



Geração Automática de horários para transporte ferroviário na TRENMO Engenharia, Lda.

Jorge Manuel Calheiros Carneiro
LEIC 2006

(047.3) LEIC
02 2006/CARj

**Faculdade de Engenharia da Universidade do Porto
Licenciatura em Engenharia Informática e Computação**



**Geração Automática de horários para transporte ferroviário na
TRENMO Engenharia, Lda.**

Relatório do Estágio Curricular da LEIC 2005/06

Jorge Manuel Calheiros Carneiro

Orientador na FEUP: Prof. Rui Camacho

Orientadora na TRENMO Engenharia, Lda: Eng^a. Oana Santos

Setembro de 2006

CONFIDENCIAL

Faculdade de Engenharia da Universidade do Porto
Licenciatura em Engenharia Informática e Computação



Gestão Automática de horários para transporte ferroviário na
TRENMO Engenharia, Lda.

Relatório do Estágio Curricular da LEIC 2006/08

Jorge Manuel Calheiros Carneiro

Orientador na FEUP: Prof. Rui Camacho
Orientadora na TRENMO Engenharia, Lda: Eng.ª Ana Santos

004(047-3) LEIC LEIC5202 2006/CARJ

Universidade do Porto Faculdade de Engenharia Biblioteca 1
Nº 389218
CDU 004.4(047-3)
Data 26/1/2007

Setembro de 2008

Resumo

No transporte público de passageiros, a geração de horários equilibrados é um problema bastante complexo. No caso do transporte ferroviário, este problema é ainda maior, dado que nem todos os horários são possíveis de pôr em prática, por restrições relacionadas com os veículos ou com a própria infra-estrutura da rede ferroviária. Assim sendo, para um dado horário, para além de ser necessário analisar a sua qualidade, é necessário verificar a sua validade.

Até hoje, este era um problema que era resolvido manualmente, e posteriormente analisado em software de modelação, que não tinha sido criado especificamente para este caso.

Foi nesta perspectiva que foi lançada a proposta de estágio "Geração Automática de Horários Ferroviários" pela TRENMO, Engenharia Lda, no âmbito do estágio curricular da Licenciatura em Engenharia Informática e de Computação, que tinha como objectivo o desenvolvimento de uma ferramenta que resolvesse este problema, gerando automaticamente os horários.

Numa primeira fase foi feito um estudo sobre as soluções actuais existentes no mercado, quais as suas potencialidades e as suas lacunas, justificando assim o porquê de partir para a criação de um software de raiz, ao invés de usar um dos produtos já existentes.

Em seguida foi feito um levantamento dos *Que a realidade continue a superar a imaginação* uma análise dos mesmos. Esta fase envolvia igualmente uma análise das tecnologias sobre as quais se poderia trabalhar.

Passou-se então à fase de desenvolvimento, desenvolvimento este que foi feito na linguagem Java, durante esta fase surgiram alguns requisitos novos, que obrigaram a reformular o problema.

Resumo

No transporte público de passageiros, a geração de horários equilibrados é um problema bastante complexo. No caso do transporte ferroviário, este problema é ainda maior, dado que nem todos os horários são possíveis de pôr em prática, por restrições relacionadas com os veículos ou com a própria infra-estrutura da rede ferroviária. Assim sendo, para um dado horário, para além de ser necessário analisar a sua qualidade, é necessário verificar a sua validade.

Até hoje, este era um problema que era resolvido manualmente, e posteriormente analisado em software de modelação, que não tinha sido criado especificamente para este caso.

Foi nesta perspectiva que foi lançada a proposta de estágio “Geração Automática de Horários Ferroviários” pela TRENMO, Engenharia Lda. no âmbito do estágio curricular da Licenciatura em Engenharia Informática e de Computação, que tinha como objectivo o desenvolvimento de uma ferramenta que resolvesse este problema, gerando automaticamente os horários.

Numa primeira fase foi feito um estudo sobre as soluções actuais existentes no mercado, quais as suas potencialidades e as suas lacunas. Justificando assim o porquê de partir para a criação de um software de raiz, ao invés de usar um dos produtos já existentes.

Em seguida foi feito um levantamento dos requisitos da ferramenta a desenvolver, seguida de uma análise dos mesmos. Esta fase envolveu, igualmente, uma análise das tecnologias sobre as quais se poderia trabalhar.

Passou-se então à fase de desenvolvimento, desenvolvimento este que foi feito na linguagem Java, durante esta fase surgiram alguns requisitos novos, que obrigaram a reformular o problema.

Agradecimentos

Primeiro que tudo queria agradecer ao meu pai, pela confiança e empenho com que me ajudou a cumprir mais um objectivo.

Queria agradecer ao professor Álvaro Costa e à minha orientadora Eng^a. Oana Santos, pela oportunidade que me foi dada de realizar o estágio na TRENMO, Engenharia Lda.

Ao Rui Fernandes, à Vera Ferreira e à Mafalda Ferreira queria agradecer por todo o apoio prestado, toda a disponibilidade demonstrada, e sobretudo pelo facto de durante estes seis últimos meses terem sido para além de excelentes colegas, excelentes amigos.

Queria igualmente agradecer ao meu orientador, o prof. Rui Camacho, por todos os conselhos dados, e pela disponibilidade demonstrada.

Aos cerca de quarenta colegas com quem tive a oportunidade de trabalhar no decorrer do curso, que me permitiram não só chegar até aqui, como me permitiram coleccionar uma grande quantidade de bons momentos, que me no futuro me farão recordar os anos passados na FEUP.

Queria também agradecer ao resto da minha família, em especial ao meu irmão e aos meus avós, assim como a todos os amigos que de uma forma ou de outra me apoiaram e me ajudaram durante os últimos tempos.

6	Implementação	25
6.1	Arquitectura	25
6.1.1	Modelo de Classes	25
6.1.2	Arquitectura dos Módulos	30
6.1.3	Nomenclatura	31
6.2	Estruturas de dados utilizadas	31
6.3	Algoritmos utilizados / descritos	34
6.3.1	Funcionamento do Motor de Pesquisa	34
6.3.2	Funcionamento do editor de dados	42
6.3.3	Outros algoritmos utilizados	48
6.4	Solução implementada	49
7	Conclusões	57
7.1	Análise dos resultados obtidos	57
7.2	Perspectivas para melhorias futuras	57
7.3	Principais dificuldades	57
7.4	Conclusões pessoais	58
	Referências e Bibliografia	59
ANEXO A	Glossário	60
ANEXO B	Explicação da fórmula de cálculo do número de hipóteses	61
ANEXO C	Soluções Encontradas	62

Índice de Conteúdos

1	Introdução.....	2
1.1	Apresentação da Instituição de Estágio (TRENMO Engenharia, Lda.).....	2
1.2	O Projecto Geração Automática de horários ferroviários na TRENMO Engenharia, Lda.	3
1.3	Organização do Relatório	3
2	Descrição do Problema	5
2.1	Situação Prévia.....	5
2.2	Descrição Geral	8
3	Estado da Arte	10
4	Requisitos.....	12
4.1	Requisitos Funcionais.....	12
4.1.1	Regras para a construção de um horário ferroviário	14
4.2	Requisitos Tecnológicos.....	20
4.3	Requisitos Não Funcionais	21
5	Análise do Problema	22
5.1	Análise prévia	22
5.2	Análise Tecnológica.....	22
5.3	Planeamento.....	24
6	Implementação	25
6.1	Arquitectura.....	25
6.1.1	Modelo de Classes	25
6.1.2	Arquitectura das Interfaces.....	30
6.1.3	Nomenclatura	31
6.2	Estruturas de dados usadas	31
6.3	Algoritmos utilizados / desenvolvidos	34
6.3.1	Funcionamento do Motor de Pesquisa	34
6.3.2	Funcionamento do editor de dados	42
6.3.3	Outros algoritmos testados.....	48
6.4	Solução implementada	49
7	Conclusões.....	57
7.1	Análise dos resultados obtidos	57
7.2	Perspectivas para melhoramentos futuros.....	57
7.3	Principais dificuldades	57
7.4	Conclusões pessoais.....	58
	Referências e Bibliografia.....	59
ANEXO A:	Glossário	60
ANEXO B:	Explicação da fórmula de cálculo do número de hipóteses	61
ANEXO C:	Soluções Encontradas	62

Índice de Figuras

Figura 1 – Organograma da TRENMO, Engenharia Lda.....	2
Figura 2 – Rede actual (simplificada) do Metro do Porto	6
Figura 3 - Ficheiro de carregamento de Linhas em <i>EMME/2</i>	7
Figura 4 – Gráfico desenhado pelo Software <i>EMME/2</i>	8
Figura 5 – Cruzamento da Senhora da Hora.....	16
Figura 6 – Cruzamento da Fonte do Cuco	16
Figura 7 – Diferenças de velocidade entre a Linha B serviço Expresso e serviço Normal.....	17
Figura 8 – Ligação Virtual Aeroporto – Póvoa de Varzim e Póvoa de Varzim – Aeroporto ..	18
Figura 9 – Sincronização de viagens do Expresso	18
Figura 10 – Linha E (Aeroporto) sincronizada.....	19
Figura 11 – Diagrama de Gantt relativo ao Projecto	24
Figura 12 – Modelo de Classes do software desenvolvido	26
Figura 13 – Fragmento do Ficheiro constraint.ugt	28
Figura 14 – Demonstração das restrições:.....	29
Figura 15 – Ficheiro constraint.ugt	30
Figura 16 - Funcionamento do Software Desenvolvido.....	34
Figura 17 – Pseudo – Código para a procura da próxima circulação a atribuir	35
Figura 18 – Pseudo – Código para contagem de veículos.....	40
Figura 19 – Parte de ficheiro HTML gerado pelo software	41
Figura 20 – Descrição dos nós no ficheiro d211.in	43
Figura 21 – Descrição dos arcos no ficheiro d211.in	43
Figura 22 – Descrição dos arcos no ficheiro d221.in	44
Figura 23 – Gráfico de Horário	48
Figura 24 – Interface de entrada do editor de dados.....	50
Figura 25 – Menu de Selecção de Opções.....	51
Figura 26 – Menu de Edição de Nós	52
Figura 27 – Janela de Edição de Arcos.....	53
Figura 28 – Janela de Edição de Linhas	54
Figura 29 – Janela de Edição da solução	55
Figura 30 – Gráfico desenhado pelo software	56

Índice de Tabelas

Tabela 1 – Análise Comparativa entre os produtos existentes no mercado.....	11
Tabela 2 – Comparação de Eficiência entre recuo com e sem cortes.....	38
Tabela 3 - Solução para o cenário C1.....	62
Tabela 4 – Solução para o cenário C3.....	63
Tabela 5 – Solução para o cenário C4.....	64
Tabela 6 – Solução para o cenário C5.....	65
Tabela 7 – Solução encontrada para o cenário C7.....	66
Tabela 8 – Solução encontrada para o cenário C9.....	67
Tabela 9 – Solução encontrada para o cenário 6A + 4B + 4C + 3E + 1EX.....	68

Figura 1 – Organização da TRENMO Engenharia Lda.

1 Introdução

Na Licenciatura em Engenharia Informática e de Computação, o habitual projecto de fim de curso, é substituído por um estágio curricular para a duração de seis meses, no qual será realizado um projecto designado ao projecto de estágio. Com este relatório pretende-se apresentar o projecto "Geração automática de horários ferroviários", realizado pelo autor intitulado na TRENMO Engenharia, Lda.

1.1 Apresentação da Instituição de Trabalho (TRENMO Engenharia, Lda.)

A TRENMO Engenharia Lda. (que daqui em diante será referida apenas como TRENMO) é uma empresa criada em 2005 a partir de um estágio de doutoramento desenvolvido na Faculdade de Engenharia da Universidade do Porto (FEUP), na área de modelação de transportes.

Com um ano de existência conta actualmente com onze colaboradores, divididos entre a sede no Porto e a delegação em Lisboa.

Apoiada em tecnologias avançadas e com forte domínio do conhecimento, a TRENMO caracteriza-se pela oferta de soluções inovadoras na área dos transportes.

Estas singularidades permitiram o primeiro reconhecimento desta empresa no programa NEST, lançado pela Agência de Inovação, SA (ANI), que apoia e promove o aprofundamento das relações entre o mundo da investigação e o mundo empresarial português.

De igual modo, a filosofia de TRENMO e a sua ligação com a FEUP foram reconhecidas através do programa NEOTEC, que apoia os projectos de criação e desenvolvimento de novas empresas de base tecnológica, integradas nas instituições do sistema científico e académico.

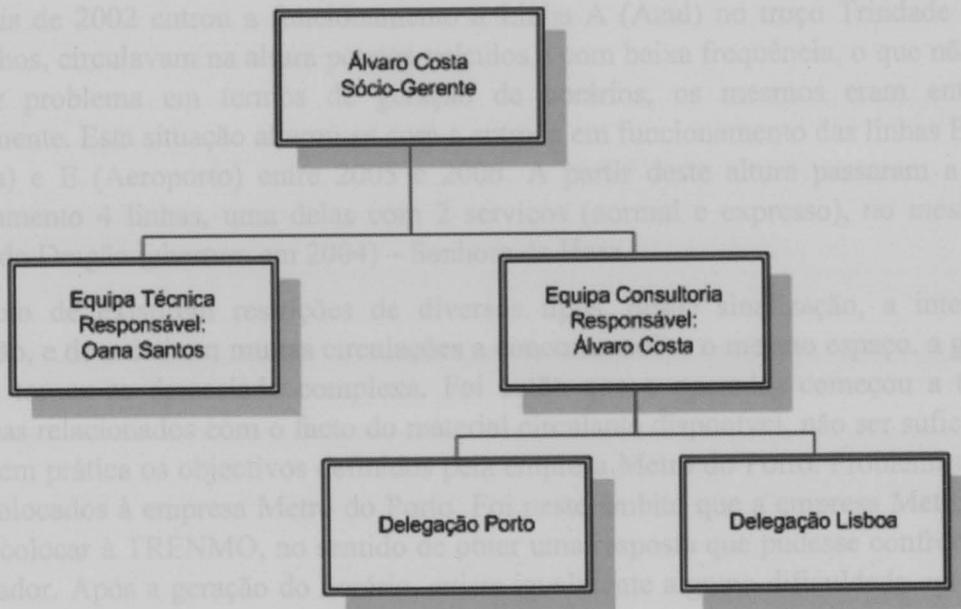


Figura 1 – Organograma da TRENMO, Engenharia Lda.

1 Introdução

Na Licenciatura em Engenharia Informática e de Computação, o habitual projecto de fim de curso, é substituído por um estágio curricular com a duração de seis meses, no qual será realizado um projecto designado de projecto de estágio. Com este relatório pretende-se apresentar o projecto “Geração automática de horários ferroviários”, realizado pelo autor intitulado na TRENMO Engenharia, Lda.

1.1 Apresentação da Instituição de Estágio (TRENMO Engenharia, Lda.)

A TRENMO, Engenharia Lda. (que doravante será referida apenas como TRENMO) é uma empresa criada em 2005 a partir de um *spin-off* do conhecimento desenvolvido na Faculdade de Engenharia da Universidade do Porto (FEUP), na área da modelação de transportes.

Com um ano de existência conta actualmente com onze colaboradores, divididos entre a sede no Porto e a delegação em Lisboa.

Apoiada em tecnologias avançadas e num forte domínio do conhecimento, a TRENMO caracteriza-se pela oferta de soluções inovadoras na área dos transportes.

Estas singularidades permitiram o perfeito enquadramento desta empresa no programa NEST, lançado pela Agência de Inovação, SA (Adi), que apoia e promove o aprofundamento das relações entre o mundo da investigação e o tecido empresarial português.

De igual modo, a filosofia da TRENMO e a sua ligação com a FEUP foram reconhecidas através do programa NEOTEC, que apoia os projectos de criação e desenvolvimento de novas empresas de base tecnológica, integrados em instituições do sistema científico e académico.

1.2 O Projecto Geração Automática de horários ferroviários na TRENMO Engenharia, Lda.

Em finais de 2002 entrou a funcionamento a Linha A (Azul) no troço Trindade – Sr. dos Matosinhos, circulavam na altura poucos veículos e com baixa frequência, o que não causava qualquer problema em termos de geração de horários, os mesmos eram então feitos manualmente. Esta situação alterou-se com a entrada em funcionamento das linhas B (Póvoa), C (Maia) e E (Aeroporto) entre 2005 e 2006. A partir deste altura passaram a estar em funcionamento 4 linhas, uma delas com 2 serviços (normal e expresso), no mesmo troço: Estádio do Dragão (abertura em 2004) – Senhora da Hora.

Pelo facto de existirem restrições de diversos tipos, desde sinalização, a intervalos de circulação, e de existirem muitas circulações a concorrer sobre o mesmo espaço, a geração de horários tornou-se demasiado complexa. Foi então que o operador começou a ter alguns problemas relacionados com o facto do material circulante disponível, não ser suficiente para colocar em prática os objectivos definidos pela empresa Metro do Porto. Problemas estes que foram colocados à empresa Metro do Porto. Foi neste âmbito que a empresa Metro do Porto decidiu colocar à TRENMO, no sentido de obter uma resposta que pudesse confrontar com a do operador. Após a geração do horário, existe igualmente alguma dificuldade em qualificar um horário, isto é, em fazer a distinção entre um bom horário e um menos bom.

A TRENMO Engenharia, Lda. apesar de não ter a seu cargo a definição dos horários, na qualidade de consultora da Metro do Porto, produzia horários que pudessem aquilatar da qualidade dos definidos pela operadora. Até este momento, na TRENMO, a sua produção era feita manual e mentalmente. No entanto a instituição de estágio utilizava para as suas actividades um software de planeamento estratégico de transportes (*EMME/2*) que entre muitas das suas funcionalidades contém a representação gráfica de horários.

No entanto o problema principal mantinha-se, a geração de um horário que cumprisse todas as restrições estabelecidas. E foi neste âmbito que se propôs como estágio curricular a análise e produção de um software que gerasse os horários para resolver este caso em concreto e outros que mais tarde surgissem.

1.3 Organização do Relatório

Este relatório é composto por sete capítulos nos quais é descrito de forma detalhada o presente projecto de estágio. Seguem-se alguns anexos com informação mais detalhada e que visa complementar o restante conteúdo deste relatório.

No presente capítulo é feita uma introdução aos conteúdos do documento, sendo feita uma breve apresentação da instituição onde decorreu o estágio, uma, igualmente breve, descrição do projecto Geração Automática de Horários Ferroviários, e uma descrição da organização do presente relatório.

No capítulo seguinte, Descrição do Problema, é feita uma descrição geral da situação actual e do problema geral que este estágio pretende resolver.

No capítulo Estado da Arte é feita uma análise detalhada ao mercado, justificando a opção de construir um software de raiz em vez de utilizar um outro já existente.

No capítulo 4 são descritos os requisitos funcionais e não funcionais que o software a desenvolver terá de ter. No capítulo seguinte, uma análise do problema, em termos tecnológicos, e em termos temporais.

Em seguida é apresentada a arquitectura do sistema, assim como alguns detalhes de implementação. São igualmente descritos e documentados alguns algoritmos usados.

O último capítulo contém as conclusões alcançadas com o trabalho. Em anexo encontram-se alguns resultados obtidos para os cenários previstos.

- Junho de 2004 – em vésperas do euro 2004 é inaugurado o troço Trindade – Estádio do Dragão, a Linha A (Azul) está aberta ao público em toda a sua extensão.
- Março de 2005 – com a abertura do troço Senhora da Hora – Pedras Rubras, é criada a linha B (Vermelha), o Metro do Porto passa então a funcionar em rede, com 23 km de linhas em operação.
- Julho de 2005 – abertura da linha C (Verde), entre o estádio do Dragão e o Fórum da Maia. A rede do metro passa a contar com 3 linhas e uma extensão global superior a 29 km.
- Setembro de 2005 – abertura da linha D (Amarela), numa primeira fase entre as estações Pólo Universitário e a Câmara de Guia.
- Março de 2006: são abertas ao público diversos novos serviços
 - o Extensão da Linha B até à Póvoa de Varzim, é igualmente introduzido o serviço Expresso, que ao contrário do corrente, só efectua paragens em algumas estações.
 - o Abertura da Linha C até à estação do ISMAI.
 - o A Linha D é estendida nos seus dois extremos, circulando agora entre as estações João de Deus e Hospital de São João, embora nem todas as circulações cheguem ao Hospital de São João.
- Maio de 2006 – Abertura da Linha E até ao Aeroporto, fica assim concluída a primeira fase do Metro do Porto.

Com todas estas alterações, a rede do metro do Porto ficou assim estabelecida:

2 Descrição do Problema

2.1 Situação Prévia

Em Dezembro de 2002, foi aberto ao público o sistema de metro ligeiro do Porto. Inicialmente apenas estava em funcionamento parte da Linha A, entre as estações Trindade e Senhor de Matosinhos num percurso de 11,7 quilómetros. Nessa altura encontravam-se em circulação poucos veículos, a geração de horários era feita manualmente e sem grandes problemas, em termos de número de veículos necessários.

No entanto a obra foi crescendo, e foram sucessivamente abertas ao público novas linhas:

- Junho de 2004 – em vésperas do euro 2004 é inaugurado o troço Trindade – Estádio do Dragão, a Linha A (Azul) está aberta ao público em toda a sua extensão.
- Março de 2005 – com a abertura do troço Senhora da Hora – Pedras Rubras, é criada a linha B (Vermelha), o Metro do Porto passa então a funcionar em rede, com 23 km de linhas em operação.
- Julho de 2005 – abertura da linha C (Verde), entre o estádio do Dragão e o Fórum da Maia. A rede do metro passa a contar com 3 linhas e uma extensão global superior a 29 km.
- Setembro de 2005 – abertura da linha D (Amarela), numa primeira fase entre as estações Pólo Universitário e a Câmara de Gaia.
- Março de 2006: são abertos ao público diversos novos serviços
 - Extensão da Linha B até à Póvoa de Varzim, é igualmente introduzido o serviço Expresso, que ao contrário do corrente, só efectua paragens em algumas estações;
 - Abertura da Linha C até à estação do ISMAI;
 - A Linha D é estendida nos seus dois extremos, circulando agora entre as estações João de Deus e Hospital de São João, embora nem todas as circulações cheguem ao Hospital de São João.
- Maio de 2006 – Abertura da Linha E até ao Aeroporto, fica assim concluída a primeira fase do Metro do Porto.

Com todas estas alterações, a rede do metro do Porto ficou assim estabelecida:

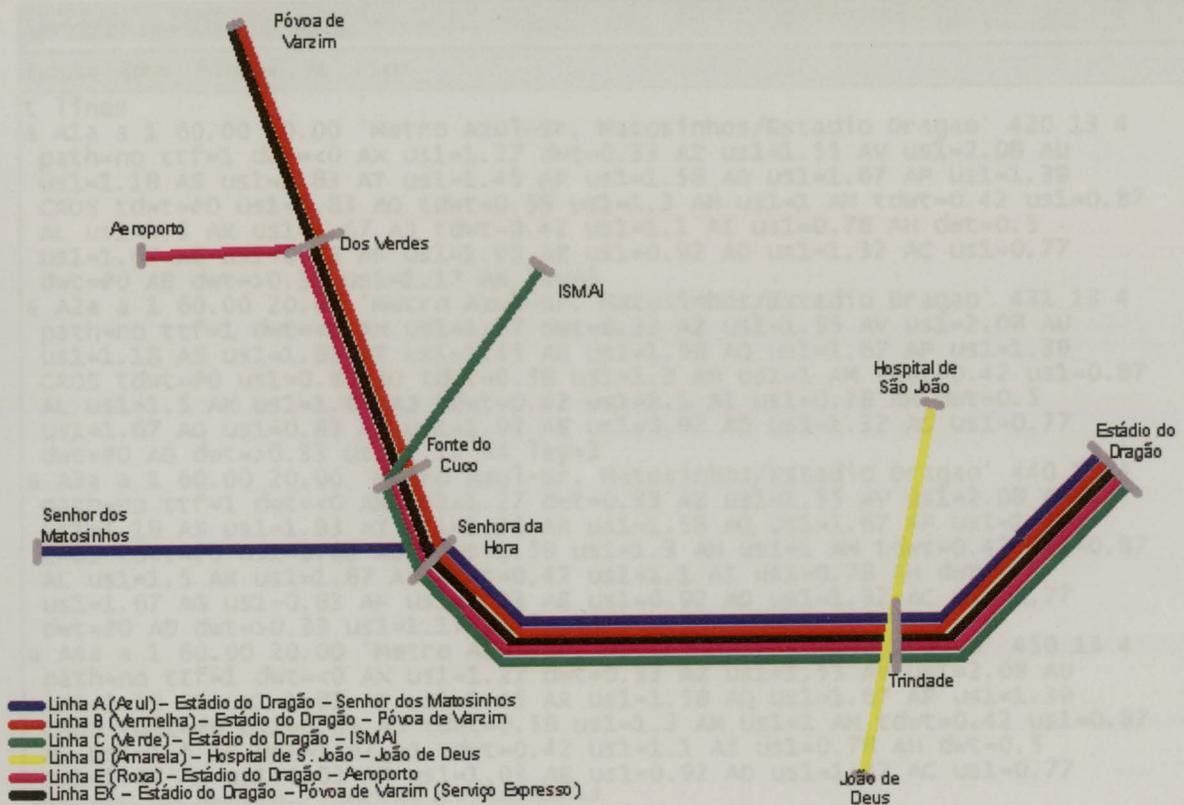


Figura 2 – Rede actual (simplificada) do Metro do Porto

Até hoje o processo de geração e criação de horários consistia nos seguintes passos:

1. Geração manual do horário;
2. Carregamento manual do horário para o formato de ficheiro *EMME/2* de acordo com a Figura 3.
3. Carregamento do ficheiro para *EMME/2*, e pedir o desenho do gráfico.
4. Analisar o gráfico, semelhante ao da Figura 4, com especial ênfase para os potenciais conflitos.

Sempre que era necessário realizar algum ajuste ao horário, era necessário repetir os pontos 2, 3 e 4, o que se revelava muitas vezes ineficiente. Pretende-se automatizar pelo menos o 1º passo deste processo, o que permitirá ao utilizador partir desde logo de uma solução mais adequada, o que deverá reduzir o número de iterações nos outros passos.

Pretende-se numa fase posterior automatizar igualmente o 2º e 3º passos.

```

d221.in - Bloco de notas
Ficheiro Editar Formatar Ver Ajuda

t lines
a A1a a 1 60.00 20.00 'Metro Azul-Sr. Matosinhos/Estadio Dragao' 420 13 4
path=no ttf=1 dwt=<0 AX us1=1.27 dwt=0.33 AZ us1=1.55 AV us1=2.08 AU
us1=1.18 AS us1=1.83 AT us1=1.45 AR us1=1.58 AQ us1=1.67 AP us1=1.39
CROS tdwt=#0 us1=0.83 AO tdwt=0.58 us1=1.3 AN us1=1 AM tdwt=0.42 us1=0.87
AL us1=1.5 AK us1=1.67 AJ tdwt=0.42 us1=1.1 AI us1=0.78 AH dwt=0.5
us1=1.67 AG us1=0.83 AF us1=1.03 AE us1=0.92 AD us1=1.32 AC us1=0.77
dwt=#0 AB dwt=>0.33 us1=1.17 AA lay=3
a A2a a 1 60.00 20.00 'Metro Azul-Sr. Matosinhos/Estadio Dragao' 431 13 4
path=no ttf=1 dwt=<0 AX us1=1.27 dwt=0.33 AZ us1=1.55 AV us1=2.08 AU
us1=1.18 AS us1=1.83 AT us1=1.45 AR us1=1.58 AQ us1=1.67 AP us1=1.39
CROS tdwt=#0 us1=0.83 AO tdwt=0.58 us1=1.3 AN us1=1 AM tdwt=0.42 us1=0.87
AL us1=1.5 AK us1=1.67 AJ tdwt=0.42 us1=1.1 AI us1=0.78 AH dwt=0.5
us1=1.67 AG us1=0.83 AF us1=1.03 AE us1=0.92 AD us1=1.32 AC us1=0.77
dwt=#0 AB dwt=>0.33 us1=1.17 AA lay=3
a A3a a 1 60.00 20.00 'Metro Azul-Sr. Matosinhos/Estadio Dragao' 440 13 4
path=no ttf=1 dwt=<0 AX us1=1.27 dwt=0.33 AZ us1=1.55 AV us1=2.08 AU
us1=1.18 AS us1=1.83 AT us1=1.45 AR us1=1.58 AQ us1=1.67 AP us1=1.39
CROS tdwt=#0 us1=0.83 AO tdwt=0.58 us1=1.3 AN us1=1 AM tdwt=0.42 us1=0.87
AL us1=1.5 AK us1=1.67 AJ tdwt=0.42 us1=1.1 AI us1=0.78 AH dwt=0.5
us1=1.67 AG us1=0.83 AF us1=1.03 AE us1=0.92 AD us1=1.32 AC us1=0.77
dwt=#0 AB dwt=>0.33 us1=1.17 AA lay=3|
a A4a a 1 60.00 20.00 'Metro Azul-Sr. Matosinhos/Estadio Dragao' 450 13 4
path=no ttf=1 dwt=<0 AX us1=1.27 dwt=0.33 AZ us1=1.55 AV us1=2.08 AU
us1=1.18 AS us1=1.83 AT us1=1.45 AR us1=1.58 AQ us1=1.67 AP us1=1.39
CROS tdwt=#0 us1=0.83 AO tdwt=0.58 us1=1.3 AN us1=1 AM tdwt=0.42 us1=0.87
AL us1=1.5 AK us1=1.67 AJ tdwt=0.42 us1=1.1 AI us1=0.78 AH dwt=0.5
us1=1.67 AG us1=0.83 AF us1=1.03 AE us1=0.92 AD us1=1.32 AC us1=0.77
dwt=#0 AB dwt=>0.33 us1=1.17 AA lay=3

```

Figura 3 - Ficheiro de carregamento de Linhas em EMME/2

Como se pode ver pela Figura 3, o carregamento das linhas, é algo difícil. Neste ficheiro (elaborado manualmente), os dados relativos aos tempos de paragens, tempos de viagem, frequências, horas de partida e todos os outros atributos de uma viagem, estão todos juntos, tornando a sua interpretação directa difícil.

Na Figura 4, pode-se ver o gráfico que resulta da introdução dos dados no software EMME/2. É assim possível efectuar a análise do horário produzido.

Como se pode ver pela Figura 2, a rede actual do Metro do Porto é constituída por 5 linhas e 6 serviços diferentes. Na Linha D, apesar de circular em dois serviços sobre a mesma linha D (João de Deus - Hospital de S. João e João de Deus - Polo Universitário), e de haver como tal alguma dificuldade em termos de modelação, este caso não será analisado.

O problema surge no troço Estadio do Dragão - Senhora da Hora, como se pode ver, circulam neste troço quatro linhas, o que dificulta em muito a definição do horário para estas linhas. Este facto deve-se a diversos factores, que se enunciam ao seguida:

Primeiro, é o facto de as linhas terem destinos diferentes, o que implica necessidades diferentes em termos de mobilidade, o que se traduz em frequências diferentes, que por sua vez implica intervalos de passagem diferentes que devem ser mantidos constantes tanto quanto possível, estes devem ser mantidos constantes tanto no troço comum (Senhora da Hora - Estadio do Dragão), como em cada uma das linhas.

¹ Baseado-se por intervalos de passagem, o tempo entre uma circulação e outra da mesma linha, serviço e sentido.

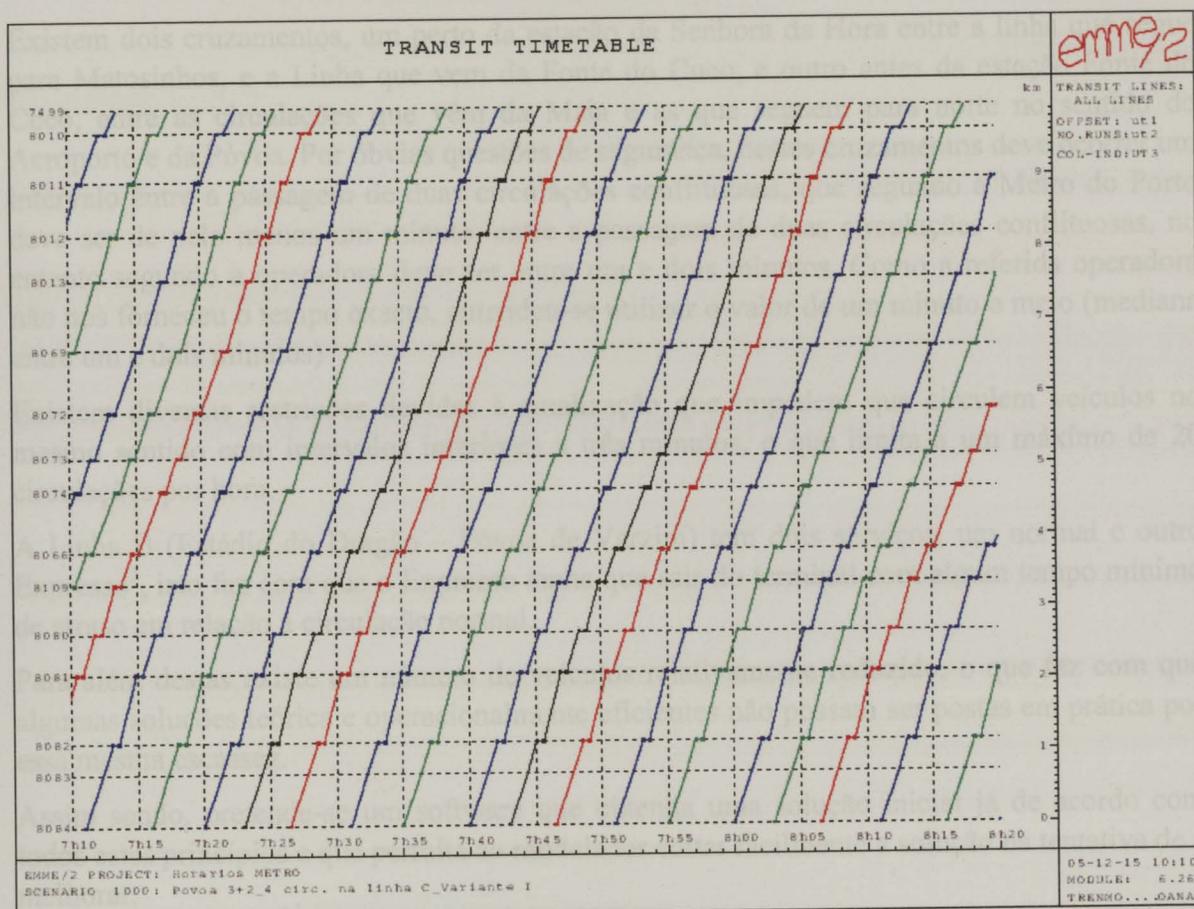


Figura 4 – Gráfico desenhado pelo Software *EMME/2*

No eixo vertical do referido gráfico encontram-se as paragens por onde a linha passa. No eixo horizontal encontram-se as horas do dia. Se a linha passa no ponto (x, y) do gráfico, significa que no momento x, a circulação se encontrar no local y.

2.2 Descrição Geral

Como se pode ver pela Figura 2, a rede actual do Metro do Porto é constituída por 5 linhas e 6 serviços diferentes. Na Linha D, apesar de circular em dois serviços sobre a mesma linha D (João de Deus – Hospital de S. João e João de Deus – Pólo Universitário), e de haver como tal alguma dificuldade em termos de modelação, este caso não será analisado.

O problema surge no troço Estádio do Dragão – Senhora da Hora, como se pode ver, circulam neste troço quatro linhas, o que dificulta em muito a definição do horário para estas linhas. Este facto deve-se a diversos factores, que se enumeram em seguida.

Primeiro, é o facto de as linhas terem destinos diferentes, o que implica necessidades diferentes em termos de mobilidade, o que se traduz em frequências diferentes, que por sua vez implica intervalos de passagem¹ diferentes que devem ser mantidos constantes tanto quanto possível, estes devem ser mantidos constantes tanto no tronco comum (Senhora da Hora – Estádio do Dragão), como em cada uma das linhas.

¹ Entende-se por intervalo de passagem, o tempo entre uma circulação e outra da mesma linha,, serviço e sentido.

Existem dois cruzamentos, um perto da estação da Senhora da Hora entre a linha que segue para Matosinhos, e a Linha que vem da Fonte do Cuco, e outro antes da estação Fonte do Cuco, entre as circulações que vêm da Maia e as que seguem para norte no sentido do Aeroporto e da Póvoa. Por óbvias questões de segurança, nestes cruzamentos deve ocorrer um intervalo entre a passagem de duas circulações conflituosas, que segundo a Metro do Porto deve ser de pelo menos um minuto, entre a passagem de duas circulações conflituosas, no entanto segundo a operadora deve ser entre um e dois minutos. Como a referida operadora não nos forneceu o tempo exacto, entendeu-se utilizar o valor de um minuto e meio (mediana entre um e dois minutos)

Existem diversas restrições devidas à sinalização que impedem que circulem veículos no mesmo sentido com intervalos inferiores a três minutos, o que limita a um máximo de 20 circulações por hora.

A Linha B (Estádio do Dragão – Póvoa de Varzim) tem dois serviços, um normal e outro Expresso², isto faz com que o Expresso tenha que sair do terminal com algum tempo mínimo de atraso em relação à circulação normal.

Para além destas existe um número de veículos relativamente reduzido, o que faz com que algumas soluções teórica e operacionalmente eficientes não possam ser postas em prática por essa mesma escassez.

Assim sendo, pretende-se um software que obtenha uma solução inicial já de acordo com todos estes princípios e que permita ao modelador editar facilmente a solução na tentativa de a melhorar.

² Por serviço Expresso, entende-se uma circulação que efectua paragens apenas em algumas estações com maior procura, o que permite uma viagem mais rápida.

3 Estado da Arte

Inicialmente, para resolver este problema, a TRENMO Engenharia, Lda. ponderou a compra de um software. Foram analisadas algumas soluções que apresentaremos em seguida, assim como as suas vantagens e desvantagens:

Open Track – Simulation of Railway Networks

O Open Track, teve início em meados dos anos 90 como um projecto de investigação dos caminhos-de-ferro suíços. O objectivo era construir uma ferramenta de modelação de caminhos-de-ferro e ao mesmo tempo uma ferramenta de simulação.

Entre as suas funcionalidades encontram-se:

- Determinar as necessidades para uma infra-estrutura ferroviária.
- Analisar a capacidade de infra-estruturas e estações.
- Permitir ao utilizador um forma interactiva de construção de horários, análise da solidez dos mesmos com simulações simples ou simulações “Monte-Carlo”³.
- Análise de diversos sistemas de sinalização.
- Análise do efeito de falhas no sistema e atrasos.
- Calcular as potências necessárias para cada cenário
- Simulação de sistemas de levitação magnética.

Para além destas funcionalidades, o Open Track inclui ainda uma ferramenta de apoio chamada Open Time Table que recebe como entrada os horários e desenha o seu gráfico, permitindo depois a sua gestão e a sua análise.

Em resumo, trata-se de um software que recebe como entradas informação sobre o material circulante, a infra-estrutura ferroviária, e o horário, devolvendo depois os gráficos, a malha de horário, a ocupação de vias e outras estatísticas.

Em conclusão, trata-se de um software que aborda o problema numa perspectiva de simulação e de análise, em que é o utilizador a definir a solução inicial. O que é um pouco diferente do que se pretende, que é o um software em que seja o sistema a dar o primeiro passo. Assim sendo, este software não resolve o problema principal da instituição de estágio, que é a geração de horários.

EMME/2

O *EMME/2* é um sistema de planeamento interactivo para transporte urbano desenvolvido pela INRO. É igualmente um software de apoio à decisão que fornece um manuseamento eficiente e uniforme de informação. Permite igualmente a análise de diversos cenários simultaneamente.

³ A simulação monte – carlo, que gera valores aleatoriamente para as variáveis, de forma repetida, criando assim um modelo.

Entre as suas funcionalidades destacam-se:

- Representação da rede multimodal.
- Modelação de procura.
- Afectação automática.
- Afectação de tráfego.
- Modelos combinados.
- Pós-processamento dos resultados obtidos.
- Análise por subsecções.
- Intercâmbio de dados com outros *packages* de software.

Este é um software já existente na TRENMO, que devido a este facto, tem sido usado como ferramenta para o desenho de horários, no entanto como se viu na Figura 3, o carregamento dos dados para este formato é algo complexo.

Trata-se de um software de modelação de transportes, e como tal não está especificamente preparado para resolver apenas este problema em concreto. Isto poderia, fazer com que se estivesse a investir num software que não teria as suas capacidades e potencialidades exploradas ao máximo. No entanto, como foi acima referido, a empresa já usa este software, pelo que não envolveria nenhum investimento.

Ao contrário do anterior, já oferece algum cálculo automático e pesquisa de soluções, infelizmente essa pesquisa limita-se a algumas situações como a afectação e contagem de veículos.

Assim sendo, não resolve igualmente o principal que é a criação de automática do horário.

Análise Comparativa

	Open Track	EMME/2
Criação Automática de Horários	Não	Não
Desenho de Gráficos	Sim	Sim
Contagem de Veículos	Sim	Sim
Usabilidade	Boa	Insuficiente

Tabela 1 – Análise Comparativa entre os produtos existentes no mercado

4 Requisitos

Como foi referido no capítulo anterior, a opção recaiu na criação de um software de raiz, que pudesse resolver o problema subjacente à geração de horários. Com este capítulo pretende-se descrever os requisitos deste mesmo software, assim como os requisitos e restrições subjacentes à criação de um horário para este caso específico.

4.1 Requisitos Funcionais

O sistema a desenvolver dividir-se-á em dois grandes módulos.

1. Interface gráfica.
2. Motor de Pesquisa

A interface gráfica é o módulo que está a um nível mais alto, e que interage directamente com o utilizador.

Aqui o utilizador tem a possibilidade de introduzir os dados relativos à rede de Metro a analisar. Esses dados consistem em:

Nós

Um nó, é o elemento mais simples de uma rede de transportes. Por definição é um ponto ao qual se ligam outros elementos. Para efeitos de modelação considera-se apenas os pontos de origem e destino de cada linha, assim como os cruzamentos.

Para efeitos de desenho do horário, é necessário considerar todas as paragens e cruzamentos da rede.

Arco

Um arco é um vector que liga dois nós, um de origem e um de destino. Para efeitos de modelação consiste em cada troço do percurso. A cada um destes percursos deve ser atribuído um tempo de percurso.

Linha

É um serviço regular de transporte público de passageiros (com frequência e tempos de viagem definidos), num itinerário fixo.

Geometricamente, o itinerário é na realidade uma polilinha, ou uma sequência de arcos. Adicionalmente uma linha contém também a lista com os tempos de paragem em cada uma das estações.

Devido ao facto de servirem locais diferentes e como tal terem padrões de procura diferentes, deve ser possível definir a prioridade dada a cada linha na geração do horário. Este facto ganha especial relevância devido ao facto de não ser possível uma atribuição perfeita, na qual todas as linhas tenham uma frequência constante. A prioridade resume-se assim à importância

que será dada no momento de avaliar a solução em relação à descontinuidade das frequências nessa linha.

Como é óbvio, é igualmente necessário indicar a frequência de cada linha, isto é, o número de circulações dessa mesma linha que passam por hora.

Restrições de Circulação

Para além das frequências, existem ainda um sem fim de restrições para a construção de um horário, estas restrições normalmente são originadas por restrições físicas da infra-estrutura como:

- Cruzamentos.
- Potências Eléctricas.
- Velocidades diferentes das composições.

Pré – Atribuições Directas

O utilizador, pode pretender que o sistema de pesquisa parta de uma solução já com algumas linhas atribuídas. Assim deverá ser possível que sejam atribuídas algumas linhas manualmente antes do sistema iniciar. Pode ser necessário por exemplo que uma dada circulação por questões de procura circule sempre a uma mesma hora, independentemente do resto do horário.

Pré – Atribuições por Referência

Por restrições diversas, e para poupar o tempo de processamento e de pesquisa pode ser necessário atribuir algumas viagens de regresso, no momento em que são atribuídas as viagens no sentido inicial.

Por exemplo: independentemente do momento em que for atribuída qualquer viagem da linha E, a sua viagem de regresso terá sempre que ser exactamente 3 minutos depois. Isto permite reduzir o número de veículos, usados nesta linha, sem impor circulações à partida.

Sincronização entre linhas

Por vezes pretende-se que duas ou mais linhas, tenham um comportamento sincronizado, de forma a criar uma ligação virtual entre os seus terminos, fazendo com que imediatamente após a chegada de uma das circulações ao local de cruzamento, esteja prestes a passar a outra circulação de continuação até ao outro termino.

Partindo de todos estes pressupostos o sistema deverá estar em execução durante um tempo determinado pelo utilizador até encontrar uma solução que cumpra todas as restrições definidas pelo utilizador. Após encontrar uma solução, o sistema deverá avaliá-la no sentido de minimizar os desvios em relação à frequência ideal.

Sempre que for encontrada uma nova solução, esta deve ser também avaliada, e deverá substituir a solução actual caso seja melhor que a existente anteriormente.

Deverá igualmente ser possível, apagando algumas das viagens atribuídas, recolocar o motor de pesquisa novamente em funcionamento no sentido de obter uma nova solução, considerando como definitivas as viagens já atribuídas anteriormente que tenham sido mantidas.

Restrições de qualificação

4.1.1 Regras para a construção de um horário ferroviário

O módulo de geração de horários, deve ter em conta vários aspectos. Nesta secção serão descritas as regras para a construção do horário para o caso específico do Metro do Porto.

Como se viu inicialmente, a rede de metro do Porto consiste em cinco linhas e seis serviços. A maior dificuldade na geração do horário prende-se com o tronco comum, entre a Senhora da Hora e o Estádio do Dragão. Neste tronco circulam simultaneamente 4 linhas e 5 serviços que servem padrões de procura diferentes, e que como tal têm frequências diferentes. Pretende-se igualmente que entre todas as linhas passem no tronco comum 18 circulações por hora.

No entanto é necessário encaixar todas essas circulações respeitando as restrições que serão explicadas em seguida.

Frequência de cada linha/serviço

É necessário analisar os cinco serviços seguintes:

- Linha A – Senhor de Matosinhos – Estádio do Dragão – é a linha mais antiga da rede de metro do Porto (aberta em 2002), e é igualmente a mais importante, pois é a que transporta mais gente diariamente. É também a que tem a frequência de passagem mais elevada das linhas analisadas, 6 por hora (em media de 10 em 10 minutos), os intervalos de passagem nesta linha não podem variar mais de 10%, que é 1 minuto apenas, podendo portanto ir dos 9 aos 11 minutos.
- Linha B – Serviço Normal – Póvoa de Varzim – Estádio do Dragão – É a linha que serve um maior número de estações, no entanto, devido ao facto de as zonas servidas não serem tão populosas como as das outras linhas, a sua periodicidade não tem a mesma relevância que a da Linha A. A sua frequência é de 3 composições por hora. Segundo alguns cenários pode ter 4 circulações por hora.
- Linha B – Serviço Expresso – Esta linha pretende servir maioritariamente as populações da Póvoa de Varzim e de Vila do Conde, efectua o mesmo percurso da linha B, com a excepção de que a partir da estação da Senhora da Hora apenas efectua paragens em Pedras Rubras, Verdes, Vila do Conde e Póvoa de Varzim. A sua frequência é de apenas uma circulação por hora, portanto como os horários se mantêm de hora para hora, o intervalo será sempre constante. Segundo alguns cenários pode ter 2 circulações por hora, portanto o intervalo de passagem pode ter que ser considerado.

- Linha C – ISMAI – Estádio do Dragão – É a linha que serve maioritariamente o concelho da Maia, e futuramente o da Trofa. Tem uma frequência de 3 circulações por hora, segundo alguns cenários de análise, pode ter até 5 circulações por hora.
- Linha E – Aeroporto – Estádio do Dragão – Foi a última linha a ser inaugurada e tem uma frequência de 3 circulações por hora. Por ser a linha que serve o aeroporto não tem uma importância muito grande em termos de intervalos de circulação constantes. No entanto, como se poderá ver mais tarde, por restrições relacionadas com o número de veículos essa frequência terá que ser mais rígida.

Restrições de sinalização

Por questões de segurança é imperativo que as composições circulem com um espaçamento entre elas. Após a passagem de uma composição num dado ponto, o sinal fica vermelho durante três minutos, e apenas após esse tempo pode passar outra circulação no mesmo sentido.

Esta é uma restrição rígida, que impede automaticamente que a frequência seja superior a 20 circulações / hora.

Intervalos de circulação

Os intervalos de circulação devem ser mantidos o mais constante possível, embora seja permitido um certo desvio, que varia de linha para linha, na linha A é de cerca de 10%, nas outras pode atingir valores na ordem dos 30%.

É igualmente necessário manter um intervalo de passagem até 6 minutos no tronco comum.

A Função Objectivo consiste em minimizar os desvios nos intervalos em relação aos da solução ideal.

Cruzamentos

Numa situação ideal, todos os locais onde as linhas tenham trajectórias conflituosas, deviam ser substituídas por cruzamentos desnivelados, no entanto, provavelmente por questões de engenharia, não foi possível nalguns casos a implementação desta solução. Neste casos, esse cruzamento é feito directamente pela trajectória da outra linha o que implica que não possam passar duas circulações em sentidos contrários simultaneamente por estes locais. É igualmente necessário dar uma certa margem entre a passagem das duas circulações conflituosas, este tempo, pode variar consoante o raio de curvatura do cruzamento.

Na rede do Metro do Porto existem dois cruzamentos potencialmente conflituosos, um pouco depois da estação Senhora da Hora e outro na estação da Fonte do Cuco. Em ambos os casos o tempo determinado como margem de segurança entre as duas circulações conflituosas foi estabelecido em 90 segundos.

Diferentes Velocidades

Na rede de Metro do Porto, de momento, todos os serviços circulam a uma única velocidade, e como tal qualquer que seja o ponto da rede, os tempos de viagem são iguais. No entanto, no tempo Senhora da Hora da linha B, os serviços Expresso e Normal têm velocidades diferentes. Isto faz com que, apesar da velocidade de ponta ser semelhante, o tempo de viagem comercial seja maior no Serviço Expresso do que no serviço normal, assim, é necessário que no momento de atribuir as saídas dos terminais, o Serviço Expresso seja atribuído um intervalo em relação às circulações mais lentas. Esta situação é ilustrada na Figura 5.

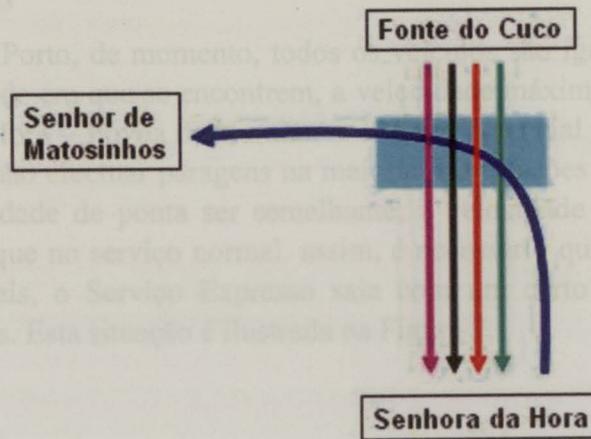


Figura 5 – Cruzamento da Senhora da Hora

Como se pode ver pela Figura 5, o veículo que segue em direcção a Matosinhos, cruza o percurso dos que vêm da estação da Fonte do Cuco.

No cruzamento da Fonte do Cuco, a situação é a seguinte:

Figura 7 – Diferenças de velocidades entre o serviço Expresso e serviço Normal

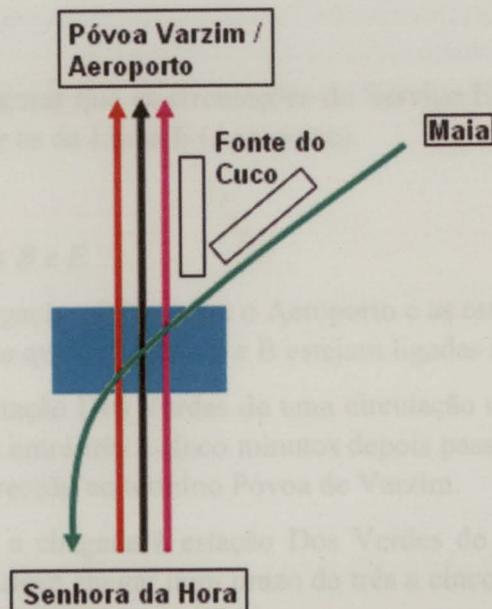


Figura 6 – Cruzamento da Fonte do Cuco

Como se pode ver pela Figura 6, as circulações provenientes da Maia e que circulam em direcção à Senhora da Hora, entram em conflito com as circulações provenientes da Senhora da Hora e que circulam em direcção à Póvoa de Varzim e Aeroporto.

Diferentes Velocidades

Na rede de Metro do Porto, de momento, todos os veículos são iguais, e como tal qualquer que seja o ponto da rede em que se encontrem, a velocidade máxima é a mesma. No entanto, no troço Senhora da Hora – Póvoa de Varzim, existe um potencial conflito criado pelo facto do Serviço Expresso não efectuar paragens na maioria das estações desse troço. Isto faz com que, apesar da velocidade de ponta ser semelhante, a velocidade comercial seja maior no Serviço Expresso do que no serviço normal. Assim, é necessário que no momento de atribuir as saídas dos terminais, o Serviço Expresso saia com um certo intervalo em relação às circulações mais lentas. Esta situação é ilustrada na Figura 7.

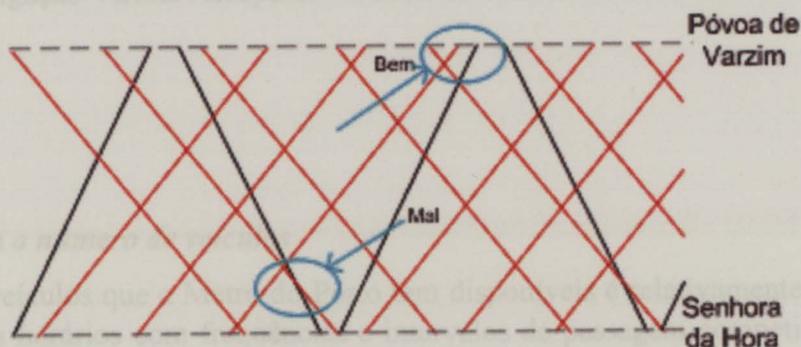


Figura 7 – Diferenças de velocidade entre a Linha B serviço Expresso e serviço Normal

Só assim será possível assegurar que as circulações do Serviço Expresso não “atropelem” as da linha B, serviço Normal e as da Linha E (Aeroporto).

Sincronização entre Linhas B e E

Com vista a fornecer uma ligação rápida entre o Aeroporto e as estações a norte em direcção à Póvoa de Varzim, procura-se que as Linhas E e B estejam ligadas na estação Dos Verdes.

Assim, após a chegada à estação Dos Verdes de uma circulação da linha E vinda do término Aeroporto, é necessário que entre três a cinco minutos depois passe nessa mesma estação uma circulação da linha B em direcção ao término Póvoa de Varzim.

No sentido contrário, após a chegada à estação Dos Verdes de uma circulação da linha B vinda da Póvoa de Varzim, deve chegar num prazo de três a cinco minutos uma circulação da linha E, em direcção ao término Aeroporto.

Este espaço mínimo de três minutos é necessário para permitir aos passageiros realizarem tranquilamente o transbordo entre as duas linhas.

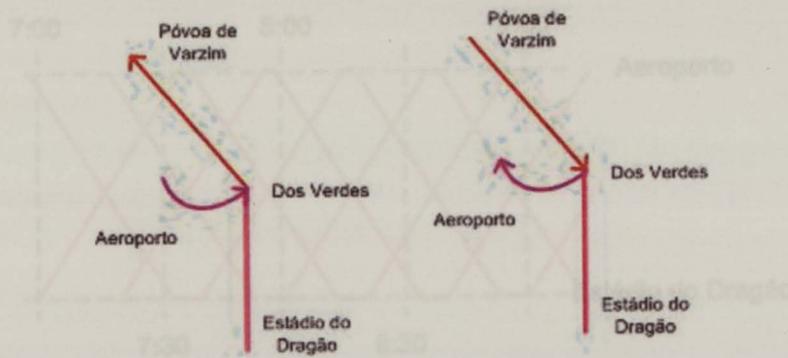


Figura 10 – Linha E (Aeroporto) sincronizada

Figura 8 – Ligação Virtual Aeroporto – Póvoa de Varzim e Póvoa de Varzim – Aeroporto

Restrições com o número de veículos

O número de veículos que a Metro do Porto tem disponíveis é relativamente reduzido, o que faz com alguns horários com frequências e intervalos de passagem competitivos, não sejam possíveis de colocar em prática devido a esta restrição. Para melhorar o desempenho, em algumas linhas, opta-se por atribuir os veículos manualmente. No caso do expresso, a viagem dura aproximadamente uma hora. Assim, caso os expressos saiam dos terminos à mesma hora, é possível que cheguem ao outro termino a tempo de saírem para a viagem de regresso da hora seguinte. Isto permite uma grande poupança em termos de veículos, como pode ser visto na Figura 9.

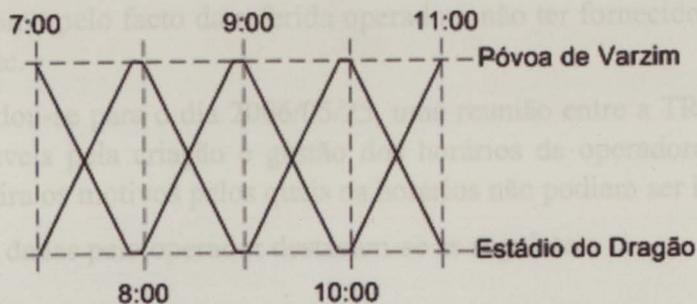


Figura 9 – Sincronização de viagens do Expresso

Na linha E, a viagem é de pouco menos de 40 minutos, isto permite que, saindo as circulações à mesma hora dos terminos, cheguem ao seu destino mesmo a tempo de partir para a viagem seguinte, como se pode ver na Figura 10.

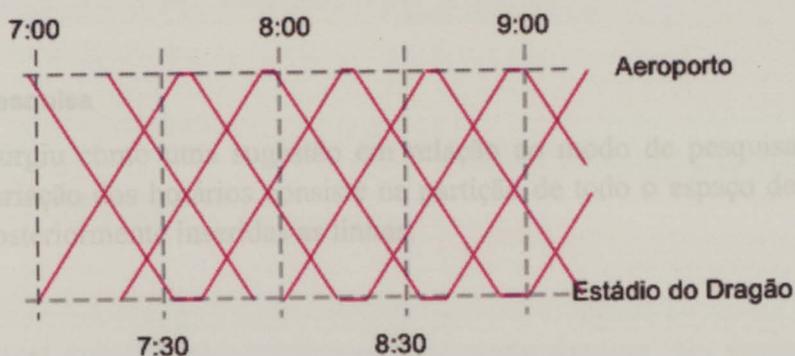


Figura 10 – Linha E (Aeroporto) sincronizada

Pretende-se portanto um *software* que resolva, por agora, este caso específico e potencialmente qualquer outro, com restrições semelhantes e dimensão também semelhante.

Dados retirados de reunião com a operadora em 2006/05/25

Em inícios de Maio, a operadora actual de toda a rede de Metro do Porto, deu indicação à TRENMO que os horários que estavam a ser produzidos, não podiam ser postos em prática, apesar de cumprirem a 100% os requisitos definidos pela empresa Metro do Porto. Este desacordo ter-se-á devido ao facto de a empresa Metro do Porto, por não operar directamente com a sua rede, não ter o conhecimento mais correcto sobre todas as restrições relativas às operações, assim como pelo facto da referida operadora não ter fornecido à TRENMO toda a informação existente.

Assim sendo, agendou-se para o dia 2006/05/25, uma reunião entre a TRENMO Engenharia, Lda. e os responsáveis pela criação e gestão dos horários da operadora, no sentido de ser esclarecido à primeira os motivos pelos quais os horários não podiam ser implementados.

Entre as indicações dadas pelo operador destacam-se as seguintes:

Tempo de passagem nos cruzamentos

Até esta reunião, o intervalo mínimo de passagem entre duas circulações conflituosas nos cruzamentos, estava definido como sendo um minuto, no entanto segundo o operador, este intervalo era manifestamente insuficiente, dado o reduzido raio de curvatura, nomeadamente do cruzamento da Senhora da Hora. Foi-nos indicado que o tempo necessário de intervalo entre as duas circulações conflituosas deveria ser entre um e dois minutos. Como não chegou a ser indicado qual o valor preciso definiu-se como tempo necessário 90 segundos (a mediana entre 1 e 2 minutos).

Esta situação está perfeitamente prevista dentro dos requisitos, sendo portanto apenas uma pequena alteração nos parâmetros de execução do software.

Imagem

Sugestões de pesquisa

Esta situação surgiu como uma sugestão em relação ao modo de pesquisar a solução. Na TRASDEV, a criação dos horários consiste na partição de todo o espaço de tempo em *slots* nas quais são posteriormente inseridas as linhas.

Velocidade alterada

Foi-nos igualmente transmitido que os tempos de viagem que estavam na posse da TRENMO Engenharia, Lda. não estavam correctos dado que a velocidade tinha sido alterada e iria ser alterada novamente nos próximos dias. Assim sendo, foi necessária uma medida directa dos tempos de passagem e paragem entre cada nó da rede. Este caso também não implicou nenhuma alteração nos requisitos iniciais, apenas consistiu numa alteração de parâmetros.

Potências Eléctricas

Existem igualmente algumas restrições em termos de potências eléctricas, que não estavam nos requisitos iniciais definidos pela Metro do Porto. No entanto, não tivemos acesso aos dados completos em relação a esta restrição, o que faz com que este requisito possa apenas ser desenvolvido em futuras versões.

Modularidade

As funcionalidades de edição de dados e apresentação de gráficos devem ser implementadas de forma independente.

4.2 Requisitos Tecnológicos

A melhor solução obtida deverá ser exportada para um dos seguintes formatos para sua melhor análise. Essa exportação deverá ser possível em qualquer um dos seguintes formatos: ficheiro, HTML, ou imagem.

Ficheiro

É o formato mais básico. A exportação para este formato, permite que mais tarde o utilizador carregue a solução para analisar e trabalhar sobre a mesma. Este ficheiro não tem que ter nenhum formato específico, tem apenas que ser reconhecido pelo próprio software a desenvolver.

HTML

Formato de páginas Web largamente divulgado e utilizado. A exportação para ficheiro de HTML tem a vantagem de permitir a apresentação dos dados mais relevantes numa única página. Tem igualmente como vantagem, devido ao facto de estar em texto, o facto de permitir a leitura a todos os utilizadores, e não apenas ao especialista / modelador.

Imagem

É apenas necessária a apresentação em imagem do horário, não é necessário exportá-la para um ficheiro à parte; por agora. Esta é uma funcionalidade torna possível analisar de uma forma global quais os pontos em que é possível melhorar a solução actual, e quais os pontos potencialmente conflituosos a corrigir.

Após a pesquisa da solução, o utilizador deverá ter a possibilidade de alterar a atribuição feita pelo sistema. Será igualmente possível realizar qualquer uma das exportações descritas anteriormente a partir desta nova solução editada.

4.3 Requisitos Não Funcionais

Para além dos requisitos funcionais definidos, anteriormente, é necessário, que o software a desenvolver, cumpra com uma série de requisitos não funcionais.

Fiabilidade

As soluções obtidas pelo motor de pesquisa, devem cumprir sempre e em qualquer situação os requisitos definidos à partida. Este facto assume especial relevo pela natureza sensível do problema abordado.

Modularidade

As funcionalidades de edição de dados e apresentação de gráficos, devem estar implementadas de forma independente.

Manutenção

Devido ao facto dos requisitos estarem em constante actualização, e de não haver aparentemente nenhum limite, ao número e tipo de restrições a implementar, todo o código deverá estar produzido numa linguagem acessível e o mais universal possível. Por linguagem acessível, pretende-se uma linguagem que seja conhecida pela grande maioria dos programadores e não obrigue a empresa a contratar alguém com um conhecimento demasiado específico para efectuar a manutenção. Todo o código deverá estar devidamente comentado.

Usabilidade

No sentido de haver um máximo aproveitamento das potencialidades do sistema, toda a interface de ligação com o utilizador deverá ser simples e intuitiva.

5 Análise do Problema

Neste capítulo pretende-se descrever a análise que foi feita antes do desenvolvimento da aplicação. Esta análise tem duas vertentes, a análise do problema em si, e a análise de algumas das tecnologias disponíveis para o desenvolvimento do software.

5.1 Análise prévia

Como se pode ver pelo capítulo anterior, a geração de horários é demasiado complexa dado que existe um grande número de restrições e se pretende ainda assim encontrar a solução. Para a correcta resolução deste problema é necessário algum poder computacional, no sentido de produzir uma solução aceitável.

No entanto, o número de soluções possíveis é demasiado elevado:

$$\frac{A_{18}^{120}}{6 \times 4! \times 4! \times 3! \times 1!} = 2.7962 \times 10^{30}$$

A Explicação desta fórmula

Isto torna impossível uma análise exaustiva, assim, nunca será possível afirmar que a solução encontrada corresponde à solução óptima.

Portanto, é necessário impedir que a pesquisa alastre em demasia para não se correr o risco do sistema derivar por demasiado tempo por hipóteses que não conduzam a nenhuma solução admissível.

Embora a abordagem possa ser semelhante, este não se trata de um problema de optimização: num problema de optimização, a pesquisa é feita dentro do conjunto de soluções admissíveis, e aí é procurada a melhor, no entanto, para que a maioria dos algoritmos tenha efeito, é necessário que a maioria das soluções/hipóteses encontradas sejam soluções admissíveis. Neste caso em concreto, a grande dificuldade prende-se com encontrar uma solução admissível, mas ainda assim, a forma de pesquisa pode ser semelhante. É necessário definir:

- Uma hipótese inicial;
- Uma estrutura de vizinhança;
- Uma função de avaliação.

5.2 Análise Tecnológica

Neste capítulo pretende-se fazer uma análise de todas as tecnologias usadas e analisadas no sentido de desenvolver a aplicação pretendida.

C# (.NET)

É uma linguagem de programação orientada a objectos, desenvolvida pela Microsoft e normalizada como parte da sua plataforma .NET.

A sua sintaxe deriva do C/C++ e utiliza o modelo de objectos da plataforma .NET que é similar ao do Java. C# foi concebido para combinar o controlo de baixo nível de linguagens como C, com a velocidade de programação de linguagens como o Visual Basic.

A principal plataforma de desenvolvimento é o Microsoft Visual Studio .NET. No entanto, esta plataforma tem um custo elevado.

Existem no entanto algumas plataformas open source como o Mono e o gnu.NET que no entanto não atingem o nível dado por plataformas como o Eclipse.

CLP(R) – Constrain Logic Programming (Real)

É uma linguagem de programação em lógica com restrições com restrições aritméticas reais, desenvolvida por técnicos da IBM e da Monash University em 1986.

Contém um *solver* automático de restrições que lida com equações aritméticas e que contém um mecanismo para transformar restrições não lineares em restrições lineares.

Existem já trabalhos desenvolvidos usando CLP (R) em diversas áreas, como a Biologia, a Finança e a Modelação de Processos Físicos.

Não foi no entanto utilizada no desenvolvimento porque era uma tecnologia que exigia um conhecimento aprofundado, e a aquisição desse conhecimento exigiria um esforço e custo que poderia pôr em causa o planeamento do projecto definido à partida.

Java (Sun)

Linguagem de programação orientada a objectos, desenvolvida por James Gosling e os seus colegas da Sun Microsystems no início dos anos 90. Ao contrário da maioria das outras linguagens, o resultado da sua compilação não é código máquina, é compilada num bytecode que é usado mais tarde numa Java Virtual Machine. Assim, qualquer projecto desenvolvido em Java é completamente portátil, uma vez que o mesmo código objecto pode ser executado em máquinas diferentes com sistemas operativos diferentes sem necessidade de recompilação.

Tem igualmente a vantagem de existirem muitas aplicações e compiladores open-source que permitem um desenvolvimento mais fácil da aplicação.

O BlueJ foi usado como plataforma para o desenvolvimento do motor de pesquisa, devido principalmente ao facto de facilmente se poder controlar se os algoritmos e as estruturas de dados implementadas estão a funcionar correctamente, através da sua funcionalidade “Object Inspector”.

Para a definição das interfaces, será usada a plataforma Eclipse, dado que é uma ferramenta mais poderosa que o BlueJ e que tem diversos mecanismos automatizados.

HTML

Sigla de Hyper Text Markup Language é uma Linguagem de Marcação (Markup Language) desenhada para estruturar textos e apresentá-los sob a forma de hipertexto, que é o formato standard das páginas web.

A utilidade desta tecnologia neste projecto, prende-se com a exportação dos dados, para um formato mais legível e apresentável para o cliente.

5.3 Planeamento

Análise do Problema	2 days	20-Fev	21-Fev
Definição de Requisitos	3 days	22-Fev	24-Fev
Análise Tecnológicas	5 days	27-Fev	03-Mar
Implementação do Motor de Pesquisa	85 days	06-Mar	30-Jun
Implementação do editor	45 days	01-Mai	30-Jun
Relatório	65 days	01-Abr	04-Ago

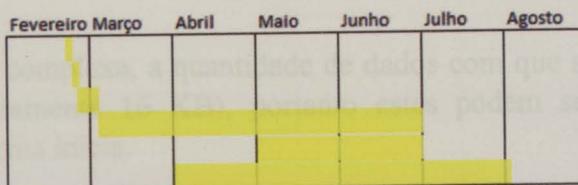


Figura 11 – Diagrama de Gantt relativo ao Projecto

O planeamento adoptado seguiu os seguintes passos, como pode ser visto na Figura 11.

Análise do Problema

Nesta fase foi feita a análise geral do problema, assim como a procura de soluções já existentes no mercado. O resultado desta análise está descrito nos capítulos 2 e 3.

Definição de Requisitos

Nesta fase foram definidos os requisitos gerais do software a desenvolver que podem ser vistos no capítulo 4 do presente relatório.

Análise Tecnológica

Foi feita a análise das tecnologias possíveis de utilizar na produção do protótipo, os prós e contras de cada, e as escolhas efectuadas. O resultado desta fase do projecto encontra-se descrito no capítulo 5.2 do presente documento.

Implementação do Motor de Pesquisa

Foi implementado o motor de pesquisa de uma solução, no capítulo 6 descrevem-se alguns detalhes do motor de pesquisa desenvolvido nesta fase.

Implementação do Editor

Nesta fase foi implementado o editor de dados, os detalhes relativos a esta fase encontram-se descritos no capítulo 6 do presente documento.

Relatório

Nesta última fase foi elaborado o presente relatório.

6 Implementação

Neste capítulo serão descritos em detalhe, os detalhes de implementação do software descrito anteriormente.

6.1 Arquitectura

Neste projecto a arquitectura está longe de ser complexa, a quantidade de dados com que se trabalha é relativamente reduzida (aproximadamente 16 KB), portanto estes podem ser carregados na sua totalidade logo quando o sistema inicia.

Isto permite um acesso mais rápido a todos os dados existentes sem sobrecarregar o sistema.

No entanto, no módulo da pesquisa da solução, optou-se por simplificar ainda mais o problema, considerando apenas os dados que acrescentassem valor, por exemplo, em vez de considerar todas as estações da rede, consideraram-se apenas as que acrescentam valor à rede, ou seja os cruzamentos e os terminais.

6.1.1 Modelo de Classes

O Modelo Relacional deste projecto consiste de 10 classes dispostas da forma que se pode ver na Figura 12.

Figura 12 – Modelo de Classes do software desenvolvido

Node (Nó)

Implementa um nó da rede ferroviária. Tipicamente um nó é apenas um ponto relevante da rede, podendo ser um nó, uma estação ou um terminal.

Devido ao facto de estarem a ser usados como entrada os ficheiros de configuração do EMME/2, os nós têm dois IDs, um corresponde ao ID do programa outro correspondente ao código do EMME/2. O ID que é usado internamente corresponde ao índice desse nó na lista de nós existentes na classe principal. O outro serve apenas de identificação na leitura das linhas. São guardados igualmente outros dados, como a zona e as coordenadas, que não contém informação relevante para a pesquisa do horário, nem para o seu desenho, mas que têm importância como ajuda ao utilizador.

Arc (Arco)

Esta classe implementa um arco da rede, segundo se pode ver no glossário, um arco é uma ligação entre dois nós, ou seja é um segmento da rede ferroviária.

Ao contrário do anterior, neste apenas é necessário um ID, dado que nos ficheiros de entrada do software EMME/2, não existe nenhuma chave primária para os arcos (são conhecidos por links), dado que a forma destes serem identificados univocamente é através da combinação de dois arcos distintos.

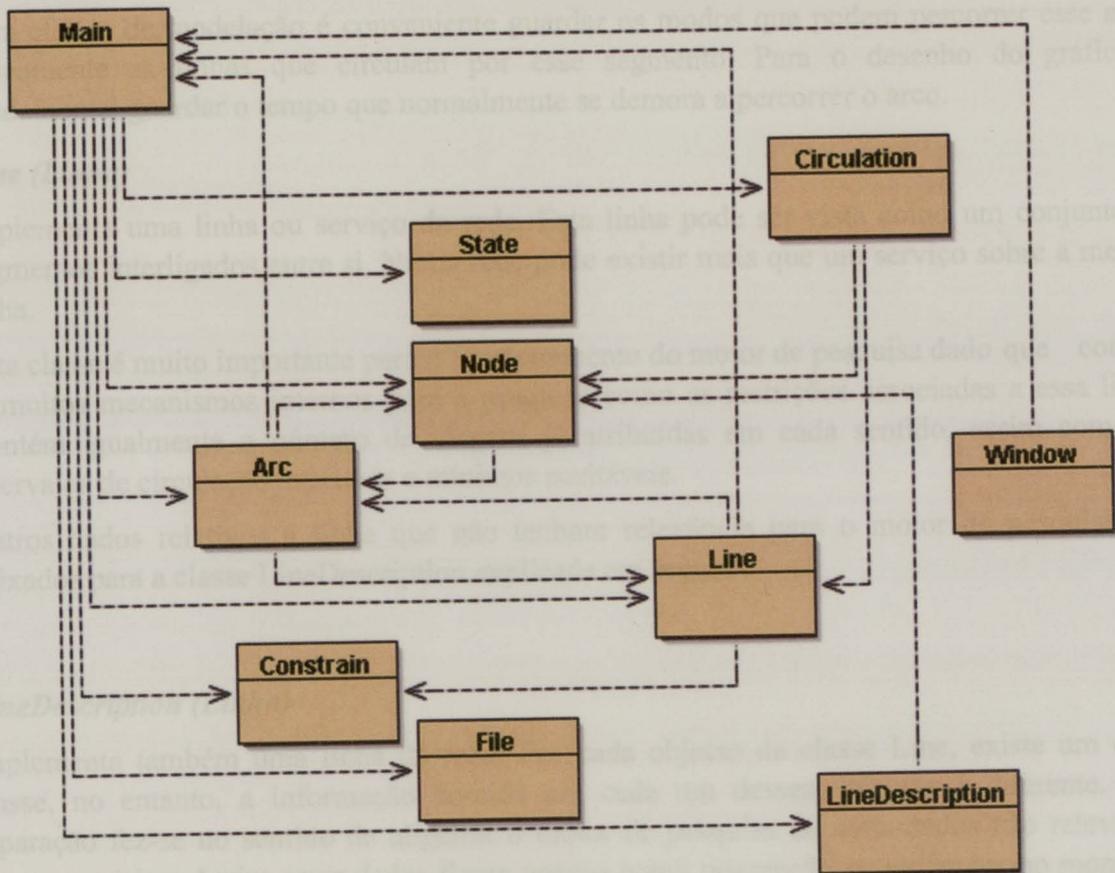


Figura 12 – Modelo de Classes do software desenvolvido

Node (Nó)

Implementa um nó da rede ferroviária. Tipicamente um nó é apenas um ponto relevante da rede, podendo ser um nó, uma estação ou um terminal.

Devido ao facto de estarem a ser usados como entrada os ficheiros de configuração do EMME/2, os nós têm dois IDs, um corresponde ao ID do programa outro correspondente ao código do EMME/2. O ID que é usado internamente corresponde ao índice desse nó na lista de nós existentes na classe principal. O outro serve apenas de identificação na leitura das linhas. São guardados igualmente outros dados, como a zona e as coordenadas, que não contêm informação relevante para a pesquisa do horário, nem para o seu desenho, mas que têm importância como ajuda ao utilizador.

Arc (Arco)

Esta classe implementa um arco da rede, segundo se pode ver no glossário, um arco é uma ligação entre dois nós, ou seja é um segmento da rede ferroviária.

Ao contrário do anterior, neste apenas é necessário um ID, dado que nos ficheiros de entrada do software EMME/2, não existe nenhuma chave primária para os arcos (são conhecidos por links), dado que a forma destes serem identificados univocamente é através da combinação de dois arcos distintos.

Para efeitos de modelação é conveniente guardar os modos que podem percorrer esse arco, tipicamente as linhas que circulam por esse segmento. Para o desenho do gráfico é fundamental guardar o tempo que normalmente se demora a percorrer o arco.

Line (Linha)

Implementa uma linha ou serviço da rede. Esta linha pode ser vista como um conjunto de segmentos interligados entre si. Numa rede pode existir mais que um serviço sobre a mesma linha.

Esta classe é muito importante para o funcionamento do motor de pesquisa dado que contém já muitos mecanismos internos para a pesquisa como as restrições associadas a essa linha. Contém igualmente o número de viagens já atribuídas em cada sentido, assim como os intervalos de circulação máximos e mínimos aceitáveis.

Outros dados relativos à linha que não tenham relevância para o motor de pesquisa são deixados para a classe *LineDescription* explicada em seguida.

LineDescription (Linha)

Implementa também uma linha da rede. Por cada objecto da classe *Line*, existe um desta classe, no entanto, a informação contida em cada um desses objectos é diferente. Esta separação fez-se no sentido de aligeirar o motor de pesquisa de meta-dados não relevantes para a pesquisa. Assim estes dados ficam apenas como orientação ao utilizador no momento em que este estiver a editar os dados.

Esta classe contém igualmente a descrição do caminho, e dos tempos de circulação e de paragem em todo o percurso da linha.

Circulation (Circulação)

Contém os dados relativos a uma circulação na rede. Nomeadamente os nós de partida e de chegada, a hora de partida e de chegada, assim como os veículos que são usados para essa circulação.

Implementa a classe *Comparable*, porque em alguns casos é necessário ordenar as circulações existentes, nomeadamente para o cálculo do número de veículos.

State (Estado)

Como o motor de pesquisa usa o algoritmo de *back-tracking*, é necessário guardar o caminho já percorrido. Assim o caminho é guardado em objectos desta classe. Estes objectos contêm todas as estruturas de dados relativas a esse estado. Contém igualmente um contador que indica a ordem em que esta solução foi escolhida no passo anterior.

Constrain (Restrição)

É a classe mais simples de todo o sistema, representa apenas uma restrição de circulação. Cada objecto desta classe contém uma linha (a linha que está restringida), o intervalo no qual

a circulação nessa linha está restringida (este intervalo é um valor relativo em relação ao momento em que outra linha foi atribuída) e um indicador relativo ao sentido em que a circulação está restringida.

Esta informação é carregada a partir de um ficheiro de configuração com o nome `constraint.ugt`, ficheiro este que se encontra no formato descrito na Figura 13.

```

2 1 4 5 21
4 1 2 -21 -5
1 0 2 2 5
1 0 3 2 5
1 0 4 2 5
    
```

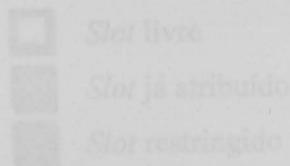
Figura 13 – Fragmento do Ficheiro `constraint.ugt`

Neste fragmento são referidas duas restrições fundamentais, a primeira (as duas primeiras linhas), refere-se à diferente velocidade entre a Linha B – Serviço Expresso e a Linha B – Serviço Normal. Na primeira delas, refere-se que quando for atribuída uma circulação da linha B (2), não pode ser atribuído nenhuma circulação no mesmo sentido (1), do Serviço Expresso (4), entre as slots 5 e 21 (dos 3 aos 11 minutos – *slots* de 30 segundos) a partir do momento da atribuição, sendo que a segunda linha, é apenas o inverso da primeira.

A segunda parte refere-se ao cruzamento da Senhora da Hora, assim sendo, sempre que for atribuída uma circulação da Linha A (1) no sentido directo, não pode ser atribuída nenhuma circulação no sentido contrário (0) das Linhas B, C e EX (2, 3 e 4) nas *slots* 2 a 5 a partir do momento da atribuição.

Figura 14 – Demonstração das restrições:

- 1 – Matriz de disponibilidades inicial;
- 2 – Matriz de disponibilidades no sentido directo após atribuição;
- 3 – Matriz de disponibilidades no sentido inverso após atribuição



Na Figura 14 mostra-se a matriz de disponibilidades com 120 slots de tempo (período de uma hora, precisão de 30 segundos), e com 4 linhas, como decorria no metro do Porto à data do infólio de projecto. Inicialmente a matriz tem todas as posições disponíveis. Seguidamente é introduzida uma circulação da linha A no momento 0, esta circulação, por ser a primeira, vai fechar grande parte dos slots para essa linha A, deixando em aberto apenas aqueles que respeitem os intervalos máximos e mínimos de circulação. De notar que nos 5 slots seguintes e anteriores ao momento 0, a circulação nas outras linhas encontra-se restringida.

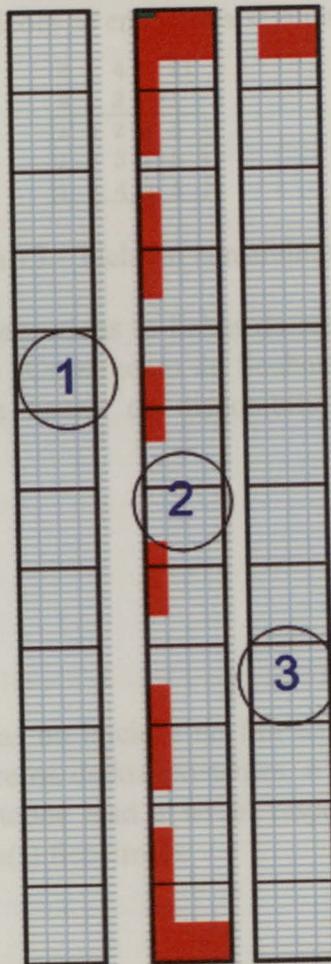


Figura 14 – Demonstração das restrições:

- 1 – Matriz de disponibilidades inicial;
- 2 – Matriz de disponibilidades no sentido directo após atribuição;
- 3 – Matriz de disponibilidades no sentido inverso após atribuição

-  Slot livre
-  Slot já atribuído
-  Slot restringido

Na Figura 14 mostra-se a matriz de disponibilidades com 120 slots de tempo (período de uma hora, precisão de 30 segundos), e com 4 linhas, como decorria no metro do Porto à data de início de projecto. Inicialmente a matriz tem todas as posições disponíveis. Seguidamente é introduzida uma circulação da linha A no momento 0, esta circulação, por ser a primeira, vai fechar grande parte das slots para essa linha A, deixando em aberto apenas aqueles que respeitem os intervalos máximos e mínimos de circulação. De notar que nas 5 slots seguintes e anteriores ao momento 0, a circulação nas outras linhas encontra-se restringida.

Como no ficheiro relativo às restrições se encontra a seguinte informação:

2	1	4	5	21
4	1	2	-21	-5
1	0	2	2	5
1	0	3	2	5
1	0	4	2	5

Figura 15 – Ficheiro constraint.ugt

Uma atribuição da linha A restringe todas as linhas do sentido inverso entre as slots 2 e 5 a partir do momento da atribuição (Cruzamento da Senhora da Hora). Como se pode ver na Figura 14, todos os pontos da matriz de disponibilidades no sentido inverso ficaram restringidos entre as slots 2 e 5.

File (Ficheiro)

É uma classe que é usada apenas no início e no fim da execução do software. Tem como função fazer a transposição entre os dados existentes, e os ficheiros de configuração. Tem implementados apenas dois métodos read e write, respectivamente que têm como função respectivamente ler e escrever do(s) ficheiro(s).

Main (Principal)

Esta é a classe principal do motor de pesquisa. É a mais complexa apesar de não ser a mais longa, nela estão incluídas grande parte das estruturas de dados referentes ao motor de pesquisa desenvolvido. Tem cerca de 20 métodos e cerca de 40 campos.

Nela estão contidos todos os métodos relativos ao processo iterativo de procura de uma solução.

Window (Janela)

Nesta classe estão contidos todos os dados relativos à interface com o utilizador.

Contém uma grande quantidade de campos que são todos os elementos que surgem na janela de interface com o utilizador. É a classe central de todo o módulo de edição de dados.

6.1.2 Arquitectura das Interfaces

A interface será criada numa classe Window que implemente as diversas interfaces Listener (ActionListener, ItemListener, ListSelectionListener, ChangeListener).

Esta classe deverá igualmente conter uma frame (JFrame), assim como os diversos componentes que venham a ser colocados sobre a frame (botões, campos de texto, texto, listas, imagens, etc.).

Aquando da inicialização da aplicação, é criado um objecto da classe Window, que já inclui uma frame, essa frame é inicializada e é criada um JPanel que será o contentPane⁴ desta janela.

Foi criada igualmente uma classe chamada GUICanvas que estende a classe JPanel, que terá como finalidade servir de suporte ao desenho do gráfico.

6.1.3 Nomenclatura

Para efeitos de organização e dado que existiam muitos componentes a colocar nas interfaces e existem também muitos ActionCommands a serem disparados por ocorrência de eventos foi definida uma nomenclatura para os componentes.

Assim sendo cada componente terá o nome da janela / menu em que está inserido, o tipo de componente e a sua função / texto.

Por exemplo:

editSolutionButtonSearchSolution está no menu editar solução, é um botão que ao ser premido procura uma solução.

Para os ActionCommand esta solução já não é viável, porque é possível que botões situados em menus diferentes tenham a mesma função (abrir um terceiro menu por exemplo). Opta-se por distribuir por tipo de evento primeiro e depois por destino (se aplicável).

Assim sendo temos os seguintes tipos de evento:

- Mudança de elemento seleccionado numa lista / combo, é necessário apenas mudar os campos, portanto é apenas necessário saber o local de onde veio, dado que o local para onde vai.
- Interação com os dados existentes, criação, alteração e eliminação de entradas / objectos, neste caso é apenas necessária detectar a janela de origem.
- Chamada a novo menu, neste caso é indiferente a origem do evento. Importa apenas a sua janela de destino.

6.2 Estruturas de dados usadas

Devido ao carácter muito peculiar deste projecto, foram utilizadas diversas estruturas de dados, que serão descritas nesta secção.

Matriz de disponibilidades

Uma parte importante deste projecto, é arranjar uma estrutura em que estivesse indicado tudo o que fosse necessário para a modelação, desde:

⁴ Na maioria das versões do Java é necessário usar este elemento para a adição de componentes, e para outras funções como alterar o Background, o foreground, a fonte e o layout. Nestas versões a tentativa de efectuar `umJFrame.add(umJComponent)` não é possível de efectuar podendo surgir um erro. Assim é necessário realizar: `umJFrame.getContentPane.add(umJComponent)`

- Circulações já atribuídas.
- Circulações por atribuir.
- Circulações já restringidas.

E tudo isto de uma forma que permitisse uma rápida pesquisa dentro da matriz na procura da próxima viagem a atribuir.

A primeira solução encontrada foi recorrendo a uma lista de valores de urgência, que no fundo não era mais que uma lista de dimensão n (em que $n = n^\circ$ de linhas a analisar), em que em cada linha existia o valor máximo (valor máximo = última viagem atribuída + intervalo de passagem máximo) em que se poderia colocar a próxima circulação. Isto permitia-nos ir construindo a solução atribuindo sempre sucessivamente a cada espaço de 3 (intervalo mínimo para circulações diferentes) minutos a viagem que menos podia esperar (que era mais urgente portanto). Este algoritmo encontra-se descrito com mais detalhe mais à frente. Esta solução, no entanto, não foi implementada porque não era satisfatória nos seguintes aspectos:

- Apesar de ser relativamente fácil de identificar os valores máximos a que cada circulação podia ser atribuída, não era possível identificar os valores mínimos, o que podia fazer com que duas circulações das linhas com maior frequência pudessem surgir demasiado próximas uma da outra, o que era errado.
- Não era possível identificar em cada momento se a solução actual ainda poderia conduzir a uma solução possível, ou se as viagens já atribuídas iriam de certa forma impedir qualquer hipótese de concluir o horário com sucesso.
- As restrições relativas à circulação tinham que ser verificadas e corrigidas no final, o que obrigaria à criação de um algoritmo para resolução de conflitos.

Surgiu assim a ideia de combinar tudo isto numa única estrutura, estrutura esta que acabou por ser a matriz de disponibilidades, que é uma array bidimensional, que no entanto, acaba por ser tridimensional. Consiste numa matriz de X por Y em que X é o número de linhas a analisar e Y o número de slots⁵ de tempo aos quais se podem atribuir viagens.

Cada elemento desta matriz pode ter o seguinte formato:

- 1 – Viagem dessa linha atribuída a esse slot.
- -1 – Viagem restringida a esse slot.
- 1XYZ – Nesse instante é possível alocar a X ésima, a Y ésima e Z ésima circulações dessa linha. Por Exemplo, o valor 11234 indica que é possível alocar a esse ponto as primeira, segunda, terceira e quarta viagens dessa mesma linha. O facto de no mesmo campo estar implícita a informação relativa ao facto de um número n de viagens ser possível ou não, indica que apesar de em termos de programação a matriz se tratar de um array bidimensional, na realidade, acaba por ser um array tridimensional (pois ao definir que viagens são ainda possíveis, está a introduzir implicitamente uma terceira dimensão ao array). Inicialmente colocou-se apenas o valor 0 para o caso da viagem ser ainda possível de atribuir, no entanto, nos casos em que o $I_{\max} > 2 \times I_{\min}$, (onde I_{\min}

⁵ Como o período é de 1 hora, e a precisão é de 30 segundos existem 120 slots.

é o intervalo mínimo de circulação e I_{\max} o intervalo máximo), poder-se ia atribuir uma viagem extra entre duas outras se estas estivessem próximas do intervalo máximo.

Por exemplo, supondo um intervalo mínimo de circulação de 13 minutos, e um intervalo máximo de 27 ($(13+27)/2 = 20$, frequência de 3 por hora), poder-se-ia atribuir uma circulação no instante 0, outra no 13 e outra no 27, o que deixaria um espaço de 33 minutos para a seguinte que seria no momento 0 da hora seguinte (60, portanto), para impedir esta situação, é preciso indicar ao software que o momento 27 está disponível, mas apenas para colocar a 2ª viagem dessa linha, e não a 3ª.

Sempre que é introduzida uma nova viagem, esta matriz é actualizada.

Esta matriz é o principal constituinte dos elementos do caminho, sempre que é necessário recuar a uma solução anterior, é recuperada a matriz de disponibilidades do passo anterior.

No entanto esta tabela só por si gera um tempo de acesso muito elevado, para algumas questões fundamentais que são necessárias durante o funcionamento motor de pesquisa: “Quais são os slots que têm viagens atribuídas?”, “A que slots estão atribuídas as viagens da linha X?”. Para qualquer uma destas questões, era necessária uma pesquisa por toda a tabela de disponibilidades, portanto esta solução só por si não era completamente satisfatória, dado que se trata de acessos, que seriam necessários repetir milhares de vezes por segundo durante a pesquisa.

Assim sendo, a solução passou por introduzir duas estruturas auxiliares, uma matriz de pares linha – tempo, em que estão incluídas as linhas e o respectivo slot a que estão atribuídas. Dentro de cada objecto da classe linha, está igualmente um array com o momento a que cada circulação dessa linha foi atribuída.

Passa então a ser possível o acesso directo sem tempo de processamento perdido a todos estes dados.

Figura 16 - Funcionamento do Software Desenvolvido

Carregamento de dados

São lidos os ficheiros de configuração. Esta leitura é feita através do método read() da classe File. Após esta leitura são criadas e preenchidas as estruturas de dados relativas ao motor de pesquisa.

Após este passo, o pesquisador entra em ciclo dentro do motor de pesquisa, a solução inicial deste ciclo é uma solução sem nenhuma circulação atribuída, portanto é uma solução não válida.

Procura da próxima circulação a atribuir

Através da função next(), é obtida a próxima viagem a atribuir, usando a matriz de disponibilidades e os contadores.

Esta função segue o algoritmo descrito na Figura 17.

```

if(c > contadorMaximo)
    return false;

for(i = 0; i < nSlots; i++)
    for(j = 0; j < nLinhas; j++)
        if((forPossivelAtribuir(matrizDisponibilidades[j][i],
            linhas[j].viagensJaAtribuidas+1) && (linhas[j].ciagensJaAtribuidas+1 <
            linhas[j].frequencia))
            if(contador-- == 0)
            {
                atribui(j, i);
                return true;
            }
    }

```

Figura 17 – Pseudo – Código para a procura da próxima circulação a atribuir

A matriz de disponibilidades chega a este ponto, ordenada pela prioridade de cada linha na atribuição. Os critérios para a escolha da próxima circulação a atribuir são os seguintes:

1. Circulação livre no primeiro slot disponível.
2. Dentro do mesmo slot, circulação com prioridade mais alta.

Sempre que é encontrada uma solução, um contador paralelo (o original é mantido) é decrementado, e a circulação apenas é atribuída se este contador já estiver em zero, caso contrário significa que essa atribuição já foi testada antes. Caso seja percorrida toda a matriz de disponibilidades e não seja encontrado nenhum espaço livre ainda não testado, a função retorna 'false' e indica ao motor de pesquisa que deve recuar.

Atribuição

É neste passo que se faz a passagem entre a solução actual, e uma sua vizinha, através da estrutura de vizinhança definida para este caso.

Uma solução A é vizinha de uma outra B, se for possível passar de A para B adicionando ou removendo uma circulação já atribuída.

A Atribuição de uma circulação de uma linha a um dado slot de tempo, consiste inicialmente em inserir a circulação às listas de viagens atribuídas na classe principal, na classe linha, assim como na matriz de disponibilidades.

Seguidamente são aplicadas as restrições que esta atribuição causa às outras linhas, restrições estas que podem ser aplicadas ao próprio sentido (diferentes velocidades), ou ao sentido inverso (cruzamentos, ligações).

São então aplicadas as restrições relativas aos intervalos mínimo e máximo de circulação, não só para a circulação seguinte, mas sim, para todas as circulações ainda por atribuir dessa mesma linha. Quando uma circulação é atribuída, esta abre uma “janela” de slots num espaço futuro onde se pode inserir a viagem seguinte, sendo que cada um destes slots abrirá outra janela num espaço a seguir, sendo que o conjunto de todos estes slots abrirá um único espaço mais alargado.

Suponhamos o seguinte caso: uma linha com frequência de 6 por hora, um intervalo mínimo de circulação de 9 minutos, e um intervalo máximo de circulação de 11 minutos. Atribuindo a primeira circulação desta linha nos 3 minutos, as janelas para as circulações seguintes serão:

- 2ª Circulação – 12 – 14 minutos
- 3ª Circulação – 21 – 25 minutos
- 4ª Circulação – 30 – 36 minutos
- 5ª Circulação – 39 – 47 minutos
- 6ª Circulação – 48 – 58 minutos

No entanto um olhar mais atento permite-nos facilmente chegar à conclusão que estas janelas não são realistas, dado que não será possível atribuir uma circulação aos 58 minutos, dado que ficaria com apenas 5 minutos de intervalo para a circulação atribuída, não respeitando portanto o intervalo mínimo definido.

Assim, é necessário repetir o procedimento mas recuando no tempo:

- 6ª Circulação – -8 – -6 (52 – 54) minutos
- 5ª Circulação – -19 – -15 (41 – 45) minutos
- 4ª Circulação – -30 – -24 (30 – 36) minutos
- 3ª Circulação – -41 – -33 (19 – 27) minutos
- 2ª Circulação – -52 – -42 (8 – 18) minutos

É necessário colocar os valores todos entre 0 e 60 minutos, isto é possível, porque o horário definido é o mesmo para a hora seguinte e também para a anterior.

Sendo assim, os slots pertencentes às janelas reais são os que pertencem a ambas as janelas, ou seja:

- 2ª Circulação – 12 – 14 minutos
- 3ª Circulação – 21 – 15 minutos
- 4ª Circulação – 30 – 36 minutos
- 5ª Circulação – 41 – 45 minutos
- 6ª Circulação – 52 – 56 minutos

Por último, o estado actual é adicionado ao caminho já percorrido.

Algoritmos desenvolvidos

- Avanço e recuo
- Função de Avaliação
- Cálculo do número de veículos

Estruturas

- Algoritmo de disponibilidades
- Linhas atribuídas

Mostrar gráficos de desempenho

- Frequência por versão
- Tempos necessários à pesquisa de uma solução

Avaliação

Sempre que é encontrada uma solução melhor que a anterior, é necessário, avaliar esta solução. Define-se uma boa solução, uma solução em que os intervalos de passagem sejam regulares. No entanto há várias formas de avaliar esta condição.

$$\sum_{i=0}^n \sum_{j=0}^k (g_i^j - g_i^b)^2$$

Em que g_i^j é o intervalo conseguido, g_i^b é o intervalo ideal, n é o número de linhas existentes e k o número de circulações de cada linha.

Nas primeiras versões utilizou-se para avaliar apenas o quadrado da diferença em slots de cada intervalo em relação aos intervalos ideais, de acordo com a seguinte fórmula:

No entanto esta solução não era a melhor, porque considerava todos os desvios como tendo a mesma importância, quando na realidade, um minuto perdido numa linha em que o intervalo ideal seria 30, não deverá ter a mesma importância que um minuto perdido, num espaço em

que o intervalo ideal é de apenas 10, portanto a opção passou a ser por somar o quadrado do rácio desvio / tempo ideal.

$$\sum_{i=0}^n \sum_{j=0}^k \frac{(g_i^j - g_i^b)^2}{g_i^b}$$

Ainda assim, mesmo que o intervalo ideal fosse semelhante, o tempo perdido numa linha podia ser mais grave que o tempo perdido noutra, era necessário dar uma certa prioridade a essa mesma linha, a solução passou por multiplicar o quadrado do rácio descrito anteriormente e multiplicá-lo pelo prioridade dessa linha, esta solução é a utilizada actualmente.

$$\sum_{i=0}^n \sum_{j=0}^k P_i \frac{(g_i^j - g_i^b)^2}{g_i^b}$$

Recuo

Inicialmente, esta função de recuo, apenas era utilizada sempre que o pesquisador atingia um beco sem saída, isto fazia com que fosse perdido demasiado tempo em caminhos que não conduziam a nada. Assim foi implementada antes da pesquisa de uma nova solução, um algoritmo que nos permitirá verificar se ainda é possível colocar todas as circulações, sem entrar em conflito com as já atribuídas, isto reduz de certa forma a frequência de processamento, no entanto aumenta drasticamente a eficiência do sistema, isto porque, apesar do número de soluções testadas ser muito inferior, o número de soluções possíveis analisadas é muito superior, conduzindo assim a soluções muito melhores.

Isto pode ser verificado na Tabela 2, onde para o cenário C7 (6 circulações da linha A, 3 da linha B, 4 da linha C, 3 da linha E e 1 Expresso), apesar de a frequência média ter sido reduzida, o número de soluções que foi necessário experimentar até encontrar uma solução válida foi cerca de 7 vezes inferior, o que fez com que o tempo necessário fosse 5 vezes inferior.

	Solução sem Cortes	Solução com cortes
Tempo para procura da primeira solução C7	3815 s	845 s
Frequência Média	42154 / s	27534 / s
Iterações até à primeira solução C7	160 milhões	23 milhões

Tabela 2 – Comparação de Eficiência entre recuo com e sem cortes

Dado que este se trata de um problema com resolução em árvore, o correcto dimensionamento dessa árvore é fundamental, caso contrário o sistema entraria num ciclo que o levaria a analisar um número exagerado de soluções. É portanto imperioso definir que cada ramo não possa ter mais que n ramificações, sendo que a correcta determinação do valor de n , é fundamental. Definiu-se então que cada ramo não teria mais que $n_{Linhas} + 1$ ramificações, isto permite que na maioria dos casos, seja testada como próxima atribuição qualquer uma das linhas, e pelo menos uma delas num *slot* distinto.

Contagem de Veículos

Aparentemente há duas formas de fazer esta atribuição, o critério é o mesmo mas pode ser orientado às partidas ou orientado às chegadas, isto é, pode-se “pegar” numa partida e tentar ligá-la a uma chegada anterior, ou pode-se pegar numa chegada e tentar ligá-la à partida seguinte.

Definição da classe circulação, que tem como elementos:

- **Linha** – elemento da classe linha, contém a linha a que se refere a circulação em causa.
- **Partida** – hora de partida, em unidades de tempo (tipicamente 30 segundos).
- **Chegada** – hora de chegada, em unidades de tempo (tipicamente 30 segundos).
- **noPartida** – elemento da classe nó, contém o nó de partida desta circulação.
- **noChegada** – elemento da classe nó, contém o nó de chegada desta circulação.

Assim, quando é obtida uma solução, é automaticamente criada a lista de circulações relativas a essa mesma solução.

Este é o primeiro passo necessário no cálculo do número de veículos necessários para pôr em prática a solução em causa.

Algoritmo em pseudo - código:

Exportação

A exportação da solução encontrada pode ser feita para um formato que possa mais tarde ser lido pelo módulo de design.

Podem igualmente ser gerados códigos HTML, no entanto, esta geração é algo complexa, dado que é feita em forma de armazenamento triplo, primeiro são ordenadas as circulações, seguidamente são calculados os tempos, por fim é gerado o código HTML, e no final este código é guardado num ficheiro.

```

while(vehicleCheck() )/*1*
{
    local = getNextFree() ;
    for(i=0 ; i < nSlots*2; )/*2*/
    {
        procuraPrimeiraSaída;
        atribui;
        local = noChegada;
        i = arrival;
    }
    vehicle ++;
}

```

Figura 18 – Pseudo – Código para contagem de veículos

Em que:

- *1* Cada iteração neste ciclo representará um veículo.
- *2* Cada iteração neste ciclo representa uma circulação.
- O método getNextFree(), devolve o nó onde se encontra a próxima saída.

Assim sendo, enquanto não estiverem atribuídas viagens a todas as circulações, procurar-se-á o local de onde saíra a primeira circulação, a essa circulação adiciona-se-lhe o tempo de viagem, o local actual passa a ser o local de destino dessa circulação e ao tempo actual é-lhe somado o tempo dessa viagem. Este ciclo é repetido até passar duas vezes o ciclo do horário. Quando for passado o dobro do ciclo do horário, significa que este veículo já tem programa definido para as próximas duas horas, e portanto passa-se a atribuir as viagens para o próximo veículo.

É simulada a circulação durante dois períodos de tempo, pelo facto de só assim ser possível gerir o intervalo de passagem entre cada período, e também pelo facto de haver linhas cujo tempo de circulação é superior a um período.

Exportação

A exportação da solução encontrada pode ser feita para um formato que possa mais tarde ser lido pelo módulo de design.

Pode igualmente ser gerado código HTML, no entanto, esta geração é algo complexa, dado que é feita em forma de varrimento triplo, primeiro são ordenadas as circulações, seguidamente são calculados, os tempos, por fim é gerado o código HTML, e no final este código é guardado num ficheiro.

The screenshot shows a web browser window with the following content:

Linha A

10:00		0:00	
19:00	9:00	11:00	11:00
28:00	9:00	20:00	9:00
39:00	11:00	29:00	9:00
48:00	9:00	40:00	11:00
59:00	11:00	50:00	10:00
70:00	11:00	60:00	10:00

Linha B

2:00		8:00	
22:00	20:00	32:00	24:00
45:00	23:00	44:00	12:00
62:00	17:00	68:00	24:00

Linha C

13:00		3:00	
31:00	18:00	23:00	20:00
42:00	11:00	37:00	14:00
53:00	11:00	47:00	10:00
73:00	20:00	63:00	16:00

Figura 19 – Parte de ficheiro HTML gerado pelo software

6.3.2 Funcionamento do editor de dados

Para o funcionamento do editor de dados, e do módulo de desenho do horário, é necessário, o carregamento de todos os dados. No entanto, estes dados não são necessários durante o funcionamento do motor de pesquisa, portanto são carregados para classes à parte. Essas classes seguem parte da estrutura seguida no EMME/2. Existem portanto as classes onde existem as classes Line, Link, Node, que serão substituídas por: LineDescription (Line já estava definido anteriormente), Arc, Node.

Nos requisitos do software a desenvolver está descrita a importação de dados neste formato, formato este que consiste em ficheiros de extensão 'in', onde os que são usados neste software são os ficheiros d211.in e d221.in, ficheiros estes que eram editados directamente através de um simples editor de texto (como o notepad).

No ficheiro d211.in estão contidos os dados relativos aos nós, e aos arcos, enquanto no ficheiro d221.in se encontra a descrição de cada circulação / linha⁶.

Figura 20 - Descrição dos nós no ficheiro d211.in

ID	X	Y	Z	...
8000	0.00	0.00	0.00	...
8001	1.00	1.00	1.00	...
8002	2.00	2.00	2.00	...
8003	3.00	3.00	3.00	...
8004	4.00	4.00	4.00	...
8005	5.00	5.00	5.00	...
8006	6.00	6.00	6.00	...
8007	7.00	7.00	7.00	...
8008	8.00	8.00	8.00	...
8009	9.00	9.00	9.00	...
8010	10.00	10.00	10.00	...
8011	11.00	11.00	11.00	...
8012	12.00	12.00	12.00	...
8013	13.00	13.00	13.00	...
8014	14.00	14.00	14.00	...
8015	15.00	15.00	15.00	...
8016	16.00	16.00	16.00	...
8017	17.00	17.00	17.00	...
8018	18.00	18.00	18.00	...
8019	19.00	19.00	19.00	...
8020	20.00	20.00	20.00	...
8021	21.00	21.00	21.00	...
8022	22.00	22.00	22.00	...
8023	23.00	23.00	23.00	...
8024	24.00	24.00	24.00	...
8025	25.00	25.00	25.00	...
8026	26.00	26.00	26.00	...
8027	27.00	27.00	27.00	...
8028	28.00	28.00	28.00	...
8029	29.00	29.00	29.00	...
8030	30.00	30.00	30.00	...

Figura 21 - Descrição dos arcos no ficheiro d211.in

⁶ No EMME/2 é feita uma descrição por linha, no entanto, na definição de linha neste software é obrigatório que uma linha tenha uma frequência fixa. Isto faz com que para induzir linhas com uma frequência não constante, se tenha que definir cada conjunto de circulações (circulações no mesmo momento do período, mas em períodos diferentes) como uma circulação independente.

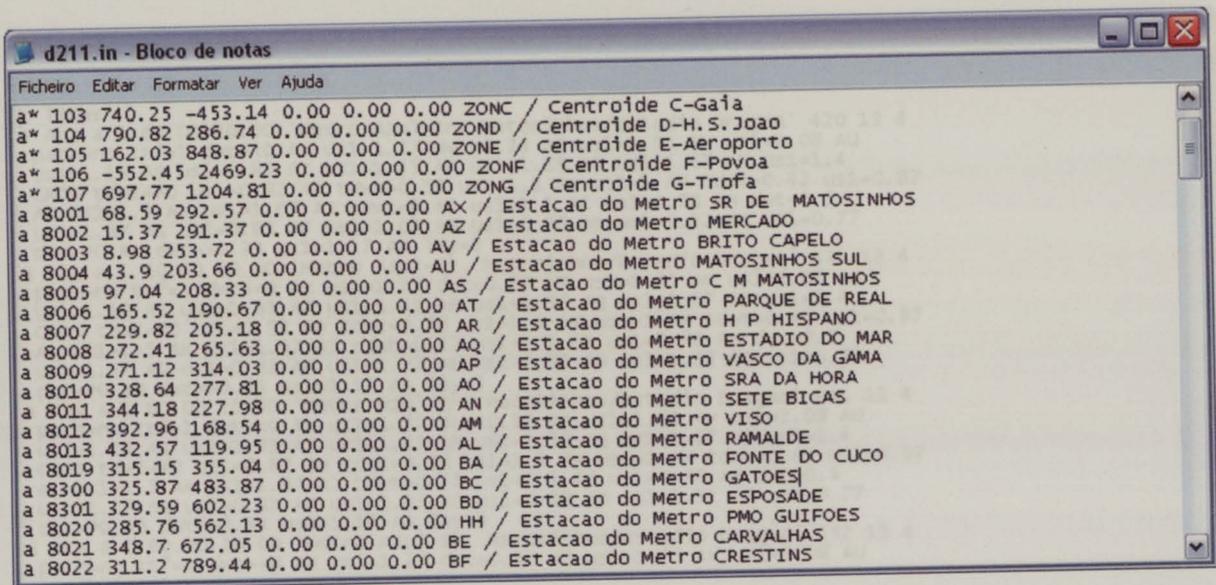


Figura 20 – Descrição dos nós no ficheiro d211.in

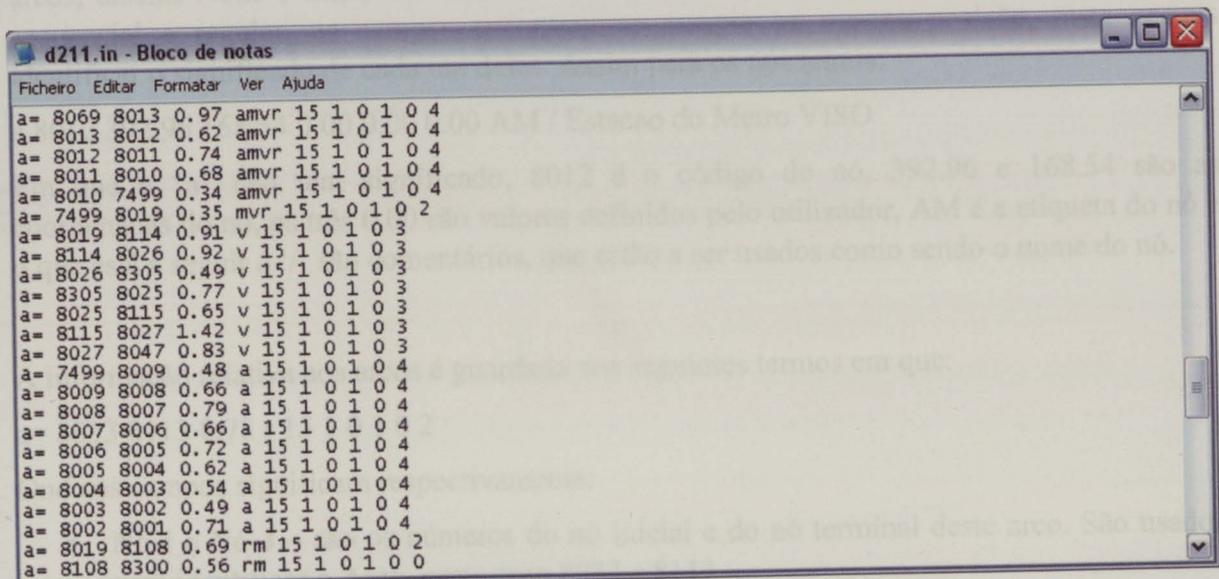


Figura 21 – Descrição dos arcos no ficheiro d211.in

```

t lines
a A1a a 1 60.00 20.00 'Metro Azul-Sr. Matosinhos/Estadio Dragao' 420 13 4
path=no ttf=1 dwt=<0 AX us1=1.27 dwt=0.33 AZ us1=1.55 AV us1=2.08 AU
us1=1.18 AS us1=1.83 AT us1=1.45 AR us1=1.58 AQ us1=1.67 AP us1=1.4
CROS tdwt=#0 us1=0.82 AO tdwt=0.58 us1=1.3 AN us1=1 AM tdwt=0.42 us1=0.87
AL us1=1.5 AK us1=1.67 AJ tdwt=0.42 us1=1.1 AI us1=0.78 AH dwt=0.5
us1=1.67 AG us1=0.83 AF us1=1.03 AE us1=0.92 AD us1=1.32 AC us1=0.77
dwt=#0 AB dwt=>0.33 us1=1.17 AA lay=3
a A2a a 1 60.00 20.00 'Metro Azul-Sr. Matosinhos/Estadio Dragao' 431 13 4
path=no ttf=1 dwt=<0 AX us1=1.27 dwt=0.33 AZ us1=1.55 AV us1=2.08 AU
us1=1.18 AS us1=1.83 AT us1=1.45 AR us1=1.58 AQ us1=1.67 AP us1=1.4
CROS tdwt=#0 us1=0.82 AO tdwt=0.58 us1=1.3 AN us1=1 AM tdwt=0.42 us1=0.87
AL us1=1.5 AK us1=1.67 AJ tdwt=0.42 us1=1.1 AI us1=0.78 AH dwt=0.5
us1=1.67 AG us1=0.83 AF us1=1.03 AE us1=0.92 AD us1=1.32 AC us1=0.77
dwt=#0 AB dwt=>0.33 us1=1.17 AA lay=3
a A3a a 1 60.00 20.00 'Metro Azul-Sr. Matosinhos/Estadio Dragao' 441 13 4
path=no ttf=1 dwt=<0 AX us1=1.27 dwt=0.33 AZ us1=1.55 AV us1=2.08 AU
us1=1.18 AS us1=1.83 AT us1=1.45 AR us1=1.58 AQ us1=1.67 AP us1=1.4
CROS tdwt=#0 us1=0.82 AO tdwt=0.58 us1=1.3 AN us1=1 AM tdwt=0.42 us1=0.87
AL us1=1.5 AK us1=1.67 AJ tdwt=0.42 us1=1.1 AI us1=0.78 AH dwt=0.5
us1=1.67 AG us1=0.83 AF us1=1.03 AE us1=0.92 AD us1=1.32 AC us1=0.77
dwt=#0 AB dwt=>0.33 us1=1.17 AA lay=3
a A4a a 1 60.00 20.00 'Metro Azul-Sr. Matosinhos/Estadio Dragao' 452 13 4
path=no ttf=1 dwt=<0 AX us1=1.27 dwt=0.33 AZ us1=1.55 AV us1=2.08 AU
    
```

Figura 22 – Descrição dos arcos no ficheiro d221.in

Como se pode ver pelas imagens anteriores o carregamento dos dados relativos aos nós e aos arcos, tabelas Node e Link, não envolve nenhuma dificuldade em especial. A informação é sequencial e regular, os campos encontram-se sempre na mesma posição, resta apenas identificar o significado de cada um deles. Assim para os nós temos:

a 8012 392.96 168.54 0.00 0.00 0.00 AM / Estacao do Metro VISO

Em que o “a” não tem significado, 8012 é o código do nó, 392.96 e 168.54 são as coordenadas do nó, os três 0.00 são valores definidos pelo utilizador, AM é a etiqueta do nó e o que está a seguir à ‘/’ são comentários, que estão a ser usados como sendo o nome do nó.

A informação relativa aos arcos é guardada nos seguintes termos em que:

a= 8033 8113 0.78 r 15 1 0 1 0 2

Onde os campos significam respectivamente:

- **Nó-I e Nó-J** – são os números do nó inicial e do nó terminal deste arco. São usados para identificar o Arco, neste caso 8033 e 8113.
- **Comprimento do arco** – definido em unidades de distância, de 0.01 a 999.99 (só são permitidas duas casa decimais). Se for inserido um * em vez de um valor, o comprimento do arco será calculado (através das coordenadas) como sendo a distância em linha recta de i a j. Atenção que, para usar a funcionalidade * correctamente as coordenadas do nó devem estar expressas em unidades de comprimento, ou o factor de conversão respectivo deve estar definido no módulo 1.23 (opção 6). Atenção também que, quando forem modificadas as coordenadas dos nós, o comprimento do arco não será recalculado automaticamente. Neste caso o comprimento do arco é 0,78.
- **Modos** – Lista de modos que são permitidos sobre este arco. Se o modo automático estiver disponível, diz-se que este é um arco automático. Neste caso o único modo que pode circular por este arco é ‘r’ que corresponde à linha vermelha (red).

- **Tipo de Arco** – classificação dos arcos definida pelo utilizador. Esta classificação pode ser útil para definir sub redes. O tipo de arco é um número positivo entre 1 e 999. Neste caso 15.
- **Número de Vias** – na escala de 0,1 a 9,9 para os arcos automático e normalmente 0 para os outros. Neste caso 1 (por ser uma via dedicada).
- **Volume Delay Function** – Fornece o tempo de viagem num arco, tempo este que é calculado a partir de parâmetros como a velocidade máxima, o número de faixas de rodagem, o comprimento, a capacidade, etc. Neste caso tem o valor 0, porque não faz sentido o seu uso no transporte ferroviário.
- **Itens definidos pelo utilizador** – Três itens definidos pelo utilizador. Neste caso 1, 0 e 2.

A leitura do ficheiro das circulações é substancialmente mais complexa, dado que não existe uma sequência dos dados, definida à partida.

Logo no momento da leitura do ficheiro, são necessários cuidados especiais, isto porque ao contrário dos outros casos em que a informação se encontrava bem definida, a informação relativa a um arco ou um nó encontrava-se devidamente separada, no entanto, no caso das linhas, a informação encontra-se colocada em múltiplas linhas, que devem ser consideradas como uma só.

A informação relativa à descrição duma linha vem no seguinte formato:

```
a A1a1 a 1 60.002 20.003 'Metro Azul-Sr. Matosinhos/Estadio Dragao'4 4205 136 47
path=no8 ttf=19 dwt=<0 AX us1=1.27 dwt=0.33 AZ us1=1.55 AV us1=2.08 AU
us1=1.18 AS us1=1.83 AT us1=1.45 AR us1=1.58 AQ us1=1.67 AP us1=1.4 CROS tdwt=#0
us1=0.82 AO tdwt=0.58 us1=1.3 AN us1=1 AM tdwt=0.42 us1=0.87 AL us1=1.5 AK
us1=1.67 AJ tdwt=0.42 us1=1.1 AI us1=0.78 AH dwt=0.5 us1=1.67 AG us1=0.83 AF
us1=1.03 AE us1=0.92 AD us1=1.32 AC us1=0.77 dwt=#0 AB dwt=>0.33 us1=1.17 AA
lay10=3
```

Onde:

1. Código de circulação/linha.
2. Período.
3. Velocidade definida por defeito, é usada apenas caso não haja nenhuma outra definida.
4. Nome.
5. Hora de Início.
6. Nº de repetições do período a partir da hora de início.
7. Cor para representação gráfica.
8. Procura do caminho, 'no' significa que o caminho é indicado directamente.

9. Forma de cálculo do tempo de viagem, 1 significa que este é indicado directamente e não é calculado a partir da velocidade.
10. Tempo de suporte - tempo mínimo de paragem no termino.

Estes elementos encontram-se sempre nas mesmas posições, sendo que os 9 primeiros estão sempre no início, enquanto o 10 (lay) encontra-se sempre no final.

É então necessário efectuar a leitura do percurso das linhas, e neste ponto há diversos tipos de elementos que podem ser lidos por uma ordem diversa:

- dwt – este elemento indica o tempo de paragem a realizar nas paragens a partir deste ponto, e até surgir um novo elemento dwt.
- tdwt – indica um tempo de paragem temporário para a paragem na última estação referida no caminho.
- dwt# – a partir deste ponto, a circulação não parará nos nós por onde passe.
- tdwt# – Indica que a composição não parará no último nó por onde passa, mantendo para as outras o tempo definido aquando do aparecimento da flag dwt.
- dwt> – tempo de paragem no nó de início da circulação.
- dwt< – tempo de paragem no nó final da circulação.
- us1 – tempo de circulação entre dois nós.
- LABEL ou CÓDIGO – É o elemento que indica o caminho percorrido, sempre que surge um elemento não identificável como um dos anteriores, considera-se que é uma label ou código de um nó, e deve ser pesquisado na rede o nó que corresponde a esta label ou código.

Desenho das linhas

O desenho do gráfico do horário é uma representação bidimensional do horário para um dado segmento num dado intervalo de tempo, não sendo possível representar toda a rede, pois existem circulações simultâneas em troços não sequenciais. Assim sendo pretende-se que o utilizador defina os extremos temporais (tempo inicial e final) e espaciais (nó inicial e final).

Como o caminho não é indicado directamente é necessário calcular quais os arcos (segmentos atómicos) é que pertencem ao intervalo seleccionado.

Sendo assim, o desenho processa-se em três fases:

- **Procura de um caminho entre os nós seleccionados** – Nesta fase, procura-se uma Linha que passe pelos dois nó seleccionados, sendo que todos os nós / arcos existentes entre estes dois serão o caminho a ser desenhado.
- **Definição dos Arcos a desenhar** – São percorridos os arcos todos de todas as circulações, e definidos quais é que estão dentro do caminho percorrido e quais é que não estão.

- **Desenho do gráfico** – Por último é desenhado o gráfico, sendo que aqui são verificados quais os arcos que estão dentro do intervalo de tempo seleccionado e quais é que não estão.

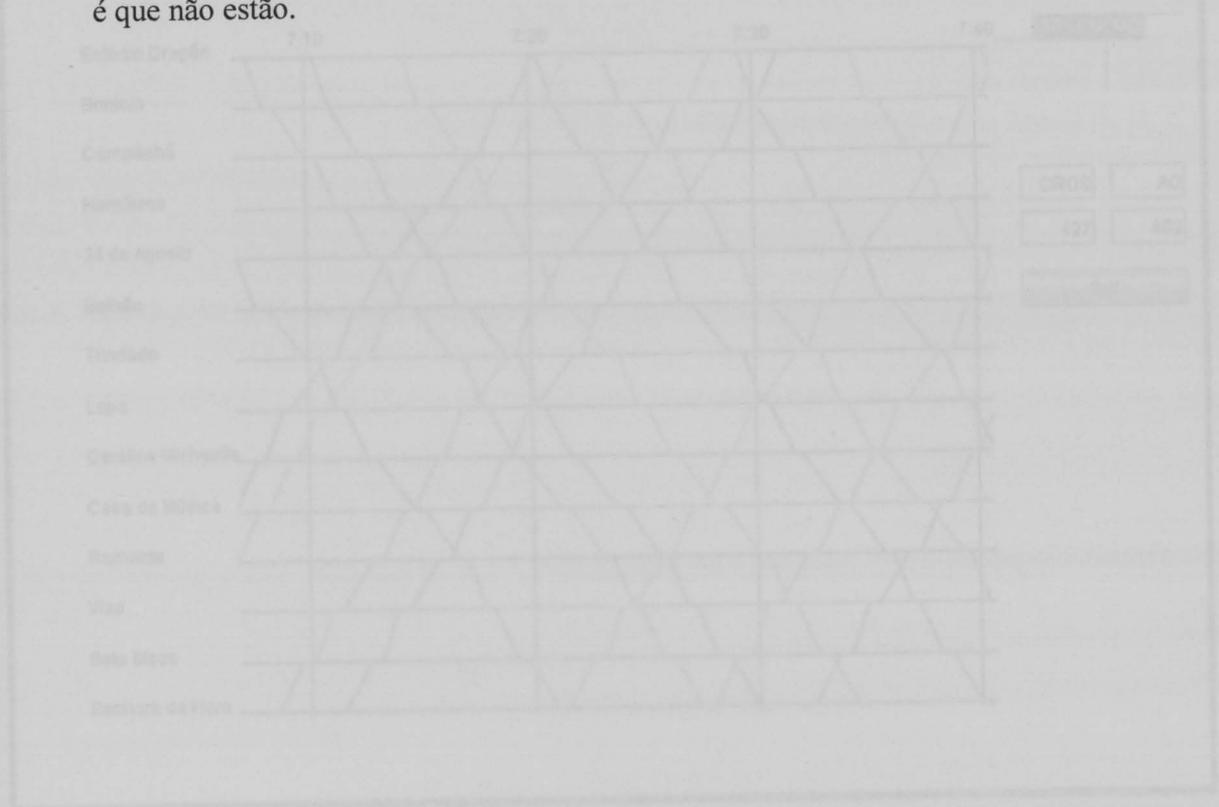


Figura 23 – Gráfico de Horário

6.2.3 Outros algoritmos testados

Para pesquisa da solução foram implementados outros algoritmos, não usados na versão final. Todos os algoritmos foram alvo de uma análise e teste, em que acabou por ser usado o que produzia melhores resultados.

Solução utilizando valores de urgência

Construção da solução inicial, partindo de um conjunto vazio, ir adicionando sucessivamente a viagem que menos tempo pode esperar.

A ideia é ir construindo a solução, indicando as viagens uma de cada vez consoante a prioridade que cada uma delas tem, ou seja, inicia-se com uma solução vazia, seguidamente adiciona-se-lhe no momento 0 a viagem mais urgente, depois passados 3 minutos (6 ciclos de 30 segundos); adiciona-se a mais urgente e por aí fora.

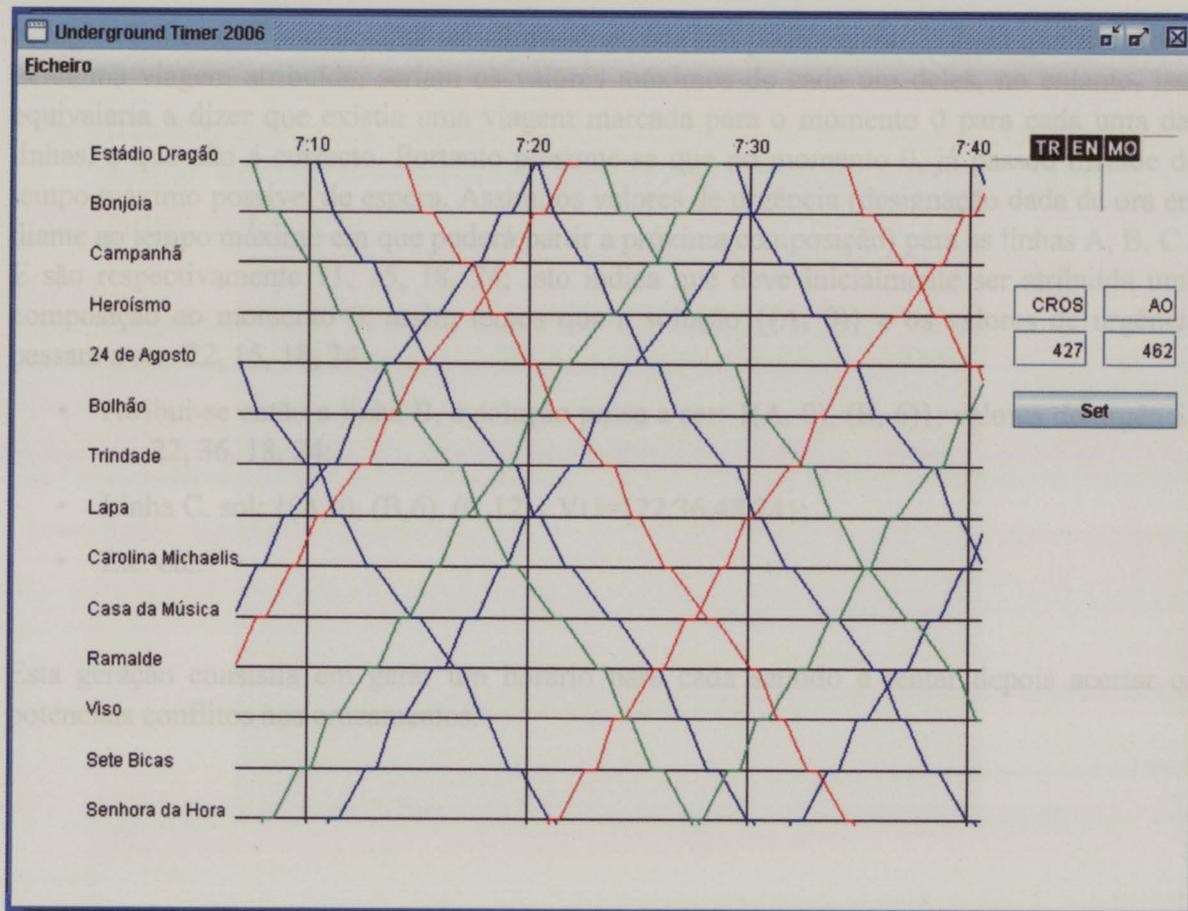


Figura 23 – Gráfico de Horário

6.3.3 Outros algoritmos testados

Para pesquisa da solução foram implementados outros algoritmos, não usados na versão final. Todos os algoritmos foram alvo de uma análise e teste, em que acabou por ser usado o que produzia melhores resultados.

Solução utilizando valores de urgência

Construção da solução inicial, partindo de um conjunto vazio, ir adicionando sucessivamente a viagem que menos tempo pode esperar.

A ideia é ir construindo a solução, indicando as viagens uma de cada vez consoante a prioridade que cada uma delas tem, ou seja, inicia-se com uma solução vazia, seguidamente adiciona-se-lhe no momento 0 a viagem mais urgente, depois passados 3 minutos (6 ciclos de 30 segundos), adiciona-se a mais urgente e por aí fora.

A lógica indica que inicialmente os valores que cada um pode esperar, e dado que não existe nenhuma viagem atribuída, seriam os valores máximos de cada um deles, no entanto, isso equivaleria a dizer que existia uma viagem marcada para o momento 0 para cada uma das linhas, o que não é correcto. Portanto presume-se que no momento 0, já passou metade do tempo máximo possível de espera. Assim, os valores de urgência (designação dada de ora em diante ao tempo máximo em que poderá partir a próxima composição) para as linhas A, B, C e E são respectivamente 11, 15, 18, 24; Isto indica que deve inicialmente ser atribuída uma composição ao momento 0, assim temos que a solução $\{(A, 0)\}$ e os valores de urgência passam a ser: 22, 15, 18, 24

- Atribui-se então a linha B, a solução passa a ser: $\{(A, 0), (B, 6)\}$, valores de urgência 22, 36, 18, 24;
- Linha C, sol: $\{(A,0), (B,6), (C,12)\}$ $VU=\{22,36,48,24\}$;
- Etc. etc.

Esta geração consistia em gerar um horário para cada sentido e tentar depois acertar os potenciais conflitos nos cruzamentos.

Recursividade aleatória

Outra das soluções tentadas foi a da recursividade aleatória, recursividade esta que foi implementada em dois locais, primeiro no momento de recuar, o sistema recuaria não uma vez, mas sim um número aleatório de vezes. É igualmente implementada no algoritmo de pesquisa de uma nova solução, que em vez de aceitar a solução proposta naturalmente, só a aceita com uma dada probabilidade p .

Esta solução não foi no entanto satisfatória dado que o pesquisador navegava durante largos períodos de tempo em soluções que estavam longe de ser as melhores.

Foi igualmente tentada a pesquisa com a recursividade aleatória instalada apenas num dos locais (primeiro apenas na atribuição, depois apenas no recuo), e os resultados embora um pouco melhores que na anterior continuaram a não ser satisfatórios, o que levou a não repetir esta solução.

6.4 Solução implementada

O editor de dados consiste num conjunto de menus e formulários nos quais se podem editar todos os dados e parâmetros relativos à execução do motor de pesquisa.

Após uma janela de entrada (Figura 24), o utilizador pode seleccionar a opção que pretende através do menu identificado na Figura 25:

- **Carregar Solução** – Nesta opção o utilizador pode seleccionar uma solução já calculada, no sentido de a editar e/ou de a desenhar.
- **Guardar Solução** – Pode ser guardada para um formato legível pelo software, a solução actualmente existente.
- **Editar Dados** – Conjunto de separadores onde são editados os dados relativos a toda a rede de transporte ferroviário.
- **Editar Solução** – Nesta situação é editada a solução que foi carregada de um ficheiro já calculado anteriormente, ou gerada automaticamente.
- **Executar** – É lançado o motor de pesquisa.
- **Desenhar Gráfico** – É desenhado o gráfico relativo ao horário existente.
- **Sair** – Abandonar o software.

Figura 25 – Menu de Selecção de Opções



Figura 24 – Interface de entrada do editor de dados

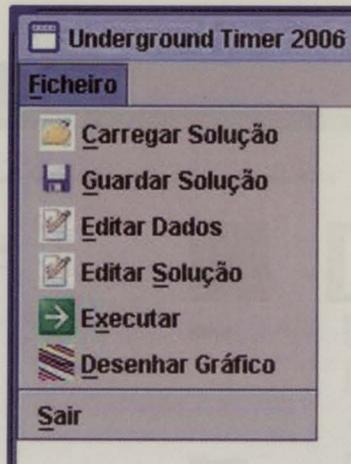


Figura 25 – Menu de Selecção de Opções

Nos menus de edição podem ser realizadas as três operações básicas em relação a cada um dos dados necessários:

- **Editar** – – Editar um nó/arco/linha já existente.
- **Novo** – – Criar um novo nó/arco/linha.
- **Eliminar** – – Apagar o nó/arco/linha definidos.

- **Controlado** – (Booleano) Indica se este nó se refere a um controlado ou não.
- **Número** – (Inteiro) Código que no software EMME/2 é usado como chave primária do nó, neste software é usado apenas como referência indicativa, dado que cada nó tem um código interno não definido. Mantém-se aqui porque os ficheiros de entrada deste software são ficheiros previamente definidos para o referido EMME/2.
- **Label** – (Texto) Tal como no anterior, não tem nenhuma relevância especial para este software, mas é igualmente usado no software EMME/2, dado que é usado na descrição do caminho das linhas.
- **X e Y** – (Decimais) Coordenadas do nó.

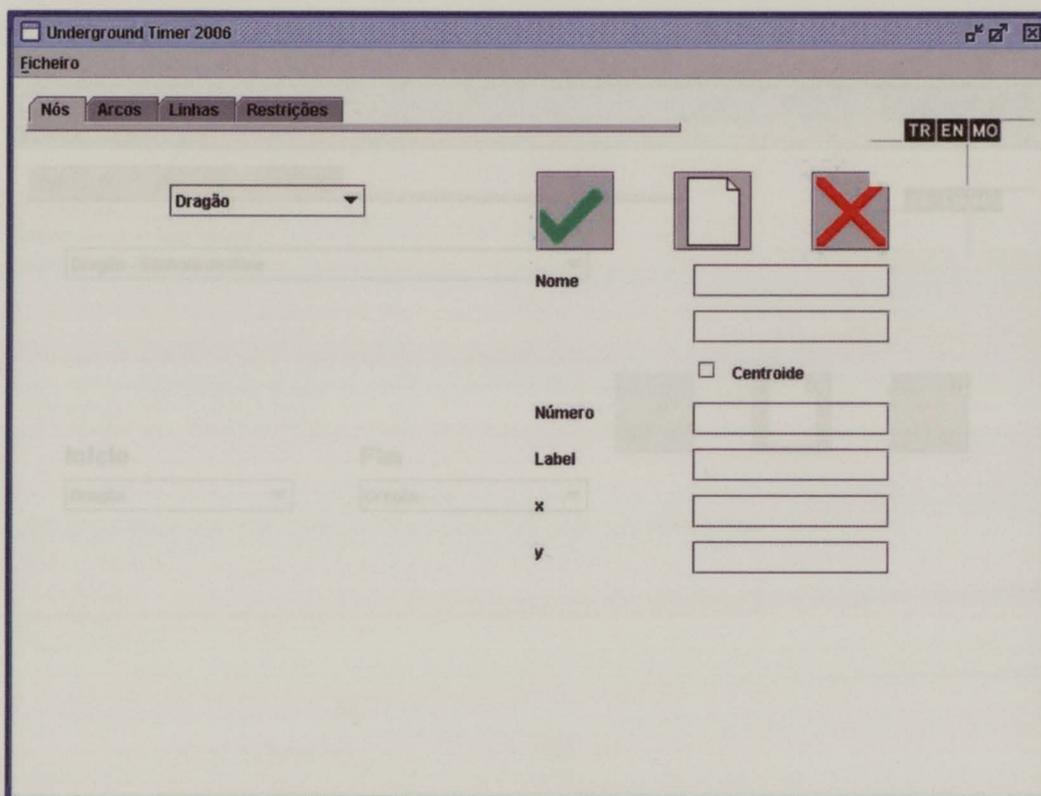


Figura 26 – Menu de Edição de Nós

Como se pode ver na Figura 26, os dados que podem ser editados são:

- **Nome** – (Texto) Nome do nó;
- **Centróide** – (Booleano) Indica se este nó se refere a um centróide ou não.
- **Número** – (Inteiro) Código que no software EMME/2 é usado como chave primária do nó, neste software é usado apenas como referência indicativa, dado que cada nó tem um código interno não definido. Mantém-se aqui porque os ficheiros de entrada deste software são ficheiros previamente definidos para o referido EMME/2.
- **Label** – (Texto) Tal como no anterior, não tem nenhuma relevância especial para este software, mas é igualmente usado no software EMME/2, dado que é usado na descrição do caminho das linhas.
- **X e Y** – (Decimais) Coordenadas do nó.

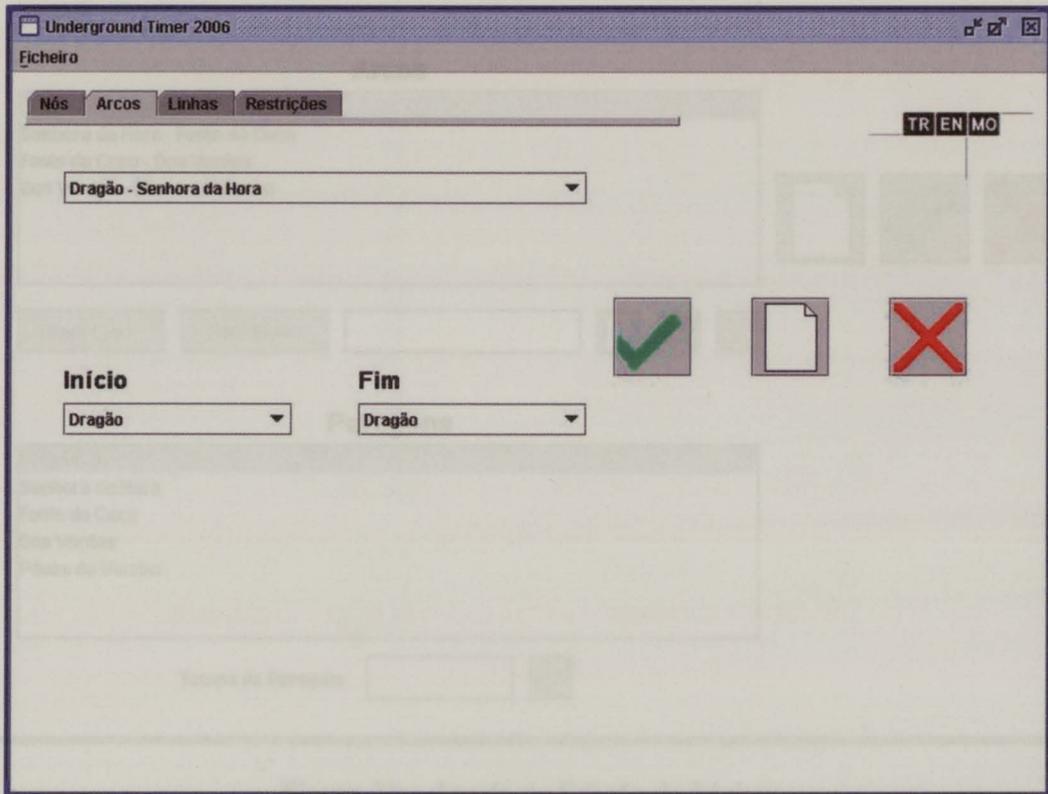


Figura 27 – Janela de Edição de Arcos

Um arco é apenas a junção de dois nós, assim sendo um arco consiste apenas em alterar o nó de início e de fim, como se pode ver na Figura 27.

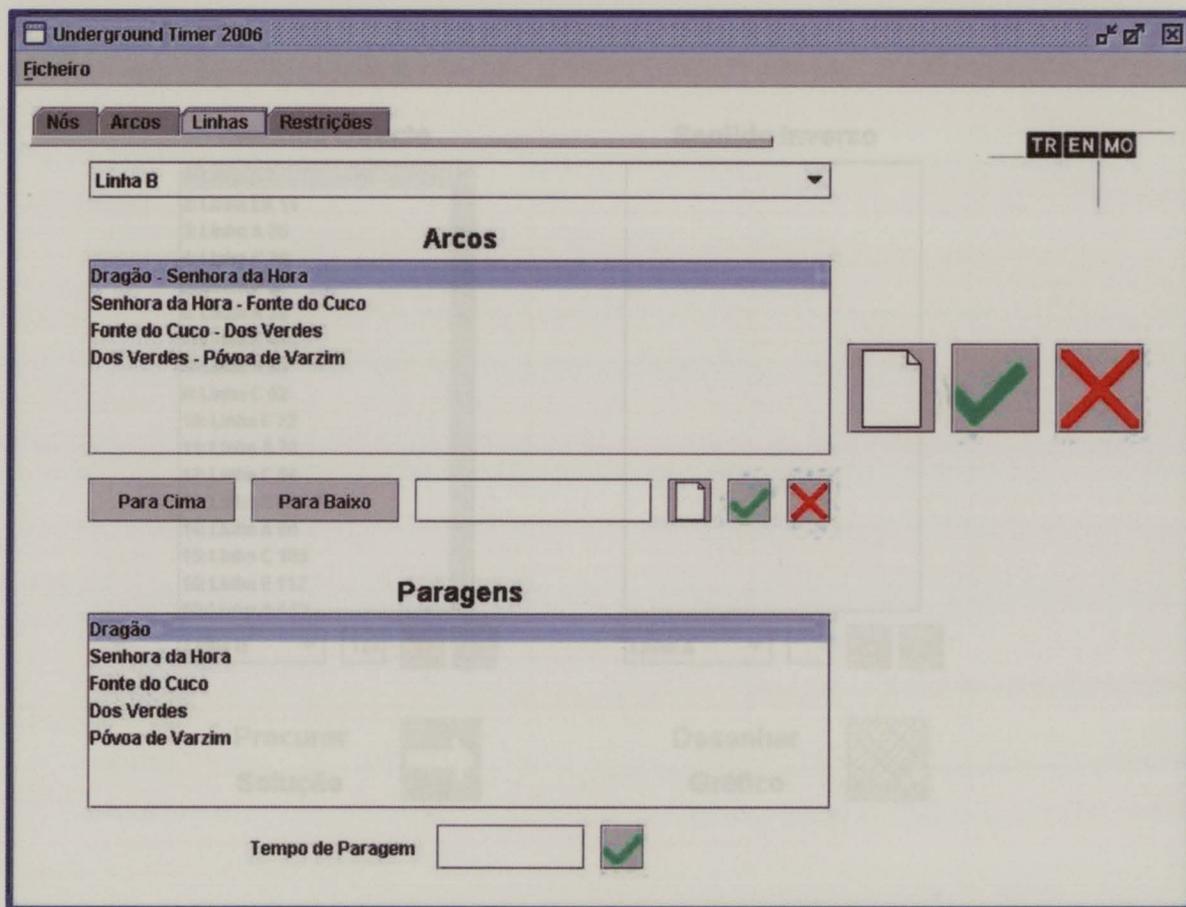


Figura 28 – Janela de Edição de Linhas

Na janela vista na Figura 28 podem ser editados os dados relativos a uma linha, que passam pela lista de paragens, o tempo de percurso entre cada uma e o tempo de paragem.

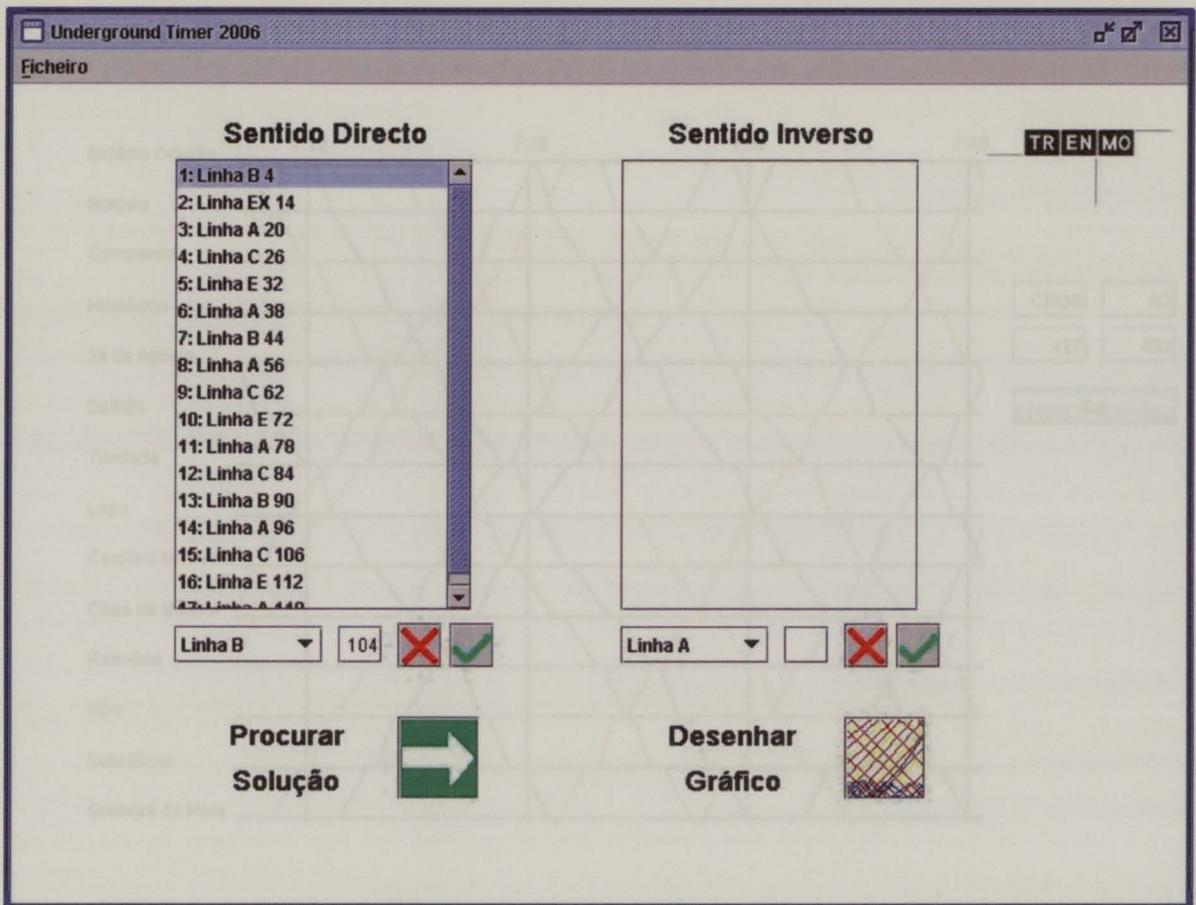


Figura 29 – Janela de Edição da solução

Na janela de descrição da solução (Figura 29), pode ser editada a solução já definida pelo software, assim como pode ser definida uma nova solução manualmente, ou até criar parte de uma solução manualmente para colocar o software a pesquisar o resto automaticamente.

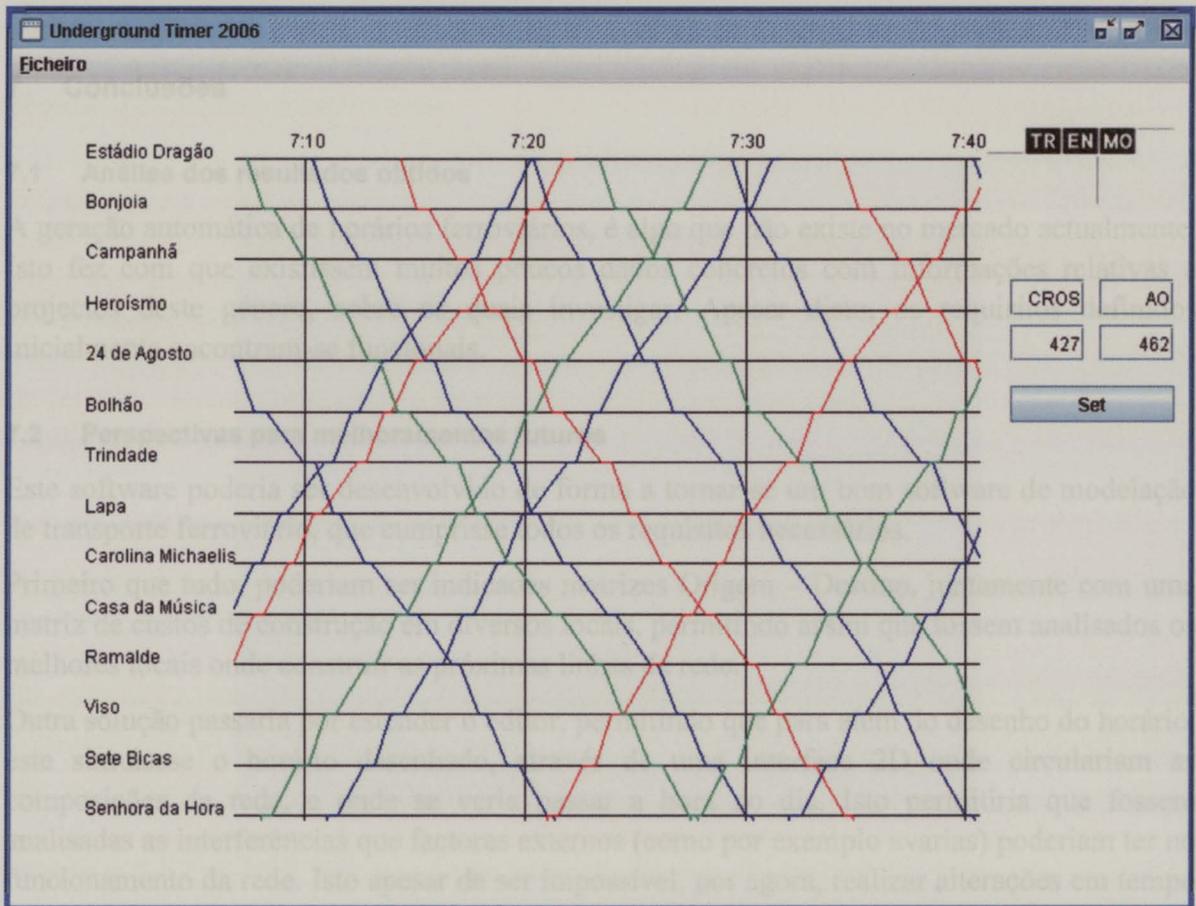


Figura 30 – Gráfico desenhado pelo software

O gráfico apresentado na Figura 30 é uma representação gráfica do horário definido pelo utilizador ou pelo software.

Devem ser indicados os códigos dos nós de início e de fim (CROS e AO), assim como o tempo mínimo e máximo a ser representado, em minutos a contar da meia – noite, neste caso 427 (7:07) e 462 (7:42).

Como solução pode ser tentado guardar para além do caminho percorrido, uma Linked List com o trajecto todo percorrido e evitar viagens já marcadas.

Outro algoritmo possível de implementar seria a definição de espaços coloridos, indicando qual seria o ponto ideal, onde colocar a solução actual, assim, cada posição da matriz de disponibilidades teria igualmente a importância que uma atribuição nesse ponto teria para a qualidade final da solução.

7.3 Principais dificuldades

Devido ao facto da instituição de estágio, estar longe de ser uma empresa no ramo da Engenharia e produção de Software, o que fez com que o estagiário fosse o único membro da empresa com formação na área da informática, todas as dúvidas, por mínimas que fossem, se

7 Conclusões

7.1 Análise dos resultados obtidos

A geração automática de horários ferroviários, é algo que não existe no mercado actualmente. Isto fez com que existissem muitos poucos dados concretos com informações relativas a projectos deste género, sobre os quais investigar. Apesar disto, os requisitos definidos inicialmente encontram-se funcionais.

7.2 Perspectivas para melhoramentos futuros

Este software poderia ser desenvolvido de forma a tornar-se um bom software de modelação de transporte ferroviário, que cumprisse todos os requisitos necessários.

Primeiro que tudo, poderiam ser indicadas matrizes Origem – Destino, juntamente com uma matriz de custos de construção em diversos locais, permitindo assim que fossem analisados os melhores locais onde construir as próximas linhas da rede.

Outra solução passaria por estender o editor, permitindo que para além do desenho do horário, este simulasse o horário desenhado, através de uma interface 2D onde circulariam as composições da rede, e onde se veria passar a hora do dia. Isto permitiria que fossem analisadas as interferências que factores externos (como por exemplo avarias) poderiam ter no funcionamento da rede. Isto apesar de ser impossível, por agora, realizar alterações em tempo real sobre as operações, seria possível planear o horário de forma a que as avarias tivessem o mínimo de influências possível.

Existem igualmente diversos algoritmos, possíveis de implementar que melhorariam a eficiência. Um destes algoritmos é o seguinte: se se atribuiu a viagem 1-70 (linha 1, slot 70), a melhor solução a partir desta será e por ordem 2-93, 1-95, 1-96, 1-97, etc. etc. Ora, se chegarmos à conclusão que o caminho percorrido a partir de 1-70 não tem saída, e atribuirmos por exemplo 1-71, vão ser novamente testadas 2-93, 1-95, 1-96, 1-97 pela mesma ordem o que provavelmente não vai dar a nenhum caminho novo.

Como solução pode ser tentado guardar para além do caminho percorrido, uma Linked List com o trajecto todo percorrido e evitar viagens já marcadas.

Outro algoritmo possível de implementar seria a definição de espaços coloridos, indicando qual seria o ponto ideal, onde colocar a solução actual, assim, cada posição da matriz de disponibilidades teria igualmente a importância que uma atribuição nesse ponto teria para a qualidade final da solução.

7.3 Principais dificuldades

Devido ao facto da instituição de estágio, estar longe de ser uma empresa no ramo da Engenharia e produção de Software, o que fez com que o estagiário fosse o único membro da empresa com formação na área da informática, todas as dúvidas, por mínimas que fossem, se

pudessem tornar em grandes problemas devido ao facto de não existir na instituição de estágio nenhum membro com formação nesta área que pudesse esclarecer a grande maioria das dúvidas e dificuldades encontradas.

Outro das dificuldades prendeu-se com a especificação de requisitos. Isto porque, a instituição de estágio não está envolvida directamente com as operações do metro do Porto (que estão a cargo de uma operadora privada), mas sim com a Empresa Metro do Porto, que em muitos casos não tinha um conhecimento aprofundado sobre as operações que eram realizadas sobre a sua própria rede. Assim sendo, surgiram alguns conflitos entre os requisitos definidos pela Empresa Metro do Porto e a operador. Após a reunião tida com a operadora estes requisitos foram esclarecidos, no entanto a informação completa apenas chegou à instituição de estágio a meio do mês de Julho.

7.4 Conclusões pessoais

Em substituição do habitual projecto de fim de curso, existente noutras licenciaturas, a LEIC proporciona aos seus finalistas um estágio curricular, estágio este que visa sobretudo dotar os finalistas de um forte sentido de responsabilidade, e sobretudo, lançar os licenciados no mercado de trabalho, integrando-os desde logo no ambiente empresarial, antes ainda de terem terminado efectivamente a licenciatura.

No meu caso, esta integração resultou ainda mais do que o esperado, pois a minha área de influência na instituição de estágio não se limitou ao projecto de estágio aqui descrito, mas abrangeu cerca de 80% dos projectos existentes na referida instituição. Este facto, permitiu-me integrar todo o processo de funcionamento de uma empresa de forma mais fácil do que teria sido a minha presença na instituição se cingisse a realização do projecto de estágio.

No entanto, não poderia deixar aqui de agradecer às pessoas na instituição com as quais aprendi muito de diversas áreas relacionadas com a área de negócio da instituição, o que me dotou de outros conhecimentos técnicos não limitados à licenciatura que agora termino. A essas pessoas, que desinteressadamente sempre me apoiaram, espero ter igualmente prestado o apoio que estas tenham necessitado.

Referências e Bibliografia

WEB – Ligação entre dois nós, segmento de linha.

<http://java.sun.com/j2se/1.3/docs/api>, documentação da API da linguagem Java.

<http://www.wikipedia.org>, informação diversa.

<http://www.ai.univie.ac.at/clpqr>, informação relativa a programação em lógica com restrições.

<http://www.trenmo.com>, informação relativa à instituição de estágio.

Bibliografia

Train Time Table and Route Generation Using a Constraint-Based Approach, Hon Wai Chun, Department of Electronic Engineering, City University of Hong Kong, Junho 1997.

Periodic Timetable Generation, Carole Gieseemann, University of Constance, 2002.

Data Structures And Problem Solving using Java, Mark Allen Weiss, 1998.

Ligação – Ligação virtual entre duas linhas, através do transbordo directo.

Nó – Ponto da rede, ao qual se ligam arcos, tipicamente uma estação.

Linha – Serviço regular de transporte público de passageiros (com frequência e tempos de viagem definidos), num itinerário fixo.

Tronco Comum – Troço existente entre as estações da Senhora da Hora e do Estádio do Dragão, é designado de tronco comum, porque é um tronco que é comum a quase todas as linhas da rede de Metro do Porto, linhas estas que derivam posteriormente para os respectivos terminos.

ANEXO A: Glossário

Arco – Ligação entre dois nós, segmento de linha.

Cruzamentos – locais onde uma linha atravessa na sua trajectória, a trajectória de outra linha. Não sendo desnivelados, são potencialmente geradores de conflitos, pois entre a passagem de duas composições pelo cruzamento é necessário deixar um relativo intervalo.

Frequência – Número de vezes que um determinado evento ocorre por unidade de tempo. ~ de passagem: número de circulações que passam no mesmo troço numa unidade de tempo (geralmente uma hora). ~ de funcionamento: neste software define-se como o número de soluções que o software consegue analisar por unidade de tempo (tipicamente 1 segundo).

Intervalo de Passagem – É o inverso da frequência, é o tempo que decorre entre a passagem de duas circulações da mesma linha.

Ligação – Ligação virtual entre duas linhas, através do transbordo directo

Nó – Ponto da rede, ao qual se ligam arcos, tipicamente uma estação.

Linha – Serviço regular de transporte público de passageiros (com frequência e tempos de viagem definidos), num itinerário fixo.

Tronco Comum – Troço existente entre as estações da Senhora da Hora e do Estádio do Dragão, é designado de tronco comum, porque é um tronco que é comum a quase todas as linhas da rede de Metro do Porto, linhas estas que derivam posteriormente para os respectivos terminos.

ANEXO B: Explicação da fórmula de cálculo do número de hipóteses

Neste anexo serão apresentadas as soluções encontradas pelo software para os cenários analisados. Estas referem o número de hipóteses analisadas, e a avaliação calculada segundo a fórmula:

$$\frac{A_{18}^{120}}{6 \times 4 \times 4 \times 3 \times 1!} = 2.7962 \times 10^{30}$$

Cenário C1 (6A + 4B + 4C)

Existem para este problema aproximadamente, 2.796.200.000.000.000.000.000.000.000. Este valor é obtido da seguinte forma, existem 18 circulações, às quais podem ser atribuídos 120 números diferentes.

Assim sendo temos:

- 120 hipóteses diferentes para a primeira circulação;
- 119 para a segunda, dado que a anterior já estava atribuída;
- 118 para a terceira

e por aí em diante.

O que corresponde a arranjos sem repetição de 120, 18 a 18.

No entanto esta solução faria com que circulações da mesma linha fossem consideradas como circulações independentes, logo é necessário ainda dividir pelas permutações de cada linha para que tenhamos o valor correcto.

Sentido SPM			Sentido EDR		
Lin	Hora de saída	Intervalo	Lin	Hora de saída	Intervalo
1	3:00		1	3:00	
2	16:00	13:00	2	16:00	13:00
3	32:00	16:00	3	32:00	16:00
4	48:00	14:00	4	47:00	16:00
5	63:00	17:00	5	63:00	16:00

Sentido SPM			Sentido EDR		
Lin	Hora de saída	Intervalo	Lin	Hora de saída	Intervalo
6	9:00		6	12:00	
7	22:00	16:00	7	25:00	14:00
8	36:00	14:00	8	42:00	16:00
9	50:00	16:00	9	57:00	16:00
10	60:00	14:00	10	72:00	15:00

Tabela 3 - Solução encontrada para o cenário C1



ANEXO C: Soluções Encontradas

Neste anexo serão apresentadas as soluções encontradas pelo software para os cenários encontrados. Estas referem o tempo de processamento, o número de hipóteses analisadas, e a avaliação calculada segundo os algoritmos descritos.

Cenário C1 (6A + 4B + 4C)

30 minutos – 65.138.734 hipóteses analisadas – 7,333 pts

Linha Azul – A					
Sentido SRM			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
A1a	0:00	.	A1b	0:00	.
A2a	9:00	09:00	A2b	9:00	09:00
A3a	19:00	10:00	A3b	19:00	10:00
A4a	29:00	10:00	A4b	29:00	10:00
A5a	39:00	10:00	A5b	39:00	10:00
A6a	49:00	10:00	A6b	50:00	11:00
A1a	60:00	11:00	A1b	60:00	10:00

Linha Vermelha – B					
Sentido PVZ			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
R11a	3:00	.	R11b	3:00	.
R12a	16:00	13:00	R12b	16:00	13:00
R13a	32:00	16:00	R13b	32:00	16:00
R14a	46:00	14:00	R13b	47:00	15:00
R11a	63:00	17:00	R11b	63:00	16:00

Linha Verde – C					
Sentido FRM			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
V1a	6:00	.	V1b	12:00	.
V2a	22:00	16:00	V2b	26:00	14:00
V3a	36:00	14:00	V3b	42:00	16:00
V4a	52:00	16:00	V4b	57:00	15:00
V1a	66:00	14:00	V1b	72:00	15:00

Tabela 3 - Solução encontrada para o cenário C1

Cenário C3 (6A + 3B + 5C + 2EX)

1 hora – 96.907.427 hipóteses analisadas – 10,967 pts

Linha Azul – A					
Sentido SRM			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
A1a	0:00	.	A1b	0:00	.
A2a	9:00	09:00	A2b	9:00	09:00
A3a	19:30	10:30	A3b	19:00	10:00
A4a	29:30	10:00	A4b	29:00	10:00
A5a	40:00	10:30	A5b	39:00	10:00
A6a	50:00	10:00	A6b	49:30	10:30
A1a	60:00	10:00	A1b	60:00	10:30

Linha Vermelha – B					
Sentido PVZ			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
R11a	3:00	.	R11b	3:00	.
R12a	23:00	20:00	R12b	22:00	19:00
R13a	43:00	20:00	R13b	42:00	20:00
R11a	63:00	20:00	R11b	63:00	21:00

Linha Verde – C					
Sentido FRM			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
V1a	6:00	.	V1b	12:00	.
V2a	14:00	08:00	V2b	25:00	13:00
V3a	26:00	12:00	V3b	34:00	09:00
V4a	37:00	11:00	V4b	45:00	11:00
V5a	53:00	16:00	V4b	57:00	12:00
V1a	66:00	13:00	V1b	72:00	15:00

Linha Expresso – EX					
Sentido PVZ			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
R21a	34:00	.	R21b	15:30	.
R22a	57:00	23:00	R22b	53:00	37:30
R21a	94:00	37:00	R21b	75:30	22:30

Tabela 4 – Solução encontrada para o cenário C3

Cenário C4 (6A + 3B + 5C + 2EX)

30 minutos – 44.524.391 hipóteses analisadas – 3,267pts

Linha Azul – A					
Sentido SRM			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
A1a	0:00	.	A1b	0:00	.
A2a	9:00	09:00	A2b	9:00	09:00
A3a	18:00	09:00	A3b	19:00	10:00
A4a	28:00	10:00	A4b	29:00	10:00
A5a	39:00	11:00	A5b	39:00	10:00
A6a	49:30	10:30	A6b	49:00	10:00
A1a	60:00	10:30	A1b	60:00	11:00

Linha Vermelha – B					
Sentido PVZ			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
R11a	3:00	.	R11b	3:00	.
R12a	12:30	09:30	R12b	22:00	19:00
R13a	31:30	19:00	R13b	34:30	12:30
R14a	46:30	15:00	R14b	52:00	17:30
R11a	63:00	16:30	R11b	63:00	11:00

Linha Verde – C					
Sentido FRM			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
V1a	6:00	.	V1b	12:00	.
V2a	21:00	15:00	V2b	26:00	14:00
V3a	36:00	15:00	V3b	42:00	16:00
V4a	52:30	16:30	V4b	57:00	15:00
V1a	66:00	13:30	V1b	72:00	15:00

Linha Expresso – EX					
Sentido PVZ			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
R21a	24:00	.	R21b	15:00	.
R22a	43:30	19:30	R22b	45:30	30:30
R21a	84:00	40:30	R21b	75:00	29:30

Tabela 5 – Solução encontrada para o cenário C4

Cenário C5 (6A + 4B + 5C + 1EX)

30 minutos – 42.392.091 hipóteses analisadas – 11,500 pts

Linha Azul – A					
Sentido SRM			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
A1a	0:00		A1b	0:00	
A2a	9:00	09:00	A2b	10:00	10:00
A3a	20:00	11:00	A3b	20:00	10:00
A4a	30:00	10:00	A4b	29:00	09:00
A5a	40:00	10:00	A5b	39:00	10:00
A6a	50:00	10:00	A6b	50:00	11:00
A1a	60:00	10:00	A1b	60:00	10:00

Linha Vermelha – B					
Sentido PVZ			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
R11a	3:00		R11b	3:00	
R12a	17:00	14:00	R12b	17:00	14:00
R13a	33:00	16:00	R13b	32:00	15:00
R14a	47:00	14:00	R14b	47:00	15:00
R11a	63:00	16:00	R11b	63:00	16:00

Linha Verde – C					
Sentido FRM			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
V1a	6:00		V1b	7:00	
V2a	14:00	08:00	V2b	23:00	16:00
V3a	25:00	11:00	V3b	35:00	12:00
V4a	37:00	12:00	V4b	43:30	08:30
V5a	53:00	16:00	V5b	55:00	11:30
V1a	66:00	13:00	V1b	67:00	12:00

Linha Expresso – EX					
Sentido PVZ			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
R21a	44:00		R21b	14:00	
R21a	104:00	60:00	R21b	74:00	60:00

Tabela 6 – Solução encontrada para o cenário C5

Linha Expresso – EX					
Sentido PVZ			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
R21a	5:00		R21b	15:00	
R21a	65:00	60:00	R21b	75:00	60:00

Tabela 7 – Solução encontrada para o cenário C7

Cenário C7 (6A + 3B + 4C + 3E + 1EX) com ligação entre linhas A e E
 15 horas, 1.256.415.759 hipóteses analisadas, 19,333 pts

Linha Azul – A					
Sentido SRM			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
A1a	8:00	.	A1b	9:00	.
A2a	19:00	11:00	A2b	18:00	09:00
A3a	28:00	09:00	A3b	27:00	09:00
A4a	39:00	11:00	A4b	38:00	11:00
A5a	48:00	09:00	A5b	49:00	11:00
A6a	59:00	11:00	A6b	59:00	10:00
A1a	68:00	09:00	A1b	69:00	10:00

Linha Vermelha – B					
Sentido PVZ			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
R11a	2:00	.	R11b	6:00	.
R12a	22:00	20:00	R12b	30:00	24:00
R13a	45:00	23:00	R13b	43:00	13:00
R11a	62:00	17:00	R11b	66:00	23:00

Linha Verde – C					
Sentido FRM			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
V1a	13:00	.	V1b	2:00	.
V2a	31:00	18:00	V2b	22:00	20:00
V3a	42:00	11:00	V3b	35:00	13:00
V4a	53:00	11:00	V4b	46:00	11:00
V1a	73:00	20:00	V1b	62:00	16:00

Linha Roxa – E					
Sentido Aeroporto			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
M1a	16:00	.	M1b	12:00	.
M2a	36:00	20:00	M2b	32:00	20:00
M3a	56:00	20:00	M3b	52:00	20:00
M1a	76:00	20:00	M1b	72:00	20:00

Linha Expresso – EX					
Sentido PVZ			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
R21a	5:00	.	R21b	15:00	.
R21a	65:00	60:00	R21b	75:00	60:00

Tabela 7 – Solução encontrada para o cenário C7

Cenário 6A 4B 4C 3E 2EX 30 minutos, 41.803.538 hipóteses analisadas, 14.883 pts

Cenário C9 (6A+3B+4C+3E+2EX) 30 minutos, 41.803.538 hipóteses analisadas, 14,883 pts

Linha Azul – A					
Sentido SRM			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
A1a	0:00	.	A1b	0:00	.
A2a	9:00	09:00	A2b	9:00	09:00
A3a	18:00	09:00	A3b	19:00	10:00
A4a	28:00	10:00	A4b	29:00	10:00
A5a	39:00	11:00	A5b	39:00	10:00
A6a	49:00	10:00	A6b	49:00	10:00
A1a	60:00	11:00	A1b	60:00	11:00

Linha Vermelha – B					
Sentido PVZ			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
R11a	3:00	.	R11b	3:00	.
R12a	21:00	18:00	R12b	22:30	19:30
R13a	45:00	24:00	R13b	45:00	22:30
R11a	63:00	18:00	R11b	63:00	18:00

Linha Verde – C					
Sentido FRM			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
V1a	6:00	.	V1b	12:00	.
V2a	24:00	18:00	V2b	26:00	14:00
V3a	36:00	12:00	V3b	42:00	16:00
V4a	52:00	16:00	V4b	57:00	15:00
V1a	66:00	14:00	V1b	72:00	15:00

Linha Roxa – E					
Sentido Aeroporto			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
M1a	12:00	.	M1b	6:00	.
M2a	32:30	20:30	M2b	32:00	26:00
M3a	55:00	22:30	M3b	52:00	20:00
M1a	72:00	17:00	M1b	66:00	14:00

Linha Expresso – EX					
Sentido PVZ			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
R21a	15:00	.	R21b	15:00	.
R22a	42:00	27:00	R22b	36:00	21:00
R21a	75:00	33:00	R21b	75:00	39:00

Tabela 8 – Solução encontrada para o cenário C9

Cenário 6A 4B 4C 3E 1EX 30 minutos, 41.803.538 hipóteses analisadas, 14,883 pts

Linha Azul – A					
Sentido SRM			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
A1a	0:00	.	A1b	0:00	.
A2a	9:00	09:00	A2b	9:30	09:30
A3a	18:00	09:00	A3b	19:30	10:00
A4a	28:00	10:00	A4b	30:00	10:30
A5a	38:00	10:00	A5b	39:00	09:00
A6a	49:00	11:00	A6b	49:00	10:00
A1a	60:00	11:00	A1b	60:00	11:00

Linha Vermelha – B					
Sentido PVZ			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
R11a	3:00	.	R11b	3:00	.
R12a	21:00	18:00	R12b	16:30	13:30
R13a	35:00	14:00	R13b	36:00	19:30
R14a	46:00	11:00	R14b	52:00	16:00
R11a	63:00	17:00	R11b	63:00	11:00

Linha Verde – C					
Sentido FRM			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
V1a	6:00	.	V1b	13:00	.
V2a	24:00	18:00	V2b	27:00	14:00
V3a	41:00	17:00	V3b	42:00	15:00
V4a	55:00	14:00	V4b	57:00	15:00
V1a	66:00	11:00	V1b	72:00	16:00

Linha Roxa – E					
Sentido Aeroporto			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
M1a	12:00	.	M1b	6:00	.
M2a	32:00	20:00	M2b	24:00	18:00
M3a	52:00	20:00	M3b	45:00	21:00
M1a	72:00	20:00	M1b	66:00	21:00

Linha Expresso – EX					
Sentido PVZ			Sentido EDR		
Cod.	Hora de saída	Intervalo	Cod.	Hora de saída	Intervalo
R21a	15:00	.	R21b	33:00	.
R21a	75:00	60:00	R21b	93:00	60:00

Tabela 9 – Solução encontrada para o cenário 6A + 4B + 4C + 3E + 1EX



Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, s/n 4200-465 Porto PORTUGAL
www.fe.up.pt



FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

BIBLIOTECA



000088928