

Estratégias Integradas de Simulação e Optimização de Processos Químicos

Aplicação a Problemas NLP e MINLP

Helder Gomes da Silva

Porto, Janeiro 2009



Licença Creative Commons
Atribuição-Uso-Não-Comercial-Compartilhamento
pela mesma Licença 2.5 Portugal Commons

Desde que não salvaguardado de outra forma, o conteúdo das versões impressa e electrónica desta tese encontra-se licenciado sob uma licença Creative Commons. Esta licença permite o uso não comercial deste trabalho, desde que a atribuição de autoria seja assegurada.

Para mais informações sobre esta licença, visite
o <http://www.creativecommons.pt>
o <http://creativecommons.org/licenses/by-nc-sa/2.5/pt/>

Versão electrónica disponível em:
Repositório aberto da Universidade do Porto
o <http://repositorio.up.pt/aberto/>
Repositório Científico de Acesso Aberto de Portugal
o <http://www.rcaap.pt>

Design da Capa pelo uTICM da FEUP
o <http://sicc.fe.up.pt/org/uticm/>

Helder Gomes da Silva © 2009

Estratégias Integradas de Simulação e Optimização de Processos Químicos

Aplicação a Problemas NLP e MINLP

Helder Gomes da Silva

Dissertação para a obtenção do grau de Doutor em Engenharia Química pela
Faculdade de Engenharia da Universidade do Porto



Universidade do Porto

Faculdade de Engenharia

FEUP

Departamento de
Engenharia Química

DEQ

tese realizada sob a orientação do

Professor Doutor Romualdo Luís Ribera Salcedo

Professor Catedrático da Faculdade de Engenharia da Universidade do Porto
e co-orientação do

Professor Doutor Nuno Manuel Clemente de Oliveira

Professor Auxiliar da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

*Para os meus Pais,
por tudo,
para sempre.*



Preâmbulo

Quando concluí a minha licenciatura, em 2002, a frequência da disciplina de Estratégia de Processo Químico, ainda no penúltimo ano, tinha já despertado em mim a curiosidade pela área da Engenharia de Sistemas de Processos. O interesse pelas técnicas e conhecimentos associados à modelização, codificação e implementação de rotinas para a solução de problemas de Engenharia Química motivaram-me, tendo por isso enveredado pela procura de uma oportunidade para realizar trabalhos de investigação.

Com a conclusão desta tese, que espero possa contribuir para que outros continuem, acredito que é dado mais um passo no desenvolvimento das ferramentas para simulação e optimização em que o Grupo de Investigação de Engenharia de Sistemas de Processos do DEQ tem investido nos últimos anos. Esta tese fecha um processo de aquisição de conhecimento sobre mim mesmo e o que me rodeia, algo que de que retiro real satisfação pessoal.

A condução a bom termo deste trabalho beneficiou inequivocamente dos contributos de ajuda, apoio e amizade que várias pessoas me ofereceram. A elas expresso aqui o meu agradecimento:

Começo por agradecer ao meu orientador científico, o Professor **ROMUALDO SALCEDO**, pela disponibilidade demonstrada para me acolher como seu orientando. Quero também deixar expresso o meu reconhecimento pela forma como me motivou, incentivando-me positivamente. Guardo comigo o à vontade com que discutimos os diferentes assuntos relacionados com os trabalhos em curso, agradecendo-lhe a postura objectiva e esclarecedora que tomou na partilha das suas opiniões e indicações com vista à orientação do trabalho desenvolvido. Obrigado por ter aceite dar-me uma oportunidade.

Ao coordenador científico do Grupo de Investigação de Engenharia de Processos do DEQ, o Professor **SEBASTIÃO FEYO DE AZEVEDO**, pelas conversas ao longo destes anos, pela partilha de opiniões, esclarecimentos e franqueza.

Ao meu co-orientador, Professor **NUNO OLIVEIRA**, pela disponibilidade continuamente demonstrada e pela forma cordial do seu trato. Fico igualmente grato pela chamada de atenção para pontos de vista alternativos, pela partilha de ideias e conhecimento ao nível da criação de modelos assim como da utilização de ferramentas computacionais. Agradeço-lhe ainda a sua colaboração pela avaliação da ocorrência de diferentes óptimos locais no modelo da coluna de destilação reactiva.

Quando cheguei à E219 ganhei três gurus. Agradeço por isso ao **RICARDO LIMA**, pelo privilégio das conversas com que brindou, pela suas capacidades científicas e de trabalho, pelo seus exemplos de perseverança e dedicação. À **JOANA PERES**, pelos seus conselhos, partilha de experiência, disponibilidade e simpatia. Ao **NUNO FARIA**, pelo constante incentivo ao *fazer bem* e à chamada de atenção para os pequenos pormenores que fazem a diferença. A todos eles agradeço, sinceramente, a forma como demonstraram com os seus exemplos que devemos ser cooperantes com os que temos em volta.

À **JOANA AZEREDO** agradeço por estes anos. Pela forma cuidada com que tratou todas as minhas solicitações administrativas e ainda pelo carinho demonstrado na convivência que estabelecemos no nosso dia-a-dia. Quero também deixar um cumprimento a todos aqueles que comigo partilharam a sala E219, a **CRISTINA OLIVEIRA**, a **PETIA** e o **PETER**, obrigado pelas boas condições de trabalho proporcionadas.

Agradeço também àqueles que ficam desejando-lhes boa sorte. Ao **LUÍS PAZ**, um exemplo de vida e dedicação, e ao **MORITZ**, uma encarnação altamente produtiva de boa disposição, digo obrigado por todos os momentos passados juntos. Ao **JÚLIO PAIVA** agradeço profundamente a amizade, disponibilidade, preocupação, ajuda e incentivo oferecidos à conclusão desta tese.

Às minhas primas, **PAULA MAIA** e **ISABEL MAIA**, pela ajuda indispensável ao cumprimento dos requisitos da escrita desta tese.

Agradeço a todos os amigos e colegas que me incentivaram em todos os momentos. Em particular, e de forma especial, o meu obrigado à **RITA ALVES** e à **MARIA COSTA** pela amizade sempre presente com que me têm presenteado.

Ao **ÁLVARO** e à **MARUJA**, os meus “abo” e “mauya”, agradeço por tanto terem dado à minha infância. Pelos passeios na praça, os bonecos de plasticina na varanda e a fraternidade com que me acolheram nas suas vidas. Nunca vos esquecerei.

Ao meu irmão, **BRUNO**, pela amizade, pelo constante exemplo de superação que demonstra e por todas as palmadas nas costas que não se absteve de me dar.

À minha namorada, **MARTA**, pelo seu brilho. Agradeço pelo Amor com que acolhe os meus desabafos e as minhas preocupações. Obrigado por não ter tido medo, pelas preocupações, por tudo o que decidiu dar-me e por tudo aquilo de que abdicou em favor desta tese.

Aos meus pais, **VITORINO** e **ALZIRA**, agradeço por todo o investimento que fizeram desde muito cedo, ainda como emigrantes, para dotar a minha educação de condições com qualidade. Agradeço-lhes pelos valores que me inculcaram e pelo exemplo com que, diariamente, me continuam a mostrar que com respeito e dedicação somos capazes de tornar as coisas possíveis. Sem o vosso apoio as linhas que hoje escrevo nunca teriam tingido o papel. Obrigado.

Porto, 4 de Janeiro de 2009.

The logo for the Fundação para a Ciência e a Tecnologia (FCT) consists of the letters 'FCT' in a bold, green, sans-serif font.

Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

Esta tese realizou-se sob financiamento revestido na forma de uma Bolsa de Doutoramento, com a referência SFRH/BD/10149/2002, atribuída pela Fundação para a Ciência e a Tecnologia ao abrigo do Programa de Formação Avançada de Recursos Humanos.

Resumo

Esta tese tem como objectivos o estudo, desenvolvimento e adaptação de estratégias integradas de simulação e optimização baseadas nos algoritmos de optimização estocásticos: MSGA (procura aleatória adaptativa) e MSIMPISA (recozimento simulado). A abordagem à resolução das restrições de igualdade é efectuada com dois procedimentos orientados por equações: o simulador ASCEND, que interpreta os problemas segundo modelos definidos com a sua própria linguagem declarativa orientada a objectos; e o SIMOP, uma metodologia que permite criar automaticamente simulações em código FORTRAN, recorrendo para isso à interpretação simbólica da definição matemática do problema.

O acoplamento do ASCEND com o MSGA e MSIMPISA constitui um exercício de programação mista analisado em detalhe. Nesta tese é apresentada uma estratégia de comunicação entre simulador e optimizadores, baseada em *scripting*, que visa criar uma sinergia que seja aplicável à resolução de problemas de optimização. Para sua avaliação são apresentados casos de estudo que revelam resultados satisfatórios. Os casos em estado estacionário e transiente conduziram a soluções concordantes com os melhores valores encontrados por abordagens alternativas. No entanto, esta estratégia conjugada apresenta um menor desempenho computacional no que diz respeito à despesa de tempo despendido na resolução dos problemas.

O estudo do SIMOP prende-se essencialmente com a sua expansão ao tratamento de problemas MINLP. Para isso identificam-se as modificações a efectuar no diagrama de procedimentos com vista à correcta manipulação das variáveis discretas. A validação da nova versão desta metodologia é efectuada por intermédio da avaliação de casos de estudo com solução conhecida, anteriormente resolvidos pelo MSGA e MSIMPISA pelo uso de simulações codificadas manualmente.

As duas estratégias integradas de simulação e optimização são estudadas à luz de um caso de estudo de destilação reactiva, exemplo não linear com múltiplos óptimos locais, e com o qual se procura avaliar o desempenho das abordagens e identificar potenciais aspectos passíveis de melhoria. O problema foi resolvido satisfatoriamente com as simulações ASCEND e em FORTRAN, registando-se um deterioramento dos resultados obtidos com o ASCEND para maiores dimensão de coluna. Os resultados obtidos permitem constatar que, para procedimentos comuns de inicialização, esquemas mais elaborados de decomposição por operação de particionamento e precedência ao nível dos sistemas de equações permitem encontrar melhores resultados. Esta é uma das vantagens do SIMOP, que assim permite reduzir a dimensão real do número de incógnitas a determinar simultaneamente, facilitando o uso dos algoritmos MSGA e MSIMPISA.

Identificam-se duas direcções para o desenvolvimento do trabalho apresentado: a reescrita de MSGA e MSIMPISA e a inclusão, no SIMOP, de modelização disjuntiva. Estas novas funcionalidades promoverão um melhor acompanhamento da evolução dos meios computacionais disponíveis, uma melhor integração com aplicações externas e, ainda, a possibilidade dos optimizadores serem aplicados simultaneamente ao mesmo problema de forma independente ou paralelizada.

Palavras chave: Optimização Global. Optimização Estocástica. NLP. MINLP. Modelização orientada por equações. Programação mista. ASCEND. MSGA. Procura aleatória adaptativa. MSIMPISA. Recozimento Simulado. SIMOP. Destilação Reactiva.

Abstract

The objectives of this thesis are the study, development and integration of simulation and optimisation strategies based on the stochastic optimisation algorithms: MSGA (adaptive random search) and MSIMPISA (simulated annealing). The equality constraints solution approach is carried out by two equation oriented procedures: the ASCEND simulator, which interprets the problems as models defined by its own object oriented declarative language; and SIMOP, a methodology which allows the automatic coding of FORTRAN simulations, a task performed by the symbolic interpretation of the problem's mathematical definition.

The coupling of ASCEND with MSGA and MSIMPISA is a mixed programming task, and was analysed in detail. A scripting based communication strategy between simulator and optimisers is presented, which aims to implement a synergy suitable to the solution of optimisation problems. Its applicability and validation through case studies shows satisfactory results. The steady state and dynamic case studies lead to solutions that agree with the best results obtained by alternative approaches. However, the coupled strategy presents a lower computational performance regarding the amount of expended CPU time during the solution process.

The study of SIMOP concerns essentially with its expansion to treat appropriate MINLP problems. So, in order to manipulate the discrete variables properly, the modifications to operate in the procedures diagram are identified. The validation of this new version of the methodology is accomplished by the evaluation of case studies with known solution, previously solved by MSGA and MSIMPISA with manually coded simulations.

The two integrated simulation and optimisation strategies are studied by their application to a reactive distillation case study, a non linear example subject to several local optima, which allows the performance evaluation of both approaches, as well as to identify workable improvements. The problem was satisfactorily solved with ASCEND and FORTRAN simulations, but a deterioration of the results for taller columns was verified when ASCEND was used. The results obtained allow to state that, when equivalent initialisation procedures are used, more elaborated partitioning and precedence decomposition schemes on the equations systems allow to find better results. This is one of the SIMOP's advantages, which allows to reduce the real dimension of the unknowns to be simultaneously determined, making easier for the successful application of MSGA and MSIMPISA.

Two development directions are identified for the work presented: the rewriting of MSGA and MSIMPISA and the inclusion, within SIMOP, of disjunctive modelling. This new features will promote a better use of the available computational means in constant evolution, a better integration with external application and, still, the possibility to simultaneously apply both optimisers to the same problem either independently or parallelized.

Keywords: Global Optimisation. Stochastic Optimisation. NLP. MINLP. Equation Oriented Modelling. Mixed Programming. ASCEND. MSGA. Adaptive Random Search. MSIMPISA. Simulated Annealing. SIMOP. Reactive Distillation.

Résumé

Cette thèse a pour but le développement et l'adaptation de stratégies intégrées de simulation et optimisation basées sur des algorithmes d'optimisation stochastiques: le MSGA (recherche aléatoire adaptative) et le MSIMPISA (recuit simulé). L'approche à la résolution de restrictions d'égalité est faite par deux procédés orientés par équations: le simulateur ASCEND qui interprète les problèmes selon des modèles définis avec son langage déclaratif orientée vis à vis des objets; et le SIMOP, une méthodologie qui permet de crier automatiquement des simulations en code FORTRAN en faisant appel à l'interprétation symbolique de la définition mathématique du problème.

L'accouplement de l'ASCEND avec le MSGA et le MSIMPISA constitue un exercice de programmation mixte analysé en détail. Dans cette thèse on présente un stratégie de communication entre simulateur et optimisateur, basée au scripting, qui a comme objectif la création d'une sinergie applicable à la résolution de problèmes d'optimisation. Pour son évaluation on présente aussi des études que démontrent des résultats satisfaisants. Les cas stationnaires et dynamiques on conduit à des solution accordantes aux meilleurs valeurs trouvés par des approches alternatives. Cependant cette stratégie conjuguée présente une plus basse efficace de la performance de l'ordinateur en ce que concerne la dépense de temps mis à la résolution des problèmes.

L'étude du SIMOP apporté, surtout, un élargissement au traitement des problèmes MINLP. Pour l'effet on identifie les modification à effectuer dans le diagramme des procédés orientés pour une bonne manipulation des variables discrètes. La validation de la nouvelle version de cette méthodologie est faite par une évaluation des cas d'étude avec des solutions déjà validées par l'MSGA et l'MSIMPISA, para l'utilisation de simulation codifiées manuellement.

Ces deux stratégies intégrées de simulation et optimisation sont étudiés selon une méthode de distillation réactive, exemple non linéaire à des plusieurs endroits avec lequel on cherche à évaluer a performance des approches et identifier des potentiels aspectas passibles d'améliorement. Le problème a été résolu d'une façon satisfatoire avec les simulation ASCEND et en FORTRAN, tout en se vérifiant une chute des résultats obtenus avec l'ASCEND pour les plus grandes dimensions de la colonne. Les résultats obtenus permettent aussi de constate que pur des procédés communs d'initiation, des schémas plus élaborés de décomposition par une opération de division et de la préséance au niveau de systèmes d'équation permettent obtenir meilleur résultats. Une des avantages du SIMOP qui permet de réduire la réelle dimension du nombre d'inconnues a déterminer en même temps, rendant plus facile l'usage des algorithmes MSGA et MSIMPISA.

On identifie deux direction pour le développement du travail présenté: la récite de l'MSGA et de l'MSIMPISA et l'inclusion au SIMOP, de la modélisation disjonctive. Ces nouvelles fonctions promouveront une meilleur e suite de l'évolution des moyeus computationnels disponibles, une plus bonne intégration avec des applications externes et, encore, la possibilité d'appliquer en même temps les optimisateurs au même problème soit d'une façon indépendante soit parallèle.

Mots-Clés: Optimisation Globale. Optimisation Stochastique. NLP. MINLP. Modélisation Orienté par Équation. Programmation mixte. ASCEND. MSGA. Recherche Aléatoire Adaptative. MSIMPISA. Recuit Simulé. SIMOP. Distillation Réactive.

Índice

Preâmbulo	vii
Resumo	xi
Abstract	xiii
Résumé	xv
Lista de Figuras	xxi
Lista de Tabelas	xxv
Lista de Quadros	xxix
1 Introdução	1
1.1 Motivação	1
1.2 Objectivos e Metodologia	2
1.3 Estrutura da Tese	3
2 Optimização	5
2.1 Contexto Histórico	5
2.2 Conceitos fundamentais	8
2.2.1 Modelos e âmbito de aplicação	8
2.2.2 Formulação e Contexto dos Problemas de Optimização	9
2.2.3 Tipologia dos Problemas de Optimização	13
2.2.4 Plausibilidade, Convexidade e Optimalidade	15

2.3	Métodos de Optimização	20
2.3.1	Optimização de problemas NLP	20
2.3.2	Optimização de problemas MINLP	22
2.3.3	Optimização Global	26
	Métodos determinísticos em GO	28
	Métodos estocásticos em GO	30
	Métodos de procura aleatória	31
	Recozimento Simulado	31
	Procura tabu	33
	Algoritmos evolucionários	33
2.4	Simulação de Processos Químicos	34
2.5	Perspectivas	36
2.5.1	Optimização baseada em agentes	37
	Nomenclatura	41
3	O ambiente ASCEND	45
3.1	Nota Histórica	45
3.2	Constituição do ASCEND e suas principais características	46
3.3	Exemplo de aplicação	49
3.4	Codificação e análise do modelo para ASCEND	52
3.4.1	Definição do modelo base	52
3.4.2	Iniciação à interface gráfica	53
3.4.3	Inclusão de métodos	60
3.4.4	A sequenciação automática da resolução de problemas	63
3.4.5	Abordagem preliminar ao tratamento de resultados em ASCEND	64
3.5	Conclusões	66
	Nomenclatura	67
4	(MSGA.ou.MSIMPSA).e.ASCEND	69
4.1	As linguagens de scripting	70
4.1.1	O caso particular da linguagem Tcl/Tk	72
4.2	A conjugação Simulação/Otimização	73
4.2.1	A estratégia de comunicação	73

Etapa I	74
Etapa II	79
4.2.2 Avaliação do desempenho das estratégias de comunicação	82
4.2.3 Aplicações em estado transiente	87
4.3 Estratégia de Comunicação: Maior usabilidade	91
4.4 Conclusões	95
Nomenclatura	96
5 Aplicação de estratégias integradas a problemas NLP/MINLP	99
5.1 SIMOP - SIMulation for OPTimization	100
5.2 SIMOP para formulações MINLP	105
5.2.1 Caso de estudo P.PRSIB15	108
5.2.2 Caso de estudo P.PRSIB19	112
5.3 Caso de estudo: Destilação Reactiva, P.CDR	117
5.3.1 Descrição e estudo preliminar do modelo	117
Detalhes sobre a inicialização	125
5.3.2 Resultados de Optimização	127
Formulação NLP Ideal	127
Formulação NLP não ideal	133
Formulação MINLP	134
5.4 O comportamento ASCEND vs. SIMOP	148
5.5 Conclusões	151
Nomenclatura	152
6 Conclusões e Perspectivas	157
6.1 Perspectivas e Trabalho Futuro	160
Referências	163
Bibliografia	179
Apêndices	181
A Casos de estudo	183
A.1 P.RSIA01	185

A.2	P.HRCV	188
A.3	P.ABSTRIP	191
A.4	P.BATCHT	208
A.5	P.FBE	212
A.6	P.PRSIB04	214
A.7	P.PRSIB08	216
A.8	P.CDR	218
B	Sobre o ASCEND	239
B.1	Ficheiro .ascendrc	240
B.2	Navegador de variáveis internas	242
C	Estratégia integrada	243
C.1	Ficheiro optprocedures.a4s	245
D	Noções complementares	251
D.1	Programação matemática de expressões lógicas	251

Lista de Figuras

2.1	Diferentes tipos de mínimos de uma função genérica.	16
2.2	Representação de uma função convexa a duas dimensões.	17
2.3	Exemplo de função não convexa, caso 1.	18
2.4	Exemplo de função não convexa, caso 2.	19
2.5	Exemplo de função não convexa, caso 3.	19
2.6	Classificação dos métodos de optimização global.	29
2.7	Sistema de optimização sequencial (adaptado de Sirola e Hauan (2007)).	38
2.8	Sistema de optimização baseado em agentes (adaptado de Sirola e Hauan (2007)).	39
2.9	Sistema de optimização polimórfica baseada na acção de multiagentes (adaptado de Sirola e Hauan (2007)).	40
3.1	Diagrama Processual do Problema P.MK.	50
3.2	Janela LIBRARY . Problema P.MK. Painéis informativos durante a importação do ficheiro <code>make_up.a4c</code> e criação da simulação <code>mk</code>	54
	(a) Lista dos ficheiros carregados.	54
	(b) Criação da simulação <code>mk</code> . O cadeado na janela de indica que a mesma requer acção imediata por parte do utilizador.	54
	(c) Lista dos ficheiros carregados e simulações compiladas.	54
3.3	Aspecto da janela BROWSER após exportação da simulação <code>mk</code> a partir da janela LIBRARY . As variáveis apresentam os valores que o ambiente lhes atribui por omissão enquanto que as constantes permanecem indefinidas. O painel inferior possibilita a filtragem imediata de determinados tipos de elementos: variáveis reais (RV) e discretas (DV); relações reais (RR) e discretas (LR); constantes reais (RC) e discretas (DC).	55

3.4	Janela BROWSER . Problema P.MK. Atribuição de valor numérico ao quociente E_1/R , 9000K.	56
3.5	Aspecto geral da janela <i>Units</i> , onde o utilizador pode escolher as unidades de trabalho para os diversos tipos de variáveis.	56
3.6	Janela <i>Browser</i> - Apresentação dos atributos da variável X_{AP}	58
3.7	Janela BROWSER . Problema P.MK. Painéis informativos mostram os valores das constantes do modelo bem como das estimativas iniciais das variáveis.	58
3.8	Janela SOLVER . Problema P.MK. Estatísticas sobre a estrutura do problema antes e após especificação das variáveis que preenchem os graus de liberdade.	59
3.9	Ambiente ASCEND. Menu SELECT METHOD	63
3.10	Janela SCRIPT . Automatização de tarefas para o modelo <i>make_up</i>	64
4.1	Diagrama generalizado da aplicação conjunto de um simulador e de um optimizador estocástico (adaptado de Lima (2005)).	74
4.2	Esquema de comunicação. Etapa I.	75
4.3	Esquema de comunicação II.	81
4.4	Exemplo P.ABSTRIP. Diagrama Processual.	82
4.5	Problema P.ABSTRIP. Matriz de incidências do problemas após decomposição e partição operados pelo ASCEND. Neste caso, pode verificar-se que no bloco 41, a variável de saída q é calculada a partir da Equação f37.	83
4.6	Problema P.ABSTRIP. Distribuições de erro relativo para as corridas efectuadas.	86
	(a) Caso I.	86
	(b) Caso II.	86
4.7	Problema P.BATCHT. Resultados obtidos.	90
	(a) Perfil óptimo de temperatura.	90
	(b) Distribuição de erro relativo.	90
4.8	Problema P.FBE. Resultados obtidos.	91
	(a) Perfil de alimentação óptimo.	91
	(b) Distribuição de erro relativo.	91
4.9	Esquema de comunicação II.	92
4.10	Estrutura geral do script de ligação. Sistematização.	94
5.1	Diagrama conceptual do SIMOP. Adaptado de Lima (2005).	101

5.2	Problema P.PRSIB15. Diagrama Processual.	109
5.3	Problema P.PRSIB15. Distribuições cumulativas do erro relativo associado às corridas de optimização efectuadas com MSGA e MSIMPISA.	111
5.4	Problema P.PRSIB15. Configuração determinada com vista à minimização dos custos de produção.	111
5.5	Problema P.PRSIB19. Estrutura de possibilidades para o esquema de separação.	113
5.6	Problema P.PRSIB19. Distribuição de erro relativo.	116
5.7	Problema P.PRSIB19. Melhor configuração encontrada.	116
5.8	Problema P.CDR. Diagrama da coluna de destilação reactiva.	119
5.9	Problema P.CDR. Modelo GAMS, ELV ideal. Soluções locais identificadas com OQNLP. Coluna com 10 andares de equilíbrio.	124
5.10	Problema P.CDR. Modelo GAMS, ELV ideal. Soluções locais identificadas com OQNLP. Coluna com 20 andares de equilíbrio.	124
5.11	Problema P.CDR. Caso NLP. ELV Ideal. Comparação dos melhores valores da função objectivo utilizando ASCEND, SIMOP e GAMS.	128
5.12	Caso NLP. ELV Ideal. Distribuição de erro relativa dos resultados óptimos obtidos com ASCEND associado a MSGA e MSIMPISA para modelos com 3, 7, 10 e 15 andares.	129
	(a) Resultados para o MSGA.	129
	(b) Resultados para o MSIMPISA.	129
5.13	Resultados SIMOP. Problema P.CDR. Caso MINLP. ELV ideal. Distribuições cumulativas do erro relativo verificado nos resultados encontrados, com domínios de procura diferenciados em termos do excesso estequiométrico, utilizando a abordagem SIMOP.	146
	(a) 10%	146
	(b) 20%	146
	(c) 30%	146
	(d) 40%	146
	(e) 50%	146
5.14	Resultados SIMOP. Problema P.CDR. Caso MINLP. ELV não ideal. Distribuições cumulativas do erro relativo verificado nos resultados encontrados, com domínios de procura diferenciados em termos do excesso estequiométrico, utilizando a abordagem SIMOP.	147

(a)	10%	147
(b)	20%	147
(c)	30%	147
(d)	40%	147
(e)	50%	147
5.15	ELV ideal. Matrizes de incidência dos conjuntos de saída obtidos com ASCEND e SIMOP para uma coluna com quinze andares.	149
5.16	ELV não ideal. Matrizes de incidência dos conjuntos de saída obtidos com ASCEND e SIMOP para uma coluna de dez andares.	150
A.1	Diagrama processual do problema P.HRCV.	188
A.2	Problema P.HRCV. Resultados para o estudo F_{obj} vs T .	190
A.3	Diagrama processual do problema P.ABSTRIP	191
A.4	Problema P.ABSTRIP. Caso I. Trajecto de procura do MSIMPISA para o melhor valor encontrado.	204
A.5	Problema P.BATCMT. Resultados encontrados com MSGA.	211
(a)	Resultados parciais da procura do melhor resultado.	211
(b)	Perfil óptimo de temperatura.	211
A.6	Problema P.PRSIB04. Distribuições de erro relativo.	215
A.7	Resultados GAMS. Perfis de temperatura e composições de Água e Etilenoglicol ao longo coluna de custo mínimo especificada.	235
(a)	Perfil de Temperaturas	235
(b)	Perfil de Composições para $x_{2,k}$ e $x_{3,k}$	235
A.8	Resultados ASCEND. Perfis de temperatura e composições de Água e Etilenoglicol ao longo coluna de custo mínimo especificada.	236
(a)	Perfil de Temperaturas	236
(b)	Perfil de Composições para $x_{2,k}$ e $x_{3,k}$	236
A.9	Resultados SIMOP. Perfis de temperatura e composições de Água e Etilenoglicol ao longo coluna de custo mínimo especificada.	237
(a)	Perfil de Temperaturas	237
(b)	Perfil de Composições para $x_{2,k}$ e $x_{3,k}$	237
B.1	ASCEND. Janela das variáveis internas.	242

Lista de Tabelas

3.1	Problema P.MK. Parâmetros cinéticos.	50
3.2	Dados do problema P.MK: Valores e nomenclatura. Representações nos contextos da tese e do modelo ASCEND.	51
3.3	Variáveis do Problema P.MK. Nomenclatura e suas representações nos contextos da tese e do modelo ASCEND.	52
4.1	Problema P.ABSTRIP. Comparação, entre abordagens baseadas na codificação em FORTRAN e o acoplamento do MSGA e MSIMPISA com o ASCEND, dos desempenhos ao nível de tempo computacional e do melhor valor obtido para a função objectivo.	85
4.2	Problema P.ABSTRIP. Melhores desempenhos encontrados.	86
4.3	Problemas P.BATCHT. Melhores resultados obtidos.	90
4.4	Problemas P.FBE. Melhores resultados obtidos.	91
5.1	SIMOP. Correspondência entre as diferentes componentes do problema e os ficheiros de texto utilizados para sua representação.	104
5.2	Problema P.PRSIB15. Parâmetros de custo.	110
5.3	Problema P.PRSIB15. Melhores resultados encontrados com MSIMPISA e MSGA.	110
5.4	Problema P.PRSIB19. Conjunto de variáveis decisão para o melhor valor encontrado. Resultado coincidente para MSIMPISA e MSGA para uma tolerância de 10^{-3}	115
5.5	Problema P.CDR. Parâmetros do modelo (Ciric e Gu, 1994).	122

5.6	Problema P.CDR. Caso NLP. ELV Ideal. Melhores resultados obtidos em função de N com a estratégia conjugada utilizando ASCEND e ainda com o GAMS ($F_{obj}, 10^7 \$/ano$).	127
5.7	Resultados ASCEND. Problema P.CDR. Caso NLP. ELV Ideal. Especificações da coluna de custo mínimo, cujo projecto corresponde a uma coluna com sete andares e uma $F_{obj} = 1.534 \cdot 10^7 \$/ano$	128
5.8	Problema P.CDR. Caso NLP. Melhores resultados encontrados para a abordagem com GAMS ($F_{obj}, 10^7 \$/ano$).	132
5.9	Resultados GAMS. Problema P.CDR. Caso NLP. ELV Ideal. Especificações da coluna de custo mínimo encontrada para o modelo com sete andares, à qual corresponde uma $F_{obj} = 1.523 \cdot 10^7 \$/ano$	132
5.10	Caso NLP. ELV não ideal. Melhores resultados obtidos ($F_{obj}, 10^7 \$/ano$).	133
5.11	Resultados ASCEND. Caso MINLP. Melhores resultados obtidos ($F_{obj}, \cdot 10^7 \$/ano$) para modelos Ideal e não ideal pela aplicação ASCEND conjugada com MSGA e MSIMPISA. Os resultados obtidos por Cardoso et al. (2000) são apresentados como referência comparativa. Os respectivos números de andar por coluna aparecem entre parêntesis.	140
5.12	Resultados ASCEND. Problema P.CDR. Caso MINLP. ELV Ideal. Especificações da coluna de custo mínimo, cujo projecto corresponde a uma coluna com seis andares e uma $F_{obj} = 1.569 \cdot 10^7 \$/ano$	141
5.13	Problema P.CDR. Caso MINLP. ELV ideal. Abordagem SIMOP. Melhores resultados encontrados para o valor da função objectivo ($F_{obj}, 10^7 \$/ano$) em corridas de optimização em intervalos de procura diferenciados em função do excesso estequiométrico.	143
5.14	Resultados SIMOP. Problema P.CDR. Caso MINLP. ELV Ideal. Especificações da coluna de custo mínimo, cujo projecto corresponde a uma coluna com oito andares e uma $F_{obj} = 1.502 \cdot 10^7 \$/ano$	143
5.15	Problema P.CDR. Caso MINLP. ELV não ideal Abordagem SIMOP. Melhores resultados encontrados para o valor da função objectivo ($F_{obj}, 10^7 \$/ano$) em corridas de optimização em intervalos de procura diferenciados em função do excesso estequiométrico.	145

5.16	Resultados SIMOP. Problema P.CDR. Caso MINLP. ELV não ideal. Especificações da coluna de custo mínimo, cujo projecto corresponde a uma coluna com sete andares e uma $F_{obj} = 1.583 \cdot 10^7$ \$/ano.	145
5.17	ELV Ideal. Comparação dos resultados do passo de partição aquando da utilização do ASCEND ou SIMOP. Os parâmetros apresentados são o número total de equações no sistema (neq), o número total de equações no subsistema (neq_s) e o número total de variáveis de reciclagem no subsistema (neq_t).	149
5.18	ELV não ideal. Comparação dos resultados do passo de partição usando ASCEND e SIMOP. Os parâmetros apresentados são o número total de equações no sistema (neq), o número total de equações no subsistema (neq_s) e o número de variáveis de reciclagem no subsistema (neq_t).	150
A.1	Problema P.ABSTRIP. Melhores resultados encontrados, Casos I e II.	203
A.3	Resultados GAMS. Especificações da coluna de custo mínimo encontrada para um problema NLP com $N = 10$	233
A.4	Resultados GAMS. Especificações da coluna de custo mínimo encontrada para um problema NLP com $N = 20$	234
A.5	Resultados GAMS. Especificações da coluna de custo mínimo encontrada para um problema NLP com $N = 7$	235
A.6	Resultados ASCEND. Especificações da coluna de custo mínimo encontrada para um problema NLP com $N = 7$	236
A.7	Resultados SIMOP. Especificações da coluna de custo mínimo encontrada para um problema MINLP.	237

Lista de Quadros

3.1	Problema P.MK. Modelo básico para ASCEND.	53
3.2	Problema P.MK. Modelo <i>make_up</i> . Resultados para $X_{AP} = 0.5$	60
3.3	Estrutura do modelo <i>make_up</i> após a inclusão de métodos.	62
3.4	Problema P.MK. <i>Script</i> para estudo de várias simulações variando X_{AP}	65
3.5	Código do ficheiro de resultados <i>outfile.dat</i>	65
3.6	Problema P.MK. Criação de um ficheiro para folha de cálculo.	66
4.1	Problema P.RSIA01. Modelo para ASCEND.	77
4.2	Problema P.RSIA01. Subrotina <i>funk</i>	77
4.3	Problema P.RSIA01. Ficheiro <i>main.opt</i>	78
4.4	Problema P.RSIA01. Ficheiro <i>opt.script</i>	79
4.5	Problema P.BATCHT. Excerto do modelo ASCEND.	88
5.1	Problema P.MK. Ficheiro <i>matrix.dat</i>	103
5.2	Problema P.MK. Ficheiro <i>matrix.res</i>	103
5.3	Problema P.MK. Ficheiro <i>equations.dat</i>	103
5.4	Problema P.PRSIB15. Formulação SIMOP.	111
5.5	Problema P.PRSIB19. Ficheiro <i>vp_fix.dat</i>	114
5.6	Problema P.PRSIB19. Ficheiro de dados de configuração das variáveis no MSIMPISA, <i>func.dat</i>	115
5.7	Problema P.PRSIB19. Ficheiro de configuração do MSGA, <i>optim.dat</i>	115
5.8	Problema P.CDR. Caso MINLP. Forma generalizada das modificações operadas no procedimento RUNSIM.	136
5.9	Problema P.CDR. Caso MINLP. Modificações operadas no modelo ASCEND e no <i>opt.script</i> correspondente.	137
A.1	Biblioteca <i>myatoms.a41</i>	203

Capítulo 1

Introdução

Neste capítulo serão abordados o contexto motivacional, os objectivos e a estrutura da tese apresentada.

1.1 Motivação

A simulação e optimização de processos industriais constitui temática onde se regista um investimento científico considerável. Ultimamente tem-se assistido a uma proliferação de aplicações que utilizam algoritmos estocásticos, isoladamente ou em combinação com algoritmos determinísticos, numa tentativa de resolução de problemas que, apresentando escala variável, aparecem em contextos diversificados.

No contexto industrial, os simuladores sequenciais modulares apresentam uma muito superior taxa de adopção face às abordagens orientadas por equações. Apesar da simplicidade apresentada pelos primeiros no que diz respeito ao modo de trabalho, os segundos providenciam ao utilizador, normalmente, uma melhor capacidade de controlo no que diz respeito aos contornos e configuração da metodologia de resolução do problema.

No âmbito da Engenharia Química é comum encontrar problemas que envolvem equações não lineares. No caso dos simuladores modulares, são diversos os exemplos que apresentam rotinas de optimização capazes de lidar com o referido tipo de problemas, casos do Aspen Plus¹, do Aspen Hysys (anteriormente HYSIM e HYSYS) e PRO/II². No entanto, muitos processos resultam numa formulação em que variáveis de decisão contínuas e discretas coexistem durante a optimização, como são exemplo as situações de formulação de topologias para diagramas processuais. Geralmente, estes problemas

¹Software comercial da ASPENTECH, <http://www.aspentech.com>

²Software comercial da Invensys Process Systems, <http://www.ips.invensys.com>

de natureza mista ao nível das variáveis envolvidas apresentam uma taxa de eficácia de resolução mais baixa, exigindo a aplicação de métodos que consigam fazer uma exploração apropriada do seu espaço combinatorial.

Desde 1998 que o grupo de investigação em Engenharia de Sistemas de Processos tem investido esforços na concepção de uma estratégia integrada de optimização estocástica com base em simulação orientada por equações. A moldura que enquadra esta linha de trabalho pressupõe conhecimento ao nível das ferramentas previamente desenvolvidas bem como experiência ao nível do trabalho com simuladores orientados por equações.

O trabalho desenvolvido nesta tese foi motivado essencialmente por duas razões: (a) experimentação dos optimizadores estocásticos desenvolvidos no grupo com ambientes de simulação que não os habitualmente utilizados; (b) alargamento da estratégia integrada anteriormente desenvolvida (SIMOP) a problemas onde estejam presentes variáveis discretas, situações que aparecem recorrentemente no contexto da Engenharia Química.

1.2 Objectivos e Metodologia

O objectivo principal da presente tese é a continuação do desenvolvimento da estratégia integrada de simulação-optimização de processos químicos. Designada por SIMOP, tal metodologia integra a linha de trabalho desenvolvida pelo grupo de investigação desde 1990, iniciada com o desenvolvimento dos optimizadores estocásticos MSGA e MSIMPISA. Fazem parte integrante desta vontade duas tarefas:

1. Acoplamento dos optimizadores MSGA e MSIMPISA ao simulador ASCEND IV. Pretende-se conseguir uma adaptação que torne os optimizadores, desenvolvidos em FORTRAN, utilizáveis pelo simulador ASCEND IV por intermédio do estabelecimento de uma interface transparente para o utilizador. Procura-se com esta ferramenta estudar diversos problemas de carácter simples que sirvam de base à sua certificação para, posteriormente, aplicá-la ao caso de estudo da síntese de colunas de destilação reactiva. Nestes casos uma formulação MINLP é a mais adequada, devido à localização variável das alimentações assim como a variabilidade no número de andares da coluna.
2. Desenvolvimento, no âmbito da estratégia de simulação-optimização integrada do grupo, de uma extensão do SIMOP para aplicação a problemas MINLP. Aproveitando as capacidades combinatoriais dos algoritmos MSGA e MSIMPISA, o

objectivo passa por dotar o SIMOP de meios para a geração automática de simulações que se enquadrem na moldura da formulação genérica MINLP, com posterior aplicação à análise para o projecto de colunas de destilação reactiva.

1.3 Estrutura da Tese

Para além do presente capítulo introdutório, os conteúdos desta tese encontram-se divididos em mais cinco capítulos e quatro apêndices.

Capítulo 2 Apresentação das bases da teoria da optimização, classificação de problemas e metodologias. Tem como objectivo fornecer uma descrição sucinta dos principais algoritmos existentes, identificando exemplos de aplicação e enquadrando o trabalho desenvolvido na presente tese. É resultado da revisão bibliográfica inerente ao presente trabalho.

Capítulo 3 Apresentação do ambiente de simulação ASCEND. Identificam-se os seus componentes e funcionalidades, assim como a forma como a aplicação é utilizada na resolução de problemas.

Capítulo 4 Apresentação da estratégia integrada de simulação e optimização com base na utilização acoplada dos optimizadores MSGA e MSIMPISA com o simulador ASCEND. Inclui-se a análise efectuada sobre as formas de estabelecimento da comunicação entre as aplicações, assim como os resultados obtidos com base em casos de estudo com vista à validação da estratégia.

Capítulo 5 Estudo da estratégia integrada de simulação e optimização SIMOP, incluindo a exposição das modificações implementadas com base à sua expansão a formulações MINLP. Avalia-se o desempenho das diferentes metodologias com base na aplicação a um caso de estudo de destilação reactiva, procurando identificar possíveis evoluções a serem implementadas.

Capítulo 6 Apresentam-se as conclusões resultantes do trabalho desenvolvido e do processo integral de aprendizagem que constitui este doutoramento. Discutem-se as perspectivas de evolução futura do trabalho desenvolvido ao nível da metodologia SIMOP e da aplicabilidade dos algoritmos estocásticos MSGA e MSIMPISA. Apresentam-se também algumas pistas para trabalhos futuros.

Apêndice A Incluem-se dados e informações complementares relativos aos casos de estudo apresentados no corpo da tese.

Apêndice B Apresentam-se alguns dos recursos relacionados com a execução do simulador ASCEND em *developer mode*.

Apêndice C Incluem-se a título ilustrativo algumas listagens de código relacionadas com a implementação da estratégia de comunicação entre o ASCEND e os optimizadores estocásticos.

Apêndice D Apresentam-se referências curtas a noções complementares referenciadas no corpo da tese, embora sem relação directa com o trabalho desenvolvido.

Capítulo 2

Optimização

Sumário do Capítulo

O presente capítulo pretende apresentar as noções gerais relacionadas com o enquadramento e importância da teoria da optimização, assim como os conceitos que estão na base da classificação dos tipos de problemas e metodologias de aproximação a soluções satisfatórias. Pretende-se providenciar uma descrição sucinta dos principais algoritmos utilizados e, também, citar exemplos no âmbito da Engenharia Química que, ilustrando situações de passível aplicação, ajudem a contextualizar o trabalho desenvolvido na área de estudo que é a optimização.

2.1 Contexto Histórico

A curiosidade intrínseca do ser humano é parte fundamental da génese da teoria da optimização. É esse espírito de procura, transversal à Humanidade, que potencia o desenvolvimento de esforços que visam a concretização de objectivos. Esses objectivos correspondem a um diversificado leque de situações, como sejam o alcançar de metas pré-estabelecidas, a produção de bens ou a obtenção de bem-estar. Esta necessidade de concretização faz com que, depois de implementada uma ideia, o passo seguinte seja tentar perceber qual a forma de a fazer progredir, qual o método para a tornar melhor — mais rápida, mais robusta — e, por consequência, mais eficiente. Este trajecto pode ser visto como uma procura de um estado de *perfeição*, que no entanto, é sempre um estado de *compromisso*.

A premissa primordial desta procura é a tomada de decisão. Do ponto de vista etimológico, a palavra *ótimo* representa precisamente essa ideia, pois tem origem no

latim, *optĭmus*, substantivo derivado do verbo *optāre* que por sua vez significa: escolher, preferir ou decidir. Esta relação permite entender *optimização*, do ponto de vista da nomenclatura, como o vocábulo que traduz a seguinte ideia: aplicação de uma metodologia que, procurando efectuar boas escolhas, permita encontrar um novo estado que apresente melhorias relativamente ao estado anterior.

Tavares e Correia (1999) referem que o Confucionismo poderá ser a primeira evidência da importância da optimização, tendo em consideração a relevância que a linha de pensamento do filósofo Chinês atribuía às preocupações que rodeiam o acto de decidir.

Partindo deste ponto os episódios sucedem-se ao longo da linha do tempo. O poeta Virgílio relata as desventuras da Rainha Dido, aquando da fundação da cidade de Cartago, então confrontada com o problema da geometria isoperimétrica. A monarca pretendia encontrar a figura geométrica que, servindo de base à construção de uma cidade circunscrita a um perímetro estabelecido, providenciasse o máximo aproveitamento da área habitável. Com o auxílio de uma corda esticada entre dois pontos a rainha acabou por desenhar um meio círculo, gesto que a conduziu finalmente até à forma circular (Wilde e Beightler, 1967). Graças à contribuição de Steiner (1841), o problema isoperimétrico passou a ter uma resposta exacta nos meados do século XIX (Bezdek, 2007). No entanto é curioso pensar que este resultado já seria empiricamente conhecido anteriormente. Muitas das cidades da antiguidade apresentavam muralhas circulares, facto que lhes permitia minimizar o comprimento de parede a construir para cercar uma povoação.

O problema isoperimétrico é apenas um exemplo. No período compreendido entre os dois momentos citados na evolução deste desafio, desencadearam-se outros desenvolvimentos. A teoria da optimização recebe as suas primeiras influências formais por intermédio do optimismo filosófico promovido por Leibniz¹. Este filósofo e matemático alemão acreditava que o mundo em que vivemos é o melhor de todos os mundos, estado resultante de um gesto criacionista dotado de perfeição. Leibniz procurou encontrar essa verdade por intermédio da matemática.

Os ventos optimistas trazidos por esta perspectiva condicionaram o contexto científico. As leis da mecânica são pela primeira vez estipuladas considerando princípios mínimos. Como exemplos práticos da aplicação desta corrente de pensamento podem citar-se os estudos iniciados por Galileu e concluídos por Johann Bernoulli, no sentido de projectar um escorrega que minimizasse o tempo de viagem de um objecto despro-

¹As ideias optimistas de Leibniz ficaram registadas numa das suas principais obras: *Essais de Théodicée sur la bonté de Dieu, la liberté de l'homme et l'origine du mal* - Ensaios de Teodicéia sobre a bondade de Deus, a liberdade do Homem e a Origem do Mal.

vido de atrito; são igualmente referência os trabalhos de Gauss no desenvolvimento do procedimento da minimização do quadrado dos resíduos para ajuste de dados; a uniformização das leis da óptica e mecânica estipulada por Hamilton e a enunciação do princípio da energia mínima por parte de Gibbs. Euler, filósofo e matemático igualmente partidário das ideias optimistas de Leibniz, notabiliza-se pelos trabalhos no âmbito do estudo do cálculo das variações, como também participa na linha do desenvolvimento das teorias do potencial cinético também estudadas por Maupertius e Lagrange (Rao, 1996; Wilde e Beightler, 1967).

Os ataques que Voltaire dirigiu à perspectiva optimista de Leibniz com a publicação de *Cândido, ou O Optimismo*, e a falta de meios que viabilizassem a concretização das ideias que tentavam responder às questões levantadas pela teoria da optimização, conduziram ao estagnamento da mesma. Tal hibernação é apenas perturbada no final da década de quarenta do século XX. Este momento marca o início do período dos desenvolvimentos mais assinaláveis na área da teoria da optimização. Nele se assiste ao aparecimento de um vasto leque de literatura em que se abordam metodologias e princípios, nomeadamente: apresentação do método simplex por Dantzig; estabelecimento do princípio da optimalidade de Bellman para problemas acíclicos; a definição das condições de mínimo e máximo por Kuhn e Tucker; a aplicação do método de Monte Carlo, entre outros. Estes resultados são impulsionados no clima de aparecimento do ENIAC e promovidos pelo subsequente desenvolvimento das ferramentas computacionais (Metropolis, 1987; Rao, 1996; Wilde e Beightler, 1967).

Nemhauser apelidou este período de “Segundo Renascimento”, atendendo à estimulação que as ciências organizacionais e do planeamento, a gestão, a economia e a engenharia promoveram no sentido de tornar o estudo das técnicas de optimização uma matéria indispensável à Investigação Operacional. Desde então que a capacidade de utilizar técnicas numéricas para determinar satisfatoriamente as soluções óptimas demonstrou ter um papel basilar à concretização de muitas das conquistas recentes nas áreas científica, económica e de engenharia. No âmbito da Engenharia de Sistemas de Processos, nomeadamente nas suas aplicações no contexto dos problemas de Engenharia Química, a área da optimização desenvolveu-se a partir de interesses académicos. Apesar da introdução de ideias mais complexas nos algoritmos que lhes servem de base, os métodos de optimização ganharam em eficiência e apresentam-se como ferramentas que oferecem garantias, encontrando-se hoje num patamar de desenvolvimento em que a teoria da optimização se assume como metodologia de importância inegável nos processos de tomada de decisão do mundo real (Biegler e Grossmann, 2004; Gill et al., 1981; Horst et al., 1995; Ponton, 1995; Rao, 1996; Tavares e Correia, 1999;

Wilde e Beightler, 1967).

A teoria da optimizaç o, que engloba o estudo quantitativo dos  ptimos e dos m todos que permitem encontr -los, n o   mais do que a materializaç o da procura do Homem pela *perfeiç o*. A sua aplicabilidade implica:

1. Conhecer a interacç o entre as vari veis de uma forma exacta e quantitativa;
2. Encontrar uma forma que permita medir o desempenho do sistema;
3. Encontrar as condiç es que conduzam a um melhor desempenho.

Este caminho de melhoria cont nua   hoje tamb m a base conceptual da estrat gia que se procura adoptar na gest o dos procedimentos laborais e administrativos nas empresas. Nesse contexto, tem-se assistido   implementaç o de normas estabelecidas e reconhecidas internacionalmente em  reas como a gest o da qualidade, a implementaç o de pr ticas de higiene e seguranç  no local de trabalho ou ainda o *lean management*.

2.2 Conceitos fundamentais

2.2.1 Modelos e  mbito de aplicaç o

A aplicabilidade de algoritmos de optimizaç o   uma am lgama, nela se fundem a implementaç o de t cnicas que visam encontrar os melhores  ndices de desempenho com a capacidade de construir modelos. A tarefa de conceber representaç es adequadas dos fen menos reais implica o conhecimento de conceitos matem ticos, bem como das pr prias leis que regem a realidade em estudo (Tavares e Correia, 1999).

Considerando um n vel de abstracç o mais elevado, podem ser utilizadas tr s categorias para a classificaç o de modelos (Thierauf e Klekamp, 1975): ic nicos (representaç es f sicas   escala), an logos (baseados em diagramas que representam relaç es) e simb licos (tamb m chamados de matem ticos). Os  ltimos s o os que interessam na aplicaç o da teoria da optimizaç o bem como em estudos do  mbito da investigaç o operacional, podendo ser ordenados com base em cinco crit rios dicot micos associ veis (Hoover e Perry, 1989): prescritivos ou descritivos; discretos ou cont nuos; probabil sticos ou determin sticos; est ticos ou din micos; e ainda de ciclo aberto ou fechado.

Particularizando para a utilizaç o de modelos no  mbito de Engenharia Qu mica, pode considerar-se ainda outra distinç o na natureza dos modelos utilizados (Edgar et al., 2001; Floudas, 1995), existindo dois tipos distintos: modelos com base em abordagens fundamentais e modelos emp ricos.

Começando pelos modelos empíricos, pode dizer-se que, essencialmente, estes são baseados em dados. Comuns em aplicações de engenharia do conhecimento, são também designados por modelos de caixa-negra.

Os modelos de abordagem fundamental assentam em princípios físico-químicos fundamentais. As equações do modelo matemático definem as leis que regem as interações dos elementos do sistema real. Quando os parâmetros disponíveis se encontrem desprovidos de incerteza, estes modelos poderão também ser designados como modelos determinísticos.

Refere-se ainda a possibilidade de existência de modelos híbridos. Nesse tipo de representação coexistem dados e equações que traduzem leis físico-químicas. Este tipo de modelos tem aplicabilidade reconhecida, por exemplo, na área de simulação de bioprocessos (Banga et al., 2003b).

2.2.2 Formulação e Contexto dos Problemas de Optimização

Tendo definido a optimização como um exercício de escolha com vista à decisão, matematicamente a tarefa de otimizar traduzirá a procura do valor de um conjunto de incógnitas, \mathbf{x} , que permitirão maximizar ou minimizar a função utilizada para exprimir o objectivo. Tavares e Correia (1999) frisam que o resultado de tal exercício deverá ser alcançado respeitando integralmente as restrições que condicionam a procura das variáveis, ao passo que Banga et al. (2003b) colocam a tónica no carácter sistemático e eficiente que esse processo deve evidenciar, descartando da definição de optimização quaisquer abordagens empíricas baseadas em tentativa e erro. Pretende-se assim encontrar a solução mais vantajosa que não comprometa a eficiência dos processos avaliados, daí que se trate, como referido na Secção 2.1, de um compromisso.

Atendendo ao enunciado apresentado, poderá definir-se a seguinte formulação algébrica para um problema de optimização genérico em estado estacionário:

$$\min F(\mathbf{x}, \mathbf{y}) \quad (2.1)$$

$$\text{sujeito a } h_i(\mathbf{x}, \mathbf{y}) = 0 \quad i = 1, \dots, m_1 \quad (2.2)$$

$$g_j(\mathbf{x}, \mathbf{y}) \geq 0 \quad j = 1, \dots, m_2 \quad (2.3)$$

$$\alpha_k \leq x_k \leq \beta_k \quad k = 1, \dots, p \quad (2.4)$$

$$\gamma_k \leq y_k \leq \xi_k \quad k = p + 1, \dots, n \quad (2.5)$$

na qual F é a função objectivo que se pretende minimizar², a sua determinação está sujeita à verificação das restrições de igualdade h_i bem como ao domínio de validação

²A formulação genérica do problema restringir-se-á ao caso da minimização. Um problema de maximização poderá sempre ser expresso como a minimização do simétrico da sua função objectivo.

imposto pelas restriç es de desigualdade g_j , tamb em designadas por restriç es suplementares. As vari veis presentes no sistema podem apresentar natureza diversa, \mathbf{x} corresponde ao vector cont nuo no dom nio $[\alpha_k, \beta_k]$ enquanto que \mathbf{y} representa o vector discreto com dom nio $[\gamma_k, \xi_k]$.

Na implementa o deste tipo de formula o, pode ainda ser considerada a possibilidade de incorporar os limites que definem o dom nio das vari veis \mathbf{x} e \mathbf{y} no conjunto das restriç es suplementares. A forma gen rica e compacta como o problema   definido por interm dio das Equa es 2.1–2.5 permite que, apesar de terem origem em  reas do conhecimento conceptualmente distintas, os problemas de optimiza o possam ser trabalhados por metodologias diversificadas e transversais aos contextos em que os problemas se enquadram (Arora, 1989; Gill et al., 1981).

Uma das caracter sticas pr prias dos problemas de optimiza o   apresentarem um n mero de graus de liberdade superior a zero. Este par metro   definido como a diferen a entre o n mero de vari veis do problema, n , e o seu n mero de equa es m_1 . Esta grandeza traduz a dimens o do subconjunto de vari veis existente em (\mathbf{x}, \mathbf{y}) designado por vari veis de projecto ou decis o. As mesmas, quando especificadas, permitir o calcular as vari veis de estado por resolu o do sistema de equa es composto por \mathbf{h} .

Como j  referido, a preocupa o em encontrar os conjuntos decis o que melhor satisfa am o cumprimento das exig ncias   transversal a  reas t o diversas como a economia, as engenharias, as teorias organizacionais, a gest o, o planeamento e as tecnologias. Siddall (1982) enquadra a tomada de decis o em cinco categorias distintas: problemas de escolha de projecto, problemas de optimiza o de projecto, problemas de especifica o, problemas de produtividade e problemas de n vel de risco. A todos eles pode ser associada uma atitude cont nuo de supervis o de desempenho da qual s o exemplo as seguintes situa es tipo (Tavares e Correia, 1999):

- O cumprimento cont nuo dos actuais sistemas legais de protec o ambiental;
- A introdu o de modifica es no fluxo do processo produtivo por altera o da configura o e/ou ordem das suas opera es unit rias;
- A obten o de maiores rendimentos;
- A minimiza o dos custos de produ o;
- A persegui o de maior competitividade comercial por interm dio da melhoria da qualidade do produto.

No panorama da engenharia n o   dif cil encontrar situa es que correspondam a cada uma das situa es tipo apresentadas, como s o os casos particulares do projecto de aeronaves e estruturas aeroespaciais com peso m nimo; a defini o de traject rias

óptimas para veículos espaciais; o projecto de redes eléctricas; o projecto de sistemas de abastecimento de água; a optimização de sistemas de alocação, distribuição e planeamento, etc. . . (Rao, 1996).

Em Engenharia Química podem considerar-se as aplicações ao projecto de redes de permutadores, de sistemas de separação por destilação, de redes de reactores, de redes de reacção com separação, à síntese de diagramas processuais completos ou ainda a implementação de estratégias de controlo óptimo (Floudas, 1995; Rao, 1996), casos em que a formulação genérica do problema de optimização se reveste de significado físico. Nestes exemplos as restrições de igualdade \mathbf{h} representam normalmente balanços mássicos e/ou energéticos, relações termodinâmicas, equações estequiométricas, leis cinéticas ou equações de projecto de equipamento. As condições de exequibilidade do processo são definidas pelas restrições de desigualdade \mathbf{g} , que correspondem normalmente a especificações e/ou limitações do processo em contextos de natureza operacional, legal e/ou de mercado concorrencial. A medida de desempenho do sistema, F , poderá traduzir um objectivo único ou múltiplo, apresentando carácter qualitativo, produtivo ou económico.

Apresentada a formulação genérica e terminologia base dos problemas de optimização, bem como algumas das suas ligações a situações práticas dos contextos científicos em que eles aparecem, é pertinente acrescentar que podem ser consideradas duas variantes conceptuais relacionadas com o carácter estacionário e determinístico dos modelos.

Em primeiro lugar podem ser consideradas situações em que o processo e/ou fenómenos em estudo acontecem em estado transiente. Logo, como não se verifica independência relativamente à variável tempo, nestas situações o modelo matemático recorre a equações algébrico-diferenciais. Está-se por isso perante um problema de optimização dinâmica cuja formulação genérica é dada por (Biegler e Grossmann, 2004):

$$\min F = \varphi(\mathbf{z}(t_f), \mathbf{v}(t_f), \mathbf{u}(t_f), t_f, \mathbf{p}_{ti}) \quad (2.6)$$

$$\text{sujeito a } \frac{d\mathbf{z}}{dt} = \Theta[\mathbf{z}(t), \mathbf{v}(t), \mathbf{u}(t), \mathbf{p}_{ti}]; \quad \mathbf{z}(t_0) = \mathbf{z}(0) \quad (2.7)$$

$$\mathbf{h}[\mathbf{z}(t), \mathbf{v}(t), \mathbf{u}(t), t, \mathbf{p}_{ti}] = 0 \quad (2.8)$$

$$\mathbf{z}^l \leq \mathbf{z}(t) \leq \mathbf{z}^S \quad (2.9)$$

$$\mathbf{v}^l \leq \mathbf{v}(t) \leq \mathbf{v}^S \quad (2.10)$$

$$\mathbf{u}^l \leq \mathbf{u}(t) \leq \mathbf{u}^S \quad (2.11)$$

$$\mathbf{p}_{ti}^l \leq \mathbf{p}_{ti} \leq \mathbf{p}_{ti}^S \quad (2.12)$$

$$t_f^l \leq t_f \leq t_f^S \quad (2.13)$$

$$0 \leq t \leq t_f \quad (2.14)$$

neste caso φ corresponde   funç o objectivo com valor F . A sua avaliaç o para o tempo final, t_f , depende do conjunto decis o do problema, neste caso o vector dos perfis das vari veis de controlo \mathbf{u} ; das restriç es impostas pelos perfis das vari veis de estado (dos transientes, \mathbf{z} , com valor inicial \mathbf{z}_0 , e dos alg bricos \mathbf{v}); das restriç es de igualdade \mathbf{h} ; do tempo t e dos par metros \mathbf{p} dele independentes.

Rao (1996) designa este tipo de problemas por problemas de traject ria. Isto prende-se com o facto de, ao contr rio do que acontece nos problemas estacion rios, em que se determinam valores para as vari veis de projecto, aqui pretende-se encontrar o melhor perfil para as vari veis de controlo, situaç o que leva a que o autor estabeleça a distinç o entre problemas param tricos e problemas de traject ria.

A segunda variante relativamente ao caso gen rico em estado estacion rio   diferente no que diz respeito   natureza determin stica do problema. Nos modelos determin sticos tem-se uma noç o completa da informaç o dispon vel. H  no entanto situaç es em que a tomada de decis o est  sujeita a incerteza e isso condiciona os modelos, que passam a estar baseados em ponderaç es probabil sticas. Um exemplo facilmente compreens vel seria o da maximizaç o do valor esperado a investir em acç es ou, noutro caso, a previs o de produç o para produtos com procura sazonais. Como exemplo ilustrativo de Engenharia Qu mica, poder-se-  pensar em casos com alimentaç es a unidades onde as composiç es sejam incertas devido   presença de impurezas (Varaiya, 1972).

Este tipo de problemas, denominados por problemas de programaç o estoc stica, apresentam a seguinte definiç o gen rica:

$$\min \quad G(\mathbf{x}, \mathbf{y}, \mathbf{w}) \quad (2.15)$$

$$\text{sujeito a} \quad h_i(\mathbf{x}, \mathbf{y}, \mathbf{w}) = 0 \quad i = 1, \dots, m_1 \quad (2.16)$$

$$g_j(\mathbf{x}, \mathbf{y}, \mathbf{w}) \geq 0 \quad j = 1, \dots, m_2 \quad (2.17)$$

$$\alpha_k \leq x_i \leq \beta_k \quad k = 1, \dots, p \quad (2.18)$$

$$\gamma_k \leq y_i \leq \xi_k \quad k = p + 1, \dots, n \quad (2.19)$$

em que a funç o objectivo G e as restriç es dependem dos factores de incerteza \mathbf{w} , que afectam processo de tomada de decis o aleatoriamente ou segundo determinadas distribuic es de probabilidade (Fu et al., 2005; Kleywegt e Shapiro, 2001).

Verifica-se assim que n o existe uma  nica formulaç o para os problemas de optimizaç o. As diferenç as entre elas s o consequ ncia do esforço desenvolvido no sentido de garantir, sempre que poss vel, a melhor aproximaç o  s diferentes realidades que lhes servem de base. Apesar disso, estes enunciados bem conhecidos na literatura asseguram as condiç es que possibilitam a implementaç o de estrat gias de resoluç o apropriadas.

Os trabalhos desenvolvidos para a presente tese correspondem a problemas desprovidos de incerteza nas suas variáveis, pelo que será às metodologias vocacionadas para a resolução desse tipo de situações que será dada maior relevância nos capítulos seguintes.

2.2.3 Tipologia dos Problemas de Optimização

A existência de uma moldura genérica para a definição de problemas de optimização potencia o desenvolvimento de diferentes filosofias que visem a obtenção de uma solução. No entanto, há distinções entre cada caso que não devem ser ignoradas. A abordagem a qualquer problema tem vantagem em identificar características especiais que possam ajudar a resolvê-lo mais eficientemente (Gill et al., 1981). Nesse sentido, é possível concluir que a adequabilidade de aplicação de cada metodologia dependerá do seu potencial para explorar as características particulares de cada problema. Assim, antes de saber quais as ferramentas disponíveis para lidar com problemas de optimização, a pergunta essencial que se coloca é a seguinte: qual o tipo de problema com que se está a trabalhar?

A classificação dos tipos de problemas de optimização é independente dos diferentes métodos empregues na sua resolução. A distinção das características matemáticas dos problemas mostra-se como a forma natural de os distinguir e, por norma, considera-se como primeiro critério de classificação a natureza das suas variáveis: contínuas ou discretas e, numa segunda instância, presta-se atenção à natureza das equações envolvidas: lineares ou não lineares.

Assim, para problemas que apenas apresentem variáveis contínuas consideram-se os casos de programação linear (LP – *Linear Programming*³) e programação não linear (NLP – *Nonlinear Programming*), quer não estejam ou estejam respectivamente presentes equações não lineares. Analogamente, se configurações deste tipo contiverem variáveis discretas, o tipo de equações determinará extensões dos dois casos anteriores respectivamente para problemas de programação linear mista (MILP – *Mixed-Integer Linear Programming*) e programação não linear mista (MINLP – *Mixed-Integer Nonlinear Programming*). A ausência de variáveis contínuas reduz a formulação a um problema de programação inteira (IP – *Integer Programming*) (Biegler e Grossmann, 2004).

Em cada um destes grupos podem encontrar-se casos particulares notabilizados pela recorrência com que aparecem ou ainda pela sua importância no desenvolvimento de algoritmos. São exemplo disso, dentro da família dos LP, o caso dos problemas da programa-

³Para garantir a consistência e coerência com a nomenclatura e literatura existentes, serão utilizados os acrónimos anglófonos no que diz respeito às classificações e designações dos tipos de problemas e algoritmos de optimização referenciados.

ção linear complementar (LCP – *Linear Complementary Programming*), e ainda os problemas de programação quadrática (QP – *Quadratic Programming*), e programação semidefinida (SP – *Semidefinite Programming*) na família dos NLP (Biegler e Grossmann, 2004).

Esta visão a dois critérios é a que mais frequentemente se pode encontrar na literatura. Existem no entanto autores, como Arora (1989) e Edgar et al. (2001) que definem mais critérios de classificação. A título exemplificativo apresentam-se os critérios enunciados por Rao (1996):

- Existência de restrições (problema condicionado ou não condicionado);
- Natureza das variáveis de decisão (paramétricos ou de trajectória);
- Estrutura física do problema (controlo óptimo e não óptimo);
- Natureza das equações presentes (não linear, linear ou geométrico);
- Valores admitidos para as variáveis de decisão (contínuas ou discretas);
- Natureza determinística das variáveis (determinísticos ou com incerteza);
- Separabilidade das funções (problemas separáveis ou não separáveis);
- Número de funções objectivo (problemas uniobjectivo ou multiobjectivo)

Em Engenharia Química os problemas que apresentam não linearidades são comuns. Pintér (2002) realça que a proeminência das funções não lineares é de compreensão imediata. Basta para isso atender a que a generalidade das funções básicas que se aprendem na matemática - polinomiais, de potência, exponenciais, logarítmicas e trigonométricas - desempenham um papel essencial na descrição dos modelos que aparecem em áreas como a física, a química, a biologia, o ambiente, a engenharia e a economia. NLP e MINLP são representações habituais para problemas de projecto, dependentes de equações que prevejam o comportamento de processos, enquanto que formulações lineares do tipo LP e MILP são normalmente utilizadas para problemas que lidam com o planeamento de operações e alocação de recursos. Na área do controlo de processos são normalmente utilizadas formulações desprovidas de variáveis discretas, ou seja, LP e NLP (Biegler e Grossmann, 2004; Grossmann e Biegler, 1995).

2.2.4 Plausibilidade, Convexidade e Optimalidade

A formulação generalizada do problema de otimização pode ser reduzida à forma

$$\min_{\mathbf{x} \in S} F(\mathbf{x}) \quad (2.20)$$

que introduz implicitamente um conceito importante: plausibilidade. Um ponto que satisfaça todas as restrições representadas pelas Equações 2.2 e 2.3 é denominado um ponto plausível (Equação 2.21).

$$\mathbf{x} \in S \subset \mathfrak{R}^n : \mathbf{h}(\mathbf{x}) = 0 \wedge \mathbf{g}(\mathbf{x}) \leq 0 \quad (2.21)$$

O grupo de todos os pontos que verifiquem as mesmas condições é designado por conjunto ou região plausível, aqui denotado por S .

O conceito de plausibilidade introduz novas terminologias ao nível das restrições do problema. As de igualdade dizem-se activas no espaço de pontos plausíveis S , uma vez que são obrigatoriamente verificadas para tais pontos. No caso das restrições de desigualdade, $g_j(\mathbf{x})$ diz-se activa quando $g_j(\mathbf{x}) = 0$, caso contrário será designada por folgada. A fronteira da região plausível, S , corresponde a todos os pontos onde, pelo menos, uma restrição de desigualdade se encontra activa. Os pontos que não pertencem à fronteira são designados por pontos interiores de S (Nash e Sofer, 1996). A definição de S como um conjunto fechado delimitado por restrições garante que o problema de otimização tem solução. O teorema de Weierstrass diz que a definição de uma função em domínio limitado assegura que a mesma tem dois valores extremos, o maior e o menor, quer seja no interior ou na fronteira do seu domínio (Luenberger, 1984).

A resolução de um problema de otimização pode assim ser encarada como a procura dos extremos da função para que, posteriormente, os valores de desempenho nesses extremos possam ser comparados entre si. Poderemos definir mínimo de F como o ponto \mathbf{x}^* que, na sua vizinhança, V_E , apresenta o menor valor de F .

$$F(\mathbf{x}^*) \leq F(\mathbf{x}) \quad \forall \quad \mathbf{x} \in V_E(\mathbf{x}^*) \cap S \quad (2.22)$$

resultado que, para efeitos de implementação em algoritmo, poderá ser convenientemente expresso em função do respeito de uma tolerância especificada, ε .

$$F(\mathbf{x}^*) \leq F(\mathbf{x}) \quad \forall \quad \mathbf{x} \in S : \|\mathbf{x} - \mathbf{x}^*\| < \varepsilon \quad (2.23)$$

Este resultado definido de forma simples pelas Equações 2.22 e 2.23 e que se designa por mínimo local, poderá ser considerado global se a vizinhança em torno de \mathbf{x}^* representar todo o domínio S .

$$F(\mathbf{x}^*) \leq F(\mathbf{x}) \quad \forall \quad \mathbf{x} \in S \quad (2.24)$$

O tipo de m nimos que F apresenta s o tamb m caracterizados pela monotonia da funç o. Um m nimo local pode ser considerado estrito quando $\mathbf{x} \neq \mathbf{x}^*$ na Equa o 2.24, caso contr rio ser  considerado n o estrito. A ilustra o deste conceito   apresentada na Figura 2.1.

Estas considera es permitem compreender que o desenvolvimento dos algoritmos de optimiza o tenha sido dominado pelo estabelecimento de considera es de natureza geom trica, aspecto que justifica a import ncia da introdu o da no o de convexidade.

O conjunto S diz-se convexo se

$$\alpha \mathbf{x}_1 + (1 - \alpha) \mathbf{x}_2 \in S \quad \forall \quad 0 \leq \alpha \leq 1 \quad (2.25)$$

A Equa o 2.25 diz que caso \mathbf{x}_1 e \mathbf{x}_2 pertençam a S , o segmento de recta que os une tamb m est  inteiramente contido no mesmo dom nio. Esta express o assim definida   a chamada combina o convexa de \mathbf{x}_1 e \mathbf{x}_2 e mostra, por exemplo, que os conjuntos definidos por restri es lineares s o conjuntos convexos. Uma caracter stica importante dos conjuntos convexos   que cont m todas as combina es convexas de qualquer par dos seus elementos (Horst et al., 1995).

Uma funç o $F(\mathbf{x})$ dir-se-  convexa no conjunto S se

$$F(\alpha \mathbf{x}_1 + (1 - \alpha) \mathbf{x}_2) \leq \alpha F(\mathbf{x}_1) + (1 - \alpha) F(\mathbf{x}_2) \quad (2.26)$$

rela o que em termos gr ficos pode ser interpretada por interm dio da Figura 2.2, onde se evidencia que para que f seja convexa, o segmento de recta que une $(a, f(a))$ a $(b, f(b))$ deve situar-se sobre a funç o ou na funç o f . Nos casos em que 2.26 verifica uma desigualdade para todo o dom nio $F(\mathbf{x})$ ser  estritamente convexa.

Em espa os dimensionais mais alargados torna-se impratic vel avaliar a convexidade de uma funç o em termos gr ficos. Nestas situa es atende-se a uma defini o alterna-

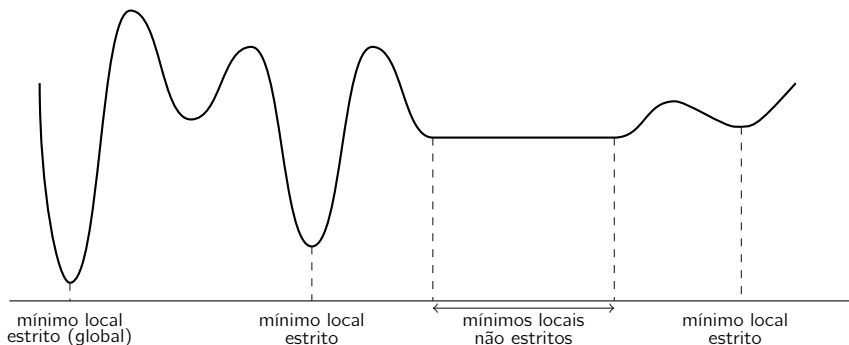


Figura 2.1 Diferentes tipos de m nimos de uma fun o gen rica.

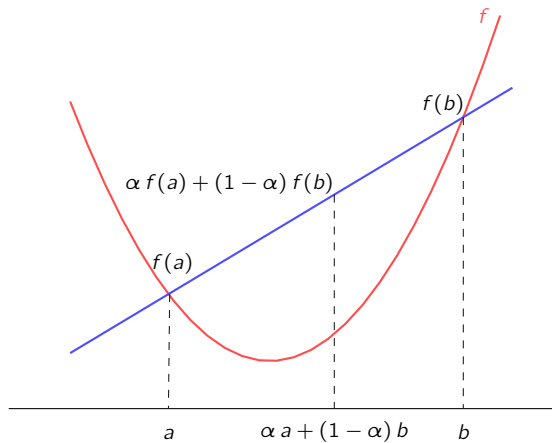


Figura 2.2 Representação de uma função convexa a duas dimensões.

tiva de convexidade. Desde que a $F(\mathbf{x})$ seja duplamente diferenciável, $F(\mathbf{x})$ será convexa se a sua segunda derivada for maior ou igual a zero. Para funções multivariáveis tal avaliação pode ser efectuada com base no cálculo do Hessiano, $H(\mathbf{x})$, da função objectivo, que, no caso ser positivo semidefinido, determinará que $F(\mathbf{x})$ é convexa. A avaliação do Hessiano pode ser efectuada pela determinação dos seus valores próprios, que caso sejam não negativos determinarão o carácter positivo semidefinido do Hessiano.

A noção de mínimo local de $F(\mathbf{x})$ dada pela Equação 2.22 não é mais prática ou acessível do que a correspondente definição de mínimo global. A determinação de condições com melhor aplicabilidade prática dos mínimos de $F(\mathbf{x})$ parte da sua expansão em série de Taylor, concluindo que a função é estacionária nos seus pontos extremos, pelo que neles se verificará a relação $\nabla F(\mathbf{x}^*) = 0$. Os pontos que obedecem a esta relação são chamados de pontos estacionários e podem ser caracterizados com base na análise da convexidade da função. O ponto estacionário \mathbf{x}^* será mínimo de $F(\mathbf{x})$ caso a função seja convexa nesse ponto, o que corresponderá a um $H(\mathbf{x}^*)$ semidefinido (Edgar e Himmelblau, 1989; Nash e Sofer, 1996).

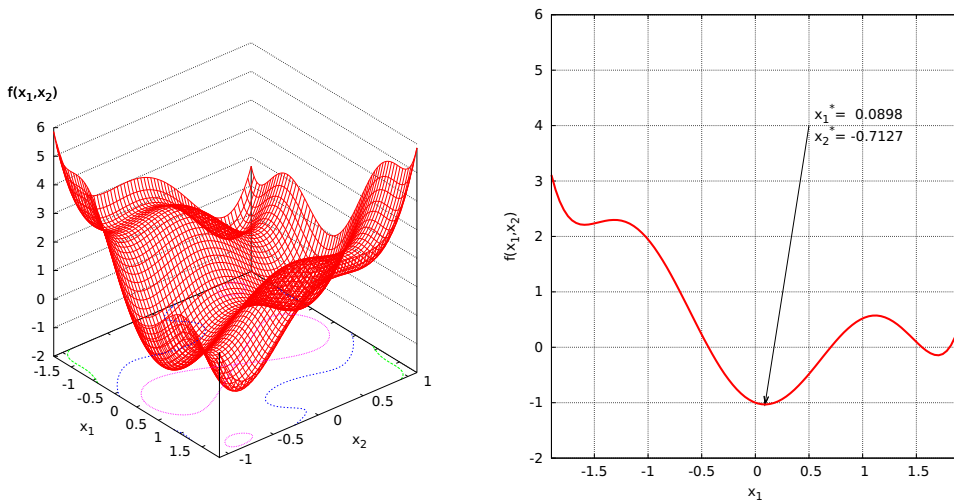
A importância do conceito de convexidade prende-se ainda com a sua utilização como critério para a determinação da existência de soluções globais. A menos que se determinem todos os mínimos locais da função objectivo, é impossível provar que a solução de um problema seja global caso a função seja não convexa (Edgar et al., 2001).

O caso particular dos problemas LP é representativo. Como a função objectivo e as restrições são lineares, logo convexas, e o conjunto S corresponde a um espaço com a forma de um poliedro, por isso convexo, a solução encontrada corresponderá

necessariamente a uma soluç o global. De referir que tal situaç o poder  assumir a forma de um conjunto m ltiplo de soluç es locais com desempenho equivalente (m nimos n o-estritos).

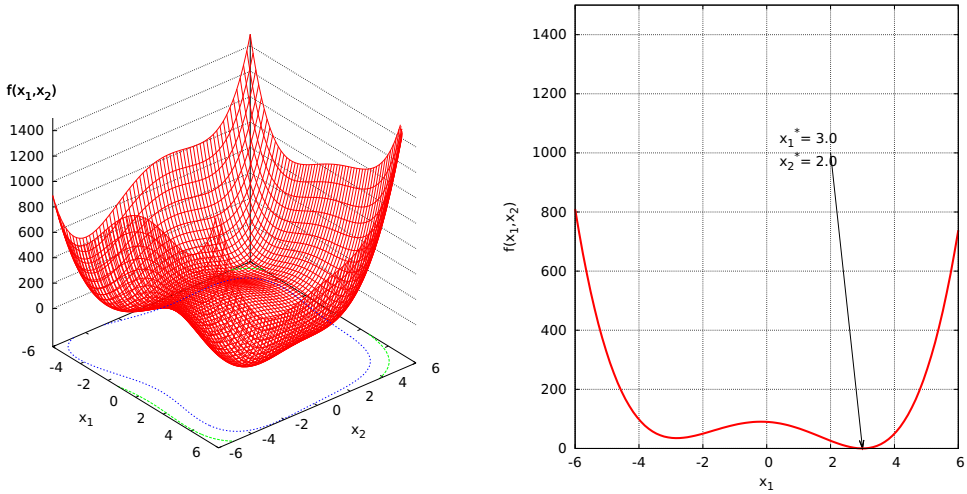
Portanto, na minimizaç o de $F(x)$, caso seja poss vel determinar que a funç o   convexa e que o conjunto de restriç es que a condicionam formam um espaço convexo, a determinaç o de um m nimo local implica que esse m nimo seja tamb m o m nimo global da funç o. No caso da funç o ser estritamente convexa, al m de global, esse m nimo ser   nico.

A an lise das condiç es de optimalidade apresentadas est o na base do desenvolvimento dos m todos de optimizaç o (Floudas, 1995). Na pr tica, a generalidade de problemas com que se trabalha no contexto de engenharia corresponde a problemas n o convexas. Basta para isso considerar que a introduç o de uma restriç o de igualdade n o linear implica obrigatoriamente que o dom nio S n o seja convexo. Atendendo a que muitos problemas apresentam um n mero de restriç es consider vel, o aparecimento de n o convexidades ser  comum e essa propriedade implica normalmente o aparecimento de uma multiplicidade de soluç es que exige a utilizaç o de t cnicas adequadas (Tavares e Correia, 1999). As Figuras 2.3, 2.4 e 2.5 s o exemplos ilustrativos de funç es n o convexas que apresentam multiplicidade de extremos.



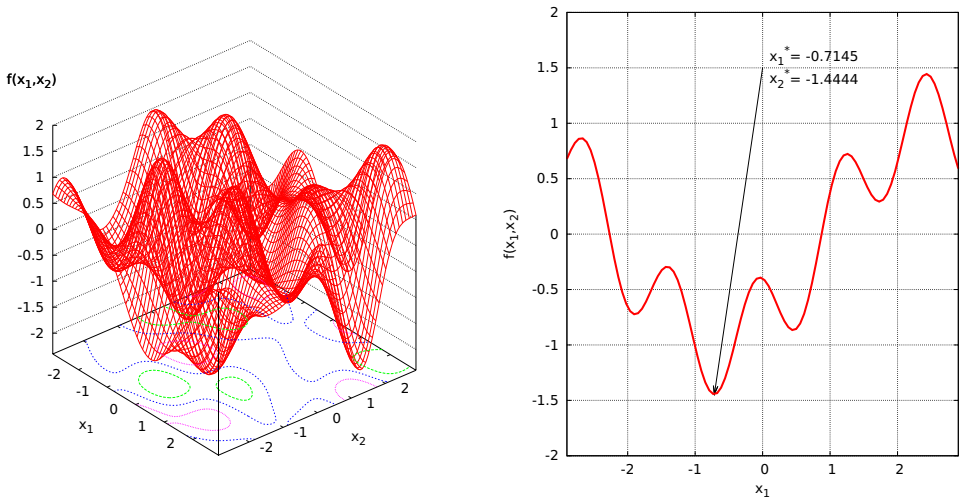
$$f(x_1, x_2) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2)x_2^2$$

Figura 2.3 Exemplo de funç o n o convexa, caso 1.



$$f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$$

Figura 2.4 Exemplo de função não convexa, caso 2.



$$f(x_1, x_2) = -0.2 (\sin(x_1 + 4x_2) - 2 \cos(2x_1 + 3x_2) - 3 \sin(2x_1 - x_2) + 4 \cos(x_1 - 2x_2))$$

Figura 2.5 Exemplo de função não convexa, caso 3.

2.3 M todos de Optimiza o

A natureza das restri es que aparecem em problemas da  rea da Engenharia Qu mica   geralmente n o linear. Atendendo ainda a que situa es como o projecto de redes de unidades, envolve vari veis discretas,   natural que os problemas do tipo NLP e MINLP assumam import ncia preponderante no contexto da  rea de Engenharia Qu mica.

Consequentemente, e apesar do vasto leque de metodologias existentes,   com base na import ncia dos problemas NLP e MINLP para a  rea cient fica que enquadra a presente tese, que se dar  aten o aos m todos para resolver problemas n o lineares condicionados. A apresenta o dos conceitos relacionados com m todos de programa o linear ou para a solu o de problemas sem restri es   por isso considerada fora do  mbito do presente trabalho.

2.3.1 Optimiza o de problemas NLP

Os problemas tipo NLP ocorrem quando, na formula o gen rica apresentada pelas Equa es 2.1–2.5 n o est o presentes vari veis discretas.

$$\min F(\mathbf{x}) \quad (2.27)$$

$$\text{sujeito a } h_i(\mathbf{x}) = 0 \quad i = 1, \dots, m_1 \quad (2.28)$$

$$g_j(\mathbf{x}) \geq 0 \quad j = 1, \dots, m_2 \quad (2.29)$$

$$\alpha_k \leq x_k \leq \beta_k \quad k = 1, \dots, n \quad (2.30)$$

Este tipo de problemas tem recebido particular aten o pela diversidade de situa es que se enquadram nesta formula o particular. A implementa o de t cnicas de programa o matem tica n o lineares com vista   sua resolu o reveste-se da maior import ncia no sentido de que esse conhecimento   tamb m aplicado a outro tipo de problemas.

Os problemas de optimiza o din mica, nomeadamente as situa es de controlo  ptimo, podem ser resolvidos como problemas NLP quando escritos sobre aproxima es discretizadas. Do mesmo modo, diversas metodologias de resolu o de problemas MINLP se baseiam na resolu o sucessiva de diferentes problemas de programa o n o linear que correspondem a diferentes configura es discretas.

Existem diferentes tipos de abordagens para lidar com a resolu o de problemas deste tipo (Floudas e Pardalos, 1995), nomeadamente:

1. M todos de penaliza o externa
2. M todos de barreira (penaliza o interna)

3. Métodos de projecção do gradiente
4. Métodos de gradiente reduzido generalizado
5. Métodos de programação linear sucessiva
6. Métodos de programação quadrática sucessiva

Os métodos de penalização externa e os métodos de barreira são muitas vezes analisados em conjunto dada a similaridade da filosofia que lhes serve de base. Ambos operam a transformação do problema no sentido de o colocar na forma não condicionada, passível de ser analisada à luz das condições de optimalidade para problemas sem restrições. Os métodos de penalização fazem-no pela adição ponderada, com recurso a parâmetros de penalização, das restrições de igualdade à função objectivo. Os métodos barreira utilizam funções que acomodam as restrições de desigualdade na função objectivo. Análises detalhadas da utilização deste tipo de métodos e de implementações de diferentes tipos de função de penalização, nomeadamente o de métodos de penalização exacta baseados em multiplicadores, podem ser encontradas em Bazaraa et al. (1993) e Gill et al. (1981).

Bazaraa et al. (1993) e Nash e Sofer (1996) consideram que os métodos 3–6 podem ser enquadrados na categoria dos chamados métodos das direcções plausíveis, visto que os mesmos partem de pontos plausíveis e utilizam direcções descendentes para encontrar novos pontos plausíveis. Dentro deste género de abordagens Biegler e Grossmann (2004) destacam os métodos de gradiente reduzido generalizado (GRG – *Generalized Reduced Gradient*) e de programação quadrática sucessiva (SQP – *Successive Quadratic Programming*), algoritmos que, dada a sua popularidade e ampla taxa de utilização, serão aqui tratados como representativos desta família de metodologias. Relativamente à aplicação prática de algoritmos para resolução de NLP, os métodos de penalização têm sido substituídos pela utilização de métodos GRG e SQP (Edgar et al., 2001).

Os algoritmos GRG justificam a sua designação pela incorporação das restrições de igualdade na função objectivo pela explicitação de algumas variáveis. Isto permite reduzir o espaço para cálculo do gradiente de n para $n - m_1$ enquanto que o tratamento das restrições de desigualdade é efectuado por intermédio de variáveis de folga. O algoritmo parte de um ponto plausível para o qual calcula $\nabla F(\mathbf{x})$, que utiliza para encontrar a direcção de procura. Sobre a direcção de procura encontra-se o passo que permita minimizar a função objectivo que garanta um novo ponto plausível sobre o qual o procedimento de procura é repetido até verificação das condições de optimalidade. Visto que garante a plausibilidade a cada iteração, este método pode exigir um maior número de iterações que a sua congénere SQP, pois requer a avaliação de restrições com vista

  plausibilidade. Uma vez que segue um trajecto plaus vel, a progress o do m todo pode ser avaliada pela an lise da evoluç o do valor da funç o objectivo, embora tal comportamento possa implicar uma maior carga computacional que, quando excessivo, poder  revelar-se como um entrave   sua aplicabilidade (Biegler e Grossmann, 2004; Biegler et al., 1997; Edgar et al., 2001; Nash e Sofer, 1996).

Os algoritmos SQP, uma evoluç o dos SLP por adiç o de informaç o de segunda ordem, correspondem a uma generalizaç o do m todo de Newton em que se procuram soluç es que satisfaç am as condiç es de optimalidade de Karush-Kuhn-Tucker (KKT) do problema NLP. O algoritmo implementa uma resoluç o recursiva a uma aproximaç o QP do problema que permite calcular a nova direcç o de procura. Uma das principais vantagens da SQP   a capacidade de convergir num n mero reduzido de avaliaç es. Para sistemas de larga escala, onde se verifica uma relativa quebra de desempenho, o algoritmo pode ser apoiado pela utilizaç o de t cnicas que permitam decompor o sistema. Os modelos SQP s o tamb m adequados em implementaç es que recorram a modelos de caixa negra (Biegler e Grossmann, 2004; Biegler et al., 1997; Edgar et al., 2001; Floudas, 1995; Nash e Sofer, 1996).

Os princ pios que regem os m todos GRG e SQP podem apresentar diversas variantes, sendo poss vel encontrar descriç es mais pormenorizadas em Bazaraa et al. (1993) e Biegler e Grossmann (2004). No que diz respeito a implementaç es de software, como algoritmos GRG mais conhecidos podem citar-se:

- CONOPT (Abadie e Carpentier, 1969; Drud, 1985, 2008, 1994)
- MINOS (Murtagh e Saunders, 1982; Murtagh et al., 2008)
- GRG2 (Lasdon e Waren, 1981; Lasdon et al., 1978), na base do SOLVER.

ao passo que para SQP podem considerar-se:

- SNOPT (Gill et al., 2008)
- rSQP (Ternet e Biegler, 1998)
- LANCELOT (Conn et al., 1992)

2.3.2 Optimizaç o de problemas MINLP

Os problemas do tipo MINLP surgem na sequ ncia da introduç o de vari veis discretas em processos que normalmente implicam alternativas estruturais, em que tais vari veis s o utilizadas para: medir ou representar situaç es que envolvam a sequ ncia de eventos; tomar decis o entre alternativas; decidir entre a exist ncia ou n o exist ncia;

dimensionar unidades de equipamento. São também consideradas situações de mistura de componentes, localização de unidades e o exemplo clássico do caixeiro viajante. A existência de um domínio combinatorial ao nível das variáveis discretas representa uma dificuldade acrescida para os problemas deste tipo, uma vez que diferentes configurações discretas corresponderão ainda a problemas não lineares não convexos (Edgar et al., 2001; Floudas, 1995; Grossmann e Turkey, 1996). Um leque alargado de exemplos com formulação MINLP são apresentados por Floudas (1995).

Biegler e Grossmann (2004) salientam que este tipo de formulação apresenta por isso uma flexibilidade considerável para a representação de problemas com variabilidade considerável, e que é nesse contexto que os engenheiros químicos têm contribuído proeminentemente no desenvolvimento deste tipo de problemas.

Edgar et al. (2001) apresentam quatro abordagens possíveis a este tipo de problemas, nomeadamente:

1. Métodos de Bifurcação e Limite
2. Métodos de Aproximação Exterior
3. Métodos alternativos de decomposição
4. Programação Disjuntiva

O métodos de bifurcação e limite (BB – *Branch and Bound* (Borchers e Mitchel, 1991; Gupta, 1980; Gupta e Ravindran, 1985; Leyffer, 2001; Ostrovsky et al., 1990)) têm como base a resolução de problemas NLP que resultam da especificação de conjuntos discretos do problema MINLP original. Como primeiro passo, o problema original é resolvido na forma de um NLP relaxado, cuja solução corresponde ao limite superior do problema. O método prossegue pela concepção de vários ramos com base na especificação discreta das variáveis com valor real na solução do NLP relaxado. Em cada um desses nodos é então resolvido um novo problema NLP, cuja solução associa a cada nodo um limite inferior do problema. Por comparação das diversas soluções encontradas, o método prossegue operando bifurcação no nodo com menor limite inferior.

Os métodos de aproximação exterior (OA – *Outer Approximation* (Duran e Grossmann, 1986; Fletcher e Leyffer, 1994)) abordam os problemas MINLP por uma decomposição NLP + MILP resolvida recursivamente. O problema NLP fornece o limite superior do problema. Com base nessa solução são linearizadas todas as funções não lineares do problema para resolução do MINLP com vista à actualização das variáveis discretas. Relativamente a esta metodologia, Floudas (1995) considera relevantes algumas das variações propostas para o algoritmo, nomeadamente: a OA com relaxação

de igualdade (OA/ER – *OA with Equality Relaxation* (Kocis e Grossmann, 1987); a OA com ER e penalizaç o aumentada (OA/ER/AP – *OA with ER and Augmented Penalty* (Viswanathan e Grossmann, 1990)); e a aproximaç o exterior generalizada (GOA – *Generalized Outer Approximation* (Fletcher e Leyffer, 1994))

De entre os m todos de decomposiç o alternativos, Biegler e Grossmann (2004) destacam a utilizaç o da decomposiç o generalizada de Bender (GBD – *Generalized Benders Decomposition* (Floudas e Ciric, 1989; Geoffrion, 1972; Paules e Floudas, 1989)). O m todo   similar ao algoritmo de OA, com a diferena de que a decomposiç o do MINLP corresponde a um NLP + MIP, em que as vari veis cont nuas n o fazem parte do problema linearizado com vista   actualizaç o das vari veis discretas. Ainda no que concerne aos m todos de decomposiç o, Floudas (1995) refere a decomposiç o generalizada de Cross (GCD – *Generalized Cross Decomposition* (Holmberg, 1990))

An lises detalhadas a estas metodologias s o apresentadas por Biegler e Grossmann (2004) que, no mesmo artigo, referem ainda a utilizaç o de m todos de plano de corte extendido (ECP – *Extended Cutting Plane*) como opç o de soluç o para problemas MINLP. Este algoritmo, proposto por Westerlund e Pettersson (1995) resultado da adaptaç o do m todo de plano corte para NLP convexos proposto por Kelley (1960), diferencia-se por n o seguir uma metodologia baseada na soluç o de diferentes subproblemas NLP. Considerando ainda outras possibilidades, refer ncia-se tamb m ao m todo da aproximaç o plaus vel (FA – *Feasible Approximation* (Mawengkang e Murtagh, 1986)), cuja procura da soluç o se baseia no arredondamento do NLP relaxado associado ao problema MINLP original (Floudas, 1995).

A utilizaç o de programaç o disjuntiva (DP – *Disjunctive Programming*) pretende incorporar na modelizaç o o potencial da representaç o por interm dio de rela es l gicas (Turkay e Grossmann, 1996; Vecchietti e Grossmann, 2000). Um exemplo dessa formulaç o   dado pelas Equa es 2.31 a 2.33 (Edgar et al., 2001):

$$\min \quad (x_1 - 3)^2 + (x_2 - 2)^2 + c \quad (2.31)$$

$$\text{sujeito a:} \quad \left[\begin{array}{c} Y_1 \\ x_1^2 + x_2^2 - 1 \leq 0 \\ c = 2 \end{array} \right] \vee \left[\begin{array}{c} Y_2 \\ (x_1 - 4)^2 + (x_2 - 1)^2 - 1 \leq 0 \\ c = 1 \end{array} \right] \\ \vee \left[\begin{array}{c} Y_3 \\ (x_1 - 2)^2 + (x_2 - 4)^2 - 1 \leq 0 \\ c = 3 \end{array} \right] \quad (2.32)$$

$$\text{e} \quad 0 \leq x_i \leq 8, \quad i = 1, 2 \quad (2.33)$$

no qual as variáveis lógicas Y_i adoptam o valor verdadeiro ou falso, determinando exclusivamente qual dos conjuntos de restrições representados pela Equação 2.32 deve ser utilizado.

Estes problemas são também considerados um caso particular dos problemas MINLP, já que apresentam uma condição que implica que apenas um determinado conjunto de restrições seja satisfeito (Edgar et al., 2001). As condições lógicas da formulação são chamadas de disjunções, as quais ocorrem normalmente em problemas de síntese (Hooker e Osorio, 1999).

A abordagem a problemas DP passa pela sua formulação como problema disjuntivo generalizado (GDP - *Generalized Disjunctive Programming*) em que variáveis binárias substituem as variáveis lógicas (Edgar et al., 2001; Lee et al., 2000). Esta operação tem como base a colocação das restrições lógicas na sua forma conjuntiva normal, conceito abordado por Biegler et al. (1997) e Floudas (1995) (ver Apêndice D) no tratamento das disjunções, estabelecendo como formulação final para estes casos uma representação habitualmente designada por *big-M*, em que a constante M é utilizada para relaxar ou apertar as restrições do problema (Edgar et al., 2001; Lee e Grossmann, 2005). As Equações 2.34 a 2.40 correspondem à representação *big-M* do problema definido pelas Equações 2.31 a 2.33.

$$\min \quad (x_1 - 3)^2 + (x_2 - 2)^2 + 2y_1 + y_2 + 3y_3 \quad (2.34)$$

$$\text{sujeito a: } x_1^2 + x_2^2 - 1 \leq M(1 - y_1) \quad (2.35)$$

$$(x_1 - 4)^2 + (x_2 - 1)^2 - 1 \leq M(1 - y_2) \quad (2.36)$$

$$(x_1 - 2)^2 + (x_2 - 4)^2 - 1 \leq M(1 - y_3) \quad (2.37)$$

$$y_1 + y_2 + y_3 = 1 \quad (2.38)$$

$$\text{e } y_i \in \{0, 1\}, \quad i = 1, 2, 3 \quad (2.39)$$

$$0 \leq x_i \leq 8, \quad i = 1, 2 \quad (2.40)$$

Grossmann e Kravanja (1995) salientam a vantagem da utilização das variáveis lógicas Y_i para escolher entre conjuntos que integram equações algébricas, disjunções lógicas e relações lógicas, explorando a utilização de diferentes métodos para lidar com a formulação lógica. Os conceitos de DP surgem como ferramentas úteis à optimização MINLP para síntese estrutural, na qual se determinam quais e quantos elementos estão presentes, como estão ligados entre si e quais os seus parâmetros. Kravanja et al. (1998a,b,c) Lee e Grossmann (2000) e Raman e Grossmann (1994) apresentam de forma detalhada os conceitos inerentes à modelização nos casos DP.

Vecchietti e Grossmann (1999) reforçam a import ncia deste tipo de abordagem pela exclus o das equa es relacionadas com elementos n o existentes. Apresenta tamb m um modelo h brido dedicado a situa es onde n o convenha existirem caudais nulos, ou onde a formula o *big-M* se mostre inapropriada. Quando se trata de tentar resolver problemas em que n o existam determinadas unidades, as equa es de balanço material e energ tico dessas unidades t m que ser satisfeitas. Isso conduz   ocorr ncia de singularidades devido a equa es redundantes em consequ ncia da configura o discreta em avalia o. O prop sito da optimiza o l gica disjuntiva   evitar que este tipo de situa o se verifique.

A DP representa uma tentativa de melhorar a aplica o dos m todos de optimiza o por uma mudança na abordagem conceptual   formula o de problemas do tipo MINLP. Como refer ncia final a esta sec o, apresentam-se a t tulo de exemplo algumas implementa es de software com base nos algoritmos para MINLP apontados anteriormente.

- DICOPT (Grossmann et al., 2008; Viswanathan e Grossmann, 1990)
- MINLP__BB (Leyffer, 2001)
- BARON (Sahinidis, 1996; Sahinidis e Tawarmalani, 2008)
- MINOPT (Schweiger e Floudas, 1998)

2.3.3 Optimiza o Global

Os problemas NLP e MINLP representam realidades descritas por modelos n o lineares complexos que conduzem ao aparecimento de m ltiplos  ptimos locais com uma elevada probabilidade, situa o reforçada quando o espaço onde se pretende minimizar a fun o objectivo   combinatorial. Uma vez que a qualidade desses  ptimos locais pode variar substancialmente, define-se assim o ponto de partida para ir ao encontro de m todos que efectuem a procura das condi es que garantem o melhor desempenho global do sistema, chamados m todos de optimiza o global (GO – *Global Optimization*), cujo desenvolvimento   considerado um dos principais desafios a atrair a aten o dos trabalhos de investiga o futuros (Grossmann e Biegler, 2004; Pint r, 2002).

Atendendo a que o n mero e a qualidade das solu es m ltiplas do dom nio do problema s o desconhecidos, podendo por isso existir situa es com um n mero alargado de solu es poss veis, a aplica o das abordagens cl ssicas descritas na sec es 2.3.1 e 2.3.2 pode mostrar-se ineficiente. Tais metodologias utilizam informa es inerentes   topologia da fun o objectivo, convergindo por isso para o  ptimo que lhes est  *mais pr ximo*. Nesse sentido, e em fun o da import ncia relativa que o melhor resultado

tenha na garantia da boa qualidade operacional dos processos em estudo, o problema de optimização poderá exigir a escolha de algoritmos que procurem encontrar o seu óptimo global. A diferença entre óptimos locais e globais depende assim da realidade física do sistema (Balakrishnan e Boyd, 1992; Edgar et al., 2001; Pintér, 2002).

Horst et al. (1995) identifica situações típicas que conduzem a problemas de optimização global, nomeadamente situações que avaliem modelos económicos ou para resolução de problemas financeiros; bases de dados; processamento de imagem e ainda áreas do conhecimento como as engenharias química e ambiental ou a biologia molecular. A importância que os problemas de optimização global assumem na área da verificação de risco é salientada por Neumaier (2004), que reforça a ideia dos perigos que podem advir de uma subestimação de avaliação do pior caso possível, situação que, como afirmam Balakrishnan e Boyd (1992), assume também importância especial nos casos da área de controlo de processos que visem análise de robustez.

Os problemas de projecto são mais susceptíveis de exigirem estratégias vocacionadas para a optimização global. Nestes casos é habitual trabalhar-se com domínios mais alargados para \mathbf{x} e \mathbf{y} , condição que facilita o aparecimento de óptimos locais na presença de não convexidades (Biegler e Grossmann, 2004). Floudas (1997) enquadra o interesse renascido pela optimização global no âmbito de Engenharia Química com base em ocorrências de três campos distintos:

- Síntese de processos – onde aparecem exemplos de estudos em áreas como o equilíbrio de fases (óptimo deve ter significado físico), a separações não lineares, modelos de minimização de energia, sistemas de reacção, estimativa de parâmetros, reconciliação de dados e vias metabólicas.
- Controlo – por intermédio da análise robusta de controlo em sistemas com parâmetros sujeitos a incerteza, análise de estabilidade, e controlo óptimo de mecanismos reaccionais complexos ou ainda em controlo preditivo não linear.
- Operação de processo – no projecto de sistemas com incerteza, em problemas de planeamento, alocação e distribuição ou ainda na análise de diagramas processuais contínuos, semicontínuos ou por partidas.

Em química computacional salienta-se a necessidade de optimalidade global para o projecto de moléculas orgânicas e a previsão de estruturas tridimensionais de proteínas e outros compostos.

Os critérios para classificar os métodos de optimização global não são consensuais entre os diversos autores que estudam a área. Há, apesar de tudo, ideias comuns a todas

essas perspectivas que permitem definir dois grandes grupos de m todos. Os m todos determin sticos, muitas vezes denominados por exactos; e os m todos estoc sticos, tamb m referenciados como heur sticos. Esta terminologia vai ao encontro da perspectiva de Edgar et al. (2001), em que o crit rio base relaciona-se com a forma como os algoritmos operam sobre o problema. Existem vis es alternativas como a de Neumaier (2004), que classifica os m todos de optimiza o global em fun o do tipo de converg ncia que verificam na aproxima o ao  ptimo. Papamichail e Adjiman (2004) atribui aos dois tipos de m todos crit rios de converg ncia distintos, associando aos determin sticos crit rios baseados no respeito por toler ncias, em contraste para com os argumentos estat sticos que suportam a aplica o dos estoc sticos.

Em cada uma das categorias h  metodologias que se destacam. Abordagens baseadas em bifurca o e limite, na implementa o de m todos intervalares ou na adop o de algoritmos de optimiza o local partindo de m ltiplos pontos iniciais s o comuns por entre a ala determin stica. No que diz respeito aos algoritmos estoc sticos h  quatro m todos que apresentam maior visibilidade na literatura, sendo por isso considerados os de maior import ncia, nomeadamente os algoritmos de procura aleat ria, de recozimento simulado, os m todos de procura tabu e as abordagens evolucion rias (Figura 2.6).

Banga e Seider (1996) atribuem a necessidade de implementa o de t cnicas de optimiza o em Engenharia Qu mica para lidar com o aparecimento de fun es objectivo multimodais, salientando o papel que abordagens estoc sticas poder o desempenhar nos casos em que se verifique a exist ncia de descontinuidades. Floudas (1997) destaca o papel dos engenheiros qu micos nesta  rea do conhecimento, e aponta tr s raz es para a import ncia dos m todos GO em Engenharia Qu mica: (1) n mero alargado de problemas de s ntese; (2) a falha de algumas abordagens locais em determinados problemas e ainda (3) a necessidade de uma interpreta o f sica v lida das solu es.

M todos determin sticos em GO

Diversos autores como Floudas (1997), Grossmann e Biegler (2004) e Pint r (1996), apresentam em detalhe classifica es extensivas de m todos determin sticos para optimiza o incluindo variantes de implementa o bem como os conceitos matem ticos que lhe servem de base.

Um vasto leque de metodologias determin sticas baseia-se na filosofia dos m todos BB (Grossmann e Biegler, 2004). O m todo recorre a uma divis o sucessiva do problema original em problemas mais pequenos que inevitavelmente acabar o por apresentar uma solu o mais acess vel. A introdu o da no o de limite tem por objectivo permitir a rejei o de subproblemas que n o interesse resolver (Neumaier, 2004). Ainda

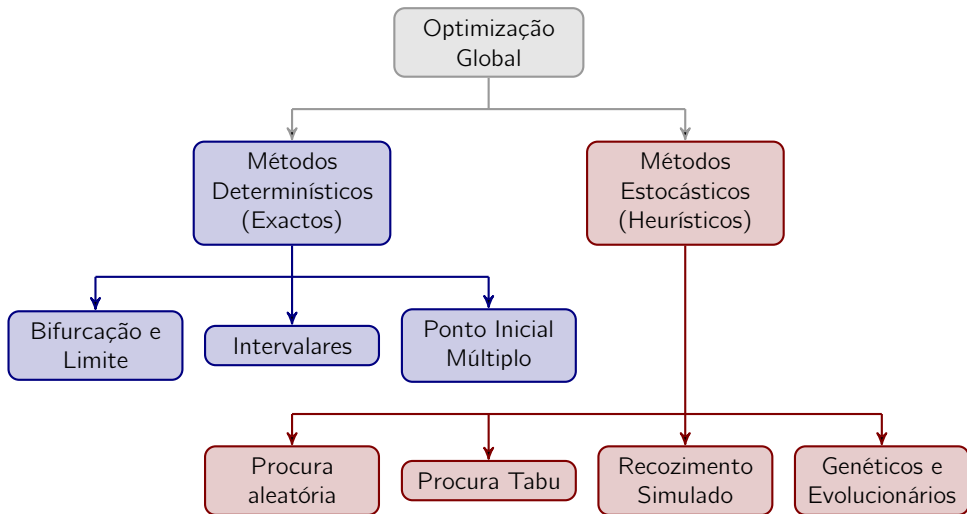


Figura 2.6 Classificação dos métodos de optimização global.

assim, apesar da ramificação no espaço, a resolução dos problemas NLP não convexos mantém-se como uma dificuldade. E é nesse cenário que Horst et al. (1995) salienta a importância da utilização de envelopes convexos na aproximação da função objectivo do problema, ferramentas utilizadas diversas vezes na subestimatação da solução do problema NLP. O algoritmo implementado mantém o histórico dos limites superior e inferior da função objectivo, os quais são garantidos na terminação (Balakrishnan e Boyd, 1992), existindo também propostas para a sua modificação com base em heurísticas (Silver, 2004). Adjiman et al. (1997) e Smith e Pantelides (1997) são referências fundamentais na aplicação de métodos BB em optimização global, com particular interesse na evolução de um método BB que opera ramificações nas variáveis contínuas e discretas, utiliza redução limite e subestimadores, o que levou à génese do optimizador global BARON (Sahinidis e Tawarmalani, 2008; Tawarmalani e Sahinidis, 2004).

Neste tipo de algoritmos as aproximações conduzidas em cada iteração para aproximação do problema, designadamente pela operação de linearizações, pode condicionar o domínio original do problema eliminando regiões onde o óptimo global possa estar contido (Biegler et al., 1997; Grossmann e Turkay, 1996).

Os métodos intervalares têm como ideia base a procura de uma solução que respeite uma tolerância. Utiliza-se a participação recursiva do espaço de procura até demonstrar que em cada um dos subdomínios não se encontra o óptimo global (McKinnon et al.).

Os m todos de m ltiplo ponto inicial implementam de uma algoritmo local ao longo do dom nio do problema em diferentes pontos iniciais simultaneamente, podendo ser uma ferramenta de abordagem inicial para defini o de zonas de explora o em optimiza o local (Torn et al., 1999).

Os algoritmos determin sticos s o utilizados na abordagem a problemas de programa o semi-infinita. Tratam-se de formula es com um n mero infinito de restri es, resolvidas por transforma o local do problema original em problema finito ou, ainda, por discretiza o do conjunto de restri es com vista a torn -lo finito (Hettich e Kortanek, 1993; Lopez e Still, 2007; R ckmann e Alfredo G mez, 2006).

M todos estoc sticos em GO

Os m todos estoc sticos s o tamb m designados por heur sticos ou probabil sticos. Estas designa es t m a ver com os tra os que caracterizam as formas como os algoritmos exploram os problemas, bem como as garantias que podem ou n o oferecer de encontrar o  ptimo global. Apesar de n o garantirem a solu o  ptima do ponto de vista matem tico, a utiliza o de heur sticas diz respeito   utiliza o de premissas que, baseadas em ju zo ou experi ncia, pare am conduzir a solu es razo veis. Os algoritmos estoc sticos partem de uma solu o do problema explorando a vizinhan a da mesma. A efici ncia de explora o desse espa o   conduzida por heur sticas. Tendem a favorecer a optimiza o global uma vez que, quando existentes, os seus crit rios de paragem n o s o baseados em gradientes (Edgar et al., 2001; Floudas, 1997; Grossmann e Biegler, 2004; Silver, 2004).

A opini o sobre as capacidades dos algoritmos estoc sticos n o   un nime. As principais vantagens apontadas aos algoritmos estoc sticos s o a sua robustez, a capacidade de obten o r pida de solu o para determinados problemas e ainda facilidade de implementa o, que possibilita o acoplamento a simuladores. Por outro lado,  -lhes apontada alguma falta de rigor e aus ncia de capacidade de acomodar problemas altamente condicionados, tendo alguns autores considerado que a sua utiliza o   relevante em problemas em que a avalia o da fun o objectivo n o seja dispendiosa ou o vector decis o n o apresente dimens o consider vel (Grossmann e Biegler, 2004; Neumaier, 2004; Silver, 2004).

Choi et al. (1999) refere o preju zo causado pela exist ncia de restri es de igualdade prejudiciais   aplica o de m todos estoc sticos pela dificuldade acrescida que acarretam para a determina o da fun o objectivo. Nesse sentido aponta a aplica o destes m todos em Engenharia Qu mica a problemas n o condicionados ou apenas com restri es de desigualdade, visto que outras situa es podem obrigar a modificar o problema

ou o algoritmo (Choi e Manosuouthakis, 2002).

De seguida apresenta-se uma descrição dos quatro principais métodos estocásticos apresentados na Figura 2.6.

Métodos de procura aleatória A procura aleatória (RS - *Random Search*) baseia a sua aplicação na exploração do espaço do domínio do problema. Quando essa procura é uniforme sobre o domínio, diz-se que a procura é considerada pura. Na prática, a aplicação desta metodologia está normalmente sujeita à incorporação de heurísticas que modificam o padrão de exploração do espaço. Surgem assim os algoritmos de procura aleatória localizada, de procura aleatória localizada melhorada ou ainda os de procura aleatória adaptativa (Banga et al., 2003a; Spall, 2003), exemplos de entre um possível leque mais alargado de modificações.

A procura aleatória adaptativa está na base da proposta de Salcedo et al. (1990) e Salcedo (1992b) para o algoritmo MSGA (MinIp Salcedo Gonçalves Azevedo), que opera melhorias nas heurísticas dos algoritmos propostos por Luus (1973) e Gaines e Gaddy (1976), explorando o espaço solução com recurso a técnicas baseadas na modificação desse domínio por intermédio da alteração dinâmica dos intervalos de procura. Adequado para problemas NLP ou MINLP, o MSGA desloca-se na região de procura ao mesmo tempo que modifica de forma diferenciada as amplitudes do vector decisão, que podem contrair ou expandir. A penalização das simulações não satisfatórias ocorre segundo morte súbita e a geração de novos pontos é baseada no melhor resultado actual registado. O MSGA permite ainda a realização de teste nos extremos do intervalo de procura, procedimento que se adequa à avaliação de situações limite sobre a fronteira do domínio. O procedimento de optimização pode considerar-se estruturado em duas fases, com relaxação prévia das restrições com vista à procura de plausibilidade, para posteriormente se iniciar o processo de optimização propriamente dito.

Melhorias na aplicação dos métodos de procura aleatória foram propostas por Price (1978) com a apresentação da variante designada por procura aleatória controlada (CRS - *Controlled Random Search*). Ali et al. (1997) e Kaelo e Ali (2006) apresentam estudos numéricos com base na inclusão de modificações no algoritmo. O uso do método simplex, de interpolações lineares ou quadráticas e ainda a ocorrência de mutação são exemplos de que conduzem a melhorias no comportamento do algoritmos existentes.

Recozimento Simulado O recozimento simulado (SA - *Simulated Annealing*) é uma metodologia que se baseia na analogia física do arrefecimento controlado de metais que, após fusão, atingem espontaneamente configurações cristalinas de equilíbrio ca-

racterizadas por apresentarem energia potencial m nima. A metodologia foi proposta por (Kirkpatrick, 1984; Kirkpatrick et al., 1983) que estabelecem a analogia entre a temperatura do metal e o n vel de energia avaliado. O SA tem por base o algoritmo de Metropolis (Metropolis et al., 1953), que implementa o c lculo do n vel de energia de um sistema de part culas discretas, onde ocorrem transiç es probabilisticamente condicionadas segundo a distribuiç o de Boltzmann (Dolan et al., 1989).

Assim, o algoritmo permite que, a qualquer momento e no decurso da avaliaç o da funç o desempenho, possam ser dados saltos para n veis desfavor veis de energia com base na probabilidade P dada pela Equa o 2.41,

$$P(\Delta E) = \exp\left[-\frac{(E_2 - E_1)}{K_B T}\right] \quad (2.41)$$

em que P se aproxima de um quando o valor da temperatura (T)   muito elevado, e decai exponencialmente   medida que a temperatura desce. A transiç o entre n veis os n veis de energia E_1 e E_2 depende da constante de Boltzmann, K_B . Este mecanismo permite que o algoritmo possa escapar aos  ptimos locais com base na aceitaç o da ocorr ncia transiç es desfavor veis mesmo em n veis de temperatura mais baixos. Dada a sua natureza probabil stica, o SA apresenta converg ncia assint tica para o  timo global quando considerado um n mero infinito de avaliaç es. Apesar deste comportamento   poss vel encontrar soluç es pr ximas do  timo global desde que se estipule um esquema de arrefecimento lento e adequado (Dolan et al., 1989).

A an lise e implementa o de variantes de aplica o do SA   apresentada por Kalivas (1995) e Ali e Storey (1997) onde se destaca recozimento simulado generalizado (GSA – *Generalized Simulated Annealing*) como estrat gia que oferece melhor converg ncia que a configura o tradicional. Azencott (1992) aponta as dificuldades demonstradas pelo algoritmo no que diz respeito aos cuidados necess rios para a boa configura o dos esquemas de arrefecimento. Atribui o  xito do algoritmo   sua simplicidade e aparente universalidade ao mesmo tempo que prop e diferentes t cnicas para a sua paraleliza o.

O recozimento simulado   a heur stica adoptada no desenvolvimento do otimizador estoc stico MSIMPISA (*Minlp SIMPLex Simulated Annealing*), apresentado e aplicado por Cardoso et al. (1994, 1996, 1997) e Cardoso (1998). O algoritmo tem como base a recria o da proposta inicialmente apresentada por Press e Teukolsky (1991), que aproveita o potencial da conjugac o do algoritmo de Metropolis com o m todo Simplex (Nelder e Mead, 1965).

Esta metodologia h brida utiliza o m todo Simplex para explorar o dom nio das vari veis, procedendo posteriormente a perturbaç es nas configuraç es do simplex. Na sequ ncia de tais perturbaç es, transiç es desfavor veis para piores n veis de desempe-

no são permitidas segundo o critério probabilístico incorporado através da inclusão do algoritmo de Metropolis, comportamento que pretende promover o escape a óptimos locais. Revelando-se independente de estimativas iniciais, o MSIMPSA também avalia as simulações não satisfatórias segundo critério de penalização por morte súbita. Opcionalmente, através de um esquema de penalização dinâmica, o algoritmo permite que a optimização progrida segundo um trajecto não plausível. Nessa situação, o valor da função objectivo é penalizado pela incorporação ponderada das violações verificada nas restrições suplementares.

Procura tabu A procura tabu é um algoritmo proposto por Glover (1989, 1990a), tendo como ideia base a retenção de um histórico de avaliações que não permita que pontos previamente avaliados sejam revisitados na exploração do espaço, pelo menos durante algumas iterações. O controlo deste comportamento é definido pela heurística que determina a dimensão do histórico de valores previamente avaliados (lista tabu), e que funciona como uma espécie de memória que mantém o registo das áreas proibidas do domínio, as quais são designadas por áreas tabu (Silver, 2004).

Tal como acontece com o SA, a procura tabu também permite movimentações a partir de óptimos locais para outros pontos. No entanto, e ao contrário do que se verifica em SA, tal opção não é de natureza probabilística como não é passível de acontecer a qualquer momento. Glover (1990b) apresenta uma lista detalhada de possíveis aplicações desta metodologia, assim como referências suplementares e esquemas detalhados com vista à sua implementação.

Algoritmos evolucionários Os algoritmos evolucionários têm por base a mimetização de processos de selecção natural, as ideias que os sustentam emergiram simultaneamente em ambos os lados do Atlântico. Nos EUA Holland (1975) apresenta os algoritmos genéticos enquanto que na Alemanha Rechenberg (1973) e Schwefel (1975) desenvolvem a ideia dos algoritmos evolucionários (Back et al., 1991).

A sua principal diferença relativamente aos métodos já apresentados é que se baseiam na gestão de populações de soluções, ao contrário das outras abordagens que avaliam vectores de decisão singulares. A procura do melhor desempenho ocorre na evolução geracional, na qual uma nova população é criada por recombinação da antecessora segundo a selecção, cruzamento e mutação dos indivíduos. Deste tipo de metodologias destacam-se os algoritmos genéticos. Os métodos partem de uma população inicial que deve encontrar-se bem dispersa no espaço de procura. Na procura de novas soluções, os algoritmos genéticos operam transformações designadas por cruzamento e mutação (Holland, 1975; Neumaier, 2004; Silver, 2004; Spall, 2003).

2.4 Simulaç o de Processos Qu micos

A aplicaç o dos diferentes m todos de optimizaç o apresentados ao contexto da Engenharia Qu mica   normalmente efectuada com o intuito de optimizar o desempenho de processos. Essa situaç o conduz a uma quest o imediata, relacionada com a representaç o do problema: qual a melhor forma de simular o comportamento dos processos intervenientes?

Essencialmente, a simulaç o de processos qu micos pode ser efectuada segundo duas abordagens: a sequencial modular (*SeqM - Sequential Modular*) e a orientada por equa es (*EO - Equation Oriented*) (Barton et al., 2000; Biegler et al., 1997; Edgar et al., 2001; Morton, 2003). Como o pr prio nome indica, a primeira baseia-se na resoluç o sequencial de modelos unit rios. No segundo caso, segue-se uma abordagem onde, ap s reunidas todas as equa es que traduzem o processo em estudo, o sistema por elas formado   resolvido simultaneamente atrav s de um algoritmo apropriado para o efeito (Barton et al., 2000; Biegler et al., 1997).

Os simuladores modulares garantem normalmente uma constru o e depura o de modelos mais acess veis, embora sejam considerados algo inflex veis. Cronologicamente, esta abordagem foi desenvolvida nos anos cinquenta, altura que a filosofia EO se confinava ao interesse acad mico. Em 1970 surge uma metodologia interm dia, modular simult nea (*SimM - Simultaneous Modular*), em consequ ncia do aparecimento do ASPEN no MIT. A chegada e desenvolvimento de EO   ind stria acontece j  na d cada de 90 do s culo XX (Biegler et al., 1997).

Na abordagem SeqM, os modelos s o constru dos ao n vel do diagrama processual, normalmente com o aux lio de interfaces gr ficas que permitem desenhar o esquema de opera es unit rias com base na escolha de blocos ligados por sinais, o que n o   favor vel no caso em que os diagramas processuais sejam muito densos. Esta filosofia corresponde normalmente a um grau de abstrac o mais elevado que diferencia claramente a forma de trabalho entre o perito do simulador que constr i blocos e o utilizador final que apenas tira partido deles. Os modelos obtidos por SeqM s o normalmente considerados limitados quanto ao seu n vel de detalhe (Marquardt, 1996).

A filosofia modular mimetiza de certa forma a sequ ncia de c lculo manual do diagrama, estabelecendo uma analogia que permite uma compreens o mais intelig vel do modelo facilitando a depura o. No que diz respeito   converg ncia, os problemas s o decompostos a dois n veis, a saber: converg ncia por unidade processual e converg ncia de todo o diagrama. Logo, a exist ncia de ciclos ou especifica es em n mero consider vel torna a resolu o muito dispendiosa pela obriga o de converg ncia nas vari veis

de reciclagem (Barton et al., 2000; Morton, 2003).

Em termos estruturais, a filosofia SeqM lida com modelos encapsulados com procedimentos distribuídos por três níveis diferentes: topologia do diagrama, modelo das operações unitárias e propriedades físicas (Biegler et al., 1997).

Barton et al. (2000) consideram que nas estratégias orientadas a equações a descrição do problema é mais acessível, uma vez que as equações são compiladas num modelo resolvido pela aplicação de um código (utilizado para determinar as raízes do sistema de equações associado). As estratégias EO apresentam por isso uma taxa de sucesso directamente dependente da robustez do código que se utiliza para resolver o modelo.

Dotada de flexibilidade, a simulação EO é considerada mais apropriada a situações de projecto com restrições ou mesmo para estruturas mais complexas com elevado nível de ciclos, visto que permite representar facilmente as restrições associadas a estes casos e é mais flexível na atribuição das especificações (Barton et al., 2000; Morton, 2003; Rodriguez-Toral et al., 2000). Os modelos são mais fáceis de expandir e manipular, mostrando-se também apropriados para aplicação em optimização estacionária e dinâmica, optimização de diagrama processual, bem como para a análise de sensibilidade de parâmetros. Como desvantagens apontam-se a menor robustez dos *solvers* de sistemas de equações face aos algoritmos de abordagens modulares (Barton et al., 2000; Ishii e Otto, 2008; Tolsma et al., 2002).

Marquardt (1996) refere que os modelos EO são construídos com base em linguagens declarativas de modelização ou pela conjugação de sub-rotinas escritas em linguagens como o FORTRAN, não existindo por isso diferenciação entre os utilizadores. A utilização de linguagens declarativas revestem a EO de alguma elegância (Ishii e Otto, 2008), apresentando ainda a vantagem de ter um nível de abstracção mais elevado do que o C ou o FORTRAN, característica que contribui para a minimização da exposição do utilizador ao erro de codificação (Banga et al., 2003b). A utilização dessas linguagens é justificada com base em três motivos: (1) menor frequência do processo de compilação/ligação/depuração; (2) a sua concretização por escrita de ficheiros de texto; (3) a possibilidade de estabelecer e utilizar estruturas recursivas, diminuindo a dimensão dos problemas. Tolsma et al. (2002) acrescentam que a utilização de linguagens declarativas com sintaxe bem definida contribui para a concepção de ferramentas que permitam a transcrição entre elas, possibilitando a conversão de códigos fonte.

Em Engenharia Química, as interacções físico químicas são fonte de não linearidades (Seider et al., 1991). Deste modo, a filosofia de modelização EO torna as unidades processuais indistinguíveis. O método explora matrizes esparsas, o que acarreta mais tempo à construção dos modelos. Problemas de maior dimensão dependem do custo

de avaliaç o do jacobiano, j  que a resoluç o do sistema de equaç es   normalmente conduzido pela aplicaç o de um m todo de Newton modificado (Barton et al., 2000; Morton, 2003).

No caso de se utilizarem modelos rigorosos, a resoluç o do sistema de equaç es pode corresponder a 90% do tempo de computa o total. Em funç o deste comportamento, Lee et al. (1966) destacam a preocupaç o com a escolha das vari veis de decis o, pois isso determina a estrutura de conjuntos de sa da, que por sua vez apresentam dificuldade de resoluç o vari vel. Uma vez que as equaç es s o resolvidas simultaneamente, os m todos de decomposiç o para matrizes esparsas por partiç o e ruptura revelam-se de grande import ncia (Biegler et al., 1997).

As matrizes n o apresentam por norma propriedades estruturais desej veis, sendo frequentes os casos em que s o esparsas em funç o da baixa taxa de incid ncia de vari veis por equaç o e que exigem m todos apropriados de resoluç o. Zitney et al. (1996) e Shacham e Kehat (1976) salientam a import ncia que os algoritmos de decomposiç o representam   obtenç o de matrizes de projecto (conjunto de sa da) mais favor veis do ponto de vista do custo de resoluç o estrutural (Steward, 1981). A aplicaç o dos mesmos tem beneficiado imenso com o desenvolvimento e avanços verificados ao n vel do desempenho das ferramentas computacionais (Biegler et al., 1996; Seader, 1985)

Em funç o das operaç es matriciais, a simulaç o EO conduz normalmente a maiores despesas computacionais. Barton et al. (2000) afirmam que a evoluç o do hardware contrabalança esta tend ncia, embora o mesmo desenvolvimento potencie a modelizaç o numa escala cada vez maior (Stadtherr, 1997).

Marquardt (1996) tece consideraç es sobre os cuidados do desenvolvimento de modelos EO em ambiente multiutilizador. Nestas situaç es, a estruturaç o das simulaç es EO segundo um procedimento sistem tico   importante para a capacidade de reutilizaç o dos modelos criados (Lohmann e Marquardt, 1996). A manutenç o de bibliotecas de modelos   dif cil, assistindo-se normalmente a uma deterioraç o da qualidade do reposit rio com o passar do tempo (Marquardt, 1996).

2.5 Perspectivas

A aplicabilidade crescente dos m todos de optimizaç o   indissoci vel da evoluç o e desenvolvimento da capacidade de computa o dispon vel (Westerberg, 2004). Grossmann e Biegler (2004) afirmam que, no  mbito da evoluç o futura das ferramentas computacionais, espera-se que o desenvolvimento das linguagens orientadas por objectos, a utilizaç o da nano tecnologia de carbono na concepç o de chips e a integraç o

das metodologias existentes em plataformas compatíveis (numa filosofia do tipo CAPE-OPEN), possam alargar o alcance e eficácia das metodologias de optimização actuais.

Villiermaux (1996) já tinha anteriormente previsto a incorporação massiva das ferramentas computacionais no quotidiano dos engenheiros químicos. Hoje, Westerberg (2004) questiona até que ponto evolutivo poderemos chegar, sublinhando a ideia de que, sob o ponto de vista prático, com ciclos computacionais suficientes poderão avaliar-se todos os pontos de um domínio de procura.

Todavia, não se deve depender exclusivamente dos computadores. Grossmann (2004) aponta o caminho da sinergia de algoritmos como base de construção para estratégias mais robustas. Grossmann e Biegler (2004) dizem-se optimistas quanto à verificação das melhorias futuras esperadas, salientando a importância futura que terão aos conceitos de modelização baseada em lógica. Os mesmos autores estabelecem que a investigação mais exaustiva na área de optimização global, e em métodos destinados a optimização sujeita a incerteza, representam os principais desafios ao desenvolvimento futuro.

Sargent (2005) destaca a importância crescente que a aplicação de métodos de inteligência artificial terá na área da Biotecnologia, acrescentando ainda que a Engenharia de Sistemas de Processos deixará definitivamente de estar confinada ao espectro dos processos. Espera-se que cada vez mais se enverede pelo estudo e análise de problemas baseados em simulação molecular (Sargent, 2004).

Grossmann (2004) identifica a necessidade de promover um relacionamento mais estreito com a indústria, salientando a importância que deve atribuir-se à inclusão nos problemas das condicionantes que garantam sustentabilidade ambiental. Prevê-se assim que a preocupação de resolver modelos do mundo real, sujeitos a incerteza e caracterizados por variabilidade de escala, seja a tónica da força directriz do desenvolvimento da teoria da optimização (Grossmann, 2005; Varma et al., 2007).

Na revisão das perspectivas apresentadas por alguns autores para o desenvolvimento das metodologias de optimização, reserva-se lugar para uma referência adicional. Trata-se de uma linha de aplicação dos métodos de optimização que, devido à disponibilidade crescente de recursos computacionais, melhorará significativamente a robustez da abordagem a problemas de maior dificuldade: a optimização baseada em agentes.

2.5.1 Optimização baseada em agentes

A diversidade do tipo de problemas de optimização potencia e justifica o leque alargado de metodologias existentes com vista à resolução dos mesmos. Uma vez que cada um dos métodos de optimização disponíveis tende a explorar aspectos distintos de

cada um dos problemas, parece oportuno colocar a quest o sobre quais os potenciais benef cios de uma abordagem multimetodol gica com vista   optimiza o.

No contexto da Engenharia de Sistemas de Processos, Siirola et al. (2003) sugerem a abordagem a optimiza o n o convexa pela aplica o de uma estrat gia baseada na utiliza o de m ltiplos agentes. A metodologia parte do princ pio colaborativo que a distribui o do problema por entre rotinas computacionais independentes, designadas por agentes, se traduzir  na melhoria na taxa de sucesso da resolu o dos problemas.

Siirola et al. (2003) apresentam como ponto de partida o comportamento t pico das metodologias orientadas por equa es que utilizam uma s  m quina, sendo aplicadas de uma forma sequencial no ataque aos problemas (Figura 2.7). Deste modo, salientam a import ncia da paraleliza o no contexto em que os computadores pessoais de processador  nico s o o equipamento inform tico mais acess vel no mercado. Siirola et al. (2003) apresentam assim uma abordagem baseada em agentes, a qual opera sustentada por c digo paralelizado que corre em *clusters*. Essa abordagem   representada pelo esquema da Figura 2.8 (Siirola e Hauan, 2007).

Verifica-se assim que o principal objectivo passa pela combina o de diferentes metodologias na resolu o do problema. Pretende-se aproveitar o comportamento do grupo de agentes, que mimetiza sistemas biol gicos onde as ac es individuais se diluem na

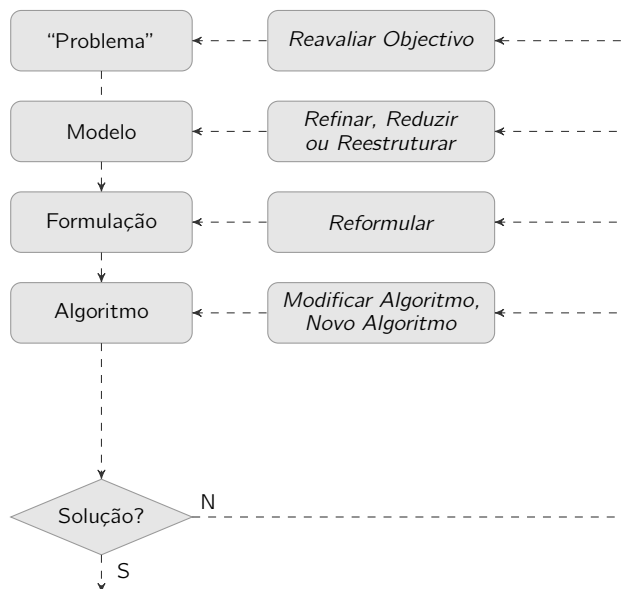


Figura 2.7 Sistema de optimiza o sequencial (adaptado de Siirola e Hauan (2007)).

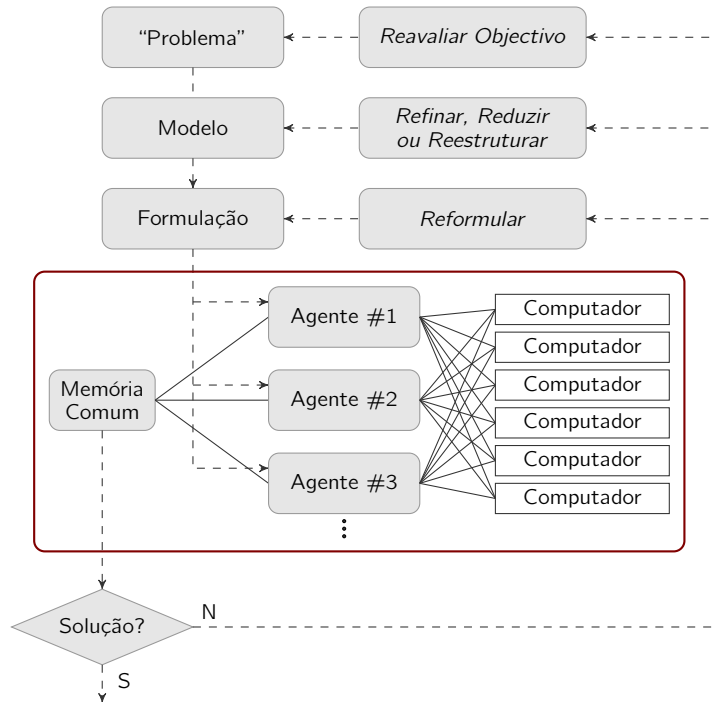


Figura 2.8 Sistema de otimização baseado em agentes (adaptado de Siirola e Hauan (2007)).

obtenção de um resultado colectivo de qualidade superior. Esta metodologia apresenta-se assim como uma plataforma adequada à abordagem de problemas de optimização ainda não caracterizados (Siirola e Hauan, 2007).

No contexto da optimização multiagente, convém referir que o estabelecimento de uma plataforma deste tipo se adequaria à aplicação conjugada de métodos estocásticos de optimização. A sua capacidade em lidar com modelos do tipo caixa-negra permite implementar, de uma forma acessível, metodologias de optimização baseadas numa procura multi-heurística.

Todavia, este tipo de abordagem pode ainda incorporar uma multiplicidade mais alargada. A evolução da aplicação do conceito de multiagente em optimização não linear foi já apresentada por Siirola e Hauan (2007). Os autores justificam-na pela dependência que determinados métodos de optimização revelam face à colocação do problema. Assumindo uma futura massificação da computação paralelizada, estes autores introduzem variabilidade ao nível da formulação do problema no esquema apresentado na Figura 2.8. O resultado é uma metodologia designada por optimização polimórfica, representada na

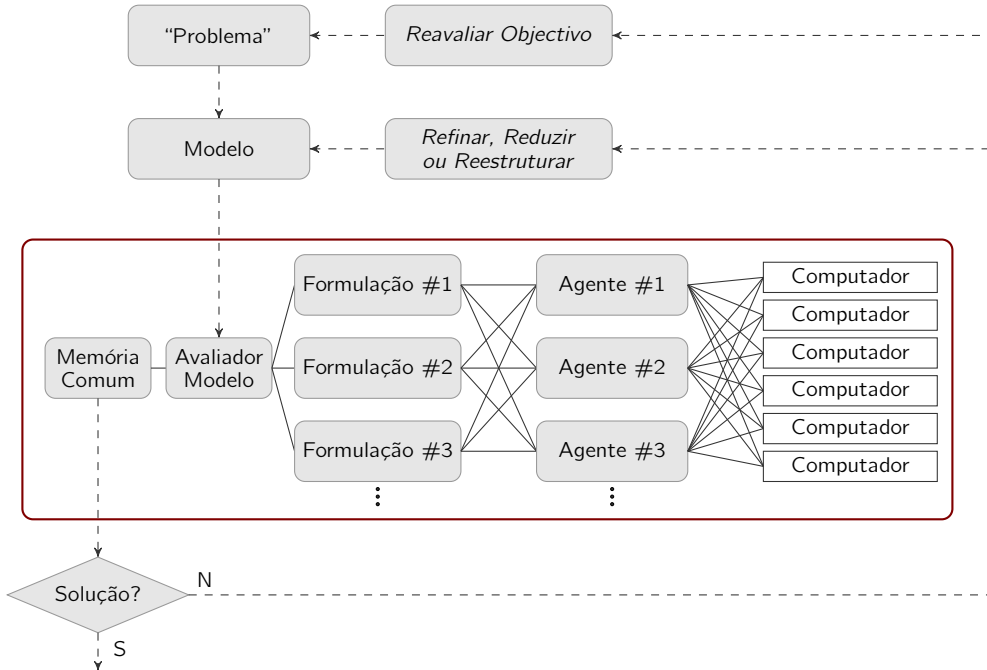


Figura 2.9 Sistema de optimizaç o polim rfica baseada na acç o de multiagentes (adaptado de Sirola e Huan (2007)).

Figura 2.9, e que revela um car ter plural n o s o nos m todos de resoluç o do problema, mas tamb m no que respeita ao seu enunciado.

A abordagem baseada em agentes e a sua extens o na forma polim rfica representam ideias emergentes na aplicaç o da teoria da optimizaç o. Em analogia com a impuls o que a filosofia de Leibniz imprimiu na import ncia do * ptimo* nos contextos social e cient fico, talvez seja hoje a vez do contexto social influenciar a evoluç o da teoria da optimizaç o. As metodologias baseadas na cooperaç o de multiagentes parecem beber inspiraç o do hoje t o discutido conceito de intelig ncia colectiva. Associado anteriormente aos resultados encontrados em meios como as redes que entrelaam parcerias entre grupos de investigaç o cient fica, o fen meno da intelig ncia colectiva apresenta hoje uma import ncia transversal. A mesma   reconhecida em diversas  reas (Surowiecki, 2005; Tapscott e Williams, 2008), e est  na de g nese de novos comportamentos sociais promovidos por meios como a Web 2.0 (O'Reilly, 2005), permanecendo enraizada em comunidades cooperativas, com   o caso, por exemplo, do modelo de desenvolvimento de aplicaç es de c digo aberto.

Nomenclatura

Caracteres romanos

$V_E(\mathbf{x}^*)$	Vizinhança esférica em torno de \mathbf{x}^*
E_i	Nível de energia i no algoritmo de recozimento simulado
EUA	Estados Unidos da América
F	Valor da Função Objectivo
G	Valor da Função Objectivo em optimização sujeita a incerteza.
\mathbf{g}	Vector das restrições de desigualdade
g_j	Restrição de desigualdade de índice j no vector \mathbf{g}
\mathbf{h}	Vector das restrições de igualdade
h_i	Restrição de igualdade de índice i no vector \mathbf{h}
K_B	Constante de Boltzmann
M	Constante M
m_1	Número de restrições de igualdade do problema de optimização
m_2	Número de restrições de desigualdade do problema de optimização
n	Número de variáveis do problema de optimização
\mathbf{p}	Vector de parâmetros
P	Probabilidade de transição de nível no recozimento simulado
S	Região plausível
T	Temperatura
t	Tempo
t_f	Tempo final
t_i	Tempo inicial
t_0	Tempo nas condições iniciais
\mathbf{u}	Vector dos perfis das variáveis de controlo
\mathbf{v}	Vector dos perfis algébrico das variáveis de estado
\mathbf{w}	Vectores dos factores de incerteza
\mathbf{x}	Vector das variáveis contínuas
x_k	Variável contínua de índice k do vector \mathbf{x}
x_i	Variável genérica de índice i
\mathbf{x}_i	Vector das variáveis genéricas de índice i
\mathbf{x}^*	Vector das coordenadas de ponto mínimo
\mathbf{y}	Vector das variáveis discretas
y_k	Variável contínua de índice k do vector \mathbf{y}
y_1	Variável binária de índice i
Y_1	Variável lógica de índice i
\mathbf{z}	Vector dos perfis diferenciais das variáveis de estado
\mathbf{z}_0	Vector dos valores das variáveis \mathbf{z} no instante inicial

Expoentes

I	Limite inferior
S	Limite superior

Caracteres gregos

α_k	Limite inferior de x_k
β_k	Limite superior de x_k
γ_k	Limite inferior de y_k
ξ_k	Limite superior de y_k
α	Partiç�o aleat�ria do intervalo a, b
ΔE	Variaç�o do n�vel de energia no recozimento simulado
ε	Toler�ncia num�rica para converg�ncia

Acr nimos

ASPEN	Advanced System for Process Engineering
BARON	Branch and Reduce Optimization Navigator
BB	Branch and Bound
CAPE-OPEN	Computer Aided Process Engineering - OPEN
DP	Disjunctive Programming
FA	Feasible Approximation
CRS	Controlled Random Search
GAMS	General Algebraic Modelling System
GBD	Generalized Benders Decomposition
GCD	Generalized Cross Decomposition
GDP	Generalized Disjunctive Programming
GO	Global Optimization
GOA	Generalized Outer Approximation
GRG	Generalized Reduced Gradient
GSA	Generalized Simulated Annealing
ECP	Extended Cutting Plane
ENIAC	Electrical Numerical Integrator and Calculator
EO	Equation Oriented
FORTTRAN	Formula Translator
GRG	Generalized Reduced Gradient
IP	Integer Programming
KKT	Karush-Kuhn-Tucker
LCP	Linear Complementary Programming
LP	Linear Programming

NLP	Nonlinear Programming
MILP	Mixed-Integer Linear Programming
MINLP	Mixed-Integer Nonlinear Programming
MIP	Mixed Integer Programming
MIT	Massachusetts Institute of Technology
MSGA	Minlp Salcedo Gonçalves Azevedo
MSIMPSA	Minlp SIMPLex Simulated Annealing
OA	Outer Approximation
OA/ER	Outer Approximation
OA/ER/AP	Outer Approximation
QP	Quadratic Programming
RS	Random Search
SeqM	Sequential Modular
SimM	Simultaneous Modular
SLP	Sequential Linear Programming
SQP	Sequential Quadratic Programming
rSQP	Recursive Sequential Quadratic Programming
SA	Simulated Annealing
SP	Semidefinite Programming
SQP	Successive Quadratic Programming

Capítulo 3

O ambiente ASCEND

Sumário do Capítulo

Partindo do seu contexto histórico, apresenta-se o ambiente de simulação ASCEND. Com base num exemplo de aplicação simples são evidenciados os principais traços do simulador, apresentando-se as estruturas da linguagem de modelização, as formas e os procedimentos utilizados para criar e resolver uma simulação. São ainda apresentados os modos de operação da interface gráfica disponível.

3.1 Nota Histórica

O acrónimo ASCEND corresponde à designação para Advanced System for Computations in ENgineering Design (Locke e Westerberg, 1983; Piela et al., 1991). O projecto nasce pela mão do Professor Arthur Westerberg¹, em 1978, na Universidade de Carnegie Mellon (Pittsburgh, EUA). Trata-se de um ambiente de simulação orientado por equações dotado de uma linguagem de modelização própria.

Os pilares do ASCEND são de natureza académica. As suas concepção e desenvolvimento foram promovidos não só por trabalhos de estudantes não graduados, mas também pelos trabalhos de doutoramento materializados nas teses de Kuru (1981), Piela (1989), Huss (1995), Zaher (1995), Abbott (1996), Safrit (1996), Thomas (1996), Allan (1997) e Rico-Ramirez (1998). Estes registos atestam o grande entusiasmo que rodeou este projecto, tendo influenciado directamente a sua concepção.

Actualmente o ASCEND já se encontra na sua versão IV. Até ao ano 2000 a sua evolução passou por quatro etapas distintas, designadas respectivamente por: I, II, III (e

¹Página pessoal: <http://www.cheme.cmu.edu/people/faculty/aw0a.htm>

IIIc) e IV. No âmbito da presente tese foi utilizada a versão IV release 0.9.1, disponível no sítio dedicado ao projecto (ASCEND Developers, 2008a), e que é ainda hoje considerada a versão estável do ambiente de simulação. Todavia, o projecto ASCEND não ficou por este estágio, voltando a reclamar atenção no período após a jubilação do Professor Westerberg. Hoje encontra-se na fase a que os seus programadores chamam de estágio pós-2004, onde um conjunto de iniciativas relacionadas com o simulador permitiram que voltasse a ter maior visibilidade por intermédio da inauguração de um *wiki* destinado à sua discussão. Informações suplementares sobre a história desta aplicação podem encontrar-se na respectiva secção do seu *wiki* (ASCEND Developers, 2008c).

O ASCEND dá o seu grande salto evolutivo na concepção da versão III, da qual se salienta o aparecimento de uma linguagem de construção de modelos bem definida, assim como o de uma nova interface interactiva para o utilizador. Depois da sua tradução para a linguagem C (versão IIIc), a versão IV já permite a inclusão de notas nos modelos de simulação, e atribui importância significativa à possibilidade de reutilizar modelos já construídos anteriormente, promovendo o intercâmbio entre utilizadores.

A equipa responsável por esta aplicação apelida o ASCEND de ambiente de modelização matemática portátil (Westerberg et al., 1998), traduzindo assim o seu carácter versátil e flexível, de grande utilidade na resolução de problemas de dimensão diversa.

3.2 Constituição do ASCEND e suas principais características

Para melhor descrever realmente o que é o sistema ASCEND citando aqueles que o desenvolveram, trata-se de um sistema de modelização concebido para permitir que os utilizadores possam construir, analisar e corrigir, solucionar ou otimizar modelos descritos por conjuntos de equações algébricas não lineares de dimensão considerável (Westerberg et al., 1998).

Uma das vantagens que lhe são apontadas tem que ver com a sua filosofia de código fonte aberto. O ASCEND pode ser obtido juntamente com o seu código fonte, de forma gratuita, podendo ser utilizado em diversas plataformas e até modificado para atender às necessidades de cada utilizador, existindo já alguns utilizadores no universo científico (ASCEND Developers, 2008d; Maltez e Oliveira, 2001).

O sistema ASCEND é constituído por três partes fundamentais:

1. A linguagem de modelização;
2. O ambiente interactivo que permite aos utilizadores compilar e executar modelos.
3. O conjunto de *solvers* e optimizadores que o próprio sistema pode incorporar.

Relativamente à linguagem de modelização, esta tese não pretende substituir os manuais da aplicação, dispensando por isso uma abordagem exaustiva da sintaxe da mesma. Na Secção 3.3 apresenta-se um problema de carácter exemplificativo, pretexto para a inclusão de comentários adicionais que se julguem necessários à boa compreensão dos conceitos até aqui expostos.

A interface que o sistema apresenta ao utilizador segue uma filosofia de organização baseada em janelas flutuantes. Pela sua importância na gestão de modelos, cinco merecem lugar de destaque, apresentando-se uma descrição sucinta das suas principais características:

- **LIBRARY** — Interface para gestão de modelos e simulações. Através dos seus comandos podem ser dadas instruções para o carregamento e compilação de modelos com vista à denominação e criação de simulações.
- **BROWSER** — Interface para navegação estrutural. Permite a gestão de todas as funcionalidades associadas à consulta e/ou modificação dos valores dos diferentes atributos dos objectos da simulação. A janela **BROWSER** é o meio a que o utilizador recorre para identificar as relações que existem entre os objectos das simulações.
- **SOLVER** — Interface para execução e resolução. Disponibiliza informação estatística sobre a estrutura do problema em termos dos seus sistemas de equações. Possibilita o visionamento de informação gráfica como a matriz de incidências do problema. Oferece opções de escolha e configuração do algoritmo a utilizar para resolver os sistemas de equações.
- **SCRIPT** — Interface de comando do ambiente de simulação. Apresenta as opções que permitem invocar e modificar as configurações base de todas as outras funcionalidades gráficas. Este painel pode ser considerado o menu principal. Está dotado de um interpretador de *scripting* embutido a que o utilizador pode recorrer para automatizar sequências de acções. Podem ainda ser gravadas para *script*, e de forma automática, as instruções resultantes da interacção do utilizador com outras janelas do ambiente gráfico.
- **CONSOLE** — Interface para monitorização das mensagens do sistema. Através deste painel o utilizador toma conhecimento dos alertas e mensagens emitidos pelo simulador quando as acções são concluídas e/ou ocorrem erros no seu processamento. Assume um papel essencial na depuração dos modelos implementados.

Westerberg et al. (1998) dão conhecimento de que o conjunto de *solvers* do sistema ASCEND contém um vasto conjunto de opções. A abordagem aos casos de estudo apresentados na presente tese recorreu ao *solver* que integra a aplicação base por omissão, designadamente o QRS1v (Westerberg, 1979; Westerberg et al., 1998). Actualmente, no estágio pós-2004, a equipa de desenvolvimento do ASCEND pretende alargar a lista de *solvers* que possam interagir com o simulador. Para além do CONOPT (Drud, 2008, 1994) que já anteriormente era utilizado para dar suporte a problemas de optimização, já foi demonstrada intenção de incorporar diferentes abordagens com base em algoritmos para a resolução de sistemas de equações algébricas lineares (LA), não lineares (NLA), assim como para equações diferenciais ordinárias (ODE) e algébrico-diferenciais (DAE). Destaca-se ainda o caso da optimização não linear (NLP), pela implementação do IPOPT (Wachter e Biegler, 2006) como ferramenta de optimização não proprietária (ASCEND Developers, 2008b).

Esta aplicação de simulação envolve um certo grau de complexidade inicial que, segundo os seus autores, só consegue ser ultrapassado se for compreendida a filosofia do seu funcionamento. Os modelos ASCEND são ficheiros de texto simples em formato ASCII, pelo que não é necessário recorrer a aplicações específicas para a sua edição, podendo o utilizador escolher a ferramenta que lhe garanta a maior produtividade. No desenvolvimento e sedimentação da sua experiência na construção de modelos e simulações de problemas, o utilizador terá de lidar essencialmente com três categorias de informação, as quais estão respectivamente associadas às seguintes três extensões de ficheiro:

- *.a4c Modelos — Ficheiros em linguagem de modelização ASCEND que contêm a descrição do problema. São os ficheiros que contêm elementos como equações, variáveis, dados, etc.
- *.a4l Bibliotecas — Correspondem a um conjunto alargado de definições simbólicas que integra o sistema. A sua incorporação nos modelos torna possível recorrer aos elementos necessários para a boa resolução dos mesmos. Dotam o utilizador do léxico base com que é possível dispor as variáveis do modelo por entre diferentes tipos matemáticos e/ou físicos. Em função das exigências apresentadas pelos problemas a resolver, o utilizador dispõe da capacidade de acrescentar elementos criados por si ao conjunto disponibilizado pela biblioteca.
- *.a4s Scripts — Visam codificar a execução automática de tarefas. A sua principal função passa pela capacidade de sistematização, baseada na codificação dos diferentes procedimentos disponíveis no ambiente de simulação. Tarefas habituais e/ou

obrigatórias à construção da simulação podem assim ser conseguidas de forma imediata, permitindo ganhos ao nível da reusabilidade e do tempo despendido.

A revisão das principais características do ambiente de simulação ASCEND exige ainda a referência de um aspecto de importância fundamental. Segundo a filosofia dos criadores do projecto, a definição das unidades de cada variável, isto é, a definição do seu tipo no contexto dimensional. Procura-se com isso, atribuir significado físico aos elementos dos modelos, criando condições para uma depuração mais eficiente e também para uma análise mais clara e acessível dos modelos à luz da realidade física do problema em estudo.

Considerando esta tarefa como *atómica* para a boa escrita, análise e resolução dos problemas (ASCEND Development Team, 1998), o sistema contém uma biblioteca designada por `atoms.a41`. A mesma, por intermédio de ligações a outras bibliotecas que definem e refinam tipos de variáveis, disponibiliza um conjunto diversificado de grandezas físicas que em número ultrapassa as 150 definições diferentes.

Os traços fortes que caracterizam o ambiente de simulação ASCEND estão apresentados. Para uma melhor e mais intuitiva compreensão do seu funcionamento recorre-se a um exemplo detalhado na Secção 3.3.

3.3 Exemplo de aplicação

O processo escolhido como exemplo demonstrativo de aplicação do ASCEND está representado pelo diagrama processual esquematizado na Figura 3.1. O problema, aqui designado por P.MK (II.b em Salcedo (2001)), corresponde a um processo químico em estado estacionário onde, num reactor perfeitamente agitado (RPA), ocorrem duas reacções consecutivas, que ocorrem competitivamente em paralelo — designadas por r_1 e r_2 (Equação 3.1).



Neste processo o produto B é desejado. A mistura obtida à saída do RPA (corrente F_P) é separada em três correntes puras de A , B e C (designadas por F_R , F_B e F_C respectivamente). O reagente remanescente é reciclado, na totalidade, para constituir, juntamente com a corrente de *make-up* de reagente (F_0), a alimentação do rector, F_1 , que é pura no componente A , não existindo acumulação de massa no RPA ($F_P = F_1$).

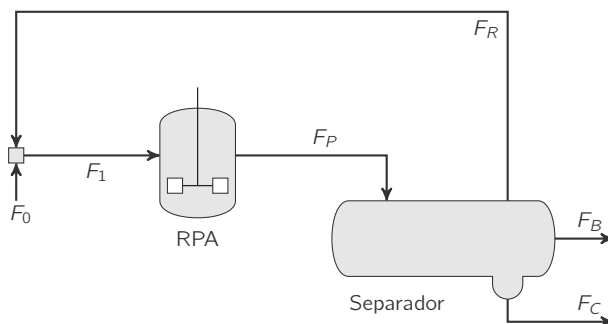


Figura 3.1 Diagrama Processual do Problema P.MK.

As reacções são de primeira ordem (Equações 3.2 e 3.3) e as respectivas constantes cinéticas são traduzidas pela lei de Arrhenius (Equação 3.4),

$$r_1 = -k_1 C_A \quad (3.2)$$

$$r_2 = -k_2 C_B \quad (3.3)$$

$$k_i = a \exp\left(-\frac{E_i}{RT}\right) \quad (3.4)$$

onde k_i é a constante cinética da reacção r_i , a é o chamado factor de frequência (ou pré-exponencial), enquanto E corresponde à energia de activação da reacção respectiva. Os valores destes parâmetros para as reacções r_1 e r_2 encontram-se especificados na Tabela 3.1. A determinação de k depende ainda da temperatura, T , enquanto que a lei de velocidade de cada reacção é função da concentração existente do respectivo reagente (C_A ou C_B).

Relativamente às condições de operação do processo considera-se que, ao nível do reactor, os calores de reacção são desprezáveis proporcionando condições isotérmicas. Assume-se ainda que o volume molar no RPA pode ser expresso por intermédio do volume molar do reagente A, pelo que, atendendo a estas simplificações, o processo P.MK poderá

Tabela 3.1 Problema P.MK. Parâmetros cinéticos.

Reacção	a (h^{-1})	E/R (K)
r_1	$4.38 \cdot 10^{11}$	9000
r_2	$6.09 \cdot 10^{11}$	15000

ser descrito pelo conjunto de Equações 3.5 a 3.10:

$$F_1 = F_0 + F_R \quad (3.5)$$

$$F_1 = F_1 X_{AP} + V_{mol} X_{AP} a_1 \exp\left(-\frac{E_1}{RT}\right) \quad (3.6)$$

$$V_{mol} = \frac{V \rho_A}{MM_A} \quad (3.7)$$

$$V_{mol} X_{AP} a_1 \exp\left(-\frac{E_1}{RT}\right) = F_1 X_{BP} + V_{mol} X_{BP} a_2 \exp\left(-\frac{E_2}{RT}\right) \quad (3.8)$$

$$F_R = F_1 X_{AP} \quad (3.9)$$

$$F_B = F_1 X_{BP} \quad (3.10)$$

onde V_{mol} e V são, respectivamente, os volumes molar e métrico do reactor enquanto que X_{AP} e X_{BP} correspondem às respectivas fracções molares dos componentes A e B na corrente F_P . Os restantes dados do problema, nomeadamente os valores de propriedades físicas e coeficientes de custo, estão especificados na Tabela 3.2. A Tabela 3.3 apresenta as variáveis do problema, um conjunto com a dimensão de sete elementos, o que confere ao problema apenas um grau de liberdade.

A Equação 3.11 corresponde à expressão para o cálculo da função objectivo, que neste caso é definida em termos da rentabilidade económica do processo. A mesma traduz a diferença entre os ganhos por venda do produto B com valor de mercado Val_B e os custos de produção associados, no caso: os custos de operação para a reciclagem

Tabela 3.2 Dados do problema P.MK: Valores e nomenclatura. Representações nos contextos da tese e do modelo ASCEND.

Dado do Problema	Tese	ASCEND	Valor
Massa molar de A	MM_A	MM_A	78.0 kg/kmol
Massa específica de A	ρ_A	ro_A	880.0 kg/m ³
Massa molar de B	MM_B	MM_B	112.5 kg/kmol
Volume do reactor	V	v	1.0 m ³
Custo de aluguer do reactor	Cst_{RPA}	Cst_RPA	50.0 \$/m ³ /h
Custo de aquisição de A	Cst_A	Cst_A	5.0 \$/kg
Custo de reciclagem	Cst_R	Cst_R	3.0 \$/kg
Valor de venda de B	Val_B	Val_B	10.0 \$/kg
Temperatura	T	T	300.0 K

Tabela 3.3 Variáveis do Problema P.MK. Nomenclatura e suas representações nos contextos da tese e do modelo ASCEND.

Variável	Tese	ASCEND
Caudal molar de F_0 (kmol/h)	F_0	F0
Caudal molar de F_1 (kmol/h)	F_1	F1
Caudal molar de F_R (kmol/h)	F_R	FR
Caudal molar de F_B (kmol/h)	F_B	FB
Fracção molar de A em F_P	X_{AP}	XAP
Fracção molar de B em F_P	X_{BP}	XBP
Capacidade molar do reactor (mol)	V_{mol}	Vmol

de reagente A, Cst_R ; os custos de investimento no reactor, Cst_{RPA} ; e os custos de aquisição do reagente A, Cst_A .

$$FOB = F_B MM_B Val_B - (Cst_R F_R + Cst_A F_0) MM_A - Cst_{RPA} V \quad (3.11)$$

A descrição do exemplo P.MK fica completa com a definição da função objectivo. Atendendo às variáveis apresentadas na Tabela 3.3 e à definição do problema pelas Equações 3.5 a 3.10, sabe-se desde já que o problema apresenta um grau de liberdade ($GL = n - m_1 = 7 - 6 = 1$). O primeiro passo, o de tradução do problema na forma de equações algébricas, está assim concluído. O próximo passo passa pela sua escrita na forma de linguagem de modelização ASCEND.

3.4 Codificação e análise do modelo para ASCEND

Com base no exemplo apresentado, pretende-se expor algumas das capacidades do ASCEND enquanto simulador. Nesse sentido, os diferentes procedimentos serão ilustrados com capturas de ecrã da interface do simulador sempre que se considerar apropriado. A criação de um modelo para ASCEND é uma tarefa que pode apresentar um grau de sofisticação diverso consoante o utilizador. Quadros de código fonte serão incluídos com o intuito de evidenciar a estrutura organizacional de modelos e *scripts*, permitindo igualmente usá-los como exemplos da linguagem de modelização.

3.4.1 Definição do modelo base

Inicialmente, e tendo apenas como preocupação exclusiva de construir um modelo que traduza a formulação do problema apresentada na Secção 3.3, o modelo para ASCEND

Quadro 3.1 Problema P.MK. Modelo básico para ASCEND.

```

1: REQUIRE "atoms.a41";
2: MODEL make_up;
3:   T, A1, A2, ER1, ER2, MM_A, MM_B, ro_A   IS_A real_constant;
4:   V, Cst_RPA, Val_B, Cst_R, Cst_A         IS_A real_constant;
5:   F0, F1, FR, FB                          IS_A molar_rate;
6:   XAp, XBp                                IS_A mole_fraction;
7:   Vmol                                    IS_A mole;
8:   FOB                                     IS_A cost_per_time;
9:
10:  eq_01:  F1 = F0 + FR;
11:  eq_02:  F1 = F1*XAp + Vmol*A1*XAp*exp(-ER1/T);
12:  eq_03:  Vmol = V*ro_A/MM_A;
13:  eq_04:  Vmol*A1*XAp*exp(-ER1/T) = F1*XBp + Vmol*A2*XBp*exp(-ER2/T);
14:  eq_05:  FR = F1*XAp;
15:  eq_06:  FB = F1*XBp;
16:  eq_FOB: FOB = FB*MM_B*Val_B - (Cst_R*FR + Cst_A*F0)*MM_A - Cst_RPA*V;
17: END make_up;

```

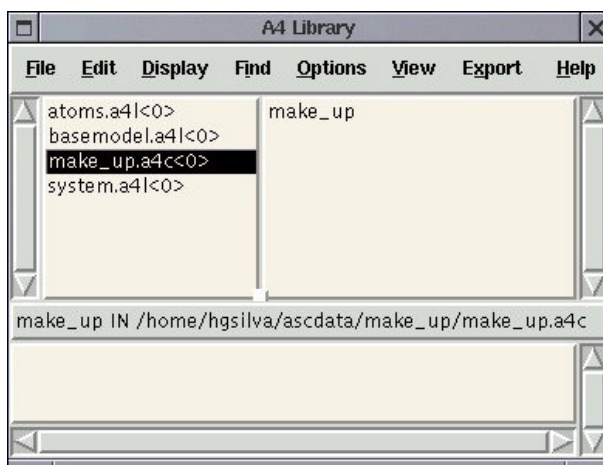
do problema P.MK será certamente semelhante ao apresentado no Quadro 3.1, aqui designado por `make_up`. Apenas são indicadas as informações essenciais à constituição de simulação, como a definição de constantes, variáveis e equações do problema.

Analisando o código apresentado no Quadro 3.1, evidencia-se que, em primeiro lugar, o código é iniciado com a instrução `REQUIRE`, que permite carregar no sistema, as bibliotecas de apoio à resolução do problema. No que diz respeito às instruções `MODEL` e `IS_A`, estas são respectivamente utilizadas para definir o nome do modelo criado e para descrever o tipo associado a cada uma das variáveis especificadas enquanto que a instrução `END` é usada para finalizar o código do modelo. Uma vez que a codificação segue uma filosofia orientada por objectos, as equações que aparecem no modelo são identificadas através de uma designação estilo etiqueta que surge no início da linha onde está escrita a equação correspondente.

3.4.2 Iniciação à interface gráfica

Com um modelo devidamente codificado, o utilizador aborda a etapa seguinte, em que recorre à interface do ASCEND para processar o modelo escrito. A primeira tarefa a efectuar é o carregamento do mesmo no sistema, através da janela `LIBRARY`. Desde que não se verifiquem erros que obriguem a depuração, a simulação poderá ser criada e o utilizador, tal como se mostra na Figura 3.2, terá disponíveis diversas informações.

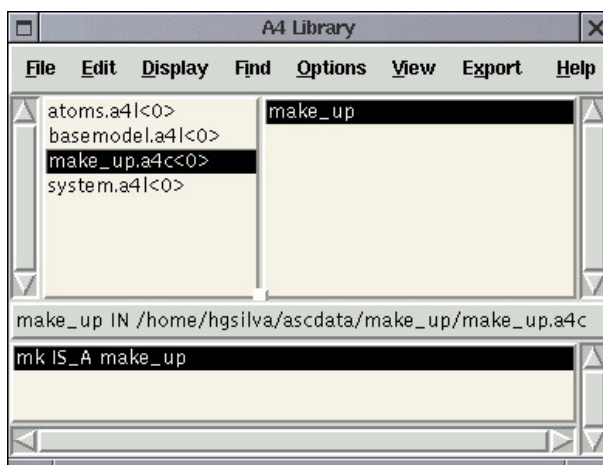
No painel superior, o lado esquerdo apresenta os nomes das bibliotecas carregadas pelo sistema enquanto que, na secção direita, surgem os nomes dos modelos incluídos no ficheiro seleccionado no momento (o utilizador pode, se assim desejar, codificar mais



(a) Lista dos ficheiros carregados.



(b) Criação da simulação mk. O cadeado na janela de indica que a mesma requer acção imediata por parte do utilizador.



(c) Lista dos ficheiros carregados e simulações compiladas.

Figura 3.2 Janela LIBRARY. Problema P.MK. Painéis informativos durante a importação do ficheiro make_up.a4c e criação da simulação mk.

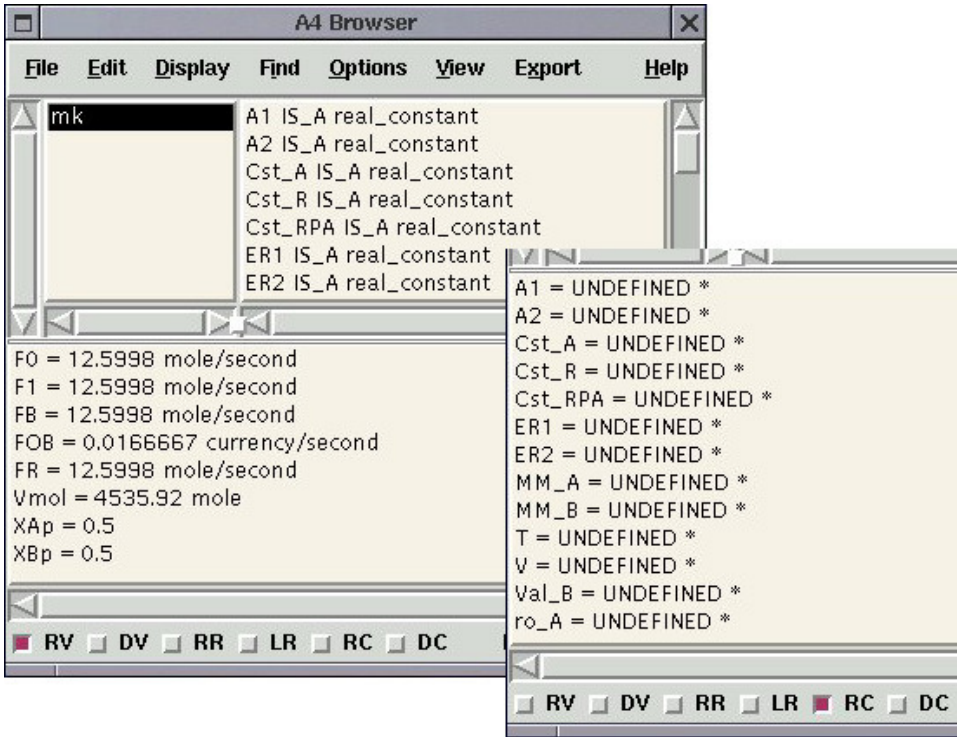


Figura 3.3 Aspecto da janela **BROWSER** após exportação da simulação *mk* a partir da janela **LIBRARY**. As variáveis apresentam os valores que o ambiente lhes atribui por omissão enquanto que as constantes permanecem indefinidas. O painel inferior possibilita a filtragem imediata de determinados tipos de elementos: variáveis reais (RV) e discretas (DV); relações reais (RR) e discretas (LR); constantes reais (RC) e discretas (DC).

do que um modelo por ficheiro). Na secção inferior aparecem ainda informações sobre o caminho de localização do ficheiro bem como a lista de correspondência entre as simulações criadas e os modelos que lhes servem de base.

Compilada, a simulação é agora uma estrutura relacional com elementos dotados de atributos caracterizáveis. As tarefas de atribuição de propriedades por modificação do valor dos atributos é conduzida pela navegação do modelo através da janela **BROWSER** (Figura 3.3). Onde o utilizador faz de uma navegação orientada por objectos para poder consultar informações relacionadas associadas a cada elemento.

Como se verifica através da Figura 3.3, são permanentemente disponibilizadas informações sobre as variáveis, nomeadamente qual o seu tipo, podendo ser consultadas, no painel inferior, listas de diferentes tipos. Atendendo a que, nesta altura, apenas se considerou a definição simbólica dos objectivos do modelo, o utilizador poderá de seguida

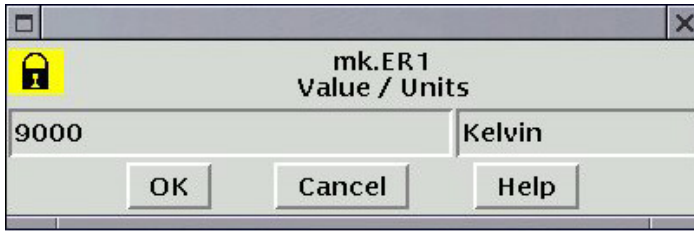


Figura 3.4 Janela **BROWSER**. Problema P.MK. Atribuição de valor numérico ao quociente E_1/R , 9000K.

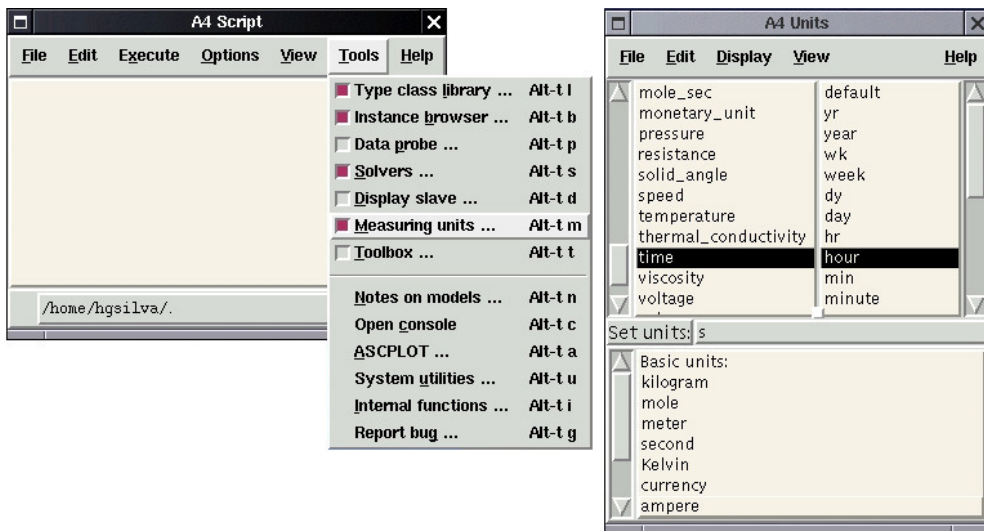


Figura 3.5 Aspecto geral da janela *Units*, onde o utilizador pode escolher as unidades de trabalho para os diversos tipos de variáveis.

atribuir os valores correctos em função dos dados que possui. A Figura 3.4 exemplifica a situação para o valor do parâmetro E_1/R .

A análise das representações no **BROWSER** serve de pretexto para evidenciar que os diferentes elementos da simulação estão sempre associados à sua natureza dimensional. O utilizador pode configurar a base de medida em que as diferentes unidades são apresentadas, podendo para isso lançar o painel **UNITS** a partir da janela **SCRIPT**. Por exemplo, para o problema P.MK, em que os dados estão na base horária, o utilizador poderá ajustar a representação das variáveis e constantes seguindo o procedimento simples ilustrado na Figura 3.5.

O utilizador é livre de definir os sistemas de unidades em que pretende basear o método de obtenção dos resultados que procura. Tais definições podem ser modificadas

em qualquer momento, podendo novas preferências serem adoptadas para efeitos de comparação com resultados existentes em sistemas de unidades diversos.

A Figura 3.6 mostra o caso particular dos atributos da variável X_{AP} , de entre os quais se salientam os limites superior e inferior do domínio da variável, o seu valor nominal (utilizado como orientação nas operações de escalonamento) e ainda o atributo lógico $X_{Ap}.fixed$. Este último é utilizado para definir se a variável integra ou não o conjunto de decisão do problema, situações a que respectivamente correspondem os valores de verdadeiro ou falso. Retomando o modelo `make_up` para o problema `P.MK`, a Figura 3.7 apresenta a informação do `BROWSER` após actualização da representação para base horária e atribuição de todos os valores relativos às constantes.

A definição de todos os atributos das variáveis permite assim especificar o conjunto decisão, os valores dos parâmetros e as estimativas iniciais das variáveis. Nesta fase a simulação poderá ser exportada para o `SOLVER`. Nesse caso, uma simulação só poderá ser resolvida quando o número de variáveis do problema igualar o número de equações, situação em que o número de graus de liberdade é nulo. Como se pode verificar na Figura 3.8, onde se apresenta o quadro estatístico relativo à simulação criada, para este problema não foi fixada ainda nenhuma variável até ao momento, pelo que o `Solver` dirá que o problema apresenta oito variáveis livres (o ASCEND contabiliza uma variável extra relativamente ao número que tinha sido apresentado na Secção 3.3. A função objectivo assim como a equação que a define são incorporadas no sistema de equações do processo).

Caso o utilizador coloque no `SOLVER` uma simulação que não tenha todos os graus de liberdade especificados, o simulador responderá com uma interpelação que visa colmatar esse lapso. Tal pormenor é evidenciado na Figura 3.8, onde o `SOLVER` apresenta uma lista de possíveis variáveis elegíveis para preencher o grau de liberdade do problema. A mesma é actualizada após cada selecção, sendo eliminadas não só a variável escolhida, como também as que deixem de ser passíveis de escolha em consequência da selecção precedente. Esta análise estrutural acontece a cada momento. Veja-se, por exemplo, que o facto da variável V_{mol} pertencer apenas à Equação 3.7 condiciona a sua exclusão da lista apresentada pelo `SOLVER` na Figura 3.8. Uma variável com incidência única é obrigatoriamente uma variável de estado.

Para este exemplo será escolhida a variável X_{AP} como variável de decisão. Tal acção permite que o `SOLVER` actualize o seu quadro estatístico, que passará a informar que o problema se encontra bem colocado (expressão usada pela equipa que desenvolve o projecto). Usa-se tal designação para nomear o estado determinado do sistema de equações, terminologia alusiva a uma matriz de incidências quadrada em que o número

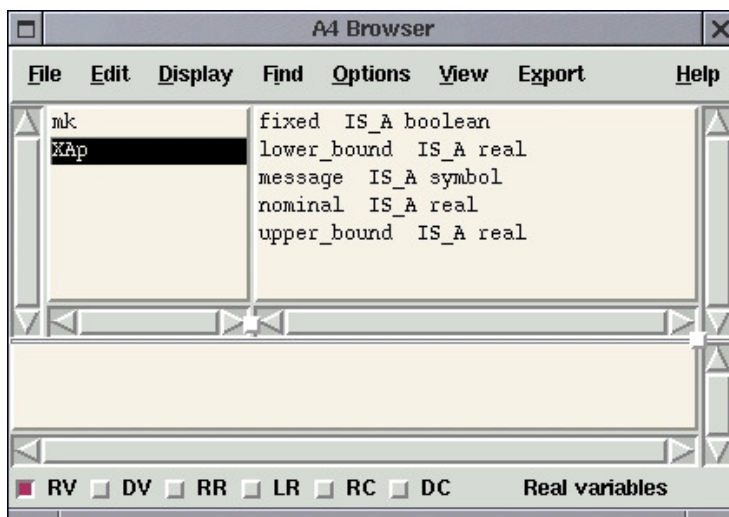


Figura 3.6 Janela *Browser* - Apresentação dos atributos da variável X_{Ap} .

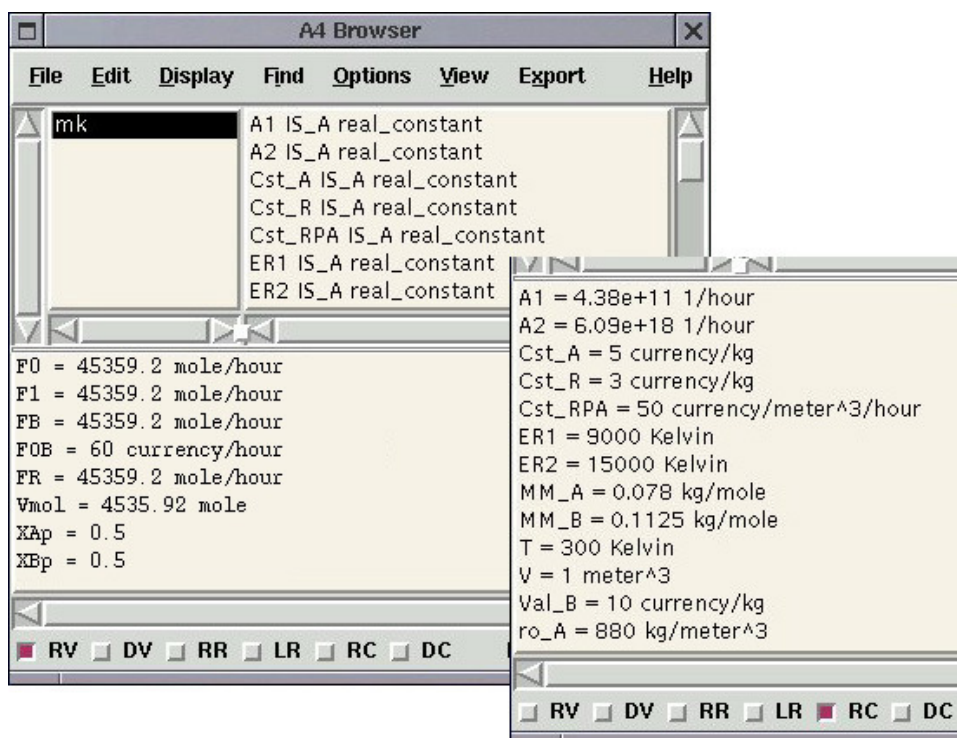


Figura 3.7 Janela *BROWSER*. Problema P.MK. Painéis informativos mostram os valores das constantes do modelo bem como das estimativas iniciais das variáveis.

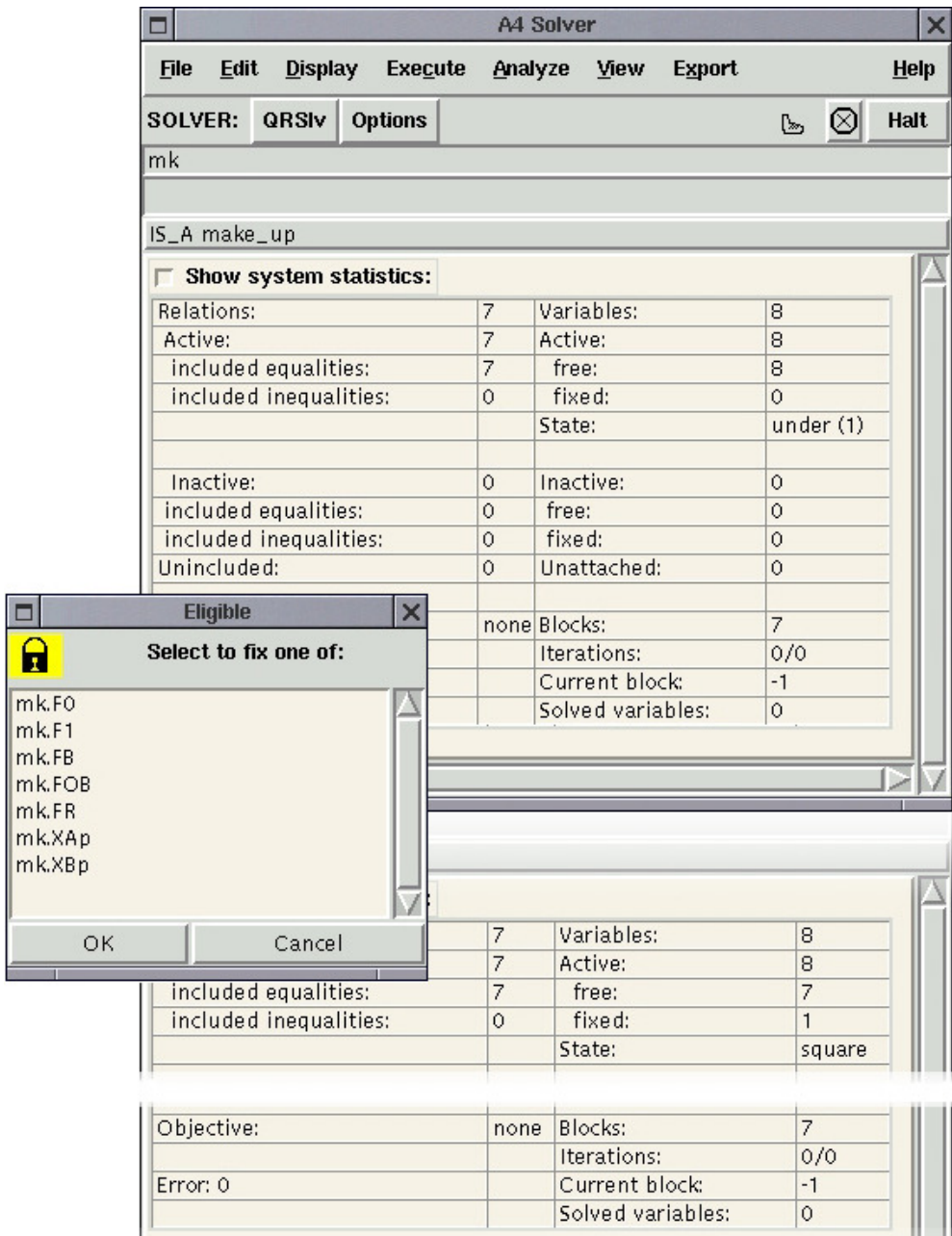


Figura 3.8 Janela SOLVER. Problema P.MK. Estatísticas sobre a estrutura do problema antes e após especificação das variáveis que preenchem os graus de liberdade.

Quadro 3.2 Problema P.MK. Modelo `make_up`. Resultados para $X_{AP} = 0.5$.

F0 = 231.205 mole/hour
F1 = 462.411 mole/hour
FB = 224.764 mole/hour
FOB = 58.5873 currency/hour
FR = 231.205 mole/hour
XBP = 0.48607

de variáveis é igual ao número de equações. Outra das informações que pode ser lida no quadro estatístico do *Solver* é o número de blocos da simulação.

A decomposição do sistema em blocos é efectuada pelo algoritmo de partição e ruptura do *solver* do ASCEND. O simulador percorre posteriormente os diversos blocos sequencialmente e, dependendo da natureza do bloco, utiliza as opções seguintes para resolver a(s) equação(ões) dentro de cada um deles:

1. Em primeiro lugar, uma tentativa para resolução simbólica de equações presentes nos blocos 1x1 da matriz, i.e., explicitação de variáveis em equações singulares;
2. Caso não seja apropriado ou possível efectuar decomposição explícita, é implementado um método numérico de procura de raiz para as equações singulares (Método de Brent (Press et al., 1992));
3. Para blocos com mais que uma equação, é utilizado um método de Newton modificado para a resolução simultânea das equações algébricas correspondentes. Após convergência o simulador avança para o bloco seguinte.

Para o problema P.MK, uma vez que a Figura 3.8 apresenta um número de sete blocos, equivalente ao número de equações, isso significa que o problema tem resolução sequencial.

Chega-se assim a uma situação em que, após estabelecimento do modelo, especificação das variáveis e análise estrutural por parte do simulador, a simulação pode ser executada. A mesma corresponde a um problema determinado com sete equações e sete variáveis. Para uma simulação com $X_{AP} = 0.5$, os resultados são os especificados no Quadro 3.2.

3.4.3 Inclusão de métodos

O código do modelo apresentado no Quadro 3.1 aparece como uma estrutura extremamente simples e compacta. No entanto, poderá também considerar-se incompleta à luz das capacidades do ASCEND. Para além das questões associadas às boas práticas de programação, nomeadamente pela inclusão de comentários nos procedimentos que

se escrevem, outras medidas podem ser tomadas para conseguir uma escrita de modelos mais robusta.

Constata-se que resolução do problema utilizando a interface gráfica do sistema terá uma simplicidade condicionada pela própria dimensão do modelo em questão. Tarefas simples como a definição dos valores dos dados do problema, ou a escolha das variáveis a fixar, podem tornar-se extremamente tediosas e contraproducentes nos casos em que o problema apresente maior dimensão.

A atenção centra-se assim em implementar procedimentos que permitam tornar a resolução dos modelos numa tarefa mais expedita. Assim, por exemplo, no que respeita à definição do conjunto de valores associados aos dados do problema, procura-se que a mesma seja efectuada directamente no modelo. Outras opções passam pela inclusão daquilo a que no âmbito do ASCEND se designam por métodos, elementos com um papel importante na capacidade de congregar a execução simultânea de funções em diferentes elementos.

Os métodos são procedimentos que podem ser requisitados facilmente através da interface do sistema (também pela linha de comando). Existem vários, com funções distintas, embora os mais usuais e importantes estejam ligados à definição das escolhas das variáveis de decisão, assim como à atribuição de estimativas iniciais para as variáveis de estado.

O Quadro 3.3 apresenta uma versão mais completa do modelo `make_up`. Neste caso os dados do problema já se encontram definidos, utilizam-se métodos e as diferentes secções do modelo são identificadas e organizadas com base na inclusão de comentários. Os métodos implementados neste caso são: `specify`, que permite estipular as variáveis como variáveis de decisão; `values`, destinado à atribuição de valores; e `default_self`, que permite definir estimativas iniciais para as variáveis do problema. Este último deverá ser sempre incluído, mesmo que vazio, já que, atendendo à importância que as estimativas iniciais têm na resolução de sistemas de equações, o simulador o invoca por omissão.

A definição de métodos permite que o utilizador execute as instruções neles contidas com uma simples chamada. Através da invocação do método respectivo por intermédio do **BROWSER**, onde se pode aceder ao menu apresentado na Figura 3.9, blocos de tarefas podem ser executados de uma só vez. Apesar disso, a codificação de procedimentos com vista à dispensa de utilização da interface gráfica traria ainda mais vantagens à reusabilidade dos modelos, tarefa a que se dá preponderância na secção seguinte.

Quadro 3.3 Estrutura do modelo *make_up* após a inclusão de métodos.

```

1: (*=====*)
2:                               M O D E L . M A K E _ U P
3: *=====*)
4:
5:   REQUIRE "atoms.a41";
6:   MODEL make_up;
7:
8: (*===== Data =====*)
9:   T, A1, A2, ER1, ER2, MM_A, MM_B, ro_A  IS_A real_constant;
10:  V, Cst_RPA, Val_B, Cst_R, Cst_A        IS_A real_constant;
11:   T      := 300 {K};
12:   A1     := 4.38e11 {1/hr};
13:   A2     := 6.09e18 {1/hr};
14:   ER1    := 9000 {K};
15:   ER2    := 15000 {K};
16:   MM_A   := 78 {g/mole};
17:   MM_B   := 112.5 {g/mole};
18:   ro_A   := 880 {kg/m^3};
19:   V      := 1 {m^3};
20:   Cst_RPA := 50 {dollar/m^3/hr};
21:   Val_B   := 10 {dollar/kg};
22:   Cst_R   := 3 {dollar/kg};
23:   Cst_A   := 5 {dollar/kg};
24:
25: (*===== Variables =====*)
26:   FO, F1, FR, FB                IS_A molar_rate;
27:   XAp, XBp                      IS_A mole_fraction;
28:   Vmol                          IS_A mole;
29:   FOB          "Objective Function" IS_A cost_per_time;
30:
31: (*===== Equations =====*)
32:   eq_01:  F1 = FO + FR;
33:   eq_02:  F1 = F1*XAp + Vmol*A1*XAp*exp(-ER1/T);
34:   eq_03:  Vmol = V*ro_A/MM_A;
35:   eq_04:  Vmol*A1*XAp*exp(-ER1/T) = F1*XBp + Vmol*A2*XBp*exp(-ER2/T);
36:   eq_05:  FR = F1*XAp;
37:   eq_06:  FB = F1*XBp;
38:   eq_FOB: FOB = FB*MM_B*Val_B - (Cst_R*FR + Cst_A*FO)*MM_A - Cst_RPA*V;
39:
40: (*===== Methods =====*)
41:   METHODS
42:     METHOD specify;
43:       XAp.fixed := TRUE;
44:     END specify;
45:
46:     METHOD values;
47:       XAp := 0.5 {};
48:     END values;
49:
50:     METHOD default_self;
51:     END default_self;
52:   END METHODS;
53: END make_up;

```

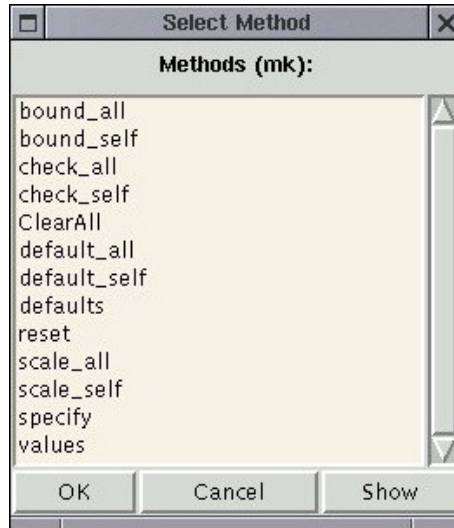


Figura 3.9 Ambiente ASCEND. Menu SELECT METHOD.

3.4.4 A sequenciação automática da resolução de problemas

Muitas vezes, um problema é resolvido recursivamente. Pode-se querer testar novos parâmetros, experimentar novas correlações, avaliar um modelo particular dentro de uma gama pré-estabelecida ou mesmo efectuar modificações pontuais no âmbito de um teste. Portanto, na eventualidade de existir um problema que o utilizador tenha de resolver inúmeras vezes, afigura-se como oportuno e necessário utilizar ferramentas, que viabilizem a automatização das tarefas comuns a tais recorrências. Tal acção permitirá libertar o utilizador da repetição contínua de acções básicas.

O sistema ASCEND apresenta essa capacidade. Antes de iniciar a resolução do problema, o utilizador pode monitorizar e gravar a sua sessão de trabalho. Seleccionando a opção **RECORD ACTIONS** disponível em **SCRIPT**, a sequência das tarefas que o utilizador executa por intermédio da interface gráfica é automaticamente convertida para um *script*. O resultado desta operação é ilustrado pela Figura 3.10, onde se revisitam algumas das tarefas já aqui enunciadas: a partir do modelo `make_up` existente no ficheiro `make_up.a4c` foi criada a simulação `mk`. Esta foi exportada para o **Browser** onde foi executado o método `reset` (que chama sequencialmente o método `specify` e `values`), sendo colocada posteriormente no **SOLVER**, onde foi resolvida.

Uma vez que este conjunto de instruções pode ser guardado num ficheiro `*.a4s`, o utilizador só precisará de carregar o *script* criado sempre que precisar de resolver no-

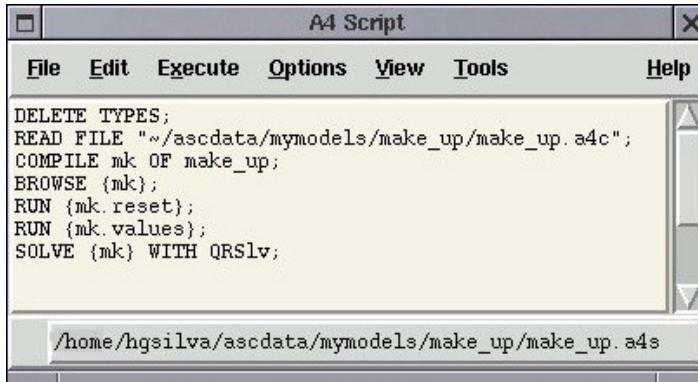


Figura 3.10 Janela `SCRIPT`. Automatização de tarefas para o modelo `make_up`.

vamente este problema. Uma vez que a janela `SCRIPT` tem capacidades de edição e depuração, o utilizador poderá testar modificações que julgue serem necessárias, avaliar apenas um conjunto restrito de passos ou compilar simultaneamente diferentes simulações partindo do mesmo modelo.

A codificação das sequências de resolução dos problemas torna-se deste modo numa ferramenta extremamente importante na utilização do ASCEND, traduzindo-se num ganho real de tempo de trabalho para o utilizador. Além disso, a codificação automática de *scripts* é uma ferramenta de aprendizagem importante, pois permite que o utilizador se familiarize com as instruções da linha de comando do ASCEND, recurso que oferece potencialidades mais alargadas do que as acessíveis através da interface gráfica. A utilização de *scripts* é também importante na implementação de procedimentos que permitam recolher conjuntos de resultados mais extensos, dando também a possibilidade de resolver sequencialmente várias simulações.

3.4.5 Abordagem preliminar ao tratamento de resultados em ASCEND

Quando uma simulação é concluída, o utilizador dispõe de funcionalidades que permitem a exportação dos resultados para ficheiros de texto. Mas essa não é, obviamente, uma das principais capacidades do ambiente de simulação.

Imagine-se a situação em que o utilizador quer averiguar como se comportam determinadas variáveis, por exemplo F_0 , F_R , F_B , ao mesmo tempo que regista a variação FOB à medida que X_{AP} , a variável de decisão, muda de especificação. Com o *script* apresentado no Quadro 3.4 tal situação é possível. O utilizador define um caso de estudo baseado na simulação criada e escolhe as variáveis que pretende monitorizar.

Posteriormente estabelece os conjuntos decisão que pretende avaliar e resolve o problema recursivamente, gravando os resultados num ficheiro externo, `outfile.dat`, cujo conteúdo se apresenta no Quadro 3.5

Quadro 3.4 Problema P.MK. *Script* para estudo de várias simulações variando X_{AP} .

```

1: DELETE TYPES;
2:
3: === Leitura da simulação ==
4: READ FILE "mymodels/make_up/make_up.a4c";
5: COMPILE mk OF make_up;
6: BROWSE {mk};
7: RUN {mk.reset};
8: RUN {mk.values};
9: ASSIGN {mk.FOB.lower_bound} -10000 {dollar/hour};
10:
11: === Preparação do case-study ==
12: STUDY {FOB} {FO} {FR} {FB}\
13: IN {mk}\
14: VARYING {{XAP} {0} {0.05} {0.1} {0.15} {0.2} {0.25} {0.3} {0.35}\
15: {0.4} {0.45} {0.5} {0.55} {0.6} {0.65} {0.7} {0.75}}\
16: USING {QRS1v}\
17: OUTFILE {/home/hgsilva/ascddata/mymodels/make_up/outfile.dat}\
18: ERROR STOP;

```

Quadro 3.5 Código do ficheiro de resultados `outfile.dat`.

```

1: Observations: (user index) (name) (units)
2: {indvar} {mk.XAP} {}
3: {1} {mk.FOB} {currency/hour}
4: {2} {mk.FO} {mole/hour}
5: {3} {mk.FR} {mole/hour}
6: {4} {mk.FB} {mole/hour}
7: indvar 1 2 3 4
8: --- --- --- --- ---
9: 0 -50 0 0 0
10: 0.05 -42.4611 23.1205 1.21687 14.9695
11: 0.1 -27.8816 46.2411 5.13789 36.7597
12: 0.15 -12.7853 69.3616 12.2403 59.6711
13: 0.2 1.86396 92.4821 23.1205 82.9708
14: 0.25 15.6548 115.603 38.5342 106.45
15: 0.3 28.2677 138.723 59.4528 130.028
16: 0.35 39.362 161.844 87.1466 153.665
17: 0.4 48.5179 184.964 123.309 177.341
18: 0.45 55.1813 208.085 170.251 201.043
19: 0.5 58.5873 231.205 231.205 224.764
20: 0.55 57.6372 254.326 310.843 248.499
21: 0.6 50.6878 277.446 416.169 272.245
22: 0.65 35.1602 300.567 558.196 295.999
23: 0.7 6.75863 323.687 755.271 319.76
24: 0.75 -42.2472 346.808 1040.42 343.526

```

Quadro 3.6 Problema P.MK. Criação de um ficheiro para folha de cálculo.

```
1: asc_merge_data_files excel\  
2: {/home/hgsilva/ascddata/mymodels/make_up/outfile_excel.txt}\  
3: {/home/hgsilva/ascddata/mymodels/make_up/outfile.dat};
```

Mas além de trabalhar os dados desta forma, o sistema ASCEND permite ainda que estes dados sejam guardados num formato passível de ser lido por outras aplicações. O Quadro 3.6 corresponde ao exemplo de instruções a implementar para se obterem dados passíveis de serem usados em folha de cálculo.

3.5 Conclusões

No presente capítulo foi dada atenção ao ambiente de simulação ASCEND, sistema constituído por uma linguagem de construção de modelos e um conjunto de ferramentas destinado à sua análise. Esses procedimentos encontram-se formalmente separados dos modelos, instâncias estruturadas segundo uma filosofia orientada por objectos, que proporcionam um nível de abstracção elevado.

No âmbito das suas funcionalidades podem ainda considerar-se a capacidade de gestão da informação de uma forma modular, assim como a sua aplicabilidade a problemas de natureza estacionária ou transiente.

Alguns factores adicionais poderão ser considerados na implementação desta aplicação como recurso aplicável a cursos de introdução à Engenharia de Sistemas de Processos ou mesmo Estratégia de Processo Químico:

- Aplicação de código fonte aberto instalável em múltiplas plataformas.
- A adopção de um estilo de escrita de modelos que potencia e encoraja a atribuição de significado físico aos diferentes elementos utilizados. Situação que cria condições favoráveis ao desenvolvimento do sentido crítico de análise importante na área de Engenharia.
- O relançamento do projecto ASCEND e a integração de novas valências, nomeadamente a criação de uma biblioteca de modelos partilhados entre os utilizadores.
- O alargado número de procedimentos que permite efectuar análises das simulações criadas, nomeadamente pelo constituição de casos de estudo que as abordem de um modo recursivo.

Nomenclatura

Caracteres romanos

a_i	Factor pré-exponencial para o cálculo de k_i , h^{-1}
A	Componente A
B	Componente B
C	Componente C
C_A	Concentração do componente A, mol/L/h
C_B	Concentração do componente B, mol/L/h
Cst_A	Custo do reagente A, \$/kg
Cst_R	Custo da operação de reciclagem, \$/kg
Cst_{RPA}	Custo de aluguer do reactor, \$/m ³ /kg
E_i	Energia de Activação da reacção r_i , J/mol
F_0	Caudal da corrente F_0 , mol/s
F_1	Caudal da corrente F_1 , mol/s
F_B	Caudal da corrente F_B , mol/s
F_C	Caudal da corrente F_C , mol/s
F_{obj}	Função Objectivo, \$/h
F_P	Caudal da corrente F_P , mol/s
F_R	Caudal da corrente F_R , mol/s
k_i	Constante cinética da reacção i , h^{-1}
n	Número de equações
m_1	Número de restrições de igualdade
MM_A	Massa molar de A, kg/mol
MM_B	Massa molar de B, kg/mol
r_i	Reacção química i
R	Constante dos Gases Perfeitos, J/mol/K
T	Temperatura, K
V	Voluma do reactor, m ³
V_{mol}	Volume Molar, mol
Val_B	Valor comercial do componente B, \$/kg
X_{AP}	Fracção molar do componente A na corrente F_P
X_{BP}	Fracção molar do componente B na corrente F_P

Caracteres gregos

ρ_A	Massa específica de A, kg/m ³
----------	--

Acrónimos e siglas

ASCEND	Advanced System for Computations in ENgineering Design
ASCII	American Standard Code for Information Interchange

DAE	Differential Algebraic Equation
EUA	Estados Unidos da América
GL	Graus de Liberdade
IPOPT	Interior Point OPTimizer
LA	Linear Algebraic
NLA	Nonlinear Algebraic
NLP	Nonlinear Programming
ODE	Ordinary Differential Equation
P.MK	Problema Make-Up
P.MK	Problema de teste make_up
RPA	Reactor Perfeitamente Agitado
SENL	Sistemas de Equações Não Lineares

Capítulo 4

(MSGA.ou.MSIMPSA).e.ASCEND

Sumário do Capítulo

O presente capítulo é atinente à descrição detalhada da estratégia integrada de simulação e optimização proposta na presente tese. A metodologia baseia-se na utilização conjugada do ASCEND com os optimizadores MSGA e MSIMPSA. Partindo da contextualização das ferramentas envolvidas, promove-se uma análise das possibilidades de conjugação das mesmas com vista à resolução de problemas de optimização. São ainda apresentados os resultados da aplicação desta metodologia a casos de estudo quer em estado estacionário, quer em estado transiente.

Considerando uma eventual classificação do software disponível, o ASCEND poderá ser visto como uma aplicação de natureza híbrida. As suas rotinas numéricas estão construídas em linguagem C enquanto a sua interface, e os procedimentos utilizados para a compilação e solução dos modelos, estão implementados sobre ferramentas baseadas em *scripting*.

Examinando os optimizadores MSGA (Salcedo, 1992b) e MSIMPSA (Cardoso et al., 1997), baseados respectivamente em procura aleatória adaptativa e recozimento simulado, constata-se que estes têm sido usados extensivamente na solução de problemas NLP e MINLP (Cardoso et al., 1997, 2000; Lima et al., 2004, 2006; Salcedo, 1992b; Salcedo et al., 2003; Silva e Salcedo, 2009). Baseados em procedimentos de procura condicionados por heurísticas, estes algoritmos permitem implementar esquemas de optimização não plausíveis com base na medida da violação verificada ao nível das restrições de desigualdade, facilitando a determinação de pontos plausíveis no domínio de procura. A sua aplicação tem demonstrado serem robustos e menos sensíveis a mau condicionamento e a não convexidades, não só na função objectivo mas também nas restrições,

apresentando-se como alternativas válidas à aplicação de metodologias baseadas na avaliação de gradientes. Ambos estão escritos em FORTRAN 77, tendo sido anteriormente implementados exclusivamente com simulações codificadas naquela linguagem de programação.

Um dos objectivos deste trabalho passa pelo projecto de uma estratégia que permita utilizar de forma conjugada simulador e ambos os optimizadores. Pretende-se uma ferramenta passível de ser aplicada na resolução de problemas de optimização, dotada de uma interface transparente ao utilizador, e sinérgica quanto ao aproveitamento das mais valias reveladas pelos algoritmos de procura do MSGA e MSIMPSA na exploração do domínio das variáveis de decisão, e a robustez e flexibilidade do ASCEND para lidar com a dificuldade inerente à resolução de sistemas de equações não lineares.

Neste capítulo abordam-se os aspectos relacionados com a concepção desta estratégia. Descrevem-se para isso as principais acções desencadeadas com o intuito de estabelecer uma base de trabalho que, além de aplicabilidade nos casos de teste, possa considerar-se suficientemente genérica e flexível para se debruçar sobre diferentes tipos de problemas. Dada a natureza mista das linguagens de programação presentes, o primeiro ponto passa pela contextualização da natureza das ferramentas envolvidas, nomeadamente o caso particular do *scripting*.

4.1 As linguagens de scripting

Nos dias de hoje ainda se assiste a um processo que, tendo sido iniciado há sensivelmente duas décadas, se traduz em mudanças na forma fundamental de como se escrevem programas para computador. Este período corresponde a uma transição de filosofia, onde a adopção das chamadas linguagens de programação convencionais, como o Fortran, Basic, C ou C++, divide atenções com as linguagens de *scripting*, casos do Perl, Python ou Tcl (Ousterhout, 1998), no pressuposto de que o todo será mais do que a soma das partes.

As linguagens de *scripting* foram concebidas para cumprir objectivos diferentes daqueles normalmente associados à programação convencional, normalmente destinadas à construção de algoritmos e estruturas de dados. Em contraposição, e apesar de ser utilizado como ferramenta para desenvolvimento de raiz, o *scripting* foi pensado para dar prioridade à celeridade de desenvolvimento e à promoção de sinergias, ou seja, partindo do princípio que um conjunto de componentes autónomos de software se encontra disponível, o propósito fundamental do *scripting* passa pelo estabelecimento de uma rede de pontes entre esses componentes. O objectivo dessa tarefa centra-se assim na implemen-

tação simultânea das diferentes valências de cada uma das aplicações, procurando obter de cada uma delas o contributo adequado para a nova funcionalidade que se pretende conseguir (Lúcio e Amaral, 2001; Ousterhout, 1998).

Na linha evolutiva do código para computador, as linguagens de programação convencionais foram introduzidas como alternativa à linguagem *assembler*. Tal mudança permitiu aos programadores um diferente grau de abstracção relativamente aos detalhes físicos de cada máquina, dos quais passaram a depender menos (Lúcio e Amaral, 2001). Apareceram assim as linguagens de alto nível como o LISP, o FORTRAN¹ e o Algol que, substituindo rapidamente a sua antecessora, se tornaram extremamente populares no desenvolvimento de aplicações de dimensão considerável (Ousterhout, 1998). A evolução deste tipo de linguagens prosseguiu, aparecendo posteriormente o Pascal, o C, o C++ e ainda mais recentemente o Java (Lúcio e Amaral, 2001).

Adoptando a designação proposta por Ousterhout (1998), as linguagens de programação convencionais podem considerar-se recursos dotados de uma escrita condensada. Isso significa que o programador deve declarar todos os fragmentos de informação que utiliza, sendo muitas vezes difícil utilizá-los sob uma forma diferente da que inicialmente lhes foi destinada. Pelo contrário, as linguagens de *scripting* tendem para um estilo de escrita mais ligeiro, isto é, todos os elementos são passíveis de apresentar o mesmo comportamento, podendo a qualquer altura permutar entre os diferentes tipos de variável disponíveis.

Tecnicamente, poder-se-á dizer que a principal diferença reside na forma como o código é executado. As linguagens convencionais são compiladas (traduzidas inicialmente para código *assembler* e subsequentemente para binário), enquanto que o *scripting* tem por base a interpretação. A utilização de *scripts* adopta uma execução sequencial, linha a linha, abrindo novas possibilidades de trabalho: o código pode ser encarado como uma entidade activa que pode gerar-se a si mesma, funcionando de uma forma dinâmica e adaptável.

Esta filosofia de trabalho permite que linguagens como o Perl, Python, Rexx, Tcl/Tk e outras representam um estilo de programação substancialmente diferente, garantindo um nível de abstracção que provém do seu desprovemento de escrita. Lúcio e Amaral (2001) afirmam que estas linguagens garantem resposta a um dos principais desafios da Engenharia de Software: a interligação de diferentes componentes já existentes, promovendo a reutilização de recursos, com o intuito de minimizar o esforço necessário à resolução de problemas.

¹Uma história detalhada sobre a linguagem de programação FORTRAN pode ser consultada no endereço <http://www.softwarepreservation.org/projects/FORTRAN/>

As situações típicas onde este tipo de linguagens podem ter interesse de aplicação são as seguintes (Lúcio e Amaral, 2001):

1. Desenvolvimentos rápidos de aplicações de tamanho pequeno/médio;
2. Aplicações que manipulem uma grande diversidade de subestruturas;
3. Aplicações que evoluam rapidamente no tempo;
4. Aplicações que necessitem de ser extensíveis.

Todavia, e apesar de existirem adeptos de ambas as filosofias que preconizam sem hesitação a adopção de uma em detrimento da outra, os dois tipos de paradigma coexistem em função da adequabilidade necessária à aplicação de cada um deles. Estudos como o de Prechelt e Informatik (2002) mostram que o nível de desempenho de cada linguagem é claramente dependente dos níveis de experiência e conhecimento de cada programador. Resultados como esse reforçam a ideia empírica de que problemas diferentes podem também exigir ferramentas diferentes, justificando por isso que a escolha da linguagem de programação não seja independente da sua tarefa alvo.

4.1.1 O caso particular da linguagem Tcl/Tk

O acrónimo Tcl (leia-se *tickle* de *Tool Command Language*) designa uma linguagem que é utilizada no desenvolvimento de ferramentas e aplicações para o ambiente X e na extensão de outras aplicações. O Tk (leia-se *tee-kay*) é a sua extensão mais útil, provavelmente a verdadeira razão que promoveu a popularidade do Tcl. O Tk acrescenta comandos que permitem criar interfaces gráficas para as aplicações desenvolvidas, assim como estabelece procedimentos de comunicação entre tais aplicações (Ousterhout, 1993, 2008; Young, 1997).

A combinação das duas, geralmente designada por Tcl/Tk, resulta numa ferramenta capaz de desenvolver aplicações dotadas de interface gráfica para o utilizador (GUI— *Graphical User Interface*). Tal como já foi referido, e uma vez que pertence à família do *scripting*, o Tcl/Tk não é uma linguagem compilada mas sim interpretada, o que se revela normalmente vantajoso em termos da despesa em tempo para depuração. Não obstante, é do conhecimento comum que a interpretação, pela leitura linha a linha, apresenta tempos de execução mais elevados que os habitualmente registados com a execução de código binário (Ousterhout, 1993). O impacto relativo da menor celeridade será mais ou menos relevante em função das características do problema.

A versatilidade do Tcl/Tk está igualmente relacionada com a possibilidade do seu interpretador poder ser embutido noutros programas. Ou seja, o interpretador da lin-

guagem é uma biblioteca em linguagem C que pode ser ligada a outras aplicações. O objectivo passa por dotar as aplicações com a capacidade de “entender” todos os comandos e instruções associados ao Tcl/Tk (Ousterhout, 1993). O ambiente de simulação ASCEND é um exemplo das aplicações que apresentam tal funcionalidade, sendo esse um aspecto explorado no seu acoplamento com os optimizadores estocásticos.

4.2 A conjugação Simulação/Otimização

O desenvolvimento de uma interface de comunicação entre os optimizadores estocásticos MSGA e MSIMPISA com o ASCEND percorreu um trajecto faseado. As primeiras abordagens incluíram testes com sistema operativo não baseado em Unix e tentativas de codificação das ferramentas de ligação necessárias usando diversas linguagens. A primeira opção adoptou a comunicação usando Perl, seguindo-se a linguagem C e, só com alguma experiência adicional, se optou por desenvolver a ligação usando Tcl.

O método de escolha da melhor abordagem mostrou-se dependente do grau de conhecimento dos aspectos relacionados com a utilização do modo avançado do ASCEND². Neste trabalho, a melhor compreensão de tais funcionalidades beneficiou objectivamente contributo dado pelo Doutor Ben Allan e o Professor Arthur Westerberg. Regista-se por isso uma nota de agradecimento a ambos por terem respondido sempre que possível às solicitações que lhes foram endereçadas.

O caminho seguido para a implementação da estratégia de simulação e optimização pode dividir-se em duas etapas distintas, aqui designadas por I e II. Na evolução da primeira para a segunda procurou-se ganhar em funcionalidade, robustez e usabilidade. A metodologia foi aplicada à análise de problemas NLP simples, o que permitiu validá-la e promover a sua extensão. A Secção 4.2.3 e o Capítulo 5 apresentam, respectivamente, a aplicação desta estratégia a problemas dinâmicos e a problemas MINLP.

4.2.1 A estratégia de comunicação

A aplicação dos optimizadores estocásticos, codificados como algoritmos independentes, depende de um procedimento externo que se encarregue do cálculo da função objectivo (Figura 4.1). Nos casos do MSGA e MSIMPISA, rotinas escritas em FORTRAN são normalmente disponibilizadas para esse efeito. Por conseguinte, a concepção de uma estratégia integrada implica uma ideia base: estabelecer uma forma onde o ASCEND,

²O ASCEND pode ser executado no chamado *developer mode*, opção que dá acesso a um menu de navegação suplementar onde é possível explorar as variáveis internas do simulador. O seu acesso depende da configuração estabelecida pelo utilizador (ver Apêndice B).

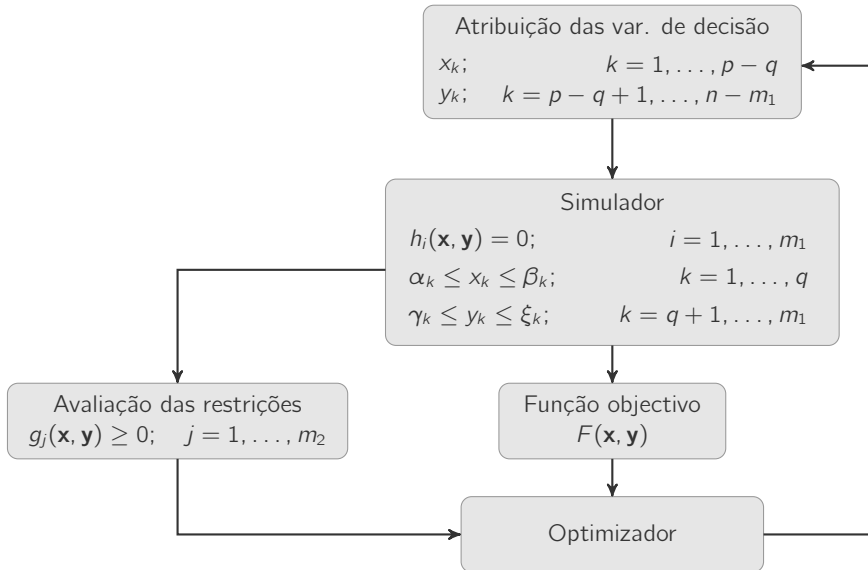


Figura 4.1 Diagrama generalizado da aplicação conjunto de um simulador e de um otimizador estocástico (adaptado de Lima (2005)).

como procedimento externo aos optimizadores, seja responsável pela medição do desempenho do sistema.

A abordagem realizada nesse sentido parte do princípio que é necessário decompor a estrutura em que os optimizadores normalmente são utilizados, isolando os seus blocos principais. Com esta análise, pretende-se definir quais os pontos em que se deve proceder a um curto-circuito no fluxo de informação habitual nos algoritmos estocásticos, viabilizando uma avaliação externa da função objectivo. Para isso, optou-se por tirar partido das ligações existentes entre os procedimentos heurísticos de procura e a avaliação de desempenho. Procurou-se com isso garantir a integridade do código fonte dos optimizadores, procedendo-se, para o cálculo da função objectivo, à substituição da simulação em FORTRAN pelas instruções de comunicação com o ASCEND.

Etapa I A primeira configuração estabelecida para o acoplamento do ASCEND aos optimizadores está apresentada na Figura 4.2. Nesse diagrama constata-se que operam três blocos de software: o optimizador, o simulador e o *script* de ligação designado por `main.opt`.

Uma vez que o optimizador e o simulador assentam em filosofias de programação distintas, as formas como o *script* `main.opt` comunica com ambas as aplicações também

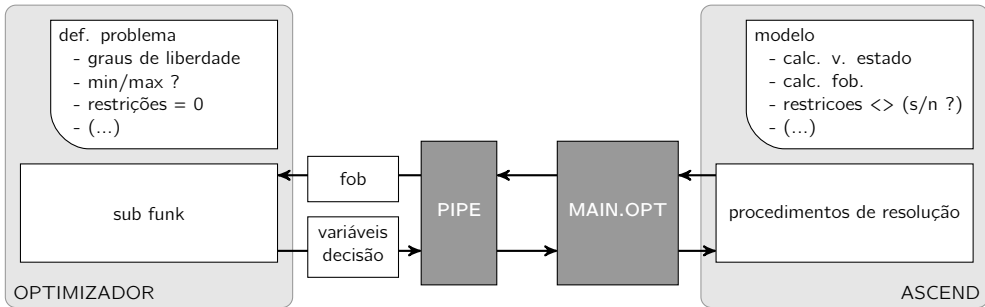


Figura 4.2 Esquema de comunicação. Etapa I.

difere. Devido à afinidade de linguagem base, a comunicação entre o `main.opt` e o ASCEND é efectuada através de um comando, o `send`, que permite executar operações remotamente entre aplicações escritas em Tcl/Tk. Por outras palavras, um determinado programa pode solicitar a execução de um procedimento que se encontra definido no contexto de uma outra aplicação.

A ligação entre o optimizador e o `main.opt` segue uma filosofia completamente diferente. O fluxo de informação deve ser dirigido para o exterior: a atribuição do conjunto de variáveis de decisão arbitradas pelo optimizador, é direccionado para o canal `stdout`, enquanto que o requerimento do valor da função objectivo ocorre a partir do `stdin`. Deste modo, no `main.opt` o optimizador recebe ordem de execução por abertura de um canal encapsulado com um *pipe*, tornando possível estabelecer correspondência de fluxo entre os dois sentidos do canal criado no `main.opt` e o I/O do optimizador. Estabelece-se assim uma sobreposição entre os descritores, o que torna possível a passagem de informação em ambos os sentidos, e que satisfaz os requisitos necessários à comunicação pretendida.

O esquema delineado delega diferentes funções aos diversos blocos. No optimizador são definidos os atributos do problema: o tipo (minimização ou maximização), a dimensão (número de graus de liberdade) e o número de restrições suplementares associadas. Através da rotina `func` assegura-se a inclusão das instruções que garantem, quer o envio das variáveis de decisão para o simulador, quer a leitura do valor da função objectivo correspondente.

O bloco do simulador pode ainda ser subdividido em duas partes funcionais: a primeira, o *script* de procedimentos, contém a sequência de comandos que permitem carregar, configurar e avaliar o modelo; a segunda corresponde ao modelo propriamente

dito, e incluí as estruturas encarregues de tarefas como:

- Calcular as variáveis de estado;
- Calcular o valor da função objectivo;
- Avaliar as restrições suplementares;
- Validar o resultado obtido na simulação (critério para penalização).

A progressão dos optimizadores estocásticos baseia-se na avaliação de sucessivas determinações do índice de desempenho do modelo. Esta situação permite que esses algoritmos sejam capazes de lidar com modelos do tipo caixa-negra, aspecto do qual se tira partido nesta configuração. No entanto, e uma vez que as restrições de desigualdade se encontram definidas no modelo ASCEND, a configuração do lado do optimizador considerará o problema não condicionado, estabelecendo as restrições de desigualdade como inexistentes e obrigando os optimizadores a progredir por intermédio de um trajecto de optimização plausível.

Os Quadros 4.1 a 4.4 apresentam o código fonte dos procedimentos envolvidos directamente na gestão da informação partilhada entre optimizador e simulador para o caso particular de um dos problemas testados, o P.RSIA01 (Salcedo, 2001). O mesmo apresenta apenas uma restrição de igualdade e um grau de liberdade (ver Apêndice A.1), simplicidade que garante uma dimensão de problema compatível com o destaque pretendido para as funcionalidades de cada um dos procedimentos.

O Quadro 4.1 apresenta o modelo ASCEND para o P.RSIA01. Nele são declaradas as variáveis, estabelecidas as equações e especificado o conjunto decisão. Esta estrutura substitui o código FORTRAN que integraria a função apresentada no Quadro 4.2, que por sua vez mostra as instruções que permitem o redireccionamento da informação para o simulador. A chamada à rotina `flush` (linha 13) garante o envio de toda a informação colocada no canal `stdout`.

A leitura e reencaminhamento dessa informação é coordenada pelo *script* `main.opt`, apresentado no Quadro 4.3. Nele são estabelecidas as configurações do canal de comunicação usando para isso as instruções disponibilizadas para o efeito pela linguagem Tcl/Tk. Por exemplo, a comando `send` escrito na linha 17 do Quadro 4.3 permite, no contexto da aplicação `ascend4`, carregar o ficheiro `opt.script.rs_ia01.a4s`. Tal acção define no simulador uma série de procedimentos, apresentados no Quadro 4.4, utilizados para definir a sequência de passos que permitem criar a simulação do problema modelizado (linhas 6–8), calcular a respectiva função objectivo (linhas 12–18) e transmitir o resultado ao optimizador (linhas 21–24).

Quadro 4.1 Problema P.RSIA01. Modelo para ASCEND.

```

1: (*=====*)
2:           M O D E L . R S _ I A 0 1
3: *=====*)
4:   REQUIRE "atoms.a41";
5:   MODEL Ia_1;
6:
7:   x1, x2, fob IS_A generic_real;
8:
9:   eq_01: -2*x1^4 - x2 + 2 = 0;
10:  eq_fob: fob = -12*x1 -7*x2 + x2^2;
11:
12:  METHODS
13:    METHOD specify;
14:      x1.fixed := TRUE;
15:    END specify;
16:
17:    METHOD bound_self;
18:      X2.upper_bound := 3;
19:      X2.lower_bound := 0;
20:    END bound_self;
21:
22:    METHOD default_self;
23:      END default_self;
24:    END METHODS;
25:  END Ia_1;

```

Quadro 4.2 Problema P.RSIA01. Subrotina funk.

```

1:   double precision function funk(v)
2:   implicit double precision (a-h,o-z)
3:   integer ncalc
4:   double precision v(100)
5:   common/no_calc/ncalc
6:   save
7: c -----
8: c Passagem das variáveis de decisão para o stdout.
9: c 'call flush(6)' garante o envio de toda a informação
10: c colocada no canal com a instrução write.
11: c -----
12:   write (*,*) v(1)
13:   call flush(6)
14:
15:   read (*,*) funk
16: c -----
17:   return
18:   end

```

Quadro 4.3 Problema P.RSIA01. Ficheiro main.opt.

```
1: #! /usr/bin/wish
2: # (*=====*)
3:                               M A I N . O P T - P . R S I A O 1
4:  *=====*)
5:
6: #   Variaveis globais
7:   global fob x2
8:   global msimpsa
9: #   Instrucao para manter o simulador pendente no canal, que
10: #   continuara atento enquanto existir actividade do optimizador
11:  global forever
12:  set record [open records.dat w+]
13: #   Abertura do canal para o optimizador
14:  set msimpsa [open "|main" "w+"]
15:  fconfigure $msimpsa -buffering none
16: #   Carreamento procedimentos no ASCEND
17:  send ascend4 [list source /home/hgsilva/ascddata/mymodels/
                                rsalcedo.Ia.1/opt.script.rs_ia01.a4s]
18:
19: #   Abertura do canal
20: fileevent $msimpsa readable {
21:   # Obter variaveis de decisao desde o MSIMPSPA
22:   # vlist - lista com todas as variaveis
23:   # v?? - designacoes individuais
24:   gets $::msimpsa v01
25:   set flag01 [llength $v01]
26:   if {$flag01 == 0} {
27:     exit
28:   }
29:   # Simulacao e leitura da funcao objectivo
30:   send ascend4 [list run_sim $v01]
31:
32:   proc fob_get {fob x2} {
33:     if { [eof $::msimpsa] } {
34:       close $::msimpsa
35:       set ::forever 1
36:     }
37:     puts $::msimpsa $fob
38:     puts $::msimpsa $x2
39:   }
40:   send ascend4 [list fob_send]
41: }
42:
43: vwait forever
```

Quadro 4.4 Problema P.RSIA01. Ficheiro opt.script.

```

1: #=====
2: #           O P T . S C R I P T . R S _ I A 0 1 . A 4 S
3: #=====
4:
5: # Construção da simulação
6: DELETE TYPES;
7: READ FILE "mymodels/rsalcedo.Ia.1/model.rs_ia.a4c";
8: COMPILE sim OF Ia_1;
9: # BROWSE {sim};
10:
11: # Procedimento RUN SIMULATION
12: proc run_sim {var01} {
13:   global fob_val
14:   RUN {sim.reset}
15:   ASSIGN {sim.x1} $var01 {}
16:   SOLVE {sim} WITH QRS1v
17:   set fob_val [u_getval sim.fob]
18: }
19:
20: # Procedimento FOB SEND
21: proc fob_send {} {
22:   global fob_val
23:   send main.opt [list fob_get $fob_val]
24: }

```

A conjugação dos procedimentos apresentados permitiu obter solução para alguns problemas de teste, nomeadamente o P.RSIA01, P.HRCV e P.ABSTRIP (ver Apêndice C). No entanto, a arquitectura de comunicação desenhada revelou alguma ineficiência. Além do baixo desempenho ao nível do tempo de computação, não foi possível levar a cabo uma gestão robusta do canal de comunicação. Tal situação condicionou a passagem de valores de variáveis de e para o optimizador, inviabilizando a opção de progressão em regiões não plausíveis, variante que requer a avaliação das restrições suplementares.

Etapa II Em função das fragilidades observadas, o desenvolvimento de uma estratégia de comunicação mais eficiente assentou essencialmente na procura de melhoria a três níveis:

1. Simplificação do esquema de comunicação;
2. Melhor desempenho quanto aos tempos de execução;
3. Disponibilidade de todas as funcionalidades do optimizador.

Quanto ao estabelecimento de um procedimento de comunicação menos complexo, pode desde já salientar-se que o recurso ao *script main.opt*, externo ao simulador, levanta questões sobre os possíveis benefícios da alternativa baseada numa ligação que

prescinda da utilização de um elemento de software intermediário. Com essa mudança seria possível modificar o esquema, passando de quatro para dois ramos de comunicação.

No diagrama da Figura 4.2, a utilização de um *script* externo justifica-se visto que a comunicação entre blocos requer instruções do Tk, as quais o ASCEND não consegue interpretar. O comando *send* é executado dentro de uma estrutura designada por *event*, cuja finalidade é estabelecer um canal de comunicação estável com outra aplicação.

A dispensa do *main.opt* pressupõe que algum procedimento do Tcl seja capaz de gerir a troca de informação entre blocos. Nesse sentido, realizaram-se testes de implementação utilizando combinações com estruturas de fluxo do Tcl.

Como consequência, verificou-se que a estrutura cíclica do tipo *while* cumpre satisfatoriamente os requisitos necessários. O seu uso sustenta o intercâmbio contínuo de informação, sem quebra do canal definido pelo *pipe*, mantendo a interacção entre os dois blocos activa enquanto o optimizador não concluir a exploração do domínio do conjunto de decisão.

A melhoria dos tempos de execução parte da análise da estrutura interna dos modelos ASCEND. Quando uma simulação é compilada em ASCEND, os objectos criados correspondem a entidades com diversos atributos. Como já referido na Secção 3.4.2, as variáveis têm associados limites, valores nominais e um atributo lógico que serve para indicar se fazem, ou não, parte integrante do conjunto de variáveis de decisão. Quanto às equações, as mesmas têm também um atributo lógico, mas que determina a sua inclusão na definição do problema.

Deste modo, a modificação de um objecto na simulação pode corresponder à actualização de mais do que um atributo, aumentando a quantidade de informação manipulada. Toda essa informação é também actualizada ao nível da interface gráfica, processo que se verifica periodicamente à medida que se sucedem as iterações do *solver*.

Uma das grandes vantagens do ASCEND é a permanente acessibilidade, em termos de informação sobre o problema, que a aplicação disponibiliza ao utilizador. Contudo, quando se está perante um cenário de procura estocástica, a manipulação computacional de toda esta informação, e correspondente actualização no ambiente gráfico, implica um consumo extra de recursos computacionais. Uma vez que o utilizador não necessita de ler as informações de que necessita através da GUI do simulador, a situação descrita revela-se contraproducente face ao cumprimento dos objectivos estabelecidos para o acoplamento entre o optimizador e o simulador.

Neste contexto o envolvimento do Dr. Ben Allan foi importante. Perante esta questão, e com base no conhecimento que possui sobre a estrutura programática do simulador, o Dr. Allan orientou o desenvolvimento e aplicação de um procedimento que

inibe as actualizações na interface gráfica do utilizador. Designada por `RESOLVE_NO_GUI` (ver Apêndice B.1), a nova instrução permite efectuar a simulação dos modelos sob o ponto de vista numérico, dispensando os procedimentos normalmente destinados à actualização no ambiente gráfico.

No que diz respeito à utilização plena dos optimizadores, a dispensa do *script* externo `main.opt` permite também ultrapassar essa desvantagem. Verificou-se que, desde que se salvguarde a inclusão da rotina `flush` do lado do optimizador, a ligação directa da via *pipe* garante condições estáveis para a partilha de um maior número de variáveis (informação) entre os dois blocos.

Dessa forma, a avaliação das restrições suplementares pode ser efectuada pelo optimizador sempre que o utilizador assim o deseje, ou necessite. Tal situação torna possível a implementação de esquemas de progressão não plausíveis baseados no relaxamento progressivo das restrições, com o `MSGGA`, ou por intermédio da aplicação de um esquema ponderado e dinâmico de penalização com o `MSIMPSSA`.

Tendo apresentado as decisões tomadas no sentido de modificar a estratégia desenvolvida para o acoplamento, o esquema patente na Figura 4.2 pode ser redesenhado, passando a apresentar a configuração evidenciada na Figura 4.3. Nesta nova versão evidencia-se que as variáveis de estado necessárias à avaliação das restrições suplementares também podem ser enviadas para os optimizadores. A aplicabilidade de ambas as configurações foi testada e medida, descrevendo-se na Secção 4.2.2 o caso particular para o problema P. ABSTRIP.

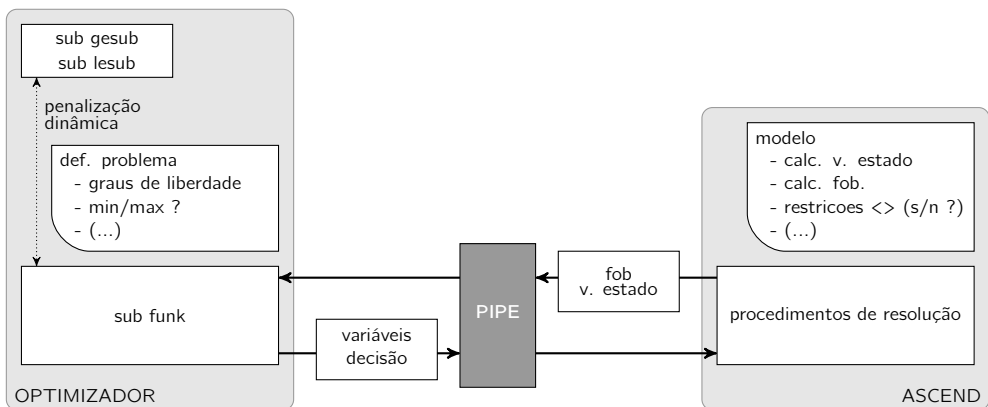


Figura 4.3 Esquema de comunicação II.

O exemplo apresenta cinco graus de liberdade, sendo definido por um sistema que contempla 45 variáveis sujeitas a 40 restrições de igualdade. A especificação como variáveis de projecto do conjunto $v_{P.ABSTRIP} = \{T_2, T_4, N, W_4, A_1\}$, conduz a um conjunto de saída sequencial após análise pelo algoritmo de partição e ruptura do *solver* do ASCEND. O problema adopta uma estrutura de resolução constituída por 42 blocos— as duas equações que definem as componentes da função objectivo também são contabilizadas. O resultado da decomposição é evidenciado pela matriz de incidências do modelo do problema, informação que está disponível através da janela **SOLVER** do ASCEND, apresentada na Figura 4.5.

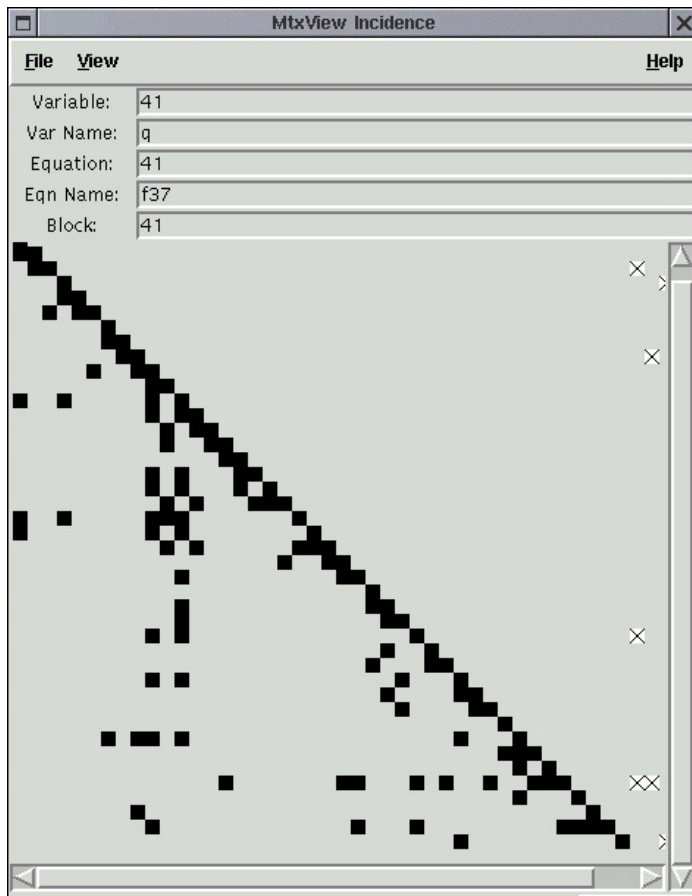


Figura 4.5 Problema P.ABSTRIP. Matriz de incidências do problemas após decomposição e partição operados pelo ASCEND. Neste caso, pode verificar-se que no bloco 41, a variável de saída q é calculada a partir da Equação $f37$.

Foram realizados ensaios de medição do tempo computacional, para comparar o desempenho das simulações codificadas em FORTRAN com o registo apresentado pela estratégia conjugada com base no ASCEND. O presente caso de estudo foi analisado segundo as duas configurações estabelecidas nas Figuras 4.2 e 4.3 para a estratégia integrada proposta. A comparação do desempenho de ambas debruça-se essencialmente sobre a medição da despesa em termos de tempo de computação, bem como, obviamente, na capacidade de obtenção de soluções plausíveis e satisfatórias do problema.

A Tabela 4.1 apresenta os resultados obtidos. Nela constam diferentes registos, como o número de corridas efectuadas pelo optimizador; o número total de avaliações da função objectivo realizado durante essas corridas e o melhor resultado obtido na sua avaliação. Relativamente aos tempos de computação, são apresentados o tempo médio despendido em cada simulação, assim como o tempo total aproximado para resolução do problema. Todos os resultados foram conseguidos com base na mesma configuração dos parâmetros do optimizador, tomando, respectivamente os seus valores típicos. Com base nos dados apresentados na Tabela 4.1, e considerando apenas a estratégia conjugada entre MSGA, MSIMPSA e ASCEND, pode dizer-se que:

1. Por comparação dos casos I e II, a desactivação da actualização dos atributos das variáveis na GUI traduz-se no ganho aproximado de uma ordem de grandeza no tempo de computação por simulação;
2. A mudança no desenho do esquema de comunicação entre os casos II e III não evidencia ter impacto no desempenho computacional por simulação, dando a indicação de que as instruções inerentes à ligação entre o optimizador e o simulador não serão as que consomem mais recursos;
3. A avaliação das restrições suplementares pelo optimizador (neste caso com o MSIMPSA), modificação operada entre os casos III e IV, potencia ganho no tempo médio por simulação, que passa para menos de metade.

A análise geral dos resultados evidenciados mostra que a melhor situação possível é o caso IV e que a utilização da conjugação entre MSGA, MSIMPSA e ASCEND beneficiará, em termos de desempenho computacional, de um ganho sensivelmente equivalente a duas ordens de grandeza com a mudança no esquema de comunicação. Todavia, a nova configuração encontra-se aquém do desempenho evidenciado pela utilização de simulações em FORTRAN, o que vai ao encontro do que se previa no início deste trabalho. Dada a natureza do simulador, que é em parte baseado em *scripting*, esperava-se uma diferença de desempenho na casa das duas ordens de grandeza, uma evidência

que se repete para outras aplicações que recorrem a linguagens interpretadas, como é o caso, por exemplo, do código desenvolvido para Matlab®.

Não sendo o ASCEND uma aplicação destinada em primeiro lugar à optimização, esta característica condicionará de certa forma a escolha desta estratégia conjugada, dado o carácter recursivo com que a função objectivo é avaliada numa abordagem estocástica. Ainda assim, e apesar de menos célere, é relevante referir que esta estratégia mostrou-se eficaz na resolução dos problemas propostos.

A análise dos dois casos do problema P.ABSTRIP conduziu aos resultados apresentados na Tabela 4.2. O conjunto $v_{P.ABSTRIP} = \{T_2, T_4, N, W_4, A_1\}$ foi estabelecido como conjunto decisão e os resultados obtidos correspondem à realização de dez corridas de optimização que partem de diferentes sementes primordiais para arbitrar os valores para as variáveis de decisão. No Caso II, onde se exige uma recuperação de composto orgânico de 99%, só foi possível encontrar soluções satisfatórias com recurso ao esquema de penalização dinâmica, apenas disponibilizado pelo MSIMPISA. A Figura 4.6 apresenta as distribuições de erro relativo no que respeita aos melhores valores de desempenho encontrados, os quais estão de acordo com os resultados apresentados na literatura (Ferreira e Salcedo, 2001a,b; Salcedo et al., 2003; Salcedo, 2001). Nos gráficos apresentados na Figura 4.6, ΔF_{obj} corresponde à diferença do valor encontrado no final de cada corrida para o melhor valor determinado em todas elas, e que cor-

Tabela 4.1 Problema P.ABSTRIP. Comparação, entre abordagens baseadas na codificação em FORTRAN e o acoplamento do MSGA e MSIMPISA com o ASCEND, dos desempenhos ao nível de tempo computacional e do melhor valor obtido para a função objectivo.

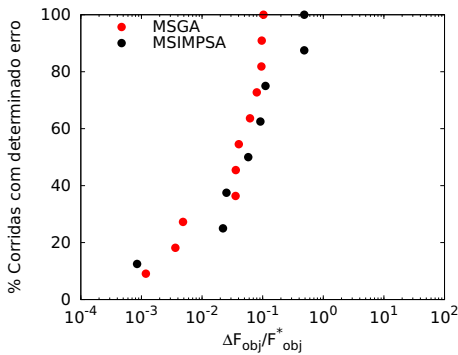
	Corridas	Avaliações	Tempo decorrido		F_{obj} (\$/ano)
			Simulação	Total	
Sim. em FORTRAN	5	44851	$2.7 \cdot 10^{-5}$ s	1.2 min	-173407

Optimizadores estocásticos & ASCEND

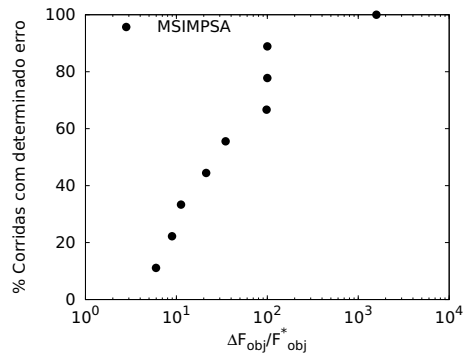
Caso	Ligação	GUI	Restrições	Corridas	Avaliações	Tempo decorrido		F_{obj} (\$/ano)
						Sim. (s)	Total (min)	
I	Externa	sim	ASCEND	1	36063	$1.1 \cdot 10^{-1}$	67	-173540
II	Externa	não	ASCEND	1	36063	$1.4 \cdot 10^{-2}$	9	-173540
III	Interna	não	ASCEND	5	115075	$1.3 \cdot 10^{-2}$	26	-173392
IV	Interna	não	MSIMPISA	5	120547	$5.2 \cdot 10^{-3}$	10	-173392

Tabela 4.2 Problema P. ABSTRIP. Melhores desempenhos encontrados.

Caso	Optimizador	V_p . ABSTRIP	F_{obj} (\$/ano)	Nº Avaliações
I	MSIMPSA	$T_2 = 326.8$ K	$-1.734 \cdot 10^6$	31915
		$T_4 = 342.7$ K		
		$W_4 = 189.1$ kg/h		
		$N = 25.9$ andares		
		$A_1 = 90.6$ m ²		
I	MSGA	$T_2 = 326.7$ K	$-1.734 \cdot 10^6$	38525
		$T_4 = 342.4$ K		
		$W_4 = 211.8$ kg/h		
		$N = 25.7$ andares		
		$A_1 = 90.1$ m ²		
II	MSIMPSA	$T_2 = 353.1$ K	$4.214 \cdot 10^3$	240063
		$T_4 = 370.4$ K		
		$W_4 = 818.2$ kg/h		
		$N = 29.4$ andares		
		$A_1 = 61.2$ m ²		



(a) Caso I.



(b) Caso II.

Figura 4.6 Problema P. ABSTRIP. Distribuições de erro relativo para as corridas efectuadas.

responde à solução do problema, F_{obj}^* . Assim, Pela sua interpretação, verifica-se que os resultados poderão ser considerados mais reprodutíveis quanto mais acentuada for a verticalidade de um determinado conjunto de pontos, os quais estarão mais próximos do melhor resultado conhecido quanto mais à esquerda estiverem representados.

A abordagem a exemplos NLP de maior dimensão é delegada para o Capítulo 5, onde é apresentado um caso de estudo para análise de um sistema de destilação reactiva, conjuntamente com a abordagem à sua formulação MINLP. Neste capítulo dá-se continuidade ao desenvolvimento e aplicação da estratégia de comunicação estabelecida, apresentando na Secção 4.2.3 a sua aplicação a casos em estado não estacionário.

4.2.3 Aplicações em estado transiente

Para além de simulações em estado estacionário, o ASCEND também tem a capacidade de lidar com problemas em estado transiente (Perry e Allan, 1996). A resolução dos sistemas de equações algébrico-diferenciais patentes nesses casos é conseguida com o integrador LSODE (Radhakrishnan e Hindmarsh, 1993). Atendendo à disponibilidade desta funcionalidade, decidiu-se alargar o âmbito de aplicação da estratégia conjugada a este tipo de problemas, permitindo usar as diferentes valências do simulador.

Em ASCEND, a representação de um modelo exige a inclusão de três métodos acessíveis através da biblioteca `ivpsystem.a41` (ASCEND Development Team, 1998). Os métodos em causa são o `set_ode`, `set_obs` e `seqmode`, que estruturam e associam a informação presente nas equações por forma a que o modelo possa ser especificado. Para ilustrar devidamente este conceito, apresenta-se o problema P.BATCHT, proposto por Ray (1981).

O exemplo descreve um processo dinâmico num reactor batch onde duas reacções em sequência acontecem envolvendo três espécies químicas diferentes. O objectivo é encontrar o perfil de temperatura óptima que conduz à produção máxima do intermediário durante um tempo de partida estipulado. As leis cinética e de dependência da constante cinética face à temperatura são conhecidas. Assume-se que o controlo de temperatura é exacto na gama 298 a 398 K. O problema foi resolvido utilizando uma aproximação por partes ao longo do tempo da partida, subdividido em 50 segmentos.

A descrição do problema e respectivo código fonte do modelo ASCEND encontram-se no Apêndice A.4. No Quadro 4.5 apresenta-se o excerto relativo à inclusão dos métodos no âmbito da modelização dinâmica. A sua análise permite verificar que a interpretação das equações diferenciais é feita com base na declaração de variáveis suplementares. Considerando dependência em ordem ao tempo, a declaração de uma variável genérica

Quadro 4.5 Problema P.BATCHT. Excerto do modelo ASCEND.

```
25: (*===== V A R I A B L E S =====*)
26:  dca_dt, dcb_dt  IS_A conc_rate;
27:  ca, cb          IS_A molar_density;

(...)

32: (*===== E Q U A T I O N S =====*)
33:  eq01: dca_dt + k1*exp(-E1/(R*T))*ca^2 = 0;
34:  eq02: dcb_dt - k1*exp(-E1/(R*T))*ca^2 + k2*exp(-E2/(R*T))*cb = 0;

(...)

36: (*===== M E T H O D S =====*)

56:  (*== SPECIFY ==*)
57:  METHOD specify;
58:    T.fixed := TRUE;
59:    RUN seqmode;
60:  END specify;
61:
62:  (*== SEQMODE ==*)
63:  METHOD seqmode;
64:    ca.fixed := TRUE;
65:    cb.fixed := TRUE;
66:  END seqmode;
67:
68:  (*== SETODE ==*)
69:  METHOD set_ode;
70:    t.ode_type := -1;
71:    dca_dt.ode_type := 2;
72:    ca.ode_type := 1;
73:    dcb_dt.ode_type := 2;
74:    cb.ode_type := 1;
75:    dca_dt.ode_id := 1;
76:    ca.ode_id := 1;
77:    dcb_dt.ode_id := 2;
78:    cb.ode_id := 2;
79:  (*== TOLERANCES ==*)
80:    ca.ode_atol := 1e-6 {};
81:    cb.ode_atol := 1e-6 {};
82:    ca.ode_rtol := 1e-10 {};
83:    cb.ode_rtol := 1e-10 {};
84:  END set_ode;
85:
86:  (*== SETOBS ==*)
87:  METHOD set_obs;
88:    ca.obs_id := 1;
89:    cb.obs_id := 2;
90:  END set_obs;
```


x_d implica que a sua taxa de variação com o tempo, dx_d/dt , também seja declarada como uma variável algébrica. Nas linhas 26–27 do Quadro 4.5 apresenta-se esta dupla declaração para o caso das variáveis ca e cb .

À partida pode pensar-se que a introdução destas novas variáveis pode modificar o número de graus de liberdade do problema original, mas tal não acontece. A modelização em estado transiente incrementa o número de atributos associado às variáveis do modelo. Através da utilização dos novos métodos, os novos atributos podem ser definidos de forma adequada com vista a acomodar um conjunto de variáveis extra que a componente diferencial do problema obriga a declarar. Em termos práticos, o método `set_ode` serve para estabelecer a classificação das variáveis no contexto dinâmico do problema, indicando ao LSODE como deve tratar e relacionar as variáveis do modelo

Os novos atributos introduzidos pelo método `set_ode` são os seguintes:

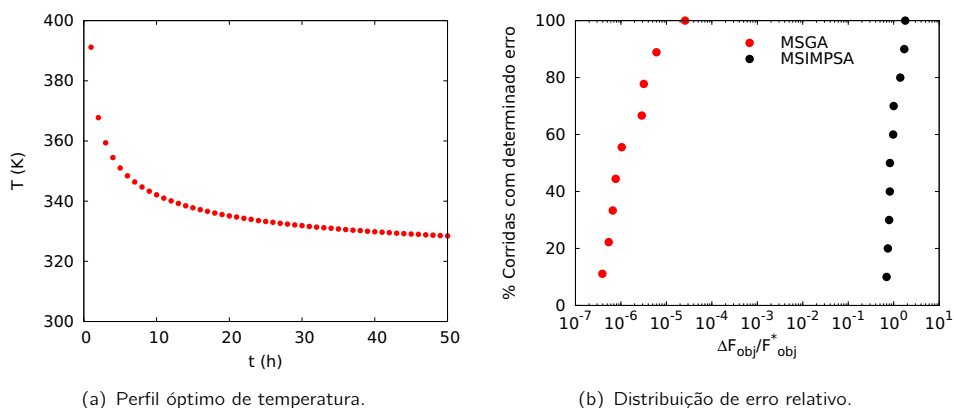
- `ode_type`—Permite designar o tipo de variável em função do valor que lhe seja atribuído. Existem quatro situações possíveis para o valor de `ode_type`:
 - variável independente, `ode_type = -1`
 - variável algébrica, `ode_type = 0`
 - variável de estado, `ode_type = 1`
 - variável diferencial, `ode_type = 2`
- `ode_id`—Este atributo é usado para associar as variáveis algébricas com os seus termos diferenciais (lados esquerdos das equações). Para o efeito são usados números inteiros positivos, quaisquer, desde que seja respeitada a regra de não existir nenhum par que partilhe o mesmo número;
- `ode_rtol` e `ode_atol`—Correspondem, respectivamente, aos atributos associados à definição das tolerâncias relativa e absoluta de cada variável no contexto do integrador LSODE.

No que diz respeito ao método `set_obs`, este utiliza apenas o atributo `ods_id`, que pela atribuição de números inteiros positivos, permite criar um conjunto de variáveis monitorizável durante a integração.

O método `seqmode` permite dispor as variáveis correctamente sem aumentar o número de graus de liberdade do problema. A sua função é gerir o comportamento das variáveis com base no valor do seu atributo `.fixed`, que determina se são, em determinado momento, variáveis de decisão. Como se estabelece na linha 58 do Quadro 4.5, a temperatura (T) é a variável assinalada como variável de decisão pelo método `specify`. Apesar de ca e cb serem as variáveis de estado do problema, o seu atributo `.fixed` é

Tabela 4.3 Problemas P.BATCHT. Melhores resultados obtidos.

Problema	Optimizador	F_{obj}	Nº Avaliações
P. BATCHT	MSIMPSA	0.6065 mol/mol	38488
	MSGA	0.6107 mol/mol	45101

**Figura 4.7** Problema P. BATCHT. Resultados obtidos.

assinalado com o valor TRUE. Isto acontece porque a modelização dinâmica em ASCEND vê o problema a dois níveis diferentes.

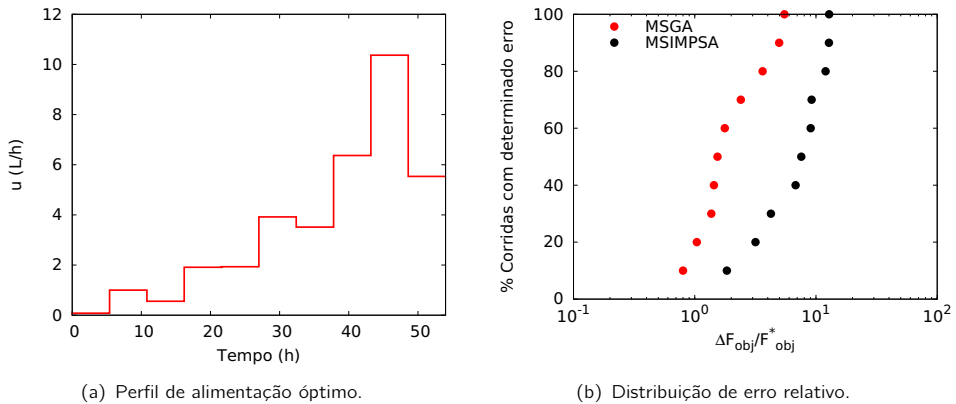
No contexto da formulação original, o exemplo apenas tem um grau de liberdade ocupado pela variável de controlo. No entanto, no contexto da sua avaliação ao longo do tempo, em cada iteração do integrador são calculados os lados esquerdos das equações, e para a realização deste passo, as variáveis de estado ca e cb são consideradas de decisão, passando novamente a variáveis de estado para actualização da função objectivo nesse instante de tempo.

Relativamente ao problema P. BATCHT, os resultados obtidos encontram-se apresentados na Tabela 4.3 e Figura 4.7, em acordo com os níveis de desempenho publicados em abordagens prévias ao problema (Faber et al., 2005).

Chen e Hwang (1990) propõem um caso de estudo que descreve a produção de etanol por *Saccharomyces cerevisiae* num reactor semi-batch. Pretende-se maximizar a produção de etanol por regulação do caudal de alimentação do reactor. O problema pode ser resolvido utilizando uma decomposição por partes da variável de decisão em função do tempo. Diferentes abordagens são apresentadas por Banga et al. (1997), sendo a formulação mais detalhada e o seu modelo ASCEND apresentados no Apêndice A.5.

Tabela 4.4 Problemas P.FBE. Melhores resultados obtidos.

Problema	Optimizador	F_{obj}	Nº Avaliações
P.FBE	MSIMPSA	19945.7 kg	39912
	MSGGA	20286.2 kg	204838

**Figura 4.8** Problema P.FBE. Resultados obtidos.

A Tabela 4.4 mostra os resultados obtidos para o tempo de partida de 54 h. O volume máximo do reactor é estabelecido, assim como os valores mínimo e máximo do caudal de alimentação do reactor. O melhor resultado foi encontrado utilizando o MSGGA, 20286.0 kg, próximo do melhor valor conhecido apresentado por Salcedo et al. (2003), no caso 20318.9 kg. O perfil de alimentação para a situação com melhor desempenho é apresentado na Figura 4.8, onde se registam as distribuições de erro relativo associadas às corridas levadas a cabo com ambos os optimizadores.

4.3 Estratégia de Comunicação: Maior usabilidade

Com base na experiência adquirida pela aplicação da estratégia integrada a casos de estudo, procura-se estabelecer uma estrutura de configuração que garanta a maior independência possível da capacidade de aplicação desta estratégia face ao tipo de problema. Nesse sentido, pretende-se sistematizar a forma de interação com a estratégia proposta numa base sistemática e bem definida, a qual funciona segundo o esquema apresentado na Figura 4.9.

Para isso, as possibilidades oferecidas pela estratégia foram estabelecidas segundo

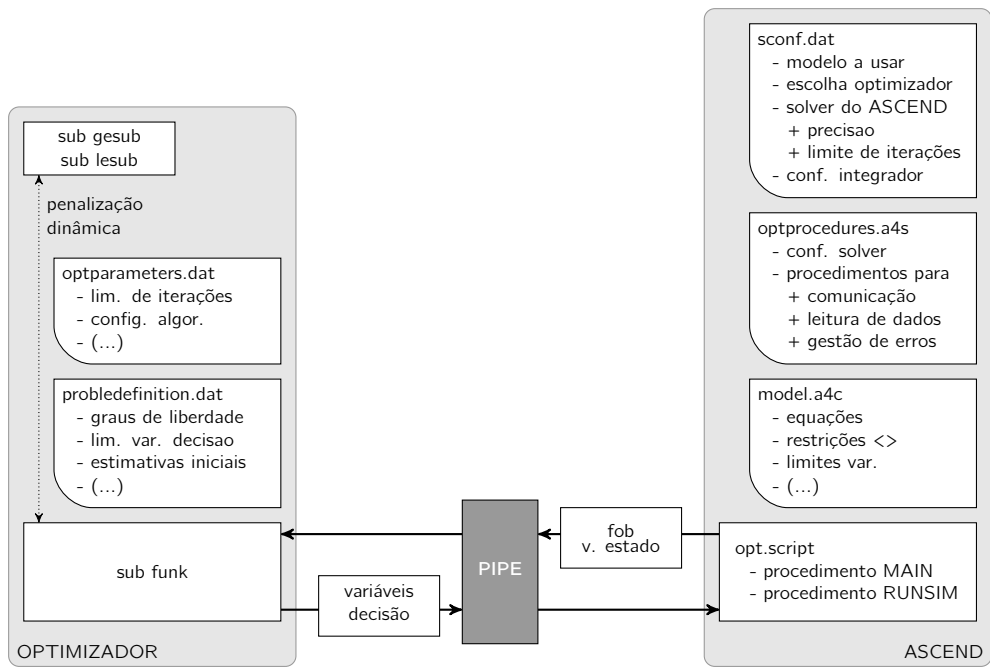


Figura 4.9 Esquema de comunicação II.

opções configuráveis. Os procedimentos mais comuns, relacionados com a leitura de dados, estabelecimento de comunicação, leitura dos modelos, etc., foram incorporados num *script* disponibilizado no arranque do simulador. No que diz respeito ao optimizador, a rotina `func` é reescrita para funcionar de uma forma independente dos graus de liberdade do problema, exigindo intervenção do utilizador nos casos onde se necessite de implementar penalização dinâmica.

Quanto ao bloco do simulador, nele são incluídos quatro elementos diferentes:

- `model.a4c`

Representa o problema a ser resolvido: equações, restrições e diversos atributos associados às variáveis, tais como domínios ou valores nominais, são aqui definidos;

- `optprocedures.a4s`

É um conjunto de procedimentos generalistas carregados imediatamente quando o ASCEND é inicializado. A instrução de chamada ao optimizador através do *pipe*, a configuração dos procedimentos de leitura de dados e detecção de erros são algumas das suas funções. O ficheiro de configuração do simulador, `.ascendrc`, é utilizado para carregar os procedimentos incluídos no `optprocedures.a4s` (ver Apêndice C). Este *script* não requer a intervenção do utilizador;

- `opt.script`

É onde o procedimento dependente do problema está localizado — RUNSIM. Define diversas instruções: leitura do vector decisão, configuração e execução da simulação, avaliação das restrições e validação da função objectivo obtida. Com base num modelo, só é preciso editar a atribuição das variáveis de decisão. Caso existam procedimentos de inicialização, é aqui que eles devem ser incluídos. Pode também ser utilizado se, fora do âmbito do problema de optimização, o utilizador pretenda realizar outro tipo de estudos sobre a simulação baseada no modelo `model.a4c`.

- `sconf.dat`

É a última peça incluída no bloco do simulador. A sua finalidade é providenciar flexibilidade pela adição de algumas opções de configuração (consultar os Apêndices A.3 e A.4 para ver alguns exemplos deste ficheiro), nomeadamente:

- escolha do modelo a utilizar;
- escolha do optimizador;
- definição de alguns parâmetros importantes do *solver* do ASCEND.

Todas estas peças interagem num puzzle representado na Figura 4.9, que mostra a estrutura dos blocos de software e o seu relacionamento com os ficheiro de dados e modelos. Numa outra perspectiva, relacionada com a sequência de acontecimentos que sucede na abordagem de um problema de optimização com a estratégia proposta, a Figura 4.10 mostra a sequência das principais acções que sistematicamente são desencadeadas na aplicação da metodologia.

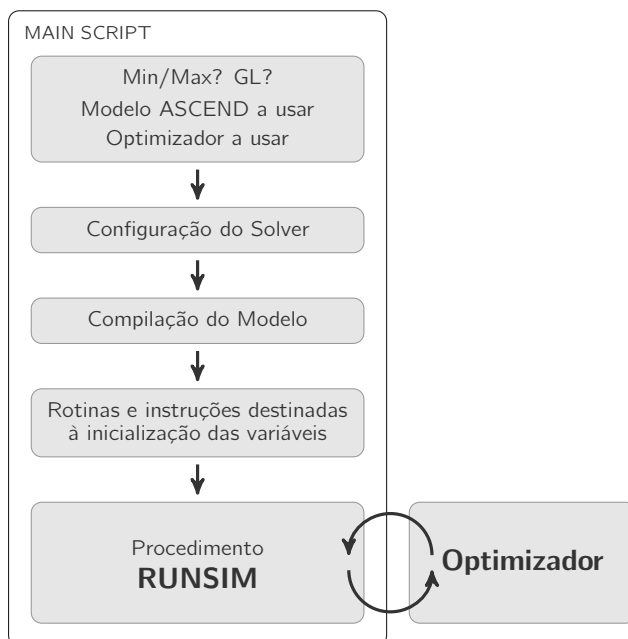


Figura 4.10 Estrutura geral do script de ligação. Sistematização.

4.4 Conclusões

A consolidação de uma estrutura de comunicação entre o simulador ASCEND e os optimizadores estocásticos MSGA e MSIMPISA, emerge como a base de construção de uma estratégia integrada de simulação e optimização. A modificação e aperfeiçoamento desta plataforma foi conduzida no sentido de possibilitar a implementação das características desejadas para uma metodologia deste tipo.

A experimentação com exemplos de aplicação permitiu desenvolver a estrutura proposta à partida para a utilização conjugada dos algoritmos disponíveis. Apresentando um desenho final estruturado e definido com base em blocos independentes, a aplicação da metodologia revela-se satisfatória na procura dos melhores valores da função objectivo em alguns casos de estudo com referência na literatura (Silva e Salcedo, 2005, 2006).

Todavia, regista-se um menor desempenho ao nível do tempo computacional necessário à resolução dos problemas, situação que, em função da utilização de componentes baseados em *scripting* já se antecipava. No contexto de casos de estudo simples, tal discrepância não acarreta problemas significativos, podendo vir a revelar um maior impacto na análise de problemas mais complexos.

Nomenclatura geral

Caracteres romanos

F	Função Objectivo
F_{obj}	Função Objectivo
F_{obj}^*	Melhor valor encontrado para a Função Objectivo
\mathbf{g}	Vector das restrições de desigualdade
g_j	Restrição de desigualdade de índice j no vector \mathbf{g}
\mathbf{h}	Vector das restrições de igualdade
h_i	Restrição de igualdade de índice i no vector \mathbf{h}
m_1	Número de restrições de igualdade
m_2	Número de restrições de desigualdade
n	Número de variáveis
p	Número de variáveis contínuas
q	Número de variáveis de estado contínuas
V_D	Conjunto de variáveis de decisão
$V_{P, ABSTRIP}$	Variáveis de decisão do problema P. ABSTRIP
\mathbf{x}	Vector das variáveis contínuas
x_k	Variável contínua de índice k do vector \mathbf{x}
\mathbf{y}	Vector das variáveis discretas
y_k	Variável discreta de índice k do vector \mathbf{y}

Caracteres gregos

α_k	Limite inferior de x_k
β_k	Limite superior de x_k
γ_k	Limite inferior de y_k
ξ_k	Limite superior de y_k
ΔF_{obj}	Diferença entre o valor de F_{obj} e F_{obj}^*

Acrónimos e siglas

Algol	ALGOrithmic Language
ASCEND	Advanced System for Computations in ENgineering Design
FORTTRAN	FORmula TRANslator
GL	Graus de Liberdade
GUI	Graphical User Interface
I/O	Input/Output
LISP	LISt Processing
LSODE	Livermore Solver for Ordinary Differential Equations
Matlab	MATrix LABoratory
MINLP	Mixed-Integer Nonlinear Programming

MSGA	Minlp Salcedo Gonçalves Azevedo
MSIMPSA	Minlp SIMPLex Simulated Annealing
NLP	Nonlinear Programming
Perl	Practical Extraction and Report Language
Rexx	Restructured Extended Executor Language
Tcl	Tool Command Language

Nomenclatura do problema P . ABSTRIP

Caracteres romanos

B_1	Bomba nº 1
B_2	Bomba nº 2
D_a	Diâmetro da coluna de absorção, m
D_s	Diâmetro da coluna de regeneração, m
D	Caudal molar de destilado (produto), kmol/h
F	Caudal molar de alimentação à coluna de absorção, mol/h
G	Caudal molar de saída do topo da coluna de absorção, kmol/h
L	Caudal molar de solvente reciclado, kmol/h
P	Pressão de operação da coluna de regeneração, mmHg
P_1	Permutador de calor nº 1, condensador da coluna de desorção
P_2	Permutador de calor nº 2, reebulidor da coluna de desorção
P_3	Permutador de calor nº 3
P_4	Permutador de calor nº 4
R	Razão de refluxo
T_1	Temperatura de operação da coluna de absorção, K
T_2	Temperatura da corrente de alimentação à coluna de regeneração, K
T_3	Temperatura da corrente de topo da coluna de regeneração, K
T_4	Temperatura da corrente de cauda da coluna de regeneração, K
T_5	Temperatura do solvente reciclado depois de arrefecido, K
T_0	Temperatura de saída da água de arrefecimento, K
T_i	Temperatura de entrada da água de arrefecimento, K
T_f	Temperatura do prato de alimentação na coluna de regeneração, K
T_{ref}	Temperatura de referência, K
T_s	Temperatura do vapor saturado no vaporizador, K
W_1	Caudal mássico de água alimentada ao condensador, kg/h
W_2	Caudal mássico de vapor alimentado ao vaporizador, kg/h
W_4	Caudal mássico de água alimentada ao permutador de calor P_4 , kg/h
x_1	Fracção molar na corrente de alimentação à coluna de desorção
x_2	Fracção molar na corrente de cauda da coluna de regeneração
x_3	Fracção molar na corrente de topo da coluna de regeneração
y_1	Fracção molar do componente a remover

y_2 Fracção molar na corrente de topo da coluna de absorção

Capítulo 5

Aplicação de estratégias integradas a problemas NLP/MINLP

Sumário do Capítulo

No presente capítulo pretende evidenciar-se a importância que o investimento na concepção de estratégias integradas de simulação e optimização pode representar como mais valia na solução de problemas. Nesse contexto, aborda-se a expansão da metodologia SIMOP à aplicação de formulações MINLP. Apresenta-se um caso de estudo não linear que corresponde a um processo de destilação reactiva, o qual é explorado nas suas formulações NLP e MINLP através da utilização da estratégia conjugada do MSGA ou MSIMPISA com o ASCEND, e da metodologia SIMOP.

A construção de simulações que ofereçam garantias numa gama alargada de condições operatórias é uma tarefa que deve tomar em consideração preocupações diversas. Em Engenharia Química, salienta-se o aparecimento frequente de sistemas de equações não lineares nas estratégias de avaliação dos problemas, situação que acarreta normalmente o aumento de dificuldade na resolução dos mesmos.

A escolha do conjunto mais adequado de variáveis de decisão é por isso importante. Esse exercício pode traduzir-se na obtenção de conjuntos de saída com menor custo de resolução, revelando por isso, taxas de sucesso mais satisfatórias aquando da aplicação de algoritmos estocásticos.

Dado um determinado conjunto de saída, resultado ou não de uma análise estrutural prévia, existem ainda preocupações de carácter numérico às quais o utilizador não deve estar alheio na abordagem ao problema.

Na modelização orientada por equações, a opção por linguagens de programação com um nível de abstracção mais baixo que o das linguagens declarativas exige uma codificação atenta, visando minorar o custo despendido com a normalmente morosa etapa de depuração dos modelos.

Estas são algumas das ideias que motivaram o desenvolvimento de ferramentas adicionais no âmbito da linha de investigação iniciada com o desenvolvimento do MSGA e MSIMPISA, continuada por Lima (2005), que propõe uma estratégia integrada, o SIMOP (SIMulation for OPTimization), cujo objectivo passa por abordar transversalmente algumas das preocupações acima enunciadas.

5.1 SIMOP - SIMulation for OPTimization

O SIMOP (Lima, 2005) é uma metodologia integrada para a simulação e optimização de Processos Químicos com aplicabilidade demonstrada (Lima et al., 2004, 2006). O seu propósito é a construção de simulações estruturalmente optimizadas, com vista à avaliação do desempenho dos processos estudados, usando para isso os princípios da optimização estocástica global, aplicando os algoritmos MSGA e MSIMPISA. A sua implementação contempla quatro tarefas essenciais:

1. Interpretação de modelos orientados por equações;
2. Escolha de conjuntos de variáveis de decisão segundo critério pré-estabelecido;
3. Integração de algoritmos numéricos para resolução de sistemas de equações resultantes da interpretação dos modelos;
4. Definição automática de procedimentos numéricos que permitam implementar os problemas de optimização resultantes com o MSGA e MSIMPISA.

A forma como estas valências se encadeiam em todo o procedimento de utilização do SIMOP é apresentada no diagrama conceptual da metodologia, apresentado na Figura 5.1 tal como proposto por Lima (2005).

O SIMOP foi desenvolvido para resolver problemas NLP do tipo definido pelas Equações 5.1 a 5.4,

$$\min F(\mathbf{x}) \quad (5.1)$$

$$\text{sujeito a } h_i(\mathbf{x}) = 0 \quad i = 1, \dots, m_1 \quad (5.2)$$

$$g_j(\mathbf{x}) \geq 0 \quad j = 1, \dots, m_2 \quad (5.3)$$

$$\alpha_k \leq x_k \leq \beta_k \quad k = 1, \dots, n \quad (5.4)$$

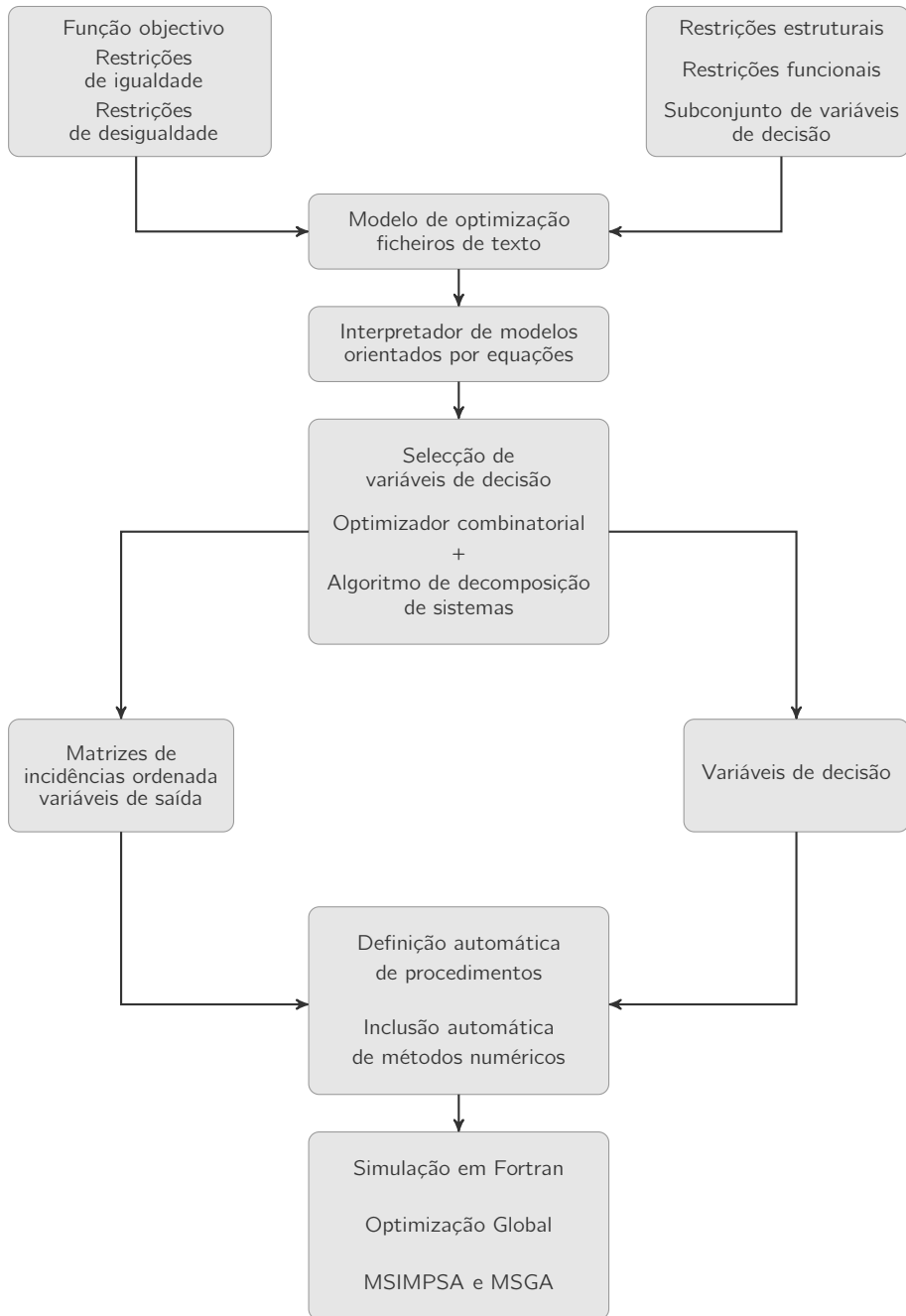


Figura 5.1 Diagrama conceptual do SIMOP. Adaptado de Lima (2005).

onde $n > m_1$ fazendo com que o problema possa ser traduzido segundo um sistema de equações rectangular. A sua resolução implica a adequada especificação dos graus de liberdade, num modelo que garanta ser isento de equações redundantes, obtendo-se como resultado um sistema de equações quadrado, em que o número de variáveis a determinar corresponde ao número de equações existentes no sistema. Biegler et al. (1997) salientam a importância que a resolução adequada do conjunto de equações resultante representa para o sucesso da aplicação de estratégias orientadas por equações.

Salcedo e Lima (1999) propõem uma metodologia para escolha de variáveis de decisão utilizando o algoritmo de Ledet e Himmelblau (1970), o MATRIX. Considerando esse exercício de escolha como um problema de optimização formal, procura-se minimizar a dificuldade de resolução do sistema, ou conjunto de sistemas, de equações resultantes da transcrição do problema segundo um modelo orientado por objectos. Para esse efeito, estão disponíveis diferentes funções objectivo que representam, sob o ponto de vista estrutural, alternativas para o critério de medição do custo de resolução do sistema. A obtenção de um conjunto de saída pode ainda ser condicionado pela inclusão de restrições estruturais, baseadas no conhecimento e análise prévios do problema. Entendem-se por restrições estruturais condições que o utilizador impõe à etapa de decomposição do sistema. Podem corresponder a especificações que excluem determinadas variáveis do conjunto de variáveis de reciclagem ou, por exemplo, podem servir para indicar equações da qual uma ou mais variáveis não devam ser variáveis de saída.

A aplicação do algoritmo MATRIX, com vista à criação de simulações em FORTRAN para optimização pelos algoritmos MSGA e MSIMPSA, deu o mote para o investimento desenvolvido na concepção e articulação de diferentes procedimentos de análise de problemas com vista à optimização numa só metodologia.

A interpretação de modelos orientados por equações disponível no SIMOP tem como finalidade melhorar a forma como o utilizador interage com os procedimentos existentes. O formato utilizado pelos mecanismo I/O do MATRIX não é de todo mais confortável para o utilizador. A leitura de dados obriga a criar um ficheiro de texto, `matrix.dat`, construído com base na tradução do sistema de equações para um registo de incidências na forma de pares (equação,variável). É com base nesta representação que o MATRIX avalia o sistema estrutural, determinando o conjunto de saída na forma de uma matriz de incidências ordenada — ficheiro `matrix.res`.

A utilização do SIMOP pretende garantir ao utilizador um nível de abstracção mais elevado. As equações do problema podem ser directamente codificadas num ficheiro de texto designado como `equations.dat`. O SIMOP interpreta esse ficheiro, identificando as ocorrências que lhe permitem construir a matriz de incidências na forma apropri-

ada do `matrix.dat`, garantindo que o utilizador interage com o algoritmo MATRIX de uma forma mais transparente e simples. Para ilustrar as diferenças existentes, nos Quadros 5.1, 5.2 e 5.3 apresentam-se respectivamente os ficheiros `matrix.dat`, `matrix.res` e `equations.dat` para o problema P.MK, apresentado na Secção 3.3.

As operações de transformação e manipulação de elementos simbólicos são efectuadas com base na ferramenta computacional Matlab[®] e do conjunto de procedimentos incluídos na sua *Symbolic Math Toolbox™*. Tal como acontece com as equações, a especificação de todas as outras componentes do problema é efectuada com o auxílio de ficheiros de texto. A correspondência entre cada uma das diferentes componentes e os respectivos ficheiros onde são incluídas é apresentada na Tabela 5.1. Uma vez que a abordagem detalhada do formato de cada uma destas estruturas está fora do âmbito deste trabalho, o leitor é remetido para a informação dedicada a esses conceitos presente no manual de utilização do SIMOP apresentado por Lima (2006).

A incorporação de algoritmos numéricos para resolução dos sistemas de equações é efectuada com base na leitura e análise da configuração de saída proposta pelo MATRIX.

Quadro 5.1 Problema P.MK. Ficheiro `matrix.dat`.

```
1: 6
2: 1 3 1 4 1 6 2 1 2 4 2 8 2 9 2 11 3 8 4 1
3: 4 2 4 4 4 8 4 9 4 10 4 11 4 12 5 6 5 7 6 5
4: 6 10 0
5: 0
6: 0
```

Quadro 5.2 Problema P.MK. Ficheiro `matrix.res`.

```
1: *output matrix ( 1, 1)*
2: *eqn.* *var.*
3: 1 4 1 . . . .
4: 3 8 . 1 . . . .
5: 2 1 1 1 1 . . .
6: 5 7 . . . 1 . .
7: 6 10 . . . . 1 .
8: 4 12 1 1 1 . 1 1
```

Quadro 5.3 Problema P.MK. Ficheiro `equations.dat`.

```
1: 'Problema P.MK - equations.dat'
2: F1-F0-FR
3: F1-F1*XAP-Vmol*XAP*a1*exp(-E1/R/T)
4: Vmol-V*roA*MMA
5: Vmol*XAP*a1*exp(-E1/R/T)-F1*XBP-Vmol*XBP*a2*exp(-E2/R/T)
6: FR-FXAP
7: FB-XBP
```

O SIMOP transcreve a matriz de incidências do `matrix.res` segundo as equações interpretadas *à priori*, rearranjando-as em função da ordem de cálculo estabelecida. A metodologia identifica os sistemas de equações presentes e estrutura as rotinas numéricas necessárias à sua resolução. No caso de blocos singulares, com apenas uma equação, o SIMOP tenta em primeiro lugar explicitar a variável de saída estabelecida para essa equação. Caso a explicitação não seja possível, a equação é resolvida numericamente por intermédio do método de Brent (Press et al., 1992). Na construção das simulações em FORTRAN, o SIMOP garante diferentes opções para a estruturação do código. O utilizador pode escolher realizar, ou não, decomposição dos sistemas de equações ou testar diferentes alternativas para o cálculo do jacobiano. As simulações criadas incluem também rotinas dedicadas à gestão da ocorrência de incongruências matemáticas, resultado que acciona o mecanismo de penalização dos optimizadores (Lima, 2006, 2005).

O resultado final da análise do SIMOP pode assim ser definido como procedimentos numéricos em FORTRAN adaptados para o MSGA e MSIMPISA. O SIMOP tem ainda a capacidade de escrever automaticamente um directório de projecto que inclui: `makefile` adaptado à arquitectura da máquina cliente, código fonte da simulação em FORTRAN configurado com base em escolhas prévias do utilizador assim como os ficheiros de configuração dos optimizadores. Lima (2005) refere que as capacidades de modelização do SIMOP ficam aquém das possibilidades oferecidas por aplicações como o GAMS ou o ASCEND. Isto prende-se com o facto de o SIMOP não possuir uma linguagem declarativa com sintaxe própria, estando apenas baseado na interpretação de informação

Tabela 5.1 SIMOP. Correspondência entre as diferentes componentes do problema e os ficheiros de texto utilizados para sua representação.

Componente do Problema	Ficheiro de texto associado
Restrições de igualdade	<code>equations.dat</code>
Parâmetros das equações	<code>parameter_eq.dat</code>
Restrições de desigualdade	<code>constraints.dat</code>
Função objectivo	<code>fobj.dat</code>
Parâmetros da função objectivo	<code>parameter_fobj.dat</code>
Restrições estruturais	<code>rfuncional.dat</code>
Especificação parcial ou total do conjunto de variáveis de decisão	<code>vp_fix.dat</code>
Domínios das variáveis, pontos iniciais e factores de escala	<code>var.dat</code>

contida em ficheiros de texto segundo um formato rígido. Todavia, Lima (2005) realça a forma como esta metodologia se diferencia na abordagem aos problemas de optimização pela escolha optimizada das variáveis de decisão, factor com importância reconhecida na configuração estrutural dos problemas (Edie e Westerberg, 1971).

5.2 SIMOP para formulações MINLP

A importância que a formulação de problemas do tipo MINLP tem em Engenharia Química foi já enfatizada na Secção 2.3, reconhecendo-se o benefício que a incorporação de variáveis discretas representa na abordagem a problemas de projecto de unidades e/ou decisão sobre a configuração de diagramas processuais. Nestes casos, a aproximação recorrendo apenas a variáveis contínuas poderá condicionar a obtenção do valor mais favorável à optimização do problema. Isto acontece por várias razões, como são: dificuldades na exploração de valores limite dos domínio de procura, a incapacidade de representar adequadamente a solução do ponto de vista físico, ou prático, e também pela boa especificação no dimensionamento de equipamento.

Uma vez que os optimizadores estocásticos MSGA e MSIMPISA conseguem lidar com variáveis discretas, leia-se formulações MINLP, a utilização do SIMOP como ferramenta que disponibilize procedimentos numéricos dedicados a estes problemas apresenta-se como uma evolução natural e desejável. Deste modo, um dos objectivos da presente tese passa pela implementação de tais funcionalidades na metodologia, tarefa baseada na análise, adaptação e modificação do código fonte SIMOP.

No contexto do diagrama conceptual do SIMOP apresentado na Figura 5.1, podem identificar-se os pontos essenciais com vista à adaptação desta metodologia para sua aplicação a problemas MINLP:

- **Estudo, anotação e decomposição do código-fonte do SIMOP.** O primeiro passo no estudo do SIMOP corresponde a um exercício que, em certa medida, pode ser visto na óptica das aplicações de engenharia invertida. Com ele se pretende estruturar o diagrama de fluxo da aplicação, identificando as suas funcionalidades assim como respectiva sequência de execução das mesmas.

O principal objectivo deste estudo é isolar os procedimentos de leitura, atribuição e manipulação do vector de variáveis de decisão; assim como os de escrita das variáveis para código FORTRAN. Deverá introduzir-se sempre que necessário a separação conceptual entre os dois tipos de variáveis presentes, discretas e contínuas. Essa distinção permitirá modificar adequadamente os parâmetros para a correcta configuração do comportamento dos optimizadores, bem como para a

escrita adequada de gestão dos diferentes tipos de variáveis em memória, como sejam a sua respectiva declaração de disponibilização em estruturas como *common blocks*.

- **Identificação**, nos ficheiros de dados, **do conjunto de variáveis de decisão discretas**. À semelhança do que acontece com o ficheiro *constraints.dat*, onde existe uma separação formal entre os dois tipos de restrições suplementares de desigualdade (de limite superior ou de limite inferior), deseja-se um formato semelhante de diferenciação para o ficheiro *vp_fix.dat*, onde as variáveis de decisão discretas possam ser diferenciadas das variáveis contínuas.

Assume-se que, na formulação de problemas na forma MINLP para trabalhar com o SIMOP, o utilizador terá toda a vantagem em incorporar as variáveis discretas do problema no seu conjunto de variáveis de decisão. As variáveis discretas estão normalmente associadas a parâmetros que influenciam directamente as configurações do sistema, o que as torna desejáveis à constituição do conjunto de variáveis de decisão. No conjunto de variáveis de projecto procura-se alocar as variáveis com impacto directo na medida de desempenho do processo, como sejam variáveis manipuláveis do ponto de vista operativo, variáveis que traduzam medidas do rendimento da eficiência dos processos, que estejam associadas a diferentes modos de operação, que digam respeito à escolha de diferentes configurações topológicas dos diagramas de fluxo ou determinem especificações que estejam relacionadas com o dimensionamento de equipamentos. Dentro de todos estes tipos há ainda preferência pelas variáveis que apresentem domínios menos amplos ou representem grandezas normalizadas, ou ainda variáveis que linearizem equações.

As variáveis discretas são empregues na representação de inúmeros exemplos de tipos de variáveis de projecto desejáveis. Em função da sua natureza, e por terem normalmente associadas amplitudes de domínio menores, a sua exploração no seu espaço de procura é habitualmente mais acessível.

No entanto, convém manter presente que a introdução de variáveis discretas contribui para o aumento do grau combinatorial dos problemas. Esse aumento de dimensão do espaço de procura global do problema é um acréscimo à complexidade de resolução do mesmo. Além disso, a utilização das variáveis discretas corresponde por norma a uma tentativa de modelizar mais rigorosamente determinado sistema, pelo que a sua inclusão pode significar um aumento da dimensão do conjunto de variáveis de projecto do problema NLP original, que passa a representar um subproblema não linear em cada um dos pontos do espaço combinatorial

do problema MINLP.

Verifica-se assim que a introdução de variáveis discretas apresenta vantagens e desvantagens. Outro pormenor importante é a análise prévia do número de variáveis discretas a considerar. Procura-se que a componente discreta modelize melhor determinadas características do sistema (promovendo uma solução eficiente), mas sem conduzir a um espaço combinatorial demasiado alargado que acarrete a perda de funcionalidade, nomeadamente pela desactivação dos procedimentos de teste nos extremos do domínio das variáveis de decisão utilizando os optimizadores estocásticos MSGA e MSIMPISA.

Do ponto de vista da matriz de incidências as variáveis são equivalentes à luz do algoritmo de decomposição, pelo que a distinção entre variáveis contínuas e discretas não é necessária. Por isso, e no que diz respeito à configuração do algoritmo MATRIX, a análise prévia do problema deverá procurar identificar pares de variáveis discretas relacionadas por uma equação. Essa situação implica que que apenas uma delas possa integrar o conjunto de variáveis de decisão.

- **Adaptação dos procedimentos que efectuem a leitura das ocorrências de variáveis nas equações do problema.** A modificação da configuração terá como base a criação de um subconjunto circunscrito às variáveis discretas devidamente identificado. Deste modo poderão adaptar-se os procedimentos numéricos presentes nos ficheiros no directório projecto, pelo que será importante determinar o número de variáveis discretas presentes bem como conhecer o seu posicionamento no vector global de variáveis.

Para isso procede-se à avaliação da dimensão dos vectores de variáveis presentes numa formulação MINLP do tipo já referenciado em 2.2.2, traduzida segundo as Equações 5.5 a 5.9,

$$\min F(\mathbf{x}, \mathbf{y}) \quad (5.5)$$

$$\text{sujeito a } h_i(\mathbf{x}, \mathbf{y}) = 0 \quad i = 1, \dots, m_1 \quad (5.6)$$

$$g_j(\mathbf{x}, \mathbf{y}) \geq 0 \quad j = 1, \dots, m_2 \quad (5.7)$$

$$\alpha_k \leq x_k \leq \beta_k \quad k = 1, \dots, p \quad (5.8)$$

$$\gamma_k \leq y_k \leq \xi_k \quad k = p + 1, \dots, n \quad (5.9)$$

e na qual se considera que o vector de variáveis de decisão do problema, designado por $v_D(\mathbf{x}, \mathbf{y})$, é composto por $n - m_1$ variáveis. O mesmo está associado a um conjunto de saída com vista ao cálculo de q variáveis de estado contínuas e $m_1 - q$ variáveis discretas. Assim, $v_D(\mathbf{x}, \mathbf{y})$ detém $p - q$ variáveis contínuas e $n - 2m_1 + q$

variáveis discretas. Estes parâmetros devem ser determinados pela modificação e adaptação dos procedimentos de manipulação simbólica já estipulados para o tratamento de variáveis contínuas da versão do SIMOP para aplicação a NLP.

Na situação comum, onde todas as variáveis discretas são parte integrante do vector decisão, dir-se-á tipicamente que $v_D(\mathbf{x}, \mathbf{y})$ será constituído por $p - m_1$ variáveis contínuas e $n - m_1$ variáveis discretas.

Com base na determinação destes parâmetros, são implementadas algumas adaptações, nomeadamente:

- Declaração das variáveis discretas como variáveis inteiras em todas as subrotinas em que as mesmas intervenham, nomeadamente na avaliação da função objectivo, no cálculo dos resíduos dos subsistemas e na avaliação das restrições suplementares.
- Preenchimento dos ficheiros de configuração do comportamento dos optimizadores, designadamente o `optim.dat` (MSGa, ver Salcedo (1992a)) e `optinesa.dat` (MSIMPsa, ver Cardoso et al. (2001)), indicando correctamente o número de variáveis discretas existentes, assim como a posição no vector decisão da que dele fizerem parte (no Apêndice A.8 apresentam-se exemplos comentados destes ficheiros).

A validação do SIMOP como metodologia para a análise de problemas MINLP foi testada com problemas apresentados na literatura por Salcedo (2001). Foram abordados os problemas P.PRISIB04, P.PRISIB08 que, dada a sua simplicidade, são remetidos respectivamente para os Apêndices A.6 e A.7.

No presente capítulo apresentam-se os resultados obtidos para a construção automática de modelos MINLP com base no SIMOP para os problemas P.PRISIB15 e P.PRISIB19.

5.2.1 Caso de estudo P.PRISIB15

O problema P.PRISIB15 (Kocis e Grossmann, 1987) corresponde a uma situação de decisão sobre configuração de uma superestrutura composta por três unidades processuais, U_1 , U_2 e U_3 (Figura 5.2).

O objectivo do processo é a produção do composto C pelo consumo das matérias primas A e/ou B , numa configuração baseada na combinação das diferentes unidades disponíveis. A produção de C está limitada superiormente (Equação 5.19), pelo que se formula o problema de optimização que visa minimizar os custos associados à produção,

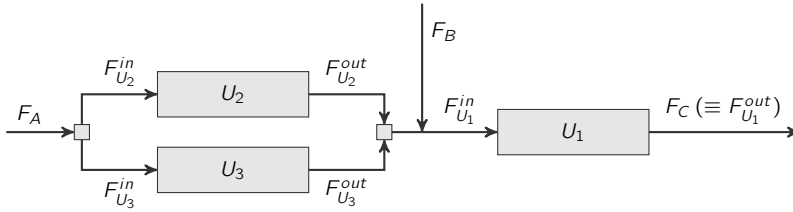


Figura 5.2 Problema P.PRSIB15. Diagrama Processual.

a qual pode ser traduzida segundo a Equação 5.10,

$$\min \left\{ F_{obj} = \sum_{k=1}^3 (y_{U_k} C_{inv,k} + C_{op,k} F_{U_k}^{out}) + \sum_{i=A,B,C} C_i F_i \right\} \quad (5.10)$$

na qual as variáveis discretas y_{U_k} , que determinam a existência das unidades U_k , são respectivamente ponderadas pelos custos de investimento associados $C_{inv,k}$. Consideram-se ainda os custos $C_{op,k}$, relativos à acomodação das correntes de saída das unidades. O cálculo da rentabilidade do processo contabiliza também o valor comercial dos diferentes componentes envolvidos, C_i . Na Equação 5.10 incidem ainda os caudais de saída das diferentes unidades, $F_{U_k}^{out}$, assim como os caudais dos diferentes componentes F_i .

O modelo do processo pode ser definido segundo as Equações 5.11 a 5.23,

- Transformações nas unidades

$$F_{U_2}^{out} - \ln(1 + F_{U_2}^{in}) = 0 \quad (5.11)$$

$$F_{U_3}^{out} - 1.2 \ln(1 + F_{U_3}^{in}) = 0 \quad (5.12)$$

$$F_C - 0.9 F_{U_3}^{in} = 0 \quad (5.13)$$

- Balanços Materiais

$$-F_{U_3}^{in} + F_B + F_{U_2}^{out} + F_{U_3}^{out} = 0 \quad (5.14)$$

$$F_A - F_{U_2}^{in} - F_{U_3}^{in} = 0 \quad (5.15)$$

- Restrições operatórias

$$F_{U_1}^{in} - 5 y_{U_1} \leq 0 \quad (5.16)$$

$$F_{U_2}^{in} - 5 y_{U_2} \leq 0 \quad (5.17)$$

$$F_{U_3}^{in} - 5 y_{U_3} \leq 0 \quad (5.18)$$

$$F_C \leq 1 \tag{5.19}$$

$$F_{U_2}^{out} \leq 5 \tag{5.20}$$

$$F_{U_3}^{out} \leq 5 \tag{5.21}$$

$$y_{U_1}, y_{U_2}, y_{U_3} \in \{0, 1\} \tag{5.22}$$

$$F_A, F_B, F_C, F_{U_1}^{in}, F_{U_2}^{in}, F_{U_3}^{in}, F_{U_2}^{out}, F_{U_3}^{out} \geq 0 \tag{5.23}$$

enquanto que na Tabela 5.2 são apresentados os valores dos coeficientes de custo que lhe estão associados. A formulação SIMOP para este caso está compilada no Quadro 5.4, fazendo já uso da separação formal da especificação das variáveis de decisão no preenchimento do `vp_fix.dat`.

Os seis graus de liberdade do problema foram preenchidos escolhendo para variáveis de decisão o conjunto $v_{P.PRISIB15} = \{y_{U_1}, y_{U_2}, y_{U_3}, F_{U_2}^{in}, F_{U_3}^{in}, F_C\}$. Foram testadas as simulações geradas automaticamente para MSGA e MSIMPISA. Os melhores valores obtidos com cada um dos optimizadores e respectivas configurações discretas encontram-se especificados na Tabela 5.3. A variabilidade dos resultados encontrados em dez corridas de optimização é ilustrada pela Figura 5.3, na qual se verifica que o MSIMPISA encontrou o óptimo em duas corridas distintas, o que explica que os seus resultados sejam representados apenas por oito pontos.

O conjunto de variáveis de decisão determinado vai ao encontro da solução anteriormente encontrada por Salcedo (2001) e que corresponde à configuração solução da estrutura que apenas utiliza as unidades U_1 e U_3 , representada pela (Figura 5.4).

Tabela 5.2 Problema P.PRISIB15. Parâmetros de custo.

(a) Custos de Investimento e Operação.			(b) Valor dos componentes.	
Unidade	C_{inv} ($10^3\$/h$)	C_{op} ($10^3\$/t$ produto)	Componente	C_i ($10^3\$/t$)
U_1	3.5	2.0	A	1.8
U_2	1.0	1.0	B	7.0
U_3	1.5	1.2	C	11.0

Tabela 5.3 Problema P.PRISIB15. Melhores resultados encontrados com MSIMPISA e MSGA.

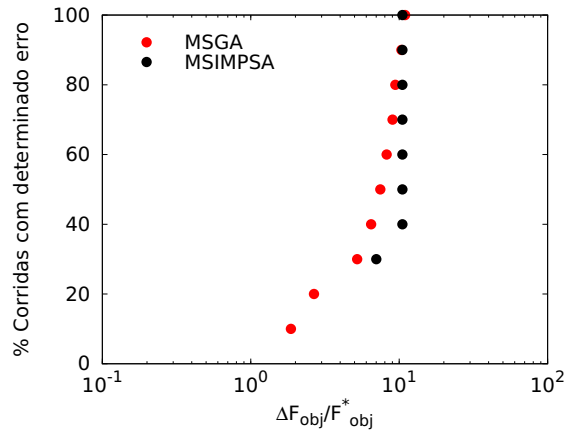
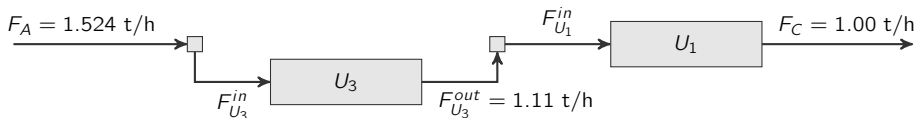
Optimizador	y_{U_1}	y_{U_2}	y_{U_3}	$F_{U_2}^{in}$	$F_{U_3}^{in}$	F_C	F_{obj} ($10^3\$/h$)	Avaliações
MSIMPISA	1	0	1	0.000	1.524	1.000	-1.923	55216
MSGA	1	0	1	0.000	1.487	1.000	-1.887	40000

Quadro 5.4 Problema P.PRSIB15. Formulação SIMOP.

```

1: '== P.RSIB015 - equations.dat ====='
2: F2out - ln(1 + F2in)
3: F3out - 1.2*ln(1 + F3in)
4: Fc - 0.9*F1in
5: -F1in + Fb + F2out + F3out
6: Fa - F2in - F3in
7:
8: '== P.RSIB15 - constraints.dat ====='
9: =>0
10: 0<=
11: F1in - 5*Y_1
12: F2in - 5*Y_2
13: F3in - 5*Y_3
14:
15: '== P.RSIB015 - fobj.dat ====='
16: fobj = (3.5*Y_1+Y_2+1.5*Y_3+F2out+1.2*F3out+1.8*Fa+7.0*Fb+11.0*Fc)*1000
17:
18: '== P.RSIB015 - vp_fix.dat ====='
19: %DISCRETE
20: Y_(i);i=1:3
21: %CONTINUOUS
22: F2in
23: F3in
24: Fc

```

**Figura 5.3** Problema P.PRSIB15. Distribuições cumulativas do erro relativo associado às corridas de otimização efectuadas com MSGA e MSIMPZA.**Figura 5.4** Problema P.PRSIB15. Configuração determinada com vista à minimização dos custos de produção.

5.2.2 Caso de estudo P.PRSIB19

O segundo caso de estudo, proposto por Floudas (1995), traduz um exemplo de sequenciação de destilação. A sua formulação contempla diversas restrições de igualdade, o que permitiu depurar a adaptação do SIMOP à manipulação de variáveis discretas no bom funcionamento dos procedimentos de decomposição do algoritmo. Atendendo aos dados disponíveis, o objectivo passa pela maximização da rentabilidade da estratégia de separação, dada pela Equação 5.24, calculada em função do valor dos principais produtos pretendidos, $P_{1,A}$ e $P_{2,B}$.

$$\max \left\{ \begin{array}{l} F_{obj} = 35 P_{1,A} + 30 P_{2,B} - 10 F_1 - 8 F_2 - F_{4,A} \\ \quad - F_{4,B} - 4 F_{5,A} - 4 F_{5,B} - 2 Y_F - 50 Y_D \end{array} \right\} \quad (5.24)$$

O modelo da estratégia de separação em estudo é dado pelas Equações 5.25 a 5.57. O papel desempenhado por cada uma das variáveis pode ser lido na representação esquemática apresentada na Figura 5.5.

- Balanços materiais

$$F_{3,A} = 0.55 F_1 + 0.50 F_2 \quad (5.25)$$

$$F_{3,B} = 0.45 F_1 + 0.50 F_2 \quad (5.26)$$

$$F_{4,A} = E_4 F_{3,A} \quad (5.27)$$

$$F_{4,B} = E_4 F_{3,B} \quad (5.28)$$

$$F_{5,A} = E_5 F_{3,A} \quad (5.29)$$

$$F_{5,B} = E_5 F_{3,B} \quad (5.30)$$

$$F_{6,A} = E_6 F_{3,A} \quad (5.31)$$

$$F_{6,B} = E_6 F_{3,B} \quad (5.32)$$

$$F_{7,A} = F_{3,A} - F_{4,A} - F_{5,A} - F_{6,A} \quad (5.33)$$

$$F_{7,B} = F_{3,B} - F_{4,B} - F_{5,B} - F_{6,B} \quad (5.34)$$

$$F_{8,A} = 0.85 F_{4,A} \quad (5.35)$$

$$F_{8,B} = 0.20 F_{4,B} \quad (5.36)$$

$$F_{9,A} = 0.15 F_{4,A} \quad (5.37)$$

$$F_{9,B} = 0.80 F_{4,B} \quad (5.38)$$

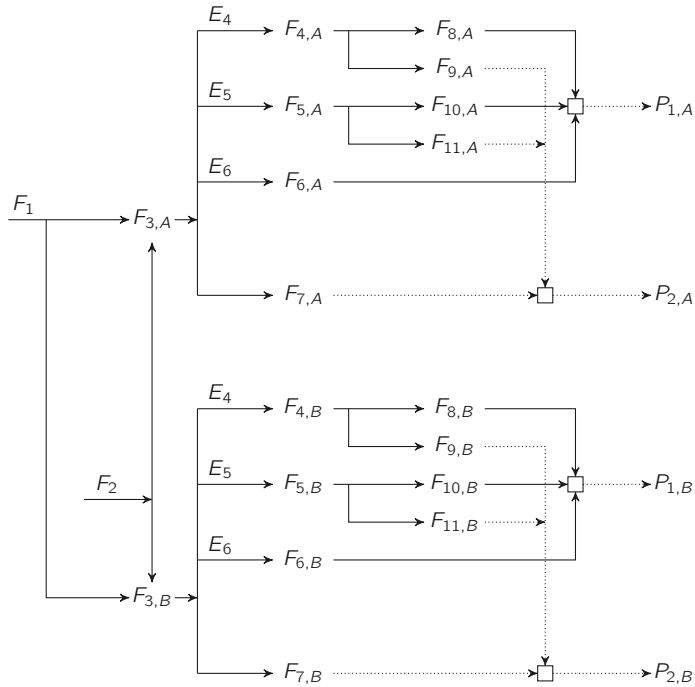


Figura 5.5 Problema P.PRSIB19. Estrutura de possibilidades para o esquema de separação.

$$F_{10,A} = 0.975 F_{5,A} \quad (5.39)$$

$$F_{10,B} = 0.050 F_{5,B} \quad (5.40)$$

$$F_{11,A} = 0.025 F_{5,A} \quad (5.41)$$

$$F_{11,B} = 0.950 F_{5,B} \quad (5.42)$$

$$P_{1,A} = F_{8,A} + F_{10,A} + F_{6,A} \quad (5.43)$$

$$P_{1,B} = F_{8,B} + F_{10,B} + F_{6,B} \quad (5.44)$$

$$P_{2,A} = F_{9,A} + F_{11,A} + F_{7,A} \quad (5.45)$$

$$P_{2,B} = F_{9,B} + F_{11,B} + F_{7,B} \quad (5.46)$$

- Restrições operatórias

$$F_{4,A} + F_{4,B} \geq 2.5 Y_F \quad (5.47)$$

$$F_{4,A} + F_{4,B} \leq 25 Y_F \quad (5.48)$$

$$F_{5,A} + F_{5,B} \geq 2.5 Y_D \quad (5.49)$$

$$F_{5,A} + F_{5,B} \leq 25 Y_D \quad (5.50)$$

$$P_{1A} \geq 4 P_{1B} \quad (5.51)$$

$$P_{2,B} \geq 3 P_{2,A} \quad (5.52)$$

$$P_{1,A} + P_{1,B} \leq 15 \quad (5.53)$$

$$P_{2,A} + P_{2,B} \leq 18 \quad (5.54)$$

$$E_4, E_5, E_6 \leq 1.0 \quad (5.55)$$

$$F_1, F_2 \leq 25.0 \quad (5.56)$$

$$Y_D, Y_F = \{0, 1\} \quad (5.57)$$

Para a resolução deste caso de estudo foi especificado o conjunto de sete variáveis de decisão encontrado pela abordagem prévia do MATRIX a este caso de estudo (Salcedo, 2001), $v_{P.PRISIB19} = \{Y_D, Y_F, E_4, E_5, E_6, F_2, F_{3,A}\}$.

A avaliação foi efectuada com simulações para ambos os optimizadores, com identificação das variáveis de decisão discretas, Y_D e Y_F , por intermédio do ficheiro de dados `vp_fix.dat` (Quadro 5.5). Os ficheiros de configuração dos optimizadores, escritos pelo SIMOP, contêm assim a informação para efectuar a procura discreta nos domínios das variáveis especificadas. Os Quadros 5.6 (linhas 2–5) e 5.7 (linhas 4–7) evidenciam respectivamente a especificação de duas variáveis discretas por entre todo o conjunto constituído por sete.

Para um grupo de dez corridas de optimização com base em diferentes sementes, ambos optimizadores permitiram encontrar o mesmo valor máximo para a função objectivo, de 510.08. O conjunto de valores para as variáveis de decisão, concordante aos apresentados previamente (Salcedo, 2001), figura na Tabela 5.4. As distribuições cumulativas de erro e a configuração do diagrama correspondente à solução encontrada apresentam-se respectivamente nas Figuras 5.6 e 5.7.

Quadro 5.5 Problema P.PRISIB19. Ficheiro `vp_fix.dat`.

```
1: == P.PRISIB19 - vp_fix.dat ==
2: % DISCRETE
3: YD
4: YF
5: % CONTINUOUS
6: E_4
7: E_5
8: E_6
9: F_2
10: F_3A
```

Quadro 5.6 Problema P.PRSIB19. Ficheiro de dados de configuração das variáveis no MSIMPISA, func.dat.

```

1: == P.RSIB19 - func.dat ==
2: ndim nminlp lim lt0 gt0 nty nstart (1=yes ,0=no)
3: 7 2 1 5 4 1 0
4: vminlp(i=1,nminlp)
5: 1 2
6: pinic(i)
7: if lim=1 xinic(i)/xfin(i)
8: 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00
9: 1.00e+00 1.00e+00 1.00e+00 1.00e+00 1.00e+00 2.50e+01 2.50e+01

```

Quadro 5.7 Problema P.PRSIB19. Ficheiro de configuração do MSGA, optim.dat.

```

1: == P.RSIB19 - optim.dat ==
2: nruns
3: 10
4: nf nminlp nty npv
5: 7 2 1 3
6: vminlp(i=1,nminlp)
7: 1 2
8: nspl m1 r1 rlm rgm shrink eps0
9: 1 5 4 0.0 0.0 0 0.5
10: nopt nestim (=1 if nf>14) nparc
11: 1 0 1
12: zecor gener
13: 0 1
14: plinic pfin factor e1 e2
15: 20 5000 2 1.0e-05 5.0e-03
16: if nopt=1 then xinic(i)/xfin(i) else xopt(i)/delt(i)
17: 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00
18: 1.00e+00 1.00e+00 1.00e+00 1.00e+00 1.00e+00 2.50e+01 2.50e+01
19: if nopt=1 and nestim=1 then xopt(i)

```

Tabela 5.4 Problema P.PRSIB19. Conjunto de variáveis decisão para o melhor valor encontrado. Resultado coincidente para MSIMPISA e MSGA para uma tolerância de 10^{-3} .

Y_D	Y_F	E_4	E_5	E_6	F_2	$F_{3,A}$	F_{obj}
1	1	0.11	0.76	0.00	25.00	16.90	510.08

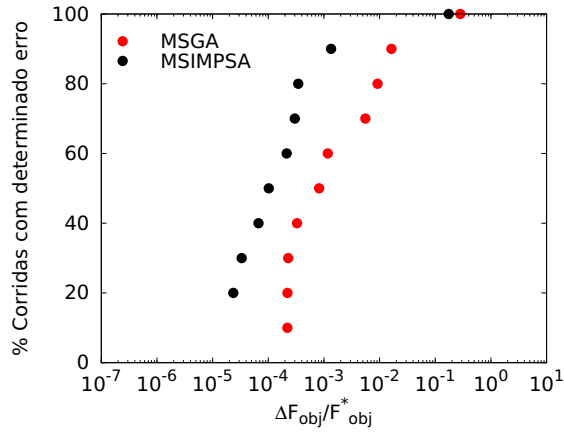


Figura 5.6 Problema P.PRSIB19. Distribuição de erro relativo.

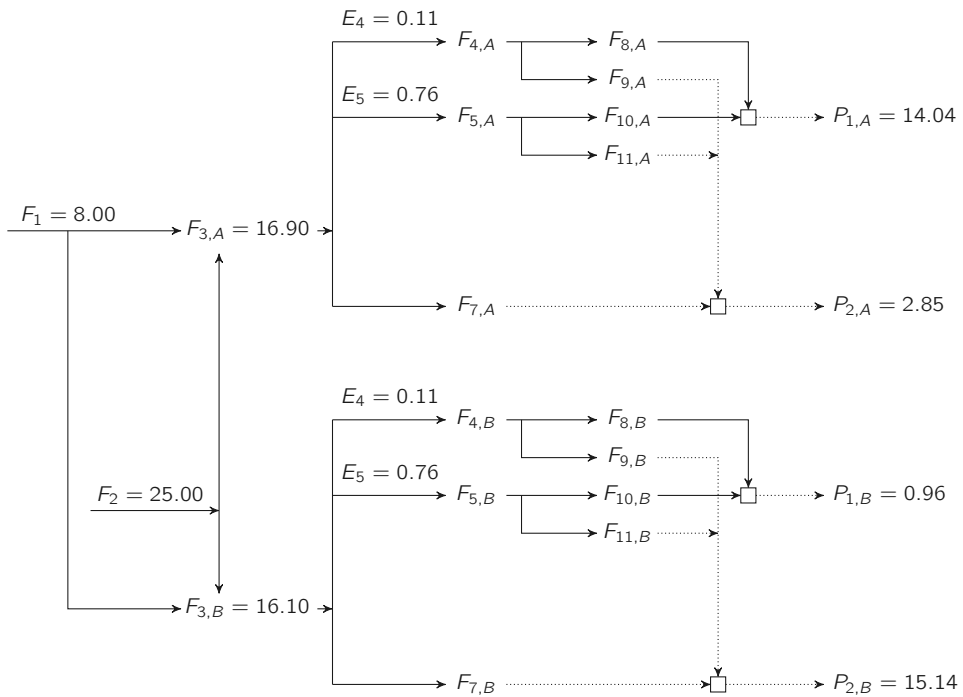


Figura 5.7 Problema P.PRSIB19. Melhor configuração encontrada.

Com a apresentação dos resultados para os casos de estudo P.PRSIB15 e P.PRSIB19, procurou-se testar a reprodutibilidade de resultados na abordagem a problemas anteriormente resolvidos com os MSGA e MSIMPISA. Atendendo aos resultados obtidos para estes dois casos de estudo, e para outros sugeridos em Salcedo (2001) e apresentados no Apêndice C, a aplicabilidade da metodologia SIMOP a problemas MINLP, com base nas adaptações previamente enunciadas sobre os procedimentos contemplados no diagrama da Figura 5.1, considerou-se estabelecida de forma satisfatória.

5.3 Caso de estudo: Destilação Reactiva, P.CDR

Na finalização do presente capítulo, pretende-se apresentar a aplicação das estratégias integradas de simulação e optimização implementadas e revistas nesta tese. O caso de estudo em análise, P.CDR, corresponde a um sistema de destilação reactiva proposto por Ciric e Gu (1994) e Ciric (1995), e revela-se adequado à aplicação destas metodologias uma vez que apresenta alta não linearidade e elevada complexidade, consequência da grande incidência de ciclos inerentes aos sistemas de destilação (Morton, 2003).

Pretende-se assim estudar a abordagem da estratégia integrada proposta no Capítulo 4, no caso com recurso ao ASCEND como avaliador de desempenho, para ambas as formulações NLP e MINLP. Resultados de abordagens anteriores ao modelo NLP e MINLP efectuadas por Cardoso et al. (2000) e Lima et al. (2006) são mantidas presentes como referências de comparação.

A formulação MINLP foi também estudada pela aplicação da nova versão do SIMOP adaptada a este tipo de problemas. Neste caso particular encontraram-se algumas dificuldades ao nível da modelização, que resultou na avaliação distribuída de subproblemas NLP, opção que coincide com o procedimento implementado na avaliação da formulação MINLP com o simulador ASCEND.

5.3.1 Descrição e estudo preliminar do modelo

O sistema em estudo corresponde a uma unidade para a produção de etilenoglicol (componente 3), processo que decorre da reacção entre o óxido de etileno (componente 1) e água (componente 2) — Equação 5.58. Pretende-se evitar a formação de dietilenoglicol (componente 4) no sistema, um produto indesejado que deriva da reacção entre o etilenoglicol e o óxido de etileno — Equação 5.59.



O processo acontece numa unidade de destilação reactiva (Figura 5.8) por forma a promover simultaneamente a reacção e separação entre os diversos componentes.

A taxa e distribuição da alimentação são desconhecidas, procurando-se estabelecer as condições operatórias que garantam uma produção desejada de etilenoglicol, P_3 , com um valor de 25 kmol/h.

O objectivo do problema passa pela minimização do custo anual de operação da unidade. O respectivo modelo é proposto segundo uma formulação MINLP (Ciric e Gu, 1994) onde o número de andares (N) está incluído no conjunto de variáveis de projecto, originando $3N+1$ graus de liberdade (Cardoso et al., 2000; Lima et al., 2006). A função objectivo (F_{obj}) que deve ser minimizada corresponde à Equação 5.60, englobando os custos de operação e dimensionamento da unidade.

$$\begin{aligned}
 F_{obj} = & c_0 + \sum_{i=1}^{N_c} \left[c_i \sum_{k=1}^N F_{i,k} \right] + c_H Q_B + c_W Q_C + \dots \\
 & + c_T D^{1.55} \sum_{k=1}^N \left(H_{min} + 1.27 \frac{W_k}{D^2} \right) + \dots \\
 & + c_{SH} D \left[H_0 + \sum_{k=1}^N \left(H_{min} + 1.27 \frac{W_k}{D^2} \right) \right]^{0.802}
 \end{aligned} \tag{5.60}$$

Na avaliação do desempenho económico da unidade são assim considerados os encargos com a alimentação dos reagentes ($F_{i,k}$) e do fornecimento de vapor e da água para arrefecimento em função dos consumos energéticos do reebulidor e do condensador (Q_B e Q_C). Também é avaliado o custo de dimensionamento da coluna através do seu diâmetro (D) e altura (calculada usando H_{min} e H_0), esta última função do volume de líquido em cada prato (W_k). As variáveis são ponderadas por factores de custo c_0 , c_T , c_i , c_W , c_H e c_{SH} , com vista à obtenção do valor da função objectivo (F_{obj}). O problema está ainda sujeito a restrições de desigualdade que especificam limites para os caudais de líquido e vapor (Equações 5.84 e 5.85) bem como para o volume de líquido em cada prato (Equação 5.86).

A descrição detalhada do modelo é apresentada pelas Equações 5.61 a 5.86:

- Balanços materiais no primeiro andar

$$\begin{aligned}
 F_{i,1} - L_1 x_{i,1} + L_2 x_{i,2} + V_0 x_{i,1} - \dots \\
 - V_1 K_{i,1} x_{i,1} + \sum_{j=1}^R v_{i,j} \xi_{j,1} = 0; \quad i = 1, \dots, N_c
 \end{aligned} \tag{5.61}$$

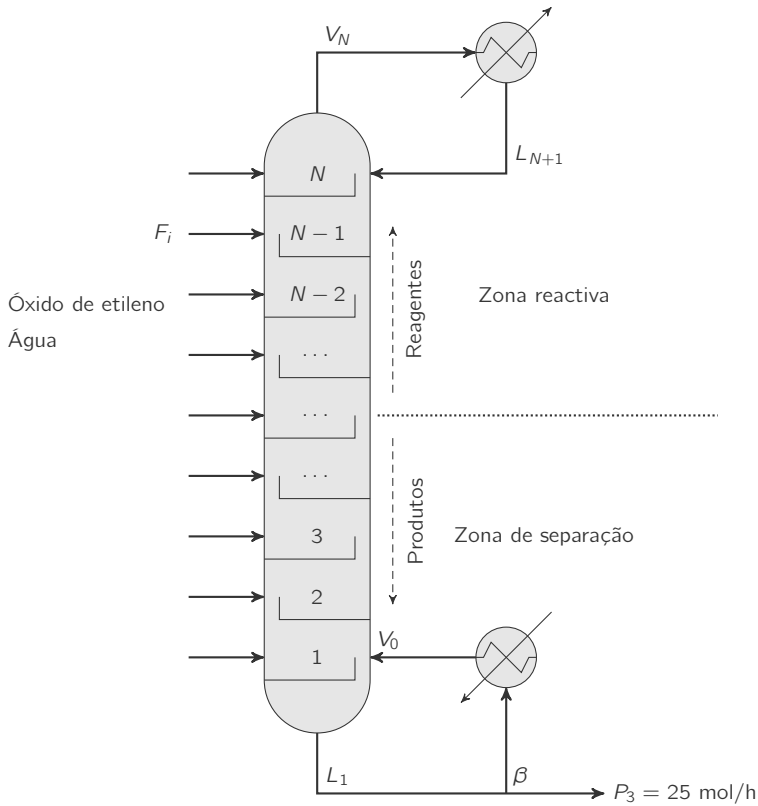


Figura 5.8 Problema P.CDR. Diagrama da coluna de destilação reactiva.

- Balanços materiais nos restantes andares

$$\begin{aligned}
 &F_{i,k} + V_{k-1} K_{i,k-1} x_{i,k-1} + L_{k+1} x_{i,k+1} + L_k x_{i,k} - \dots \\
 &- V_k K_{i,k} x_{i,k} + \sum_{j=1}^{N_R} \nu_{i,j} \xi_{j,k} = 0; \quad i = 1, \dots, N_C; \quad k = 2, \dots, N
 \end{aligned} \tag{5.62}$$

- Equações estequiométricas

$$\sum_{i=1}^C x_{i,k} - 1 = 0; \quad k = 1, \dots, N \tag{5.63}$$

$$\sum_{i=1}^C K_{i,k} x_{i,k} - 1 = 0; \quad k = 1, \dots, N \tag{5.64}$$

- Balanços energéticos de andar

$$\lambda (V_{k-1} - V_k) - \sum_{j=1}^{N_R} \Delta H_j \xi_{j,k} = 0; \quad k = 1, \dots, N \quad (5.65)$$

- Balanços materiais no reebulidor

$$B_i = (1 - \beta) L_1 x_{i,1}; \quad i = 1, \dots, N_C \quad (5.66)$$

- Balanços globais por componente

$$x_{d,i} Dist + B_i = P_i; \quad i = 1, \dots, N_C \quad (5.67)$$

- Balanço ao condensador

$$Dist = V_N - L_{N+1} \quad (5.68)$$

onde o refluxo total implica que

$$x_{i,N+1} = x_{d,i}; \quad i = 1, \dots, N_C \quad (5.69)$$

com as fracções de destilado calculadas segundo o equilíbrio

$$x_{d,i} = K_{i,N} x_{i,N}; \quad i = 1, \dots, N_C \quad (5.70)$$

- Fracção da corrente líquida vaporizada no reebulidor

$$\beta = \frac{V_0}{L_1} \quad (5.71)$$

- Definição das extensões das reacções

$$\xi_{1,k} = W_k x_{1,k} x_{2,k} k_{0,1} \exp\left(\frac{E_a}{RT_k}\right); \quad k = 1, \dots, N \quad (5.72)$$

$$\xi_{2,k} = W_k x_{1,k} x_{3,k} k_{0,2} \exp\left(\frac{E_a}{RT_k}\right); \quad k = 1, \dots, N \quad (5.73)$$

- Definição das constantes de equilíbrio

Equilíbrio ideal

$$K_{i,k} = a_i \exp\left[b_i \frac{(T_k - t_{a,i})}{(T_k - t_{b,i})}\right]; \quad i = 1, \dots, N_C, \quad k = 1, \dots, N \quad (5.74)$$

Equilíbrio não ideal, com $\gamma_{i,k} \neq 1$

$$K_{i,k} - \frac{P_{i,k}^{sat}}{P} \gamma_{i,k} = 0; \quad i = 1, \dots, N_C, \quad k = 1, \dots, N \quad (5.75)$$

com

$$\ln(P_{i,k}^{sat}) = A_{i,1} - \frac{A_{i,2}}{T_k + A_{i,3}}; \quad \text{com } P \text{ em Pa e } T \text{ em K} \quad (5.76)$$

onde, pela equação de Wilson, para $i = 1, \dots, N_C; k = 1, \dots, N$ e $p = 1, \dots, N_C$

$$\ln \gamma_{i,k} = 1 - \ln \left(\sum_{p=1}^{N_C} x_{p,k} \Lambda_{i,p,k} \right) - \sum_{m=1}^{N_C} \left[\frac{x_{m,k} \Lambda_{m,i,k}}{\sum_{p=1}^{N_C} x_{p,k} \Lambda_{m,p,k}} \right] \quad (5.77)$$

com

$$\Lambda_{i,p,k} = \frac{V_{M,p}}{V_{M,i}} \exp \left(-\frac{w_{i,p}}{RT_k} \right) \quad (5.78)$$

- Quantidade de calor transferida na operação de condensador e reebulidor

$$Q_B = \beta \lambda L_1 \quad (5.79)$$

$$Q_C = \beta_C V_N \quad (5.80)$$

- Dimensionamento da coluna

$$D^4 = C_D \beta^2 L_1^2 \quad (5.81)$$

$$H = H_0 + \sum_{k=1}^N H_k; \quad k = 1, \dots, N \quad (5.82)$$

$$H_k = H_{min} + 1.27 \frac{W_k}{D^2}; \quad k = 1, \dots, N \quad (5.83)$$

- Limites de operação para $k = 1, \dots, N$ e $i = 1, \dots, N_C$

$$L_k \leq L_{max} \quad (5.84)$$

$$V_k \leq V_{max} \quad (5.85)$$

$$W_k \leq W_{max} \quad (5.86)$$

$$F_{i,k} \leq F_{max} \quad (5.87)$$

O conjunto de Equações 5.61 a 5.86 corresponderá a uma formulação NLP se a sua avaliação for efectuada para uma determinada dimensão de coluna, enquanto que a liberdade na especificação de N se traduzirá numa concepção MINLP, onde o problema

Tabela 5.5 Problema P.CDR. Parâmetros do modelo (Ciric e Gu, 1994).

(a) Parâmetros Económicos.					(b) Dados Cinéticos e Termodinâmicos.			
Dado	Valor				Dado	Valor		
c_1	$43.7 \cdot 10^{-3}$ \$/mol				E_a/R	9547.0 K		
c_2	$21.9 \cdot 10^{-3}$ \$/mol				ΔH_1	$-80.0 \cdot 10^3$ J/mol		
c_{SH}	222.0 \$/ano				ΔH_2	$-13.1 \cdot 10^3$ J/mol		
c_T	15.7 \$/ano				$k_{0,1}$	$3.15 \cdot 10^5$ mol/m ³ /s		
c_R	$146.8 \cdot 10^{-6}$ \$s/J/ano				$k_{0,2}$	$6.3 \cdot 10^5$ mol/m ³ /s		
c_C	$24.5 \cdot 10^{-6}$ \$s/J/ano				λ	$20.0 \cdot 10^3$ J/mol		
c_0	10000.0 \$/ano							

(c) Especificações.		(d) Parâmetros para dimensionamento.	
Dado	Valor	Dado	Valor
$Dist$	0.0 mol/s	C_D	0.01331 m ² /mol ²
L_{max}	277.8 mol/s	H_0	3.0 m
V_{max}	277.8 mol/s	H_{min}	0.61 m
W_{max}	2.0 m ³		
P_3	6.9444 mol/s		

(e) Dados para ELV ideal.					(f) Constantes de Antoine.			
i	a_i	b_i	$t_{a,i}$	$t_{b,i}$	i	$A_{i,1}$	$A_{i,2}$	$A_{i,3}$
1	71.9	5.720	469	35.9	1	21.3066	2428.2	-35.3880
2	221.2	6.310	647	52.9	2	23.2256	3835.18	-45.3430
3	77.0	9.940	645	71.4	3	25.1431	6022.18	-28.2500
4	47.0	10.42	681	80.6	4	34.4262	16224.92	190.015

(g) Dados para ELV não ideal.					
i	Constantes Wilson (cal/mol)				V_M (mL/mol)
	1	2	3	4	
1	0.0	124.965	-79.4710	220.233	49.909
2	1905.77	0.0	1266.0109	-180.210	18.07
3	635.823	-1265.7398	0.0	856.192	55.92
4	157.874	923.330	-112.337	0.0	94.78

tomará a forma de optimização com vista a projecto. A definição completa do modelo só é conseguida com a definição dos seus parâmetros, apresentados na Tabela 5.5.

A adaptação desta formulação para os ambientes ASCEND e SIMOP pressupõe a correcta codificação dos modelos apropriados respectivamente a cada uma das metodologias, apresentando-se essas mesmas estruturas nos modelos incluídos no Apêndice A.8.

Além da sua natureza não linear, o presente caso de estudo combina reacção e separação, fenómenos que, nas soluções óptimas, podem ocorrer simultaneamente ou em zonas adjacentes da coluna. Adicionalmente, a unidade funciona em condições operatórias não isotérmicas a que se associam perfis de temperatura abruptos (Lima et al., 2006). Estas características estão normalmente associadas à presença de não convexidades que originam várias soluções locais. A possibilidade de especificar o número de andares de equilíbrio revela-se uma forma eficaz de controlar a dimensão deste problema de optimização, que atendendo às características enunciadas, se apresenta como um exemplo interessante para a avaliação das estratégias integradas propostas.

O estudo da presença de óptimos locais múltiplos foi efectuado com base na modelização do problema P.CDR em ambiente GAMS (apresenta-se o modelo associado no Apêndice A.8). Para tal, o método de ponto inicial múltiplo OQNLP (Lasdon et al., 2004) foi utilizado em conjunto com a CONOPT (Drud, 2008) para a resolução dos diferentes NLP para cada estimativa inicial.

As Figuras 5.9 e 5.10 mostram os resultados obtidos para a experiência de avaliação do caso de estudo. Utilizando as tolerâncias definidas por omissão, foram respectivamente encontradas um total de 158 e 357 soluções distintas para colunas com 10 e 20 andares de equilíbrio (ELV ideal). A forma monótona da função objectivo na região esquerda dos gráficos é maioritariamente provocada pelo facto do custo total dos reagentes estar incluído na avaliação da função objectivo. Esta contribuição domina sobre outras, tais como os custos energéticos, uma característica já identificada anteriormente para este caso de estudo por Cardoso et al. (2000).

Ainda do lado esquerdo de cada uma das figuras, os resultados encontrados evidenciaram variabilidade considerável nos perfis de composição sem que as dimensões da coluna variem significativamente. Denota-se assim importância de bons esquemas de inicialização bem como de metodologias para optimização global em exemplos de reacção com separação e, por isso, a potencial utilidade de utilizar métodos estocásticos.

A avaliação do caso de estudo proposto centra-se nas suas formulações NLP e MINLP, das quais se pretende construir simulações bem condicionadas numericamente. Em abordagens anteriores a este caso de estudo, Cardoso et al. (2000) e Lima et al. (2006) apresentam um conjunto de variáveis de decisão que garante a estabilidade dese-

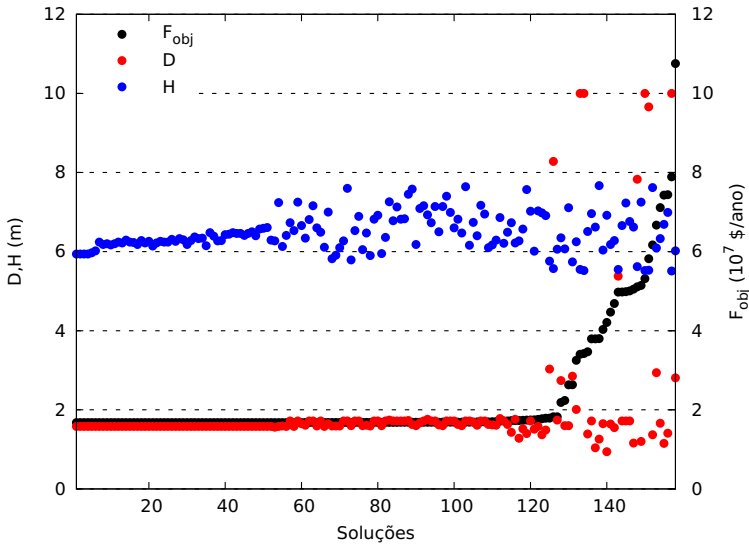


Figura 5.9 Problema P.CDR. Modelo GAMS, ELV ideal. Soluções locais identificadas com QNLP. Coluna com 10 andares de equilíbrio.

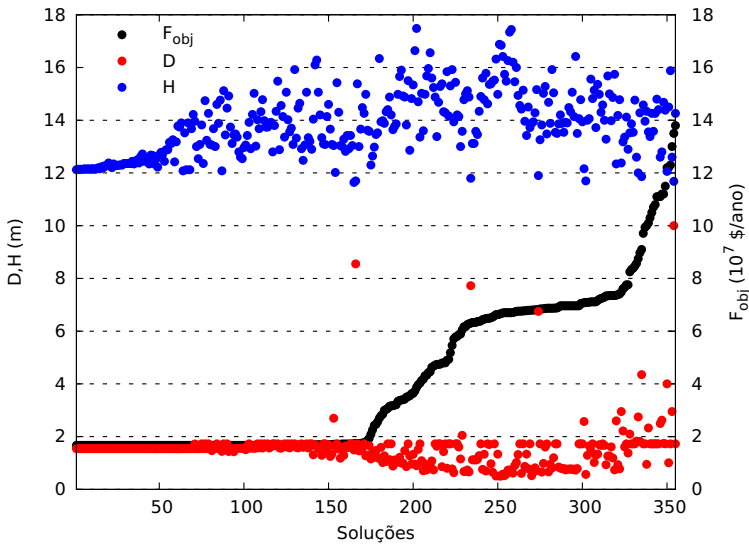


Figura 5.10 Problema P.CDR. Modelo GAMS, ELV ideal. Soluções locais identificadas com QNLP. Coluna com 20 andares de equilíbrio.

jada, nomeadamente $v_{P.CDR} = \{F_{1,k}, F_{2,j}, \beta, W_k\}$ com $(k = 1, \dots, N; j = 2, \dots, N)$, que representa o conjunto escolhido nas abordagens apresentadas nesta tese. Este conjunto decisão corresponde a $3N$ graus de liberdade para a formulação NLP. Na formulação MINLP, em que a variável discreta N também é parte integrante do conjunto, o número de graus de liberdade do problema corresponderá a um total de $3N + 1$.

Verificou-se que dimensões de coluna atípicas (em particular o valor do rácio entre o diâmetro e a altura) podem ser evitadas restringindo os caudais de líquido e vapor a um valor máximo de 1000 kmol/h (Ciric, 1995; Lima et al., 2006), restrições que foram consideradas na avaliação dos modelos pospostos no trabalho aqui apresentado.

Detalhes sobre a inicialização

Morton (2003) refere que a modelização de sistemas por equações pode exigir a implementação de procedimentos robustos de inicialização. Mesmo em situações que se considere ELV ideal, o carácter não linear destes problemas implica cuidado especial sobre a geração de estimativas iniciais, uma vez que a implementação robusta de métodos de Newton e quasi-Newton na solução de sistemas de equações simultâneas é condicionada pela existência de boas estimativas iniciais.

Tendo presente que a resolução dos sistemas é determinante na avaliação do modelo, estratégias de inicialização adequadas garantem uma taxa de sucesso superior na avaliação da função objectivo, condição que favorece a aplicação mais satisfatória de algoritmos de optimização estocásticos.

Cardoso et al. (2000) propõem um procedimento de inicialização do modelo apresentado, esquema que foi também implementado nos estudos da formulação NLP apresentada por Lima et al. (2006) utilizando o SIMOP. No âmbito deste trabalho, o procedimento de inicialização seguido é essencialmente baseado na proposta de Cardoso et al. (2000), o qual é descrito de seguida:

1. É obtida uma estimativa conservativa para $F_{2,1}$ (que não é parte integrante do conjunto decisão) considerando que o caudal de água representa, por um excesso de 50%, a produção desejada de etilenoglicol. Assim, considerando ambos os reagentes e a estequiometria da reacção, é obtida uma primeira estimativa normalizada para a composição do primeiro andar;
2. O valor inicial para as extensões das reacções é calculado em cada prato, considerando uma distribuição igualitária da produção desejada nos andares reactivos. Nos outros andares, onde não existe retenção de líquido, o valor das mesmas é considerado nulo;

3. Perfil de composições:

- a) A composição do último andar é assumida como sendo 95% de água e 5% de etilenoglicol. A partir deste ponto a fracção molar de óxido de etileno é calculada considerando a ocorrência de reacção. A composição do andar é finalmente ajustada por normalização;
- b) O perfil de composições é então calculado assumindo uma fracção nula de óxido de etileno em todos os pratos. Para os outros componentes é efectuada uma aproximação por interpolação linear e, para os andares reactivos, a fracção de óxido de etileno é ajustada pelo mesmo procedimento utilizado no último prato da coluna;

4. A computação do perfil de temperaturas inicial é efectuada considerando ELV ideal em todos os andares e através da equação que garante a consistência estequiométrica da fase de vapor;

5. A partir daqui as diferentes abordagens seguem alternativas diferentes:

- a) Nas simulações para SIMOP, este procedimento providencia estimativas iniciais para as composições de líquido utilizadas no modelo da coluna escrito em FORTRAN (Lima et al., 2006). Na implementação do método de Newton-Raphson, os valores de temperatura são usados na forma normalizada.
- b) Quando se utiliza o ASCEND, as capacidades de amortecimento de escala das variáveis do modelo foram activadas (como operação equivalente a normalização das variáveis) e, uma vez que o conjunto de saída obtido após partição determina a resolução simultânea das equações nos sistemas identificados, duas atribuições adicionais foram incorporadas no procedimento de inicialização.

- i. Admitiu-se um perfil uniforme em toda a coluna para os caudais de líquido e vapor, onde:

$$L_k = L_1 \text{ com } L_1 = \frac{1}{N_C \beta} \sum_{i=1}^{N_C} \frac{P_i}{x_{i,1}}; \quad k = 1, \dots, N + 1 \quad (5.88)$$

$$V_k = V_0 \text{ com } V_0 = \beta L_1; \quad k = 0, \dots, N \quad (5.89)$$

sendo L_1 e V_0 calculados pelas Equações 5.88 e 5.89, onde P_i corresponde à produção do componente i e $x_{i,1}$ é a fracção do componente i no primeiro andar da coluna;

- ii. Uma primeira estimativa para as constantes do ELV em cada prato foi calculada com base no perfil inicial de temperaturas.

5.3.2 Resultados de Optimização

Formulação NLP Ideal

Considerando o modelo escrito para ASCEND, a formulação NLP apresenta $14N + 23$ equações, sendo estas particionadas em $12 + N$ blocos após análise estrutural do algoritmo de decomposição do simulador. O maior bloco aloca um sistema de $13N + 13$ equações que são simultaneamente resolvidas enquanto as restantes $11 + N$ equações são distribuídas por blocos singulares que são trabalhados sequencialmente numa fase posterior.

A estratégia conjugada foi aplicada por intermédio da avaliação de dez corridas de optimização. Foram testadas colunas com dimensão igual a 3, 7, 10 e 15 andares. Os melhores resultados encontrados para o valor da função objectivo estão apresentados na Tabela 5.6 e Figura 5.11. Conjuntamente com os resultados para o ASCEND, são também apresentados resultados obtidos com outras abordagens a este caso de estudo. Consideram-se assim a aplicação da metodologia SIMOP (Lima et al., 2006) e a resolução do modelo NLP da coluna de destilação no ambiente GAMS (Brooke et al., 1998, ver Apêndice A.8) com recurso ao *solver* CONOPT (Drud, 2008).

Os resultados obtidos com a estratégia conjugada são considerados equivalentes até uma dimensão de coluna de dez andares. Na abordagem com ASCEND, o menor custo anual foi registado para uma coluna com sete andares teóricos, nomeadamente $1.534 \cdot 10^7$ \$/ano, e que corresponde a uma unidade com as especificações apresentadas na Tabela 5.7.

Tabela 5.6 Problema P.CDR. Caso NLP. ELV Ideal. Melhores resultados obtidos em função de N com a estratégia conjugada utilizando ASCEND e ainda com o GAMS (F_{obj} , 10^7 \$/ano).

N	Método		MSGa		MSIMPsa		CONOPT
	ASCEND		SIMOP ¹	ASCEND	SIMOP ¹	GAMS	
3	1.655	-	1.661	1.685	-	1.662	1.593
7	1.534	-	1.552	1.546	-	1.550	1.523
10	1.549	-	1.550	1.535	-	1.548	1.516
15	2.625	1.689 ²	1.567	8.069	1.818 ²	1.574	1.512
20	-	-	-	-	-	-	1.511

¹Lima et al. (2006)

²Resultados obtidos usando o esquema de inicialização com normalização em $F_{i,k}$ e W_k .

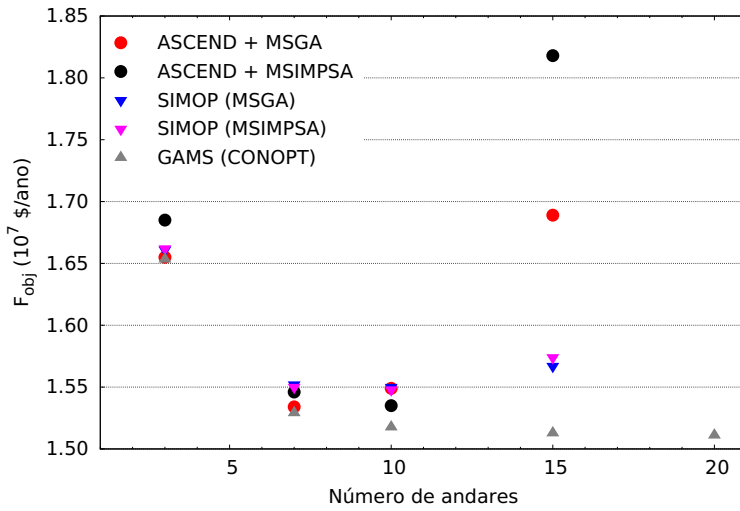


Figura 5.11 Problema P.CDR. Caso NLP. ELV Ideal. Comparação dos melhores valores da função objectivo utilizando ASCEND, SIMOP e GAMS.

Tabela 5.7 Resultados ASCEND. Problema P.CDR. Caso NLP. ELV Ideal. Especificações da coluna de custo mínimo, cujo projecto corresponde a uma coluna com sete andares e uma $F_{obj} = 1.534 \cdot 10^7$ \$/ano.

Andar	F_{1k} (mol/s)	F_{2k} (mol/s)	W_k m^3	T_k (K)	V_k mol/s	L_k mol/s
8	-	-	-	-	-	277.6
7	2.334	1.259	0.799	373.3	277.6	273.0
6	1.770	4.308	0.553	373.4	272.2	274.3
5	2.060	0.499	1.013	373.7	269.0	271.0
4	1.294	0.000	0.702	373.9	265.2	269.2
3	0.000	0.051	1.213	379.6	263.1	269.2
2	0.001	1.011	0.527	432.2	263.1	270.2
1	0.001	0.141	1.545	467.7	263.1	270.4
0	-	-	-	-	263.1	-

$D = 1.68$ m, $H = 10.20$ m, $\beta = 0.973$
 $Q_B = 10.5$ MW, $Q_C = 11.1$ MW

As simulações avaliadas com o SIMOP (dez corridas de optimização) permitem dizer que é notório que os resultados obtidos simulando com o ASCEND acompanham a tendência existente entre o custo determinado e a dimensão da coluna (Lima et al., 2006). Apesar disso, os resultados da Tabela 5.6 mostram existir um evidente desemparelhamento nas avaliações efectuadas pela estratégia conjugada para uma dimensão de 15 andares, verificando-se tal situação quer com o MSGA ou com o MSIMPSPA.

A esta deterioração de resultados pode associar-se um maior esforço numérico na resolução dos sistemas de equações, o que condiciona a eficácia de avaliação da função objectivo. Esse efeito é ilustrado pela Figura 5.12, na qual se apresenta a medição de reprodutibilidade dos resultados com respeito às avaliações da formulação NLP com ELV ideal mediante a representação da distribuição cumulativa dos erros relativos das diferentes corridas de optimização realizadas.

A Figura 5.12 mostra que, no geral, uma distribuição de erros mais alargada está associada a uma coluna de maior dimensão, i.e., a um maior sistema de equações. Tal como será discutido posteriormente, a deterioração pode ser explicada pela expansão do número de equações que tem que ser resolvido simultaneamente quando dimensões de coluna superiores são avaliadas.

Uma vez que os resultados obtidos com a estratégia conjugada para colunas maiores não revelam a mesma robustez que os obtidos para colunas mais pequenas, modificações foram testadas no procedimento de inicialização utilizado nas simulações para ASCEND.

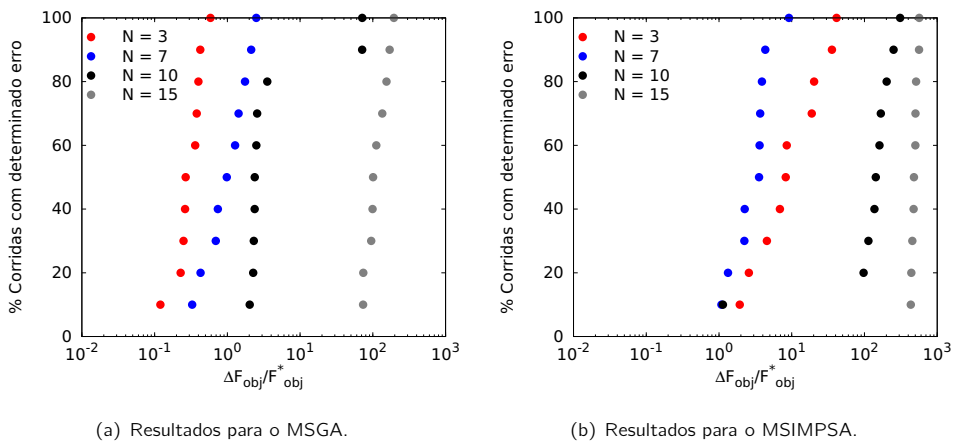


Figura 5.12 Caso NLP. ELV Ideal. Distribuição de erro relativa dos resultados óptimos obtidos com ASCEND associado a MSGA e MSIMPSPA para modelos com 3, 7, 10 e 15 andares.

As alterações introduzidas têm como objectivo estabelecer melhores estimativas iniciais para o sistema de equações a resolver ou, por ora, testar uma forma diferente de tratar as restrições. Nesse sentido foi avaliado o impacto das seguintes modificações:

- Introdução de variáveis de folga na definição das restrições superiores dos caudais de líquido e vapor na coluna. Os resultados obtidos foram equivalentes aos previamente verificados, pelo que esta variante de formulação não se mostrou vantajosa.
- Atendendo a que os perfis de composição obtidos nas soluções não são lineares (Cardoso et al., 2000), o procedimento de inicialização foi modificado no sentido de garantir uma estimativa de perfil de composição de melhor qualidade. Em vez de uma interpolação linear, a primeira estimativa para o perfil de composições foi calculada utilizando uma aproximação sigmóide. Esta assegura que a estimativa inicial para o perfil de temperaturas também segue uma forma sigmóide, estando mais próximo do perfil real obtido na solução. O procedimento de inicialização modificado foi testado para modelos NLP com dez e quinze andares, mas os resultados obtidos não foram diferentes dos anteriormente registados.
- Para contornar as possíveis dificuldades do método de Newton (*solver* QRSlv) quando lida com limites nos caudais de líquido e vapor, os mesmos foram relaxados, mas não foram registadas melhorias nos resultados obtidos.
- Para além do procedimento de inicialização estabelecido como padrão nas diferentes abordagens a este exemplo (Cardoso et al., 2000; Lima et al., 2006), condições adicionais são introduzidas implementando duas modificações sobre o conjunto de variáveis de decisão gerado aleatoriamente pelos optimizadores estocásticos, nomeadamente:
 - Normalização dos caudais de alimentação para um intervalo controlado de excesso estequimétrico, nomeadamente $[0, 30\%]$;
 - Normalização da quantidade total do volume reaccional existente em todos os pratos da coluna para o intervalo $[2, 4] \text{ m}^3$.

Ambas as condições são resultado da análise dos resultados obtidos com o GAMS. As mesmas foram testadas com ambos optimizadores utilizando modelos de coluna com quinze andares e os resultados obtidos, também apresentados na Figura 5.6, mostram uma melhoria significativa quando comparados com os obtidos previamente.

Os melhores resultados para a formulação NLP do caso de estudo foram obtidos utilizando o GAMS, tendo este modelo sido escrito após uma reformulação cuidada do modelo original de equações (ver Apêndice A.8), considerando os aspectos de escala e não linearidade.

Atendendo às soluções previamente determinadas, o procedimento de inicialização para o modelo em GAMS atribui um perfil inicial sigmóide como estimativa para as temperaturas em cada andar. Tal comportamento foi propagado ao longo da coluna considerando as composições de vapor, extensões das reacções e caudais. Este garantiu a plausibilidade necessária para a procura de optimalidade local com recurso aos *solvers* CONOPT e IPOPT. Contudo, não foram registadas soluções plausíveis pela implementação de outros algoritmos, casos do MINOS e do SNOPT, que também revelaram dificuldades adicionais em colunas de dimensão superior.

Uma vez que o caso de estudo apresentado não pretende ser um instrumento de *benchmarking* para *solvers* NLP, não se optou por experimentar esquemas de inicialização mais sofisticados, casos de métodos como o Wang-Henke (Seader e Henley, 1998). Não se considerou igualmente necessário efectuar comparações com a aplicação de *solvers* dedicados a optimização global, como é o caso do BARON (Tawarmalani e Sahinidis, 2004).

Embora melhores resultados tenham sido encontrados utilizando o GAMS, refere-se que esquemas de inicialização mais simples foram utilizados com os algoritmos estocásticos para encontrar pontos plausíveis nas metodologias ASCEND e SIMOP.

Os resultados obtidos com GAMS indicam que o valor óptimo, apesar do perfil monótono da função objectivo, estará localizado perto dos 30 andares (ver Tabela 5.8). As colunas obtidas com o GAMS ou com os optimizadores estocásticos são essencialmente as mesmas no que diz respeito ao diâmetro, energia despendida e valor da razão de refluxo no reebulidor. Contudo, uma vez que as alimentações da coluna têm a mais pesada influência no cálculo da função objectivo, as diferenças encontradas e apresentadas na Tabela 5.6 são consequência das diferentes distribuições de caudais a que se chega na solução. Nas simulações operadas com o GAMS, as soluções reflectem uma clara divisão da coluna em duas zonas diferentes, nomeadamente a zona reactiva, junto ao topo, e a zona de separação, junto à base, com os caudais a serem distribuídos apenas na zona reactiva (Figura 5.9).

Este comportamento conduz normalmente a uma diminuição do caudal total alimentado à coluna, o que garante condições para que a unidade opere com um menor nível de volume reaccional acumulado nos pratos, que são nulos na zona de separação. O estabelecimento de soluções com uma alimentação menor também contribui para que a

Tabela 5.8 Problema P.CDR. Caso NLP. Melhores resultados encontrados para a abordagem com GAMS (F_{obj} , 10^7 \$/ano).

N	3	7	10	20	30	60
F_{obj}	1.594	1.523	1.516	1.511	1.510	1.513

Tabela 5.9 Resultados GAMS. Problema P.CDR. Caso NLP. ELV Ideal. Especificações da coluna de custo mínimo encontrada para o modelo com sete andares, à qual corresponde uma $F_{obj} = 1.523 \cdot 10^7$ \$/ano.

Andar	F_{1k} (mol/s)	F_{2k} (mol/s)	W_k m^3	T_k (K)	V_k mol/s	L_k mol/s
8	-	-	-	-	-	277.8
7	-	5.676	0.072	372.9	277.8	277.8
6	3.255	1.521	0.134	373.2	274.0	278.0
5	1.937	-	0.407	373.6	270.3	273.4
4	2.231	-	0.501	373.8	266.7	270.5
3	-	-	-	374.9	263.3	270.5
2	-	-	-	404.6	263.3	270.5
1	-	-	-	460.0	263.3	270.5
0	-	-	-	-	263.3	-

$D = 1.68$ m, $H = 7.82$ m, $\beta = 0.973$
 $Q_B = 10.5$ MW, $Q_C = 11.1$ MW

unidade tenha menor altura, permitindo assim obter colunas mais baratas com a mesma carga de alimentação. Esta configuração de coluna está de acordo com a previamente apresentada por Ciric e Gu (1994), que também utilizam o GAMS para este problema. Curiosamente, estes autores apresentam um valor óptimo de dez andares quando efetuada uma procura no intervalo limitado até 20 andares teóricos.

Ainda no contexto da análise da formulação NLP, deve ser referido que nem todas as soluções são completamente intercambiáveis entre diferentes esquemas de simulação. Por exemplo, a solução obtida com o GAMS para 7 andares não é plausível com o ASCEND ou SIMOP, a não ser que a tolerância para resolver o sistema de equações seja relaxada para 10^{-2} . Isto acontece devido às diferentes estratégias utilizadas pelos *solvers*, embora, no âmbito do presente trabalho, esta possa ser considerada uma diferença que não constitui obstáculo à análise dos resultados aqui apresentados.

Formulação NLP não ideal

Okasinski e Doherty (1998) propõem o estudo do funcionamento da unidade em condições operatórias diferentes, aumentando a pressão de uma para quinze atmosferas e modificando a representação do modelo de equilíbrio líquido vapor para uma descrição não ideal caracterizada pelas equações de Wilson. Este novo conjunto de definições foi testado nas abordagens prévias realizadas por Cardoso et al. (2000) e (Lima et al., 2006), pelo que a existência de dados para comparação motivou a avaliação deste novo modelo através do acoplamento MSGA/MSIMPISA com ASCEND.

Quando o modelo não ideal é avaliado por intermédio da estratégia conjugada proposta, verifica-se a obtenção de piores resultados do que os registados com abordagens alternativas apresentadas na Tabela 5.10. Relativamente à determinação do custo de operação anual mínimo, os resultados obtidos mostram-se bastante díspares do nível de desempenho observado nos modelos que recorrem à definição ideal para descrever o equilíbrio (dispensando-se por isso a representação das distribuições de erro relativo para este caso). Salienta-se aqui que, apesar da mudança de condições que caracterizam o comportamento e modo de operação da coluna, os resultados obtidos com o SIMOP para o modelo de dez andares não apresentam deterioração.

No sentido de procurar resultados mais favoráveis, o esquema de inicialização melhorado foi aplicado às simulações ASCEND para modelos com dez andares. À semelhança do comportamento verificado nas simulações para equilíbrio ideal, os novos valores encontrados para a função objectivo são significativamente melhores do que os resultantes da aplicação do procedimento de inicialização sem normalização dos caudais de alimentação, conforme apresentado na Tabela 5.10

Tabela 5.10 Caso NLP. ELV não ideal. Melhores resultados obtidos (F_{obj} , 10^7 \$ano).

N	Método	MSGA		MSIMPISA		
		ASCEND	SIMOP ¹	ASCEND	SIMOP ¹	
3	2.357	-	-	3.548	-	-
7	2.133	-	-	3.579	-	-
10	2.501	1.865 ²	1.579	4.513	1.982 ²	1.586
15	3.928	-	-	9.066	-	-

¹Lima et al. (2006)

²Resultados obtidos usando o esquema de inicialização com normalização em $F_{i,k}$ e W_k .

Formulação MINLP

Os optimizadores estocásticos MSGA e MSIMPISA possibilitam a inclusão de variáveis discretas no conjunto de variáveis de decisão, capacidade essa, aliás, que motivou a adaptação da metodologia SIMOP à análise desse tipo de problemas como descrito na Secção 5.2. Nesse contexto, a estratégia conjugada foi também aplicada à formulação MINLP do caso de estudo proposto, permitindo observar o comportamento das estratégias integradas estudadas nesta tese na aplicação a formulações MINLP.

A introdução da variável discreta N no conjunto de variáveis condiciona a estrutura dos modelos criados para ASCEND e SIMOP. A solução passa por uma de duas opções: (1) a codificação de um modelo indexado à dimensão da coluna e que se possa considerar elástico na passagem entre sucessivas avaliação da função objectivo ou (2) a configuração de uma leque de problemas NLP pré-compilados, para várias dimensões de coluna, que possam depois ser avaliados em função do valor adoptado por N . Esta última opção assemelha-se, em certa medida, à filosofia de base seguida pela métodos determinísticos para optimização MINLP.

Apesar disso, tal não corresponde à resolução de diferentes sub-problemas NLP. Nos modelos SIMOP, a dimensão da coluna não é a única variável discreta do problema pelo que os modelos de coluna disponibilizados não correspondem ao conjunto de todas as configurações discretas possíveis. A colecção de modelos criada representa a solução encontrada para modelizar a variabilidade de dimensão da unidade em análise neste caso de estudo, pois permite que os optimizadores explorem o espaço combinatorial adequadamente.

Estabelecida a forma de modelização do problema MINLP, apresentam-se algumas das considerações e adaptações operadas sobre os intervenientes nas estratégias integradas de simulação com vista a uma melhor abordagem ao problema, adequada exploração do espaço de procura e aproveitamento de estruturas previamente codificadas na resolução dos casos NLP. Os ajustes efectuados ao nível do modelo, da estratégia conjugada e do SIMOP foram os seguintes:

- **Modelo** O MSGA e o MSIMPISA comportam-se de maneira diferente na exploração do espaço de procura definido para as variáveis discretas. O MSGA explora o domínio de procura, livremente, permitindo que, entre duas corridas de optimização consecutivas, as variáveis discretas sejam alteradas com uma amplitude superior a uma unidade. No caso do MSIMPISA este comportamento é diferente. O algoritmo modifica as variáveis discretas em apenas uma unidade (positiva ou negativa) entre procuras estocásticas que aconteçam sequencialmente.

O comportamento evidenciado pelo MSIMPISA prejudica a procura em domínios discretos alargados, onde a rigidez de variabilidade da componente discreta torna o processo de procura muito dependente da estimativa inicial para a configuração discreta. Nesse sentido, e com o objectivo de estabelecer uma variabilidade mais adequada à exploração do intervalo especificado para a variável N , a determinação da mesma é efectuada por intermédio da sua representação por expansão binária (Cardoso et al., 1996; Floudas, 1995). O procedimento define o cálculo da variável discreta genérica $y_{bin} \in \{\gamma_{bin}, \varepsilon_{bin}\}$, que é função do somatório de variáveis binárias Y_z , como definido na Equação 5.90,

$$y_{bin} = \gamma_{bin} + \sum_{z=1}^{n_{exp}} (z Y_z) + (\varepsilon_{bin} - C_z) Y_{z+1} \quad (5.90)$$

A dimensão do somatório, n_{exp} , é determinada com base no limite inferior do domínio de y_{bin} respeitando a condição $C_z \leq \varepsilon_{bin}$. Por seu turno, a constante C_z é dada pela Equação 5.91, que relaciona todos esses parâmetros,

$$C_z = \gamma_{bin} + \sum_{z=1}^{n_{exp}} z \quad (5.91)$$

Na abordagem efectuada à formulação MINLP do caso em estudo, foram considerados os intervalos de exploração $N = [3, 10]$, $N = [3, 15]$ e $N = [3, 20]$ nos diferentes optimizações efectuadas. A tais intervalos corresponderão respectivamente as expansões binárias determinadas pelas Equações 5.93–5.95, relativas aos coeficientes apresentados em 5.92 e calculados usando a Equação 5.91:

$$\gamma_{bin} = 3 \Rightarrow C_1 = 4, C_2 = 6, C_3 = 9, C_4 = 13, C_5 = 18, C_6 = 24, \dots \quad (5.92)$$

$$\begin{aligned} N \leq 10, \quad n_{exp} = 3, \\ N = 3 + Y_1 + 2Y_2 + 3Y_3 + Y_4 \end{aligned} \quad (5.93)$$

$$\begin{aligned} N \leq 15, \quad n_{exp} = 4, \\ N = 3 + Y_1 + 2Y_2 + 3Y_3 + 4Y_4 + 2Y_5 \end{aligned} \quad (5.94)$$

$$\begin{aligned} N \leq 20, \quad n_{exp} = 5, \\ N = 3 + Y_1 + 2Y_2 + 3Y_3 + 4Y_4 + 5Y_5 + 2Y_6 \end{aligned} \quad (5.95)$$

- **Estratégia conjugada** A primeira opção para implementação do modelo MINLP para avaliação pela estratégia conjugada com recurso ao ASCEND partiu da utilização de um modelo genérico, indexado à variável N , e cuja dimensão seria estabelecida de cada vez que os optimizadores especificassem um conjunto de variáveis de decisão.

A escolha de um modelo deste tipo implicaria que, na reincidência de solicitação de cálculo da função objectivo, o modelo ASCEND tivesse que compilar a simulação partindo do modelo a cada chamada do optimizador, acrescentando despesa considerável ao nível do tempo computacional. Atendendo a que a estratégia, pela sua natureza, já apresenta um desempenho menos interessante a este nível quando comparada com as simulações codificadas em FORTRAN, decidiu-se abandonar a opção de recurso a este tipo de modelo.

Em alternativa, optou-se por criar um conjunto de simulações codificadas para cada uma das dimensões de coluna no domínio do problema. Para esse efeito modificou-se o *script* de ligação como apresentado no Quadro 5.8, no qual a simulação do problema é escolhida por intermédio de uma selecção caso a caso das diferentes simulações compiladas. Desta forma, o modelo codificado para o

Quadro 5.8 Problema P.CDR. Caso MINLP. Forma generalizada das modificações operadas no procedimento RUNSIM.

```

proc RUNSIM {} {
  set nt [expr 3 + $y1 + 2*$y2 + 3*$y3 + 4*$y4 + 2*$y5]

  case $nt {
    3 {set ascSimsVect(selectedsim) n03}
    4 {set ascSimsVect(selectedsim) n04}
    5 {set ascSimsVect(selectedsim) n05}
    6 {set ascSimsVect(selectedsim) n06}
    7 {set ascSimsVect(selectedsim) n07}
    8 {set ascSimsVect(selectedsim) n08}
    9 {set ascSimsVect(selectedsim) n09}
    10 {set ascSimsVect(selectedsim) n10}
    11 {set ascSimsVect(selectedsim) n11}
    12 {set ascSimsVect(selectedsim) n12}
    13 {set ascSimsVect(selectedsim) n13}
    14 {set ascSimsVect(selectedsim) n14}
    15 {set ascSimsVect(selectedsim) n15}
  }

  set n_trays $nt

  (...) inicialização (...)

  RUN $ascSimsVect(selectedsim).default_self

```


Quadro 5.9 Problema P.CDR. Caso MINLP. Modificações operadas no modelo ASCEND e no `opt.script` correspondente.

```

---- MODIFICAÇÃO DA DECLARAÇÃO DO MODELO ----

MODEL column_model(n_trays WILL_BE integer_constant);

---- INCLUSÃO DA DEFINIÇÃO RECURSIVA DE MODELOS (ex. N=7) ----

(*== MODELS FOR DIFFERENT DIMENSIONS ==*)

MODEL n07;
n_trays IS_A integer_constant;
n_trays := 7;
column IS_A column_model(n_trays);
METHODS
METHOD setcolumn;
  RUN column.default_self;
  RUN column.specify;
  RUN column.values;
  RUN column.bound_self;
  RUN column.scale_self;
END setcolumn;
METHOD default_self;
RUN setcolumn;
END default_self;
END n07;

```

caso NLP é aproveitado como estrutura base para a criação dos novos modelos. O ASCEND possibilita este tipo de recorrência segundo a estrutura apresentada no Quadro 5.9, promovendo assim a reutilização de código já existente para a simulação da unidade processual.

- **SIMOP** A modelização da formulação MINLP para o SIMOP está sujeita a condicionantes diferentes das já enunciadas. No seu estágio de desenvolvimento actual, o SIMOP não permite a escrita de modelos indexados, ou seja, todas as variáveis associadas ao modelo são declaradas individualmente e não como vectores. Desde o início decidiu-se não alterar este comportamento por parte da metodologia, uma vez que se considera que há ainda outros aspectos mais importantes que devem ser amadurecidos na experimentação desta ferramenta.

Logo, face a esta condicionante, a hipótese de escrever um modelo elástico de dimensão variável indexada a N está excluída, pelo que a representação MINLP deve ser sujeita a considerações alternativas.

Apesar de aumentar o grau combinatorial do problema, a primeira tentativa de

abordagem para representação do modelo MINLP para SIMOP tem como base a inclusão de variáveis binárias para além da dimensão da coluna, N . A introdução dessas variáveis prende-se com a especificação da existência ou não existência de elementos ou fenómenos presentes na unidade. A primeira representação considera dois conjuntos de variáveis binárias, $Y_{ANDAR,k}$ e $Y_{F,k}$, que determinam respectivamente as existências do andar k e de alimentação nesse mesmo andar.

A inclusão das variáveis $Y_{F,k}$ determina que nas Equações 5.85 a 5.86 sejam substituídas todas as ocorrências das variáveis $F_{i,1}$ e $F_{i,2}$ pela combinação $Y_{F,k} \cdot F_{i,1}$ e $Y_{F,k} \cdot F_{i,2}$ respectivamente. Uma vez que o modelo não está indexado, a sua rigidez de codificação leva à concepção de uma simulação, para $N = N_{max}$, com andares intercalares de comportamento inerte quando à cinética e ao equilíbrio de líquido e vapor. Isto obriga a que matematicamente esse pratos sirvam apenas para garantir a continuidade das correntes de líquido e vapor, que deverão manter os seus caudais e composições. Esta concepção obriga a que o modelo matemático tenha que garantir o cálculo de uma constante de equilíbrio imaginária, K^* , baseada unicamente na premissa que garanta composições constantes à entrada e saída do andar.

A definição desta constante imaginária pode ser escrita em função do número de andares, e para todos os componentes, da forma apresentada pelas Equações 5.96 e 5.97:

– Para $k = 1 \vee k = N$

$$K_{i,k}^* = f(T_k) \quad (5.96)$$

– Para $k = 2 \dots, N - 1$

$$K_{i,k}^* = Y_{ANDAR,k} \left[f(T_k) - \frac{K_{i,k-1} X_{i,k-1}}{X_{i,k+1}} \right] + \frac{K_{i,k-1} X_{i,k-1}}{X_{i,k+1}} \quad (5.97)$$

A introdução das variáveis $Y_{ANDAR,k}$ condicionou também a modificação da equação estequiométrica da fase de vapor em cada andar (Equação 5.64), que para flexibilizar a matriz de incidências a dar ao algoritmo de decomposição, passa a ser definida explicitamente em função das composições de vapor,

$$\sum_{i=1}^C y_{i,k} - 1 = 0, \quad k = 1, \dots, N \quad (5.98)$$

também calculadas através da lei de equilíbrio diferenciada para todos os componentes nos andares intermédios que não existem, isto é, para aqueles que $Y_{ANDAR,k} = 0$ (Equações 5.99 e 5.100).

– Para $k = 1 \vee k = N$

$$y_{i,k} = K_{i,k} x_{i,k} \quad (5.99)$$

– Para $k = 2 \dots, N - 1$

$$y_{i,k} = Y_{ANDAR,k} (K_{i,k} x_{i,k} - y_{i,k-1}) - y_{i,k-1} \quad (5.100)$$

O recurso a este tipo de modelo utilizando o SIMOP não permitiu a obtenção de perfis de temperatura e composição suficientemente estáveis à convergência da rotina NRES sobre o sistema de equações, pelo que a frequente incidência de resultados penalizados não permitiu encontrar resultados satisfatórios nas corridas de optimização estocásticas realizadas.

Optou-se assim por implementar um esquema de avaliações semelhante ao que se utilizou com o ASCEND na estratégia conjugada, situação que, para além de ser compatível com as estruturas codificadas pelo SIMOP, permite que as estratégias integradas experimentadas registem resultados com base em filosofias de simulação semelhantes. Assim, e com base no SIMOP, foram criadas diferentes rotinas de avaliação de custo da coluna, em função da dimensão, que posteriormente foram compiladas conjuntamente. Todavia, decidiu-se incluir no conjunto de variáveis de decisão o grupo de variáveis binárias $Y_{F,k}$, com o intuito dos optimizadores poderem testar diferentes configurações de alimentação, criando directamente diversos cenários de ausência de alimentação em diversos andares. Tais configurações são estabelecidas com menor probabilidade quando os caudais em questão dependem apenas da definição de variáveis que sejam contínuas. Atendendo à presença da variável discreta N , o problema apresentará $4N + 1$ graus de liberdade quando resolvido com o MSGA. Para o MSIMPISA, uma vez que a definição de N recorre à expansão binária pela Equação 5.90, o problema terá $4N + Z + 1$ graus de liberdade. Resta ainda referir que, como o problema é MINLP e optimizadores efectuem a procura em todas as rotinas simultaneamente, o espaço de procura corresponderá sempre ao do modelo com maior dimensão, caso que determinará todas as configurações relacionadas com partilha de variáveis e estabelecimento de *common blocks*.

Convém referir que, apesar de os optimizadores permitirem efectuar testes nos extremos do domínio de procura, essa situação é na prática inviável atendendo à dimensão do conjunto de variáveis de decisão em análise para as dimensões de coluna estudadas.

Para a formulação MINLP, a aplicação da estratégia conjugada considerou modelos para definições de equilíbrio ideal e não ideal. Foram realizadas respectivamente 30 e 10 corridas de optimização para esses casos, explorando a variável N no intervalo [3, 15], segundo problemas com 46 graus de liberdade para o MSGA e 50 para o MSIMPSPA.

Os melhores resultados encontrados para o valor da função objectivo estão apresentados na Tabela 5.11. Os resultados do modelo MINLP obtidos por Cardoso et al. (2000) são incluídos apenas como referência uma vez que correspondem a uma estratégia de simulação diferente onde, ao contrário do que acontece com o ASCEND e o SIMOP, não há análise estrutural e partição do sistema de equações com base em procedimentos programados.

O projecto de coluna com menor custo anual encontrado com o ASCEND utilizando um modelo MINLP corresponde a seis pratos teóricos, um custo anual de $1.569 \cdot 10^7$ \$/ano e com especificações apresentadas na Tabela 5.12. A aplicação do procedimento de inicialização melhorado por recurso a normalização conduz novamente a melhorias significativas nos valores da função objectivo. Para além disso, as colunas associadas a estes resultados mais favoráveis estão mais próximas da gama considerada habitual no que diz respeito ao rácio altura/diâmetro (Douglas, 1988).

A avaliação da formulação MINLP com o SIMOP também se debruçou sobre as duas situações distintas para a descrição ELV. No que diz respeito à exploração do domínio das variáveis de decisão, é necessário ter em consideração que, incluindo variáveis binárias que determinam a existência ou não existência de alimentação em cada andar, existe a possibilidade de toda a alimentação se concentrar apenas num único prato. Logo, para cada um dos caudais que integram o conjunto de variáveis de decisão, é necessário considerar um intervalo de procura que, no mínimo, corresponda à quantidade

Tabela 5.11 Resultados ASCEND. Caso MINLP. Melhores resultados obtidos ($F_{obj} \cdot 10^7$ \$/ano) para modelos Ideal e não ideal pela aplicação ASCEND conjugada com MSGA e MSIMPSPA. Os resultados obtidos por Cardoso et al. (2000) são apresentados como referência comparativa. Os respectivos números de andar por coluna aparecem entre parêntesis.

ELV	MSGA		MSIMPSPA		C-MINLP ¹
Ideal	1.753 (5)	1.583 ² (11)	1.569 (6)	1.587 ² (13)	1.526 (10)
Wilson	3.162 (5)	1.871 ² (13)	3.317 (8)	1.623 ² (12)	1.512 (11) ³

¹Cardoso et al. (2000).

²Resultados obtidos usando o esquema de inicialização com normalização em $F_{i,k}$ e W_k .

³Restrições sobre L_k e V_k relaxadas.

Tabela 5.12 Resultados ASCEND. Problema P.CDR. Caso MINLP. ELV Ideal. Especificações da coluna de custo mínimo, cujo projecto corresponde a uma coluna com seis andares e uma $F_{obj} = 1.569 \cdot 10^7$ \$/ano.

Andar	F_{1k} (mol/s)	F_{2k} (mol/s)	W_k m^3	T_k (K)	V_k mol/s	L_k mol/s
7	-	-	-	-	-	274.8
6	2.556	0.570	0.553	373.3	274.8	269.1
5	1.744	0.769	1.013	373.5	267.0	267.0
4	1.926	0.284	0.702	373.7	265.9	263.7
3	1.317	1.236	1.213	374.3	262.3	263.1
2	0.000	3.571	0.527	393.6	260.2	266.6
1	0.001	1.172	1.545	454.2	260.2	267.8
0	-	-	-	-	260.2	-

$D = 1.67$ m, $H = 9.34$ m, $\beta = 0.972$

$Q_B = 10.4$ MW, $Q_C = 11.0$ MW

estequiométrica mínima necessária para garantir a produção desejada.

A avaliação da formulação MINLP com o SIMOP também se debruçou sobre as duas situações apresentadas para descrição do ELV. Uma vez que este caso de estudo pretende servir de teste a esta nova funcionalidade da metodologia, as corridas efectuadas foram conduzidas de uma forma mais exaustiva do que em situações anteriores (trinta corridas de optimização para ambas as definições de equilíbrio), e seguindo as seguintes directrizes:

- **Graus de liberdade** No caso do ELV ideal, foram exploradas dimensões de coluna até 20 andares, traduzindo-se em 81 graus de liberdade para os problemas resolvidos com o MSGA e em 86 para o caso do MSIMPISA.

Na análise do problema definido segundo ELV não ideal, o algoritmo de partição e ruptura revelou dificuldades na obtenção de conjuntos de saída análogos para dimensões de coluna superiores a dez andares. Tal situação condicionou a procura estocástica ao intervalo $N = [3, 10]$, conduzindo a simulações para MSGA com 41 graus de liberdade, que pela expansão binária aumentam para 44 quando usado o MSIMPISA.

- **Inicialização** Atendendo à dimensão dos conjuntos decisão e partindo do conhecimento dos resultados obtidos com as abordagens anteriores, decidiu-se que o

método de inicialização das colunas utilizaria normalização prévia dos caudais de entrada na coluna. Ao contrário da inicialização melhorada por normalização para os problemas resolvidos com ASCEND, nesta situação particular não são normalizados os volumes reaccionais integrantes do conjunto de variáveis de decisão.

Na avaliação do caso MINLP com o SIMOP, foram realizadas corridas numa gama de procura que abrange cinco situações diferentes no que diz respeito ao limite superior estabelecido para o excesso estequiométrico admissível na alimentação total. Consideraram-se os casos para 10, 20, 30, 40 ou 50%.

Pretende-se com esta condição melhorar a qualidade da procura estocástica, promovendo uma sinergia positiva entre a normalização dos caudais de entrada e a utilização de variáveis binárias que condicionam a sua existência.

Quanto às variáveis binárias, com a sua introdução para a modelização da ocorrência, ou não, de alimentação em cada andar, é necessário contemplar a situação em que toda a alimentação é efectuada num único andar. Nesse sentido, é necessário definir limites de procura suficientemente largos nas variáveis caudal de alimentação. Tal situação promove a penalização de diversos conjuntos de variáveis de decisão que correspondem a um nível de alimentação excessivo. Desse modo, pela normalização aleatória da alimentação total da coluna em intervalos pré-estabelecidos, consegue-se uma maior taxa de simulações satisfatórias.

Para colunas a funcionar segundo ELV ideal, os melhores valores encontrados para as avaliações efectuadas encontram-se na Tabela 5.13. O valor mínimo para o custo, $1.502 \cdot 10^7$ \$/ano, corresponde ao projecto de uma coluna de oito andares, determinada com o MSIMPISA, numa avaliação com intervalo limitado a 30% de excesso estequiométrico. Este resultado é ligeiramente melhor que o determinado por Cardoso et al. (2000), $1.526 \cdot 10^7$ \$/ano, que se compreende face às diferenças na especificação das unidades. Salienta-se que, para o resultado apresentado na Tabela 5.13, a introdução das variáveis binárias permitiu determinar um perfil de alimentação semelhante ao determinado no GAMS, com o caudal a dar entrada junto ao topo, na zona reactiva da coluna. Ao contrário do teoricamente expectável na comparação dos resultados entre abordagens MINLP e NLP, em que os problemas NLP, menos restringidos, conduzem a melhores resultados, a abordagem estocástica apresentada revela um comportamento diferente. A introdução de variáveis discretas no conjunto de decisão traduz-se numa melhoria da precisão de procura nos domínios definidos, possibilitando a modelização de situações limite (ausência de alimentação) que conduzem a melhores valores para a função objectivo. A Tabela 5.14 apresenta a especificação dessa coluna.

Tabela 5.13 Problema P.CDR. Caso MINLP. ELV ideal. Abordagem SIMOP. Melhores resultados encontrados para o valor da função objectivo (F_{obj} , 10^7 \$/ano) em corridas de optimização em intervalos de procura diferenciados em função do excesso estequiométrico.

Exc.	MSGA		MSIMPISA	
	F_{obj}	N	F_{obj}	N
10%	1.565	6	1.510	10
20%	1.551	8	1.506	8
30%	1.549	8	1.502	8
40%	1.540	8	1.504	8
50%	1.550	7	1.506	8

Tabela 5.14 Resultados SIMOP. Problema P.CDR. Caso MINLP. ELV Ideal. Especificações da coluna de custo mínimo, cujo projecto corresponde a uma coluna com oito andares e uma $F_{obj} = 1.502 \cdot 10^7$ \$/ano.

Andar	$Y_{F,k}$	F_{1k} (mol/s)	F_{2k} (mol/s)	W_k m^3	T_k (K)	V_k mol/s	L_k mol/s
9	0	-	-	-	-	-	264.0
8	1	1.357	4.291	1.780	373.2	264.0	263.9
7	1	2.256	2.593	0.684	373.3	260.2	263.1
6	1	1.956	0.223	1.606	373.5	256.5	259.7
5	1	1.737	0.006	1.721	373.6	252.7	256.8
4	0	-	-	0.003	375.1	249.7	256.8
3	0	-	-	0.846	409.3	249.7	256.8
2	0	-	-	0.757	461.8	249.7	256.8
1	0	-	-	1.112	470.5	249.7	256.8
0	-	-	-	-	-	249.7	-

$$D = 1.72 \text{ m}, H = 9.60 \text{ m}, \beta = 0.980$$

$$Q_B = 10.5 \text{ MW}, Q_C = 11.1 \text{ MW}$$

Na avaliação para colunas de ELV não ideal, os melhores resultados encontrados apresentam-se na Tabela 5.15. O valor mínimo para o custo, $1.583 \cdot 10^7$ \$/ano, foi determinado com o MSGA numa avaliação de caudais normalizadas na gama até 20% de excesso estequiométrico. As especificações da coluna de custo mínimo associada estão apresentadas na Tabela 5.16. Neste caso particular a alimentação da coluna está ainda mais concentrada do que no resultado encontrado para o ELV ideal, mostrando os benefícios da introdução das variáveis binárias no modelo do problema. As mesmas permitem encontrar topologias de coluna mais exequíveis do ponto de vista prático, por sua vez mais próximas dos comportamentos registados nas soluções com o GAMS.

Nos resultados obtidos, em contraponto do que sucede para ELV ideal, os dois optimizadores não apresentam desempenho semelhante, registando-se resultados de qualidade inferior para o MSIMPSA. Os resultados obtidos para esse caso são melhores nas gamas intermédias dos intervalos de normalização de caudais na estimativa inicial, embora apresentem um erro relativo superior aos registados com MSGA.

A reprodutibilidade dos resultados encontrados foi avaliada para as duas situações da definição de ELV. A Figura 5.13 apresenta as distribuições cumulativas do erro relativo associado às determinações efectuadas com o SIMOP para o modelo MINLP ideal. A implementação da normalização do somatório dos caudais como ajuste dos valores estipulados pelos optimizadores para os caudais garantiu a obtenção de um resultado plausível para todas as corridas efectuadas. Todavia, o cálculo de erro face ao melhor resultado encontrado permite evidenciar que o desempenho da procura conduzida pelos optimizadores é relativamente insensível à amplitude do intervalo especificado para a normalização, mantendo-se sempre abaixo dos 10%.

As distribuições de erro relativo para as determinações em condições não ideais de equilíbrio apresentam-se na Figura 5.14. Foram encontrados resultados plausíveis grande maioria das corridas efectuadas. No entanto, e para além da maior amplitude de erro associada aos resultados quando comparados com o caso MINLP ideal, não foi encontrada uma taxa de resultados plausíveis satisfatória para o intervalo estequiométrico de normalização de 10%, razão pela qual tais registos não são considerados na Tabela 5.15.

Em termos gerais, regista-se que uma gama considerável de resultados está acima dos 10% em termos de erro relativo, factor que resultará da associação do grau combinatorial do problema com a sua maior dimensão no que diz respeito ao sistema de equações. Contudo, tal situação não inviabiliza a obtenção de resultados de qualidade semelhante aos registados por Lima et al. (2006) para colunas a funcionar em ELV não ideal e sujeitas a restrições de limite superior nos caudais de líquido e de vapor, onde a coluna de custo mínimo corresponde uma $F_{obj} = 1.586 \cdot 10^7$ \$/ano.

Tabela 5.15 Problema P.CDR. Caso MINLP. ELV não ideal Abordagem SIMOP. Melhores resultados encontrados para o valor da função objectivo (F_{obj} , 10^7 \$/ano) em corridas de optimização em intervalos de procura diferenciados em função do excesso estequiométrico.

Exc.	MSGA		MSIMPSPA	
	F_{obj}	N	F_{obj}	N
20%	1.583	7	1.839	6
30%	1.592	7	1.625	6
40%	1.592	7	1.806	8
50%	1.622	6	1.869	5

Tabela 5.16 Resultados SIMOP. Problema P.CDR. Caso MINLP. ELV não ideal. Especificações da coluna de custo mínimo, cujo projecto corresponde a uma coluna com sete andares e uma $F_{obj} = 1.583 \cdot 10^7$ \$/ano.

Andar	$Y_{F,k}$	F_{1k} (mol/s)	F_{2k} (mol/s)	W_k m^3	T_k (K)	V_k mol/s	L_k mol/s
8	0	-	-	-	-	-	277.5
7	1	5.870	4.141	0.122	475.7	277.5	270.3
6	1	1.778	3.481	0.284	476.8	266.2	270.4
5	0	-	-	0.542	477.0	262.8	270.4
4	0	-	-	0.442	479.7	262.8	270.4
3	0	-	-	0.218	506.1	262.8	270.4
2	0	-	-	0.007	553.6	262.8	270.4
1	0	-	-	0.006	608.2	262.8	277.0
0	-	-	-	-	-	262.8	-

$$D = 1.68 \text{ m}, H = 8.05 \text{ m}, \beta = 0.949$$

$$Q_B = 10.5 \text{ MW}, Q_C = 11.1 \text{ MW}$$

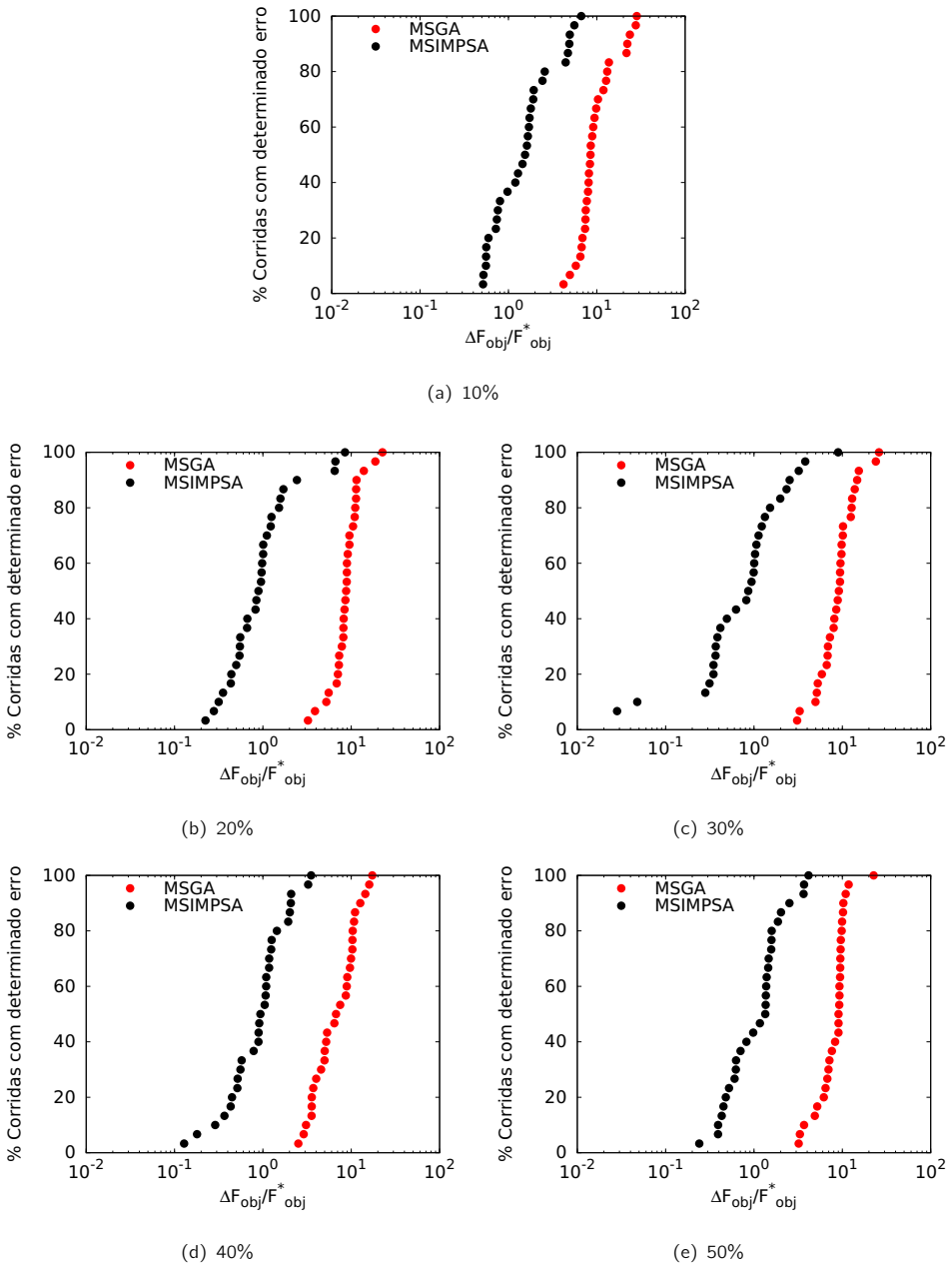
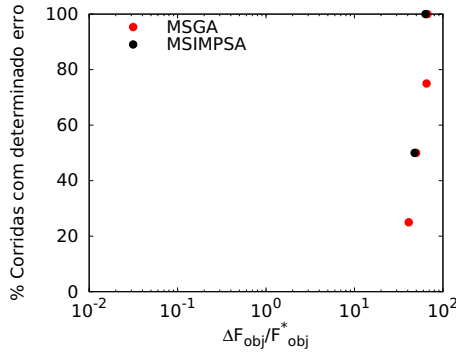
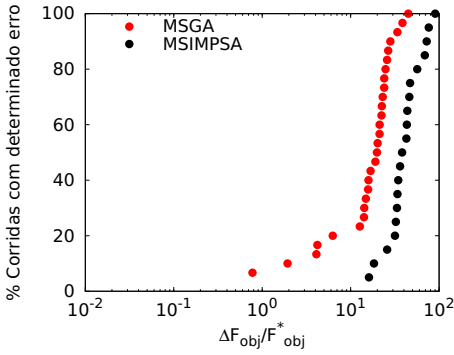


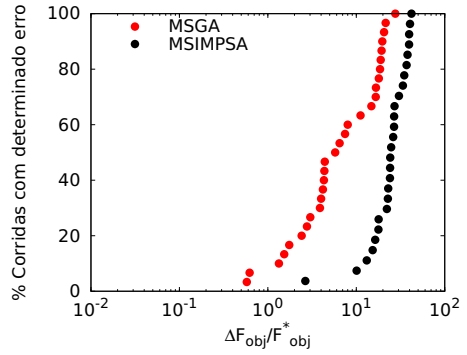
Figura 5.13 Resultados SIMOP. Problema P.CDR. Caso MINLP. ELV ideal. Distribuições cumulativas do erro relativo verificado nos resultados encontrados, com domínios de procura diferenciados em termos do excesso estequiométrico, utilizando a abordagem SIMOP.



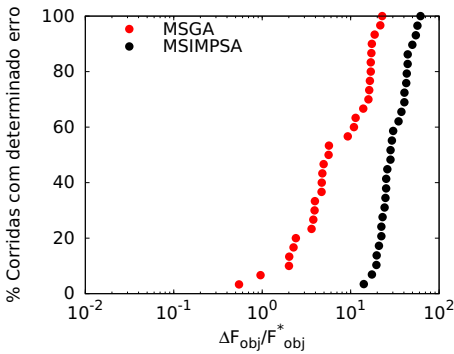
(a) 10%



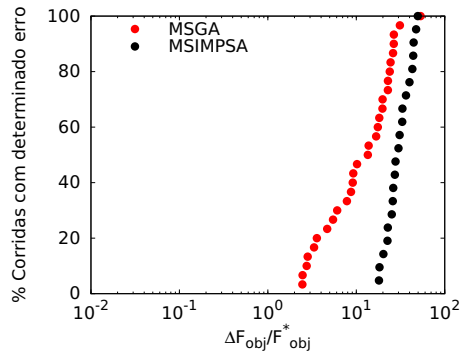
(b) 20%



(c) 30%



(d) 40%



(e) 50%

Figura 5.14 Resultados SIMOP. Problema P.CDR. Caso MINLP. ELV não ideal. Distribuições cumulativas do erro relativo verificado nos resultados encontrados, com domínios de procura diferenciados em termos do excesso estequiométrico, utilizando a abordagem SIMOP.

5.4 O comportamento ASCEND vs. SIMOP

Os resultados determinados mediante a aplicação da estratégia conjugada e do SIMOP mostram que as duas abordagens apresentam sensibilidades diferentes ao aumento de dimensão da coluna. Com o SIMOP, verifica-se que a rentabilidade da unidade piora ligeiramente à medida que se avaliam colunas com mais do que dez andares, resultado igualmente verificado por Cardoso et al. (2000) e Lima et al. (2006). Nas simulações com o ASCEND a deterioração dos resultados é significativamente mais acentuada, pelo que se procura aqui explicar os motivos que estão na base desta discrepância de comportamento.

Para isso serão efectuadas considerações de natureza estrutural sobre os modelos avaliados. Uma vez que a formulação MINLP se baseou na avaliação de diferentes casos NLP em ambas as metodologias, as conclusões da análise efectuada para os modelos NLP consideram-se aplicáveis às determinações efectuadas na formulação MINLP.

A Figura 5.15 representa os conjuntos de saída obtidos com o ASCEND e o SIMOP para uma coluna com quinze andares e ELV ideal. Nela se evidencia que em ambos os casos existe um subsistema de equações de dimensão considerável (S1). A sua dimensão relativa, quando comparada com todo o sistema de equações, é apresentada na Tabela 5.17. A dimensão total do sistema é diferente nas duas abordagens uma vez que a equação para o cálculo da função objectivo está incluída na contagem efectuada nos modelos para ASCEND.

A principal diferença entre as duas abordagens de simulação é que, no SIMOP, o subsistema obtido pelo algoritmo de partição e ruptura é dividido em duas partes (Luus, 1999; Shacham et al., 2002). A opção por esta divisão permite um tratamento diferente para cada uma das partes do subsistema. O subconjunto S1a corresponde a variáveis que são explicitamente calculadas enquanto que as remanescentes, alocadas no subconjunto S1b, são as variáveis de reciclagem do subsistema que são calculadas simultaneamente. Como apresentado na Tabela 5.17, a divisão do subsistema em duas partes conduz a uma significativa redução no número de equações que deve ser resolvido simultaneamente. Adicionalmente, e porque as variáveis do subconjunto S1a são explicitamente calculadas, a computação dos seus valores iniciais (estimativa) é desnecessária. A possibilidade que o SIMOP oferece de operar este rearranjo permite tornar o problema mais acessível do ponto de vista numérico. Para este caso particular, as únicas variáveis que precisam de ser inicializadas no SIMOP são os perfis de temperatura e de composição que são tratados como variáveis de reciclagem. A mudança do modelo de equilíbrio ideal para o de Wilson modifica a dimensão do modelo de $14N + 23$ para $38N + 23$ equações. A

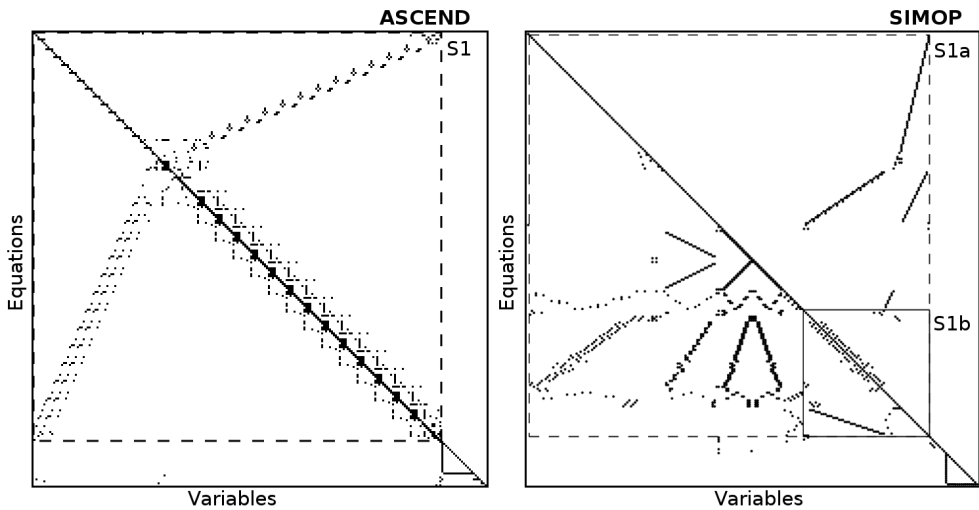


Figura 5.15 ELV ideal. Matrizes de incidência dos conjuntos de saída obtidos com ASCEND e SIMOP para uma coluna com quinze andares.

Tabela 5.17 ELV Ideal. Comparação dos resultados do passo de partição quando da utilização do ASCEND ou SIMOP. Os parâmetros apresentados são o número total de equações no sistema (neq), o número total de equações no subsistema (neq_s) e o número total de variáveis de reciclagem no subsistema (neq_t).

N	ASCEND		SIMOP		
	neq	neq_s	neq	neq_s	neq_t
7	121	103	120	102	22
10	163	142	162	141	44
15	233	207	232	206	68

aplicação dos algoritmos de partição produz um resultado análogo no que diz respeito aos conjuntos de saída dos modelos. (ver Figura 5.16 e Tabela 5.18).

As novas $24N$ equações fazem todas parte do bloco maior bloco (S1), incrementando o número de equações a ser resolvido simultaneamente e aumentando a carga numérica que deve ser ultrapassada para atingir uma solução satisfatória. Assim, se o aumento do número de andares condiciona directamente o número de equações a resolver simultaneamente, a mudança da descrição do sistema de equilíbrio para não ideal tem um impacto ainda mais significativo a este nível, pelo que o desvio verificado nos resultados é uma consequência da diferente dimensão nos sistemas de equações que têm que ser simultaneamente resolvidas pelo ASCEND (quando se usa o SIMOP, o número

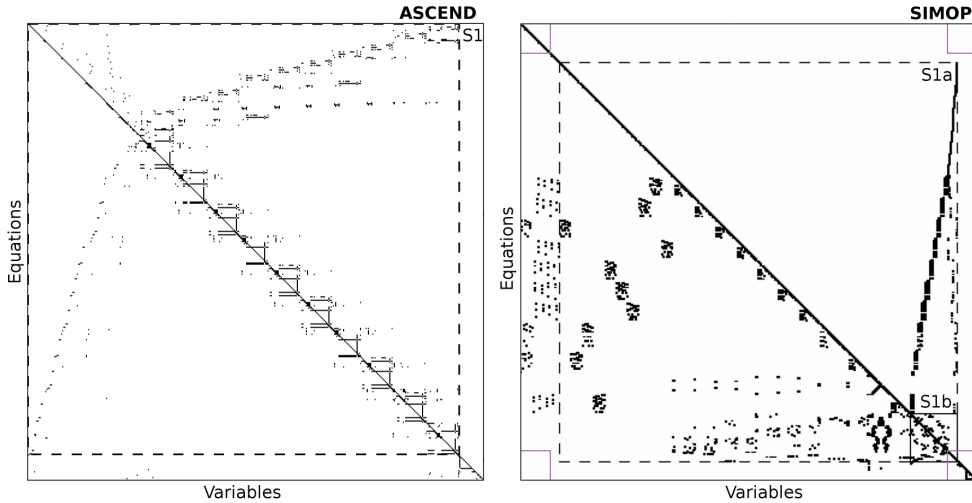


Figura 5.16 ELV não ideal. Matrizes de incidência dos conjuntos de saída obtidos com ASCEND e SIMOP para uma coluna de dez andares.

Tabela 5.18 ELV não ideal. Comparação dos resultados do passo de partição usando ASCEND e SIMOP. Os parâmetros apresentados são o número total de equações no sistema (neq), o número total de equações no subsistema (neq_s) e o número de variáveis de reciclagem no subsistema (neq_t).

N	ASCEND		SIMOP		
	neq	neq_s	neq	neq_s	neq_t
10	403	382	482	421	50

de equações apresentado na Tabela 5.18 é maior que $18N + 23$ uma vez que a definição de equilíbrio foi quebrada num maior número de termos explícitos para obedecer à estrutura dos ficheiros de dados).

O ASCEND não é uma aplicação dedicada a optimização como sua principal tarefa. Como explicado previamente, este ambiente de simulação orientado a equações não providencia ferramentas para escolher o conjunto de variáveis de decisão nem quaisquer características para modificar o conjunto de saída que se obtém no passo de decomposição estrutural do seu *solver*. Por outro lado, o SIMOP é capaz de construir simulações que permitem ao utilizador um gama de configuração mais alargada, o que pode fazer a diferença neste problema particular. O SIMOP permite gerir e modificar as variáveis de reciclagem do problema e estabelecer restrições sobre os conjuntos de saída gerados por forma a evitar problemas numéricos relacionados com mau condicionamento. Tal

tipo de procedimentos resulta em simulações mais estáveis, condição necessária para a utilização eficaz de optimizadores estocásticos.

5.5 Conclusões

Neste capítulo são apresentados os conceitos na base da adaptação da metodologia SIMOP a problemas com formulação MINLP. Os resultados obtidos para os dois casos de estudo apresentados garantem a reprodutibilidade dos resultados obtidos previamente com base em simulações codificadas manualmente, indicando que a nova versão da aplicação pode ser utilizada satisfatoriamente.

O estudo de um caso complexo e altamente não linear, um sistema de destilação reactiva, é pretexto para análise do desempenho do SIMOP e da estratégia conjugada proposta no Capítulo 4, que recorre ao simulador ASCEND como avaliador de desempenho. Regista-se que a total adaptação do SIMOP ao estudo de problemas MINLP poderá necessitar de algumas mudanças ao nível da forma como os seus modelos são construídos, assunto delegado para revisão do trabalho futuro no Capítulo 6. No entanto, a possibilidade de introdução de variáveis binárias permite modelizar mais fielmente a unidade em estudo, permitindo que os perfis de alimentação se adaptem melhor às duas zonas existentes na unidade, solução evidenciada pela avaliação do problema com recurso ao GAMS.

Na aplicação das estratégias integradas ao problema de destilação reactiva, na utilização do ASCEND verifica-se que os resultados se deterioram para modelos não ideais ou para ideais com mais de dez andares, onde um alargado número de equações tem que ser resolvido simultaneamente. A introdução de variáveis de folga nos caudais bem como a utilização de perfis sigmóide para temperaturas e composições foi implementada como alternativa ao procedimento de inicialização standard sem que se verificassem melhorias significativas na qualidade dos resultados obtidos. Contudo, melhores resultados podem ser obtidos quando um esquema de inicialização melhorado, baseado na normalização das variáveis de projecto é implementado.

A carga numérica associada a sistemas de equações de dimensão alargada é a razão provável que conduz à deterioração de resultados verificada quando se utiliza o conjunto de saída sugerido pelo algoritmo de decomposição do ASCEND. A necessidade de esquemas de inicialização muito detalhados pode ser minorada pela decomposição estrutural operada no SIMOP, que com a separação das variáveis de reciclagem dos subsistemas consegue reduzir o número de equações a ser resolvido simultaneamente.

Nomenclatura geral

Caracteres romanos

C_z	Coeficientes utilizados na expansão binária
$F(\mathbf{x})$	Função objectivo
$h_i(\mathbf{x})$	Restrição de igualdade de índice i
$g_j(\mathbf{x})$	Restrição de igualdade de índice j
m_1	Número de restrições de igualdade
m_2	Número de restrições de desigualdade
n	Número de equações
n_{exp}	Parâmetro para determinação do número de termos da expansão binária
n_{bin}	Número de termos associados à expansão binária de y
p	Número de variáveis contínuas
q	Número de variáveis de estado contínuas
v_D	Conjunto de variáveis de decisão
\mathbf{x}	Vector das variáveis contínuas
x_k	Variável contínua de índice k do vector \mathbf{x}
\mathbf{y}	Vector das variáveis discretas
y_{bin}	Variável discreta calculada por expansão binária
y_k	Variável discreta de índice k do vector \mathbf{y}
Y_z	Variável binária utilizada na expansão binária

Caracteres gregos

α_k	Limite inferior do domínio de x_k
β_k	Limite superior do domínio de x_k
γ_k	Limite inferior do domínio de y_k
ξ_k	Limite superior do domínio de y_k
γ_{bin}	Limite inferior do domínio de y_{bin}
ξ_{bin}	Limite superior do domínio de y_{bin}

Acrónimos e siglas

BARON	Branch and Reduce Optimization Navigator
C-MINLP	Abordagem MINLP do problema P.CDR por Cardoso et al. (2000)
ELV	Equilíbrio Líquido-Vapor
FORTTRAN	FORmula TRANslator
GAMS	General Algebraic Modeling System
IPOPT	Interior Point OPTimizer
I/O	Input/Output
MINLP	Mixed-Integer NonLinear Programming
Matlab	Matrix Laboratory

MSGA	Minlp Salcedo Gonçalves Azevedo
MSIMPASA	Minlp Simplex Simulated Annealing
NLP	NonLinear Programming
NNES	Nonmonotonic Nonlinear Equation Solver
OQNLP	OptQuest Nonlinear Programming
SIMOP	SIMulation for OPTimization

Nomenclatura do problema P.PRSIB15

Caracteres romanos

A	Componente A
B	Componente B
C	Componente C
$C_{inv,k}$	Custo de investimento na unidade U_k , \$/h
$C_{op,k}$	Custo de operação da unidade U_k , \$/t produto
$C_{st,i}$	Valor comercial do componente i , \$/t
F_i	Caudal mássico do componente i , t/h
F_{obj}	Função objectivo, \$/h
$F_{U_k}^{in}$	Caudal mássico de entrada na unidade U_k , t/h
$F_{U_k}^{out}$	Caudal mássico de saída na unidade U_k , t/h
U_k	Unidade k
$V_{P.PRSIB15}$	Conjunto de variáveis de decisão de P.PRSIB15
y_{U_k}	Variável binária que determina a existência da unidade U_k

Nomenclatura do problema P.PRSIB19

Caracteres romanos

F_1	Corrente de alimentação
F_2	Corrente de alimentação
$F_{j,i}$	Caudal do componente i na andar j
F_{obj}	Função Objectivo
$P_{j,i}$	Produção do componente i no ramo j
$V_{P.PRSIB19}$	Conjunto de variáveis de decisão de P.PRSIB19
Y_D	Variável binária
Y_F	Variável binária

Nomenclatura do problema P.CDR

Caracteres romanos

a_i	Constante para o cálculo de $K_{i,k}$ em ELV ideal
-------	--

$A_{i,n}$	Constantes de Antoine para o componente i , com $n = 1, 2, 3$
b_i	Constante para o cálculo de $K_{i,k}$ em ELV ideal
B_i	Caudal molar do componente i na base da coluna, mol/s
C_D	Constante relacionada com o cálculo de D , m^4s^2/mol^2
c_0	Parâmetro de custo, \$/ano
c_H	Custo do vapor de água, \$/W/ano
c_i	Custo do componente i , \$/mol
c_{SH}	Parâmetro de custo associado à coluna, \$/ano
c_T	Parâmetro de custo associado à coluna, \$/ano
c_W	Custo da água de arrefecimento, \$/W/ano
D	Diâmetro da coluna, m
$Dist$	Caudal molar de destilado, mol/s
E_a	Energia de activação, J/mol
$F_{i,k}$	Caudal molar do componente i alimentado ao andar k , mol/s
F_{max}	Limite máximo para os caudais molares de alimentação, mol/s
F_{obj}	Função objectivo a minimizar, \$/ano
F_{obj}^*	Melhor valor conhecido da função objectivo, \$/ano
H	Altura da coluna, m
H_0	Altura suplementar da coluna, m
H_k	Altura do andar k , m
H_{min}	Espaçamento entre andares, m
K_{ik}	Constante de ELV do componente i no andar k
$k_{0,j}$	Factor pré-exponencial para a reacção j , $mol/m^3/s$
$k_{j,k}$	Constante cinética da reacção j no andar k , $mol/m^3/s$
L_k	Caudal molar de líquido que sai do andar k , mol/s
L_{max}	Limite superior para o caudal de líquido L_k , mol/s
neq	Número total de equações do problema
neq_s	Numero de equações do subsistema
neq_t	Número de equações com incidência das variáveis reciclagem do subsistema
N	Número total de andares da coluna
N_C	Número de componentes
N_R	Número de reacções
P	pressão de operação da coluna, atm
P_i	Produção do componente i na coluna, mol/s
$P_{i,k}^{sat}$	Pressão de saturação do componente i no andar k , Pa
Q_B	Quantidade de calor transferida no reebulidor, W
Q_C	Quantidade de calor transferida no condensador, W
R	Constante universal dos gases, J/mol/K
$t_{a,i}$	Constante para o cálculo de $K_{i,k}$ em ELV ideal, K

$t_{b,i}$	Constante para o cálculo de $K_{i,k}$ em ELV ideal, K
T_k	Temperatura no andar k , K
$V_{P.CDR}$	Conjunto de variáveis de decisão de P.CDR
V_k	Caudal molar de vapor que sai do andar k , mol/s
V_{max}	Limite superior para o caudal de vapor V_k , mol/s
$V_{M,i}$	Volume molar do componente i , mL/mol
W_k	Volume de líquido no andar k , m ³
W_{max}	Volume máximo de líquido no andar k , m ³
$w_{i,j}$	Constantes de Wilson, cal/mol
$x_{d,i}$	Fracção molar do componente i no destilado
$x_{i,k}$	Fracção molar do componente i na fase líquida no andar k
$y_{i,k}$	Fracção molar do componente i na fase de vapor no andar k
$Y_{ANDAR,k}$	Variável binária que determina a existência do andar k
$Y_{F,k}$	Variável binária que determina a existência de alimentação no andar k

Caracteres gregos

β	Fracção de L_1 vaporizada no reebulidor
ΔH_j	Entalpia da reacção j , J/mol
ΔF_{obj}	Diferença entre o valor de F_{obj} e F_{obj}^*
$\gamma_{i,k}$	Coefficiente de actividade do componente i no andar k
λ	Entalpia de vaporização, J/mol
$\Lambda_{i,p,k}$	Parâmetro de equilíbrio de Wilson, interacção i com p no andar k
$\nu_{i,j}$	Coefficiente estequiométrico do componente i na reacção j
$\xi_{j,k}$	Extensão da reacção j no andar k , mol/s

Índices

i	Componente
j	Reacção
k	Andar
p	Componente

Capítulo 6

Conclusões e Perspectivas

A presente tese constitui o registo do trabalho desenvolvido para o estudo, desenvolvimento e adaptação de estratégias integradas de simulação e optimização, enquadradas na linha de investigação seguida pelo Grupo de Investigação em Engenharia de Sistemas de Processos do DEQ, FEUP. Parte substancial desse trabalho, que incidiu no estudo, interpretação e anotação de código-fonte, é dedicado ao diagnóstico do comportamento dessas abordagens na resolução de problemas, à aprendizagem dos diferentes modos de modelização associados e ainda à implementação de novas funcionalidades que possibilitem e ampliem o seu âmbito de aplicação.

A metodologia proposta nesta tese passa pela conjugação dos optimizadores estocásticos MSGA e MSIMPISA, com o ASCEND, um simulador orientado por equações, por intermédio do qual se pretende dar o primeiro passo para a exportação dos algoritmos estocásticos como *solvers* de optimização externos disponíveis para incorporação em simuladores. A nova etapa de desenvolvimento do ASCEND, iniciada em 2004 e que ainda não deu frutos sob a forma de uma nova versão etiquetada como estável, poderá representar uma oportunidade futura de adaptação destes optimizadores estocásticos naquele ambiente de simulação.

A utilização conjugada do ASCEND com os optimizadores MSGA e MSIMPISA é estabelecida segundo uma estratégia de comunicação, na forma de um esquema organizado de procedimentos, e com o qual se pretende gerir a partilha de informação entre o ASCEND e o MSGA e MSIMPISA. O seu uso torna o utilizador dependente do conhecimento prévio da sintaxe da linguagem de modelização ASCEND, assim como de alguns comandos básicos de atribuição de Tcl/Tk. Uma vez que o ambiente de simulação é capaz de comportar toda a decomposição estrutural do sistema autonomamente, o utilizador só precisa de considerar os parâmetros numéricos do solver

(ex. limite de iterações, tolerâncias absolutas e relativas para o critério de convergência, etc.). Problemas em que a activação de trajectos de optimização não plausíveis possam ser considerados úteis, implicam que o utilizador detenha conhecimentos básicos da linguagem FORTRAN.

A aplicação desta metodologia a casos de estudo em estado estacionário e transitente, propícios ao aparecimento de óptimos locais, demonstrou ser eficaz na obtenção dos melhores valores conhecidos para as respectivas funções objectivo. No entanto, atendendo ao facto da comunicação entre optimizadores e simulador ser baseada em *scripting*, a sua aplicação demonstra ter um tempo de computação cerca de vinte vezes mais elevado do que o registado com linguagens de programação compiladas, caso do FORTRAN. Apesar de tudo, a sua aplicação a um caso de estudo de destilação reactiva nas suas formulações NLP e MINLP com (respeito ao parâmetro N), demonstrou ser reprodutível e eficaz para dimensões de coluna até dez andares de equilíbrio.

A deterioração dos resultados que se verifica para colunas com um maior número de pratos, ou para situações de equilíbrio não ideal, que implicam um maior número de equações, é atribuída ao facto dos sistemas de equações envolvidos serem abordados simultaneamente na sua resolução. Os modelos escritos para ASCEND seguem a mesma estrutura dos codificados para o SIMOP. Contudo, a menor qualidade dos resultados obtidos com o ASCEND poderá eventualmente ser ultrapassada por uma reformulação cuidada do modelo, incluindo procedimentos de inicialização mais elaborados como os que, por exemplo, estiveram na base dos resultados obtidos com a aplicação GAMS.

O SIMOP, metodologia proposta por Lima (2005), é uma ferramenta dedicada ao tratamento sistemático das restrições de igualdade em formulações NLP. A sua expansão para abordagem sistemática a problemas MINLP, também objectivo desta tese, demonstrou ser eficaz na obtenção dos melhores valores registados anteriormente com simulações codificadas manualmente, o que permitiu considerar um funcionamento correcto da metodologia ao nível da geração dos procedimentos numéricos para resolução em FORTRAN.

A separação formal das variáveis discretas e contínuas no conjunto de variáveis de decisão dos problemas constituiu o primeiro passo na exploração mais detalhada de outras componentes da formulação dos problemas que não as restrições de igualdade. A diferenciação das variáveis discretas permite implementar modelos MINLP através do SIMOP. Modificações futuras do SIMOP, ao nível por exemplo, da análise estrutural das restrições de desigualdade, tornarão possível a escrita automática de modelos MINLP em diferentes formulações complementares.

A aplicação do SIMOP ao modelo MINLP do caso de estudo da coluna de desti-

lação revelou algumas condicionantes da metodologia. A esse nível, a possibilidade da utilização de disjunções poderá ser tomada em consideração na escrita de modelos, que poderão incorporar definições explícitas de alternativas topológicas.

Relativamente aos resultados obtidos para o caso de estudo da destilação reactiva com formulação MINLP, verifica-se que os mesmos diferem essencialmente num ponto relativamente a abordagens anteriores utilizando modelos NLP com o MSGA e MSIMPISA. A incorporação de variáveis binárias na definição da existência de alimentação em cada andar permite encontrar esquemas de alimentação semelhantes aos determinados com o modelo GAMS, com a alimentação a ser efectuada junto ao topo da coluna, na zona reactiva, e sem valores residuais de alimentação junto à base. Pode considerar-se que este tipo de resultados confirma a importância que as formulações MINLP apresentam no contexto dos problemas de projecto em Engenharia Química. Neste caso particular, a introdução de variáveis binárias permite colmatar a dificuldade que os optimizadores têm em aproximar-se dos limites do domínio contínuo das variáveis de decisão. Apesar dos algoritmos disponibilizarem a funcionalidade do teste nos extremos nesse domínio, em determinados casos essa avaliação torna-se incomportável dada a dimensão do espaço combinatorial associado a vectores de decisão com dimensão considerável.

No que diz respeito aos valores obtidos para função objectivo, considerando o mesmo esquema de inicialização utilizado com o ASCEND, as simulações implementadas com o SIMOP comportam-se melhor para dimensões de coluna superiores. Esse comportamento é atribuído à capacidade de reorganização estrutural que pode ser operada ao nível dos subsistemas, nomeadamente ao posicionamento das variáveis de reciclagem, o que permite reduzir o número real de incógnitas a determinar simultaneamente, reduzindo assim a dimensão do vector de inicialização.

O estudo das estratégias integradas de simulação e optimização apresentadas conduziu a resultados que, na generalidade, podem ser considerados satisfatórios. A obtenção de resultados de menor qualidade em alguns casos ajudam a identificar aspectos que se pretende melhorar. A introdução do SIMOP como ferramenta da disciplina de Estratégia e Optimização de Processos, no âmbito do Mestrado Integrado em Engenharia Química leccionado pela Faculdade de Engenharia da Universidade do Porto, será um instrumento importante na recolha de dados sobre a utilização desta metodologia. O seu alargamento a utilizadores independentes ao ambiente de investigação onde o SIMOP é desenvolvido fornecerá certamente pistas que ajudem a enquadrar as prioridades do trabalho a desenvolver no futuro.

6.1 Perspectivas e Trabalho Futuro

O desenvolvimento futuro da estratégia conjugada, baseada na utilização do MSGA e MSIMPISA com recurso ao ASCEND, deverá incidir na adopção de formas alternativas de modelização. Pretende-se com isso explorar as características de análise recursiva do ASCEND, permitindo que os modelos sejam resolvidos de uma forma mais expedita. Para além disso, a incorporação robusta destes algoritmos como *solvers* daquele ambiente de simulação só poderá ser conseguida por intermédio da cooperação com o grupo de programadores que desenvolve o ASCEND, o que possibilitaria o aproveitamento da interface já disponibilizada para o algoritmo CONOPT. A integração dos optimizadores com o ASCEND beneficiará de uma das direcções de trabalho futuro apontada para as ferramentas desenvolvidas no Grupo de Investigação em Engenharia de Sistemas de Processos.

No que diz respeito às linhas de desenvolvimento futuro do SIMOP e dos optimizadores MSGA e MSIMPISA, devem ter-se em consideração as observações a esse nível apresentadas por Lima (2005), acrescentando-se aqui mais três direcções importantes, nomeadamente:

1. **Reimplementação dos optimizadores MSGA e MSIMPISA.** A evolução dos meios computacionais, dos sistemas operativos e, conseqüentemente, dos compiladores e bibliotecas disponíveis pode apresentar-se como uma entrave à aplicação contínua destas ferramentas de optimização estocástica escritas em FORTRAN 77. Atendendo a esse facto, a transcrição dos algoritmos estocásticos para uma variante mais recente da linguagem FORTRAN permitiria um melhor aproveitamento de valências disponíveis.

As novas versões do MSIMPISA e MSGA incluiriam a possibilidade de compilação com vista à integração em ambientes de simulação externos (como o ASCEND ou o GAMS); de execução paralelizada, aproveitando o potencial de mais que uma máquina ou, simplesmente, da disponibilidade actual de máquinas com processadores *multi-core* que possam estar dedicadas exclusivamente à resolução de problemas. O aproveitamento de formulações do tipo caixa-negra poderia criar as condições necessárias para a utilização de MSGA e MSIMPISA numa plataforma do tipo optimização baseada em agentes.

2. **Inclusão automática de procedimentos de inicialização.** A análise de casos de estudo como o exemplo de destilação reactiva evidencia a importância dos procedimentos de inicialização na obtenção de boas estimativas iniciais na resolução dos

sistemas de equações não lineares. Nesse sentido, a existência de um ficheiro de dados adicional, que inclua as relações a ter em consideração num procedimento de inicialização, permitiria ao SIMOP a incorporação directa do cálculo preliminar de estimativas no respectivo código FORTRAN de simulação.

3. **Mudança das técnicas de modelização no SIMOP.** A este nível apontam-se duas situações distintas: (1) A possibilidade de escrita de código indexado, que permite codificar mais facilmente rotinas de inicialização para o caso dos problemas que assim o justifiquem, bem como a operação de pequenas alterações nas simulações criadas e (2) A possibilidade de modelização segundo uma lógica disjuntiva, que permitirá poupar tempo de execução bem como a ocorrência de incongruências matemáticas derivadas da ausência de determinados equipamentos nos diagramas processuais dos problemas avaliados.

Referências

- Abadie, J. e Carpentier, J. (1969), *Optimization*, Academic Press, Capítulo Generalization of the Wolfe reduced gradient method to the case of nonlinear constraints, Páginas 37–47.
- Abbott, K. A. (1996), *Very Large Scale Modeling*, Tese de Doutorado, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.
- Adjiman, C. S., Androulakis, I. P. e Floudas, C. A. (1997), 'Global optimization of MINLP problems in process synthesis and design', *Computers & Chemical Engineering*, **21**, S445–S450.
- Ali, M. M. e Storey, C. (1997), 'Aspiration based simulated annealing algorithm', *Journal of Global Optimization*, **11**(2), 181–191.
- Ali, M. M., Torn, A. e Viitanen, S. (1997), 'A numerical comparison of some modified controlled random search algorithms', *Journal of Global Optimization*, **11**(4), 377–385.
- Allan, B. A. (1997), *A More Reusable Modeling System*, Tese de Doutorado, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.
- Arora, J. S. (1989), *Introduction to Optimum Design*, Mechanical Engineering Series, McGraw-Hill International Editions.
- ASCEND Developers (2008a), 'ASCEND modelling environment', URL <http://ascend.cheme.cmu.edu/>.
- ASCEND Developers (2008b), 'ASCEND Wiki', URL <http://ascendwiki.cheme.cmu.edu/>.
- ASCEND Developers (2008c), 'ASCEND Wiki - History', URL http://ascendwiki.cheme.cmu.edu/History_of_ASCEND.
- ASCEND Developers (2008d), 'ASCEND Wiki - Users', URL http://ascendwiki.cheme.cmu.edu/ASCEND_Users_and_Friends.
- ASCEND Development Team (1998), 'ASCEND how-to', URL <http://www.ndim.edrc.cmu.edu/ascend/pdfhelp/ascend-help.pdf>.
- Azencott, R. (Editor) (1992), *Simulated Annealing - Parallelization Techniques*, John Wiley &

- Sons, Inc.
- Back, T., Hoffmeister, F. e paul Schwefel, H. (1991), 'A survey of evolution strategies', em *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann, Páginas 2–9.
- Balakrishnan, V. e Boyd, S. (1992), 'Global optimization in control system analysis and synthesis', Information Systems Laboratory; Departament of Electrical Engineering; Stanfor University; Stanford, CA 94305; U.S.A., Volume 53 de *Control and Dynamic Systems*.
- Banga, J. R., Alonso, A. A. e Singh, R. P. (1997), 'Stochastic dynamic optimization of batch and semicontinuous bioprocesses', *Biotechnology Progress*, **13**(3), 326 – 335.
- Banga, J. R., Balsa-Canto, E., Moles, C. G. e Alonso, A. A. (2003a), 'Dynamic optimization of bioreactors: A review', *Proceedings of the Indian National Science Academy*.
- Banga, J. R., Balsa-Canto, E., Moles, C. G. e Alonso, A. A. (2003b), 'Improving food processing using modern optimization methods', *Trends in Food Science & Technology*, **14**(4), 131 – 144.
- Banga, J. R. e Seider, W. D. (1996), 'Global optimization of chemical processes using stochastic algorithms', em *State Of The Art In Global Optimization - Computational Methods And Applications*, Kluwer Academic Publ, Po Box 17, 3300 Aa Dordrecht, Netherlands, Volume 7 de *Nonconvex Optimization And Its Applications*, Páginas 563 – 583.
- Barton, P. I., Banga, J. R. e Galan, S. (2000), 'Optimization of hybrid discrete/continuous dynamic systems', *Computers & Chemical Engineering*, **24**(9-10), 2171 – 2182.
- Bazaraa, M. S., Sherali, H. D. e Shetty, C. (1993), *Nonlinear Programming - Theory and Algorithms*, John Wiley & Sons, Inc, 2 Edição.
- Bezdek, D. (2007), 'A proof of an extension of the icosaedral conjecture of Steiner for generalizes deltaedra', *Contributions to Discrete Mathematics*, **2**, 86–92.
- Biegler, L. T. e Grossmann, I. (2007), 'Capd short course - notes', .
- Biegler, L. T. e Grossmann, I. E. (2004), 'Retrospective on optimization', *Computers & Chemical Engineering*, **28**(8), 1169 – 1192.
- Biegler, L. T., Grossmann, I. E. e Westerbeg, A. W. (1997), *Systematic Methods of Chemical Process Design*, International Series in the Physical and Chemical Engineering Sciences, Prentice Hall.
- Biegler, L. T., seader, J. D. e Seider, W. D. (1996), 'Process simulation', *Computers in Chemical Engineering Education*, 161–170, URL <http://www.che.utexas.edu/cache/compeng.html>.
- Borchers, B. e Mitchel, J. E. (1991), *An Improved branch and bound algorithm for mixed integer*

- nonlinear programs*, RPI Math Report 200, Rensselaer Polytechnic Institute.
- Brooke, A., Kendrick, D., Meeraus, A. e Raman, R. (1998), *GAMS - A user's guide*, URL <http://www.gams.com>.
- Cardoso, M. F., Lima, R. M. e Salcedo, R. L. (2001), *The MSIMPISA Program User Manual*, Departamento de Engenharia Química - Faculdade de Engenharia da Universidade do Porto.
- Cardoso, M. F., Salcedo, R. L. e de Azevedo, S. F. (1994), 'Nonequilibrium simulated annealing: A faster approach to combinatorial minimization', *Industrial & Engineering Chemistry Research*, **33**(8), 1908–1918.
- Cardoso, M. F., Salcedo, R. L. e de Azevedo, S. F. (1996), 'The simplex-simulated annealing approach to continuous non-linear optimization', *Computers & Chemical Engineering*, **20**(9), 1065 – 1080.
- Cardoso, M. F., Salcedo, R. L., de Azevedo, S. F. e Barbosa, D. (1997), 'A simulated annealing approach to the solution of MINLP problems', *Computers & Chemical Engineering*, **21**(12), 1349 – 1364.
- Cardoso, M. F., Salcedo, R. L., de Azevedo, S. F. e Barbosa, D. (2000), 'Optimization of reactive distillation processes with simulated annealing', *Chemical Engineering Science*, **55**(21), 5059 – 5078.
- Cardoso, M. M. F. C. (1998), *À Procura do Ótimo Global - Desenvolvimento de uma Estratégia para Optimização Global Não-Linear com Variáveis Contínuas e Discretas*, Tese de Doutoramento, Faculdade de Engenharia da Universidade do Porto.
- Chen, C. T. e Hwang, C. (1990), 'Optimal-control computation for differential-algebraic process systems with general constraints', *Chemical Engineering Communication*, **97**, 9 – 26.
- Choi, S. H., Ko, J. W. e Manousiouthakis, V. (1999), 'A stochastic approach to global optimization of chemical processes', *Computers & Chemical Engineering*, **23**(9), 1351 – 1356.
- Choi, S. H. e Manosuouthakis, V. (2002), 'Global optimization methods for chemical process design: Deterministic and stochastic approaches', *Korean Journal of Chemical Engineering*, **19**(2), 227–232.
- Circic, A. R. (1995), 'Personal communication', .
- Circic, A. R. e Gu, D. Y. (1994), 'Synthesis of nonequilibrium reactive distillation processes by MINLP optimization', *AIChE Journal*, **40**(9), 1479 – 1487.
- Clocksinn, W. e Mellish, C. (1981), *Programmin in prolog*, Springer-Verlag New York, Inc.
- Conn, A. R., Gould, N. I. M. e Toint, P. L. (1992), *Lancelot: A FORTRAN Package for Large-Scale Nonlinear Optimization (Release A)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

- Dolan, W. B., Cummings, P. T. e Levan, M. D. (1989), 'Process optimization via simulated annealing - application to network design', *AIChE Journal*, **35**(5), 725–736.
- Douglas, J. (1988), *Conceptual design of chemical processes*, McGraw-Hill Higher Education, 4 Edição.
- Drud, A. (1985), 'CONOPT - a grg code for large sparse dynamic nonlinear optimization problems', *Mathematical Programming*, **31**, 2.
- Drud, A. (2008), *CONOPT*, URL <http://www.gams.com/dd/docs/solvers/conopt.pdf>, <http://www.gams.com/docs/document.htm>.
- Drud, A. S. (1994), 'CONOPT - a large-scale GRG code', *ORSA Journal on Computing*, **6**(2), 207–218.
- Duran, M. A. e Grossmann, I. E. (1986), 'Simultaneous-optimization and heat integration of chemical processes', *Aiche Journal*, **32**(1), 123–138.
- Edgar, T. F. e Himmelblau, D. M. (1989), *Optimization of Chemical Processes*, McGraw-Hill International Editions.
- Edgar, T. F., Himmelblau, D. M. e Lasdon, L. S. (2001), *Optimization of Chemical Processes*, McGraw-Hill, 2 Edição.
- Eddie, F. e Westerberg, A. (1971), 'Computer aided design: Part 3 - design variable selection to avoid hidden singularities in resulting recycle calculations', *Chemical Engineering Journal*, **2**, 114–124.
- Faber, R., Jockenhovel, T. e Tsatsaronis, G. (2005), 'Dynamic optimization with simulated annealing', *Computers & Chemical Engineering*, **29**(2), 273 – 290.
- Ferreira, E. C. e Salcedo, R. (2001a), 'Can spreadsheet solvers solve demanding optimization problems?', *Computer Applications in Engineering Education*, **9**(1), 49 – 56.
- Ferreira, E. C. e Salcedo, R. (2001b), 'A tool to optimize VOC removal during absorption and stripping', *Chemical Engineering*, **108**(1), 94 – 98.
- Fletcher, R. e Leyffer, S. (1994), 'Solving mixed-integer nonlinear programs by outer approximation', *Mathematical Programming*, **66**(3), 327–349.
- Floudas, C. A. (1995), *Nonlinear and Mixed-Integer Optimization - Fundamentals and Applications*, Oxford University Press.
- Floudas, C. A. (1997), 'Deterministic global optimization in design, control, and computational chemistry', em L. C. L.T. Biegler, A. Conn e F. Santosa (Editores), *Large Scale Optimization with Applications. Part III: Optimal Design and Control*, Department of Chemical Engineering; Princeton University; Princeton, N.J. 08544-5263; U.S.A., Volume 93, Páginas 129–184.

- Floudas, C. A. e Ciric, A. R. (1989), 'Strategies for overcoming uncertainties in heat-exchanger network synthesis', *Computers & Chemical Engineering*, **13**(10), 1133–1152.
- Floudas, C. A. e Pardalos, P. M. (1995), 'State-of-the-art in global optimization - Computational methods and applications - Preface', *Journal of Global Optimization*, **7**(2), 113–113.
- Fu, M. C., Glover, F. W. e April, J. (2005), 'Simulation optimization: A review, new developments, and applications', em *Proceedings of the 2005 Winter Simulation Conference*.
- Gaines, L. D. e Gaddy, J. L. (1976), 'Process optimization by flow sheet simulation', *Industrial & Engineering Chemistry Process Design And Development*, **15**(1), 206–211.
- Geoffrion, A. M. (1972), 'Generalized benders decomposition', *Journal of Optimization Theory and Applications*, **10**(4), 237–260, URL <http://dx.doi.org/10.1007/BF00934810>.
- Gill, P. E., Murray, W., Saunders, M. A., Drud, A. e Kalvelagen, E. (2008), *SNOPT*, URL <http://www.gams.com/dd/docs/solvers/snopt.pdf>, <http://www.gams.com/docs/document.htm>.
- Gill, P. E., Murray, W. e Wright, M. H. (1981), *Practical Optimization*, Academic Press Limited.
- Glover, F. (1989), 'Tabu search - part i', *ORSA Journal on Computing*, **1**(3), 190.
- Glover, F. (1990a), 'Tabu search - part ii', *ORSA Journal on Computing*, **2**(1), 4.
- Glover, F. (1990b), 'Tabu search: A tutorial', *Interfaces*, **4**, 74–94.
- Grossmann, I. (2005), 'Enterprise-wide optimization: A new frontier in process systems engineering', *AIChE Journal*, **51**(7), 1846–1857.
- Grossmann, I. E. (2004), 'Challenges in the new millennium: product discovery and design, enterprise and supply chain optimization, global life cycle assessment', *Computers & Chemical Engineering*, **29**(1), 29–39.
- Grossmann, I. E. e Biegler, L. T. (1995), 'Optimizing chemical processes', *Chemtech*, **25**(12), 27–35.
- Grossmann, I. E. e Biegler, L. T. (2004), 'Part II. Future perspective on optimization', *Computers & Chemical Engineering*, **28**(8), 1193 – 1218.
- Grossmann, I. E. e Kravanja, Z. (1995), 'Mixed-integer nonlinear-programming techniques for process systems-engineering', *Computers & Chemical Engineering*, **19**, S189–S204.
- Grossmann, I. E. e Turkay, M. (1996), 'Solution of algebraic systems of disjunctive equations', *Computers & Chemical Engineering*, **20**, S339–S344.
- Grossmann, I. E., Viswanathan, J., Vecchiotti, A., Raman, R. e Kalvelagen, E. (2008), *DICOPT*, URL <http://www.gams.com/dd/docs/solvers/dicopt.pdf>,

- <http://www.gams.com/docs/document.htm>.
- Gupta, O. K. (1980), *Branch And Bound Experiments In Nonlinear Integer Programming*, Tese de Doutorado, Purdue University, United States – Indiana.
- Gupta, O. K. e Ravindran, A. (1985), 'Branch and bound experiments in convex nonlinear integer programming', *Management Science*, **31**(12), 1533–1546.
- Hettich, R. e Kortanek, K. O. (1993), 'Semiinfinite programming - theory, methods, and applications', *Siam Review*, **35**(3), 380–429.
- Holland, J. H. (1975), *Adaptation in natural and artificial systems - an introductory analysis with applications to biology, control, and artificial intelligence*, University of Michigan Press.
- Holmberg, K. (1990), 'On the convergence of cross decomposition', *Mathematical Programming*, **47**(2), 269–296.
- Hooker, J. N. e Osorio, M. A. (1999), 'Mixed logical-linear programming', *Discrete Applied Mathematics*, **97**, 395–442.
- Hoover, V., Stewart e Perry, R. F. (1989), *Simulation - a problem-solving approach*, Addison-Wesley.
- Horst, R., Pardalos, P. M. e Thoai, N. V. (1995), *Introduction to Global Optimization*, Volume 3 de *Nonconvex Optimization and its Applications*, Kluwer Academic Publishers.
- Huss, R. S. (1995), *Collocation Methods for flexible distillation design*, Tese de Doutorado, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.
- Ishii, Y. e Otto, F. D. (2008), 'Novel and fundamental strategies for equation-oriented process flowsheeting part i: A basic algorithm for inter-linked, multicolumn separation processes', *Computers & Chemical Engineering*, **32**(8), 1842–1860.
- Kaelo, P. e Ali, M. (2006), 'Some variants of the controlled random search algorithm for global optimization', *Journal of Optimization Theory and Applications*, **130**(2), 253–264, URL <http://dx.doi.org/10.1007/s10957-006-9101-0>.
- Kalivas, J. H. (Editor) (1995), *Adaptation of simulated annealing to chemical optimization problems*, Volume 15 de *Data Handling in Science and Technology*, Elsevier.
- Kelley, J. E. (1960), 'The cutting-plane method for solving convex programs', *Journal Of The Society For Industrial And Applied Mathematics*, **8**(4), 703–712.
- Kirkpatrick, S. (1984), 'Optimization by simulated annealing - quantitative studies', *Journal Of Statistical Physics*, **34**(5-6), 975–986.
- Kirkpatrick, S., Gelatt, C. D. e Vecchi, M. (1983), 'Optimization by simulated annealing', *Science*, (4598).

- Kleywegt, A. e Shapiro, A. (2001), *Handbook of Industrial Engineering*, John Wiley & Sons, New York, NY, Capítulo Stochastic Optimization, 3 Edição, Páginas 2625–2649.
- Kocis, G. R. e Grossmann, I. E. (1987), 'Relaxation strategy for the structural optimization of process flow sheets', *Industrial & Engineering Chemistry Research*, **26**(9), 1869–1880.
- Kocis, G. R. e Grossmann, I. E. (1988), 'Global optimization of nonconvex mixed-integer nonlinear-programming (minlp) problems in process synthesis', *Industrial & Engineering Chemistry Research*, **27**(8), 1407–1421.
- Kravanja, S., Kravanja, Z. e Bedenik, B. S. (1998a), 'The minlp optimization approach to structural synthesis - part i: A general view on simultaneous topology and parameter optimization', *International Journal for Numerical Methods in Engineering*, **43**(2), 263–292.
- Kravanja, S., Kravanja, Z. e Bedenik, B. S. (1998b), 'The minlp optimization approach to structural synthesis - part ii: Simultaneous topology, parameter and standard dimension optimization by the use of the linked two-phase minlp strategy', *International Journal for Numerical Methods in Engineering*, **43**(2), 293–328.
- Kravanja, S., Kravanja, Z. e Bedenik, B. S. (1998c), 'The minlp optimization approach to structural synthesis - part iii: Synthesis of roller and sliding hydraulic steel gate structures', *International Journal for Numerical Methods in Engineering*, **43**(2), 329–364.
- Kuru, S. (1981), *Dynamic Simulation With an Equation Based Flowsheeting System Thesis*, Tese de Doutorado, Carnegie Mellon University.
- Lasdon, L., Plummer, J., Ugray, Z. e Bussieck, M. (2004), *Improved Filters and Randomized Drivers for Multistart Global Optimization*, Relatório Técnico, MSIS Department, McCombs College of Business, The University of Texas at Austin.
- Lasdon, L. S. e Waren, A. D. (1981), 'Grg2: an all fortran general purpose nonlinear optimizer', *SIGMAP Bull.*, (30), 10–11.
- Lasdon, L. S., Waren, A. D., Jain, A. e Ratner, M. (1978), 'Design and testing of a generalized reduced gradient code for nonlinear programming', *ACM Trans. Math. Softw.*, **4**(1), 34–50.
- Ledet, W. e Himmelblau, D. (1970), *Advances in Chemical Engineering*, Elsevier, Volume 8, Capítulo Decomposition procedures for the solving of large scale systems, Páginas 185–254.
- Lee, J. W., Huan, S., Lien, K. M. e Westerberg, A. W. (2000), 'Difference points in extractive and reactive cascades. ii - generating design alternatives by the lever rule for reactive systems', *Chemical Engineering Science*, **55**(16), 3161–3174.
- Lee, S. e Grossmann, I. E. (2000), 'New algorithms for nonlinear generalized disjunctive programming', *Computers & Chemical Engineering*, **24**(9-10), 2125–2141.
- Lee, S. e Grossmann, I. E. (2005), 'Logic-based modeling and solution of nonlinear discrete/-continuous optimization problems', *Annals Of Operations Research*, **139**(1), 267–288.

- Lee, W., Christensen, J. H. e Rudd, D. F. (1966), 'Design variable selection to simplify process calculations', *AIChE Journal*, **12**(6), 1104–1115, URL <http://dx.doi.org/10.1002/aic.690120613>.
- Leyffer, S. (2001), 'Integrating SQP and branch-and-bound for mixed integer nonlinear programming', *Computational Optimization and Applications*, **18**(3), 295 – 309.
- Lima, R. (2006), *SIMOP - Manual de utilização*, Departamento de Engenharia Química - Faculdade de Engenharia da Universidade do Porto.
- Lima, R. M., Francois, G., Srinivasan, B. e Salcedo, R. L. (2004), 'Dynamic optimization of batch emulsion polymerization using MSIMPISA, a simulated-annealing-based algorithm', *Industrial & Engineering Chemistry Research*, **43**(24), 7796 – 7806.
- Lima, R. M., Salcedo, R. L. e Barbosa, D. (2006), 'SIMOP: Efficient reactive distillation optimization using stochastic optimizers', *Chemical Engineering Science*, **61**, 1718–1739.
- Lima, R. M. P. (2005), *SIMOP - Uma estratégia Integrada de Simulação-Optimização de Processos Químicos*, Tese de Doutorado, Faculdade de Engenharia da Universidade do Porto.
- Locke, M. H. e Westerberg, A. W. (1983), 'The ASCEND-II system - a flowsheeting application of a successive quadratic programming methodology', *Computers & Chemical Engineering*, **7**(5), 615 – 630.
- Lohmann, B. e Marquardt, W. (1996), 'On the systematization of the process of model development', *Computers & Chemical Engineering*, **20**, S213–S218.
- Lopez, M. e Still, G. (2007), 'Semi-infinite programming', *European Journal Of Operational Research*, **180**(2), 491–518.
- Luenberger, D. G. (1984), *Linear and Nonlinear Programming*, Addison-Wesley Publishing Company.
- Luus, R. (1973), 'Handling of equality constraints in optimization', *Aiche Journal*, **19**(5), 1087–1087.
- Luus, R. (1999), 'Effective solution procedure for systems of non-linear algebraic equations', *Hungarian Journal of Industrial Chemistry*, **27**(4), 307–310.
- Lúcio, L. e Amaral, V. (2001), *Programação em PERL*, FCA, 1 Edição.
- Maltez, N. N. F. e Oliveira, N. M. C. (2001), 'Using flowsheeting tools for process optimization: Case studies', em F. R. Ribeiro e J. J. C. C. Pinto (Editores), *CHEMPOR 2001 - 8th International Chemical Engineering Conference*, Páginas 957 – 964.
- Marquardt, W. (1996), 'Trends in computer aided process modeling', *Computers & Chemical Engineering*, **20**, 591–609.

- Mawengkang, H. e Murtagh, B. (1986), 'Solving nonlinear integer programs with large-scale optimization software', *Annals of Operations Research*, **5**(1), 425–437, URL <http://dx.doi.org/10.1007/BF02739232>.
- McKinnon, K. I. M., Millar, C. G. e Mongeau, M. (), 'Global optimization for the chemical and phase equilibrium problem using interval analysis', URL <http://www.mip.ups-tlse.fr/~mongeau/>.
- Metropolis, N. (1987), 'The beginning of the monte carlo method', *Los Alamos Science*, 125–130.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. e Teller, E. (1953), 'Equation of state calculations by fast computing machines', *The Journal of Chemical Physics*, **21**(6), 1087–1092, URL <http://link.aip.org/link/?JCP/21/1087/1>.
- Morton, W. (2003), 'Equation-Oriented Simulation and Optimization', em *Proc Indian Natn Sci Acad*, Volume 69, A, Páginas 317–357.
- Murtagh, B. A. e Saunders, M. A. (1982), 'A projected lagrangian algorithm and its implementation for sparse non-linear constraints', *Mathematical Programming Study*, **16**(MAR), 84–117.
- Murtagh, B. A., Saunders, M. A., Murray, W., Gill, P. E., Raman, R. e Kalvelagen, E. (2008), *MINOS*, URL <http://www.gams.com/dd/docs/solvers/minos.pdf>, <http://www.gams.com/docs/document.htm>.
- Nash, S. e Sofer, A. (1996), *Linear and Nonlinear Programming*, McGraw-Hill International Editions.
- Nelder, J. A. e Mead, R. (1965), 'A simplex method for function minimization', *The Computer Journal*, **7**(4), 308–313, URL <http://comjnl.oxfordjournals.org/cgi/content/abstract/7/4/308>.
- Neumaier, A. (2004), 'Complete search in continuous global optimization and constraint satisfaction', em A. Iserles (Editor), *ACTA NUMERICA 2004*, Cambridge University Press, Volume 13, Páginas 271–369, URL <http://www.mat.univie.ac.at/~neum/papers.html#g1opt03>.
- Okasinski, M. J. e Doherty, M. F. (1998), 'Design method for kinetically controlled, staged reactive distillation columns', *Industrial & Engineering Chemistry Research*, **37**(7), 2821–2834.
- O'Reilly, T. (2005), 'What is web 2.0 - design patterns and business models for the next generation of software', URL <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>.
- Ostrovsky, G. M., Ostrovsky, M. G. e Mikhailow, G. W. (1990), 'Discrete optimization of chemical processes', *Computers & Chemical Engineering*, **14**(1), 111–117.

- Ousterhout, J. K. (1993), *Tcl and Tk Toolkit*, Addison-Wesley Publishing Company, Inc.
- Ousterhout, J. K. (1998), 'Scripting: Higher level programming for the 21st century', *IEEE Computer Magazine*, **31**(3), 23 – 30.
- Ousterhout, J. K. (2008), 'History of Tcl', World Wide Web Publication, URL <http://home.pacbell.net/ouster/tclHistory.html>.
- Papamichail, I. e Adjiman, C. S. (2004), 'Global optimization of dynamic systems', *Computers & Chemical Engineering*, **28**(3), 403 – 415.
- Paules, G. E. e Floudas, C. A. (1989), 'Apros - algorithmic development methodology for discrete-continuous optimization problems', *Operations Research*, **37**(6), 902–915.
- Perry, J. L. e Allan, B. A. (1996), *Design and Use of Dynamic Modeling in ASCEND IV*, EDRC Technical Report 06-224-96, Carnegie Mellon University, Pittsburgh, PA, USA.
- Piela, P. (1989), *ASCEND: An Object-Oriented Computer Environment for Modeling and Analysis*, Tese de Doutorado, Carnegie Mellon University, ph.D. Thesis, Carnegie Mellon University, available as EDRC 02-09-89 from the Engineering Design Research Center, Carnegie Mellon University.
- Piela, P. C., Epperly, T. G., Westerberg, K. M. e Westerberg, A. W. (1991), 'ASCEND - an object-oriented computer environment for modeling and analysis - the modeling language', *Computers & Chemical Engineering*, **15**(1), 53 – 72.
- Pintér, J. D. (1996), 'Continuous global optimization software: A brief review', *OPTIMA - Mathematical Programming Society Newsletter*, (52), URL <http://www.ise.ufl.edu/~optima/>.
- Pintér, J. D. (2002), *Handbook of Global Optimization*, Kluwer Academic Press, Capítulo 15, Páginas 512–569.
- Ponton, J. (1995), 'Process systems engineering: Halfway through the first century', *Chemical Engineering Science*, **50**(24), 405–4059.
- Prechelt, L. e Informatik, F. F. (2002), 'Are scripting languages any good? a validation of perl, python, rexx, and tcl against c, c++, and java', .
- Press, W. H. e Teukolsky, S. A. (1991), 'Simulated annealing over continuous spaces', *Computers in Physics*, **5**(4), 426.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T. e Flannery, B. P. (1992), *Numerical Recipes in Fortran 77*, Cambridge University Press, Capítulo Root Finding and Nonlinear Sets of Equations, 1 Edição.
- Price, W. L. (1978), 'Global optimization by controlled random search', *Journal of Optimization Theory and Applications*, **40**, 333–348.

- Radhakrishnan, K. R. e Hindmarsh, A. C. (1993), *Description and Use of LSODE, the Livermore Solver for Ordinary Differential Equations*, LLNL Report UCRL-ID-113855, U.S. Department of Energy Lawrence Livermore National Laboratory, Lewis Research Center, Cleveland, OH, USA.
- Raman, R. e Grossmann, I. E. (1994), 'Modeling and computational techniques for logic-based integer programming', *Computers & Chemical Engineering*, **18**(7), 563–578.
- Rao, S. S. (1996), *Engineering Optimization - Theory and Practice*, Wiley Interscience - John Wiley & Sons, Inc.
- Ray, W. H. (1981), *Advanced process control*, McGraw-Hill, New York.
- Rechenberg, I. (1973), *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologisch Evolution*, Frommann-Holzboog Verlag.
- Rico-Ramirez, V. (1998), *Representation, Analysis and Solution of Conditional Models in an Equation-Based Environment*, Tese de Doutorado, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.
- Rodriguez-Toral, M. A., Morton, W. e Mitchell, D. R. (2000), 'Using new packages for modeling, equation oriented simulation and optimization of a cogeneration plant', *Computers & Chemical Engineering*, **24**(12), 2667 – 2685.
- Rückmann, J. e Alfredo Gómez, J. (2006), 'On generalized semi-infinite programming', *TOP*, **14**(1), 1–32, URL <http://dx.doi.org/10.1007/BF02578994>.
- Safrit, B. T. (1996), *Synthesis of Azetropic batch Distillation Separation Systems*, Tese de Doutorado, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.
- Sahinidis, N. (1996), 'BARON: A general purpose global optimization software package', *Journal of Global Optimization*, **8**, 201–105.
- Sahinidis, N. e Tawarmalani, M. (2008), *BARON Manual*, URL <http://www.gams.com/dd/docs/solvers/baron.pdf>, <http://www.gams.com/docs/document.htm>.
- Salcedo, R. (1992a), *MSGa - User's Manual*, Departamento de Engenharia Química - Faculdade de Engenharia da Universidade do Porto.
- Salcedo, R., Goncalves, M. J. e de Azevedo, S. F. (1990), 'An improved random-search algorithm for nonlinear optimization', *Computers & Chemical Engineering*, **14**(10), 1111–1126.
- Salcedo, R. L. (1992b), 'Solving Nonconvex Nonlinear-Programming and Mixed-Integer Nonlinear-Programming Problems with Adaptive Random Search', *Industrial & Engineering Chemistry Research*, **31**(1), 262 – 273.
- Salcedo, R. L. e Lima, R. M. (1999), 'On the optimum choice of decision variables for equation-oriented global optimization', *Industrial & Engineering Chemistry Research*, **38**(12), 4742 –

4758.

Salcedo, R. L., Lima, R. P. e Cardoso, M. F. (2003), 'Simulated annealing for the global optimization of chemical processes', *Proceedings of the Indian National Science Academy*, **69 A**(3 & 4), 359 – 401.

Salcedo, R. L. R. (2001), *Problemas de Optimizaç o N o-Linear (para resoluç o em computador)*, FEUP Ediç es, 1^a Ediç o.

Sargent, R. W. H. (2004), 'Introduction: 25 years of progress in process systems engineering', *Computers & Chemical Engineering*, **28**(4), 437–439.

Sargent, R. W. H. (2005), 'Process systems engineering: A retrospective view with questions for the future', *Computers & Chemical Engineering*, **29**(6), 1237–1241.

Schwefel, H. P. (1975), *Evolutionsstrategie und numerische Optimierung*, Tese de Doutorado, Technische Universitat Berlin.

Schweiger, C. A. e Floudas, C. A. . (1998), *MINOPT: A Modeling Language and Algorithmic Framework for Linear, Mixed-Integer, Nonlinear, Dynamic, and Mixed-Integer Nonlinear Optimization, Version 3.1, User's Manual*, Princeton University, Princeton, NJ 08544-5263, URL <http://titan.princeton.edu/MINOPT/minopt.html>.

Seader, J. D. (1985), 'Equilibrium-stage operations', *Chemical Engineering Education*, 88–103.

Seader, J. D. e Henley, E. S. (1998), *Separation process principles*, John Wiley & Sons, New York.

Seider, W. D., Brengel, D. D. e Widagdo, S. (1991), 'Nonlinear-analysis in process design', *AIChE Journal*, **37**(1), 1–38.

Shacham, M., Brauner, N. e Cutlip, M. B. (2002), 'A web-based library for testing performance of numerical software for solving nonlinear algebraic equations', *Computers & Chemical Engineering*, **26**(4-5), 547–554.

Shacham, M. e Kehat, E. (1976), 'A direct method for the solution of large sparse systems of linear equations', *The Computer Journal*, **19**(4), 353–359, URL <http://comjnl.oxfordjournals.org/cgi/content/abstract/19/4/353>.

Siddall, J. N. (1982), *Optimal Engineering Design*, Mechanical Engineering, Marcel Dekker, Inc.

Siirola, J. D. e Huan, S. (2007), 'Polymorphic optimization', *Computers & Chemical Engineering*, **31**(10), 1312–1325.

Siirola, J. D., Huan, S. e Westerberg, A. W. (2003), 'Toward agent-based process systems engineering: proposed framework and application to non-convex optimization', *Computers & Chemical Engineering*, **27**(12), 1801–1811.

- Silva, H. G. e Salcedo, R. L. (2005), 'Stochastic algorithms and ASCEND IV: a coupled equation-oriented strategy for non-linear optimization', em D. E. Q. U. Coimbra (Editor), *CHEMPOR 2005 - 9th International Chemical Engineering Conference*, Dep. Eng. Química. - Univ. Coimbra, Página 250.
- Silva, H. G. e Salcedo, R. L. R. (2006), 'Modeling and optimization of chemical processes: ASCEND IV and stochastic optimizers', em R. Wamkeue (Editor), *Proceedings of the 17th IASTED International Conference on MODELING AND SIMULATION*, IASTED - International Association of Science and Technology for Development, ACTA Press, Páginas 493–498.
- Silva, H. G. e Salcedo, R. L. R. (2009), 'A coupled strategy for the solution of nlp and minlp optimisation problems: benefits and pitfalls', *Industrial & Engineering Chemistry Research*, in press.
- Silver, E. A. (2004), 'An overview of heuristic solution methods', *Journal of the Operational Research Society*, **55**(9), 936–956.
- Smith, E. M. B. e Pantelides, C. (1997), 'Global optimisation of nonconvex minlps', *Computers & Chemical Engineering*, **21**, S791–S796.
- Spall, J. C. (2003), *Introduction to Stochastic Search and Optimization - Estimation, Simulation and Control*, Discrete Mathematics and Optimization, Wiley Interscience - John Wiley & Sons.
- Stadtherr, M. A. (1997), 'Large-scale process simulation and optimization in a high performance computing environment', em *AspenWorld 1997*.
- Steiner, J. (1841), 'Über maximum und minimum bei den figuren in der ebene, auf der kugeläche und in raume überhaupt', *C. R. Acad. Sci. Paris*, **12**, 177–308, les Comptes rendus de l'Académie des sciences.
- Steward, D. V. (1981), 'The design structure sytem: A method for managing the design of comples systems', *IEEE Transacticons on Engineering Management*, **EM-28**(3), 71–74.
- Surowiecki, J. (2005), *The Wisdom of Crowds*, Anchor.
- Tapscott, D. e Williams, A. D. (2008), *Wikinomics: How Mass Collaboration Changes Everything*, Penguin Group USA, URL <http://www.wikinomics.com/book/>.
- Tavares, L. V. e Correia, F. N. (1999), *Optimização Linear e Não Linear - Conceitos, Métodos e Algoritmos*, Fundação Calouste Gulbenkian.
- Tawarmalani, M. e Sahinidis, N. V. (2004), 'Global optimization of mixed-integer nonlinear programs: A theoretical and computational study', *Mathematical Programming*, **99**(3), 563–591.
- Ternet, D. J. e Biegler, L. T. (1998), 'Recent improvements to a multiplier-free reduced hessian

- successive quadratic programming algorithm', *Computers & Chemical Engineering*, **22**(7-8), 963–978.
- Thierauf, R. J. e Klekamp, R. C. (1975), *Decision Making Through Operations Research*, Wiley International.
- Thomas, M. (1996), *Tool and Information Management in Engineering Design*, Tese de Doutorado, Carnegie Mellon University, Pittsburgh, PA.
- Tolsma, J. E., Clabaugh, J. A. e Barton, P. I. (2002), 'Symbolic incorporation of external procedures into process modeling environments', *Industrial & Engineering Chemistry Research*, **41**(16), 3867–3876, URL <http://pubs.acs.org/doi/abs/10.1021/ie0107946>.
- Torn, A., Ali, M. M. e Viitanen, S. (1999), 'Stochastic global optimization: Problem classes and solution techniques', *Journal Of Global Optimization*, **14**(4), 437–447.
- Turkay, M. e Grossmann, I. E. (1996), 'Logic-based minlp algorithms for the optimal synthesis of process networks', *Computers & Chemical Engineering*, **20**(8), 959–978.
- Varaiya, P. (1972), *Notes on Optimization*, Van Nostrand Reinhold Company.
- Varma, V., Reklaitis, G. E., Blau, G. E. e Pekny, J. F. (2007), 'Enterprise-wide modeling & optimization-an overview of emerging research challenges and opportunities', *Computers & Chemical Engineering*, **31**, 692–711.
- Vecchietti, A. e Grossmann, I. E. (1999), 'Logmip: a disjunctive 0-1 non-linear optimizer for process system models', *Computers & Chemical Engineering*, **23**(4-5), 555–565.
- Vecchietti, A. e Grossmann, I. E. (2000), 'Modeling issues and implementation of language for disjunctive programming', *Computers & Chemical Engineering*, **24**(9-10), 2143–2155.
- Villermaux, J. (1996), 'Future prospects for chemical engineering research and technology', *Chemical Engineering Science*, 21–23.
- Viswanathan, J. e Grossmann, I. E. (1990), 'A combined penalty-function and outer-approximation method for minlp optimization', *Computers & Chemical Engineering*, **14**(7), 769–782.
- Wachter, A. e Biegler, L. T. (2006), 'On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming', *Mathematical Programming*, **106**(1), 25–57.
- Westerberg, A. (1979), *A modified least squares algorithm for solving sparse nxn sets of nonlinear equations*, Edrc technical report, Carnegie Mellon University.
- Westerberg, A., Allan, B., Rico-Ramirez, V., Thomas, M. e Tyner, K. (1998), *ASCEND IV - Advanced System for computations in ENgineering Design*, Department of Chemical Engineering, Carnegie Mellon University, URL <http://www-2.cs.cmu.edu/~ascendFTP/pdfhelp/>

ascend-help.pdf.

- Westerberg, A. W. (2004), 'A retrospective on design and process synthesis', *Computers & Chemical Engineering*, **28**(4), 447 – 458, URL <http://www.sciencedirect.com/science/article/B6TFT-4B1XP1W-2/2/89668861c0d94e0d2b3a4f4b6ca99618s>.
- Westerlund, T. e Pettersson, F. (1995), 'An extended cutting plane method for solving convex minlp problems', *Computers & Chemical Engineering*, **19**, S131–S136.
- Wilde, D. J. e Beightler, C. S. (1967), *Foundations of Optimization*, Prentice-Hall, Inc.
- Williams, P. (1988), *Model building in mathematical programming*, Wiley.
- Young, D. H. (1997), *The Visual Tcl Handbook*, Prentice Hall PTR.
- Zaher, J. J. (1995), *Conditional Modeling*, Tese de Doutorado, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.
- Zitney, S. E., Mallya, J., Davis, T. A. e Stadtherr, M. A. (1996), 'Multifrontal vs frontal techniques for chemical process simulation on supercomputers', *Computers & Chemical Engineering*, **20**(6-7), 641–646.

Bibliografia

- (2003), *The Chicago Manual of Style*, The University of Chicago Press, 15 Edição, URL www.chicagomanualofstyle.org.
- Cadete, M. O. R. (1996), *A linguagem FORTRAN77*, Fundação Calouste Gulbenkian, 2 Edição.
- Chapra, S. C. e Canale, R. P. (2002), *Numerical Methods for Engineers - With Software and Programming Applications*, McGraw-Hill, 4 Edição.
- Dams, L. M. D. (2002), *Linguagem C*, FCA Editores.
- Davis e Hoffmann (1983), *FORTRAN 77: A Structured, Disciplined Style*, International Student Editions, McGraw-Hill, 2 Edição.
- Estrela, E., Soares, M. A. e Leitão, M. J. (2003), *Saber Escrever Saber Falar*, Dom Quixote.
- Estrela, E., Soares, M. A. e Leitão, M. J. (2006), *Saber Escrever Uma Tese e Outros Textos*, Dom Quixote.
- Goossens, M., Mittelbach, F., Rahtz, S., Roegel, D. e Voß, H. (2007), *The L^AT_EX Graphics Companion*, Tools and Techniques for Computer Typesetting, Addison-Wesley, Boston, Massachusetts, 2 Edição.
- Loukides, M. (1990), *UNIX for FORTRAN Programmers*, O'Reilly & Associates, Inc.
- Loukides, M. e Oram, A. (1997), *Programming with GNU Software*, O'Reilly.
- Mittelbach, F. e Goossens, M. (2004), *The L^AT_EX Companion*, Tools and Techniques for Computer Typesetting, Addison-Wesley, Boston, Massachusetts, 2 Edição, with Johannes Braams, David Carlisle, and Chris Rowley.
- Mould, C. (1991), *O Sistema Operativo UNIX*, Presença.
- Oliveira, N. M. C. (2006), *Prática de Computação*, Imprensa da Universidade de Coimbra.
- Pereira, F. (2003), *Linux - Curso Completo*, FCA, 4 Edição.
- Raines, P. (1998), *Tcl/Tk Pocket Reference*, O'Reilly, 1 Edição.

- Raymond, E. S. (2004), *The Art of UNIX Programming*, Professional Computing Series, Addison-Wesley.
- SDI - Serviços de Documentação e Informação (FEUP) (2008), 'Guia de apoio à publicação', Documento electrónico., URL <http://paginas.fe.up.pt/~seagull/wiki/index.php>.
- Smirnov, G. e Bushenkov, V. (2005), *Curso de Optimização - Programação Matemática, Cálculo de Variações, Controlo Óptimo*, Escolar Editora.
- Smith, J. M., Van Ness, H. C. e Abbot, M. M. (1996), *Introduction to Chemical Engineering Thermodynamics*, McGraw-Hill International Editions - Chemical Engineering Series, 5 Edição.
- Wainwright, P. (2001), *Perl Programming*, Wrox Press Ltd, 1 Edição.
- Walas, S. M. (1990), *Chemical Process Equipment - Selection and Design*, Butterworth Heinemann.
- Welch, B. B. e Jones, K. (2003), *Practical Programming in Tcl and Tk*, Prentice Hall PTR, 4 Edição.

Apêndices

Apêndice A

Casos de estudo

Apresentam-se as formulações, modelos e informações suplementares relativamente aos casos de estudo focados na tese. São igualmente indicadas referências bibliográficas relacionadas com a generalidade dos exemplos.

A.1 P.RSIA01

Problema genérico, simples, utilizado como objecto de testes preliminares na avaliação da eficácia da ligação entre simulador e optimizador no contexto da estratégia integrada apresentada (Salcedo, 2001).

Equações

$$\min f = -12x_1 - 7x_2 + x_2^2 \quad (\text{A.1})$$

$$\text{sujeito a: } -2x_1^4 - x_2 - 2 = 0 \quad (\text{A.2})$$

$$0 \leq x_1 \leq 2 \quad (\text{A.3})$$

$$0 \leq x_2 \leq 2 \quad (\text{A.4})$$

Modelo ASCEND

```

1: (*=====*)
2:                               M O D E L . R S _ I A 0 1
3: *=====*)
4:   REQUIRE "atoms.a41";
5:   MODEL Ia_1;
6:
7:   x1, x2, fob IS_A generic_real;
8:
9:   eq_01: -2*x1^4 - x2 + 2 = 0;
10:  eq_fob: fob = -12*x1 -7*x2 + x2^2;
11:
12:  METHODS
13:    METHOD specify;
14:      x1.fixed := TRUE;
15:    END specify;
16:
17:    METHOD bound_self;
18:      X2.upper_bound := 3;
19:      X2.lower_bound := 0;
20:    END bound_self;
21:
22:    METHOD default_self;
23:    END default_self;
24:  END METHODS;
25:  END Ia_1;

```

Função funk

```

1:   double precision function funk(v)

```

```
2:      implicit double precision (a-h,o-z)
3:      integer ncalc
4:      double precision v(100)
5:      common/no_calc/ncalc
6:      save
7: c -----
8: c  Passagem das variáveis de decisão para o stdout.
9: c  'call flush(6)' garante o envio de toda a informação
10: c  colocada no canal com a instrução write.
11: c -----
12:      write (*,*) v(1)
13:      call flush(6)
14:
15:      read (*,*) funk
16: c -----
17:      return
18:      end
```

Ficheiro opt.script

```
1: #=====
2: #           O P T . S C R I P T . R S _ I A O 1 . A 4 S
3: #=====
4:
5: # Construção da simulação
6: DELETE TYPES;
7: READ FILE "mymodels/rsalcedo.Ia.1/model.rs_ia.a4c";
8: COMPILE sim OF Ia_1;
9: # BROWSE {sim};
10:
11: # Procedimento RUN SIMULATION
12: proc run_sim {var01} {
13:   global fob_val
14:   RUN {sim.reset}
15:   ASSIGN {sim.x1} $var01 {}
16:   SOLVE {sim} WITH QRS1v
17:   set fob_val [u_getval sim.fob]
18: }
19:
20: # Procedimento FOB SEND
21: proc fob_send {} {
22:   global fob_val
23:   send main.opt [list fob_get $fob_val]
24: }
```

Ficheiro main.opt

```
1: #! /usr/bin/wish
2: # (*=====*)
3: #           M A I N . O P T - P . R S I A O 1
4: # (*=====*)
5:
6: #  Variaveis globais
7:   global fob x2
8:   global msimpsa
```

```

9: #   Instrucao para manter o simulador pendente no canal, que
10: #   continuara atento enquanto existir actividade do optimizador
11:   global forever
12:   set record [open records.dat w+]
13: #   Abertura do canal para o optimizador
14:   set msimpsa [open "|main" "w+"]
15:   fconfigure $msimpsa -buffering none
16: #   Carreamento procedimentos no ASCEND
17:   send ascend4 [list source /home/hgsilva/ascddata/mymodels/
                    rsalcedo.Ia.1/opt.script.rs_ia01.a4s]

18:
19: #   Abertura do canal
20: fileevent $msimpsa readable {
21:     # Obter variaveis de decisao desde o MSIMPSPA
22:     # vlist - lista com todas as variaveis
23:     # v?? - designacoes individuais
24:     gets $::msimpsa v01
25:     set flag01 [llength $v01]
26:     if {$flag01 == 0} {
27:         exit
28:     }
29:     # Simulacao e leitura da funcao objectivo
30:     send ascend4 [list run_sim $v01]
31:
32:     proc fob_get {fob x2} {
33:         if { [eof $::msimpsa] } {
34:             close $::msimpsa
35:             set ::forever 1
36:         }
37:         puts $::msimpsa $fob
38:         puts $::msimpsa $x2
39:     }
40:     send ascend4 [list fob_send]
41: }
42:
43: vwait forever

```

Alguns resultados

O problema é resolvido estabelecendo x_1 como variável de decisão, tendo-se obtido como melhor solução $\{x_1; f\} = \{0.7158; -16.74\}$, que é concordante com o melhor resultado obtido reportado na literatura (Salcedo, 2001).

A.2 P.HRCV

Trata-se da análise da equação de projecto de um permutador localizado junto à base de uma coluna de destilação. O exemplo serve para testar a eficiência da opção *safe calculations* que o ASCEND disponibiliza. Tal funcionalidade é essencial para que a ocorrência de incongruências matemáticas não condicione a avaliação de função objectivo, o que prejudicaria o processo de comunicação com o optimizador. O objectivo é maximizar a recuperação de energia pelo cruzamento das correntes F_1 e F_2 .

Diagrama Processual

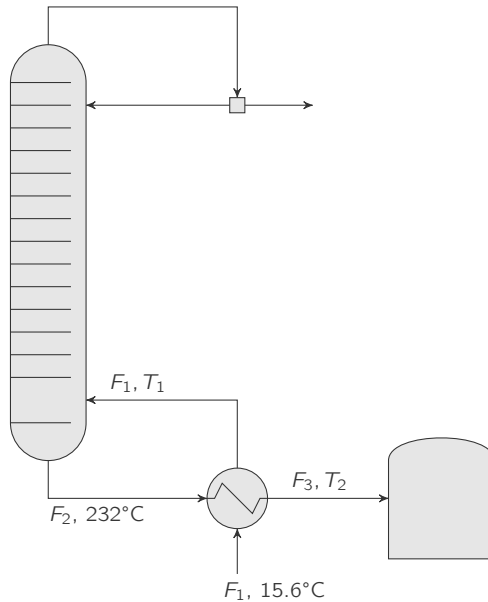


Figura A.1 Diagrama processual do problema P.HRCV.

Equações

$$Q = U A \Delta T_{In} \quad (\text{A.5})$$

$$Q = F_1 C_P (T_1 - 288.75) \quad (\text{A.6})$$

$$Q = F_2 C_P (505.15 - T_2) \quad (\text{A.7})$$

$$\Delta T_{ln} = \frac{(505.15 - T_1) - (T_2 - 288.75)}{\ln\left(\frac{505.15 - T_1}{T_2 - 288.75}\right)} \quad (\text{A.8})$$

Modelo ASCEND

```

1: (*=====*)
2:                               M O D E L . H E A T _ R E C O V E R
3: (*=====*)
4:   REQUIRE "myatoms.a41";
5:   MODEL heat_recover;
6:
7: (*===== D A T A =====*)
8:   F1, F2, U, CP IS_A real_constant;
9:
10:  F1 := 272760 {kg/hr};
11:  F2 := 136380 {kg/hr};
12:  U  := 1.23e6 {J/hr/m^2/K};
13:  CP := 2092 {J/kg/K};
14:
15: (*===== V A R I A B L E S =====*)
16:  Q      IS_A energy_rate;
17:  T1, T2 IS_A temperature;
18:  A      IS_A area;
19:  DTln   IS_A dif_temperature;
20:
21: (*===== E Q U A T I O N S =====*)
22:  eq01: Q = U * A * DTln;
23:  eq02: Q = F1 * CP * (T1 - 288.75 *1{K});
24:  eq03: Q = F2 * CP * (505.15*1{K} - T2);
25:  eq04: DTln = ((505.15*1{K}-T1)-(T2-288.75*1{K}))/
26:              ln((505.15*1{K}-T1)/(T2-288.75*1{K}));
27:
28: (*===== O B J E C T I V E   F U N C T I O N =====*)
29:  FOB IS_A factor;
30:  eqFOB: FOB = 1.66e-6*(Q*1{hr/J}) - 153.675*(A*1{1/m^2})^0.8;
31:
32: (*===== M E T H O D S =====*)
33:  METHODS
34:    METHOD specify;
35:      T2.fixed := TRUE;
36:    END specify;
37:
38:    METHOD values;
39:    END values;
40:
41:    METHOD default_self;
42:    END default_self;
43:
44:    METHOD bound_self;
45:    END bound_self;
46:  END METHODS;
47: END heat_recover;

```

Alguns resultados

A formulação apresenta apenas um grau de liberdade (4 equações e 5 variáveis), tendo sido escolhida a variável T_2 para variável de decisão. O melhor resultado obtido corresponde ao conjunto $\{T_2, F_{obj}\} = \{317.7 \text{ K}; 60810.5 \text{ \$/ano}\}$

Os resultados para um estudo do modelo, segundo estrutura similar à apresentada no Quadro 3.4 para o exemplo P.MK na Secção 3.4.5, são apresentados na Figura A.2. Nela representa-se a evolução da F_{obj} em função da variável do valor tomado pela variável de decisão T_2 .

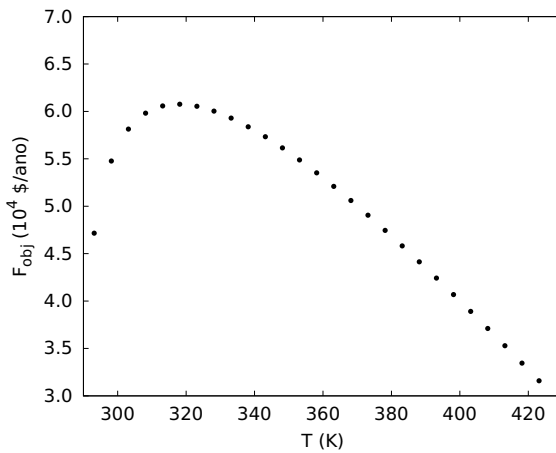


Figura A.2 Problema P.HRCV. Resultados para o estudo F_{obj} vs T .

A.3 P. ABSTRIP

Processo químico onde ocorre a extracção e recuperação de solvente. Como se pode verificar o sistema é constituído por uma coluna de absorção, outra de desorção, quatro permutadores (P_i , $i = 1...4$) e duas bombas (B_1 e B_2). Na referência onde o processo se encontra extensiva e convenientemente explicado (Ferreira e Salcedo, 2001a,b) são abordados dois casos cujas diferenças residem nas restrições que condicionam a resolução do problema. Resultados adicionais, com simulações em FORTRAN, podem ser encontrados em Salcedo et al. (2003); Salcedo (2001).

Diagrama Processual

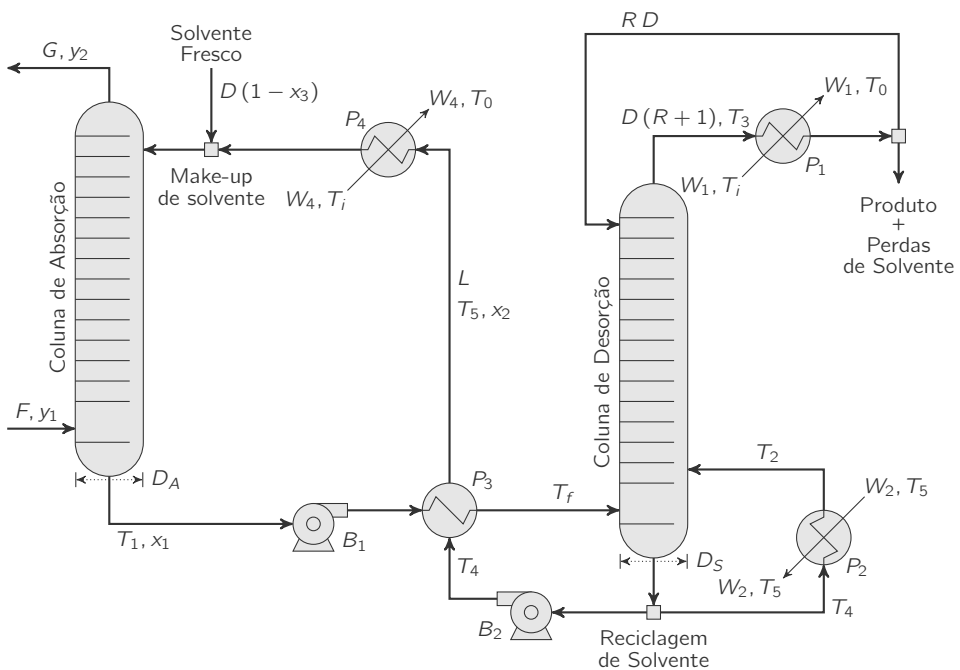


Figura A.3 Diagrama processual do problema P. ABSTRIP

Equações

$$F - G - D x_3 = 0 \quad (\text{A.9})$$

$$F y_1 + L x_2 - (L + D) x_1 - G y_2 = 0 \quad (\text{A.10})$$

$$N_{og} - \frac{1}{1 - \frac{HG}{PL^*}} \ln \left[\left(1 - \frac{HG}{PL^*} \right) \left(\frac{y_1 - \frac{H}{P} x_2^*}{y_2 - \frac{H}{P} x_2^*} \right) + \frac{HG}{PL^*} \right] = 0 \quad (\text{A.11})$$

$$H_{og} - H_g - \frac{HG H_I}{PL^*} = 0 \quad (\text{A.12})$$

$$H_g - \alpha \left(\frac{F M_g}{A_A} \right)^\beta \sqrt{\frac{\mu_g}{\rho_g D_g}} \left(\frac{(L + D) M_I}{A_A} \right)^\gamma = 0 \quad (\text{A.13})$$

$$H_I - \left(\frac{(L + D) M_I}{A_A \mu_I} \right)^\delta \phi \sqrt{\frac{\mu_I}{\rho_I D_I}} = 0 \quad (\text{A.14})$$

$$Z - N_{og} H_{og} = 0 \quad (\text{A.15})$$

$$A_A - \pi \frac{D_a^2}{4} = 0 \quad (\text{A.16})$$

$$D_a - \sqrt{\frac{4 G M_g}{0.75 \pi G_f}} = 0 \quad (\text{A.17})$$

$$\log \left[G_f^2 \left(\frac{a_p}{\epsilon^3} \right) \left(\frac{1}{g \rho_g \rho_I} \right) \left(\frac{\mu_I}{\mu_g} \right)^{0.2} \right] - 1.74 \left(\frac{L^* M_I}{G M_g} \right)^{0.25} \left(\frac{\rho_I}{\rho_g} \right)^{-0.125} = 0 \quad (\text{A.18})$$

- Bomba n^o 1

$$H P_1 - K_{P_1} (L + D) N = 0 \quad (\text{A.19})$$

- Bomba nº 2

$$HP_2 - K_{P_2} L Z = 0 \quad (\text{A.20})$$

- Permutador de calor nº 4

$$Q_4 - L c_p (T_5 - T_1) = 0 \quad (\text{A.21})$$

$$Q_4 - W_4 c_{pw} (T_0 - T_i) = 0 \quad (\text{A.22})$$

$$Q_4 - U_4 A_4 \Delta T_{In_4} = 0 \quad (\text{A.23})$$

$$\Delta T_{In_4} - \frac{(T_5 - T_0) - (T_1 - T_i)}{\ln \left(\frac{T_5 - T_0}{T_1 - T_i} \right)} = 0 \quad (\text{A.24})$$

- Permutador de calor nº 3

$$Q_3 - (L + D) c_p (T_f - T_1) = 0 \quad (\text{A.25})$$

$$Q_3 - L c_p (T_4 - T_5) = 0 \quad (\text{A.26})$$

$$Q_3 - U_3 A_3 \Delta T_{In_3} = 0 \quad (\text{A.27})$$

$$\Delta T_{In_3} - \frac{(T_1 - T_5) - (T_f - T_4)}{\ln \left(\frac{T_1 - T_5}{T_f - T_4} \right)} = 0 \quad (\text{A.28})$$

- Stripper e permutadores de calor

$$(L + D) x_1 - L x_2 - D x_3 = 0 \quad (\text{A.29})$$

$$(L + D) c_p T_f + Q_2 - D c_p T_3 - L c_p T_4 - Q_1 = 0 \quad (\text{A.30})$$

$$\pi \frac{D_s^2}{4} - \frac{22.4 Q_2 T_4}{273.2 \cdot 3600 \lambda M_l K \sqrt{\frac{(\rho_l - \rho_g)}{\rho_g}}} = 0 \quad (\text{A.31})$$

$$R_m - \left[\frac{1}{a-1} \right] \left[\frac{x_3}{x_1} - a \frac{(1-x_3)}{(1-x_1)} \right] = 0 \quad (\text{A.32})$$

$$N_m - \frac{\ln \left[\frac{x_3}{(1-x_3)} \frac{(1-x_2)}{x_2} \right]}{\ln(a)} = 0 \quad (\text{A.33})$$

$$X - \frac{(N - N_m)}{(1 + N)} = 0 \quad (\text{A.34})$$

$$X - 0.75 \left[1 - \left(\frac{R - R_m}{R + 1} \right)^{0.5668} \right] = 0 \quad (\text{A.35})$$

$$Q_1 - D \lambda M_l (R + 1) = 0 \quad (\text{A.36})$$

$$Q_1 - W_1 c_{pw} (T_0 - T_i) = 0 \quad (\text{A.37})$$

$$Q_1 - U_1 A_1 \Delta T_{ln1} = 0 \quad (\text{A.38})$$

$$\Delta T_{ln1} - \frac{(T_3 - T_0) - (T_3 - T_i)}{\ln \left(\frac{T_3 - T_0}{T_3 - T_i} \right)} = 0 \quad (\text{A.39})$$

$$Q_2 - W_2 \lambda_w = 0 \quad (\text{A.40})$$

$$Q_2 - U_2 A_2 \Delta T_{ln2} = 0 \quad (\text{A.41})$$

$$\Delta T_{ln2} - (T_s - T_4) = 0 \quad (\text{A.42})$$

$$x_2 \exp\left(A'_{1,1} - \frac{A'_{2,1}}{(T_4 + T_{ref})}\right) + (1 - x_2) \exp\left(A'_{1,2} - \frac{A'_{2,2}}{(T_4 + T_{ref})}\right) - P = 0 \quad (\text{A.43})$$

$$\frac{P x_3}{\exp\left(A'_{1,1} - \frac{A'_{2,1}}{(T_3 + T_{ref})}\right)} + \frac{P(1 - x_3)}{\exp\left(A'_{1,2} - \frac{A'_{2,2}}{(T_3 + T_{ref})}\right)} - 1 = 0 \quad (\text{A.44})$$

$$T_f - T_2 - (q - 1) \frac{\lambda}{C_p} = 0 \quad (\text{A.45})$$

$$x_1 \exp\left(A'_{1,1} - \frac{A'_{2,1}}{(T_2 + T_{ref})}\right) + (1 - x_1) \exp\left(A'_{1,2} - \frac{A'_{2,2}}{(T_2 + T_{ref})}\right) - P = 0 \quad (\text{A.46})$$

$$x_2^* - \frac{L x_2}{L + D(1 - x_3)} = 0 \quad (\text{A.47})$$

$$L^* - D(1 - x_3) - L = 0 \quad (\text{A.48})$$

- Cálculo da função objectivo

$$\begin{aligned} Inv = & C_z Z D_a^{1,0} + C_n N D_s^{1,085} + \dots \\ & + C_a (A_1^{0,556} + A_2^{0,556} + A_3^{0,556} + A_4^{0,556}) + C_{hp} (HP_1^{0,3} + HP_2^{0,3}) \end{aligned} \quad (\text{A.49})$$

$$\begin{aligned} F_{obj} = & P_P D M_l x_3 - C_g F M_g - C_l D(1 - x_3) M_l - \dots \\ & - C_w (W_1 + W_4) - C_s W_2 - C_e (HP_1 + HP_2) - Inv F_c \end{aligned} \quad (\text{A.50})$$

Modelo ASCEND

```

1: (*=====*)
2:
3:           M O D E L . A B _ S T R I P P E R
4:
5: *=====*)
6:
7: REQUIRE "myatoms.a41";
8:
9: (*===== D A T A =====*)
10:
11: MODEL ab_stripper;
12:
13: F, P, y1, x3, T1, Ti, To, Ts, Tref, ro_liq, ro_gas, M_liq, M_gas,
14: cpw, cp, lambda, lambda_w, miu_liq, miu_gas, dif_liq, dif_gas,
15: antA1, antA2, antB1, antB2, alfa, beta, gama, delta, fi, K, sp_area,
16: a, U1, U2 , U3, U4, H, kp1, kp2
17:
18: IS_A real_constant;
19:
20: F      := 60 {kmol/hr};
21: P      := 1 {atm};
22: y1     := 0.4 {};
23: x3     := 0.99 {};
24: T1     := 303 {K};
25: Ti     := 293 {K};
26: To     := 298 {K};
27: Ts     := 393 {K};
28: Tref   := 230 {K};
29: ro_liq := 1500 {kg/m^3};
30: ro_gas := 5 {kg/m^3};
31: M_liq  := 154 {g/mol};
32: M_gas  := 58 {g/mol};
33: cpw    := 1 {kcal/kg/K};
34: cp     := 0.2 {kcal/kg/K};
35: lambda := 50 {kcal/kg};
36: lambda_w := 500 {kcal/kg};
37: miu_liq := 1.00e-3 {kg/m/s});
38: miu_gas := 1.85e-5 {kg/m/s});
39: dif_liq := 2.0e-9 {m^2/s};
40: dif_gas := 2.2e-5 {m^2/s};
41: antA1  := 32.9 {};
42: antA2  := 30.4 {};
43: antB1  := 14300 {K};
44: antB2  := 13800 {K};
45: alfa   := 0.557 {};
46: beta   := 0.32 {};
47: gama   := 0.51 {};
48: delta  := 0.22 {};
49: fi     := 0.00235 {};
50: K      := 0.1 {m/s};
51: sp_area := 490 {m^2/m^3};
52: a      := 2 {};
53: U1     := 300 {kcal/m^2/hr/K});
54: U2     := 500 {kcal/m^2/hr/K});
55: U3     := 100 {kcal/m^2/hr/K});

```

```

56: U4      := 200 {kcal/m^2/hr/K};
57: H       := 608 {mmHg};
58: kp1     := 0.10 {hp*hr/kmol};
59: kp2     := 0.05 {hp*hr/m/kmol};
60:
61: (*===== V A R I A B L E S =====)
62:
63: N, Nm, q, Nog      IS_A factor;
64: W1, W2, W4        IS_A mass_rate;
65: T2, T3, T4, T5, Tf IS_A temperature;
66: DTln1, DTln2, DTln3, DTln4 IS_A dif_temperature;
67: A1, A2, A4, Aa     IS_A area;
68: A3                IS_A area;
69: Q1, Q2, Q3, Q4    IS_A energy_rate;
70: x1, x2, y2, X, x2_ast IS_A mole_fraction;
71: Rm, R             IS_A factor;
72: D, G, L, L_ast    IS_A molar_rate;
73: Ds, Da, Hg, Hl, Hog, Z IS_A distance;
74: HP1, HP2         IS_A energy_rate;
75: Gf                IS_A factor_Gf;
76:
77: (*===== E Q U A T I O N S =====)
78:
79: (*== A B S O R B E R ==*)
80:
81: f01: F - G - D*x3 = 0*1{mol/hr};
82:
83: f02: F*y1 + L*x2 - (L+D)*x1 - G*y2 = 0*1{mol/hr};
84:
85: f03: Nog - (1/(1-H*G/(P*L_ast))) * ln( (1-H*G/(P*L_ast)) *
86:      ((y1-H*x2_ast/P)/(y2-H*x2_ast/P)) + H*G/(P*L_ast) ) = 0;
87:
88: f04: Hog - Hg - H*G*Hl/(P*L_ast) = 0*1{m};
89:
90: f05: Hg - alfa*1{m}*((F/1{kmol/hr})*(M_gas/1{g/mol}))(Aa/1{m^2}))^beta*
91:      (miu_gas/(ro_gas*dif_gas))^0.5/(((L+D)/1{kmol/hr})*(M_liq/1{g/mol}))/
92:      (Aa/1{m^2}))^gama = 0*1{m};
93:
94: f06: Hl - 1{m}*(((L+D)/1{kmol/hr}) * (M_liq/1{g/mol})) /
95:      ((Aa/1{m^2})*(miu_liq/1{kg/m/s}*3600))^delta * fi * (miu_liq/
96:      (ro_liq*dif_liq))^0.5 = 0*1{m};
97:
98: f07: Z - Nog*Hog = 0*1{m};
99:
100: f08: Aa - 1{PI}*Da^2/4 = 0*1{m^2};
101:
102: f09: Da - (4*G*M_gas/(0.75*1.0{PI}*Gf))^0.5 = 0*1{m};
103:
104: f10: log( Gf^2*sp_area*(1/(1{EARTH_G}*ro_gas*ro_liq))*(miu_liq/miu_gas)^0.2)
105:      -1.74*(L_ast*M_liq/(G*M_gas))^0.25*(ro_liq/ro_gas)^(-0.125) = 0;
106:
107: (*== P U M P 1 ==*)
108:
109: f11: HP1 - kp1*(L+D)*N = 0*1{hp};
110:
111: (*== P U M P 2 ==*)

```

```

112:
113: f12: HP2 - kp2*L*Z = 0*1{hp};
114:
115: (=== H E A T   E X C H A N G E R   4 ===)
116:
117: f13: Q4 - L*cp*M_liq*(T5-T1) = 0*1{kcal/hr};
118:
119: f14: Q4 - W4*cpw*(To-Ti) = 0*1{kcal/hr};
120:
121: f15: Q4 - U4*A4*DTln4 = 0*1{kcal/hr};
122:
123: f16: DTln4 - ((T5-To)-(T1-Ti))/ln((T5-To)/(T1-Ti)) = 0*1{K};
124:
125: (=== H E A T   E X C H A N G E R   5 ===)
126:
127: f17: Q3 - (L+D)*cp*M_liq*(Tf-T1) = 0*1{kcal/hr};
128:
129: f18: Q3 - L*cp*M_liq*(T4-T5) = 0*1{kcal/hr};
130:
131: f19: Q3 - U3*A3*DTln3 = 0*1{kcal/hr};
132:
133: f20: DTln3 - ((T5-T1)-(T4-Tf))/ln((T5-T1)/(T4-Tf)) = 0*1{K};
134:
135: (=== S T R I P P E R   A N D   E X C H A N G E R S ===)
136:
137: f21: (L+D)*x1 - L*x2 - D*x3 = 0*1{mol/hr};
138:
139: f22: (L+D)*cp*M_liq*Tf + Q2 - D*cp*M_liq*T3 - L*cp*M_liq*T4 - Q1 = 0*1{kcal/hr};
140:
141: f23: 1.0{PI}*Ds^2/4 - 22.4*(Q2/1{kcal/hr})*(T4/1{K})*1{m^2}/ (273.2*3600*
142: (lambda/1{kcal/kg})*(M_liq/1{g/mole})*(K/1{m/s})*( ro_liq-ro_gas)/
143: ro_gas)^0.5) = 0*1{m^2};
144:
145: f24: Rm - 1/(a-1)*(x3/x1-a*(1-x3)/(1-x1)) = 0;
146:
147: f25: Nm - ln(x3*(1-x2)/((1-x3)*x2))/ln(a) = 0;
148:
149: f26: X - (N-Nm)/(1+N) = 0;
150:
151: f27: R - (Rm + (1- X/0.75)^(1/0.5668))/(1-(1- X/0.75)^(1/0.5668)) = 0;
152:
153: f28: Q1 - D*lambda*M_liq*(R+1) = 0*1{kcal/hr};
154:
155: f29: Q1 - W1*cpw*(To-Ti) = 0*1{kcal/hr};
156:
157: f30: Q1 - U1*A1*DTln1 = 0*1{kcal/hr};
158:
159: f31: DTln1 - ((T3-To)-(T3-Ti))/ln((T3-To)/(T3-Ti)) = 0*1{K};
160:
161: f32: Q2 - W2*lambda_w = 0*1{kcal/hr};
162:
163: f33: Q2 - U2*A2*DTln2 = 0*1{kcal/hr};
164:
165: f34: DTln2 - (Ts-T4) = 0*1{K};
166:
167: f35: x2*exp(antA1-antB1/(T4+Tref)) + (1-x2)*exp(antA2-antB2/(T4+Tref))

```

```

168:      -P/1{mmHg} = 0;
169:
170:      f36: (P/1{mmHg})*x3/exp(antA1-antB1/(T3+Tref)) + (P/1{mmHg})*(1-x3)/
171:      exp(antA2-antB2/(T3+Tref)) - 1 = 0;
172:
173:      f37: Tf - T2 - (q-1)*lambda/cp = 0*1{K};
174:
175:      f38: x1*exp(antA1 - antB1/(T2+Tref)) + (1-x1)*exp(antA2-antB2/(T2+Tref))
176:      -P/1{mmHg} = 0;
177:
178:      (*** M A K E - U P ***)
179:
180:      f39: x2_ast - L*x2/(L+D*(1-x3)) = 0;
181:
182:      f40: L_ast - D*(1-x3) - L = 0*1{mol/hr};
183:
184:
185:      (*****O B J E C T I V E   F U N C T I O N ******)
186:
187:      (* ******)
188:      Some of the parameters present units that ASCEND cannot handle, like
189:      meter^(non integer). So, all the parameters are declared as factors at
190:      the same time that the variables are written in a dimensionless form
191:      within the equations
192:      ******)
193:
194:      Cg, Pp, Cw, Cs, Cl, Ce, Ca, Chp, Cz, Cn, Fc IS_A real_constant;
195:
196:      Cg  := 50 {};
197:      Pp  := 20 {};
198:      Cw  := 0.0635 {};
199:      Cs  := 35.2 {};
200:      Cl  := 5 {};
201:      Ce  := 89.7 {};
202:      Ca  := 350 {};
203:      Chp := 1000 {};
204:      Cz  := 600 {};
205:      Cn  := 363 {};
206:      Fc  := 0.14286 {};
207:
208:      FOB, Inv IS_A factor;
209:
210:      f41: FOB = Pp * (D/1{mol/hr}) * (M_liq/1{kg/mol}) * x3 - ( Cg * (F/1{mol/hr}) *
211:      (M_gas/1{kg/mol}) + Cl * (D/1{mol/hr}) * (1-x3) * (M_liq/1{kg/mol}) +
212:      Cw* ( (W1/1{kg/hr}) + (W4/1{kg/hr}) ) + Cs * (W2/1{kg/hr}) + Ce * ((HP1/1{hp})
213:      + (HP2/1{hp}) ) + Inv * Fc );
214:
215:      f42: Inv = Cz*(Z/1{m})*(Da/1{m})^1.0 + Cn*N*(Ds/1{m})^1.085 + Ca*(
216:      (A1/1{m^2})^0.556 +(A2/1{m^2})^0.556 + (A3/1{m^2})^0.556 + (A4/1{m^2})^0.556 )
217:      +Chp*( (HP1/1{hp})^0.3 +(HP2/1{hp})^0.3 );
218:
219:      (***** F L A G -- C O N S T R A I N T S   E V A L U A T I O N ******)
220:
221:      flag IS_A factor;
222:
223:      eqflag: flag = 1;

```

```
224:
225: (*===== S P E C I F Y   M E T H O D =====*)
226:
227: METHODS
228:   METHOD specify;
229:     N.fixed := TRUE;
230:     W4.fixed := TRUE;
231:     T4.fixed := TRUE;
232:     T2.fixed := TRUE;
233:     A1.fixed := TRUE;
234:   END specify;
235:
236: (*===== V A L U E S   M E T H O D =====*)
237:
238:   METHOD values;
239:     FOB := 1.0001+38;
240:     R := 1.0001;
241:   END values;
242:
243: (*===== M E T H O D S   A N D   C O N S T R A I T S =====*)
244:
245:   METHOD default_self;
246:     FOB.nominal := 1.0001e+38;
247:     R.nominal := 1.0001;
248:   END default_self;
249:
250:   METHOD bound_self;
251:     L.upper_bound := 300 {kmol/hr};
252:     R.upper_bound := 20;
253:     HP1.lower_bound := 0 {kg*m^2/hr^3};
254:     HP2.lower_bound := 0 {kg*m^2/hr^3};
255:     Nm.lower_bound := 0;
256:     Nog.lower_bound := 0;
257:     Rm.lower_bound := 0;
258:   END bound_self;
259:
260: (*===== C H E C K   S E L F =====*)
261:
262: (* Setting criteria for penalization of the FOB value
263:   when it occurs a violation of the supplementary constraints *)
264:
265:   METHOD check_self;
266:
267:   (== check bounds for the efficiency parameters ==)
268:
269:   IF ((1 - (G*y2)/(F*y1)) < 0 {}) THEN
270:     flag := 0 {};
271:   END IF;
272:
273:   IF ((1 - (G*y2)/(F*y1)) > 1 {}) THEN
274:     flag := 0 {};
275:   END IF;
276:
277:   (== (H/P)*G/L_ast is bounded like a fraction ==)
278:
279:   IF ((H/P)*G/L_ast < 0 {}) THEN
```



```

280:     flag := 0 {};
281:     END IF;
282:
283:     IF (((H/P)*G/L_ast) > 1 {}) THEN
284:         flag := 0 {};
285:     END IF;
286:
287: END check_self;
288:
289: END ab_stripper;

```

Ficheiro opt.script

```

1: #=====
2: #           O P T . S C R I P T . A B S T R I P
3: #=====
4:
5: ## == Procedure RUNSIM ==
6: proc RUNSIM {} {
7:     global sproc ascSolvStatVect vlist validsim
8:
9:     RUN {sim.default_self}
10:
11:     ASSIGN {sim.T2} [lindex $vlist 0] {K}
12:     ASSIGN {sim.T4} [lindex $vlist 1] {K}
13:     ASSIGN {sim.W4} [lindex $vlist 2] {kg/hr}
14:     ASSIGN {sim.N} [lindex $vlist 3] {}
15:     ASSIGN {sim.A1} [lindex $vlist 4] {m^2}
16:
17:     if {$sproc == 1} {
18: : SOLVE {sim} WITH QRS1v
19:         # First successfull simulation -> turn off gui
20: : if {$ascSolvStatVect(converged) == 1} {
21: :     if {$sproc == 1} {set sproc 2}
22: : }
23:     } elseif {$sproc == 2} {
24: : RESOLVE_NO_GUI
25:     }
26:
27:
28:     if {$ascSolvStatVect(converged) == 1} {
29: : incr validsim 1:
30: : RUN {sim.check_self}
31:     } else {
32: : return "0 [lindex [lindex [u_getval sim.fob] 0] 0] $validsim"
33:     }
34:
35:     set flag [lindex [lindex [u_getval sim.flag] 0] 0]
36:
37:     if {$flag == 1} {
38: : return "1 [lindex [lindex [u_getval sim.fob] 0] 0] $validsim"
39:     } else {
40: : return "0 [lindex [lindex [u_getval sim.fob] 0] 0] $validsim"
41:     }
42: }
43:

```

```
44: ## == MAIN PROGRAM ==
45:
46: global ndim optexe modelfile modeltype dyprob nstep lstep ustep
47: global sproc ascSolvStatVect vlist validsim
48: global ascSolvVect ascSolvQRSlvVect
49: global stopflag
50:
51: set vflag 0
52: set sproc 1
53: set validsim 0
54: set stopflag 0
55:
56: DEFAULTSLV
57: READSCRIPTDATA
58:
59: if {$stopflag==1} {
60:     puts " "
61: } else {
62:     MAIN
63: }
```

Implementação e Alguns resultados

O conjunto de variáveis de decisão, $V_{P.ABSTRIP}$, é constituído pelas seguintes variáveis:

- T_2 - Temperatura da corrente recirculada na cauda do stripper
- T_4 - Temperatura da corrente de saída da cauda do stripper
- N - Número de andares do stripper
- W_4 - Caudal mássico do fluído térmico no permutador P_4
- A_1 - Área de transferência do permutador P_1

O problema está ainda sujeito a duas restrições suplementares, $L \leq 300$ kmol/h e $R \leq 20$. Dois casos podem ser considerados em função do nível de rendimento de recuperação dos compostos na torre de absorção $(G y_2) / (F y_1)$. No primeiro caso a restrição é considerada inactiva, devendo obedecer a um mínimo de 99% no segundo caso.

Ao nível do modelo para ASCEND, aparecem variáveis que não são contempladas pela biblioteca `atoms.a41`. Uma vez que a definição dimensional é importante na modelização para este simulador, torna-se necessário criar uma nova biblioteca, da responsabilidade do utilizador, que define tipos suplementares de variáveis. O código dessa biblioteca, designada por `myatoms.a41`, é apresentado na Quadro A.1, onde se evidenciam os dois novos tipos de variável criados, bem como os quatro pontos necessários à sua definição:

1. O nome do tipo de variável;
2. As suas dimensões;
3. O seu valor por defeito;
4. O conjunto formado pelos seus limites superior e inferior do seu domínio bem como o seu valor nominal.

Quadro A.1 Biblioteca `myatoms.a41`

```

REQUIRE "atoms.a41";
PROVIDE "myatoms.a41";

ATOM dif_temperature
REFINES solver_var
  DIMENSION TMP
  DEFAULT 100{K};
  lower_bound := -10000{K};
  upper_bound := 10000{K};
  nominal := 100{K};
END dif_temperature;

ATOM factor_Gf
REFINES solver_var
  DIMENSION M/L^2/T
  DEFAULT 1{kg/m^2/s};
  lower_bound := 0{kg/m^2/s};
  upper_bound := 100{kg/m^2/s};
  nominal := 1{kg/m^2/s};
END factor_Gf;

```

Além da definição dos novos tipos de variável, o código da biblioteca `myatoms.a41` faz com que a biblioteca `atoms.a41` seja também carregada no sistema, pelo que o utilizador usufrui de todos os tipos de variável especificados na biblioteca que acompanha originalmente o ASCEND.

Os melhores resultados determinados para os Casos I e II do problema encontram-se na Tabela A.1.

Tabela A.1 Problema P.ABSTRIP. Melhores resultados encontrados, Casos I e II.

Caso	Optimizador	T_2 (K)	T_4 (K)	W_4 (kg/h)	N (andares)	A_1 (m ²)	F_{obj}
I	MSIMPISA	326.8	342.7	189.1	25.9	90.6	$-1.733 \cdot 10^6$
II	MSIMPISA	353.1	370.4	818.2	29.4	61.2	$4.214 \cdot 10^3$

A título complementar, apresenta-se na Figura A.4 o percurso do otimizador na procura do melhor valor encontrado para o Caso I. A representação evidencia os saltos

para níveis mais desfavoráveis pela aplicação de um algoritmo de recozimento simulado.

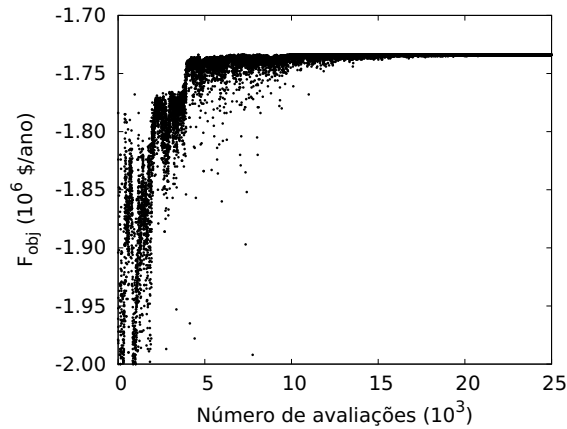


Figura A.4 Problema P.ABSTRIP. Caso I. Trajecto de procura do MSIMPSA para o melhor valor encontrado.

Nomenclatura do problema

Caracteres romanos

$A'_{1,i}$	Constante de Antoine, componente i
$A'_{2,i}$	Constante de Antoine, componente i , K
A_1	Área de transferência do condensador P_1 , m ²
A_2	Área de transferência do vaporizador P_2 , m ²
A_3	Área de transferência do permutador P_3 , m ²
A_4	Área de transferência do permutador P_4 , m ²
A_A	Área da coluna de absorção, m ²
a_p	Área específica, m ⁻¹
a	Volatilidade relativa
B_1	Bomba nº 1
B_2	Bomba nº 2
C_a	Custo dos permutadores, $\$/(\text{m}^2)^{0.556}$
C_e	Custo da electricidade, $\$/\text{ano}/\text{hp}$
C_g	Custos da alimentação, $\$/\text{ano}/\text{kg}$
C_{hp}	Custos de funcionamento das bombas, $\$/\text{hp}^{0.30}$
C_l	Custo associado ao solvente, $\$/\text{ano}/\text{kg}$
C_n	Custo da coluna de regeneração, $\$/(\text{m larg.})^{1.085}$
C_s	Custo do vapor para o vaporizador, $\$/\text{ano}/\text{kg}$

C_w	Custo da água de arrefecimento, \$/ano/kg
C_z	Custo da coluna de absorção, \$/(m alt.)/(m larg.)
c_p	Capacidade calorífica dos permutadores P_3 e P_4 , kcal/kg/K
c_{pw}	Capacidade calorífica das correntes W_1 e W_4 , kcal/kg/K
D_a	Diâmetro da coluna de absorção, m
D_s	Diâmetro da coluna de regeneração, m
D	Caudal molar de destilado (produto), kmol/h
F_c	Retorno do investimento
F	Caudal molar de alimentação à coluna de absorção, mol/h
G_f	Caudal de massa de gás por unidade de área, kg/s/m ²
G	Caudal molar de saída do topo da coluna de absorção, kmol/h
H_g	Altura da unidade de transferência no filme gasoso, m
H_l	Altura da unidade de transferência no filme líquido, m
H_{og}	Altura da unidade global de transferência de gás, m
H	Constante de Henry, mmHg
H_{P1}	Potência da bomba B_1 , hp
H_{P2}	Potência da bomba B_2 , hp
I_{nv}	Investimento económico no processo, \$
K	Constante empírica (eq. Souders-Brown), m/s
K_{P1}	Constante associada à bomba B_1 , hph/N/kmol
K_{P2}	Constante associada à bomba B_2 , hph/N/kmol
L^*	Caudal molar de solvente total alimentado à coluna de absorção, kmol/h
L	Caudal molar de solvente reciclado, kmol/h
M_g	Massa molar do gás, kg/mol
M_l	Massa molar do líquido, kg/mol
N_m	Número máximo de andares da coluna de regeneração
N_{og}	Número de unidades globais de transferência de gás
N	Número de andares da coluna de desorção
P_p	Valor do produto Dx_3 , \$/ano/kg
P	Pressão de operação da coluna de regeneração, mmHg
P_1	Permutador de calor nº 1, condensador da coluna de desorção
P_2	Permutador de calor nº 2, reebulidor da coluna de desorção
P_3	Permutador de calor nº 3
P_4	Permutador de calor nº 4
Q_1	Calor transferido no condensador P_1 , kcal/h
Q_2	Calor transferido no vaporizador P_2 , kcal/h
Q_3	Calor transferido no permutador P_3 , kcal/h
Q_4	Calor transferido no permutador P_4 , kcal/h
q	Constante operatória da coluna de absorção

R_m	Razão de refluxo mínima
R	Razão de refluxo
T_1	Temperatura de operação da coluna de absorção, K
T_2	Temperatura da corrente de alimentação à coluna de regeneração, K
T_3	Temperatura da corrente de topo da coluna de regeneração, K
T_4	Temperatura da corrente de cauda da coluna de regeneração, K
T_5	Temperatura do solvente reciclado depois de arrefecido, K
T_0	Temperatura de saída da água de arrefecimento, K
T_i	Temperatura de entrada da água de arrefecimento, K
T_f	Temperatura do prato de alimentação na coluna de regeneração, K
T_{ref}	Temperatura de referência, K
T_s	Temperatura do vapor saturado no vaporizador, K
U_1	Coeficiente global de transferência de calor do condensador P_1 , kcal/m ² /h/K
U_2	Coeficiente global de transferência de calor do vaporizador P_2 , kcal/m ² /h/K
U_3	Coeficiente global de transferência de calor do permutador P_3 , kcal/m ² /h/K
U_4	Coeficiente global de transferência de calor do permutador P_4 , kcal/m ² /h/K
V_M	Volume molar, m ³ /kmol
W_1	Caudal mássico de água alimentada ao condensador, kg/h
W_2	Caudal mássico de vapor alimentado ao vaporizador, kg/h
W_4	Caudal mássico de água alimentada ao permutador de calor P_4 , kg/h
x_2^*	Fracção molar na corrente de topo da coluna de regeneração
x_1	Fracção molar na corrente de alimentação à coluna de desorção
x_2	Fracção molar na corrente de cauda da coluna de regeneração
x_3	Fracção molar na corrente de topo da coluna de regeneração
X	Variável de substituição
y_1	Fracção molar do componente a remover
y_2	Fracção molar na corrente de topo da coluna de absorção
Z	Altura da coluna de absorção, m

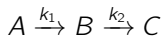
Caracteres gregos

α	Constante utilizada no cálculo de H_g , m(kg/h/m ²) ^{$\alpha-\beta$}
β	Constante utilizada no cálculo de H_g
γ	Constante utilizada no cálculo de H_g
δ	Constante utilizada no cálculo de H_l
ΔT_{ln1}	Diferença de temperatura média logarítmica no condensador P_1 , K
ΔT_{ln2}	Diferença de temperatura média logarítmica no vaporizador P_2 , K
ΔT_{ln3}	Diferença de temperatura média logarítmica no permutador de calor P_3 , K
ΔT_{ln4}	Diferença de temperatura média logarítmica no permutador de calor P_4 , K
ϵ	Porosidade

η_{min}	Eficiência de remoção da coluna de absorção
λ	Calor latente do líquido da corrente de cauda da coluna de regeneração, kcal/kg
λ_w	Calor latente do vapor alimentado ao vaporizador, kcal/kg
μ_l	Viscosidade do líquido, kg/m/s
μ_g	Viscosidade do gás, kg/m/s
ρ_l	Massa específica do líquido, kg/m ³
ρ_g	Massa específica do gás, kg/m ³
ϕ	Constante utilizada no cálculo de H_I , m ^{1+δ}

A.4 P.BATCHT

O exemplo é proposto por Ray (1981). Trata-se de um reactor por partidas onde decorrem duas reacções simultâneamente, envolvendo três espécies químicas diferentes.



O objectivo é encontrar o perfil de temperatura óptima que conduz à produção máxima do intermediário durante o tempo de partida estipulado. Um controlo de temperatura exacto é assumido e a cinética, bem como a dependência da temperatura da constante cinética, são conhecidos.

Equações

$$\frac{dc_A}{dt} = -c_A^2 \left[k_{1,0} \exp\left(\frac{E_1}{RT}\right) \right] \quad (\text{A.51})$$

$$\frac{dc_B}{dt} = c_A^2 \left[k_{1,0} \exp\left(\frac{E_1}{RT}\right) \right] - c_B \left[k_{2,0} \exp\left(\frac{E_2}{RT}\right) \right] \quad (\text{A.52})$$

Uma vez que o reactor opera por partidas, em cada instante de tempo verifica-se o balanço:

$$c_A + c_B + c_C = c_A(t_0) + c_B(t_0) + c_C(t_0) \quad (\text{A.53})$$

Modelo ASCEND

```

1: (*=====
2:                               M O D E L - P . B A T C H T
3:                               =====*)
4:
5: REQUIRE "ivpsystem.a41";
6: REQUIRE "atoms.a41";
7:
8: MODEL biotemp;
9:
10: (*===== D A T A =====*)
11:   k1, k2 IS_A real_constant;
12:
13:   k1 := 4000 {l/mol/hr};
14:   k2 := 620000 {1/hr};
15:
16:   E1, E2 IS_A molar_energy_constant;
17:
18:   E1 := 5000 {cal/mol};
19:   E2 := 10000 {cal/mol};
20:

```



```

21: R IS_A real_constant;
22:
23: R := 1.98721 {cal/mol/K};
24:
25: (*===== V A R I A B L E S =====*)
26: dca_dt, dcb_dt IS_A conc_rate;
27: ca, cb IS_A molar_density;
28:
29: T IS_A temperature;
30: t IS_A time;
31:
32: (*===== E Q U A T I O N S =====*)
33: eq01: dca_dt + k1*exp(-E1/(R*T))*ca^2 = 0;
34: eq02: dcb_dt - k1*exp(-E1/(R*T))*ca^2 + k2*exp(-E2/(R*T))*cb = 0;
35:
36: (*===== M E T H O D S =====*)
37: METHODS
38:
39: (*== DEFAULT ==*)
40: METHOD default_self;
41: dca_dt := 0 {mol/l/hr};
42: dcb_dt := 0 {mol/l/hr};
43: ca := 1 {mol/l};
44: cb := 0 {mol/l};
45: t := 0 {hr};
46: END default_self;
47:
48: (*== BOUNDS ==*)
49: METHOD bound_self;
50: dca_dt.lower_bound := -1e20 {mol/l/hr};
51: dcb_dt.lower_bound := -1e20 {mol/l/hr};
52: T.lower_bound := 298 {K};
53: T.upper_bound := 398 {K};
54: END bound_self;
55:
56: (*== SPECIFY ==*)
57: METHOD specify;
58: T.fixed := TRUE;
59: RUN seqmode;
60: END specify;
61:
62: (*== SEQMODE ==*)
63: METHOD seqmode;
64: ca.fixed := TRUE;
65: cb.fixed := TRUE;
66: END seqmode;
67:
68: (*== SETODE ==*)
69: METHOD set_ode;
70: t.ode_type := -1;
71: dca_dt.ode_type := 2;
72: ca.ode_type := 1;
73: dcb_dt.ode_type := 2;
74: cb.ode_type := 1;
75: dca_dt.ode_id := 1;
76: ca.ode_id := 1;

```

```
77:   dcb_dt.ode_id := 2;
78:   cb.ode_id := 2;
79:   (*== TOLERANCES ==*)
80:   ca.ode_atol := 1e-6 {};
81:   cb.ode_atol := 1e-6 {};
82:   ca.ode_rtol := 1e-10 {};
83:   cb.ode_rtol := 1e-10 {};
84: END set_ode;
85:
86: (*== SETOBS ==*)
87: METHOD set_obs;
88:   ca.obs_id := 1;
89:   cb.obs_id := 2;
90: END set_obs;
91: END biotemp;
```

Ficheiro opt.script

```
1: #=====
2: #           O P T . S C R I P T . P . B A T C H T
3: #=====
4:
5: ## == PROCEDURE RUNSIM ==
6: proc RUNSIM {} {
7:   global sproc ascSolvStatVect vlist validsim
8:
9:   RUN {sim.default_self}
10:
11:   for {set i 0} {$i < 50} {incr i 1} {
12: : ASSIGN {sim.T} [lindex $vlist $i] {K}
13: : INTEGRATE sim FROM $i TO [expr $i +1] WITH BLSODE
14:
15: : if {$ascSolvStatVect(converged) == 0} {
16: :   return "0 [lindex [lindex [u_getval sim.cb] 0] 0] $validsim"
17: : }
18:   }
19:
20:   if {$ascSolvStatVect(converged) == 0} {
21: : return "0 [lindex [lindex [u_getval sim.cb] 0] 0] $validsim"
22:   } else {
23:     incr validsim 1
24: : return "1 [lindex [lindex [u_getval sim.cb] 0] 0] $validsim"
25:   }
26: }
27:
28:
29: ## == MAIN PROGRAM ==
30: global ndim optexe modelfile modeltype dyprob nstep lstep ustep
31: global sproc ascSolvStatVect vlist validsim
32: global ascSolvVect ascSolvQRSlvVect
33: global stopflag
34:
35: set vflag 0
36: set sproc 1
37: set validsim 0
38: set stopflag 0
```

```

39:
40: DEFAULTSLV
41: READSCRIPTDATA
42:
43: if {$stopflag==1} {
44:     puts " "
45: } else {
46:     MAIN
47: }

```

Ficheiro sconf.dat

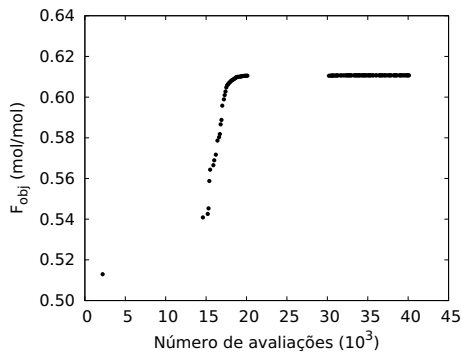
```

1: 'P.BATCH sconf.dat'
2: 'Optimizer'
3: msimpsa
4: 'Model file'
5: model_p.a4c
6: 'It limit, SINGTOL, FEASTOL'
7: 5 1e-12 1e-10
8: 'Dynamic optimization: dyprob'
9: 1
10: 'If dyprob 1: nstep lstep ustep (ustep between braces)'
11: 51 0.02 hr

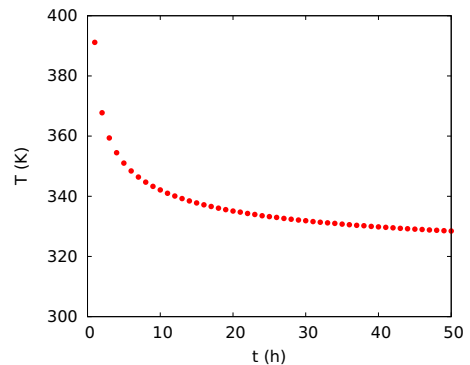
```

Alguns resultados

A integração do modelo permitiu determinar como valores máximos para a produção de intermediário de 0.6065 mol/mol e 0.6107 mol/mol respectivamente com MSIMPISA e MSGA. Na Figura A.5 apresentam-se o perfil de temperatura óptimo e o percurso de resultados parciais determinados até o valor máximo determinado.



(a) Resultados parciais da procura do melhor resultado.



(b) Perfil óptimo de temperatura.

Figura A.5 Problema P.BATCJT. Resultados encontrados com MSGA.

A.5 P.FBE

Descreve-se a produção de etanol por *Saccharomyces cerevisiae* num rector semi-batch (Chen e Hwang, 1990). O objectivo é maximizar a produção de etanol (J) por regulação do caudal de alimentação do rector. O problema pode resolvido utilizando uma decomposição por partes do perfil da variável de decisão em função do tempo. Diferentes abordagens podem ser encontradas na literatura (Banga et al., 1997).

Equações

- Balanços Materiais

$$\frac{dx_1}{dt} = g_1 x_1 - u \frac{x_1}{x_4} \quad (\text{A.54})$$

$$\frac{dx_2}{dt} = -10 g_1 x_1 + u \frac{(150 - x_2)}{x_4} \quad (\text{A.55})$$

$$\frac{dx_3}{dt} = g_2 x_1 - u \frac{x_3}{x_4} \quad (\text{A.56})$$

$$\frac{dx_4}{dt} = u \quad (\text{A.57})$$

- Cinética

$$g_1 = 0.408 \left(\frac{x_2}{0.22 + x_2} \right) \left(1 + \frac{x_3}{16} \right)^{-1} \quad (\text{A.58})$$

$$g_2 = \left(\frac{x_2}{0.44 + x_2} \right) \left(1 + \frac{x_3}{71.5} \right)^{-1} \quad (\text{A.59})$$

- Função Objectivo

$$F_{obj} = J = \max \{x_3(t_f) x_4(t_f)\} \quad (\text{A.60})$$

Alguns resultados

Mostra os resultados obtidos para o tempo de partida de 54 h. O volume máxima do reactor é estabelecido, bem como os valores mínimo e máximo do caudal de alimentação do reactor. O melhor resultado foi encontrado utilizando o MSGA e corresponde a 20157.0 kg, valor que pode ser comparado com outros encontrados na literatura, nomeadamente o melhor valor conhecido de 20318.9 kg (Salcedo et al., 2003).

A.6 P.PRSIB04

Problema proposto por Kocis e Grossmann (1987). Apresenta dois graus de liberdade e não convexidade na função objectivo e restrição de igualdade.

Equações

$$\min F_{obj} = -y + 2x_1 + x_2 \quad (\text{A.61})$$

$$\text{sujeito a } x_1 - 2 \exp(-x_2) = 0 \quad (\text{A.62})$$

$$-x_1 + x_2 + y \leq 0 \quad (\text{A.63})$$

$$0.5 \leq x_1 \leq 1.4 \quad (\text{A.64})$$

$$y \in \{0, 1\} \quad (\text{A.65})$$

Ficheiros SIMOP

- equations.dat

```
'P.RSIB04 - equations.dat'  
x1 - 2*exp(-x2)
```

- constraints.dat

```
'P.RSIB04 - constraints.dat'  
=>0  
x1 - 0.5  
0<=  
-x1 + x2 + y  
x1 - 1.4
```

- fobj.dat

```
'P.RSIB04 - fobj.dat'  
fobj=-y+2*x1+x2
```

- vp_fix.dat

```
'P.RSIB04 - vp_fix.dat'  
% DISCRETE  
y  
% CONTINUOUS
```

Alguns resultados

O melhor resultado apresentado por Salcedo (2001) corresponde a $\{x_1, x_2, y, f\} = \{1.375, 0.375, 1, 2.124\}$. A sua experimentação na validação da metodologia SIMOP

para a formulação MINLP conduziu à verificação do mesmo resultado previamente apresentado.

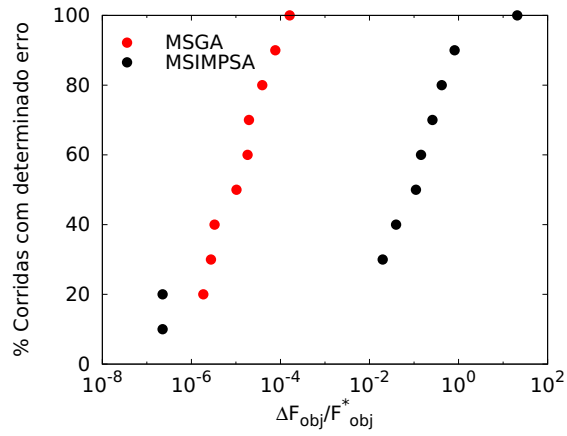


Figura A.6 Problema P.PRSIB04. Distribuições de erro relativo.

A.7 P.PRSIB08

Problema de minimização proposto por Kocis e Grossmann (1988), com três graus de liberdade e três variáveis discretas. O conjunto decisão escolhido é o seguinte:

$$V_{P.PRSIB08} = \{x_1, x_2, y_1, y_2, y_3\}$$

Equações

$$\min \quad \{F_{obj} = 2x_1 + 3x_2 + 1.5y_1 + 2y_2 - 0.5y_3\} \quad (A.66)$$

$$\text{sujeito a } x_1^2 + y_1 = 1.25 \quad (A.67)$$

$$x_2^{1.5} + 1.5y_2 = 3.00 \quad (A.68)$$

$$x_1 + y_1 \leq 1.60 \quad (A.69)$$

$$1.333x_2 + y_2 \leq 3.00 \quad (A.70)$$

$$-y_1 - y_2 + y_3 \leq 0 \quad (A.71)$$

$$x \geq 0 \quad (A.72)$$

$$y \in \{0, 1\}^3 \quad (A.73)$$

Ficheiros SIMOP

- equations.dat

```
'P.RSIB08 - equations.dat'
x1^2 + y_1 - 1.25
x2^1.5 + 1.5 * y_2 - 3.00
```

- constraints.dat

```
'P.RSIB08 - constraints.dat'
=>0
x1
x2
0<=
x1 + y_1 -1.60
1.333*x2 + y_2 - 3.00
-y_1 -y_2 + y_3
```

- fobj.dat

```
'P.RSIB08 - fobj.dat'
fobj=2*x1+3*x2+1.5*y_1+2*y_2-0.5*y_3
```


- vp_fix.dat

```
'P.RSIB08 - vp_fix.dat'  
% DISCRETE  
y_(i);i=1:3  
% CONTINUOUS
```

Alguns resultados

Com base em dez corridas de otimização par cada otimizador, os resultados foram inteiramente reprodutíveis em todas as corridas, onde foi encontrado o valor de 7.667 para a F_{obj} , que está associado ao conjunto decisão $v_{P.PRSIB08} = \{1.118, 1.310, 0, 1, 1\}$ e que corresponde aos resultados previamente encontrados por Salcedo (2001).

A.8 P.CDR

Caso de estudo de destilação reactiva proposto por Ciric e Gu (1994), com abordagens prévias utilizando MSIMPSA e MSGA apresentadas por Cardoso et al. (2000) e Lima et al. (2006). As equações deste exemplo encontram-se na Secção 5.3.1 do Capítulo 5, assim como os principais resultados associados. Neste apêndice apresentam-se os modelos ASCEND e GAMS na base dos resultados encontrados.

Modelo ASCEND

```

1: (*=====
2:
3:   S I M U L A T I O N   M O D E L
4:   R E A C T I V E   D I S T I L L A T I O N   C O L U M N
5:   CIRIC (1994)
6:
7: =====*)
8:
9: REQUIRE "atoms.a41";
10: MODEL destilacao;
11:
12: (*** MODEL DIMENSION DEFINITION =====*)
13:
14: n_trays, n_component, n_reaction IS_A integer_constant;
15:
16: n_trays := 10;
17: n_component := 4;
18: n_reaction := 2;
19:
20:
21: (*** VARIABLES DEFINITION =====*)
22:
23: (*** TRAYS ***)
24:
25: FOR k IN [1..n_trays+1] CREATE
26:   L[k] IS_A molar_rate;
27:
28:   FOR i IN [1..n_component] CREATE
29:     x[i][k] IS_A mole_fraction;
30:   END FOR;
31: END FOR;
32:
33: FOR k IN [0..n_trays] CREATE
34:   V[k] IS_A molar_rate;
35: END FOR;
36:
37:
38: FOR k IN [1..n_trays] CREATE
39:   w[k] IS_A volume;
40:   T[k] IS_A temperature;
41:
42:   FOR i IN [1..n_component] CREATE
43:     F[i][k] IS_A molar_rate;
44:     K[i][k] IS_A factor;
45:   END FOR;
46: END FOR;
47:
48: (*** CONDENSER AND REBOILER ***)
49:
50: beta IS_A fraction;
51: Dist IS_A molar_rate;

```

```

52: Qc, Qb IS_A energy_rate;
53:
54: FOR i IN [1..n_component] CREATE
55:   xd[i] IS_A mole_fraction;
56: END FOR;
57:
58:
59: (*== REACTIONS ==*)
60:
61: FOR j IN [1..n_reaction] CREATE
62:   dH[j] IS_A enthalpy_of_formation_constant;
63:
64:   FOR i IN [1..n_component] CREATE
65:     v[i][j] IS_A real_constant;
66:   END FOR;
67:
68:   FOR k IN [1..n_trays] CREATE
69:     E[j][k] IS_A molar_rate;
70:   END FOR;
71: END FOR;
72:
73: lambda IS_A heat_of_vaporization_constant;
74:
75: lambda := 40000 {J/mol};
76: dH[1] := -80.0e3 {J/mol};
77: dH[2] := -13.1e3 {J/mol};
78: v[1][1] := -1 {};
79: v[2][1] := -1 {};
80: v[3][1] := 1 {};
81: v[4][1] := 0 {};
82: v[1][2] := -1 {};
83: v[2][2] := 0 {};
84: v[3][2] := -1 {};
85: v[4][2] := 1 {};
86:
87:
88: (*== COLUMN OVERALL VARIABLES ==*)
89:
90: FOR i IN [1..n_component] CREATE
91:   B[i] IS_A molar_rate;
92:   P[i] IS_A molar_rate;
93: END FOR;
94:
95: D,H[1..n_trays], Height IS_A distance;
96: HO, Hmin, Cd IS_A real_constant;
97:
98: Cd := 0.01331 {};
99: HO := 10 {ft};
100: Hmin := 2 {ft};
101:
102: (*== DATA FOR K(i,j) calculation ==*)
103:
104: FOR i IN [1..n_component] CREATE
105:   a[i], b[i], t1[i], t2[i] IS_A real_constant;
106: END FOR;
107:
108: a[1] := 71.9 {};
109: a[2] := 221.2 {};
110: a[3] := 77.0 {};
111: a[4] := 47.0 {};
112: b[1] := 5.720 {};
113: b[2] := 6.310 {};
114: b[3] := 9.940 {};
115: b[4] := 10.42 {};
116: t1[1] := 469 {K};
117: t1[2] := 647 {K};
118: t1[3] := 645 {K};

```

```

119: t1[4] := 681 {K};
120: t2[1] := 35.9 {K};
121: t2[2] := 52.9 {K};
122: t2[3] := 71.4 {K};
123: t2[4] := 80.6 {K};
124:
125: (=== EQUATIONS DEFINITION =====)
126:
127: (=== MASS BALANCES
128:     TRAY 1     ===)
129:
130: FOR i IN [1..n_component] CREATE
131:     zmassbal_tray1[i]: F[i][1]-L[1]*x[i][1]*(1-beta) + L[2]*x[i][2] - V[1]*K[i][1]*x[i][1]
        + (v[i][1]*E[1][1] + v[i][2]*E[2][1]) = 0{mol/s};
132: END FOR;
133:
134: (=== TRAYS 2 .. N ===)
135:
136: FOR k IN [2..n_trays] CREATE
137:     FOR i IN [1..n_component] CREATE
138:         zmassbal_tray[i][k]: F[i][k] + V[k-1]*K[i][k-1]*x[i][k-1] + L[k+1]*x[i][k+1] - L[k]*x[i][k]
            - V[k]*K[i][k]*x[i][k] + (v[i][1]*E[1][k] + v[i][2]*E[2][k]) = 0{mol/s};
139:     END FOR;
140: END FOR;
141:
142: (=== OTHER EQUATIONS FOR TRAYS 1 .. N ===)
143:
144: FOR k IN [1..n_trays] CREATE
145:     (=== EXTENSION OF REACTIONS EVOLVED ===)
146:     zr_ext_1[k]: E[1][k]
        - w[k]*3.15e15*1{mol/s/m^3}*exp(-9547*1{K}/T[k])*x[1][k]*x[2][k] = 0{mol/s};
147:     zr_ext_2[k]: E[2][k]
        - w[k]*6.3e15*1{mol/s/m^3}*exp(-9547*1{K}/T[k])*x[1][k]*x[3][k] = 0{mol/s};
148:
149:     (=== GLOBAL MASS BALANCE ===)
150:     zgmassbal[k]: F[1][k] + F[2][k] + F[3][k] + F[4][k] + V[k-1] + L[k+1] - L[k] - V[k]
        + (v[1][1]+v[2][1]+v[3][1]+v[4][1])*E[1][k]
        + (v[1][2]+v[2][2]+v[3][2]+v[4][2])*E[2][k] = 0{mol/s};
151:
152:     (=== STOICHEOMETRIC EQUATION FOR VAPOUR PHASE ===)
153:     zy_stoich[k]: K[1][k]*x[1][k] + K[2][k]*x[2][k] + K[3][k]*x[3][k]
        + K[4][k]*x[4][k] - 1 = 0;
154:
155:     (=== ENERGY BALANCE ===)
156:     zenergybal[k]: V[k-1] - V[k] - (dH[1]*E[1][k] + dH[2]*E[2][k])/lambda = 0{mol/s};
157:
158:     (=== CALCULATION OF THE TRAY HEIGHT ===)
159:     ztray_height[k]: H[k]/1{ft} - (Hmin/1{ft}) + 1.27*((w[k]/1{m^3})/0.3048^3/(D/1{ft})^2) = 0;
160:
161:     FOR i IN [1..n_component] CREATE
162:         (=== VLE CONSTANT DEFINITION ===)
163:         zkdef[i][k]: K[i][k] - a[i]*exp(b[i]*(T[k]-t1[i])/(T[k]-t2[i])) = 0;
164:     END FOR;
165: END FOR;
166:
167: (=== EQUATIONS FOR REBOILER; CONDENSER AND OVERALL COLUMN ===)
168:
169: FOR i IN [1..n_component] CREATE
170:     zmassbal_reb[i]: B[i] - (1-beta)*L[1]*x[i][1] = 0{mol/s};
171:     zmassbal_overall[i]: xd[i]*Dist + B[i] - P[i] = 0{mol/s};
172:     zcond_eq1[i]: x[i][n_trays+1] - xd[i] = 0;
173:     zcond_eq2[i]: xd[i] - K[i][n_trays]*x[i][n_trays] = 0;
174: END FOR;
175:
176: (=== ANOTHER PARAMETERS ===)
177:
178: zbeta_def: beta*L[1] - V[0] = 0{mol/s};

```

```

179:
180: zdist_def: Dist - V[n_trays] + L[n_trays+1] = 0{mol/s};
181:
182: zQb_eq: Qb - beta*lambda*L[1] = 0{J/s};
183:
184: zQc_eq: Qc - lambda*V[n_trays] = 0{J/s};
185:
186: zdiameter_def: (D/(1{ft}))^4 - Cd*beta^2*(L[1]/1{mol/s})^2 = 0;
187:
188: zheight_def: Height - SUM[H[k] | k IN [1..n_trays]] = 0{ft};
189:
190: (=== O B J E C T I V E   F U N C T I O N =====)
191:
192: (=== COST DATA ===)
193:
194: c1, c2, csh, ct, cr, cc, c0 IS_A real_constant;
195:
196: c1 := 43.7e-3 {dollar/mol};
197: c2 := 21.9e-3 {dollar/mol};
198: csh := 222 {dollar/year};
199: ct := 15.7 {dollar/year};
200: cr := 146.8e-6 {dollar*s/J/year};
201: cc := 24.5e-6 {dollar*s/J/year};
202: c0 := 10000 {dollar/year};
203:
204: fob IS_A cost_per_time;
205:
206: FOB_eq: fob - (c0 + c1*SUM[F[1][k] | k IN [1..n_trays]] + c2*SUM[F[2][k] | k IN [1..n_trays]]
      + cr*Qb + cc*Qc + ct*(D/1{ft})^1.55*(Height/1{ft})
      + csh*(D/1{ft})*((H0+Height)/1{ft})^0.802) = 0{dollar/year};
207:
208:
209: (=== M E T H O D =====)
210:
211: METHODS
212:
213: (=== D E F A U L T   S E L F =====)
214:
215: METHOD default_self;
216:   FOR k IN [1..n_trays] DO
217:     H[k] := 2 {ft};
218:   END FOR;
219:
220:   FOR k IN [1..n_trays+1] DO
221:     L[k] := 500 {kmol/hour};
222:   END FOR;
223:
224:   FOR k IN [0..n_trays] DO
225:     V[k] := 500 {kmol/hour};
226:   END FOR;
227:
228:   Height := 10 {m};
229: END default_self;
230:
231:
232: (=== S P E C I F Y =====)
233:
234: (=== F(i,k) and w(k) as decision variables with beta replacing F(2,1) ===)
235:
236: METHOD specify;
237:   FOR i IN [1..n_component] DO
238:     FOR k IN [1..n_trays] DO
239:       F[i][k].fixed := TRUE;
240:     END FOR;
241:   END FOR;
242:
243:   F[2][1].fixed := FALSE;

```

```

244:
245:   FOR k IN [1..n_trays] DO
246:     w[k].fixed := TRUE;
247:   END FOR;
248:
249:   Dist.fixed := TRUE;
250:
251:   P[3].fixed := TRUE;
252:
253:   beta.fixed := TRUE;
254: END specify;
255:
256: (*== B O U N D   S E L F =====*)
257:
258: METHOD bound_self;
259:   FOR k IN [1..n_trays] DO
260:     T[k].lower_bound := 300 {K};
261:     T[k].upper_bound := 600 {K};
262:   END FOR;
263:
264:   (*== FLOW LIMITS ==*)
265:   FOR k IN [1..n_trays+1] DO
266:     L[k].upper_bound := 8000 {mol/s};
267:   END FOR;
268:
269:   FOR k IN [0..n_trays] DO
270:     V[k].upper_bound := 8000 {mol/s};
271:   END FOR;
272: END bound_self;
273:
274:
275: (*== S C A L E   S E L F =====*)
276:
277: METHOD scale_self;
278:   FOR k IN [1..n_trays] DO
279:     w[k].nominal := 1 {m^3};
280:     E[1][k].nominal := 1 {mol/s};
281:     E[2][k].nominal := 1 {mol/s};
282:     H[k].nominal := 2 {ft};
283:     T[k].nominal := 350 {K};
284:
285:     FOR i IN [1..n_component] DO
286:       F[i][k].nominal := 10 {mol/s};
287:       K[i][k].nominal := 10 {};
288:     END FOR;
289:   END FOR;
290:
291:   FOR i IN [1..n_component] DO
292:     xd[i].nominal := 1 {};
293:     P[i].nominal := 1 {mol/s};
294:     B[i].nominal := 1 {mol/s};
295:   END FOR;
296:
297:   FOR k IN [0..n_trays] DO
298:     V[k].nominal := 500 {kmol/hr};
299:   END FOR;
300:
301:   FOR k IN [1..n_trays+1] DO
302:     L[k].nominal := 500 {kmol/hr};
303:
304:     FOR i IN [1..n_component] DO
305:       x[i][k].nominal := 1 {};
306:     END FOR;
307:   END FOR;
308:
309:   beta.nominal := 1 {};
310:

```

```

311: Dist.nominal := 1 {mol/s};
312:
313: D.nominal := 1 {m};
314:
315: Height.nominal := 10 {ft};
316: END scale_self;:
317:
318: (*== V A L U E S =====*)
319:
320: METHOD values;
321:   FOR i IN [3..4] DO
322:     FOR k IN [1..n_trays] DO
323:       F[i][k] := 0 {mol/s};
324:     END FOR;
325:   END FOR;
326:
327: Dist := 0 {mol/s};
328:
329: P[3] := 0.69444E+01 {mol/s};
330: END values;
331:
332: END destilacao;

```

Modelo GAMS

```

1: *-----
2: * Reactive Distillation Process - Production of Ethylene Glycol
3: * Model from A.R. Ciric, D. Gu, "Synthesis of Nonequilibrium Reactive
4: * Distillation Processes by MINLP Optimization", AIChE J., 40(9),
5: * 1479-1487, (1994).
6: *
7: * Initial version (0.1).
8: *   Scaling based on magnitudes of groups of variables and equations.
9: *   Values for cD corrected.
10: *   Constraints for the column diameter on each tray.
11: *   Initialization based on estimatives of variables.
12: *   7 equilibrium stages + total condenser + partial reboiler.
13: *
14: * (c) 2007, R. Lima, H. Silva, R. Salcedo and N. Oliveira.
15: *-----
16:
17: *$OFFUPPER
18: *$OFFSYMXREF
19: *$OFFSYMLIST
20: OPTION LIMROW=0;
21: OPTION LIMCOL=0;
22: *option limcol=1000;
23: *option limrow=1000;
24: OPTION DECIMALS = 8;
25:
26: * Model parameters:
27: SETS
28:   j          'Global index for trays (0..NT+1)'          /n0*n8/
29:   i          'Components'                                /i1*i4/
30:   k          'Reaction paths'                            /r1*r2/;
31:
32: SCALARS
33:   a2          'Reduced activation energy'                /9.547/
34:   b          'Residue flowrate [mol/s]'
35:   beta        'Boilup ratio'
36:   botfeed    'Index of bottom feed tray'
37:   c0          'Fixed cost [$/yr]'                        /1e4/
38:   cc          'Cooling water cost [$/W.yr]'              /24.5e-3/
39: *   cD        'Cost coefficient [m^2.s/mol]'              /0.01026/
40:   cD          'Cost coefficient [m^2.s/mol]'              /0.01331/

```

```

41: cr          'Steam cost [$/W.yr]' /146.8e-3/
42: costsep     'Cost of downstream separation of H2O [$/yr]'
43: csep        'Cost coefficient for H2O separation [$/mol]' /0.15e-3/
44: csh         'Cost coefficient [$/yr]' /222/
45: ct          'Cost coefficient [$/yr]' /15.7/
46: eachfeed    'Individual feed for each component [mol/s]'
47: egdes       'Desired production rate of EG [mol/s]' /6.944444/
48: egreal      'Obtained production rate of EG [mol/s]'
49: exc1        'Excess of EO [%]'
50: exc2        'Excess of H2O [%]'
51: Hmin        'Min spacing between trays [m]' /0.6096/
52: lambda      'Enthalpy of vaporization [J/mol]' /40e3/
53: n           'Integer count'
54: ntrays      'Number of trays in column'
55: ta          'Normalization constant for T [K]' /1000/
56: tb          'Normalization constant for T' /1.9/
57: topfrac     'Top fraction of trays with feed, reaction'
58: totfeed     'Total feed for each component [mol/s]'
59: Vb          'Vapor flowrate in reboiler [mol/s]'
60: weach       'Estimate of each reaction volume [m^3]'
61: wttotal     'Total reaction volume [m^3]'
62: x3b         'Mass fraction of EG in bottom stream';
63:
64: PARAMETER a1(k) 'Normalized pre-exponential factor of reaction k [mol/m^3.s]'
65: / r1 = 17.546879, r2 = 18.240026 /;
66:
67: PARAMETER c1(i) 'VLE coefficient'
68: / i1 = 71.9, i2 = 221.2, i3 = 77.0, i4 = 47.0 /;
69:
70: PARAMETER c2(i) 'VLE coefficient'
71: / i1 = 5.720, i2 = 6.310, i3 = 9.940, i4 = 10.42 /;
72:
73: PARAMETER c3(i) 'VLE coefficient'
74: / i1 = 469, i2 = 647, i3 = 645, i4 = 681 /;
75:
76: PARAMETER c4(i) 'VLE coefficient'
77: / i1 = 35.9, i2 = 52.9, i3 = 71.4, i4= 80.6 /;
78:
79: PARAMETER cf(i) 'Cost of feed components [$/mol]'
80: / i1 = 43.7e-3, i2 = 21.9e-3, i3 = 0, i4 = 0 /;
81:
82: PARAMETER DH(k) 'Reaction enthalpy [J/mol]'
83: / r1 = -80.0e3, r2 = -13.1e3/;
84:
85: PARAMETER li(i,j) 'Liquid flowrate of component i out of tray j [mol/s]';
86: PARAMETER T(j) 'Tray temperatures';
87: PARAMETER totx(j) 'Normalization factor for compositions';
88: PARAMETER toty(j) 'Normalization factor for compositions';
89: PARAMETER vi(i,j) 'Vapor flowrate of component i out of tray j [mol/s]';
90:
91: TABLE vs(i,k) 'Stoichiometric coefficient of component i in reaction k'
92: r1 r2
93: i1 -1 -1
94: i2 -1 0
95: i3 1 -1
96: i4 0 1;
97:
98: * Variables:
99: POSITIVE VARIABLES
100: D 'Column diameter [m]'
101: er(k,j) 'Extension of reaction k in tray j [mol/s]'
102: F(i,j) 'Feed flowrate of component i in tray j [mol/s]'
103: h(j) 'Height of equilibrium stage j [m]'
104: ht 'Total column height [m]'
105: lnbeta 'Negative log of the boilup ratio'
106: lnkr(k,j) 'Log of kinetic constant of reaction k in tray j [mol/m^3.s]'
107: lnkv(i,j) 'Log of VLE constant for component i in tray j'

```



```

108: L(j)          'Liquid flowrate out of tray j [mol/s]'
109: Qb           'Reboiler duty [MW]'
110: Qc           'Condenser duty [MW]'
111: theta(j)     'Normalized temperature in tray j'
112: V(j)         'Vapor flowrate out of tray j [mol/s]'
113: w(j)         'Reaction volume in tray j [m^3]'
114: x(i,j)       'Liquid molar fraction of component i in tray j'
115: y(i,j)       'Vapor molar fraction of component i in tray j';
116:
117: VARIABLES
118: cost         'Objective value [M$]';
119:
120: * Equations:
121: EQUATIONS
122: ebc, eqrec(j), ebt(j), ebr(j), eqdi(j), eqhs(j), eqht, eqkr(k,j),
123: er1(j), er2(j), exy(i,j), mbc(i,j), mbc1, mbr1(i,j), mbr2(j),
124: mbt(i,j), objfun, sumx(j), sumy(j), vle(i,j);
125:
126:
127: * Objective function:
128: objfun .. cost =e= 1e-6*(c0 + cr*1e6*Qb + cc*1e6*Qc +
129: 3600*24*365*(SUM((i,j)$((ord(j) ne 1) and (ord(j) ne card(j))),
130: cf(i)*F(i,j))) + ct*(D/0.3048)**1.55*(ht/0.3048) +
131: csh*(D/0.3048)*(ht/0.3048 + 10)**0.802);
132:
133:
134: * Mass balances:
135: * Condenser (j = 0). Note: x(i,'0') = xD(i), V('0') = D.
136: mbc(i,j)$ (ord(j) eq 1) .. x(i,j) =e= y(i,j+1);
137: mbc1 .. V('n1') =e= L('n0') + V('n0');
138:
139: * Trays (j = 1..N):
140: mbt(i,j)$ ((ord(j) ne 1) and (ord(j) ne card(j))) ..
141: F(i,j) + L(j-1)*x(i,j-1) + V(j+1)*y(i,j+1) + SUM(k, vs(i,k)*er(k,j))
142: =e= L(j)*x(i,j) + V(j)*y(i,j);
143:
144: * Reboiler (j = N+1):
145: * Note: x(i,N+1) = xB(i), L(N+1) = B.
146: mbr1(i,j)$ (ord(j) eq card(j)) .. L(j-1)*x(i,j-1) =e= L(j)*x(i,j)
147: + V(j)*y(i,j);
148: mbr2(j)$ (ord(j) eq card(j)) .. (1 - exp(-lnbeta))*L(j-1) =e= V(j);
149:
150: * Equilibrium constraints (j = 1..NT):
151: exy(i,j)$ (ord(j) ne 1) .. y(i,j) =e= exp(lnkv(i,j))*x(i,j);
152:
153: * Summations (j = 1..NT):
154: sumx(j)$ (ord(j) ne 1) .. sum(i, x(i,j)) =e= 1;
155: sumy(j)$ (ord(j) ne 1) .. sum(i, y(i,j)) =e= 1;
156:
157:
158: * Energy balances:
159: * Condenser:
160: ebc .. V('n1') =e= Qc*1e6/lambda;
161:
162: * Trays (j = 1..NT):
163: ebt(j)$ ((ord(j) ne 1) and (ord(j) ne card(j))) ..
164: V(j+1) =e= V(j) + sum(k, DH(k)*er(k,j))/lambda;
165:
166: * Reboiler:
167: ebr(j)$ (ord(j) eq card(j)) .. Qb*1e6/lambda =e= V(j);
168:
169: * Auxiliary equations:
170: * Extension of reaction (j = 1..NT);
171: er1(j)$ ((ord(j) ne 1) and (ord(j) ne card(j))) ..
172: er('r1',j) =e= w(j)*exp(lnkr('r1',j))*x('i1',j)*x('i2',j);
173: er2(j)$ ((ord(j) ne 1) and (ord(j) ne card(j))) ..
174: er('r2',j) =e= w(j)*exp(lnkr('r2',j))*x('i1',j)*x('i3',j);

```

```

175:
176: * Kinetic constants (j = 1..NT):
177: eqkr(k,j)$((ord(j) ne 1) and (ord(j) ne card(j))) ..
178:   lnkr(k,j)=e= a1(k) - a2*theta(j);
179:
180: * VLE constants:
181: vle(i,j)$((ord(j) ne 1) .. (lnkv(i,j) - log(C1(i)))*(1 -
182:   (C4(i)/ta)*(theta(j) + tb)) =e= C2(i)*(1 - (C3(i)/ta)*(theta(j) + tb));
183:
184:
185: * Physical dimensions:
186: * Diameter:
187: eqdi(j)$((ord(j) ne 1) .. D**2 =g= cD*V(j);
188:
189: * Height of each stage:
190: eqhs(j)$((ord(j) ne 1) and (ord(j) ne card(j))) .. (h(j) - Hmin)*D**2 =e=
191:   1.27*0.3048*w(j);
192:
193: * Total height:
194: eqht .. ht =e= SUM(j$((ord(j) ne 1) and (ord(j) ne card(j))), h(j));
195:
196:
197: * Problem specifications:
198: * Recovery constraint:
199: eqrec(j)$((ord(j) eq card(j)) .. L(j)*x('i3',j) =g= egdes;
200:
201: * Component feeds:
202: F.FX('i3',j)$((ord(j) ne 1) and (ord(j) ne card(j))) = 0;
203: F.FX('i4',j)$((ord(j) ne 1) and (ord(j) ne card(j))) = 0;
204:
205:
206: * Model initialization:
207: * Estimatives of variables:
208: display "Initialization phase:";
209:
210: * Number of trays and boilup ratio:
211: ntrays = card(j) - 2;
212: if (ntrays < 1,
213:   abort "ntrays < 1";
214: elseif(ntrays <= 7),
215:   lnbeta.L = 6.9975 - 0.1125*ntrays;
216: elseif (ntrays <= 10.2),
217:   lnbeta.L = 10.6069 - 0.628125*ntrays;
218: elseif (ntrays <= 30),
219:   lnbeta.L = 4.76667 - 0.0555556*ntrays;
220: else
221:   lnbeta.L = 3;
222: );
223: beta = 1 - exp(-lnbeta.L);
224:
225: * Tray and column height:
226: h.L(j)$((ord(j) ne 1) and (ord(j) ne card(j))) = Hmin;
227: ht.L = sum(j$((ord(j) ne 1) and (ord(j) ne card(j))), h.L(j));
228:
229: * Top range where feeds and reaction occur:
230: if (ntrays <= 60,
231:   topfrac = 0.655 + 0.00461*ntrays;
232: else
233:   topfrac = 0.94;
234: );
235: botfeed = round(topfrac*ntrays);
236:
237: * Feeds:
238: * Components fed to the top range, with a 10% excess. H2O only fed to
239: * top tray:
240: totfeed = 1.1*egdes;
241: eachfeed = totfeed/botfeed;

```

```

242: F.L('i1',j)$((ord(j) ne 1) and (ord(j) <= botfeed + 1)) = eachfeed;
243: F.L('i2',j)$((ord(j) eq 2) = totfeed;
244:
245: * Reaction volumes (only the top range):
246: w.L(j)$((ord(j) ne 1) and (ord(j) ne card(j))) = 1e-7;
247: if (ntrays <= 10,
248:   wttotal = 0.874 + 0.223*ntrays;
249: else
250:   wttotal = log(17.2 + 0.446*ntrays);
251: );
252: weach = wttotal/botfeed;
253: w.L(j)$((ord(j) ne 1) and (ord(j) <= botfeed + 1)) = weach;
254:
255: * Temperature distribution (sigmoid) and dependent functions:
256: T(j)$((ord(j) ne 1) = 373.1 + 91.16/(1 + exp(20*(-(ord(j)-1)/(ntrays+1) + 0.8)));
257: theta.L(j)$((ord(j) ne 1) = ta/T(j) - tb;
258:
259: lnkr.L(k,j)$((ord(j) ne 1) and (ord(j) ne card(j))) = a1(k) - a2*theta.L(j);
260: lnkv.L(i,j)$((ord(j) ne 1) = log(C1(i)) + c2(i)*(ta - C3(i)*(theta.L(j) +
261:   tb))/(ta - C4(i)*(theta.L(j) + tb));
262:
263: * Composition estimates in column (sigmoids, like T):
264: * Liquid phase:
265: x.L('i1',j) = 0.001;
266: x.L('i2',j) = 1 - 1/(1 + exp(20*(-(ord(j)-1)/(ntrays+1) + 0.8)));
267: x.L('i3',j) = 1/(1 + exp(20*(-(ord(j)-1)/(ntrays+1) + 0.8)));
268: x.L('i4',j) = 0.001;
269:
270: * Normalize compositions:
271: totx(j) = sum(i, x.L(i,j));
272: x.L(i,j) = x.L(i,j)/totx(j);
273:
274: * Vapor phase:
275: y.L(i,j)$((ord(j) ne 1) = exp(lnkv.L(i,j))*x.L(i,j);
276: toty(j)$((ord(j) ne 1) = sum(i, y.L(i,j));
277: y.L(i,j)$((ord(j) ne 1) = y.L(i,j)/toty(j);
278:
279: * Extension of reactions:
280: eR.L('r1',j)$((ord(j) ne 1) and (ord(j) ne card(j))) =
281:   w.L(j)*exp(lnkr.L('r1',j))*x.L('i1',j)*x.L('i2',j);
282:
283: eR.L('r2',j)$((ord(j) ne 1) and (ord(j) ne card(j))) =
284:   w.L(j)*exp(lnkr.L('r2',j))*x.L('i1',j)*x.L('i3',j);
285:
286: * Flows in the reboiler:
287: x3b = sum(j$(ord(j) eq card(j)), x.L('i3',j));
288: b = egdes/x3b;
289: Vb = beta*b/(1 - beta);
290: L.L(j)$((ord(j) eq card(j)) = b;
291: L.L(j)$((ord(j) eq (card(j)-1)) = b/(1 - beta);
292: V.L(j)$((ord(j) eq card(j)) = Vb;
293:
294: * Vapor flows (from energy balance):
295: for (n = card(j) - 1 downto 2 by 1,
296:   V.L(j)$((ord(j) eq n) = V.L(j+1) - sum(k, DH(k)*er.L(k,j))/lambda;
297: );
298: V.L(j)$((ord(j) eq 1) = 0;
299:
300: * Heat transfered:
301: Qb.L = 1e-6*lambda*Vb;
302: Qc.L = 1e-6*lambda*V.L('n1');
303:
304: * Column diameter:
305: D.L = sqrt(sqrt(cD)*Vb);
306:
307: * Liquid flows (from mass balances):
308: for (n = card(j) - 1 downto 2 by 1,

```

```

309:   L.L(j-1)$ord(j) eq n) = L.L(j) + V.L(j) - SUM(i, F.L(i,j)) -
310:   V.L(j+1) - SUM(i, SUM(k, vs(i,k)*er.L(k,j)));
311: );
312: V.L('n0') = V.L('n1') - L.L('n0');
313:
314: display b, beta, botfeed, eachfeed, ntrays, T, totfeed, Vb, x3b;
315: display D.L, er.L, F.L, h.L, ht.L, L.L, lnbeta.L, lnkr.L, lnkv.L;
316: display Qb.L, Qc.L, theta.L, V.L, w.L, x.L, y.L;
317: * $exit end
318: * Note: More accurate composition estimates can be obtained by iteration,
319: * using the previous values.
320:
321: * Bounds on variables:
322: **** Generalizar ****
323: lnbeta.UP = 8;
324: D.UP = 10;
325:
326: er.UP(k,j)$((ord(j) ne 1) and (ord(j) ne card(j))) = 50;
327: F.UP('i1',j)$((ord(j) ne 1) and (ord(j) ne card(j))) = 1.5*totfeed;
328: F.UP('i2',j)$((ord(j) ne 1) and (ord(j) ne card(j))) = 1.5*totfeed;
329:
330: H.UP(j)$((ord(j) ne 1) and (ord(j) ne card(j))) = 1;
331: Ht.UP = ntrays;
332:
333: lnkr.LO(k,j)$((ord(j) ne 1) and (ord(j) ne card(j))) = 1.8;
334: lnkr.UP(k,j)$((ord(j) ne 1) and (ord(j) ne card(j))) = 20;
335:
336: lnkv.LO('i1',j)$ord(j) ne 1) = -0.0001;
337: lnkv.UP('i1',j)$ord(j) ne 1) = 5;
338: lnkv.LO('i2',j)$ord(j) ne 1) = -5;
339: lnkv.UP('i2',j)$ord(j) ne 1) = 4;
340: lnkv.LO('i3',j)$ord(j) ne 1) = -13;
341: lnkv.UP('i3',j)$ord(j) ne 1) = 2;
342: lnkv.LO('i4',j)$ord(j) ne 1) = -17;
343: lnkv.UP('i4',j)$ord(j) ne 1) = -0.0001;
344:
345: L.UP(j) = 8000;
346: Qb.UP = 500;
347: Qc.UP = 500;
348:
349: theta.LO(j)$ord(j) ne 1) = 0.01;
350: theta.UP(j)$ord(j) ne 1) = 1.8;
351:
352: V.UP(j) = 8000;
353: w.up(j)$((ord(j) ne 1) and (ord(j) ne card(j))) = 3;
354: x.UP(i,j) = 1;
355: y.UP(i,j)$ord(j) ne 1) = 1;
356:
357:
358: * Scale factors:
359: **** Generalizar ****
360: * Equations:
361: ebc.scale = 5500;
362: ebr.scale(j) = 5500;
363: ebt.scale(j) = 5500;
364: eqdi.scale(j) = 640;
365: eqhs.scale(j) = 390;
366: eqht.scale = 2;
367: eqkr.scale(k,j) = 18;
368: eqrec.scale(j) = 7;
369: er1.scale(j) = 5;
370: er2.scale(j) = 1;
371: exy.scale(i,j) = 1;
372: mbc1.scale = 5500;
373: mbc.scale(i,j) = 1;
374: mbr1.scale(i,j) = 4000;
375: mbr2.scale(j) = 5500;

```

```

376: mbt.scale(i,j) = 5500;
377: vle.scale(i,j) = 5;
378:
379: * Variables:
380: D.scale = 25;
381: er.scale(k,j) = 7;
382: F.scale(i,j) = 7;
383: h.scale(j) = 1;
384: ht.scale = 2;
385: lnbeta.scale = 7;
386: lnkr.scale(k,j) = 10;
387: lnkv.scale(i,j) = 5;
388: L.scale(j) = 5500;
389: Qb.scale = 200;
390: Qc.scale = 200;
391: theta.scale(j) = 1;
392: V.scale(j) = 5500;
393: w.scale(j) = 1;
394: x.scale(i,j) = 1;
395: y.scale(i,j) = 1;
396:
397:
398: * Solution:
399: model teste /all/;
400: teste.scaleopt = 1;
401: option nlp = conopt;
402: option iterlim = 5000;
403:
404: solve teste minimizing cost using nlp;
405:
406: * Additional solve to make sure that large problems are
407: * effectively solved:
408: if (ntrays > 30,
409:   solve teste minimizing cost using nlp;
410: );
411:
412: * Additional variables:
413: * Absolute temperature:
414: T(j)$ord(j) ne 1) = ta/(theta.L(j) + tb);
415: display T;
416:
417: * Boilup ratio:
418: beta = 1 - exp(-lnbeta.L);
419: display beta;
420:
421: * EG recovery:
422: egreal = sum(j$(ord(j) eq card(j)), L.L(j)*x.L('i3',j));
423: display egreal;
424:
425: * Individual flows:
426: li(i,j) = L.L(j)*x.L(i,j);
427: vi(i,j) = V.L(j)*y.L(i,j);
428: display f.L, li, vi;
429:
430: * Excess of reactants:
431: exc1 = 100*(sum(j$((ord(j) ne 1) and (ord(j) ne card(j))),
432:   F.L('i1',j))/egreal - 1);
433: exc2 = 100*(sum(j$((ord(j) ne 1) and (ord(j) ne card(j))),
434:   F.L('i2',j))/egreal - 1);
435: display exc1, exc2;
436:
437: * Total reaction volume:
438: wttotal = sum(j$((ord(j) ne 1) and (ord(j) ne card(j))), w.L(j));
439: display wttotal;
440:
441: * Cost of downstream separation of H2O:
442: costsep = csep*3600*24*365*sum(j$(ord(j) eq card(j)), li('i2',j));

```

```

443: display costsep;
444:
445: * All variables;
446: display cost.L, D.L, er.L, F.L, h.L, ht.L, lnbeta.L, lnkr.L,
447:   lnkv.L, L.L, Qb.L, Qc.L, theta.L, V.L, w.L, x.L, y.L;

```

Ficheiros SIMOP

Dada a dimensão de alguns dos ficheiros do modelo, apresentam-se apenas os mais representativos, omitindo-se os relativos às restrições.

• equations.dat

```

1: 'P.CDR - equations.dat '
2: YF_(1)*zF_(i,1)-L_(1)*x_(i,1)*(1-beta)+L_(2)*x_(i,2)-V_(1)*zzK_(i,1)*x_(i,1)
   +v_(i,1)*zzer_(1,1)+v_(i,2)*zzer_(2,1);i=1:2
3: -L_(1)*x_(i,1)*(1-beta)+L_(2)*x_(i,2)-V_(1)*zzK_(i,1)*x_(i,1)+v_(i,1)*zzer_(1,1)
   +v_(i,2)*zzer_(2,1);i=3:4
4: (zF_(j,k)+V_(k-1)*zzK_(j,k-1)*x_(j,k-1)+L_(k+1)*x_(j,k+1)-L_(k)*x_(j,k)
   -V_(k)*zzK_(j,k)*x_(j,k)+v_(j,1)*zzer_(1,k)+v_(j,2)*zzer_(2,k));j=1:2;k=2:20
5: (V_(k-1)*zzK_(j,k-1)*x_(j,k-1)+L_(k+1)*x_(j,k+1)-L_(k)*x_(j,k)-V_(k)*zzK_(j,k)
   *x_(j,k)+v_(j,1)*zzer_(1,k)+v_(j,2)*zzer_(2,k));j=3:4;k=2:20
6: (YF_(k)*(zF_(1,k)+zF_(2,k))+V_(k-1)+L_(k+1)-L_(k)-V_(k)+v_(1,1)*zzer_(1,k)+v_(1,2)
   *zzer_(2,k)+v_(2,1)*zzer_(1,k)+v_(2,2)*zzer_(2,k)+v_(3,1)*zzer_(1,k)+v_(3,2)
   *zzer_(2,k)+v_(4,1)*zzer_(1,k)+v_(4,2)*zzer_(2,k));k=1:20
7: (x_(1,j) * a_(1)*exp(b_(1)*(T_(j)-t1_(1))/(T_(j)-t2_(1)))+ x_(2,j)*a_(2)*
   exp(b_(2)*(T_(j)-t1_(2))/(T_(j)-t2_(2)))+ x_(3,j) * a_(3)*exp(b_(3)*(T_(j)-t1_(3))
   /(T_(j)-t2_(3)))+ x_(4,j) * a_(4)*exp(b_(4)*(T_(j)-t1_(4))/(T_(j)-t2_(4))) - 1);j=1:20
8: (V_(j-1)-V_(j)-(dH_(1)*zzer_(1,j)+dH_(2)*zzer_(2,j))/lambda);j=1:20
9: B_(j)-(1-beta)*L_(1)*x_(j,1);j=1:4
10: xd_(j)*Dist+B_(j)-P_(j);j=1:4
11: Dist-V_20+L_21
12: x_(i,21)-xd_(i);i=1:4
13: xd_(i)-zzK_(i,20)*x_(i,20);i=1:4
14: beta-V_0/L_1
15: zzK_(i,k) - a_(i)*exp(b_(i)*(T_(k)-t1_(i))/(T_(k)-t2_(i)));i=1:4;k=1:20
16: zzer_(1,k) - (w_(k)*3.15e15*exp(-9547/T_(k))*x_(1,k)*x_(2,k));k=1:20
17: zzer_(2,k) - (w_(k)*6.3e15*exp(-9547/T_(k))*x_(1,k)*x_(3,k));k=1:20
18: zQb - beta*lambda*L_1
19: zQc - lambda*V_20
20: D^4 - Cd*beta^2*L_1^2
21: H_(k) - (Hmin+1.27*w_(k)/(0.3048)^3/D^2);k=1:20
22: height - (H_1+H_2+H_3+H_4+H_5+H_6+H_7+H_8+H_9+H_10+H_11+H_12+H_13+H_14+H_15+H_16+H_17
   +H_18+H_19+H_20)

```

• vp_fix.dat

```

1: 'P.CDR - vp_fix.dat '
2: % DISCRETE
3: YF_(k);k=1:20
4: % CONTINUOUS
5: zF_(i,k);i=1:2;k=1:20
6: beta
7: w_1;k=1:20

```


GAMS, N=10 Resultados obtidos com o GAMS para o caso NLP com ELV Ideal. Apresentam-se as especificações de uma coluna de custo mínimo, cujo projecto corresponde a uma coluna com dez andares andares e uma $F_{obj} = 1.516 \cdot 10^7$ \$/ano.

Tabela A.3 Resultados GAMS. Especificações da coluna de custo mínimo encontrada para um problema NLP com $N = 10$.

Prato	F_{1k} (mol/s)	F_{2k} (mol/s)	W_k m^3	T_k (K)	V_k mol/s	L_k mol/s
11	-	-	-	-	-	277.8
10	-	3.269	0.025	372.5	277.8	277.8
9	-	3.228	0.107	373.1	275.6	277.8
8	2.549	0.674	0.074	373.2	273.5	277.8
7	1.224	-	0.186	373.4	271.3	275.8
6	1.126	-	0.254	373.5	269.2	273.8
5	1.173	-	0.282	373.7	267.2	271.9
4	1.307	-	0.292	373.8	265.1	270.6
3	-	-	-	375.4	263.4	270.6
2	-	-	-	410.6	263.4	270.6
1	-	-	-	462.2	263.4	270.6
0	-	-	-	-	263.4	-

$D = 1.68$ m, $H = 9.69$ m, $\beta = 0.973$
 $Q_B = 10.54$ MW, $Q_C = 11.11$ MW

GAMS, N=20 Resultados obtidos com o GAMS para o caso NLP com ELV Ideal. Apresentam-se as especificações de coluna de custo mínimo, cujo projecto corresponde a uma coluna com vinte andares andares e uma $F_{obj} = 1.511 \cdot 10^7$ \$/ano.

Tabela A.4 Resultados GAMS. Especificações da coluna de custo mínimo encontrada para um problema NLP com $N = 20$.

Prato	F_{1k} (mol/s)	F_{2k} (mol/s)	W_k m^3	T_k (K)	V_k mol/s	L_k mol/s
21	-	-	-	-	-	277.8
20	-	1.492	0.009	372.2	277.8	277.8
19	-	1.491	0.067	373.1	276.8	277.8
18	-	1.466	0.072	373.1	275.8	277.8
17	-	1.467	0.050	373.1	274.8	277.8
16	0.232	1.230	0.037	373.1	273.8	277.8
15	1.469	-	0.033	373.1	272.9	277.8
14	1.140	-	0.053	373.3	271.9	277.5
13	0.478	-	0.103	373.4	270.9	276.5
12	0.544	-	0.109	373.4	270.0	275.6
11	0.477	-	0.118	373.5	269.0	274.7
10	0.563	-	0.117	373.5	268.1	273.8
9	0.542	-	0.132	373.6	267.2	272.9
8	0.461	-	0.148	373.7	266.2	272.0
7	0.524	-	0.141	373.7	265.3	271.1
6	0.915	-	0.154	373.7	264.4	270.6
5	-	-	-	373.9	263.4	270.6
4	-	-	-	380.0	263.4	270.6
3	-	-	-	433.4	263.4	270.6
2	-	-	-	467.0	263.4	270.6
1	-	-	-	470.5	263.4	270.6
0	-	-	-	-	263.4	-

$D = 1.68$ m, $H = 15.84$ m, $\beta = 0.974$
 $Q_B = 10.54$ MW, $Q_C = 11.11$ MW

GAMS, N=7 Resultados obtidos com o GAMS para o caso NLP com ELV Ideal. Apresentam-se as especificações de coluna de custo mínimo, cujo projecto corresponde a uma coluna com sete andares e uma $F_{obj} = 1.523 \cdot 10^7$ \$/ano. Incluem-se igualmente os perfis de temperatura e composições dos dois reagentes ao longo da coluna.

Tabela A.5 Resultados GAMS. Especificações da coluna de custo mínimo encontrada para um problema NLP com $N = 7$.

Andar	F_{1k} (mol/s)	F_{2k} (mol/s)	W_k m^3	T_k (K)	V_k mol/s	L_k mol/s
8	-	-	-	-	-	277.8
7	-	5.676	0.072	372.9	277.8	277.8
6	3.255	1.521	0.134	373.2	274.0	278.0
5	1.937	-	0.407	373.6	270.3	273.4
4	2.231	-	0.501	373.8	266.7	270.5
3	-	-	-	374.9	263.3	270.5
2	-	-	-	404.6	263.3	270.5
1	-	-	-	460.0	263.3	270.5
0	-	-	-	-	263.3	-

$D = 1.68$ m, $H = 7.82$ m, $\beta = 0.973$
 $Q_B = 10.5$ MW, $Q_C = 11.1$ MW

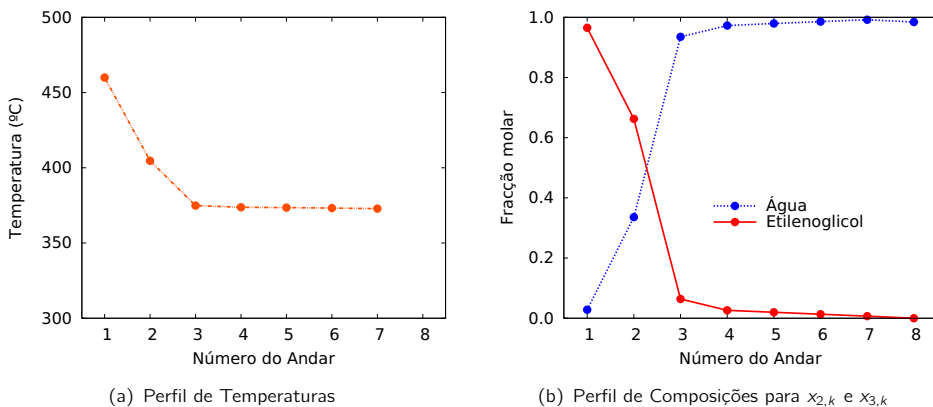


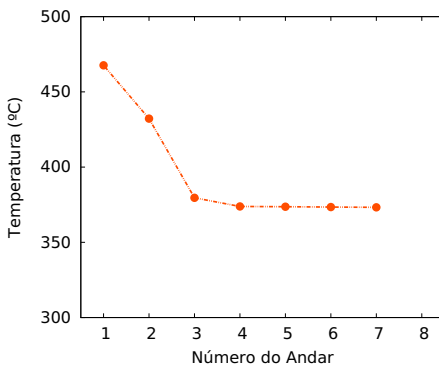
Figura A.7 Resultados GAMS. Perfis de temperatura e composições de Água e Etilenoglicol ao longo coluna de custo mínimo especificada.

ASCEND, N=7 Resultados obtidos com o ASCEND para o caso NLP com ELV Ideal. Apresentam-se as especificações de coluna de custo mínimo, cujo projecto corresponde a uma coluna com sete andares e uma $F_{obj} = 1.534 \cdot 10^7$ \$/ano. Incluem-se igualmente os perfis de temperatura e composições dos dois reagentes ao longo da coluna.

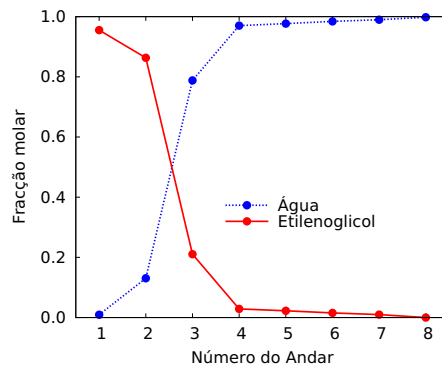
Tabela A.6 Resultados ASCEND. Especificações da coluna de custo mínimo encontrada para um problema NLP com $N = 7$.

Andar	F_{1k} (mol/s)	F_{2k} (mol/s)	W_k m^3	T_k (K)	V_k mol/s	L_k mol/s
8	-	-	-	-	-	277.6
7	2.334	1.259	0.799	373.3	277.6	273.0
6	1.770	4.308	0.553	373.4	272.2	274.3
5	2.060	0.499	1.013	373.7	269.0	271.0
4	1.294	0.000	0.702	373.9	265.2	269.2
3	0.000	0.051	1.213	379.6	263.1	269.2
2	0.001	1.011	0.527	432.2	263.1	270.2
1	0.001	0.141	1.545	467.7	263.1	270.4
0	-	-	-	-	263.1	-

$D = 1.68$ m, $H = 10.20$ m, $\beta = 0.973$
 $Q_B = 10.5$ MW, $Q_C = 11.1$ MW



(a) Perfil de Temperaturas



(b) Perfil de Composições para $x_{2,k}$ e $x_{3,k}$

Figura A.8 Resultados ASCEND. Perfis de temperatura e composições de Água e Etilenoglicol ao longo coluna de custo mínimo especificada.

SIMOP, N=8

Resultados obtidos com o SIMOP para o caso MINLP com ELV Ideal. Especificações da coluna de custo mínimo, cujo projecto corresponde a uma coluna com oito andares e uma $F_{obj} = 1.502 \cdot 10^7$ \$/ano. Incluem-se igualmente os perfis de temperatura e composições dos dois reagentes ao longo da coluna.

Tabela A.7 Resultados SIMOP. Especificações da coluna de custo mínimo encontrada para um problema MINLP.

Andar	$Y_{F,k}$	F_{1k} (mol/s)	F_{2k} (mol/s)	W_k m^3	T_k (K)	V_k mol/s	L_k mol/s
9	0	-	-	-	-	-	264.0
8	1	1.357	4.291	1.780	373.2	264.0	263.9
7	1	2.256	2.593	0.684	373.3	260.2	263.1
6	1	1.956	0.223	1.606	373.5	256.5	259.7
5	1	1.737	0.006	1.721	373.6	252.7	256.8
4	0	-	-	0.003	375.1	249.7	256.8
3	0	-	-	0.846	409.3	249.7	256.8
2	0	-	-	0.757	461.8	249.7	256.8
1	0	-	-	1.112	470.5	249.7	256.8
0	-	-	-	-	-	249.7	-

$D = 1.72$ m, $H = 9.60$ m, $\beta = 0.980$
 $Q_B = 10.5$ MW, $Q_C = 11.1$ MW

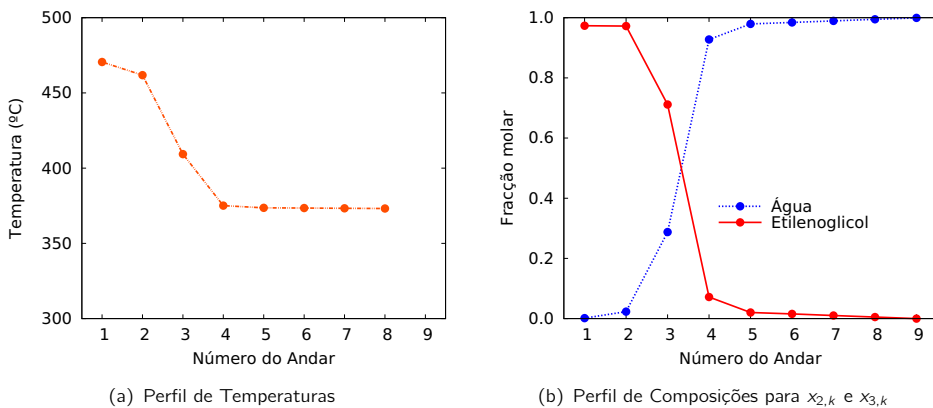


Figura A.9 Resultados SIMOP. Perfis de temperatura e composições de Água e Etilenoglicol ao longo coluna de custo mínimo especificada.

Apêndice B

Sobre o ASCEND

Apresentam-se informações sobre a utilização do ASCEND em *developer mode*. Opção não abrangida pela documentação da aplicação.

São também disponibilizadas versões do código-fonte de alguns dos procedimentos utilizados no estabelecimento da estratégia de simulação e optimização que utiliza os optimizadores MSGA e MSIMPISA.

B.1 Ficheiro `.ascendrc`

Quando a aplicação ASCEND é executada, o ficheiro de configurações de utilizador, `.ascendrc`, é carregado pela aplicação. Nele estão incluídas instruções que são automaticamente interpretadas, podendo o utilizador criar os seus próprios procedimentos e até simular modelos logo no arranque da aplicação.

No âmbito da estratégia conjugada proposta nesta tese, o ficheiro `.ascendrc` utiliza-se para:

- executar o ASCEND em *developer mode*, activando para isso a variável binária `ascScripVect(developer)`.
- O procedimento `RESOLVE_NO_GUI` que permite executar as simulações sem actualização das informações dos modelos na GUI.
- O carregamento de `optprocedures.a4s`, ficheiro que contém o conjunto de procedimentos que suporta a ligação do ASCEND com os algoritmos estocásticos.

Código Fonte do `.ascendrc`

```
1: # This is the .ascendrc file for running Ascend4
2: # a copy of it should be put in each user's startup directory.
3: # This file is called by the ascend binary.
4: #
5: # $Date: 98/06/01 12:42:45 $
6: # $Revision: 1.9 $
7: #
8: global env tcl_platform
9: source "$env(ASCENDTK)/ascend.tcl"
10: # ----- do not change anything above this line -----
11:
12: # Run ASCEND in developer mode by default
13: set ascScripVect(developer) 1
14:
15: # -----
16: # Ben Allan, Livermore, California, Feb 10, 2004
17: # GPL
18: #
19: # This feature is not loaded in ascend by default,
20: # as it breaks the design of the GUI in fundamental ways--
21: # it deprives the interactive user of immediate feedback.
22: #
23: # A new script function to handle re-solving efficiently
24: # for QRSLV.
25: # This function can be used to re-solve a model already
26: # in the solver if and only if:
27: # 1) The model has been solved (at least partially) once.
28: # 2) The .fixed flags on variables and .included flags on relations
29: #    have not been changed. (Thus the matrix structure is unchanged).
30: # 3) The choice of objective is not changed.
31: #
32: # We are allowed to change:
33: #   Bounds on variables (.lower_bound, .upper_bound)
34: #   Scaling values on variables (.nominal)
35: #   Values of input variables (those with .fixed == true)
36: #   Numerical solver controls like convergence tolerances.
```



```

37: #
38: # Side effects:
39: # This function updates Tcl variables related to the solver,
40: # but does not update the GUI. To update the GUI, one will
41: # have to hit the GUI Solve button or call the script SOLVE again.
42: #
43: # Unlike other solver processes, the tracing functions
44: # are not used, in the interest of speed. (entertrace, leavetrace
45: # commented out).
46: # The GUI interrupt is not going to work.
47: # The solver retains control for all iterations, with no tcl
48: # in between blocks.
49: #
50: # proc RESOLVE_NO_GUI {}
51: #-----
52: # Execute resolve internals
53: #-----
54: proc RESOLVE_NO_GUI {} {
55: # entertrace
56:   if {[slv_checksyst]} { return }
57:   # needs better sanity checking and interrupt checking.
58:   global ascSolvVect ascSolvStatVect ascSolv32767Vect ascSolv3Vect
59:   set ascSolvStatVect(menubreak) 0
60:   slv_set_haltflag 0
61:   if {[slv_checksyst]} {
62:     set ascSolvStatVect(ready2solve) 1
63:   }
64:   Solve_Downdate_ParmBox
65:   slv_resolve
66:   slv_solve
67:   Solve_Update_StatVect
68:   # puts "converged= $ascSolvStatVect(converged)"
69:   # puts "max err= $ascSolvStatVect(maxblockerr)"
70: #leavetrace
71: }
72:
73: #
74: # Force an update of the gui
75: # after the resolve is all over.
76: #
77: proc RESOLVE_GUI {} {
78:   RESOLVE_NO_GUI
79:   Solve_Update_StatusBox 1
80: }
81:
82: #
83: # Example of intended use (pseudo code):
84: # 1) SOLVE sim;
85: # 2) set someFixedVariable to a new value;
86: # RESOLVE_NO_GUI
87: # if {[ascSolvStatVect(converged)]} { die } else { do something else }
88:
89: # -----
90: # Optimization procedures loading for MSGA and MSIMPSPA coupling
91: # -----
92:
93: proc LOAD_OPTPROCEDURES {} {
94: source "/home/hgsilva/ascddata/optimizers/optprocedures.a4s"
95: }
96:
97: proc RELOAD_OPTPROCEDURES {} {
98: LOAD_OPTPROCEDURES
99: }
100:
101: LOAD_OPTPROCEDURES
102:

```

B.2 Navegador de variáveis internas

Com a activação do *developer mode*, o ASCEND disponibiliza um menu de funções e variáveis internas, navegável, apresentado na Figura B.1. Através deste painel o utilizador pode saber quais as variáveis internas associadas aos parâmetros de configuração do simulador.

Podem também ser efectuadas pesquisas no domínio das variáveis utilizadas para representar os diferentes estados do ambiente de simulação. Por exemplo, a variável `ascSolvStatVect(ok)` permite averiguar o estado de uma simulação, sabendo-se assim se ocorreu, ou não, convergência; a variável `iterationlimit` guarda o número máximo de iterações do `SOLVER`.

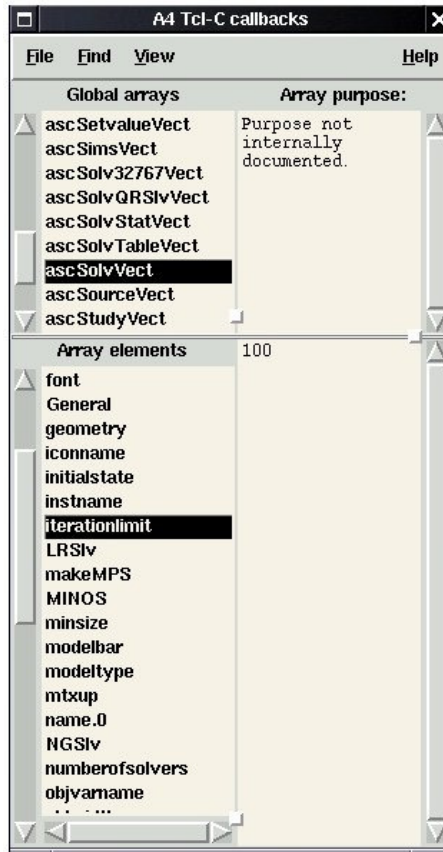


Figura B.1 ASCEND. Janela das variáveis internas.

Apêndice C

Estratégia integrada

Na Secção 4.3 é apresentada a estrutura final definida para o esquema de comunicação implementado entre o simulador e os optimizadores. Os procedimentos na base da sua implementação constituem um *script* carregado logo no arranque do simulador, o `optprocedures.a4s`.

C.1 Ficheiro optprocedures.a4s

```

1: # =====
2: #           O P T P R O C E D U R E S . A 4 S
3: # =====
4:
5: # ERROR HANDLING
6: #-----
7:
8: proc ERROR {type} {
9:     global stopflag
10:    switch $type {
11: : 001 {
12: :     puts ":: ERROR :::::::::::::::::::: 001 ::"
13: :     puts "    Define Optimizer in script.dat:"
14: :     set stopflag 1
15: : }
16:
17: : 002 {
18: :     puts ":: ERROR :::::::::::::::::::: 002 ::"
19: :     puts "    Define Problem Type in script.dat:"
20: :     puts "        0 -> Steady State"
21: :     puts "        1 -> Dynamic"
22: :     set stopflag 1
23: : }
24:
25: :     003 {
26: :     puts ":: ERROR :::::::::::::::::::: 003 ::"
27: :     puts "    Define MINLP FLAG in script.dat:"
28: :     puts "        0 -> NLP"
29: :     puts "        1 -> MINLP"
30: :     set stopflag 1
31: : }
32: : }
33: }
34:
35: # SIMULATOR CONFIGURATION
36: #-----
37:
38: proc SETASCEND {} {
39:     global ascBrowVect ascLibrVect
40:
41:     # Perform dimensional consistency
42:     set ascBrowVect(dimconsistency) 1
43:     set ascBrowVect(TypeorValue) VALUE
44:
45:     # Show all parser warnings
46:     set ascLibrVect(parserWarnings) 1
47: }
48:
49: # SOLVER CONFIGURATION
50: #-----
51:
52: proc DEFAULTSLV {value} {
53:     global ascSolvVect ascSolvQRSlvVects tcl_precision
54:     # INTERPRETER precision (number of significant figures)

```

```
55:     set tcl_precision $value
56:
57:     # ASCEND precision (number of significant figures)
58:     u_setprec $value
59:
60:     # Automatic pop-up solver report (yes = 1, no = 0)
61:     set ascSolvVect(statreport) 0
62:
63:     # Enable safe calculations (yes = 1, no = 0)
64:     set ascSolvQRSlvVect(safe_calc) 1
65:
66:     # Truncate whole step vector (yes = 1, no = 0)
67:     set ascSolvQRSlvVect(btrunc) 0
68: }
69:
70: #   READ SCONF.DAT
71: #-----
72:
73: proc READSCRIPTDATA {} {
74:
75:     global ndim optexe modelfile modeltype dyprob nstep lstep ustep
76:     global ascSolvQRSlvVect ascSolv32767Vect minlpflag modelsarray
77:     set ndim 0
78:     set optexe optmsimpsa
79:     set modelfile model.a4c
80:     set modeltype model
81:     set dyprob 0
82:     set nstep 10
83:     set lstep 1
84:     set ustep hr
85:
86:     set datafile [open sconf.dat r]
87:     gets $datafile text
88:     gets $datafile text
89:     gets $datafile optname
90:     gets $datafile text
91:
92:     set optname [string tolower $optname]
93:     set optexe "./opt"
94:     append optexe $optname
95:
96:     if {[string compare $optname msimpsa] == 0} {
97: :
98: : set datafilename optinesa.dat
99: :
100: : set optdatafile [open $datafilename r]
101: : gets $optdatafile text
102: : gets $optdatafile text
103: : gets $optdatafile text
104: : gets $optdatafile text
105: : gets $optdatafile fdatafilename
106: : close $optdatafile
107: :
108: : set fdatafilename [string trimright $fdatafilename ']
109: : set fdatafilename [string trimleft $fdatafilename ']
110: : set fdatafilename [lindex $fdatafilename 0]
```

```
111: :
112: : set fdatafile [open $datafilename r]
113: : gets $fdatafile text
114: : gets $fdatafile values
115: : close $fdatafile
116: :
117: : set ndim [lindex $values 0]
118: :
119: :   } elseif {[string compare $optname msga] == 0} {
120: :
121: : set datafilename optim.dat
122: :
123: : set optdatafile [open $datafilename r]
124: : gets $optdatafile text
125: : gets $optdatafile text
126: : gets $optdatafile values
127: : close $optdatafile
128: :
129: : set ndim [lindex $values 0]
130: :
131: :   } else {
132: :
133: : ERROR 001:
134: :   }
135: :
136: :   gets $datafile modelfile
137: :   set modeltype [exec grep MODEL $modelfile]
138: :   set modeltype [lindex $modeltype 1]
139: :
140: :   gets $datafile text
141: :   gets $datafile minlpdata
142: :
143: :   set minlpflag [lindex $minlpdata 0]
144: :
145: :   if {$minlpflag == 1} {
146: :
147: :     set dim [llength $minlpdata]
148: :     set modelsarray [lrange $minlpdata 1 [expr $dim -1]]
149: :
150: :   } elseif {$minlpflag == 0} {
151: :   } else {
152: :     ERROR 003
153: :   }
154: :
155: :   gets $datafile text
156: :   gets $datafile parameters
157: :
158: :   set ascSolvQRSlvVect(iterationlimit) [lindex $parameters 0]
159: :   set ascSolvQRSlvVect(singtol) [lindex $parameters 1]
160: :   set ascSolvQRSlvVect(feastol) [lindex $parameters 2]
161: :
162: :   gets $datafile text
163: :   gets $datafile dyprob
164: :
165: :   if {$dyprob == 1} {
166: : gets $datafile text
```

```

167: : gets $datafile dydata
168: : set nstep [lindex $dydata 0]
169: : set lstep [lindex $dydata 1]
170: : set ustep [lindex $dydata 2]
171:     set ascSolv32767Vect(moststeps) [lindex $dydata 3]
172: : close $datafile
173:     } elseif {$dyprob == 0} {
174: : close $datafile
175:     } else {
176: : ERROR 002
177:     }
178: }
179:
180: # DYNAMIC PROBLEMS DATA
181: #-----
182:
183: proc DYNAMIC {nstep lstep ustep} {
184:     global env:
185:     source "$env(ASCENDDIST)/models/set_intervals.tcl"
186:     RUN {sim.set_ode}
187:     RUN {sim.set_obs}
188:     set brace \{
189:     set ustep [append brace $ustep]
190:     set ustep [append ustep \}]
191:     set_int $nstep $lstep $ustep
192: }
193:
194: # =====
195:
196:
197: # =====
198: #                               MAIN SCRIPT
199: # =====
200:
201: proc MAIN {} {
202:
203:     global ndim optexe modelfile modeltype dyprob nstep lstep ustep
204:     global sproc ascSolvStatVect vlist validsim
205:     global ascSolvVect ascSolvQRSlvVect ascSolv32767Vect
206:     global ascBrowVect ascLibrVect
207:     global stopflag
208:     global minlpflag modelsarray
209:
210:     DELETE TYPES
211:     READ FILE "$modelfile"
212:
213:     if {$minlpflag == 0} {
214:         COMPILE sim OF $modeltype
215:         RUN {sim.reset}
216:         RUN {sim.default_self}
217:         RUN {sim.bound_self}
218:     } elseif {$minlpflag == 1} {
219:         foreach i $modelsarray {
220:             COMPILE $i OF $i
221:         }
222:     }

```



```
223:     }
224:
225:     # -- browsing disable to save time --
226:     # BROWSE {sim};
227:     # -----
228:
229:     if {$dyprob == 1} {
230: : DYNAMIC $nstep $lstep $ustep
231:     }
232:
233:     #
234:     # == STARTING OPTIMIZATION =====
235:
236:     set optimizer [open "|$optexe" "w+"]
237:
238:     gets $optimizer vlist
239:     set vflag [llength $vlist]
240:
241:     while {$vflag == $ndim} {
242:
243: : set inf [RUNSIM]
244:
245: #: puts "fob = $inf"
246:
247: : puts $optimizer $inf
248: : flush $optimizer
249:
250: : gets $optimizer vlist
251: #         puts "main: $vlist"
252: : set vflag [llength $vlist]
253:     }
254:
255:     close $optimizer
256:
257:     puts "Problem Solved"
258:     puts $validsim
259: }
```


Apêndice D

Noções complementares

D.1 Programação matemática de expressões lógicas

Neste apêndice apresentam-se as noções básicas associadas à transcrição de condições lógicas sob a forma de variáveis binárias discretas. O conteúdo é essencialmente baseado nas notas de Biegler e Grossmann (2007), podendo-se encontrar mais informações em Floudas (1995) e Biegler et al. (1997), que por sua vez citam Williams (1988)

Em termos da lógica proposicional, à acção de escolha da unidade i ou execução da tarefa j poderá associar-se a condição lógica P_i que assume os valores de verdadeiro, ou falso. O acontecimento contrário ao determinado pela condição P_i é considerado a sua negação, podendo ainda ser estabelecidas relações de intercepção, reunião ou opção entre cada uma das condições, operadores lógicos representados pelos seguintes símbolos:

Operador lógico	Notação
ou	\vee
e	\wedge
ou exclusivo	\oplus
negação	\neg
implicação	\Rightarrow

O operador implicação pode ser representado de forma alternativa, pelo que quando se diz que a condição P_1 implica P_2 , $P_1 \Rightarrow P_2$, tal situação pode ser representada pela expressão $\neg P_1 \vee P_2$.

A representação de restrições de desigualdade lineares que modelizam a ocorrência das condições é feita com base em duas atribuições. Quando a condição P_i é verdadeira,

isso corresponde a dizer que a variável binária associada será unitária, $y_i = 1$, assumindo o valor nulo a condição P_i for falsa.

As relações existentes entre as condições (proposições) poderão assim ser modelizadas da seguinte forma:

1. Opção entre condições, escolhendo, pelo menos, uma delas.

$$P_1 \vee P_2 \vee \dots \vee P_r \quad (\text{D.1})$$

representa-se por:

$$y_1 + y_2 + \dots + y_r \geq 1 \quad (\text{D.2})$$

2. Ocorrência simultânea das condições.

$$P_1 \wedge P_2 \wedge \dots \wedge P_r \quad (\text{D.3})$$

representa-se por:

$$\begin{cases} y_1 \geq 1 \\ y_2 \geq 1 \\ \dots \\ y_r \geq 1 \end{cases} \quad (\text{D.4})$$

3. Implicação, $P_1 \Rightarrow P_2$

$$\neg P_1 \vee P_2 \quad (\text{D.5})$$

representa-se por

$$1 - y_1 + y_2 \geq 1, \text{ o mesmo que, } y_2 \geq y_1 \quad (\text{D.6})$$

4. Equivalência, $P_1 \Leftrightarrow P_2 \rightarrow (P_1 \Rightarrow P_2) \vee (P_2 \Rightarrow P_1)$

$$(\neg P_1 \vee P_2) \wedge (\neg P_2 \vee P_1) \quad (\text{D.7})$$

representa-se por

$$\begin{cases} 1 - y_1 + y_2 \geq 1 \\ 1 - y_2 + y_1 \geq 1 \end{cases} \Leftrightarrow \begin{cases} y_1 \leq y_2 \\ y_1 \geq y_2 \end{cases} \Leftrightarrow y_1 = y_2 \quad (\text{D.8})$$

5. Escolha de apenas uma das condições

$$P_1 \oplus P_2 \oplus \dots \oplus P_r \quad (\text{D.9})$$

representa-se por

$$y_1 + y_2 + \dots + y_r = 1 \quad (\text{D.10})$$

6. Implicação antecedida de opção, $P_1 \vee P_2 \Rightarrow P_3$

$$(\neg P_1 \vee P_3) \wedge (\neg P_2 \vee P_3) \quad (\text{D.11})$$

representa-se por

$$\left. \begin{array}{l} 1 - y_1 + y_3 \geq 1 \\ 1 - y_2 + y_3 \geq 1 \end{array} \right\} \begin{array}{l} y_3 \geq y_1 \\ y_3 \geq y_2 \end{array} \quad (\text{D.12})$$

Com base nas representações apresentadas, a escrita das restrições passa pela colocação das proposições na forma conjuntiva normal, dada pela proposição D.13, na qual a condição Q_i corresponde a $P_1 \vee P_2 \vee \dots \vee P_r$.

$$Q_1 \wedge Q_2 \wedge \dots \wedge Q_s \quad (\text{D.13})$$

O procedimento sistemático para derivar desigualdades lineares é composto por três passos (Clocksin e Mellish, 1981):

1. Substituição das implicações por disjunções

$$P_1 \Rightarrow P_2 \Leftrightarrow \neg P_1 \vee P_2 \quad (\text{D.14})$$

2. Aplicação da propriedade distributiva da operação negação

$$\neg(P_1 \wedge P_2) \Leftrightarrow \neg P_1 \vee \neg P_2 \quad (\text{D.15})$$

$$\neg(P_1 \vee P_2) \Leftrightarrow \neg P_1 \wedge \neg P_2 \quad (\text{D.16})$$

3. Distribuir recursivamente “ou” sobre “e”

$$(P_1 \wedge P_2) \vee P_3 \Leftrightarrow (P_1 \vee P_3) \wedge (P_2 \vee P_3) \quad (\text{D.17})$$