



UFOP

Universidade Federal
de Ouro Preto

**Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas**

**Um método baseado em Estratégia
Evolutiva e *Variable Neighborhood
Descent* para o Problema Quadrático
de Alocação**

Kelly Márcia de Oliveira

**TRABALHO DE
CONCLUSÃO DE CURSO**

ORIENTAÇÃO:
Fernando Bernardes de Oliveira

**Setembro, 2017
João Monlevade–MG**

Kelly Márcia de Oliveira

**Um método baseado em Estratégia Evolutiva e
Variable Neighborhood Descent para o
Problema Quadrático de Alocação**

Orientador: Fernando Bernardes de Oliveira

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Universidade Federal de Ouro Preto

João Monlevade

Setembro de 2017

O482m Oliveira, Kelly Marcia de.
Um método baseado em estratégia evolutiva e Variable Neighborhood Descent para o problema quadrático de alocação [manuscrito] / Kelly Marcia de Oliveira. - 2017.

40f.:

Orientador: Prof. Dr. Fernando Bernardes de Oliveira.

Monografia (Graduação). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Departamento de Computação e Sistemas de Informação.

1. Sistemas de informação. 2. Heurística. 3. Otimização combinatória. I. Oliveira, Fernando Bernardes de. II. Universidade Federal de Ouro Preto. III. Título.

CDU: 004.023

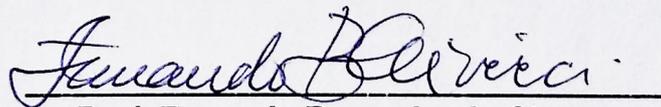
Catálogo: ficha@sisbin.ufop.br

FOLHA DE APROVAÇÃO DA BANCA EXAMINADORA

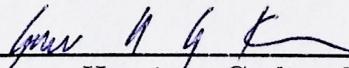
Um método baseado em Estratégia Evolutiva e *Variable Neighborhood Descent* para o Problema Quadrático de Alocação

Kelly Márcia de Oliveira

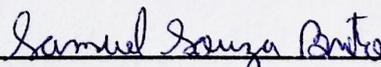
Monografia apresentada ao Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto como requisito parcial da disciplina CSI499 – Trabalho de Conclusão de Curso II do curso de Bacharelado em Sistemas de Informação e aprovada pela Banca Examinadora abaixo assinada:



Prof. Fernando Bernardes de Oliveira
Doutor
DECSI – UFOP



Prof. George Henrique Godim da Fonseca
Doutor
Examinador
DECSI – UFOP

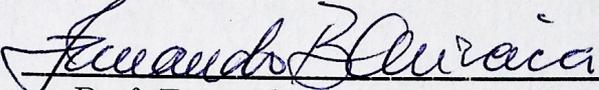


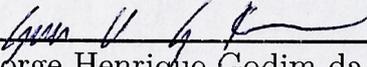
Prof. Samuel Souza Brito
Mestre
Examinador
DECSI – UFOP

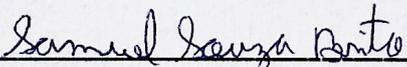
João Monlevade, 5 de setembro de 2017

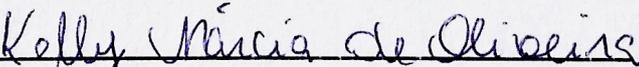
ATA DE DEFESA

No dia 5 do mês de Setembro de 2017, às 13:00 horas, na sala C304 do Instituto de Ciências Exatas e Aplicadas, foi realizada a defesa de Monografia pelo(a) aluno(a) **Kelly Márcia de Oliveira**, sendo a Comissão Examinadora constituída pelos professores: Prof. Fernando Bernardes de Oliveira, Prof. George Henrique Godim da Fonseca, Prof. Prof. Samuel Souza Brito. O(a) candidato(a) apresentou a monografia intitulada: "Um método baseado em Estratégia Evolutiva e *Variable Neighborhood Descent* para o Problema Quadrático de Alocação". A comissão examinadora deliberou, por unanimidade, pela aprovação do candidato, com nota 9,8 (doze pontos e oito décimos), concedendo-lhe o prazo de 15 dias para incorporação das alterações sugeridas ao texto final. Na forma regulamentar, foi lavrada a presente ata que é assinada pelos membros da Comissão Examinadora e pelo(a) graduando(a).


Prof. Fernando Bernardes de Oliveira
Doutor
DECSI - UFOP


Prof. George Henrique Godim da Fonseca
Doutor
Examinador(a)
DECSI - UFOP


Prof. Samuel Souza Brito
Mestre
Examinador(a)
DECSI - UFOP


Kelly Márcia de Oliveira

“Science is more than a body of knowledge; it is a way of thinking.”

— Carl Sagan (1934 – 1996),
in: The Demon-Haunted World: Science as a Candle in the Dark.

Resumo

O Problema Quadrático de Alocação (PQA) é bastante conhecido na área de otimização combinatória e possui ampla aplicação na indústria. O PQA faz parte da classe de problemas definidos na computação como NP-difícil, ou seja, para certas instâncias não é possível encontrar uma solução ótima em tempo computacional viável devido ao grande número de soluções a serem avaliadas. Assim, o uso de heurísticas torna-se fundamental, uma vez que essas indicam uma maneira de reduzir o número de avaliações necessárias para problemas com uma grande quantidade de soluções candidatas. O objetivo deste trabalho é o desenvolvimento de uma meta-heurística aplicável ao PQA por meio da Computação Evolucionária. Para isso, faz-se necessário o estudo do Problema Quadrático de Alocação, das estruturas de vizinhança, e algoritmos utilizados na resolução do problema. Este trabalho apresenta uma meta-heurística híbrida combinando Estratégias Evolutivas com a VND. Os resultados obtidos sugerem que não há diferença significativa entre as soluções do método proposto e da literatura, indicando que a meta-heurística proposta pode ser uma boa alternativa na busca pela melhor solução. Quando se compara a média das soluções obtidas, a literatura tende a apresentar um melhor resultado. Entretanto, a comparação dos valores médios do ES-VND com a literatura pode ser injusta, pois somente os melhores resultados estão disponíveis na literatura para comparação.

Palavras-chaves: Problema Quadrático de Alocação. Meta-heurística, Estratégia Evolutiva, *Variable Neighborhood Descent*.

Abstract

The Quadratic Assignment Problem (QAP) is well known in the field of combinatorial optimization and it is widely applied in the industry. Although it has a wide application, the QAP is part of the class of problems defined in computing as NP-Hard. That is, for some instances it is not possible to find an optimal solution in viable computational time due to a large number of solutions to be evaluated. Thus, the use of heuristics becomes feasible, since these indicate a way to reduce the number of evaluations required for problems with a large number of candidate solutions. The objective of this work is the development of a metaheuristic applied to QAP by means of Evolutionary Computing. For this, it is necessary to study the Quadratic Assignment Problem, the neighborhood structures, and algorithms used to solve the problem. This paper presents a hybrid metaheuristic combining Evolution Strategies with Variable Neighborhood Descent method. The results suggest that there is no significant difference between the solutions of the proposed method and the literature, indicating that the proposed metaheuristic can be a good alternative in the search for the best solution. When the average of the obtained solutions is compared, the literature tends to present a better result. However, the comparison of mean values of ES-VND with the literature may be unfair, since only the best results are available in the literature for comparison.

Key-words: Quadratic Assignment Problem. Metaheuristic. Evolution Strategy. Variable Neighborhood Descent.

Lista de ilustrações

Figura 1 – Diferença entre as médias dos níveis do fator	29
Figura 2 – <i>Bloxpot</i> – Comparação entre mínimos obtidos pelo ES-VND e literatura	32
Figura 3 – <i>Bloxpot</i> – Comparação entre o valor médio obtido pelo ES-VND e literatura	32
Figura 4 – Teste de Tukey para os menores valores obtidos entre o ES-VND e a literatura	33
Figura 5 – Teste de Tukey para os valores médios obtidos entre o método ES-VND e literatura	34

Lista de tabelas

Tabela 1 – Parâmetros	27
Tabela 2 – Definição das instâncias para teste de parametrização	28
Tabela 3 – ANOVA para Fatores e Valor P	28
Tabela 4 – Parâmetros	29
Tabela 5 – Definição das instâncias para testes de avaliação	30
Tabela 6 – GAP entre o menor resultado e o valor médio do método proposto e a literatura	31
Tabela 7 – ANOVA em relação ao menor e valor médio dos resultados	33

Lista de abreviaturas e siglas

ACO *Ant Colony Optimization*

ANOVA *Analysis of Variance*

ES *Estratégias Evolutivas*

FLP *Facility Layout Problem*

GA *Genetic Algorithms*

ILS *Iterated Local Search*

PLI *Programação Inteira*

PLIM *Programação Inteira Mista*

PLS *Pareto Local Search*

PQA *Problema Quadrático de Alocação*

PSO *Particle Swarm Optimization*

SA *Simulated Annealing*

TS *Tabu Search*

VND *Variable Neighborhood Descent*

VNS *Variable Neighborhood Search*

Lista de algoritmos

1	Estratégia Evolutiva ($\mu + \lambda$)	21
2	<i>Variable Neighborhood Descent</i>	22
3	ES-VND	23
4	<i>2-opt move</i>	25

Sumário

1	INTRODUÇÃO	14
1.1	Identificação do Problema	14
1.2	Objetivos	15
1.3	Justificativa	16
1.4	Organização do trabalho	16
2	REVISÃO BIBLIOGRÁFICA	17
2.1	Trabalhos correlatos	17
2.2	Aplicações do PQA	18
2.3	Considerações Finais	19
3	DESENVOLVIMENTO	20
3.1	Definição da representação do PQA	20
3.2	Definição de uma meta-heurística	20
3.3	Implementação	23
3.3.1	A classe Indivíduo	24
3.3.2	A classe Meta-heurística	24
3.3.3	A classe Movimento	24
3.4	Considerações Finais	25
4	EXPERIMENTOS COMPUTACIONAIS	26
4.1	Ambiente computacional	26
4.2	Parametrização	26
4.2.1	Definição dos parâmetros	28
4.3	Avaliação do método proposto	30
4.4	Resultados e análise	30
4.5	Considerações Finais	34
5	CONCLUSÃO	35
5.1	Propostas para trabalhos futuros	35
	REFERÊNCIAS	37

1 Introdução

O Problema Quadrático de Alocação (PQA) foi descrito por [Koopmans e Beckmann \(1957\)](#) e é definido por [Ahuja et al. \(2007\)](#) como o problema de alocar n facilidades para n localidades com o menor custo de interações entre facilidades. Problemas de agendamento, *layout* de fábricas e inserção de componentes eletrônicos em placas são algumas das aplicações do PQA. Além do mais, vários problemas da otimização combinatória como o problema do caixeiro viajante, partição de grafos e clique maximal podem ser interpretados como problemas quadráticos de alocação ([BURKARD, 2013](#)).

Embora tenha um grande número de aplicações, de acordo com [Ahuja et al. \(2007\)](#), o Problema Quadrático de Alocação é NP-difícil, e pode ser resolvido em tempo viável apenas para pequenas instâncias ($n \leq 30$). Assim, o uso de heurísticas é justificável, uma vez que essas indicam uma maneira de reduzir o número de avaliações necessárias para problemas com um grande número de soluções candidatas. Ainda que as heurísticas nem sempre encontrem a melhor solução para o problema em questão, elas realizam essa tarefa com frequência aceitável ([PEARL, 1984](#)).

O presente trabalho propõe uma meta-heurística híbrida, combinando Estratégias Evolutivas com a busca local *Variable Neighborhood Descent* na procura por soluções ótimas. Para avaliar a performance do método proposto, testes serão realizados por meio de instâncias de *benchmark* disponíveis na literatura.

A seguir, a Seção 1.1 descreve o Problema Quadrático de Alocação. Na Seção 1.2 são descritos os objetivos gerais e específicos do trabalho. A Seção 1.3 apresenta a justificativa para o desenvolvimento do trabalho e a Seção 1.4 descreve a organização deste documento.

1.1 Identificação do Problema

Proposto inicialmente por [Koopmans e Beckmann \(1957\)](#) como um problema de alocação de recursos indivisíveis e com ampla utilização na indústria, o PQA possui diversas formulações. [Loiola, Abreu e Netto \(2004\)](#) apresentam algumas das formulações usadas para representar o PQA. Entre elas estão a formulação por permutações, a formulação por grafos, as formulações por Programação Inteira Mista (PLIM) e as formulações por Programação Inteira (PLI), sendo a formulação por permutações escolhida para o desenvolvimento do projeto e descrita a seguir.

Seja S_n o conjunto de todas as permutações a n elementos, f_{ij} o fluxo entre as facilidades i e j , $d_{\pi(i)\pi(j)}$ a distância entre as localizações $\pi(i)$ e $\pi(j)$, dada pela permutação

$\pi \in S_n$. Se cada permutação π representar uma alocação de facilidades a localidades, o modelo matemático para o PQA é representado pela Equação 1.1. O objetivo é minimizar o fluxo e a distância entre as localidades.

$$\min_{\pi \in S_n} \sum_{\substack{i, \\ j=1}}^n f_{ij} d_{\pi(i)\pi(j)}. \quad (1.1)$$

O problema está sujeito às seguintes restrições:

1. Todas as facilidades serão alocadas.
2. Cada localidade deve alocar no máximo uma facilidade.

1.2 Objetivos

O objetivo geral deste trabalho é o estudo do Problema Quadrático de Alocação e as características relacionadas às estruturas de vizinhança. Um algoritmo baseado em técnicas de computação evolucionária será utilizado com a incorporação de busca local nessas estruturas.

Este trabalho possui os seguintes objetivos específicos:

1. Definir e implementar um algoritmo baseado em computação evolucionária para a geração de soluções para o Problema Quadrático de Alocação.
2. Estudar mecanismos de busca local para o problema.
3. Validar o algoritmo proposto por meio de experimentos computacionais utilizando instâncias de teste disponíveis na literatura.

Os passos para a execução deste trabalho são definidos a seguir:

1. Revisar a literatura sobre PQA e suas classes, trabalhos correlatos na literatura, e demais conceitos, bem como identificar métodos e técnicas utilizadas na solução desse tipo de problema;
2. Definir uma ou mais classes do PQA para estudo, com suas características gerais, aplicações e restrições utilizadas. Avaliar também as principais técnicas utilizadas na solução do problema;
3. Estudar métodos para representação do problema, identificando abordagens e estruturas de dados específicas para o PQA;

4. Definir e implementar uma meta-heurística para o problema baseada em técnicas de computação evolucionária;
5. Estudar e incorporar mecanismos de busca local aplicados ao problema;
6. Planejar e realizar experimentos para avaliar a performance da meta-heurística proposta;
7. Analisar e discutir os resultados obtidos, além de identificar possíveis melhorias e considerações gerais sobre o processo.

1.3 Justificativa

Fatores como o grande número de vizinhos a serem avaliados e a necessidade do cálculo de uma função objetivo quadrática para avaliar cada solução gerada dificultam a busca por um melhor resultado e fazem que apenas instâncias de ordem $n \leq 30$ sejam resolvidas em tempo computacional viável. Por estes motivos, o [PQA](#) faz parte da classe de problemas conhecidos na computação como NP-difícil.

Uma vizinhança pode ser entendida como o resultado da troca de posição de duas ou mais facilidades no [PQA](#). Geralmente, quanto maior a vizinhança, melhor a precisão do resultado final obtido. Todavia, quanto maior a vizinhança, maior o tempo de execução do algoritmo [Ahuja, Orlin e Tiwari \(2000\)](#). Sendo assim, este trabalho justifica-se pela necessidade de enumerar bons vizinhos heurísticamente, já que a tarefa de gerar e avaliar uma grande quantidade de vizinhos é muito custosa.

1.4 Organização do trabalho

Este trabalho é organizado como segue. O [Capítulo 2](#) contém a revisão bibliográfica referente ao Problema Quadrático de Alocação, no qual são introduzidas algumas técnicas utilizadas na resolução do problema por meio do estudo de trabalhos correlatos. O [Capítulo 3](#) introduz e descreve o novo método proposto. No [Capítulo 4](#) são apresentados os resultados obtidos enquanto o [Capítulo 5](#) traz uma discussão sobre a meta-heurística desenvolvida e possíveis melhoras para o algoritmo.

2 Revisão bibliográfica

O Problema Quadrático de Alocação é bastante conhecido na área de otimização combinatória e possui ampla aplicação na indústria. Existem diversos trabalhos na literatura que fornecem soluções para o PQA por meio de diferentes métodos. [Loiola, Abreu e Netto \(2004, p. 86\)](#) afirmam que “*com o surgimento das meta-heurísticas, o Problema Quadrático de Alocação ganhou um novo impulso, dado que toda meta-heurística ao ser implementada usa o PQA como elemento de prova de sua eficiência*”.

A Seção 2.1 apresenta trabalhos correlatos na literatura. Algumas aplicações do PQA são apresentadas na Seção 2.2.

2.1 Trabalhos correlatos

A Têmpera Simulada (do inglês, *Simulated Annealing – SA*), proposta por [Metropolis et al. \(1953\)](#), remete ao processo usado na metalurgia para temperar ou endurecer metais e vidros, conhecido como têmpera. Neste procedimento os materiais citados são aquecidos em temperatura elevada e depois arrefecidos gradualmente, o que faz com que os sólidos alcancem um estado de baixa energia. De acordo com [Russell e Norvig \(2004\)](#), a Têmpera Simulada pode ser entendida como uma versão da Subida de Encosta Estocástica (*Stochastic Climbing Hill*). Porém, ao contrário da Subida de Encosta, a Têmpera Simulada admite movimentos que pioram a situação com certa probabilidade. Movimentos encosta abaixo são permitidos com menor frequência no decorrer do tempo. [Misevičius \(2003\)](#) introduz um algoritmo de SA modificado na resolução do PQA que mostrou-se superior à versões da Têmpera Simulada utilizadas anteriormente.

Os Algoritmos Genéticos (do inglês, *Genetic Algorithms – GA*), propostos por [Holland \(1975\)](#), são uma analogia à seleção natural. Geralmente, o problema é representado por uma cadeia de *bits* de tamanho n denominada indivíduo, embora outras representações sejam aceitas. No Algoritmo Genético descrito por [Russell e Norvig \(2004\)](#) é gerada uma população inicial de indivíduos. Em seguida dois indivíduos são selecionados nessa população para a operação de *crossover*, sendo que cada indivíduo possui uma probabilidade calculada de ser escolhido. No *crossover*, partes dos indivíduos eleitos são mescladas e dão origem à próxima geração. Por último, a mutação, operação que altera um *bit* do indivíduo com probabilidade p , é aplicada nos indivíduos gerados. As operações de *crossover* e mutação são repetidas até que se atinja o critério de parada. [Tate e Smith \(1995\)](#) utilizam Algoritmos Genéticos na resolução do PQA e concluem que este método tem performance igual ou melhor do que heurísticas conhecidas para o PQA.

A Busca Tabu (do inglês, *Tabu Search* – TS), proposta por Glover (1986), utiliza o conceito de tabu (algo sagrado, que não pode ser tocado). Luke (2013) apresenta um algoritmo para a TS no qual uma lista tabu de tamanho predeterminado é mantida pela busca. Essa lista contém os candidatos à solução visitados recentemente e o algoritmo recusa-se a visitar esses candidatos por um determinado tempo, por isso o nome lista tabu. Essa abordagem amplia o espaço de busca, uma vez que evita que o algoritmo fique preso em um mínimo (ou máximo) local. James, Rego e Glover (2009) propõem uma busca tabu paralela cooperativa para o PQA, a qual encontrou soluções de boa qualidade para instâncias conhecidas do PQA.

Algoritmos de Colônia de Formigas (do inglês, *Ant Colony Optimization* – ACO), propostos por Dorigo (1992), são definidos por Dorigo e Stützle (2004) como uma meta-heurística na qual uma colônia de formigas artificiais cooperam para encontrar boas soluções para problemas de otimização discretos. A cooperação é essencial para ACO. As formigas artificiais se comunicam por meio de informações deixadas nos caminhos visitados, o que resulta em boas soluções para o problema em questão. Saremi, Abedin e Kermani (2008) aplicam o algoritmo de colônias de formiga ao PQA para melhorar a estrutura de *links* e navegação em um *website*.

Ainda é possível aplicar algoritmos híbridos ao PQA. Chiang et al. (2011) utilizam a Busca Tabu e a Otimização por Enxame de Partículas (*Particle Swarm Optimization*) em uma extensão do PQA conhecida como *Facility Layout Problem*. Qi, Tian e Sun (2009) propõem a combinação da Colônia de Formigas com a *Pareto Local Search*, em que um conjunto de soluções é gerado pela ACO e aprimorado pela PLS. Xia (2009) realiza um trabalho similar ao de Qi, Tian e Sun (2009), utilizando o Algoritmo de Colônia de Formigas e a *Iterated Local Search*.

Muitos dos algoritmos, inclusive TS e SA citados acima, utilizam a estrutura de vizinhança na busca por uma solução do PQA. Deve-se então buscar trabalhos na literatura que forneçam métodos eficientes para a busca de vizinhança. Luong, Melab e Talbi (2010) utilizam a busca paralela na unidade de processamento gráfico ou GPU, obtendo assim uma maior velocidade no processo de busca. Campeotto et al. (2014) realizam um trabalho correlato ao de Luong, Melab e Talbi (2010), o qual apresenta um *framework* que utiliza paralelismo na busca em vizinhanças.

2.2 Aplicações do PQA

Existem alguns trabalhos na literatura em que podem ser encontradas aplicações práticas do Problema Quadrático de Alocação. Steinberg (1961) utilizou o Problema Quadrático de Alocação para minimizar o total de conexões em uma placa de circuitos eletrônicos. Dickey e Hopkins (1972) discutem o planejamento de um *campus* com o

objetivo de minimizar a distância caminhada entre os prédios. [Geoffrion e Graves \(1976\)](#) descrevem uma aplicação prática para o escalonamento de horários em um reator nuclear utilizando o [PQA](#).

Estes foram somente alguns exemplos de aplicações do Problema Quadrático de Alocação. [Loiola, Abreu e Netto \(2004, p.74\)](#) e [Burkard \(2013, p.2744\)](#) fornecem outras aplicações para o problema.

2.3 Considerações Finais

Por meio da revisão da literatura referente ao Problema Quadrático de Alocação foi possível identificar os principais conceitos relacionados ao [PQA](#), observar algumas aplicações práticas e diagnosticar os desafios enfrentados na resolução do problema. Várias abordagens foram propostas ao longo do tempo utilizando meta-heurísticas conhecidas na área de otimização combinatória, sempre comparando os resultados obtidos com outros trabalhos presentes na literatura. O próximo capítulo apresenta a definição e conceitos da meta-heurística desenvolvida neste trabalho.

3 Desenvolvimento

Para o desenvolvimento do trabalho optou-se pela representação por permutação do PQA e a adoção das Estratégias Evolutivas como base para o algoritmo proposto juntamente com a *Variable Neighborhood Descent*. A definição da representação por permutação encontra-se na Seção 3.1, enquanto a Seção 3.2 aborda os principais conceitos da meta-heurística construída. A Seção 3.3 fornece detalhes sobre a implementação do algoritmo.

3.1 Definição da representação do PQA

Uma das possíveis representações para o PQA é a permutação de n números em um conjunto $N = \{1, 2, \dots, n\}$, na qual o i -ésimo número na permutação representa a facilidade atribuída a localidade i Ahuja, Orlin e Tiwari (2000). Esta representação será utilizada na implementação do algoritmo.

3.2 Definição de uma meta-heurística

Após o estudo das abordagens utilizadas no PQA bem como suas representações, foi decidido que este trabalho utilizará o método de Estratégias Evolutivas (do inglês, *Evolution Strategies – ES*). As ES podem ser classificadas como métodos baseados em população, na qual um grupo de soluções candidatas são avaliadas na busca por uma solução. Essa abordagem permite o descarte de boas soluções que geram descendentes ruins ou ainda sua modificação de forma que gerem melhores soluções (LUKE, 2013).

Como mecanismo de evolução será utilizada a estratégia $(\mu + \lambda)$ descrita em Luke (2013) na resolução do Problema Quadrático de Alocação, na qual μ representa o número de pais e λ o número de filhos gerados pelos pais. A próxima geração é composta pelos μ melhores indivíduos selecionados do conjunto formado por $(\mu + \lambda)$. O pseudocódigo para a Estratégia Evolutiva $(\mu + \lambda)$, disponível em Luke (2013), é apresentado no Algoritmo 1.

Algoritmo 1 Estratégia Evolutiva ($\mu + \lambda$)

```

1:  $\mu \leftarrow$  número de pais selecionados
2:  $\lambda \leftarrow$  número de filhos gerados pelos pais
3:  $P \leftarrow \{\}$ 
4: //Gera população inicial
5: for  $\lambda$  vezes do
6:    $P \leftarrow P \cup \{\text{novo indivíduo aleatório}\}$ 
7:  $Melhor \leftarrow \emptyset$ 
8: //Busca por uma solução enquanto o critério de parada
9: //não for atingido
10: repeat
11:   //Avalia cada indivíduo de acordo com a aptidão calculada
12:   for cada indivíduo  $P_i \in P$  do
13:      $CalcularAptidão(P_i)$ 
14:     if  $Melhor = \emptyset$  ou  $Aptidão(P_i) \geq Aptidão(Melhor)$  then
15:       //O melhor indivíduo da geração é selecionado
16:        $Melhor \leftarrow P_i$ 
17:   //Os  $\mu$  melhores indivíduos da geração são selecionados
18:    $Q \leftarrow$  os  $\mu$  melhores indivíduos em  $P$  cuja aptidão seja a maior
19:    $P \leftarrow Q$ 
20:   //Cada indivíduo da geração é submetido ao processo
21:   //de alteração
22:   for cada indivíduo  $Q_j \in Q$  do
23:     for  $\lambda/\mu$  vezes do
24:       //A nova geração é formada pelos  $\mu$  melhores indivíduos
25:       //e novos indivíduos resultantes do processo de
26:       //alteração
27:        $P \leftarrow P \cup \{Alterar(Copiar(Q_j))\}$ 
28: until  $Melhor$  é a melhor solução ou o tempo acabou
29: //Retorna o melhor valor encontrado
30: return  $Melhor$ 

```

Ainda é possível refinar os melhores indivíduos selecionados na ES aplicando algoritmos que utilizam a estrutura de vizinhança. O método escolhido foi o *Variable Neighborhood Descent* (VND), uma variação da meta-heurística *Variable Neighborhood Search* (VNS) proposta por Mladenović e Hansen (1997), que permite explorar vizinhanças distantes. O algoritmo parte do princípio que um ótimo global é um ótimo local comum a todas vizinhanças. Nesse caso é mais provável que se encontre o ótimo global se uma maior quantidade de vizinhos forem explorados. O Algoritmo 2 apresenta o pseudocódigo para a meta-heurística VND baseado no trabalho de (QU; XU; KENDALL, 2009).

Algoritmo 2 *Variable Neighborhood Descent*

```
1: //A melhor solução recebe a solução inicial
2:  $Solução_{melhor} \leftarrow Solução_{atual}$ 
3: //Incrementa a iteração
4:  $k \leftarrow 1$ 
5: //Enquanto o número máximo de tentativas  $k_{max}$ 
6: //não for atingido
7: while  $k \leq k_{max}$  do
8:   //Um novo vizinho da melhor solução é gerado
9:    $Solução \leftarrow gerarVizinho(Solução_{melhor})$ 
10:  //Se o novo vizinho for melhor que a solução atual
11:  //adota-se esse vizinho
12:  if  $Fitness(Solução) \leq Fitness(Solução_{melhor})$  then
13:     $Solução_{melhor} \leftarrow Solução$ 
14:    //Continua buscando por melhores vizinhos
15:     $k \leftarrow 1$ 
16:  else
17:    //Se não houve melhora no vizinho incrementa-se
18:    //o número de tentativas
19:     $k \leftarrow k + 1$ 
20: //Retorna a melhor solução encontrada
21: return  $Solução_{melhor}$ 
```

O método proposto para implementação consiste, portanto, em uma meta-heurística híbrida que se beneficia das vantagens proporcionadas pela Estratégia Evolutiva e a *Variable Neighborhood Descent*. O pseudocódigo para abordagem proposta trata-se de uma junção dos algoritmos apresentados por Qu, Xu e Kendall (2009) e Luke (2013) e é apresentado no Algoritmo 3.

Algoritmo 3 ES-VND

```

1:  $\mu \leftarrow$  número de pais selecionados
2:  $\lambda \leftarrow$  número de filhos gerados pelos pais
3: TaxaMutaçao  $\leftarrow$  probabilidade de mutação de um indivíduo
4: TaxaBuscaLocal  $\leftarrow$  probabilidade de que ocorra busca local em um indivíduo
5:  $P \leftarrow \{\}$ 
6: //Gera população inicial
7: for  $\lambda$  vezes do
8:    $P \leftarrow P \cup \{\text{novο indivíduo aleatório}\}$ 
9: Melhor  $\leftarrow \emptyset$ 
10: //Busca por uma solução enquanto o critério de parada
11: //não for atingido
12: repeat
13:   //Avalia cada indivíduo de acordo com a aptidão calculada
14:   for cada indivíduo  $P_i \in P$  do
15:     CalcularAptidao( $P_i$ )
16:     if Melhor =  $\emptyset$  ou Aptidao( $P_i$ )  $\geq$  Aptidao(Melhor) then
17:       //O melhor indivíduo da geração é selecionado
18:       Melhor  $\leftarrow P_i$ 
19:   //Os  $\mu$  melhores indivíduos da geração são selecionados
20:    $Q \leftarrow$  os  $\mu$  melhores indivíduos em  $P$  cuja aptidão seja a maior
21:    $P \leftarrow Q$ 
22:   //Cada indivíduo da geração é submetido ao processo
23:   //de alteração e busca local
24:   for cada indivíduo  $Q_j \in Q$  do
25:     for  $\lambda/\mu$  vezes do
26:       Porcentagem  $\leftarrow$  Número aleatório entre 0 e 100
27:       if Porcentagem < TaxaMutaçao then
28:         //indivíduo é submetido ao processo de mutação
29:          $Q_j \leftarrow \text{Alterar}(\text{Copiar}(Q_j))$ 
30:       Porcentagem  $\leftarrow$  Número aleatório entre 0 e 100
31:       if Porcentagem < TaxaBuscaLocal then
32:         //indivíduo é submetido ao processo de busca local
33:          $Q_j \leftarrow \text{VND}(Q_j)$ 
34:       //A nova geração é formada pelos  $\mu$  melhores indivíduos
35:       //e novos indivíduos resultantes do processo de
36:       //alteração e mutação
37:        $P \leftarrow P \cup \{(Q_j)\}$ 
38: until Melhor é a melhor solução ou o tempo acabou
39: //Retorna o melhor valor encontrado
40: return Melhor

```

3.3 Implementação

A linguagem Java, versão 8, foi escolhida para a implementação do método proposto, pois é uma linguagem simples, portátil, robusta e de alta performance. O programa

desenvolvido é composto por três classes principais, a classe Indivíduo, a classe Meta-Heurística e a classe Movimento descritas a seguir.

3.3.1 A classe Indivíduo

A classe Indivíduo tem como atributos os genes, uma lista encadeada de inteiros que representa uma permutação do PQA, e o *fitness* que indica o custo da uma permutação. A classe também contém métodos que auxiliam na execução da meta-heurística.

3.3.2 A classe Meta-heurística

A classe Meta-heurística é a principal classe do programa e responsável por executar a maioria das operações que levam ao resultado. O pseudocódigo descrito no Algoritmo 3 demonstra a ordem de execução do programa. Entre os métodos da classe que merecem destaque estão:

- Para gerar um novo indivíduo, Linha 8, utiliza-se o método *shuffle* disponível pela linguagem Java, criando-se, assim, uma nova combinação a partir de uma lista de inteiros informada.
- O método *calcularAptidão*, Linha 15, calcula a aptidão de um indivíduo utilizando a Equação 1.1.
- O método *Alterar*, Linha 29, da classe Meta-heurística é responsável pela mutação de um indivíduo. Para isso, a quantidade de genes que serão modificados é gerada aleatoriamente e então realiza-se operação de troca entre os genes selecionados.
- O método *VND*, Linha 33, executa uma série de movimentos com objetivo de melhorar um indivíduo enquanto o número máximo de tentativas não for atingido. Os movimentos implementados são descritos na Seção 3.3.3.

3.3.3 A classe Movimento

A classe Movimento contém uma série de movimentos baseados no trabalho de Prins (2004) usados pela busca VND para melhorar um indivíduo. Dadas duas localidades u e v do Problema Quadrático de Alocação, e duas localidades x e y sucessoras de u e v respectivamente, cada movimento é descrito a seguir:

- Movimento 1: Remover a facilidade da localidade u e inserir depois de v .
- Movimento 2: Remover as facilidades em u e x e inserir na ordem (u, x) depois de v .
- Movimento 3: Remover as facilidades em u e x e inserir na ordem (x, u) depois de v .

- Movimento 4: Trocar as facilidades entre u e v .
- Movimento 5: Trocar as facilidades entre (u, x) e v .
- Movimento 6: Trocar as facilidades entre (u, x) e (v, y) .
- Movimento 7: *2-opt move*.

O *2-opt move*, proposto por Croes (1958), consiste em aplicar “inversões” em uma solução com o propósito de obter um novo resultado de menor custo. Os passos descritos por He e Xiang (2017) foram empregados na implementação do *2-opt move*, e são ilustrados no Algoritmo 4.

Algoritmo 4 *2-opt move*

```

1: //Armazena-se o tamanho da solução atual
2:  $tamanho \leftarrow obterTamanho(soluçãoAtual)$ 
3: //Insere as facilidades 1 até  $u - 1$  em ordem na nova solução
4: for  $i = 0$  to  $u - 1$  do
5:    $novaSolução(i) \leftarrow soluçãoAtual(i)$ 
6: //Auxiliar para o processo de inversão
7:  $j \leftarrow 0$ 
8: //Insere as facilidades  $u$  até  $v$  em ordem reversa na nova solução
9: for  $i = u$  to  $v$  do
10:   $novaSolução(i) \leftarrow soluçãoAtual(v - j)$ 
11:   $j \leftarrow j + 1$ 
12: //Insere o restante das facilidades em ordem na nova solução
13: for  $i = v + 1$  to  $tamanho$  do
14:   $novaSolução(i) \leftarrow soluçãoAtual(i)$ 
return  $novaSolução$ 

```

3.4 Considerações Finais

A Estratégia Evolutiva ($\mu + \lambda$) foi a escolhida como algoritmo base para aplicação no projeto, visto que a computação evolucionária apresenta vantagens como simplicidade da abordagem, robustez, dentre outras. O método *Variable Neighborhood Descent* foi incorporado à meta-heurística já que permite explorar um maior número de vizinhanças na busca pela melhor solução. O capítulo seguinte apresenta os resultados obtidos e provém a análise e discussão do método proposto.

4 Experimentos Computacionais

Este capítulo descreve os experimentos realizados para determinar os valores dos parâmetros utilizados e identifica se existe alguma diferença estatisticamente significativa entre eles. Também é feita a avaliação do método proposto comparando os resultados com os disponíveis na literatura. As instâncias de teste escolhidas pertencem à biblioteca QAPLib¹, já que a biblioteca proporciona diversas instâncias do PQA e suas soluções, sendo amplamente utilizada na literatura revisada. A implementação do ES-VND e os dados coletados durante o experimento podem ser encontrados no seguinte endereço: <<https://github.com/kmdeoliv/TCC>>.

A seguir, a Seção 4.1 apresenta o ambiente computacional de execução dos testes, a Seção 4.2 descreve como os testes de parametrização do problema foram realizados e discute os resultados obtidos, enquanto a Seção 4.3 apresenta os testes realizados para a avaliação da qualidade do método proposto e a análise dos resultados.

4.1 Ambiente computacional

Todos os experimentos foram executados em um Desktop Dell modelo Vostro 260S com processador Intel®Core(TM) i3-2120 CPU @ 3.30GHz e 4GB de memória RAM com sistema operacional *Elementary OS 64 bits*. Durante todo o processamento apenas a aplicação desenvolvida foi executada para que não houvesse interferência nos resultados obtidos.

4.2 Parametrização

A primeira etapa dos testes consiste em determinar qual a combinação de valores dos parâmetros proporciona os melhores resultados na amostra selecionada. Os experimentos podem ser classificados como um Planejamento Fatorial 2^n com blocagem, pois, de acordo com Montgomery, Runger e Calado (2000), esse experimento permite estudar o efeito conjunto dos fatores (ou parâmetros) no resultado, sendo que cada fator apresenta dois níveis (ou valores) quantitativos. Os fatores e os níveis a serem estudados são apresentados na Tabela 1.

¹ <<http://anjos.mgi.polymtl.ca/qaplib/>>

Tabela 1 – Parâmetros

Fatores	Níveis	Descrição
μ	5 – 10	Determina o tamanho do conjunto com as melhores soluções.
λ	30 – 100	Determina o número de soluções que serão geradas para avaliação.
Tempo	3 – 10 min	Critério de parada de execução do algoritmo.
Taxa de Mutação	0,4 – 0,7	Probabilidade que uma solução tem de sofrer o processo de mutação.
Taxa de Busca Local	0,4 – 0,7	Probabilidade que uma solução tem de ser submetida ao processo de melhora.
Iterações Sem Melhora	5 – 10	Critério de parada para a busca local.

Fonte: Produzido pelo autor

Foram selecionadas dez instâncias de teste com tamanhos distintos da QAPLib. Para que os diferentes tamanhos de cada instância não influenciassem nos resultados, utilizou-se a blocagem por esse fator. As características de cada instância selecionada estão presentes na Tabela 2. O experimento consiste em 64 observações (seis fatores com dois níveis cada). Para cada configuração o algoritmo foi replicado cinco vezes. Considerando as 64 observações, as dez instâncias e as cinco replicações, o experimento possui 3.200 observações totais. Para obter os resultados e evitar qualquer efeito no experimento, a ordem de execução das observações foi aleatorizada.

Tabela 2 – Definição das instâncias para teste de parametrização

Instância	Número de facilidades
bur26a	26
chr12c	12
els19	19
esc32g	32
had18	18
kra32	32
lipa90a	90
nug20	20
nug30	30
tai25a	25

Fonte: Produzido pelo autor

4.2.1 Definição dos parâmetros

A Tabela 3 contém os resultados da análise de variância, (do inglês, *Analysis of Variance-ANOVA*). O intervalo de confiança utilizado é de 95%, o que implica em um nível de significância $\alpha = 0,05$. Os valores destacados em negrito na tabela são menores que o nível de significância utilizado. A observação do Valor P sugere que os parâmetros *Tempo*, *Taxa de Mutação* e *Iterações Sem Melhora* são os que mais interferem na qualidade da solução obtida ($Valor P \leq \alpha$).

Tabela 3 – ANOVA para Fatores e Valor P

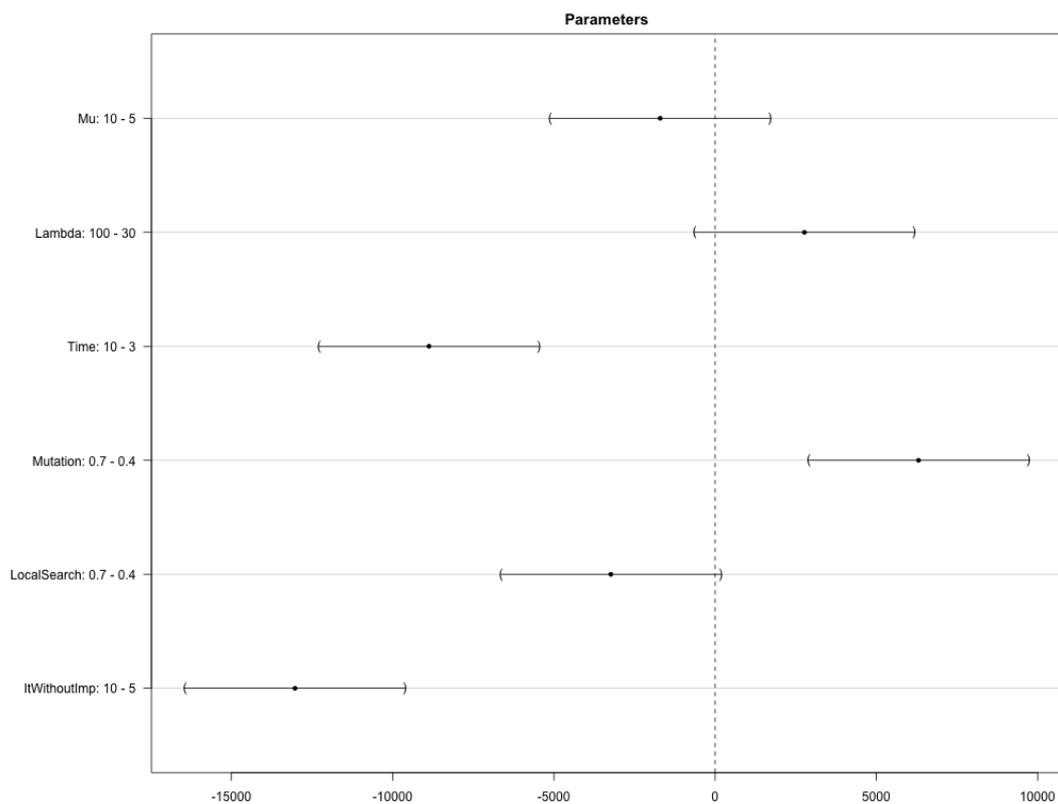
Fatores	Valor P
μ	0,1899
λ	0,0331
Tempo	1, 02⁻¹¹
Taxa de Mutação	1, 26⁻⁰⁶
Taxa de Busca Local	0,0128
Iterações Sem Melhora	< 2⁻¹⁶
$\mu:\lambda$:Taxa de Busca Local	0,0273

Fonte: Produzido pelo autor

Considerando que existe diferença estatisticamente significativa entre os fatores no contexto experimental apresentado, a próxima etapa é identificar qual o nível que deverá ser utilizado. Para isso é aplicado um teste de comparações múltiplas. O teste escolhido foi o Teste de Tukey, um método proposto por [Tukey \(1953\)](#), o qual sugere os valores ideais para cada parâmetro. A Figura 1 representa os resultados obtidos pelo teste. Considerando um nível de significância de 5%, para parâmetros cujos valores apresentam-se

totalmente à esquerda da média adota-se o maior nível, enquanto que para parâmetros cujos valores apresentam-se totalmente à direita da média adota-se o menor nível, pois há diferença significativa entre eles, ou seja, o nível escolhido interfere no resultado final. Para os parâmetros cujos valores encontram-se sobre a média não há diferença significativa, podendo-se escolher qualquer nível. Neste caso, utilizou-se o nível para que o parâmetro tendia. A Tabela 4 apresenta os parâmetros adotados e seus respectivos valores.

Figura 1 – Diferença entre as médias dos níveis do fator



Fonte: Produzido pelo autor

Tabela 4 – Parâmetros

Parâmetro	Valor
μ	10
λ	30
Tempo	10 min
Taxa de Mutação	0,4
Taxa de Busca Local	0,7
Iterações Sem Melhora	10

Fonte: Produzido pelo autor

4.3 Avaliação do método proposto

A segunda etapa dos testes permite avaliar a qualidade do algoritmo proposto utilizando os parâmetros definidos. Para este experimento foram utilizadas 20 instâncias com solução ótima conhecida, incluindo aquelas utilizadas na parametrização do problema. Considerando 20 observações (instâncias) e que cada observação foi replicada 10 vezes, o experimento possui um total de 200 observações. As instâncias utilizadas nessa fase de teste e os trabalhos que apresentam as soluções ótimas ² encontram-se na Tabela 5 e a análise dos resultados obtidos encontram-se na Seção 4.4.

Tabela 5 – Definição das instâncias para testes de avaliação

Instância	Número de facilidades	Referências
Bur26a	26	Pardalos e Resende (1994)
Bur26h	26	Pardalos e Resende (1994)
Chr12c	12	Christofides e Benavent (1989)
Chr25a	25	Christofides e Benavent (1989)
Els19	19	Mautor (1993)
Esc16a	16	Clausen e Perregaard (1997)
Esc32g	32	Brungger et al. (1997)
Esc128	128	Fischetti, Monaci e Salvagnin (2012)
Had16	16	Hahn, Grant e Hall (1998)
Had18	18	Clausen e Perregaard (1997)
Had20	20	Clausen e Perregaard (1997)
Kra30b	20	Anstreicher et al. (2002)
Kra32	32	Anstreicher et al. (2002)
Lipa60a	60	Li e Pardalos (1992)
Lipa90a	90	Li e Pardalos (1992)
Nug20	20	Clausen e Perregaard (1997)
Nug24	24	Clausen et al. (1996)
Nug30	30	Anstreicher e Brixius (2001)
Tai25a	25	Hahn, Grant e Hall (1998)
Tai64c	64	Drezner (2006)

Fonte: Produzido pelo autor

4.4 Resultados e análise

A primeira fase para análise dos resultados consiste em observar o GAP entre as soluções obtidas e as soluções encontradas na literatura. O cálculo do GAP é definido pela Equação 4.1.

$$GAP(\%) = \frac{\text{Solução do ES-VND} - \text{Solução da Literatura}}{\text{Solução da Literatura}} \times 100 \quad (4.1)$$

² As referências com as soluções ótimas estão definidas na QAPLib.

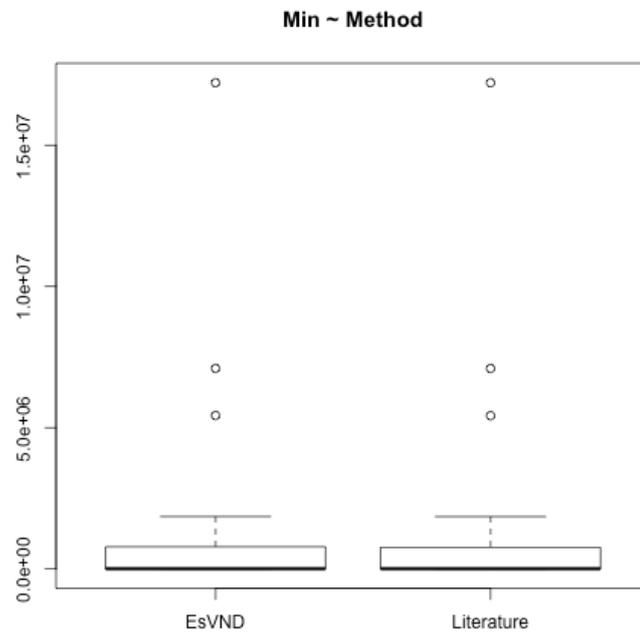
Nota-se na Tabela 6 que o GAP para o menor valor encontrado pelo método apresenta melhores valores em relação aos valores médios. Isso sugere que, ao menos uma vez, durante as dez replicações, foram encontrados valores próximos aos da literatura. Dentre as instâncias observadas, destacam-se os melhores resultados obtidos com as instâncias Chr12c, Esc16a e Esc32g, nas quais não houve GAP tanto para o mínimo quanto para o valor médio. Destacam-se também as instâncias Had16, Had18 e Had 20, nas quais não houve GAP para o menor valor encontrado. As instâncias Chr25a e Esc18 apresentaram os piores resultados para o mínimo e o valor médio, com valores muito superiores ao restante da amostra. Acredita-se que o tamanho da instância Esc128 e particularidades da instância Chr25 podem influenciar no resultado.

Tabela 6 – GAP entre o menor resultado e o valor médio do método proposto e a literatura

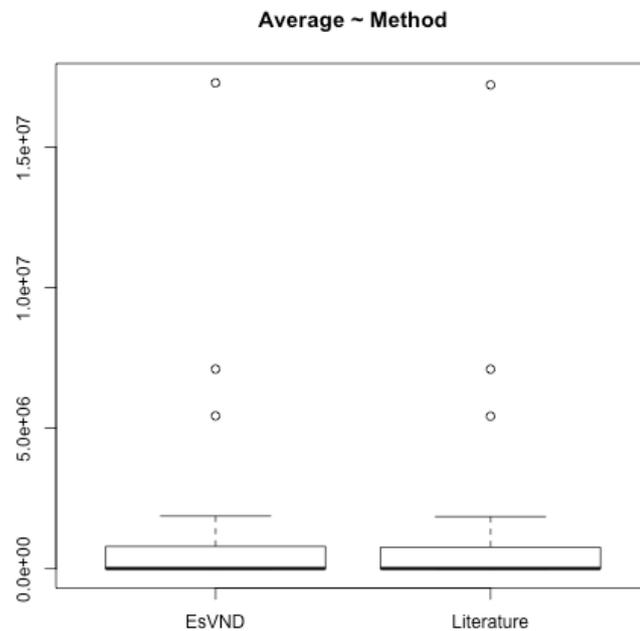
Instância	Literatura	Menor	GAP%	Valor Médio	GAP%
Bur26a	5426670	5433008	0,12	5438856,6	0,22
Bur26h	7098658	7099924	0,02	7103168,3	0,06
Chr12c	11156	11156	0,00	11156	0,00
Chr25a	3796	5046	32,93	5774,8	52,13
Els19	17212548	17212548	0,00	17284270,6	0,42
Esc128	64	100	56,25	119,4	86,56
Esc16a	68	68	0,00	68	0,00
Esc32g	6	6	0,00	6	0,00
Had16	3720	3720	0,00	3721,4	0,04
Had18	5358	5358	0,00	5367,6	0,18
Had20	6922	6922	0,00	6930,2	0,12
Kra30b	91420	94930	3,84	97538	6,69
Kra32	88900	94360	6,14	97083	9,20
Lipa60a	107218	108702	1,38	108768,4	1,45
Lipa90a	360630	364335	1,03	364747,1	1,14
Nug20	2570	2574	0,15	2627,2	2,22
Nug24	3488	3532	1,26	3613,6	3,60
Nug30	6124	6242	1,93	6414	4,73
Tai25a	1167256	1210436	3,70	1226917,6	5,11
Tai64c	1855928	1857646	0,09	1873497,8	0,95
Média	1672625	1676030,65	0,002	1682032,28	0,006

Fonte: Produzido pelo autor

A análise do *boxplot* para o menor valor obtido (*Min*, Figura 2) e para o valor médio (*Average*, Figura 3) sugere que a diferença entre o método proposto e a literatura não é aparentemente observável. Alguns *outliers* foram encontrados tanto no método proposto quanto na literatura, o que pode ser explicado pela diversidade das instâncias escolhidas, nas quais encontram-se custos muito abaixo ou acima da média observada.

Figura 2 – *Bloxpot* – Comparação entre mínimos obtidos pelo ES-VND e literatura

Fonte: Produzido pelo autor

Figura 3 – *Bloxpot* – Comparação entre o valor médio obtido pelo ES-VND e literatura

Fonte: Produzido pelo autor

A análise estatística dos resultados foi realizada por meio da análise de variância. Os resultados obtidos encontram-se na Tabela 7. O intervalo de confiança utilizado é de 95%, o que implica em um nível de significância $\alpha = 0,05$. Considerando o contexto experimental estabelecido, os resultados sugerem que existem uma diferença estatística

para os valores médios entre o método proposto e a literatura. Para os valores mínimos, os resultados sugerem que não existe diferença.

Tabela 7 – ANOVA em relação ao menor e valor médio dos resultados

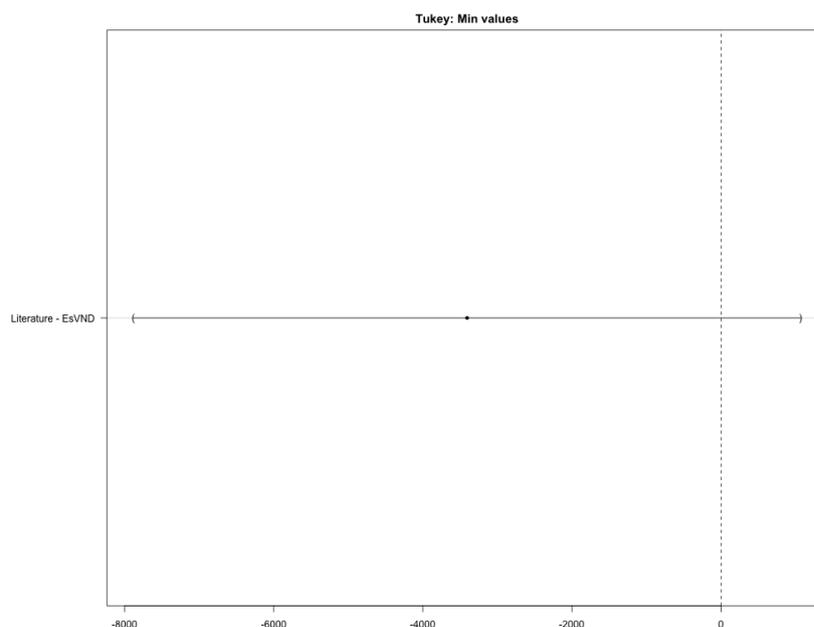
Fator	Valor P
Mínimo	0,1280
Valor médio	0,0482

Fonte: Produzido pelo autor

Para identificar se a diferença é em favor do ES-VND ou da literatura, utilizou-se o Teste de Tukey. Considerando um nível de significância de 5%, os resultados sugerem que não há diferença estatística entre os valores mínimos obtidas pelo ES-VND e os valores reportados na literatura. Esse resultado é apresentado na Figura 4, e é mais um indício que as observações acerca do Valor P estão corretas. Esses resultados sugerem que o método proposto pode ser aperfeiçoado com a incorporação de outras estratégias de busca local.

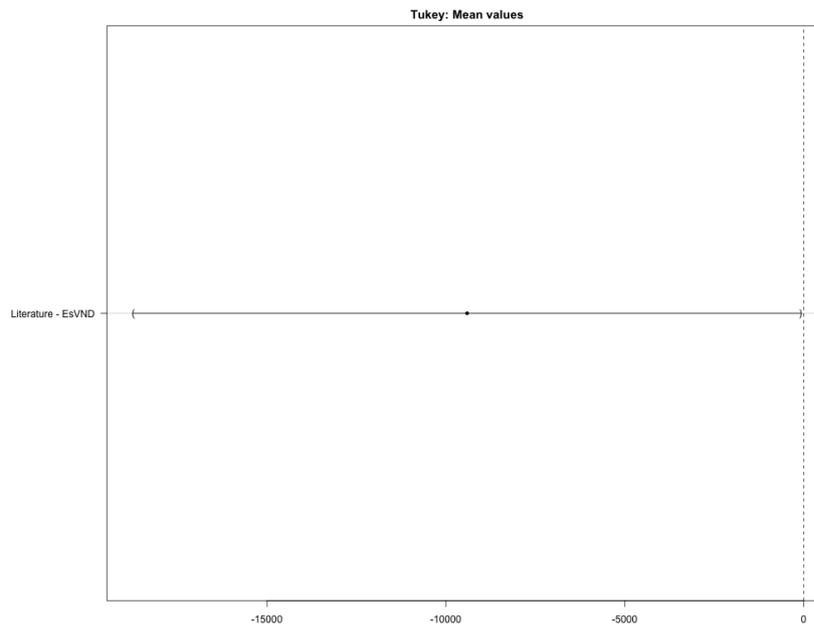
A Figura 5 apresenta a comparação entre os valores médios do ES-VND e a literatura. Os resultados sugerem que o ES-VND possui valores relativamente inferiores aos da literatura. Entretanto, a comparação dos valores médios do ES-VND com a literatura pode ser injusta, pois os valores médios dos resultados não são reportadas. Somente os melhores resultados estão disponíveis para comparação.

Figura 4 – Teste de Tukey para os menores valores obtidos entre o ES-VND e a literatura



Fonte: Produzido pelo autor

Figura 5 – Teste de Tukey para os valores médios obtidos entre o método ES-VND e literatura



Fonte: Produzido pelo autor

4.5 Considerações Finais

O planejamento experimental foi fundamental para o estudo do problema, pois foi possível estabelecer de maneira adequada os valores dos parâmetros necessários para o método proposto por meio da análise de variância ANOVA e o Teste de Tukey. Os testes também indicam que não há diferença para a melhor solução disponível na literatura e a fornecida pelo método ES-VND. Foram encontradas soluções iguais ou muito próximas às da literatura, sugerindo que a meta-heurística proposta pode ser uma boa alternativa na busca pela melhor solução. Todavia, a análise ANOVA e o Teste de Tukey revelam que a literatura provém melhores valores médios para as soluções quando comparada com o método apresentado. Porém, deve-se considerar que somente o melhor resultado é conhecido na literatura consultada. O que pode resultar em uma comparação injusta. As conclusões referentes ao trabalho desenvolvido e as propostas para trabalhos futuros encontram-se no próximo capítulo.

5 Conclusão e trabalhos futuros

O presente trabalho teve como objetivo o desenvolvimento de uma meta-heurística usando técnicas da Computação Evolucionária e Busca Local aplicável ao Problema Quadrático de Alocação e a avaliação da qualidade do método proposto a partir das soluções geradas. O maior desafio encontrado foi a busca por métodos que permitissem gerar boas soluções heurísticamente e incorporá-las ao algoritmo, pois fatores como o grande número de vizinhos a serem avaliados e a necessidade do cálculo de uma função objetivo quadrática para avaliar cada solução gerada fazem com que o PQA tenha uma alta complexidade, dificultando a busca por um melhor resultado.

No Capítulo 2 foi realizada a revisão de diversos trabalhos na literatura referentes ao Problema Quadrático de Alocação, o que proporcionou uma melhor compreensão acerca do problema. Realizou-se ainda o estudo de diversos métodos e conceitos empregados na resolução do PQA a fim de escolher uma abordagem a ser implementada. O Capítulo 3 apresentou os principais conceitos da meta-heurística escolhida e detalhes referentes à implementação do algoritmo.

Os resultados de testes para parametrização e avaliação do método proposto estão descritos no Capítulo 4. Os resultados sugerem que não há diferença significativa entre as soluções encontradas pelo método proposto e a literatura. Quando se compara o GAP entre a meta-heurística proposta e a literatura, (Tabela 6), observa-se que foram encontradas soluções iguais ou muito próximas às da literatura, sugerindo que a meta-heurística proposta pode ser uma boa alternativa na busca pela melhor solução. Todavia, a análise ANOVA e o Teste de Tukey revelam que a literatura provém uma melhor média para as soluções quando comparada com o método apresentado. Entretanto, a comparação dos valores médios do ES-VND com a literatura pode ser injusta, pois somente os melhores resultados estão disponíveis na literatura para comparação.

5.1 Propostas para trabalhos futuros

Durante a execução desse trabalho, algumas melhorias e inovações foram identificadas. Elas não foram implementadas em virtude de limitações do escopo e do tempo. A seguir, algumas sugestões de propostas de continuidade e trabalhos futuros são apresentadas:

- A incorporação de outros métodos de Busca Local na meta-heurística proposta;
- Utilização de outros modos para representação do problema;
- Aplicar métodos paralelos para cálculo do custo;

- Utilização da estrutura de vizinhança de larga escala;
- Realização de testes com outras instâncias do PQA;
- Comparação dos resultados obtidos com trabalhos correlatos;

Referências

- AHUJA, R. K. et al. Very large-scale neighborhood search for the quadratic assignment problem. *Inform journal on computing*, INFORMS, v. 19, n. 4, p. 646–657, 2007. Citado na página 14.
- AHUJA, R. K.; ORLIN, J. B.; TIWARI, A. A greedy genetic algorithm for the quadratic assignment problem. *Computers & Operations Research*, Elsevier, v. 27, n. 10, p. 917–934, 2000. Citado 2 vezes nas páginas 16 e 20.
- ANSTREICHER, K. et al. Solving large quadratic assignment problems on computational grids. *Mathematical Programming*, Springer, v. 91, n. 3, p. 563–588, 2002. Citado na página 30.
- ANSTREICHER, K. M.; BRIXIUS, N. W. Solving quadratic assignment problems using convex quadratic programming relaxations. *Optimization Methods and Software*, Taylor & Francis, v. 16, n. 1-4, p. 49–68, 2001. Citado na página 30.
- BRUNGER, A. et al. Joining forces in solving large-scale quadratic assignment problems in parallel. In: *Proceedings 11th International Parallel Processing Symposium*. [S.l.: s.n.], 1997. p. 418–427. ISSN 1063-7133. Citado na página 30.
- BURKARD, R. E. *Quadratic assignment problems*. [S.l.]: Springer, 2013. Citado 2 vezes nas páginas 14 e 19.
- CAMPEOTTO, F. et al. A gpu implementation of large neighborhood search for solving constraint optimization problems. In: *Proceedings of the Twenty-first European Conference on Artificial Intelligence*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2014. (ECAI'14), p. 189–194. ISBN 978-1-61499-418-3. Citado na página 18.
- CHIANG, W.-C. et al. Two-stage tabu—particle swarm algorithms for the facility layout problem with size constraints. In: IEEE. *Evolutionary Computation (CEC), 2011 IEEE Congress on*. [S.l.], 2011. p. 1679–1686. Citado na página 18.
- CHRISTOFIDES, N.; BENAVENT, E. An exact algorithm for the quadratic assignment problem on a tree. *Operations Research*, INFORMS, v. 37, n. 5, p. 760–768, 1989. Citado na página 30.
- CLAUSEN, J. et al. Benchmark testing for quadratic assignment problems on a portable parallel branch-and-bound library. *Work in progress*, 1996. Citado na página 30.
- CLAUSEN, J.; PERREGAARD, M. Solving large quadratic assignment problems in parallel. *Computational Optimization and Applications*, Springer, v. 8, n. 2, p. 111–127, 1997. Citado na página 30.
- CROES, G. A. A method for solving traveling-salesman problems. *Operations research*, INFORMS, v. 6, n. 6, p. 791–812, 1958. Citado na página 25.
- DICKEY, J.; HOPKINS, J. Campus building arrangement using topaz. *Transportation Research*, Elsevier, v. 6, n. 1, p. 59–68, 1972. Citado na página 18.

- DORIGO, M. *Optimization, Learning and Natural Algorithms (in Italian)*. Tese (Doutorado) — Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy, 1992. Citado na página 18.
- DORIGO, M.; STÜTZLE, T. *Ant Colony Optimization*. [S.l.]: BRADFORD BOOK, 2004. (A Bradford book). ISBN 9780262042192. Citado na página 18.
- DREZNER, Z. Finding a cluster of points and the grey pattern quadratic assignment problem. *OR Spectrum*, Springer, v. 28, n. 3, p. 417–436, 2006. Citado na página 30.
- FISCHETTI, M.; MONACI, M.; SALVAGNIN, D. Three ideas for the quadratic assignment problem. *Operations research*, INFORMS, v. 60, n. 4, p. 954–964, 2012. Citado na página 30.
- GEOFFRION, A. M.; GRAVES, G. W. Scheduling parallel production lines with changeover costs: Practical application of a quadratic assignment/lp approach. *Operations Research*, INFORMS, v. 24, n. 4, p. 595–610, 1976. Citado na página 19.
- GLOVER, F. Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, Elsevier, v. 13, n. 5, p. 533–549, 1986. Citado na página 18.
- HAHN, P.; GRANT, T.; HALL, N. A branch-and-bound algorithm for the quadratic assignment problem based on the hungarian method. *European Journal of Operational Research*, Elsevier, v. 108, n. 3, p. 629–640, 1998. Citado na página 30.
- HE, Y.; XIANG, M. An empirical analysis of approximation algorithms for the euclidean traveling salesman problem. *arXiv preprint arXiv:1705.09058*, 2017. Citado na página 25.
- HOLLAND, J. H. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press, 1975. Citado na página 17.
- JAMES, T.; REGO, C.; GLOVER, F. A cooperative parallel tabu search algorithm for the quadratic assignment problem. *European Journal of Operational Research*, Elsevier, v. 195, n. 3, p. 810–826, 2009. Citado na página 18.
- KOOPMANS, T. C.; BECKMANN, M. Assignment problems and the location of economic activities. *Econometrica: journal of the Econometric Society*, JSTOR, p. 53–76, 1957. Citado na página 14.
- LI, Y.; PARDALOS, P. M. Generating quadratic assignment test problems with known optimal permutations. *Computational Optimization and Applications*, Springer, v. 1, n. 2, p. 163–184, 1992. Citado na página 30.
- LOIOLA, E. M.; ABREU, N. M. M. d.; NETTO, P. O. B. Uma revisão comentada das abordagens do problema quadrático de alocação. *Pesquisa Operacional*, SciELO Brasil, v. 24, n. 1, p. 73–109, 2004. Citado 3 vezes nas páginas 14, 17 e 19.
- LUKE, S. *Essentials of Metaheuristics*. second. [S.l.]: Lulu, 2013. Available for free at <http://cs.gmu.edu/~sean/book/metaheuristics/>. Citado 3 vezes nas páginas 18, 20 e 22.

- LUONG, T. V.; MELAB, N.; TALBI, E. G. Large neighborhood local search optimization on graphics processing units. In: *2010 IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW)*. [S.l.: s.n.], 2010. p. 1–8. Citado na página 18.
- MAUTOR, T. *Contribution a la resolution des problemes d'implantation-algorithmes sequentiels et paralleles pour l'affectation quadratique*. Tese (Doutorado) — Paris 6, 1993. Citado na página 30.
- METROPOLIS, N. et al. Equation of state calculations by fast computing machines. *The journal of chemical physics*, AIP, v. 21, n. 6, p. 1087–1092, 1953. Citado na página 17.
- MISEVIČIUS, A. A modified simulated annealing algorithm for the quadratic assignment problem. *Informatica*, IOS Press, Amsterdam, The Netherlands, The Netherlands, v. 14, n. 4, p. 497–514, 2003. ISSN 0868-4952. Citado na página 17.
- MLADENOVIĆ, N.; HANSEN, P. Variable neighborhood search. *Computers & operations research*, Elsevier, v. 24, n. 11, p. 1097–1100, 1997. Citado na página 21.
- MONTGOMERY, D. C.; RUNGER, G. C.; CALADO, V. *Estatística Aplicada E Probabilidade Para Engenheiros*. [S.l.]: Grupo Gen-LTC, 2000. Citado na página 26.
- PARDALOS, L.; RESENDE, M. A greedy randomized adaptive search procedure for the quadratic assignment problem. *Quadratic Assignment and Related Problems, DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, v. 16, p. 237–261, 1994. Citado na página 30.
- PEARL, J. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1984. ISBN 0-201-05594-5. Citado na página 14.
- PRINS, C. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, Elsevier, v. 31, n. 12, p. 1985–2002, 2004. Citado na página 24.
- QI, C.; TIAN, W.; SUN, Y. Hybrid ant colony algorithm for qap. In: IEEE. *Computing, Communication, Control, and Management, 2009. CCCM 2009. ISECS International Colloquium on*. [S.l.], 2009. v. 3, p. 213–216. Citado na página 18.
- QU, R.; XU, Y.; KENDALL, G. A variable neighborhood descent search algorithm for delay-constrained least-cost multicast routing. In: SPRINGER. *International Conference on Learning and Intelligent Optimization*. [S.l.], 2009. p. 15–29. Citado 2 vezes nas páginas 21 e 22.
- RUSSELL, S.; NORVIG, P. *Inteligência artificial*. [S.l.]: CAMPUS - RJ, 2004. ISBN 9788535211771. Citado na página 17.
- SAREMI, H. Q.; ABEDIN, B.; KERMANI, A. M. Website structure improvement: Quadratic assignment problem approach and ant colony meta-heuristic technique. *Applied Mathematics and Computation*, v. 195, n. 1, p. 285 – 298, 2008. ISSN 0096-3003. Citado na página 18.
- STEINBERG, L. The backboard wiring problem: A placement algorithm. *Siam Review*, SIAM, v. 3, n. 1, p. 37–50, 1961. Citado na página 18.

TATE, D. M.; SMITH, A. E. A genetic approach to the quadratic assignment problem. *Computers and Operations Research*, v. 22, n. 1, p. 73 – 83, 1995. ISSN 0305-0548. Citado na página 17.

TUKEY, J. W. The problem of multiple comparisons. *Multiple Comparisons*, Chapman & Hall, 1953. Citado na página 28.

XIA, M. An ant colony algorithm hybridized with iterated local search for the qap. In: IEEE. *Computational Intelligence and Industrial Applications, 2009. PACIIA 2009. Asia-Pacific Conference on*. [S.l.], 2009. v. 2, p. 80–83. Citado na página 18.

ANEXO III – Termo de Responsabilidade

TERMO DE RESPONSABILIDADE

Eu, Kelly Márcia de Oliveira,
declaro que o texto do trabalho de conclusão de curso intitulado
"Um método baseado em estratégia evolutiva e variable
neighborhood descent para o problema quadrático de alocação" é de
minha inteira responsabilidade e que não há utilização de texto, material fotográfico, código
fonte de programa ou qualquer outro material pertencente a terceiros sem as devidas
referências ou consentimento dos respectivos autores.

João Monlevade, 05 de Setembro de 2017

Kelly Márcia de Oliveira
Assinatura do aluno

Certifico que a aluna **Kelly Márcia de Oliveira**, autora do trabalho de conclusão de curso intitulado “**Um método baseado em Estratégia Evolutiva e *Variable Neighborhood Descent* para o Problema Quadrático de Alocação**”, efetuou as correções sugeridas pela banca examinadora e que estou de acordo com a versão final do trabalho.

A handwritten signature in blue ink, reading "Fernando B. Oliveira", is written over a horizontal line.

Prof. Fernando Bernardes de Oliveira

Orientador

Ouro Preto, 18 de julho de 2018.