



Universidade do Porto
Faculdade de Engenharia

FEUP

Optimização na Programação de uma Conferência

Relatório Intercalar

Inteligência Artificial

3º ano do Mestrado Integrado em Engenharia Informática e Computação

Elementos do Grupo:

Anderson Rogério da Silva Gralha – 201710810– up201710810–

António Miguel Silva Pereira – 201307910– up201307910–

Vitor Emanuel Fernandes Magalhães – 201503447– up201503447–

08 de abril de 2018

Objetivo

O objetivo deste trabalho reside na otimização da programação de uma conferência que dura alguns dias, geralmente três dias.

Em cada dia, ocorrem sessões temáticas, que duram, no máximo, 2 horas e contêm, no mínimo 2 full-papers (30 min cada). Existem também short-papers, que duram 20 minutos.

Cada sessão tem um apresentador do paper(que é autor ou coautor). Como cada paper cobre um ou mais temas, é necessário garantir que os papers que são apresentados contenham o tema da sessão.

Existe um máximo de sessões em paralelo e, por dia, há, no máximo, 4 horários para sessões, intervalados por coffee-breaks e almoço.

Descrição

Especificação

O objectivo deste trabalho é organizar uma programação de uma conferência de sessões. Para este objetivo, vai ser utilizada a metodologia de otimização de Algoritmos Genéticos.

Estados

Um indivíduo representa uma solução possível do problema, composto por uma *string* gerada através de N repetições de uma sequência composta pelo dia e pelas suas respectivas sessões.

Cada sessão contém um tema, uma duração e é composta por vários papers, que alteram a duração. Em caso de sessões em paralelo, poderá haver, no máximo, M sessões representadas pelo horário.

Função de cruzamento

Irá ser utilizada uma estratégia HUX(half uniform crossover). Com uma probabilidade P_x , o gene de um indivíduo é trocado pelo gene do outro e vice-versa.

Função de mutação

Uma mutação poderá ocorrer em qualquer bit, com uma probabilidade P_n bastante reduzida.

Função de avaliação

Para a função de avaliação, são aplicadas diversas verificações para garantir que os indivíduos selecionados para a próxima geração possuam características mínimas, ou seja, características que se não estiverem presentes nos indivíduos, não podem ser replicadas de forma nenhuma para as próximas gerações. Para uma conferência, é essencial que:

- Um dos temas do paper esteja relacionado com o tema da sessão em que está a ser apresentado.
- Haja no mínimo dois full-papers em cada sessão.
- O apresentador do paper, que pode ser coautor ou autor, não possa estar em duas apresentações ao mesmo tempo.

Após descartar os indivíduos que não satisfaçam as condições indicadas anteriormente, atribui-se um valor entre 0 e 1, conforme as combinações dos papers. O valor mínimo é 0 quando há apenas dois full papers na sessão e o valor máximo é 1 quando há 2 full papers e 3 short papers.

A tabela abaixo representa as possíveis combinações de papers, as suas durações e a sua representação em binário, já considerando os dois full-papers obrigatórios, mas sem os indicar.

Combinação	Duração	Representação
20 + 20 + 20	60	111
30 + 30	60	110
20 + 30(ou 30 + 20)	50	101
20 + 20	40	100
30	30	011
20	20	010
0	0	001

Tabela 1: Tabela de combinações de Papers

A terceira etapa consiste em verificar as diferenças de tempo das apresentações dos papers durante todos os dias, gerando o valor máximo caso todas os dias de apresentação contenham a mesma duração.

Critérios de Paragem

Para parar o algoritmo, haverá um limite de iterações, assim como uma verificação do *score* do melhor indivíduo: se este não aumentar passadas n iterações, então esse será considerado o melhor indivíduo e o algoritmo para.

Trabalho Efetuado

O trabalho até agora realizado foi a divisão sucinta do problema em várias classes de Java. Também já foi construído o cromossoma de um indivíduo, que representará uma conferência, e cada parte do cromossoma de maneira a facilitar o processo das verificações da função de avaliação.

Resultados Esperados

Para verificar se a solução obtida vai de acordo com o que se espera, a função de avaliação contém filtros que rejeitam soluções inválidas e, de modo a confirmar a exatidão da solução, os critérios de passagem que dão a solução quando o *score* do melhor indivíduo não aumentar passadas *n* iterações, serão utilizados.

Conclusões

Até o momento conclui-se que a utilização de algoritmos genéticos fornece um ganho significativo no que diz respeito ao tempo gasto para se solucionar o problema, pois é possível, em apenas algumas iterações, remover indivíduos que representam soluções inválidas e cruzar os melhores, formando rapidamente uma solução ideal.

Recursos

1. Oliveira, Eugénio: Métodos de Resolução de Problemas e Algoritmos para a Evolução
2. <https://bcc.ime.usp.br/tccs/2003/anselmo/node12.html>
3. Eclipse IDE for Java Developers