**Campo De Férias**

João Furriel nº201104913

Ventura Pereira nº201404690

Faculdade de Engenharia da Universidade do Porto

**Abstract.** O trabalho proposto consistia na organização de um campo de férias, que procurava fomentar ao máximo a interação entre os participantes- tal era atingido por diminuir as co-ocorrências de elementos em relação a eventos anteriores, tendo em atenção as competências dos mesmos e requisitos das atividades.

Como tal, foram utilizadas restrições na formação de grupos para as atividades a decorrer num dia.

1. **Introdução**

O objetivo para este trabalho era a distribuição otimizada, no que toca a co-ocorrências de participantes em grupos nas atividades, tendo em conta os requisitos das mesmas e as competências dos participantes.

No decorrer deste artigo abordaremos a nossa abordagem ao problema, de forma detalhada – analisando as restrições utilizadas, as variáveis de decisão, função de avaliação e estratégia de pesquisa. Para além disso, analisaremos os resultados e conclusões do trabalho realizado.

1. **Descrição do Problema**

O nosso trabalho, Campo de Férias, era um problema de otimização.

Este consistia na formação de grupos para variadas atividades, podendo estas estar sujeitas, ou não, a requisitos obrigatórios. Tendo cada participante um set de competências, os grupos eram formados de acordo com a compatibilidade dessas duas vertentes.  
 Para além disso, a otimização referencia-se a minimizar a co-ocorrência de elementos em grupos, tendo em conta atividades prévias.

1. **Abordagem**
   1. **Variáveis de decisão:**

A nossa estrutura de dados é baseada numa lista de listas de listas.

A primeira componente de listas é referente às atividades a realizar.

A segunda componente de listas é referente aos grupos. A quantidade dos mesmos é calculada através da divisão inteira entre o número de participantes e o número máximo de elementos que a atividade requer.

As nossas variáveis de decisão são todas aquelas que constituem o conteúdo dos grupos. Como domínio terão desde o caso base, 1, até ao número de participantes.

* 1. **Restrições:**

Em termos de restrições, decidimos implementar dois tipos de restrições – físicas e de competências.

Todas as atividades podem, ou não, ter certas exigências a nível físico – sexo, altura – e em termos de idade. Assim, haverá sempre uma verificação dos participantes, para os valores que a atividade requisita, referente aos parâmetros referidos. De referir que, caso a atividade não tenha requisitos, são utilizados valores default.   
Assim, temos o sexo identificado com 1 ou 2 – feminino e masculino, respetivamente, altura em centímetros e idade em inteiros. Para valores default utilizámos o 0 para sexo, altura e idade mínima. A altura máxima ficou definida nos 300cm, enquanto que a idade máxima ficou nos 75 anos.

Por último, as atividades podem possuir um set de competências necessárias em x número (pode precisar que haja 2 participantes a ter determinada competência). Assim, cada participante tem o seu próprio set de competências, que serão avaliadas, conforme a atividade.

* 1. **Função de avaliação**

De forma a avaliar a solução obtida poderemos olhar para a consola do SICstus, de forma a observar, quer se os participantes cumprem com as restrições impostas e se são heterogéneos.   
  
Quanto à implementação, criamos variáveis já instanciadas, que representam todas as combinações de pares possíveis. Associadas, temos variáveis não instanciadas que guardarão e acumularão a ocorrência desses pares. A sua soma pesada será utilizada para avaliar a otimização. Quanto menor for a soma, mais optimizador estará.

* 1. **Estratégia de pesquisa**

Para o labeling, de forma a aplicar as restrições, utilizámos uma matriz, de atividades por competência. À semelhança de uma matriz de adjacências, o valor 1 representará quando uma atividade está conectada a uma competência necessária e 0 para quando não o é. Analisando essa matriz, será criada uma lista por atividade, que guardará as competências necessárias e será utilizada nas restrições.

Posteriormente, as restrições ao grupo são feitas com os operadores, precedidos de cardinais.

1. **Visualização da solução**

De forma a visualizar a solução em modo texto, utilizámos, após a execução do labeling, um predicado printSolution(List). Esta lista é referente a uma lista de listas de listas, que está dividida por atividades, cada uma com um determinado número de grupos.

Dentro desse predicado, percorrendo essa lista, chama-se o printActivity que dará print aos valores que lhe dizem respeito (requisitos, nome) e, percorrendo os grupos, no printGroup, mostrámos os elementos com o printParticipant.

1. **Resultados**

Para visualizar os resultados, registámos os resultados numa tabela. Esta contém o número de atividades a realizar, o timeout enviado como parâmetro na execução e o fator de co-ocorrência dos elementos nos grupos.

A primeira tabela continha apenas a restrição de duas das três atividades exigirem um elemento do sexo feminino, enquanto que na segunda tabela representa a execução do programa com as seguintes restrições:

* Atividade “Futebol”:
  + Um elemento com habilidade para jogar futebol;
  + Um elemento do sexo masculino;
  + Um elemento com idade mínima de 20 anos.
* Atividade “Jogar às Cartas”:
  + Um elemento com habilidade para jogar às cartas;
* Atividade “Caminhar”:
  + Um elemento do sexo feminino.

|  |  |  |
| --- | --- | --- |
| Nº Atividades | TimeOut | Co-ocurrences Factor |
| 3 | 1s | 230 |
| 3 | 5s | 216 |
| 3 | 1200s | 198 |

Tabela 1

|  |  |  |
| --- | --- | --- |
| Nº Atividades | TimeOut | Co-ocurrences Factor |
| 3 | 1s | 210 |
| 3 | 5s | 186 |
| 3 | 1200s | 178 |

Tabela 2

1. **Conclusões e trabalho**

Ao longo deste projeto, podemos concluir inúmeras coisas.

Primeiramente, a cadeira de PLOG obrigou-nos a uma mudança de mindset, no que à programação toca. Contudo, com restrições, ainda tivemos que nos adaptar mais, especialmente no que toca à abordagem do problema.

Habituados a utilizar uma abordagem bottom-up, iniciámos mal a estruturação do programa. Com as dificuldades sentidas nessa execução, sentimos a necessidade de mudar para top-down. Sentimos portanto, que evoluímos no sentido de avaliação à melhor abordagem a seguir a certos problemas, como os de otimização em PLOG, com restrições.

De seguida, os resultados obtidos demonstram que a programação por restrições é útil na procura de soluções que procuram ser otimizadas, consoante vários fatores -como o tempo ou número de condições prévias.

Note-se, também, que apesar de haver vantagens como a facilidade de definir restrições, existe a desvantagem de, sem timeout, existir a possibilidade de a execução ser muito demorada. Tal torna a obtenção de soluções ótimas, no que diz respeito à heterogeneidade dos grupos, muito complicada em tempo útil.

Por último, para melhorar o trabalho, poderíamos por uma melhor interface de utilizador, de forma a adicionar à nossa base de dados, de forma dinâmica, mais participantes, atividades ou restrições.

1. **Bibliografia**

* Cardoso, Henrique Lopes; Programação em Lógica com Restrições, DEI,FEUP,2013;
* Cardoso, Henrique Lopes; PLR com SICStus, DEI,FEUP,2013
* SWI Prolog Reference Manual. Disponível em <http://www.swi-prolog.org/pldoc/doc_for?object=manual>
* StackOverflow. https://stackoverflow.com/

1. **Anexos**

* Exemplo de execução do programa na tabela 2 com timeout de 1200s:

Activity:

Name Min Height Max Height Min Age Max Age Elements Sex (Of at least 1 Element) Required Skills (At Least 1 Element)

------------------ ---------- ---------- ------- ------- -------- --------------------------- ------------------------------------

football 0 300 0 75 6 None None

Grup 1:

Participants:

Name Sex Age Height Skills

------------------ --- --- ------ ------

Joao M 20 170 2

Pedro M 12 179 2,7

Ventura M 21 175 6

Luis M 14 167 6

Vasco M 20 161 1,5

Maria F 13 151 1,3

Grup 2:

Participants:

Name Sex Age Height Skills

------------------ --- --- ------ ------

Joana F 18 167 8

Manuela F 9 160 8,7

Elisabete F 11 165 3

Manuel M 20 160 3

Ricardo M 25 167 3

Marta F 25 170 3,8

Activity:

Name Min Height Max Height Min Age Max Age Elements Sex (Of at least 1 Element) Required Skills (At Least 1 Element)

------------------ ---------- ---------- ------- ------- -------- --------------------------- ------------------------------------

cards 0 300 0 75 6 F None

Grup 1:

Participants:

Name Sex Age Height Skills

------------------ --- --- ------ ------

Joao M 20 170 2

Pedro M 12 179 2,7

Ventura M 21 175 6

Luis M 14 167 6

Vasco M 20 161 1,5

Maria F 13 151 1,3

Grup 2:

Participants:

Name Sex Age Height Skills

------------------ --- --- ------ ------

Joana F 18 167 8

Manuela F 9 160 8,7

Elisabete F 11 165 3

Manuel M 20 160 3

Ricardo M 25 167 3

Marta F 25 170 3,8

Activity:

Name Min Height Max Height Min Age Max Age Elements Sex (Of at least 1 Element) Required Skills (At Least 1 Element)

------------------ ---------- ---------- ------- ------- -------- --------------------------- ------------------------------------

walk 0 300 0 75 6 F None

Grup 1:

Participants:

Name Sex Age Height Skills

------------------ --- --- ------ ------

Joao M 20 170 2

Pedro M 12 179 2,7

Ventura M 21 175 6

Joana F 18 167 8

Manuela F 9 160 8,7

Elisabete F 11 165 3

Grup 2:

Participants:

Name Sex Age Height Skills

------------------ --- --- ------ ------

Luis M 14 167 6

Vasco M 20 161 1,5

Maria F 13 151 1,3

Manuel M 20 160 3

Ricardo M 25 167 3

Marta F 25 170 3,8