

Trabalho Prático (2018/2019)

Objetivo

Formulação de problemas de satisfação de restrições em domínios finitos e problemas de otimização combinatória. Utilização de sistemas de Programação Lógica por Restrições para resolução desses problemas.

Tema

Estudo de **dois problemas** de calendarização de tarefas com alguma partilha de recursos (*scheduling problems*).

• Problema 1

Um projeto envolve um certo número de tarefas. É conhecida a duração de cada tarefa (dada por um número inteiro de dias) e a relação de precedência entre tarefas. Cada tarefa requer um certo número de trabalhadores (a tempo inteiro) e a contratação de um número de trabalhadores maior do que o necessário em nada altera a duração da mesma. Não há partilha de outros recursos entre tarefas (como, por exemplo, máquinas). Os trabalhadores contratados para o projeto estão habilitados a desempenhar qualquer tarefa. Determinar o número mínimo de trabalhadores a contratar de forma a concluir o projeto o mais cedo possível.

No estudo do problema, terão de ser analisadas as questões seguintes.

- Qual é a duração mínima do projeto?
 - Quantos trabalhadores são necessários se todas as tarefas tiverem início na sua data de início mais próxima?
 - Quantos trabalhadores são necessários no mínimo para realizar as tarefas críticas?
 - Qual o número mínimo de trabalhadores a contratar? Quando é que cada tarefa tem início?
- Existem soluções ótimas alternativas?

Para este problema, pretende-se o(s) modelo(s) matemático(s), uma implementação no sistema ECLiPSe que use programação por restrições, e a comparação com um programa desenvolvido para resolução do problema (em C/C++, Java ou Python).

• Problema 2

Um projeto envolve um certo número de tarefas. É conhecida a duração de cada tarefa (dada por um número inteiro de horas) e a relação de precedência entre tarefas. Cada tarefa requer uma equipa de trabalhadores de várias especialidades, sendo conhecido o número de trabalhadores necessários de cada especialidade. Cada trabalhador tem habilitações em uma ou mais especialidades. Se for escolhido para fazer parte da equipa que realizará uma certa tarefa, então nessa equipa só terá a seu cargo trabalho de uma especialidade.

Nenhuma tarefa pode ser interrompida.

~~Existem tarefas que podem ser interrompidas no fim de um período de trabalho e retomadas novamente no período seguinte. Existem outras que não podem ser interrompidas mas a sua duração permite que sejam realizadas num período. Se uma tarefa for interrompida então é~~

~~retomada no início do período seguinte, exatamente pela mesma equipa.~~

Podem existir tarefas que terão de começar dentro de um intervalo de tempo após a conclusão de uma outra tarefa de que dependem. O projeto deverá estar concluído até ao fim de um certo dia. O horário de trabalho é de 8 horas por dia, de segunda a sexta-feira, excepto feriados (das 8:00 às 12:00 e das 13:00 às 17:00). Na definição do início de cada tarefa, deverá ser considerado um calendário real.

~~Preferencialmente, nenhum trabalhador deverá estar mais do que um certo número de horas desocupado por semana, sendo tal número definido por uma percentagem do seu horário semanal. Considera-se que o trabalhador está ocupado num certo instante se fizer parte de uma equipa que está a realizar uma tarefa nesse instante.~~

Se não for possível concluir o projeto dentro do prazo previsto, terá de ser determinado o número mínimo de trabalhadores a contratar (se desse modo se puder concluir o projeto no prazo). Nesse caso, assumir-se-á que cada um dos novos trabalhadores só tem habilitações para uma especialidade. Pode ser útil também minimizar a duração do projeto, se for possível concluí-lo no prazo sem contratar mais trabalhadores.

Desenvolvimento e Prazos de entrega

O trabalho será realizado em **grupo**, preferencialmente de **três** elementos.

A constituição dos grupos deve ser indicada até 26 de Abril. Os trabalhos deverão ser desenvolvidos até ~~20 de Maio~~. **23 de Maio (23:00)** ~~Não haverá alteração das datas de entrega.~~ Os trabalhos serão submetidos por e-mail (com assunto "TRABALHO - MAD 1819").

Para cada problema, pretende-se:

- Definição dos **modelo(s) matemático(s)**. As variáveis de decisão serão inteiras ou booleanas. As restrições podem ser lineares ou não.
- **Estudo computacional do problema no sistema ECLiPSe** (<http://eclipseclp.org/>). Apresentação de resultados para algumas instâncias. Para o problema 1, estudo comparativo com resolutor específico implementado em C/C++/Java/Python.
- Um **relatório** que descreva o problema, o(s) modelo(s) matemático(s), a implementação, os resultados obtidos, as alternativas analisadas (nomeadamente, para tentar tratar instâncias de dimensão maior), o comportamento do programa para vários exemplos de diferentes complexidades e, se aplicável, indique sugestões sobre possíveis melhoramentos. Não esquecer as referências bibliográficas que consultar.

Critérios de Avaliação

A classificação será a média ponderada das notas obtidas na resolução do Problema 1 e do Problema 2, com pesos de 70% e 30% respetivamente. Os relatórios e a apresentação oral dos trabalhos são fundamentais para a sua avaliação.

Espera-se que todos os elementos do grupo dêem o seu contributo para as diferentes fases do trabalho (especificação do modelo, implementação do modelo e relatório). O desconhecimento que algum elemento demonstre poderá penalizar a sua classificação e a do grupo. Serão igualmente penalizadas situações de plágio. A experiência adquirida pela elaboração do trabalho prático será também aferida no segundo teste escrito e nos exames finais.

Formato para os dados

Os dados necessários são definidos por predicados, num ficheiro.

Problema 1

`tarefa(Idt, ListaIdts, Duracao, Ntrabs)`

Problema 2

~~`tarefa(Idt, ListaIdts, Duracao, Reqtrabalhadores, PodeInterromper)`~~
 `tarefa(Idt, ListaIdts, Duracao, Reqtrabalhadores, 0)`
 `trabalhador(Codigo, ListaEspecialidades)`
 `intervalo(Idt, Idtprev, Min, Max)`
 ~~`taxadesocupacao(Taxa)`~~
 `prazo(Data)`
 `calendario(ListaOrdenadaDiasUteis)`

onde

- `Idt` é um inteiro que identifica a tarefa;
- `ListaIdts` é uma lista de inteiros que são os identificadores de tarefas que só podem ter início depois de a tarefa `Idt` estar concluída;
- `Duracao` é um inteiro e representa a duração da tarefa (em dias, no Problema 1, e em horas, no Problema 2);
- `Ntrabs` é um inteiro e representa o número de trabalhadores necessários;
- `Codigo` é um inteiro e identifica o trabalhador, e `ListaEspecialidades` é a lista de especialidades para as quais tem habilitações;
- `Reqtrabalhadores` lista de termos da forma `r(E, Ntrabs)` que definem o número de trabalhadores da especialidade `E` necessários para a realização da tarefa;
- `PodeInterromper` tem valor 1 ou 0 consoante a tarefa possa ou não possa ser interrompida;
- `Idtprev` identifica uma tarefa que precede `Idt` e condiciona o início de `Idt` a no mínimo `Min` (máximo `Max`) horas depois da conclusão de `Idtprev`;
- `Taxa` é um inteiro entre 0 e 100 e deve ser entendido como uma percentagem (limite superior para a taxa de desocupação semanal);
- `Data` define o dia limite para conclusão do projeto;
- `ListaOrdenadaDiasUteis` é o calendário de dias úteis;
- Cada data é um termo `d(DiaMes, Mes, DiaSemana)`, por exemplo, `d(20, 5, 2)`, 20 de maio, segunda-feira.