

TRABALHO PRÁTICO Nº 2

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

Descrição

Objetivo: O objetivo deste trabalho é a construção de um programa em Programação em Lógica com Restrições para a resolução de um problema de otimização ou decisão combinatória. Adicionalmente, deverá ser elaborado um relatório no formato de um artigo, descrevendo o trabalho realizado e os resultados obtidos.

Sistema de Desenvolvimento: Deve ser usado o SICStus Prolog versão 4.6, o qual inclui um módulo de resolução de restrições sobre domínios finitos: *clp(FD)*. Caso seja necessária alguma configuração (para além da instalação padrão do software), ou seja usada uma fonte diferente da fonte por omissão, isso deverá estar expresso na documentação, que deverá ainda incluir os passos necessários para configurar e/ou instalar as componentes necessárias (em Windows e Linux). A impossibilidade de testar o código desenvolvido resultará em penalizações na avaliação. Todo o código deve ser devidamente comentado.

Condições de Realização

Constituição dos Grupos: Grupos de 2 estudantes, inscritos na mesma turma teórico-prática. Excepcionalmente, podem aceitar-se trabalhos individuais. Estudantes dispensados da frequência das aulas teórico-práticas (ex. TEs) deverão contactar os docentes no sentido de marcar sessões de acompanhamento do progresso e avaliação do trabalho.

Datas Importantes:

A partir de 2/12/2020	Formação dos grupos e escolha do enunciado no <i>Moodle</i> .
04/01/2021	Entrega, via <i>Moodle</i> , do artigo (formato PDF) e do código fonte desenvolvido. Submeter um único ficheiro ZIP com nome da forma: <code>PLOG_TP2_TX_#Grupo.ZIP</code> . (Exemplo: <code>PLOG_TP2_T3_Xpto_1.ZIP</code>)
05/01/2021 - 8/01/2021	Demonstrações dos trabalhos, em horário a combinar com o docente da turma respectiva.

Peso da Avaliação: ver ficha da Unidade Curricular no SIGARRA.

Escolha do Trabalho

Os estudantes devem formar grupos de trabalho e selecionar o tema no *Moodle*, no formulário a disponibilizar para o efeito a partir de 2 de Dezembro de 2020. Haverá um limite do número de grupos que podem escolher o mesmo tema, de modo a procurar garantir que todos os temas são igualmente selecionados. No final deste enunciado encontra-se uma lista dos temas disponíveis.

TRABALHO PRÁTICO Nº 2

Artigo

Cada grupo deve elaborar e entregar um artigo, e ainda realizar uma demonstração da aplicação desenvolvida. O artigo poderá ser escrito em Português ou Inglês no formato LNCS (Lecture Notes in Computer Science) da Springer. Para tal, a Springer disponibiliza no seu sítio web *templates* 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:

- a) **Título:** que transparea o trabalho realizado (não tem que ser apenas o tema do trabalho).
- b) **Autores e Afiliações:** identificação do trabalho e do grupo. Nas afiliações colocar a turma e o grupo, segundo o seguinte exemplo: FEUP-PLOG, Turma 3MIEIC08, Grupo Xpto_1.
- c) **Resumo / Abstract:** Contextualizar e resumir o trabalho, salientando o objetivo, o método utilizado e referindo os principais resultados e conclusões.
- d) **1. Introdução / Introduction:** Descrição da contextualização, objetivos e motivação do trabalho (não do ponto de vista de relatório da UC, mas de um ponto de vista computacional / aplicacional), referência sucinta ao problema em análise (idealmente, incluindo referências a outros trabalhos sobre o mesmo problema e sua abordagem), e descrição sucinta da estrutura do resto do artigo.
- e) **2. Descrição do Problema / Problem Description:** Descrever com detalhe o problema de otimização ou decisão em análise, incluindo todas as restrições envolvidas.
- f) **3. Abordagem / Approach:** Descrever a modelação do problema como um PSR / POR:
 - g) **3.1 Variáveis de Decisão / Decision Variables:** Descrever as variáveis de decisão e respectivos domínios, assim como o seu significado no contexto do problema em análise.
 - h) **3.2 Restrições / Constraints:** Descrever as restrições rígidas e flexíveis do problema e a sua implementação utilizando o SICStus Prolog.
 - i) **3.3 Função de Avaliação / Evaluation Function** (apenas para problemas de otimização): Descrever a forma de avaliar a solução e a sua implementação usando o SICStus Prolog.
- j) **4. Visualização da Solução / Solution Presentation:** Explicar os predicados que permitem visualizar a solução em modo de texto.
- k) **5. Experiências e Resultados / Experiments and Results:**
 - l) **5.1 Análise Dimensional:** Incluir exemplos de execução em instâncias do problema com diferentes dimensões e analisar os resultados obtidos.
 - m) **5.2 Estratégias de Pesquisa:** Devem ser testadas diferentes estratégias de pesquisa (heurísticas de escolha de variável e de valor), comparando os resultados obtidos. Devem ser usadas formas convenientes para apresentar os resultados (tabelas/gráficos).
- n) **6. Conclusões e Trabalho Futuro / Conclusions and Future Work:** Que conclusões retira deste projeto? O que mostram os resultados obtidos? Quais as vantagens e limitações da solução proposta? Como poderia melhorar o trabalho desenvolvido?
- o) **Referências / References:** Fontes bibliográficas usadas, incluindo Livros, artigos, páginas Web, entre outros, e apresentados segundo o formato sugerido no *template*.

TRABALHO PRÁTICO Nº 2

- p) **Anexo / Annex:** Resultados detalhados, e outros elementos úteis que não sejam essenciais ao relatório (não contabilizados para o limite de páginas).

Problemas de Otimização/Decisão Propostos

Puzzles (2D/3D)

1. Chess-Num: <https://erich-friedman.github.io/puzzle/chessnum/>
2. Chess-Position: <https://erich-friedman.github.io/puzzle/attack/>
3. C-Note: <https://erich-friedman.github.io/puzzle/100/>
4. Crypto-Product: <https://erich-friedman.github.io/puzzle/crypto/>
5. CuOxybon: <https://www.jaapsch.net/puzzles/culica.htm> (inclui otimização)
6. Diamond: <https://erich-friedman.github.io/puzzle/diamond/>
7. Dominosweeper:
<https://logicmastersindia.com/lmitests/dl.asp?attachmentid=790&view=1>
8. Gold Star: <https://erich-friedman.github.io/puzzle/star/>
9. Grape: <https://erich-friedman.github.io/puzzle/grapes/>
10. Light Bulb: <https://erich-friedman.github.io/puzzle/bulb/>
11. Multi-Balance: <https://erich-friedman.github.io/puzzle/2Dweight/>
12. Square: <https://erich-friedman.github.io/puzzle/square/>
13. Tricky Triple: <https://erich-friedman.github.io/puzzle/shape/>
14. White and Tan: <https://erich-friedman.github.io/puzzle/whitetan/>
15. Wrong Products:
<https://logicmastersindia.com/lmitests/dl.asp?attachmentid=790&view=1>

Problemas de Otimização

16. A Biblioteca do Professor Asdrúbal
17. Abertura de um Restaurante de Sushi
18. Cal.CoDe - Calendarização do Competições Desportivas
19. TeCoTuTeCo - Tecnologia de Colocação em Turmas em Tempos de Covid
20. O Padeiro da Vila em Época Covid

Descrição Resumida dos Problemas

Puzzles 2D/3D

Consultar os sítios para informações sobre os problemas. A abordagem deve permitir lidar com tamanhos diferentes de tabuleiros e números diferentes de peças. É valorizada a geração

TRABALHO PRÁTICO Nº 2

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 otimização

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.

16. A Biblioteca do Professor Asdrúbal

Com a mudança de ano letivo, o Professor Asdrúbal está a remodelar o seu gabinete. Pretende comprar um conjunto de prateleiras onde colocar todos os seus livros. Existem E tipos de prateleiras. A prateleira de tipo i tem um custo C_{e_i} , largura L_{e_i} e altura A_{e_i} . A única loja que vende prateleiras do agrado do Professor Asdrúbal, a Akolá, tem um stock limitado de cada tipo.

O Prof. Asdrúbal tem uma vasta coleção de L livros de várias áreas de estudo, como biologia ou física. Cada livro i tem uma largura L_i e altura A_i . Para facilitar a consulta dos livros, em cada prateleira, todos os livros de uma área devem ficar juntos, e ordenados por data de publicação.

O Professor Asdrúbal pretende minimizar o dinheiro gasto em prateleiras, de forma a que todos os seus livros estejam numa prateleira e fiquem organizados a seu gosto. Além disso, pretende saber como ficarão dispostos os seus livros pelas prateleiras.

Modele este problema como um problema de otimização / satisfação de restrições e resolva-o em PLR de forma a ser possível satisfazer problemas desta classe com diferentes parâmetros.

17. Abertura de um Restaurante de Sushi

O Sr. Eleutério pretende abrir um restaurante de sushi, com o nome Pequenos Prazeres Inesquecíveis de Nagoya (PPIN). Tem de decidir, de uma vasta seleção de pratos, quais deve incluir na ementa, e, de uma seleção de cozinheiros especializados, quais deve contratar. Cada cozinheiro é apenas capaz de confeccionar certos pratos, e tem uma exigência salarial distinta. Tendo em conta o feitio dos cozinheiros, não é possível negociar o seu salário. Cada prato tem um lucro médio mensal esperado e um tipo (maki, sashimi, nigiri, ...).

Para o seu restaurante ser mais atrativo, o Sr. Eleutério pretende que os seus menus tenham pelo menos X_i pratos do tipo i . Além disso, para não sobrecarregar o seu menu, tem também um número máximo de pratos.

O Sr. Eleutério pretende maximizar o seu lucro mensal esperado.

Modele este problema como um problema de otimização / satisfação de restrições e resolva-o em PLR de forma a ser possível satisfazer problemas desta classe com diferentes parâmetros.

18. Cal.CoDe - Calendarização do Competições Desportivas

Pretende-se implementar um sistema capaz de construir calendários para campeonatos nacionais de futebol e outras competições desportivas. O campeonato nacional de futebol é jogado por 18 equipas que devem jogar duas vezes com cada uma das restantes ao longo do campeonato (uma em casa e outra no campo do adversário). Em cada uma das duas voltas do campeonato, composta por 17 jornadas, cada equipa joga uma vez com cada oponente. O

TRABALHO PRÁTICO Nº 2

calendário da segunda volta do campeonato deve ser idêntico ao da primeira, variando unicamente o local de realização de cada jogo (ie, se uma equipa jogou um dado jogo em casa na primeira volta deverá jogá-lo fora na segunda). Cada equipa é de uma dada cidade, devendo-se evitar a realização de mais do que um jogo na mesma cidade numa dada jornada. Esta restrição flexível pode ser estendida para zonas do país. Por outro lado, interessa definir grupos de equipas cujos jogos simultâneos em casa se devem balancear. Um exemplo disto seria incluir como restrição flexível que do grupo Porto, Sporting, Braga e Benfica, dois deles devem jogar em casa em cada jornada. Deve-se também alternar ao máximo os jogos em casa e fora de cada equipa, ou seja, se numa dada jornada, uma equipa jogou em casa, na seguinte deverá jogar fora. O sistema deve ainda permitir a inserção de restrições de precedência entre jogos e jornadas mínimas e máximas para a ocorrência de cada jogo. O sistema deve ainda permitir gerar calendários com diferentes números de equipas e voltas.

Modele este problema como um problema de otimização / satisfação de restrições e resolva-o em PLR de forma a ser possível satisfazer problemas desta classe com diferentes parâmetros.

19. TeCoTuTeCo - Tecnologia de Colocação em Turmas em Tempos de Covid

A alocação em turmas é um problema recorrente nas universidades portuguesas, tornando-se ainda mais complexo em tempos pandémicos. Em cada curso existem diversas unidades curriculares (UC), cada uma com um conjunto de turmas, cada uma destas com o seu horário de funcionamento (eventualmente sobrepostos). Cada estudante encontra-se inscrito num conjunto de UCs e pode indicar entre 3 a 6 combinações de turmas (não sobrepostas), por ordem decrescente de preferência. Pretende-se um programa capaz de alocar todos os estudantes a turmas, procurando atingir vários objetivos:

- equilibrar o número de estudantes inscritos por turma (o mais próximo possível do número médio de estudantes por turma em cada UC)
- equilibrar o número de estudantes de número par e ímpar em cada turma (para permitir aulas alternadas com metade da turma em sala de aula)
- tentar não ultrapassar a capacidade das turmas (a capacidade da turma é usualmente um pouco maior do que a média)
- dar preferência na alocação a estudantes com média mais elevada

Adicionalmente, pode-se considerar maximizar a colocação de estudantes em opções declaradas (ie, um estudante pode eventualmente ser alocado numa opção não declarada). Pode-se ainda considerar lidar com combinações de turmas sobrepostas, tentando minimizar a quantidade e duração das sobreposições na solução.

Modele este problema como um problema de otimização / satisfação de restrições e resolva-o em PLR de forma a ser possível satisfazer problemas desta classe com diferentes parâmetros.

20. O Padeiro da Vila em Época Covid

Em época de pandemia, o Sr. Marciliano, padeiro da vila, tem de vender o seu produto levando-o diretamente à casa dos seus clientes com a sua carrinha. Ele parte da padaria às 7h da manhã e deve passar por um conjunto de habitações. Quando terminar todas as suas entregas, dirige-se de volta à padaria. O tempo de viagem entre quaisquer duas habitações, e entre a padaria e uma habitação é conhecido.

TRABALHO PRÁTICO Nº 2

Cada cliente indica a hora preferencial para que o pão lhe seja entregue e tem uma tolerância de 30 minutos. As entregas só podem ser feitas até 10 minutos antes da hora preferencial. Quando o padeiro chega a uma habitação, demora 5 minutos a efetuar a entrega.

O Sr. Marciliano precisa de um programa que defina qual o itinerário a seguir, de forma a otimizar vários objetivos:

- Minimizar o tempo total do itinerário.
- Equilibrar o tempo de atraso nas entregas (entregas que sejam mais tardias que as janelas de tempo de 40 minutos).

Modele este problema como um problema de otimização / satisfação de restrições e resolva-o em PLR de forma a ser possível satisfazer problemas desta classe com diferentes parâmetros.