

# Procura e Planeamento

Campus da Alameda

IST @ 2018/2019

26 de Outubro de 2018

## 1. Introdução

Este projecto tem dois objectivos: (1) Desenvolver um programa em ANSI Common Lisp que resolva diferentes problemas de afectação de recursos. Devem ser utilizadas diferentes estratégias de procura estudadas na cadeira, e desenvolvidas heurísticas que permitam resolver este tipo de problemas da forma mais eficaz possível; (2) Produzir um estudo que avalie as alternativas implementadas tanto do ponto de vista quantitativo como qualitativo.

## 2. Descrição do problema de afectação em empresas de transporte

O problema consiste em obter uma afectação completa das tarefas de condução de veículos de transporte, a realizar pelos trabalhadores durante um dia de trabalho, a turnos de serviço, respeitando todas as restrições do problema.

Este é um problema de optimização pois o objectivo é minimizar uma função de custo. A função de custo a utilizar é a seguinte:

$$CustoTotal = \sum_{i=1}^n C(i)$$

em que o  $C(i)$  é o custo de cada um dos turnos de serviço que constituem a afectação completa. Este custo representa o tempo de trabalho e a sua granularidade é o minuto.

Uma tarefa é definida pelo tuplo:

(<Local de partida>, <Local de chegada>, <Instante de partida>, <Instante de chegada>)

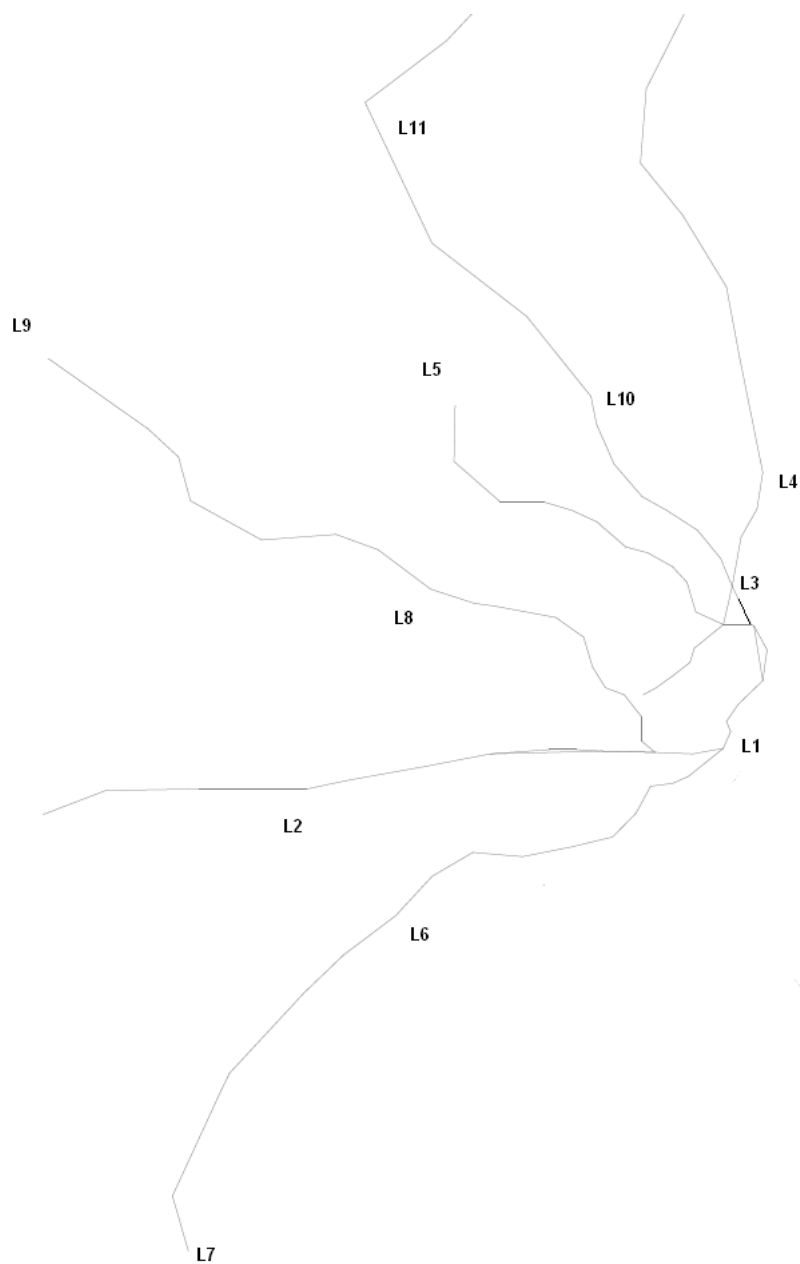
Na modelação do problema todas as durações devem ser apresentadas em minutos. Também os instantes devem ser apresentados em minutos, referindo-se o valor ao número de minutos que decorreram desde o início do dia de trabalho – note-se que um turno pode terminar depois do fim do dia de trabalho sendo nesse caso o instante final representado por um inteiro maior que 1440.

Neste problema uma das restrições mais importantes é a necessidade de garantir a continuidade espacial das tarefas num mesmo turno de serviço. Para garantir esta continuidade pode ser necessário introduzir viagens sem serviço que desloquem um trabalhador de um ponto da rede para outro. Vamos assumir que é possível deslocar um trabalhador em qualquer altura do dia e que o tempo necessário para o fazer é sempre de 40 minutos.

A afectação de tarefas a turnos de serviço tem que ser feita respeitando as seguintes restrições:

1. Todos os turnos devem começar no local L1;
2. A duração máxima de um turno de serviço é de 8:00;
3. A duração de um turno de serviço é calculada do início do serviço até ao fim do serviço. Se a duração for menos de 6 horas, então conta como se tivesse 6 horas;
4. Os turnos de serviço devem ter no máximo uma pausa para refeição;
5. Uma pausa de refeição deve ter a duração de 40 minutos;
6. O trabalhador não pode tomar refeição enquanto está a ser transportado;
7. O tempo de condução antes de ser necessário uma pausa de refeição não pode exceder as 4:00;
8. O tempo de condução inclui potenciais espaços entre tarefas que não sejam tomadas de refeição, isto é conta-se todo o tempo desde o início da primeira tarefa do bloco de tarefas até ao fim da última tarefa do bloco de tarefas. Nesta contabilização, no caso de a primeira tarefa do bloco ser a primeira tarefa do turno e o seu local de início não ser L1, deve-se incluir o tempo de deslocar o trabalhador de L1 até esse local (é necessário usar o mesmo raciocínio para a última tarefa de um turno e o seu local de fim).

O mapa da rede viária onde se realizam os serviços de transporte é apresentado em baixo na **Figura 1**, a sua apresentação serve apenas como ilustração dos dados dos problemas de testes, não possuindo, no entanto, qualquer informação relevante para a modelação do problema.



**Figura 1:** Mapa da rede viária onde se realizam os serviços de transporte.

### **3. Resolução do problema**

A obtenção de uma solução válida para o problema de afectação descrito é trivial visto que não é imposto qualquer limite para o número de turnos de serviço. Já a obtenção de uma boa solução ou mesmo da solução óptima, o objectivo deste trabalho, não é de todo trivial.

Deve ser escolhida uma formulação adequada que permita resolver qualquer problema da forma mais eficiente possível. Os problemas fornecidos na página da cadeira são apenas exemplos dos problemas que vão ser usados no teste das soluções produzidas.

## 4. Implementação

Parte da avaliação do trabalho desenvolvido vai decorrer automaticamente, pelo que é essencial que a especificação da interface seja seguida rigorosamente.

O conjunto de funções que implementa o *solucionador* deve ser definido num único ficheiro, este deve poder ser compilado sem erros nem avisos. Uma dos avisos produzidos pelos compiladores é a falta de uma declaração de package num ficheiro de código, de forma a evitar este aviso deve ser incluída a seguinte forma Lisp:

```
(in-package :user)
```

Esta forma deve ser a primeira forma do ficheiro.

Refira-se ainda que os testes a realizar automaticamente impõem alguns limites espaciais e temporais considerados razoáveis: (1) um limite à heap de 256 MBytes e (2) um tempo máximo por problema de afectação de 5 minutos num Core i5 a 2,6 GHz, ou seja a eficiência temporal e espacial das soluções apresentadas é relevante, tal como já tinha sido afirmado anteriormente.

### 4.1. Interface

Deve ser implementada a função *faz-afectacao*, que produz uma solução para um dado problema de afectação.

Esta função recebe como argumentos: um problema na sua representação externa definida em baixo e a estratégia de procura a usar. A função deve devolver a solução do problema usando a representação externa de uma solução, também definida em baixo.

As estratégias a suportar são:

```
"melhor.abordagem"  
"a*.melhor.heuristica"  
"a*.melhor.heuristica.alternativa"  
"sondagem.iterativa"  
"ILDS"  
"abordagem.alternativa"
```

A representação externa de um problema de afectação é uma lista de listas, em que cada uma das sub-listas representa uma tarefa que é necessário afectar a um turno.

Na *Figura 2*, representa-se uma tarefa que parte de L1 no instante 666 e chega a L10 no instante 733.

(L1 L10 666 733)

**Figura 2:** Exemplo de uma tarefa.

Na *Figura 3* representa-se a invocação a *faz-afectacao*, com um problema para o qual se quer obter um solução. Na *Figura 4* representa-se a solução obtida.

```
(faz-afectação '((L2 L1 1 25) (L1 L2 34 60) (L5 L1 408 447) (L1 L1 448  
551) (L1 L1 474 565)) "melhor.abordagem")
```

**Figura 3:** Exemplo de um problema de afectação.

A representação externa de uma solução é uma lista de listas, em que cada uma das sub-listas representa um dos turnos de serviço da solução. Por sua vez cada turno de serviço é uma lista de tarefas.

```
(( (L2 L1 1 25) (L1 L2 34 60)) ((L5 L1 408 447) (L1 L1 448 551)) ((L1 L1 474  
565)))
```

**Figura 4:** Solução do problema de afectação da Figura 3.

A estratégia "melhor.abordagem" deve usar a modelação, estratégia de procura, heurística e estratégia de corte de sucessores que produz os melhores resultados de entre tudo o que foi implementado.

## 4.2. Código Lisp fornecido

Para a realização deste projecto devem ser utilizados os algoritmos de procura, desenvolvidos em ANSI Common Lisp, disponíveis no *site* da cadeira. O código fornecido encontra-se no ficheiro *'procura.lisp'*. Este contém a implementação de vários algoritmos de procura. O funcionamento dos mesmos encontra-se descrito em *'procura.txt'*. O mais importante é compreender para que servem, não perceber exactamente como é que as funções estão definidas. Este ficheiro não deve ser alterado, se houver necessidade de alterar definições incluídas neste ficheiro estas devem ser feitas no ficheiro de código desenvolvido que contém a implementação realizada pelo grupo.

## 5. Estudo a desenvolver

A abordagem a desenvolver deve modelar o problema como uma procura num espaço de estados usando uma abordagem incremental e uma perspectiva de optimização. Ou seja, deve devolver a melhor solução possível de acordo com o definido anteriormente.

- Devem ser testadas todas as técnicas de procura fornecidas no ficheiro *"procura.lisp"*.
- Deve ser implementada e testada a estratégia de sondagem iterativa.
- Deve ser implementada e testada a estratégia de discrepância ILDS com as necessárias adaptações.
- Deve ser implementada pelo menos uma abordagem alternativa diferente das listadas acima.
- Adicionalmente, novas estratégias que facilitem a resolução dos problemas devem ser desenvolvidas. Estas podem incluir variações aos algoritmos básicos de procura, novos algoritmos de procura, estratégias híbridas, macro-operadores, sub-objectivos, etc. A análise do problema e dos primeiros resultados obtidos deverá fornecer indicadores quanto a abordagens válidas.

Note-se que as estratégias que visam a optimização devem usar todo o tempo disponível para devolverem a melhor solução que conseguirem encontrar nesse período.

Para as técnicas de procura informadas, são necessárias heurísticas.

- Devem ser apresentadas pelo menos duas heurísticas de desempenho relevante. Essas heurísticas devem, tanto quanto possível, basear-se em ideias distintas (heurísticas que variam apenas em constantes, por exemplo, são consideradas a mesma heurística).

O desempenho das várias estratégias de procura utilizadas, bem como das heurísticas e das estratégias de corte desenvolvidas, deve ser discutido no relatório, fazendo uma comparação dos resultados obtidos para alguns dos problemas usados para teste. Toda esta avaliação deve ser baseada em resultados quantitativos nomeadamente: custo das soluções, tempo da procura, nós gerados, nós expandidos, factor médio de ramificação, profundidade atingida, etc.

A discussão deve limitar-se a problemas que salientem características ou limitações relevantes do trabalho desenvolvido. Os alunos são encorajados a criar novos problemas, para além dos fornecidos como exemplo, que sejam adequados para ilustrar aspectos interessantes da discussão.

*Todo o processo de decisão deve ser documentado no relatório a elaborar.*

O relatório do projecto deve incluir ainda:

- A descrição da(s) modelação(ões) do problema e as razões subjacentes às decisões tomadas;
- A descrição das estruturas de dados desenvolvidas e as razões subjacentes ao seu desenvolvimento;
- A descrição das heurísticas desenvolvidas e as razões subjacentes ao seu desenvolvimento;
- A descrição das estratégias de corte implementadas e as razões subjacentes ao seu desenvolvimento;
- As opções tomadas durante o desenvolvimento do projecto;



- Uma discussão sobre a forma de conseguir obter melhores resultados, indicando genericamente o tipo de heurísticas/estratégias a desenvolver para isso ou alterações à modelação.

Ainda quanto ao relatório, convém salientar que a legibilidade do mesmo (para a qual contribuem não só a organização dos capítulos/secções como a correcção ortográfica, a construção das frases e a fluidez do discurso) é de importância capital para a correcta compreensão do seu conteúdo. Devem evitar-se frases complexas (a utilização da voz passiva, de duplas negativas, etc.). Todas as figuras e gráficos apresentados devem ser legíveis e estar adequadamente legendados.

## **6. Critérios de avaliação**

Os factores mais importantes para a avaliação do projecto são (não necessariamente por esta ordem):

- Execução correcta e elegância do programa;
- Qualidade da(s) modelação(ões) do problema;
- Qualidade das heurísticas e estratégias desenvolvidas;
- Eficiência das estruturas de dados escolhidas para a representação do problema;
- Qualidade do estudo desenvolvido (inclui a descrição do trabalho desenvolvido, os resultados obtidos e conclusões);
- Apreciação global.

## **7. Inscrições e entrega do projecto**

Os elementos de cada grupo, que deverão ser no máximo 2, devem inscrever-se no Fénix.

Deve ser feita a entrega do relatório e do código desenvolvido.

A entrega do código desenvolvido deve ser feita até às 23h59 do dia 7 de Dezembro de 2018 da seguinte forma:

- o ficheiro com o programa deve ser entregue de forma electrónica usando o sistema Fénix. O nome do ficheiro Lisp a entregar deve ser "GXXX.lisp", em que "XXX" é o número do grupo, por exemplo "G007.lisp";

A entrega do relatório deve ser feita até às 23h59 do dia 7 de Dezembro de 2018 da seguinte forma:

- o ficheiro com o relatório em formato electrónico (ficheiro em formato “pdf”) deve ser entregue de forma electrónica usando o sistema Fénix. O nome do ficheiro a entregar deve ser "GXXX.pdf" ou "GXXX.doc", em que "XXX" é o número do grupo, por exemplo "G007.pdf";

Todo o material entregue pelos alunos (documentação e ficheiro com o código) deve conter a identificação do grupo (número do grupo, fornecido durante a inscrição do grupo) e o número e nome de cada elemento do grupo.

**Atenção:** Não são aceites entregas fora do prazo!

## **8. Novidades do projecto**

No caso de haver novidades relativas ao projecto, estas serão afixadas na página da cadeira pelo que esta página deve ser visitada diariamente.

## Procura e Planeamento

Campus da Alameda

Projecto (2018/2019)

Número do grupo (obtido a partir do sistema Fénix):

Nome:

Número:

Nome:

Número:

Classificação:

Soma das horas gastas exclusivamente para fazer este trabalho:

Limite para entrega: dia 23h59 do dia 7 de Dezembro de 2018.

---