

LEONARDO MOREIRA FAÊDA

ABORDAGENS HEURÍSTICAS APLICADAS AO THIEF  
ORIENTEERING PROBLEM

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*.

Orientador: André Gustavo dos Santos

VIÇOSA - MINAS GERAIS  
2022

**Ficha catalográfica elaborada pela Biblioteca Central da Universidade  
Federal de Viçosa - Campus Viçosa**

T

F147a Faêda, Leonardo Moreira, 1995-  
2022 Abordagens heurísticas aplicadas ao Thief Orienteering  
Problem / Leonardo Moreira Faêda. – Viçosa, MG, 2022.  
1 dissertação eletrônica (62 f.): il.

Texto em português e inglês.

Orientador: André Gustavo dos Santos.

Dissertação (mestrado) - Universidade Federal de Viçosa,  
Departamento de Informática, 2022.

Referências bibliográficas: f. 60-62.

DOI: <https://doi.org/10.47328/ufvbbt.2023.163>

Modo de acesso: World Wide Web.

1. Algoritmos genéticos. 2. Multiprogramação  
(Computadores). 3. Programação heurística. I. Santos, André  
Gustavo dos, 1974-. II. Universidade Federal de Viçosa.  
Departamento de Informática. Programa de Pós-Graduação em  
Ciência da Computação. III. Título.

CDD 23. ed. 006.31

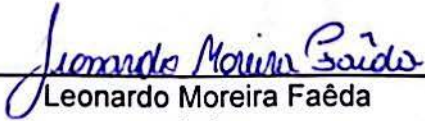
LEONARDO MOREIRA FAÊDA

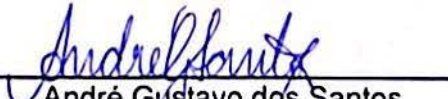
**ABORDAGENS HEURÍSTICAS APLICADAS AO THIEF  
ORIENTEERING PROBLEM**

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: 15 de julho de 2022.

Assentimento:

  
\_\_\_\_\_  
Leonardo Moreira Faêda  
Autor

  
\_\_\_\_\_  
André Gustavo dos Santos  
Orientador

# Agradecimentos

Agradeço a Deus por ter me dado saúde, força e fé, principalmente quando tudo parecia impossível.

Aos meus pais, Rafael Antônio e Leila Rosana, e irmão, Felipe e Ana Clara, pelo amor e incentivo em todas as fases da minha vida, sempre dispostos a me ajudar em todas as minhas idas e vindas, me colocando em suas orações e por toda compreensão da minha ausência.

Ao meu orientador, André Gustavo dos Santos, por apoiar minha ideia, pela paciência, disposição e incentivo.

À Universidade Federal de Viçosa pela estrutura oferecida.

À Capes pelo apoio financeiro.

Aos meus amigos por todo incentivo e por acreditarem em mim.

Às pessoas que cruzaram meu caminho, que me apoiaram de alguma forma e torceram por mim.

A todos os meus sinceros agradecimentos.

# Resumo

FAËDA, Leonardo Moreira, M.Sc., Universidade Federal de Viçosa, julho de 2022. **Abordagens Heurísticas Aplicadas ao Thief Orienteering Problem**. Orientador: André Gustavo dos Santos.

Este trabalho aborda o *Thief Orienteering Problem*, um problema multicomponente que combina dois problemas combinatórios conhecidos: o problema da mochila e o problema de orientação. Nesse problema, uma pessoa (denominada de ladrão) carrega uma mochila capacitada e possui um limite de tempo para coletar itens distribuídos em um conjunto de pontos. Os pontos de partida e chegada são fixos. O ladrão inicia sua trajetória com a mochila vazia e viaja com a velocidade inversamente proporcional ao peso da mochila. Enquanto tiver tempo, o ladrão pode passar pelos pontos coletando novos itens. O objetivo do problema é determinar qual rota e quais itens o ladrão deve coletar para maximizar o lucro da mochila. Dois métodos heurísticos baseados em metaheurísticas conhecidas são propostos para encontrar soluções boas que maximize o lucro obtido pelo ladrão em tempo hábil. O primeiro método heurístico é inspirado em algoritmos genéticos e os experimentos computacionais foram realizados com as instâncias de *benchmark*, a fim de comparar o desempenho do método desenvolvido com os algoritmos existentes na literatura. Os resultados mostraram que nosso método foi superior na maioria dos casos. Posteriormente um novo método baseado na metaheurística de busca local iterada foi proposto e novos experimentos computacionais foram realizados. Os resultados mostraram que a abordagem superou os trabalhos existentes e o método inspirado em algoritmo genéticos em mais de 80% das instâncias do *benchmark*, com uma melhoria média de mais de 30%. Propomos também uma ampliação do problema para utilizar múltiplos ladrões. Nós descrevemos formalmente o problema apresentando uma formulação de programação não linear inteira mista e utilizamos adaptações dos métodos heurísticos propostos para gerar soluções iniciais para o problema. Os resultados mostram que, com a utilização de mais ladrões, o lucro total dos itens coletados aumentou em até 32%

Palavras-chave: Metaheurísticas. Problemas Multicomponentes. Thief Orienteering Problem

# Abstract

FAËDA, Leonardo Moreira, M.Sc., Universidade Federal de Viçosa, July 2022. **Heuristic Approaches Applied to the Thief Orienteering Problem.** Advisor: André Gustavo dos Santos.

This work approaches the Thief Orienteering Problem, a multicomponent problem that combines two known combinatorial problems: the knapsack problem and the orientation problem. In this problem, a person (called a thief) carries a capacitated backpack and has a time limit to collect items distributed in a set of points. Departure and arrival points are fixed. The thief starts his trajectory with an empty backpack and travels with a speed inversely proportional to the weight of the backpack. As long as he has time, the thief can go through the points collecting new items. The purpose of the problem is to determine which route and which items the thief should collect to maximize the backpack's profit. Two heuristic methods based on known metaheuristics are proposed to find good solutions that maximize the profit obtained by the thief in a timely manner. The first heuristic method is inspired by genetic algorithms and computational experiments were carried out with the benchmark instances, in order to compare the performance of the developed method with existing algorithms in the literature. The results showed that our method was superior in most cases. Subsequently, a new method based on the iterated local search metaheuristic was proposed and new computational experiments were performed. The results showed that the approach outperformed existing work and the method inspired by genetic algorithms in over 80% of the benchmark instances, with an average improvement of over 30%. We also propose an expansion of the problem to use multiple thieves. We formally describe the problem by presenting a mixed integer nonlinear programming formulation and adapt the proposed methods to generate initial solutions to the problem. The results show that with the use of more thieves, the total profit from collected items increased by up to 32%.

Keywords: Metaheurísticas. Multi-component problems. Thief Orienteering Problem

# Lista de ilustrações

Figura 1 – ThOP example. . . . .	16
Figura 2 – Flowchart of the thopGA algorithm. . . . .	18
Figura 3 – Quality of convergence per number of items per city. . . . .	25
Figura 4 – Quality of convergence per weight-profit relation type. . . . .	25
Figura 5 – Quality of convergence per maximum travel time class. . . . .	25
Figura 6 – Quality of convergence per knapsack capacity class. . . . .	26
Figura 7 – Comparative analysis of the convergence of BRKGA algorithm and thopGA for all instances. . . . .	28
Figura 8 – Exemplo para o <i>Thief Orienteering Problem</i> . . . . .	32
Figura 9 – Qualidade da convergência dos algoritmos heurísticos agrupados por relações das instâncias. . . . .	41
Figura 10 – Análise comparativa da convergência dos algoritmos heurísticos para todas as instâncias . . . . .	42
Figura 11 – Exemplo exemplo de instância do ThOP para solucionar o MThOP. Fonte: (FAËDA; SANTOS, 2020b) . . . . .	49
Figura 12 – Proximidade dos resultados dos métodos heurísticos com o ótimo global do problema da mochila. . . . .	54
Figura 13 – Proximidade dos resultados dos métodos heurísticos com o ótimo global do problema da mochila, para as instâncias com tempo de conclusão reduzido para 25%. . . . .	55
Figura 14 – Análise comparativa da convergência dos algoritmos heurísticos para todas as instâncias . . . . .	56

# Lista de tabelas

Tabela 1 – Quality of convergence grouped by number of items per city. . . . .	24
Tabela 2 – Quality of convergence grouped by item relation type. . . . .	24
Tabela 3 – Quality of convergence grouped by maximum travel time class. . . . .	24
Tabela 4 – Quality of convergence grouped by knapsack capacity class. . . . .	25
Tabela 5 – Number of best results found for each group . . . . .	27
Tabela 6 – Number of best average results for each group . . . . .	27
Tabela 7 – Número de médios e melhores resultados encontrados para cada grupo	40

# Sumário

<b>1</b>	<b>Introdução</b>	<b>9</b>
1.1	Problema e sua importância	10
1.2	Hipótese	11
1.3	Objetivo Geral	11
1.3.1	Objetivos Específicos	11
1.4	Estrutura da Dissertação	12
<b>2</b>	<b>A Genetic Algorithm for the Thief Orienteering Problem</b>	<b>14</b>
2.1	Introduction	15
2.2	Literature Review	17
2.3	Heuristic Approach	18
2.3.1	Solution representation and evaluation procedure	19
2.3.2	Local Search	19
2.3.3	Genetic Algorithm	21
2.4	Computational Experiments	23
2.5	Conclusions and Further Investigations	28
<b>3</b>	<b>Busca Local Iterada Aplicada ao Thief Orienteering Problem</b>	<b>29</b>
3.1	Introdução	30
3.2	Definição Formal e Modelo Matemática	33
3.3	Proposta Heurística	34
3.3.1	Representação da Solução e Procedimento de Avaliação	35
3.3.2	Busca Local	35
3.3.3	Busca Local Iterada	37
3.4	Experimentos Computacionais	38
3.5	Conclusão e Trabalhos Futuros	42
<b>4</b>	<b>Multiple Thief Orienteering Problem: Formulação e Abordagens Heurísticas</b>	<b>43</b>
4.1	Introdução	44
4.2	Descrição do Problema	46
4.3	Definição Formal e Formulação Matemática	49
4.4	Abordagens Heurísticas	51
4.5	Análise dos Resultados	52
4.6	Conclusões	56
<b>5</b>	<b>Conclusões Gerais e Trabalhos Futuros</b>	<b>58</b>
5.1	Trabalhos Futuros	58
	<b>Referências</b>	<b>60</b>

# 1 Introdução

Nos últimos cinquenta anos, cientistas de diferentes áreas têm estudado diversos problemas de otimização. Os problemas de estudos são pertencentes à classe dos problemas NP-difíceis, que são caracterizados de forma que não é possível otimizar grandes instâncias (GAREY; JOHNSON, 1979). Essa classe inclui problemas clássicos na área de otimização, entre eles estão: problema do caixeiro viajante, problema da mochila, problema de roteamento de veículos, problema de coloração de grafos, problema de sequenciamento de tarefas entre muitos outros.

Para solucionar estes problemas, muitos métodos exatos foram desenvolvidos, como: *branch-and-bound*, programação dinâmica e de memorização. Apesar desses métodos terem obtido bons resultados, dois impasses cruciais foram detectados. O primeiro é que nenhum desses métodos eram eficientes em relação ao tempo de execução para instâncias grandes e a outra é que para cada problema diferente é necessária uma adaptação do método utilizado.

Como os problemas da classe NP-difíceis possuem uma grande relevância para os problemas do mundo real, foram realizadas muitas iniciativas para solucionar estes dois impasses. Muitos métodos heurísticos foram propostos. Com eles, soluções quase ótimas ou até ótimas foram encontradas de maneiras muito mais rápidas que os métodos exatos. Por exemplo, o método proposto por Lin e Kernighan (1973) para o problema do caixeiro viajante ou o método desenvolvido por Pirkul (1987) para o problema da mochila. Mesmo obtendo sucesso para o primeiro impasse em relação ao tempo de execução dos métodos, eles ainda sofriam com o segundo, sendo muito específicos para cada problema abordado.

Havia ainda a necessidade de uma nova classe de métodos mais genéricos, que com apenas pequenas modificações se adaptasse ao problema abordado. Então surgiu a classe de métodos metaheurísticos (ROTHLAUF, 2011). Muitos desses métodos são inspirados pela natureza, entre os exemplos estão a *simulated annealing*, busca tabu, colônia de formigas e algoritmos evolutivos. À proporção que novas metaheurísticas foram sendo propostas, a necessidade de compará-las ficou cada vez relevante. Em vista disso, para avaliar o desempenho de várias metaheurísticas aplicadas a um determinado problema NP-difícil, foram criadas bases de instâncias padrões (por exemplo, (REINELT, 1991) e (BEASLEY, 1990)). Com essas comparações foi possível mostrar que muitos métodos metaheurísticos conseguem alcançar soluções muito próximas de ótima ou até mesmo ótima para problemas NP-difíceis grandes, de maneira rápida.

Entretanto, alguns cientistas concordam que existe uma grande distância entre os problemas que são estudados e o que realmente são problemas do mundo real (MICHA-

LEWICZ, 2012; BONYADI; MICHALEWICZ; BARONE, 2013). O principal motivo é que há muito tempo cientistas propõem heurísticas eficientes para bases de instâncias conhecidas, porém esses problemas não relatam algumas características de problemas da realidade. Assim, métodos metaheurísticos que apresentam bons resultados para esses banco de instâncias não necessariamente vão possuir bons desempenhos para os problemas que refletem a realidade.

Para Bonyadi, Michalewicz e Barone (2013), a principal diferença entre os problemas que utilizam as bases de instâncias conhecidas para os problemas do mundo real está na definição da complexidade. Duas principais características retiradas de problemas do mundo real são a combinação, que geralmente os problemas são compostos por dois ou mais subproblemas, e a interdependência, no sentido que uma solução para um subproblema influencia na qualidade das soluções dos outros subproblemas. Estes tipos de problemas foram denominados de problemas multicomponentes. Para ser um problema multicomponente ele deve ter presente na sua formulação as características de combinação e interdependência.

Os problemas multicomponentes abordados na literatura pertencem à classe NP-difícil. Portanto, é necessário um esforço computacional muito grande para encontrar soluções ótimas para um problema, podendo levar muito tempo para serem resolvidos dependendo do tamanho e das restrições do problema. Encontrar uma solução ótima torna-se inviável devido sua complexidade, então é necessária a utilização de algoritmos eficientes que encontrem uma boa solução para o problema em tempo razoável e tolerável. Esses algoritmos são classificados como métodos heurísticos e metaheurísticas, que determinam soluções aproximadas de alta qualidade em tempo computacional relativamente baixo.

## 1.1 Problema e sua importância

Este trabalho aborda o *Thief Orienteering Problem* (ThOP), um problema multicomponente com interdependência que combina dois problemas combinatórios clássicos: o problema da mochila e o problema de orientação. Neste problema, um indivíduo (denominado de ladrão) possui uma mochila capacitada e possui um tempo limite para percorrer um conjunto de locais coletando os itens presentes nelas. O ladrão tem o local de partida e chegada previamente fixados, e enquanto tiver tempo e capacidade em sua mochila, ele vai percorrendo os locais coletando os itens, e cada item tem seu peso e lucro. No entanto, à medida que os itens são coletados, a mochila fica mais pesada e o ladrão caminha mais devagar. O objetivo do problema é determinar uma rota e os itens a serem coletados, para maximizar o lucro da mochila, respeitando as restrições de capacidade da mochila e tempo total para a realização do percurso. O problema foi proposto recentemente por Santos e Chagas (2018).

Uma aplicação imediata para esse problema é na logística. Supondo que uma companhia possua veículos com capacidade limitada e o tempo de trabalho de seus funcionários não deve ultrapassar um tempo limite por dia. A companhia utiliza os caminhões para recolher mercadorias de descarte, para realizar o processo de descarte adequado. Para cada produto, um benefício está associado. Além disso, a velocidade do caminhão diminui à medida que os itens são coletados, assim o tempo de deslocamento entre duas localidades de coleta aumenta. Portanto, o objetivo é encontrar uma rota e os itens que devem ser coletados, para maximizar o lucro em cada caminhão da companhia.

Utilizar os recursos disponíveis de maneira eficiente e o tempo total disponível de maneira correta proporciona às empresas de manufatura alcançar seus dois principais objetivos: reduzir os custos e aumentar a lucratividade. Portanto, o problema que se propõe resolver neste trabalho é encontrar boas soluções que utilizam o tempo gasto para a realização da coleta e a capacidade total de transporte de maneira eficientemente. A programação da coleta dos itens utilizando técnicas de otimização visa oferecer uma resposta rápida e de qualidade para esse problema.

## 1.2 Hipótese

A nossa hipótese é que utilizando métodos heurísticos inspirados em metaheurísticas conhecidas para resolver o problema do *Thief Orienteering Problem* podemos melhorar a qualidade das soluções e eficiência em tempo computacional analisando um conjunto de instâncias.

## 1.3 Objetivo Geral

Esta dissertação tem como objetivo propor métodos heurísticos inspirados nas principais metaheurísticas existentes, para solucionar o *Thief Orienteering Problem*, com o intuito de encontrar um conjunto de itens e pontos que maximize o lucro obtido pelo ladrão e analisar os seus desempenhos para gerar soluções boas em tempo hábil. Para isso são propostos dois algoritmos heurísticos, um inspirado em algoritmos genéticos e outro na metaheurística de busca local iterada.

### 1.3.1 Objetivos Específicos

1. Realizar experimentos computacionais com as heurísticas desenvolvidas para o ajustamento dos parâmetros utilizados.
2. Aplicar técnicas da estatística experimental para verificar se os algoritmos desenvolvidos possuem diferenças significantes em relação aos algoritmos já existentes.

3. Propor uma ampliação do problema utilizando vários ladrões, descrevendo formalmente e uma formulação matemática. Utilizar os algoritmos propostos para solucionar a ampliação do problema.

## 1.4 Estrutura da Dissertação

Esta dissertação foi elaborada em formato de coletânea de artigos que foram produzidos durante a pesquisa. Este primeiro capítulo apresenta uma introdução geral do problema discutido nessa dissertação e apresenta a hipótese e os objetivos gerais e específicos da pesquisa. Os capítulos a seguir apresentam três artigos, que representam todo trabalho desenvolvido nessa dissertação.

O Capítulo 2 apresenta o artigo "*A Genetic Algorithm for the Thief Orienteering Problem*", publicado nos anais do IEEE Congress on Evolutionary Computation (CEC 2020). O artigo propõe um método heurístico inspirado em algoritmos genéticos e denominado de thopGA. Testamos o thopGA para as 432 instâncias disponíveis na literatura e comparamos com os algoritmos heurísticos já existentes até o momento da publicação. Os resultados mostraram uma superioridade do thopGA quando comparado com os demais algoritmos, principalmente na instâncias grandes. Avaliamos que essa superioridade é devido a combinação de indivíduos que possuem boas características, obtidas pela busca local utilizada na geração da população inicial e após os novos indivíduos gerados, assim gerando outros com melhores resultados.

O segundo artigo, "*Busca Local Iterada Aplicada ao Thief Orienteering Problem*", publicado nos anais do LII Simpósio Brasileiro de Pesquisa Operacional (SBPO 2020), é apresentado no Capítulo 3. O artigo apresenta um método heurístico baseado na metaheurística de busca local iterada, denominado de thopILS. Realizamos os experimentos computacionais comparando o desempenho do thopILS com os algoritmos existentes na literatura. Os resultados mostraram que a abordagem superou os trabalhos existentes em mais de 80% das instâncias do benchmark, com uma melhoria média de mais de 30%.

O terceiro artigo, "*Multiple Thief Orienteering Problem: Formulação e Abordagens Heurísticas*", está presente no Capítulo 4. O artigo propõe uma ampliação do ThOP para múltiplos ladrões. No *Multiple Thief Orienteering Problem* (MThOP) passamos a utilizar um grupo de ladrões carregando uma mochila capacitada com capacidade coletiva e cada ladrão tem um limite de tempo para coletar os itens. Descrevemos formalmente o MThOP por uma formulação de programação não linear inteira mista e apresentamos duas abordagens heurísticas baseadas nas heurísticas thopGA e thopILS propostas para o ThOP. Os resultados mostraram que o ILS superou o GA para a grande maioria das instâncias, e com a utilização de mais ladrões, o lucro total dos itens coletados aumentou em até 32%.

Por fim, o Capítulo 5 contém um resumo dos resultados encontrados durante a pesquisa e as conclusões obtidas. Neste capítulo também incluímos possíveis trabalhos futuros para continuação da pesquisa.

## 2 A Genetic Algorithm for the Thief Orienteering Problem

### Resumo

This paper approaches the Thief Orienteering Problem, a multi-component problem that combines two combinatorial problems: Orienteering Problem (OP) and Knapsack Problem (KP). In this problem, a person (called thief) has a capacitated knapsack and has a time limit to collect objects distributed in a set of points. The departure and arrival points are fixed. The thief begins his journey with an empty knapsack and travels with speed inversely proportional to the weight of the knapsack. As long as he has time, the thief can go through the points collecting the objects. The objective of the problem is to define which route and which objects the thief must collect to maximize the profit of the knapsack. We developed a heuristic algorithm based on the genetic algorithm (GA) metaheuristic and computational experiments were carried out in order to compare the performance of the developed algorithm with the existing algorithms in the literature. Our results showed that our GA was superior in the majority of the cases.

## 2.1 Introduction

A multicomponent problem is characterized by the combination of two or more subproblems that are interdependent, in the sense that the solution for a subproblem influences the quality of the solutions of the other subproblems (BONYADI; MICHALEWICZ; BARONE, 2013). This class of problems was proposed due to the need to approach the complexity of real world problems, as scientists like Bonyadi, Michalewicz e Barone (2013) and Michalewicz (2012) argue that there is a big distance between the classic problems that are largely studied and the real world problems. The reason is that many studies improve techniques for classic NP-Hard problems to known benchmarks, but these problems lack the main features of reality, such as combination and interdependence, which make problems even more complex (MICHALEWICZ, 2012).

This work approaches the Thief Orienteering Problem (ThOP), a multicomponent problem proposed by Santos e Chagas (2018), which combines two combinatorial problems: the Orienteering Problem and the Knapsack Problem. In this problem, the thief carries a capacitated knapsack and has a time limit to travel through a set of cities, collecting the items (KP component), knowing for each item its weight, profit and location. The thief has a fixed place of departure and arrival, and as long as he has time and capacity in his knapsack, he goes around the cities (called checkpoints) collecting items (OP component). However, as items are collected, the knapsack becomes heavier and the thief walks more slowly. Formally, the minimum and maximum speeds ( $v_{min}$  e  $v_{max}$ ) are the thief speeds when the knapsack is full (at maximum capacity  $W$ ) or empty, respectively. The thief speed ( $v$ ) for a knapsack for a given weight ( $w$ ), with  $0 \leq w \leq W$ , is represented by  $v = v_{max} - (v_{max} - v_{min})w/W$ . The objective of the problem is to determine a route and the items to be collected, in order to maximize the profit of the knapsack (SANTOS; CHAGAS, 2018).

An example for this problem is depicted in Figure 1, where there are four points: the first (start point) and the last (end point) have no items; the other points have a set of items, each item  $i$  with its profit ( $p_i$ ) and weight ( $w_i$ ). For example, the item with identifier 31 is at point 3, has a profit of 110 and weight 3. The distances between each pair of points are presented on the edges. In addition to the information contained in the figure, the thief has a minimum speed 0.1, maximum speed 1, a knapsack with maximum capacity 3 and a time limit of 75 to arrive at the end point.

A solution to the ThOP can be represented only by the items collected in an orderly manner, as the route is derived from the location of the items. The collection of items 32 and 23 has a total profit of  $45 + 50 = 95$ . This solution is feasible, as the capacity and the timeout constraints have been respected, as detailed below:

- the thief travels from the start point (1) to the point (3) at maximum speed, as the

knapsack is empty: time is the distance divided by the speed,  $5 / 1.0 = 5$ ;

- at point (3) item 32 is collected, with weight 2: the speed decreases to 0.4;
- he then travels from point (3) to point (2): travel time is  $8/0.4 = 20$ ;
- at point (2) item 23 is collected, with weight 1: the speed drops to 0.1;
- finally, he travels from point (2) to the end point (4): travel time is  $5/0.1 = 50$ . The total time is  $5+20+50=75$ .

If we reverse the order in which the items are collected (23 and then 32), the solution is not feasible, as the time required to collect the items in this sequence is 77.43, which is over the time limit. For this example, the ideal solution is to collect item 21, thus having a profit of 120, spending time 56 and respecting the capacity 3 of the backpack.

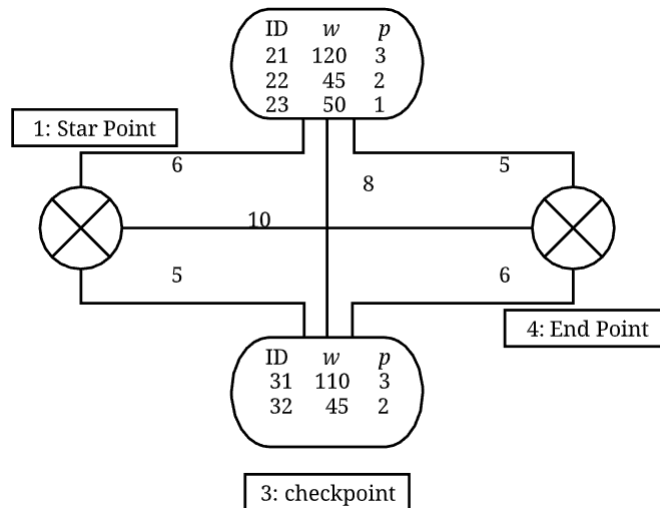


Figura 1 – ThOP example.

An applicability to this problem arises in logistics, assuming that a company has vehicles with limited capacity and the working time of its employees cannot exceed a time limit per day. The company uses vehicles to collect recyclable goods, to carry out the material to reuse process, and each item has a weight and a value. The vehicle’s speed decreases as objects are collected, so the travel time between two collection locations increases. Therefore, the objective is to find a route and the items that must be collected, in order to maximize the profit of using the vehicle.

This problem was idealized by Santos e Chagas (2018), where the problem was formally defined, a mathematical model was formulated and two heuristics were proposed. The model could not be used due to its complexity: the number of variables is exponential in the number of items at a given point, due to the number of possible subsets; and the constraint that defines the speed is non-linear, as the distance must be divided by a

continuous variable. The heuristics were developed based on the Iterative Local Search (ILS) metaheuristic and the Biased Random Key Genetic Algorithm (BRKGA). The results of the computational experiments showed a superiority of BRKGA when compared to ILS, for larger instances.

In this work we present a Genetic Algorithm and analyze its performance to generate upper bounds in the optimal solution of the Thief Orienteering Problem. Through techniques of experimental statistics we show that the developed algorithm produces results with significant improvements in comparison to the already existing algorithms.

This work is organized as follows: in Section II, we present a short bibliographic review of the problem; in Section III, we describe in detail the proposed heuristic; then, in Section IV, we report the computational experiments and analyze the performance of the proposed heuristic algorithm; finally, in Section V, we present our conclusions and suggest further investigations.

## 2.2 Literature Review

The evolution of multi-component problems with interdependence started in 2013 with the Traveling Thief Problem (TTP) ([BONYADI; MICHALEWICZ; BARONE, 2013](#)). This problem consists of a combination of two well-known classic problems: the Traveling Salesman Problem (TSP) and the Knapsack Problem (KP). In TTP, the thief must visit all cities from a set of cities and during the visit he may collect items present in those cities to fill his knapsack. The speed at which the thief moves from one city to another is inversely proportional to the weight of his knapsack, so the heavier the thief's knapsack, the slower he goes. The knapsack is rented and the price to pay is proportional to the time of use. The thief objective is at the same time maximize the profit from the collected items, respecting the capacity of the knapsack, and minimize the total time of the route, as the total revenue is the profit collected minus the amount payed for the knapsack rental.

Since the presentation of the TTP, a benchmark of 9,720 instances has been made available ([POLYAKOVSKIY et al., 2014a](#)). Several heuristics have been proposed: Hill Climbing and Simulated Annealing ([YAFRANI; AHOD, 2018](#)), Ant Colony ([WAGNER, 2016](#)), Variable Neighborhood Descent ([ARAUJO et al., 2018](#)), Profit Guided Coordination ([NAMAZI et al., 2019](#)) and Genetic Algorithm ([VIEIRA et al., 2017](#)). Moreover, an exact algorithm was proposed in ([WU et al., 2017](#)) to verify the efficiency of the existing heuristic algorithms. The TTP was also used in competitions in the main events of Evolutionary Computing ([CEC, 2014](#)) e ([GECCO, 2017](#)). A research was developed in ([WAGNER et al., 2018](#)) that selects the most suitable algorithm, among the twenty one existing, according to the characteristics of the instance.

The ThOP is the most recent problem approaching multi-components with interdependence (SANTOS; CHAGAS, 2018). It was inspired by TTP, but instead of the Traveling Salesman Problem it is based on the Orienteering Problem. The Orienteering Problem is based on an orientation sports game. In this game, players start at a specific checkpoint (origin) and try to visit as many checkpoints as possible and return to a checkpoint within a time limit. Each control point has a score and the objective is to maximize the total score collected. At ThOP, score is not given by solely visiting the checkpoints, the score is given for collected items, which must be carried by the participants to the finish line. However, since they must carry the items, each one has a knapsack to store the items collected, and the speed on the route, and consequently the time, varies according to the weight of the knapsack, just as in the TTP.

For ThOP, to the best of our knowledge, only two heuristics have already been proposed to find good solutions in reasonable time, a BRKGA and an ILS (SANTOS; CHAGAS, 2018). The performance of the algorithm based on the BRKGA was superior to that of ILS. The use of other metaheuristics for ThOP may improve the upper bounds on the optimal solution. A Genetic Algorithm was used in TTP and had good performance, finding for some instances the best known solutions. Due to these good results, we developed a heuristic for ThOP based on the Genetic Algorithm.

## 2.3 Heuristic Approach

In this section we describe the Genetic Algorithm developed for the problem (thopGA). An overview of the algorithm is presented in the flowchart of Figure 2. In the following, we detail each procedure.

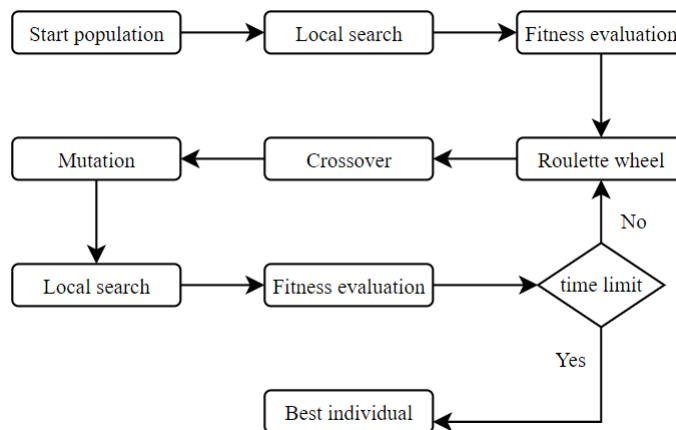


Figura 2 – Flowchart of the thopGA algorithm.

### 2.3.1 Solution representation and evaluation procedure

The solution is represented by two sequences: the first ( $R$ ) contains the points to be visited in the order in which the journey will be carried out (as the initial and final location is fixed, it is not necessary to store this information); the second ( $C$ ) contains the items to be collected (in any order, since the first sequence already informs the collection order). The example of solution given for Figure 1 would be represented by  $R = \{3, 2\}$  and  $C = \{32, 23\}$  (or  $C = \{23, 32\}$ ).

The evaluation procedure starts with the route  $R$  and the collection  $C$  filled and the backpack  $K$  empty. For each location in  $R$ , following the order defined by the permutation, we check the items in  $C$  that are related to the  $R$  location, so all items in a given location is collected in a single visit. For each item, its weight is accumulated and we check if the capacity of the backpack has been respected and if the remaining time is sufficient for the thief to arrive at the final location. If any of these constraints is violated, this solution is considered invalid, otherwise the profit obtained is returned.

### 2.3.2 Local Search

Algorithm 1 presents the local search technique used as a subroutine in the main algorithm here proposed. It explores the neighborhood for a current solution to search for a better quality neighbor. If a better solution is detected, that solution becomes the current solution and the process is repeated until a stopping criterion is reached *it<sub>max</sub>*, thus returning an improved solution.

This local search process has a simple structure that consists of finding a better route to collect the items in the solution, thus reducing the time spent to collect them. Subsequently, new items are inserted in the route, thus increasing the profit of the backpack. These two procedures respect the maximum capacity of the backpack and the route time limit, and are executed until there is no improvement for several iterations.

The process to find a better route is the 2-Opt neighborhood (2). On each iteration, two cities in the route are randomly selected and the visit order of the cities between them are reversed. For example, consider the route  $\{B, C, D, E, F, G, H, I, J\}$  with start point A and end point K. If the selected cities are D and G, the new route is built in three steps: the cities up to the first selected remain the same  $\{B, C, D\}$ ; the route from there until the second city selected is reversed  $\{B, C, D, G, F, E\}$ ; finally, the remaining cities remain in the same order  $\{B, C, D, G, F, E, H, I, J\}$ .

The process for inserting new items is performed to include items not yet collected, in order to improve the profit. The set of items is randomly shuffled and items are inserted one-by-one in the solution, if the item does not belong to the solution. For each item included, the new solution is checked if the time and capacity constraints are still satisfied.

Otherwise, the item is removed. Algorithm 3 show the steps of this neighborhood.

---

**Algoritmo 1:** Local search procedure
 

---

```

1 procedure LOCAL-SEARCH (  $s, it_{max}$  )
2  $k \leftarrow 0$ 
3 while  $k < it_{max}$  do
4    $s' \leftarrow$  2-OPT (  $s, it_{max}$  )
5    $s' \leftarrow$  ITEM-INSERTION (  $s'$  )
6   if  $f(s') > f(s)$  then
7      $s \leftarrow s'$ 
8      $k \leftarrow 0$ 
9   else
10     $k \leftarrow k + 1$ 
11  end
12 end
13  $s$ 

```

---



---

**Algoritmo 2:** Neighborhood 2-OPT procedure
 

---

```

1 procedure 2-OPT (  $s, it_{max}$  )
2  $k \leftarrow 0$ 
3 while  $k < it_{max}$  do
4    $index1 \leftarrow$  RANDOM (1,  $s.number\ of\ points$ )
5    $index2 \leftarrow$  RANDOM (1,  $s.number\ of\ points$ )
6   if  $index1 > index2$  then
7     SWAP (  $index1, index2$  )
8   end
9    $s' \leftarrow$   $s.route$  [1 ..  $index1$ ] +
10     $s.route$  [ $index2$  ..  $index1 + 1$ ] +
11     $s.route$  [ $index2 + 1$  ..  $s.numberofpoints$ ]
12   if  $f(s').feasible$  then
13      $s \leftarrow s'$ 
14      $k \leftarrow 0$ 
15   else
16      $k \leftarrow k + 1$ 
17   end
18 end
19  $s$ 

```

---

**Algoritmo 3:** Neighborhood Item-Insertion procedure

---

```

1 procedure ITEM-INSERTION ( s )
2 i ← number of items
3 t ← SHUFFLE (items)
4 k ← 0
5 while k < i do
6   if IN-SOLUTION (t[k].idItem, s) == False then
7     INSERT-ITEM (s, t[k].idItem)
8     belongs ← IN-SOLUTION (t[k].idPoint)
9     if belongs == False then
10      INSERT-ROUTE (s, t[k].idPoint)
11    end
12  end
13  if f(s).feasible == False then
14    REMOVE-ITEM (s, t[k].idItem)
15    if belongs == False then
16      REMOVE-ROUTE (s, t[k].idPoint)
17    end
18  end
19  k ← k + 1
20 end
21 s

```

---

### 2.3.3 Genetic Algorithm

The Genetic Algorithm (GA) is an evolutionary metaheuristic (HOLLAND, 1992) inspired by the Darwinian principle of the evolution of species (DARWIN, 1956). Their methods use the evolution of a population of individuals, where each individual has a solution to the problem studied. Over the course of evolution, the characteristics of individuals with greater aptitude tend to survive, thus leading the algorithm to more promising solutions.

In the thopGA proposed here, each individual is represented by two vectors, one containing the route and the other the items that should be collected. The population has  $p_k$  individuals, that are generated initially at random, respecting the constraints of the problem: the backpack capacity and time limit. While items fit and the thief has time to arrive at the end point. The vector containing the route is filled in the order that the items are in the collection vector. After the generation of the initial population, all individuals in the population undergo an improvement through the local search explained above.

The thopGA proceeds to the next stage, creating new individuals by recombination

and mutation. The process continues until a maximum number of generations is reached, controlled by a parameter  $Tmax$ . To generate a new child, two parents from the initial population are drawn using the roulette technique, where the probability of choosing a parent is proportional to the quality of his solution. Suppose the parent 1 and 2 collects  $i$  and  $j$  items respectively. Then, the crossover processes are carried out: the first  $i/2$  items from parent 1 and the last  $j/2$  items from parent 2, without repeating items, are inserted in the child and the route is generated from the selected items respecting this sequence, without repeating cities. After, the mutation process is carried out, where an item is selected randomly in the child and exchanged for another random item that is not present in the child. The crossover and mutation processes are carried out with a given probability.

The child generated from the crossover and processes undergoes an improvement through local search. The thopGA has then a component like Memetic Algorithms. The solution value of the child is compared to the value of the worst individual in the population. If the child's value is higher, the worst individual is removed from the population and the child is inserted. At the end of the entire process, the one with the most valuable solution is chosen from the population of individuals. Algorithm 4 summarizes the previously mentioned steps of thopGA.

---

**Algoritmo 4:** Genetic Algorithm procedure
 

---

```

1 procedure thopGA (  $T_{max}, p_c, p_m, it_{max}$  )
2  $Population \leftarrow$  START-POPULATION ( )
3 for  $s$  in  $Population$  do
4   | LOCAL-SEARCH (  $s, it_{max}$  )
5 end
6 while time limit  $T_{max}$  do
7   |  $i \leftarrow$  RANDOM (0, 1)
8   | if  $i < p_c$  then
9     |  $parent1 \leftarrow$  ROULETTE-WHEEL ( )
10    |  $parent2 \leftarrow$  ROULETTE-WHEEL ( )
11    |  $child \leftarrow$  CROSSOVER (  $parent1, parent2$  )
12    |  $i \leftarrow$  RANDOM (0, 1)
13    | if  $i < p_m$  then
14      | |  $child \leftarrow$  MUTATION (  $child$  )
15    | end
16    | LOCAL-SEARCH (  $child, it_{max}$  )
17    | if  $f(child) > f(WORST-INDIVIDUAL (Population))$  then
18      | |  $Population.REMOVE (WORST-INDIVIDUAL (Population))$ 
19      | |  $Population.INSERT (child)$ 
20    | end
21   | end
22 end
23 BEST-INDIVIDUAL (  $Population$  )

```

---

## 2.4 Computational Experiments

The proposed heuristic algorithm was implemented in the C++ language and compiled in Cygwin 3.1.2. The experiments were performed on a notebook with an Intel Core i7 processor with 2.50 GHz clock, 8 GB of memory and Windows 10 64-bit operating system.

The heuristic algorithm was evaluated using different instances, from the ThOP benchmark proposed by Santos e Chagas (2018). The characteristics of each set vary according to the number and location of cities, the number of items in each city, the weight-profit relationship type of the items, the capacity of the backpack and the maximum travel time. The following characteristics are considered in this benchmark:

- TSP based instance groups: `eil51`, `pr107`, `a280` and `dsj1000`;
- number of items per city: 01, 03, 05, and 10;
- weight-profit relation type: bounded-strongly-correlated (`bsc`), uncorrelated (`unc`), uncorrelated-similar-weights (`usw`);
- knapsack capacity class: 01, 05, 10;
- maximum travel time class: 01, 02, 03.

Regarding the parameters of the algorithm, we defined as the stopping criterion the execution time equal to  $m/10$  seconds, the same used by Santos e Chagas (2018), given in terms of the number of items  $m$  of each instance. In addition to the stop parameter, it has four more parameters: the local search stop condition  $it_{max}$ , the number of individuals in the population  $p_k$ , which is the number of individuals that are stored as the best found, the probability of crossover  $p_c$  and the probability of mutation  $p_m$ . All parameters were calibrated empirically and in the final experiment they were performed with  $it_{max} = 50$ ,  $p_k = 100$ ,  $p_c = 0.90$ ,  $p_m = 0.85$ .

As the algorithm has random components, it was executed 10 times for each instance with different random seeds. The average value of the objective function and the best found in these 10 runs is used to compare the algorithm with those from the literature. Therefore we have 10 results for 432 instances, a total of 4320 results, from which we extract two values per instance, the best and the average. In this paper, we present and analyze the results grouped by instance and types, in Tables 1, 2, 3 and 4 and the graphs on Figures 3, 4, 5 and 6.

The metric  $\chi$  used in the work where ThOP was proposed (SANTOS; CHAGAS, 2018) to establish a measure of the quality of convergence of the algorithm was adopted in this paper to verify the quality of the proposed algorithm. The equations (2.1) and (2.2)

show how the metric  $\chi$  is calculated for the thopGA proposed in this work and for the BRKGA proposed by Santos e Chagas (2018). The  $thopGA^{avg}$  and  $BRKGA^{avg}$  values refer to the average of the results found by the thopGA and BRKGA algorithms, respectively, for the corresponding group, and  $thopGA^{best}$  and  $BRKGA^{best}$  are the best solutions for each algorithm. The metric then measures how close the algorithm approaches, on average, to the best known solution. Note that the higher the  $\chi$  metric, the greater the convergence of the algorithm to the best known solution.

$$\chi^{thopGA} = \frac{thopGA^{avg}}{\max(thopGA^{best}, BRKGA^{best})} \cdot 100\% \quad (2.1)$$

$$\chi^{BRKGA} = \frac{BRKGA^{avg}}{\max(BRKGA^{best}, thopGA^{best})} \cdot 100\% \quad (2.2)$$

Tabela 1 – Quality of convergence grouped by number of items per city.

Group	$\chi^{BRKGA}$				$\chi^{thopGA}$			
	01	03	05	10	01	03	05	10
ei151	<b>93</b>	83	74	64	91	<b>95</b>	<b>93</b>	<b>95</b>
pr107	<b>96</b>	71	63	56	77	<b>91</b>	<b>92</b>	<b>93</b>
a208	<b>90</b>	54	47	35	85	<b>96</b>	<b>97</b>	<b>98</b>
dsj1000	83	44	56	74	<b>87</b>	<b>99</b>	<b>98</b>	<b>97</b>

Tabela 2 – Quality of convergence grouped by item relation type.

Group	$\chi^{BRKGA}$			$\chi^{thopGA}$		
	bsc	unc	usw	bsc	unc	usw
ei151	80	79	78	<b>97</b>	<b>90</b>	<b>94</b>
pr107	73	70	71	<b>93</b>	<b>85</b>	<b>89</b>
a208	55	55	57	<b>96</b>	<b>92</b>	<b>94</b>
dsj1000	69	62	62	<b>95</b>	<b>95</b>	<b>96</b>

Tabela 3 – Quality of convergence grouped by maximum travel time class.

Group	$\chi^{BRKGA}$			$\chi^{thopGA}$		
	01	02	03	01	02	03
ei151	77	79	81	<b>94</b>	<b>93</b>	<b>93</b>
pr107	73	71	71	<b>88</b>	<b>89</b>	<b>89</b>
a208	55	56	58	<b>93</b>	<b>94</b>	<b>95</b>
dsj1000	61	65	67	<b>95</b>	<b>95</b>	<b>96</b>

Tabela 4 – Quality of convergence grouped by knapsack capacity class.

Group	$\chi^{BRKGA}$			$\chi^{thopGA}$		
	01	05	10	01	05	10
ei151	<b>94</b>	78	65	91	<b>98</b>	<b>94</b>
pr107	<b>96</b>	65	54	90	<b>92</b>	<b>86</b>
a208	83	46	41	<b>92</b>	<b>96</b>	<b>96</b>
dsj1000	74	62	57	<b>89</b>	<b>99</b>	<b>99</b>

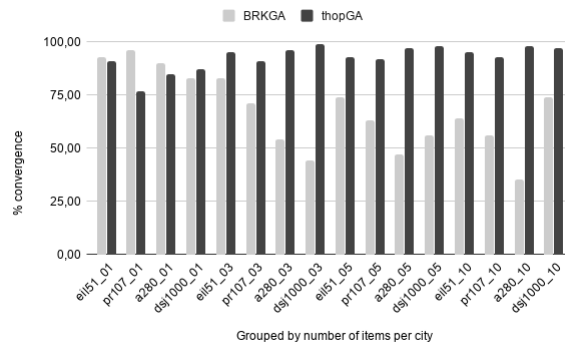


Figura 3 – Quality of convergence per number of items per city.

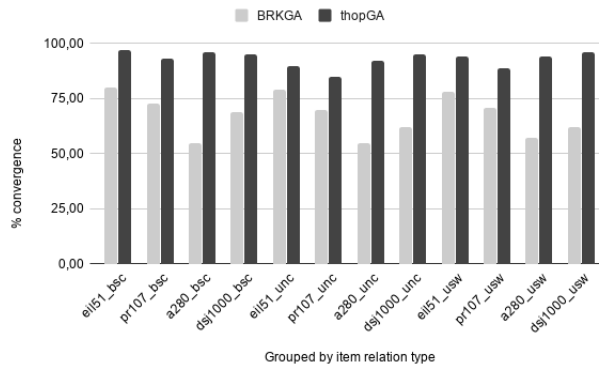


Figura 4 – Quality of convergence per weight-profit relation type.

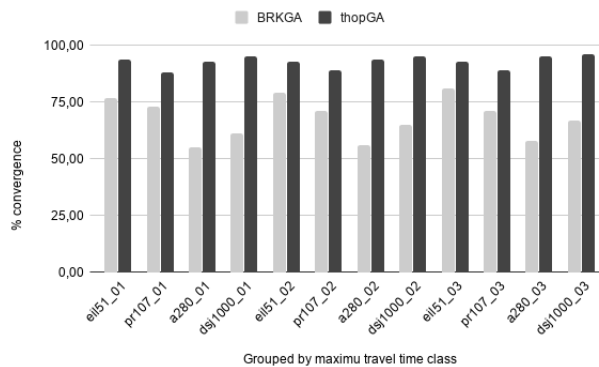


Figura 5 – Quality of convergence per maximum travel time class.

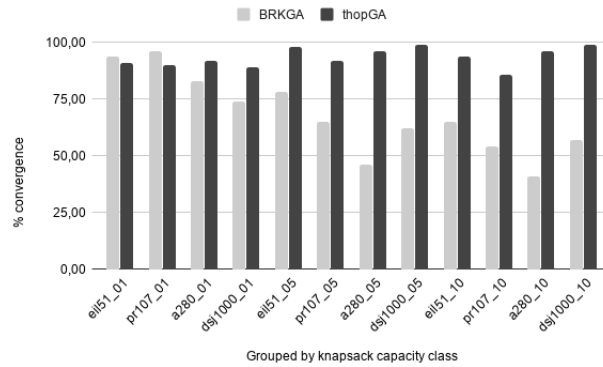


Figura 6 – Quality of convergence per knapsack capacity class.

One can see that the thopGA algorithm remains stable for most groups (convergence around 85%), while the convergence of the BRKGA algorithm decreases as the number of checkpoint increases and the number of items per checkpoint increases for the same number of checkpoint, from 93% to approximately 44%. The BRKGA has a better convergence only for the group with only one item per checkpoint (except for class `dsj1000`) and for the group with the lowest capacity of the backpack (except in classes `a208` and `dsj1000`). We believe that the machines used influenced the worse performance of thopGA, because the fewer items available, the tight is the execution time limit, hence the solution for these instances are more sensitive to the performance of the machine used. The processor of the machine used to carry out the experiments of the BRKGA algorithm is superior to that used for the experiments of the thopGA algorithm.

The behavior of the algorithms changes for different weight-profit relationship, as can be seen in Table 2. Both algorithms perform better for the `bsc` type, with the thopGA algorithm being superior in all types compared to the BRKGA algorithm. The worst performance is for the `unc` type. This indicates that the algorithms still depend on a good relationship between the weight and the value of the items. Regarding the maximum travel time, the algorithms tend to have a better convergence as this parameter increases, as reported in Table 3, except in the BRKGA algorithm in the cases of `pr107`, where the behavior is the opposite. And for the convergence of the algorithms regarding the backpack capacity class, the BRKGA algorithm has a better performance for group 01 (in classes `eil51` and `pr107`) and worse for the others, and the thopGA algorithm was better in groups 01 (in classes `a208` and `dsj1000`), 05 and 10. These results indicate that the algorithms behave better when time and capacity limits are not highly constrained. Instances with tighter limits are more difficult.

Besides the convergence, we also analyze how often a method find an overall best solution, compared to the others. The results are presented in Tables 5 and 6, grouped by the type of instance. The values for each method and group of instances inform in how many of the 108 instances of the group the best (or average) solution found by the

respective algorithm has the best value among the methods. The column “equals” informs in how many there is a tie, i.e., more than one method found the best value. The thopGA was superior to the other methods in all groups, in both cases, the best solution and the average solution in 10 runs. Regarding the best solution, thopGA found the best one in 304 out of 432 instances, which corresponds to more than 70%.

Tabela 5 – Number of best results found for each group

Instances	Best Solutions			
	ILS	BRKGA	thopGA	equals
eil51	12	26	64	6
pr107	17	25	64	2
a208	20	4	84	0
dsj1000	1	15	92	0

Tabela 6 – Number of best average results for each group

Instances	Average Solutions			
	ILS	BRKGA	thopGA	equals
eil51	11	30	64	3
pr107	14	29	65	0
a208	19	2	87	0
dsj1000	0	13	95	0

In order to verify whether the results found by the algorithms already developed for the ThOP problem, including the one proposed in this work, have a significant difference among them, the Friedman test was applied. The Friedman test was chosen because we cannot say that the samples used in the tests are independent (ZIMMERMAN; ZUMBO, 1993). The hypothesis observed in this test is that the algorithms are the same; if this statement is denied, we can say that there is at least one different algorithm. Using the p-value = 0.05, the result found was less than 0.05, thus concluding that there is at least one different algorithm with 95% confidence.

As a way to observe the results, Figure 7 has a boxplot graph that illustrates the results of the experiments. On the x-axis we have the BRKGA (best results currently in the literature) and the thopGA algorithm proposed in this work, and on the y-axis the percentage of solution convergence for each instance. The normalization used to verify the percentage of improvement in the solution of each instance was the  $\chi$  metric presented in equations (2.1) and (2.2). The thopGA algorithm has a higher average result than BRKGA and we can also observe that the thopGA algorithm has greater convergence than BRKGA.

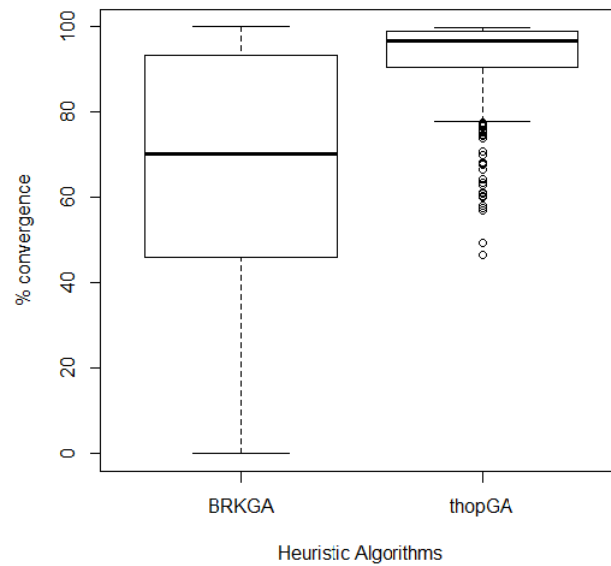


Figura 7 – Comparative analysis of the convergence of BRKGA algorithm and thopGA for all instances.

The algorithm that is apparently the best one, observed in the graph previously, must pass the Wilcoxon test (GEHAN, 1965). In this test, it is verified if the visually better algorithm has a significant difference in relation to the other algorithms. Initially, the p-value is adjusted using the Bonferroni correction (PERNEGER, 1998), and the hypothesis observed for the test is that the pairwise compared algorithms are the same. The value found for (thopGA with BRKGA) is less than 0.05, then we can say that the thopGA algorithm is significantly different from BRKGA and has a better performance.

## 2.5 Conclusions and Further Investigations

In this paper, we approach the Thief Orienteering Problem (ThOP), a combinatorial problem with interdependence. We proposed the thopGA, a Genetic Algorithm based heuristic. We tested thopGA for the 432 instances available in the literature and compared it with the existing heuristic algorithms (ILS and BRKGA). The results showed a superiority of thopGA when compared to the other algorithms, mainly in large instances. We believe that this superiority is due to the combination of individuals who have good characteristics, and also by the local search used in the generation of the initial population and after the new individuals generated, generating others with better results. As a future work, we would like to develop an exact mathematical programming model to test the quality of the developed heuristics algorithms, at least for the smaller instances, and to analyze the performance of new heuristics based on other metaheuristics.

# 3 Busca Local Iterada Aplicada ao Thief Orienteering Problem

## Resumo

Neste trabalho abordamos o *Thief Orienteering Problem* (ThOP), um problema de multicomponente que combina características de dois problemas de otimização combinatória: o problema de orientação e o problema da mochila. Deve-se definir um trajeto e itens a serem coletados ao longo do caminho, com restrições de tempo máximo de percurso e velocidade dependente do peso da mochila, o que torna o ThOP um problema complexo e desafiador. Propomos uma abordagem heurística baseada na metaheurística de busca local iterada e realizamos os experimentos computacionais para comparar o seu desempenho com os algoritmos existentes na literatura. Os resultados mostraram que a abordagem superou os trabalhos existentes em mais de 80% das instâncias do benchmark, com uma melhoria média de mais de 30%.

## 3.1 Introdução

Os problemas multicomponentes possuem as características de combinar dois ou mais subproblemas e eles serem interdependentes, na percepção que a solução para um subproblema influencia na qualidade da solução de outro subproblema (BONYADI; MICHALEWICZ; BARONE, 2013). Apesar de muitos estudos dedicados aos problemas clássicos de otimização poderem ser aplicados diretamente a soluções de problemas do mundo real, como problemas de empacotamento e problema de roteamento de veículos, alguns cientistas argumentam que existe uma grande distância entre os problemas que são estudados e o que realmente são problemas do mundo real, como (MICHALEWICZ, 2012) e (BONYADI; MICHALEWICZ; BARONE, 2013). O motivo é que muitos estudos aprimoram técnicas para problemas clássicos NP-difíceis para *benchmark* conhecidos, mas esses problemas não possuem as principais características da realidade, tais como combinação e interdependência, que tornam os problemas ainda mais complexos (MICHALEWICZ, 2012).

Um problema multicomponente estudado por vários pesquisadores recentemente é o *Traveling Thief Problem* (TTP), que combina dois problemas clássicos: o problema do caixeiro viajantes (TSP) e o problema da mochila (KP). O TTP foi proposto por Bonyadi, Michalewicz e Barone (2013) para oferecer uma generalização de problemas multicomponentes para estudos científicos. No TTP, um indivíduo, chamado de ladrão, deve visitar todas as cidades de um conjunto de cidades e durante a visita pode coletar itens presentes nessas cidades para preencher sua mochila. A velocidade em que o ladrão se desloca de uma cidade para outra é inversamente proporcional ao peso de sua mochila, portanto quanto mais pesada, mais lento ele anda. A mochila possui uma taxa de aluguel e o preço a pagar é proporcional ao tempo de utilização. O objetivo do ladrão é maximizar o custo-benefício que envolve aumentar o valor dos itens coletados e diminuir o tempo total da rota, visto que a taxa de aluguel deve ser descontada no total obtido.

O *Thief Orienteering Problem* (ThOP) proposto Santos e Chagas (2018) foi idealizado como um problema acadêmico multicomponentes que combina dois problemas combinatórios: o problema de orientação (OP) e o clássico problema da mochila (KP). O OP é baseado em um jogo esportivo de orientação (CHAO; GOLDEN; WASIL, 1996a). Neste jogo, os jogadores começam num ponto de controle (origem) especificado, percorrer uma região visitando pontos de controle e retornar a um ponto de controle dentro de um período de tempo limite. No OP, cada ponto de controle tem uma pontuação, então o objetivo é encontrar a rota que maximiza a pontuação total, ou seja, cuja soma das pontuações dos pontos de controle visitados é máxima. Esse problema é amplamente estudado na literatura (ver, por exemplo, (CHAO; GOLDEN; WASIL, 1996b; VANSTEENWEGEN; SOUFFRIAUX; OUDHEUSDEN, 2011)).

No ThOP, assim como no OP, o participante tem um ponto pré-definido de início e

de fim e enquanto houver tempo visita postos de controle para obter pontos. Então, assim como no TTP, um indivíduo denominado de ladrão transporta uma mochila capacitada e possui um tempo limite para percorrer um conjunto de cidades, coletando os itens presentes nelas (componente KP) e cada item possui seu peso e lucro. No entanto, à medida que os itens são coletados, a mochila fica mais pesada e o ladrão caminha mais lentamente. O objetivo do problema é determinar uma rota e os itens a serem coletados, para maximizar o lucro da mochila que corresponde a maximizar a pontuação do OP.

Um exemplo ilustrativo para este problema é apresentado na figura 8. Nela existem quatro locais sendo o primeiro, ponto de partida e o último, ponto de chegada, que não possuem itens. Os demais locais são pontos de controle e contêm um conjunto de itens, e cada item possui seu peso e valor. Por exemplo, o item com identificador 6 está no local 3 e possui o valor 95 e peso 1. As distâncias entre cada par de pontos estão sobre as arestas. Além das informações apresentadas na figura, são consideradas as velocidades mínima e máxima, 0.1 e 1 respectivamente, o tempo limite para atingir o ponto de chegada 90 e capacidade máxima da mochila 3.

Uma solução para este problema é o ladrão coletar os itens 6 e 1 percorrendo os pontos de controle 1, 3, 2 e 4. Esta solução possui um lucro total de  $95 + 100 = 195$ , sendo uma solução viável e ótima, visto que foram respeitadas as restrições de capacidade da mochila e tempo limite de conclusão, conforme detalhado abaixo:

- percorrer do ponto inicial (1) para o ponto de controle (3) com a velocidade máxima, pois a mochila está vazia: o tempo gasto é a distância dividida pela velocidade,  $8/1.0 = 8$ ;
- no ponto de controle (3) é coletado o item 6, de peso 1: assim a velocidade diminui para 0.7;
- percorrer do ponto de controle (3) para o (2): o tempo gasto é de  $14/0.7 = 20$ ;
- no ponto de controle (2) é coletado o item 1: a velocidade diminui para 0.1;
- e para percorrer do ponto de controle (2) para o ponto de chegada (4): o tempo gasto é  $6/0.1 = 60$ , totalizando 88, portanto dentro do limite de tempo.

Se trocarmos a ordem em que os itens são coletados para 1 e 6, a nova rota passará a ser 1,2,3 e 4 e a solução inviável. O tempo necessário para coletar os itens nessa sequência é 145, superior ao tempo limite. Pode-se notar a interdependência dos problemas verificando que a solução ótima para o problema da mochila, coletar o item 4 de peso 3 e valor 200, não é a solução válida para o ThOP, pois a restrição de tempo limite não é respeitada, sendo gasto um tempo de 108.

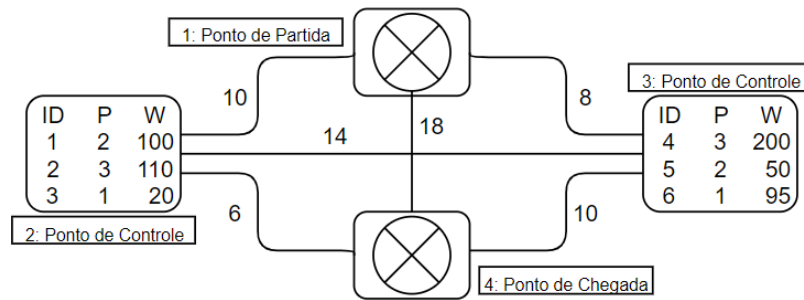


Figura 8 – Exemplo para o *Thief Orienteering Problem*.

O ThOP, apesar de ser um problema acadêmico, está presente em vários cenários do mundo real. Um exemplo é o de coleta de mercadorias de maneira ecológica, utilizando bicicletas. As bicicletas possuem capacidades limitadas e o tempo de trabalho do ciclista não deve ultrapassar um tempo limite por dia. Os materiais que devem ser recolhidos possuem um peso e seu valor, e a velocidade da bicicleta diminui à medida que os itens são coletados, assim o tempo de deslocamento entre dois pontos de coleta aumenta. Logo, o objetivo é determinar uma rota e os materiais que devem ser recolhidos, para maximizar o lucro em cada bicicleta. Outro exemplo, citado por Santos e Chagas (2018), acontece na coleta de material reciclável. Se a empresa não possui capacidade para coletar todo o material, deve selecionar cuidadosamente quais coletar para obter o maior benefício, sabendo que o veículo fica mais lento (ou gasta mais combustível) à medida que aumenta a carga transportada.

Para o ThOP, Santos e Chagas (2018) apresentaram uma formulação matemática de programação não-linear inteira mista, mas nenhum resultado desta formulação foi apresentado devido à sua complexidade, uma vez que o número de variáveis é exponencial no número de itens de uma determinada cidade, relacionado ao número de subconjuntos possíveis; e a restrição que define a velocidade é não linear, pois a distância deve ser dividida por uma variável contínua. Por sua vez, dois algoritmos heurísticos foram propostos, um baseado em busca local iterada (ILS, do inglês *Iterated Local Search*) e outro baseado no algoritmo genético de chaves aleatórias viciadas (BRKGA, do inglês *Biased Random-Key Genetic Algorithms*). Os resultados dos experimentos computacionais mostraram uma superioridade do BRKGA quando comparado ao ILS, principalmente para as instâncias maiores. Os autores atribuíram essa superioridade à variedade introduzida dos indivíduos mutantes.

Seguindo os bons resultados do algoritmo populacional proposto por Santos e Chagas (2018), propomos anteriormente um algoritmo populacional denominado thopGA que realiza melhoramentos em seus indivíduos utilizando busca local por uma combinação de vizinhanças (FAËDA; SANTOS, 2020a). Uma das razões do thopGA ter superado resultados anteriores pode ter sido o melhorado dos indivíduos que incorporamos. Então, neste trabalho, propomos o uso de uma abordagem heurística para o ThOP baseada na

metaheurística de busca local iterada utilizando a busca local apresentada anteriormente. Essa abordagem foi adotada para verificar o desempenho da busca local utilizando um algoritmo não populacional e por meio de técnicas da estatística experimental mostramos que o algoritmo desenvolvido produz resultados com diferenças significantes em relação aos algoritmos já existentes.

O restante deste artigo está estruturado da seguinte forma. Na Seção 2, descrevemos formalmente o ThOP. Na Seção 3, apresentamos nossa abordagem de solução proposta para abordar o ThOP. A Seção 4 relata os experimentos e faz uma análise do desempenho da abordagem proposta em relação às anteriores já propostas na literatura. Finalmente, na Seção 5, apresentamos nossas conclusões e sugerimos novas investigações.

## 3.2 Definição Formal e Modelo Matemática

O modelo a seguir foi proposto por Santos e Chagas (2018), e é transcrito aqui para definir formalmente o problema. Seja um conjunto de  $n$  pontos: 1 é o ponto de partida,  $n$  o ponto de chegada e  $(2, \dots, n-1)$  são pontos de controle. Em cada ponto de controle, existe um ou mais itens e cada item tem seu peso  $w_i$  e lucro  $p_i$ . Também é conhecido o tempo limite  $T$  para se chegar ao ponto final  $n$ , a capacidade máxima da mochila  $W$ , a distância  $d_{ij}$  entre quaisquer dois pontos  $i$  e  $j$  e a velocidade máxima e mínima do ladrão ( $v_{min}$  e  $v_{max}$ ). O objetivo é determinar uma rota que começa pelo ponto de partida até o ponto de chegada, colocando na mochila os itens selecionados nos pontos de controle visitados, com a finalidade de maximizar o lucro total da mochila, respeitando sua capacidade máxima e não ultrapassando o tempo limite para realizar o percurso.

O modelo matemática contém todas as características do problema. Seja  $S_i$  o conjunto de itens pertencentes ao ponto de controle  $i$ . Para cada  $s \subseteq S_i$ ,  $w_i^s$  representa o peso total dos itens contidos em  $s$  enquanto  $p_i^s$  é o lucro total desse subconjunto. Além disso, seja a constante  $v = (v_{max} - v_{min}) / W$  a perda de velocidade para cada unidade de peso dentro da mochila e  $M'$  e  $M''$  constantes suficientemente grandes. Apresentam-se abaixo as variáveis de decisão do modelo:

- $x_{ij}^s$ : variável binária que obtém 1 se o ladrão percorre o arco  $(i, j)$  após coletar os itens do subconjunto  $s \subseteq S_i$  e 0 caso contrário.
- $q_i$ : variável que informa o peso da mochila após a coleta dos itens na cidade  $i$ .
- $t_i$ : variável que informa o tempo de chegada do ladrão na cidade  $i$ .

$$\max \sum_{i=1}^{n-1} \sum_{j=2}^n \sum_{s \subseteq S_i} p_i^s * x_{ij}^s \quad (3.1)$$

sujeito a:

$$\sum_{j=2}^n x_{1j}^{\circ} = 0 \quad (3.2)$$

$$\sum_{i=1}^{n-1} \sum_{s \subseteq S_i} x_{in}^s = 1 \quad (3.3)$$

$$\sum_{i=1}^{n-1} \sum_{s \subseteq S_i} x_{ij}^s - \sum_{i=2}^n \sum_{s \subseteq S_i} x_{ji}^s = 0, \forall j = 2..n-1 \quad (3.4)$$

$$q_j \geq q_i + \sum_{j'=2}^n \sum_{s \subseteq S_i} w_j^s * x_{jj'}^s - M' * \left( 1 - \sum_{s \subseteq S_i} x_{ij}^s \right), \forall i = 1..n, \forall j = 1..n \quad (3.5)$$

$$t_j \geq t_i + \frac{d_{ij}}{v_{max} - v * q_i} - M'' * \left( 1 - \sum_{s \subseteq S_i} x_{ij}^s \right), \forall i = 1..n, \forall j = 1..n \quad (3.6)$$

$$x_{ij}^s \in \{0, 1\}, \forall i = 1..n, \forall j = 1..n, \forall s \subseteq S_i \quad (3.7)$$

$$0 \leq q_i \leq W, \forall i = 1..n, 0 \leq t_i \leq T, \forall i = 1..n \quad (3.8)$$

Segundo Santos e Chagas (2018), o primeiro termo da função objetivo (3.2) tem por finalidade maximizar o lucro total dos itens coletados. O ladrão deve começar no ponto 1 carregando uma mochila vazia (3.1) e chegar ao ponto  $n$  (3.2). Ao visitar um ponto de controle, o ladrão pode coletar um subconjunto de itens deste ponto (3.3), assim aumentando o peso de sua mochila (3.4) que conseqüentemente diminui sua velocidade, afetando o tempo para chegar no próximo ponto de controle (3.5). O peso da mochila e o tempo total de caminhada devem estar sempre dentro dos limites indicados (3.7).

### 3.3 Proposta Heurística

No trabalho de Faêda e Santos (2020a) bons resultados foram encontrados para o problema do ThOP e a justificativa para o bom desempenho do algoritmo foi a combinação de indivíduos que possuem boas características e o melhoramento dos indivíduos, mas neste trabalho mostramos que utilizando a busca local em uma metaheurística não populacional gerou resultados de melhor qualidade. Portanto, nesta seção apresentamos o algoritmo heurístico baseado na metaheurística de busca local iterada (ILS), utilizando as vizinhanças da busca local desenvolvidas no trabalho de Faêda e Santos (2020a), de-

nominado de thopILS, a fim de obter soluções de qualidade.

### 3.3.1 Representação da Solução e Procedimento de Avaliação

A solução é representada por duas sequências: a primeira ( $R$ ) contém os locais a serem visitados e a ordem em que será realizado o percurso (como o local inicial e final são fixos não é necessário armazenar essas informações); a segunda ( $C$ ) contém os itens que foram selecionados para serem coletados (em qualquer ordem, já que a primeira sequência já informa a ordem das coletas). Utilizando o exemplo da Figura 1, uma solução é representada por  $R = \{3, 2\}$  e  $C = \{6, 1\}$  (ou  $C = \{1, 6\}$ ).

O procedimento de validação é iniciado com a rota  $R$ , a coleta  $C$  preenchidas e a mochila  $K$  vazia. Para cada localidade de  $R$ , seguindo a ordem definida pela permutação, verificamos os itens em  $C$  que esta relacionada a localidade de  $R$ , assim todos os itens de uma determinada localidade serão coletados em uma única visita. Para cada item é acumulado o seu peso, e verificado se a capacidade da mochila foi respeitada e se o tempo restante é suficiente para o ladrão chegar à localidade final. Se alguma das restrições não forem respeitadas esta solução é considerada inválida, caso contrário é retornado o lucro obtido.

### 3.3.2 Busca Local

A busca local utilizada nesse trabalho foi proposta por [Faêda e Santos \(2020a\)](#) e implementada conforme as instruções apresentadas no algoritmo 5 e utilizada como sub-rotina no principal algoritmo proposto neste trabalho. Ele explora a vizinhança de uma solução atual para buscar um vizinho de melhor qualidade. Se uma solução melhor for detectada, essa solução passa a ser a solução atual e esse processo é repetido até que atinja o critério de parada, assim retornando uma solução melhorada.

Esse processo de busca local possui uma estrutura simples que consiste em encontrar uma rota melhor para coletar os itens da solução, assim diminuindo o tempo gasto para recolher os itens. Posteriormente, novos itens são inseridos na rota, assim aumentando o lucro da mochila. Todos esses procedimentos respeitam a restrição de capacidade máxima da mochila e o tempo da rota.

O procedimento de encontrar uma rota melhor é realizada através da vizinhança 2-opt (algoritmo 6). Em cada iteração, o algoritmo sorteia duas localidades na sequência da rota e realiza uma inversão na rota entre estas localidades. Como exemplo, suponha a seguinte rota:  $\{2, 3, 4, 5, 6, 7, 8, 9, 10\}$  e a localidade inicial 1 e final  $n$ . Se os índices sorteados foram os das posições 4 e 7, o ajuste possui três passos: mantêm-se a sequência da primeira localidade até o primeiro índice sorteado  $\{2, 3, 4\}$ ; inverte-se a rota do primeiro índice até o segundo índice sorteado  $\{2, 3, 4, 7, 6, 5\}$ ; por fim, mantêm as

localidades restantes  $\{2, 3, 4, 7, 6, 5, \mathbf{8}, \mathbf{9}, \mathbf{10}\}$ .

O procedimento de inserção de novos itens é realizado com os itens que ainda não pertencem ao conjunto solução. O conjunto dos itens é embaralhado aleatoriamente, e um a um são inseridos na solução, se o item não pertence a solução. Para cada item inserido é verificado se as restrições foram respeitadas, assim passando a fazer parte da solução, caso contrario é retirado. O algoritmo 7 resume as etapas mencionadas anteriormente da vizinhança.

---

**Algoritmo 5:** Procedimento de busca local

---

```

1 procedimento BUSCA-LOCAL (  $s, b_{max}, v_{2-opt}$  )
2  $k \leftarrow 0$ 
3 while  $k < b_{max}$  do
4    $s' \leftarrow 2\text{-OPT} (s, v_{2-opt})$ 
5    $s' \leftarrow \text{INSERÇÃO-ITEM} (s')$ 
6   if  $f(s') > f(s)$  then
7      $s \leftarrow s'$ 
8      $k \leftarrow 0$ 
9   else
10     $k \leftarrow k + 1$ 
11  end
12 end
  Saída:  $s$ 

```

---



---

**Algoritmo 6:** Procedimento de vizinhança 2-OPT

---

```

1 procedimento 2-OPT (  $s, v_{2-opt}$  )
2  $k \leftarrow 0$ 
3 while  $k < v_{2-opt}$  do
4    $\text{índice}^1 \leftarrow \text{ALEATÓRIO} (1, s.\text{número de pontos})$ 
5    $\text{índice}^2 \leftarrow \text{ALEATÓRIO} (1, s.\text{número de pontos})$ 
6   if  $\text{índice}^1 > \text{índice}^2$  then
7     TROCA ( $\text{índice}^1, \text{índice}^2$ )
8   end
9    $s' \leftarrow s.\text{rota} [1 .. \text{índice}^1] +$ 
10     $s.\text{rota} [\text{índice}^2 .. \text{índice}^1 + 1] +$ 
11     $s.\text{rota} [\text{índice}^2 + 1 .. s.\text{número de pontos}]$ 
12   if  $f(s').\text{viável}$  then
13      $s \leftarrow s', k \leftarrow 0$ 
14   else
15      $k \leftarrow k + 1$ 
16   end
17 end
  Saída:  $s$ 

```

---

**Algoritmo 7:** Procedimento de vizinhança de Inserção-Item

---

```

1 procedimento INSERÇÃO-ITEM (  $s$  )
2  $m \leftarrow$  número de itens,  $k \leftarrow 0$ 
3  $t \leftarrow$  EMBARALHAR (itens)
4 while  $k < m$  do
5     if PERTENCE-SOLUÇÃO ( $t[k].idItem$ ,  $s$ ) == Falso then
6         INSERIR-ITEM( $s$ ,  $t[k].idItem$ )
7          $pertence \leftarrow$  PERTENCE-SOLUÇÃO ( $t[k].idLocal$ )
8         if  $pertence ==$  Falso then
9             | INSERIR-ROTA( $s$ ,  $t[k].idLocal$ )
10        end
11        if  $f(s).viável ==$  Falso then
12            | REMOVER-ITEM ( $s$ ,  $t[k].idItem$ )
13            | if  $pertence ==$  Falso then
14                | REMOVER-ROTA ( $s$ ,  $t[k].idLocal$ )
15            end
16        end
17    end
18     $k \leftarrow k + 1$ 
19 end
Saída:  $s$ 

```

---

### 3.3.3 Busca Local Iterada

A busca local iterada é uma metaheurística que explora o espaço da solução por meio de um método de busca local e um método que aplica perturbações a cada ótimo local encontrado. O método de perturbação é capaz e responsável por escapar dos ótimos locais e, assim, explorar outras regiões do espaço da solução (veja (LOURENÇO; MARTIN; STÜTZLE, 2003) para obter mais detalhes sobre a metaheurística ILS).

Como a metaheurística de ILS é amplamente utilizada na literatura para resolver problemas combinatórios (LOURENÇO; MARTIN; STÜTZLE, 2019), e como o ILS possui sub-rotinas que utilizam buscas locais, resolvemos aplicar as estratégias de busca local propostas por Faêda e Santos (2020a) para o problema do ThOP utilizando a metaheurística ILS.

Este método é escrito em detalhes no algoritmo 8, que começa com uma solução  $s$  gerada aleatoriamente e respeitando as restrições do ThOP. Em seguida, aplica uma busca local (algoritmo 5) em  $s$ . Obedecendo o tempo de execução  $t_{max}$ , as seguintes etapas são realizadas: uma perturbação na solução, em que, respeitando o parâmetro  $p_{max}$ , o conjunto  $s$  é dividido em  $p_{max}$  subconjuntos e nos  $d$  últimos subconjuntos, os itens são eliminados e

novos itens aleatórios são inseridos. Ainda na perturbação, se as cidades dos itens retirados não tiverem outros itens para serem coletados, essas cidades são retiradas da rota, e os novos itens inseridos. Se suas cidades não estiverem na rota, elas são inseridas no final da rota, respeitando as restrições do problema, assim produzindo uma nova solução  $s'$ . Em seguida uma nova busca local (algoritmo 5) é realizada em  $s'$ , resultando em  $s''$ . Se a solução  $s''$  é melhor que  $s$ ,  $s$  se torna a solução atual  $s''$  e o tamanho da perturbação  $d$  é redefinido para 1 caso contrário, o tamanho aumenta em uma unidade até um limite  $d_{max}$ .

---

**Algoritmo 8:** Heurística Proposta utilizando Busca Local Iterada
 

---

```

1 procedimento thopILS (  $s, p_{max}, t_{max}, b_{max}, v_{2-opt}$  )
2  $s \leftarrow$  solução inicial aleatória
3  $s \leftarrow$  BUSCA-LOCAL (  $s, b_{max}, v_{2-opt}$  )
4  $d \leftarrow 1$ 
5  $d_{max} \leftarrow p_{max}$ 
6 while tempo limite  $t_{max}$  do
7    $s' \leftarrow$  PERTUBAÇÃO (  $s, d$  )
8    $s'' \leftarrow$  BUSCA-LOCAL (  $s', b_{max}, v_{2-opt}$  )
9   if  $f(s'') > f(s)$  then
10     $s \leftarrow s''$ 
11     $d \leftarrow 1$ 
12  else
13     $d \leftarrow \text{MAX}( d + 1, d_{max} )$ 
14  end
15 end
  
```

**Saída:**  $s$

---

### 3.4 Experimentos Computacionais

Nessa seção são comparados os algoritmos existentes na literatura para o problema do ThOP com o algoritmo heurístico proposto neste trabalho, o thopILS. O thopILS foi desenvolvido na linguagem de programação C++ e compilado no Cygwin 3.1.2. Os experimentos foram realizados em um notebook com processador Intel Core i7 com 2.50 GHz de clock, 8 GB de memória e sistema operacional Windows 10 64 bits. Para permitir uma comparação justa, pois o processamento dos algoritmos é baseado em tempo, o algoritmo thopGA foi compilado no mesmo cenário que o thopILS, os demais algoritmos não foram encontrados para serem executados no mesmo cenário, portanto, foram utilizados os resultados encontrados na literatura.

Para avaliar a qualidade do algoritmo heurístico proposto, foram utilizadas todas as instâncias do ThOP criadas por Santos e Chagas (2018). De acordo com os autores, as

instâncias foram desenvolvidas utilizando com base as instâncias do TTP (POLYAKOVSKIY et al., 2014b), removendo os itens da última cidade da instância e adicionando um tempo máximo para a conclusão da viagem. Foram criadas 432 instâncias que variam de acordo os seguintes recursos:

- número de cidades: 51, 107, 280 e 1000 (denominadas de eil51, pr107, a280, dsj1000);
- número de itens por cidade: 01, 03, 05 e 10;
- tipos de relações dos itens: pesos e valores dos itens são limitados e fortemente correlacionados (bsc), não correlacionados (não) ou não correlacionados com pesos semelhantes (usw);
- classe de capacidade da mochila: 01, 05 e 10;
- classe de tempo de conclusão: 01, 02 e 03. Esses valores se referem a 50%, 75% e 100% dos tempos de referência específicos da instância definidos em (BONYADI; MICHALEWICZ; BARONE, 2013).

Em relação aos parâmetros do algoritmo, definimos como critério de parada como nos trabalhos anteriores (SANTOS; CHAGAS, 2018; FAËDA; SANTOS, 2020a), onde o tempo de execução é igual a  $m/10$  segundos, dado em termos do número de itens  $m$  de cada instância. Além do parâmetro de parada, possui outros três parâmetros: o tamanho da permutação dos itens  $d_{max}$ , que no caso desse algoritmo é o número de perturbações necessárias para reiniciar a busca de uma solução aleatória, o número de interações da busca local  $b_{max}$  e o número de interações da vizinhança 2-opt  $v_{2-opt}$ . Todos os parâmetros foram calibrados empiricamente e no experimento final foram realizadas com  $d_{max} = 4$ ,  $b_{max} = 50$  e  $v_{2-opt} = 50$ .

Como o algoritmo possui componentes aleatórios, ele foi executado 10 vezes para cada instância com diferentes sementes aleatórias. O valor médio da função objetivo e o melhor encontrado nessas 10 execuções são usados para comparar o algoritmo com os da literatura. Portanto, temos 10 resultados para 432 instâncias, um total de 4320 resultados, dos quais extraímos dois valores por instância, o melhor e a média. Neste artigo, apresentamos e analisamos agrupados por instâncias e tipos, nos gráficos da Figura 9 e na Tabela 7.

Os resultados encontrados pelos algoritmos são apresentados na Tabela 7, agrupados pelo tipo de instância. Os valores para cada método e grupo de instâncias informam em quantas das 108 instâncias do grupo a melhor (ou média) solução encontrada pelo respectivo algoritmo tem o melhor valor entre os métodos. A coluna “igual” informa quantos empates existem, ou seja, mais de um método encontrou o melhor valor. O thopILS foi superior aos outros métodos em todos os grupos, em ambos os casos, a melhor solução e

a solução média em 10 execuções. Em relação à melhor solução, o thopILS encontrou a melhor em 356 das 432 instâncias, o que corresponde a mais de 82%.

Tabela 7 – Número de médios e melhores resultados encontrados para cada grupo

Grupos	Melhor Solução					Solução Média				
	ILS	BRKGA	thopGA	thopILS	igual	ILS	BRKGA	thopGA	thopILS	igual
eil51	9	15	20	56	8	7	12	20	66	3
pr107	12	17	3	74	2	12	15	9	72	0
a208	9	1	0	98	0	6	2	10	90	0
dsj1000	0	9	1	98	0	0	8	5	95	0

A métrica  $\chi$  utilizada no trabalho Santos e Chagas (2018) para estabelecer uma medida da qualidade de convergência dos algoritmos foi adotada neste artigo para verificar a qualidade do algoritmo heurístico proposto. A equação 3.9 mostra como a métrica  $\chi$  é calculada para os algoritmos analisados. O valor  $\alpha$  é substituído pelo nome do algoritmo que está sendo analisado. O valor  $\alpha^{média}$  é substituído pela variável correspondente ao algoritmo analisado,  $thopILS^{média}$ ,  $thopGA^{média}$ ,  $ILS^{média}$  e  $BRKGA^{média}$  que se referem ao valor médio da função objetivo encontrada pelos algoritmos, e  $thopILS^{melhor}$ ,  $thopGA^{melhor}$ ,  $ILS^{melhor}$  e  $BRKGA^{melhor}$  são as melhores soluções por algoritmo. A métrica mede então quão próximo o algoritmo, em média, se aproxima da melhor solução conhecida. Observe que quanto maior a métrica  $\chi$ , maior a convergência do algoritmo para a melhor solução conhecida.

$$\chi^\alpha = \frac{\alpha^{média}}{\max(ILS^{melhor}, BRKGA^{melhor}, thopGA^{melhor}, thopILS^{melhor})} \cdot 100\% \quad (3.9)$$

Podemos observar que os algoritmos thopILS e thopGA permanecem estáveis para a maioria dos grupos (convergência superior à 85% e 73%, respectivamente), enquanto a convergência dos algoritmos BRKGA e ILS diminui à medida que o número de pontos de controle aumenta e a quantidade de itens por ponto de controle para um mesmo número de pontos de controle, de 96% e 95% para aproximadamente 57% e 37%. Interpretando todos os gráfico da figura 9, o algoritmo thopILS possui melhor convergência em todos os grupos, perdendo apenas nas classes pr107\_01 no grupo de número de itens por cidade, para o algoritmo ILS e BRKGA e na classe a280\_bsc no grupo de tipo da instância, para o algoritmo thopGA. Estes resultados indicam uma grande distância entre o thopILS aos demais algoritmos, sendo superior em todos grupos de instâncias.

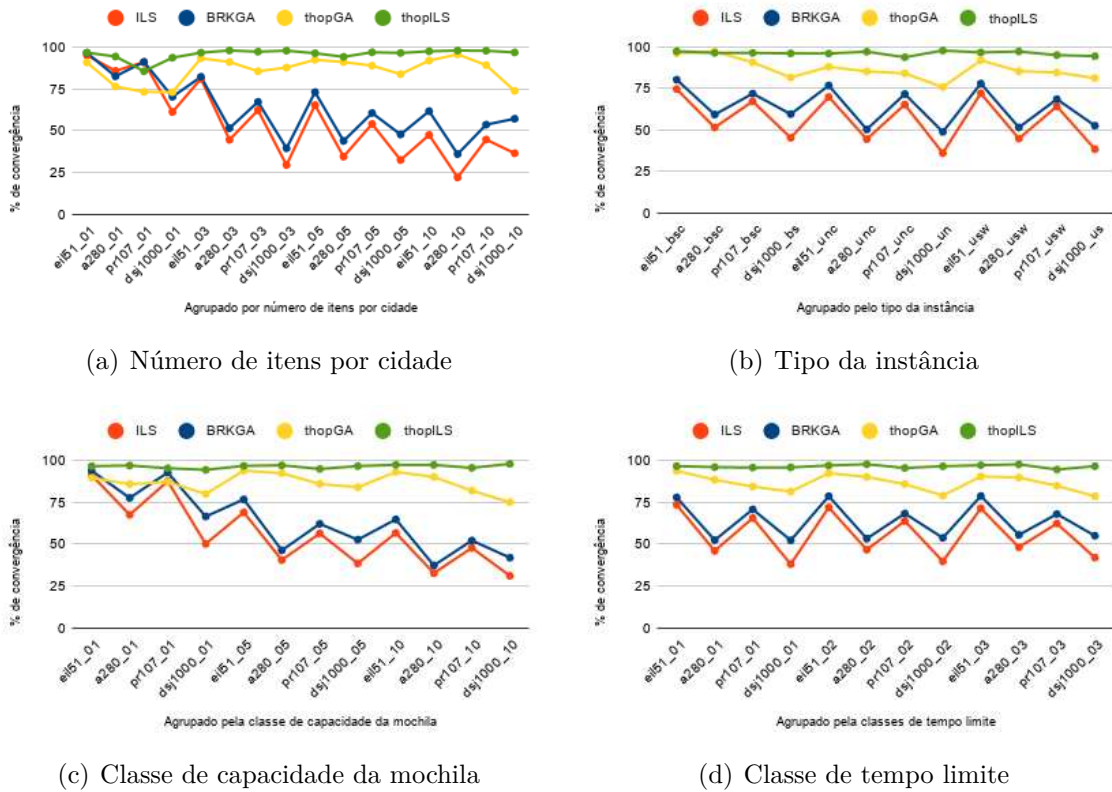


Figura 9 – Qualidade da convergência dos algoritmos heurísticos agrupados por relações das instâncias.

Para comparar se os resultados encontrados pelos algoritmos já desenvolvidos para o problema do ThOP e o thopILS, possuem diferença significativa entre eles, foi aplicado o teste de Friedman. O teste de Friedman foi escolhido pois não podemos afirmar que as amostras utilizadas nos testes são independentes (ZIMMERMAN; ZUMBO, 1993). A hipótese observada para o teste é que os algoritmos são iguais, caso essa afirmação seja negada, podemos afirmar que existe pelo menos um algoritmo diferente. Utilizando o  $p$ -valor = 0,05, o resultado encontrado foi menor que 0,05, assim concluindo que existe pelo menos um algoritmo diferente com 95% de confiança.

Como forma de observar os resultados, a figura 10 possui um gráfico boxplot que ilustra os resultados dos experimentos. No eixo  $x$  estão os algoritmos ILS, BRKGA e thopGA e o thopILS proposto neste trabalho, e no eixo  $y$  a porcentagem de convergência da solução de cada instância. A normalização utilizada para verificar a porcentagem de melhoramento na solução de cada instância foi a mesma métrica apresentada na equação (3.9). O algoritmo thopILS possui a média dos resultados maior que os demais algoritmos e também conseguimos observar que possui uma maior convergência.

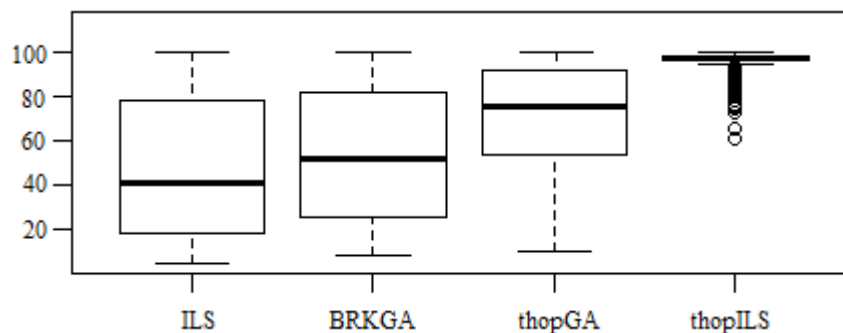


Figura 10 – Análise comparativa da convergência dos algoritmos heurísticos para todas as instâncias

O algoritmo aparentemente melhor observado no gráfico da Figura 10 deve passar pelo teste de Wilcoxon. Neste teste, é verificado se o algoritmo visualmente melhor possui diferença significativa em relação aos demais algoritmos (GEHAN, 1965). Inicialmente o p-valor é ajustado utilizando a correção de Bonferroni (PERNEGER, 1998), e a hipótese observada para o teste é que os algoritmos comparados par-a-par são iguais. Os valores encontrados comparando o thopILS par-a-par com os demais algoritmos são inferiores a 0,05, assim podemos afirmar que o algoritmo thopILS é diferente significativamente dos demais algoritmos e possui um desempenho melhor.

### 3.5 Conclusão e Trabalhos Futuros

Neste trabalho abordamos um problema combinatório com interdependência, o *Thief Orienteering Problem* (ThOP), onde propusemos um algoritmo heurístico baseado na metaheurística de busca local iterada e denominado de thopILS. Testamos o thopILS para as 432 instâncias disponíveis na literatura e comparamos com os algoritmos heurísticos já existentes (ILS, BRKGA e thopGA). Os resultados mostraram uma superioridade do thopILS quando comparado com os demais algoritmos, principalmente nas instâncias grandes, sendo melhor em mais de 80% das 432 instâncias, com uma melhoria média superior a mais de 30%. Com base em nossa análise, isso se deve à eficiência da otimização da busca local utilizada. Como trabalho futuro, gostaríamos de investigarmos algoritmos exatos para resolver instâncias ThOP pequenas e médias para testar a eficiência dos algoritmos heurísticos desenvolvidos para resolver instâncias ThOP e estabelecer ótimos globais.

## 4 Multiple Thief Orienteering Problem: Formulação e Abordagens Heurísticas

### Resumo

Este artigo apresenta o Multiple Thief Orienteering Problem (MThOP), uma ampliação do Thief Orienteering Problem (ThOP) para vários ladrões. No MThOP um grupo de pessoas (chamada de ladrões) carrega uma mochila capacitada que possui uma capacidade coletivamente e cada ladrão tem um limite de tempo para coletar itens distribuídos em um conjunto de pontos de controle. Os ladrões têm pontos iniciais e finais fixos, iniciam sua jornada com a mochila vazia e viajam com velocidade inversamente proporcional ao peso da mochila. Enquanto há tempo, os ladrões podem visitar os pontos de controle e coletar itens. O objetivo do problema é determinar as rotas dos ladrões e os itens a serem coletados para maximizar o lucro da mochila. Descrevemos formalmente o MThOP por uma formulação de programação não linear inteira mista e apresentamos duas abordagens heurísticas baseadas nas metaheurísticas de Algoritmo Genético (GA) e Busca Local Iterada (ILS). Nossos resultados mostraram que o ILS superou o GA para a grande maioria das instâncias e que com a utilização de mais ladrões, o lucro total dos itens coletados aumentou em até 32%.

## 4.1 Introdução

O *Traveling Thief Problem* (TTP) (BONYADI; MICHALEWICZ; BARONE, 2013) é um problema multicomponente que consiste na combinação de dois problemas clássicos, o Problema do Caixeiro Viajante (PCV) e o Problema da Mochila (PM). Os problemas foram combinados de forma que a solução ideal para ambos não corresponde necessariamente a uma solução ideal do TTP. A motivação do TTP é ter um problema em que as interações de seus componentes tenham que ser investigadas sistematicamente.

No TTP, apenas um indivíduo, denominado ladrão, tem que visitar todas as cidades de um conjunto de cidades (componente PCV) e pode lucrar roubando itens presentes nestes pontos e acumulando em uma mochila alugada (componente PM). A velocidade em que o ladrão se desloca de uma cidade para outra é inversamente proporcional ao peso de sua mochila, portanto quanto mais pesada, o ladrão caminha mais lentamente. A mochila possui uma taxa de alocação e o preço a pagar é proporcional ao tempo de utilização. O objetivo do ladrão é maximizar o lucro dos itens coletados, respeitando a capacidade da mochila, e também minimizar o tempo total da rota, já que isto reduz o custo de uso da mochila.

O TTP vem sendo estudado desde sua apresentação, e um *benchmark* de 9.720 instâncias foi disponibilizado (POLYAKOVSKIY et al., 2014a). Alguns algoritmos heurísticos abordando diversas metaheurísticas foram propostos (YAFRANI; AHIOD, 2018) (*Hill Climbing e Simulated Annealing*), (WAGNER, 2016) (Colônia de Formigas), (ARAUJO et al., 2018) (*Variable Neighbourhood Descent*) e (VIEIRA et al., 2017) (Algoritmo Genético), e um algoritmo exato foi proposto em (WU et al., 2017) para verificar a eficiência dos algoritmos heurísticos existentes. O TTP também foi utilizado em competições nos principais eventos de Computação Evolutiva (CEC, 2014) e (GECCO, 2017). Uma pesquisa foi desenvolvida por Wagner et al. (2018) que seleciona o algoritmo mais adequado, entre os vinte e um existentes, para cada tipo de instância do problema.

O *Multiple Traveling Thieves Problem* (MTTP) (CHAND; WAGNER, 2016) é uma extensão do TTP para incluir vários ladrões. No MTTP, assim como no TTP, os ladrões partem de uma cidade inicial e após realizar suas trajetórias recolhendo os itens devem retornar a cidade inicial. A velocidade de cada ladrão varia de acordo com o peso da mochila individualmente, e em cada cidade podem recolher um ou mais itens. No entanto, não é mais necessário os ladrões passarem por todas as cidades. As cidades que não possuem itens interessantes ou muito distantes, podem não ser visitadas. Uma cidade pode ser visitada por diferentes ladrões, mas um item pode ser coletado apenas uma vez, por algum dos ladrões. O peso máximo da mochila deve ser respeitado globalmente. Isso significa que as mochilas não possuem limites individuais, assim os ladrões podem recolher a quantidade de itens que quiserem, mas a somatória de todas as mochilas dos ladrões devem respeitar o peso máximo da coleta. Assim, eles cooperam na coleta dos itens. A

ideia principal do problema é que, após planejar a rota e os itens que devem ser coletados, cada ladrão carrega uma mochila capacitada com o limite necessário para a realização da coleta dos itens. No final todos os ladrões se encontram para reunir os itens numa única mochila.

O *Thief Orienteering Problem* (ThOP) (SANTOS; CHAGAS, 2018) também foi projetado como um problema acadêmico de múltiplos componentes com diferentes interações e restrições em mente: ele combina o Problema de Orientação (PO) e o Problema da Mochila (MP). O PO é um problema bem estudado em pesquisa operacional (como podemos ver em (CHAO; GOLDEN; WASIL, 1996a) (GOLDEN; LEVY; VOHRA, 1987) (GUNAWAN; LAU; VANSTEENWEGEN, 2016) (VANSTEENWEGEN; SOUFFRIAUX; OUDHEUSDEN, 2011)), onde um participante começa em um determinado ponto, viaja através de uma região visitando postos de controle e tem que chegar a um ponto de controle dentro um determinado momento. Cada ponto de controle possui uma pontuação, e o objetivo do participante é encontrar a rota que maximize a pontuação total, ou seja, cuja soma das pontuações dos pontos de controle visitados seja máxima.

No ThOP, o ladrão transporta uma mochila capacitada e possui um tempo limite para percorrer um conjunto de pontos de controle, coletando os itens presentes nelas (componente PM) e cada item possui seu peso e lucro. Os locais de partida e chegada são pré-definidos. Enquanto tiver tempo e capacidade em sua mochila vai percorrendo os pontos de controle coletando os itens (componente OP). No entanto, à medida que os itens são coletados, a mochila fica mais pesada e o ladrão caminha mais lentamente. O objetivo do problema é determinar uma rota e os itens a serem coletados, para maximizar o lucro da mochila.

Para o ThOP, Santos e Chagas (2018) propuseram uma formulação de Programação Não Linear Inteira Mista, mas nenhum resultado computacional da formulação foi apresentado devido à complexidade da formulação. Entretanto, dois algoritmos heurísticos simples foram propostos, um baseado em Pesquisa Local Iterada (ILS) e um baseado em um Algoritmo Genético de Chave Aleatória Polarizada (BRKGA) (SANTOS; CHAGAS, 2018). O BRKGA superou o ILS em grandes instâncias, e os autores atribuíram isso à diversificação introduzida nos indivíduos mutantes.

No trabalho de Faêda e Santos (2020a) é apresentado o thopGA, uma nova abordagem de algoritmo heurístico para o ThOP, baseada em algoritmo genético (GA) que realiza melhoramentos em seus indivíduos utilizando busca local que possui uma combinação de vizinhanças. A busca local utilizada possui um procedimento simples, onde é utilizada a vizinhança 2-opt para melhorar a rota de coleta dos itens selecionados. Com o melhoramento da rota sobra mais tempo para recolher novos itens, portanto novos itens são inseridos para serem coletados. Os resultados dos experimentos computacionais mostraram uma superioridade do thopGA em relação aos algoritmos ILS e BRKGA.

Os autores caracterizaram essa melhoria à combinação de indivíduos que possuem boas características e as estratégias de melhoramentos dos indivíduos.

No trabalho de [Faêda e Santos \(2020b\)](#) foi desenvolvido um algoritmo heurístico para o ThOP, baseado na metaheurística de Pesquisa Local Iterada (ILS) utilizando a busca local apresentada em [\(FAÊDA; SANTOS, 2020a\)](#). Essa abordagem foi adotada para verificar o desempenho da busca local utilizando um algoritmo não populacional e foram utilizadas técnicas da estatística experimental para mostrar que o algoritmo desenvolvido produziu resultados com diferenças significantes em relação aos algoritmos já existentes.

Posteriormente, [Chagas e Wagner \(2020\)](#) propuseram o ACOThOP, um algoritmo de otimização de colônias de formigas juntamente com uma nova heurística de empacotamento para lidar de forma individual e interativa com os componentes do problema. Eles investigaram a importância dos componentes por meio da configuração automatizada de algoritmos e avaliaram a abordagem em um amplo conjunto de instâncias.

Neste artigo, propomos uma variante do ThOP denominada de *Multiple Thief Orienteering Problem* (MThOP). O MThOP é uma ampliação da presente formulação do ThOP para incluir vários ladrões, assim como foi feito no MTTP para o TTP. Essa variante, além de tornar o problema mais realista, utilizando grupos de ladrões para a realização da coleta, também adiciona uma dificuldade maior ao problema, por haver várias planos de coletas que precisam ser otimizados simultaneamente. Também existe interdependência entre os ladrões que realizaram as coletas, significando que otimizar os ladrões separadamente não produz necessariamente o melhor resultado. Além de propor este novo problema de múltiplos componentes, são propostas duas heurísticas diferentes utilizando as heurísticas propostas em [\(FAÊDA; SANTOS, 2020a\)](#) (thopGA) com base em Algoritmo Genético e [\(FAÊDA; SANTOS, 2020b\)](#) (thopILS) com base na metaheurística Busca Local Iterada desenvolvida para o ThOP, a fim de gerar soluções para o MThOP, como método de partida para lidar com o problema.

O restante do artigo está organizado da seguinte forma. A Seção 2 fornece detalhes do proposto MThOP e apresenta um exemplo detalhado, para demonstrar a característica do problema. Na Seção 3, descrevemos formalmente e propomos uma formulação matemática para o MThOP. As seções 4 e 5 discutem os detalhes das heurísticas usadas para resolver o MThOP e os resultados obtidos. O artigo conclui com um resumo geral e uma discussão sobre possíveis direções futuras de pesquisa na Seção 6.

## 4.2 Descrição do Problema

O MThOP, assim com o ThOP [\(SANTOS; CHAGAS, 2018\)](#), consiste em percorrer um conjunto de pontos de controle com o objetivo de coletar itens que possuem valores e pesos, e que estão situados em cada um desses pontos. Utilizando vários ladrões, nova

característica do MThOP, eles precisam iniciar sua caminhada em uma mesma ponto e finalizar em outro mesmo ponto, pois podemos supor que os ladrões estão trabalhando juntos e precisam chegar todos ao mesmo ponto para juntar o saque. Vários ladrões podem visitar o mesmo ponto de controle, mas cada item pode ser coletado por um único ladrão. Um ladrão pode optar por visitar um ponto e pegar um ou vários itens. Visitar todas os pontos não é obrigatório.

Cada ladrão pode pegar o maior número possível de itens, desde que respeite a capacidade da mochila coletivamente, o que significa que os ladrões não possuem mochilas com capacidades individuais. No entanto, à medida que os itens são coletados, a mochila fica mais pesada e o ladrão caminha mais lentamente. De maneira mais clara, as velocidades pré-definidas mínima e máxima são as velocidades do ladrão quando a mochila estiver respectivamente cheia (em sua capacidade máxima) ou vazia, e a velocidade do ladrão é inversamente proporcional ao peso da mochila.

Uma justificativa para adotar essa estratégia, também utilizada para o MTTP (CHAND; WAGNER, 2016), é a seguinte. Supomos que os itens saqueados e reunidos no ponto final devem ser transportados juntos para outro local e o veículo que realizará o transporte tenha uma capacidade limitada. Como os ladrões sabem as rotas e os planos de coletas antes de iniciar a jornada, eles podem escolher mochilas apropriadas, portanto, um ladrão com rota menor pode pegar uma mochila menor e outro pode pegar uma mochila maior. Os ladrões devem respeitar um tempo limite total para realizar as coletas e chegar ao ponto final. Cada ladrão possui uma velocidade de deslocamento individual, que é calculada exatamente como no ThOP, portanto quanto mais pesados os itens coletados por um ladrão, mais devagar é o seu deslocamento. O objetivo é encontrar um caminho do ponto inicial ao ponto final, para cada ladrão, enchendo a mochila com itens cuidadosamente escolhidos nos pontos de controle visitados no caminho, de forma a maximizar o lucro total da mochila, desde que a capacidade coletiva da mochila não seja ultrapassada e o tempo individual de caminhada dos ladrões não ultrapasse o limite de tempo.

Para elucidar as propriedades do MThOP, ilustramos na figura 11 um exemplo de instância do ThOP que envolve 4 pontos de controle e 6 itens. Observe que não há itens nos pontos de controle inicial (1) e final (4), e nos demais pontos de controle (2 e 3) existem alguns itens de diferentes pesos ( $P$ ) e valores ( $W$ ) distribuídos. As distâncias de cada par de pontos são fornecidas. A seguir, apresentamos em detalhes algumas soluções para essa instância. Para esse exemplo, vamos considerar a velocidade mínima = 0.1, velocidade máxima = 1.0, capacidade da mochila coletivamente = 4 e tempo limite = 90.

O MThOP pode ser tratado com ThOP quando utilizamos apenas um ladrão. Uma solução utilizando apenas um ladrão é coletar os itens 6 e 1 percorrendo os pontos de controle 1, 3, 2 e 4. Esta solução possui um lucro total de  $95 + 100 = 195$ , sendo uma solução viável, visto que foram respeitadas as restrições de capacidade da mochila e tempo

limite de conclusão, conforme detalhado abaixo:

- percorrer do ponto inicial (1) para o ponto de controle (3) com a velocidade máxima, pois a mochila está vazia: o tempo gasto é a distância dividida pela velocidade,  $8/1.0 = 8$ ;
- no ponto de controle (3) é coletado o item 6, de peso 1: assim a velocidade diminui para 0.775;
- percorrer do ponto de controle (3) para o (2): o tempo gasto é de  $14/0.78 = 18.064$ ;
- no ponto de controle (2) é coletado o item 1, de peso 2, totalizando 3, portanto dentro da capacidade da mochila. A velocidade diminui para 0.325;
- e para percorrer do ponto de controle (2) para o ponto de chegada (4): o tempo gasto é  $6/0.325 = 18.461$ , totalizando 44.525, portanto dentro do limite de tempo.

Considere agora o caso do MThOP com dois ladrões, com o primeiro coletando os itens na rota acima. O segundo ladrão coleta o item 3 percorrendo os pontos de controle 1, 2 e 4. Este ladrão possui o lucro individual de 20, e podemos ver que também foram respeitadas as restrições de tempo:

- percorrer do ponto inicial (1) para o ponto de controle (2) com a velocidade máxima, pois a mochila do segundo ladrão está vazia: o tempo gasto é a distância dividida pela velocidade,  $10/1.0 = 10$ ;
- no ponto de controle (2) é coletado o item 3, de peso 1: assim a velocidade diminui para 0.775;
- e para percorrer do ponto de controle (2) para o ponto de chegada (4): o tempo gasto é  $6/0.775 = 7.742$ , totalizando 17.742, portanto dentro do limite de tempo.

Esta solução possui um lucro total de 195 (primeiro ladrão) + 20 (segundo ladrão) = 215, sendo que ambos os ladrões chegaram no ponto de chegada dentro do limite de tempo e a soma dos pesos das mochilas 3 (primeiro ladrão) + 1 (segundo ladrão) = 4 respeitaram a capacidade coletiva da mochila.

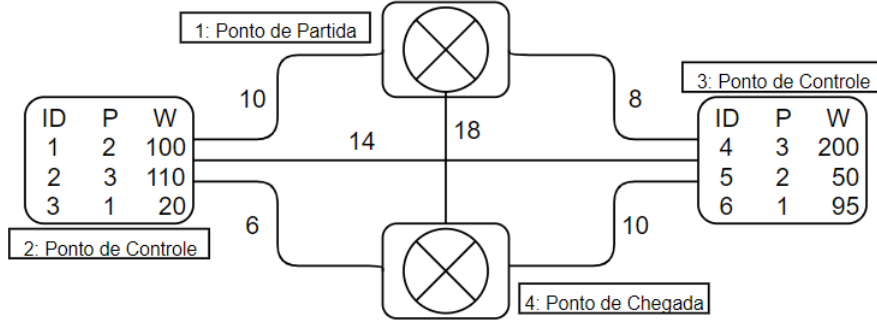


Figura 11 – Exemplo exemplo de instância do ThOP para solucionar o MThOP. Fonte: (FAÊDA; SANTOS, 2020b)

### 4.3 Definição Formal e Formulação Matemática

A presente pesquisa ajustou a definição formal e modelo matemático do ThOP apresentados por Santos e Chagas (2018) para o MThOP. Portanto, a definição formal para o MThOP é descrita a seguir: Seja um conjunto de  $n$  pontos: 1 é o ponto de partida,  $n$  o ponto de chegada e  $(2, \dots, n-1)$  são pontos de controle. Em cada ponto de controle, existe um ou mais itens e para cada item  $i$  são informados seu peso  $w_i$  e lucro  $p_i$ . Os itens podem ser recolhidos por um conjunto de  $m$  ladrões, mas cada item pode ser recolhido apenas por um ladrão. Também é conhecido o tempo limite para chegar ao ponto final  $T$ , a capacidade máxima da mochila  $W$  coletivamente, a distância  $d_{ij}$  entre quaisquer dois pontos  $i$  e  $j$  e a velocidade máxima e mínima dos ladrões ( $v_{max}$  e  $v_{min}$ ). O objetivo é determinar uma rota que começa pelo ponto de partida até o ponto de chegada para cada ladrão e os itens selecionados nos locais de verificação visitados, com a finalidade de maximizar o lucro total da mochila, respeitando a capacidade máxima da mochila globalmente e não ultrapassando o tempo limite individualmente para realizar o percurso.

A seguinte formulação matemática contém todas as características do problema. Para descrevê-lo adequadamente, seja  $S_i$  o conjunto de itens localizados no vértice  $i$ . Para cada  $s_k \subseteq S_i$ ,  $w_i^{s_k}$  representa o peso total dos itens contidos em  $s_k$  enquanto  $p_i^{s_k}$  é o lucro total desse subconjunto para o ladrão  $k$ . Além disso, seja a constante  $v_k = v_{max} - w_k \cdot (v_{max} - v_{min}) / W$  a perda de velocidade para cada unidade de peso dentro da mochila, sendo  $w_k$  o peso da mochila do ladrão  $k$ , e  $M'$  e  $M''$  constantes suficientemente grandes. As variáveis de decisão são:

- $x_{ki}^{s_k}$  : variável binária que recebe 1 se o ladrão  $k$  cruzar o arco  $(i, j)$  após coletar os itens do subconjunto  $s_k \subseteq S_i$  e 0 caso contrário.
- $q_{ki}$  : variável que informa o peso da mochila do ladrão  $k$  após a coleta dos itens na cidade  $i$ .
- $t_{ki}$  : variável que informa o horário de chegada do ladrão  $k$  na cidade  $i$ .

$$\max \sum_{k=1}^m \sum_{i=1}^{n-1} \sum_{j=2}^n \sum_{s_k \subseteq S_i} p_i^{s_k} * x_{ijk}^{s_k} \quad (4.1)$$

sujeito a:

$$\sum_{j=2}^n x_{k1j}^{\emptyset} = 1, \forall k = 1..m \quad (4.2)$$

$$\sum_{i=1}^{n-1} \sum_{s_k \subseteq S_i} x_{kin}^{s_k} = 1, \forall k = 1..m \quad (4.3)$$

$$\sum_{i=1}^{n-1} \sum_{s_k \subseteq S_i} x_{ijk}^{s_k} - \sum_{i=2}^n \sum_{s_k \subseteq S_j} x_{jik}^{s_k} = 0, \forall j = 2..n-1, \forall k = 1..m \quad (4.4)$$

$$q_{kj} \geq q_{ki} + \sum_{j'=2}^n \sum_{s_k \subseteq S_{j'}} w_j^{s_k} * x_{kjj'}^{s_k} - M' * \left( 1 - \sum_{s_k \subseteq S_i} x_{ijk}^{s_k} \right), \forall i = 1..n, \forall j = 1..n, \forall k = 1..m \quad (4.5)$$

$$t_{kj} \geq t_{ki} + \frac{d_{ij}}{v_{max} - v_k * q_{ki}} - M'' * \left( 1 - \sum_{s_k \subseteq S_i} x_{kij}^{s_k} \right), \forall i = 1..n, \forall j = 1..n, \forall k = 1..m \quad (4.6)$$

$$x_{kij}^{s_k} \in \{0, 1\}, \forall i = 1..n, \forall j = 1..n, \forall k = 1..m, \forall s_k \subseteq S_i \quad (4.7)$$

$$0 \leq \sum_{k=1}^m q_{kn} \leq W \quad (4.8)$$

$$0 \leq t_{kn} \leq T, \forall k = 1..m \quad (4.9)$$

$$x_{kij}^{s_k} + x_{k'ij}^{s_{k'}} \leq 1, \forall i = 2..n-1, \forall k = 1..m, \forall k' = 1..m, s_k \in S_i, s_{k'} \in S_i, s_k \cap s_{k'} \neq \emptyset \quad (4.10)$$

A função objetivo (4.1) tem por finalidade maximizar o lucro total dos itens coletados por todos os ladrões. Todos os ladrões devem começar no ponto 1 carregando uma mochila vazia (4.2) e todos os ladrões devem chegar no ponto  $n$  (4.3). Ao visitar um ponto de controle, os ladrões podem coletar um subconjunto de itens deste ponto (4.4). Assim, aumentando o peso da mochila individual (4.5), que consequentemente diminui

sua velocidade individual, afetando o tempo para chegar ao próximo ponto de controle (4.6). O peso da mochila coletivamente deve estar sempre dentro dos limites de capacidade máxima (4.8) e o tempo total individual de cada ladrão deve estar dentro do limite de tempo máximo (4.9). Os ladrões não podem coletar o mesmo item (4.10).

## 4.4 Abordagens Heurísticas

As heurísticas abordadas neste trabalho utilizam das heurísticas propostas para o ThOP em (FAËDA; SANTOS, 2020a) (thopGA) com base em algoritmo genético e (FAËDA; SANTOS, 2020b) (thopILS) com base na metaheurística de Busca Local Iterada. Os algoritmos de cada heurística foram executados sequencialmente para selecionar a rota e os itens a serem recolhidos de cada ladrão. Os itens coletados por cada ladrão são excluídos, para que nenhum item seja coletado mais de uma vez.

A solução é representada pelo conjunto solução  $S$ . O componente  $s_k$  é a solução encontrada para cada ladrão presente no conjunto de  $m$  ladrões, que é formado por dois conjuntos: o primeiro ( $R_k$ ) contém os locais a serem visitados e a ordem em que será realizada a rota (o ponto inicial e final não são armazenadas por serem fixos); o segundo ( $C_k$ ) possui os itens que serão coletados pelo ladrão  $k$  (utilizando qualquer ordem, pois o primeiro conjunto armazena a ordem dos pontos de controle). Utilizando o exemplo da Figura 1, a solução para MThOP é representada por  $R_1 = \{1, 3, 2, 4\}$ ,  $C_1 = \{6, 1\}$  (ou  $C_1 = \{1, 6\}$ ) e  $R_2 = \{1, 2, 4\}$ ,  $C_2 = \{3\}$

O procedimento de avaliação é realizado para cada subconjunto  $s_k$  do conjunto solução  $S$ . Para cada ponto de controle de  $R_k$ , seguindo a ordem definida pela permutação, verificamos os itens em  $C_k$  que estão relacionados ao ponto de controle de  $R_k$ , assim todos os itens de um determinado ponto de controle são coletados de uma única vez. Para cada item é acumulado o seu peso, e verificado se a capacidade da mochila foi respeitada globalmente e se o tempo restante do ladrão  $k$  é suficiente para chegar ao ponto final. Se alguma das restrições não forem respeitadas esta solução é classificada como inválida, caso contrário o lucro dos itens coletados pelos  $k$  ladrões é retornado.

O Algoritmo 9 apresenta um pseudocódigo genérico para as abordagens heurísticas propostas para o MThOP. Ele possui duas estratégias diferentes, uma utiliza todos os itens disponíveis e outra restringe os ladrões em recolher apenas itens da solução ótima do problema da mochila. Essa restrição aborda o *upper bound*, que é o maior lucro possível que os ladrões podem recolher, mas recolher apenas esses itens restritos pode ser muito custoso em tempo, portanto, restringindo os itens podemos estar deixando de encontrar a solução ótima para o MThOP.

Na primeira linha são passados os parâmetros da proposta heurística. O parâmetro  $m$  é a quantidade de ladrões utilizados para coletar os itens, o conjunto de itens  $C$ , onde

foram utilizadas duas estratégias, uma com todos os itens disponíveis e outra apenas com os itens da solução ótima para o problema da mochila, e o *Algoritmo* que foram abordados o thopGA (FAËDA; SANTOS, 2020a) e o thopILS (FAËDA; SANTOS, 2020b). Para cada ladrão  $k$  é gerada a solução  $s_k$  e para que os planos de coleta dos subconjuntos não possuam itens iguais e inviabilize a solução, os itens coletados no componente  $s_k$  são removidos do conjunto de itens disponíveis  $C$ . No final é retornado os componentes de soluções para cada ladrão.

Portanto, o Algoritmo 9 representa quatro abordagens diferentes: O M-thopGA que utiliza o algoritmo thopGA e todos os itens disponíveis; o M-thopGA\* que também utiliza o algoritmo thopGA, mas com os itens da solução ótima do problema da mochila; o M-thopILS que faz uso do algoritmo thopILS e todos os itens à disposição; e o M-thopILS\* que possui no seu escopo o algoritmo thopILS e recolhe apenas itens da solução ótima do problema da mochila.

---

**Algoritmo 9:** Heurísticas Propostas para o M-thop

---

```

1 procedimento: M-thop (  $m, C, Algoritmo$  )
2 foreach  $k$  in  $1..m$  do
3    $s_k \leftarrow Algoritmo(C)$ 
4    $C \leftarrow \text{REMOVER-ITENS}(s_k)$ 
5 end

```

**Saída:**  $s_1, s_2, \dots, s_k$

---

## 4.5 Análise dos Resultados

Os experimentos computacionais realizados para verificar o desempenho dos algoritmos propostos foram executados em um notebook com processador Intel Core i7 com 2.50 GHz de clock, 8 GB de memória e sistema operacional Windows 10 64 bits. Os algoritmos foram desenvolvidos em C++ e compilados no Cygwin 3.1. Essas configurações são as mesmas aplicadas para implementação e experimentos dos algoritmos thopGA e thopILS utilizados no escopo do M-thopGA e M-thopILS.

As instâncias utilizadas para validar o desempenho dos algoritmos heurísticos propostos para o MThOP foram as mesmas utilizadas para o ThOP criados por Santos e Chagas (2018). Foram usadas todas as 432 instâncias. Essas instâncias apresentam as seguintes características:

- número de pontos de controle: 51, 107, 280 e 1000 (denominadas de eil51, pr107, a280, dsj1000);
- número de itens por ponto de controle: 01, 03, 05 e 10;

- tipos de relações dos itens: pesos e valores dos itens são limitados e fortemente correlacionados (bsc), não correlacionados (não) ou não correlacionados com pesos semelhantes (usw);
- classe de capacidade da mochila: 01, 05 e 10.
- classe de tempo de conclusão: 01, 02 e 03. Esses valores se referem a 50%, 75% e 100% dos tempos de referência específicos da instância definidas em (SANTOS; CHAGAS, 2018).
- número de ladrões: 01, 02, 03, 04, 05;

Os parâmetros dos algoritmos thopGA e thopILS utilizados no escopo do M-thopGA e M-thopILS foram os mesmos valores utilizados em (FAÊDA; SANTOS, 2020a; FAÊDA; SANTOS, 2020b) no problema do ThOP. Como os algoritmos possuem componentes aleatórios, eles foram executados 10 vezes para cada instância, com diferentes sementes. Portanto, possuímos 10 resultados para 432 instâncias, e cada resultado foi analisado para 1, 2, 3, 4 e 5 ladrões.

Os gráficos da Figura 12 ilustram para cada quantidade de ladrões a proximidade dos melhores resultados das abordagens heurísticas com o ótimo global do problema da mochila, que é o *upper bound* para o MThOP. Analisando os gráficos observamos uma melhor convergência das abordagens heurísticas (M-thopGA\* e M-thopILS\*) que utilizam a estratégia de operar apenas com os itens da solução ótima do problema da mochila. Também podemos analisar que, com o aumento do número de ladrões, a convergência chega cada vez mais próxima do *upper bound*. Um exemplo para demonstrar o benefício de utilizar vários ladrões pode ser observado na Figura 12, onde a heurística do M-thopILS\* possui um aumento de até 32% do lucro. Ainda analisando os resultados dos gráficos da Figura 12 podemos observar uma convergência dos resultados das abordagens heurísticas (M-thopGA\* e M-thopILS\*) para o *upper bound* a partir da utilização de quatro ladrões.

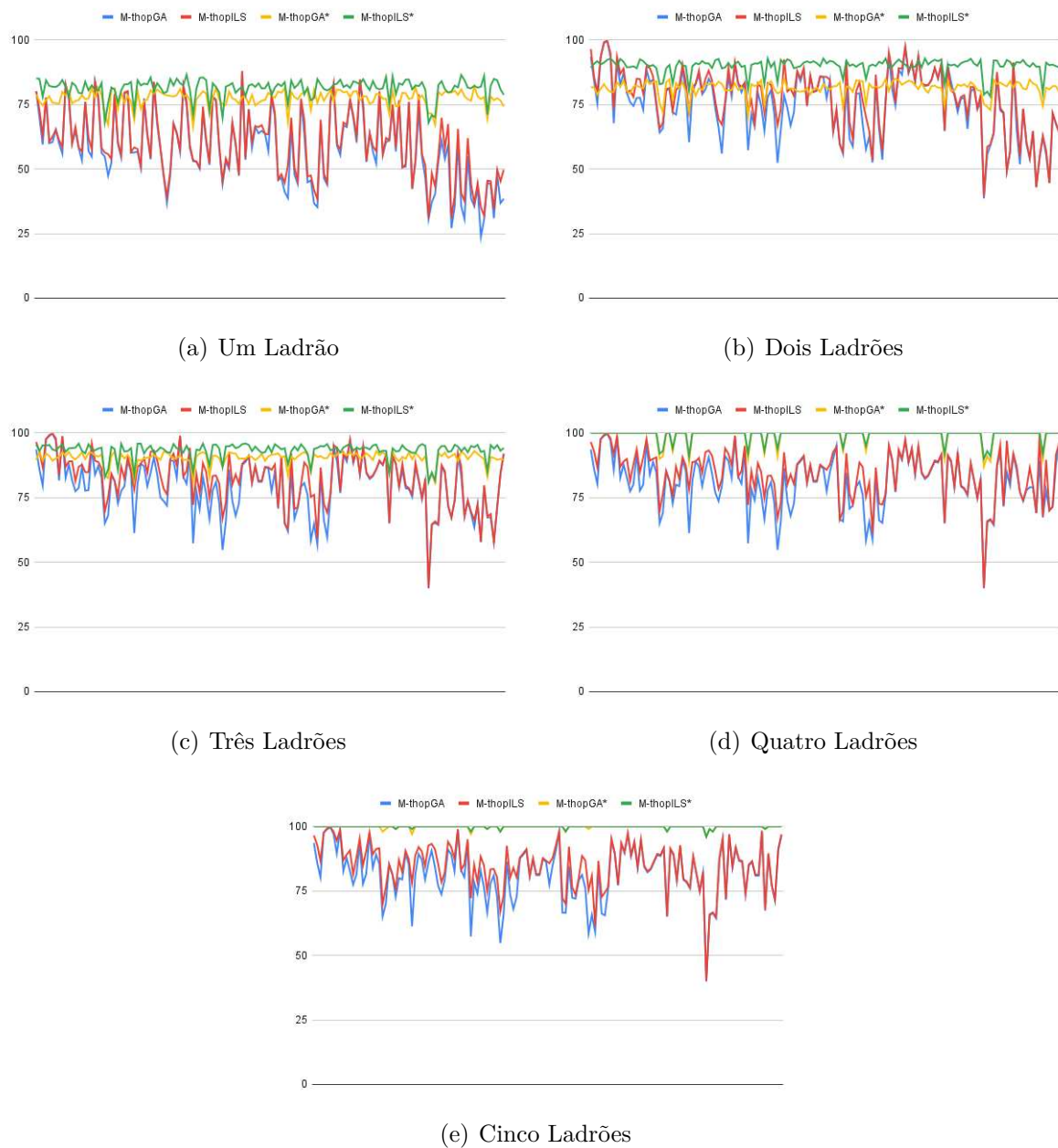


Figura 12 – Proximidade dos resultados dos métodos heurísticos com o ótimo global do problema da mochila.

Para verificar um melhor comportamento das abordagens heurísticas com instância mais competitivas para o problema, criamos instâncias que reduzem o tempo de conclusão de coleta de cada ladrão para 25% de referência específicos da instância definidas em (BONYADI; MICHALEWICZ; BARONE, 2013). Portanto, foram criadas 144 novas instâncias, com as mesmas características apresentadas anteriormente, mas com o limite de tempo reduzido. Os resultados são apresentados nos gráficos da Figura 13. Podemos analisar com os resultados que as abordagens heurísticas que utilizaram a estratégia de operar com os itens da solução ótima do problema da mochila continuaram melhores e a convergência para o *upper bound* diminuíram, portanto, diminuindo o tempo que o ladrão

possui para recolher os itens, passamos a ter instâncias mais competitivas para o MThOP.

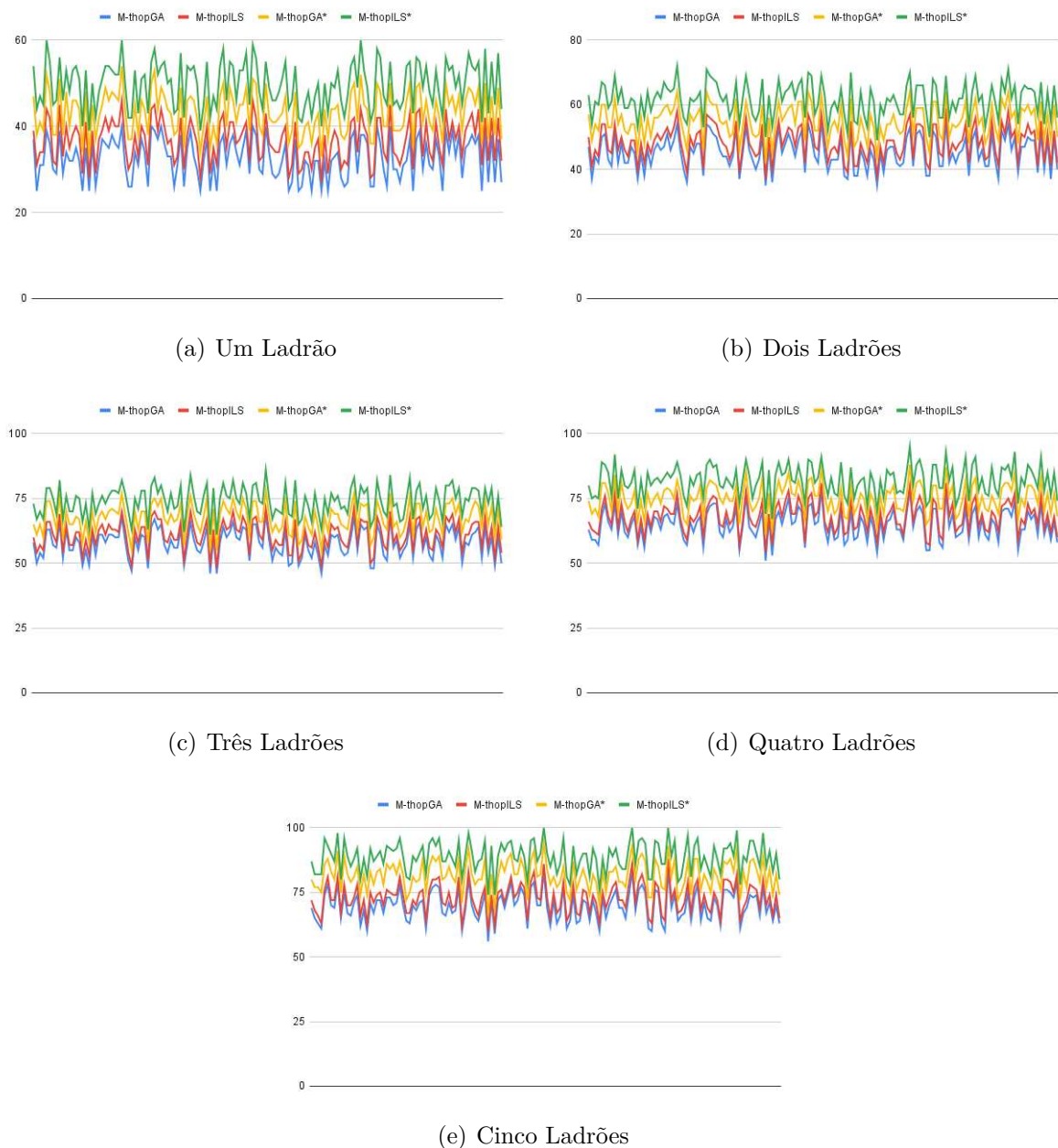


Figura 13 – Proximidade dos resultados dos métodos heurísticos com o ótimo global do problema da mochila, para as instâncias com tempo de conclusão reduzido para 25%.

Para confirmar que a estratégia de utilizar o conjunto dos itens ótimos do problema da mochila possui diferença significativa para a estratégia que utiliza todos os itens, o teste de *Friedman* foi aplicado nos resultados dos algoritmos M-thopILS e M-thopILS\*. Escolhemos o teste de *Friedman* pois não podemos afirmar que as amostras utilizadas nos testes são independentes (ZIMMERMAN; ZUMBO, 1993). Para afirmar que os resultados dos algoritmos são diferentes, a hipótese observada no teste de que os algoritmos são iguais, deve ser negada. O p-valor utilizado foi de 0.05 e o resultado encontrado é 0.014, que é

menor que 0.05, portanto concluímos que os algoritmos são diferentes com confiança de 95%.

Na figura 14 podemos observar o gráfico *boxplot* com os resultados dos experimentos. No eixo horizontal estão os algoritmos M-thopILS e M-thopILS\* apresentado nesse trabalho e na vertical a porcentagem de aproximação da solução com o *upper bound* de cada instância, incluindo as novas instâncias criadas. Visualizando o gráfico, aparentemente o algoritmo M-thopILS\* possui a média e convergência maior que o algoritmo M-thopILS, mas não é possível comprovar apenas visualmente.

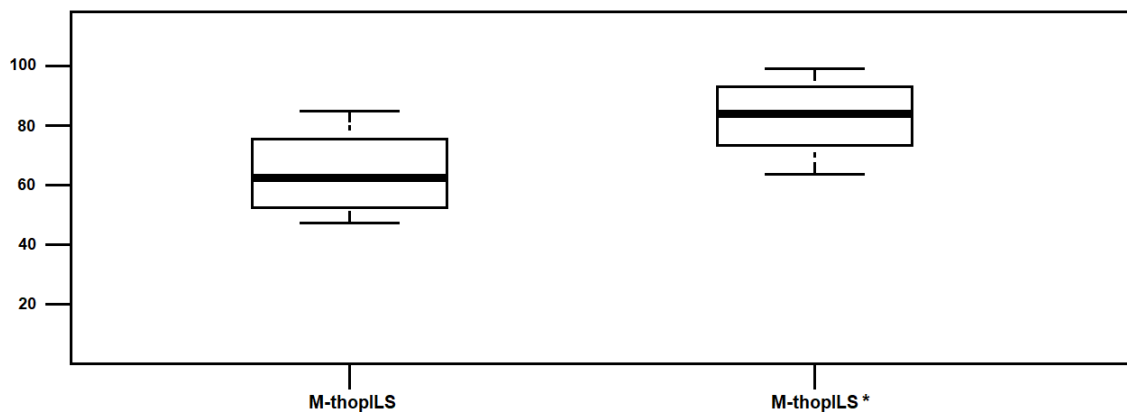


Figura 14 – Análise comparativa da convergência dos algoritmos heurísticos para todas as instâncias

Para verificar se existe realmente diferença significativa entre os algoritmos é necessário passar pelo teste de *Wilcoxon*. Nele é verificado se o algoritmo que apresentou melhores resultados visualmente possui diferença significativa em relação ao outro algoritmo proposto (GEHAN, 1965). Primeiramente, o p-valor é ajustado utilizando a correção de Bonferroni (PERNEGER, 1998). A hipótese observada no teste é que, comparando os algoritmos par-a-par, eles são iguais. O valor encontrado comparando o M-thopILS com o M-thopILS\* é 0.012, que é inferior a 0.05, portanto, podemos afirmar que os algoritmos são diferentes e que o M-thopILS\* possui melhor desempenho.

## 4.6 Conclusões

Neste artigo apresentamos o Multiple Thief Orienteering Problem, uma ampliação do Thief Orienteering Problem para vários ladrões. O MThOP foi descrito formalmente por um formulação de programação não linear mista e as abordagens heurísticas para o MThOP foram implementadas e analisadas. Testamos nossas abordagens em 432 instâncias existentes desenvolvidas para o ThOP e outras 144 novas instâncias, utilizando até cinco ladrões. Os resultados mostraram uma melhor convergência das abordagens heurísticas que utilizam a estratégia de operar apenas com os itens da solução ótima do problema

da mochila, com aumento de até 32% do lucro. As abordagens heurísticas que utilizaram o algoritmo thopILS obtiveram os melhores resultados para o MThOP. Como trabalhos futuros, gostaríamos de analisar o desempenho de heurísticas que realizam o processo de seleção de itens e rotas simultaneamente e a utilização de outras metaheurísticas.

## 5 Conclusões Gerais e Trabalhos Futuros

Nesta dissertação foi abordado o *Thief Orienteering Problem*, um problema multicomponente que combina características de dois problemas de otimização combinatória: o problema de orientação e o problema da mochila. No ThOP, deve-se determinar uma rota e itens a serem coletados ao longo do caminho, com restrições de tempo máximo de percurso, capacidade da mochila e velocidade dependente do peso da mochila, o que torna o ThOP um problema complexo e desafiador. Propomos duas abordagens heurísticas para o ThOP, uma inspirado em algoritmos genéticos (thopGA) e outra na metaheurística de busca local iterada (thopILS).

Nas duas abordagens foi utilizada uma sub-rotina de busca local que explora a vizinhança de uma solução atual para buscar um vizinho de melhor qualidade. Se uma solução melhor for encontrada, essa solução passa a ser a solução atual e esse processo é repetido até que atinja o critério de parada, assim retornando uma solução melhorada. O processo de busca local possui uma estrutura simples que consiste em encontrar uma rota melhor para coletar os itens da solução, assim diminuindo o tempo gasto para recolher os itens. Posteriormente, novos itens são inseridos na rota, assim aumentando o lucro da mochila.

As abordagens propostas foram testadas com as instâncias de *benchmark* e superaram os algoritmos existentes em grande parte das instâncias. Comparando as duas abordagens, o thopILS obteve melhor desempenho em mais de 80% da instância, com uma melhoria média de mais de 30%. Com base em nossa análise, isso se deve à eficiência da otimização da busca local utilizada.

O *Multiple Thief Orienteering Problem* foi apresentado e descrito formalmente por um formulação de programação não linear mista. Esse problema é uma ampliação do ThOP que considera múltiplos ladrões trabalhando em conjunto. Os algoritmos (thopGA e thopILS) propostos para o ThOP foram utilizados para gerar soluções iniciais para o MThOP. Os resultados foram analisados e mostrou uma superioridade do abordagem que utilizou o thopILS. Também foi verificado que o uso de mais ladrões, combinado com a estratégia de utilizar apenas os itens da solução ótima do problema da mochila, aumenta o lucro total em até 32%. Entretanto, a partir de 5 ladrões não há aumento significativo no lucro, exigindo novas instâncias apropriadas para o problema.

### 5.1 Trabalhos Futuros

Como trabalho futuro, gostaríamos de investigar algoritmos exatos para resolver instâncias ThOP pequenas e médias para testar a eficiências dos algoritmos heurísticas

desenvolvidos para resolver instâncias ThOP e estabelecer ótimos globais. Além de pesquisar o desempenho de heurísticas para o MThOP que realizam a construção das soluções dos ladrões simultaneamente e a utilização de outras metaheurísticas.

# Referências

- ARAUJO, R. P. et al. A novel list-constrained randomized vnd approach in gpu for the traveling thief problem. **Electronic Notes in Discrete Mathematics**, Elsevier, v. 66, p. 183–190, 2018.
- BEASLEY, J. E. Or-library: distributing test problems by electronic mail. **Journal of the operational research society**, Taylor & Francis, v. 41, n. 11, p. 1069–1072, 1990.
- BONYADI, M. R.; MICHALEWICZ, Z.; BARONE, L. The travelling thief problem: The first step in the transition from theoretical problems to realistic problems. In: IEEE. **2013 IEEE Congress on Evolutionary Computation**. [S.l.], 2013. p. 1037–1044.
- CEC. **Optimisation of Problems with Multiple Interdependent Components**. 2014. Disponível em: <<https://cs.adelaide.edu.au/optlog/CEC2014Comp>>. Acesso em: 01-12-2019.
- CHAGAS, J. B.; WAGNER, M. Ants can orienteer a thief in their robbery. **Operations Research Letters**, Elsevier, v. 48, n. 6, p. 708–714, 2020.
- CHAND, S.; WAGNER, M. Fast heuristics for the multiple traveling thieves problem. In: **Proceedings of the Genetic and Evolutionary Computation Conference 2016**. [S.l.: s.n.], 2016. p. 293–300.
- CHAO, I.-M.; GOLDEN, B. L.; WASIL, E. A. A fast and effective heuristic for the orienteering problem. **European journal of operational research**, Elsevier, v. 88, n. 3, p. 475–489, 1996.
- CHAO, I.-M.; GOLDEN, B. L.; WASIL, E. A. The team orienteering problem. **European journal of operational research**, Elsevier, v. 88, n. 3, p. 464–474, 1996.
- DARWIN, C. **The Origin of Species: By means of natural selection or the preservation of favoured races in the struggle for life**. [S.l.], 1956.
- FAÊDA, L. M.; SANTOS, A. G. A genetic algorithm for the thief orienteering problem. In: IEEE. **2020 IEEE Congress on Evolutionary Computation (CEC)**. [S.l.], 2020a. p. 1–8.
- FAÊDA, L. M.; SANTOS, A. G. Busca local iterada aplicada ao thief orienteering problem. In: SBPO. **LII Simpósio Brasileiro de Pesquisa Operacional (SBPO)**. [S.l.], 2020b.
- GAREY, M. R.; JOHNSON, D. S. A guide to the theory of np-completeness. **Computers and intractability**, p. 641–650, 1979.
- GECCO. **“Optimisation of Problems with Multiple Interdependent Components**. 2017. Disponível em: <<http://gecco-2017.sigevo.org/index.html/Competitions>>. Acesso em: 01-12-2019.
- GEHAN, E. A. A generalized wilcoxon test for comparing arbitrarily singly-censored samples. **Biometrika**, Oxford University Press, v. 52, n. 1-2, p. 203–224, 1965.

- GOLDEN, B. L.; LEVY, L.; VOHRA, R. The orienteering problem. **Naval Research Logistics (NRL)**, Wiley Online Library, v. 34, n. 3, p. 307–318, 1987.
- GUNAWAN, A.; LAU, H. C.; VANSTEENWEGEN, P. Orienteering problem: A survey of recent variants, solution approaches and applications. **European Journal of Operational Research**, Elsevier, v. 255, n. 2, p. 315–332, 2016.
- HOLLAND, J. H. Genetic algorithms. **Scientific american**, JSTOR, v. 267, n. 1, p. 66–73, 1992.
- LIN, S.; KERNIGHAN, B. W. An effective heuristic algorithm for the traveling-salesman problem. **Operations research**, INFORMS, v. 21, n. 2, p. 498–516, 1973.
- LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search. In: **Handbook of metaheuristics**. [S.l.]: Springer, 2003. p. 320–353.
- LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search: Framework and applications. In: **Handbook of metaheuristics**. [S.l.]: Springer, 2019. p. 129–168.
- MICHALEWICZ, Z. Quo vadis, evolutionary computation? In: SPRINGER. **IEEE World Congress on Computational Intelligence**. [S.l.], 2012. p. 98–121.
- NAMAZI, M. et al. A profit guided coordination heuristic for travelling thief problems. In: **Twelfth Annual Symposium on Combinatorial Search**. [S.l.: s.n.], 2019.
- PERNEGER, T. V. What’s wrong with bonferroni adjustments. **Bmj**, British Medical Journal Publishing Group, v. 316, n. 7139, p. 1236–1238, 1998.
- PIRKUL, H. A heuristic solution procedure for the multiconstraint zero-one knapsack problem. **Naval Research Logistics (NRL)**, Wiley Online Library, v. 34, n. 2, p. 161–172, 1987.
- POLYAKOVSKIY, S. et al. A comprehensive benchmark set and heuristics for the traveling thief problem. In: **Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation**. [S.l.: s.n.], 2014. p. 477–484.
- POLYAKOVSKIY, S. et al. A comprehensive benchmark set and heuristics for the traveling thief problem. In: **Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation**. [S.l.: s.n.], 2014. p. 477–484.
- REINELT, G. Tsplib—a traveling salesman problem library. **ORSA journal on computing**, INFORMS, v. 3, n. 4, p. 376–384, 1991.
- ROTHLAUF, F. **Design of modern heuristics: principles and application**. [S.l.]: Springer Science & Business Media, 2011.
- SANTOS, A. G.; CHAGAS, J. B. The thief orienteering problem: Formulation and heuristic approaches. In: IEEE. **2018 IEEE Congress on Evolutionary Computation (CEC)**. [S.l.], 2018. p. 1–9.
- VANSTEENWEGEN, P.; SOUFFRIAUX, W.; OUDHEUSDEN, D. V. The orienteering problem: A survey. **European Journal of Operational Research**, Elsevier, v. 209, n. 1, p. 1–10, 2011.

- VIEIRA, D. K. et al. A genetic algorithm for multi-component optimization problems: the case of the travelling thief problem. In: SPRINGER. **European Conference on Evolutionary Computation in Combinatorial Optimization**. [S.l.], 2017. p. 18–29.
- WAGNER, M. Stealing items more efficiently with ants: a swarm intelligence approach to the travelling thief problem. In: SPRINGER. **International Conference on Swarm Intelligence**. [S.l.], 2016. p. 273–281.
- WAGNER, M. et al. A case study of algorithm selection for the traveling thief problem. **Journal of Heuristics**, Springer, v. 24, n. 3, p. 295–320, 2018.
- WU, J. et al. Exact approaches for the travelling thief problem. In: SPRINGER. **Asia-Pacific Conference on Simulated Evolution and Learning**. [S.l.], 2017. p. 110–121.
- YAFRANI, M. E.; AHIOD, B. Efficiently solving the traveling thief problem using hill climbing and simulated annealing. **Information Sciences**, Elsevier, v. 432, p. 231–244, 2018.
- ZIMMERMAN, D. W.; ZUMBO, B. D. Relative power of the wilcoxon test, the friedman test, and repeated-measures anova on ranks. **The Journal of Experimental Education**, Taylor & Francis, v. 62, n. 1, p. 75–86, 1993.