

MATHEUS ROBERTI RIBEIRO OLIVEIRA

**APLICAÇÃO DE HEURÍSTICAS PARA SOLUCIONAR CENÁRIOS DE  
MÉDIA ESCALA DO PROBLEMA DO MOCHILEIRO VIAJANTE**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

VIÇOSA  
MINAS GERAIS – BRASIL  
2017

**Ficha catalográfica preparada pela Biblioteca Central da Universidade  
Federal de Viçosa - Câmpus Viçosa**

T

O48a  
2017

Oliveira, Matheus Roberti Ribeiro, 1991-

Aplicação de heurísticas para solucionar cenários de média  
escala do problema do mochileiro viajante / Matheus Roberti  
Ribeiro Oliveira. – Viçosa, MG, 2017.

xiii, 95 f. : il. (algumas color.) ; 29 cm.

Inclui apêndice.

Orientador: André Gustavo dos Santos.

Dissertação (mestrado) - Universidade Federal de Viçosa.

Referências bibliográficas: f. 86-90.

1. Otimização combinatória. 2. Heurística. 3. Programação  
heurística. I. Universidade Federal de Viçosa. Departamento de  
Informática. Programa de Pós-Graduação em Ciência da  
Computação. II. Título.

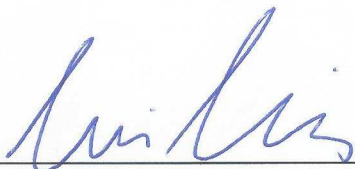
CDD 22. ed. 003

MATHEUS ROBERTI RIBEIRO OLIVEIRA

**APLICAÇÃO DE HEURÍSTICAS PARA SOLUCIONAR  
CENÁRIOS DE MÉDIA ESCALA DO PROBLEMA DO  
MOCHILEIRO VIAJANTE**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.


APROVADA: 08 de março de 2017.



Levi Henrique Santana de Lelis



Luciana Brugio Gonçalves



André Gustavo dos Santos  
Orientador

*A verdade pode ser intrigante. Pode dar algum trabalho lidar com ela.  
Pode ser contra-intuitiva. Ela pode contradizer preconceitos  
profundamente enraizados. Pode não se coadunar com o que queremos  
desesperadamente que seja verdade. Mas nossas preferências não  
determinam o que é a verdade.*

**Carl Sagan (1934 – 1996)**

# AGRADECIMENTOS

Uma série de fatores reunidos foram responsáveis pela conclusão desta dissertação. Alguns deles, como tudo na vida, ficam na conta do acaso; outros, sendo os mais importantes, aconteceram devido ao esforço e interesse de muita gente envolvida direta e indiretamente em minha vida. A primeira pessoa a quem devo agradecer é o André Gustavo dos Santos, por não apenas ser um excelente orientador e estar diretamente envolvido no trabalho, mas também ser uma ótima pessoa com quem pude ter a oportunidade de ter conversas agradáveis sobre os mais diversos assuntos, além de ter me guiado de forma exemplar na descoberta de uma área com a qual pouco contato tive antes de iniciarmos nossos trabalhos.

Aos meus professores e amigos da FAGOC que me possibilitaram iniciar uma etapa em minha vida onde foi possível reunir duas coisas que desde novo me fascinaram: computação e ciência. Um abraço especial ao Marcelo S. Daibert e o Sérgio M. Stempliuć, pelos bons tempos que passamos durante minha graduação, pela confiança que tiveram em mim e por sempre me incentivarem a ir além.

Um grande abraço a todos os meus amigos que estiveram presentes em minha vida durante parte dos últimos anos: vocês foram, de certa forma, responsáveis pela manutenção da minha sanidade nesse período, muito obrigado por isso! Aos meus pais e familiares: vocês me deram toda a base para tudo em minha vida, muito obrigado por me permitirem crescer pensando livremente e sem imposições irracionais, certamente essa é a coisa pela qual sou mais grato. Meus sinceros agradecimentos a todos citados acima e também a todas as pessoas que de forma direta ou indireta, através da arte ou razão, acrescentaram experiências em minha vida cuja consequência resultou em quem sou hoje. Agradeço também a CAPES pelo financiamento da pesquisa e a UFV, especialmente ao Departamento de Informática,

pelo acolhimento e viabilização da mesma.

# SUMÁRIO

LISTA DE TABELAS .....	viii
LISTA DE FIGURAS .....	xi
RESUMO .....	xii
ABSTRACT .....	xiii
1. INTRODUÇÃO .....	1
1.1 O problema e a sua importância .....	2
1.2 Hipótese .....	4
1.3 Objetivos .....	4
1.3.1 Objetivos específicos .....	4
1.4 Organização da Dissertação .....	5
2. O PROBLEMA DO MOCHILEIRO VIAJANTE .....	7
2.1. Definição do Problema .....	7
2.2. Exemplo .....	8
2.3. Trabalhos Relacionados .....	9
3. REFERENCIAL TEÓRICO .....	15
3.1. Problema do Caixeiro Viajante .....	15
3.2. Problema da Mochila .....	16
3.3. Técnicas de Otimização Combinatória .....	17
3.3.1. Heurísticas .....	18
3.3.1.1. Heurísticas Construtivas .....	18
3.3.1.2. Busca Local .....	19
3.3.2. Meta-Heurísticas .....	21
3.3.2.1 Simulated Annealing .....	22
4. HEURÍSTICAS PROPOSTAS PARA SOLUCIONAR O PROBLEMA ...	26
4.1. Plano de coleta .....	26
4.1.1. Travel Packing Planner Heuristic (TPPH) .....	26
4.1.2. TPPHx .....	28
4.2. Look and Construct Simulated Annealing .....	29
4.2.1. Solução Inicial .....	31
4.2.2. Critério de parada .....	32
4.2.3. Taxa de Atualização da Mochila .....	33

4.2.4. Estrutura de vizinhança .....	34
4.2.5. Critério de Aceitação.....	36
4.2.6. Ajuste da Temperatura .....	36
4.3. Outras Abordagens .....	39
5. EXPERIMENTOS COMPUTACIONAIS .....	40
5.1. Instâncias.....	40
5.2. Experimentos para Configuração dos Parâmetros .....	41
5.2.1. Instâncias .....	41
5.2.2. Planejamento dos Experimentos.....	41
5.2.3. Conjunto de Testes Estatísticos para Problemas de Otimização.....	42
5.2.3.1. Normalização dos Resultados.....	42
5.2.3.2. ANOVA .....	43
5.2.3.3. Durbin-Watson.....	45
5.2.3.4. Levene.....	45
5.2.3.5. Shapiro-Wilk .....	46
5.2.3.6. Scott-Knott .....	46
5.2.3.7. Kruskal-Wallis e o Método de Fischer LSD .....	46
5.2.4. Análise dos Resultados.....	47
5.2.4.1. Análise Estatística Considerando Todas as Instâncias .....	48
5.2.4.2. Análise Estatística: Instâncias com 76 Cidades.....	54
5.2.4.3. Análise Estatística: Instâncias com 195 Cidades .....	58
5.2.4.4. Análise Estatística: Instâncias com 3 Itens por Cidade .....	61
5.2.4.5. Análise Estatística: Instâncias com 5 itens por Cidade .....	64
6. RESULTADOS E DISCUSSÃO .....	68
6.1. Comparação entre TPPH e TPPH $\kappa$ .....	68
6.2. Direcionamento da Busca LaCSA por Iteração .....	70
6.3. Resultados dos Experimentos Computacionais e Análise de Desempenho.....	75
7. CONCLUSÃO .....	82
7.1. Sugestões para Trabalhos Futuros .....	84
REFERÊNCIAS BIBLIOGRÁFICAS .....	86
APÊNDICE A .....	91
A.1 Valores reais dos resultados .....	91

A.1.1 Valores das abordagens LaCSA, MATLS, S2, S5, C5 e C6.....	91
A.1.2 Valores das abordagens S1, S3, S4, C1, C2, C3, C4, RLS e EA .....	93
A.2 Divulgações dos resultados para a comunidade acadêmica .....	95

# LISTA DE TABELAS

Tabela 1 – Mudança na Taxa de Atualização usada pelo LaCSA de acordo com o tempo de busca em segundos .....	33
Tabela 2 – Probabilidade do LaCSA aceitar uma solução inferior para uma respectiva temperatura baseada em uma solução com 60 mil de lucro para o PMV .....	38
Tabela 3 - Probabilidade do LaCSA aceitar uma solução inferior para uma respectiva temperatura baseada em uma solução com 150 mil de lucro para o PMV .....	38
Tabela 4 – Resultados reais para algumas instâncias do PMV destacando a diferença entre os valores .....	43
Tabela 5 – Resultados do teste ANOVA sobre o impacto dos valores dos parâmetros no algoritmo quando aplicado nos resultados de todas as instâncias.....	48
Tabela 6 – Resultados do teste Levene para validação da Pressuposição II da ANOVA com todas as instâncias .....	50
Tabela 7 – Resultado do teste Shapiro-Wilk para validação da Pressuposição III da ANOVA com todas as instâncias .....	50
Tabela 8 – Resultados do teste Scott-Knott usando a análise de variância com todas as instâncias .....	51
Tabela 9 – Resultado do teste Kruskal-Wallis com o teste Fischer LSD para comparações múltiplas analisando todas as instâncias .....	53
Tabela 10 – Resultados do teste ANOVA sobre o impacto dos valores dos parâmetros no algoritmo quando aplicado somente nos resultados das instâncias com 76 cidades .....	55
Tabela 11 - Resultados do teste Levene para validação da Pressuposição II da ANOVA com as instâncias de 76 cidades.....	55
Tabela 12 - Resultado do teste Shapiro-Wilk para validação da Pressuposição III da ANOVA com as instâncias de 76 cidades .....	55
Tabela 13 - Resultados do teste Scott-Knott usando a análise de variância com as instâncias de 76 cidades .....	56
Tabela 14 - Resultado do teste Kruskal-Wallis com o teste Fischer LSD para comparações múltiplas analisando todas as instâncias.....	57
Tabela 15 - Resultados do teste ANOVA sobre o impacto dos valores dos parâmetros no algoritmo quando aplicado nos resultados de todas as instâncias.....	58

Tabela 16 - Resultados do teste Levene para validação da Pressuposição II da ANOVA com as instâncias de 195 cidades.....	58
Tabela 17 - Resultado do teste Shapiro-Wilk para validação da Pressuposição III da ANOVA com as instâncias de 195 cidades .....	58
Tabela 18 - Resultados do teste Scott-Knott usando a análise de variância com as instâncias de 195 cidades.....	59
Tabela 19 - Resultado do teste Kruskal-Wallis com o teste Fischer LSD para comparações múltiplas analisando as instâncias de 195 cidades.....	61
Tabela 20 - Resultados do teste ANOVA sobre o impacto dos valores dos parâmetros no algoritmo quando aplicado nos resultados das instâncias com 3 itens por cidade.....	61
Tabela 21 - Resultados do teste Levene para validação da Pressuposição II da ANOVA com as instâncias de 3 itens por cidade .....	62
Tabela 22 - Resultado do teste Shapiro-Wilk para validação da Pressuposição III da ANOVA com as instâncias de 3 itens por cidade.....	62
Tabela 23 - Resultados do teste Scott-Knott usando a análise de variância com as instâncias de 3 itens por cidade.....	62
Tabela 24 - Resultados do teste Kruskal-Wallis com o teste Fischer LSD para comparações múltiplas analisando as instâncias com 3 itens por cidade .....	64
Tabela 25 - Resultados do teste ANOVA sobre o impacto dos valores dos parâmetros no algoritmo quando aplicado nos resultados das instâncias de 5 itens por cidade .....	65
Tabela 26 - Resultados do teste Levene para validação da Pressuposição II da ANOVA com as instâncias de 5 itens por cidade .....	65
Tabela 27 - Resultado do teste Shapiro-Wilk para validação da Pressuposição III da ANOVA com as instâncias de 5 itens por cidade.....	65
Tabela 28 - Resultados do teste Scott-Knott usando a análise de variância com as instâncias de 5 itens por cidade.....	66
Tabela 29 - Resultado do teste Kruskal-Wallis com o teste Fischer LSD para comparações múltiplas analisando as instâncias com 5 itens por cidade .....	67
Tabela 30 – Resultados das heurísticas TPPH e TPPHx com diversos valores para as taxas usadas.....	70
Tabela 31 - Resultados de todas as abordagens analisadas para as instâncias com 51, 76 e 100 cidades.....	77

Tabela 32 – Resultados de todas as abordagens analisadas para as instâncias com 195, 225 e 280 cidades. ....	79
Tabela 33 – Resultados gerais para todas as instâncias analisadas em relação a cada abordagem .....	79
Tabela 34 – Resultados do teste <i>post hoc</i> de.....	80
Tabela 35 – Valores reais dos resultados para as menores instâncias solucionadas pelas abordagens que encontraram o melhor resultado para pelo menos uma instância: LaCSA, MATLS, S2, S5, C5 e C6 .....	91
Tabela 36 - Valores reais dos resultados para as maiores instâncias solucionadas pelas abordagens que encontraram o melhor resultado para pelo menos uma instância: LaCSA, MATLS, S2, S5, C5 e C6 .....	92
Tabela 37 - Valores reais dos resultados para as menores instâncias solucionadas pelas abordagens que não conseguiram encontrar o melhor resultado para qualquer instância: S1, S3, S4, C1, C2, C3, C4, RLS e EA .....	93
Tabela 38 - Valores reais dos resultados para as maiores instâncias solucionadas pelas abordagens que não conseguiram encontrar o melhor resultado para qualquer instância: S1, S3, S4, C1, C2, C3, C4, RLS e EA .....	93

# LISTA DE FIGURAS

Figura 1 - Exemplo de um cenário do PMV com 4 cidades e 4 itens.....	8
Figura 2 – Gráfico de análise do desempenho das abordagens propostas por Faulkner (2015).....	14
Figura 3 - Fluxograma de todo o processo para realização da calibração dos parâmetros do algoritmo LaCSA por meio de testes estatísticos .....	48
Figura 4 – Gráfico do resultado do teste Scott-Knott aplicado nos tratamentos de todas as instâncias: tratamentos considerados iguais usam a mesma cor.....	52
Figura 5 - Gráfico do resultado do teste Scott-Knott aplicado nos tratamentos das instâncias com 76 cidades: tratamentos considerados iguais usam a mesma cor .....	56
Figura 6 - Gráfico do resultado do teste Scott-Knott aplicado nos tratamentos das instâncias com 195 cidades: tratamentos considerados iguais usam a mesma cor.....	60
Figura 7 - Gráfico do resultado do teste Scott-Knott aplicado nos tratamentos das instâncias com 3 itens por cidade: tratamentos considerados iguais usam a mesma cor ....	63
Figura 8 - Gráfico do resultado do teste Scott-Knott aplicado nos tratamentos das instâncias com 76 cidades: tratamentos considerados iguais usam a mesma cor .....	66
Figura 9 – Iterações do algoritmo LaCSA na instância rat195_n582_bsc_07 em 10 minutos.....	71
Figura 10 - Iterações do algoritmo LaCSA na instância rat195_n582_bsc_07 em 1 hora de execução.....	72
Figura 11 - Iterações do algoritmo LaCSA na instância eil76_n225_bsc_03 em 10 minutos.....	73
Figura 12 - Iterações do algoritmo LaCSA na instância eil76_n225_bsc_03 em 1 hora de execução.....	73
Figura 13 – Avaliação da relação entre os componentes das soluções usadas pela busca LaCSA a cada iteração (rat195_n582_bsc_07).....	75
Figura 14 - Avaliação da relação entre os componentes das soluções usadas pela busca LaCSA a cada iteração (eil76_n225_bsc_03).....	75
Figura 15 – Diagrama de Caixa e Bigode ( <i>Box Plot and Whiskers</i> ) mostrando as médias dos RPD obtidos pelas abordagens existentes para solucionar o PMV .....	81

# RESUMO

OLIVEIRA, Matheus Roberti Ribeiro, M.Sc., Universidade Federal de Viçosa, março de 2017. **Aplicação de heurísticas para solucionar cenários de média escala do Problema do Mochileiro Viajante.** Orientador: André Gustavo dos Santos.

Nesta dissertação é apresentado o Problema do Mochileiro Viajante, sendo este um problema multicomponente composto da combinação de outros dois problemas bastante conhecidos e estudados pela comunidade acadêmica: o Problema do Caixeiro Viajante e o Problema da Mochila. Todo problema multicomponente possui duas características importantes: combinação e interdependência. A combinação existe porque um problema multicomponente deve ser composto de dois ou mais problemas individuais de otimização. A interdependência ocorre quando a combinação desses subproblemas torna-os capazes de afetar o resultado um do outro, como consequência, se forem resolvidos separadamente e suas melhores soluções individuais colocadas em união, não levarão, necessariamente, à melhor solução para o problema completo. Na definição do problema aqui tratado, a interdependência age através do impacto no tempo para percorrer a rota escolhida conforme diversos itens, com pesos específicos, são coletados (ou não) nas cidades onde estão localizados. Esse problema foi proposto através de uma discussão teórica sobre a distância encontrada entre os problemas enfrentados pelas indústrias e corporações, por vezes compostos por diversos subproblemas, e a maneira como, em geral, os pesquisadores tentam solucioná-los individualmente, impossibilitando assim uma visão maior do cenário onde se encontram e no impacto do relacionamento entre eles. Neste trabalho são realizados diversos estudos sobre a maneira como a interdependência dos componentes do PMV afeta o espaço de busca do mesmo, havendo propostas de heurísticas para solucioná-lo apenas com base em seus dados de entrada e de uma meta-heurística capaz de iniciar uma busca com uma solução inicial previamente fornecida e melhorá-la significativamente através de uma exploração de parte desse espaço de busca, sendo o percurso dessa exploração guiado por outra heurística que usa conhecimentos sobre a forma que os componentes do problema interagem entre si. Experimentos computacionais e análises estatísticas foram realizados para que as heurísticas aqui propostas pudessem ser melhor adaptadas para solucionar um conjunto de instâncias com as mais diversas características. Os resultados obtidos após esses experimentos são comparados com os resultados encontrados em outros três trabalhos presentes na literatura, revelando uma maior eficácia por parte das heurísticas aqui propostas quando cenários de pequeno e médio tamanho são selecionados.

# ABSTRACT

OLIVEIRA, Matheus Roberti Ribeiro, M.Sc., Universidade Federal de Viçosa, March, 2017. **Heuristics approaches to solve medium-scale instances of the Travelling Thief Problem.** Advisor: André Gustavo dos Santos.

In this dissertation the Travelling Thief Problem is presented, being this a multicomponent problem composed of the combination of two others well known and studied problems by the academic community: the Traveling Salesman Problem and the Knapsack Problem. Every multi-component problem has two important characteristics: combination and interdependence. The combination exists because a multi-component problem must be composed of two or more individual optimization problems. Interdependence occurs when the combination of these subproblems makes them capable of affecting each other's result, as a consequence, if solved separately and their best individual solutions put together, will not necessarily lead to the best solution to the whole problem. In the definition of the problem treated here, interdependence acts through the impact in time to go through the chosen route as several items, with specific weights, are collected (or not) in the cities where they are located. This problem has been proposed through a theoretical discussion about the distance found between the problems faced by industries and corporations, sometimes composed of several subproblems, and the way in which researchers in general try to solve them individually, thus preventing a whole view of the problem that they are in and the impact of the relationship between them. In this work several studies are carried out on how the interdependence of TTP components affects the search space of the problem, with heuristics proposals to solve it only based on its input data and a metaheuristic capable of initiating a search with a previously provided initial solution and significantly improve it by exploiting part of that search space with the exploration path being guided by another heuristic that uses knowledge about how the problem components interact with each other. Computational experiments and statistical analyzes were performed so that the heuristics proposed here could be better adapted to solve a set of instances with the most diverse characteristics. The results obtained after these experiments are compared with the results found in three other papers present in the literature, revealing a better efficiency by the heuristics proposed here when small and medium size instances are selected.

# 1. INTRODUÇÃO

Diversos problemas encontrados no mundo são estudados pelo campo da otimização combinatória para serem solucionados de forma computacional em tempo aceitável. Alguns desses problemas, devido a sua alta complexidade e aplicabilidade prática, se tornaram conhecidos como problemas clássicos da literatura, por exemplo: o Problema do Caixeiro Viajante (DANTZIG; FULKERSON; JOHNSON, 1954), o Problema de Roteamento de Veículos (RADIY, 2010), o Problema de Sequenciamento de Tarefas (MIHAILA, 2011), o Problema da Mochila (PISINGER, 1995), entre outros. Esses problemas foram extraídos do mundo real e são estudados por pesquisadores com o objetivo de encontrar formas satisfatórias de solucioná-los e, com isso, contribuir para os setores industriais e comerciais onde esses problemas são encontrados.

Em alguns casos esses problemas clássicos estudados pela otimização combinatória eram parecidos com outros problemas encontrados no mundo real, mas, devido a qualquer pequena diferença entre eles, uma boa solução para esses problemas poderia não ser uma boa solução para suas versões alternativas, ou sequer uma solução viável. Para resolver esse conflito foram criadas variações dos problemas clássicos com alguma adaptação, como a adição/remoção de alguma(s) restrição(ões) e/ou uma variação em sua função de avaliação.

A criação de diversas variações desses problemas serviu para encurtar a distância entre os problemas estudados na teoria e os problemas encontrados na prática. Entretanto, alguns pesquisadores ainda enxergam uma grande distância entre esses dois tipos de problemas e argumentam que parte desta distância existe porque muitos dos problemas encontrados no mundo real são multicomponentes (BONYADI et al, 2014), e não somente uma variação de outro problema. Um problema multicomponente consiste de dois ou mais problemas individuais, que, se resolvidos separadamente de forma ótima, não levarão necessariamente a uma solução ótima quando colocados em união (BONYADI; MICHALEWICZ; BARONE, 2013). Todo problema multicomponente possui duas características importantes: combinação e interdependência. A combinação existe porque um problema multicomponente é um problema composto de dois ou mais componentes, de forma que esses componentes sejam problemas de otimização quando avaliados individualmente. A interdependência ocorre quando a combinação dos componentes é feita de forma com que um afete o resultado do outro.

As indústrias e corporações têm se mostrado mais interessadas em ter seus problemas solucionados como um todo, em vez de obter a solução dos diversos subproblemas que o compõe separadamente, ou, ainda pior, solucionar somente parte de seus problemas (BONYADI et al, 2014). Foram realizadas diversas pesquisas em otimização combinatória sobre formas de solucionar problemas individuais, e, com isso,

existem hoje diversas técnicas para resolvê-los, que foram sendo aprimoradas com o passar do tempo e em vários casos nos fornecem uma garantia de qualidade da solução encontrada.

Existem estudos com o objetivo de identificar quais os fatores que elevam a complexidade dos problemas de otimização e quais as principais dificuldades encontradas na hora de propor abordagens para solucioná-los (WEISE, 2009), entretanto, pouco se tem pesquisado sobre a aplicabilidade e efetividade das técnicas tradicionais quando aplicadas a problemas multicomponentes (BONYADI; MICHALEWICZ; BARONE, 2013). Este trabalho tratará o Problema do Mochileiro Viajante (PMV, do inglês: *Traveling Thief Problem*), que é um problema multicomponente composto da combinação de dois problemas bem conhecidos: o Problema do Caixeiro Viajante (PCV) e o Problema da Mochila (PM).

A proposta desse problema foi o primeiro passo para o início de uma exploração que vem ocorrendo nos últimos anos para descobrir o quanto seria aumentada a complexidade dos problemas convencionais (monocomponente) encontrados na literatura se os mesmos fossem combinados de uma forma comum de serem encontrados no cotidiano e criassem assim um novo problema com característica multicomponente.

Fatores como múltiplos componentes presentes em um mesmo problema, a incerteza sobre seus dados e análises de riscos estão frequentemente encontrados nos problemas enfrentados pelas empresas e possuem impacto significativo na complexidade dos mesmos. Entretanto, esses fatores não vem sendo suficientemente explorados pela comunidade acadêmica e estudos teóricos sobre eles estão apenas começando a ser realizados (PRZYBYLEK et al 2016; MICHALEWICZ, 2016).

Atualmente vem ocorrendo um movimento entre os próprios pesquisadores da área de otimização combinatória para que a comunidade acadêmica direcione mais seus esforços em direção aos problemas complexos encontrados nos mais diversos setores da sociedade e que necessitam de atenção especial graças a sua alta complexidade, incertezas e elevado número de restrições, criando um campo onde as técnicas convencionais ainda precisam ter sua efetividade avaliadas e, talvez, novas técnicas possam surgir para solucionar essa nova classe de problemas, conhecida como problemas *Multi-Hard*, com maior eficácia (BONYADI et al, 2013; MICHALEWICZ, 2016; PRZYBYLEK et al, 2016).

## 1.1 O problema e a sua importância

O Problema do Mochileiro Viajante (PMV) foi proposto com o intuito de dar início à exploração da efetividade das técnicas atuais de otimização combinatória para problemas multicomponentes (BONYADI; MICHALEWICZ; BARONE, 2013; POLYAKOVSKIY et al, 2014). Mais especificamente, o PMV é um problema 2-

componentes criado através da junção de dois problemas clássicos da literatura, sendo eles o Problema do Caixeiro Viajante e o Problema da Mochila. Os autores propuseram duas modelagens que combinam esses dois problemas de formas diferentes de modo a fazer com que o PMV se tornasse interdependente em ambas abordagens (BONYADI; MICHALEWICZ; BARONE, 2013).

Nas duas modelagens propostas grande parte da definição do problema se mantém, variando somente pequenas partes em relação à avaliação de qualidade da solução. Após isolar suas diferenças, o PMV pode ser definido em ambos modelos da seguinte forma: dado um conjunto de cidades, cada qual com uma distância atribuída para se mover a qualquer outra cidade, defina uma rota de forma com que cada cidade seja visitada uma única vez, saindo de uma cidade de origem e voltando para ela no final. Cada cidade possui seu próprio conjunto de itens, cada qual com seu próprio valor e peso, e os mesmos podem ser coletados quando a cidade na qual eles se encontram for visitada, contanto que a soma dos pesos de todos os itens coletados não ultrapasse a capacidade máxima de peso suportada pela mochila do mochileiro. Há também uma velocidade máxima e mínima as quais o mochileiro possui como limites de sua velocidade. A velocidade usada para percorrer a distância entre uma cidade e outra é dada em função dessas duas velocidades e o estado atual da mochila, iniciando com uma velocidade máxima quando não há itens coletados e diminuindo sempre que um novo item for selecionado e o peso atual da mochila aumentar, até alcançar a velocidade mínima quando a mochila estiver cheia.

No modelo 1 proposto para o problema é definido também que o mochileiro deverá pagar uma taxa  $\$R$  para cada unidade de tempo gasta para percorrer toda a rota carregando os itens coletados. A solução ótima para este modelo é aquela que retorne o maior lucro possível, sendo o lucro a diferença entre a soma dos valores dos itens coletados e o custo de percorrer a rota. A função objetivo para obter o lucro é dada como:  $L(x, y) = g(y) - R * f(x, y)$ , sendo  $x$  a rota escolhida,  $y$  os itens coletados,  $R$  o valor da taxa de aluguel da mochila, a função  $f$  o custo em percorrer a rota  $x$  contendo os itens  $y$  e a função  $g$  o valor total dos itens coletados  $y$  (BONYADI; MICHALEWICZ; BARONE, 2013). É percebido que ao aumentar a taxa  $R$  aumenta-se também o impacto do custo de percorrer a rota, fazendo com que em casos extremos o ato de coletar um item nunca aumente a qualidade da solução, reduzindo o problema assim para a versão clássica do PCV. Por outro lado, se  $R = 0$ , o PMV se reduz ao PM, pois o custo de percorrer a rota não interferirá na qualidade da solução.

No modelo 2 proposto para o problema não existe uma taxa  $\$R$  a ser paga por unidade de tempo, porém, o valor de cada item diminui em relação ao tempo total gasto para percorrer toda a rota com os itens coletados. Esse modelo possui dois objetivos:

$$G(x, y) = \left\{ \begin{array}{l} \min f(x, y) \\ \max g(x, y) \end{array} \right\}, \text{ sendo } x \text{ a rota escolhida, } y \text{ os itens coletados, a função } f$$

retorna o custo em percorrer a rota  $x$  e a função  $g$  retorna o valor total dos itens  $y$  após o mochileiro chegar em seu destino (BONYADI; MICHALEWICZ; BARONE, 2013). Essa segunda modelagem do problema não foi abordada em nenhum trabalho presente na literatura até o momento, acredita-se que a complexidade inerente aos problemas multiobjetivo possa dificultar a análise da complexidade vinda da combinação e interdependência dos componentes do problema, portanto, o modelo 1 vem sendo usado como plataforma para o estudo desses fatores. Através do modelo 1 é possível estudar o impacto da característica multicomponente presente em um problema real: quando veículos de transporte aumentam o peso total de sua carga transportada, aumenta-se a quantidade de combustível necessário para sua locomoção.

Através do PMV podem ser estudadas formas de adaptar técnicas existentes de otimização para encontrar melhores soluções em problemas multicomponentes ou até mesmo a criação de novas técnicas específicas que levem em consideração a interdependência dos problemas para encontrar melhores resultados.

## **1.2 Hipótese**

O problema abordado é de otimização combinatória e pertencente à classe NP-Difícil. Para resolução dessa classe de problemas, em tempo aceitável, devem ser utilizados métodos aproximados, tais como heurísticas e/ou meta-heurísticas, as quais não oferecem garantias de que a melhor solução sempre será encontrada, entretanto, podem ser capazes de encontrar soluções de qualidade próximas a ela. Dessa forma, este trabalho baseia-se na hipótese de que é possível encontrar boas soluções para o Problema do Mochileiro Viajante utilizando algoritmos heurísticos, baseados em meta-heurísticas, em um tempo computacional aceitável.

## **1.3 Objetivos**

O objetivo geral desta dissertação é desenvolver abordagens heurísticas capazes de encontrar boas soluções para o Problema do Mochileiro Viajante em cenários de pequena e/ou média escala, analisando estatisticamente sua eficácia ao resolver o problema em relação às outras técnicas de otimização combinatória existentes na literatura.

### **1.3.1 Objetivos específicos**

Os objetivos específicos deste trabalho para alcançar o objetivo geral são:

- a) Fazer um estudo sobre os métodos heurísticos existentes na literatura para elaboração de abordagens próprias que possam solucionar o PMV.
- b) Apresentar um levantamento bibliográfico sobre o problema abordado e os métodos utilizados para resolvê-los;
- c) Propor algoritmos heurísticos para resolução do problema;
- d) Realizar testes e experimentos a fim de encontrar a melhor configuração de parâmetros para executar os algoritmos propostos por meio de testes estatísticos;
- e) Realizar experimentos computacionais para comparação entre as propostas deste trabalho e as existentes na literatura;
- f) Analisar estatisticamente os resultados obtidos.
- g) Avaliar a forma que a abordagem aqui proposta se comporta quando aplicada para solucionar instâncias com características diferentes.
- h) Analisar a interdependência entre os dois componentes do problema e a forma que a mesma afeta sua resolução.

#### **1.4 Organização da Dissertação**

As próximas seções deste trabalho estão organizadas da seguinte forma:

- No capítulo 2 é apresentado um levantamento bibliográfico sobre os conteúdos abordados nessa dissertação, de forma a fornecer ao leitor o conhecimento necessário para a correta interpretação do restante do texto.
- No capítulo 3 o Problema do Mochileiro Viajante é abordado detalhadamente e é fornecida uma revisão bibliográfica sobre os principais trabalhos sobre o problema encontrados na literatura atualmente.
- No capítulo 4 são descritas todas as abordagens propostas para solucionar o problema e são analisadas todas as características que foram desenvolvidas para tratar a interdependência do mesmo com o objetivo de encontrar um melhor resultado.
- No capítulo 5 é feita uma análise estatística para encontrar o melhor conjunto de valores para os parâmetros dos algoritmos e o impacto de cada estratégia usada pelas abordagens propostas com o objetivo de tirar proveito do conhecimento da interdependência do problema para melhorar a qualidade das soluções encontradas.

- No capítulo 6 os resultados de todos os experimentos efetuados são comparados com os resultados existentes nos principais trabalhos encontrados na literatura.
- Por fim, no capítulo 7 são discutidas algumas conclusões sobre os resultados analisados e a natureza do problema em relação à forma em que o mesmo foi tratado, assim como também quais são as possíveis direções futuras que podem ser tomadas a partir daqui.

## 2. O PROBLEMA DO MOCHILEIRO VIAJANTE

Neste capítulo o PMV é definido e um exemplo é usado para um melhor entendimento de suas possíveis soluções e demonstração de como sua interdependência afeta a qualidade das mesmas; em seguida, são informadas as contribuições dos trabalhos relacionados disponíveis na literatura.

### 2.1. Definição do Problema

O Problema do Mochileiro Viajante (PMV) foi proposto inicialmente com o intuito de dar início à exploração da efetividade das técnicas atuais de otimização combinatória para problemas multicomponentes (BONYADI; MICHALEWICZ; BARONE, 2013). O PMV pode ser definido da seguinte forma: (i) dado um conjunto de cidades, cada qual com uma distância atribuída para se mover a qualquer outra cidade, defina uma rota de forma com que cada cidade seja visitada uma única vez, saindo de sua cidade de origem e voltando para ela no final; (ii) cada cidade possui seu próprio conjunto de itens, cada qual com seu próprio valor e peso, e os mesmos podem ser coletados quando a cidade na qual eles se encontram for visitada, exceto ao chegar na cidade de destino, contanto que a soma dos pesos de todos os itens coletados não ultrapasse a capacidade máxima de peso suportada pela mochila do mochileiro; (iii) há também uma velocidade máxima e mínima as quais o mochileiro usa como limites de sua velocidade, e a velocidade usada para percorrer a distância entre uma cidade e outra é dada em função dessas duas velocidades e o estado atual da mochila, e pode ser definida através da seguinte fórmula:  $V_a = V_{max} - W_a \left( \frac{V_{max} - V_{min}}{W} \right)$ , sendo  $V_a$  a velocidade atual,  $V_{max}$  a velocidade máxima,  $V_{min}$  a velocidade mínima,  $W_a$  o peso atual da mochila e  $W$  a capacidade máxima da mochila. Desta forma, a velocidade do mochileiro será  $V_{max}$  quando a mochila estiver vazia ( $W_a = 0$ ), diminuindo sempre que um novo item for coletado até alcançar  $V_{min}$  quando a mochila estiver cheia ( $W_a = W$ ); (iv), por fim, o mochileiro deverá pagar uma taxa  $\$R$  para cada unidade de tempo gasta para percorrer toda a rota carregando os itens coletados. A solução ótima para este modelo é aquela que retorne o maior lucro possível, sendo o lucro a diferença entre a soma dos valores dos itens coletados e o custo para percorrer a rota. A função objetivo para obter o lucro é dada como:  $L(x, y) = g(y) - R * f(x, y)$ , sendo  $x$  a rota escolhida,  $y$  os itens coletados,  $R$  o valor da taxa de aluguel da mochila, a função  $f$  o custo em percorrer a rota  $x$  contendo os itens  $y$  e a função  $g$  o valor total dos itens coletados  $y$  (BONYADI; MICHALEWICZ; BARONE, 2013). É percebido que ao aumentar a taxa  $R$  aumenta-se também o impacto do custo de percorrer a rota, havendo casos onde o  $R$  pode ser grande o suficiente para fazer com que nenhum item aumente a qualidade da solução, reduzindo

assim o problema para a versão clássica do PCV. Por outro lado, se  $R = 0$ , o PMV se reduz ao PM, pois o custo de percorrer a rota não interferirá na qualidade da solução.

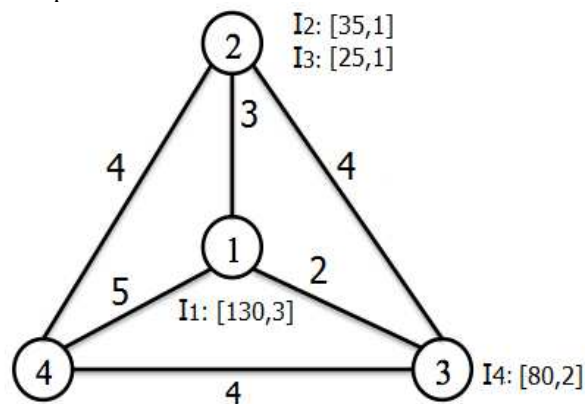
A interdependência no PMV não o tornou um problema multiobjetivo. Sua função objetivo deve somente maximizar o lucro obtido pelo mochileiro ao término de sua viagem. Entretanto, uma segunda definição do problema foi proposta (BONYADI; MICHALEWICZ; BARONE, 2013) onde sua função de avaliação possui dois objetivos. Enquanto na primeira definição a interdependência funciona somente em uma via, na segunda definição é encontrada uma interdependência entre os componentes do problema de forma com que a solução de um afete a solução do outro: como na primeira definição, os itens coletados vão interferir na velocidade para percorrer a rota, mas, além disso, os itens vão ter seu valor deteriorado em razão do tempo em que ficarem na mochila.

Por ser um problema recente, até o presente momento não há qualquer trabalho na literatura que propõe uma abordagem para solucionar a segunda definição do problema. Nesta dissertação somente a primeira definição do PMV é tratada.

## 2.2. Exemplo

Considere o exemplo apresentado na Figura 1. É possível visualizar que existem 4 cidades, sendo a cidade 1 a origem e destino final do mochileiro; há 4 itens, sendo o item 1 com valor de 130, peso 3 e localizado na cidade 1, os itens 2 e 3 localizados na cidade 2 e o item 4 localizado na cidade 3. O mochileiro deve definir sua rota de forma a sair da cidade 1, passar em todas as outras cidades e, por fim, retornar à cidade 1. O mochileiro pode tentar diminuir o seu prejuízo ao percorrer a rota coletando itens, mas deve estar atento para evitar os itens que aumentarão o custo de percorrer a rota ao ponto de superar os valores obtidos ao coletá-los, diminuindo assim o seu lucro final.

**Figura 1** - Exemplo de um cenário do PMV com 4 cidades e 4 itens.



**Fonte:** Elaborada pelo autor.

Considere um cenário para o exemplo da Figura 1 onde a capacidade de peso da mochila ( $W$ ) seja 3, o custo do aluguel ( $R$ ) da mochila por unidade de tempo seja 2,5, a velocidade máxima seja 1 e a mínima 0,1. Uma possível solução do PMV para esse cenário pode ser composta de:  $x = \{1, 4, 3, 2\}$  e  $y = \{0, 1, 0, 1\}$ , isto é, o mochileiro passará na seguinte ordem pelas cidades 1, 4, 3 e 2, coletando os itens 2 e 4 ao passar pelas cidades 2 e 3, respectivamente. Ao sair da cidade 1 e ir para a cidade 4 o mochileiro não coleta item algum, então, sua velocidade é a velocidade máxima e o tempo necessário para percorrer esse trajeto é igual a:  $5 \div \{1 - 0 \times [(1 - 0,1) \div 3]\} = T_{14} = 5$ . O mochileiro segue sua rota saindo da cidade 4 e indo para a cidade 3 sem coletar item algum, portanto, segue em velocidade máxima e tem o seu tempo de percurso calculado como:  $4 \div \{1 - 0 \times [(1 - 0,1) \div 3]\} = T_{43} = 4$ . Ao chegar na cidade 3 o item 4 é coletado, portanto, o peso atual da mochila passa a ser 2, afetando assim a velocidade do mochileiro, fazendo com que o tempo gasto para ir da cidade 3 até a cidade 2 seja:  $4 \div \{1 - 2 \times [(1 - 0,1) \div 3]\} = T_{32} = 10$ . Chegando na cidade 2 o mochileiro coleta o item 3 tornando seu peso atual igual  $(1 + 2)$  e alcançando a capacidade máxima da mochila e atingindo sua velocidade mínima como visto no cálculo do tempo levado para fazer a viagem de volta a cidade 1 (origem e destino):  $3 \div \{1 - 3 \times [(1 - 0,1) \div 3]\} = T_{21} = 30$ . Quando o mochileiro retorna à cidade de origem o seu lucro total com a viagem é equivalente a  $(80 + 35) - 2,5 \times (T_{14} + T_{43} + T_{32} + T_{21}) = -7,5$ .

No exemplo acima o lucro da viagem ficou negativo porque o valor dos itens coletados não supera o custo da viagem, mas, se considerarmos a mesma rota sem coletar item algum, o resultado final será de  $2,5 \times -(5 + 4 + 4 + 3) = -40$ , mostrando que houve um ganho de 32,5 por causa dos itens coletados.

É importante notar que a forma como a interdependência dos subproblemas age não permite que o PMV possa ser resolvido de forma ótima encontrando a solução ótima para o componente PCV e a solução ótima para o componente PM e as unindo em uma só solução. Essa situação pode ser vista quando tenta-se unir a rota ótima  $x = \{1, 2, 4, 3\}$  e a mochila ótima  $y = \{1, 0, 0, 0\}$  e a solução final para o PMV obtém um lucro de -195, valor menor do que o encontrado pela solução anterior. Nota-se também que a mesma solução sem coletar itens possui um lucro de  $0 - 2,5 \times 13 = -32,5$ , mostrando que o item coletado ocasionou em uma perda de -162,5 no lucro final. Encontrar a solução ótima para a rota e depois encontrar a solução ótima para a mochila usando a rota encontrada também não garante a solução ótima para o PMV.

### 2.3. Trabalhos Relacionados

Bonyadi, Michalewicz e Barone (2013) argumentaram que o volume dos dados tratados não é o fator que mais influencia na complexidade dos problemas encontrados hoje no

mundo real, sendo um desses fatores a característica multicomponente desses problemas. Um problema multicomponente foi descrito como sendo um problema de otimização composto de dois ou mais problemas. Foram questionadas a efetividade e a aplicabilidade das técnicas de otimização combinatória que são utilizadas em problemas 1-componente quando utilizadas em problemas multicomponentes. Para iniciar os estudos sobre essa classe de problemas os autores propuseram o Problema do Mochileiro Viajante. Foram propostas duas definições para o problema, cada qual com uma pequena diferença em relação ao outro, mas nenhuma forma de solucioná-los foi apresentada.

Esse e outros fatores que também influenciam na complexidade dos problemas reais e que não são suficientemente explorados são relatados também em outros trabalhos (PRZYBYLEK et al 2016), tais como a incerteza sobre os dados de um problema e análise de riscos dos mesmos. Atualmente vem ocorrendo um movimento entre os próprios pesquisadores da área de otimização combinatória para que a comunidade acadêmica direcione mais seus esforços em direção aos problemas complexos encontrados nos mais diversos setores da indústria e negócios e que necessitam de atenção especial dada sua alta complexidade, grande número de restrições e incertezas, criando um campo onde as técnicas convencionais ainda precisam ter sua efetividade avaliadas e, talvez, novas técnicas possam surgir para solucionar essa nova classe de problemas, também chamados como problemas *Multi-Hard*, com maior eficácia (MICHALEWICZ, 2016; PRZYBYLEK et al, 2016, BONYADI et al, 2013).

Foi definido um modelo matemático para o PMV onde o mesmo é apresentado como um problema de otimização não linear inteira pertencente à classe NP-Difícil (MEI; LI; YAO, 2014b). Se  $P \neq NP$ , problemas dessa natureza não podem ser resolvidos de forma determinística em tempo polinomial, tornando impossível a solução de instâncias grandes em tempo aceitável (GAREY; JOHNSON, 1979). Abordagens aproximativas como meta-heurísticas têm se mostrado eficazes para encontrar soluções de boa qualidade (TALBI, 2009; BLUM; ROLI, 2003).

Em (POLYAKOVSKIY et al, 2014) os autores realizaram um *benchmark* de instâncias para o PMV, e propuseram uma heurística construtiva e duas meta-heurísticas, uma busca local e um algoritmo genético, para encontrar soluções para esse *benchmark*. Para testarem as duas abordagens criaram também um conjunto de instâncias para o problema contendo 9720 instâncias. Os autores informaram no artigo os resultados para algumas instâncias de tamanhos diferentes, mas somente para instâncias de um determinado tipo. Entretanto, os resultados de suas três abordagens para cada uma das 9720 instâncias foram gentilmente disponibilizados por e-mail pelos autores do referente trabalho. No capítulo 6 desta dissertação serão mostrados os resultados para algumas dessas instâncias usando a meta-heurística aqui proposta e

definida no capítulo 3, em seguida será feita uma análise comparando os resultados obtidos com os resultados fornecidos pelos autores.

Em (MEI; LI; YAO, 2014a) os autores propuseram um Algoritmo Memético (AM) que usa uma estratégia de solução em duas etapas: uma rota é obtida através de uma busca local 2-opt e em seguida uma heurística é chamada para selecionar quais os itens que serão coletados para a rota fornecida, dando o nome para essa abordagem de Algoritmo Memético com Busca Local de Dois Estágios (MATLS, do inglês: *Memetic Algorithm with Two Stage Local Search*).

Além do MATLS, os autores também informaram algumas otimizações que foram feitas para reduzir a complexidade assintótica do algoritmo de modo com que o mesmo consiga resolver instâncias maiores do problema. As três estratégias de redução de complexidade propostas foram:

- Redução do tamanho da vizinhança: em vez de utilizar todos os vizinhos, gerados pelo operador 2-opt, somente os vizinhos que também pertenciam a Triangulação de Delaunay (ŽALIK, 2005) são analisados.
- Avaliação incremental: quando são aplicados os operadores de vizinhança sobre uma solução um novo vizinho é gerado e muitas vezes poucos componentes da solução são modificados em relação a solução de origem. Os autores criaram um algoritmo de avaliação que pudesse tirar proveito dessa característica, evitando recalcular partes das soluções vizinhas que não foram modificadas.
- Modificação eficiente da solução: os autores informaram que, inicialmente, para modificar uma rota, estava sendo utilizado um algoritmo de complexidade  $O(n)$ , mas, ao modificar a estrutura de dados da mesma para uma *splay tree* (FREDMAN et al, 1995) houve uma redução de complexidade para  $O(\log n)$ .

Os autores destacaram a importância da redução da complexidade para o tratamento de problemas de larga escala, mesmo que para isso seja necessário abrir mão de maiores garantias, como não avaliar todos os vizinhos 2-OPT e somente levar em consideração aqueles com maior possibilidade de formarem rotas mais curtas, isto é, os vizinhos pertencentes a Triangulação de Delaunay (MEI; LI; YAO, 2014a; ŽALIK, 2005).

Polyakovskiy e Neumanm (2015) elaboraram uma abordagem usando Programação Inteira Mista relaxada (MIP, do inglês *Mixed Integer Programming*) com o objetivo de encontrar um plano de coleta de itens subótimo para uma dada rota que deve ser fornecida como entrada e que não será alterada, isto é, a abordagem soluciona o Componente da Mochila encontrando a solução ótima ou muito próxima levando em consideração as características do PMV, mas, para realmente encontrar a solução ótima

para o PMV usando essa abordagem seria necessário descobrir de antemão qual a melhor rota, não apenas a de menor distância, mas a que permitiria coletar um conjunto de itens durante seu percurso que totalizaria no maior lucro possível (valor dos itens coletados – custo para percorrer a rota).

A abordagem MIP para solucionar a mochila não poderia ser realizada usando as características originais do problema, pois o mesmo é um problema não-linear, portanto, uma relaxação das restrições do problema foi realizada para linearizar o mesmo. Essa relaxação não foi realizada para diminuir a complexidade do problema, mantendo-o assim na classe de problemas NP-Difícil, mas viabilizando a proposta dessa abordagem para solucionar o componente da mochila de forma a encontrar resultados muito próximo do ótimo (POLYAKOVSKIY e NEUMANN, 2015).

Os resultados apresentados por Polyakovskiy e Neumann e suas análises sobre a interdependência dos componentes do problema foram interessantes, porém, não é viável o uso de sua abordagem incorporada a algum método heurístico para solucionar o problema completamente e de maneira eficiente, pois apenas o MIP teve um tempo limite de 1 dia para execução de cada instância, havendo instâncias menores, com 100 ou menos itens, em que o algoritmo conseguia encontrar a solução com poucos segundos, mas para instâncias com mais de 250 itens o tempo limite foi alcançado e o algoritmo, suspenso, retornou sua melhor solução encontrada durante esse período.

Um trabalho de grande contribuição em relação às diversas abordagens propostas foi realizado por Faulkner et al (2015), onde um conjunto de 11 abordagens diferentes foram usadas para solucionar o PMV e tiveram seus resultados comparados com outras abordagens existentes na literatura, tais como o MATLS proposto por Mei et al (2014a) e as abordagens propostas por Polyakovskiy et al (2014).

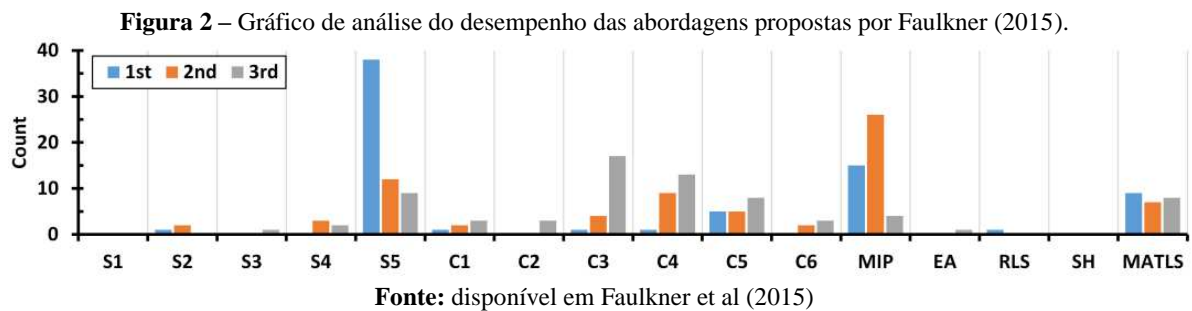
As abordagens propostas por Faulkner et al (2015) serão descritas com mais detalhes por serem as que detêm os melhores resultados contidos na literatura até o momento. Entretanto, por mais que o número de abordagens propostas seja grande, muitas apenas usam um conceito definido em outra e acrescentam ou modificam alguma característica, existindo assim 5 heurísticas ditas como “simples” (S1 ~ S5), 6 heurísticas definidas como mais “complexas” (C1 ~ C6) e uma abordagem usando Programação Inteira Mista baseada em outra abordagem MIP relaxada proposta por Polyakovskiy e Neumann (2015) com o objetivo de encontrar um plano de coleta de itens subótimo para uma dada rota que deve ser fornecida como entrada e que não será alterada:

- S1: executar a heurística de Lin-Kernighan (Lin e Kernighan, 1973) para encontrar uma rota e em seguida usar uma heurística construtiva para criação de um plano de coleta.

- S2: executar S1 e em seguida realizar na solução obtida uma busca local simples (*Hill Climbing*) usando a estrutura de vizinhança *Bitflip* apenas no componente da mochila até não haver como melhorar a solução.
- S3: executar S2 e em seguida realizar na solução obtida um algoritmo evolucionário 1+1 ([1+1]-EA) com sua mutação uniforme padrão, sendo esse um algoritmo simples que distribui uma probabilidade  $\frac{1}{M}$  para cada um dos  $M$  itens na solução terem seu valor invertido de coletado para não-coletado e vice-versa (RECHENBERG, 1973). Por esse não ser um algoritmo de convergência como a busca local da abordagem S2, seu critério de parada é definido como 10 minutos de tempo de execução.
- S4: executar S1 e em seguida realizar na solução obtida uma busca local simples (*Hill Climbing*) usando a estrutura de vizinhança *Insertion* apenas no componente da rota (PCV) até não haver como melhorar a solução.
- S5: executar S1 e repetir sua execução até que o tempo limite de 10 minutos se esgote. A heurística de Lin e Kernighan (1973) possui elementos estocásticos e é provável que ela retorne diversas soluções de boa qualidade para o PCV durante esse intervalo de tempo.
- C1: executar S1 e em seguida aplicar uma busca local usando a estrutura de vizinhança *Bitflip* e depois outra busca local usando a vizinhança *Insertion*, repetir as duas buscas locais em sequência até que o algoritmo convirja em uma solução contida em um mínimo local de ambas vizinhanças.
- C2: executar S1 e em seguida aplicar uma busca local usando a estrutura de vizinhança *Bitflip*, depois outra busca local usando a vizinhança *Insertion* e por último uma busca (1+1)-EA. Repetir as três buscas em sequência até que o tempo limite seja alcançado.
- C3: executar a heurística de Lin e Kernighan (1973) por 10% do tempo total de busca e em seguida usar a melhor solução (PCV) obtida por essa heurística para servir de solução inicial para a abordagem C2.
- C4: executar a heurística de Lin e Kernighan (1973) por 10% do tempo total de busca e em seguida usar a melhor solução (PCV) obtida por essa heurística para servir de solução inicial para a abordagem C3.
- C5: executar C1. Repetir esse processo até que o tempo se esgote.
- C6: executar C2. Repetir esse processo até que o tempo se esgote.
- MIP: executar a heurística de Lin-Kernighan para geração da rota e em seguida executar a abordagem baseada na proposta de Polyakovskiy e Neumann (2015) para criação de um plano de coleta para uma dada rota, repetindo esse processo até alcançar um tempo limite.

Em seu trabalho, Faulkner et al (2015) relataram que todas essas 12 abordagens foram usadas para solucionar 72 instâncias do PMV e seus resultados foram comparados entre si e também entre os resultados disponíveis no trabalho de I) Polyakovskiy et al (2014a) para suas abordagens EA (*Evolutionary Algorithm*, em português: Algoritmo Evolucionário) e RLS (*Random Local Search*, em português: Busca Local Aleatória); e II) de Mei et al (2014) para sua abordagem MATLS, sendo que esta abordagem havia previamente se mostrado mais eficiente em resolver o PMV do que suas duas antecessoras (EA e RLS).

Na Figura 2, extraída do trabalho de Faulkner et al (2015), é possível visualizar a eficiência dos algoritmos em encontrar a 1ª, 2ª e 3ª melhores soluções encontradas para cada instância entre todas as abordagens. Por exemplo, temos a abordagem S5 que encontrou para as 72 instâncias analisadas aproximadamente: 40 dos melhores resultados, 10 resultados que foram o segundo melhor e 7 resultados que foram o terceiro melhor.



A abordagem S5 foi considerada por Faulkner et al (2015) como sendo a que obteve os melhores resultados de maneira geral, sendo acompanhada de perto pelos resultados obtidos por Mei et al (2014a) em sua abordagem MATLS. A abordagem MIP apresenta resultados aparentemente significativos, porém, relataram em seu trabalho que essa solução conseguiu se sair muito bem em alguns grupos específicos de instâncias, mas não em outros, indo de acordo com o que foi relatado anteriormente nesta seção sobre a abordagem MIP original (MEI et al, 2014a) demorar poucos segundos para solucionar instâncias com menos de 100 itens e chegando a demorar 1 dia para solucionar instâncias com mais de 250 itens. No capítulo 6 esses resultados serão analisados em conjunto aos obtidos pelas abordagens propostas nesta dissertação.

### 3. REFERENCIAL TEÓRICO

Neste capítulo são abordados os tópicos necessários para um melhor entendimento sobre o problema aqui proposto e as abordagens usadas para conseguir solucioná-lo.

#### 3.1. Problema do Caixeiro Viajante

O Problema do Caixeiro Viajante (PCV) pode ser definido da seguinte forma: dado um conjunto de cidades e um custo para ir de uma cidade a outra para cada par de cidades, encontre a rota de menor custo onde cada cidade seja visitada somente uma vez e a cidade de origem seja também a última cidade a ser visitada (DANTZIG; FULKERSON; JOHNSON, 1954).

Esse é um dos problemas mais antigos de Otimização Combinatória, tendo sido definido em 1930, desde então tem sido muito estudado por pesquisadores devido as suas aplicações práticas e grande complexidade (LIN, 1965). O PCV é um problema NP-Difícil e sua complexidade assintótica é  $(n - 1)!$ , sendo  $n$  o número de cidades, tornando impraticável a utilização de métodos exatos para resolver instâncias grandes, isto é, para um grande número de cidades (PEDRO, 2013).

Foram criadas diversas variações do PCV para aproximá-lo de diversos problemas encontrados no mundo real. Algumas delas, citadas por Pedro (2013), podem ser vistas abaixo:

- O Problema do Caixeiro Viajante Assimétrico: variação do PCV original onde o custo de sair de uma cidade A e ir para uma cidade B pode ser diferente do custo de sair da cidade B e ir para a cidade A.
- Problema do Caixeiro Viajante com Coleta de Prêmios: variação onde não é necessário visitar todas as cidades que compõem o problema, mas há um ganho para cada cidade visitada e uma possível perda para as não visitadas. O objetivo consiste em maximizar o lucro, sendo este definido como o ganho em visitar o subconjunto de cidades da rota menos o custo total de percorrê-la.
- Problema do Caixeiro Viajante Suficientemente Próximo: variação onde cada cidade possui, além de sua localidade, um raio de proximidade definido como suficiente para visitá-la. Nessa variação o caixeiro não precisa visitar uma determinada cidade se a mesma estiver dentro do raio de proximidade de outra cidade já visitada.
- Problema do Caixeiro Viajante com Janela de Tempo: variação onde cada cidade possui uma faixa de horário na qual pode ser visitada. Soluções

que visitam uma cidade fora de sua faixa de horário são consideradas inválidas para o problema.

No Problema do Mochileiro Viajante (PMV) temos dois componentes compostos pelas versões clássicas do Problema do Caixeiro Viajante (PCV) e do Problema da Mochila (PM), havendo regras que associam ambos componentes criando assim uma interdependência forte entre eles.

### 3.2. Problema da Mochila

O Problema da Mochila (PM) pode ser definido da seguinte forma: dado um conjunto de itens, cada qual com seu respectivo valor e peso, e uma mochila com uma capacidade máxima de peso, selecione o subconjunto de itens que resulte na maior soma de valores possível, não permitindo que a soma do peso dos itens selecionados ultrapasse a capacidade de peso da mochila (PISINGER, 1995).

O PM é um problema NP-Difícil e sua complexidade assintótica é  $2^n$ , sendo  $n$  o número de itens, tornando impossível, se  $P \neq NP$ , obter resultados em tempo polinomial utilizando técnicas exatas de otimização combinatória para instâncias grandes, isto é, com muitos itens (MARTELLO, 1990).

É possível visualizar diversas aplicações do PM no mundo real, tais como: orçamento, carregamento de veículos, seleção de projetos, designação de tarefas, entre outras. Porém, para aproximar o PM clássico com outros problemas também encontrado no mundo real, foram criadas diversas variações do problema. Algumas delas, citadas por Pisinger (1995), são descritas a seguir:

Problema da Mochila Limitado: variação onde no máximo uma unidade de cada item pode ser selecionada para entrar na mochila. Essa variação é conhecida também como Problema da Mochila Binário ou Problema da Mochila 0-1.

Problema de Múltiplas Mochilas: variação do Problema da Mochila onde existem diversas mochilas e o valor e peso dos itens variam dependendo da mochila escolhida. Essa variação engloba outros tipos de problemas, como o Problema da Designação Generalizada.

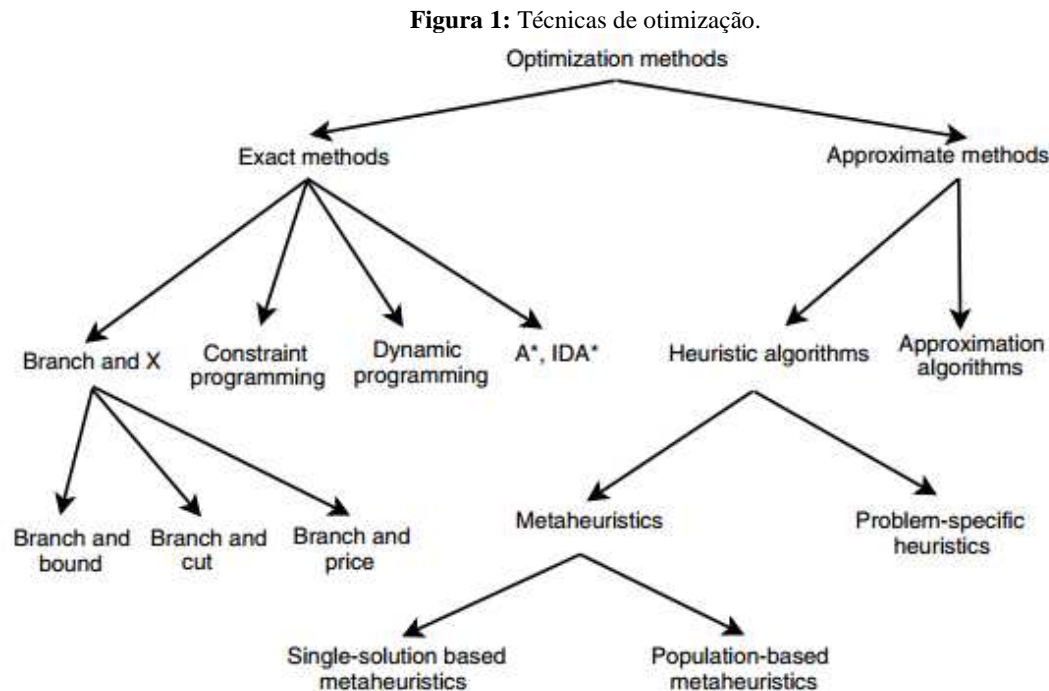
Problema da Mochila Multidimensional: variação onde outras dimensões são incluídas. Exemplo: além da capacidade de peso máxima suportada pela mochila há

também uma restrição sobre o seu volume, onde cada item também possua um volume definido.

No PMV é usada a versão clássica do PM, combinada com a versão clássica do PCV.

### 3.3. Técnicas de Otimização Combinatória

Existem diversas técnicas de otimização combinatória que podem ser utilizadas para encontrar a solução de problemas extraídos do mundo real e modelados computacionalmente para serem solucionados por uma máquina. Cada técnica possui sua própria característica e finalidade, conforme mostrado na Figura 1, há uma primeira e importante separação entre essas técnicas que parte da ramificação entre os métodos exatos e métodos aproximados.



Fonte: (TALBI, 2009).

Um método exato garante que a melhor solução, também conhecida como solução ótima, será encontrada para qualquer problema de otimização com uma entrada de tamanho finito (BLUM; ROLI, 2003). Entretanto, para os problemas de otimização combinatória que se encontram na classe NP-Difícil, assumindo que  $P \neq NP$ , não é possível garantir que seja encontrada a solução ótima em tempo polinomial (GAREY; JOHNSON, 1979).

Os métodos aproximados podem ser utilizados nos casos onde a instância do problema é grande o suficiente para não ser possível obter uma solução ótima em tempo aceitável através da utilização de técnicas exatas. Nesses métodos não há a garantia de que a melhor solução seja encontrada, porém, o seu tempo de execução é muito menor comparado aos métodos exatos.

Entre os métodos aproximados existem duas separações: as heurísticas e algoritmos de aproximação. Os algoritmos de aproximação podem ser usados para solucionar um problema através de uma dada abordagem específica para o mesmo;

possibilitando também, em certos casos, que certas propriedades possam ser provadas de forma a fornecer uma garantia de que a solução encontrada por esse tipo de algoritmo esteja a uma aproximação mínima do resultado ótimo para o problema tratado.

As heurísticas, por sua vez, possuem como objetivo encontrar uma boa solução para um problema, contrapondo-se aos algoritmos de aproximação no sentido de não ser possível fornecer uma garantia da qualidade dessa solução encontrada, porém, em geral, são mais simples de serem criadas e encontram soluções mais rapidamente. Essa técnica possui dois subconjuntos de métodos conhecidos como “heurísticas específicas ao problema” e “meta-heurísticas”. As heurísticas específicas ao problema são algoritmos criados para problemas específicos com o objetivo de criar uma boa solução a partir de uma solução inicial parcial ou inexistente. Em geral essas heurísticas são algoritmos rápidos e capazes de encontrar soluções com uma qualidade aceitável, entretanto, é possível encontrar soluções melhores ou melhorar uma solução encontrada por uma heurística usando técnicas conhecidas como meta-heurísticas (BLUM; ROLI, 2003). Essas técnicas são descritas em maior detalhes nas próximas seções.

### **3.3.1. Heurísticas**

Existem diferentes tipos de algoritmos heurísticos capazes de solucionar problemas, cada qual com sua própria características. Dois grupos precisam ser abordados em detalhes devido sua alta importância nas abordagens propostas neste trabalho: as heurísticas construtivas e as heurísticas de busca local.

#### **3.3.1.1. Heurísticas Construtivas**

São conhecidas como heurísticas construtivas os algoritmos capazes de analisar os dados de um problema e criar uma solução iniciando-se de uma solução vazia e a construindo parte por parte. Exemplos de algoritmos construtivos facilmente encontrados na literatura são:

- Problema do Caixeiro Viajante:
  - I. Heurística Aleatória: crie N rotas aleatórias e guarde a melhor encontrada, sendo que, quanto maior o valor N, maior a chance de uma boa rota ser gerada. A qualidade dessa heurística está fortemente relacionada a dois fatores: a quantidade de cidades existentes e o tempo disponível para a que o algoritmo exerça sua busca.
  - II. Heurística do Vizinho Mais Próximo: o PCV deve informar qual a cidade de origem do caixeiro, coloque essa cidade no começo da rota, após isso selecione as próximas cidades, uma após a outra,

escolhendo sempre a cidade mais perto de onde o caixeiro está no momento e que ainda não foi visitada por ele.

- Problema da Mochila:
  - I. Seleção por Valor: selecione os itens de maior valor, um após o outro, até que a mochila esteja com um peso alto o suficiente para não ser possível coletar qualquer outro item.
  - II. Seleção por Razão: selecione os itens de maior razão entre seu valor e peso até que nenhum outro item consiga ser coletado, priorizando assim uma relação custo-benefício.

As quatro heurísticas citadas acima, duas para cada problema, são todas consideradas como heurísticas construtivas, porém, a eficácia delas para resolver o mesmo problema são diferentes. Uma Heurística Aleatória como a definida acima dificilmente conseguirá obter uma boa solução entre as  $5,59 \times 10^{430}$  possibilidades para um cenário com 225 cidades, portanto, é provável que a Heurística do Vizinho Mais Próximo consiga obter melhores resultados, por mais que este também possa estar longe do resultado ótimo. O mesmo ocorre com as Heurística de Seleção pelo Valor ou pela Razão: avaliar a razão do valor de um item pelo seu peso é uma opção mais inteligente do que simplesmente selecionar os itens com maiores valores, uma vez que um único item de valor  $V_0$  pode impedir de selecionar dois ou mais itens em conjunto com valores  $V_1$  e  $V_2$ , sendo  $V_1 + V_2 > V_0$ ; portanto, conclui-se que podem existir diversas heurísticas construtivas para solucionar um mesmo problema e cada uma delas encontrar resultados bem diferentes entre si, sendo sua efetividade diretamente relacionada à forma como a mesma tenta extrair informações do problemas para criar sua solução.

### 3.3.1.2. Busca Local

Existem também heurísticas capazes de melhorar uma solução inicial, fornecida previamente, analisando o espaço de busca perto da região onde sua solução atual está contida à procura de uma solução superior à solução atual que, se encontrada, se tornará a solução corrente e uma nova iteração da busca ocorrerá usando-a como referência e repetindo o mesmo processo feito anterior para encontrá-la. Essa busca é suspendida assim que convergir para um mínimo local, isto é, lugares no espaço de busca em que a estrutura de vizinhança não é capaz de encontrar soluções melhores do que a atual.

Essas heurísticas, conhecidas também como Busca Local de Escalada (*Hill Climbing*) ou Heurística de Melhoramento, podem ser aplicadas a qualquer problema onde suas soluções podem ser avaliados quantitativamente, fornecendo assim um valor que possa ser maximizado ou minimizado para alcançar sua solução ótima, bastando:

- I. Obter uma solução inicial para o problema em questão.

- II. Definir uma estrutura de vizinhança a ser usada para explorar o espaço de busca em volta da solução atual.

Para fornecer uma solução inicial para a Busca Local, pode-se usar uma Heurística Construtiva baseada em algum critério guloso extraído de propriedades do problema. No Algoritmo 1 é possível entender a estratégia usada pela Busca Local para tentar melhorar uma dada solução.

**Algoritmo 1** – Pseudocódigo de uma Busca Local do tipo Melhor Aprimorante em um problema de maximização

---

```

1: função BuscaLocal(s0)
2:    $s \leftarrow s0$ 
3:   enquanto (tempo de execução permitido) faça
4:      $s^0 \leftarrow \text{MelhorVizinho}(s)$ 
5:     se ( $\text{avaliar}(s^0) > \text{avaliar}(s)$ ) então
6:        $s \leftarrow s^0$ 
7:     senão
8:       retorna  $s$ 
9:     fim se
10:  fim enquanto
11:  retorna  $s$ 
12: fim função

```

---

A vizinhança de uma solução é composta de todas as soluções que podem ser geradas ao aplicar um padrão qualquer de perturbação em cima da solução original (TALBI, 2009). É possível encontrar na literatura diversos tipos de estruturas de vizinhanças para os mais diversos problemas de otimização combinatória baseados em Busca Local. Algumas dessas estruturas de vizinhança se tornaram conhecidas por seu bom desempenho, tais como as listadas abaixo; porém, deve-se sempre avaliar se a estrutura de vizinhança usada está de acordo com as restrições do problema.

- Estrutura de Vizinhança para o PCV:
  - I. *Insertion*: com base em uma solução válida para o PCV, crie uma nova solução (vizinha) para cada movimento de inserção que possa ser realizado. Um movimento de inserção consiste em selecionar uma cidade da rota e colocá-la em qualquer outra posição além de sua atual. O espaço de busca dessa estrutura de vizinhança é composto de todas as soluções criadas dessa maneira.
  - II. *Swap*: com base em uma solução válida para o PCV, crie uma nova solução para cada dupla de cidades que possa ser trocada de lugar uma com a outra. O conjunto dessas novas soluções será a vizinhança de sua solução.
- Estrutura de Vizinhança para o PM:
  - I. *Bitflip*: obtenha uma solução válida para o PM, escolha um de seus itens e mude seu estado de coletado para não coletado ou vice-

versa, sendo a estrutura completa da vizinha composta por todas as soluções criadas modificando o estado de cada um dos itens por vez.

- II. *Swap*: obtenha uma solução válida para o PM, escolha dois itens e troque o estado de um com o outro, sendo desnecessário realizar essa operação quando ambos estados forem iguais.

Um fator que pode interferir no desempenho da busca é a forma como a mesma será conduzida: sabe-se que é necessário avaliar os vizinhos à procura de uma solução melhor do que a que os gerou, porém, a forma como esses vizinhos serão avaliados pode variar de uma abordagem para a outra, havendo, inclusive, duas maneiras bastante presentes na literatura, sendo elas:

- *First Improvement* (Primeiro Aprimorante): ao usar essa estratégia a busca local escolherá o primeiro vizinho que for avaliado como sendo melhor do que a solução atual que o gerou, levando a busca para sua próxima iteração com base nessa nova solução selecionada.
- *Best Improvement* (Melhor Aprimorante): ao usar essa estratégia a busca local avaliará todos os vizinhos e, se houver vizinhos que sejam melhores do que a solução atual que o gerou, a busca escolherá o melhor dentre eles para ser usado em sua próxima iteração.

Em geral, busca locais são algoritmos rápidos e capazes de encontrar soluções de qualidade em tempo aceitável. Entretanto, é possível usar outras técnicas heurísticas (e.g.: meta-heurísticas), capazes de percorrer o espaço de busca de forma mais inteligente e não se prendendo facilmente em regiões como mínimos locais.

### 3.3.2. Meta-Heurísticas

Enquanto as heurísticas são criadas para solucionar um problema em específico, as meta-heurísticas foram propostas com o objetivo de solucionar problemas de otimização de forma mais geral, havendo necessidade de poucas adaptações ao se aplicar a mesma meta-heurística em problemas diferentes (TALBI, 2009; BLUM; ROLI, 2003). As heurísticas construtivas são responsáveis por criar uma solução para um determinado problema, de forma distinta, as meta-heurísticas realizam buscas locais usando estratégias de intensificação e diversificação capazes de guiar a busca de forma mais inteligente do que uma simples busca local. Existem também meta-heurísticas conhecidas como “populacionais” por começarem sua busca com um conjunto de soluções

iniciais e possuem uma estratégia, específica à abordagem usada, para combinar os componentes dessas soluções contidas na população de uma forma que novas soluções possam ser criadas e, possivelmente, encontrar uma ou mais soluções melhores do que as contidas na população atual.

As meta-heurísticas devem ser usadas levando em consideração dois fatores: diversificação e intensificação. A diversificação é a característica que faz com que as meta-heurísticas explorem o espaço de busca de forma mais ampla, mesmo que para isso a busca vá, em algum momento, para locais aleatórios ou de piores avaliações. De forma contraditória à diversificação, a intensificação faz com que o espaço de busca se concentre mais em torno das vizinhanças onde foram encontradas as melhores soluções durante a busca (TALBI, 2009). A seguir é apresentada a meta-heurística usada neste trabalho.

### 3.3.2.1 Simulated Annealing

Na metalurgia existe um processo chamado de “recozimento de metais” (do inglês: *annealing*) que é usado para tornar sólidos, tais como bronze, prata, ferro, entre outros, mais maleáveis para poderem ser trabalhados da maneira necessária. Um modelo computacional para simular esse processo foi proposto por Metropolis (1953) para avaliar o comportamento dos átomos desses sólidos quando expostos a esse processo. Após 30 anos, Kirkpatrick et al (1983) inspirou-se em sua modelagem para propor uma nova meta-heurística capaz de solucionar problemas de otimização combinatória inspirado em reações químicas e físicas que ocorrem durante o processo de recozimento de metais. Essa meta-heurística ficou conhecida como *Simulated Annealing* (SA), sendo conhecida em português com o nome de Arrefecimento Simulado (ou Recozimento Simulado). Pouco tempo após seu surgimento houve um estudo de sua aplicação (Cerný, 1985) para solucionar o Problema do Caixeiro Viajante e resultados significantes foram obtidos através de seu uso, ocorrendo assim uma popularização dessa meta-heurística e incentivando sua aplicação nos mais diversos tipos de problemas.

A realização do recozimento em um metal ocorre em duas etapas: inicialmente o metal precisa ser levado a uma temperatura alta o suficiente para entrar em sua fase austenita; em seguida, o mesmo é resfriado lentamente até sua solidificação, resultando assim em um metal mais ductível e com menores tensões internas. Esse resultado é obtido porque o metal, quando aquecido, aumenta a energia interna de seus átomos permitindo que os mesmos possam se movimentar. Após serem adequadamente resfriados, a tendência é que esses átomos consigam encontrar uma configuração interna de menor energia.

As características definidas no processo físico também devem ser encontradas e ajustadas quando essa meta-heurística é aplicada a um problema de otimização

combinatória: o metal pode ser considerado a solução inicial a ser trabalhada pelo SA e as temperaturas podem ser controladas através de parâmetros no algoritmo. Entretanto, encontrar os valores corretos para a temperatura e a taxa de queda de temperatura (comumente chamada de *alpha*, da letra grega  $\alpha$ ) é um trabalho que necessita ser feito para cada tipo de problema em específico, sendo possível até mesmo haver variações nos valores ideais a serem usados para um mesmo problema quando cenários de tamanhos diferentes forem analisados; para resolver tal problema, pesquisadores costumam realizar experimentos sob uma amostra do conjunto total de cenários com alguns diferentes valores para esses parâmetros e, em seguida, analisam os resultados desses experimentos usando métodos estatísticos que os permitam verificar, com certo grau de confiança, quais valores para os parâmetros (temperatura e  $\alpha$ ) conseguiram obter um resultado mais significativo (BARTZ-BEIELSTEIN e LASARCZYK, 2005; CHIARANDINI et al, 2007; RARDIN e UZSOY, 2001).

No Algoritmo 2 são mostrados os processos existentes na versão clássica do *Simulated Annealing*. Nele está contido, como parâmetros, todos os conceitos descritos acima: solução inicial (metal), temperatura e taxa de queda da temperatura ( $\alpha$ ) por iteração. Em seu início, a busca está em uma temperatura máxima e é finalizada quando a mesma cai para zero, porém, outros critérios de parada podem ser usados no lugar deste, bastando manter a queda de temperatura em razão de  $\alpha$  conforme as iterações forem sendo realizadas. A cada iteração do algoritmo serão criados os vizinhos da solução usada pelo *Simulated Annealing* naquele momento. A estrutura de vizinhança a ser usada depende da representação da solução e é uma informação que necessita ser definida após análise das propriedades do problema sendo trabalhado, havendo, assim como em outras meta-heurísticas, uma forte dependência entre esses dois fatores. A ideia central do *Simulated Annealing* é que seja feita uma busca local probabilística usando a cada iteração os vizinhos da solução atual; a cada vizinho analisado olha-se seu valor de aptidão, isto é, o valor da qualidade dessa solução (quantitativamente), através de uma função objetivo que é definida em relação a forma como se espera solucionar o problema. Se o vizinho analisado for uma solução melhor do que qualquer outra encontrada até o momento, ele será armazenado como a melhor solução e será também usado como solução atual da busca na próxima iteração, caso contrário, sua aceitação como solução atual ainda pode ocorrer estocasticamente através de uma probabilidade P calculada por um critério de aceitação.

**Algoritmo 2** – Pseudocódigo da meta-heurística Simulated Annealing em sua versão clássica

---

1: **função** Simulated Annealing(*solucaoInicial*, *tempMax*, *tempMin*, *alpha*, *K*)  
2:   *melhorSolucao*  $\leftarrow$  *solucaoInicial*,  
3:   *solucaoAtualSA*  $\leftarrow$  *solucaoInicial*,  
4:   *temp*  $\leftarrow$  *tempMax*

---

```

5:  enquanto  $temp > tempMin$  faça
6:    para  $i \leftarrow 1$  até  $K$  faça
7:       $vizinho \leftarrow GerarVizinho(solucaoAtualSA)$ 
8:      se  $avaliar(vizinho) > avaliar(melhorSolucao)$  então
9:         $melhorSolucao \leftarrow vizinho$ 
10:        $solucaoAtualSA \leftarrow vizinho$ 
11:      senão
12:        se  $CriterioDeAceitacao(temp, vizinho)$  então
13:           $solucaoAtualSA \leftarrow vizinho$ 
14:      fim se
15:    fim se
16:  fim para
17:   $temperatura \leftarrow AjustarTemperatura(temperatura, alpha)$ 
18: fim enquanto
19:  retorna  $melhorSolucao$ 
20: fim função

```

---

O critério de aceitação mais usado pelo *Simulated Annealing* foi criado por Metropolis (1953) em seu modelo computacional de sólidos e é amplamente baseado nas leis da termodinâmica estatística (mecânica estatística). Kirkpatric et al (1983) percebeu que a função definida por Metropolis poderia também ser usada como critério de aceitação pelo *Simulated Annealing*, sendo essa função baseada I) na diferença entre a qualidade do vizinho e a qualidade da melhor solução encontrada até o momento, II) a temperatura atual e III) na constante de Euler; podendo ser definida como  $P(Vv, Sv, T) = e^{(Vv - Sv) \div T}$ , sendo  $Vv$  o valor do vizinho,  $Sv$  o valor da melhor solução e  $T$  a temperatura atual;  $e$  o Número de Euler, sendo uma constante de valor aproximado a 2,71, sendo este número também usado por Metropolis (1953) como base de sua exponenciação. O critério de aceitação definido aqui se comporta da seguinte forma: com uma temperatura alta o suficiente qualquer solução pode ser aceita, porém, conforme a temperatura for decrescendo, a função de probabilidade pode reduzir drasticamente a chance de uma solução ruim ser selecionada, tornando o *Simulated Annealing* em uma busca local *Hill-Climbing* onde dificilmente são admitidas mudanças para vizinhos que são piores do que a melhor solução encontrada, sendo esse o principal motivo para que seja bem definida a taxa de redução de temperatura por iteração.

Ao longo de uma iteração um desses três diferentes cenários pode ocorrer: I) um vizinho melhor do que a solução atual foi encontrado, II) um vizinho pior do que a solução atual foi aceito pela probabilidade dada através do critério de aceitação ou III) nenhum vizinho foi selecionado e o algoritmo usará a mesma solução atual na próxima iteração. Independente de qual cenário ocorrer, quando chegar ao final de sua iteração o algoritmo diminuirá a temperatura atual com base em seu  $\alpha$  definido como taxa de queda de temperatura e continuará com suas iterações até que o critério de parada seja satisfeito.

Para solucionar o PMV uma versão alternativa do *Simulated Annealing* com o nome de *Look and Construct Simulated Annealing* (LaCSA) foi proposta para esse trabalho. Nessa versão da meta-heurística existem algumas diferenças em relação sua versão original, sendo a maior delas relacionada à forma como essa abordagem guia sua busca não somente usando uma estrutura de vizinhança, mas com o auxílio também de uma heurística construtiva para criar parte das soluções vizinhas que foram encontradas, usando assim informações sobre o fator multicomponente do problema para reduzir seu espaço de busca. Mais detalhes sobre essa abordagem serão fornecidos no capítulo 4, onde sua proposta, algoritmos e características serão analisadas.

## 4. HEURÍSTICAS PROPOSTAS PARA SOLUCIONAR O PROBLEMA

O PMV é redutível ao PCV quando o valor e peso de todos os itens for zero; e o PCV, por sua vez, foi demonstrado ser um problema NP-Difícil (PAPADIMITRIOU, 1977). Se  $P \neq NP$ , então não é possível usar métodos exatos para obter uma solução ótima de forma determinística em tempo polinomial para grandes instâncias deste problema, entretanto, métodos aproximados podem ser utilizados para obter soluções competitivas em tempo computacional aceitável.

Abordagens heurísticas são aconselháveis a serem usadas em situações como essa, portanto, neste capítulo, são abordadas todas as heurísticas propostas para solucionar o problema, assim como todas as minúcias implementadas em cada uma delas com o objetivo de melhorar o resultado final encontrado pela abordagem completa quando executada para solucionar cenários de média escala do PMV.

### 4.1. Plano de coleta

Um plano de coleta no PMV tem as mesmas restrições impostas a um PM simples, portanto, quaisquer itens podem ser coletados, contanto que não exceda a capacidade de peso suportada pela mochila. Apesar de não ter restrições adicionais que torne o trabalho de encontrar uma solução viável para o PMV pior do que encontrar para o PM, pode haver uma grande diferença na avaliação da qualidade de uma mesma solução quando avaliada em ambos os problemas: no PM, quando um conjunto de itens com lucro alto é coletado há um ganho real, porém, não necessariamente essa alteração é positiva se considerado o PMV, uma vez que o custo para percorrer a rota pode aumentar ao ponto de não valer a pena coletar determinados itens em determinadas rotas.

Ao coletar um item de lucro alto, mas peso também alto, altera-se significativamente o tempo gasto para realizar o percurso de todas as cidades, aumentando assim o custo final do aluguel da mochila que é cobrado por unidade de tempo. Esse mesmo item, se coletado no começo da rota, mudará o tempo para visitar todas as próximas cidades, porém, se for coletado no final da rota, mudará o tempo para percorrer uma distância muito menor até voltar para a cidade inicial. Essa propriedade do problema faz com que um plano de coleta somente possa ser avaliado no âmbito do PCV como “ruim” ou “bom” se houver uma rota pré-definida para ele.

#### 4.1.1. Travel Packing Planner Heuristic (TPPH)

Uma heurística de seleção de itens para o PMV precisa avaliar tanto os efeitos positivos quanto os negativos de se coletar um determinado item, portanto, uma rota deve ser fornecida como requisito para obter essas informações. Para uma determinada rota é possível saber qual o verdadeiro impacto que um item possui na solução do problema, bastando verificar a diferença entre os valores retornados pela função de avaliação para uma solução contendo esse item e outra não.

No Algoritmo 3 é mostrada uma heurística construtiva proposta para solucionar o componente da mochila pertencente ao PMV. Esse algoritmo se assemelha às heurísticas para o PM que selecionam os itens que possuam a maior razão entre seu valor e peso até que a capacidade da mochila seja alcançada ou os itens esgotados. Porém, essa estratégia de seleção de itens não é apropriada para o PMV, uma vez que coletar um item traz consigo não só um efeito positivo (seu próprio valor), mas também um negativo: o aumento de tempo para percorrer a rota e, conseqüentemente, o preço a ser pago pelo aluguel da mochila, podendo esse aumento ser maior que o próprio valor do item.

A estratégia de seleção usada no Algoritmo 3 consiste em escolher os itens que terão o maior impacto positivo se acrescentados a solução. Para isso, é necessário definir o impacto de cada item como sendo:  $item_{IMP} = [L(x, y \cup item) - L(x, y)] \div item_{PESO}$ , dessa forma, os itens com os maiores impactos serão os que mais beneficiarão uma solução composta pela rota  $x$ . Entre as linhas 4 e 8 os itens são ordenados de forma decrescente pelo seu impacto se acrescentado na solução e as variáveis necessárias para a busca são inicializadas. Na linha 10 é definido o critério de parada do algoritmo de seleção dos itens verificando se não há mais itens a serem analisados, se a capacidade da mochila foi alcançada ou se os itens restantes possuem um impacto negativo; caso alguma dessas condições seja verdadeira, não é possível coletar qualquer item restante de modo a afetar de forma positiva a solução. Na linha 13 é verificado se o item selecionado naquela iteração pode ser coletado sem violar a restrição de capacidade da mochila, se sim, o mesmo é selecionado para a solução, pois, devido a ordenação da lista de itens, há uma garantia de que o impacto deste seja maior ou igual a qualquer outro item posterior a ele na lista, sendo essa sequência repetida até que um dos critérios de parada ocorra.

**Algoritmo 3** – Heurística Construtiva TPPH para elaboração de um plano de coleta de itens para uma rota predeterminada

---

1: **função** TPPH(*rota*,*itens*,*N*,*W*)

---

2:     *planoDeColeta*  $\leftarrow \emptyset$   
3:     *w*  $\leftarrow 0$   
4:     *avaliacaoTTPBase*  $\leftarrow$  *avaliarTTP(rota,planoDeColeta)*  
5:     **para** *i*  $\leftarrow 1$  **até** *N* **faça**

```

6:     avaliacaoTTPComItem ← avaliarTTP(rota,planoDeColeta ∪ itens[i])
7:     itens[i].imp ← (avaliacaoTTPComItem - avaliacaoTTPBase) ÷ itens[i].peso
8:     fim para
9:     Ordenar decrescentemente os itens pelos seus impactos
10:    enquanto (( $N \geq 1$ ) ∧ ( $w < W$ ) ∧ (itens[1].imp > 0)) faça
11:        item ← RemoverItem(itens[1])
12:         $N \leftarrow N - 1$ 
13:        se (item.peso +  $w \leq W$ ) então
14:            planoDeColeta ← planoDeColeta ∪ item
15:             $w \leftarrow w + item.peso$ 
16:        fim se
17:    fim enquanto
18:    retorna planoDeColeta
19: fim função

```

---

#### 4.1.2. TPPHx

Se um item é coletado, o seu peso é acrescentado à solução e o tempo (e custo) para percorrer a rota é aumentado. Quanto mais cheia a mochila estiver, maior será o custo para percorrer a rota, isto é, coletar um item quando o peso da mochila for um valor  $P$  qualquer de sua capacidade total, aumentará em  $T$  o tempo do percurso; mas, coletá-lo quando o peso da mochila for maior que  $P$  de sua capacidade, aumentará o tempo em uma quantidade maior que  $T$ . Essa propriedade mostra a importância em recalculer o lucro de cada item restante quando o peso da mochila for atualizado, principalmente quando o peso atual da mochila estiver se aproximando da capacidade máxima da mesma. A heurística deve então ser modificada para recalculer os impactos e reordenar os itens a cada inserção. Entretanto, existem itens com peso tão baixo que se coletados não causam grande impacto no tempo para percorrer a rota, então, quando apenas um item leve é coletado, recalculer os lucros dos itens restantes e reordenar toda a lista dos mesmos se torna uma operação praticamente desnecessária.

Uma evolução da heurística TPPH se faz necessária para levar essa característica do problema em consideração. A heurística TPPHx mostrada no Algoritmo 4 faz essa consideração da seguinte forma: após receber por parâmetro uma “taxa de atualização dos itens” é feito um cálculo para descobrir um limite de tolerância de peso que a heurística pode usar para coletar sucessivos itens sem a necessidade de um extensivo recálculo e reordenação dos efeitos reais dos itens, sendo ambos realizados apenas quando a coleta de um item fizer exceder essa tolerância de peso calculada. Essa reavaliação dos impactos dos itens enquanto os mesmos estiverem sendo selecionados para coleta não é realizada pela heurística TPPH, o que faz com que o desempenho da mesma seja inferior quando comparada com a TPPHx. Há uma verificação na linha 19 do Algoritmo 4 que mostra o funcionamento dessa reavaliação dos impactos dos itens feita pela heurística, onde é usado o parâmetro informado a ela para poder definir quando essa reavaliação se faz necessário: o objetivo é verificar se os itens remanescentes na lista de itens

disponíveis precisam ter seus impactos recalculados e a lista reordenada, ocorrendo esse evento somente se uma determinada porcentagem da capacidade máxima da mochila for preenchida após a última reordenação, caso contrário, o algoritmo continuará sua execução com a mesma ordem na lista de itens usada antes do último item coletado.

**Algoritmo 4** – Heurística Construtiva TPPHx baseada na TPPH com reavaliação dos impactos dos itens de acordo com o peso da mochila e da taxa de atualização fornecida para ela

---

```

1: função TPPHx(rota, itens, N, W, taxaDeReavaliacao)
2:   planoDeColeta  $\leftarrow \emptyset$ 
3:   pesoCicloAtual  $\leftarrow 0$ 
4:   w  $\leftarrow 0$ 
5:   toleranciaSemReavaliar  $\leftarrow W \times taxaDeReavaliacao$ 
6:   avaliacaoTTPBase  $\leftarrow avaliarTTP(rota, planoDeColeta)$ 
7:   para i  $\leftarrow 1$  até N faça
8:     avaliacaoTTPComItem  $\leftarrow avaliarTTP(rota, planoDeColeta \cup itens[i])$ 
9:     itens[i].imp  $\leftarrow (avaliacaoTTPComItem - avaliacaoTTPBase) \div itens[i].peso$ 
10:  fim para
11:  Ordenar decrescentemente os itens pelos seus impactos
12:  enquanto  $((N \geq 1) \wedge (w < W) \wedge (itens[1].imp > 0))$  faça
13:    item  $\leftarrow$  RemoverItem(itens[1])
14:    N  $\leftarrow N - 1$ 
15:    se (item.peso + w  $\leq W$ ) então
16:      planoDeColeta  $\leftarrow$  item
17:      w  $\leftarrow w + item.peso$ 
18:      pesoCicloAtual  $\leftarrow pesoCicloAtual + item.peso$ 
19:      se (pesoCicloAtual  $\geq$  toleranciaSemReavaliar) então
20:        Reavaliar os impactos dos itens e reordenar a lista
21:        pesoCicloAtual  $\leftarrow 0$ 
22:      fim se
23:    fim se
24:  fim enquanto
25:  retorna planoDeColeta
26: fim função

```

---

## 4.2. Look and Construct Simulated Annealing

A abordagem proposta nesta dissertação para solucionar o PMV é baseada em uma meta-heurística que faz uso dos algoritmos descritos anteriormente nessa seção, sendo o *Simulated Annealing* escolhido para essa função por ser uma meta-heurística probabilística não populacional capaz de realizar de forma diversificada uma busca local usando uma estrutura de vizinhança de maneira mais eficiente do que tratando diversas soluções em conjunto. Porém, algumas modificações em relação a sua versão clássica foram realizadas para adaptar a mesma à natureza multicomponente do problema: mais

de um subproblema é trabalhado simultaneamente e possuem suas soluções combinadas de forma que o espaço de busca se tornou muito grande para ser explorado sem definir critérios, heurísticos, que possam guiar a busca de maneira a encontrar bons vizinhos sem a necessidade de avaliar uma grande quantidade de soluções.

O nome *Look and Construct Simulated Annealing* (em português: Arrefecimento Simulado – Olhar e Construir) é usado porque a abordagem aqui proposta não possui apenas o uso de uma busca local em uma estrutura de vizinhança: estruturas de vizinhança que trabalham afetando os dois componentes ao mesmo tempo são grandes demais e geram muitos vizinhos de baixa qualidade, fazendo com que para serem exploradas de forma eficiente seja necessário restringir a probabilidade de aceitação de soluções inferiores e permitir que o algoritmo as avalie em maior escala, demandando um grande tempo de execução. Para diminuir o número de vizinhos e o tempo de busca, a heurística TPPHx, que é um algoritmo guloso para elaboração de um plano de coleta para uma dada rota do PMV, é usada em conjunto a uma exploração da estrutura de vizinhança 2-OPT aplicada somente sobre o componente PCV do problema, sendo que o componente da mochila é solucionado através do uso da heurística construtiva. Esse processo pode ser entendido através do Algoritmo 5 mostrado abaixo, porém, o mesmo possui diversas estratégias que são explicadas nas próximas subseções.

A ideia geral do Algoritmo 5 é permitir que uma solução inicial seja aperfeiçoada através de uma busca local 2-OPT usando a rota da solução atual do LaCSA como base. Após gerados os vizinhos, esses são analisados e os que forem considerados promissores terão seu componente da mochila criado um a um através de sucessivas chamadas à heurística TPPHx. Se algum vizinho encontrado for melhor do que a melhor solução encontrada até o momento, o mesmo é escolhido como melhor solução e também como solução atual do LaCSA, sendo a próxima iteração da busca baseada nele. Caso contrário, mesmo um vizinho não mostrando melhorias em relação à melhor solução encontrada, ele terá uma certa probabilidade de ser aceito. Após esse processo, a temperatura atual da busca é atualizada com base no que ocorreu nessa iteração, e, se o tempo de busca avançar o suficiente, a taxa de atualização da mochila fornecida para a heurística TPPHx é também atualizada, permitindo que mais tempo seja gasto na elaboração de um plano de coleta para corrigir pequenos erros de coletas de itens que possam não ser vantajosos. Todos os componentes do algoritmo LaCSA são explicados em detalhes nas próximas subseções.

---

**Algoritmo 5** – Look and Construct Simulated Annealing (LaCSA)

---

1: **função** LaCSA(*solucaoInicial*, *tempMax*, *tempMin*, *alpha*)

---

2:     *melhorSolucao* ← *solucaoInicial*  
3:     *solucaoAtualSA* ← *solucaoInicial*  
4:     **enquanto** criterio de parada não for atingido **faça**  
5:         *temp* ← *tempMax*

```

6:      enquanto temp > tempMin faça
7:          taxaMochila ← AtualizarTaxaMochila(tempoRestante)
8:          vizinhosTSPV alidos ← GerarVizinhosTSPValidos(solucaoAtualSA)
9:          vizinhosTSP ← ObterBonsVizinhos(vizinhosTSPV alidos)
10:         vizinhosTSP ← OrdenarVizinhosPeloComponenteTSP(vizinhosTSP)
11:         enquanto existir vizinho TSP não verificado faça
12:             vizinhoSelecionado ← RemoverMelhorVizinhoTSP(vizinhosTSP)
13:             vizinhoTTP ← TPPHx(vizinhoSelecionado,taxaMochila)
14:             se avaliar(vizinhoTTP) > avaliar(melhorSolucao) então
15:                 melhorSolucao ← vizinhoTTP
16:                 solucaoAtualSA ← vizinhoTTP
17:                 break
18:             senão
19:                 se CriterioDeAceitacao(temp,vizinhoTTP) então
20:                     solucaoAtualSA ← vizinhoTTP
21:                     break
22:             fim se
23:         fim se
24:     fim enquanto
25:     temp ← AjustarTemperatura()
26: fim enquanto
27: fim enquanto
28:     retorna melhorSolucao
29: fim função

```

---

#### 4.2.1. Solução Inicial

Para criar uma solução inicial a rota é criada de forma independente, mas, para definir o plano de coleta, a rota é levada em consideração. A heurística de Lin-Kernighan (1973) foi usada para criar uma rota válida para o componente PCV sem levar em consideração os itens das cidades, sendo essa heurística amplamente conhecida por conseguir obter resultados próximos do ótimo para esse problema. Em seguida, essa rota é fornecida para uma heurística construtiva (TPPHx) que define quais itens serão coletados, modificando somente o componente PM da solução. Foram realizados testes computacionais com diferentes heurísticas de criação de rotas para a geração da solução inicial: vizinho mais próximo, seleção aleatória de cidades, o Algoritmo de Kruskal (KRUSKAL, 1956) adaptado para as restrições do PCV e o Algoritmo de Economias (CLARKE; WRIGHT, 1964). O Algoritmo de Kruskal e o Algoritmo de Economia encontraram rotas melhores do que a heurística do vizinho mais próximo, e a heurística de seleção aleatória de cidades para a criação da rota foi a que se saiu pior. A heurística de Lin-Kernighan (LK) conseguiu resultados significativamente superiores para uma solução somente com o componente PCV e também para a versão final da solução inicial, contendo a mochila e sendo uma solução PMV completa.

Duas conclusões puderam ser tiradas com base nesse estudo das possibilidades de soluções iniciais:

- I. A seleção aleatória de cidades para criação de uma rota criou rotas muito ruins, o que era esperado, e que resultaram em soluções PMV também ruins. Essa é apenas a confirmação de uma conclusão talvez bastante intuitiva: uma rota ruim pode levar muito tempo para ser percorrida, aumentando assim o custo da viagem até que nenhum plano de coleta consiga reverter o prejuízo com seus itens coletados.
- II. A heurística de LK quando usada para criar a rota da solução inicial levava às soluções PCV e PMV de qualidade significativamente superiores a qualquer outra, influenciando completamente o resultado do LaCSA.

Devido a essas conclusões, a heurística LK foi selecionada para ser usada como parte da heurística construtiva da solução inicial. Além disso, seu resultado é tão importante que 10% do tempo total da busca é dedicado apenas para ela conseguir encontrar uma solução PCV de alta qualidade. A heurística LK é estocástica, portanto, a aleatoriedade influencia em seu resultado e a cada execução ela pode encontrar uma rota diferente: até completar 10% do tempo limite, essa heurística é executada e para cada rota obtida é criada uma solução PMV contendo um plano de coleta criado através da heurística TPPHx. Todo o processo para definir uma solução inicial resultou em uma heurística chamada de CHTTP (*Constructive Heuristic for the Travelling Thief Problem*), sendo ela descrita no Algoritmo 6 e seu resultado escolhido como solução inicial para a LaCSA.

**Algoritmo 6** – Heurística Construtiva para o Problema do Mochileiro Viajante

---

```

1: função CHTTP(idades, itens, N, W)
2:   enquanto (tempo de execução não atingir 10% do tempo limite) faça
3:     rotaLK ← LK(idades)
4:     planoDeColeta ← TPPHx(rotaLK, itens, N, W, 0.21)
5:     solucaoTTP ← rotaLK ∪ planoDeColeta
6:     se (avaliar(solucaoTTP) > avaliar(melhorSolucaoTTP)) então
7:       melhorSolucaoTTP ← solucaoTTP
8:   fim se
9:   fim enquanto
10:  retorna melhorSolucaoTTP
11: fim função

```

---

#### 4.2.2. Critério de parada

O critério de parada usado para executar o algoritmo é o mesmo definido em todos os outros trabalhos existentes na literatura: 10 minutos de tempo de execução. Entretanto,

1 minuto desse tempo total é dedicado para que a CHTTP possa encontrar uma solução inicial que será fornecida para a LaCSA, sendo os 9 minutos restantes do tempo de busca gastos por esta abordagem com a finalidade de melhorar a qualidade da solução fornecida pela heurística construtiva.

Antes de haver qualquer proposta de abordagens para solucionar o PMV, quando apenas a definição formal do mesmo era encontrada na literatura, houve uma competição no CEC<sup>1</sup> usando esse problema como tema, onde o critério de parada dos algoritmos submetidos deveriam ser de 10 minutos de tempo de execução, conseqüentemente, todos os trabalhos presentes na literatura mantiveram esse critério de parada para facilitar as comparações dos resultados encontrados por diversas abordagens.

### 4.2.3. Taxa de Atualização da Mochila

O uso da taxa de atualização é a principal diferença entre as heurísticas TPPH e TPPHx, sendo que nesta há o uso dessa taxa para que a mesma consiga determinar em sua execução quais são os momentos em que os itens precisam ter seu impacto na solução reavaliados e em seguidas reordenados para serem coletados, característica esta ausente na TPPH, onde há apenas uma avaliação e ordenação realizada antes de iniciar as escolhas dos itens a serem coletados. Essa taxa permite não apenas que a heurística TPPHx, diferente da TPPH, consiga atualizar esse impacto dos itens e, assim, realize melhores escolhas na elaboração de um plano de coleta: o uso dessa taxa de forma variável permite também que essa heurística consiga ser usada de forma eficiente pela meta-heurística LaCSA.

Ao usar a heurística TPPHx com uma taxa de atualização baixa, como, por exemplo, exigindo que os itens sejam reavaliados a cada 1% que a capacidade total da mochila for preenchida, ocasionará em aproximadamente 100 reavaliações de cada item da lista de itens e 100 reordenações da mesma, que, por sua vez, pode conter uma quantidade significativamente grande de itens. O LaCSA usa uma taxa de atualização dinâmica de acordo com o tempo de busca fornecido a ela: 1 minuto é gasto apenas para encontrar a solução inicial, portanto, os outros 9 minutos de busca são usados pela LaCSA, que, por sua vez, controla qual o valor da taxa de reavaliação que deve ser usada em suas iterações baseando-se no tempo de busca consumido até o momento, usando os seguintes valores:

**Tabela 1** – Mudança na Taxa de Atualização usada pelo LaCSA de acordo com o tempo de busca em segundos

Período de Tempo	Taxa de Atualização
0 60	-
60 120	21,0%

<sup>1 1</sup> Competição do CEC 2014 sobre o PMV: <http://cs.adelaide.edu.au/~optlog/CEC2014Comp/>

120 180	18,0%
180 240	16,0%
240 300	15,0%
300 360	13,0%
360 420	11,8%
420 480	10,6%
480 540	09,5%
540 600	02,0%

**Fonte:** Elaborada pelo autor

Como visto na Tabela 1, a LaCSA inicia sua busca a partir dos 60 segundos gastos pela heurística CHTTP e é usada uma taxa de reavaliação de 21%, isto é, há uma tolerância de peso equivalente a 21% da capacidade total da mochila em que os itens não precisam ser reavaliados enquanto essa tolerância não for alcançada. Essa tolerância é diminuída conforme a busca vai sendo realizada, até alcançar um mínimo de tolerância equivalente a 2% da capacidade da mochila quando a busca está chegando em seu fim.

Esse uso de uma taxa de atualização dinâmica de acordo com o tempo de busca restante permitiu que uma maior quantidade de iterações da LaCSA ocorresse, pois, a diferença na quantidade de reavaliações de todos os itens disponíveis pode cair de 50 (tolerância de 2% da capacidade da mochila preenchida) para, no máximo, 8 (21% da capacidade) vezes por execução da heurística. Se nenhum limite mínimo de tolerância (2%) sem atualização for fornecido, a heurística pode chegar a reavaliar os seus itens  $N$  vezes, sendo  $N$  a quantidade de itens selecionadas para coleta durante sua execução. O uso dessa estratégia tornou viável solucionar instâncias com 280 cidades e 1395 itens. Sem ela, para essas instâncias maiores, o algoritmo não conseguia realizar iterações suficientes para melhorar a solução inicial, havendo uma diferença significativa após a adoção desse dinamismo na taxa de atualização dos itens fornecida para a heurística TPPHx.

Na seção 6.1 são apresentados resultados sobre o uso das heurísticas TPPH e TPPHx com diversas taxas, onde, através de uma análise estatística, os resultados revelaram que o uso dessa taxa de reavaliação pela LaCSA de forma dinâmica (variável em relação ao tempo) permite que a busca seja acelerada sem ser guiada por maus caminhos em suas iterações iniciais: quando a taxa de reavaliação usada é mais alta, com o objetivo de tornar a busca mais rápida, cria-se a possibilidade da mesma se tornar menos precisa, entretanto, os resultados analisados indicam que na prática não é isso que ocorre.

#### 4.2.4. Estrutura de vizinhança

A estrutura de vizinhança usada pela LaCSA é a estrutura 2-OPT para soluções representadas por rotas, nesse caso, apenas o componente PCV é perturbado, ficando sob a responsabilidade da heurística TPPHx selecionar um plano de coleta para cada um dos vizinhos considerados promissores, transformando-os assim em soluções completas e válidas para o PMV. Com o objetivo de otimizar o tempo necessário para realizar esse processo, duas estratégias de redução no número de vizinhos gerados foram usadas, de forma a diminuir o espaço de busca do problema, sendo essas estratégias usadas para:

- I. Evitar que sejam gerados vizinhos 2-OPT que dificilmente seriam boas soluções para o PCV, ou seja, vizinhos com rotas de grande distância.
- II. Diminuir a quantidade de vizinhos 2-OPT que teriam seu componente da mochila criado através da heurística TPPHx, diminuindo assim o número de vezes que essa heurística fosse executada.

A primeira estratégia derivou-se da ideia de usar o conceito da Triangulação de Delaunay (DELAUNAY, 1934) para limitar a quantidade de vizinhos 2-OPT gerados para o problema (YI MEI et al, 2014a), entretanto, por mais que essa estratégia tenha se mostrado benéfica quando aplicada para resolver o TSP (KRASNOGOR, 1995), o mesmo não ocorreu quando ela foi usada pelo LaCSA para solucionar o PMV: os resultados mostraram uma queda de qualidade, possivelmente ocorrida devido ao corte de bons vizinhos PMV que não passaram pela restrição imposta na Triangulação de Delaunay que considera apenas o componente PCV. Entretanto, outra estratégia usada para não gerar vizinhos que não fossem interessantes funcionou bem com o problema: foram removidos da estrutura de vizinhança 2-OPT todos aqueles vizinhos cujas cidades escolhidas para serem conectadas estejam a uma distância que exceda a um limite imposto para cada uma delas, isto é, existirão agora vizinhos *inviáveis*, sendo eles aqueles compostos por conexões entre um par de cidade que possua uma distância considerada como grande demais para ser interessante sua existência. Cada cidade tem seu próprio limite de distância a qual ela pode se conectar com outras cidades, impedindo assim que duas cidades que estão longes uma da outra sejam escolhidas para serem conectadas, evitando dessa forma gerar um vizinho 2-OPT de baixa qualidade. Um total de 20% das (piores) conexões que envolvam cada uma das duas cidades escolhidas para serem conectadas são o que compõe esse grupo de conexões que se enquadram fora do limite de distância permitido, sendo que esse valor foi encontrado através de experimentos computacionais que mostraram uma diminuição no tempo de cada iteração do LaCSA sem impactar em perda de qualidade na solução encontrada, sendo essa mais uma evidência da importância em encontrar boas rotas para serem testadas como soluções PMV.

A segunda estratégia foi necessária porque o tempo gasto para criar o componente da mochila para cada vizinho 2-OPT, mesmo esses tendo seu número reduzido em 20%, estava atrapalhando com o desempenho do LaCSA de forma geral, tornando essa abordagem inviável para as instâncias com uma maior quantidade de cidades e/ou itens. Conhecendo o problema e sabendo que soluções PCV com tempo de percurso elevado dificilmente se tornam boas soluções para o PMV (OLIVEIRA et al, 2015), foi implementada uma redução na quantidade de vizinhos PMV completos da seguinte forma: apenas os 1000 melhores vizinhos PCV (com menores rotas) terão sua parte da solução pertencente ao componente da mochila criada através da heurística TPPHx. Por mais que seja uma estratégia que poderia ocasionar em cenários onde bons vizinhos fossem ignorados, após a realização de experimentos usando esse limite no número de vizinhos foi percebido uma melhora significativa em relação ao não uso do mesmo: acontece que o PMV é um problema com um espaço de busca muito grande e que o LaCSA consegue obter melhores resultados verificando o que considera ser a parte mais promissora desse espaço por uma maior quantidade de iterações do que verificando indiscriminadamente todos os vizinhos 2-OPT gerados, sendo que para isso é necessário usar uma outra heurística para cada um deles, ocasionando em um número menor de iterações.

#### **4.2.5. Critério de Aceitação**

O critério de aceitação escolhido aqui é o mesmo usado por Metropolis (1953) e é descrito com detalhes na seção 3.3.2.1.

#### **4.2.6. Ajuste da Temperatura**

Existem três informações capazes de afetar o desempenho do algoritmo de forma significativa, sendo elas: temperatura inicial, temperatura mínima e taxa de resfriamento. O impacto que cada uma delas causam no algoritmo LaCSA e a forma como foram definidas é descrita abaixo, porém, a análise estatística citada a seguir é mostrada em detalhes apenas no capítulo 5 deste trabalho.

A temperatura inicial necessitou ser definida de forma adaptativa em relação a solução inicial informada para o Simulated Annealing. Essa estratégia se fez necessária porque o PMV possui instâncias com tamanho e características completamente diferentes entre si, fazendo com que uma solução possa ter um lucro menor que 10.000,00 para um cenário e para outro conseguir alcançar 200.000,00: se usada a mesma temperatura inicial para ambos, o algoritmo se comportará de forma diferente para cada cenário, pois aceitará com a mesma probabilidade uma solução com diferença de valor de -5000 para ambos, sendo que para o primeiro essa é uma diferença muito alta, porém,

para o segundo, não tanto. Para evitar esse problema, optou-se por fazer com que a temperatura inicial fosse definida com o valor de 0,8% do valor retornado pela função objetivo ao analisar a solução inicial entregue para o LaCSA. Esse valor (0,8%) foi encontrado como sendo o melhor a ser usado pelo algoritmo após uma série de experimentos computacionais e análises estatísticas relatados no capítulo 5 deste trabalho.

A taxa de queda de temperatura poder-se-ia ser definida de diversas formas diferentes, como, por exemplo, uma queda linear, exponencial ou logarítmica (KRASNOGOR et al, 1998). A abordagem LaCSA possui como critério de parada 10 minutos de execução, entretanto, seu espaço de busca é muito grande e o tempo necessário para explorá-lo pode ser muito diferente para as instâncias com os diversos tamanhos encontradas na literatura, devido a isso, optou-se por usar uma taxa de queda exponencial para que a busca possa começar com uma temperatura maior, explorando de forma mais diversificada o espaço de busca, mas não permitindo que a mesma se perca em mínimos locais de baixa qualidade ou não consiga realizar iterações o suficiente para chegar em sua temperatura mínima. Através de experimentos computacionais e análises estatísticas dos resultados coletados (capítulo 5), concluiu-se que a melhor taxa de queda de temperatura a ser usada pelo algoritmo deve ser equivalente a 30% de queda na temperatura atual por iteração.

A temperatura mínima é que define quando o LaCSA deve ser reaquecido após uma série de buscas que ocasionem em queda de temperatura grande o suficiente para que a busca se torne similar a uma busca local *Hill-Climbing*. Entretanto, como discutido acima, os valores retornados pela função objetivo ao analisar soluções para cenários distintos pode ser muito diferente, dificultando assim a detecção de quando o algoritmo começa a se comportar como uma busca local sem diversificação. Para resolver esse problema, a temperatura mínima aqui proposta é adaptativa e tem seu valor definido através I) da temperatura inicial e II) do critério de aceitação:  $T_{min} = \sqrt[6]{T_0}$ . Esse valor não precisou ser calibrado devido a equação conseguir distribuir a probabilidade de aceitação das soluções de forma satisfatória independente do tamanho ou da característica da instância sendo solucionada.

A probabilidade do algoritmo aceitar soluções piores usando todos os critérios definidos acima são mostradas nas Tabela 2 e 3, onde é possível observar que soluções com diferenças iguais a -500, -1000, -2500 ou -5000 em relação à melhor solução possuem diferentes probabilidades de serem aceitas se a instância sendo solucionada possuir características que permitem encontrar soluções com valores de 60.000 (Tabela 2) ou 150.000 (Tabela 3), sendo esse comportamento benéfico para o algoritmo, pois há uma grande diferença de diferenças quando se cai de 60 mil para 55 mil em relação a cair de

150 mil para 145 mil: por mais que a diferença absoluta seja a mesma (-5000), a solução de 60 mil sofreu uma queda proporcional de qualidade maior do que a de 150 mil.

**Tabela 2** – Probabilidade do LaCSA aceitar uma solução inferior para uma respectiva temperatura baseada em uma solução com 60 mil de lucro para o PMV

Ta	-5000	-2500	-1000	-500	-100
4800,00	35%	59%	81%	90%	98%
3360,00	23%	48%	74%	86%	97%
2352,00	12%	35%	65%	81%	96%
1646,40	5%	22%	55%	74%	94%
1152,48	1%	12%	42%	65%	92%
806,74	0%	5%	29%	54%	88%
564,72	0%	1%	17%	41%	84%
395,30	0%	0%	8%	28%	78%
276,71	0%	0%	3%	17%	70%
193,70	0%	0%	1%	8%	60%
135,59	0%	0%	0%	3%	48%
94,91	0%	0%	0%	1%	35%
66,44	0%	0%	0%	0%	22%
46,51	0%	0%	0%	0%	12%
32,55	0%	0%	0%	0%	5%

Fonte: Elaborada pelo autor

**Tabela 3** - Probabilidade do LaCSA aceitar uma solução inferior para uma respectiva temperatura baseada em uma solução com 150 mil de lucro para o PMV

Ta	-5000	-2500	-1000	-500	-100
12000,00	66%	81%	92%	96%	99%
8400,00	55%	74%	89%	94%	99%
5880,00	43%	65%	84%	92%	98%
4116,00	30%	55%	78%	89%	98%
2881,20	18%	42%	71%	84%	97%
2016,84	8%	29%	61%	78%	95%
1411,79	3%	17%	49%	70%	93%
988,25	1%	8%	36%	60%	90%
691,78	0%	3%	24%	49%	87%
484,24	0%	1%	13%	36%	81%
338,97	0%	0%	5%	23%	75%
237,28	0%	0%	1%	12%	66%
166,10	0%	0%	0%	5%	55%
116,27	0%	0%	0%	1%	42%
81,39	0%	0%	0%	0%	29%
56,97	0%	0%	0%	0%	17%
39,88	0%	0%	0%	0%	8%

Fonte: Elaborada pelo autor

Se uma nova solução é aceita como solução atual do algoritmo, o mesmo interrompe sua iteração e aplica a queda de temperatura definida. Sua execução segue dessa forma, iteração após iteração, até que o critério de parada seja atingido ou a

temperatura caia até uma temperatura mínima definida através da equação mostrada acima. Esse é o comportamento padrão do algoritmo, entretanto, para algumas instâncias o algoritmo se movia para certos locais do espaço de busca onde não conseguia sair devido ao tempo gasto para isso ser possível em sua atual temperatura, portanto, para resolver essa deficiência, foi implementado um mecanismo que eleva a temperatura se ocorrer o cenário descrito, permitindo assim que a diversificação probabilística do LaCSA possa movê-lo da região do espaço de busca onde o mesmo se localiza: foi implementada uma estratégia de reaquecimento da temperatura antes que a mesma alcançasse seu valor mínimo, bastando o algoritmo não encontrar mais do que duas soluções, entre seus vizinhos atuais, com 5% de chance de serem aceitas para que sua temperatura atual fosse elevada novamente para sua temperatura inicial, aumentando assim sua chance de sair desse dessa região e começar a explorar novas soluções.

### **4.3. Outras Abordagens**

Nesta dissertação a abordagem LaCSA tem suas características e resultados fortemente analisados por ter sido capaz de encontrar boas soluções para o problema, entretanto, ela não foi a única abordagem desenvolvida pelo autor deste trabalho. Inicialmente uma Busca Tabu (OLIVEIRA et al, 2015) foi proposta e conseguiu, na época, encontrar resultados melhores do que os encontrados por outras abordagens presentes na literatura naquele momento. Além da Busca Tabu e do Simulated Annealing, foram também implementadas outras meta-heurísticas, que não foram capazes de encontrar resultados melhores do que os existentes na literatura: Algoritmo Genético, Algoritmo Memético, Scatter-Search e uma Iterated Greedy (IG) em conjunto a uma busca Iterated Local Search (ILS).

## 5. EXPERIMENTOS COMPUTACIONAIS

Neste capítulo serão apresentados os resultados de todos os testes computacionais realizados com as heurísticas propostas. As instâncias usadas são descritas na seção 5.1. A calibração para encontrar o melhor conjunto de valores para os parâmetros e a efetividade de algumas heurísticas são encontradas na seção 5.2.

Todos os testes computacionais foram realizados em um computador com a CPU AMD A8 7600 Quad Core 3.1 GHz, 4 GB memória RAM e o sistema operacional Windows 8.1. Os algoritmos propostos nesta dissertação foram programados usando a linguagem C++, cujos parâmetros (-o e -march) de compilação foram configurados para favorecer o desempenho do executável em relação à arquitetura do processador usado, elevando assim a velocidade de processamento pelo mesmo.

### 5.1. Instâncias

Para iniciarem os testes computacionais sobre o PMV, Polyakovskiy *et al* (2014) criaram 9720 instâncias dos mais diversos tipos. Foram criadas 10 instâncias com variação no custo do aluguel e na capacidade da mochila para cada uma das seguintes três características: mesma quantidade de cidades, mesma quantidade de itens por cidade e mesmo tipo de itens disponíveis. Para cada número de cidades diferentes existem instâncias com 1, 3, 5 ou 10 itens por cidade, e para cada um desses subgrupos existem instâncias com as seguintes três características de itens disponíveis:

- *bounded-strongly-correlated* (**bsc**): os valores dos itens estão altamente relacionados aos seus pesos.
- *uncorrelated* (**u**): os valores dos itens não estão relacionados ao seus pesos.
- *uncorrelated-similar-weights* (**usw**): os valores dos itens não estão relacionados ao peso, mas os pesos de todos os itens são similares.

Um exemplo de instância é a *eil51\_n50\_bsc\_10* e sua leitura pode ser feita da seguinte forma: é a décima instância (\_10) do grupo *eil* com 51 cidades e 50 itens com valor e peso fortemente relacionados. Dentro do grupo *eil* existem também 10 instâncias com 150 itens, 250 itens e 500 itens, sendo estas as instâncias com 3, 5 e 10 itens por cidade, respectivamente. Essas 10 instâncias, enumeradas de 1 a 10, para cada grupo de características diferentes, foram criadas modificando apenas a capacidade da mochila disponibilizada para o mochileiro e o custo de seu aluguel por unidade de tempo. Foram escolhidas as de numerações 3 e 7, por serem elas as instâncias que foram estudadas em outros trabalhos disponíveis na literatura. Há também instâncias do tipo *u*, *usw* e *bsc* para cada número de cidades e de itens diferentes.

## 5.2. Experimentos para Configuração dos Parâmetros

Para definir qual o melhor conjunto de valores para os parâmetros “taxa de temperatura” e “taxa de resfriamento” ( $\alpha$ ) da abordagem LaCSA foram realizados testes estatísticos paramétricos e não paramétricos nos resultados coletados através dos experimentos efetuados sob uma amostra das instâncias que serão solucionadas por esse algoritmo. Os testes mostraram que a escolha de certos valores para esses parâmetros exerce uma diferença estatisticamente significativa sobre os resultados finais encontrados pelo LaCSA. Os detalhes são mostrados a seguir.

### 5.2.1. Instâncias

Foram selecionadas 24 instâncias para serem usadas como parte dos experimentos responsáveis pela coleta dos resultados a serem analisados para definir os valores para os parâmetros, sendo esse o critério de escolha dessas instâncias: duas instâncias ( $\times 2$ ) de 76 e 195 cidades ( $\times 2$ ), com 3 e 5 itens por cidade ( $\times 2$ ), dos tipos *bsc*, *u* e *usw* ( $\times 3$ ). As duas instâncias escolhidas que atendem cada uma das permutações acima foram a terceira (*\_3*) e a sétima (*\_7*) de cada um dos grupos. No capítulo 6 há a análise dos resultados finais obtidos pelo LaCSA em relação aos existentes na literatura e é possível encontrar resultados pertencentes às instâncias citadas acima e também de outras instâncias com tamanhos diferentes (51, 76, 100, 225 e 280 cidades).

### 5.2.2. Planejamento dos Experimentos

Com as instâncias definidas, o próximo passo é definir quais os valores para os parâmetros serão testados para então realizar a análise dos resultados. Através de testes empíricos menos formais, foram definidos os seguintes valores: temperatura = {0,8%, 1,1%, 1,4%, 1,7%} e  $\alpha = \{-10\%, -20\%, -30\%, -40\%\}$ . A temperatura está com seu valor em porcentagem, mas seu valor real é equivalente a essa porcentagem do valor da solução inicial e o motivo do uso desse recurso é descrito no capítulo 4. Como o valor de  $\alpha$  escolhido é baseado em porcentagem, significará que a temperatura perderá  $\alpha$  (%) de seu valor atual por iteração, sendo essa perda baseada no valor da taxa escolhida para o parâmetro, havendo, portanto, uma perda maior de temperatura (em seu valor absoluto) quando a mesma estiver alta e menor quando estiver baixa.

Para a calibração dos parâmetros foram realizadas 15 execuções independentes com cada uma das 16 permutações possíveis com os valores de temperatura e  $\alpha$  escolhidos para os testes, totalizando em 240 execuções para cada uma das 24 instâncias. O critério de parada é atingir 10 minutos de execução, sendo o mesmo que foi usado na competição

do CEC<sup>2</sup> 2014<sup>3</sup>/2015<sup>4</sup> para solucionar o PMV e mantido também em outros trabalhos disponíveis na literatura sobre o tema.

### 5.2.3. Conjunto de Testes Estatísticos para Problemas de Otimização

De forma a descobrir, com certo grau de confiança, se há algum impacto ocasionado pela escolha dos valores dos parâmetros em relação ao desempenho do LaCSA, os resultados coletados pelos experimentos podem ser submetidos a um teste de análise de variância, como o teste paramétrico ANOVA (KIRK, 1982) e, se acusar alguma diferença entre os tratamentos (média dos resultados para cada parâmetro usado) em relação aos fatores (parâmetros), pode-se, então, aplicar um teste *post hoc* como o de Tukey (TUKEY, 1953) ou o de Scott-Knott (SCOTT e KNOTT, 1974) que usam a análise de variância para realizarem comparações entre cada dupla de configuração de parâmetros possíveis e indica quais dentre elas são capazes de encontrar resultados com diferença estatisticamente significativa uma da outra, sendo essa uma ferramenta de grande apoio na garantia de uma boa escolha de valores para os parâmetros.

Para efetuar os testes estatísticos apresentados neste capítulo foi usada a plataforma R<sup>5</sup>, com sua linguagem e ambiente de programação fornecendo meios para a realização dos testes de forma mais segura e eficiente do que realizar seus cálculos manualmente.

#### 5.2.3.1. Normalização dos Resultados

O processo citado acima é uma das formas válidas de realizar um teste estatístico capaz de analisar os resultados coletados e fornecer uma resposta confiável sobre qual(is) valor(es) de parâmetro(s) pode(m) ser usado(s) de forma a maximizar a qualidade do resultado final da busca, porém, os resultados coletados não podem ser analisados sem previamente passarem por algum tipo de transformação de dados, aceita pelo teste *ANOVA*, para corrigir um problema ocasionado pelas diferenças de características contidas nas instâncias analisadas. Na Tabela 4 são mostrados alguns resultados de algumas dessas instâncias que teriam seus valores reunidos para serem analisados.

---

<sup>2</sup> Mais informações no site da conferência: <http://ieeexplore.ieee.org/servlet/opac?punumber=1000284>

<sup>3</sup> Competição do CEC 2014 sobre o PMV: <http://cs.adelaide.edu.au/~optlog/CEC2014Comp/>

<sup>4</sup> Competição do CC 2015 sobre o PMV, porém, cancelada por não obter um número mínimo de participantes: <http://cs.adelaide.edu.au/~optlog/CEC2015Comp/>

<sup>5</sup> Para mais informações sobre a linguagem **R** e seu ambiente de desenvolvimento, acesse: <https://www.r-project.org/>

**Tabela 4** – Resultados reais para algumas instâncias do PMV destacando a diferença entre os valores

<b>INSTÂNCIA</b>	<b>RESULTADO</b>
eil76_n225_bsc_03	26046,50
eil76_n225_bsc_07	34991,00
eil76_n225_u_03	13533,40
eil76_n225_usw_03	8514,07
eil76_n375_bsc_07	60941,50
rat195_n582_bsc_03	87358,90
rat195_n582_u_03	57091,80
rat195_n582_usw_03	31621,70
rat195_n582_usw_07	52793,20
rat195_n970_bsc_07	197794,00
rat195_n970_u_03	92656,30
rat195_n970_usw_03	51558,50
rat195_n970_usw_07	93293,50
<b>MÉDIA:</b>	<b>62168,80</b>
<b>DESVIO PADRÃO:</b>	<b>49476,88</b>

**Fonte:** Elaborada pelo autor

Os resultados mostrados na Tabela 4 contêm dados que indicam uma grande diferença nos valores das soluções para instâncias com tamanhos e/ou características diferentes. Se esses dados forem usados dessa maneira por um teste de análise de variância, considerando que todas as instâncias são igualmente importantes, os resultados das instâncias que possuem valores absolutos maiores teriam um peso maior de influência para decidir qual conjunto de parâmetros se saiu melhor. Para corrigir esse problema, foi aplicada uma normalização Max-Min Equalizada (do inglês: *Feature Scaling*) nos resultados, transformando-os de  $X$  para  $X'$  usando a seguinte equação:  $X' = (X - X_{min}) / (X_{max} - X_{min})$ , sendo  $X$  o resultado original ao qual se quer aplicar a normalização,  $X_{min}$  o menor resultado encontrado para a mesma instância de  $X$  e  $X_{max}$  o maior resultado encontrado também para essa instância, não importando qual a configuração dos parâmetros usados que conseguiram encontrar tais resultados, devendo-se restringir somente às instâncias com características diferentes. Ao aplicar tal normalização, os resultados de cada instância terão agora a mesma base de valores entre 0 e 1, sendo que os valores normalizados que mais se aproximarem de 1 serão aqueles que mais se aproximarem do melhor resultado encontrado, sendo o valor 1 reservado para as execuções que obtiverem esse resultado. Em contrapartida, os valores normalizados que estiverem perto de 0 serão os piores valores encontrados, sendo o valor 0 reservado para as execuções que encontrarem a pior solução encontrada por todas as execuções nessa instância.

### 5.2.3.2. ANOVA

Após realizar a normalização nos resultados foi possível realizar uma análise de variância sobre eles. O teste *ANOVA* é um teste paramétrico que permite testar duas diferentes hipóteses, sendo elas definidas abaixo. A hipótese nula (*H0*) representa a informação de que todas as médias analisadas são iguais, indicando assim que não há diferença estatisticamente significativa entre as populações analisadas. Os resultados fornecidos pelo teste *ANOVA* contém informações que podem rejeitar ou falhar em rejeitar a hipótese nula, sendo possível, através deles, responder o seguinte questionamento: “Há pelo menos uma configuração de parâmetros que quando usada muda o resultado do algoritmo?”.

$$H0: M_1 = M_j \quad \forall (2 = 1 \dots n)$$

$$H1: M_i \neq M_j \quad \exists (i = 1 \dots n \mid j = 1 \dots n \mid i \neq j)$$

Como os dados de entrada para o teste foram os resultados das soluções (tratamento) encontradas por cada repetição independente do algoritmo separados pelos valores dos parâmetros (fatores) usados para consegui-los, rejeitar a hipótese nula indicaria, com certo grau de confiança (definido em  $\alpha$ ), que pelo menos uma entre as médias dos resultados analisados possuem diferenças estatísticas significativas, havendo sim um impacto entre os valores usados por pelo menos um dos parâmetros sobre a qualidade da solução final encontrada pelo algoritmo.

Não rejeitar a hipótese nula, por outro lado, afirmaria que os valores usados pelos parâmetros não impactaram no resultado do algoritmo, isto é, o algoritmo não é sensível aos valores usados pelos parâmetros e é capaz de encontrar resultados similares independente desse fator: se houver alguma diferença entre as médias dos tratamentos, pode-se afirmar com uma probabilidade<sup>6</sup> definida em  $P = (1 - \alpha)$  de que essa variação nos resultados ocorreu devido aos processos aleatórios inerentes ao algoritmo e não foram ocasionados pela mudança nos valores usados pelos parâmetros.

O teste *ANOVA* necessita que algumas pressuposições sejam cumpridas para garantir que seu resultado seja confiável. Devido a isso, após realizar o teste *ANOVA* é necessário realizar outros testes estatísticos em cima de seus resíduos para garantir que essas pressuposições estão de acordo com o esperado (GELMAN e HILL, 2007):

- Pressuposto I: independências dos erros.
- Pressuposto II: igualdade de variância (homocedasticidade).
- Pressuposto III: distribuição normal.

Existem duas formas de se verificar se as três pressuposições acima são válidas: executar o teste *ANOVA* e em seguida:

- I. Exibir em gráficos seus resíduos e fazer uma análise visual; ou

---

<sup>6</sup> Esse  $\alpha$  citado em P significa o grau de confiança definido para o teste ANOVA, e não o parâmetro “taxa de resfriamento”, também chamado de  $\alpha$ , usado pelo algoritmo e que está sendo testado. Em todas as seções contendo análise estatística o parâmetro  $\alpha$  do algoritmo será chamado de “alpha” para evitar confusões.

- II. Realizar outros testes estatísticos com base nos resíduos e analisar seus resultados.

Por mais que o uso de recursos visuais como gráficos ajudem na maior parte das vezes a compreender mais facilmente uma informação, nesse caso, analisar visualmente os gráficos necessários levaria mais tempo para os leitores deste trabalho que não estiverem familiarizados com esse processo, desconhecendo assim quais os padrões que indicam os problemas que podem ser visualizados. Portanto, a opção de realizar outros testes estatísticos foi escolhida e esses testes serão definidos a seguir.

### 5.2.3.3. Durbin-Watson

A primeira pressuposição que deve ser cumprida para usar o teste *ANOVA* pode ser validada usando o teste estatístico de *Durbin-Watson*, onde seu resultado permite descobrir se os resíduos de uma análise de variância estão autocorrelacionados, se estiverem, a pressuposição de independência dos erros não é satisfeita e o teste *ANOVA* não pode ser realizado devido aos dados não estarem de acordo com uma de suas pressuposições.

Após aplicar o teste de *Durbin-Watson* deve-se verificar quais os limites superiores ( $dU$ ) e inferiores ( $dL$ ) serão usados em sua análise após a aplicação do teste e da obtenção do valor  $d$  retornado por ele: se  $d < dL$ , rejeita-se a hipótese nula; caso contrário, se  $d > dU$ , não se rejeita a hipótese nula; por fim, se  $dL < d < dU$ , então os resultados são inconclusivos. Para os testes aqui propostos, devem ser usados os valores de  $dL = 1,08$  e  $dU = 1,36$ , números esses extraídos com base no tipo do teste usado, com 15 repetições e nível de confiança definido em  $\alpha = 0,05$ .

Se os erros não estiverem correlacionados, isto é, a hipótese nula não for rejeitada, significa que os resultados que não condizem com as previsões do teste *ANOVA* estão pelo menos de acordo com os erros que eram esperados de serem obtidos, o que é o suficiente para garantir o pressuposto de independência dos erros exigido pela análise de variância.

### 5.2.3.4. Levene

O teste de *Levene* é capaz de garantir a segunda pressuposição sobre os resíduos da *ANOVA*, verificando se os resíduos retornados pela mesma encontram-se em homocedasticidade, isto é, os resultados passados pela análise de variância encontram-se menos dispersos e mais aproximados da reta de regressão (LEVENE, 1960).

Quando o teste é efetuado e um valor de significância (*P-Value*) maior que  $\alpha$  (definido em 0,05) pode ser assumido que os resultados possuem uma variância homogênea, satisfazendo assim um dos critérios para a realização da *ANOVA*, caso

contrário ( $P - Value < 0,05$ ), o teste indicada que os resultados avaliados possuem diferenças de valores que não os permitem ser considerados como uma amostra satisfatória da população e não podem ser usados para análise por testes paramétricos de análise de variância.

### 5.2.3.5. Shapiro-Wilk

Por fim, o teste de *Shapiro – Wilk* (SHAPIRO e WILK, 1965) pode ser usado para garantir a terceira pressuposição sobre os resíduos da ANOVA, sendo capaz de verificar se os resíduos pertencem a uma distribuição normal ou não. Esse teste faz uso do princípio da Hipótese Nula ( $H_0$ ) para verificar se uma amostra foi retirada de uma distribuição normal. Se o valor de significância ( $P - Value$ ) do teste for menor do que  $\alpha$  (definido em 0,05), então, significa que  $H_0$  é rejeitada e que há evidências de que a amostra não segue uma distribuição normal.

### 5.2.3.6. Scott-Knott

O teste paramétrico de *Scott – Knott* (SCOTT e KNOTT, 1974) é um dos testes *post hoc* capazes de usar uma análise de variância para distinguir as médias encontradas para cada dupla de tratamentos. Esse teste fornece uma análise de resultados através da classificação dos resultados por grupos ordenados do melhor para o pior, sendo fácil de ser interpretado por não haver ambiguidade de resultados: cada tratamento é alocado a um único grupo, contrário, por exemplo, ao teste de *Tukey* (TUKEY, 1953; PURCHASE, 2012), que é um teste *post hoc* onde um único tratamento pode ser categorizado em mais de um grupo, contendo tratamentos diferentes em cada um deles, mas com a conclusão de que dentro de cada um desses grupos não há diferença entre as médias obtidas por seus membros.

Através das médias, desvios padrões e classificações dos tratamento por grupos é possível inferir, com certo grau de confiança, quais valores usados pelos parâmetros conseguiram fazer com que o algoritmo LaCSA alcançasse seu melhor desempenho médio, havendo como diferir se a média de um tratamento é melhor do que outra devido a mudança nos valores dos parâmetros ou se essa diferença é apenas consequência dos processos aleatórios presentes no algoritmo.

### 5.2.3.7. Kruskal-Wallis e o Método de Fischer LSD

Quando um teste paramétrico como o ANOVA não pode ser usado por violar uma de suas pressuposições, a opção mais segura é realizar um teste não paramétrico como o de *Kruskal – Wallis* (KW) e usar sua análise em algum outro teste *post hoc* também não

paramétrico. Um dos testes *post hoc* baseados na análise de KW é o teste de Fischer para Menor Diferença Significante (do inglês: *Fischer Least Significant Difference*, ou Fischer LSD) por ser capaz de realizar uma análise por pares de tratamentos, assim como sua versão paramétrica *Scott – Knott*. Esses dois testes serão realizados para todas as análises estatísticas a seguir, sendo sua aplicação melhor entendida nas próximas seções.

#### 5.2.4. Análise dos Resultados

Através dos dados coletados pelos experimentos realizados (seção 5.2.2) é possível obter, com certo grau de confiança, a informação de qual o conjunto de valores para os parâmetros possibilita o algoritmo a obter sua melhor performance. Para isso, os testes paramétricos ou não paramétricos relatados anteriormente podem ser usados. De forma a facilitar a leitura dos resultados, os parâmetros “temperatura” e “alpha” foram substituídos, respectivamente, pelos rótulos T e A. Seus valores foram alterados para serem usados como numeração do parâmetro, de forma a ficarem como o mostrado no Quadro 1.

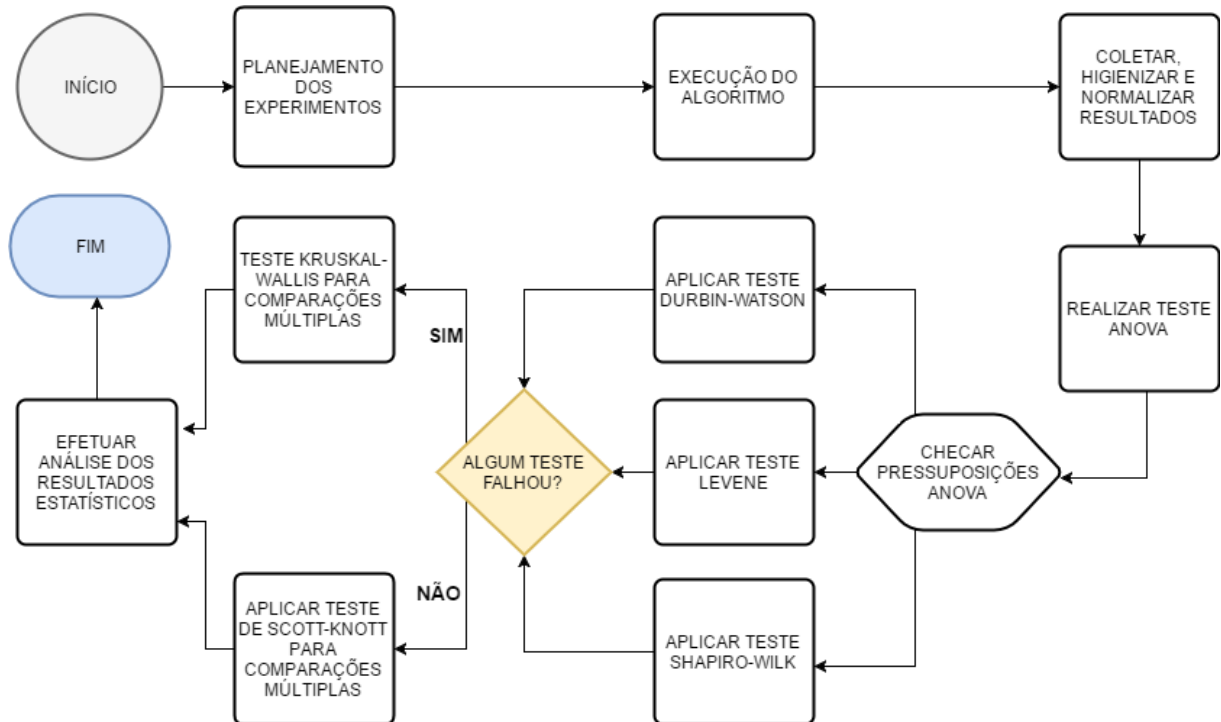
**Quadro 1** – Nomenclatura para os valores dos parâmetros Temperatura e Alpha.

<b>Temperatura</b>	<b>Valor</b>
T1	0,8%
T2	1,1%
T3	1,4%
T4	1,7%
<b>Alpha</b>	<b>Valor</b>
A1	40% p/ It.
A2	30% p/ It.
A3	20% p/ It.
A4	10% p It.

**Fonte:** Elaborada pelo autor.

A Figura 3 contém um fluxograma mostrando todos os processos, em ordem, que devem ser realizados para uma correta aplicação de testes estatísticos sob o conjunto de resultados coletados.

**Figura 3** - Fluxograma de todo o processo para realização da calibração dos parâmetros do algoritmo LaCSA por meio de testes estatísticos



Fonte: Elaborada pelo autor.

### 5.2.4.1. Análise Estatística Considerando Todas as Instâncias

Inicialmente o teste estatístico ANOVA deve ser usado para descobrir se há alguma diferença entre as médias obtidas pelos tratamentos, isto é, se algum valor para os parâmetros obteve resultados estatisticamente diferente em relação aos outros valores experimentados. Após a realização de tal teste, os resultados mostrados na Tabela 5 foram obtidos.

**Tabela 5** – Resultados do teste ANOVA sobre o impacto dos valores dos parâmetros no algoritmo quando aplicado nos resultados de todas as instâncias

	Grau de Liberdade	Soma dos Quadrados	Média dos Quadrados	F-Value	P-Value
ANOVA	15	11,8	0,785	11,85	2.00E-16

Fonte: Elaborada pelo autor

Para descobrir se a hipótese nula da ANOVA foi rejeitada, isto é, se há alguma diferença nas médias dos tratamentos devido mudança nos valores dos parâmetros, deve-se analisar o *P – Value* do resultado obtido pelo teste, sendo essa rejeição verdadeira se o *P – Value* for menor do que o nível de confiança (definido em 0,05), a conclusão é de que há diferença estatisticamente significativa entre as médias dos tratamentos e pelo menos um dos conjuntos de valores usados pelos parâmetros do algoritmo o fez obter resultados superiores do que a execução do mesmo algoritmo para pelo menos um conjunto de parâmetros diferentes. Entretanto, o desempenho de cada um dos

parâmetros usados não pode ser medido somente através do teste *ANOVA*, sendo necessário usar outro teste baseado na análise de variância, como o de *Tukey* ou o de *Scott – Knott*, para alcançar esse objetivo. Uma observação sobre o valor do *P – Value* encontrado: como relatado anteriormente, a plataforma R para testes estatísticos foi usada para programar e executar os testes cujos resultados são aqui mostrados, nessa plataforma o número  $2.00E - 16$  é uma constante usada como *P – Value* para indicar com um nível máximo de confiança possível sobre a existência de uma diferença significativa entre os tratamentos, portanto, será comum vê-lo nos próximos testes que serão realizados.

Antes de dar sequência à análise com os testes *post hoc*, deve-se primeiro validar as três pressuposições da *ANOVA*. Para isso, pode-se começar usando o teste *Durbin – Watson* para descobrir se os resíduos da análise de variância estão autocorrelacionados. Caso os resíduos se mostrem autocorrelacionados, os dados não estarão aptos a serem usados pelo teste *ANOVA* por não cumprir com um de seus pressupostos, portanto, seu resultado apresentado acima na Tabela 5 deve ser ignorado.

Após a realização do teste de *Durbin – Watson* obteve-se um resultado  $d = 1,5267$ . Para os testes aqui propostos, devem ser usados, respectivamente, como limites inferior e superior, os valores de  $dL = 1,08$  e  $dU = 1,36$ ; sendo esses números extraídos com base no tipo do teste usado, com 15 repetições e nível de confiança definido em  $\alpha = 0,05$ . Os critérios para análise do valor  $d$  do teste de *Durbin – Watson* são:

- Se  $d < dL$ , então não rejeita-se a hipótese de autocorrelação.
- Se  $d > dU$ , então rejeita-se a hipótese de autocorrelação.
- Se  $dL < d < dU$ , então os resultados são inconclusivos.

Se os erros estiverem correlacionados, isto é, a hipótese nula não for rejeitada ( $d < dL$ ), significa que o pressuposto de independência dos erros exigido pela análise de variância não é satisfeito; entretanto, se a hipótese de autocorrelação for rejeitada ( $d > dU$ ), seu pressuposto é garantido e os resíduos do teste *ANOVA* devem ser submetidos aos outros testes para validar seus dois pressupostos que ainda estão em aberto. Como o valor obtido por  $d$  foi relatado anteriormente como sendo  $1,5267$ , seu resultado indica que  $d > dU$ , portanto, acredita-se que há independência de erros dos resíduos e a análise prosseguirá para seu próximo teste.

O teste de *Levene* é capaz de garantir a segunda pressuposição sobre os resíduos da análise de variância realizada pelo teste *ANOVA*, verificando se os resíduos retornados pela mesma encontram-se em homocedasticidade. Após sua realização, o resultado fornecido pelo teste mostrado pela Tabela 6 possui um *P – Value* =  $0,3665$ , sendo este maior que o valor  $0,05$  definido como nível de significância, portanto, acredita-se que os resíduos possuem uma variância homogênea e o segundo pressuposto exigido pelo teste *ANOVA* é satisfeito.

**Tabela 6** – Resultados do teste Levene para validação da Pressuposição II da ANOVA com todas as instâncias

	<b>Grau de Liberdade</b>	<b>F-Value</b>	<b>P-Value</b>
<b>Levene</b>	15	1,0827	0,3665

**Fonte:** Elaborada pelo autor

Para validar o terceiro pressuposto da ANOVA pode ser aplicado o teste *Shapiro – Wilk* que indica através de seu resultado se há uma distribuição normal de resíduos. Entretanto, após a aplicação desse teste, como mostrado na Tabela 7, obteve-se um  $P - Value = 2,2E - 16$ , indicando que hipótese nula não pode ser rejeitada ( $2,2E - 16 < 0,05$ ), portanto, as amostras não seguem uma distribuição normal e o último pressuposto da ANOVA não foi satisfeito.

É provável que os resultados não seguiram um padrão de distribuição normal porque as soluções existentes no espaço de busca do PMV não possuem valores distribuídos seguindo essa distribuição, assim como ocorre com outros problemas de otimização combinatória (CHIARANDINI et al, 2004).

**Tabela 7** – Resultado do teste Shapiro-Wilk para validação da Pressuposição III da ANOVA com todas as instâncias

	<b>W</b>	<b>P-Value</b>
<b>Shapiro-Wilk</b>	0,9598	2,2E-16

**Fonte:** Elaborada pelo autor.

Com seu pressuposto de distribuição normal de resíduos não sendo satisfeito sobram dois caminhos a serem seguidos: realizar testes estatísticos não paramétricos que não necessitam que esse pressuposto seja atendido (FEIR-WALSH e TOOTHAKER, 1974; SHESKIN, 2003) ou ignorar o pressuposto de normalidade da ANOVA e continuar com o teste paramétrico usando outros testes *post hoc* como planejado. A primeira opção é comum de ser encontrada na literatura como solução viável para esse problema, porém, testes não paramétricos possuem poder estatístico menor do que os paramétricos e devem ser evitados em prol deste sempre que possível (SHESKIN, 2003; CHIARANDINI et al, 2004); porém, a segunda opção é menos comum de ser encontrada, mas há indícios de que o pressuposto de normalidade, se não satisfeito, é o que menos afeta o teste ANOVA e pode não impedir que a mesma entregue resultados mais precisos do que suas alternativas não paramétricas (GELMAN e HILL, 2016; CHIARANDINI et al, 2007). Para dar sequência nesse processo, foi decidido executar os dois caminhos e analisar seus resultados para ver se ambos estão condizentes, fornecendo assim uma maior confiança sobre a conclusão final obtida, portanto, o teste de *Scott – Knott* será executado para continuação da análise paramétrica e o teste de *Kruskal – Wallis* para comparações

múltiplas será usado como teste não paramétrico. Se ocorrer algum conflito entre os resultados dos diferentes testes realizados, o resultado fornecido pelo teste de *Kruskal – Wallis* será usado como verdadeiro por ser essa a abordagem mais aconselhável quando ocorre a inviabilização dos testes paramétricos (GELMAN e HILL, 2016; CHIARANDINI et al, 2007; SHESKIN, 2003).

Para finalizar a aplicação dos testes paramétricos o resultado da ANOVA será aplicado no teste de *Scott – Knott* de forma a permitir descobrir quais duplas de tratamentos diferem entre si. Os resultados fornecidos pelo teste de *Scott – Knott* podem ser analisados através da classificação dos tratamentos por grupos, sendo mostrado na Tabela 8 por letras organizadas em ordem alfabética em relação aos tratamentos que obtiveram os melhores resultados.

**Tabela 8** – Resultados do teste Scott-Knott usando a análise de variância com todas as instâncias

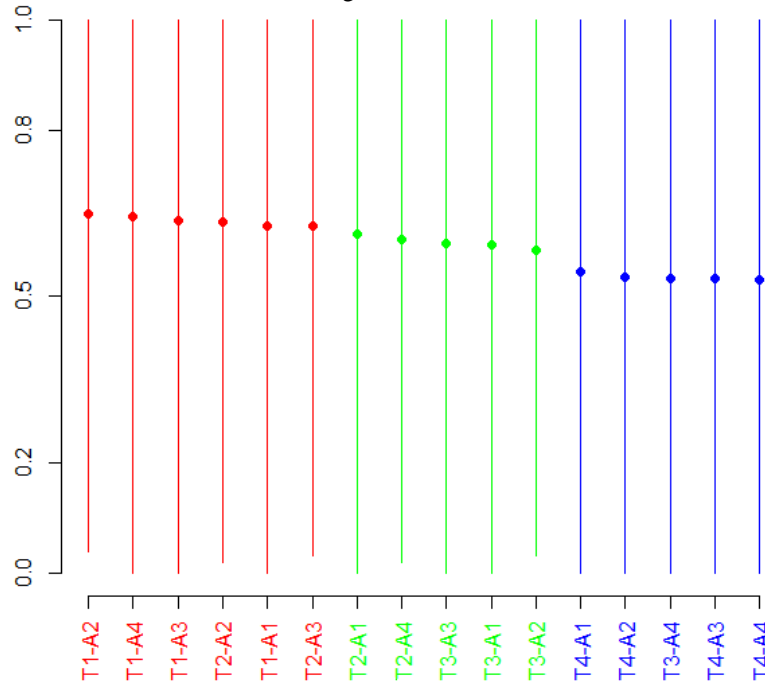
PARÂMETRO	MÉDIA	CLASSIFICAÇÃO
T1-A2	0.650	a
T1-A4	0.644	a
T1-A3	0.637	a
T2-A2	0.635	a
T1-A1	0.628	a
T2-A3	0.627	a
T2-A1	0.614	b
T2-A4	0.603	b
T3-A3	0.595	b
T3-A1	0.594	b
T3-A2	0.583	b
T4-A1	0.544	c
T4-A2	0.534	c
T3-A4	0.533	c
T4-A3	0.533	c
T4-A4	0.531	c

**Fonte:** Elaborada pelo autor

Como pode ser visto na Tabela 8, o tratamento com os parâmetros T1-A2 (Quadro 1) obteve a melhor média de resultados quando usado para solucionar todas as instâncias usadas pelo experimento de calibração, porém, não pode ser afirmado que ele consegue obter resultados melhores do que todos os outros, pois o teste de *Scott – Knott* o definiu como participante do grupo “a”, indicando que o mesmo não consegue obter resultados estatisticamente diferentes em relação aos outros 5 tratamentos também classificados como participantes desse grupo, portanto, deve-se considerar todos os grupos com a letra “a” como estatisticamente capazes de convergirem para a mesma média se um número muito alto de repetições forem realizadas. Na Figura 4 esses resultados são apresentados em formato de gráfico mostrando as médias obtidas por cada tratamento e usando cores iguais para todas as médias que não se diferem de forma

significativa, facilitando assim a visualização da classificação feita pelo teste de *Scott – Knot* desses tratamentos.

**Figura 4** – Gráfico do resultado do teste Scott-Knott aplicado nos tratamentos de todas as instâncias: tratamentos considerados iguais usam a mesma cor



**Fonte:** Elaborada pelo autor.

A escolha do valor de 0,8% para a temperatura (T1) e de 30% para a taxa de queda da temperatura por iteração ( $\alpha$ ) ocorreu por meio de uma análise estatística usando testes paramétricos, porém, como discutido anteriormente, a ANOVA necessita que seus três pressupostos sejam atendidos para que seu resultado possa ser validado. Como o teste *Shapiro – Wilk* para verificar a distribuição normal dos resíduos não mostrou resultados que indicam uma falha em uma das pressuposições da ANOVA, optamos por realizar o teste de *Scott – Knott* com a intenção de realizar, em sequência, o teste de *Kruskal – Wallis* por ser um teste não paramétrico e não necessitar que os resultados sigam uma distribuição normal para serem analisados. Na Tabela 9 pode ser visto o resultado do teste de *Kruskal – Wallis* em cima do mesmo conjunto de dados.

**Tabela 9** – Resultado do teste Kruskal-Wallis com o teste Fischer LSD para comparações múltiplas analisando todas as instâncias

PARÂMETROS	GRUPO	KW	MÉDIA	DESVIO PADRÃO	MIN	MAX
T1-A2	a	3589.367	0.650	0.256	0.038	1.000
T1-A4	ab	3537.270	0.644	0.246	0.000	1.000
T1-A3	ab	3507.708	0.637	0.257	0.000	1.000
T2-A2	abc	3483.362	0.635	0.249	0.020	1.000
T1-A1	abc	3454.492	0.628	0.262	0.000	1.000
T2-A3	abcd	3416.039	0.627	0.249	0.031	1.000
T2-A1	bcde	3334.493	0.614	0.260	0.000	1.000
T2-A4	cde	3244.160	0.603	0.256	0.020	1.000
T3-A3	de	3181.298	0.595	0.252	0.000	1.000
T3-A1	de	3169.082	0.594	0.254	0.000	1.000
T3-A2	e	3104.596	0.583	0.253	0.032	1.000
T4-A1	f	2834.130	0.544	0.261	0.000	1.000
T4-A2	f	2775.475	0.534	0.271	0.000	1.000
T4-A4	f	2754.056	0.531	0.272	0.000	1.000
T3-A4	f	2749.819	0.533	0.258	0.000	1.000
T4-A3	f	2745.982	0.533	0.261	0.000	1.000

Fonte: Elaborada pelo autor

Como observado acima, o conjunto de parâmetros T1(0,8%)-A2(30%) também foi o que obteve a melhor classificação seguindo o valor estatístico (terceira coluna, KW) do teste *Kruskal – Wallis*, indicando que o teste de *Scott – Knott*, mesmo sendo realizado sobre o teste ANOVA contendo dados que não seguem uma distribuição normal, conseguiu relatar resultados suficientemente precisos. Há pequenas diferenças entre os resultados do teste paramétrico (SK) e não paramétrico (KW), sendo essas diferenças presentes apenas nos três últimos (piores) tratamentos, não influenciando assim na escolha dos valores para os parâmetros. Esse resultado condiz com o que é discutido em outros trabalhos da literatura, onde há argumentos de que o teste ANOVA pode ser resiliente a sua pressuposição de normalidade em certos cenários (GELMAN e HILL, 2016; CHIARANDINI et al, 2007; SHESKIN, 2003), entretanto, a aplicação de testes não paramétricos ainda assim é aconselhável por ainda não haver uma forma de descobrir quais são esses cenários em que essa pressuposição não necessita ser cumprida.

A escolha do valor de 0,8% para a temperatura (T1) e de 30% para a taxa de queda da temperatura por iteração (A2) ocorreu porque esses valores conseguiram obter a melhor média para todas as instâncias, entretanto, como foi utilizada a normalização dos resultados descrita na seção 5.2.3.1, não se pode concluir que essa seria a melhor configuração de parâmetros para elevar o lucro total: o uso da normalização dos resultados fez com que as instâncias com soluções possuindo valores acima de 100.000,00 obtivessem a mesma influência em decidir quais parâmetros usar em relação às instâncias com soluções possuindo valores inferiores a 20.000,00. Se aplicado em um cenário onde o maior objetivo é a maximização do capital (\$), é possível que o algoritmo

consiga encontrar soluções melhores quando usado com parâmetros diferentes para essas instâncias com maior valor.

Para descobrir tal informação, poder-se-ia inicialmente pensar em realizar o teste de *Scott – Knott ou Kruskal – Wallis* usando os valores reais, não normalizados, para cada tratamento de cada grupo de instância em interesse. Porém, esses valores reais não poderiam ser usados sem a normalização por esses testes estatísticos para avaliar o desempenho do algoritmo em todas as instâncias: os dados variam muito de valor entre os diversos tipos de instâncias e a análise de variância não conseguiria explicar essa mudança através dos parâmetros do modelo: ora um parâmetro funciona muito bem, obtendo valores altos; em outros momentos, com os mesmos parâmetros, são obtidos valores significativamente inferiores. Esse problema ocorre porque a instância não é incluída no modelo estatístico como um fator, uma vez que a mesma está mais para uma variável interveniente<sup>7</sup> e seria impossível de ser controlada, sendo que ela é fornecida como dados de entrada para ser solucionada pelo algoritmo, e não usada por ele para aumentar seu desempenho durante a busca: sem ela não há busca.

#### 5.2.4.2. Análise Estatística: Instâncias com 76 Cidades

Uma das conclusões tiradas da discussão apresentada anteriormente é a que os valores ideais para os parâmetros podem variar em relação aos tipos de instâncias que estiverem sendo solucionadas. Para explorar a forma como o algoritmo se comporta para os diversos tipos de instância, será feita também uma análise estatística com base nos resultados de grupos específicos de instâncias, sendo essa separação feita em 4 grupos: 76 cidades, 195 cidades, 3 itens por cidade ou 5 itens por cidade. Inicia-se nesta seção a análise para os resultados das instâncias com 76 cidades.

O teste *ANOVA* é realizado e seu resultado é mostrado na Tabela 10. Para testar suas pressuposições, esse novo conjunto de dados necessita passar também pelos três testes (*Durbin – Watson, Levene e Shapiro – Wilk*) capazes de validar cada uma delas. Através da aplicação do teste *Durbin – Watson* obteve-se um valor  $d = 1,668$ , sendo esse valor maior do que o valor de  $dU (1,36)$  encontrado anteriormente, portanto, os dados analisados possuem independência de erros (Pressuposto I) dos resíduos e a análise prosseguirá para seu próximo teste. Para testar a igualdade de variância o teste de *Levene* foi aplicado e seu resultado é visto na Tabela 11. Por obter um  $P - Value > 0,05$ , é possível concluir que esse novo (sub)conjunto de dados está em homocedasticidade, validando assim o Pressuposto II da ANOVA e avançando para o próximo teste. Ao realizar o teste de *Shapiro – Wilk*, novamente, os tratamentos

---

<sup>7</sup> Uma variável interveniente se localiza entre a variável independente (parâmetros) e a variável dependente (resultado): é capaz de alterar a variável dependente sem haver modificação na variável independente, i.e., afeta o resultado do algoritmo, no entanto, não pode ser controlada ou medida.

fornecidos para a ANOVA foram considerados não pertencentes a uma distribuição normal (Tabela 12), portanto, o Pressuposto III é novamente rejeitado. Devido a isso, o mesmo processo contendo testes paramétricos e não paramétricos usados no conjunto completo de todas as instâncias será também necessário para essa análise.

**Tabela 10** – Resultados do teste ANOVA sobre o impacto dos valores dos parâmetros no algoritmo quando aplicado somente nos resultados das instâncias com 76 cidades

	<b>Grau de Liberdade</b>	<b>Soma dos Quadrados</b>	<b>Média dos Quadrados</b>	<b>F-Value</b>	<b>P-Value</b>
<b>ANOVA</b>	15	1,74	0,115	1,623	0,0602

**Fonte:** Elaborada pelo autor

**Tabela 11** - Resultados do teste Levene para validação da Pressuposição II da ANOVA com as instâncias de 76 cidades

	<b>Grau de Liberdade</b>	<b>F-Value</b>	<b>P-Value</b>
<b>Levene</b>	15	1,0827	0,3665

**Fonte:** Elaborada pelo autor

**Tabela 12** - Resultado do teste Shapiro-Wilk para validação da Pressuposição III da ANOVA com as instâncias de 76 cidades

	<b>W</b>	<b>P-Value</b>
<b>Shapiro-Wilk</b>	0,9598	2,2E-16

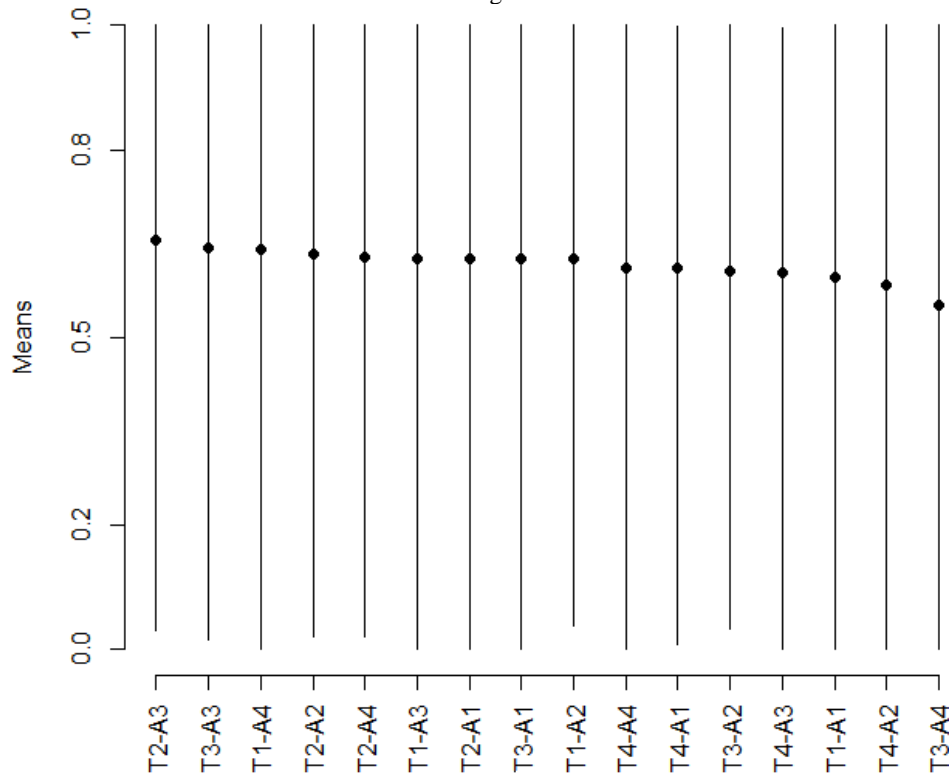
**Fonte:** Elaborada pelo autor

Como verificado na seção anterior, o teste ANOVA tem se mostrado resiliente aos tratamentos não pertencerem à distribuição normal, portanto, o teste paramétrico de *Scott – Knott* para comparações em duplas de tratamentos será aplicado para descobrir se há algum tratamento que conseguiu ser estatisticamente melhor do que algum outro, mas, antes disso, deve-se dar atenção ao fato de que o teste ANOVA (Tabela 10) obteve um  $P - Value = 0,0602$ , indicando não haver diferença significativa entre as médias dos tratamentos analisados. Na Tabela 13 o resultado do teste *Scott – Knott* é mostrado, validando assim o resultado entregue pela ANOVA. Na Figura 5 são mostradas as médias dos tratamentos em uma única cor representando a igualdade em obter resultados estatisticamente iguais por todos os tratamentos para as instâncias com 76 cidades.

**Tabela 13** - Resultados do teste Scott-Knott usando a análise de variância com as instâncias de 76 cidades

PARÂMETRO	MÉDIA	CLASSIFICAÇÃO
T2-A3	0.6570	a
T3-A3	0.6439	a
T1-A4	0.6410	a
T2-A2	0.6342	a
T2-A4	0.6276	a
T1-A3	0.6267	a
T2-A1	0.6260	a
T3-A1	0.6257	a
T1-A2	0.6255	a
T4-A4	0.6124	a
T4-A1	0.6112	a
T3-A2	0.6058	a
T4-A3	0.6050	a
T1-A1	0.5967	a
T4-A2	0.5844	a
T3-A4	0.5520	a

Fonte: elaborada pelo autor

**Figura 5** - Gráfico do resultado do teste Scott-Knott aplicado nos tratamentos das instâncias com 76 cidades: tratamentos considerados iguais usam a mesma cor

Fonte: elaborada pelo autor

Apesar de haver diferenças entre os tratamentos no que diz respeito a ordem em que eles obtiveram a melhor média, não foi possível acusar uma diferença grande o suficiente para considerar uma relação de causa-efeito entre os valores usados pelos

parâmetros e essa variação entre as médias de cada tratamento, sendo assim, qualquer um entre os valores pode ser escolhido e a diferença nos resultados encontrados existe devido aos processos aleatórios os quais o algoritmo está sujeito. Essa é a conclusão tirada com base nos testes paramétricos mostrados até agora para o conjunto de instâncias com 76 cidades, entretanto, após a realização do teste não paramétrico *Kruskal – Wallis*, o mesmo indicou com seus resultados mostrados na Tabela 14 que há valores que devem ser evitados por conseguirem resultados piores do que os outros, contradizendo assim os resultados obtidos pelo SK.

**Tabela 14** - Resultado do teste Kruskal-Wallis com o teste Fischer LSD para comparações múltiplas analisando todas as instâncias

PARÂMETROS	GRUPO	KW	MÉDIA	DESVIO PADRÃO	MIN	MAX
T2-A3	a	1605.982	0.657	0.263	0.031	1.000
T3-A3	ab	1554.414	0.644	0.254	0.015	1.000
T1-A4	ab	1551.768	0.641	0.266	0.000	1.000
T2-A2	ab	1541.332	0.634	0.271	0.020	1.000
T2-A4	ab	1512.787	0.628	0.273	0.020	1.000
T1-A3	ab	1525.227	0.627	0.281	0.000	1.000
T2-A1	ab	1504.036	0.626	0.273	0.001	1.000
T3-A1	ab	1493.874	0.626	0.258	0.000	1.000
T1-A2	ab	1504.490	0.626	0.276	0.038	1.000
T4-A4	ab	1449.853	0.612	0.253	0.000	1.000
T4-A1	abc	1441.726	0.611	0.257	0.008	0.998
T3-A2	abc	1431.699	0.606	0.260	0.032	1.000
T4-A3	bc	1431.046	0.605	0.262	0.000	0.996
T1-A1	bc	1419.976	0.597	0.279	0.000	1.000
T4-A2	bc	1381.011	0.584	0.280	0.000	1.000
T3-A4	c	1269.688	0.552	0.267	0.000	1.000

**Fonte:** elaborada pelo autor

Com base nos resultados dessa análise, pode-se constatar que a configuração T1-A2, quando usada pelo algoritmo LaCSA, consegue obter os melhores resultados quando analisadas todas as instâncias usadas pelo experimento e também quando apenas as instâncias com 76 cidades são selecionadas. Se a configuração T1-A2 estivesse classificada como as configurações T4-A3, T1-A1, T4-A2 e T3-A4, sua troca poderia ser realizada para executar esse grupo de instâncias em prol de uma melhoria no desempenho do algoritmo, podendo essa configuração ser substituída por qualquer uma das configurações de parâmetros que estiverem com seus tratamentos classificadas como grupo “a” para os testes com todas as cidades e também para o teste com 76 cidades, sendo elas: T1-A2, T1-A3, T1-A4, T2-A2 ou T2-A3. As demais configurações se mostraram piores quando usadas para solucionar as instâncias com todas as cidades ou quando usadas para solucionar apenas as instâncias com 76 cidades, entretanto, os testes estatísticos usando as instâncias com 195 cidades, 3 itens por cidades e 5 itens por

idades serão mostrados a seguir e essa lista de parâmetros “aconselháveis” pode vir a ser alterada.

### 5.2.4.3. Análise Estatística: Instâncias com 195 Cidades

Usando agora como fonte de dados as instâncias contendo 195 cidades, o teste *ANOVA* tem seu resultado mostrado na Tabela 15. Para testar suas pressuposições, esse novo conjunto de dados necessita passar também pelos três testes (*Durbin – Watson, Levene e Shapiro – Wilk*) capazes de validar cada uma delas. Através da aplicação do teste *Durbin – Watson* obteve-se um valor  $d = 1,496$ , sendo esse valor maior do que o valor de  $dU (1,36)$  encontrado anteriormente, portanto, os dados analisados possuem independência de erros (Pressuposto I) dos resíduos e a análise prosseguirá para seu próximo teste. Para testar a igualdade de variância o teste de *Levene* foi aplicado e seu resultado é visto na Tabela 16. Por obter um  $P - Value > 0,05$ , é possível concluir que esse novo conjunto de dados, restrito às instâncias com 195 cidades, está em homocedasticidade, validando assim o Pressuposto II da ANOVA e avançando para o próximo teste. Após a realização do teste *Shapiro – Wilk* os tratamentos fornecidos para a ANOVA foram considerados não pertencentes a uma distribuição normal (Tabela 17), portanto, o Pressuposto III é novamente rejeitado. Devido a isso, o mesmo processo contendo testes paramétricos e não paramétricos usados anteriormente será necessário também para essa análise.

**Tabela 15** - Resultados do teste ANOVA sobre o impacto dos valores dos parâmetros no algoritmo quando aplicado nos resultados das instâncias com 195 cidades

	<b>Grau de Liberdade</b>	<b>Soma dos Quadrados</b>	<b>Média dos Quadrados</b>	<b>F-Value</b>	<b>P-Value</b>
<b>ANOVA</b>	15	15,27	1,0181	17,11	2.00E-16

Fonte: Elaborada pelo autor

**Tabela 16** - Resultados do teste Levene para validação da Pressuposição II da ANOVA com as instâncias de 195 cidades

	<b>Grau de Liberdade</b>	<b>F-Value</b>	<b>P-Value</b>
<b>Levene</b>	15	1,0317	0,4182

Fonte: Elaborada pelo autor

**Tabela 17** - Resultado do teste Shapiro-Wilk para validação da Pressuposição III da ANOVA com as instâncias de 195 cidades

	<b>W</b>	<b>P-Value</b>
<b>Shapiro-Wilk</b>	0,983	0,0003339

Fonte: Elaborada pelo autor

A Tabela 15 mostra que o teste ANOVA conseguiu obter um  $P - Value = 2,2E - 16$ , indicando haver diferenças significativas entre as médias dos tratamentos analisados. O teste ANOVA se mostrou resiliente aos tratamentos não pertencerem à distribuição

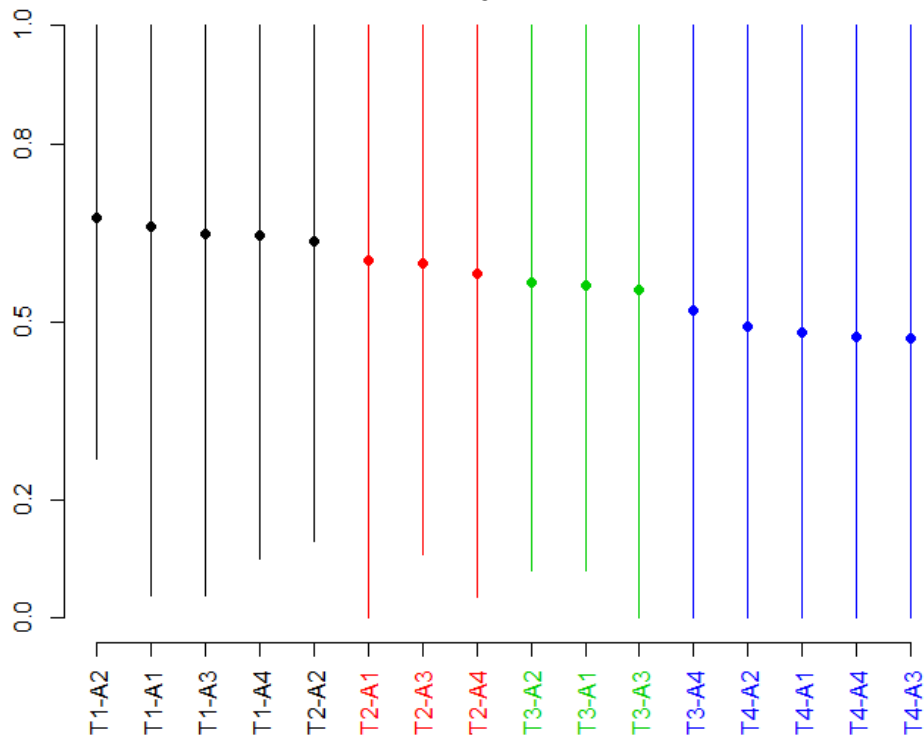
normal, portanto, o teste paramétrico de *Scott – Knott* para comparações em duplas de tratamentos será aplicado para descobrir se há algum tratamento que conseguiu ser estatisticamente melhor do que algum outro. Na Tabela 18 o resultado desse teste é mostrado, validando assim o resultado entregue pela ANOVA. Na Figura 6 são mostradas as médias dos tratamentos, classificados por cores, agrupando assim os resultados estatisticamente iguais e distinguindo visualmente os conjuntos de parâmetros que impactaram no desempenho do algoritmo.

**Tabela 18** - Resultados do teste Scott-Knott usando a análise de variância com as instâncias de 195 cidades

<b>PARÂMETRO</b>	<b>MÉDIA</b>	<b>CLASSIFICAÇÃO</b>
T1-A2	0.675	a
T1-A1	0.662	a
T1-A3	0.648	a
T1-A4	0.646	a
T2-A2	0.637	a
T2-A1	0.603	b
T2-A3	0.600	b
T2-A4	0.582	b
T3-A2	0.566	c
T3-A1	0.562	c
T3-A3	0.554	c
T3-A4	0.519	d
T4-A2	0.493	d
T4-A1	0.484	d
T4-A4	0.475	d
T4-A3	0.474	d

**Fonte:** Elaborada pelo autor

**Figura 6** - Gráfico do resultado do teste Scott-Knott aplicado nos tratamentos das instâncias com 195 cidades: tratamentos considerados iguais usam a mesma cor



Fonte: Elaborada pelo autor

Novamente o parâmetro T1-A2 é encontrado entre os tratamentos que conseguiram obter resultados estatisticamente superiores aos demais, portanto, seu uso também é aconselhado para solucionar esse conjunto de instâncias. Após a realização do teste não paramétrico *Kruskal – Wallis*, o mesmo indicou com seus resultados mostrados na Tabela 19 que, novamente, mesmo com sua Pressuposição III sendo violada, o teste ANOVA em conjunto ao de *Scott – Knott* conseguiu entregar resultados similares aos obtidos pelo teste não paramétrico, confirmando assim a vantagem em usar o parâmetro T1-A2.

**Tabela 19** - Resultado do teste Kruskal-Wallis com o teste Fischer LSD para comparações múltiplas analisando as instâncias de 195 cidades

PARÂMETRO	GRUPO	KW	MÉDIA	DESVIO PADRÃO	MIN	MAX
T1-A2	a	2117.508	0.675	0.232	0.269	1.000
T1-A1	a	2078.541	0.662	0.239	0.038	1.000
T1-A3	ab	2017.291	0.648	0.231	0.039	1.000
T1-A4	ab	2008.711	0.646	0.229	0.099	1.000
T2-A2	abc	1959.988	0.637	0.227	0.129	1.000
T2-A1	bcd	1847.186	0.603	0.248	0.000	1.000
T2-A3	cde	1820.532	0.600	0.233	0.109	1.000
T2-A4	def	1735.855	0.582	0.240	0.037	1.000
T3-A2	def	1678.856	0.566	0.246	0.081	1.000
T3-A1	efg	1655.279	0.562	0.246	0.080	1.000
T3-A3	fg	1623.580	0.554	0.243	0.000	1.000
T3-A4	gh	1479.332	0.519	0.250	0.000	1.000
T4-A2	h	1386.220	0.493	0.257	0.000	1.000
T4-A1	h	1362.297	0.484	0.249	0.000	1.000
T4-A4	h	1332.442	0.475	0.272	0.000	1.000
T4-A3	h	1307.120	0.474	0.246	0.000	1.000

Fonte: Elaborada pelo autor

#### 5.2.4.4. Análise Estatística: Instâncias com 3 Itens por Cidade

Usando agora como fonte de dados as instâncias contendo 3 itens por cidade, sem fazer distinção entre o número de cidades contidas nas mesmas, o teste *ANOVA* tem seu resultado mostrado na Tabela 20. Para testar suas pressuposições, esse novo conjunto de dados necessita passar também pelos três testes (*Durbin – Watson*, *Levene e Shapiro – Wilk*) capazes de validar cada uma delas. Através da aplicação do teste *Durbin – Watson* obteve-se um valor  $d = 1,433$ , sendo esse valor maior do que o valor de  $dU(1,36)$  encontrado anteriormente, portanto, os dados analisados possuem independência de erros (Pressuposto I) dos resíduos e a análise prosseguirá para seu próximo teste. Para testar a igualdade de variância o teste de *Levene* foi aplicado e seu resultado é visto na Tabela 21. Por obter um  $P - Value > 0,05$ , é possível concluir que esse novo conjunto de dados, restrito às instâncias com 3 itens por cidade, está em homocedasticidade, validando assim o Pressuposto II da ANOVA e avançando para o próximo teste. Após a realização do teste *Shapiro – Wilk* os tratamentos fornecidos para a ANOVA foram considerados não pertencentes a uma distribuição normal (Tabela 22), portanto, o Pressuposto III é novamente rejeitado. Devido a isso, o mesmo processo contendo testes paramétricos e não paramétricos usados anteriormente será necessário também para essa análise.

**Tabela 20** - Resultados do teste ANOVA sobre o impacto dos valores dos parâmetros no algoritmo quando aplicado nos resultados das instâncias com 3 itens por cidade

	Grau de Liberdade	Soma dos Quadrados	Média dos Quadrados	F-Value	P-Value
ANOVA	15	8,89	0,5927	10,07	2.00E-16

**Fonte:** Elaborada pelo autor

**Tabela 21** - Resultados do teste Levene para validação da Pressuposição II da ANOVA com as instâncias de 3 itens por cidade

	<b>Grau de Liberdade</b>	<b>F-Value</b>	<b>P-Value</b>
<b>Levene</b>	15	0,941	0,517

**Fonte:** Elaborada pelo autor

**Tabela 22** - Resultado do teste Shapiro-Wilk para validação da Pressuposição III da ANOVA com as instâncias de 3 itens por cidade

	<b>W</b>	<b>P-Value</b>
<b>Shapiro-Wilk</b>	0,921	0,0003339

**Fonte:** Elaborada pelo autor

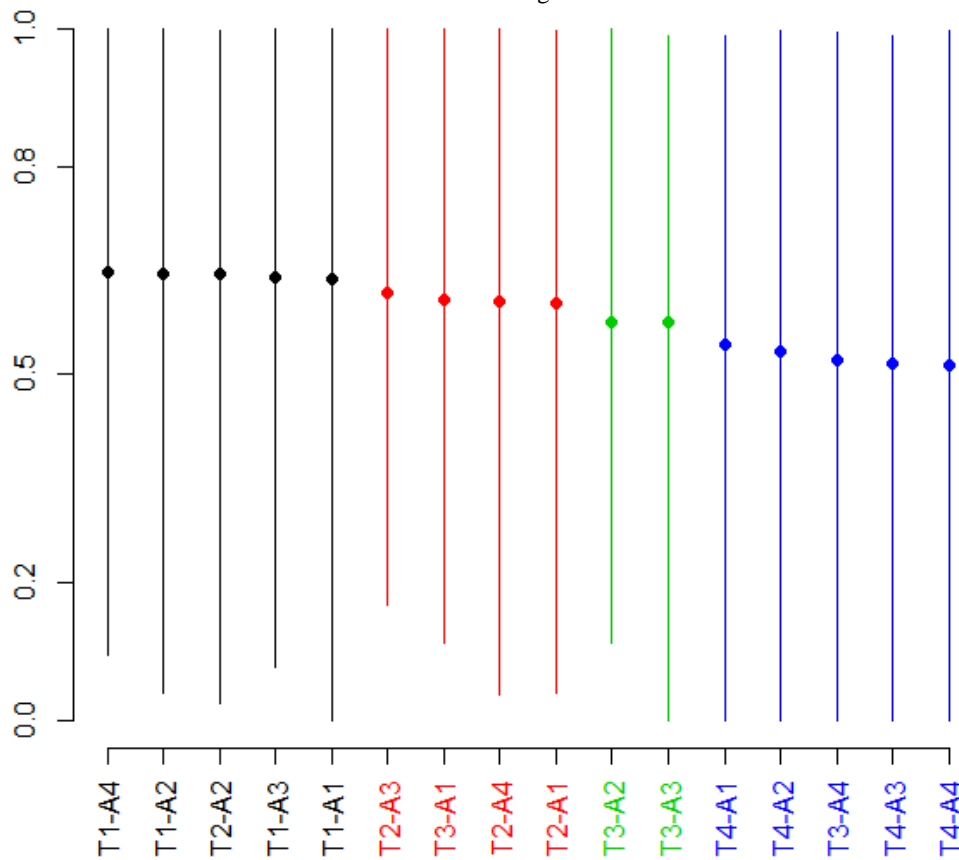
A Tabela 20 mostra que o teste ANOVA conseguiu obter um  $P - Value = 2,0E - 16$ , indicando haver diferenças significativas entre as médias dos tratamentos analisados. O teste ANOVA se mostrou resiliente aos tratamentos não pertencerem à distribuição normal, portanto, o teste paramétrico de *Scott – Knott* para comparações em duplas de tratamentos será aplicado para descobrir se há algum tratamento que conseguiu ser estatisticamente melhor do que algum outro. Na Tabela 23 o resultado desse teste é mostrado, validando assim o resultado entregue pela ANOVA. Na Figura 7 são mostradas as médias dos tratamentos, classificados por cores, agrupando assim os resultados estatisticamente iguais e distinguindo visualmente os conjuntos de parâmetros que impactaram no desempenho do algoritmo.

**Tabela 23** - Resultados do teste Scott-Knott usando a análise de variância com as instâncias de 3 itens por cidade

<b>PARÂMETRO</b>	<b>MÉDIA</b>	<b>CLASSIFICAÇÃO</b>
T1-A4	0.6481	a
T1-A2	0.6467	a
T2-A2	0.6454	a
T1-A3	0.6411	a
T1-A1	0.6393	a
T2-A3	0.6179	b
T3-A1	0.6088	b
T2-A4	0.6054	b
T2-A1	0.6036	b
T3-A2	0.5758	c
T3-A3	0.5751	c
T4-A1	0.5431	d
T4-A2	0.5346	d
T3-A4	0.5219	d
T4-A3	0.5158	d
T4-A4	0.5137	d

Fonte: Elaborada pelo autor

Figura 7 - Gráfico do resultado do teste Scott-Knott aplicado nos tratamentos das instâncias com 3 itens por cidade: tratamentos considerados iguais usam a mesma cor



Fonte: Elaborada pelo autor

Novamente o parâmetro T1-A2 é encontrado entre os tratamentos que conseguiram obter resultados estatisticamente superiores aos demais, portanto, seu uso também é aconselhado para solucionar esse conjunto de instâncias. Após a realização do teste não paramétrico *Kruskal – Wallis*, o mesmo indicou com seus resultados mostrados na Tabela 24 que, novamente, mesmo com sua Pressuposição III sendo violada, o teste ANOVA em conjunto ao de *Scott – Knott* conseguiu entregar resultados similares aos obtidos pelo teste não paramétrico, confirmando assim a vantagem em usar o parâmetro T1-A2 para o algoritmo LaCSA.

**Tabela 24** - Resultados do teste Kruskal-Wallis com o teste Fischer LSD para comparações múltiplas analisando as instâncias com 3 itens por cidade

PARÂMETRO	GRUPO	KW	MÉDIA	DESVIO PADRÃO	MIN	MAX
T1-A4	a	2145.598	0.648	0.235	0.095	1.000
T1-A2	a	2143.257	0.647	0.246	0.038	1.000
T2-A2	a	2138.188	0.645	0.236	0.024	0.997
T1-A1	ab	2119.626	0.639	0.245	0.000	1.000
T1-A3	ab	2115.573	0.641	0.234	0.077	1.000
T2-A3	ab	2002.583	0.618	0.240	0.167	1.000
T3-A1	bc	1947.934	0.609	0.233	0.112	1.000
T2-A4	bc	1946.495	0.605	0.245	0.037	1.000
T2-A1	bc	1944.552	0.604	0.233	0.038	0.998
T3-A3	cd	1806.979	0.575	0.242	0.000	0.990
T3-A2	cd	1805.658	0.576	0.238	0.112	1.000
T4-A1	de	1669.189	0.543	0.247	0.000	0.992
T4-A2	de	1627.231	0.535	0.248	0.000	0.999
T3-A4	e	1571.125	0.522	0.245	0.000	0.996
T4-A4	e	1546.626	0.514	0.263	0.000	0.999
T4-A3	e	1542.745	0.516	0.250	0.000	0.991

Fonte: Elaborada pelo autor

#### 5.2.4.5. Análise Estatística: Instâncias com 5 itens por Cidade

Essa é a última seção que trata sobre a calibração do algoritmo LaCSA em relação aos tipos de instâncias selecionadas para o experimento, havendo no final desta seção a conclusão sobre qual a configuração dos parâmetros que será usada na hora de efetuar a execução do algoritmo sobre as instâncias que terão seus resultados analisados no capítulo 6 e comparados com outras abordagens presentes na literatura.

Usando agora como fonte de dados as instâncias contendo 5 itens por cidade, sem fazer distinção entre o número de cidades contidas nas mesmas, o teste *ANOVA* tem seu resultado mostrado na Tabela 25. Para testar suas pressuposições, esse novo conjunto de dados necessita passar também pelos três testes (*Durbin – Watson*, *Levene e Shapiro – Wilk*) capazes de validar cada uma delas. Através da aplicação do teste *Durbin – Watson* obteve-se um valor  $d = 1,635$ , sendo esse valor maior do que o valor de  $dU (1,36)$  encontrado anteriormente, portanto, os dados analisados possuem independência de erros (Pressuposto I) dos resíduos e a análise prosseguirá para seu próximo teste. Para testar a igualdade de variância o teste de *Levene* foi aplicado e seu resultado é visto na Tabela 26. Por obter um  $P - Value > 0,05$ , é possível concluir que esse novo conjunto de dados, restrito às instâncias com 5 itens por cidade, está em homocedasticidade, validando assim o Pressuposto II da ANOVA e avançando para o próximo teste. Após a realização do teste *Shapiro – Wilk* os tratamentos fornecidos para a ANOVA foram considerados não pertencentes a uma distribuição normal (Tabela 27), portanto, o Pressuposto III é novamente rejeitado. Devido a isso, o mesmo processo

contendo testes paramétricos e não paramétricos usados anteriormente será necessário também para essa análise.

**Tabela 25** - Resultados do teste ANOVA sobre o impacto dos valores dos parâmetros no algoritmo quando aplicado nos resultados das instâncias de 5 itens por cidade

	<b>Grau de Liberdade</b>	<b>Soma dos Quadrados</b>	<b>Média dos Quadrados</b>	<b>F-Value</b>	<b>P-Value</b>
<b>ANOVA</b>	15	3.9	0.25984	3.37	1.12E-05

**Fonte:** Elaborada pelo autor

**Tabela 26** - Resultados do teste Levene para validação da Pressuposição II da ANOVA com as instâncias de 5 itens por cidade

	<b>Grau de Liberdade</b>	<b>F-Value</b>	<b>P-Value</b>
<b>Levene</b>	15	1,0677	0,3816

**Fonte:** Elaborada pelo autor

**Tabela 27** - Resultado do teste Shapiro-Wilk para validação da Pressuposição III da ANOVA com as instâncias de 5 itens por cidade

	<b>W</b>	<b>P-Value</b>
<b>Shapiro-Wilk</b>	0,9478	2,2E-16

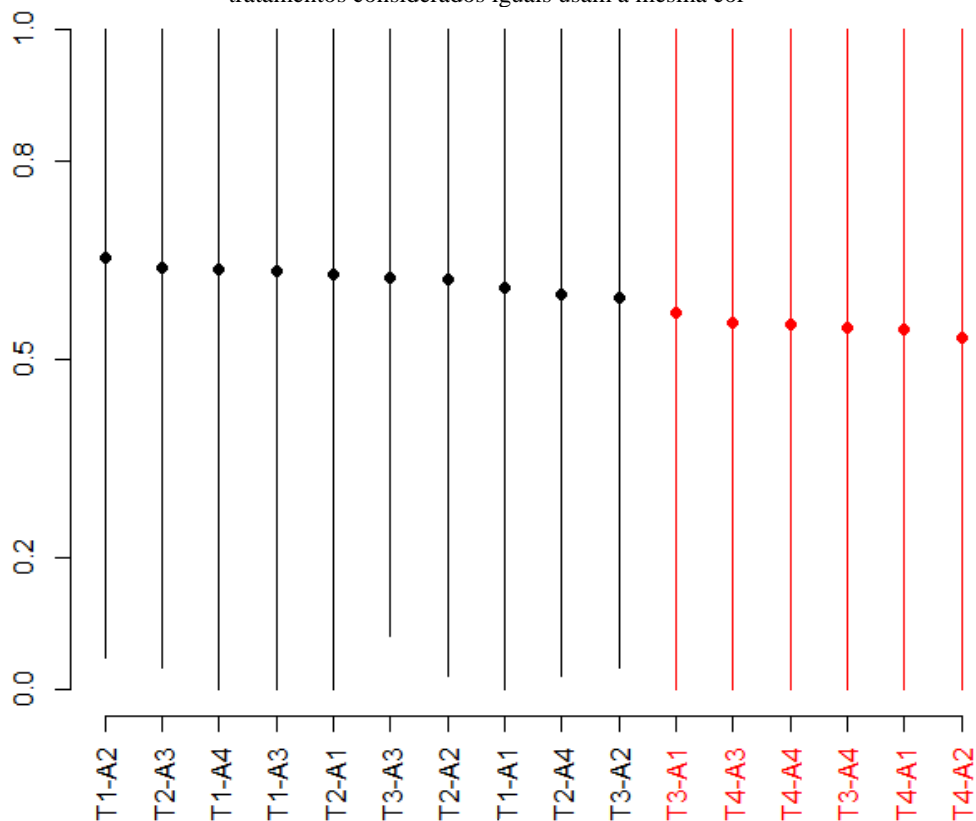
**Fonte:** Elaborada pelo autor

A Tabela 25 mostra que o teste ANOVA conseguiu obter um  $P - Value = 1.12E - 05$ , indicando haver diferenças significativas entre as médias dos tratamentos analisados. O teste ANOVA se mostrou resiliente aos tratamentos não pertencerem à distribuição normal, portanto, o teste paramétrico de *Scott – Knott* para comparações em duplas de tratamentos será aplicado para descobrir se há algum tratamento que conseguiu ser estatisticamente melhor do que algum outro. Na Tabela 28 o resultado desse teste é mostrado, validando assim o resultado entregue pela ANOVA. Na Figura 8 são mostradas as médias dos tratamentos, classificados por cores, agrupando assim os resultados estatisticamente iguais e distinguindo visualmente os conjuntos de parâmetros que impactaram no desempenho do algoritmo.

**Tabela 28** - Resultados do teste Scott-Knott usando a análise de variância com as instâncias de 5 itens por cidade

PARÂMETRO	MÉDIA	CLASSIFICAÇÃO
T1-A2	0.6551	a
T2-A3	0.6391	a
T1-A4	0.6373	a
T1-A3	0.6328	a
T2-A1	0.6276	a
T3-A3	0.6242	a
T2-A2	0.6209	a
T1-A1	0.6082	a
T2-A4	0.5992	a
T3-A2	0.5932	a
T3-A1	0.5698	b
T4-A3	0.5558	b
T4-A4	0.5522	b
T3-A4	0.5487	b
T4-A1	0.5463	b
T4-A2	0.5337	b

Fonte: Elaborada pelo autor

**Figura 8** - Gráfico do resultado do teste Scott-Knott aplicado nos tratamentos das instâncias com 76 cidades: tratamentos considerados iguais usam a mesma cor

Fonte: Elaborada pelo autor

Novamente o parâmetro T1-A2 é encontrado entre os tratamentos que conseguiram obter resultados estatisticamente superiores aos demais, portanto, seu uso

também é aconselhado para solucionar esse conjunto de instâncias. Após a realização do teste não paramétrico *Kruskal – Wallis*, o mesmo indicou com seus resultados mostrados na Tabela 29 que, novamente, mesmo com sua Pressuposição III sendo violada, o teste ANOVA em conjunto ao de *Scott – Knott* conseguiu entregar resultados similares aos obtidos pelo teste não paramétrico, confirmando assim a vantagem em usar o parâmetro T1-A2 para esse grupo de instâncias.

**Tabela 29** - Resultado do teste Kruskal-Wallis com o teste Fischer LSD para comparações múltiplas analisando as instâncias com 5 itens por cidade

PARÂMETRO	GRUPO	KW	MÉDIA	DESVIO PADRÃO	MIN	MAX
T1-A2	a	1452.483	0.655	0.274	0.048	1.000
T2-A3	ab	1413.389	0.639	0.261	0.031	1.000
T1-A3	ab	1404.571	0.633	0.283	0.000	1.000
T2-A1	ab	1392.682	0.628	0.292	0.000	1.000
T1-A4	abc	1391.351	0.637	0.263	0.000	1.000
T3-A3	abc	1366.616	0.624	0.264	0.080	1.000
T2-A2	abc	1357.458	0.621	0.266	0.020	1.000
T1-A1	abcd	1333.539	0.608	0.290	0.000	1.000
T2-A4	abcde	1299.348	0.599	0.273	0.020	1.000
T3-A2	bcde	1283.839	0.593	0.271	0.032	1.000
T3-A1	cde	1230.693	0.570	0.283	0.000	1.000
T4-A3	de	1188.190	0.556	0.275	0.000	1.000
T4-A4	de	1180.194	0.552	0.283	0.000	1.000
T3-A4	de	1166.756	0.549	0.274	0.000	1.000
T4-A1	de	1163.222	0.546	0.280	0.000	1.000
T4-A2	e	1147.087	0.534	0.303	0.000	1.000

**Fonte:** Elaborada pelo autor

Depois de realizar testes com os dados de todas as instâncias em conjunto e outros testes as separando por características em comum, é possível afirmar que os valores de 0,8% de temperatura (T1) e 30% de queda de temperatura por iteração (A2, alpha) quando usados pelo algoritmo LaCSA conseguem fazer com que o mesmo obtenha um melhor desempenho, encontrando resultados significativamente melhores do que vários dos outros valores de parâmetros aqui testados, com uma única exceção para a análise realizada sobre as instâncias com 76 cidades: para essas, nenhum parâmetro se saiu melhor, sendo todos classificados como capazes de serem usados pelo algoritmo sem haver variação significativa nas soluções encontradas por ele. Conclui-se então que os parâmetros definidos com esses valores, T1 (0,8%) e A2 (30%), devem ser usados pelo algoritmo com o objetivo de permiti-lo explorar melhor o espaço de busca.

## 6. RESULTADOS E DISCUSSÃO

Nesta seção são mostrados todos os resultados dos experimentos computacionais realizados, seguindo o método de experimentação definido previamente no capítulo 5, porém, usando mais instâncias do que lá relatadas, sendo elas: eil51, eil76, kroB100, rat195, st225, a280; mantendo as mesmas escolhas de características: 3 e 5 itens por cidade, tipos de itens b, u e usw, com capacidade `_3` e `_7` (Seção 5.1). Os resultados aqui usados foram obtidos pela média de 15 execuções independentes do algoritmo LaCSA para cada instância analisada, sendo os resultados das abordagens propostas por outros trabalhos, quando citados, tendo sido disponibilizados pelos autores das mesmas.

### 6.1. Comparação entre TPPH e TPPHx

Como discutido anteriormente, é de alta importância que uma heurística para selecionar um plano de coleta para uma dada rota seja capaz de descobrir o impacto real na solução quando um item é coletado, não sendo suficiente contar apenas com a razão entre seu peso e valor, pois o mesmo influencia com seu peso não apenas seu lucro, mas também o custo para percorrer a rota. Essa é a principal diferença entre as heurísticas TPPH e TPPHx, sendo que o “x” acrescentado ao nome se dá ao fato desta heurística usar uma taxa informada a ela para decidir quando que ela deve reavaliar o impacto real de cada item antes de decidir qual será coletado. Essa taxa não é fixada em um único valor, pois o algoritmo LaCSA se mostrou capaz de usar essa heurística de maneira a se guiar pelo espaço de busca de forma mais rápida usando valores para a taxa de atualização “x” que variam conforme a busca vai sendo realizada. Até o término da busca essa taxa de atualização tem seu valor diminuído para que ocorra uma maior quantidade de reavaliações do impacto real dos itens, chegando no término da busca com uma taxa suficientemente pequena, permitindo que essa reavaliação ocorra sempre que um item de peso significativo seja coletado. A ideia dessa abordagem originou-se da observação de que ao usar taxas que permitam uma busca mais rápida, isto é, com um menor número de reavaliações dos itens, a solução encontrada será pior, porém, será igualmente pior para todas as soluções do espaço de busca, permitindo que o algoritmo caminhe pelo mesmo mais rapidamente usando um valor heurístico alto para a taxa de atualização no início da busca e, quando estiver mais avançado nela, termine por priorizar a busca intensificando-a também na escolha de um bom plano de coleta, esperando até lá ter encontrado as rotas mais promissoras para o PMV.

Para analisar o desempenho entre as heurísticas TPPH e TPPHx, sendo esta última com diversas taxas usadas, foi realizado o seguinte experimento: foram geradas 40 rotas únicas para a instância `rat195_n970_bsc_07` usando a heurística de Lin-Kernighan, após isso, foram criados diversos planos de coletas para essa mesma rota

usando as configurações mostradas na Tabela 30 com seus devidos resultados. É usado o método de comparação conhecido como Desvio Percentual Relativo (do inglês: *Relative Percentage Deviation*, conhecido como RPD), que define os resultados através de um desvio percentual em relação ao melhor resultado encontrado. O RPD  $X^{RPD}$  de uma solução  $X$  qualquer pode ser dado como:  $X^{RPD} = \frac{X-X'}{X'}$ , sendo  $X'$  a melhor solução encontrada para o problema em questão; portanto, quanto menor for o valor de  $X^{RPD}$ , maior a aproximação do resultado da melhor solução encontrada, sendo 0,00 o valor RPD significante para as melhores soluções encontradas.

A heurística TPPHx com a configuração de 0% para a taxa de atualização dos impactos dos itens conseguiu, como era o esperado, obter os melhores resultados, ignorando o tempo que ela levou para isso. O tempo levado para a execução dessa heurística na instância analisada é menor do que 0,5 segundos, porém, a mesma heurística com uma taxa de valor 10% consegue ser executada com um tempo menor que 0,001 segundos: ambas configurações são muito rápidas, entretanto, quando usadas em um número grande de soluções, e.g. vizinhos 2-OPT, essa diferença se torna significativa. Porém, como conciliar o quanto se deve abrir mão da taxa de atualização em prol de uma busca mais diversificada feita pelo algoritmo LaCSA? A resposta para essa pergunta pode ser obtida através da análise do RPD da Tabela 30, sendo o primeiro o valor RPD entre a taxa com 15% e 0% e o segundo o valor RPD usando as taxas de 10% e 0%. Percebe-se que não há muita variação do RPD em nenhuma das 40 replicações. Esse efeito significa que, quando usada uma taxa pior, porém mais rápida, o LaCSA cria para todos os vizinhos daquela iteração um plano de coleta que tende a ser igualmente inferior para todos eles, o que significa que não influenciará na escolha de qual rota deve ser escolhida para seguir com a busca do PMV. Sendo assim, o LaCSA realiza uma “busca relaxada” durante parte do início de sua execução usando taxas para a heurística TPPHx que permita sua execução ser feita mais rapidamente, entregando um valor heurístico aproximado do real (usando a taxa de 0%) para que o LaCSA faça sua escolha de qual rota ser selecionada para a próxima iteração. No final do algoritmo, o mesmo recria o plano de coleta da melhor solução encontrada usando a heurística TPPHx com uma taxa de 0%, sendo essa a solução final retornada por essa abordagem.

**Tabela 30** – Resultados das heurísticas TPPH e TPPHx com diversos valores para as taxas usadas

ROTA	TPPH	TPPHx(21%)	TPPHx(15%)	TPPHx(10%)	TPPHx(5%)	TPPHx(2%)	TPPHx(0%)	RDP: 15% 0%	RDP 10% 0%
2333,87	-66860,06	146940,08	166983,37	167865,49	178167,62	179279,13	179914,47	0,07	0,01
2337,18	-65675,47	156176,02	169299,24	168583,90	179516,85	181092,07	181196,53	0,07	0,01
2337,18	-44980,45	175045,67	206157,72	208843,34	208880,85	212104,10	212557,28	0,03	0,02
2336,53	-67735,88	127767,68	166674,77	164756,73	178100,13	179826,98	180084,43	0,07	0,01
2338,14	-41641,99	176526,20	207822,26	210917,05	211322,59	213965,09	214314,58	0,03	0,01
2336,62	-69511,44	116230,20	167810,74	167734,14	178012,65	179870,00	180334,17	0,07	0,01
2338,41	-65912,50	152810,02	167716,08	167547,98	177589,52	180549,73	180670,87	0,07	0,02
2338,54	-65986,28	150822,72	166221,12	165238,67	177530,30	179737,26	180052,04	0,08	0,01
2338,14	-60842,81	160406,97	167230,18	166558,35	176764,73	178561,34	179155,58	0,07	0,01
2338,41	-48386,16	173315,73	204967,81	207511,52	208364,22	210982,75	212387,43	0,03	0,02
2344,36	-75608,74	112723,47	174222,65	177303,15	184785,80	185967,74	186636,71	0,07	0,01
2339,78	-64172,70	158459,22	198486,64	199588,14	200545,78	202826,30	203183,08	0,02	0,01
2339,22	-60722,72	160918,39	200697,17	202240,02	202366,23	205808,99	206070,83	0,03	0,02
2335,98	-35842,68	178756,70	209801,22	211847,69	212666,77	215010,17	215820,92	0,03	0,01
2338,05	-48679,79	163610,09	167027,60	165348,99	176583,06	177695,95	178985,23	0,07	0,01
2336,47	-49058,08	161446,04	200245,37	202564,56	202509,11	206355,84	206447,68	0,03	0,02
2336,47	-77805,06	111621,13	173032,13	176576,21	183937,72	185102,74	186175,34	0,07	0,01
2339,78	-72289,84	112993,59	176773,10	176366,62	184796,15	187093,26	187490,57	0,06	0,01
2340,78	-61868,32	160190,39	196952,39	199207,70	199624,17	204104,04	204658,36	0,04	0,02
2341,20	-39996,92	177172,37	208347,21	209011,31	211380,65	213317,10	214161,59	0,03	0,01
2339,93	-66407,61	150499,29	166816,32	165938,00	175985,38	178909,01	179311,83	0,07	0,02
2341,14	-64147,78	159142,11	198653,70	201194,68	200275,66	202969,57	203729,89	0,02	0,02
2338,37	-102911,73	106565,83	173407,59	174680,51	182553,53	183956,09	184331,04	0,06	0,01
2339,13	-60554,77	161162,52	200728,01	202360,59	202391,69	205834,41	206096,26	0,03	0,02
2341,14	-84754,94	109435,82	173869,11	177053,63	183513,93	185580,95	185844,55	0,06	0,01
2339,50	-48735,94	161891,19	166191,80	165480,20	175499,74	177609,73	178033,62	0,07	0,01
2340,09	-67877,90	126334,02	166378,55	168250,00	176801,29	179298,68	179377,86	0,07	0,01
2341,01	-78780,17	109996,55	174521,05	177129,88	184724,04	185864,74	186536,96	0,06	0,01
2344,58	-71797,53	115621,70	166431,24	164583,35	177557,99	178960,17	179100,27	0,07	0,01
2335,98	-68998,03	118444,83	168594,77	168214,97	177767,01	179535,04	180206,21	0,06	0,01
2339,22	-76470,40	112091,96	174411,34	177478,36	183734,78	185492,28	186246,90	0,06	0,01
2340,04	-66658,28	147660,02	169058,90	169880,31	180244,63	181110,24	181476,32	0,07	0,01
2337,18	-41697,82	175045,67	206157,72	208843,34	208880,85	212104,10	212557,28	0,03	0,02
2333,87	-37906,16	178438,80	209451,38	211932,63	212443,77	214858,79	215618,74	0,03	0,01
2335,98	-28824,16	178756,70	209801,22	211847,69	212666,77	215010,17	215820,92	0,03	0,01
2338,41	-46193,65	173315,73	204967,81	207511,52	208364,22	210982,75	212387,43	0,03	0,02
2337,47	-69251,14	117045,08	177656,76	179955,63	187291,81	187817,17	188252,25	0,06	0,01
2341,01	-64539,75	157818,43	197857,32	200230,85	200338,47	202908,84	203224,64	0,03	0,01
2340,78	-68377,03	125721,68	167200,54	169234,94	179215,57	180889,44	181130,94	0,08	0,01

Fonte: Elaborada pelo autor

Muitas das características do LaCSA foram elaboradas através da experimentação do mesmo e da análise de como ele se comportava nesses experimentos. O resultado disso é que algumas abordagens heurísticas começaram a usar outras informações heurísticas dentro de si, o que poderia ocasionar em uma forma baseada em estimativas, por vezes erradas, para solucionar o problema; entretanto, os resultados finais, discutidos até o final deste capítulo, mostraram que o uso de informações específicas sobre a interdependência do problema ajudou a diminuir o espaço de busca a ser explorado, focando mais em lugares promissores e diminuindo o número de avaliações necessárias para encontrar a melhor solução, permitindo também que instâncias um pouco maiores conseguissem ser solucionadas de forma satisfatória.

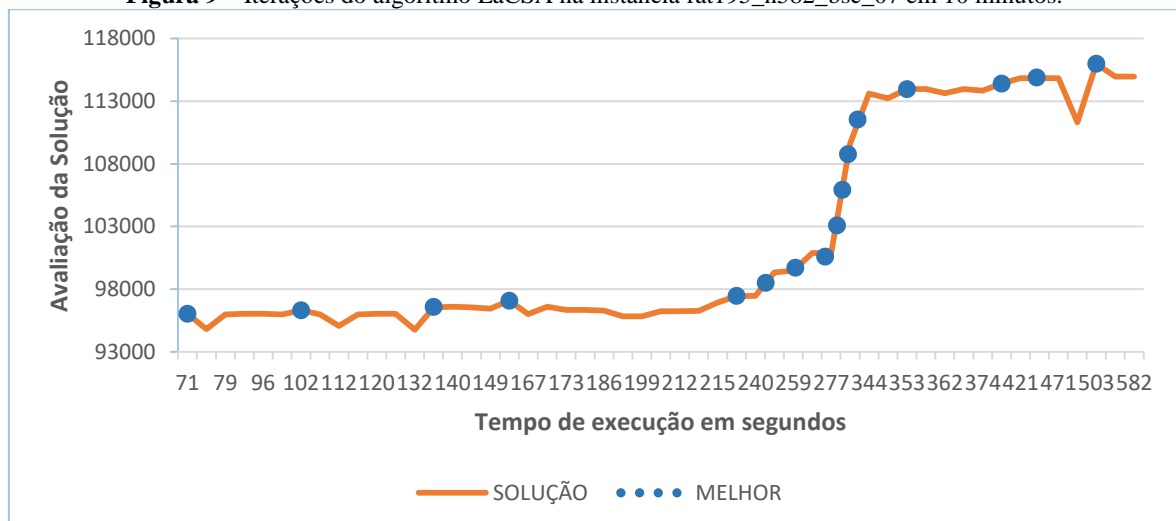
## 6.2. Direcionamento da Busca LaCSA por Iteração

É importante conhecer a forma como qualquer heurística percorre seu espaço de busca à procura de melhores soluções que a atual ou, pelo menos, novas soluções capazes de apontar para um caminho promissor. Meta-heurísticas com características

probabilísticas como o *Simulated Annealing* necessitam ser ajustadas para que sua busca não fique diversificada demais ou muito intensificada em uma mesma região. Essa importância é ainda maior quando o problema tratado é de alta complexidade e possui um espaço de busca muito grande e diversificado. Para piorar, relacionamentos como os presentes no PMV possibilitam cenários onde uma boa solução, após uma fração de sua representação ser alterada, mantendo grande parte da solução anterior, consegue causar grande impacto, negativo ou positivo, na qualidade da nova solução, tornando mais difícil, por exemplo, saber quais partes dos componentes de uma solução podem ser consideradas como interessantes, bastando mudar uma pequena parte na solução do outro componente (trocar de lugar duas cidades da rota ou coletar um item pesado no início da mesma) para que a qualidade da solução completa seja significativamente alterada.

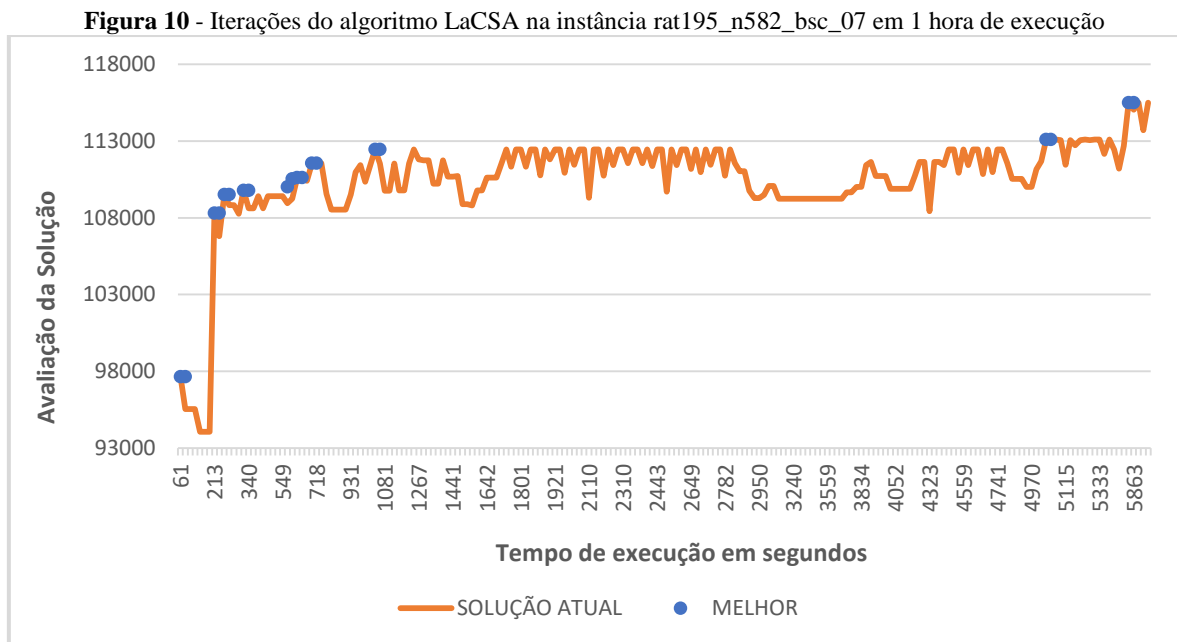
Para avaliar a forma que o LaCSA conduz sua busca, o mesmo foi analisado iteração por iteração e o resultado de suas escolhas são mostradas na Figura 9 e 11, sendo, respectivamente, para as soluções `rat195_n582_bounded-strongly-corr_07` e `eil76_n225_bounded-strongly-corr_03`. Em ambas figuras são mostradas as iterações de uma execução real do algoritmo LaCSA e suas escolhas em relações às soluções aceitas pelo critério de aceitação ou por serem as melhores encontradas até o momento. A seguir, nas Figuras 10 e 12 são mostradas as iterações para as mesmas instâncias, porém, o tempo de execução de ambas foi aumentado de 10 minutos para 1 hora, mantendo apenas 1 minuto como tempo de preparação da solução inicial, pois o objetivo é verificar a forma como a busca ocorre sob circunstâncias normais, porém, por um tempo mais prolongado, verificando se há anormalidades nas escolhas das soluções aceitas.

**Figura 9** – Iterações do algoritmo LaCSA na instância `rat195_n582_bsc_07` em 10 minutos.



Fonte: Elaborada pelo autor

É possível observar nas Figuras 9 e 10 que o LaCSA percorre o espaço de busca da mesma instância de maneira bem diferente, mesmo iniciando com uma solução inicial de valores similares e conseguindo encontrar uma solução final de valor aproximado em ambos cenários. As melhores soluções encontradas, representadas pelos pontos azuis, mostram também que foram encontradas em momentos diferentes, sendo que para o cenário com 1 hora de tolerância houve descobertas de novas “melhores soluções” também no final da busca.

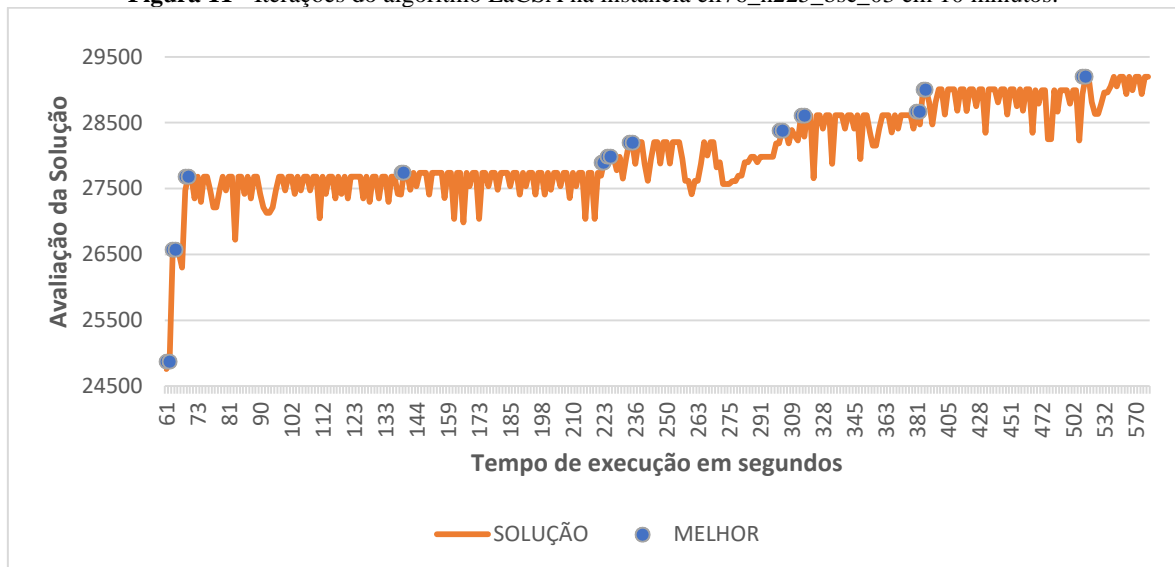


Fonte: Elaborada pelo autor

Nas Figuras 11 e 12 a mesma conclusão pode ser tirada: o algoritmo LaCSA consegue explorar o espaço de busca do problema de maneira organizada, variando por diversas vezes entre as mais variadas soluções com valores próximos uma das outras até, eventualmente, encontrar algumas soluções que se destacam durante esse processo. É interessante notar que para essas duas imagens também é possível ver uma busca que se intensifica em soluções com valores similares, mas, vez ou outra, ocorre uma maior diversificação e soluções piores são aceitas, mas, em seguida, novas “melhores soluções” são encontradas, levando o algoritmo para um espaço de busca de maior qualidade (linha laranja constante mais alta). Na Figura 11 é possível observar que o algoritmo LaCSA aceitou algumas soluções ruins que levou a busca a explorar locais de baixa qualidade do espaço de busca, sendo que o algoritmo conseguiu se recuperar apenas no final. Esse evento pode ter ocorrido porque o algoritmo teve seus parâmetros calibrados para levar em consideração apenas buscas com 10 minutos como critério de parada: elevar a busca para 1 hora pode ter aumentado o número de vezes que a probabilidade de soluções ruins serem aceitas, porém, com temperaturas um pouco menores isso poderia não ter acontecido, mas essa possibilidade não existia com a busca limitada a 10 minutos e

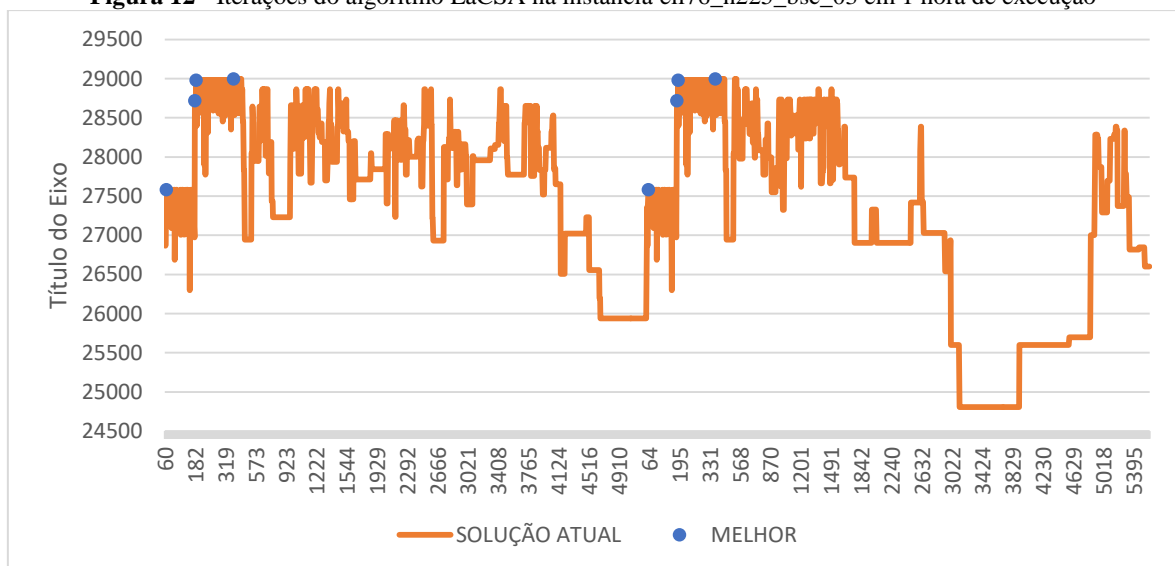
certamente não foi levada em consideração no momento da calibração, não que isso seja ruim: o algoritmo deve saber se comportar bem em situações onde ele será usado, sendo irrealistas esses gráficos com o tempo de 1 hora de execução, servindo apenas para verificar se há anomalias no processo de busca ou se novas soluções promissoras podem ser encontradas se mais tempo for fornecido para a exploração do problema.

**Figura 11** - Iterações do algoritmo LaCSA na instância eil76\_n225\_bsc\_03 em 10 minutos.



Fonte: Elaborada pelo autor

**Figura 12** - Iterações do algoritmo LaCSA na instância eil76\_n225\_bsc\_03 em 1 hora de execução



Fonte: Elaborada pelo autor

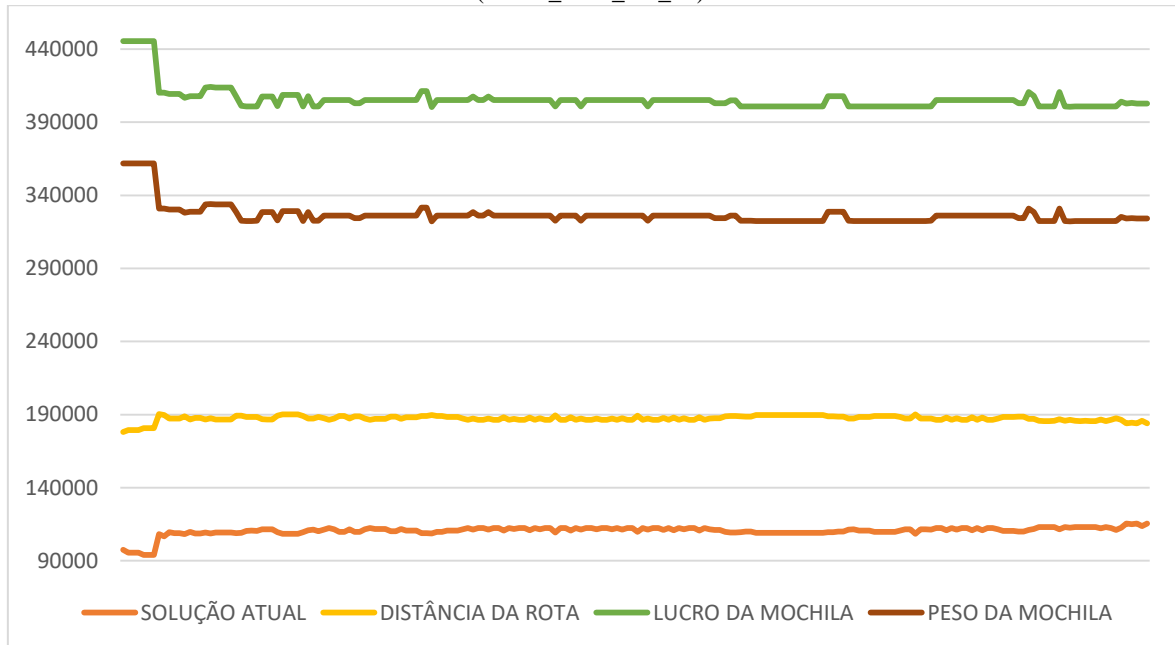
Outra análise bastante interessante pode ser feita através da compreensão dos resultados mostrados nas Figuras 13 e 14, sendo essas figuras responsáveis, novamente, por mostrar as iterações da busca LaCSA, porém, desta vez, destacando a mudança nos valores dos componentes do problema (PCV e PM) quando sua solução completa (PMV)

é alterada a cada iteração. As mesmas instâncias anteriores foram usadas para gerar os gráficos dessas figuras, isso porque é importante que os itens das mochilas sigam uma característica de que seus pesos e valores estão relacionados (bsc) para que conclusões erradas não sejam tiradas devido à existência de itens desproporcionais.

No início do gráfico mostrado pela Figura 13 é possível observar uma situação interessante: entre a solução inicial e a primeira melhoria significativa que ocorre há I) uma queda no lucro dos itens coletados, II) uma queda no peso total dos itens coletados e III) um aumento na distância da rota. Se uma solução do PMV é avaliada como sendo a diferença do lucro dos itens coletados pelo custo gasto para percorrer a rota, seria intuitivo concluir que essas mudanças ocasionariam em uma queda grande de qualidade para a solução PMV completa, entretanto, o contrário ocorreu, elevando a solução de um valor aproximado de 95.000 para 108.000. Isso pode ocorrer se o LaCSA conseguir encontrar uma rota maior, mas com itens que sejam mais interessantes em seu final, fazendo com que menos itens sejam coletados, diminuindo assim o lucro dos mesmos, porém, aumentando o lucro final do PMV devido ao somatório do peso desses itens ser menor e impactar menos no custo para percorrer a rota, diminuindo assim o fator negativo da avaliação da solução a um ponto que essa diminuição possa superar a diminuição do lucro dos itens que deixaram de ser coletados.

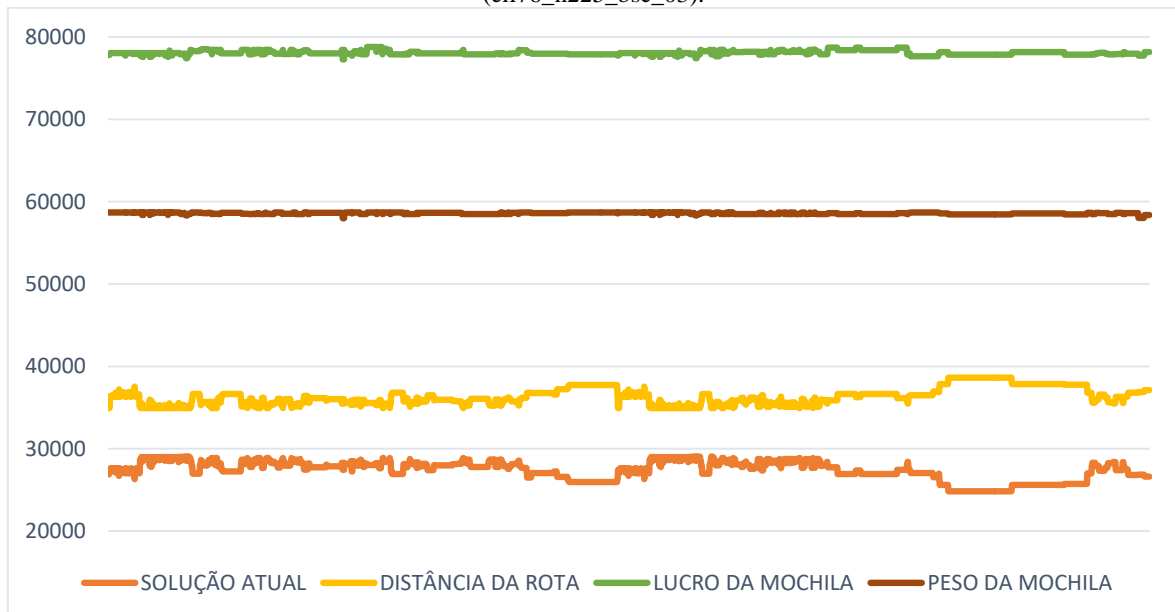
Outra relação nítida nas Figuras 13 e 14 ocorre entre I) o peso da mochila e o lucro dos itens coletados e II) a distância da rota e a qualidade da solução PMV, sendo possível concluir que a estratégia de explorar rotas diversificadas dando prioridade para as menores é uma boa abordagem e que a heurística TPPHx consegue obter um lucro proporcional ao peso dos itens selecionados para as diversas soluções que compõe os gráficos, sendo esta característica uma possível resposta da reavaliação dos impactos reais que os itens causam na solução quando são coletados, impedindo que más escolhas possam interferir no resultado final da heurística conforme a mesma vai aumentando o peso dos itens coletados.

**Figura 13** – Avaliação da relação entre os componentes das soluções usadas pela busca LaCSA a cada iteração (rat195\_n582\_bsc\_07).



Fonte: Elaborada pelo autor

**Figura 14** - Avaliação da relação entre os componentes das soluções usadas pela busca LaCSA a cada iteração (eil76\_n225\_bsc\_03).



Fonte: Elaborada pelo autor

### 6.3. Resultados dos Experimentos Computacionais e Análise de Desempenho

Nas Tabelas 31 e 32 são apresentados os resultados obtidos pela meta-heurística LaCSA aqui proposta e outras abordagens presentes na literatura sobre o Problema do

Mochileiro Viajante. Dentre os algoritmos escolhidos da literatura, estão presentes o I) Algoritmo Evolucionário (EA) e II) uma Busca Local Aleatória (RLS) propostos por Polyakovskiy et al (2014); o Algoritmo Memético (MATLS) proposto por Mei et al (2014a) e os 11 diferentes algoritmos propostos mais recentemente por Faulkner et al (2015). Em seu trabalho, Faulkner relatou que, dentre suas abordagens, uma conhecida como “S5” foi capaz de encontrar os melhores resultados para as instâncias analisadas pelo mesmo, entretanto, essa conclusão não é verdadeira para o conjunto de instâncias aqui analisados, com foco em instâncias de pequeno e médio tamanho; devido a isso, todas suas 11 abordagens serão mostradas e discutidas a seguir, uma vez que todas elas conseguiram, para uma ou outra instância, resultados significativos que merecem ser citados.

Todos os algoritmos aqui descritos usaram como critério de parada um tempo de 10 minutos de execução, isto é, a busca é interrompida após esse tempo ser alcançado e a melhor solução encontrada até o momento é retornada. Informações mais detalhadas sobre o método de experimentação usado pelo algoritmo LaCSA podem ser lidas no capítulo 5 deste trabalho, sendo ele condizente com o método usado nos demais trabalhos onde as abordagens citadas anteriormente foram propostas.

Os resultados para cada instância com característica diferente para os grupos eil51, eil76, kroB100 são mostradas na Tabela 32 e na Tabela 32 para os grupos rat195, st225 e a280. O método de comparação usado é o Desvio Percentual Relativo (do inglês: *Relative Percentage Deviation*, conhecido como RPD), que define os resultados através de um desvio percentual em relação ao melhor resultado encontrado. O RPD  $X^{RPD}$  de uma solução  $X$  qualquer pode ser dado como:  $X^{RPD} = \frac{X-X'}{X'}$ , sendo  $X'$  a melhor solução encontrada para o problema em questão; portanto, quanto menor for o valor de  $X^{RPD}$ , maior a aproximação do resultado da melhor solução encontrada, sendo 0,00 o valor RPD significante para as melhores soluções encontradas.

É possível observar que para as instâncias com 51, 76 ou 100 cidades, contendo, respectivamente, 150 e 250, 225 e 375 ou 297 e 495 itens; com seus resultados mostrados na Tabela 31, que a abordagem LaCSA consegue obter os melhores resultados para quase todos os cenários, sendo na Tabela 34 a relação entre o desempenho dessas diversas abordagens mostradas de forma mais direta. Ainda na Tabela 31, é possível verificar, através dos valores em negrito, que a abordagem LaCSA consegue obter os melhores resultados para todas as 12 instâncias do grupo eil51 aqui analisadas, não conseguindo obter a melhor média de resultados para apenas 1 das 12 instância do grupo eil76 e perdendo em 4 de 12 instâncias do grupo kroB100. Esses três conjuntos de instância são os que melhores foram solucionados pelo algoritmo LaCSA, comprovando assim a eficácia do mesmo em solucionar instâncias menores para o PMV.

Tabela 31 - Resultados de todas as abordagens analisadas para as instâncias com 51, 76 e 100 cidades.

Instância	LaCSA	MATLS	S1	S2	S3	S4	S5	C1	C2	C3	C4	C5	C6	RLS	EA
eil51_n150_bsc_03	<b>0.000</b>	0,135	0,113	0,113	0,111	0,113	0,113	0,313	0,313	0,113	0,113	0,113	0,111	0,157	0,115
eil51_n150_bsc_07	<b>0.000</b>	0,176	0,162	0,583	0,162	0,160	0,160	0,160	0,160	0,160	0,160	0,160	0,160	0,163	0,162
eil51_n150_u_03	<b>0.000</b>	0,088	0,225	0,207	0,206	0,207	0,207	0,207	0,206	0,207	0,207	0,207	0,206	0,221	0,225
eil51_n150_u_07	<b>0.000</b>	0,208	0,280	0,277	0,277	0,277	0,277	0,277	0,277	0,277	0,277	0,277	0,277	0,280	0,280
eil51_n150_usw_03	<b>0.000</b>	0,057	0,245	0,245	0,152	0,149	0,149	0,149	0,148	0,149	0,149	0,149	0,148	0,163	0,156
eil51_n150_usw_07	<b>0.000</b>	0,181	0,264	0,287	0,285	0,264	0,264	0,264	0,264	0,264	0,264	0,264	0,264	0,268	0,268
eil51_n250_bsc_03	<b>0.000</b>	0,090	0,234	0,108	0,234	0,234	0,108	0,234	0,108	0,108	0,108	0,108	0,108	0,117	0,112
eil51_n250_bsc_07	<b>0.000</b>	0,209	0,373	0,211	0,372	0,373	0,212	0,372	0,211	0,211	0,211	0,211	0,211	0,214	0,215
eil51_n250_u_03	<b>0.000</b>	0,122	0,221	0,248	0,217	0,248	0,219	0,217	0,217	0,217	0,217	0,217	0,217	0,219	0,219
eil51_n250_u_07	<b>0.000</b>	0,182	0,239	0,238	0,262	0,262	0,239	0,238	0,262	0,238	0,238	0,238	0,238	0,241	0,241
eil51_n250_usw_03	<b>0.000</b>	0,150	0,375	0,221	0,364	0,375	0,227	0,221	0,363	0,221	0,220	0,221	0,220	0,227	0,224
eil51_n250_usw_07	<b>0.000</b>	0,190	0,241	0,240	0,242	0,241	0,241	0,240	0,240	0,240	0,240	0,240	0,240	0,242	0,242
eil76_n225_bsc_03	<b>0.000</b>	0,045	0,160	0,161	0,232	0,163	0,013	0,013	0,158	0,013	0,013	0,013	0,012	0,161	0,162
eil76_n225_bsc_07	<b>0.000</b>	0,117	0,039	0,161	0,161	0,157	0,037	0,157	0,161	0,038	0,038	0,037	0,037	0,162	0,162
eil76_n225_u_03	<b>0.000</b>	0,200	0,299	0,231	0,297	0,233	0,233	0,248	0,230	0,230	0,230	0,230	0,230	0,231	0,231
eil76_n225_u_07	<b>0.000</b>	0,116	0,262	0,156	0,346	0,156	0,153	0,156	0,262	0,153	0,153	0,153	0,153	0,156	0,156
eil76_n225_usw_03	<b>0.000</b>	0,076	0,558	0,276	0,272	0,557	0,213	0,275	0,270	0,236	0,234	0,213	0,213	0,278	0,279
eil76_n225_usw_07	<b>0.000</b>	0,116	0,559	0,218	0,431	0,557	0,214	0,213	0,556	0,209	0,209	0,209	0,209	0,218	0,218
eil76_n375_bsc_03	<b>0.000</b>	0,033	0,103	0,136	0,102	0,103	0,020	0,133	0,180	0,021	0,020	0,020	0,019	0,141	0,138
eil76_n375_bsc_07	0,002	0,033	0,235	0,095	0,095	0,234	<b>0.000</b>	0,234	0,235	0,000	0,000	0,000	0,000	0,096	0,097
eil76_n375_u_03	<b>0.000</b>	0,122	0,165	0,161	0,216	0,164	0,164	0,172	0,334	0,160	0,160	0,160	0,160	0,161	0,161
eil76_n375_u_07	<b>0.000</b>	0,103	0,203	0,136	0,133	0,136	0,133	0,402	0,402	0,133	0,133	0,133	0,133	0,136	0,136
eil76_n375_usw_03	<b>0.000</b>	0,096	0,456	0,249	0,451	0,454	0,157	0,407	0,452	0,164	0,157	0,157	0,156	0,247	0,249
eil76_n375_usw_07	<b>0.000</b>	0,145	0,211	0,211	0,531	0,479	0,208	0,530	0,211	0,208	0,208	0,208	0,208	0,210	0,210
kroB100_n297_bsc_03	0,043	0,020	0,262	0,262	0,262	0,262	0,001	0,001	0,001	0,003	0,002	0,001	<b>0.000</b>	0,263	0,263
kroB100_n297_bsc_07	<b>0.000</b>	0,068	0,357	0,001	0,357	0,001	0,001	0,001	0,327	0,002	0,001	0,001	0,001	0,357	0,357
kroB100_n297_u_03	<b>0.000</b>	0,005	0,141	0,017	0,017	0,017	0,017	0,017	0,141	0,017	0,018	0,017	0,017	0,141	0,141
kroB100_n297_u_07	<b>0.000</b>	0,013	0,249	0,020	0,020	0,234	0,020	0,020	0,201	0,020	0,020	0,020	0,020	0,249	0,249
kroB100_n297_usw_03	<b>0.000</b>	0,025	0,675	0,042	0,671	0,647	0,042	0,609	0,609	0,042	0,042	0,042	0,042	0,674	0,671
kroB100_n297_usw_07	<b>0.000</b>	0,008	0,544	0,007	0,007	0,536	0,007	0,007	0,536	0,007	0,007	0,007	0,007	0,543	0,543
kroB100_n495_bsc_03	0,002	0,032	0,226	0,001	0,225	0,226	0,001	0,001	0,225	0,001	0,001	0,001	<b>0.000</b>	0,234	0,227
kroB100_n495_bsc_07	<b>0.000</b>	0,009	0,006	0,006	0,326	0,309	0,006	0,006	0,006	0,006	0,006	0,006	0,006	0,327	0,326
kroB100_n495_u_03	0,009	0,018	0,161	<b>0.000</b>	<b>0.000</b>	0,000	0,000	0,157	0,157	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	0,161	0,161
kroB100_n495_u_07	<b>0.000</b>	0,009	0,004	0,004	0,270	0,004	0,004	0,265	0,004	0,004	0,004	0,004	0,004	0,270	0,270
kroB100_n495_usw_03	0,169	0,010	0,721	0,010	0,719	0,721	0,002	0,719	0,719	0,010	0,011	<b>0.000</b>	<b>0.000</b>	0,719	0,719
kroB100_n495_usw_07	<b>0.000</b>	0,002	0,515	0,007	0,007	0,505	0,008	0,481	0,007	0,007	0,007	0,007	0,007	0,514	0,514

Fonte: Elaborada pelo autor

Na Tabela 32 é possível ver os resultados para as instâncias com 195, 225 e 280 cidades. Para o primeiro grupo dessas instâncias, rat195, a abordagem LaCSA conseguiu 9 dos 12 melhores resultados para as instâncias resolvidas, perdendo em duas delas para a abordagem C5 e uma derrota para a abordagem C6. Pode ser observado também que para as instâncias com 225 o LaCSA conseguiu 8 dos 12 melhores resultados, perdendo para o algoritmo MATLS em duas situações e para o algoritmo C5 nos dois cenários com 3 itens por cidade do tipo de itens “bsc”. As duas instâncias em que a abordagem MATLS conseguiu vencer foram com 5 itens por cidade e de capacidade máxima da mochila mais restritiva (\_03). Conforme as instâncias vão crescendo, como visto nos casos com 280 cidades, o desempenho do algoritmo LaCSA vai caindo, chegando a ganhar em 7 dos 12 casos analisados, entretanto, é importante notar que desses 5 cenários perdidos o algoritmo S5 foi responsável por vencer em 3, sendo este um total de quase 43% em relação ao número de vitórias obtidas pelo LaCSA para esse mesmo grupo que pode se considerado o mais difícil de ser resolvido por esta abordagem.

Como mostrado na Tabela 33, a abordagem LaCSA conseguiu vencer em 55 das 72 instâncias avaliadas, havendo encontrado o segundo melhor resultado em duas situações e também o pior resultado em outros dois cenários. Uma informação importante se destaca também desta tabela: o algoritmo MATLS, por mais que tenha

vencido em apenas 3 das 72 instâncias, empatando nesses números com a abordagem C6 e perdendo para a abordagem LaCSA, S4 e C5; por sua vez, conseguiu obter as 30 segundas melhores soluções e 2 das terceiras melhores soluções, não apresentando qualquer pior resultado para qualquer instância. Em seu trabalho, Faulkner et al (2015) relatou que sua abordagem de melhor desempenho teria sido a S5, entretanto, não foram esses os resultados mostrados para as instâncias aqui analisadas: a S5 consegue apenas 4 dos melhores resultados e 7 dos segundos melhores, enquanto sua heurística C5 consegue obter 6 dos melhores resultados e 10 dos segundos melhores, tornando a heurística S5 de pior eficácia quando comparadas com as abordagens C5 e MATLS. Uma hipótese: a não separação dos grupos de instâncias na hora de analisar os resultados pode ter influenciado no resultado dos autores: a abordagem S5 pode ter conseguido resultados melhores para as instâncias maiores analisadas por Faulkner, sendo estas fora do escopo deste trabalho. A premissa é que as instâncias maiores podem ser bastante difíceis de serem bem conduzidas por uma busca devido a seu grande espaço de busca, sendo assim, heurísticas simples, como a S5, podem conseguir bons resultados onde outras heurísticas mais complexas não conseguem realizar suas buscas de forma eficiente, com menos iterações do que o necessário para uma diversificação em busca de uma intensificação bem localizada. Além disso, heurísticas mais complexas como a MATLS, C5, C6, que obtiveram resultados superiores a S5 para as instâncias de pequena e média escala, também estão condizentes com o que se esperaria delas: resultados superiores do que os encontrados por heurísticas mais simples (i.e.: S1,S2,...,S5) em cenários onde o espaço de busca é menor e consegue ser suficientemente explorado.

As abordagens propostas por Polyakovsky (2014<sup>a</sup>), EA e RLS, não conseguiram obter nenhum dos três melhores resultados para qualquer uma das instâncias aqui analisadas, mostrando que os diversos algoritmos propostos para solucionar o problema desde sua definição vem, um após o outro, melhorando a qualidade das soluções encontradas ao solucionar o problema usando informações específicas de como sua interdependência pode ser usada para guiar melhor a busca.

**Tabela 32** – Resultados de todas as abordagens analisadas para as instâncias com 195, 225 e 280 cidades.

Instância	LaCSA	MATLS	S1	S2	S3	S4	S5	C1	C2	C3	C4	C5	C6	RLS	EA
rat195_n582_bsc_03	0,032	0,040	0,014	0,007	0,014	0,001	0,001	0,178	0,007	0,001	0,001	0,000	<b>0,000</b>	0,176	0,177
rat195_n582_bsc_07	<b>0,000</b>	0,058	0,061	0,054	0,194	0,060	0,043	0,185	0,054	0,046	0,054	0,046	0,046	0,184	0,184
rat195_n582_u_03	<b>0,000</b>	0,040	0,059	0,030	0,045	0,042	0,016	0,028	0,068	0,028	0,028	0,028	0,028	0,054	0,054
rat195_n582_u_07	<b>0,000</b>	0,061	0,086	0,092	0,079	0,087	0,060	0,086	0,092	0,074	0,079	0,074	0,079	0,086	0,086
rat195_n582_usw_03	<b>0,000</b>	0,075	0,137	0,117	0,152	0,134	0,109	0,134	0,131	0,104	0,104	0,104	0,104	0,138	0,138
rat195_n582_usw_07	<b>0,000</b>	0,063	0,103	0,145	0,163	0,165	0,103	0,102	0,145	0,102	0,102	0,102	0,102	0,104	0,104
rat195_n970_bsc_03	0,008	0,036	0,007	0,007	0,194	0,168	0,001	0,196	0,007	0,001	0,001	<b>0,000</b>	0,000	0,194	0,195
rat195_n970_bsc_07	0,002	0,088	0,176	0,026	0,026	0,138	0,011	0,014	0,026	0,011	0,011	<b>0,000</b>	0,011	0,174	0,174
rat195_n970_u_03	<b>0,000</b>	0,057	0,046	0,045	0,085	0,104	0,046	0,085	0,085	0,045	0,045	0,045	0,045	0,084	0,084
rat195_n970_u_07	<b>0,000</b>	0,057	0,065	0,064	0,080	0,064	0,053	0,064	0,052	0,052	0,052	0,052	0,043	0,080	0,080
rat195_n970_usw_03	<b>0,000</b>	0,083	0,101	0,108	0,099	0,129	0,098	0,117	0,132	0,097	0,087	0,087	0,096	0,097	0,097
rat195_n970_usw_07	<b>0,000</b>	0,098	0,172	0,170	0,170	0,172	0,107	0,107	0,170	0,107	0,107	0,092	0,107	0,107	0,107
ts225_n672_bsc_03	0,030	0,003	0,021	0,057	0,030	0,050	0,002	0,021	0,024	0,011	0,015	<b>0,000</b>	0,002	0,041	0,027
ts225_n672_bsc_07	0,030	0,007	0,141	0,181	0,068	0,135	0,004	0,023	0,123	0,004	0,004	<b>0,000</b>	0,003	0,065	0,065
ts225_n672_u_03	<b>0,000</b>	0,007	0,035	0,044	0,030	0,007	0,004	0,063	0,028	0,006	0,009	0,004	0,003	0,027	0,027
ts225_n672_u_07	<b>0,000</b>	0,003	0,035	0,023	0,102	0,042	0,012	0,064	0,096	0,018	0,016	0,010	0,008	0,033	0,033
ts225_n672_usw_03	<b>0,000</b>	0,041	0,209	0,275	0,098	0,230	0,029	0,268	0,258	0,082	0,045	0,032	0,035	0,143	0,144
ts225_n672_usw_07	<b>0,000</b>	0,025	0,099	0,119	0,100	0,119	0,032	0,082	0,064	0,030	0,052	0,036	0,032	0,081	0,081
ts225_n1120_bsc_03	0,012	<b>0,000</b>	0,041	0,027	0,028	0,035	0,003	0,052	0,037	0,004	0,003	0,002	0,003	0,012	0,012
ts225_n1120_bsc_07	<b>0,000</b>	0,075	0,141	0,067	0,054	0,149	0,039	0,140	0,042	0,038	0,040	0,038	0,041	0,086	0,086
ts225_n1120_u_03	<b>0,000</b>	0,059	0,069	0,051	0,063	0,087	0,038	0,050	0,067	0,042	0,043	0,043	0,048	0,063	0,063
ts225_n1120_u_07	<b>0,000</b>	0,013	0,064	0,063	0,069	0,078	0,024	0,045	0,087	0,023	0,026	0,022	0,025	0,045	0,045
ts225_n1120_usw_03	0,020	<b>0,000</b>	0,160	0,073	0,165	0,215	0,034	0,050	0,076	0,038	0,034	0,032	0,035	0,086	0,087
ts225_n1120_usw_07	<b>0,000</b>	0,029	0,144	0,115	0,095	0,100	0,038	0,113	0,059	0,032	0,038	0,033	0,035	0,076	0,076
a280_n837_bsc_03	0,021	0,024	0,064	0,006	0,004	0,077	<b>0,000</b>	0,007	0,067	0,001	0,001	0,001	0,001	0,066	0,067
a280_n837_bsc_07	0,156	0,011	0,017	0,017	0,037	0,039	<b>0,000</b>	0,037	0,040	0,001	0,001	0,001	0,002	0,026	0,026
a280_n837_u_03	<b>0,000</b>	0,021	0,012	0,064	0,016	0,011	0,004	0,016	0,021	0,005	0,004	0,004	0,003	0,008	0,008
a280_n837_u_07	<b>0,000</b>	0,003	0,025	0,024	0,021	0,016	0,011	0,029	0,021	0,011	0,015	0,011	0,012	0,020	0,020
a280_n837_usw_03	0,023	0,022	0,053	0,070	0,046	0,069	<b>0,000</b>	0,058	0,027	0,001	0,010	0,000	0,002	0,033	0,033
a280_n837_usw_07	<b>0,000</b>	0,030	0,049	0,049	0,025	0,041	0,017	0,035	0,037	0,017	0,017	0,017	0,018	0,046	0,046
a280_n1395_bsc_03	0,024	0,042	0,046	0,046	0,047	0,046	0,001	0,050	0,057	0,000	0,000	<b>0,000</b>	0,001	0,045	0,046
a280_n1395_bsc_07	0,213	<b>0,000</b>	0,020	0,031	0,024	0,021	0,010	0,056	0,013	0,013	0,013	0,012	0,013	0,016	0,016
a280_n1395_u_03	<b>0,000</b>	0,018	0,016	0,024	0,026	0,025	0,007	0,015	0,020	0,008	0,008	0,007	0,007	0,007	0,007
a280_n1395_u_07	<b>0,000</b>	0,012	0,029	0,026	0,015	0,034	0,011	0,027	0,023	0,010	0,013	0,010	0,012	0,026	0,026
a280_n1395_usw_03	<b>0,000</b>	0,014	0,057	0,063	0,047	0,039	0,009	0,057	0,046	0,006	0,015	0,013	0,006	0,054	0,054
a280_n1395_usw_07	<b>0,000</b>	0,019	0,059	0,076	0,039	0,050	0,015	0,061	0,072	0,015	0,017	0,014	0,015	0,060	0,060

Fonte: Elaborada pelo autor

**Tabela 33** – Resultados gerais para todas as instâncias analisadas em relação a cada abordagem

ABORDAGEM	MÉDIA RDP	DES. P.	MELHOR RDP	PIOR RDP	MELHOR	2º MELHOR	3º MELHOR	PIOR
LaCSA	0,011	0,037	0,000	0,213	55	2	0	2
MATLS	0,062	0,058	0,000	0,209	3	30	2	0
S1	0,177	0,164	0,004	0,721	0	1	1	18
S2	0,111	0,105	0,000	0,583	1	3	6	10
S3	0,161	0,153	0,000	0,719	0	1	5	11
S4	0,177	0,164	0,000	0,721	0	0	1	8
S5	0,070	0,085	0,000	0,277	4	7	10	0
C1	0,150	0,149	0,001	0,719	0	0	1	6
C2	0,160	0,153	0,001	0,719	0	1	2	7
C3	0,072	0,084	0,000	0,277	0	6	14	0
C4	0,072	0,084	0,000	0,277	0	1	6	0
C5	0,070	0,084	0,000	0,277	6	10	11	0
C6	0,070	0,084	0,000	0,277	3	8	5	0
RLS	0,164	0,140	0,007	0,719	0	0	0	5
EA	0,163	0,140	0,007	0,719	0	0	0	1

Fonte: Elaborada pelo autor

Apesar dos resultados indicarem que a abordagem LaCSA consegue obter resultados satisfatórios e, aparentemente, de maior significância do que as outras abordagens aqui analisadas, apenas após a realização de testes estatísticos é possível haver certo grau de confiança sobre a veracidade dessa conclusão. Para isso, tentou-se realizar testes paramétricos com os fatores sendo as abordagens comparadas e os tratamentos sendo as médias do RPD para todas as instâncias de cada um desses fatores. Após a realização da ANOVA testou-se seus resíduos de forma a tentar validar suas

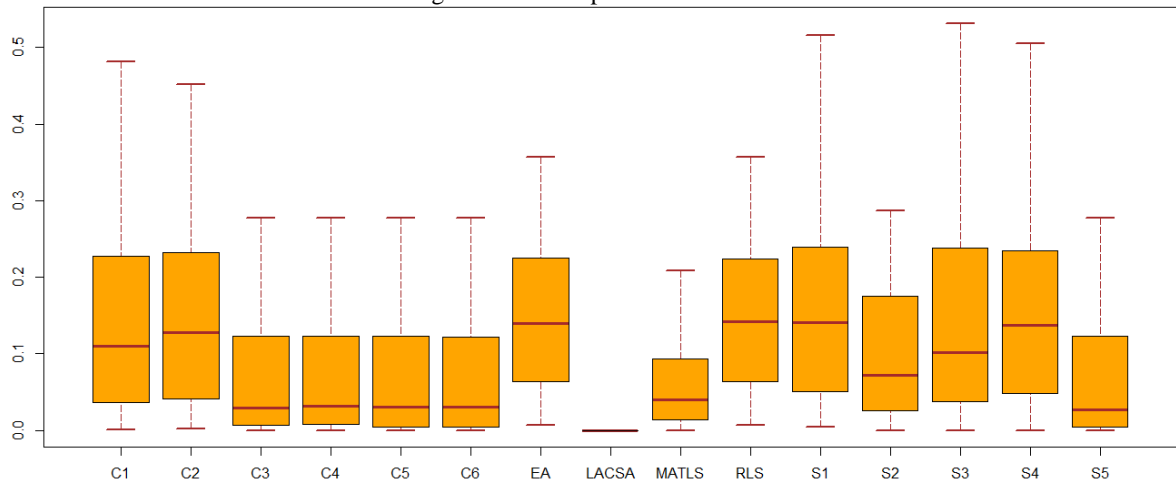
pressuposições, entretanto, as Pressuposições II e III foram violadas, indicando assim o uso de testes não paramétricos para trabalhar com esse conjunto de dados, tal como o teste *Kruskal – Wallis* em conjunto ao teste *Fischer LSD*. Os resultados desses testes podem ser vistos na Tabela 35, onde há uma separação direta na classificação entre o algoritmo LaCSA e os algoritmos C5, C6, S5 e MATLS, sendo estes as melhores abordagens encontradas na literatura para solucionar esse conjunto de instância do problema, mostrando uma diferença estatisticamente significativa entre todas as abordagens que não compartilham grupos, havendo uma ordenação decrescente em relação às abordagens que obtiveram os melhores resultados, i.e., menor valor estatístico (KW) do teste, ocupando assim o topo do *rank* do teste de menor significância mínima de Fischer (*Fischer LSD*). Como a abordagem LaCSA possui o menor valor de KW, média e desvio padrão, conclui-se que ela consegue entregar resultados significativamente melhores e de forma mais confiável. Na Figura 15 esses resultados são colocados visualmente em foco através do uso do diagrama de caixa e bigodes (do inglês: *Box Plot and Whisker*), revelando uma menor mediana (traço central) encontrada no algoritmo LaCSA em relação às outras abordagens e em uma grande garantia em se obter as melhores soluções entre elas: em conjunto, a mediana próxima do zero e a não visibilidade da caixa quando usado o RPD no *Box Plot* indicam que pelo menos 50% das soluções avaliadas pelo LaCSA são as melhores encontradas ou muito próximas delas.

**Tabela 34** – Resultados do teste *post hoc* de

ABORDAGEM	GRUPO	KW	MÉDIA	DESVIO PADRÃO	MIN	MAX
LACSA	a	119,6319	0,0110	0,0368	0,0000	0,2130
C5	b	423,8889	0,0697	0,0846	0,0000	0,2769
C6	b	429,0486	0,0703	0,0845	0,0000	0,2769
S5	b	431,2639	0,0703	0,0852	0,0000	0,2769
C3	b	442,8889	0,0719	0,0847	0,0000	0,2769
C4	b	447,1319	0,0721	0,0841	0,0000	0,2769
MATLS	b	451,6875	0,0617	0,0586	0,0000	0,2085
S2	c	583,8750	0,1112	0,1054	0,0000	0,5829
C1	cd	645,0556	0,1504	0,1499	0,0009	0,7192
C2	cd	665,8194	0,1601	0,1541	0,0015	0,7192
S3	cd	668,2014	0,1609	0,1544	0,0000	0,7192
S4	d	690,9514	0,1769	0,1653	0,0001	0,7213
S1	d	698,2153	0,1766	0,1650	0,0038	0,7213
EA	d	703,3681	0,1629	0,1409	0,0072	0,7192
RLS	d	706,4722	0,1640	0,1407	0,0072	0,7192

Fonte: Elaborada pelo autor

**Figura 15** – Diagrama de Caixa e Bigode (*Box Plot and Whiskers*) mostrando as médias dos RPD obtidos pelas abordagens existentes para solucionar o PMV



Fonte: elaborada pelo autor

Os resultados com seus valores reais são mostrados no apêndice A deste trabalho, onde é possível verificar quais foram os valores usados na análise estatística para cada abordagem e para cada instância analisada nesta seção.

## 7. CONCLUSÃO

Nesta dissertação foi abordado o Problema do Mochileiro Viajante, sendo este um problema multicomponente composto pela combinação do Problema do Caixeiro Viajante e do Problema da Mochila. O PMV foi proposto inicialmente com o intuito de servir de plataforma para o estudo de uma nova classe de problemas, conhecidos como multicomponente, permitindo que os pesquisadores da área de otimização combinatória pudessem realizar experimentos e verificar I) o quão efetiva são as abordagens mais usadas atualmente na área quando as mesmas forem aplicadas para solucionar problemas multicomponente e II) como a interdependência dos mesmos afeta na forma usada para procurar boas soluções em um espaço de busca grande e volátil, onde uma mudança em uma pequena parte da solução de um de seus componentes pode afetar significativamente a qualidade da solução completa em relação ao seu estado original.

Para solucionar o problema aqui discutido, foram usadas heurísticas capazes de solucionar os componentes do problema de forma individual, como a de Lin-Kernighan para geração de rotas curtas e a heurística TPPHx proposta neste trabalho para definir um plano de coleta para uma dada rota. Ambas heurísticas foram usadas pela meta-heurística *Simulated Annealing* que, por sua vez, foi adaptada à característica multicomponente do problema e teve seu nome dado como *Look and Construct Simulated Annealing*, sendo este nome devido a sua característica de procurar (*look*) vizinhos que são considerados promissores devido as suas rotas e, um após o outro, usar a heurística TPPHx para criar (*construct*) um plano de coleta para o vizinho escolhido no momento e verificar, sob a temperatura atual do algoritmo, qual será o comportamento do LaCSA em relação a ele: se ele for um vizinho superior a melhor solução encontrada ele será escolhido, caso contrário, haverá uma probabilidade em mesmo assim ser escolhido, sendo esta probabilidade amplamente baseada no quão próximo dessa melhor solução esse vizinho está. A cada iteração do algoritmo essa temperatura cai em razão a uma taxa pré-estabelecida, entretanto, se a mesma cai a um valor mínimo também definido pelo algoritmo, o mesmo reinicia sua busca elevando a temperatura novamente a seu valor máximo, repetindo todo o processo enquanto os 10 minutos de execução definidos como critério de parada não forem atingidos.

Os valores usados pelos parâmetros do algoritmo LaCSA foram encontrados após uma análise estatística dos resultados obtidos por experimentos computacionais realizados através de uma permutação entre os valores mais promissores. Para realização desses experimentos foram usadas as instâncias do problema disponibilizadas por autores de outros trabalhos. Esses autores (POLYAKOVSKIY et al, 2014) propuseram um conjunto vasto de instâncias, com as mais diversas características, e duas abordagens heurísticas, que foram executadas para solucionar as instâncias propostas por eles e disponibilizarem assim um *benchmark* do problema com propostas heurísticas que até

então não existiam, baseando-se somente no trabalho teórico onde o PMV foi proposto. Essas instâncias propostas por Polyakovskiy et al (2014) foram usadas também por outros trabalhos futuros a ele, onde as abordagens propostas nesses trabalhos mais recentes foram comparadas às melhores abordagens encontradas na literatura em seu próprio tempo.

A meta-heurística LaCSA foi proposta para solucionar instâncias que podem ser consideradas de pequeno e médio porte para o PMV, com um tamanho mínimo de 51 cidades e 150 itens, mas alcançando tamanhos com 280 cidades e 1395 itens. Existem instâncias muito maiores disponibilizadas pelos mesmos autores das instâncias estudadas nesta dissertação, chegando a existir cenários com mais de 85 mil cidades e 850 mil itens; entretanto, o autor desta dissertação acredita que uma mesma abordagem heurística não consegue ser eficiente para solucionar o problema quando fornecidas instâncias dos mais diversos tamanhos existentes na literatura, isto é, uma abordagem heurística capaz de resolver instâncias grandes do problema de forma satisfatória, dificilmente conseguirá obter soluções tão boas quanto as obtidas pelo LaCSA e outros algoritmos encontrados na literatura com bom desempenho para solucionar instâncias menores, isso porque uma abordagem que soluciona de forma satisfatória as instâncias grandes devem usar de estratégias e heurísticas mais simples, capazes de dar um “bom chute” sem ter que avaliar parte do gigantesco espaço de busca derivado da combinação de ambos problemas, enquanto as abordagens centradas em solucionar cenários menores podem se permitir a explorar o espaço de busca de forma mais inteligente, por vezes usando as características do problema em seu favor ao encontro de boas soluções.

As instâncias solucionadas pelo LaCSA tiveram seus resultados comparados com outras abordagens presentes na literatura, onde uma análise estatística baseada no RPD entre essas diversas abordagens classificou cada uma delas em grupos com valor estatístico igual, revelando assim quais conseguiam obter melhores resultados para as 72 instâncias selecionadas. A meta-heurística LaCSA revelou-se como a abordagem que obteve os melhores resultados, sendo estatisticamente superior às outras abordagens comparadas, principalmente se consideradas as instâncias de menores tamanhos, onde houve uma predominância da LaCSA em encontrar os melhores resultados. Para a maior parte das instâncias com mais de 200 cidades a LaCSA também foi capaz de obter as melhores soluções, entretanto, a diferença entre seu resultado e o resultado obtido por outras abordagens foi se encurtando, ocorrendo também mais cenários onde ela não foi capaz de encontrar a melhor solução. De forma geral, o algoritmo LaCSA conseguiu obter 55 dos 72 melhores resultados, perdendo, principalmente, para as outras abordagens quando avaliadas as instâncias maiores. Além da LaCSA, as abordagens que conseguiram obter os melhores resultados para alguns cenários foram as: C5 (6 de 72), S5 (4 de 72), MATLS (3 de 72), C6 (3 de 72) e S2 (1 de 72), relevando que mesmo entre

as melhores abordagens existentes na literatura não há entre elas uma dominância em solucionar o PMV tão grande quanto a apresentada pela LaCSA para as instâncias de tamanhos aqui explorados.

O Problema do Mochileiro Viajante se apresentou como sendo bastante desafiador, em grande parte devido a dois tipos diferentes de complexidades: uma derivada apenas do grande tamanho do espaço de busca resultante da combinação de dois problemas NP-Difícil e a outra derivada da forma como esses componentes foram combinados, isto é, a interdependência entre eles, que é capaz de permitir que uma leve alteração na solução de um de seus componentes possa afetar significativamente a qualidade da solução completa se o outro componente não for alterado em conjunto, dificultando assim um bom direcionamento da busca sem que haja um processamento mais elevado de possibilidades que devem ser cobridas.

### **7.1. Sugestões para Trabalhos Futuros**

Algumas ideias interessantes podem vir a serem trabalhadas no futuro, como, por exemplo:

- Proposta de novas heurísticas mais eficientes para solucionar o problema com foco nas instâncias de tamanho aproximado as usadas nesta dissertação.
- Proposta de novas abordagens para solucionar as maiores instâncias do problema (3.000 ~ 85.000 cidades e 9.000 ~ 850.000 itens), sendo necessário averiguar a aplicabilidade prática para justificar a realização de tal esforço.
- Novas heurísticas construtivas para o componente da mochila que sejam mais rápidas do que a TPPHx, porém, igualmente eficientes; possibilitando assim que a abordagem LaCSA consiga solucionar instâncias maiores do problema ou explorar uma maior parte do espaço de busca para as instâncias aqui trabalhadas. Diversas tentativas foram realizadas nesse sentido, entretanto, nenhuma delas se mostrou mais eficaz para ser aplicada ao problema.
- Proposta de uma variação do problema contendo mais de um mochileiro, isto é, transformando o componente PCV em um Problema de Roteamento de Veículos, mantendo o relacionamento entre o tempo para percorrer a rota e o peso dos itens coletados. Essa divisão em múltiplas rotas pode fazer com que estratégias que necessitem usar algoritmos gulosos para criação do componente da mochila, como a usada neste trabalho, se tornem mais rápidas, uma vez que a complexidade da mesma é exponencial e o número de itens possíveis de serem coletados por cada

veículo será diminuído em relação ao número de cidades que ele visitará: assintoticamente temos que  $(2^{n/v} \times v) < (2^n)$ , sendo  $n$  o número de itens e  $v$  o número de veículos.

- Estudo sobre a efetividade de certas estratégias quando usadas para solucionar mais de um problema multicomponente: espera-se que estratégias distintas, como a de “Olhar e Construir” usada pelo LaCSA, possam ser aplicadas em outros problemas mantendo suas características principais, havendo assim a possibilidade de comparar se alguma abordagem pode ser dominante a alguma(s) outra(s) quando aplicada a essa classe de problemas.

## REFERÊNCIAS BIBLIOGRÁFICAS

BARTZ-BEIELSTEIN, T.; LASARCZYK, C. W.; PREUB, M. Sequential parameter optimization. *In Evolutionary Computation, 2005*. The 2005 IEEE Congress on Vol. 1, 2005, p. 773-780.

BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, v. 35, n. 3, 2003, p. 268-308.

BONYADI, M. R.; MICHALEWICZ, Z.; BARONE, L. The Travelling Thief Problem: The First Step in the Transition From Theoretical Problems to Realistic Problems. *IEEE Congress on Evolutionary Computation (CEC)*, Cancún, 2013, p. 1037-1044.

BONYADI, M. R., MICHALEWICZ, Z., PRZYBYLEK, M. R., & WIERZBICKI, A. Socially inspired algorithms for the travelling thief problem. *In Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*. 2014, p. 421-428.

BONYADI, M. R., MICHALEWICZ, Z., NEUMANN, F., & WAGNER, M. Evolutionary computation for multicomponent problems: opportunities and future directions. *ArXiv preprint arXiv: 1606.06818*. 2016.

BONYADI, M. R., & MICHALEWICZ, Z. Evolutionary computation for real-world problems. *In Challenges in Computational Statistics and Data Mining*. Springer International Publishing, 2016, p. 1-24.

CANTERI, M. G., ALTHAUS, R. A., VIRGENS FILHO, J. D., GIGLIOTI, E. A., & GODOY, C. V. SASM-Agri: Sistema para análise e separação de médias em experimentos agrícolas pelos métodos Scott-Knott, Tukey e Duncan. *Revista Brasileira de Agrocomputação*, 1(2), 2001, p. 18-24.

CERNÝ, V. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of optimization theory and applications*, 45(1), 1985, p. 41-51.

CHIARANDINI, M.; PAQUETE, L.; PREUSS, M., & RIDGE, E. *Experiments on metaheuristics: Methodological overview and open issues*. 2007.

CLARKE, G. U; WRIGHT, JOHN W. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, v. 12, n. 4, 1964, p. 568-581.

- DANTZIG, G.; FULKERSON, R.; JOHNSON, S. Solution of a Large-Scale Traveling Salesman Problem. *Operations Research*. 1954, p. 393–410.
- DELAUNAY, B. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, vol. 7, 1934, p. 793–800.
- FAULKNER, H., POLYAKOVSKIY, S., SCHULTZ, T., & WAGNER, M. Approximate approaches to the traveling thief problem. *In Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, 2015, p. 385-392.
- FEIR-WALSH, B. J., TOOTHAKER, L. E. An empirical comparison of the ANOVA F-test, normal scores test and Kruskal-Wallis test under violation of assumptions. *Educational and Psychological Measurement*, 34(4), 1974, p. 789-799.
- FREDMAN, M. L.; JOHNSON, D. S.; MCGEOCH, L. A. Data structures for traveling salesmen. *Journal of Algorithms*, v. 18, n. 3, 1995, p. 432-479.
- GAREY, M. R.; JOHNSON, D. S. *A Guide to the Theory of NP-Completeness*. San Francisco: WH Freeman & Co. 1979.
- GELMAN, A.; HILL, J. Data Analysis Using Regression and Multilevel/Hierarchical Models. *Cambridge University Press*. 2006, p. 45–46.
- HANSEN, P., MLADENOVIC, N. First vs. best improvement: An empirical study. *Discrete Applied Mathematics*, 154(5), 2006, p. 802-817.
- KIRK, R. E. *Experimental design*. John Wiley & Sons, Inc. 1982.
- KIRKPATRICK, S., GELATT, C. D., & VECCHI, M. P. Optimization by Simulated Annealing. *Science*, 220(4598), 1983, p. 671-680.
- KRUSKAL, J. B. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, v. 7, n. 1, 1956, p. 48-50.
- LEVENE, H. "Robust tests for equality of variances". *In Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*. Stanford University Press. 1960, p. 278–292.
- LIN, S., & KERNIGHAN, B. W. An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, 21(2), 1973, p. 498-516.

LIN, S. Computer Solutions of the Traveling Salesman Problem. *Bell System Technical Journal*, v.44, n. 10, 1965, p. 2245-2269.

KRASNOGOR, N.; MOSCATO, P.; NORMAN, N. A new hybrid heuristic for large geometric traveling salesman problems based on the delaunay triangulation. *in Anales del XXVII Simposio Brasileiro de Pesquisa Operacional*. Citeseer, 1995, p. 6–8.

MARTELLO, S.; TOTH, P. *Knapsack problems*. New York: Wiley. 1990.

MEI, YI; LI, X.; YAO, X. Improving Efficiency of Heuristics for the Large Scale Traveling Thief Problem. *Simulated Evolution and Learning*. Springer International Publishing. 2014a, p. 631-643.

MEI, YI; LI, X.; YAO, XIN. On investigation of interdependence between sub-problems of the Travelling Thief Problem. *Soft Computing*, 2014b, p. 1-16.

MICHALEWICZ, Z. *Evolutionary Algorithms for Real-World Applications*. School of Computer Science University of Adelaide Adelaide, SA, 5005. 2015.

MIHAILA, C. *Evolutionary Computation in Scheduling*. 54 p. Tese de doutorado - Babes-Bolyai University. Cluj-Napoca. 2011.

NOURANI, Y.; ANDRESEN, B. A comparison of simulated annealing cooling strategies. *Journal of Physics A: Mathematical and General*, v. 31, n. 41, 1998, p. 8373.

OLIVEIRA, M. R. R., DOS SANTOS, A. G., ARAUJO, M. F. Uma Heurística Busca Tabu Para o Problema Do Mochileiro Viajante. Publicado em *XLVII Simpósio Brasileiro de Pesquisa Operacional (SBPO)*. 2015.

PAPADIMITRIOU, C. H. The Euclidean Travelling Salesman Problem is NP-Complete. *Theoretical Computer Science*, v. 4, n. 3, 1997, p. 237-244.

PEDRO, O. R. *Uma abordagem de Busca Tabu para o Problema do Caixeiro Viajante com Coleta de Prêmios*. Dissertação de mestrado - Universidade Federal de Minas Gerais. Belo Horizonte. 2013.

PISINGER, D. *Algorithms for Knapsack Problems*. Tese de doutorado - University of Copenhagen, Copenhagen. 1995.

POLYAKOVSKIY, S. ET AL. A Comprehensive Benchmark Set and Heuristics for the Travelling Thief Problem. *Genetic and Evolutionary Computation Conference (GECCO)*, Vancouver, 2014, p. 477-484.

POLYAKOVSKIY, S., & NEUMANN, F. Packing while traveling: Mixed integer programming for a class of nonlinear knapsack problems. *In International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Springer International Publishing, 2015, p. 332-346.

PRZYBYLEK, M. R., WIERZBICKI, A., & MICHALEWICZ, Z. Multi-hard Problems in Uncertain Environment. *In Proceedings of the 2016 on Genetic and Evolutionary Computation Conference*, 2016 p. 381-388.

PURCHASE, H. C. Experimental human-computer interaction: a practical guide with visual examples. *Cambridge University Press*, 2012.

R DEVELOPMENT CORE TEAM. R: A language and environment for statistical computing. Vienna, Austria: *R Foundation for Statistical Computing*, 2011.

RARDIN, R. L., & UZSOY, R. Experimental evaluation of heuristic optimization algorithms: A tutorial. *Journal of Heuristics*, 7(3), 2001, 261-304.

RECHENBERG, I. Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. frommann-holzboog, Stuttgart. Step-Size Adaptation Based on Non-Local Use of Selection Information. *In Parallel Problem Solving from Nature (PPSN3)*. 1973.

SHAPIRO, S. S.; WILK, M. B. An analysis of variance test for normality. *Biometrika*. doi:10.1093/biomet/52.3-4.591. JSTOR 2333709. MR 205384. 1965, p. 591-611.

SHEKIN, D. J. *Handbook of parametric and nonparametric statistical procedures*. crc Press, 2003.

SCOTT, A.J.; KNOTT, M. A Cluster analysis method for grouping means in the analysis of variance. *Biometrics*, Washington, v. 30, n. 3, 1974, p. 507-512.

TALBI, E. G. *Metaheuristics: from design to implementation*. New Jersey: John Wiley, 2009.

TUKEY, J.W. The problem of multiple comparisons. Mimeographs Princeton University, Princeton, N.J., 1953.

WEISE, T., ZAPF, M., CHIONG, R., & NEBRO, A. J. Why is optimization difficult? *In Nature-Inspired Algorithms for Optimisation*. Springer Berlin Heidelberg, 2009, p. 1-50.

ŽALIK, B. An efficient sweep-line Delaunay triangulation algorithm. *Computer Aided Design*, v. 37, n. 10, 2005, p. 1027-1038.

# APÊNDICE A

## A.1 Valores reais dos resultados

Nesta seção do apêndice são mostrados os valores reais obtidos pela função objetivo do problema para suas instâncias solucionadas pelas abordagens analisadas no capítulo 6.

### A.1.1 Valores das abordagens LaCSA, MATLS, S2, S5, C5 e C6

**Tabela 35** – Valores reais dos resultados para as menores instâncias solucionadas pelas abordagens que encontraram o melhor resultado para pelo menos uma instância: LaCSA, MATLS, S2, S5, C5 e C6

Instância	LaCSA	S2	S5	C5	C6	MATLS
eil51 n150 bsc 03	17838,60	15826,41	15831,01	15831,01	15854,52	15426,70
eil51 n150 bsc 07	25273,30	10541,46	21224,04	21225,76	21225,76	20813,10
eil51 n150 u 03	12252,80	9718,26	9718,26	9718,26	9723,82	11179,90
eil51 n150 u 07	13937,40	10078,54	10078,54	10078,54	10078,54	11038,30
eil51 n150 usw 03	11804,90	8914,58	10048,90	10048,90	10057,37	11129,20
eil51 n150 usw 07	14198,70	10119,67	10448,14	10448,93	10448,93	11632,90
eil51 n250 bsc 03	33987,20	30325,89	30324,82	30325,89	30328,50	30932,40
eil51 n250 bsc 07	43566,10	34355,34	34344,14	34355,34	34358,73	34482,50
eil51 n250 u 03	17772,50	13367,94	13877,17	13913,51	13915,83	15609,00
eil51 n250 u 07	23533,80	17922,87	17918,95	17922,87	17922,87	19251,10
eil51 n250 usw 03	14359,65	11186,24	11096,23	11186,24	11200,75	12208,10
eil51 n250 usw 07	24653,33	18726,17	18716,54	18726,17	18726,17	19966,20
eil76 n225 bsc 03	26896,70	22559,93	26552,81	26552,81	26577,91	25685,30
eil76 n225 bsc 07	38905,50	32627,98	37483,32	37483,46	37483,46	34371,10
eil76 n225 u 03	13369,20	10285,04	10249,06	10290,42	10290,42	10690,40
eil76 n225 u 07	22663,00	19134,69	19188,59	19189,38	19189,38	20041,00
eil76 n225 usw 03	9656,32	6995,51	7602,28	7602,28	7602,28	8921,40
eil76 n225 usw 07	17982,70	14067,35	14131,33	14218,24	14218,24	15899,40
eil76 n375 bsc 03	48408,00	41804,78	47439,49	47463,53	47474,72	46823,60
eil76 n375 bsc 07	65356,50	59232,31	65481,73	65481,72	65481,72	63336,90
eil76 n375 u 03	23311,40	19557,31	19481,53	19580,99	19580,99	20471,00
eil76 n375 u 07	37645,40	32509,71	32626,62	32626,62	32626,62	33751,30
eil76 n375 usw 03	17711,40	13301,30	14923,04	14923,04	14942,80	16015,80
eil76 n375 usw 07	31306,70	24715,25	24780,75	24791,09	24791,09	26752,40
kroB100 n297 bsc 03	37241,60	28722,63	38860,54	38860,54	38917,79	38122,30
kroB100 n297 bsc 07	60040,10	59965,64	59962,59	59965,64	59965,64	55931,60
kroB100 n297 u 03	21726,00	21357,79	21357,79	21357,79	21361,82	21609,90
kroB100 n297 u 07	28935,60	28365,75	28365,75	28365,75	28369,34	28564,10
kroB100 n297 usw 03	15834,10	15161,23	15161,23	15161,23	15176,23	15445,30
kroB100 n297 usw 07	23210,50	23054,53	23054,53	23054,53	23055,63	23022,90
kroB100 n495 bsc 03	64621,70	64667,62	64667,62	64667,62	64724,47	62637,00
kroB100 n495 bsc 07	94698,00	94164,48	94163,22	94164,48	94166,82	93849,20
kroB100 n495 u 03	37557,20	37880,73	37876,12	37880,73	37880,73	37206,40

kroB100 n495 u 07	49965,10	49775,05	49775,05	49775,05	49775,05	49518,70
kroB100 n495 usw 03	17961,60	21393,85	21570,67	21616,73	21616,73	21409,50
kroB100 n495 usw 07	39843,60	39548,63	39525,55	39548,63	39552,15	39771,30

**Tabela 36** - Valores reais dos resultados para as maiores instâncias solucionadas pelas abordagens que encontraram o melhor resultado para pelo menos uma instância: LaCSA, MATLS, S2, S5, C5 e C6

Instância	LaCSA	S2	S5	C5	C6	MATLS
rat195 n582-bsc-03	83807,80	85967,83	86498,75	86533,03	86546,93	83063,40
rat195 n582-bsc-07	115003,00	108811,23	110115,08	109759,36	109759,36	108308,00
rat195 n582 u 03	57951,80	56185,70	57006,43	56319,63	56319,63	55641,00
rat195 n582 u 07	75838,30	68831,13	71267,36	70229,34	69830,07	71183,70
rat195 n582-usw-03	31306,30	27642,57	27907,97	28049,05	28061,55	28945,10
rat195 n582-usw-07	53043,90	45364,31	47600,53	47633,75	47633,75	49691,40
rat195 n970-bsc-03	120084,00	120315,77	121034,08	121104,77	121055,95	116703,00
rat195 n970-bsc-07	196069,00	191402,69	194331,65	196556,92	194333,23	179298,00
rat195 n970 u 03	90915,50	86866,27	86777,81	86866,72	86869,02	85694,70
rat195 n970 u 07	127969,00	119803,54	121182,43	121319,79	122490,73	120693,00
rat195 n970-usw-03	50241,40	44808,14	45309,82	45869,81	45407,49	46079,60
rat195 n970-usw-07	92016,60	76355,08	82146,33	83536,88	82209,11	82999,30
ts225 n672 bsc 03	91173,12	88666,06	93803,23	93992,91	93809,58	93750,40
ts225 n672 bsc 07	125489,03	106003,08	128864,49	129370,13	128925,11	128462,00
ts225 n672-u-03	62185,00	59423,13	61958,63	61942,47	61977,32	61737,20
ts225 n672-u-07	81958,00	80040,02	80982,51	81105,74	81280,67	81687,80
ts225 n672-usw-03	37281,00	27030,83	36195,84	36091,02	35963,99	35769,00
ts225 n672-usw-07	71929,30	63368,37	69607,92	69317,95	69637,42	70100,00
ts225 n1120 bsc 03	154836,00	152572,26	156278,10	156452,23	156247,84	156784,00
ts225 n1120 bsc 07	265110,43	247338,21	254647,02	254906,16	254147,52	245277,00
ts225 n1120 u 03	99372,94	94296,66	95550,90	95119,22	94595,92	93514,90
ts225 n1120 u 07	137241,20	128562,66	134001,11	134217,25	133849,31	135404,00
ts225 n1120 usw 03	66140,49	62590,81	65175,31	65340,56	65145,68	67490,30
ts225 n1120 usw 07	121952,00	107880,52	117304,75	117886,60	117626,39	118400,00
a280 n837 bsc _03	110552,20	112228,67	112940,88	112850,80	112784,79	110250,00
a280 n837 bsc _07	134344,00	156318,40	159087,46	158861,82	158835,70	157320,00
a280 n837 u 03	82127,90	76842,39	81810,38	81835,63	81919,97	80430,20
a280 n837 u 07	104461,58	101947,89	103295,32	103312,47	103253,97	104139,00
a280 n837 usw 03	46629,60	44353,39	47703,07	47686,84	47625,23	46654,60
a280 n837 usw 07	84730,20	80612,15	83261,11	83328,40	83163,26	82200,20
a280 n1395 _bsc 03	176911,00	172905,51	181188,11	181284,68	181016,69	173654,00
a280 n1395 _bsc 07	203888,00	251033,05	256357,63	255947,96	255753,51	259057,00
a280 n1395 u 03	125556,28	122485,80	124640,52	124717,41	124717,41	123310,00
a280 n1395 u 07	175557,51	170970,13	173662,28	173816,56	173512,26	173509,00
a280 n1395-usw-03	81497,89	76351,87	80733,16	80422,84	80969,63	80361,10
a280 n1395-usw-07	134880,00	124631,64	132853,60	133020,53	132838,62	132384,00

### A.1.2 Valores das abordagens S1, S3, S4, C1, C2, C3, C4, RLS e EA

**Tabela 37** - Valores reais dos resultados para as menores instâncias solucionadas pelas abordagens que não conseguiram encontrar o melhor resultado para qualquer instância: S1, S3, S4, C1, C2, C3, C4, RLS e EA

Instância	S1	S3	S4	C1	C2	C3	C4	RLS	EA
eil51-n150-bsc 03	15831,01	15849,78	15831,01	12263,83	12263,83	15831,01	15831,01	15036,85	15786,47
eil51-n150-bsc 07	21185,82	21188,24	21224,04	21224,04	21225,76	21224,04	21224,04	21153,60	21188,24
eil51-n150-u 03	9498,55	9723,82	9718,26	9718,26	9723,82	9718,26	9718,26	9547,93	9496,27
eil51-n150-u 07	10040,39	10072,94	10078,54	10074,21	10078,54	10078,54	10078,54	10038,80	10038,80
eil51-n150-usw 03	8914,58	10008,99	10048,90	10048,90	10057,37	10048,90	10048,90	9879,15	9966,09
eil51-n150-usw 07	10448,14	10149,69	10448,14	10448,93	10448,93	10448,93	10448,93	10397,29	10397,29
eil51-n250-bsc 03	26023,80	26050,36	26023,80	26023,80	30325,89	30320,69	30325,89	29998,05	30181,38
eil51-n250-bsc 07	27336,83	27340,12	27336,83	27340,12	34355,34	34355,34	34355,34	34229,99	34190,89
eil51-n250-u 03	13848,95	13915,83	13358,61	13913,51	13913,51	13913,51	13915,83	13887,23	13885,29
eil51-n250-u 07	17918,95	17372,36	17372,36	17922,87	17372,36	17922,87	17922,87	17870,70	17870,70
eil51-n250-usw 03	8974,72	9137,96	8974,72	11186,24	9150,32	11186,24	11197,83	11093,15	11140,43
eil51-n250-usw 07	18716,54	18683,48	18716,54	18726,17	18726,17	18726,17	18726,17	18683,48	18683,48
eil76-n225-bsc 03	22604,37	20654,90	22524,72	26552,81	22650,65	26552,81	26552,81	22560,93	22538,26
eil76-n225-bsc 07	37405,43	32638,91	32789,06	32790,02	32636,24	37416,23	37416,23	32584,11	32610,91
eil76-n225-u 03	9370,45	9394,07	10249,06	10049,84	10290,42	10290,42	10290,42	10285,04	10285,04
eil76-n225-u 07	16717,31	14821,14	19137,25	19138,16	16731,99	19189,38	19189,38	19134,69	19134,12
eil76-n225-usw 03	4270,62	7027,59	4276,41	7002,55	7046,87	7374,43	7393,87	6971,48	6957,75
eil76-n225-usw 07	7928,98	10227,40	7962,34	14146,76	7976,39	14218,24	14218,24	14067,35	14067,35
eil76-n375-bsc 03	43420,11	43448,54	43420,11	41979,87	39696,34	47400,18	47463,51	41568,19	41727,77
eil76-n375-bsc 07	50098,53	59236,29	50153,92	50153,92	50084,38	65481,72	65481,72	59171,53	59137,19
eil76-n375-u 03	19475,70	18272,02	19476,96	19297,95	15533,87	19580,99	19580,99	19553,73	19553,73
eil76-n375-u 07	30018,78	32623,15	32514,58	22526,61	22526,61	32626,62	32626,62	32509,71	32509,71
eil76-n375-usw 03	9640,20	9716,10	9662,15	10503,15	9709,42	14806,47	14923,04	13328,77	13292,86
eil76-n375-usw 07	24699,20	14683,60	16307,09	14720,93	24715,27	24791,09	24791,09	24721,24	24721,24
kroB100 n297 bsc 03	28705,28	28727,82	28705,28	38860,54	38860,54	38809,06	38828,38	28665,68	28698,49
kroB100 n297 bsc 07	38616,64	38624,94	59962,59	59965,64	40387,39	59920,29	59965,64	38622,66	38588,56
kroB100 n297 u 03	18662,70	21361,82	21357,79	21357,79	18662,99	21357,79	21335,13	18662,99	18662,99
kroB100 n297 u 07	21726,12	28369,34	22169,70	28365,75	23106,66	28365,75	28365,75	21733,80	21733,57
kroB100 n297 usw 03	5139,15	5216,62	5590,31	6185,18	6193,38	15161,23	15170,26	5167,82	5207,40
kroB100 n297 usw 07	10594,34	23055,63	10760,54	23054,53	10780,65	23054,53	23055,63	10610,53	10610,53
kroB100 n495 bsc 03	50075,37	50145,24	50075,37	64667,62	50141,77	64635,43	64667,62	49587,02	50030,59
kroB100 n495 bsc 07	94163,22	63864,36	65403,69	94164,48	94166,82	94164,48	94164,66	63767,19	63864,36
kroB100 n495 u 03	31766,83	37880,73	37876,12	31914,75	31914,75	37880,73	37880,73	31764,77	31775,61
kroB100 n495 u 07	49775,05	36456,39	49775,05	36700,32	49775,05	49775,05	49775,05	36456,39	36456,39
kroB100 n495 usw 03	6024,08	6069,40	6024,08	6069,40	6069,40	21393,85	21383,68	6069,40	6069,40
kroB100 n495 usw 07	19312,04	39552,15	19729,68	20677,27	39551,47	39548,63	39550,09	19374,86	19374,86

**Tabela 38** - Valores reais dos resultados para as maiores instâncias solucionadas pelas abordagens que não conseguiram encontrar o melhor resultado para qualquer instância: S1, S3, S4, C1, C2, C3, C4, RLS e EA

Instância	S1	S3	S4	C1	C2	C3	C4	RLS	EA
rat195 n582 bsc 03	85315,83	85347,42	86498,75	71122,29	85967,83	86498,75	86498,75	71320,26	71228,97

rat195 n582 bsc 07	108009,18	92662,79	108053,40	93766,60	108811,99	109759,36	108811,99	93877,20	93867,17
rat195 n582 u 03	54543,32	55330,91	55538,94	56319,18	54038,95	56319,63	56319,63	54805,70	54820,60
rat195 n582 u 07	69336,85	69830,07	69222,54	69296,99	68831,13	70229,34	69828,73	69283,24	69283,24
rat195 n582 usw 03	27010,20	26547,17	27120,28	27106,52	27201,97	28046,25	28042,33	26977,77	26982,89
rat195 n582 usw 07	47572,67	44415,02	44280,62	47633,75	45360,60	47633,75	47633,75	47540,52	47540,52
rat195 n970 bsc 03	120246,27	97662,00	100699,16	97367,81	120315,77	121017,36	121031,68	97551,99	97496,20
rat195 n970 bsc 07	161888,59	191404,21	169495,77	193899,96	191402,69	194331,88	194460,02	162301,71	162306,40
rat195 n970 u 03	86777,36	83172,51	81441,87	83219,89	83172,51	86866,72	86869,02	83245,12	83253,98
rat195 n970 u 07	119700,09	117676,50	119720,83	119822,78	121319,50	121319,79	121319,79	117788,05	117787,54
rat195 n970 usw 03	45154,06	45287,47	43782,66	44387,02	43624,91	45390,69	45861,34	45366,27	45363,31
rat195 n970 usw 07	76206,56	76356,68	76206,56	82189,16	76356,10	82189,16	82189,16	82209,11	82209,11
ts225 n672 bsc 03	91988,53	91171,88	89261,59	92064,67	91760,92	92945,21	92608,71	90118,66	91460,69
ts225 n672 bsc 07	111146,20	120524,71	111926,45	126451,11	113428,06	128848,14	128897,59	120960,82	120965,80
ts225 n672 u 03	59998,28	60330,28	61750,59	58289,12	60428,41	61836,60	61653,10	60495,37	60501,66
ts225 n672 u 07	79087,98	73606,30	78478,11	76712,11	74121,50	80462,76	80617,58	79273,35	79273,35
ts225 n672 usw 03	29491,51	33639,58	28723,33	27295,81	27651,97	34217,80	35599,60	31958,58	31925,08
ts225 n672 usw 07	64779,69	64754,49	63389,91	66032,67	67338,16	69739,42	68187,39	66099,73	66099,73
ts225 n1120 bsc 03	150334,90	152449,97	151348,20	148666,45	151023,17	156117,74	156254,05	154846,37	154840,27
ts225 n1120 bsc 07	227618,89	250747,55	225499,44	227977,80	254038,81	254913,88	254530,85	242333,67	242344,53
ts225 n1120 u 03	92547,15	93119,75	90680,04	94378,60	92724,28	95160,58	95084,28	93145,87	93143,61
ts225 n1120 u 07	128502,04	127809,07	126559,56	131102,09	125316,14	134033,20	133632,99	131071,73	131071,73
ts225 n1120 usw 03	56669,48	56379,78	52987,35	64106,63	62387,76	64893,84	65180,07	61660,91	61638,52
ts225 n1120 usw 07	104445,86	110333,21	109746,62	108163,02	114772,47	118045,34	117358,21	112704,82	112704,82
a280 n837 bsc 03	105726,52	112437,99	104190,79	112114,07	105342,68	112853,12	112849,09	105433,06	105412,69
a280 n837 bsc 07	156447,11	153216,87	152910,24	153271,25	152761,50	158953,27	158880,54	154965,91	154974,13
a280 n837 u 03	81161,76	80843,10	81259,29	80829,15	80368,77	81710,29	81807,90	81478,41	81478,41
a280 n837 u 07	101897,17	102273,74	102820,79	101381,13	102288,18	103322,63	102864,79	102395,23	102395,23
a280 n837 usw 03	45180,43	45510,37	44397,52	44931,94	46406,79	47656,30	47221,27	46152,05	46152,05
a280 n837 usw 07	80546,38	82571,31	81274,22	81735,91	81620,66	83286,28	83305,34	80866,86	80866,86
a280 n1395 bsc 03	172904,85	172836,40	173030,27	172156,10	170911,13	181249,77	181211,48	173093,71	173004,09
a280 n1395 bsc 07	253876,49	252768,86	253737,13	244499,75	255734,88	255734,95	255602,36	254947,58	254959,80
a280 n1395 u 03	123499,77	122297,52	122427,40	123659,05	123078,84	124526,25	124559,57	124651,16	124656,45
a280 n1395 u 07	170391,63	172898,52	169648,94	170760,76	171579,82	173790,96	173279,21	170997,41	170997,41
a280 n1395 usw 03	76823,36	77684,29	78344,18	76851,84	77727,07	81017,34	80303,28	77088,22	77088,46
a280 n1395 usw 07	126901,16	129577,85	128112,11	126650,74	125137,44	132846,75	132575,46	126770,21	126770,21

## **A.2 Divulgações dos resultados para a comunidade acadêmica**

OLIVEIRA, M. R. R., DOS SANTOS, A. G., ARAUJO, M. F. Uma Heurística Busca Tabu Para o Problema Do Mochileiro Viajante. Publicado em *XLVII Simpósio Brasileiro de Pesquisa Operacional (SBPO)*. 2015, p. 1-12.

Está sendo escrito mais um artigo contendo os resultados descritos nessa dissertação para ser submetido a alguma revista ou congresso da área de otimização combinatória.