



Faculdade de Engenharia da Universidade do Porto  
Mestrado Integrado em Engenharia Informática e Computação

## Programação em Lógica

2012/2013

MIEIC

(3º Ano)

1º Sem

Docentes: Henrique Lopes Cardoso, Daniel C. Moura, Mário A. Alves, Tiago Fernandes

### TRABALHO PRÁTICO Nº 2

## Tema

**Resolução de Problema de Decisão/Otimização usando Programação em Lógica com Restrições**

## Descrição

**Objectivo:** O objectivo deste trabalho é a construção de um programa em Programação em Lógica com Restrições para a resolução de um dos problemas de optimização ou decisão combinatoria sugeridos neste enunciado. Adicionalmente, deverá ser elaborado um artigo descrevendo o trabalho realizado e os resultados obtidos.

**Sistema de Desenvolvimento:** O sistema de desenvolvimento recomendado é o SICStus Prolog, que inclui um módulo de resolução de restrições sobre domínios finitos clp(FD).

## Condições de Realização

**Grupos:** Os grupos serão os mesmos do primeiro trabalho a menos que haja algum factor impeditivo. Nesse caso, os estudantes deverão entrar em contacto com o docente das aulas teórico-práticas.

### Datas Importantes:

A partir de 16/11/2012	Escolha do enunciado no <i>Moodle</i> e posterior envio de <i>email</i> para <a href="mailto:hlc@fe.up.pt">hlc@fe.up.pt</a> com a constituição do grupo (caso seja o mesmo do TP1 basta indicar o número do grupo) e com a escolha concretizada no <i>Moodle</i> . O <u>assunto</u> do <i>email</i> deverá ser da forma <b>PLOG 2012 TP2 Turma #TURMA</b>
6/12/2012, até às 12h00	(Opcional) Submissão, via <i>Moodle</i> , de versão preliminar do artigo (em formato Word ou PDF), com vista à obtenção de feedback do docente das aulas teórico-práticas. Esta submissão é opcional e não tem efeito na avaliação, sendo o seu objectivo alertar os grupos para eventuais falhas no artigo que possam comprometer a sua avaliação final. O nome do ficheiro deve ser da forma <b>PLOG_TP2_Pre_G#Grupo_T#Trabalho.DOC</b> . (Exemplo: PLOG_TP2_Pre_G60_T2.DOC)
9/12/2012	Entrega, via <i>Moodle</i> , da versão final do artigo (formato PDF) e do código fonte desenvolvido. Submeter um único ficheiro ZIP com nome da forma <b>PLOG_TP2_FINAL_G#Grupo_T#Trabalho.ZIP</b> . (Exemplo: PLOG_TP2_FINAL_G60_T2.ZIP)
10-14/12/2012	Demonstrações dos trabalhos nas aulas teórico-práticas.

**Valorização da Avaliação:** Ver ficha da Unidade Curricular no SiFEUP.

**Esclarecimentos:** Esclarecimentos sobre o enunciado e a realização do trabalho devem ser obtidos, preferencialmente, junto dos docentes no decorrer das aulas teórico-práticas.

# Artigo

Cada grupo deve elaborar e entregar um artigo e realizar uma demonstração da aplicação desenvolvida. O artigo deverá ser escrito em Português ou Inglês no formato LNCS (Lecture Notes in Computer Science) da Springer. Para tal, a Springer disponibiliza no seu sítio templates em Word e LaTeX (<http://www.springer.com/computer/lncs?SGWID=0-164-6-793341-0>), que deverão ser utilizados para a escrita do artigo. O artigo deverá ter entre 6 a 8 páginas (excluindo anexos). Sugere-se que contenha as seguintes partes:

**Título:** Título que transpareça o trabalho realizado (não tem que ser apenas o tema do trabalho).

**Autores e Afiliações:** Elementos de identificação do trabalho e do grupo. Nas afiliações colocar a turma e o grupo segundo o seguinte exemplo: FEUP-PLOG, Turma 3MIEIC9, Grupo 901.

**Resumo/Abstract:** Deve contextualizar e resumir o trabalho, salientando o objectivo, o método utilizado e fazendo referência aos principais resultados e à principal conclusão que esses resultados permitem obter.

**1. Introdução/Introduction:** Descrição dos objectivos e motivação do trabalho, descrição do problema, (idealmente, referência a outros trabalhos na mesma área + descrição muito sucinta da aproximação utilizada, salientando as principais diferenças em relação aos outros trabalhos), e estrutura do resto do artigo.

**2. Descrição do Problema/Problem Description:** Descrever sucintamente o problema de optimização ou decisão em análise.

**3. Ficheiros de Dados/Data Files:** Descrever os problemas a resolver e o conteúdo dos respectivos ficheiros de dados. Devem ser construídos pelo menos três problemas distintos em ficheiros de texto separados. (Idealmente: utilização de *datasets* usados noutros trabalhos de outros autores).

**4. Variáveis de Decisão/Decision Variables:** Descrever as variáveis de decisão e os seus domínios.

**5. Restrições/Constraints:** Descrever as restrições rígidas e flexíveis do problema e a sua implementação utilizando o SICStus Prolog.

**6. Função de Avaliação/Evaluation Function.** Descrever, quando for o caso, a forma de avaliar a solução obtida e a sua implementação utilizando o SICStus Prolog.

**7. Estratégia de Pesquisa/Search Strategy:** Descrever a estratégia de etiquetagem (“*labeling*”) implementada, nomeadamente no que diz respeito à ordenação de variáveis e valores. Deve também ser descrita a forma de implementação desta estratégia utilizando a linguagem.

**8. Visualização da Solução/Solution Presentation.** Explicar os predicados para visualizar a solução em modo de texto (obrigatório) ou modo gráfico (opcional).

**9. Resultados/Results:** Demonstrar exemplos de aplicação em casos práticos com diferentes complexidades e analisar os resultados obtidos. Devem utilizar as formas que considerem mais convenientes para apresentação dos resultados (tabelas e/ou gráficos). (Idealmente: comparação de resultados com outros autores/trabalhos.)

**10. Conclusões e Perspectivas de Desenvolvimento/Conclusions and Future Work:** Que conclusões retiram deste projeto? O que mostram os resultados obtidos? Quais as vantagens e limitações da solução proposta? (Idealmente: conclusão sobre a comparação com os resultados obtidos por outros autores/trabalhos.) Como poderiam melhorar o trabalho desenvolvido?

**Bibliografia:** Livros, artigos, páginas Web, usados para desenvolver o trabalho, apresentados segundo o formato sugerido no *template*.

**Anexos:** Código fonte, ficheiros de dados e resultados, e outros elementos úteis que não sejam essenciais ao relatório (não são contabilizados para o limite de 6 a 8 páginas).

## Problemas de Optimização/Decisão Sugeridos

### Puzzles 3D

1. Bolygok - <http://www.jaapsch.net/puzzles/bolygok.htm>
2. The Great Pyramid Pocket Puzzle - <http://www.jaapsch.net/puzzles/pyramid.htm>
3. CuFrog - <http://www.jaapsch.net/puzzles/culica.htm>
4. Instant Indecision - <http://www.jaapsch.net/puzzles/indecision.htm>

### Puzzles 2D

5. Izzi 2 - <http://www.jaapsch.net/puzzles/izzi2.htm>
6. Spectra - <http://www.jaapsch.net/puzzles/spectra.htm>
7. Number Snake - <http://logicmastersindia.com/lmitests/dl.asp?view=1&attachmentid=277>
8. Nurikabe - <http://logicmastersindia.com/lmitests/dl.asp?view=1&attachmentid=277>

### Problemas de Optimização

9. Construções Bricky
10. Escola de Línguas
11. MiMi Programação, Lda.

## Descrição Resumida dos Problemas

### Puzzles 2D/3D

Consultar os sítios para informações de problemas. A abordagem deve permitir lidar com tamanhos diferentes de tabuleiros e números diferentes de peças. É valorizada a geração dinâmica de problemas, e.g. gerar aleatoriamente o problema a ser resolvido. Deve ser possível visualizar a solução em modo de texto, de uma forma que facilite a sua validação.

### Problemas de optimização

Seguem-se as descrições dos problemas. As abordagens devem permitir problemas com diferentes dimensões. São valorizadas experiências com dimensões elevadas. Deve ser possível visualizar a solução em modo de texto, de uma forma que facilite a sua validação.

A empresa de construção Bricky subcontrata quatro outras empresas como subempreiteiras, nomeadamente para aluguer de veículos pesados, carpintaria, electricidade e segurança. A própria Bricky tem ao seu encargo uma equipa de trabalhadores de construção que trata da parte inicial de edificação da obra. No entanto, na fase inicial da obra é necessário fazer o aluguer de veículos pesados para transporte e deslocação de matéria-prima. Por outro lado, é ainda preciso fazer o pedido da matéria-prima, como cimento, betão e tijolos a fornecedores, o que implica um certo tempo e custo associados, apresentados abaixo. A equipa de construção tem que ser a primeira a começar o trabalho. Só quando o trabalho de construção estiver 75% completo é que a equipa de electricidade poderá começar a fazer a instalação. Tipicamente a empresa de carpintaria só poderá começar o trabalho num piso quando os electricistas tiverem terminado a instalação eléctrica desse piso. No entanto, caso seja necessário devido a restrições derivadas dos prazos de conclusão da obra, as duas equipas poderão trabalhar em simultâneo, o que implicará um custo 40% superior, trazendo também uma redução de 25% no tempo total das tarefas das duas equipas. Porém, as equipas de electricidade e de carpintaria não poderão trabalhar mais do que 60% do tempo em sobreposição, devido a regras de segurança.

- Considere que uma semana tem cinco dias úteis.
- O fornecedor de matéria-prima não realiza a entrega da matéria-prima, pelo que deve ser feita com veículos alugados pela Bricky. Cada caminhão consegue transportar 25 toneladas de matéria-prima e demora 3 dias a percorrer a distância entre o fornecedor e a obra. O aluguer do caminhão custa 80€ por dia. Cada piso acima do nível do solo requer 15 toneladas de matéria-prima para ser construído. Cada piso no subsolo requer 20 toneladas de matéria-prima.
- A equipa de construção/edificação requer duas semanas para concluir um piso em altura e quatro semanas caso se trate de um piso subterrâneo, devido à logística de escavação. Tem um custo de 150€ por piso. No caso de pisos subterrâneos é necessário contratar pelo menos uma máquina de escavação, que tem um custo de 75€ por dia. O aluguer de cada máquina de escavação adicional reduz o tempo necessário para construir um piso subterrâneo em 3 dias.
- O aluguer de uma grua reduz em 25% a duração do trabalho da equipa de edificação. Tem um custo de 120€ por dia.
- A equipa de electricidade demora uma semana a concluir a instalação eléctrica de um piso, mais uma semana para finalizar as ligações de todos os pisos quando o edifício estiver concluído. Tem um custo de 200€ por piso.
- A equipa de carpintaria demora dois dias a produzir as peças necessárias para cada piso (portas, rebordos de janela, rodapés) e durante este período não precisará de estar em obra. Considere que podem fazer as peças para todos os pisos de uma só vez. Por exemplo, se um edifício tiver 10 pisos, demorarão 20 dias a fazer as peças todas e podem escolher fazer as peças nos 20 dias de forma contínua ou demorar dois dias a fazer as peças de um piso e ir montar, para depois fazer as peças do piso seguinte e daí por diante. Tem um custo de 100€ por piso.
- A equipa de segurança entra quando todas as outras equipas tiverem terminado. A instalação de equipamentos de segurança demora duas semanas e tem um custo de 75€ por piso.

Sabendo que se pretende construir um arranha-céus com 150 pisos acima de terra e 15 pisos no subsolo, como deve ser organizada a construção de modo a maximizar o lucro e diminuir o tempo necessário? Defina o problema como um problema de satisfação de restrições, e resolva-o com PLR, de modo a que seja possível resolver problemas desta classe com diferentes parâmetros, isto é, deverá ser possível resolver problemas para edifícios com mais ou menos andares, com restrições parametrizáveis relativas aos tempos de sobreposição de equipas, etc.

## 10. Escola de Línguas

Uma escola de línguas oferece cursos de Espanhol, Francês e Inglês. Para tal, tem três professores, com estatutos diferentes. A prof. Annette é natural de França, mas também fala fluentemente Português, Espanhol e Alemão. O prof. Bóris é proveniente da Rússia mas teve que se refugiar em Itália, França, Espanha e Portugal por estar a ser procurado pelo KGB. Como tal, fala fluentemente as línguas desses países. Por último, o prof. Charles é americano e estava em Cuba de férias durante o período da Guerra Fria e da crise dos mísseis. Dessa forma, é versado também em Espanhol, Russo e Italiano (só porque gosta de ver filmes italianos).

A escola tem ainda duas funcionárias administrativas necessárias para o registo e manutenção dos cursos de línguas. Possui ainda duas salas, que são usadas para ministrar os cursos. Devido à valência dos professores, a direcção da escola pretende agora alargar o número de cursos para incluir Italiano, Alemão e Russo.

- Cada curso tem que ter um mínimo de 4h por semana e um máximo de 8h.
- Um professor tem que dar no mínimo um curso, mas pode dar no máximo dois cursos.
- A prof. Annette é a mais nova, pelo que ganha 25€ por hora de aula. Como está em início de carreira pode trabalhar até um máximo de 40h por semana.
- O prof. Bóris é o mais velho, mas por ordens médicas só pode trabalhar um máximo de 20h por semana. O seu vencimento é de 40€ à hora.
- O prof. Charles teve recentemente um filho e só poderá trabalhar um máximo de 30h por semana. O seu vencimento é de 30€ à hora.
- A manutenção de um curso requer cerca de 15h semanais a uma funcionária administrativa.
- Cada funcionária administrativa trabalha 40h por semana e aufer 15€ por hora. No entanto, poderá realizar até um máximo de 5h extra por semana, com um custo de 25€ à hora. Por outro lado, pode-se contratar uma funcionária adicional em part-time, que ganha 10€ à hora e trabalha um máximo de 10h por semana.
- Cada sala só pode ter uma aula de cada vez.
- Cada aluno paga mensalmente o equivalente a 10€ por cada hora de aula.
- Cada curso tem a duração de um ano.

Ao abrir os novos cursos, a escola pretende maximizar o lucro, reduzindo os custos operacionais e admitindo que cada curso preencherá todas as vagas disponíveis (15 para cada curso).

Defina o problema como um problema de satisfação de restrições, e resolva-o com PLR, de modo a que seja possível resolver problemas desta classe com diferentes parâmetros, isto é, deverá ser possível resolver problemas com mais ou menos línguas ou número de vagas diferentes por curso, com restrições parametrizáveis relativas aos cursos ministráveis por professor, tempos de contratação, salas, etc.

## 11. MiMi Programação, Lda.

A empresa MiMi Programação, Lda. realiza projetos de programação. A empresa funciona com um plano atualizado anualmente, no início do ano, data em já que são conhecidas novas encomendas (ainda não iniciadas). O objetivo do novo plano anual é obter o máximo lucro (receitas - despesas) possível nesse ano.

O novo plano não precisa de incluir (aceitar) todas as novas encomendas. Os projetos já em curso têm que ser cumpridos sem alteração. O novo plano pode incluir projetos que se prolongam para lá do ano. Estes não devem dar prejuízo à empresa até ao fim do projeto mais longo.

Cada encomenda tem os seguintes parâmetros: a) número de linhas de código fonte; b) complexidade (simples ou complexo); c) data de entrega intermédia; d) data de entrega final. O preço de cada encomenda deriva do número de linhas de código e da complexidade, de acordo com os respetivos preços unitários  $PU_S$ ,  $PU_C$ .

Projetos grandes (com mais de  $N_G$  linhas de código fonte) são sempre complexos. Projetos de dimensão média ( $>N_{med}$ ) têm sempre entrega intermédia. A entrega intermédia tem de ter uma percentagem do número total de linhas de código entre  $I_{min}$  e  $I_{max}$ . A data final pode ser excedida incorrendo num custo para a MiMi de  $A$  (%) do preço total) por dia de atraso, retirado ao pagamento final a receber. Este custo não pode ser superior a  $A_{max}$  (%) do que falta pagar, nem pode haver atraso superior a  $A_{dias}$  dias.

Os pagamentos são efetuados no momento de cada entrega. O valor do pagamento corresponde ao número de (novas) linhas de código entregues, descontado de eventuais custos por atraso.

O pessoal da empresa consiste em pessoal permanente e pessoal contratado por períodos de seis meses. Cada contratação dum novo programador tem um custo fixo  $C$ . Os contratos podem ser renovados sem custos. A dotação inicial da MiMi consiste em  $P$  seniores permanentes (não há juniores permanentes) e capitais próprios no montante de  $E$ . As despesas mensais, pagas no último dia de cada mês, consistem nos custos com pessoal  $Sr_C$ ,  $Sr_P$ ,  $Jr$ , e outras despesas, fixas, no montante total de  $M$ .

Todos os contratos se iniciam no primeiro dia do mês. A produtividade dum programador recém-contratado é nula nos primeiros  $Z$  dias, passando depois para o nível nominal de  $N$  linhas de código por dia. Cada colaborador trabalha 5 dias por semana, e usufrui de 11 dias de férias pagas em cada período de 6 meses.

Cada projeto tem de ter um programador sénior com uma dedicação mínima do seu tempo de  $D_{min}$  (%) por mês. Os projetos complexos têm de ter um programador sénior a  $D_C$  (%) no mínimo. Os projetos grandes ( $>N_G$ ) têm de ter um programador sénior permanente.

$A$	=	1	%	( custo de atraso diário	)
$A_{dias}$	=	60	dias	( atraso máximo	)
$A_{max}$	=	50	%	( custo máximo de atraso	)
$C$	=	500	€	( custo de contratação de novo programador	)
$D_C$	=	50	%	( dedicação mínima em projeto complexo	)
$D_{min}$	=	33	%	( dedicação mínima	)
$E$	=	20K	€	( dotação inicial de capitais próprios	)
$I_{max}$	=	90	%	( entrega intermédia	)
$I_{min}$	=	10	%	( entrega intermédia	)
$Jr$	=	1000	€	( custo mensal de programador júnior	)
$M$	=	5000	€	( outras despesas mensais (total)	)
$N$	=	35	linha de código fonte / dia	( produtividade nominal	)
$N_G$	=	100K	linhas de código fonte	( projeto complexo	)
$P$	=	4	pessoas	( dotação inicial de programadores permanentes	)
$PU_C$	=	25	€	( preço da linha de código complexo	)
$PU_S$	=	15	€	( preço da linha de código simples	)
$Sr_C$	=	3000	€	( custo mensal de programador sénior contratado	)
$Sr_P$	=	4000	€	( custo mensal de programador sénior permanente	)
$Z$	=	15	dias	( período de produtividade nula	)

Pretende-se um programa que permita gerar o plano anual para conjuntos variados de encomendas. Note que cada plano anual deve respeitar a continuação do plano do ano anterior (no primeiro ano tal é nulo, isto é, não há projetos em curso). Defina o problema como um problema de satisfação de restrições, e resolva-o com PLR, de modo a que seja possível resolver problemas desta classe com diferentes parâmetros, isto é, deverá ser possível resolver problemas com mais ou menos encomendas (crie estes dados de teste), com restrições parametrizáveis relativas ao tempo de dedicação mínima a diferentes tipos de projetos, etc.