



Licenciatura em Engenharia Informática

Relatório de trabalho prático

Trabalho prático nº2 – Problema de Otimização

Leandro Adão Fidalgo / 2017017144
Pedro dos Santos Alves / 2019112789

05/01/2020

Índice

1. Introdução.....	3
2. Implementação.....	4
2.1 Trepa colinas normal.....	4
2.2 Trepa colinas alterado.....	4
2.3 Algoritmo evolutivo (Seleção por torneio, Crossover double-point, Mutação por troca)	4
2.4 Algoritmo evolutivo (Seleção por torneio, Crossover double-point, Mutação por inserção)	5
2.5 Algoritmo híbrido (1 e 3).....	5
2.6 Algoritmo híbrido (2 e 3).....	5
2.7 Algoritmo híbrido (1 e 4).....	5
2.8 Algoritmo híbrido (2 e 4).....	5
3. Análise de resultados.....	6
4. Conclusão	7

1. Introdução

A realização deste trabalho serve para consolidar todos os assuntos desenvolvidos nas aulas e aprender outras funcionalidades que são propostas na realização do trabalho prático.

No início da realização do trabalho, foi elaborado um pequeno esquema para ser mais perceptível de como este tinha de ser elaborado e todos os processos necessários para a sua realização inclusive o estudo dos diversos algoritmos.

O principal objetivo foi a implementação de um algoritmo de pesquisa local com dois critérios de vizinhança diferentes, um algoritmo evolutivo, no qual deve ser explorado diferentes tamanhos para a população e operadores genéticos e ainda um método híbrido combinando as duas abordagens anteriores, os quais foram implementados sem grandes dificuldades devido ao facto de nas aulas práticos ter-nos sido falado um pouco sobre tudo a utilizar no trabalho.

No segundo capítulo é apresentada a implementação e uma breve descrição de todos os tipos de algoritmos utilizados e algumas das alterações que tiveram de ser efetuadas.

No terceiro capítulo é feita uma análise dos resultados e uma descrição dos testes realizados.

Por fim, no último capítulo, consta a conclusão e onde é retirada a conclusão do trabalho.

2. Implementação

O projeto foi baseado nos exercícios disponibilizados pelo professor das aulas práticas.

2.1 Trepa colinas normal

No início do projeto do trepa colinas foi feito um estudo sobre o que era o trepa colinas e só após esse estudo foi inicializado o projeto, sendo assim o trepa colinas o algoritmo de pesquisa local.

O projeto foi inicializado começando por alterar o código do exercício 7 das aulas práticas, a primeira função a ser alterada foi a inicialização dos dados, pois teve de se ir buscar ao ficheiro, os dados dos vértices, das arestas e ainda criar a matriz das adjacências de modo a colocar “1” nas ligações com os vértices que estão ligados e “0” nos que não estão ligados.

De seguida, foram efetuadas alterações no trepa colinas de modo a que este avaliasse a solução inicial e gerasse uma solução vizinha para ser avaliada e de seguida ambas serem comparadas. Neste primeiro caso o vizinho apenas altera dois vértices da solução.

2.2 Trepa colinas alterado

Neste trepa colinas apenas foi alterado o modo como é gerado os vizinhos para depois ser comparado o custo deste, com o custo da solução, sendo assim alterados 3 ou 4 vértices da solução para este trepa colinas alterado.

2.3 Algoritmo evolutivo (*Seleção por torneio, Crossover double-point, Mutação por troca*)

Antes de iniciar o algoritmo evolutivo foi feito um estudo e só após esse estudo é que foram iniciadas as alterações necessárias no exercício 8 das aulas práticas. A primeira função a ser alterada foi a inicialização dos dados, na qual são definidos alguns parâmetros iniciais, tais como a população, a probabilidade de recombinação e a probabilidade de mutação. De seguida é utilizada a recombinação com dois pontos de corte e no caso deste projeto irá ter dois descendentes, bastante idêntico á recombinação utilizada no problema do caixeiro viajante. Após a recombinação com dois pontos foi feita a mutação por troca na qual, são escolhidos dois valores da solução e são trocados.

2.4 Algoritmo evolutivo (Seleção por torneio, Crossover double-point, Mutação por inserção)

Neste algoritmo, a diferença está na mutação por inserção, na qual é escolhido um valor e é colocado mais á frente na solução, fazendo assim com que a solução fique bastante alterada.

2.5 Algoritmo híbrido (1 e 3)

Neste algoritmo foi conjugado o trepa colinas normal com o algoritmo evolutivo com mutação por troca.

2.6 Algoritmo híbrido (2 e 3)

Neste algoritmo foi conjugado o trepa colinas alterado com o algoritmo evolutivo com mutação por troca.

2.7 Algoritmo híbrido (1 e 4)

Neste algoritmo foi conjugado o trepa colinas normal com o algoritmo evolutivo com mutação por inserção.

2.8 Algoritmo híbrido (2 e 4)

Neste algoritmo foi conjugado o trepa colinas alterado com o algoritmo evolutivo com mutação por inserção.

3. Análise de resultados

Foram realizados vários testes com o objetivo de retirar algumas conclusões acerca do que acontecia ao variar pequenas variáveis no ambiente.

O primeiro teste realizado foi a análise ao trepa colinas normal e no qual se verificou que ao aumentar o número de iterações e mantendo o número de runs o “melhor resultado” tende para valores mais satisfatórios e a “média” também. Tem um rápido processamento e não precisa de muita memória.

No segundo teste realizado, foi utilizado o trepa colinas alterado e verificou-se que este método é ligeiramente pior que o anterior, porém constatou-se exatamente a mesma coisa que no anterior, simplesmente tinha valores ligeiramente menos satisfatórios.

O terceiro teste realizado foram analisados os dois algoritmos evolutivos, nos quais se constatou que ao aumentar o número da população o resultado final melhora. Ao aumentar a probabilidade de recombinação piora o resultado final pretendido, como foi concluído ao ser alterado o critério de vizinhança no trepa colinas anteriormente. Ao aumentar a probabilidade de mutação o resultado melhora ligeiramente. A mutação por troca parece ser ligeiramente melhor que a mutação por inserção, pois a mutação por troca apenas troca dois vértices enquanto a mutação por inserção mexe com parte da solução. Acontece o mesmo quando se mexe muito com a vizinhança ou se aumenta a probabilidade de recombinação. Ao aumentar o número de gerações os resultados também melhoram. Todo este processo é mais lento e necessita de mais memória ao contrário do trepa colinas e para além disso os resultados finais são piores.

No quarto teste realizado foram analisados os diversos algoritmos híbridos e todas as suas combinações, chegou-se á conclusão que a relação entre os resultados finais é o mesmo que acima explicado, quando o trepa colinas normal é utilizado é melhor do que o trepa colinas com a vizinhança alterada e quando é utilizada a mutação por troca é melhor do que a mutação por inserção. Ao adicionar o trepa colinas ao algoritmo evolutivo melhorou significativamente os resultados, com parâmetros iniciais baixos (população, probabilidade de recombinação e mutação, etc..). Resultados finais relativamente bons.

4. Conclusão

Com a realização deste trabalho, concluímos que ao realizar este trabalho percebemos melhor o funcionamento do trepa colinas e do algoritmo evolutivo e de toda a teórica associada aos algoritmos.

Todos os objetivos tanto os do trabalho como também aqueles que impusemos a nós mesmos durante a realização do mesmo foram alcançados sem grandes dificuldades.