57º SBPO
SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL
Gramado • RS
DE 5 A 9 DE OUTUBRO DE 2025

Análise de Variantes da Metaheurística Greedy Randomized Adaptive Search Procedure Aplicadas ao Problema do Ladrão Viajante

Giovane de Morais[1], Felipe Miranda de Oliveira[2], Antonio Augusto Chaves[2]

[1] Doutorando em Engenharia Aeronáutica e Mecânica, Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos, São Paulo, Brasil.
E-mail: giovane.morais.101868@ga.ita.br
[2] Mestrando em Pesquisa Operacional, Universidade Federal de São Paulo (UNIFESP), São José dos Campos, São Paulo, Brasil.
E-mail: fmoliveira13@unifesp.br
[2] Docente, Universidade Federal de São Paulo (UNIFESP), São José dos Campos, São Paulo, Brasil.
E-mail: antonio.chaves@unifesp.br

RESUMO

Este trabalho propõe seis variantes da metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*) aplicadas ao Problema do Ladrão Viajante, que integra o Problema do Caixeiro Viajante e o Problema da Mochila. As variantes diferem pela parametrização (reativa ou fixa), pela intensificação (*Path Relinking*) e diversificação (*Restart*). Experimentos em 18 instâncias, com 30 execuções e dois tempos-limite (600 e 3600 segundos), avaliaram os resultados médios, desvio padrão, intervalo de confiança e aplicaram testes de Friedman e Nemenyi. Resultados mostram que o tempo de execução impacta significativamente, com ganhos superiores a 10% em instâncias complexas. A variante GRASP reativo com *Path Relinking* e *Restart* obteve os melhores resultados, mostrando que a combinação de reatividade, intensificação e diversificação melhora a qualidade das soluções para o problema abordado, principalmente em instâncias de maior porte.

PALAVRAS CHAVE. Problema do Ladrão Viajante, GRASP Reativo, Path Relinking.

MH – Metaheurísticas, OD – Otimização Discreta

ABSTRACT

This work proposes six variants of the GRASP (Greedy Randomized Adaptive Search Procedure) metaheuristic applied to the Travelling Thief Problem, which integrates the Travelling Salesman Problem and the Knapsack Problem. The variants differ in their parametrization (reactive or fixed), intensification mechanism (Path Relinking), and diversification mechanism (Restart). Experiments were conducted on 18 instances, with 30 runs each and two time limits (600 and 3600 seconds), evaluating average results, standard deviation, confidence intervals, and applying the Friedman and Nemenyi tests. The results show that execution time significantly impacts solution quality, with improvements exceeding 10% in complex instances. The reactive GRASP variant with Path Relinking and Restart achieved the best results, indicating that the combination of reactivity, intensification, and diversification improves solution quality for the addressed problem, particularly in large-scale instances.

KEYWORDS. Travelling Thief Problem, Reactive GRASP, Path Relinking.

MH – Metaheuristics, OA – Discrete Optimization

DOI: 10.59254/sbpo-2025-212146
https://proceedings.science/p/212146?lang=pt-br

1. Introdução

O Problema do Ladrão Viajante (TTP, do inglês *Travelling Thief Problem*), introduzido por (Bonyadi e Michalewicz, 2013), combina dois problemas clássicos e NP-difíceis: o Problema do Caixeiro Viajante (TSP) (Croes, 1958) e o Problema da Mochila (KP) (Martello e Toth, 1990). No TTP, a seleção de itens influencia diretamente o peso transportado, que por sua vez afeta a velocidade de deslocamento e o tempo total de viagem, estabelecendo uma forte interdependência entre as decisões de roteamento e coleta (Polyakovskiy et al., 2014). Essa complexidade reflete diversos cenários reais de logística, onde o peso transportado impacta o consumo de combustível, o tempo de operação e os custos envolvidos (Rodriguez e Smith, 2023). Devido à dificuldade do problema, métodos exatos, como programação inteira e *branch-and-bound*, são aplicáveis apenas a instâncias de pequeno porte, o que torna necessária a adoção de metaheurísticas para tratar instâncias de maior escala.

Este estudo compara seis variantes de uma metaheurística bem consolidada para a resolução do TTP: o Greedy Randomized Adaptive Search Procedure (GRASP), explorando diferentes combinações dos mecanismos de reatividade, Path Relinking e Restart. As variantes analisadas incluem: (i) GRASP Reativo com Path Relinking e Restart; (ii) GRASP Tradicional com Path Relinking; (iii) GRASP Tradicional com Path Relinking e Restart; (iv) GRASP Reativo com Restart; e (v) GRASP Tradicional com Restart. A comparação entre essas abordagens visa avaliar o impacto individual e combinado desses mecanismos sobre a eficácia da metaheurística.

Essas abordagens serão comparadas com o GRASP clássico. A hipótese central é que a inclusão de reatividade, *Path Relinking* e *Restart* melhora a exploração global e local da busca, gerando soluções de maior qualidade. As métricas de avaliação incluem gap percentual em relação à melhor solução conhecida e tempo de execução.

Este trabalho está organizado da seguinte forma. A seção 2 apresenta o modelo matemático do TTP. A seção 3 descreve a metaheurística GRASP e as variações propostas. A seção 4 apresenta os resultados computacionais e a seção 5 apresenta a análise estatística. Na seção 6 fazem-se as considerações finais.

2. Modelagem Matemática

O problema *Travelling Thief Problem* (TTP) busca maximizar o lucro líquido combinando dois problemas clássicos: o Problema do Caixeiro Viajante (TSP) e o Problema da Mochila (KP). A complexidade do TTP está no fato de que a velocidade de deslocamento depende do peso acumulado na mochila, o que gera acoplamento entre as decisões de rota e de coleta de itens.

Definições

Seja $G = (V, E)$ um grafo completo, onde: $V = \{0, 1, \ldots, n\}$ representa as cidades (a cidade 0 é o ponto de origem e retorno) e E é o conjunto de arestas com distâncias d_{ij} entre as cidades i e j.

Cada cidade $i \in V$ pode conter um conjunto de itens I_i. Para cada item $k \in I_i$, temos: valor $p_k > 0$ e peso $w_k > 0$.

Parâmetros Globais

- W: capacidade máxima da mochila,
- $v_{\min}, v_{\max}$: velocidades mínima e máxima de deslocamento,
- R: taxa de aluguel por unidade de tempo.

Variáveis de Decisão

- $x_{ij} \in \{0, 1\}$: 1 se a rota vai da cidade i para a cidade j,
- $z_k \in \{0, 1\}$: 1 se o item k é coletado,
- $u_i \in [1, n]$: posição da cidade i no tour (para eliminação de subtours).

DOI: 10.59254/sbpo-2025-212146
https://proceedings.science/p/212146?lang=pt-br

Função Objetivo

O objetivo é maximizar o lucro líquido, dado pela diferença entre o valor dos itens coletados e o custo do tempo de viagem:

$$\max \left(\sum_{k \in I} p_k z_k - R \cdot T \right)$$

O tempo total T é calculado como:

$$T = \sum_{(i,j) \in E} \frac{d_{ij}}{v_{ij}}$$

A velocidade v_{ij} ao sair da cidade i depende do peso transportado até aquele ponto:

$$v_{ij} = v_{\max} - \left(\frac{v_{\max} - v_{\min}}{W} \right) \cdot w_i$$

Onde w_i representa o peso acumulado ao sair da cidade i, isto é:

$$w_i = \sum_{\substack{k \in I \\ \text{coletado antes de } i}} w_k z_k$$

Cujas restrições aplicadas, conforme tabela, são:

Tabela 1: Restrições do modelo matemático do TTP

Restrição	Índices / Âmbito	Descrição
$\sum_{j \neq i} x_{ij} = 1$	$\forall i \in V$	Cada cidade possui exatamente uma saída
$\sum_{i \neq j} x_{ij} = 1$	$\forall j \in V$	Cada cidade possui exatamente uma entrada
$\sum_{k \in I} w_k z_k \leq W$	global	Capacidade da mochila
$z_k \leq \sum_{j \in V} x_{ij}$	$\forall k \in I_i,\ \forall i \in V$	Só coleta se visitar a cidade
$x_{00} = 0, \quad z_k = 0$	$\forall k \in I_0$	Nada é coletado na cidade de origem
$u_i - u_j + n x_{ij} \leq n - 1$	$\forall i \neq j,\ i, j \in V \setminus \{0\}$	Eliminação de subtours (MTZ)

O modelo MTZ (Miller–Tucker–Zemlin) é uma formulação clássica proposta por (Miller et al., 1960) que, neste contexto do TTP, introduz variáveis auxiliares u_i à ordem de visita das cidades. A principal ideia é impedir ciclos parciais (subtours) que não passam por todas as cidades, impondo restrições do tipo $u_i - u_j + n x_{ij} \leq n - 1$, onde x_{ij} indica a escolha do arco de i para j, e n representa o número total de cidades.

3. Greedy Randomized Adaptive Search Procedure

O algoritmo GRASP (*Greedy Randomized Adaptive Search Procedure*) (Feo e Resende, 1995) é uma meta-heurística construtiva iterativa composta por duas fases principais: construção gulosa aleatorizada de soluções iniciais e aplicação de busca local para refinamento. Cada iteração do GRASP gera uma solução viável e potencialmente ótima local. Ao final de todas as iterações, a melhor solução global encontrada é retornada.

A construção da solução é guiada por uma função gulosa que ordena os candidatos com base em um critério de avaliação. Para introduzir diversidade, utiliza-se uma Lista de Candidatos Restrita (RCL), composta pelos melhores candidatos conforme um limiar determinado pelo parâmetro $\alpha \in [0, 1]$. Em cada passo, um candidato é selecionado aleatoriamente da RCL, balanceando aleatoriedade e qualidade.

A busca local atua sobre a solução gerada, explorando sua vizinhança até atingir um ótimo local. A cada iteração do GRASP, a solução encontrada é comparada à melhor global e, se superior, atualiza o registro da solução ótima. Para problemas de alta complexidade como o TTP, técnicas adicionais são incorporadas para potencializar a exploração e a intensificação da busca.

DOI: 10.59254/sbpo-2025-212146
https://proceedings.science/p/212146?lang=pt-br

Extensões para o TTP

Neste trabalho, três mecanismos complementares foram integrados ao GRASP clássico, com o objetivo de lidar com a complexidade acoplada do TTP: seleção reativa do parâmetro α, intensificação por meio de *Path Relinking*, e reinicialização adaptativa. Essas estratégias têm respaldo na literatura como formas eficazes de melhorar a convergência e a robustez de meta-heurísticas em problemas combinatórios complexos (Resende e Ribeiro, 2010).

Seleção Reativa de α: em vez de manter um valor fixo para α, utiliza-se um conjunto de valores candidatos $\mathscr{A} = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$. A cada iteração, escolhe-se um α_i com base em uma roleta adaptativa, cujos pesos w_{α_i} são atualizados segundo o desempenho histórico das soluções geradas com aquele valor. A atualização é feita por:

$$w_{\alpha_i} \leftarrow w_{\alpha_i} \cdot \left(1 + \rho \cdot \frac{\bar{f}_{\alpha_i} - f_{min}}{f_{max} - f_{min}}\right)$$

onde $\bar{f}_{\alpha_i}$ é o valor médio das soluções com α_i, f_{min} e f_{max} representam os valores extremos observados, e ρ é o parâmetro de aprendizado. Essa técnica foi inspirada por mecanismos de autoajuste como os discutidos em (Prazeres e Ribeiro, 2017).

***Path Relinking*:** trata-se de um método de intensificação que explora trajetórias entre uma solução corrente S e uma solução guia S', retirada de um conjunto elite $\mathcal{E}$ (Glover e Laguna, 1997). Um conjunto de diferenças Δ é identificado entre S e S', e modificações incrementais são aplicadas a S para aproximá-la de S'. Cada solução intermediária é avaliada, e a melhor é preservada. No caso do TTP, a técnica é aplicada prioritariamente à parte TSP da solução, com foco em cidades onde há itens coletados.

Reinicialização Adaptativa: caso a solução ótima permaneça inalterada por mais de t_r segundos (critério de estagnação temporal), o algoritmo reinicializa os pesos w_α, o conjunto elite $\mathcal{E}$ e variáveis auxiliares. Isso permite escapar de regiões saturadas do espaço de busca e restaurar a diversidade, evitando convergência prematura.

A seguir, apresenta-se a versão estendida da metaheurística GRASP, que incorpora reatividade, *Path Relinking* e reinicialização adaptativa, adaptada para a resolução do TTP.

Algoritmo 1: GRASP Reativo com *Path Relinking* e Reinicialização Adaptativa

Entrada: Grafo $G = (V, E)$, itens I, capacidade da mochila W, conjunto de candidatos $\mathscr{A}$, variabilidade aleatória c, tempo de reinicialização t_r, tamanho do conjunto elite g, tempo limite total T_{max}
Saída: Melhor solução S_{best}
Inicializar $S_{best} \leftarrow \emptyset$, $f_{\text{best}} \leftarrow -\infty$;
Inicializar pesos w_α e conjunto elite $\mathcal{E}$;
while *tempo decorrido* $< T_{max}$ **do**
  Selecionar $\alpha \in \mathscr{A}$ via roleta adaptativa;
  Construir solução S com RCL parametrizada por α;
  if $|\mathcal{E}| \geq g$ **then**
    Selecionar guia $S' \in \mathcal{E}$;
    Aplicar *Path Relinking* entre S e S';
  end
  Aplicar busca local em S (TSP com Two-Opt; KP com Bit Flip);
  Atualizar S_{best}, pesos w_α e conjunto $\mathcal{E}$;
  if *tempo sem melhoria* $> t_r$ **then**
    Reinicializar pesos w_α e conjunto elite $\mathcal{E}$;
  end
end
return S_{best};

Análises empíricas sugerem que a aplicação combinada dessas estratégias resulta em maior robustez frente à variabilidade de instâncias do TTP, como reportado por (Polyakovskiy et al., 2014; Faulkner et al., 2015).

3.1. Trabalhos Relacionados

O *Travelling Thief Problem* (TTP) foi proposto para estudar a interdependência entre subproblemas combinatórios clássicos (Bonyadi e Michalewicz, 2013). A primeira suíte abrangente de benchmarks, com 9.720 instâncias, foi apresentada por Polyakovskiy et al. (2014), tornando-se o

DOI: 10.59254/sbpo-2025-212146
https://proceedings.science/p/212146?lang=pt-br

padrão de comparação para a área. Diversas heurísticas e metaheurísticas foram exploradas desde então, incluindo abordagens aproximadas e baseadas em seleção de algoritmos (Faulkner et al., 2015; Wagner et al., 2018), heurísticas de busca local e variantes eficientes (El Yafrani e Ahiod, 2018), além de métodos exatos e híbridos (Wu et al., 2018). Trabalhos recentes investigam coordenação explícita entre as componentes TSP/KP e aprendizagem (Namazi et al., 2023). Esta base motiva as variantes de GRASP avaliadas aqui, focadas em reatividade, *Path Relinking* e *Restart*.

Os valores f^* utilizados no cálculo do RPD foram coletados do conjunto de desempenho consolidado pela OptLog (University of Adelaide), que compila 21 algoritmos sobre as 9.720 instâncias originais do TTP, bem como do estudo de seleção de algoritmos em Wagner et al. (2018). Quando um valor não estava disponível nesse repositório, adotamos o melhor valor reportado nos artigos correspondentes às instâncias avaliadas (e.g., Faulkner et al., 2015; El Yafrani e Ahiod, 2018; Wu et al., 2018). As referências específicas por instância constam no abaixo.

Selecionamos 18 instâncias por amostragem estratificada (tamanho do TSP e tipo de distribuição BSC/USW/U), visando representar cenários com diferentes níveis de acoplamento e dificuldade computacional. A suíte completa do TTP contém 9.720 instâncias (Polyakovskiy et al., 2014); por limitações de tempo de processamento e espaço editorial, optamos por este recorte. Como ameaça à validade externa, reconhecemos que conclusões podem mudar em outras faixas de tamanho/distribuição; mitigamos o risco reportando variabilidade (desvios/IC95%) e testes não-paramétricos (Friedman/Nemenyi). Um estudo futuro pode ampliar para subconjuntos adicionais da suíte (p.ex., mais níveis de capacidade, outras bases TSP).

4. Modelagem do Experimento

Os testes foram conduzidos em um ambiente virtualizado por meio da plataforma VMware vSphere 8.0, utilizando o sistema operacional Ubuntu 22.04.5 LTS. A infraestrutura computacional consistiu em uma CPU Intel Xeon Platinum 8473C (com 12 núcleos virtuais, frequência de 1,7GHz e 24 threads) e 24GB de memória RAM. Os scripts do experimento foram desenvolvidos na linguagem de programação Python, versão 3.12. Foram consideradas 18 instâncias do problema conhecido como *Travelling Thief Problem* (TTP).

As instâncias utilizadas neste trabalho seguem a convenção de nome `#cXXX#iYYY`, na qual `#cXXX` indica a quantidade de cidades e `#iYYY` representa a quantidade de itens disponíveis, bem como o tipo de distribuição associada. Três categorias de distribuição são consideradas: `BSC` (Bounded Strongly Correlated), em que os valores dos itens são fortemente correlacionados com seus respectivos pesos, de modo que itens mais valiosos tendem a ser mais pesados; `USW` (Uncorrelated Similar Weights), caracterizada por itens com valores não correlacionados e pesos semelhantes entre si; e `U` (Uncorrelated), que contempla itens cujos valores e pesos são distribuídos de forma independente, sem qualquer correlação entre essas variáveis.

A Tabela 2 apresenta as 18 instâncias empregadas, baseadas no conjunto de benchmarks proposto por (Polyakovskiy et al., 2014; Mei et al., 2014).

Tabela 2: Instâncias testadas pela metaheurística proposta

Código das instâncias TTP		
51EIL50BSC	51EIL50USW	51EIL50U
51EIL150BSC	**51EIL150USW**	51EIL150U
51EIL250BSC	51EIL250USW	51EIL250U
152PR151BSC	152PR151USW	**152PR151U**
152PR453BSC	**152PR453USW**	152PR453U
280A279BSC	280A279USW	280A279U

DOI: 10.59254/sbpo-2025-212146
https://proceedings.science/p/212146?lang=pt-br

Cinco instâncias (em negrito) foram utilizadas na calibração, com 36 configurações resultantes da combinação dos parâmetros descritos na Tabela 3.

Tabela 3: Parâmetros utilizados na calibração da metaheurística

Parâmetros	Definição	Valores
$\mathscr{A}$	Roleta Alpha	3 níveis de granularidade
g	Estratégia PR	0: melhor, 1: pior, 2: alternado
$n_{\mathscr{E}}$	Tamanho do pool	10, 20
c	Aleatoriedade na construção	Sim / Não
$\mathscr{B}$	Roleta Beta	(1, 2, 3, 5, 8)
t_r	Tempo de Restart (s)	150

Tabela 4: Heurísticas GRASP utilizadas

Código	Estratégia
H1	Reativo + PR + Restart
H2	Tradicional + PR (sem Restart)
H3	Tradicional + PR + Restart
H4	Reativo + Restart (sem PR)
H5	Tradicional + Restart (sem PR)
H6	Tradicional puro (sem PR, sem Restart)

4.1. Cálculo de Desempenho do Desvio Percentual Relativo

A avaliação do desempenho das soluções obtidas foi realizada por meio do Desvio Percentual Relativo (*Relative Percentage Deviation*, RPD), métrica que indica o quanto a solução encontrada se distancia da melhor solução conhecida para uma determinada instância.

Seja f_i^* o melhor valor conhecido para a instância i, e $f_{i,h}^{(\ell)}$ o valor obtido pela heurística h na execução ℓ. O RPD é calculado da seguinte forma:

$$RPD_{i,h}^{(\ell)} = \frac{f_{i,h}^{(\ell)} - f_i^*}{f_i^*} \cdot 100\%$$

Para cada instância, são calculadas a média e o desvio padrão do RPD com base em 30 execuções independentes:

$$\overline{RPD}_{i,h} = \frac{1}{30}\sum_{\ell=1}^{30} RPD_{i,h}^{(\ell)}, \quad s_{i,h} = \sqrt{\frac{1}{29}\sum_{\ell=1}^{30}\left(RPD_{i,h}^{(\ell)} - \overline{RPD}_{i,h}\right)^2}$$

A média geral do desempenho da heurística h ao longo das 18 instâncias é expressa por:

$$\overline{\overline{RPD}}_h = \frac{1}{18}\sum_{i=1}^{18} \overline{RPD}_{i,h}$$

Por fim, o intervalo de confiança de 95% (IC95%) para essa média é estimado com base na distribuição t de Student com 17 graus de liberdade:

$$IC_{95\%} = \overline{\overline{RPD}}_h \pm t_{0,975} \cdot \frac{S_h}{\sqrt{18}}$$

DOI: 10.59254/sbpo-2025-212146
https://proceedings.science/p/212146?lang=pt-br

Em que S_h representa o desvio padrão amostral das médias $\overline{RPD}_{i,h}$ para a heurística h.

5. Calibração de Parâmetros

Para a calibração do GRASP, foram avaliadas 36 combinações distintas de parâmetros, envolvendo variações nos valores de controle do algoritmo (por exemplo, o número de iterações, valores de α, e tuplas de seleção). Inicialmente, cada configuração foi executada sobre um conjunto de instâncias menores, com um limite de 200 iterações, visando uma análise preliminar de desempenho e viabilidade. Posteriormente, as 18 instâncias do conjunto principal foram utilizadas para testar todas as configurações, com 10 repetições por instância-configuração, usando diferentes sementes aleatórias e execução em paralelo. O tempo médio de execução observado nessas repetições foi utilizado para definir os limites de tempo dos cenários de teste dos experimentos principais: 600 segundos e 3600 segundos.

As métricas consideradas na avaliação foram: **Gap percentual médio:** diferença percentual entre a melhor solução conhecida e a média das soluções obtidas pela configuração e **Gap percentual da melhor solução:** diferença percentual entre a melhor solução conhecida e a melhor solução encontrada pela configuração.

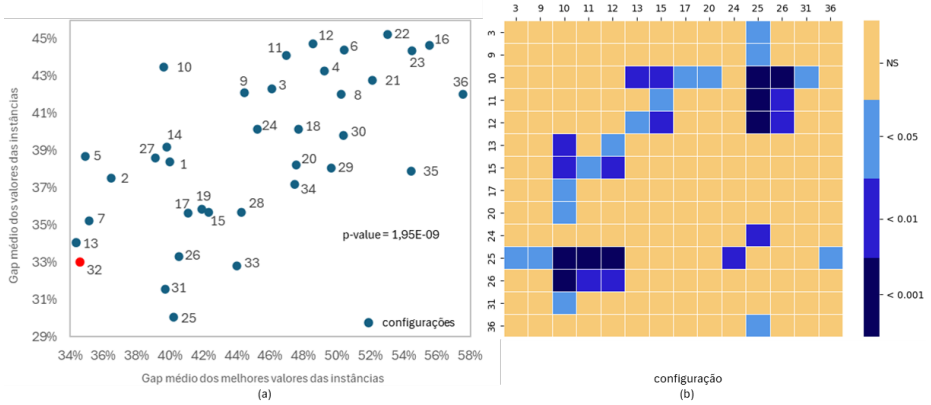


Figura 1: Comparativo das 36 configurações GRASP. (a) Gap percentual médio por configuração. (b) Heatmap de *p-values* do teste de Nemenyi entre configurações, com destaque para pares estatisticamente distintos (rejeição da hipótese nula)

A Figura 1(a) apresenta um panorama do desempenho médio de cada configuração. Já a Figura 1(b) mostra os resultados do teste de Nemenyi aplicado entre todas as combinações pareadas de configurações, onde valores baixos de *p-value* indicam diferenças estatisticamente significativas. A análise conjunta dos gráficos permitiu identificar configurações dominantes — aquelas que não apenas apresentaram os menores gaps, mas também se mostraram estatisticamente superiores às demais.

A partir dessa análise, foram selecionadas as configurações com melhor desempenho e significância estatística para compor os cenários de execução dos experimentos principais. Essas configurações servirão de base para comparar o GRASP com diferentes estratégias de pós-otimização, bem como para avaliar o impacto de cada cenário de tempo (600s e 3600s) no desempenho final.

6. Análise dos Resultados

Esta seção apresenta uma análise conjunta dos resultados obtidos pelas heurísticas H1 a H6 aplicadas ao TTP, combinando interpretações gráficas com testes estatísticos. São consideradas 18 instâncias, com diferentes tipos de distribuição e estruturas, executadas com e sem aleatoriedade (moeda), sob dois limites de tempo (600 s e 3600 s).

DOI: 10.59254/sbpo-2025-212146
https://proceedings.science/p/212146?lang=pt-br

6.1. Padrões por Tipo de Distribuição

A Figura 2 apresenta um heatmap dos valores de RPD_best, RPD_avg e desvio-padrão por instância, agrupados por tipo de distribuição. Observa-se que as instâncias do grupo BSC tendem a apresentar os menores valores médios e variâncias, sugerindo maior estabilidade e menor complexidade relativa (ex: `051EIL050BSC`, `152PR151BSC`). Em contraste, as instâncias dos grupos USW e U mostram maior variabilidade, com destaque para `152PR453U` e `280A279USW`.

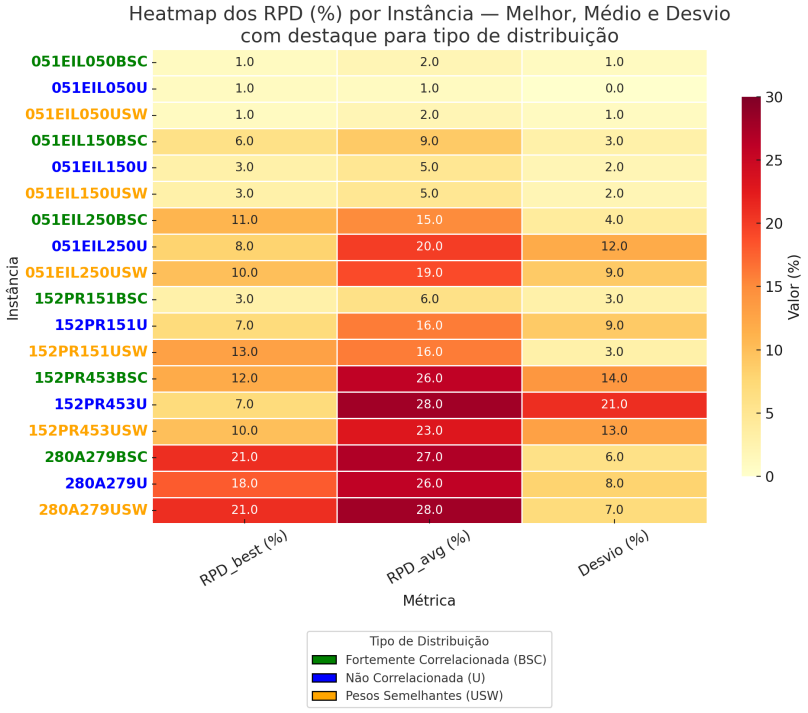


Figura 2: Heatmap dos valores de RPD_best, RPD_avg e desvio-padrão (%) por instância, agrupados por tipo de distribuição.

Ainda que o grupo U apresente a menor média de RPD, sua variabilidade interna é alta. O grupo BSC mostra comportamento mais consistente. Já o USW concentra instâncias com desempenho variado, incluindo outliers com RPD acima de 20%. Esses resultados sugerem que a classificação por tipo de distribuição não é suficiente para prever a dificuldade de resolução — características como o número de cidades, itens e estrutura da mochila também influenciam significativamente.

6.2. Efeito do Lançamento de Moeda e Tempo de Execução

A Figura 3 avalia o impacto do operador aleatório (lançamento de moeda) durante a construção. A maioria das instâncias mostra variações inferiores a 3%, mas outras como `152PR453BSC` e `152PR453U` ultrapassam 5%, indicando que a aleatoriedade pode ter impacto relevante em instâncias específicas, especialmente nas de maior porte ou estrutura irregular. Adicionalmente, observa-se que a ampliação do tempo de execução melhora os resultados na maioria das instâncias. Os maiores ganhos aparecem em `152PR453U` (+15,9%) e `280A279USW` (+10%). Já instâncias menores tendem a estabilizar suas soluções em 600 segundos.

DOI: 10.59254/sbpo-2025-212146
https://proceedings.science/p/212146?lang=pt-br

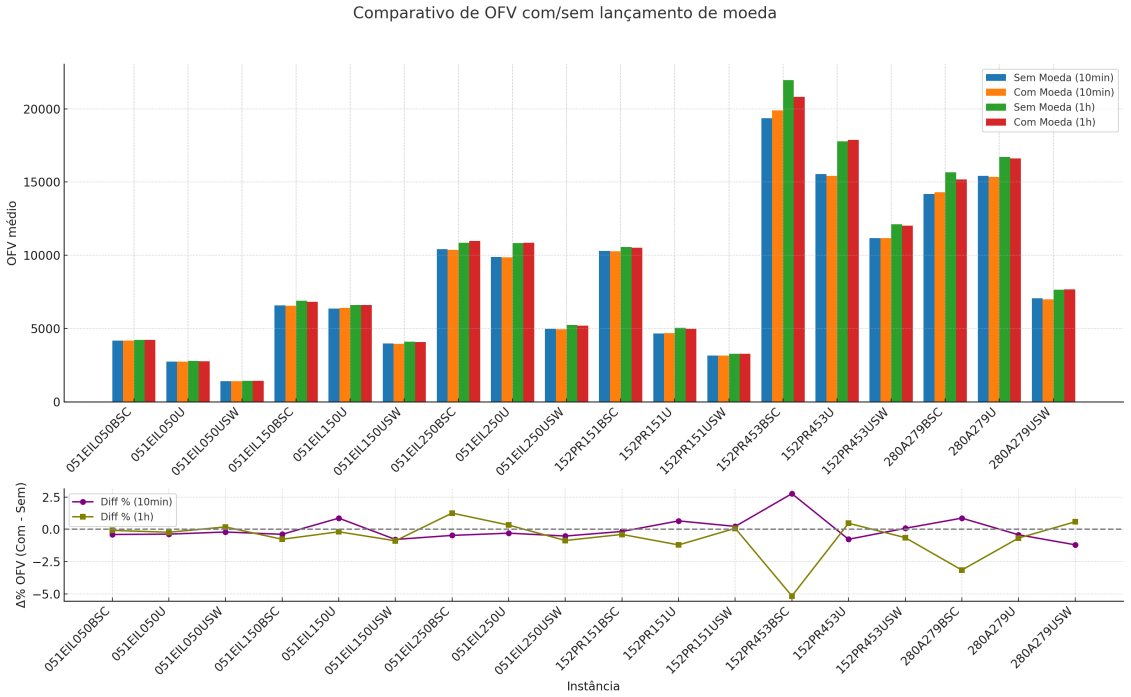


Figura 3: Comparativo do valor médio de OFV com e sem moeda, para 600 segundos e 3600 segundos.

6.3. Comparação entre Heurísticas: Desempenho e Significância

As Figuras 4 e 5 comparam o desempenho médio das heurísticas H1 a H6 com e sem moeda, nos tempos de 600 s e 3600 s. H1, que combina GRASP reativo, Path Relinking e Restart, apresenta os melhores resultados médios, especialmente em tempos mais curtos. H2 e H3 também se destacam, enquanto H4, H5 e H6 (sem Path Relinking) têm desempenho inferior.

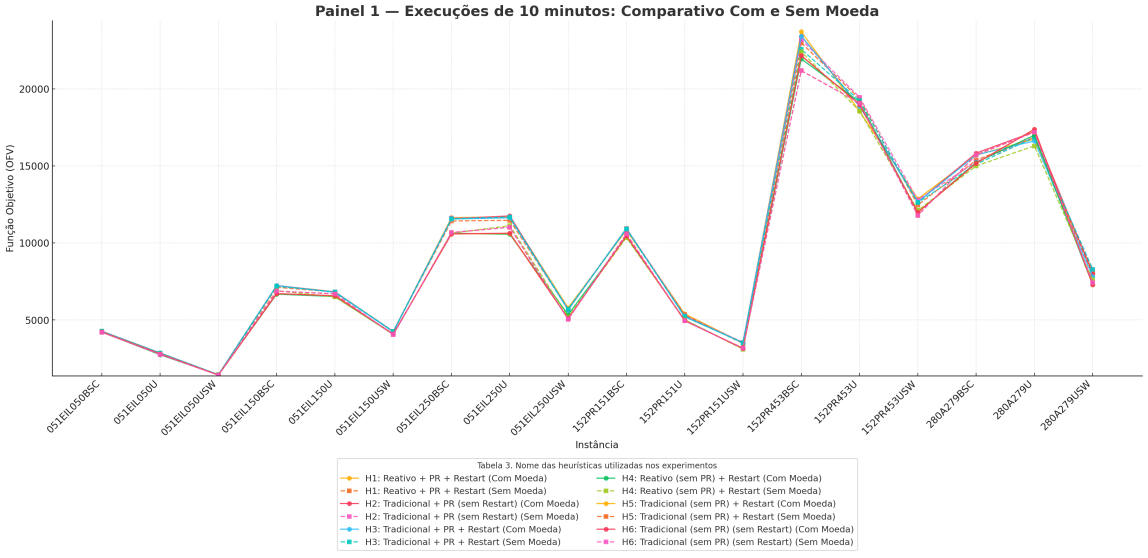


Figura 4: Comparativo das heurísticas H1 a H6 com e sem moeda — 600 segundos.

DOI: 10.59254/sbpo-2025-212146
https://proceedings.science/p/212146?lang=pt-br

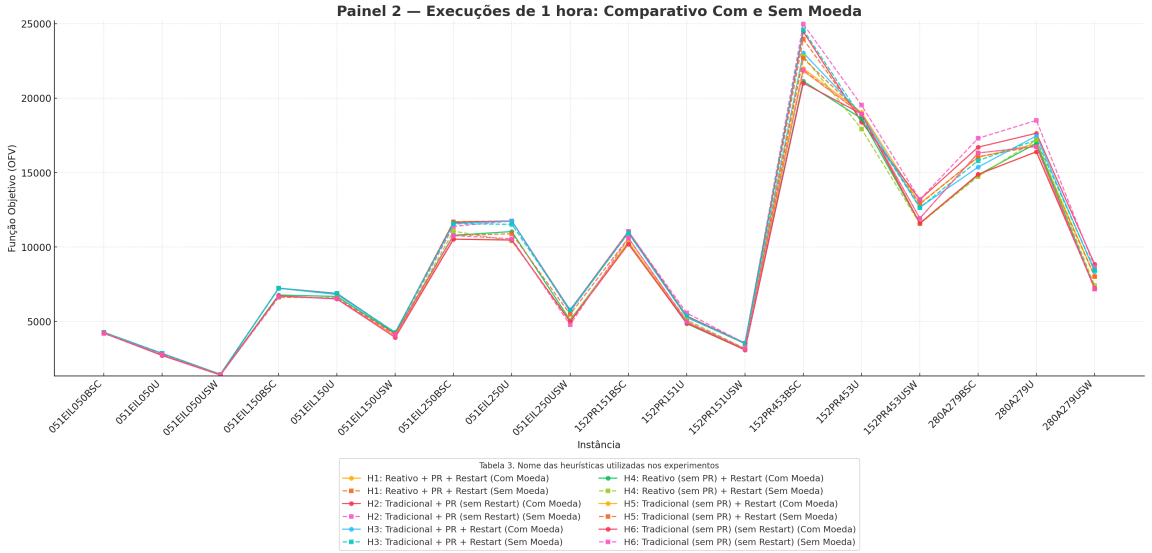


Figura 5: Comparativo das heurísticas H1 a H6 com e sem moeda — 3600 segundos.

Para validar se essas diferenças são estatisticamente significativas, realizou-se testes conforme proposto por Demšar (2006). Inicialmente, uma ANOVA univariada indicou diferenças globais significativas ($F = 3{,}12$, $p = 0{,}010$), mas foram detectadas violações das premissas de normalidade e homocedasticidade. Assim, adotou-se o teste de Friedman, adequado para dados emparelhados e não paramétricos.

6.4. Teste de Friedman

Aplicado separadamente para cada tempo-limite: **600 s:** $\chi^2_F = 18{,}76,\ p < 0{,}01$ e **3600 s:** $\chi^2_F = 20{,}85,\ p < 0{,}005$. Os resultados indicam rejeição da hipótese nula de equivalência entre heurísticas, com maiores diferenças à medida que o tempo de execução aumenta.

6.5. Teste de Nemenyi (Pós-Hoc)

A análise posterior, pelo teste de Nemenyi, revelou que a única diferença estatisticamente significativa a 5% foi entre H1 e H4. O valor da Diferença Crítica (CD) foi de aproximadamente 1,94.

6.6. Ranks Médios e Interpretação

Com base nos ranks médios, observa-se que:

Tabela 5: Ranks médios das heurísticas no teste de Friedman

Heurística	Rank Médio (600 s)	Rank Médio (3600 s)
H1	1,67	1,56
H2	2,56	2,89
H3	2,94	3,00
H4	4,06	4,17
H5	4,33	4,22
H6	5,44	5,17

Os ranks confirmam a superioridade de H1 sobre as demais, em especial sobre H4, que utiliza as mesmas estratégias, exceto pelo Path Relinking. Isso evidencia a importância dessa técnica na intensificação da busca, especialmente com mais tempo de execução. H2 e H3 mantêm

DOI: 10.59254/sbpo-2025-212146
https://proceedings.science/p/212146?lang=pt-br

desempenho intermediário, mas sem diferença estatisticamente relevante entre si. Estratégias que omitem o Path Relinking (H4, H5, H6) têm desempenho claramente inferior e mais instável. Estes resultados reforçam a necessidade de abordagens híbridas e configuráveis para lidar com a diversidade estrutural do TTP.

7. Conclusão

A resolução eficiente do Travelling Thief Problem (TTP) exige equilibrar exploração e intensificação em um espaço de busca complexo. Os experimentos confirmaram que o *Path Relinking* é decisivo para a qualidade das soluções, como apontado por (Marzo e Ribeiro, 2020). A heurística H1, que combina reatividade, *Path Relinking* e *Restart*, obteve os melhores resultados médios, superando significativamente H4 (sem *Path Relinking*). O aumento do tempo de execução (600s para 3600s) trouxe ganhos relevantes, especialmente em instâncias maiores.

A aleatoriedade na fase construtiva teve efeito limitado e não alterou o ranking global. O teste de Friedman confirmou diferenças significativas entre as abordagens, e o de Nemenyi apontou significância apenas entre H1 e H4, possivelmente pela alta variância das demais. Esses achados reforçam estudos prévios sobre a importância da intensificação via *Path Relinking* (Marzo e Ribeiro, 2020; Resende e Ribeiro, 2011; Maity e Das, 2020), mantidos mesmo com instâncias maiores e maior tempo-limite.

Conclui-se que a combinação de mecanismos exploratórios (reatividade), intensificadores (*Path Relinking*) e diversificadores (*Restart*) é fundamental para lidar robustamente com a complexidade combinatória do TTP.

8. Agradecimentos

Os autores agradecem à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), pelo apoio por meio do processo nº 2024/08848-3. O presente trabalho também foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. Esta obra foi desenvolvida de forma conjunta entre a Universidade Federal de São Paulo (UNIFESP) – Campus São José dos Campos e o Centro de Competência em Manufatura (CCM) do Instituto Tecnológico de Aeronáutica (ITA), instituições às quais os autores expressam sua gratidão pelo suporte acadêmico, científico e pela infraestrutura disponibilizada.

Referências

Bonyadi, M. R. e Michalewicz, Z. (2013). The travelling thief problem: The first step in the transition from theoretical problems to realistic problems. In *2013 IEEE Congress on Evolutionary Computation*, p. 1037–1044. IEEE.

Croes, G. A. (1958). A method for solving traveling-salesman problems. *Operations Research*, 6 (6):791–812.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(Jan):1–30.

El Yafrani, M. e Ahiod, B. (2018). Efficiently solving the traveling thief problem using hill climbing and simulated annealing. *Information Sciences*, 432:231–244.

Faulkner, H., Wang, H., e Michalewicz, Z. (2015). Approximate approaches to the ttp based on simple heuristics. In *Proceedings of the Genetic and Evolutionary Computation Conference*, p. 385–392.

Feo, T. A. e Resende, M. G. C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2):109–133.

DOI: 10.59254/sbpo-2025-212146
https://proceedings.science/p/212146?lang=pt-br

Glover, F. e Laguna, M. (1997). Tabu search and adaptive memory programming—advances, applications and challenges. In *Interfaces in Computer Science and Operations Research*, p. 1–75. Springer.

Maity, A. e Das, S. (2020). Efficient hybrid local search heuristics for solving the travelling thief problem. *Applied Soft Computing*, 93:106284. URL `https://www.sciencedirect.com/science/article/pii/S1568494620302246`.

Martello, S. e Toth, P. (1990). *Knapsack Problems: Algorithms and Computer Implementations*. Wiley.

Marzo, R. G. e Ribeiro, C. C. (2020). A grasp with path-relinking and restarts heuristic for the prize-collecting generalized minimum spanning tree problem. *International Transactions in Operational Research*, 27(3):1419–1446.

Mei, Y., Zhang, M., e Liang, Y. (2014). Heuristic approaches for the travelling thief problem. *IEEE Transactions on Evolutionary Computation*, 18(5):741–756.

Miller, C. E., Tucker, A. W., e Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM*, 7(4):326–329.

Namazi, M., Newton, M. A. H., Sanderson, C., e Sattar, A. (2023). Solving travelling thief problems using coordination based methods. *Journal of Heuristics*.

Polyakovskiy, S., Bonyadi, M. R., Michalewicz, Z., e Wagner, M. (2014). A comprehensive benchmark set and heuristics for the traveling thief problem. In *Proceedings of GECCO (Companion 2014)*, p. 453–460.

Prazeres, B. e Ribeiro, C. C. (2017). Adaptive parameter selection in grasp for combinatorial optimization. *Optimization Letters*, 11:1535–1550.

Resende, M. G. C. e Ribeiro, C. C. (2010). Grasp: Greedy randomized adaptive search procedures. In *Handbook of Metaheuristics*, p. 283–319. Springer.

Resende, M. G. C. e Ribeiro, C. C. (2011). Restart strategies for grasp with path-relinking heuristics. *Optimization Letters*, 5:467–478.

Rodriguez, J. e Smith, A. (2023). Sequence prediction for solving the travelling thief problem using machine learning. *Artificial Intelligence Review*, 56:1243–1261.

Wagner, M., Lindauer, M., Misir, M., Nallaperuma, S., e Hutter, F. (2018). A case study of algorithm selection for the traveling thief problem. *Journal of Heuristics*, 24:295–320. Dataset de desempenho: 21 algoritmos em 9.720 instâncias.

Wu, J., Wagner, M., Polyakovskiy, S., e Neumann, F. (2018). Exact approaches for the travelling thief problem. In *Simulated Evolution and Learning (SEAL)*.

DOI: 10.59254/sbpo-2025-212146
https://proceedings.science/p/212146?lang=pt-br