



Universidade do Porto
Faculdade de Engenharia
FEUP

INTELIGÊNCIA ARTIFICIAL

3º ANO DO MESTRADO INTEGRADO EM ENGENHARIA
INFORMÁTICA E COMPUTAÇÃO

Otimização da gestão de projetos

Authors:

Duarte PINTO

- up201304777 - up201304777@fe.up.pt

Filipa RAMOS

- up201305378 - up201305378@fe.up.pt

Gustavo SILVA

- up201304143 - up201304143@fe.up.pt

13 de Abril de 2016

Conteúdo

1	Introdução	2
2	Especificação	4
2.1	Problematização	4
2.2	Aquitetura	4
2.3	Formato do input	5
2.4	Fases	7
2.5	Algoritmos Genéticos	7
2.6	Arrefecimento Simulado	7
3	Trabalho Realizado	7
4	Testes	7
5	Conclusões	7

1 Introdução

No âmbito da unidade curricular de Inteligência Artificial pretende-se desenvolver um programa que, com base em algoritmos genéticos e arrefecimento simulado, faça a gestão de um projeto balançando os elementos participantes e as tarefas a realizar do mesmo. O sistema é composto por um conjunto de tarefas que pertencem ao projeto em análise. Cada tarefa tem uma competência indispensável ao seu cumprimento e uma duração. Cada elemento tem um conjunto de competências sendo que o mesmo tem um nível de capacidade para cumprir cada uma. A gestão a ser realizada tem em vista minimizar o tempo ocupado para satisfazer todas as tarefas do projeto usando a melhor combinação de elementos para cada tarefa. Será feita uma análise comparativa entre o desempenho das solução encontradas com algoritmos genéticos e arrefecimento simulado.

Os objetivos principais do projeto passam pela exploração da implementação prática dos algoritmos genéticos e do algoritmo de arrefecimento simulado. Através dos dados obtidos, visa-se também realizar uma comparação da solução encontrada com ambos os algoritmos. Este processo irá fomentar o conhecimento adquirido, evidenciando as vantagens principais de cada algoritmo e as suas dicotomias principais.

Espera-se que surjam dificuldades na implementação prática dos algoritmos estudados teoricamente, principalmente na construção dos cromossomas pois existem dúvidas em relação à sua influência na eficiência da solução encontrada. Para além disto, a melhor adaptação da função de avaliação ao problema por forma a obter os melhores resultados revela-se um processo tumultuoso. Os membros decidiram optar por otimizar o tempo utilizado a concluir todas as tarefas do projeto em estudo. Desta forma, a melhor solução será a que implicará um menor tempo de conclusão do projeto em questão.

2 Especificação

2.1 Problematização

O sistema tem por objetivo otimizar a atribuição de membros por tarefas num dado projeto. Os dados do mesmo são introduzidos por input através de um ficheiro.

Um projeto em análise caracteriza-se por um conjunto de tarefas (Task) a cumprir, tendo estas um nome (por motivos de identificação) e uma duração. Existe ainda um conjunto de elementos (Element) que podem ser atribuídos a essas mesmas tarefas. Um elemento é identificado por um nome e tem uma lista de competências (Skill) avaliadas em função da sua capacidade. Por exemplo, o elemento "joão" tem competências na área da informática a um nível 6 e na área da economia com nível 10.

2.2 Arquitetura

O projeto foi dividido em três "packages" principais que representam os três níveis mais importantes. Um dos packages diz respeito às classes que guardam informação sobre as tarefas, os elementos e as competências. Os outros dois dizem respeito à implementação do algoritmo genético ou do arrefecimento simulado.

A arquitetura pensada para o projeto passa pela implementação de três classes principais que interagem entre si próprias. A classe "Task" representa uma tarefa e guarda a sua duração. Um elemento é representado pela classe "Element" que mantém a identificação do mesmo. A classe "Skill" corresponde a uma competência. Estas ligam-se entre si por forma a que um elemento tenha vários skills e um skill tenha várias tarefas tal como é visível na figura 1. A relação que se pretende obter será a que liga os elementos a tarefas.

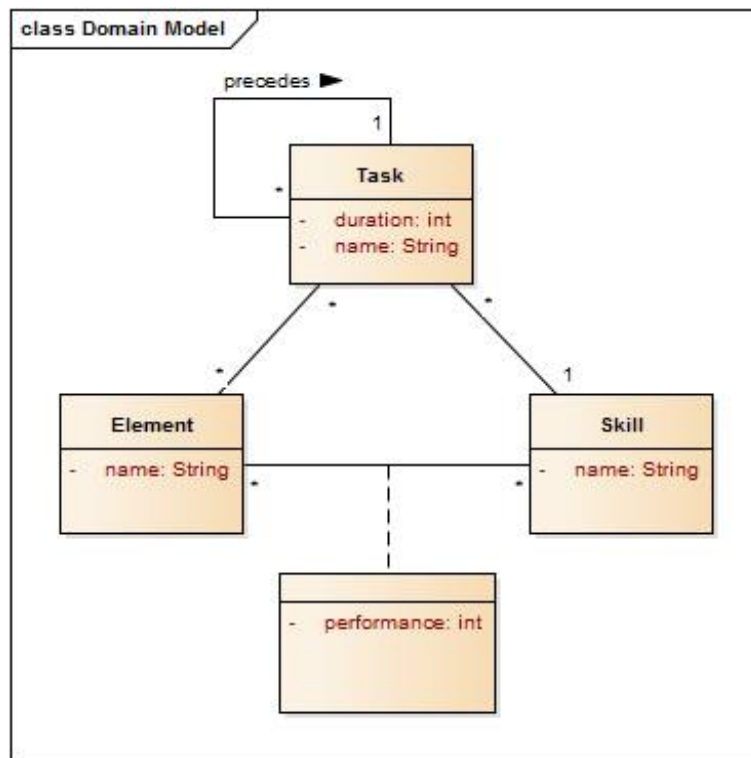


Figura 1: Diagrama de classes UML

2.3 Formato do input

O input será colhido de um ficheiro de formato *json* e será estruturado conforme está representado no exemplo apresentado a seguir. Espera-se receber:

- uma lista de *skills*
- uma lista de *tasks* sendo que cada uma tem um **nome** e uma **competência**
- uma lista de *elements* sendo que cada um tem um **nome** e um array de **competências** com as respectivas **capacidades**

```

1  {{
2    "skills": [
3      "Z",
4      "Y",
5      "Z"
6    ], "tasks": [
7      {
8        "name": "A",
9        "skill": 0
10     },
11     {
12       "name": "B",
13       "skill": 1
14     }
15   ], "elements": [
16     {
17       "name": "Duarte Pinto",
18       "skills": [
19         [0, 0.5],
20         [2, 0.1]
21       ]
22     },
23     {
24       "name": "Filipa Ramos",
25       "skills": [
26         [0, 1]
27       ]
28     },
29     {
30       "name": "Gustavo Silva",
31       "skills": [
32         [1, 0.5],
33         [2, 0.1]
34       ]
35     }
36   ]
37 }

```

2.4 Fases

O projeto será dividido em fases de trabalho. A inicial passa pelo desenvolvimento da arquitetura supracitada. Seguidamente, proceder-se-á implementação dos algoritmos genéticos. Finalmente, será desenvolvido o arrefecimento simulado. As fases abordadas até ao presente relatório foram as duas primeiras. A arquitetura explicitada pelo diagrama de classes uml já foi implementada e a obtenção de solução por algoritmos genéticos encontra-se numa fase avançada.

2.5 Algoritmos Genéticos

Explicar o algoritmo usado: - estrutura do cromossoma - função de avaliação - seleção - cruzamento - mutações

2.6 Arrefecimento Simulado

Explicar o algoritmo usado.

3 Trabalho Realizado

Incluir trechos do código já implementado.

4 Testes

Explicar os testes planeados.

5 Conclusões

Conclusões retiradas.

Palavras Chave

Algoritmos Genéticos	7	Skill	4
Element	4	Task	4