

Faculdade de Engenharia da Universidade do Porto



**Problemas de corte de peças com incerteza na
qualidade da matéria-prima - uma abordagem
com algoritmos genéticos**

Inês Pinto Coimbra Queirós

Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Orientador: Beatriz Brito Oliveira

30 de Julho de 2019

© I. Queirós, 2019

Resumo

Em ambiente industrial existe a necessidade de controlar o desperdício, para maximizar a rentabilidade dos processos, não provocando custos excessivos e não planeados na realização dos processos.

Nos processos industriais que envolvem cortes de objetos grandes em itens mais pequenos, é importante cortar o número de objetos certo com os padrões de corte adequados, de forma a que a indústria não seja prejudicada ficando com excesso de itens que não necessita ou não tendo itens suficientes para corresponder à procura. No entanto, mesmo que os itens cortados correspondam exatamente à procura, existe a possibilidade de a matéria-prima apresentar defeito, o que inutiliza um certo número de peças. A possibilidade de a matéria-prima apresentar defeitos não é, por norma, considerada no planeamento de cortes.

Este trabalho tem como objetivo estudar um problema de cortes bidimensionais em ambiente industrial, considerando que existe incerteza na qualidade da matéria-prima.

Para resolver este problema foi utilizado o Algoritmo Genético BRKGA (*Biased Random-Key Genetic Algorithm*), com a adaptação do decodificador. Este algoritmo é uma metaheurística que permite obter boas soluções em curtos intervalos de tempo.

De modo a analisar o impacto da presença de incerteza no problema em relação à qualidade da matéria-prima, foram gerados vários conjuntos de cenários para a percentagem de área de cada objeto inutilizada por defeitos. Cada conjunto de cenários foi associado a uma diferente parametrização da distribuição de probabilidades Beta, representando assim diferentes tipos de distribuições, como a Normal, monótona crescente e decrescente, entre outras. Foi realizada uma análise às soluções geradas para cada cenário, de forma a entender o impacto destas distribuições de probabilidades no problema.

Pode verificar-se que o BRKGA, como seria de esperar, apresenta boas soluções em espaços de tempo curtos. A utilização da distribuição de probabilidades Beta como gerador de cenários para criar incerteza permitiu conhecer de forma mais detalhada o impacto da incerteza neste problema.

Abstract

In an industrial environment, there is a need to control waste to maximize process profitability, without causing excessive and unplanned costs to perform the processes.

In industrial processes that involve cutting large objects into smaller items, it is important to cut the right number of objects to the appropriate cutting patterns, so that the industry is not harmed by getting too many items that it does not need or by not having enough to meet demand. However, even if the cut items match the exact demand, there is a possibility that the raw material may be defective, which makes a number of parts unusable. The possibility that the raw material may be defective is not normally considered in cutting problems.

This paper aims to study a two-dimensional cutting problem in an industrial environment, considering that there is uncertainty in the quality of the raw material.

To solve this problem, the Biased Random-Key Genetic Algorithm (BRKGA) was used, by adapting the decoder procedure. This algorithm is a metaheuristic that allows to obtain good solutions in short time periods.

In order to analyze the impact of the presence of uncertainty on the problem regarding the quality of the raw material, several sets of scenarios were generated for the percentage of area of each object rendered unusable by defects. Each set of scenarios was associated with a different parameterization of the Beta probability distribution, thus representing different types of distributions, such as Normal, monotonous increasing and decreasing, among others. An analysis was made of the solutions generated for each scenario in order to understand the impact of these probability distributions on the problem.

It can be seen that BRKGA, as expected, achieves good solutions in short time periods. Using the Beta probability distribution as a scenario generator to create uncertainty allowed to further understand the impact of uncertainty in this problem.

Agradecimentos

Em primeiro lugar agradeço à Professora Beatriz Brito Oliveira, por toda a disponibilidade ao longo do desenvolvimento deste trabalho, pela motivação e positivismo.

Agradeço ao Professor José Fernando Oliveira por nunca me ter fechado portas e me ter dado a oportunidade de desenvolver este trabalho numa área que é do meu interesse.

Agradeço aos meus pais e à minha irmã pelo apoio incondicional ao longo deste percurso cheio de altos e baixos.

Ao meu sobrinho agradeço pela alegria constante, pela energia e motivação que me dá para ser um exemplo melhor.

Por fim agradeço ao Rúben todo o lado racional e todo o suporte emocional.

Índice

Resumo.....	iii
Abstract.....	v
Agradecimentos	vii
Índice.....	ix
Lista de figuras	x
Lista de tabelas.....	xii
Abreviaturas e Símbolos	xiii
Capítulo 1. Introdução	1
1.1. Descrição do Problema.....	2
1.2. Modelo Matemático.....	2
Capítulo 2. Revisão de Literatura.....	5
2.1. Problemas de Corte e Empacotamento	5
2.2. Incerteza	8
2.2.1. Problemas de corte com incerteza	9
2.2.2. Cenários	9
2.3. Metodologia	11
2.3.1. Algoritmos Genéticos	11
2.3.1. BRKGA	16
Capítulo 3. Desenvolvimento do Método de Resolução.....	19
3.1. BRKGA adaptado ao problema - <i>decoder</i>	19
3.2. Cenários	20
Capítulo 4. Testes e Resultados	23
4.1. Instâncias.....	23
4.2. Análise de sensibilidade aos parâmetros que controlam o elitismo e a aleatoriedade do BRKGA	24
4.3. Análise da solução face aos cenários	28
Capítulo 5. Conclusão	37
Referências.....	38
Apêndices	40

Lista de figuras

Figura 1. Exemplo de um padrão de corte.	3
Figura 2. Tipos de problemas básicos (Wäsher et al. (2007)).	8
Figura 3. Estrutura de um algoritmo genético (baseada em [18]).	13
Figura 4. Evolução da população entre gerações consecutivas no BRKGA [22].	16
Figura 5. <i>Framework</i> do BRKGA (baseado em [24]).	17
Figura 6. Funcionamento do decodificador (exemplo).	19
Figura 7. Função densidade de probabilidade para a distribuição Beta [23].	21
Figura 8. Gráfico de convergência da instância grande para a solução final durante 1000 gerações.	27
Figura 9. Gráfico de convergência da instância pequena para solução final durante 1000 gerações.	27
Figura 10. Valores dos componentes da função objetivo para cada cenário pertencente ao conjunto gerado pela distribuição $\alpha = 0.5$; $B = 0.5$ à esquerda e $\alpha = 1$; $B = 3$ à direita, e valor da função objetivo se não considerada incerteza.	28
Figura 11. Valores dos componentes da função objetivo para cada cenário pertencente ao conjunto gerado pela distribuição $\alpha = 2$; $B = 2$ à esquerda e $\alpha = 2$; $B = 5$ à direita, e valor da função objetivo se não considerada incerteza.	28
Figura 12. Valores dos componentes da função objetivo para cada cenário pertencente ao conjunto gerado pela distribuição $\alpha = 5$; $B = 1$, e valor da função objetivo se não considerada incerteza.	29
Figura 13. Valores dos componentes da função objetivo para cada cenário pertencente ao conjunto gerado pela distribuição $\alpha = 0.5$; $B = 0.5$ à esquerda e $\alpha = 1$; $B = 3$ à direita, e valor da função objetivo se não considerada incerteza.	29
Figura 14. Valores dos componentes da função objetivo para cada cenário pertencente ao conjunto gerado pela distribuição $\alpha = 2$; $B = 2$ à esquerda e $\alpha = 2$; $B = 5$ à direita, e valor da função objetivo se não considerada incerteza.	29

Figura 15. Valores dos componentes da função objetivo para cada cenário pertencente ao conjunto gerado pela distribuição $\alpha = 5$; $\beta = 1$, e valor da função objetivo se não considerada incerteza.	30
Figura 16. Gráfico de padrões cortados na solução final obtida para cada cenário do conjunto gerado pela distribuição Beta(0.5,0.5).	31
Figura 17. Gráfico de padrões cortados na solução final obtida para cada cenário do conjunto gerado pela distribuição Beta(1,3).	31
Figura 18. Gráfico de padrões cortados na solução final obtida para cada cenário do conjunto gerado pela distribuição Beta(2,2).	32
Figura 19. Gráfico de padrões cortados na solução final obtida para cada cenário do conjunto gerado pela distribuição Beta(2,5).	32
Figura 20. Gráfico de padrões cortados na solução final obtida para cada cenário do conjunto gerado pela distribuição Beta(5,1).	33
Figura 21. Gráfico de padrões cortados na solução final obtida para cada cenário do conjunto gerado pela distribuição Beta(0.5,0.5).	33
Figura 22. Gráfico de padrões cortados na solução final obtida para cada cenário do conjunto gerado pela distribuição Beta(1,3).	34
Figura 23. Gráfico de padrões cortados na solução final obtida para cada cenário do conjunto gerado pela distribuição Beta(2,2).	34
Figura 24. Gráfico de padrões cortados na solução final obtida para cada cenário do conjunto gerado pela distribuição Beta(2,5).	35
Figura 25. Gráfico de padrões cortados na solução final obtida para cada cenário do conjunto gerado pela distribuição Beta(5,1).	35

Lista de tabelas

Tabela 1. Resultados da Função Objetivo obtidos na instância grande para os diferentes valores de p_e - probabilidade de herança do alelo de elite.	25
Tabela 2. Resultados da Função Objetivo obtidos na instância pequena para os diferentes valores de p_e - probabilidade de herança do alelo de elite.	26

Abreviaturas e Símbolos

Lista de abreviaturas (ordenadas por ordem alfabética):

AG	Algoritmo Genético
BRKGA	Biased random-key genetic algorithms
F.O.	Função Objetivo
RKGA	Random-key genetic algorithms

Capítulo 1. Introdução

Os problemas de corte e empacotamento são problemas de otimização combinatória que surgem como uma dificuldade recorrente em ambiente industrial e de logística. Estes problemas têm vindo a ser estudados desde 1960, após o aparecimento do trabalho de Kantorovich [1], e de forma mais exaustiva nas últimas décadas.

Os problemas de corte e empacotamento surgem quando se pretende cortar um determinado recurso em objetos menores ou inserir esse conjunto de pequenos itens num espaço maior, de forma a minimizar ou maximizar uma determinada função objetivo, otimizando o espaço, minimizando o desperdício de área em vazio ou número de recursos a cortar. São problemas reais de otimização combinatória, segundo Alvarez-Valdes, Carravilla e Oliveira (2018) muito recorrentes na *“indústria de corte de papel em rolos mais estreitos na indústria de papel, corte de grandes painéis de madeira em painéis retangulares menores ou corte de peças de forma irregular de rolos de tecido, mas também incluir caixas de embalagens em recipientes ou carregamentos de objetos em paletes em aplicações logísticas”*[2].

Para definir um problema de corte e empacotamento é necessário conhecer a matéria prima que se pretende cortar ou que contenha o empacotamento, a qual se denomina objeto, e o material que se pretende gerar ou armazenar, denominado item. Ao longo do processo de corte ou de empacotamento dá-se a construção de padrões, obtida através de combinações geométricas dos itens nos objetos.

Os problemas de cortes consistem em cortar peças maiores, denominadas objetos, transformando-as em peças mais pequenas, denominadas itens. Estes problemas têm como objetivo dar resposta a uma determinada procura, otimizando uma função objetivo.

Os problemas de empacotamento são também problema de otimização combinatória.

A relação entre este tipo de problemas é grande, pois os problemas de cortes representam a divisão de um espaço vazio que será ocupado por itens enquanto que nos problemas de empacotamento se pretende empacotar os itens no espaço que os objetos ocupam.

No desenvolvimento deste trabalho considerou-se o problema de corte bidimensional de peças com incerteza na qualidade da matéria-prima em placas com simetria, e padrões de corte pré-definidos.

Este trabalho é composto por cinco capítulos. Neste primeiro capítulo de introdução, é apresentado o problema e o modelo matemático proposto para o resolver. No segundo capítulo, é feito o estudo de literatura onde são abordados os problemas de corte e empacotamento, outros problemas que consideram incerteza, e a utilização de cenários neste tipo de problemas. É ainda também revista a metodologia utilizada para a resolução do

problema: os algoritmos genéticos e, em específico, o BRKGA. No terceiro capítulo é explicado o método de resolução proposto para o problema. O quarto capítulo contém a análise de resultados do problema, apresentando as instâncias utilizadas para a realização de testes, e os resultados da análise de sensibilidade e da análise da solução face aos cenários. Por fim, no quinto capítulo encontra-se a conclusão do trabalho.

1.1. Descrição do Problema

O problema estudado no desenvolvimento desta dissertação trata de um problema de corte bidimensional com incerteza na qualidade da matéria prima. Os problemas de corte na indústria têm sido largamente estudados nas últimas décadas, não sendo tão frequentemente estudado o impacto que a qualidade da matéria prima pode provocar no produto final. A incerteza na qualidade da matéria prima é uma situação que pode ocorrer recorrentemente, provocando desperdício de peças ou padrões inteiros, alterando os planos de produção, o número de placas cortadas, podendo provocar atrasos nos processos e custos superiores. Estas situações podem ocorrer em diversos tipos de indústrias, como de madeira, de papel, metalúrgica.

Os algoritmos para problemas de cortes bidimensionais já foram estudados por Beasley em 1985 [3], para o corte com guilhotina em indústrias de corte de chapas de aço, chapas de madeira para fazer móveis e no corte de papelão para caixas. Também Morabito e Arenales, em 1992 [4], propuseram uma abordagem diferente para os cortes com guilhotina, chamada *and-or graph*, tendo como referência os processos de produção das indústrias de vidro, aço, madeira ou papel. Já Alvarez-Valdés, Parajón e Tamarit, em 2002, [5] propuseram a metaheurística *Tabu Search* para problemas de cortes com guilhotina, apresentando resultados favoráveis,

Neste trabalho foi proposto um Algoritmo Genético evolutivo denominado *Biased Random Key Genetic Algorithm* para encontrar uma solução rápida, e adequada para o problema proposto.

1.2. Modelo Matemático

Com intenção de facilitar o entendimento e a construção do problema, desenvolveu-se um modelo matemático determinístico com base nas variáveis de decisão e nos parâmetros mais relevantes, para a resolução do problema. A Figura 1 apresenta-se para um melhor entendimento dos parâmetros e da sua relação, sendo um exemplo de padrão de corte onde são cortados quatro tipos de itens diferentes em diferentes quantidades, e ainda uma área de desperdício.

Índices

i - peça, $i \in \{1, \dots, P\}$
 j - padrão, $j \in \{1, \dots, T\}$

Parâmetros

P - número total de peças
 T - número total de padrões de corte
 δ - custo associado ao desperdício por 1 cm^2
 w_j - área (cm^2) de placa desperdiçada com o corte do padrão j
 θ_i - custo de produzir uma peça i a menos
 β_i - custo de produzir uma peça i a mais
 γ_j - custo de cortar uma placa com padrão j
 α - custo/peso da *soft constraint*
 a_{ij} - número de vezes que a peça i é cortada no padrão j (exemplificado na Figura 1)
 d_i - procura (em número de peças) da peça i

Variáveis de decisão

x_j - número de vezes que o padrão de corte T_j é cortado - $x_j \in Z_0^+$
 e_i - número de peças do tipo i em excesso - $e_i \in Z_0^+$
 f_i - número de peças do tipo i em falta - $f_i \in Z_0^+$

$i=1$	$i=1$	$i=1$			Padrão $j=1$ $a_{11} = 3$ $a_{21} = 2$ $a_{31} = 3$ $a_{41} = 1$ $w_1 = \text{desperdício}$
$i=2$		$i=2$			
$i=3$	$i=3$	$i=3$	$i=4$	w_1	

Figura 1. Exemplo de um padrão de corte.

Função Objetivo

$$F.O. \min \sum_j x_j \cdot \gamma_j + \delta \sum_j w_j \cdot x_j + \alpha \sum_i (\theta_i \cdot f_i + \beta_i \cdot e_i) \quad (1)$$

A função objetivo (1) é uma função de custos e pode ser dividida em três parcelas aditivas: FO_1 , FO_2 e FO_3 .

- $FO_1 = \sum_j x_j \cdot \gamma_j$; FO_1 representa o mínimo de cortar os padrões de corte o número de vezes necessárias;
- $FO_2 = \delta \sum_j w_j \cdot x_j$; FO_2 representa o custo associado ao desperdício de matéria prima dos padrões de corte efetivamente cortados;
- $FO_3 = \alpha \sum_i (\theta_i \cdot f_i + \beta_i \cdot e_i)$; FO_3 introduz o custo associado à produção de peças em

excesso ou em falta. Esta função apresenta uma *soft constraint* que penaliza a função objetivo por soluções em que haja excesso ou falta de peças, não impedindo de obter uma solução em que o número de peças seja diferente do pretendido, mas apenas penalizando essa situação. Uma *soft constraint* representa uma restrição que se pretende que seja atendida pelo modelo, mas que se permite que possa não ser para não comprometer a admissibilidade das soluções; é considerada uma restrição flexível.

Restrições

$$\sum_j a_{ij} \cdot x_j + f_i = d_i + e_i, \forall_i \quad (2)$$

A função de restrição (2) estabelece a relação entre a procura das peças do tipo i , as peças produzidas, e se tal resulta num excesso ou falta de peças.

Capítulo 2. Revisão de Literatura

Neste capítulo são apresentadas as diferentes tipologias dos Problemas de Corte e Empacotamento. É ainda introduzido o tema da Incerteza, apresentando-se problemas que consideram incerteza e introduzindo o conceito de cenário. Por fim, é revista a metodologia usada na realização da dissertação.

2.1. Problemas de Corte e Empacotamento

Desde a publicação dos primeiros artigos sobre os problemas de corte e empacotamento, em 1960 por Kantorovich, “*cujo trabalho original foi publicado em russo em 1939*” [6] o número de publicações e investigação sobre os problemas de corte e empacotamento tem vindo a aumentar. Também os métodos e técnicas de abordagem têm sido estudados no sentido de melhorar cada vez mais os resultados, com as abordagens heurísticas a apresentar bons resultados e vantagens significativas a nível dos tempos de processamento e do tamanho dos problemas que conseguem resolver.

O aumento dos trabalhos de investigação sobre os problemas de corte e empacotamento criou a necessidade de criar diferentes tipologias, sendo a primeira proposta por Dyckhoff (1990) [7]. Mais tarde, a tipologia proposta por Wäshel et al. (2007) [8] tornou-se bastante disseminada e utilizada (mais de 1300 citações no Google Scholar).

Tipologia de Dyckhoff (1990)

A tipologia de Dyckhoff divide os problemas de corte e empacotamento em quatro critérios diferentes, e atribui-lhes assim as seguintes características:

1. Dimensão
 - (1) Uma dimensão
 - (2) Duas dimensões
 - (3) Três dimensões
 - (4) N dimensões, com $N > 3$

2. Tipos de atribuição
 - (B) Todos os objetos e uma seleção de itens
 - (V) Uma seleção de objetos e todos os itens
3. Características dos objetos grandes
 - (O) Um objeto
 - (I) Formas idênticas
 - (D) Formas diferentes
4. Características dos itens pequenos
 - (F) Poucos itens (de formas diferentes)
 - (M) Muitos itens de muitas formas diferentes
 - (R) Muitos itens de relativamente poucas formas diferentes (não congruentes)
 - (C) Formas congruentes

O primeiro critério, a dimensão, representa o número mínimo de dimensões geométricas necessárias para descrever totalmente os padrões necessários (por exemplo a altura, a largura, o comprimento). O segundo critério refere-se ao tipo de atribuição de itens pequenos a objetos grandes. Os dois casos são distinguidos por Dyckhoff atribuindo as designações em alemão de (B) - *Beladeproblem*, a qual significa que todos os objetos grandes devem ser usados, ou seja, uma seleção de pequenos itens deve ser atribuída aos objetos grandes; e (V) - *Verladeproblem*, que caracteriza uma situação na qual todos os itens pequenos devem ser atribuídos a uma seleção de objetos grandes. O terceiro critério representa as características dos objetos grandes. (O) representa um objeto grande, (I) representa vários objetos grandes de formas idênticas, e (D) representa vários objetos grandes de formas diferentes. O quarto e último critério caracteriza os itens pequenos. Dyckhoff distingue os itens pequenos considerando (F) poucos itens de formas diferentes, (M) muitos itens de muitas formas diferentes, (R) muitos itens de relativamente poucas formas diferentes e (C) formas congruentes.

Tipologia de Wäsher et al. (2007)

Wäsher et al. considerou a tipologia apresentada por Dyckhoff útil para a classificação dos problemas de corte e empacotamento, no entanto com capacidades insuficientes. Em 2007, apresentou uma melhoria para caracterizar os problemas de corte e empacotamento com base na tipologia original de Dyckhoff, com novos critérios de categorização e diferentes categorias de problemas.

Os critérios de categorização usados por Wäsher et al. para definir os tipos de problemas combinados de corte e empacotamento são os cinco seguintes que se apresentam:

- 1) Dimensão - distingue-se entre problemas unidimensionais, bidimensionais e tridimensionais. Ocasionalmente, consideram-se problemas com mais de três dimensões geométricas.
- 2) Tipo de atribuição - introduzem-se duas situações básicas como Dyckhoff (1990), tendo em conta o objetivo final, mas a designação é alterada, pretendendo-se atingir uma maximização do output ou uma minimização do input.

- 3) Características de pequenos itens - são distinguidos três casos, itens pequenos idênticos (mesma forma e tamanho), itens pequenos pouco heterogêneos e itens pequenos muito heterogêneos.
- 4) Características de objetos grandes - um objeto grande, podendo a dimensão ser fixada em todas as dimensões ou ser variável em uma ou mais dimensões; vários objetos grandes, onde apenas dimensões fixas serão consideradas. Distinguem-se entre objetos grandes idênticos, objetos grandes pouco heterogêneos e objetos grandes muito heterogêneos.
- 5) Forma dos pequenos itens - distinguem-se pequenos itens regulares (retângulos, círculos, caixas, cilindros, bolas, etc.) e irregulares (também chamados de não regulares). No caso bidimensional, às vezes são distinguidos em itens retangulares, itens circulares e outros.

Cada tipologia de problemas de corte e empacotamento deve oferecer uma opção para caracterizar um determinado problema em relação ao número de dimensões relevantes para o problema. Portanto, são definidos três tipos de problemas, os problemas básicos, os problemas intermédios e os problemas refinados (*basic, intermediate and refined problem types*) Wäsher et al. (2007).

Os problemas do tipo básico são definidos combinando os critérios 2 e 3, ou seja, o tipo de atribuição dos itens ao objeto e as características dos pequenos itens. Este tipo de problemas é dividido em problemas do tipo de maximização de *output* e problemas do tipo de minimização de *input*. Os autores definiram três problemas básicos de corte e empacotamento para cada um deles. Para os problemas do tipo de maximização de *output* foram definidos o Problema de empacotamento de itens idênticos (*Identical Item Packing Problem*), o Problema de posicionamento (*Placement Problem*) e o Problema da mochila (*Knapsack Problem*). No caso dos problemas do tipo de minimização de *input* foram definidos o Problema de Dimensão Aberta (*Open Dimension Problem*), o Problema de corte (*Cutting Stock Problem*) e o Problema de empacotamento (*Bin Packing Problem*). A Figura 2 representa de forma esquemática os tipos de problemas básicos.

De modo a definir os tipos de problemas de forma mais homogênea, os tipos básicos de problemas são divididos em problemas do tipo intermédio, sendo os problemas do tipo intermédio gerados através da combinação dos problemas do tipo básico com o quarto critério, o qual caracteriza os objetos grandes. Os problemas do tipo refinado são a combinação dos problemas do tipo intermédio com o primeiro e quinto critérios: a dimensão e a forma dos pequenos itens.

Segundo a tipologia de Wäsher et al., o problema tratado nesta dissertação é um problema de dimensão bidimensional, sendo o tipo de atribuição minimização de *input*. Nas características de itens pequenos, o problema enquadra-se nas três situações e nas características de objetos grandes também pode ser enquadrada nas três situações, pois não se geram os padrões. Quanto à forma dos itens, também se aplica a todos os casos.

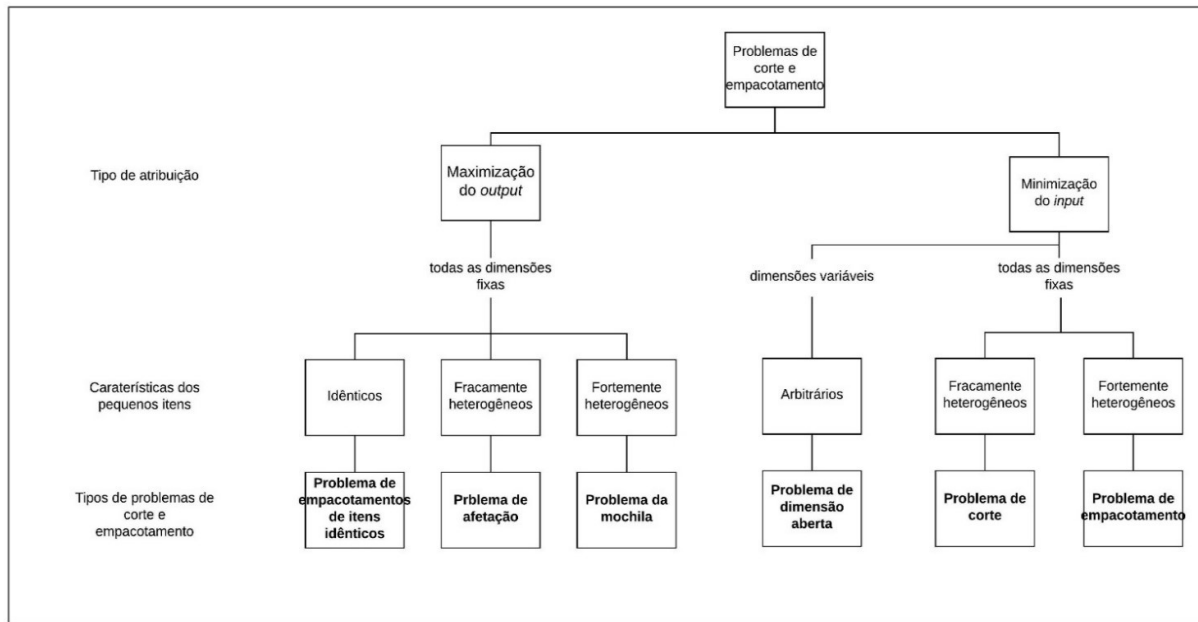


Figura 2. Tipos de problemas básicos (Wäsher et al. (2007)).

2.2. Incerteza

A maioria dos ambientes industriais com produção, caracterizam-se por existirem vários tipos de incertezas. As quantidades produzidas nem sempre correspondem às quantidades procuradas, pois a procura não é conhecida. Esse valor de incerteza pode afetar o planeamento e controlo de produção. No caso específico onde matéria-prima é dividida em diferentes padrões de corte, o estado físico da matéria-prima é muitas vezes desconhecido para o operador, sendo difícil garantir o rendimento exato de cada padrão cortado. O planeamento de produção pretende determinar a quantidade ideal de padrões cortados para responder aos pedidos de forma otimizada, diminuindo os custos tanto por excesso de peças como por falta, tendo em consideração essa possibilidade de problemas de qualidade na matéria-prima.

Recentemente, a incerteza já foi abordada na literatura para vários problemas, como por exemplo problemas de agendamento. Para este tipo de problemas, por exemplo, Li e Ierapetritou [9] compararam um método otimização robusta com o método de programação estocástica tradicional.

2.2.1. Problemas de corte com incerteza

Na pesquisa para o desenvolvimento desta dissertação, não foram encontrados muitos trabalhos sobre problemas de corte com incerteza na literatura. No entanto, este é um tópico de interesse atual, pois foram encontrados os seguintes projetos em desenvolvimento:

- “Incerteza em problemas de cortes e empacotamentos: planeamento robusto e replaneamento otimizado na produção e nos transportes” a decorrer no INESC TEC, conjuntamente com a Universidade de São Paulo - Instituto de Ciências Matemáticas e de Computação com data prevista para terminar a 31 de Agosto de 2021;
- “O problema de corte bidimensional com sobras aproveitáveis e incerteza na demanda” a decorrer na Faculdade de Engenharia da Universidade Estadual Paulista, com data prevista para terminar a 31 de Julho de 2021.

Alem e Morabito [10] já abordaram o tema de corte de matéria-prima sob incertezas e planeamento de produção, obtendo soluções a partir de otimização robusta. Mais tarde, Alem e Morabito [11] analisaram a programação estocástica e a otimização robusta para o mesmo problema, verificando que a otimização robusta é mais fácil de resolver, mas que cada problema deve ser analisado para perceber qual o método mais eficaz.

Zanjani, Noureldath e Ait-Kadi [12], propuseram uma abordagem estocástica no planeamento de produção com incerteza na matéria-prima, incluindo processos de cortes. O problema foi validado através da Simulação de Monte Carlo.

2.2.2. Cenários

Os cenários são combinações de possíveis resultados com incerteza. O processo de obtenção de cenários ou de árvores de cenários é conhecido como geração de cenários. Consiste na geração de resultados discretos para variáveis aleatórias.

Durante o estudo deste problema, foram utilizados cenários de forma a validar o grau de robustez dos resultados obtidos.

Um conjunto de cenários representa todo o universo probabilístico sobre o qual é efetuado o processo de otimização do problema. O objetivo das técnicas de geração de cenários é selecionar um conjunto de cenários que minimize o erro de aproximação. A qualidade da solução depende significativamente do método de geração de cenários aplicado, ou seja, da capacidade do método de capturar as características do processo aleatório. A utilização de cenários torna-se útil quando o valor da solução ótima é afetado com mudanças nas variáveis estocásticas.

Não existe um padrão geralmente aceite para medir a qualidade de um cenário, mas existem dois métodos propostos. O método de Ruszczyński e Shapiro (2003) [13] afirma o princípio da geração ideal de cenários. Este princípio requer que o processo de geração de cenários replique a distribuição original com a maior precisão possível. Consequentemente, a distribuição de probabilidades (discreta) associada a um conjunto de cenários é comparada com a original por alguma medida de distância. O método proposto por Kaut e Wallace (2003) [14] considera que a geração de cenários deve ser baseada no desempenho e o mais importante é ter ferramentas de geração de cenários que proporcionem um controlo razoável sobre a árvore com um número limitado de cenários. O que se pretende é um método que

seja estável, imparcial e produza pequenas árvores. Uma consequência desse critério é que um método de geração de cenários pode ser considerado adequado para um modelo, mas não para outro modelo.

A qualidade da geração de cenários é avaliada pela estabilidade da amostra e pelo teste de tendência. A estabilidade da amostra significa que a solução ótima da função objetivo não deve apresentar uma variação significativa relativamente à alteração da amostra estatística usada na geração de cenários, ou seja, se gerarmos vários conjuntos de cenários (com os mesmos parâmetros iniciais) e resolvermos o problema de otimização com esses conjuntos, devemos obter o mesmo valor ótimo da função objetivo. Para melhorar a estabilidade de uma amostra pode-se aumentar o número de cenários. O teste de tendência pretende determinar se o método de geração de cenário introduz tendência na obtenção da solução ótima, ou seja, o método de geração de cenário não deve introduzir qualquer tendência em comparação com a solução verdadeira, onde não são incluídos cenários.

O processo de gerar cenários deve cumprir algumas propriedades de modo a ser eficaz, como a consistência do procedimento, o objetivo a atingir, os fatores necessários a incorporar para ser possível tomar decisões e a consistência com os aspetos empíricos dos dados.

Os métodos de geração de cenários podem ser classificados como:

- Geração de Cenários Baseados em Amostragem;
- Geração de Cenários Baseados em Métodos Estatísticos;
- Geração de Cenários Baseados em Simulação.

Na Geração de Cenários Baseados em Amostragem os cenários são obtidos através de amostragem de uma distribuição por algum método de amostragem. Cada valor amostrado constitui o valor de um cenário, logo cada método de amostragem dá origem a diferentes métodos de geração de cenários. Incluem-se nos métodos de Geração de Cenários Baseados em Amostragem os métodos Monte Carlo e Amostragem de Importância, Amostragem Interna, Amostragem *Bootstrap* e Amostragem de Monte Carlo da Cadeia de Markov (MCMC). Outros métodos de geração de cenários baseados em amostragem incluem Amostragem Condicional, Amostragem Aleatória, Amostragem Baseada em Árvores ou Treliças Binomiais e Amostragem Estratificada.

Na Geração de Cenários Baseados em Métodos Estatísticos consideram-se processos aleatórios que possuem propriedades estatísticas específicas e essas propriedades são usadas para gerar cenários. Existem também métodos estatísticos que podem fornecer cenários (por exemplo, Análise de Componentes Principais). Nos métodos de Geração de Cenários Baseados em Métodos Estatísticos incluem-se Correspondência de Propriedade Estatística e Análise de Componentes Principais (PCA). Outros métodos de Geração de Cenários Baseados em Métodos Estatísticos incluem Geração de Cenários Baseada em Regressão e Métodos Relacionados e Geração de Cenários Multivariados Recorrendo ao Uso de Cópulas.

A Geração de Cenários Baseada em Métodos Estatísticos apresenta vantagem de poder gerar cenários melhores numa árvore de cenário de tamanho menor em comparação à Geração de Cenário Baseada em Amostragem ou Simulação. Isso acontece porque a Geração de Cenários Baseada em Métodos Estatísticos normalmente não depende do número de cenários. A desvantagem deste método é que os métodos estatísticos podem resultar em cenários que não refletem com precisão o processo aleatório.

Na Geração de Cenários Baseados em Simulação, números aleatórios são extraídos do domínio da parte estocástica da equação, que é então usado para calcular a resposta da equação. A simulação foi inventada para gerar soluções para equações analiticamente intratáveis.

Nos métodos de Geração de Cenários Baseados em Simulação incluem-se Simulação de Equações Diferenciais Estocásticas e Simulação de Modelo Econométrico. Além dos métodos referidos, existem métodos que se encontram em pesquisa, nos quais se destacam os Métodos Híbridos que combinam os métodos atuais de geração de cenários (April et al., 2003) para construir um método de geração de cenários mais poderoso.

A Geração de Cenário Baseada em Simulação apresenta a vantagem de facilmente poder criar vários modelos analiticamente intratáveis, e de poder capturar com precisão o comportamento de muitos modelos complicados de tempo contínuo. No entanto, a geração de cenários baseada em simulação apresenta a desvantagem de poder resultar em árvores de cenário com propriedades estatísticas significativamente diferentes num pequeno número de cenários.

A utilização de cenários tornou-se relevante no meio empresarial, pois as empresas trabalham com muitas tomadas de decisão sob incerteza, o que acontece, segundo Mitra e Domenica(2010) [15] nos meios financeiro (Fleten e Kristoffersen, 2007), planeamento de recursos (Wang et al., 2008; Huang e Ahmed, 2008), produção de energia (Fleten e Kristoffersen, 2008), e planeamento agrícola (Birge e Louveaux, 1997). Em Lempert et al. (2006) [16] é abordada a utilização de cenários nas mudanças na economia, na demografia e no meio ambiente.

2.3. Metodologia

Neste sub-capítulo é feita uma apresentação geral dos Algoritmos Genéticos e mais em específico do método utilizado: o BRKGA.

2.3.1. Algoritmos Genéticos

Os Algoritmos Genéticos (AG) são métodos de pesquisa probabilística, usados na resolução de problemas de busca e otimização de resultados. São baseados na teoria da seleção natural de Darwin, da evolução natural das espécies, que diz: *“Quanto melhor um indivíduo se adaptar ao seu meio ambiente, maior será a sua chance de sobreviver e gerar descendentes.”* (A Origem das Espécies, Charles Darwin 1859). Ou seja, a sobrevivência é considerada uma medida de desempenho, onde o indivíduo mais apto ao meio ambiente tem mais hipóteses de sobreviver e perpetuar as suas características genéticas para as gerações futuras.

Estes métodos surgiram na década de 1960, por John H. Holland, que estudou a relação da evolução natural das espécies e da seleção natural e o método de aplicar este fenómeno à computação evolucionária, mas apenas foi publicado em 1975 [17].

Os AG simulam os processos naturais de sobrevivência e da reprodução das populações. Os indivíduos de uma mesma população competem entre si, procurando sobreviver, consequentemente os indivíduos mais aptos conseguem um maior número de descendentes, do que os indivíduos menos aptos. Designa-se por “população” um conjunto de soluções possíveis para o problema.

Na representação dos AG os indivíduos (as soluções) são designados “cromossomas”. Cada “cromossoma” é representado por um vetor de valores (os “alelos”) e é também uma solução do problema. As características da solução, designadas “genes”, são os valores dos “alelos” do “cromossoma”. Cada iteração do processo de otimização designa-se “geração”.

Para perceber o processo de funcionamento dos AG é necessário considerar as várias etapas de implementação do algoritmo. Inicia-se com a escolha de uma população inicial, normalmente formada por cromossomas aleatórios, que contenha diversidade suficiente para o algoritmo ter capacidade de combinar diferentes características e gerar novas soluções. Esta população é submetida às várias etapas que constituem o processo de evolução de um algoritmo genético ao longo de várias gerações até atingir o critério de paragem, como representa a Figura 3. As etapas são constituídas pela fase de “avaliação”, onde a população de cromossomas é avaliada segundo um critério determinado pela função objetivo, que avalia a qualidade do “cromossoma”. Esta função é designada por função de aptidão ou “*fitness*”. Após a avaliação, segue-se a “seleção”, na qual são escolhidos os cromossomas com melhor valor, que é dado pelo valor “*fitness*”, e com estes cromossomas é criado um novo conjunto de possíveis soluções - uma nova “geração”. Através dos operadores “cruzamento” (“*crossover*”) e “mutação” são introduzidos novos “genes” a esta nova geração. Este caminho é repetido até um dos critérios de paragem ser atingido. Os critérios podem ser encontrar um certo valor de aptidão ou até que o número de voltas estipuladas seja atingido ou ainda quando toda a população tem um certo grau de homogeneidade.

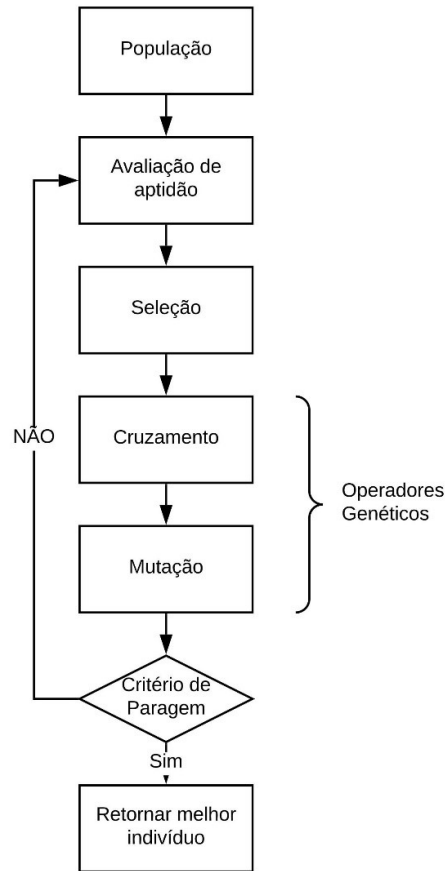


Figura 3. Estrutura de um algoritmo genético (baseada em [18]).

Um aspeto importante a considerar na pesquisa dos AG, é que estes procuram encontrar os melhores valores para os parâmetros comportamentais. O seu desempenho depende fundamentalmente do valor dos seus parâmetros, como o tamanho da população, o cruzamento e as taxas de mutação.

De seguida, nesta secção, são explicados os principais componentes de um AG.

População Inicial (Indivíduos)

A população de um AG é formada por um conjunto de indivíduos aleatoriamente escolhidos para criar uma população inicial. O tamanho da população inicial e o método de seleção dos indivíduos são as principais questões em ter em conta devido ao impacto na solução final. O tamanho da população é um fator que afeta o desempenho e a eficiência dos algoritmos genéticos. Uma população pequena corresponde a pouca diversidade genética e pode convergir rapidamente para uma solução cujo *fitness* não seja o ótimo. Já uma grande população requer maior esforço computacional para avaliar todos os indivíduos, o que poderá ser pouco eficiente pelo tempo gasto na procura. O método de seleção tem também um grande impacto. Nesta fase são selecionados os indivíduos para futura reprodução: quanto mais aptos forem estes indivíduos selecionados, maior probabilidade existe de obter bons resultados.

Para representar adequadamente os cromossomas ao estudo dos problemas, estes são codificados como um vetor de caracteres de tamanho fixo, formado pela concatenação dos caracteres “0” e “1”.

Avaliação de Aptidão (Fitness)

Na fase de avaliação de aptidão é calculada a aptidão dos indivíduos que constituem a população, fazendo uma análise para perceber a resposta de cada indivíduo ao problema proposto. Esta avaliação deve fazer uma separação clara entre boas e más soluções.

Seleção

Na fase de seleção os indivíduos são selecionados para reprodução da geração seguinte. A população é submetida a um critério de seleção de modo a selecionar os indivíduos mais aptos a gerar descendentes. A probabilidade de um indivíduo ser selecionado é proporcional à sua aptidão.

Existem vários critérios para efetuar a seleção dos indivíduos. A seleção pode ser realizada através da Seleção por Roleta (*Roulette Wheel*) (Michalewicz, 1996) [21], Seleção por Torneio (*Tournament Selection*) ou Seleção por Amostragem Universal Estocástica (*SUS - Stochastic Universal Sampling*).

No caso da Seleção por Roleta todos os indivíduos são colocados numa roleta com um espaço proporcional ao seu valor de aptidão. Assim, os indivíduos com maior aptidão possuem uma parte maior da roleta, tendo maior probabilidade de serem escolhidos para a geração seguinte. Roda-se a agulha da roleta e permanece o indivíduo escolhido para a geração seguinte. É de salientar que deste modo pode o indivíduo com melhor aptidão não ser o escolhido. O processo é repetido até encontrar o número de indivíduos necessários.

Outro método existente é a Seleção por Torneio. Este é um método muito atrativo pois é diferente da seleção proporcional, é rápido e exige pouco esforço computacional. Utiliza disputas sucessivas para obter uma seleção de indivíduos. Para selecionar k indivíduos, este método realiza k disputas onde em cada disputa participam n indivíduos escolhidos de forma aleatória. Neste método o indivíduo com maior aptidão na disputa é o selecionado. Este processo é repetido até terminar as k disputas.

O caso da Seleção por Amostragem Universal Estocástica é semelhante à Seleção por Roleta. A grande diferença encontra-se no fato de na Seleção por Amostragem Universal Estocástica as agulhas se encontram igualmente espaçadas e são giradas numa só vez. Consequentemente os resultados são menos variados que na Seleção por Roleta.

Após terminar o processo de seleção dos indivíduos, passa-se para as operações genéticas, Cruzamento e Mutação, para gerar indivíduos mais aptos.

Operadores Genéticos: Crossover e Mutação

Os operadores genéticos são fundamentais no processo dos AG para existir diversidade na população e as características das gerações anteriores sejam mantidas. Estes têm como objetivo principal transformar a população através de gerações sucessivas, até atingir uma solução que satisfaça o objetivo.

Crossover

Durante a fase de *Crossover* as características das soluções previamente escolhidas são recombinadas gerando novos indivíduos. Isto acontece trocando os genes de um progenitor pelos genes correspondentes de outro progenitor, gerando novos indivíduos, nos quais a informação genética é formada por sequências genéticas parciais de cada progenitor.

Existem vários tipos de operadores de *crossover*. Este pode ser simples (crossover de um ponto), que é o mais comum, porém o que dá piores resultados (Caruana et al. 1989) [19], multiponto (crossover de dois pontos ou mais), ordenado (Goldberg, 1989) [20], aritmético, geométrico e esférico (Michalewicz, 1996). Não existe um operador considerado o melhor. Este deve ser escolhido de acordo com o problema a estudar

Mutação

A mutação é um processo aleatório de manipulação. Durante o processo de mutação ocorre a alteração de características dos indivíduos selecionados para reprodução, o que introduz diversidade à população. Qualquer indivíduo em qualquer posição de descendência pode sofrer mutação, gerando um novo valor não proveniente da informação genética dos progenitores. No caso do operador mutação pode ocorrer mutação simples, onde apenas um gene de cada vez sofre mutação ou mutação aleatória, onde uma percentagem de genes sofre mutação. A mutação pode ainda ser uniforme ou não uniforme [21].

Atualização

Na fase de atualização, os novos indivíduos gerados são introduzidos na população e indivíduos já pertencentes à população são removidos.

Critério de paragem

Nesta fase final todas as condições de paragem da evolução são verificadas com intuito de perceber se os critérios foram atingidos. Em caso positivo a execução do algoritmo é encerrada, em caso negativo retorna à etapa de avaliação. A paragem pode ocorrer se atingir o tempo limite estipulado para a operação, ou número de gerações pré-definidas for atingido.

Os AG conseguem apresentar valores satisfatórios em diversas áreas, sendo aplicados com sucesso a problemas de otimização (problemas de escalonamento, desenvolvimento de controladores *fuzzy* para robots móveis, otimização de rotas, controlo de irrigação, entre outros). No entanto por vezes apresentam um problema de convergência prematura, onde o algoritmo atinge o critério de paragem sem obter resultados eficientes devido à falta de diversidade de material genético e longe do seu valor ótimo. De forma a combater essa situação, várias melhorias foram propostas, entre as quais o BRKGA (*Biased Random-Key Genetic Algorithm*), do qual se falará a seguir.

2.3.1. BRKGA

O *Biased Random Key Genetic Algorithm* (BRKGA - algoritmo genético de chaves aleatórias tendenciosas) é uma metaheurística de busca geral que tem como objetivo encontrar soluções ótimas ou quase ótimas para problemas de otimização combinatória. Foi desenvolvido para problemas de otimização e proposto por Gonçalves e Resende [22]. Faz parte de uma classe dos algoritmos genéticos baseados em chaves aleatórias tendenciosas no qual as soluções são codificadas como vetores de chaves aleatórias e são número reais compreendidos no intervalo entre [0,1] gerados aleatoriamente a partir de uma distribuição uniforme. O BRKGA representa uma evolução relativamente aos AG mais simples. Também como evolução dos AG mais simples surgiu o *Random Key Genetic Algorithm* (RKGA - algoritmo genético de chaves aleatórias) proposto por Bean [23]. O RKGA surgiu com objetivo de resolver problemas de sequenciamento. As soluções podem ser representadas como permutações de um vetor, ou seja, ordenando um vetor obtemos a ordem sequencial. No entanto, o vetor necessita passar por um processo de decodificação.

O BRKGA diferencia-se do RKGA, que é também baseado no uso de vetores de chave aleatória, por dividir a população, p , em população elite (indivíduos com maior aptidão para sobreviver), p_e , e população não-elite, $p-p_e$, (maior dos dois conjuntos) no processo de *crossover*. Enquanto no RKGA os dois progenitores são selecionados aleatoriamente de toda a população para proceder ao *crossover*, no BRKGA cada elemento é gerado através da combinação de um elemento selecionado de forma aleatória da população elite e um elemento da população não-elite, como mostra a Figura 4. Designa-se seleção tendenciosa pois um dos progenitores é sempre um indivíduo de elite, o qual tem uma maior probabilidade de transmitir os seus genes para a geração seguinte. Esta probabilidade é controlada pelo parâmetro *rhoe* (p_e - probabilidade de herança de alelo de elite).

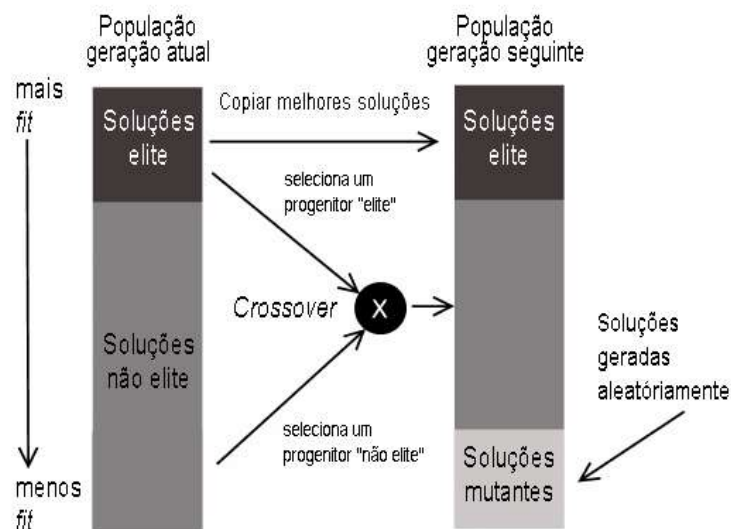


Figura 4. Evolução da população entre gerações consecutivas no BRKGA [22].

O decodificador é um algoritmo determinístico que tem como valor de entrada um vetor que representa a solução (o cromossoma), gera uma solução possível do problema, e

avalia a sua aptidão (cálculo do *fitness*). O decodificador deve ter custos baixos a nível computacional.

O fluxograma da Figura 5 representa esquematicamente o funcionamento de um algoritmo BRKGA. O processo é iniciado com um conjunto de p vetores aleatórios de tamanho n (população). O parâmetro n depende da solução codificada, isto é, do problema, enquanto o parâmetro p é definido pelo usuário. A partir da população inicial, o algoritmo gera uma série de populações. Cada iteração do algoritmo é chamada de geração. A população evolui ao longo das gerações, combinando pares de soluções de uma geração para produzir soluções descendentes para a geração seguinte.

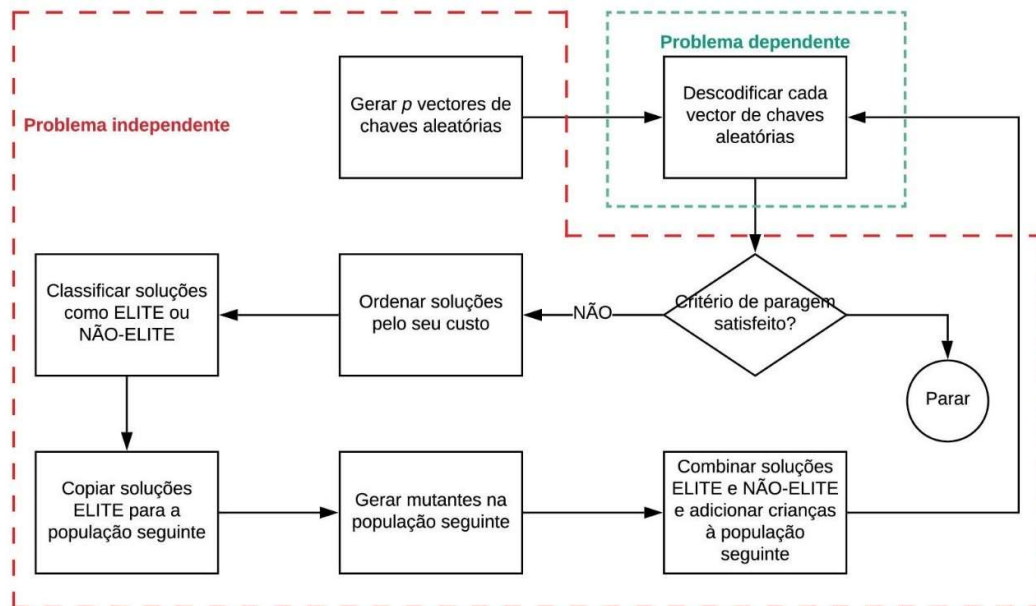


Figura 5. Framework do BRKGA (baseado em [24]).

A estrutura do BRKGA é dividida em duas partes, como se verifica na Figura 5: uma parte que é independente do problema e a outra parte, o decodificador (*decoder*), que depende do problema ao qual é aplicado. Os componentes da parte independente do BRKGA são desenvolvidos sem conhecimento do problema. A parte dependente do BRKGA é responsável pela ligação ao problema.

Como foi dito anteriormente, a população inicial é composta por vetores de chave aleatória p , cada um com n chaves. O valor de cada vetor é gerado aleatoriamente no intervalo $[0,1]$. A população evolui ao longo das gerações e em cada geração é ordenada pelo valor da sua aptidão. A população designada elite, contém as melhores soluções e é copiada sem alterações para a população da geração seguinte. A população designada mutante, p_m , constituída por vetores de chaves aleatórias geradas aleatoriamente, é também introduzida na população da geração seguinte. A última partição da população da geração seguinte é constituída pelo cruzamento do progenitor da população elite com o progenitor da população não-elite. É possível ocorrer a repetição de um progenitor, logo um indivíduo pode produzir mais do que um filho na mesma geração. A avaliação de aptidão é efetuada quando a geração se encontra completa. Os valores de aptidão são calculados para todos os novos vetores de chave aleatória e assim se faz a nova partição da população em elite e não-elite. Um BRKGA

é descrito mostrando como as soluções são codificadas como vetores de chaves aleatórias e como esses mesmo vetores são decodificados para soluções viáveis dos problemas de otimização.

Para se proceder à implementação do BRKGA, existe uma API disponibilizada por Gonçalves e Resende, onde toda a parte independente se encontra implementada e cabe ao utilizador implementar apenas a parte dependente, o decodificador. Alguns parâmetros devem ser ajustados a cada problema específico e outros parâmetros, mesmo sendo independentes do problema, podem e devem ser afinados. No entanto, a API tem pré-definidos alguns valores destes parâmetros:

- tamanho da população (p) = 1000
- tamanho da população de elite (p_e) = 20%
- tamanho da população mutante (p_m) = 10%
- probabilidade de herança de alelo de elite (p_e) = 70%
- número de populações independentes (K) = 3
- número de *threads* para decodificação paralela ($MAXT$) = 1
- número de gerações ao fim das quais existe troca de melhores indivíduos entre populações independentes (X_INTVL) = 100

Os últimos dois parâmetros tornam a utilização desta API do BRKGA significativamente eficiente. Por um lado, permite que haja várias populações a evoluir independentemente, com trocas de elementos a intervalos de gerações pré-definidos. Por outro lado, a API está preparada para funcionar em *threads* paralelas aumentando a rapidez do algoritmo.

Para validar este método e compreender o funcionamento do BRKGA, este foi testado para o problema genérico *Travelling Salesman Problem*, usando instâncias padrão da literatura, onde foi possível verificar a eficiência e rapidez do método.

Capítulo 3. Desenvolvimento do Método de Resolução

Neste capítulo é apresentado o método de resolução para o problema. Primeiro é explicado o decodificador do BRKGA e de seguida a utilização de cenários para o problema em estudo.

3.1. BRKGA adaptado ao problema - *decoder*

Como anteriormente já foi referido, o BRKGA é dividido em duas partes, uma independente do problema e outra parte dependente do problema. A parte dependente do problema, o decodificador, foi implementado em C++ a partir da API já existente. O modelo matemático previsto em 1.2 foi adaptado à função de avaliação de *fitness* de modo a se conseguir obter boas soluções para o problema.

Todos os parâmetros foram definidos de acordo com o modelo matemático.

O decodificador, como representado na Figura 6, transforma um cromossoma numa solução. O cromossoma é representado por um vetor de alelos que corresponde ao valor de entrada, e é transformado pelo decodificador em valores para as variáveis de decisão.

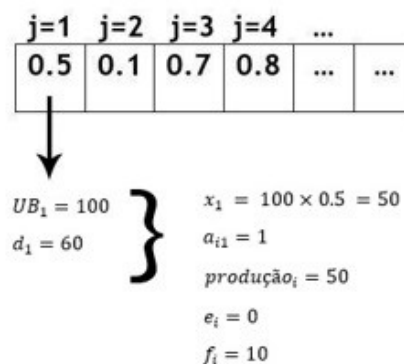


Figura 6. Funcionamento do decodificador (exemplo).

No caso do problema estudado, o cromossoma pode ter 15 alelos ou 30 alelos, respetivamente para a instância pequena e para a instância grande (apresentadas em 4.1). Cada alelo corresponde a um padrão de corte j e apresenta um valor real entre 0 e 1.

De forma auxiliar à descodificação, foi calculado um *upper bound* para cada padrão de corte j (UB_j) para estabelecer um limite máximo de corte de padrões, garantindo que a procura dos itens presentes no padrão seria atendida se apenas se cortasse aquele padrão. O número de vezes que cada padrão é cortado (x_j) é, então, dado pelo valor do alelo a multiplicar pelo valor do *upper bound*, como se pode verificar na Equação (3).

$$x_j = UB_j \times alelo_j, \forall j \quad (3)$$

Conhecendo o valor do parâmetro a_{ij} , obtém-se a quantidade de peças produzidas e, consequentemente, as variáveis de excesso e falta de peças são calculadas a partir da fórmula das restrições (2) apresentada em 1.2, pois os valores da procura, do número de vezes que a peça i é cortada no padrão j , e do número de vezes que o padrão de corte j é cortado são valores conhecidos.

Caso haja incerteza, isso provoca impacto no parâmetro a_{ij} .

Por fim, conhecidos todos os valores, pode calcular-se o *fitness* através da função objetivo.

3.2. Cenários

No desenvolvimento deste problema foram aplicados diferentes cenários ao problema estudado. Como não se conheciam dados históricos para o problema, e de forma a garantir que se estudava o impacto de diferentes distribuições de probabilidades, para desenvolver os cenários foi utilizada a distribuição Beta.

A distribuição Beta é uma família de distribuições de probabilidade contínuas definida por dois parâmetros positivos: α e β . Esta é usada para modelar o comportamento de variáveis aleatórias pertencentes ao intervalo $[0,1]$, pois a distribuição está definida neste intervalo. A distribuição é definida pelo valor dos parâmetros. A distribuição é simétrica quando $\alpha = \beta$, quando $\alpha > \beta$ a distribuição apresenta assimetria negativa e quando $\alpha < \beta$ a assimetria apresentada pela distribuição é positiva. A distribuição Beta é considerada apropriada para modelar devido à variedade de formas que a sua densidade pode obter, dependendo do valor definido nos parâmetros. A Figura 7 representa as distribuições de probabilidade usadas para gerar cenários para o problema estudado. No total, para cada instância, foram gerados cinco conjuntos de cenários (um associado a cada distribuição), tendo cada conjunto 10 cenários.

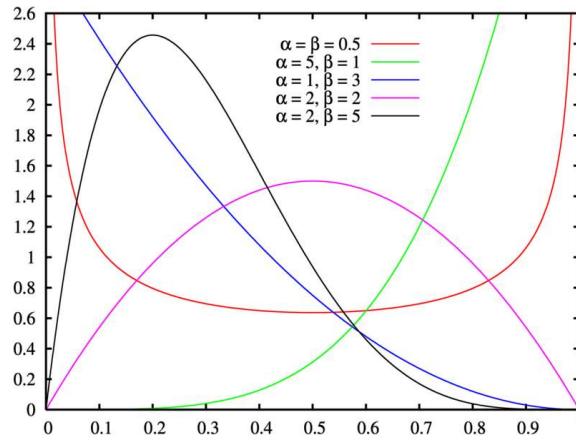


Figura 7. Função densidade de probabilidade para a distribuição Beta [23].

Os parâmetros usados para gerar os cinco conjuntos de cenários e os respectivos nomes atribuídos encontram-se de seguida:

- $\alpha = 0.5; \beta = 0.5$ (extremos);
- $\alpha = 5; \beta = 1$ (crescente);
- $\alpha = 1; \beta = 3$ (decréscante);
- $\alpha = 2; \beta = 2$ (aproximadamente normal);
- $\alpha = 2; \beta = 5$ (não central).

Os cenários foram introduzidos neste problema para representar a incerteza na qualidade da matéria-prima a cortar. Cada cenário indica-nos qual a percentagem de área de cada padrão que se encontra afetada em cada padrão j , provocando desperdício na matéria-prima, e, consequentemente, qual percentagem se encontra em bom estado.

Os cenários são combinações de possíveis resultados com incerteza e, neste caso, são valores que vão ter impacto no valor do parâmetro a_{ij} (número de vezes que a peça i é cortada no padrão j), pois se alguma peça apresentar defeito na matéria-prima terá de ser novamente cortada.

A representação da curva de distribuição indica para que padrões existe maior ou menor probabilidade de defeito, dependendo se se encontra mais centrada, ou mais densa nos extremos.

Os parâmetros de distribuição foram escolhidos para se obter diversidade nas distribuições. A distribuição Beta(2,2) aproxima-se a uma distribuição Normal, com média central. A distribuição Beta(0.5,0.5) apresenta os valores de maior probabilidade nos dois extremos. Ambas estas distribuições são simétricas. A distribuição Beta(5,1) (crescente) apresenta maior probabilidade à medida que a distribuição cresce, contrariamente à distribuição Beta(1,3) (decréscante), que apresenta maior probabilidade no início e apresenta uma distribuição a decrescer. Por ultimo, a distribuição Beta(2,5) (não central), apresenta

uma distribuição descentrada, com maior probabilidade a crescer no início e a decrescer no fim.

Para o problema específico assumiu-se um valor máximo de desperdício de padrão devido a problema de matéria-prima de 25%. Esse valor foi calculado pela distribuição de probabilidade para dar o valor real de máximo desperdício. Assumiu-se que a percentagem de desperdício devido a problema na matéria-prima do objeto onde o padrão é cortado afeta todos os itens i em cada padrão j de igual forma.

Esta simplificação e limitação foi considerada adequada para os testes a realizar para o problema estudado e poderá ser revista numa fase posterior do projeto.

Capítulo 4. Testes e Resultados

Neste capítulo são apresentados os testes efetuados e resultados obtidos para o problema apresentado. Primeiramente são apresentadas as instâncias utilizadas para efetuar os testes, e de seguida são apresentados resultados e análises ao problema.

Todos os testes foram realizados num computador da marca Lenovo, modelo ideapad 710S-13IKB, com um processador Intel Core i7, 7th Gen @ 2.70GHz 2.90GHz, memória RAM 8.00GB, com o sistema operativo Windows 10 Home de 64bits. A implementação foi feita em linguagem C++ e testado no programa Visual Studio 2019.

4.1. Instâncias

Na fase de testes do problema foram usadas duas instâncias geradas aleatoriamente. Estas instâncias apresentam tamanhos diferentes, sendo uma mais pequena, contendo 15 padrões e 50 peças ou itens, e uma maior, contendo 30 padrões e 100 peças. O tamanho das instâncias foi projetado para se aproximar às necessidades reais da resolução do problema. Achou-se relevante ter duas instâncias de tamanho diferente de modo a testar e validar o método proposto. Em anexo encontram-se os quadros com os valores das instâncias utilizadas.

É relevante salientar que no parâmetro θ_i (o custo de produzir peça i a menos) é um parâmetro muito importante, pois se o valor definido não for alto suficiente corre-se o risco do AG preferir não cortar peças devido ao custo. O valor de θ_i usado foi de 1000, derivado de testes preliminares. O critério de paragem foi definido em 1000 gerações.

4.2. Análise de sensibilidade aos parâmetros que controlam o elitismo e a aleatoriedade do BRKGA

No desenvolvimento desta dissertação foi feita uma análise de sensibilidade ao método proposto, especialmente aos fatores que impactam o elitismo e a aleatoriedade, como abordagem prévia para acomodar incerteza, variando o valor de alguns parâmetros de modo a verificar o impacto que estes têm na função objetivo definida. A análise de sensibilidade pretende prever o resultado gerado por alterações em parâmetros ou atividades em processo, medindo o impacto das alterações sobre o problema. Assim mede o grau de sensibilidade do processo perante uma mudança, podendo esta análise servir para tomada de decisões nas mudanças de variáveis do modelo estudado.

Decidiu-se estudar o impacto alterando o tamanho da percentagem da população de elite, da população mutante e da probabilidade de herança de alelo de elite, pois são os parâmetros mais relacionados com o elitismo e a aleatoriedade da parte independente do problema.

De seguida apresentam-se os valores para qual o problema foi testado, começando por apresentar o valor padrão da API BRKGA (sublinhado) e seguindo-se os dois outros valores que foram utilizados:

- População elite $\underline{p_e = 0.2}$; $p_e = 0.1$; $p_e = 0.4$
- População mutante $\underline{p_m = 0.1}$; $p_m = 0.05$; $p_m = 0.2$
- Probabilidade de herança de alelo de elite $\underline{\rho_e = 0.7}$; $\rho_e = 0.6$; $\rho_e = 0.8$

Para efetuar as análises foram testados valores maiores e menores que os valores pré-definidos pelo BRKGA. Os valores de população elite e população mutante para os quais a análise foi efetuada correspondem ao dobro e à metade do valor pré-definido pelo BRKGA. Os valores da probabilidade de herança de alelo de elite correspondem ao valor pré-definido somando e subtraindo 0.1.

Na realização desta análise foram usadas 5 *seeds* diferentes, sendo que os resultados apresentados correspondem aos valores médios obtidos.

Na análise de sensibilidade nesta secção assume-se que a matéria-prima não apresenta defeitos, isto é, não é considerada incerteza.

Valor da Função Objetivo - custo (instância grande)

$$\rho_e = 0.6$$

pe/pm	0.05	0.1	0.2
0.1	419.2	477	403.2
0.2	423.3	402.2	415.7
0.4	328.1	305.9	301.2

$$\rho_e = 0.7$$

pe/pm	0.05	0.1	0.2
0.1	525	438.9	371.4
0.2	450.4	403.8	429.1
0.4	344.5	357.6	351.7

$$\rho_e = 0.8$$

pe/pm	0.05	0.1	0.2
0.1	519	494.3	587
0.2	418.3	673.2	399.1
0.4	437.9	392.5	439.3

Tabela 1. Resultados da Função Objetivo obtidos na instância grande para os diferentes valores de ρ_e - probabilidade de herança do alelo de elite.

Analisando os valores obtidos nas Tabelas 1 e 2, pode-se verificar que quanto maior é a população elite, melhores são as soluções obtidas. De forma global, verifica-se que os melhores resultados são obtidos para o valor de probabilidade de herança de alelo de elite mais pequeno. A população mutante não apresenta um padrão relevante nos resultados.

Valor da Função Objetivo - custo (instância pequena)

$$\rho_e = 0.6$$

pe/pm	0.05	0.1	0.2
0.1	279.9	280.6	255.3
0.2	266.9	253.3	259.8
0.4	256.4	259.5	259.3

$$\rho_e = 0.7$$

pe/pm	0.05	0.1	0.2
0.1	280.9	253.0	265.8
0.2	285.6	253.3	263.4
0.4	264.2	266.6	252.7

$$\rho_e = 0.8$$

pe/pm	0.05	0.1	0.2
0.1	309.2	261.2	270.9
0.2	339.5	305.7	269.4
0.4	278.9	279.7	272.1

Tabela 2. Resultados da Função Objetivo obtidos na instância pequena para os diferentes valores de ρ_e - probabilidade de herança do alelo de elite.

No caso da instância pequena as melhores soluções nem sempre são encontradas quando a população de elite é maior, mas tendencialmente os melhores valores são encontrados aliados às maiores populações de elite. Ainda que nesta instância a melhor solução não se encontre para o menor valor da probabilidade de herança de alelo de elite, de forma geral as melhores soluções são encontradas para $\rho_e = 0.6$.

Nos dois casos se verifica que as piores soluções são encontradas para $\rho_e = 0.8$.

Sendo a população de elite e a probabilidade de herança de alelo de elite dois indicadores de elitismo, nas soluções obtidas verifica-se que as soluções tendem a ser melhores quando a população de elite é maior, pois tendo melhores indivíduos, a probabilidade de ter melhores soluções aumenta. Nesta análise foi possível verificar que o parâmetro da probabilidade de herança de alelo de elite, que é também um parâmetro que controla o elitismo, é menor para as soluções mais favoráveis. Seria necessário efetuar mais testes para perceber o comportamento inverso relativamente ao parâmetro de população de elite.

Valor médio dos tempos de processamento das instâncias

O valor médio do tempo de processamento da instância grande é de 9.5 segundos, já da instância pequena é 2.4 segundos. Verifica-se que a instância grande demora cerca de quatro vezes mais tempo a correr, comparativamente à instância pequena. A instância grande analisa o dobro de padrões a cortar comparativamente à instância pequena, gerando um

cromossoma com o número de alelos correspondente ao número de padrões, e pretende cortar o dobro do número de peças. Pode verificar-se que o tempo de processamento das instâncias aumenta na mesma proporção, no qual tanto o número de padrões como o número de peças parecem ter impacto.

Convergência da função objetivo para a solução final

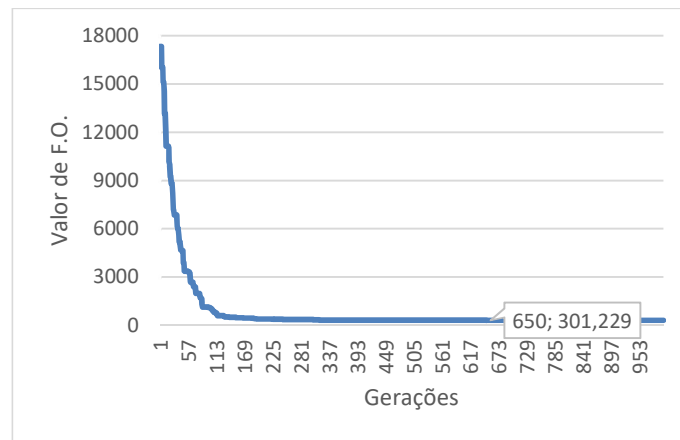


Figura 8. Gráfico de convergência da instância grande para a solução final durante 1000 gerações.

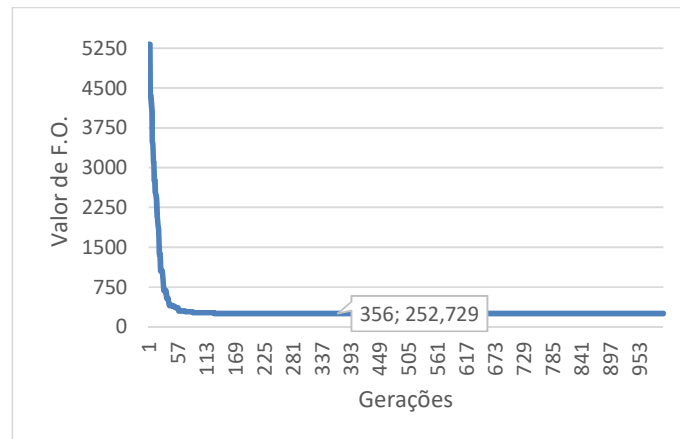


Figura 9. Gráfico de convergência da instância pequena para solução final durante 1000 gerações.

Pode-se verificar nas Figuras 8 e 9 que tanto na instância grande como na instância pequena a solução converge muito rapidamente para o valor da solução final, ainda que na instância grande o número de gerações até atingir o valor final da solução seja superior. Na Figura 8 pode-se verificar que na instância grande a solução final FO = 301.2 é atingida na geração 650. Já na Figura 9, verifica-se que a solução final FO = 252.7 é atingida na geração 356.

4.3. Análise da solução face aos cenários

Através dos resultados extraídos da Análise de Sensibilidade, as melhores combinações de parâmetros foram utilizadas para realizar a análise de comportamento face aos diferentes cenários. Para a instância grande utilizaram-se os seguintes parâmetros com os seguintes valores: $p_e = 0.4$; $p_m = 0.2$; $p_e = 0.6$. Já para a instância pequena utilizaram-se: $p_e = 0.4$; $p_m = 0.2$; $p_e = 0.7$.

Na realização destes testes, o BRKGA foi corrido independentemente para cada cenário dos conjuntos gerados através das diferentes distribuições de probabilidade.

De seguida são apresentados os valores da Função Objetivo e de cada uma das suas três componentes face a cada cenário, para cada uma das distribuições. Em cada gráfico é apresentado como *baseline* o valor da Função Objetivo quando não existem defeitos na matéria-prima, isto é, quando não há incerteza.

Resultados da instância grande

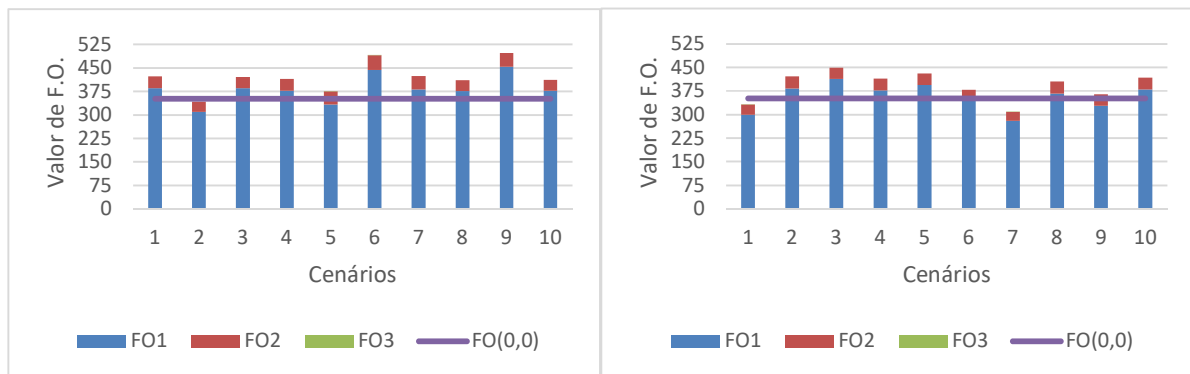


Figura 10. Valores dos componentes da função objetivo para cada cenário pertencente ao conjunto gerado pela distribuição $\alpha = 0.5$; $B = 0.5$ à esquerda e $\alpha = 1$; $B = 3$ à direita, e valor da função objetivo se não considerada incerteza.

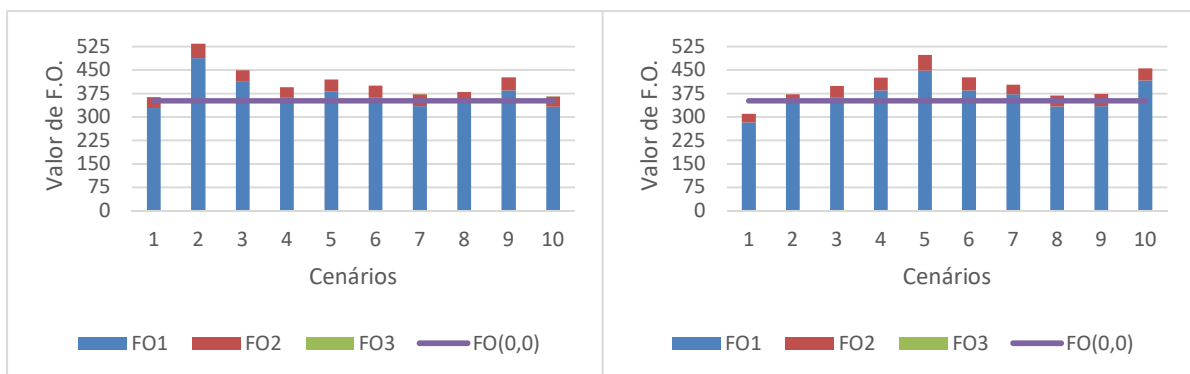


Figura 11. Valores dos componentes da função objetivo para cada cenário pertencente ao conjunto gerado pela distribuição $\alpha = 2$; $B = 2$ à esquerda e $\alpha = 2$; $B = 5$ à direita, e valor da função objetivo se não considerada incerteza.

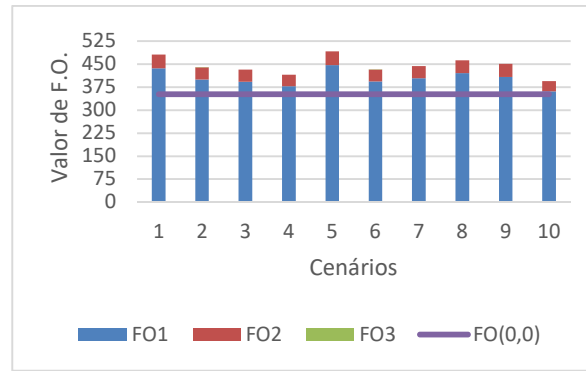


Figura 12. Valores dos componentes da função objetivo para cada cenário pertencente ao conjunto gerado pela distribuição $\alpha = 5$; $\beta = 1$, e valor da função objetivo se não considerada incerteza.

Resultados para a instância pequena:

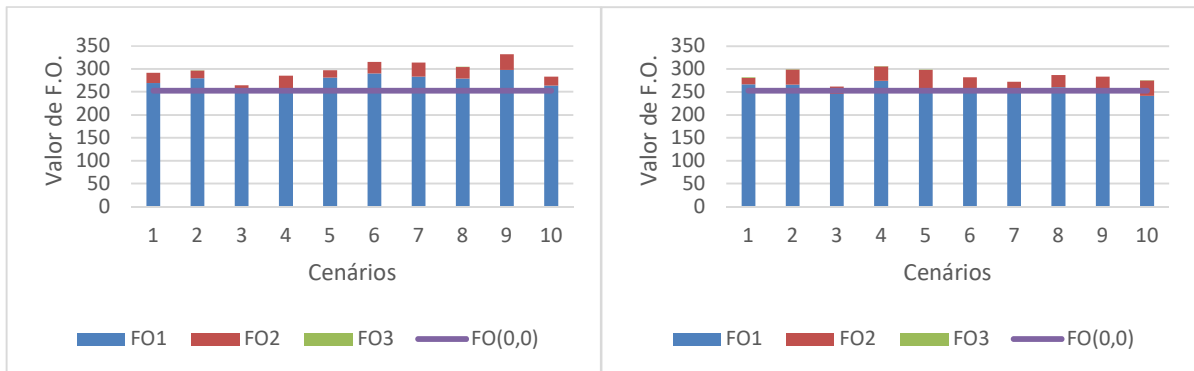


Figura 13. Valores dos componentes da função objetivo para cada cenário pertencente ao conjunto gerado pela distribuição $\alpha = 0.5$; $\beta = 0.5$ à esquerda e $\alpha = 1$; $\beta = 3$ à direita, e valor da função objetivo se não considerada incerteza.

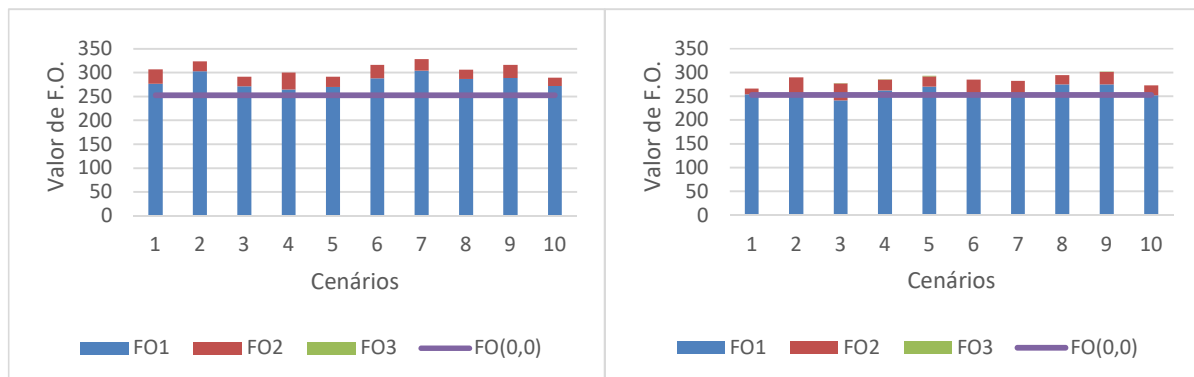


Figura 14. Valores dos componentes da função objetivo para cada cenário pertencente ao conjunto gerado pela distribuição $\alpha = 2$; $\beta = 2$ à esquerda e $\alpha = 2$; $\beta = 5$ à direita, e valor da função objetivo se não considerada incerteza.

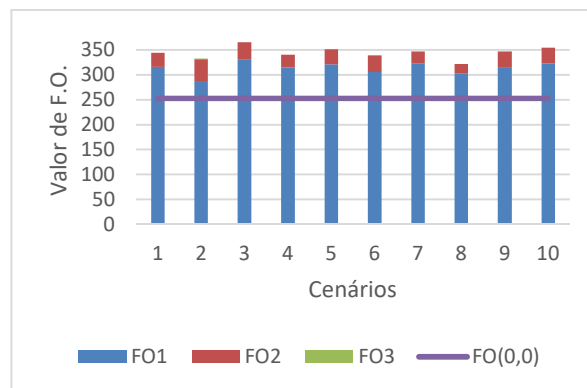


Figura 15. Valores dos componentes da função objetivo para cada cenário pertencente ao conjunto gerado pela distribuição $\alpha = 5$; $\beta = 1$, e valor da função objetivo se não considerada incerteza.

Pode-se verificar, analisando os gráficos das Figuras 10 a 15, que tanto para a instância grande como para a instância pequena a solução mais favorável é encontrada na distribuição decrescente ($Beta(1,3)$). Na instância grande, o cenário 7 é o mais favorável, dado que a melhor solução permite obter o menor custo com $FO=309.7$. Já para a instância pequena o cenário 3 é o mais favorável, onde se consegue obter $FO=262.4$. A melhor média de soluções na instância grande é também encontrada na distribuição decrescente ($Beta(1,3)$), no entanto, na instância pequena a média de soluções mais favoráveis é encontrada na distribuição não central ($Beta(2,5)$) mas com valores muito próximos à distribuição decrescente ($Beta(1,3)$). Não sendo a distribuição com a solução menos favorável na instância grande, a distribuição crescente ($Beta(5,1)$) apresenta a média de soluções que causam um impacto mais negativo. A distribuição de probabilidades que causa um impacto mais negativo à empresa é a distribuição aproximada à normal ($Beta(2,2)$), dado que no cenário 2 o valor de FO é o mais elevado, com $FO=534.2$. Na instância pequena a distribuição de probabilidades com impacto mais negativo é a distribuição crescente ($Beta(5,1)$) com $FO=364.9$. Também esta distribuição apresenta a média de soluções menos favoráveis.

Analisando os gráficos pode ainda verificar-se que os pesos das componentes FO1, FO2 e FO3 são muito diferentes. Sendo FO1 o custo de cortar os padrões e FO2 o custo associado ao desperdício de matéria-prima, uma solução favorável deve apresentar a componente FO1 com um valor muito superior à componente FO2, como é o caso das soluções dos testes efetuados. FO2 deve ter o menor valor possível, indicando o menor desperdício. Não se verificando para todos os casos, as soluções com os menores valores na componente FO2 tendem a coincidir com as soluções com menor valor da componente geral FO.

A componente FO3 é praticamente inexistente em todas as soluções, e após análise verificou-se que o custo presente na componente FO3, quando existente, é referente a peças cortadas em excesso.

Verifica-se que é possível obter melhores soluções quando a probabilidade de desperdício se encontra no início da distribuição, e as piores soluções ocorrem quando a probabilidade de desperdício se encontra no fim da distribuição. Note-se que padrões estão ordenados de forma crescente pelo valor de w (desperdício devido aos cortes, não à qualidade. Então, se a probabilidade de desperdício devido à qualidade se encontra mais nos

últimos padrões, o desperdício devido à qualidade tende a afetar padrões que de si já tinham maior desperdício devido ao próprio padrão de cortes.

Instância grande - corte de padrões

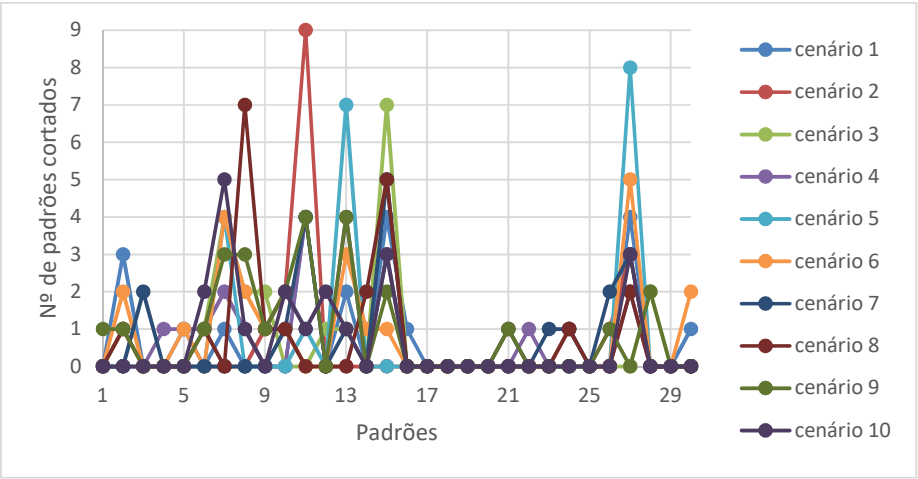


Figura 16. Gráfico de padrões cortados na solução final obtida para cada cenário do conjunto gerado pela distribuição Beta(0.5,0.5).

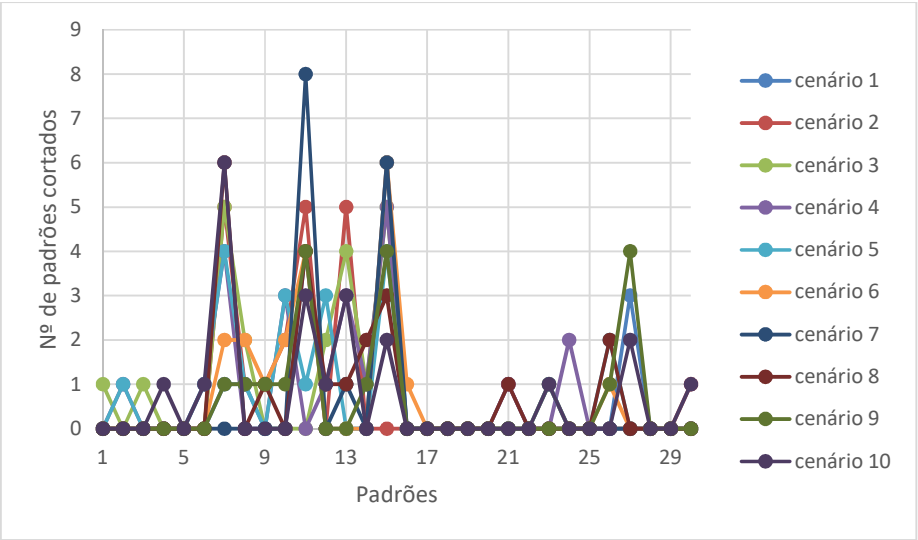


Figura 17. Gráfico de padrões cortados na solução final obtida para cada cenário do conjunto gerado pela distribuição Beta(1,3).

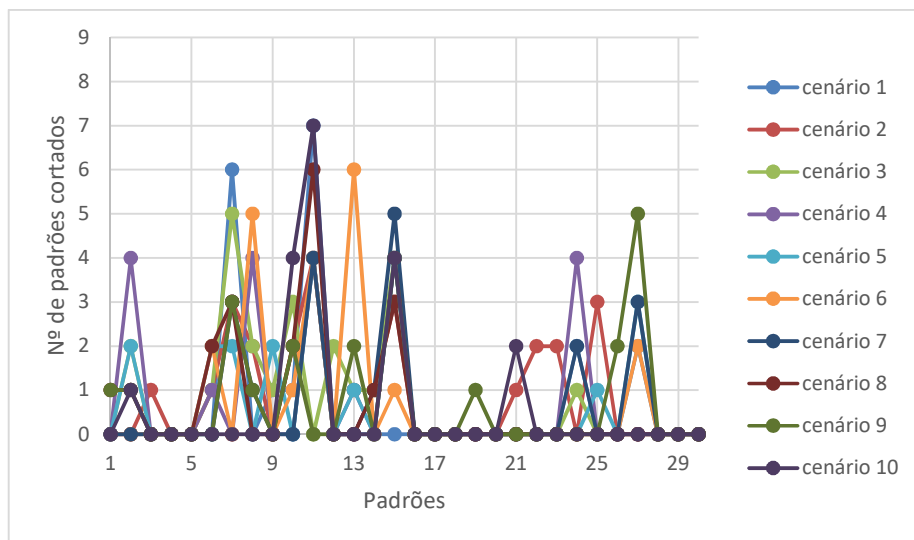


Figura 18. Gráfico de padrões cortados na solução final obtida para cada cenário do conjunto gerado pela distribuição Beta(2,2).

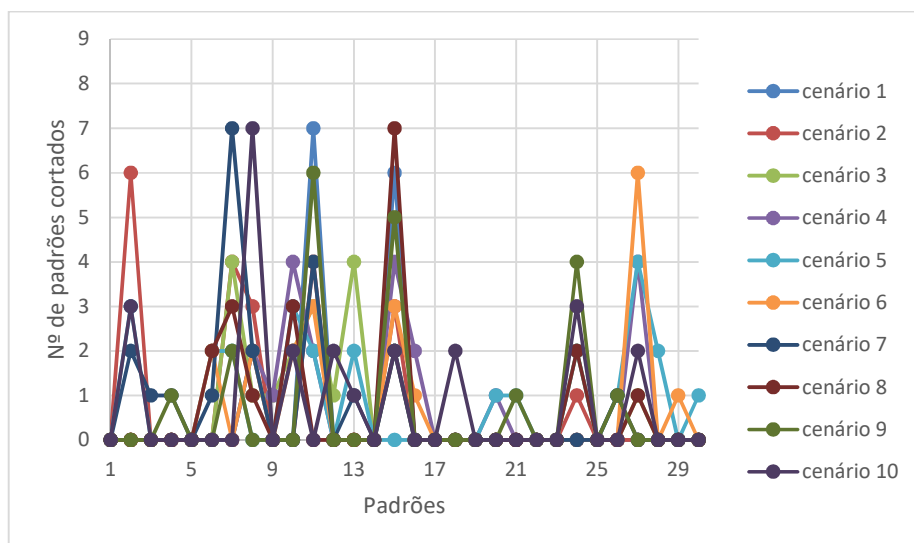


Figura 19. Gráfico de padrões cortados na solução final obtida para cada cenário do conjunto gerado pela distribuição Beta(2,5).

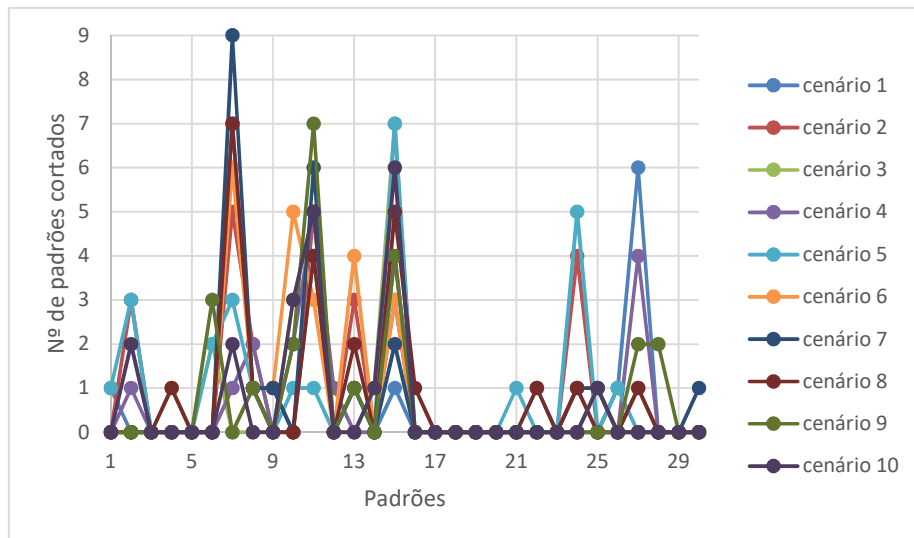


Figura 20. Gráfico de padrões cortados na solução final obtida para cada cenário do conjunto gerado pela distribuição Beta(5,1).

Instância pequena - corte de peças

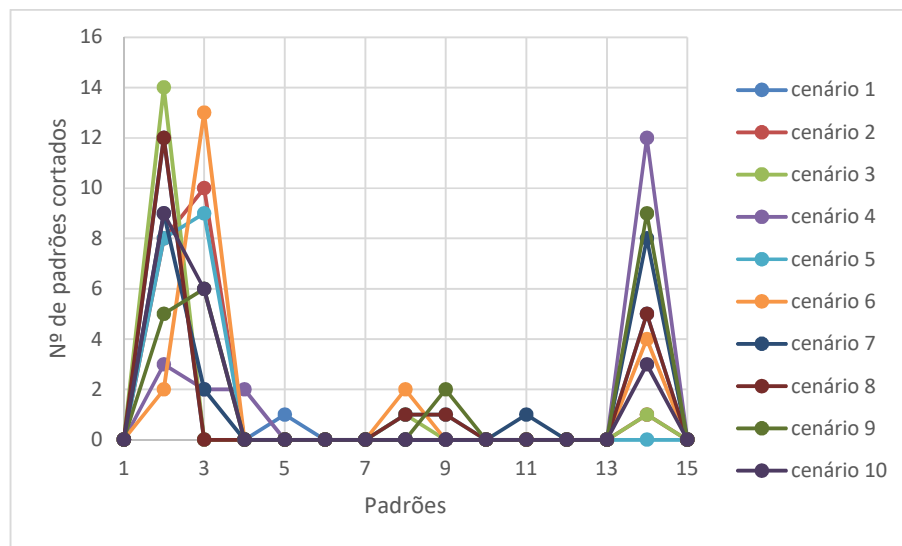


Figura 21. Gráfico de padrões cortados na solução final obtida para cada cenário do conjunto gerado pela distribuição Beta(0.5,0.5).

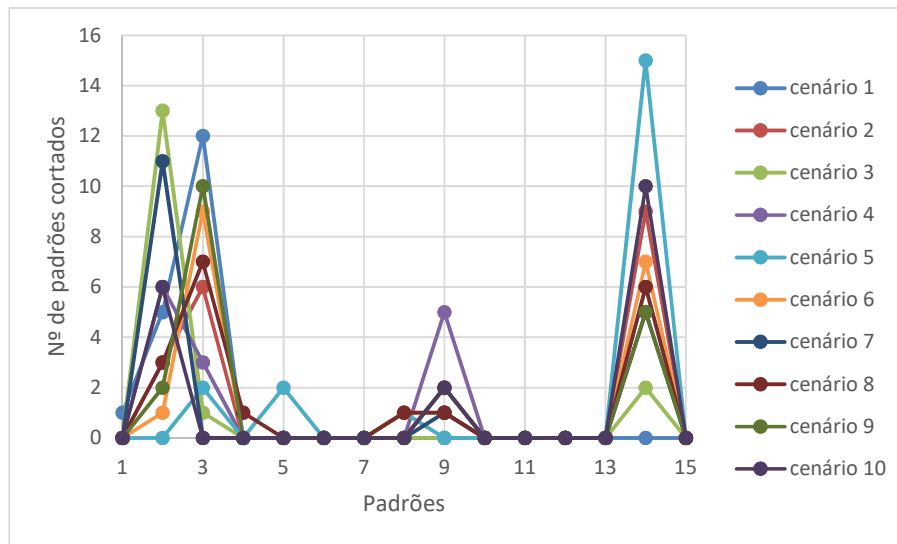


Figura 22. Gráfico de padrões cortados na solução final obtida para cada cenário do conjunto gerado pela distribuição Beta(1,3).

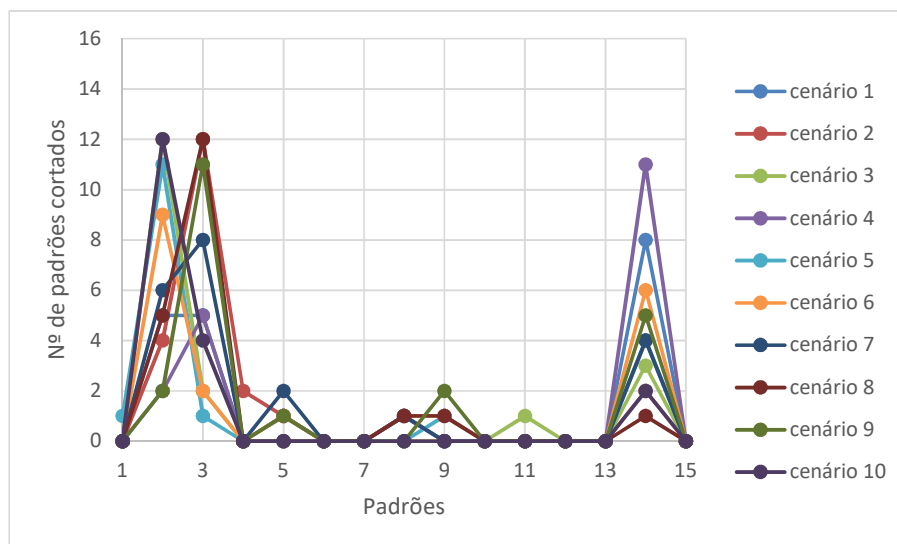


Figura 23. Gráfico de padrões cortados na solução final obtida para cada cenário do conjunto gerado pela distribuição Beta(2,2).

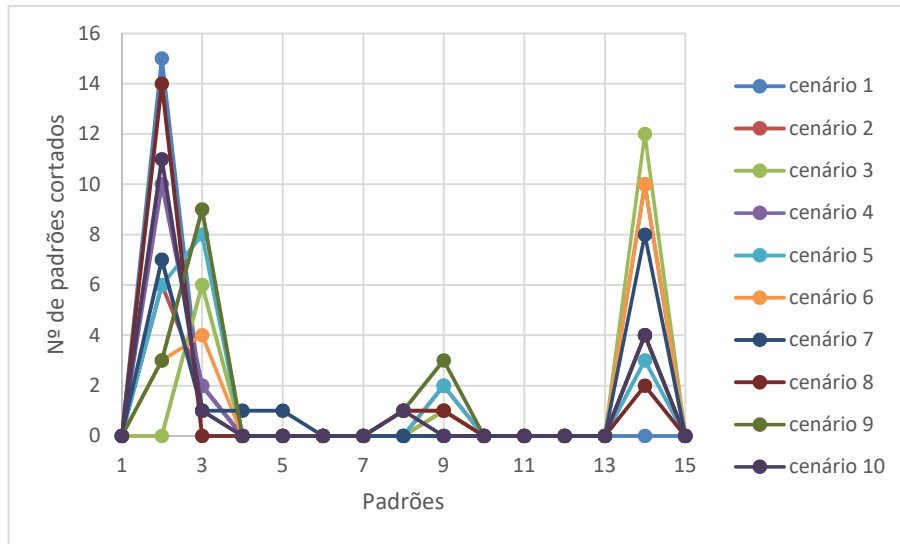


Figura 24. Gráfico de padrões cortados na solução final obtida para cada cenário do conjunto gerado pela distribuição Beta(2,5).

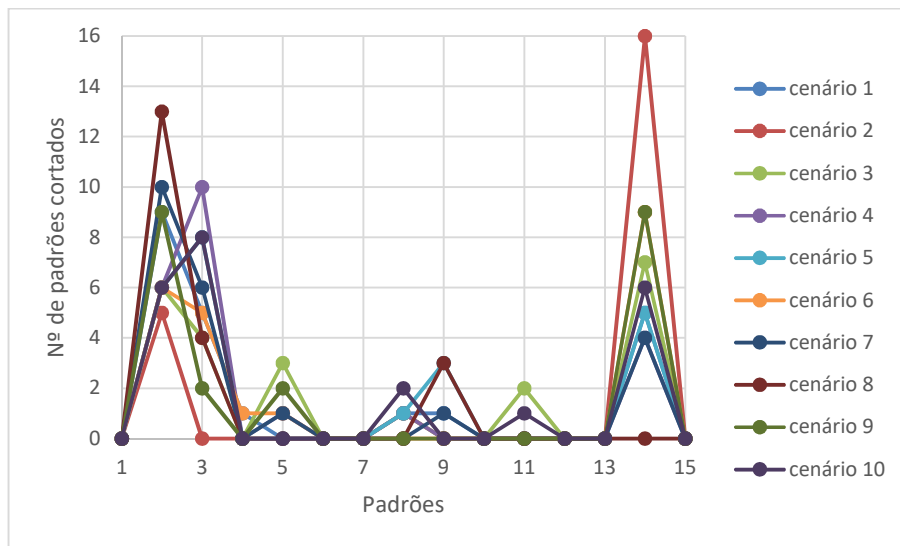


Figura 25. Gráfico de padrões cortados na solução final obtida para cada cenário do conjunto gerado pela distribuição Beta(5,1).

Os gráficos apresentados entre as Figuras 16 e 20 mostram, para a instância grande, os padrões cortados para cada cenário, sendo estes diferentes para cada distribuição de probabilidades. Os padrões cortados por cenário para a instância pequena encontram-se entre as Figuras 21 e 25.

A instância grande apresenta o menor número de padrões cortados nas distribuições decrescente (Beta(1,3)) e não central (Beta(2,5)), sendo que a solução mais favorável da Função Objetivo se encontra para o valor da distribuição decrescente (Beta(1,3)). Ambas apresentam o resultado mínimo de 15 padrões cortados no total no cenário 7. Este resultado é melhor que o caso sem cenários, onde na melhor solução são cortados 17 padrões. A média

de menor número de padrões cortados é encontrado na distribuição decrescente (Beta(1,3)) com 19.7 padrões cortados. A solução com maior impacto negativo é encontrada na distribuição crescente (Beta(5,1)) com o corte de 27 padrões, sendo que a média das soluções menos favoráveis também se encontra nesta distribuição com 22.9 padrões cortados. Este resultado encontra-se de acordo com a melhor solução encontrada para a Função Objetivo.

Na instância pequena a solução mais favorável é obtida para as distribuições decrescente (Beta(1,3)), não central (Beta(2,5)) e extremos (Beta(0.5,0.5)), com 16 padrões cortados, igualando o valor de padrões cortados na solução sem cenários aplicados. O valor mais favorável da Função Objetivo corresponde à solução da distribuição decrescente Beta(1,3), sendo esta considerada a melhor solução obtida. No entanto o valor médio de padrões cortados por distribuição é inferior na distribuição não central (Beta(2,5)) mas muito semelhante ao valor da distribuição decrescente (Beta(1,3)).

Sendo este problema um problema de minimização de *input* é pretendido que sejam cortados o menor número de padrões possível, no entanto para este problema é mais relevante obter uma solução melhor na Função Objetivo. Nas duas instâncias é verificada uma correspondência entre o menor número de padrões cortados e a melhor solução da Função Objetivo.

É possível analisar em detalhe o impacto da incerteza no corte de cada padrão, isto é, o número de vezes que cada padrão é cortado em cada cenário de cada distribuição. Isto permite ter algumas conclusões que podem ser úteis para a empresa. Por exemplo, para a instância pequena, pode verificar-se que a variabilidade do número de vezes que cada padrão é cortado, varia muito dependendo do padrão. No caso do padrão 14, verifica-se que o número de vezes que é cortado varia entre 0 e 16, sendo 16 o valor máximo de vezes que o padrão é cortado. Por outro lado, há padrões que nunca são cortados, 12 e 13, e outros que têm uma variabilidade muito pequena, como o 11 que apenas varia entre 0 e 2. Este tipo de orientação permite à empresa orientar para os padrões mais críticos os objetos que passaram por um controlo de qualidade mais apertado.

Capítulo 5. Conclusão

O estudo do problema de corte de matéria-prima considerando a existência de incerteza provou que o BRKGA é um método de solução eficaz para encontrar boas soluções num curto período. Ainda que nem sempre as soluções encontradas sejam soluções ótimas, é possível encontrar soluções muito favoráveis à resolução do problema.

A utilização da distribuição de probabilidades Beta para gerar cenários provou ser capaz de gerar um conjunto de cenários diversificados para considerar diferentes casos de incerteza possíveis de ocorrer.

Durante todo o trabalho foi mantido o objetivo de minimizar o corte de padrões e minimizar o custo da função objetivo. Com o estudo da incerteza verificou-se que é possível obter soluções próximas das condições de base, onde não é considerada incerteza.

Este trabalho é um trabalho inicial numa área que ainda se encontra em fase inicial de investigação sobre a qual as referências e resultados ainda são escassos. Com ele, pretende-se contribuir para perceber e analisar o impacto da incerteza em problemas que envolvam processos de corte, servindo de base para estudos futuros em que possam ser desenvolvidos métodos que considerem a incerteza de forma intrínseca.

Referências

- [1] Kantorovich, L. V. (1958). Mathematical Methods of Organizing as Planning Production, 1960 6:4, 366-422. <https://doi.org/10.1287/mnsc.6.4.366>
- [2] Alvarez-Valdes, R., Carravilla, M. A., Oliveira, J. F. (2018). Cutting and packing. In: Martí R., Pardalos P., Resende M. (eds) Handbook of Heuristics. Springer, Cham
- [3] Beasley, J. E. (1985). Algorithms for unconstrained two-dimensional guillotine Cutting. Journal of the Operational Research Society, 36(4), 297-306.
- [4] Morabito, R. N. & Arenales, M. N., (1992). An and-or-graph approach for two-dimensional cutting problems. European Journal of Operational Research 58, 263-271.
- [5] Alvarez-Valdés, R. & Parajón (2002). A tabu search algorithm for large-scale guillotine (un)constrained two-dimensional cutting problems. Computers & Operations Research, 29(7), pp.925-947.
- [6] Silva, E. (2011). Modelos e métodos de otimização para problemas de corte e empacotamento a duas dimensões. Tese de Doutorado. Universidade do Minho, Escola de Engenharia.
- [7] Dyckhoff, Harald. (1990). A Typology of Cutting and Packing Problems. European Journal of Operational Research. 44. 145-159. [https://doi.org/10.1016/0377-2217\(90\)90350-K](https://doi.org/10.1016/0377-2217(90)90350-K).
- [8] Wäscher, G., Haußner, H., & Schumann, H. (2007). An improved typology of cutting and packing problems. European Journal of Operational Research, 183(3), 1109-1130. <https://doi.org/10.1016/j.ejor.2005.12.047>
- [9] Li, Z., & Ierapetritou, M. G. (2008). Robust Optimization for Process Scheduling Under Uncertainty. 4148-4157.
- [10] Alem, D., & Morabito, R. (2013). O problema combinado de planejamento da produção e corte de estoque sob incertezas: aplicação em fábricas de móveis de pequeno porte. Gestão & Produção, 20(1), pp.111-133.
- [11] Alem, D., & Morabito, R. (2015). Planejamento da produção sob incerteza : programação estocástica versus otimização robusta. Gestao da Producao, 22(3), pp.539-551.
- [12] Zanjani, M. K., Nourelfath, M., & Ait-kadi, D. (2011). A Stochastic Programming Approach for Production Planning with Uncertainty in the Quality of Raw Materials: A Case in Sawmills, Journal of the Operational Research Society, 62(7), pp.1334-1343.
- [13] Ruszczyński, A. & Shapiro, A. (2003). Stochastic Programming, Handbook in Operations Research and Management Science, Elsevier Science, Amsterdam.
- [14] Kaut, M., & Wallace, S. W. (2003). Evaluation of scenario-generation methods for stochastic programming. Stochastic Programming E-Print Series, 14(1).
- [15] Mitra, S., Domenica, N. Di. (2010). A review of scenario generation methods, Int. J. Computing Science and Mathematics, Vol. 3, No. 3, 226-244.
- [16] Lempert, R., Groves, D. G., Popper, S. W. & Bankes, S. C. (2006). A General, analytic method for generating robust strategies and narrative scenarios. Management science, 52(4), pp.514-528.
- [17] Holland, J. H. (1975). Adaptation in natural and artificial systems. Ann Arbor, MI: University of Michigan Press.

- [18] Jafari-marandi, R., & Smith, B. K. (2017). Fluid Genetic Algorithm (FGA). *Journal of Computational Design and Engineering*, 4(2), 158-167. <https://doi.org/10.1016/j.jcde.2017.03.001>
- [19] Caruana, R. A., Schaffer, J. D., & Eschelman, L. J., (1989). Using multiple representations to improve inductive bias: Gray and binary coding for genetic algorithms. In *Proceedings of the sixth international workshop on machine learning*, pp. 375-378.
- [20] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning* (1st ed.). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- [21] Michalewicz Z. (2013). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Science & Business Media.
- [22] Gonçalves, J.F. & Resende, M.G.C., (2010). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487-525.
- [23] Bean, J. C., (1994). Genetic Algorithms and Random Keys for Sequencing and Optimization, *ORSA Journal on Computing*, Vol. 6, No. 2. <https://doi.org/10.1287/ijoc.6.2.154>
- [24] Resende, M. G. C. (2012). BRKGA tutorial Biased random-key genetic algorithms: A tutorial. Tutorial given at CLAIO/SBPO 2012.
- [25] Gonçalves, J. F. e Resende, M. G. C. (2018). Random-key genetic algorithms. *Handbook of Heuristics*. In: Martí R., Pardalos P., Resende M. (eds) *Handbook of Heuristics*. Springer, Cham.

Apêndices

Instâncias usadas

Instância grande

Padrão(j)	$\gamma(j)$	$w(j)$
1	27	0.15
2	14	0.19
3	23	1.05
4	16	1.21
5	28	1.25
6	19	1.75
7	16	1.78
8	18	1.8
9	20	2.38
10	18	2.44
11	19	2.64
12	29	2.66
13	14	2.66
14	17	2.76
15	19	2.87
16	20	3.03
17	18	3.08
18	14	3.21
19	21	3.23
20	27	3.23
21	19	3.26
22	25	3.29
23	22	3.31
24	12	3.44
25	21	3.68
26	11	3.79
27	19	3.99
28	18	4.01
29	27	4.44
30	23	4.52

T	30
P	100

teta	1000
beta	0.7

		a (i,j)													
		Padrão(j)													
Peça(i)	d(i)	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	94	3	3	2	2	3	4	5	5	4	5	7	6	4	4
2	93	6	1	3	5	4	4	3	3	5	5	1	3	5	3
3	89	2	4	5	2	4	3	1	5	5	2	3	1	2	5
4	88	5	4	4	5	3	1	6	0	3	3	5	4	2	2
5	85	3	4	3	3	4	4	4	3	4	4	1	4	5	5
6	85	3	5	2	4	2	3	2	5	5	4	4	6	2	2
7	84	5	2	2	3	4	3	5	5	3	4	4	3	6	4
8	83	3	5	7	3	6	6	5	0	2	6	2	4	2	5
9	83	6	4	5	5	2	5	5	3	3	6	2	4	3	3
10	82	2	1	5	4	4	3	3	4	2	3	1	5	6	2
11	80	2	1	4	4	6	5	3	3	4	4	4	1	3	5
12	80	4	3	2	4	1	2	3	4	1	2	5	3	1	3
13	79	7	2	6	3	5	5	3	1	6	5	4	1	4	5
14	78	2	2	3	4	6	3	2	5	6	5	3	1	2	5
15	78	2	5	3	1	5	6	5	4	3	1	5	3	1	2
16	78	4	5	3	4	5	3	5	1	4	2	3	4	3	3
17	78	5	2	1	1	1	6	1	4	1	5	3	5	3	1
18	78	3	4	6	4	2	4	4	3	5	3	6	2	5	5
19	77	5	5	3	2	3	3	1	3	5	0	2	2	5	2
20	77	3	5	4	3	0	5	5	2	5	0	5	2	2	2
21	76	7	5	1	6	3	4	1	2	3	2	1	3	3	5
22	75	4	5	4	3	5	6	6	3	6	3	3	2	4	5
23	75	4	3	3	3	6	4	3	3	6	6	5	5	5	4
24	75	4	3	5	2	5	2	1	5	2	1	2	3	2	2
25	75	4	5	1	4	3	3	2	7	3	2	4	1	6	5
26	74	6	1	5	4	3	5	4	4	4	4	6	1	5	5
27	74	7	1	5	3	4	2	5	0	2	1	6	3	4	6
28	74	5	4	5	6	3	5	4	4	4	1	6	4	4	5
29	74	2	1	3	4	3	3	3	1	3	6	5	2	5	5
30	73	1	5	4	2	3	1	4	5	2	2	4	2	5	2
31	73	5	4	5	3	3	3	6	2	4	6	2	2	3	3
32	71	3	2	3	6	6	6	3	3	3	4	4	4	2	5
33	71	6	4	6	5	6	2	4	4	5	2	4	3	4	2
34	71	2	2	4	6	1	3	4	5	2	5	4	3	6	2
35	70	5	5	5	4	3	5	2	6	6	5	3	4	2	3
36	69	6	5	3	2	6	5	4	3	4	2	6	5	3	3
37	69	3	6	5	1	3	3	5	3	5	3	5	5	6	4
38	68	2	5	6	2	4	3	3	2	2	2	5	5	4	3
39	66	1	2	3	4	4	5	1	5	5	4	7	3	4	2
40	66	4	2	5	4	6	5	2	3	4	3	3	5	1	1
41	65	4	2	1	5	2	4	1	2	5	1	2	3	4	4
42	60	3	0	3	5	5	1	1	3	3	0	3	1	6	2

		a (i,j)													
		Padrão(j)													
Peça(i)	d(i)	1	2	3	4	5	6	7	8	9	10	11	12	13	14
43	60	4	6	4	2	5	2	3	3	4	2	3	4	4	4
44	60	3	4	6	2	3	1	2	3	6	2	5	2	2	1
45	59	2	2	5	2	2	4	4	2	5	5	1	4	2	6
46	59	3	2	5	6	5	7	1	1	2	2	3	4	6	5
47	58	2	4	2	5	4	0	5	4	5	2	2	1	4	4
48	58	5	3	5	3	4	4	3	5	3	2	2	3	3	4
49	57	3	4	3	3	1	6	3	2	3	4	2	5	6	3
50	57	1	2	6	4	3	4	6	1	1	2	4	6	6	6
51	56	2	4	1	2	1	5	7	5	1	4	3	5	7	5
52	56	6	4	3	3	3	7	4	3	3	1	2	6	5	2
53	55	4	2	3	5	4	4	5	1	3	5	2	3	5	5
54	52	4	2	2	6	6	1	5	5	5	3	5	3	2	4
55	51	3	3	4	3	6	6	4	4	4	1	6	3	2	2
56	51	4	4	2	4	1	5	4	4	2	3	2	1	3	3
57	50	5	5	2	2	5	6	2	2	1	2	2	5	5	2
58	49	5	7	4	4	6	1	4	1	1	1	4	3	2	3
59	48	2	0	4	4	4	4	3	5	1	3	5	4	3	6
60	48	3	6	3	3	6	5	7	5	4	4	2	5	4	5
61	48	3	4	4	1	5	6	3	1	4	3	5	1	3	5
62	47	4	5	3	5	4	3	3	1	5	4	1	2	4	4
63	47	4	6	6	2	2	4	4	3	6	4	2	2	3	5
64	47	5	2	5	3	5	5	3	5	5	1	5	2	3	5
65	45	4	3	7	2	4	5	1	5	4	3	4	4	1	4
66	44	6	5	5	3	4	3	5	6	4	3	4	5	5	4
67	43	1	4	3	4	3	2	3	4	6	2	6	5	4	3
68	43	2	5	4	3	3	4	1	2	2	1	4	3	6	4
69	43	2	2	5	2	4	3	3	4	5	3	5	3	4	3
70	42	4	5	4	5	4	1	0	2	5	5	2	2	4	6
71	41	1	2	2	1	2	3	7	2	3	5	6	1	4	4
72	41	4	1	5	3	0	2	6	2	1	6	6	4	4	6
73	41	2	5	4	6	3	4	1	5	4	3	5	2	6	2
74	40	1	3	2	2	4	6	2	2	2	2	6	1	4	3
75	40	4	3	3	2	4	2	2	2	2	5	3	6	5	3
76	39	4	2	4	5	6	1	1	6	5	4	4	5	2	3
77	39	1	2	2	5	4	5	2	6	5	2	5	3	2	2
78	38	4	4	3	5	2	3	5	5	1	1	2	5	0	4
79	38	4	1	3	4	2	5	6	5	4	6	3	6	3	4
80	38	1	4	4	4	4	4	5	4	6	6	5	3	3	2
81	37	2	4	4	1	0	5	3	4	3	2	6	3	7	4
82	37	4	2	4	2	4	4	3	5	0	6	3	5	6	6
83	35	5	2	4	1	3	3	5	1	1	3	6	4	5	3
84	35	7	6	6	0	6	1	5	6	1	3	4	5	4	6

		a (i,j)													
		Padrão(j)													
Peça(i)	d(i)	1	2	3	4	5	6	7	8	9	10	11	12	13	14
85	35	6	6	2	0	3	5	4	4	2	4	3	1	2	3
86	32	3	2	5	2	6	3	3	4	3	6	7	2	5	4
87	31	4	3	6	2	4	4	1	1	3	7	3	5	4	3
88	31	4	4	2	5	1	2	5	3	3	1	4	2	6	3
89	29	5	1	3	4	5	5	7	1	2	1	2	3	4	4
90	28	4	4	5	2	2	2	3	2	7	2	4	2	4	6
91	23	5	4	5	1	5	4	6	3	5	2	3	4	6	3
92	22	3	4	1	1	3	6	6	3	4	4	2	2	3	3
93	21	1	3	3	4	3	4	3	4	2	2	1	2	4	6
94	20	1	5	4	3	4	2	3	2	5	5	3	2	3	3
95	15	6	5	3	2	4	1	3	3	3	5	3	5	2	5
96	15	2	4	2	2	6	4	4	7	2	5	1	6	1	5
97	15	6	5	3	3	6	1	1	2	2	3	2	1	3	4
98	14	5	2	1	1	4	4	2	5	2	3	6	5	2	4
99	14	4	3	2	5	2	2	3	1	5	2	2	6	5	0
100	12	4	1	4	5	4	3	4	3	4	2	6	3	4	1

		a (i,j)															
		Padrão(j)															
Peça(i)	d(i)	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	94	7	2	1	1	3	2	3	2	2	3	4	2	6	1	2	4
2	93	3	3	3	2	4	1	4	4	1	4	1	2	4	2	3	2
3	89	5	4	2	3	1	3	5	4	3	4	1	3	1	4	4	2
4	88	3	3	5	4	5	6	5	5	3	2	5	3	5	4	4	4
5	85	4	1	2	1	5	4	4	5	3	4	4	2	1	6	2	3
6	85	4	0	4	4	1	3	5	2	4	4	1	4	5	4	5	3
7	84	2	4	5	3	4	6	3	5	4	0	5	1	4	4	4	5
8	83	1	2	2	6	2	2	1	5	2	3	2	4	3	1	2	1
9	83	5	3	1	5	5	1	1	4	5	4	2	6	5	1	5	4
10	82	5	3	4	5	0	1	6	4	1	5	2	5	4	3	2	6
11	80	5	3	2	5	3	3	1	2	5	6	2	3	2	3	2	5
12	80	2	5	0	4	2	2	3	5	4	3	1	2	4	2	5	2
13	79	2	4	4	2	2	3	5	5	1	4	4	4	2	5	5	1
14	78	4	2	3	3	6	6	3	4	2	2	1	4	6	2	5	2
15	78	3	3	5	6	5	2	3	3	3	3	3	5	3	7	4	5
16	78	2	5	1	3	4	3	5	5	3	2	3	6	1	5	4	5
17	78	5	2	4	4	2	5	5	2	3	3	6	3	4	2	4	5
18	78	3	3	3	6	5	2	3	2	5	1	4	6	5	3	5	4
19	77	6	4	5	2	5	3	2	3	5	3	4	6	3	2	4	2

		a (i,j)																
		Padrão(j)																
Peça(i)	d(i)	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
20	77	6	1	2	6	4	4	3	2	5	4	5	6	4	5	3	4	
21	76	1	6	3	2	4	2	5	3	3	4	6	3	2	4	4	4	
22	75	3	1	6	6	4	1	1	4	5	1	4	3	1	1	2	4	
23	75	5	4	3	1	5	3	4	6	4	2	4	2	3	5	4	4	
24	75	2	5	0	2	3	4	3	2	4	5	3	2	3	4	3	5	
25	75	4	3	6	4	5	5	2	4	2	1	4	3	2	6	2	2	
26	74	3	6	4	6	4	4	1	5	0	4	2	6	3	4	5	5	
27	74	1	1	2	4	5	2	2	2	5	4	2	3	1	1	2	6	
28	74	3	5	5	3	5	2	2	2	4	4	5	7	2	2	2	6	
29	74	6	4	6	3	3	5	3	6	3	5	5	4	4	5	4	6	
30	73	2	5	2	3	2	6	6	2	5	1	2	3	4	1	5	2	
31	73	4	4	4	4	6	5	3	6	5	1	6	3	6	2	4	6	
32	71	1	1	3	1	3	3	5	2	1	6	1	2	4	5	5	1	
33	71	3	3	3	4	1	4	3	2	2	5	4	6	3	5	1	3	
34	71	3	4	3	2	4	6	2	4	2	6	4	1	5	2	3	6	
35	70	4	1	2	4	1	5	2	2	3	3	4	0	4	2	2	3	
36	69	3	2	4	2	3	4	5	4	3	5	4	4	2	2	4	4	
37	69	1	6	3	3	5	3	2	2	2	5	4	2	2	3	4	3	
38	68	1	2	5	2	1	4	5	5	3	6	2	6	5	6	7	5	
39	66	3	1	1	4	4	4	1	5	2	1	3	1	1	4	0	3	
40	66	6	1	5	4	4	1	5	3	3	4	3	5	5	4	1	3	
41	65	5	3	5	2	5	7	2	5	0	1	3	2	3	1	5	4	
42	60	3	2	3	5	2	2	2	1	1	2	4	5	5	5	4	5	
43	60	2	4	5	3	4	2	1	1	5	1	5	4	3	5	4	2	
44	60	3	1	3	2	3	6	4	2	1	2	4	1	4	4	5	2	
45	59	1	1	1	6	2	5	2	5	3	2	4	6	3	5	7	3	
46	59	4	6	3	5	6	1	5	3	5	4	3	3	6	4	3	5	
47	58	4	3	4	2	3	2	1	6	4	3	4	4	5	4	5	4	
48	58	1	5	5	5	5	6	2	4	3	4	4	4	4	2	4	4	
49	57	2	6	1	4	5	2	6	4	6	4	1	4	1	3	5	4	
50	57	4	2	2	1	3	1	1	4	3	2	3	2	6	5	3	3	
51	56	4	1	3	5	3	5	6	3	5	5	3	5	2	5	4	5	
52	56	3	5	1	3	1	4	3	5	5	5	2	4	5	2	6	6	
53	55	6	0	2	2	2	4	4	5	4	5	6	3	3	5	4	4	
54	52	2	4	6	2	3	4	4	3	1	0	1	3	6	2	3	3	
55	51	6	2	4	3	6	4	1	2	4	4	2	6	4	0	3	4	
56	51	2	5	5	4	5	3	2	3	5	5	3	4	6	2	1	3	
57	50	1	3	5	2	5	1	2	6	6	2	1	2	2	4	2	1	
58	49	5	7	2	3	2	6	4	4	1	2	2	5	3	5	4	1	
59	48	1	2	3	3	5	3	2	4	2	3	3	5	3	5	0	4	
60	48	0	5	6	3	2	6	4	3	4	6	3	2	1	4	6	2	
61	48	1	2	2	3	3	1	5	4	3	2	3	4	2	3	6	1	

		a (i,j)																	
		Padrão(j)																	
Peça(i)	d(i)	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		
62	47	3	4	2	2	2	6	2	5	6	2	1	0	3	1	1	5		
63	47	1	6	6	4	2	3	5	4	2	6	5	5	0	1	1	3		
64	47	3	4	2	3	5	4	4	5	7	3	1	3	2	2	3	5		
65	45	6	2	6	3	2	5	4	4	5	5	3	2	4	3	5	3		
66	44	4	6	2	2	4	4	1	3	2	4	2	2	2	4	2	4		
67	43	2	5	4	3	5	6	2	3	1	4	5	3	6	4	2	3		
68	43	4	3	6	5	1	5	3	6	5	3	6	2	4	2	4	4		
69	43	4	4	5	5	1	1	3	4	2	7	4	2	2	0	4	5		
70	42	1	5	4	1	3	2	4	1	4	3	1	4	2	5	4	1		
71	41	1	5	4	4	1	6	5	4	5	2	6	2	2	3	4	2		
72	41	5	4	3	2	5	4	7	4	1	4	3	4	3	5	1	3		
73	41	4	3	1	1	3	1	4	5	5	6	4	4	6	4	1	5		
74	40	4	2	3	7	4	5	1	4	2	4	4	2	6	3	3	2		
75	40	2	3	5	3	4	1	4	6	3	5	1	1	2	6	6	3		
76	39	7	3	6	5	6	3	4	4	4	4	5	7	1	1	1	2		
77	39	6	3	3	3	5	3	2	7	1	2	3	3	4	1	3	6		
78	38	5	6	3	2	7	2	5	3	1	5	2	1	2	5	4	1		
79	38	4	4	3	3	1	2	2	6	4	4	5	5	2	5	1	3		
80	38	1	4	1	3	4	3	5	3	2	4	3	4	4	5	3	3		
81	37	0	4	2	3	2	3	3	6	3	1	2	1	4	4	2	6		
82	37	4	7	6	6	6	6	3	6	3	2	3	4	2	6	6	3		
83	35	1	6	3	5	5	0	2	2	5	3	3	4	1	1	6	4		
84	35	4	3	6	5	3	1	1	2	5	5	3	1	3	3	2	5		
85	35	4	6	6	6	4	5	5	3	2	3	4	6	6	5	7	2		
86	32	2	2	3	4	5	6	1	4	1	3	4	4	2	5	4	3		
87	31	3	5	4	3	3	2	5	2	3	3	5	5	2	2	4	2		
88	31	3	5	0	5	7	6	2	3	2	2	3	6	2	3	6	6		
89	29	4	6	3	7	4	6	4	5	3	3	3	4	6	1	3	1		
90	28	4	5	1	5	5	2	2	2	3	6	2	5	2	1	1	4		
91	23	1	5	5	3	1	1	2	6	3	4	6	4	4	3	5	5		
92	22	2	2	4	7	5	4	5	4	4	2	6	3	3	4	1	3		
93	21	3	3	3	1	4	1	4	3	3	3	0	2	5	5	1	3		
94	20	6	2	3	2	1	6	2	6	3	6	5	6	1	5	4	4		
95	15	1	5	5	3	4	5	1	3	4	2	4	2	1	2	4	6		
96	15	3	5	3	2	6	3	3	2	3	5	4	5	4	2	3	2		
97	15	1	5	6	2	5	5	6	1	2	4	1	3	3	6	5	4		
98	14	6	2	1	1	2	4	2	4	2	5	4	4	3	6	1	4		
99	14	5	5	1	2	1	3	3	3	5	2	2	4	4	1	6	5		
100	12	3	4	0	4	3	6	4	2	4	4	2	1	2	6	1	6		

Instância pequena

Padrão (j)	$\gamma(j)$	$w(j)$
1	19	0.81
2	16	1.11
3	14	1.12
4	15	1.36
5	17	1.39
6	23	1.88
7	25	2.12
8	14	2.28
9	13	2.5
10	17	2.52
11	15	2.98
12	22	3.04
13	12	3.24
14	12	3.56
15	15	3.77

T	15
P	50

teta	1000
beta	0.7

		a (i,j)														
		Padrão (j)														
Peça (i)	d (i)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	89	4	6	5	4	5	2	4	4	4	1	2	5	1	5	1
2	81	2	6	6	3	2	1	4	4	0	3	5	2	4	4	2
3	80	5	5	0	4	4	3	5	1	4	1	2	5	3	5	2
4	79	4	2	5	4	4	3	4	4	2	1	6	5	4	5	3
5	76	3	2	2	4	4	5	2	3	5	5	2	6	2	3	2
6	76	4	2	2	3	5	2	1	3	3	4	4	2	5	2	5
7	76	5	5	5	5	4	1	3	4	2	5	5	5	3	5	2
8	74	6	2	2	5	1	5	2	4	2	5	5	1	4	3	5
9	73	2	1	5	4	4	1	5	2	3	2	2	5	5	6	6
10	72	3	1	1	2	6	5	6	2	5	5	1	5	2	2	4
11	72	3	3	4	3	2	4	2	6	5	3	5	1	3	2	3
12	72	3	6	3	1	3	1	5	2	2	2	1	3	4	2	5
13	70	2	3	1	1	1	5	2	6	2	2	4	4	2	2	2
14	69	6	5	2	4	5	2	3	2	3	5	2	3	5	4	4
15	68	3	5	2	0	5	3	4	4	2	4	4	5	5	3	3
16	66	3	3	4	3	5	2	5	3	0	2	3	1	5	1	4
17	64	6	4	2	6	2	2	6	6	3	0	0	4	2	4	2

		a (i,j)														
		Padrão (j)														
Peça (i)	d (i)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
18	64	6	4	2	5	6	3	4	5	5	6	5	5	6	5	3
19	62	4	7	6	1	4	3	5	2	3	3	2	4	4	4	4
20	62	4	4	1	2	4	3	6	3	6	4	2	3	3	6	6
21	62	6	2	1	2	5	4	3	6	4	3	5	4	2	4	2
22	61	2	2	3	6	6	6	4	1	3	6	5	0	2	1	2
23	61	6	2	2	1	5	6	2	2	5	3	4	3	3	1	4
24	59	1	4	1	4	3	3	2	3	4	3	4	5	4	4	3
25	58	3	3	5	7	1	6	6	5	4	6	2	1	3	6	2
26	58	4	2	6	1	4	2	2	1	4	5	5	3	3	4	4
27	58	4	4	4	6	4	5	6	4	4	2	4	4	4	4	2
28	58	4	2	2	2	1	4	5	0	5	4	2	0	6	3	1
29	56	4	6	4	6	3	3	4	3	5	2	4	3	3	3	3
30	52	1	6	5	1	3	1	7	3	3	3	4	2	3	6	3
31	52	3	4	7	4	3	2	3	6	2	6	6	1	5	3	5
32	51	1	3	4	4	4	6	3	2	3	4	2	4	4	4	3
33	49	3	5	3	1	4	6	2	1	5	3	3	4	5	2	4
34	48	6	3	3	2	6	1	5	5	4	2	4	5	1	1	4
35	47	4	0	1	4	4	2	6	3	6	6	2	4	5	5	3
36	47	6	4	4	2	5	5	4	5	5	4	5	3	3	3	3
37	46	1	3	2	5	6	3	4	1	4	5	5	4	2	3	5
38	44	5	6	0	4	5	5	4	2	3	4	2	1	1	4	3
39	37	3	2	4	4	1	3	6	2	6	5	1	6	5	4	2
40	27	6	6	4	1	4	4	3	2	5	5	3	2	3	4	3
41	27	4	1	3	2	4	6	6	1	6	1	6	4	5	3	3
42	27	5	4	4	1	0	3	3	2	4	2	1	4	4	2	2
43	26	6	6	6	3	3	4	1	5	3	4	2	1	1	2	5
44	25	3	1	5	4	5	2	1	3	4	2	6	3	7	5	6
45	23	3	4	2	6	4	1	2	4	3	4	4	4	5	4	5
46	22	1	4	6	2	4	5	5	4	1	1	4	2	2	3	4
47	20	6	5	4	3	1	2	1	5	6	3	2	4	3	3	5
48	17	1	2	4	4	1	4	5	5	1	2	2	2	1	3	2
49	16	6	4	5	2	1	4	3	4	5	4	5	3	2	3	2
50	13	5	1	4	3	1	0	5	4	5	4	2	1	3	5	2