

Busca Local Eficiente para Problemas de Roteamento de Veículos com Demandas Incertas

Carlos Vinícius, Anand Subramanian, Marcelo de Freitas

Universidade Federal da Paraíba – Centro de Informática
Rua dos Escoteiros, s/n – Mangabeira, João Pessoa – PB, 58055-000
carlosneves@eng.ci.ufpb.br, anand@ci.ufpb.br,
marcelo.freitas@academico.ufpb.br

RESUMO

Este trabalho aborda a busca local eficiente para problemas de roteamento de veículos sob incerteza na demanda. Estruturas de dados novas foram propostas para calcular de forma eficiente a demanda total de uma rota no pior caso durante a avaliação de movimentos. Usando essas estruturas de dados, novos limites superiores assintóticos para a complexidade temporal de avaliação de um movimento são teoricamente derivados, melhorando os resultados existentes na literatura. Extensos experimentos computacionais demonstram que nosso método reduz significativamente o tempo computacional na prática em comparação com as abordagens atuais.

PALAVRAS CHAVE. Roteamento de veículos, incertezas, otimização robusta, busca local.

Tópicos: OC - Otimização Combinatória

ABSTRACT

This work addresses efficient local search for vehicle routing problems under demand uncertainty. We propose novel data structures for efficiently computing the worst-case total demand of a route during move evaluation. Using these data structures, we theoretically derive new asymptotic upper bounds for the time complexity of evaluating a move, improving upon existing results in the literature. Extensive computational experiments demonstrate that our method significantly reduces computational time in practice compared to current approaches.

KEYWORDS. Vehicle routing, uncertainties, robust optimization, local search.

Paper topics: CO - Combinatorial Optimization

1. Introdução

O Problema de Roteamento de Veículos (PRV) é um dos mais famosos problemas na área da Otimização Combinatória (OC). Nele, visa-se obter um conjunto de rotas de custo mínimo para servir um grupo de clientes — cada um dos quais possui uma certa demanda por um produto — empregando-se uma frota de veículos. Além disso, aplicações práticas normalmente incorrem restrições adicionais, como janelas de tempo de atendimento dos clientes, frotas limitadas, etc. Os métodos projetados para resolver problemas de OC como o PRV geralmente consistem em abordagens determinísticas, i.e., parte-se do pressuposto de que os parâmetros associados ao problema (e.g., demandas dos clientes, tempos de viagem) são certos, e conhecidos previamente. No entanto, na prática, é comum que esse não seja o caso. Se, por exemplo, as demandas dos clientes de uma rota forem 10% maiores do que o esperado, uma solução obtida por uma abordagem determinística pode acabar violando a restrição de capacidade do veículo, tornando-se inválida. Isso exigiria que a solução fosse “consertada”, ou que uma nova etapa de otimização fosse feita a partir do zero.

Uma abordagem comum na resolução de PRVs envolvendo parâmetros incertos consiste na Otimização Robusta (OR) Aharon Ben-Tal et al. [2009]. Nela, parâmetros incertos do problema são tratados como variáveis aleatórias pertencentes a um conjunto de incertezas bem-definido. Uma solução para o problema é considerada viável se e somente se for viável para todas as realizações possíveis de tais variáveis aleatórias. Por consequência, uma grande vantagem da OR é que não é necessário conhecer a natureza da distribuição dos parâmetros incertos associados ao problema. Além disso, o tomador de decisões possui liberdade para projetar os conjuntos de incertezas de acordo com suas necessidades: conjuntos de incertezas mais amplos resultam em soluções mais “conservadoras”, que embora custosas, são protegidas contra diversos cenários, enquanto conjuntos mais restritos resultam em soluções menos protegidas, mas de custo possivelmente menor.

Dentre as heurísticas estado-da-arte na resolução da versão determinística do PRV, pode-se citar, por exemplo, os algoritmos propostos por Subramanian et al. [2012]; Penna et al. [2019]; Máximo et al. [2022], que fazem uso da meta-heurística de Busca Local Iterada (*Iterated Local Search*, ILS) para resolver variantes do PRV com múltiplos atributos (e.g., múltiplos depósitos, frotas heterogêneas). Vidal et al. [2013], por sua vez, empregou um algoritmo genético na resolução de diversas variantes do PRV com janelas de tempo. Quanto a variantes do PRV envolvendo demandas incertas, pode-se destacar a abordagem proposta por Pessoa et al. [2021], que faz uso da meta-heurística ILS para resolver o CVRP com demandas incertas sob o paradigma da OR. De forma similar, Subramanyam et al. [2018] propõe uma abordagem baseada em Programação de Memória Adaptativa (*Adaptive Memory Programming*, AMP) para resolver uma variante geral do PRV sob demandas incertas, envolvendo vários atributos e conjuntos de incerteza distintos.

Embora consistam em meta-heurísticas diversas, no geral, todas as abordagens apresentadas possuem em comum o uso de buscas locais no decorrer do método. Uma busca local consiste em, dada uma solução corrente, examinar o custo de todas as soluções “vizinhas”, i.e., que diferem da solução atual em apenas um movimento (e.g., trocar a posição de dois clientes em uma rota, mover um cliente de uma posição na rota para outra), e determinar a melhor delas, e trocá-la pela solução atual. Embora isso seja comumente trivial em versões determinística do PRV, a avaliação de soluções vizinhas pode tornar-se custosa quando deseja-se empregar o paradigma de OR. Isso ocorre porque verificar se uma solução vizinha é adequada envolve determinar sua viabilidade. Caso seja feita de forma ingênua, o tempo computacional exigido por essa verificação — que depende do conjunto de incertezas empregado — pode ser proibitivo.

Portanto, o objetivo deste trabalho é propor estruturas de dados auxiliares avançadas para verificar de maneira eficiente a viabilidade de soluções vizinhas em problemas de roteamento com demandas incertas sob o paradigma da OR. A abordagem é versátil, podendo ser acoplada a qual-

quer meta-heurística que faça uso de etapas de busca local. O restante do trabalho é dividido como segue. Na Seção 2, a versão robusta do PRV com demandas incertas é caracterizada de forma mais precisa, bem como os conjuntos de incerteza abordados. A Seção 3, por sua vez, descreve as estruturas auxiliares utilizadas, e as provas de corretude e de complexidade assintótica associadas a elas. Por fim, as Seções 4 e 5 discorrem brevemente acerca dos resultados obtidos, e das considerações finais do trabalho, respectivamente.

2. Definição do problema e conjuntos de incerteza

Já que o método proposto neste trabalho visa endereçar especificamente incertezas da demanda, a versão do PRV utilizada será a versão canônica, denominada *Capacitated Vehicle Routing Problem* (CVRP), que envolve apenas restrições de capacidade. Note, no entanto, que os conceitos apresentados (bem como os métodos propostos nas seções seguintes), são gerais e podem ser aplicados a qualquer variante do PRV que envolva demandas incertas e restrições de capacidade. O CVRP é definido como segue.

Seja $G = (V, A)$ um grafo, e V e A conjuntos de vértices e arcos, respectivamente. O subconjunto $V' = \{1, \dots, n\}$ representa os clientes, enquanto o vértice 0 é o depósito, i.e., $V = V' \cup \{0\}$. Cada arco $(i, j) \in A, i \neq j$ está associado a um custo c_{ij} , e a matriz de custos satisfaz a desigualdade triangular. Cada cliente $i \in V'$ possui uma certa demanda q_i . Um conjunto de veículos homogêneos K está disponível no depósito, cada um com capacidade C . O objetivo é minimizar a soma dos custos de viagem considerando: (i) todo cliente deve ser visitado uma única vez; (ii) a capacidade do veículo não é excedida; e (iii) cada rota começa e termina no depósito. Para abordar o problema sob a ótica da OR, define-se, ainda, um conjunto de incertezas \mathcal{U} , e diz-se que um elemento de \mathcal{U} consiste em um vetor de demandas $\mathbf{q} = (q_1, \dots, q_n)$. Para que uma rota seja considerada robusta viável, ela deve, conforme definido acima, respeitar a restrição de capacidade do veículo associado para todo $\mathbf{q} \in \mathcal{U}$. Note que é necessário apenas encontrar o valor máximo que a soma das demandas em cada rota pode atingir. Note, ainda, que se \mathcal{U} for composto por exatamente um único elemento, tem-se a versão determinística do problema.

Além do conjunto determinístico $\mathcal{U}^0 = \{\mathbf{q}^0\}$, três outros conjuntos de incertezas bastante abordados na literatura foram abordados. Cada um desses conjuntos possui vantagens, fundamentadas em estatística, que os tornam adequados em diferentes situações práticas. Contudo, explorá-las não é o foco do trabalho. Portanto, ao leitor interessado em uma leitura aprofundada sobre os conjuntos de incertezas abordados, bem como as motivações associadas a cada um, recomenda-se que consulte Subramanyam et al. [2018]; Aharon Ben-Tal et al. [2009]. Os conjuntos utilizados foram $\mathcal{U}^{\text{knap}}$ (mochila), $\mathcal{U}^{\text{card}}$ (cardinalidade), e $\mathcal{U}^{\text{disc}}$ (discreto).

Dessa forma, sejam $\underline{\mathbf{q}}, \bar{\mathbf{q}}$ e $\hat{\mathbf{q}}$ vetores pertencentes a \mathbb{R}^n , $\mathbf{e} \in \mathbb{R}^n$ um vetor em que todas as componentes são iguais a 1, $\text{diag}(\cdot)$ uma função que transforma um vetor em uma matriz diagonal, e $\{B_1, \dots, B_L\}$ uma partição do conjunto V' . Os conjuntos de incerteza podem então ser definidos pelas seguintes equações.

$$\begin{aligned}
 \mathcal{U}^0 &= \{\mathbf{q}^0\} \\
 \mathcal{U}^{\text{card}} &= \{\mathbf{q} \in [\mathbf{q}^0, \mathbf{q}^0 + \hat{\mathbf{q}}] : \mathbf{q} = \mathbf{q}^0 + \text{diag}(\hat{\mathbf{q}})\xi, \mathbf{e}^T \xi \leq \Gamma, \xi \in [0, 1]^n\} \\
 \mathcal{U}^{\text{knap}} &= \left\{ \mathbf{q} \in [\mathbf{q}^0, \mathbf{q}^0 + \hat{\mathbf{q}}] : \sum_{i \in B_l} q_i \leq b_l, l \in \{1, \dots, L\} \right\} \\
 \mathcal{U}^{\text{disc}} &= \{\mathbf{q}^1, \mathbf{q}^2, \dots, \mathbf{q}^D\}
 \end{aligned}$$

Assim, seja S um conjunto de clientes em uma rota arbitrária. A demanda total da rota

associada a S no pior-caso pode ser calculada conforme as seguintes equações.

$$Q^0(S) = \sum_{i \in S} q_i^0 \quad (1)$$

$$Q^{\text{knap}}(S) = \sum_{i \in S} q_i^0 + \sum_{l=1}^L \min \left\{ b_l, \sum_{i \in B_l} \hat{q}_i \right\} \quad (2)$$

$$Q^{\text{card}}(S) = \sum_{i \in S} q_i^0 + \sum_{k=1}^{\min\{\Gamma, |S|\}} \hat{g}_k \quad (3)$$

$$Q^{\text{disc}}(S) = \max_{d=1, \dots, D} \left\{ \sum_{i \in S} q_i^d \right\} \quad (4)$$

Observe que, no caso específico do conjunto $\mathcal{U}^{\text{card}}$, o termo \hat{g}_k na equação (3) é o k -ésimo maior desvio de demanda dentre os clientes em S .

3. Cálculo eficiente da demanda no pior-caso

Para computar a demanda total no pior-caso durante a avaliação de um movimento de busca local, Subramanyam et al. [2018] faz uso de uma abordagem incremental. Mais precisamente, ele propõe um algoritmo que, dado um dos conjuntos de incerteza citados anteriormente e uma rota composta por um conjunto de clientes S , permite computar a demanda total após a adição (ou remoção) de um único cliente. Isso significa que o algoritmo precisa ser executado uma vez para cada cliente adicionado ou removido em um movimento. Em outras palavras, o tempo computacional gasto pelo método depende diretamente do número de clientes trocados entre rotas em um movimento de busca local. A Tabela 1 mostra as complexidades assintóticas do método para a avaliação de um movimento que envolve a adição de um conjunto de clientes S' a uma rota arbitrária associada a um conjunto de clientes S .

Tabela 1: Complexidade assintótica de avaliação de movimento ao adicionar (ou remover) um cliente no método de Subramanyam et al. [2018]

Conjunto	Complexidade
$\mathcal{U}^{\text{knap}}$	$\mathcal{O}(S')$
$\mathcal{U}^{\text{card}}$	$\mathcal{O}(S' \log S)$
$\mathcal{U}^{\text{disc}}$	$\mathcal{O}(S' D)$

O método proposto neste trabalho, em contrapartida, faz uso de estruturas auxiliares associadas a cada uma das subsequências de uma rota pertencente à solução corrente, de modo semelhante à abordagem originalmente proposta por Savelsbergh [1985] para checagem de viabilidade de rotas do PRV com janelas de tempo. Dessa forma, um movimento que leva a solução corrente a uma solução vizinha (cuja viabilidade deseja-se avaliar) pode ser visto como um rearranjo de tais subsequências. De forma mais precisa, define-se $\sigma = (\sigma_i, \dots, \sigma_j)$ e $\sigma' = (\sigma'_{i'}, \dots, \sigma'_{j'})$ como duas sequências de clientes consecutivos que fazem parte de rotas da solução corrente, e diz-se que a sequência $\sigma \oplus \sigma' = (\sigma_i, \dots, \sigma_j, \sigma'_{i'}, \dots, \sigma'_{j'})$ é obtida “concatenando-se” σ e σ' .

Dessa forma, se algumas informações específicas sobre σ e σ' forem armazenadas previamente, é possível obter a demanda total da sequência $\sigma \oplus \sigma'$ no pior-caso de maneira eficiente. Além disso, já que a maioria dos operadores de busca local na literatura consiste em rearranjar um

número constante de subsequências, a complexidade assintótica desse método depende apenas do tempo de uma única operação concatenação. Considere, então, que $\sigma = (i)$ é uma subsequência composta por exatamente um cliente. Define-se as seguintes estruturas de dados auxiliares para σ .

$$Q_l^{\text{knap}}(\sigma) = \begin{cases} \hat{q}_i & \text{se } i \in B_l \\ 0 & \text{caso contrário} \end{cases} \quad l = 1, \dots, L \quad (5)$$

$$Q_d^{\text{disc}}(\sigma) = q_i^d \quad d = 1, \dots, D \quad (6)$$

$$Q_0^{\text{card}}(\sigma) = q_i \quad (7)$$

$$Q_\gamma^{\text{card}}(\sigma) = q_i + \hat{q}_i \quad \gamma = 1, \dots, \Gamma \quad (8)$$

Em casos nos quais uma subsequência pode ser descrita como uma concatenação de outras duas, i.e., a subsequência é $\sigma \oplus \sigma'$.

$$Q_l^{\text{knap}}(\sigma \oplus \sigma') = Q_l^{\text{knap}}(\sigma) + Q_l^{\text{knap}}(\sigma'), \quad l = 1, \dots, L \quad (9)$$

$$Q_d^{\text{disc}}(\sigma \oplus \sigma') = Q_d^{\text{disc}}(\sigma) + Q_d^{\text{disc}}(\sigma'), \quad d = 1, \dots, D \quad (10)$$

$$Q_\gamma^{\text{card}}(\sigma \oplus \sigma') = \max_{\gamma'=0, \dots, \gamma} \{Q_{\gamma'}^{\text{card}}(\sigma) + Q_{\gamma-\gamma'}^{\text{card}}(\sigma')\} \quad \gamma = 0, \dots, \Gamma \quad (11)$$

A partir disso, o seguinte resultado mostra como obter a demanda total da sequência $\sigma \oplus \sigma'$ no pior-caso.

Proposição 1. *Sejam σ e σ' duas subsequências, a demanda total no pior-caso da subsequência $\sigma \oplus \sigma'$ para cada conjunto de incertezas pode ser calculada como segue:*

$$Q^0(\sigma \oplus \sigma') = Q^0(\sigma) + Q^0(\sigma') \quad (12)$$

$$Q^{\text{knap}}(\sigma \oplus \sigma') = Q^0(\sigma \oplus \sigma') + \sum_{l=1}^L \min \{b_l, Q_l^{\text{knap}}(\sigma \oplus \sigma')\} \quad (13)$$

$$Q^{\text{card}}(\sigma \oplus \sigma') = Q_\Gamma^{\text{card}}(\sigma \oplus \sigma') \quad (14)$$

$$Q^{\text{disc}}(\sigma \oplus \sigma') = \max_{d=1, \dots, D} \{Q_d^{\text{disc}}(\sigma \oplus \sigma')\} \quad (15)$$

Demonstração. Primeiro, note que o caso particular em que a subsequência consiste em um único cliente é trivialmente correto. Para mostrar que as equações (12)–(15) estão correto para os casos restantes, considera-se cada conjunto individualmente. Primeiro, supõe-se que a demanda total no pior-caso é conhecida para duas subsequências arbitrárias σ e σ' . Em seguida, mostra-se que quando $S = \sigma \cup \sigma'$, o lado direito das equações (1)–(4) é equivalente ao do seu respectivo par nas equações (12)–(15). A corretude então segue através de indução no operador de concatenação \oplus .

((1) \Leftrightarrow (12)) Suponha que $Q^0(\sigma) = \sum_{i \in \sigma} q_i^0$ e $Q^0(\sigma') = \sum_{j \in \sigma'} q_j^0$. Então,

$$Q^0(\sigma \oplus \sigma') = Q^0(\sigma) + Q^0(\sigma') = \sum_{i \in \sigma} q_i^0 + \sum_{j \in \sigma'} q_j^0.$$

((2) \Leftrightarrow (13)) Suponha que $Q_l^{\text{knap}}(\sigma) = \sum_{i \in \sigma} \hat{q}_i$, e $Q_l^{\text{knap}}(\sigma') = \sum_{j \in \sigma'} \hat{q}_j$. Então,

$$\begin{aligned}
 Q^{\text{knap}}(\sigma \oplus \sigma') &= Q^0(\sigma \oplus \sigma') + \sum_{l=1}^L \min \left\{ b_l, Q_l^{\text{knap}}(\sigma \oplus \sigma') \right\} \\
 &= \sum_{i \in \sigma} q_i^0 + \sum_{j \in \sigma'} q_j^0 + \sum_{l=1}^L \min \left\{ b_l, \sum_{i \in \sigma} \hat{q}_i + \sum_{j \in \sigma'} \hat{q}_j \right\}.
 \end{aligned}$$

((4) \Leftrightarrow (15)) Suponha que $Q_d^{\text{disc}} = \sum_{i \in \sigma} q_i^d$, e $Q_d^{\text{disc}}(\sigma') = \sum_{j \in \sigma'} q_j^d$. Então,

$$\begin{aligned}
 Q^{\text{disc}}(\sigma \oplus \sigma') &= \max_{d=1, \dots, D} \left\{ Q_d^{\text{disc}}(\sigma \oplus \sigma') \right\} \\
 &= \max_{d=1, \dots, D} \left\{ \sum_{i \in \sigma} q_i^d + \sum_{j \in \sigma'} q_j^d \right\}.
 \end{aligned}$$

((3) \Leftrightarrow (14)) Suponha que $Q_\gamma^{\text{card}}(\sigma) = \sum_{i \in \sigma} q_i^0 + \sum_{i=1}^{\min\{|\sigma|, \gamma\}} \hat{g}_i^\sigma$, e $Q_\gamma^{\text{card}}(\sigma') = \sum_{j \in \sigma'} q_j^0 + \sum_{j=1}^{\min\{|\sigma'|, \gamma\}} \hat{g}_j^{\sigma'}$ para todo $\gamma = 0, \dots, \Gamma$. Então,

$$\begin{aligned}
 Q_\gamma^{\text{card}}(\sigma \oplus \sigma') &= \max_{\gamma'=0, \dots, \gamma} \left\{ Q_{\gamma'}^{\text{card}}(\sigma) + Q_{\gamma-\gamma'}^{\text{card}}(\sigma') \right\} \\
 &= \max_{\gamma'=0, \dots, \gamma} \left\{ \sum_{i \in \sigma} q_i^0 + \sum_{i=1}^{\min\{|\sigma|, \gamma'\}} \hat{g}_i^\sigma + \sum_{j \in \sigma'} q_j^0 + \sum_{j=1}^{\min\{|\sigma'|, \gamma-\gamma'\}} \hat{g}_j^{\sigma'} \right\},
 \end{aligned}$$

e portanto,

$$\begin{aligned}
 Q^{\text{card}}(\sigma \oplus \sigma') &= Q_\Gamma^{\text{card}}(\sigma \oplus \sigma') \\
 &= \max_{\gamma'=0, \dots, \Gamma} \left\{ \sum_{i \in \sigma} q_i^0 + \sum_{i=1}^{\min\{|\sigma|, \gamma'\}} \hat{g}_i^\sigma + \sum_{j \in \sigma'} q_j^0 + \sum_{j=1}^{\min\{|\sigma'|, \Gamma-\gamma'\}} \hat{g}_j^{\sigma'} \right\}.
 \end{aligned}$$

É fácil verificar que maximizar a última expressão equivale a escolher os $\min\{|\sigma \oplus \sigma'|, \Gamma\}$ maiores elementos do conjunto de desvios $\hat{g}_1^\sigma, \dots, \hat{g}_{|\sigma|}^\sigma, \hat{g}_1^{\sigma'}, \dots, \hat{g}_{|\sigma'|}^{\sigma'}$ e adicioná-los a $Q^0(\sigma \oplus \sigma')$, assim concluindo a prova. \square

A Tabela 2 é um resultado direto da Proposição 1, e ilustra a complexidade assintótica de uma operação de concatenação para cada um dos conjuntos de incertezas abordados, através das estruturas propostas acima. Como argumentado anteriormente, se o número de operações de concatenação for constante, a complexidade assintótica da avaliação inteira é a mesma que a de uma única concatenação. No geral, isso torna as estruturas acima mais adequadas, embora exijam muito mais memória, no geral, que as estruturas propostas por Subramanyam et al. [2018].

4. Resultados

Os experimentos computacionais realizados consistem em uma comparação entre as estruturas propostas e, o método de Subramanyam et al. [2018]. Para isso, um algoritmo de busca local simples foi implementado. No algoritmo, uma única estrutura de vizinhança é explorada de maneira exaustiva, i.e., todos os vizinhos da solução corrente tiverem seu custo e viabilidade avaliados, e o vizinho de menor custo (ou o de menor violação de capacidade, no caso de soluções

Tabela 2: Complexidade assintótica da avaliação dos movimentos

Conjunto	Complexidade
$\mathcal{U}^{\text{knap}}$	$\mathcal{O}(L)$
$\mathcal{U}^{\text{card}}$	$\mathcal{O}(\Gamma)$
$\mathcal{U}^{\text{disc}}$	$\mathcal{O}(D)$

inviáveis) é escolhido. Uma vez que o melhor vizinho é determinado, ele passa a ser a solução corrente, e o procedimento se repete até que melhoras não sejam mais possíveis. As soluções iniciais usadas nos testes foram obtidas de maneira gulosa e aleatória, de forma similar à proposta em [Subramanian et al., 2013]. Três estruturas de vizinhança clássicas na literatura de PRVs foram testadas de forma separada, e para cada uma delas, 500 soluções iniciais foram construídas. As estruturas de vizinhança utilizadas são descritas a seguir.

- SHIFT(1,0) – Mover um cliente de uma rota para outra.
- SWAP(1,1) – Trocar um cliente de uma rota com um cliente de outra rota.
- 2-OPT* – Escolher duas rotas e remover um arco de cada, e reconstruir a solução das duas maneiras possíveis.

Os métodos foram implementados na linguagem C++ (g++, 9.4.0) e executados em um processador Intel® Core™ i7-3770 com 3,40 GHz e 16 GB de memória RAM. O sistema operacional utilizado foi o Ubuntu 20.04. O conjunto clássico de instâncias A, B, E, F, M, P para o CVRP foi utilizado nos testes. Os conjuntos de incerteza foram gerados conforme descrito por Subramanyam et al. [2018]. Diferentes parâmetros foram testados para gerar os conjuntos de incertezas $\mathcal{U}^{\text{card}}$ e $\mathcal{U}^{\text{disc}}$. Mais precisamente, o conjunto $\mathcal{U}^{\text{card}}$ foi testado com os valores $\Gamma = 0,05n$ (caso 1), $\Gamma = 0,1n$ (caso 2), e $\Gamma = 0,2n$ (caso 3). O conjunto $\mathcal{U}^{\text{disc}}$, por sua vez, foi testado com $D = 0,25n$ (caso 1), $D = 0,5n$ (caso 2), e $D = 0,75n$ (caso 3).

As Figuras 1–3 ilustram o tempo médio de uma única busca local (em milissegundos) para cada estrutura de vizinhança, e em cada conjunto de incertezas. As instâncias foram ordenadas em ordem crescente, em função do número de clientes. Assim, cada figura apresenta um gráfico que contém o tempo médio de busca local em função da instância, tanto para a abordagem incremental de Subramanyam et al. [2018] (INCREMENTAL), quanto a abordagem baseada em subsequências proposta neste trabalho (SUBSEQ). Analisando-se as figuras, é possível perceber que, na maioria dos casos, a abordagem baseada em subsequências apresenta tempos de execução inferiores, sobretudo em instâncias com um número de clientes maior. Isso é facilmente observado, por exemplo, em todas as vizinhanças para o conjunto $\mathcal{U}^{\text{knap}}$, onde a abordagem baseada em subsequências domina a abordagem incremental.

Por outro lado, uma análise mais aprofundada — focando-se em cada conjunto de incertezas individualmente — revela que, no conjunto $\mathcal{U}^{\text{card}}$, o desempenho da abordagem incremental nas vizinhanças 2OPT* e SWAP(1,1) torna-se superior a medida que o número de clientes n aumenta. Note que, especificamente nos casos em que Γ é menor, o desempenho da abordagem baseada em subsequências torna-se superior. Contudo, abordagem baseada em subsequências é mais eficiente para ambas as parametrizações de Γ no caso específico da vizinhança SHIFT(1,0). Para o conjunto $\mathcal{U}^{\text{disc}}$, no caso em que $D = 0,25n$, o desempenho das duas abordagens é similar. Note, no entanto, que no caso em que $D = 0,5n$, a diferença entre as duas abordagens torna-se cada vez mais acen-

tuada a medida que n cresce, sendo a abordagem de subsequências substancialmente superior na vizinhança 2OPT*, embora seja ligeiramente pior para as vizinhanças SHIFT(1,0) e SWAP(1,1).

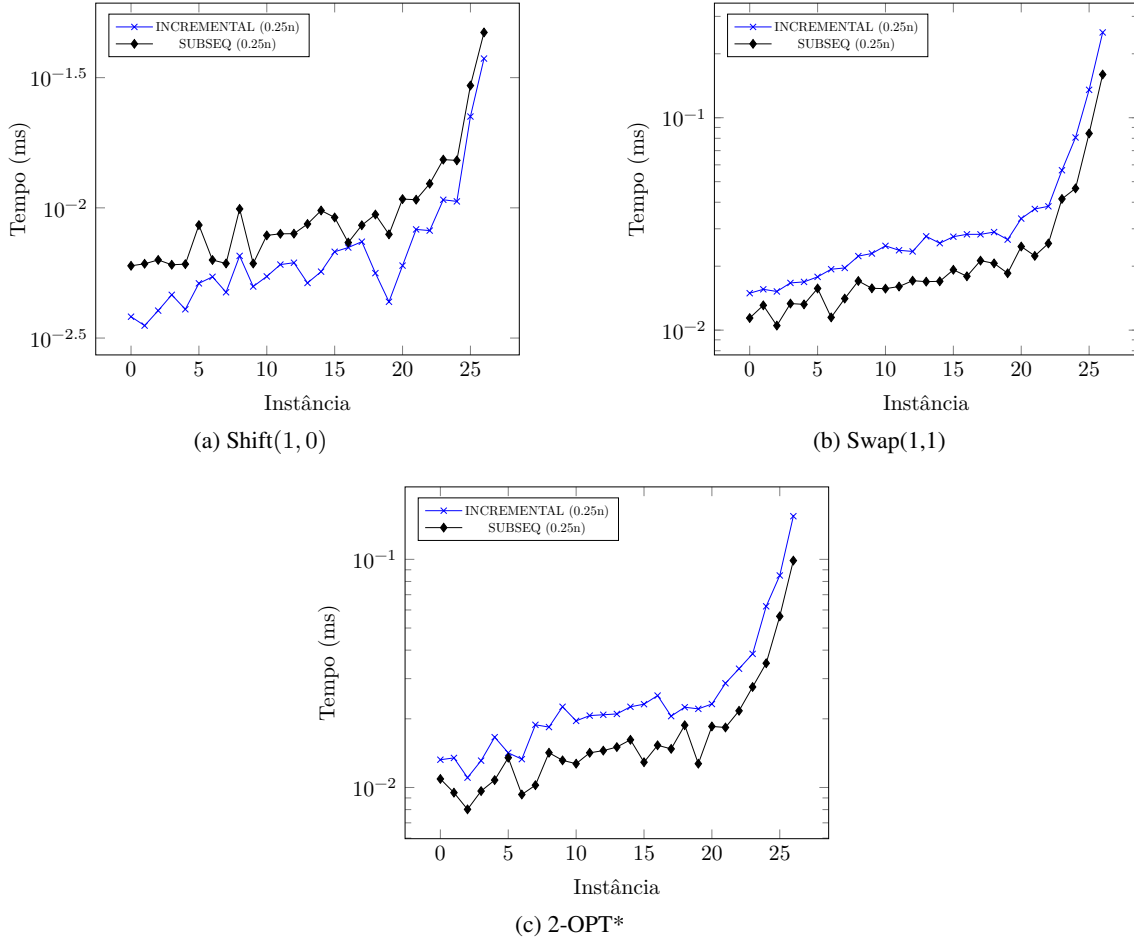


Figura 1: Tempos de execução para o conjunto $\mathcal{U}^{\text{knap}}$

5. Considerações finais

Neste trabalho, foram propostas estruturas de aceleração que permitem realizar buscas locais em PRVs com incerteza de maneira mais eficiente. Além disso, os limites assintóticos proporcionados por tais estruturas são, no geral, melhores que os do método estado-da-arte da literatura para todos os conjuntos de incerteza considerados. Esse resultado teórico também foi complementado por uma análise experimental em instâncias clássicas. Portanto, é possível afirmar que o método proposto é promissor.

Dentre as sugestões para trabalhos futuros, propõe-se a adaptação dos métodos a novos conjuntos de incerteza (e.g., conjuntos cuja definição depende da solução corrente). Propõe-se, ainda, acoplar o método a meta-heurísticas estado-da-arte para PRVs com múltiplos atributos, para verificar se novos limitantes superiores podem ser obtidos para instâncias em aberto. Por fim, convém investigar se os métodos utilizados podem ser adaptados (ou servir como base) para lidar com incertezas em outros parâmetros, como tempos de viagem e tempos de serviço.

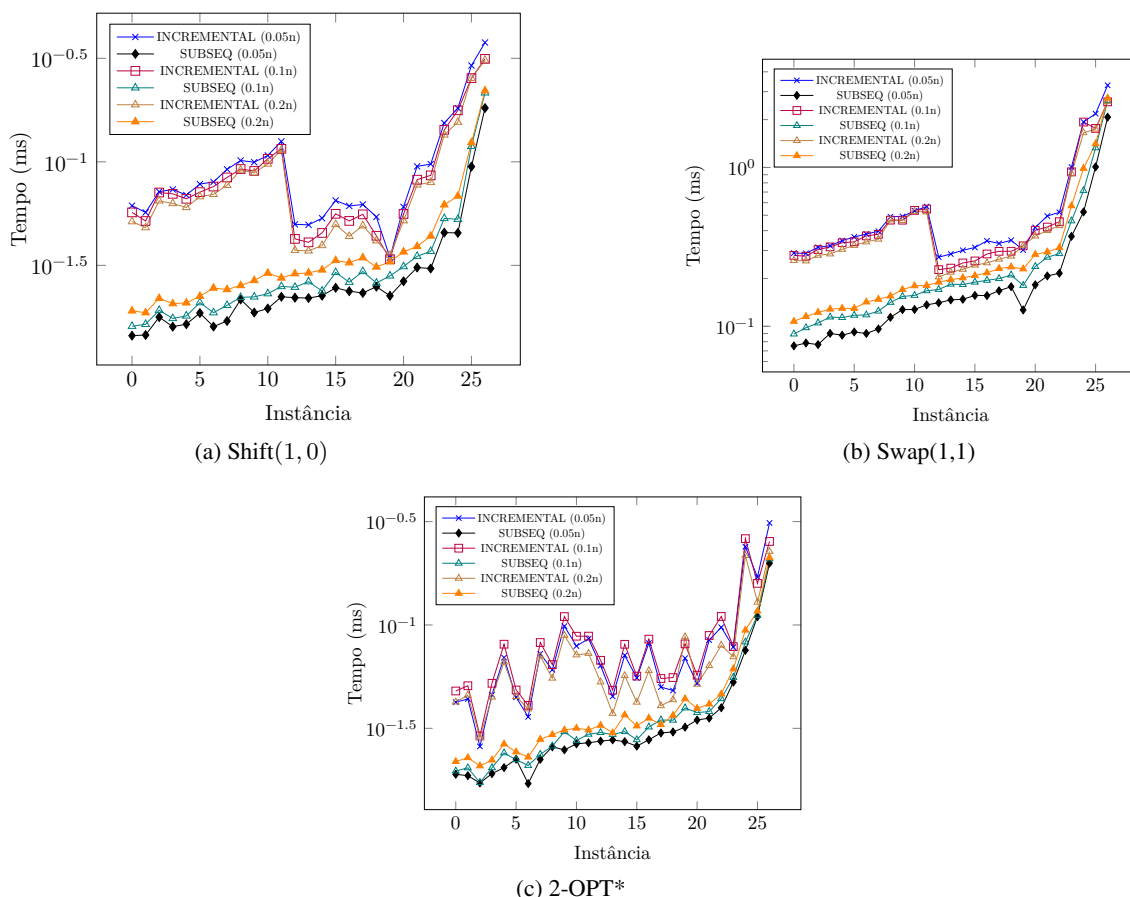


Figura 2: Tempos de execução para o conjunto \mathcal{U}^{card}

Referências

- Aharon Ben-Tal, Laurent El Ghaoui, e Arkadi Nemirovski (2009). Robust Optimization. Princeton series in applied mathematics. ISBN 978-0-691-14368-2. URL <https://press.princeton.edu/books/hardcover/9780691143682/robust-optimization>.
- Máximo, V. R., Cordeau, J.-F., e Nascimento, M. C. (2022). An adaptive iterated local search heuristic for the Heterogeneous Fleet Vehicle Routing Problem. Computers & Operations Research, 148:105954. ISSN 03050548. URL <https://linkinghub.elsevier.com/retrieve/pii/S0305054822002052>.
- Penna, P. H. V., Subramanian, A., Ochi, L. S., Vidal, T., e Prins, C. (2019). A hybrid heuristic for a broad class of vehicle routing problems with heterogeneous fleet. Annals of Operations Research, 273(1):5–74. ISSN 1572-9338. URL <https://doi.org/10.1007/s10479-017-2642-9>.
- Pessoa, A. A., Poss, M., Sadykov, R., e Vanderbeck, F. (2021). Branch-Cut-and-Price for the Robust Capacitated Vehicle Routing Problem with Knapsack Uncertainty. Operations Research, 69(3):739–754. ISSN 0030-364X, 1526-5463. URL <http://pubsonline.informs.org/doi/10.1287/opre.2020.2035>.
- Savelsbergh, M. W. P. (1985). Local search in routing problems with time windows. Annals

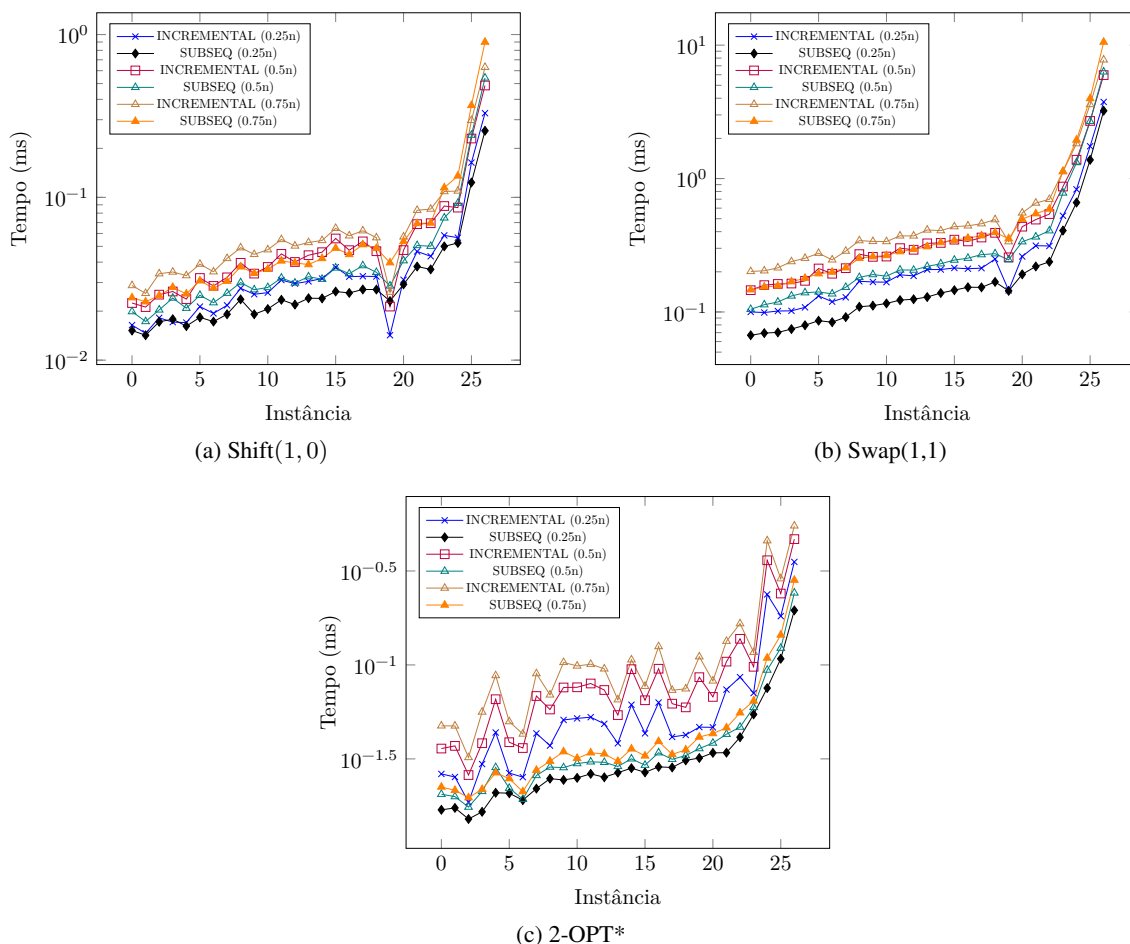


Figura 3: Tempos de execução para o conjunto $\mathcal{U}^{\text{disc}}$

of Operations Research, 4(1):285–305. ISSN 0254-5330, 1572-9338. URL <http://link.springer.com/10.1007/BF02022044>.

Subramanian, A., Penna, P. H. V., Uchoa, E., e Ochi, L. S. (2012). A hybrid algorithm for the Heterogeneous Fleet Vehicle Routing Problem. *European Journal of Operational Research*, 221(2): 285–295. ISSN 03772217. URL <https://linkinghub.elsevier.com/retrieve/pii/S0377221712002093>.

Subramanian, A., Uchoa, E., e Ochi, L. S. (2013). A hybrid algorithm for a class of vehicle routing problems. *Computers & Operations Research*, 40(10):2519–2531. ISSN 03050548. URL <https://linkinghub.elsevier.com/retrieve/pii/S030505481300021X>.

Subramanyam, A., Repoussis, P. P., e Gounaris, C. E. (2018). Robust optimization of a broad class of heterogeneous vehicle routing problems under demand uncertainty. URL <http://arxiv.org/abs/1810.04348>. arXiv:1810.04348 [cs, math].

Vidal, T., Crainic, T. G., Gendreau, M., e Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows.

Computers & Operations Research, 40(1):475–489. ISSN 0305-0548. URL <https://www.sciencedirect.com/science/article/pii/S0305054812001645>.