



Universidade Federal da Paraíba
Centro de Informática
Graduação em Ciência da Computação

Uma heurística eficiente pra o problema da mínima latência com janelas de tempo

Iderval da Costa e Silva Neto

João Pessoa - PB
2024

Iderval da Costa e Silva Neto

Uma heurística eficiente pra o problema da mínima latência com janelas de tempo

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal da Paraíba (UFPB), como requisito para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Teobaldo Leite Bulhões Júnior

Catálogo na publicação
Seção de Catalogação e Classificação

S586h Silva Neto, Iderval da Costa e.
Uma heurística eficiente pra o problema da mínima
latência com janelas de tempo / Iderval da Costa e
Silva Neto. - João Pessoa, 2024.
35 f. : il.

Orientação: Teobaldo Leite Bulhões Júnior.
TCC (Graduação) - UFPB/CI.

1. Roteamento. 2. Problema da mínima latência com
janelas de tempo. 3. Iterated Local Search. 4.
Algoritmo. I. Bulhões Júnior, Teobaldo Leite. II.
Título.

UFPB/CI

CDU 004.421

Iderval da Costa e Silva Neto

Uma heurística eficiente pra o problema da mínima latência com janelas de tempo

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal da Paraíba (UFPB), como requisito para obtenção do grau de Bacharel em Ciência da Computação.

Trabalho aprovado. João Pessoa - PB, 10 de Maio de 2024:

Prof. Teobaldo Leite Bulhões Júnior
Orientador

Eduardo Vieira Queiroga
Examinador

Prof. Anand Subramanian
Examinador

João Pessoa - PB
2024

Agradecimentos

Gostaria de agradecer a Deus, fonte de força e inspiração dado os obstáculos encontrados ao decorrer do curso.

À minha família, em especial a meus pais Alípia e Idinaldo, e a minha irmã Maiara, por não medirem esforços no apoio e encorajamento providos ao longo do curso e da vida.

À Larissa, que entrou na minha vida ao final do curso, mas que se tornou uma excelente companheira e motivadora essencial, dando um gigante apoio em momentos difíceis.

Ao Prof. Teobaldo Bulhões por sua orientação nesse trabalho e contribuição para meu desenvolvimento durante todos os anos em que fui membro do LOG.

Ao Prof. Bruno Bruck pelas oportunidades de aprendizado e orientação oferecidas enquanto membro do LOG.

Ao Prof. Anand Subramanian por aceitar compor a banca desse trabalho, e também por toda a paciência estendida a mim, assim como toda a motivação e ensinamentos de vida.

Ao Dr. Eduardo Queiroga, por aceitar compor essa banca e pelas sugestões e contribuições para a melhora desse trabalho.

A Carlos e Marcelo por oferecerem uma contribuição essencial para a concretização desse trabalho.

Por fim, aos demais membros do LOG, pelo companheirismo, apoio e amizade oferecidas, além dos momentos de descontração e das conversas trocadas sobre os tópicos mais aleatórios.

Resumo

Este estudo apresenta um método heurístico para lidar com o Problema da Mínima Latência com Janelas de Tempo. A abordagem proposta combina técnicas derivadas de *iterated local search* e *random variable neighbourhood descent*. Uma contribuição inovadora deste estudo é a proposta de uma avaliação eficiente em $O(1)$ amortizada para a função objetivo durante a busca local. Os experimentos foram conduzidos em 178 instâncias de uma variedade de conjuntos, obtendo uma solução melhor que a literatura em 91 casos.

Palavras-chave: Roteamento. Problema da Mínima Latência com Janelas de Tempo. *Iterated Local Search*.

Abstract

This study presents a heuristic method for dealing with the Minimum Latency Problem with Time Windows. The proposed approach combines techniques derived from iterated local search and random variable neighborhood descent. An innovative contribution of this study is the proposal of an efficient $O(1)$ amortized evaluation for the objective function during local search. Experiments were conducted on 178 instances from a variety of sets, obtaining a better solution than the literature in 91 cases.

Keywords: Routing. Minimum Latency Problem with Time Windows. Iterated Local Search.

Lista de tabelas

Tabela 1 – Resultados para MLPTW em instâncias propostas por Dumas et al. (1995) e Silva e Urrutia (2010)	26
Tabela 2 – Resultados para MLPTW em instâncias propostas por Gendreau et al. (1998) e Ohlmann e Thomas (2007)	28

Lista de ilustrações

Figura 1 – Exemplo de uma solução para o MLPTW	11
Figura 2 – Exemplos das estruturas de vizinhança utilizadas.	20
Figura 3 – Quebra de uma solução em 3 subsequências e reconstrução durante o movimento $\mathcal{Q}$ -opt.	23
Figura 4 – Exemplo da perturbação com Double-Bridge.	24

Sumário

1	INTRODUÇÃO	10
1.1	Definição do tema	10
1.2	Justificativa	12
1.3	Objetivos	12
1.3.1	Objetivo Geral	12
1.3.2	Objetivos específicos	13
1.4	Estrutura do trabalho	13
2	TRABALHOS RELACIONADOS	14
3	METODOLOGIA	17
3.1	Algoritmo construtivo	18
3.2	Busca local	18
3.2.1	Estruturas de vizinhança	19
3.2.2	Avaliação eficiente	20
3.3	Mecanismo de perturbação	23
4	EXPERIMENTOS COMPUTACIONAIS	25
4.1	Instâncias	25
4.2	Resultados	25
5	CONSIDERAÇÕES FINAIS	31
	REFERÊNCIAS	32

1 Introdução

1.1 Definição do tema

A gestão dos tempos de espera e de entrega desempenha um papel de extrema relevância na logística, especialmente quando se busca minimizar as distâncias percorridas em rotas. Nesse contexto, são desenvolvidos métodos para o roteamento de veículos com clientes geograficamente dispersos. Dentre os mais conhecidos na literatura pode ser citado o Problema do Caixeiro Viajante (TSP, do inglês, *Traveling Salesman Problem*). Devido a existência de uma grande variedade de cenários, inúmeras variantes desse problema existem e são discutidas em Gutin e Punnen (2006), tais como o Problema do Caixeiro Viajante com Coleta e Entrega (PDTSP, do inglês, *Pickup and Delivery Traveling Salesman Problem*) e o Problema do Caixeiro Viajante com Janelas de Tempo (TSPTW, do inglês, *Traveling Salesman Problem with Time Windows*).

Dentre as variantes do TSP pode ser citado o Problema da Mínima Latência (MLP, do inglês *Minimum Latency Problem*). Essa variante é conhecida na literatura por diversos nomes: *Traveling Repairman Problem* Tsitsiklis (1992), *Delivery Man Problem* Fischetti, Laporte e Martello (1993), *Cumulative Traveling Salesman Problem* Bianco, Mingozzi e Ricciardelli (1993), e *School Bus Driver Problem* Chaudhuri et al. (2003). Nesse trabalho, nos referiremos a todos os problemas equivalentes como MLP.

Algoritmos voltados a resolver problemas de otimização combinatória podem ser categorizados como exatos ou heurísticos. Algoritmos exatos obtêm soluções provadamente ótimas por uma busca exaustiva. Dada a inúmera quantidade de permutações que o conjunto de soluções de um problema de otimização pode ter, tais abordagens podem exigir tempo exponencial, o que faz com que o algoritmos heurísticos sejam favorecidos. Tais algoritmos buscam soluções de alta qualidade — mesmo que não sejam comprovadamente ótimas — em um tempo razoável. Alguns desses algoritmos heurísticos podem ser chamados de “meta-heurísticas”, sendo gerais o suficiente para abordar toda uma classe de problemas.

Este trabalho busca, através de métodos heurísticos, resolver um MLP com restrições de janela de tempo (MLPTW, do inglês *Minimum Latency Problem with Time Windows*). Por conter o MLP como um caso especial (i.e., quando todas as janelas de tempo são abertas), e sabendo-se que o MLP é NP-Difícil Tsitsiklis (1992), o MLPTW também é, por consequência, NP-Difícil. Portanto, não existem algoritmos que resolvam o MLPTW de forma ótima em tempo polinomial. O problema pode ser definido como segue.

Seja $G = (N \cup \{0\}, A)$ um grafo completo, orientado e assimétrico, em que $N = \{1, \dots, n\}$ é um conjunto de nós de cliente e 0 é o depósito. Um tempo de viagem c_{ij} está associado a cada arco $(i, j) \in A$. Cada nó i de G também está associado a um intervalo

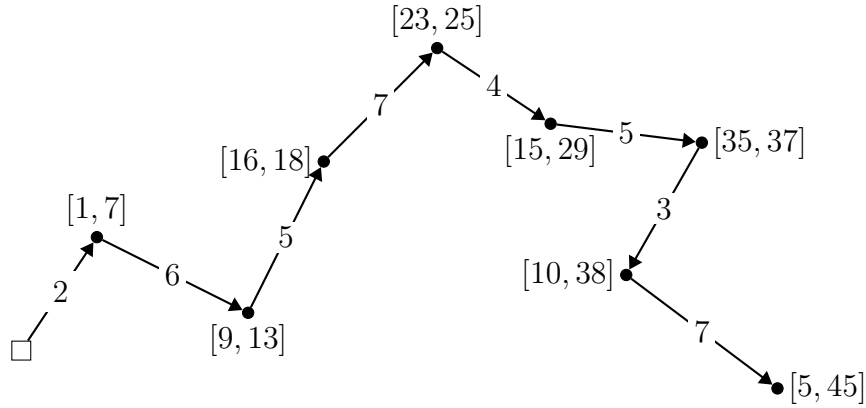


Figura 1 – Exemplo de uma solução para o MLPTW

$[e_i, l_i]$, que descreve o horário no qual o início do serviço pode ocorrer no nó i . Também define-se a duração do caminho até o nó i como a diferença entre o início do serviço no nó i e o início do serviço no depósito. O MLPTW consiste em determinar um caminho hamiltoniano — isso é, um caminho em que todo nó cliente é visitado exatamente uma vez — em G , começando no nó de depósito 0, que respeite as restrições de janela de tempo, e que minimize a duração de tempo de início de serviço de todos os clientes. Mais precisamente, sejam $t_0, t_1, \dots, t_n$ os tempos de início de serviço nos vértices $0, 1, \dots, n$, o objetivo é minimizar

$$\sum_{i=1}^n (t_i - t_0).$$

Vale ressaltar que é possível que o tempo de início dos serviços (que ocorre no depósito) t_0 possa ser livre e assumir qualquer valor entre e_0 e l_0 . Isso é considerado bom, pois pode evitar possíveis tempos de espera nos diversos clientes. Neste caso, diz-se que a instância admite saída tardia. É possível, no entanto, restringir t_0 a e_0 (como feito em alguns trabalhos) fazendo-se $l_0 = e_0$.

A versão do problema que permite uma saída tardia do depósito é claramente uma relaxação da versão fixa. Para compreender o impacto causado por essa relaxação, considere a solução para uma instância de 9 vértices, ilustrada na Figura 1. Se o veículo sair do depósito no tempo 0 e iniciar os serviços o mais cedo possível em cada vértice, os tempos de chegada serão

$$t = (0, 2, 9, 16, 23, 27, 35, 38, 45),$$

e o custo acumulado total será 195. Se, no entanto, o veículo esperar 5 unidades de tempo para sair do depósito, os tempos de início de serviço serão

$$t = (5, 7, 13, 18, 25, 29, 35, 38, 45),$$

e conseqüentemente, o custo acumulado será 170.

1.2 Justificativa

Preocupações ambientais e os desafios relacionados ao planejamento urbano vem crescendo nos últimos anos. Dado este cenário, a otimização de problemas logísticos existe como uma ferramenta para enfrentar a complexidade desses problemas. Ao utilizar princípios de otimização combinatória, soluções dinâmicas e adaptáveis podem ser oferecidas, tendo um papel importante na popularização de práticas urbanas mais sustentáveis.

O MLP representa uma extensão do TSP com uma perspectiva centrada nas necessidades do cliente. Enquanto o TSP visa minimizar o tempo total de viagem, o MLP tem como objetivo reduzir o tempo médio de espera de cada cliente. Como resultado, o MLP encontra aplicações variadas em setores como logística, transporte e serviços, destacando-se pela sua abordagem mais orientada ao cliente.

As aplicações significativas do Problema do MLP abrangem uma variedade de áreas, como serviços de entrega em domicílio, logística para ajuda humanitária e a minimização do consumo de energia. Esses setores se beneficiam da capacidade do MLP de reduzir o tempo médio de espera do cliente e otimizar a eficiência dos percursos, contribuindo para uma prestação de serviço mais ágil, eficaz e sustentável (HEILPORN; CORDEAU; LAPORTE, 2010).

Por outro lado, devido às suas restrições de janelas de tempo, o MLPTW demonstra grande utilidade em diversas aplicações práticas. Por exemplo, no roteamento de ônibus escolares, onde é essencial seguir horários específicos de partida e chegada à escola. Além disso, o transporte de produtos perecíveis se beneficia do MLPTW, pois esses itens não podem permanecer por períodos prolongados em trânsito, garantindo sua qualidade e segurança (HEILPORN; CORDEAU; LAPORTE, 2010).

Dada a ampla gama de aplicações, abordamos e propomos nesse trabalho uma solução para o MLPTW. Enquanto o MLP recebe considerável atenção na literatura, o MLPTW carece desse mesmo nível de foco e estudo. Essa lacuna na literatura é um dos motivos para que direcione-se esforços para resolver o problema e contribuir para o avanço do conhecimento nesse campo específico.

1.3 Objetivos

1.3.1 Objetivo Geral

Apresentar um algoritmo heurístico eficiente para a resolução do Problema da Mínima Latência com Janelas de Tempo.

1.3.2 Objetivos específicos

- Revisar as abordagens da literatura para esse problema.
- Desenvolver uma abordagem heurística para a resolução desse problema.
- Testar o algoritmo desenvolvido em instâncias da literatura.
- Comparar os resultados do algoritmo proposto com o melhor método encontrado na literatura.

1.4 Estrutura do trabalho

Este trabalho é organizado da seguinte maneira:

- O Capítulo 1 consiste na introdução do trabalho e na definição do tema.
- O Capítulo 2 apresenta uma revisão de literatura e trabalhos relacionados.
- O Capítulo 3 descreve a abordagem meta-heurística proposta.
- O Capítulo 4 apresenta os experimentos computacionais.
- Por fim, o Capítulo 5 traz as considerações finais.

2 Trabalhos relacionados

Embora poucos trabalhos tenham sido publicados sobre o tema específico do MLPTW, várias abordagens, exatas e heurísticas, foram propostas na literatura para o MLP e suas variantes. Esta seção visa apresentar tais trabalhos, bem como alguns dos trabalhos que abordam o MLPTW.

O MLP, foi formalmente proposto por Lucena (1990) com o nome de *Time-dependent Traveling Salesman Problem* (TDTSP). O trabalho apresenta uma abordagem baseada em programação não linear e relaxação lagrangeana para derivar limitantes inferiores para o problema. Também é descrito um algoritmo de *branch-and-bound* baseado nesse esquema. O problema foi resolvido para instâncias com até 30 clientes. O trabalho proposto por Sahni e Gonzalez (1976), que pode ser equiparado a um MLP, demonstra que o MLP é NP-difícil.

Desde então, muitos outros artigos sobre o tema foram publicados. Bianco, Mingozzi e Ricciardelli (1993) propuseram uma abordagem similar a Lucena (1990), utilizando programação não linear e relaxação lagrangeana. Fischetti, Laporte e Martello (1993) propuseram uma abordagem baseada em programação linear inteira, obtendo resultados ótimos para instâncias com até 60 vértices. Outras formulações de programação linear e algoritmos exatos podem ser encontrados na literatura, como os propostos por Eijl (1995), Méndez-Díaz, Zabala e Lucena (2008) e Naeni e Salehipour (2019). A formulação de Naeni e Salehipour (2019) de programação inteira mista, baseada no modelo proposto por Fischetti, Laporte e Martello (1993), tem o desempenho superior às outras em termos de resultados, obtendo 49 soluções ótimas dentre as 70 instâncias propostas na literatura. Além disso, Bulhões, Sadykov e Uchoa (2018) incorporam técnicas de relaxação de *ng-path*. Experimentos computacionais mostraram melhorias significativas em relação ao estado-da-arte, conseguindo atingir novos ótimos para várias instâncias.

Vários algoritmos de aproximação para o MLP foram propostos. Blum et al. (1994), por exemplo, propuseram um algoritmo de aproximação para um problema de escalonamento de requisições a servidores. O método faz uso de uma redução a árvores geradoras, obtendo um fator constante de aproximação para espaços métricos. Desde então, subsequentes melhoras à taxa de aproximação obtida foram propostas por Goemans e Kleinberg (1998), Chaudhuri et al. (2003), e Frederickson e Wittman (2007), com o último tendo o melhor fator de aproximação na literatura. Por fim, Wu, Huang e Zhan (2004) apresentam um algoritmo exato combinando programação dinâmica e *branch-and-bound*.

Muitas abordagens heurísticas para o MLP também foram propostas. No geral, tais abordagens fazem uso de estratégias similares, dentre as quais constam a busca com vizinhança variável (VNS, do inglês *Variable Neighborhood Search*), o procedimento de

busca gulosa adaptativa aleatória (GRASP, do inglês *Greedy Random Adaptative Search Procedure*) (CLARKE; WRIGHT, 1964), descida em vizinhança variável (VND, do inglês *Variable Neighborhood Descent*) e a meta-heurística *iterated local search* (ILS) (LOURENÇO; MARTIN; STÜTZLE, 2019). Salehipour et al. (2011) propõem duas abordagens, GRASP+VND e GRASP+VNS, realizando experimentos em instâncias com até 1000 clientes. Silva et al. (2012), por sua vez, propôs uma meta-heurística denominada GILS-RVND, baseado na combinação de GRASP, ILS e um VND aleatório (RVND) (SUBRAMANIAN et al., 2010). A abordagem obteve melhoras quando comparada com Salehipour et al. (2011), encontrando valores ótimos para instâncias previamente em aberto.

Utilizando o VND durante a etapa de busca local, o método proposto por Mladenović, Urošević e Hanafi (2012) foi capaz de melhorar os resultados obtidos por Salehipour et al. (2011), mas ainda é pior que o GILS-RVND em termos de qualidade de solução. Ha e Duc (2013) implementa um algoritmo que combina a estratégia de busca tabu (TS, do inglês *Tabu Search* com o método VNS, que utiliza estruturas de memória auxiliares para impedir buscas não promissoras. Tal algoritmo é comparável aos melhores da literatura, porém não melhora os resultados do GILS-RVND em termos de tempo computacional. Além disso, o GILS-RVND possui um equilíbrio entre simplicidade, qualidade e tempo computacional, e, por isso, é comum que alguns trabalhos o utilizem como base. RIOS (2016), por exemplo, propõe versões do GILS-RVND com computação paralela em sistemas híbridos de CPU/GPU. Santana, Plastino e Rosseti (2019) utilizou técnicas de *data mining* em conjunção com o GILS-RVND, obtendo resultados melhores em termos de qualidade em instâncias pequenas (com até 50 clientes), melhor em termos computacionais em instâncias médias (entre 50 e 200 clientes) e melhor tanto em qualidade de solução como em tempo computacional em instâncias maiores (mais que 200 clientes).

Vários algoritmos para variantes do MLP também foram propostos. Onder, Kara e Derya (2017) propôs um modelo de programação linear inteira para uma generalização do MLP com múltiplos agentes. Akbari e Shiri (2021) propôs duas heurísticas para uma variante *online* do MLP, na qual arestas são desbloqueadas enquanto agente está em trânsito. Em contrapartida, o método desenvolvido por Mikula e Kulich (2022) consiste em escolher dentre três meta-heurísticas com diferentes parametrizações para resolver uma versão estocástica do problema.

O primeiro artigo a abordar o MLPTW foi publicado por Tsitsiklis (1992). Em particular, um caso especial do MLPTW (bem como alguns outros problemas) é provado como NP-difícil. O trabalho de Heilporn, Cordeau e Laporte (2010) é provavelmente o primeiro a definir o MLPTW abordado neste trabalho, propondo duas formulações de programação linear inteira. A primeira formulação é de fluxo, enquanto a segunda utiliza variáveis de posição associadas aos clientes, e é unida a resultados poliedrais para o caso especial em que há uma janela de tempo fechada apenas no depósito. Uma heurística baseada em busca tabu também é proposta. Mais recentemente, Ban (2021) propôs uma

meta-heurística com duas fases. Na primeira etapa, uma busca local é aplicada após um método construtivo para encontrar uma solução viável. Após isso, uma busca tabu é realizada para evitar que o algoritmo fique preso em ótimos locais. Os resultados obtidos demonstram que o algoritmo alcança soluções melhores em muitos casos. Vale ressaltar que o trabalho de Heilporn, Cordeau e Laporte (2010) considera saída tardia do depósito, enquanto Ban (2021) considerou a saída no tempo mais cedo possível. Uzun e Kara (2021) propõe quatro novas formulações matemáticas para o MLPTW e realiza experimentos computacionais comparando-as com formulações existentes. Os resultados mostram que as novas formulações são significativamente mais rápidas, conseguindo resolver instâncias com até 150 clientes de forma ótima dentro de segundos.

O caso do MLP com múltiplos agentes também foi estudado em múltiplos trabalhos. Frederickson e Wittman (2007), por exemplo, aborda duas variantes através de algoritmos de aproximação: em uma delas um subconjunto de clientes é visitado de forma a atender as janelas de tempo e maximizar o lucro, e na segunda variante, todos os clientes devem ser visitados, e a velocidade do agente é dinâmica. Um método baseado em inserções para lidar com o MLPTW com múltiplos agentes homogêneos é proposto por Meer (2000) e aplicado a um problema de controle operacional *offline* de uma frota de veículos guiados automaticamente (VGA) em escala real. Bjelić, Vidović e Popović (2013) por outro lado, resolveu uma variante do MLPTW que considera uma frota heterogênea de agentes. Foram propostos tanto um modelo de programação linear inteira quanto um algoritmo heurístico baseado em VNS unido a uma estratégia de redução de vizinhanças. Os experimentos realizados mostram que os algoritmos propostos possuem aplicabilidade em sistemas reais para resolver instâncias de pequeno e médio porte.

Este trabalho visa preencher uma lacuna na literatura ao propor uma abordagem eficiente com complexidade $O(1)$ para a avaliação da função objetivo durante a busca local para o MLPTW.

3 Metodologia

A abordagem utilizada é inspirada na meta-heurística proposta por Silva et al. (2012), chamado de GILS-RVND. É uma combinação das técnicas GRASP, ILS e RVND.

O pseudocódigo de tal abordagem pode ser visto no Algoritmo 1. No algoritmo, a linha 3 inicia o valor da função objetivo como infinito. Na linha 4 é iniciado o laço de execução do GILS-RVND, no qual uma solução inicial é gerada a cada iteração. Nas linhas 5 e 6 é realizada a construção da solução inicial, que é atribuída à melhor solução daquela iteração na linha 7. Tal solução é iterativamente melhorada através do ILS, o laço interior do algoritmo, executado por um máximo de I_{ILS} vezes (linhas 8-9). Na linha 10 é realizado o RVND e atribuído a s . Se o custo da solução s encontrada for melhor do que o custo $f(s')$, a solução é atualizada e o contador $iterILS$ é reiniciado (linhas 11-13). A solução então é perturbada, e o contador $iterILS$ é incrementado (14 e 15). Quando $iterILS$ for maior ou igual a I_{ILS} , o ILS termina. Se o custo da solução s' encontrada for menor do que o custo f^* , a melhor solução e a melhor função objetivo são atualizadas (linhas 16-18). Esse processo é repetido I_{Max} vezes, e na linha 19, a melhor solução é retornada.

Algoritmo 1 GILS-RVND

```

1: procedimento GILS-RVND( $I_{Max}, I_{ILS}, R$ )
2:    $f^* \leftarrow \infty$ 
3:    $s^* \leftarrow \{\}$ 
4:   para  $i \leftarrow 1$  to  $I_{Max}$  faça
5:      $\alpha \leftarrow$  Escolha valor aleatório em  $R$ 
6:      $s \leftarrow$  Construção( $\alpha$ )
7:      $s' \leftarrow s$ 
8:      $iterILS \leftarrow 0$ 
9:     enquanto  $iterILS < I_{ILS}$  faça
10:       $s \leftarrow$  RVND( $s$ )
11:      se  $f(s) < f(s')$  então
12:         $s' \leftarrow s$ 
13:         $iterILS \leftarrow 0$ 
14:       $s \leftarrow$  Perturbar( $s'$ )
15:       $iterILS \leftarrow iterILS + 1$ 
16:    se  $f(s') < f^*$  então
17:       $s^* \leftarrow s'$ 
18:       $f^* \leftarrow f(s')$ 
19:  devolve  $s^*$ 

```

A Seção 3.1 descreve de maneira detalhada o procedimento de construção. A Seção 3.2 apresenta a etapa de busca local do algoritmo, bem como as estruturas de vizinhança. Na Seção 3.3, as estratégias de perturbação são descritas.

3.1 Algoritmo construtivo

O procedimento construtivo utiliza uma abordagem GRASP, uma heurística construtiva que utiliza uma mistura de estratégias gulosa e aleatória para gerar soluções iniciais de qualidade razoável. É frequentemente utilizado em problemas de otimização combinatória onde a estrutura do problema não permite a aplicação direta de algoritmos de otimização exata.

Usa uma abordagem gulosa semelhante ao método do Vizinho Mais Próximo. O pseudocódigo de tal algoritmo pode ser visto no Algoritmo 2. Nas linhas 2-5 a solução é iniciada contendo apenas o depósito, é iniciada uma lista de candidatos CL que não contém o depósito e uma variável l é atribuída como a posição atual da solução. Na linha 6 o laço de construção é iniciado e continua até que CL esteja vazia. Nas linhas 7-11 seleciona-se aleatoriamente um dos $\lfloor \alpha n \rfloor$ vizinhos mais próximos ao nó atual para ser adicionado à rota, o removendo da lista de candidatos e adicionando-o a solução. A linha 12 retorna a solução encontrada.

Algoritmo 2 Construção

```

1: procedimento CONSTRUÇÃO( $\alpha$ )
2:    $s \leftarrow \{0\}$ 
3:   Inicialize uma lista com os possíveis candidatos  $CL$ 
4:    $CL \leftarrow CL - \{0\}$ 
5:    $l \leftarrow 0$ 
6:   enquanto  $CL \neq \emptyset$  faça
7:     Ordene  $CL$  em ordem ascendente conforme a distância em relação a  $r$ 
8:     Escolha  $c$  aleatoriamente entre os  $\alpha\%$  melhores em  $CL$ 
9:      $CL \leftarrow CL - \{c\}$ 
10:     $s \leftarrow s \cup \{c\}$ 
11:     $l \leftarrow c$ 
12:  devolve  $s$ 

```

3.2 Busca local

O funcionamento do algoritmo GILS-RVND ocorre ao longo de uma série de iterações, continuando até que um número máximo I_{MAX} de iterações seja alcançado, conforme apresentado no Algoritmo 1. Cada iteração individual executa o procedimento ILS, que persiste até que um total de I_{ILS} iterações sem melhoria seja registrado.

A ILS é uma meta-heurística que combina a busca local com uma perturbação aleatória. Após obter uma solução inicial, a busca local é aplicada iterativamente até que não haja mais melhorias em um número pré-definido de iterações. Em seguida, a perturbação é aplicada para escapar de ótimos locais e explorar novas regiões do espaço de busca.

A busca local, chamada de RVND, é detalhada no Algoritmo 3. É uma técnica que visa diversificar a busca explorando diferentes vizinhanças de uma solução. As linhas 2-3 iniciam as estruturas de vizinhança e de subsequências. Na linha 4 é iniciado o laço do RVND, que continua até que a estrutura de vizinhanças esteja vazia. Nas linhas 5-6 é escolhida e aplicada uma vizinhança aleatória. Caso a solução encontrada por essa vizinhança seja melhor do que a solução vigente, a solução vigente é atualizada, a estrutura de vizinhança é reinicializar e as estruturas de subsequência são atualizadas (linhas 7-11). Caso contrário, a vizinhança é removida da estrutura de vizinhanças (linha 13). Por fim, a melhor solução encontrada é retornada (linha 14).

Algoritmo 3 RVND

```

1: procedimento RVND( $s$ )
2:   Inicialize a Lista de Vizinhança  $NL$ .
3:   Inicialize estruturas de dados de subsequências.
4:   enquanto  $NL \neq \emptyset$  faça
5:     Escolha uma vizinhança  $N^{(\eta)}$  de  $NL$  aleatoriamente.
6:     Aplique a vizinhança  $N^{(\eta)}$  em  $s$  encontrando um novo melhor vizinho  $s$ 
7:     se  $f(s') < f(s)$  então
8:        $s \leftarrow s'$ .
9:        $f(s) \leftarrow f(s')$ .
10:    Reinicialize  $NL$ .
11:    Atualize estruturas de dados de subsequências.
12:   senão
13:     Remova  $N^{(\eta)}$  de  $NL$ .
14:   devolve  $s$ 

```

No final de todas as iterações, o GILS-RVND retorna a melhor solução encontrada durante sua execução. Essa solução é representada como “rota”, e a qualidade (ou custo) de cada solução é avaliada através da função objetivo f .

3.2.1 Estruturas de vizinhança

O método RVND (Algoritmo 3) é a base para a realização da busca local. Nele, é definido um conjunto de estruturas de vizinhança de tamanho s , sendo $N = N^1, N^2, N^3, \dots, N^s$ o conjunto que as representa. Sempre que uma vizinhança de N não melhora a solução vigente, o RVND opta de forma aleatória por uma diferente vizinhança dentro do mesmo conjunto para continuar a busca. As estruturas de vizinhança implementadas para a resolução são descritas a seguir:

- **Swap** ($N^{(1)}$) — Permuta dois clientes da solução.
- **Reinserção** ($N^{(2)}$) — Realoca um cliente para outra posição da solução.
- **2opt** ($N^{(3)}$) — Um segmento da solução é invertido.

- **Or-opt2** ($N^{(4)}$) — Realoca dois clientes adjacentes para outra posição da solução.
- **Or-opt3** ($N^{(4)}$) — Realoca três clientes adjacentes para outra posição da solução.

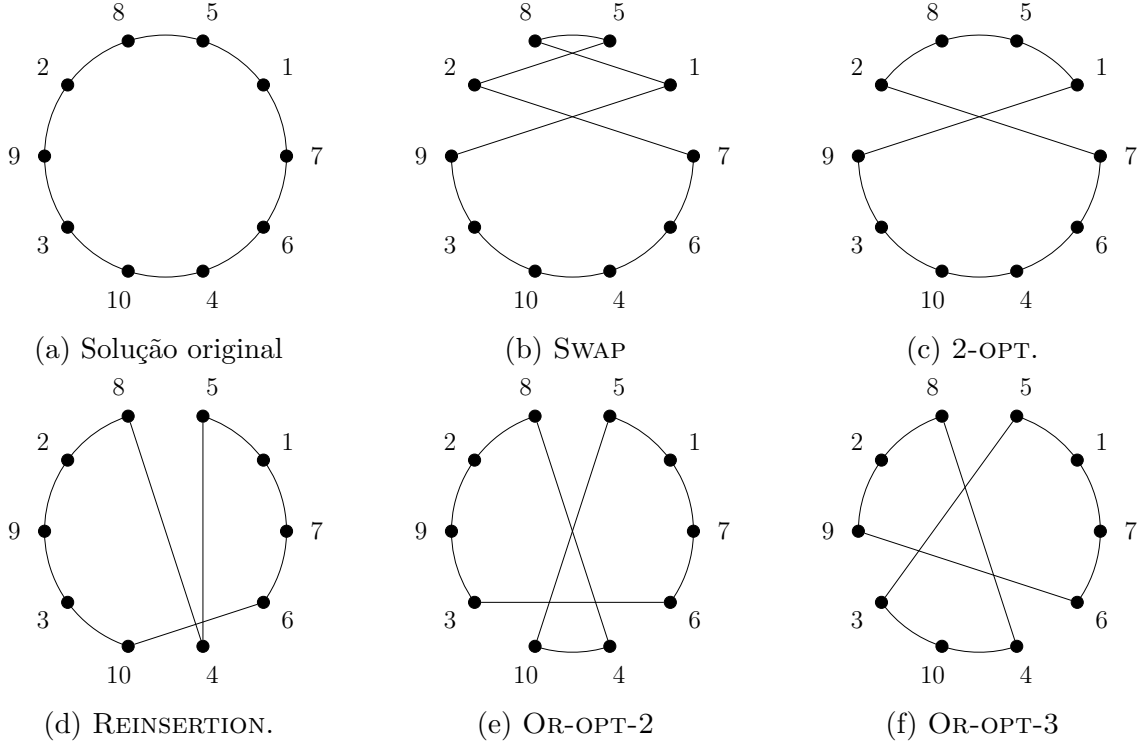


Figura 2 – Exemplos das estruturas de vizinhança utilizadas.

A Figura 2 ilustra as vizinhanças utilizadas.

3.2.2 Avaliação eficiente

Para verificar a viabilidade da solução com respeito às restrições de janela de tempo, o algoritmo emprega estruturas auxiliares baseadas no conceito de subsequências. Para isso, considere a sequência $s = (\sigma_1, \dots, \sigma_{n+1})$ uma solução viável para uma instância do MLPTW. Uma subsequência é uma sequência $\sigma = (\sigma_i, \dots, \sigma_j)$, em que $i, j \in \{1, \dots, n+1\}$. Note há exatamente $(n+1)^2$ subsequências associadas à sequência s . Ao computar previamente algumas informações para todas as subsequências associadas a uma solução, é possível checar a viabilidade de uma nova solução em tempo $\mathcal{O}(1)$ para todas as estruturas de vizinhança utilizadas. As estruturas de subsequência são baseadas nas propostas em Vidal et al. (2013), e são definidas a seguir.

- $C(\sigma)$: Custo acumulado de σ ;
- $TW(\sigma)$: Quantidade de tempo excedida do final da janela de tempo ou *time-warp* mínimo;

- $E(\sigma)$: Horário mais cedo de visita ao primeiro vértice tal que a duração e o *time-warp* sejam mínimos;
- $L(\sigma)$: Horário mais tarde de visita ao primeiro vértice tal que a duração e o *time-warp* sejam mínimos.
- $W(\sigma)$: O custo de atraso, relacionado a um atraso de uma unidade de tempo no tempo de início.

Uma solução σ é considerada viável com respeito às restrições de janela de tempo se e somente se $TW(\sigma) = 0$. Em casos nos quais $TW(\sigma) > 0$, a solução é penalizada. Mais precisamente, seja $C(\sigma)$ o custo acumulado de σ . Então, o custo total de σ durante a busca local é

$$C(\sigma) + \beta \times TW(\sigma),$$

em que β é um fator de penalização muito grande. Dessa forma, naturalmente, sempre que uma solução inviável é retornada pelos procedimentos construtivo ou de perturbação, a etapa de busca local prioriza encontrar primeiramente uma solução viável. Uma vez que isso ocorre, a busca local passa a aceitar apenas soluções viáveis, já que um movimento inviável causaria um aumento indesejável no custo total.

Considerando o caso trivial em que a subsequência σ' é composta apenas pelo vértice i , tem-se que $D(\sigma') = 0$, $TW(\sigma') = 0$, $E(\sigma') = e_i$, $L(\sigma') = l_i$, e $W(\sigma)$ assume o valor 0 se o vértice for o depósito, e 1 caso contrário. Utilizando o operador $\oplus$ para representar a concatenação de duas subsequências diferentes, observamos que uma dada subsequência σ , onde $|\sigma| > 1$, pode ser originada a partir de outras duas subsequências, σ_1 e σ_2 . Um exemplo de quebra e reconstrução de subsequências pode ser visto na Figura 3. A concatenação pode ser descrita pelas seguintes equações, conforme Vidal et al. (2013) e Silva et al. (2012):

$$D(\sigma^1 \oplus \sigma^2) = D(\sigma^1) + D(\sigma^2) + t_{\sigma^1(|\sigma^1|)\sigma^2(1)} + \Delta_{WT} \quad (3.1)$$

$$E(\sigma^1 \oplus \sigma^2) = \max\{E(\sigma^2) - \Delta, E(\sigma^1)\} - \Delta_{WT} \quad (3.2)$$

$$L(\sigma^1 \oplus \sigma^2) = \min\{L(\sigma^2) - \Delta, L(\sigma^1)\} + \Delta_{TW} \quad (3.3)$$

$$TW(\sigma^1 \oplus \sigma^2) = TW(\sigma^1) + TW(\sigma^2) + \Delta_{TW} \quad (3.4)$$

$$W(\sigma^1 \oplus \sigma^2) = W(\sigma^1) + W(\sigma^2), \quad (3.5)$$

onde:

$$\Delta = D(\sigma^1) - TW(\sigma^1) + t_{\sigma^1(|\sigma^1|)\sigma^2(1)} \quad (3.6)$$

$$\Delta_{WT} = \max\{E(\sigma^2) - \Delta - L(\sigma^1), 0\} \quad (3.7)$$

$$\Delta_{TW} = \max\{E(\sigma^1) + \Delta - L(\sigma^2), 0\}. \quad (3.8)$$

Seja t_i o tempo de atendimento do i -ésimo nó na sequência. A latência total (função objetivo) é definida como

$$\sum_{i=1}^n (t_i - t_0), \quad (3.9)$$

onde t_0 é o tempo de partida do primeiro nó. A seguinte proposição estabelece condições suficientes sob as quais é possível agendar de forma ótima os tempos de início de atendimento para qualquer solução viável.

Proposição 1. *Seja t_0 o maior tempo de partida viável do primeiro nó, e seja $t_i = \max\{t_{i-1} + c_{s_{i-1}s_i}, e_{s_i}\}$ para todo $i = 1, \dots, n$. Então, para qualquer outro conjunto viável de tempos de atendimento $t'_0, t'_1, \dots, t'_n$, temos que $t_i - t_0 \leq t'_i - t'_0$ para todo $i \in \{1, \dots, n\}$.*

Demonstração. Primeiro, nota-se que como t_0 é o maior tempo de partida viável a partir do primeiro nó, temos que $t'_0 \leq t_0$. No caso em que $t'_i \geq t_i$ para $i = 1, \dots, n$. Se deduz diretamente que

$$t_i - t_0 \leq t'_i - t'_0$$

para todo $i = 1, \dots, n$. Nosso objetivo agora é examinar o caso em que $t'_k < t_k$ para algum inteiro $k \in \{1, \dots, n\}$. Suponhamos que k is the é o menor tal inteiro. Temos então que

$$t'_k < \max\{t_{k-1} + c_{s_{k-1}s_k}, e_{s_k}\} \leq \max\{t'_{k-1} + c_{s_{k-1}s_k}, e_{s_k}\},$$

o que necessitaria que o veículo chegasse o mais cedo possível no k -ésimo cliente. Já que tal atribuição de tempos de atendimento é inviável, a demonstração está completa. $\square$

Uma consequência direta da proposição 1 é que ela nos dá uma política que minimiza a equação (3.9) para qualquer solução viável.

Proposição 2. *Dada uma sequência viável $\sigma = (\sigma_0, \dots, \sigma_m)$, seja t_0^σ o maior tempo latest de saída possível e seja $t_i^\sigma = \max\{t_{i-1}^\sigma + c_{\sigma_{i-1}\sigma_i}, e_i\}$ para todo $i = 1, \dots, m$. A seguinte equação é verdadeira:*

$$\sum_{i=1}^m (t_i^\sigma - t_0^\sigma) = \sum_{i=1}^m D(\sigma_{0,i}), \quad (3.10)$$

onde $\sigma_{i,j} = (\sigma_i, \dots, \sigma_j)$.

Demonstração. Conforme mostrado em Vidal et al. (2013), sabe-se que $L(\sigma)$ é o maior tempo de início de serviço do primeiro nó σ_0 . Assim, tem-se que $t_0^\sigma = L(\sigma)$, e os tempos de início de serviço restantes podem ser inicializados conforme estabelecido na proposição. Para alcançar o resultado desejado, prova-se que

$$t_i^\sigma - t_0^\sigma = D(\sigma_{0,i})$$

para todo $i = 1, \dots, m$. A equivalência é óbvia no caso em que $i = m$, pois segue diretamente da Proposição 1 e da definição de $D(\cdot)$.

Resta mostrar que a equivalência é verdadeira para os outros casos, isto é, quando $i = 1, \dots, m - 1$. Note que a Proposição 1 também estabelece que a escolha de tempo de serviços utilizada minimiza $t_i^\sigma - t_0^\sigma$ para qualquer inteiro i . Por isso, precisa-se apenas mostrar que o conjunto de tempos de início de serviço $t_0^\sigma, \dots, t_i^\sigma$ também causa uma duração $D(\sigma_{0,i})$ para a subsequência $\sigma_{0,i}$, em que $i = 1, \dots, m - 1$. Como mostrado em Vidal et al. (2013), uma condição suficiente é que os clientes sejam servidos o mais cedo possível (o que já é garantido), e que t_0^σ pertença ao intervalo $[E(\sigma_{0,i}), L(\sigma_{0,i})]$. É fácil mostrar que para quaisquer duas subsequências σ^1, σ^2 , as desigualdades

$$E(\sigma^1) \leq E(\sigma^1 \oplus \sigma^2) \leq L(\sigma^1 \oplus \sigma^2) \leq L(\sigma^1)$$

são válidas. A demonstração é completa através de uma simples indução em i , fazendo-se $\sigma^1 = \sigma_{0,i}$, $\sigma^2 = \sigma_{i+1,i+1}$ e lembrando-se de que $t_0^\sigma = L(\sigma) = L(\sigma^1 \oplus \sigma^2)$. $\square$

Quando combinada com a Proposição 1, a Proposição 2 nos permite computar a latência total $C(\sigma)$ para uma subsequência σ em $\mathcal{O}(1)$ de forma semelhante a de Silva et al. (2012), como mostrado na equação (3.11).

Com isso, é possível calcular o custo acumulado da subsequência $\sigma^1 \oplus \sigma^2$ como

$$C(\sigma^1 \oplus \sigma^2) = C(\sigma^1) + W(\sigma^1) \times (D(\sigma^1) + \Delta_{WT} + t_{\sigma^1 \sigma^2_1}) + C(\sigma^2) \quad (3.11)$$

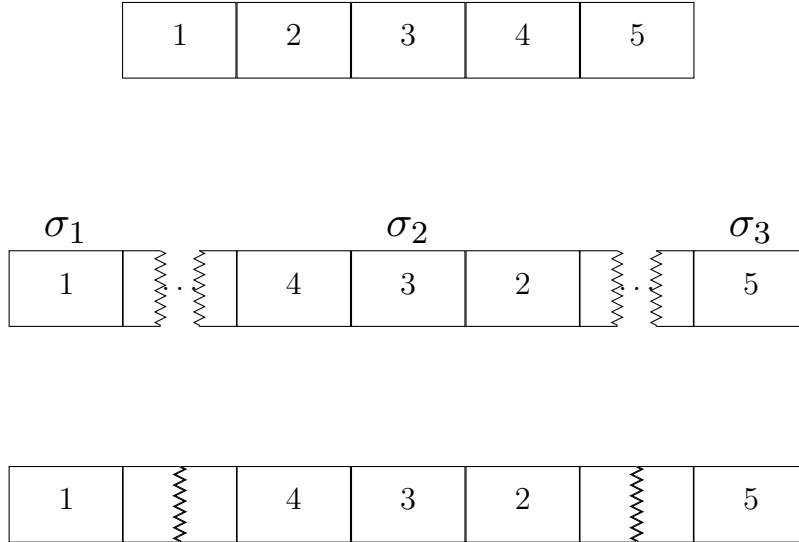


Figura 3 – Quebra de uma solução em 3 subsequências e reconstrução durante o movimento $2-opt$.

3.3 Mecanismo de perturbação

O procedimento de perturbação foi baseado no movimento *Double-Bridge* apresentado por Martin, Otto e Felten (1991). No movimento, a rota é dividida em 4 segmentos.

Em seguida, os segmentos são rearranjados de uma maneira específica, resultando em um novo padrão que destaca interações cruzadas entre os segmentos. A Figura 4 ilustra o mecanismo de perturbação utilizado em uma rota envolvendo 8 nós.

Esse movimento de perturbação é essencial para a diversificação da busca e para evitar que o algoritmo fique preso em mínimos locais. Ao introduzir aleatoriedade na estrutura da rota, o algoritmo pode explorar novas regiões do espaço de busca, potencialmente encontrando soluções de melhor qualidade. Isso é crucial em problemas de otimização, onde a busca por soluções ótimas pode ser dificultada por muitos mínimos locais.

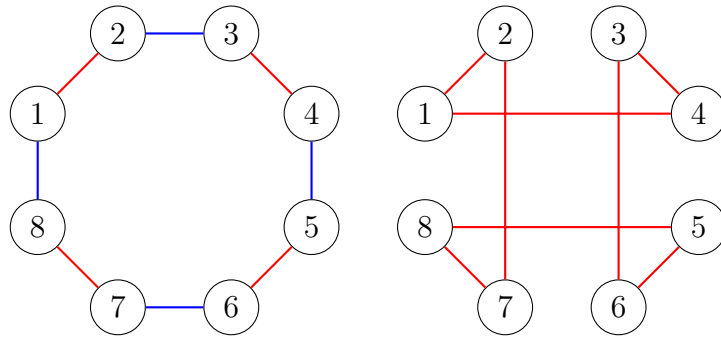


Figura 4 – Exemplo da perturbação com Double-Bridge.

4 Experimentos computacionais

Para avaliar o algoritmo meta-heurístico proposto, experimentos computacionais foram executados em um Intel Core i7-12700 com 2.10 GHz e 16GB de RAM, e sistema operacional Ubuntu 20.04. O algoritmo foi implementado em C++ (g++ 9.4.0) e todos os experimentos foram realizados em uma máquina com processador Intel i7-12700 com 4,9 GHz e 16 GB RAM e sistema operacional Linux Ubuntu 20.04.6 LTS. Durante os experimentos, utilizou-se a seguinte parametrização, baseada na proposta por Silva et al. (2012):

- $I_{Max} = 20$;
- $I_{ILS} = 200$;
- $R = \{0, 00, 0, 01, 0, 02, \dots, 0, 25\}$.

4.1 Instâncias

Os experimentos foram conduzidos em 4 conjuntos de instâncias, obtidos no site <<https://homepages.dcc.ufmg.br/~rfsilva/tsptw/>>. Os conjuntos possuem as seguintes características:

- Dumas et al. (1995) propuseram um conjunto de 135 instâncias para o VRPTW, e encontrou todos os seus ótimos. Tais instâncias têm entre 20 e 200 clientes e as janelas de tempo variam de 20 a 100 unidades de tempo.
- Gendreau et al. (1998), por outro lado, utilizaram as instâncias de Dumas et al. (1995) como base para criar 130 novas instâncias, com a diferença de que as janelas de tempo variam entre 120 e 200 unidades de tempo.
- Ohlmann e Thomas (2007) propuseram 25 novas instâncias, estendendo em 100 unidades as janelas de tempo, as instâncias de 150 e 200 clientes propostas por Dumas et al. (1995).
- Por fim, Silva e Urrutia (2010) propõem 25 novas instâncias, com mais clientes e janelas de tempo maiores.

4.2 Resultados

As Tabelas 1 e 2 a seguir mostram os resultados obtidos para os diferentes conjuntos de instâncias mostrados na seção anterior, respectivamente os conjuntos propostos por

Dumas et al. (1995) e Gendreau et al. (1998), e os propostos por Ohlmann e Thomas (2007) e Silva e Urrutia (2010). Nestas, as duas primeiras colunas apresentam informações da instâncias, isso é, o nome da instância (ID) e a quantidade de clientes que ela possui (n). As colunas TS_{best} , TS_{avg} e $TS_{T(s)}$ mostram, respectivamente, o melhor resultado, o resultado médio e o tempo computacional obtidos por Ban (2021). Foi escolhido a comparação com Ban (2021) pois o mesmo possui os melhores resultados da literatura. As próximas duas colunas apresentam o melhor e o médio resultado obtido pelo GILS-RVND. Por fim, as duas últimas colunas apresentam o gap, segundo a Equação 4.1, e o tempo em segundos. Os resultados destacados em negrito mostram instâncias em que obtivemos resultados melhores que os da literatura.

A partir da Tabela 1, pode-se observar que nosso método alcançou resultados de alta qualidade para esse grupo de instâncias, superando ou igualando o método de Ban (2021) em um número substancial de casos. Especificamente, o algoritmo GILS-RVND melhorou em relação à literatura em 49 casos e igualou em 41 casos. Nosso método não igualou ou melhorou a literatura em apenas 1 instância.

Para o segundo grupo de instâncias, também pode ser observado na Tabela 2 soluções de alta qualidade, superando ou igualando a literatura na maioria dos casos. Mais especificamente, nosso algoritmo obteve resultados melhores que a literatura em 42 casos, igualou em 27 casos e não igualou nem melhorou em 18 casos.

Não foi possível descobrir que máquina Ban (2021) utilizou para executar seus experimentos. Além disso, não está claro se o valor de tempo reportado é referente à média das execuções ou uma execução individual. Dessa forma, não foi possível realizar uma comparação entre os tempos de execução.

$$Gap(\%) = \frac{GILS-RVND_{\text{best}} - TS_{\text{best}}}{TS_{\text{best}}} \times 100 \quad (4.1)$$

Tabela 1 – Resultados para MLPTW em instâncias propostas por Dumas et al. (1995) e Silva e Urrutia (2010)

ID	n	TS_{best}	TS_{avg}	$TS_{T(s)}$	$GILS-RVND_{\text{best}}$	$GILS-RVND_{\text{avg}}$	Gap(%)	T(s)
n20w20.001	20	2528	2528	0,49	2528	2528	0,00	0,14
n20w20.002	20	2560	2560	0,54	2560	2560	0,00	0,14
n20w20.003	20	2671	2671	0,5	2671	2671	0,00	0,14
n20w20.004	20	2975	2975	0,52	2975	2975	0,00	0,14
n20w40.001	20	2270	2270	0,49	2270	2270	0,00	0,15
n20w40.002	20	2679	2679	0,53	2679	2679	0,00	0,14
n20w40.003	20	2774	2774	0,5	2774	2774	0,00	0,14
n20w40.004	20	2568	2568	0,53	2568	2568	0,00	0,14
n20w60.001	20	2421	2421	0,54	2421	2421	0,00	0,14
n20w60.002	20	2176	2176	0,54	2176	2176	0,00	0,14
n20w60.003	20	2694	2694	0,52	2694	2694	0,00	0,14
n20w60.004	20	2020	2020	0,49	2020	2020	0,00	0,19
n20w80.001	20	2990	2990	0,5	2990	2990	0,00	0,16
n20w80.002	20	2669	2669	0,55	2669	2669	0,00	0,15

Table 1: Resultados para MLPTW em instâncias propostas por Dumas et al. (1995) e Silva e Urrutia (2010)(continuação)

ID	n	TS_{best}	TS_{avg}	$TS_{T(s)}$	$GILS-RVND_{best}$	$GILS-RVND_{avg}$	Gap(%)	T(s)
n20w80.003	20	2643	2643	0,49	2643	2643	0,00	0,16
n20w80.004	20	2627	2627	0,55	2552	2552	-2,85	0,16
n20w100.001	20	2294	2294	0,52	2269	2269	-1,09	0,16
n20w100.002	20	2082	2082	0,56	2082	2082	0,00	0,15
n20w100.003	20	2416	2416	0,49	2416	2416	0,00	0,14
n20w100.004	20	2914	2914	0,52	2862	2862	-1,78	0,15
n40w20.001	40	7875	7875	2,55	7875	7875	0,00	0,86
n40w20.002	40	7527	7527	2,76	7527	7527	0,00	0,95
n40w20.003	40	7535	7535	2,77	7535	7535	0,00	0,96
n40w20.004	40	7031	7031	2,93	7031	7031	0,00	0,95
n40w40.001	40	7663	7663	2,74	7663	7663	0,00	0,90
n40w40.002	40	7104	7104	2,7	7104	7104	0,00	1,36
n40w40.003	40	7483	7483	2,84	7483	7483	0,00	0,98
n40w40.004	40	6917	6917	2,87	6915	6915	-0,03	0,98
n40w60.001	40	7066	7066	2,76	7066	7066	0,00	1,13
n40w60.002	40	7247	7247	2,67	7247	7247	0,00	1,27
n40w60.003	40	6758	6758	2,58	6758	6758	0,00	1,10
n40w60.004	40	5548	5548	2,79	5527	5527	-0,38	1,55
n40w80.001	40	8229	8229	2,63	8152	8152	-0,94	1,36
n40w80.002	40	7176	7176	2,52	7123	7123	-0,74	1,22
n40w80.003	40	7075	7075	2,88	7075	7075	0,00	1,49
n40w80.004	40	7166	7166	2,62	7166	7166	0,00	1,26
n40w100.001	40	6858	6858	2,72	6773	6773	-1,24	1,32
n40w100.002	40	6778	6778	2,84	6687	6687	-1,34	1,46
n40w100.003	40	6178	6178	2,68	6178	6178	0,00	1,60
n40w100.004	40	7019	7019	2,87	6912	6912	-1,52	1,61
n60w20.002	60	13996	14041,63	14	13996	13996	0,00	2,86
n60w20.003	60	13782	13808,25	14,8	13782	13782	0,00	3,27
n60w20.004	60	12965	12980,4	14,15	12965	12965	0,00	3,05
n60w40.003	60	15034	15081,67	14,69	15012	15013	-0,15	4,93
n60w40.004	60	12584	12605	14,48	12584	12584	0,00	3,67
n60w60.001	60	13459	13534,8	14,17	13455	13455,2	-0,03	5,06
n60w60.002	60	14506	14846,08	13,49	14506	14506,1	0,00	5,49
n60w60.003	60	12523	12738,17	14,33	12462	12462	-0,49	4,74
n60w60.004	60	13923	14060,56	13,17	13801	13801	-0,88	5,12
n60w80.001	60	13572	13850,79	14,25	13512	13512	-0,44	6,58
n60w80.002	60	13498	13990,34	14,32	13438	13438	-0,44	5,75
n60w80.003	60	13787	14096,03	14,46	13600	13600	-1,36	5,13
n60w80.004	60	12966	13141,9	14,78	12919	12923,3	-0,36	6,06
n60w100.001	60	11882	11982,27	14,96	11479	11479	-3,39	5,33
n60w100.002	60	13699	14066,78	14,54	13420	13420	-2,04	6,15
n60w100.003	60	12086	12260,93	14,16	12013	12013	-0,60	4,82
n60w100.004	60	13289	13613,33	14,86	13158	13160	-0,99	5,38
n80w20.001	80	19565	19616,09	24,54	19564	19564	-0,01	11,52
n80w20.002	80	20680	20696,5	27,56	20680	20680	0,00	9,04
n80w20.003	80	20531	20556,67	26,78	20530	20530	0,00	9,01
n80w20.004	80	20192	20267,08	26,48	20181	20181	-0,05	8,70
n80w40.001	80	20713	20861,43	26,81	20661	20661	-0,25	12,96
n80w40.003	80	19860	19895,36	26,26	19860	19860	0,00	9,75
n80w40.004	80	21179	21288,17	27,88	21172	21172	-0,03	9,71
n80w60.001	80	19429	19623,11	27,91	19366	19372,3	-0,32	14,20
n80w60.002	80	19332	19542,45	27,15	19294	19295,3	-0,20	13,30
n80w60.003	80	19315	19425,41	26,12	19257	19262,3	-0,30	13,38
n80w60.004	80	20317	20414,38	26,47	20263	20264,4	-0,27	13,72
n100w20.001	100	30604	30676,53	48,28	30604	30604	0,00	19,63

Table 1: Resultados para MLPTW em instâncias propostas por Dumas et al. (1995) e Silva e Urrutia (2010)(continuação)

ID	n	TS_{best}	TS_{avg}	$TS_{T(s)}$	$GILS-RVND_{best}$	$GILS-RVND_{avg}$	Gap(%)	T(s)
n100w20.003	100	30031	30083,54	45,17	30030	30030	0,00	21,22
n100w20.004	100	28316	28359	45,68	28316	28316	0,00	15,23
n100w40.001	100	25914	25992,51	47,6	25904	25904,9	-0,04	25,23
n100w40.002	100	28819	29070,29	47,93	28771	28773	-0,17	30,04
n100w40.003	100	28008	28121,64	47,59	28008	28008,9	0,00	27,61
n100w40.004	100	27489	27730,64	46,8	27438	27442,8	-0,19	30,99
n100w60.001	100	27308	27567,99	47,19	27260	27272,8	-0,18	30,12
n100w60.002	100	27630	27696,79	46,19	27522	27534,5	-0,39	29,10
n100w60.003	100	27418	27566,71	47,98	27354	27364,4	-0,23	28,67
n100w60.004	100	26154	26302,6	45,76	26064	26072,7	-0,34	24,64
n150w20.001	150	54150	5427,6	124,12	54140	54141,3	-0,02	83,86
n150w20.002	150	53396	5352,9	121,1	53378	53378,1	-0,03	73,80
n150w20.003	150	47715	4788,6	122,21	47703	47706,1	-0,03	81,68
n150w20.004	150	49534	4963,1	123,75	49518	49518	-0,03	67,08
n150w40.001	150	47695	4787	124,68	47674	47681	-0,04	94,54
n150w40.002	150	52804	5287,6	120,49	52770	52775,2	-0,06	90,82
n150w40.003	150	49919	5023,4	125,58	49931	49942,3	0,02	97,99
n150w40.004	150	48230	4841,8	124,65	48217	48226,5	-0,03	106,74
n200w20.001	200	78574	7870,2	171,31	78559	78565,8	-0,02	206,79
n200w20.002	200	85262	8537,7	181,9	85194	85197	-0,08	225,74
n200w20.003	200	80473	8059,6	180,77	80427	80430,4	-0,06	230,59
n200w20.004	200	76178	7629	174,9	76178	76182,8	0,00	213,23

Tabela 2 – Resultados para MLPTW em instâncias propostas por Gendreau et al. (1998) e Ohlmann e Thomas (2007)

ID	n	TS_{best}	TS_{avg}	$TS_{T(s)}$	$GILS-RVND_{best}$	$GILS-RVND_{avg}$	Gap(%)	T(s)
n20w120.001	20	2175	–	0,46	2175	2175	0,00	0,16
n20w120.002	20	2193	2193	0,51	2193	2193	0,00	0,18
n20w120.003	20	2337	2337	0,48	2337	2337	0,00	0,17
n20w120.004	20	2686	2686	0,43	2686	2686	0,00	0,16
n20w140.001	20	1846	–	0,45	1826	1826	-1,10	0,17
n20w140.002	20	2330	2330	0,5	2330	2330	0,00	0,17
n20w140.003	20	2194	2194	0,48	2194	2194	0,00	0,18
n20w140.004	20	2279	2279	0,45	2278	2278	-0,04	0,17
n20w160.001	20	2146	–	0,46	2146	2146	0,00	0,19
n20w160.002	20	1830	1830	0,49	1830	1830	0,00	0,20
n20w160.003	20	2286	2286	0,45	2286	2286	0,00	0,15
n20w160.004	20	1616	1616	0,5	1616	1616	0,00	0,14
n20w180.001	20	2477	–	0,45	2477	2477	0,00	0,16
n20w180.002	20	2315	2315	0,52	2315	2315	0,00	0,15
n20w180.003	20	2414	2414	0,46	2414	2414	0,00	0,16
n20w180.004	20	2624	2624	0,51	1924	1924	-36,38	0,17
n20w200.001	20	1975	–	0,45	1975	1975	0,00	0,17
n20w200.002	20	1799	1799	0,46	1799	1799	0,00	0,16
n20w200.003	20	2144	2144	0,44	2075	2075	-3,33	0,19
n20w200.004	20	2624	2624	0,49	2595	2595	-1,12	0,19
n40w120.001	40	6800	–	2,25	6800	6800	0,00	1,00
n40w120.002	40	6265	6265	2,84	6265	6265	0,00	1,06
n40w120.003	40	6411	6411	2,85	6411	6411	0,00	1,08
n40w120.004	40	5855	5855	2,72	5855	5855	0,00	1,87
n40w140.001	40	6290	–	2,7	6290	6290	0,00	1,18
n40w140.002	40	5746	5746	2,67	5746	5746	0,00	1,39
n40w140.003	40	6572	6572	2,71	6572	6572	0,00	1,37

Table 2: Resultados para MLPTW em instâncias propostas por Gendreau et al. (1998) e Ohlmann e Thomas (2007) (continuação)

ID	n	TS_{best}	TS_{avg}	$TS_{T(s)}$	$GILS-RVND_{best}$	$GILS-RVND_{avg}$	Gap(%)	T(s)
n40w140.004	40	5719	5719	2,64	5680	5681,1	-0,69	1,65
n40w160.001	40	6143	–	2,51	6143	6143	0,00	1,48
n40w160.002	40	6368	6368	2,91	6348	6348	-0,32	1,26
n40w160.003	40	5850	5850	2,72	5850	5850	0,00	1,17
n40w160.004	40	4468	4468	2,94	4440	4440	-0,63	1,24
n40w180.001	40	6952	–	2,7	6890	6890	-0,90	1,55
n40w180.002	40	6104	6104	2,88	6104	6104	0,00	1,62
n40w180.003	40	6040	6040	2,7	6031	6031	-0,15	1,62
n40w180.004	40	6103	6103	2,9	6283	6283	2,86	1,14
n40w200.001	40	6169	–	2,88	6084	6084	-1,40	1,22
n40w200.002	40	6674	6674	2,69	5791	5791	-15,25	1,47
n40w200.003	40	5542	5542	2,61	5230	5230	-5,97	1,18
n40w200.004	40	6103	6103	2,9	5977	5977	-2,11	1,74
n60w120.001	60	11120	–	14,73	11120	11120,3	0,00	5,96
n60w120.002	60	12517	13097,23	13,97	12517	12518,3	0,00	6,64
n60w120.003	60	11690	12157,1	14,69	11673	11675,2	-0,15	5,51
n60w120.004	60	11132	11721,03	13,42	11112	11112,4	-0,18	5,37
n60w140.001	60	10814	–	14,1	10808	10834,1	-0,06	5,95
n60w140.002	60	11782	12269,62	14,26	11749	11762	-0,28	5,55
n60w140.003	60	13128	13897,28	13,06	13031	13043,2	-0,74	9,01
n60w140.004	60	13189	13366,8	14,23	12514	12514	-5,39	3,39
n60w160.001	60	11574	–	13,72	11548	11555,8	-0,23	4,99
n60w160.002	60	12471	13790,48	13,1	12541	12598,3	0,56	6,17
n60w160.003	60	10682	11684,83	13,98	10634	10648,2	-0,45	5,37
n60w160.004	60	11645	12544,8	13,39	11599	11599	-0,40	5,52
n60w180.001	60	11363	–	13,25	11315	11315	-0,42	6,09
n60w180.002	60	12015	12629,89	13,41	11950	11950	-0,54	4,83
n60w180.003	60	12214	13018,66	13,29	11955	11972,4	-2,17	6,04
n60w180.004	60	11101	11996,46	13,38	11095	11115,5	-0,05	6,15
n60w200.001	60	10128	–	13,09	9911	9911	-2,19	5,18
n60w200.002	60	11748	12672,98	14,27	11654	11654	-0,81	5,22
n60w200.003	60	10697	12237,75	13,56	10666	10666	-0,29	3,19
n60w200.004	60	11441	12243,66	14,08	11341	11344,3	-0,88	5,53
n80w120.001	80	11122	–	21,32	16604	16693,3	33,02	12,91
n80w120.002	80	18181	18536,79	25,3	18146	18170,6	-0,19	14,02
n80w120.003	80	17878	18774,41	25,67	17812	17836,1	-0,37	12,80
n80w120.004	80	17318	18499,71	25,92	17323	17352,3	0,03	17,26
n80w140.001	80	14131	–	25,29	17888	17990,9	21,00	12,99
n80w140.002	80	17815	19278,19	24,93	17778	17805,9	-0,21	13,94
n80w140.003	80	17315	17981,11	23,57	17278	17279,1	-0,21	14,49
n80w140.004	80	18936	19902,45	21,89	18648	18689,8	-1,54	16,33
n80w160.001	80	9108	–	22,99	17298	17328,8	47,35	12,52
n80w160.002	80	17091	18281,9	21,67	16869	16873,8	-1,32	12,14
n80w160.003	80	16606	18398,57	21,15	16489	16496,4	-0,71	15,57
n80w160.004	80	17804	19203,29	25,7	17787	17789,4	-0,10	14,69
n80w180.001	80	11222	–	22,51	17089	17092,8	34,33	14,64
n80w180.002	80	17339	18272,36	22,48	17324	17329,1	-0,09	13,70
n80w180.003	80	17271	17597,4	22,66	16891	16912,8	-2,25	12,91
n80w180.004	80	16729	17594,16	23,34	16823	16878,1	0,56	16,35
n80w200.001	80	8302	–	39,02	16115	16123,8	48,48	15,01
n100w120.001	100	22269	–	47,61	31151	31328,9	28,51	21,46
n100w120.002	100	29882	31112,66	46,68	30047	30168,3	0,55	25,84
n100w120.003	100	25275	26665,72	45,88	24697	24793,5	-2,34	21,24
n100w120.004	100	30102	31010,51	46,04	30235	30363,5	0,44	25,79
n100w140.001	100	22078	–	49,53	27267	27529,6	19,03	21,69

Table 2: Resultados para MLPTW em instâncias propostas por Gendreau et al. (1998) e Ohlmann e Thomas (2007) (continuação)

ID	n	TS_{best}	TS_{avg}	$TS_{T(s)}$	$GILS-RVND_{best}$	$GILS-RVND_{avg}$	Gap(%)	T(s)
n100w140.002	100	30192	31575,11	48,38	30402	30467	0,69	24,19
n100w140.003	100	28309	29872,99	47,34	28714	28859,5	1,41	26,38
n100w140.004	100	27448	28081,39	49,56	27818	27905,1	1,33	24,01
n150w120.002	150	27192	–	120,32	72906	73051,9	62,70	101,01
n200w120.003	200	18010	–	160,2	47940	48081,3	62,43	206,79
Média		10618,85	11324,73	16,28	11957,85	11981,18	3,12	10,15

5 Considerações finais

Neste trabalho, um algoritmo meta-heurístico para resolver o problema da mínima latência com janelas de tempo foi apresentado. Em particular, propôs-se um método baseado na meta-heurística GILS-RVND, bem como um novo método para uma avaliação eficiente da função objetivo em cada busca local.

Experimentos computacionais realizados em um conjunto de 178 instâncias propostas por (DUMAS et al., 1995), (GENDREAU et al., 1998), (OHLMANN; THOMAS, 2007) e (SILVA; URRUTIA, 2010) demonstraram que o algoritmo proposto é competitivo quando comparado com os melhores da literatura. Nosso método encontrou melhores soluções para 91 instâncias e igualou em 68 casos. Além disso, a avaliação eficiente proposta

Quanto a trabalhos futuros, sugere-se estudar a aplicação de subsequências na busca local e saída tardia ao problema do MLPTW com múltiplas rotas.

Referências

AKBARI, V.; SHIRI, D. Weighted online minimum latency problem with edge uncertainty. *European Journal of Operational Research*, Elsevier BV, v. 295, n. 1, p. 51–65, nov. 2021. ISSN 0377-2217. Disponível em: <<http://dx.doi.org/10.1016/j.ejor.2021.02.038>>.

BAN, H.-B. A metaheuristic for the delivery man problem with time windows. *Journal of Combinatorial Optimization*, Springer Science and Business Media LLC, v. 41, n. 4, p. 794–816, mar. 2021. ISSN 1573-2886. Disponível em: <<http://dx.doi.org/10.1007/s10878-021-00716-2>>.

BIANCO, L.; MINGOZZI, A.; RICCIARDELLI, S. The traveling salesman problem with cumulative costs. *Networks*, Wiley, v. 23, n. 2, p. 81–91, mar. 1993. ISSN 1097-0037. Disponível em: <<http://dx.doi.org/10.1002/net.3230230202>>.

BJELIĆ, N.; VIDOVIĆ, M.; POPOVIĆ, D. Variable neighborhood search algorithm for heterogeneous traveling repairmen problem with time windows. *Expert Systems with Applications*, Elsevier BV, v. 40, n. 15, p. 5997–6006, nov. 2013. ISSN 0957-4174. Disponível em: <<http://dx.doi.org/10.1016/j.eswa.2013.05.036>>.

BLUM, A. et al. The minimum latency problem. In: *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing - STOC '94*. ACM Press, 1994. (STOC '94). Disponível em: <<http://dx.doi.org/10.1145/195058.195125>>.

BULHÕES, T.; SADYKOV, R.; UCHOA, E. A branch-and-price algorithm for the minimum latency problem. *Computers and Operations Research*, Elsevier BV, v. 93, p. 66–78, maio 2018. ISSN 0305-0548. Disponível em: <<http://dx.doi.org/10.1016/j.cor.2018.01.016>>.

CHAUDHURI, K. et al. Paths, trees, and minimum latency tours. In: *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings*. IEEE Computer Soc, 2003. (SFCS-03). Disponível em: <<http://dx.doi.org/10.1109/SFCS.2003.1238179>>.

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

DUMAS, Y. et al. An optimal algorithm for the traveling salesman problem with time windows. *Operations Research*, v. 43, n. 2, p. 367–371, 1995. Disponível em: <<https://doi.org/10.1287/opre.43.2.367>>.

EIJL. *A Polyhedral Approach to the Delivery Man Problem*. Eindhoven University of Technology, Department of Mathematics and Computing Science, 1995. (Memorandum COSOR). Disponível em: <<https://books.google.com.br/books?id=lsoXzwEACAAJ>>.

FISCHETTI, M.; LAPORTE, G.; MARTELLO, S. The delivery man problem and cumulative matroids. *Operations Research*, Institute for Operations Research and the Management Sciences (INFORMS), v. 41, n. 6, p. 1055–1064, dez. 1993. ISSN 1526-5463. Disponível em: <<http://dx.doi.org/10.1287/opre.41.6.1055>>.

FREDERICKSON, G. N.; WITTMAN, B. Approximation algorithms for the traveling repairman and speeding deliveryman problems with unit-time windows. In: _____.

Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007. p. 119–133. ISBN 9783540742081. Disponível em: <http://dx.doi.org/10.1007/978-3-540-74208-1_9>.

GENDREAU, M. et al. A generalized insertion heuristic for the traveling salesman problem with time windows. *Operations Research*, v. 46, n. 3, p. 330–335, 1998. Disponível em: <<https://doi.org/10.1287/opre.46.3.330>>.

GOEMANS, M.; KLEINBERG, J. An improved approximation ratio for the minimum latency problem. *Mathematical Programming*, Springer Science and Business Media LLC, v. 82, n. 1–2, p. 111–124, jun. 1998. ISSN 1436-4646. Disponível em: <<http://dx.doi.org/10.1007/BF01585867>>.

GUTIN, G.; PUNNEN, A. P. *The traveling salesman problem and its variations*. [S.l.]: Springer Science & Business Media, 2006. v. 12.

HA, B. B.; DUC, N. N. A meta-heuristic algorithm combining between tabu and variable neighborhood search for the minimum latency problem. In: *The 2013 RIVF International Conference on Computing and Communication Technologies - Research, Innovation, and Vision for Future (RIVF)*. IEEE, 2013. Disponível em: <<http://dx.doi.org/10.1109/RIVF.2013.6719892>>.

HEILPORN, G.; CORDEAU, J.-F.; LAPORTE, G. The delivery man problem with time windows. *Discrete Optimization*, Elsevier BV, v. 7, n. 4, p. 269–282, nov. 2010. ISSN 1572-5286. Disponível em: <<http://dx.doi.org/10.1016/j.disopt.2010.06.002>>.

LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated Local Search: Framework and Applications. In: GENDREAU, M.; POTVIN, J.-Y. (Ed.). *Handbook of Metaheuristics*. Cham: Springer International Publishing, 2019, (International Series in Operations Research & Management Science). p. 129–168. ISBN 978-3-319-91086-4. Disponível em: <https://doi.org/10.1007/978-3-319-91086-4_5>.

LUCENA, A. Time-dependent traveling salesman problem—the deliveryman case. *Networks*, Wiley, v. 20, n. 6, p. 753–763, out. 1990. ISSN 1097-0037. Disponível em: <<http://dx.doi.org/10.1002/net.3230200605>>.

MARTIN, O.; OTTO, S. W.; FELTEN, E. W. *Large-step Markov chains for the traveling salesman problem*. [S.l.]: Citeseer, 1991.

MEER, J. R. van der. *Operational Control of Internal Transport*. Tese (Doutorado) — University of Rotterdam, 2000.

MIKULA, J.; KULICH, M. Solving the traveling delivery person problem with limited computational time. *Central European Journal of Operations Research*, Springer Science and Business Media LLC, v. 30, n. 4, p. 1451–1481, jan. 2022. ISSN 1613-9178. Disponível em: <<http://dx.doi.org/10.1007/s10100-021-00793-y>>.

MLADENović, N.; UROŠEVIć, D.; HANAFI, S. Variable neighborhood search for the travelling deliveryman problem. *4OR*, Springer Science and Business Media LLC, v. 11, n. 1, p. 57–73, set. 2012. ISSN 1614-2411. Disponível em: <<http://dx.doi.org/10.1007/s10288-012-0212-1>>.

- MÉNDEZ-DÍAZ, I.; ZABALA, P.; LUCENA, A. A new formulation for the traveling deliveryman problem. *Discrete Applied Mathematics*, Elsevier BV, v. 156, n. 17, p. 3223–3237, out. 2008. ISSN 0166-218X. Disponível em: <<http://dx.doi.org/10.1016/j.dam.2008.05.009>>.
- NAENI, L. M.; SALEHIPOUR, A. A new mathematical model for the traveling repairman problem. In: *2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE, 2019. Disponível em: <<http://dx.doi.org/10.1109/IEEM44572.2019.8978898>>.
- OHLMANN, J. W.; THOMAS, B. W. A compressed-annealing heuristic for the traveling salesman problem with time windows. *INFORMS Journal on Computing*, v. 19, n. 1, p. 80–90, 2007. Disponível em: <<https://doi.org/10.1287/ijoc.1050.0145>>.
- ONDER, G.; KARA, I.; DERYA, T. New integer programming formulation for multiple traveling repairmen problem. *Transportation Research Procedia*, Elsevier BV, v. 22, p. 355–361, 2017. ISSN 2352-1465. Disponível em: <<http://dx.doi.org/10.1016/j.trpro.2017.03.042>>.
- RIOS, E. F. S. *Exploração de Estratégias de Busca Local em Ambientes CPU/GPU*. Tese (Doutorado) — UNIVERSIDADE FEDERAL FLUMINENSE, 2016.
- SAHNI, S.; GONZALEZ, T. P-complete approximation problems. *Journal of the ACM*, Association for Computing Machinery (ACM), v. 23, n. 3, p. 555–565, jul. 1976. ISSN 1557-735X. Disponível em: <<http://dx.doi.org/10.1145/321958.321975>>.
- SALEHIPOUR, A. et al. Efficient grasp+vnd and grasp+vns metaheuristics for the traveling repairman problem. *4OR*, Springer Science and Business Media LLC, v. 9, n. 2, p. 189–209, jan. 2011. ISSN 1614-2411. Disponível em: <<http://dx.doi.org/10.1007/s10288-011-0153-0>>.
- SANTANA, ; PLASTINO, A.; ROSSETI, I. *Improving a State-of-the-Art Heuristic for the Minimum Latency Problem with Data Mining*. arXiv, 2019. Disponível em: <<https://arxiv.org/abs/1908.10705>>.
- SILVA, M. M. et al. A simple and effective metaheuristic for the minimum latency problem. *European Journal of Operational Research*, v. 221, n. 3, p. 513–520, 2012. ISSN 0377-2217. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S037722171200269X>>.
- SILVA, R. F. da; URRUTIA, S. A general vns heuristic for the traveling salesman problem with time windows. *Discrete Optimization*, v. 7, n. 4, p. 203–211, 2010. ISSN 1572-5286. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1572528610000289>>.
- SUBRAMANIAN, A. et al. A parallel heuristic for the Vehicle Routing Problem with Simultaneous Pickup and Delivery. *Computers & Operations Research*, v. 37, n. 11, p. 1899–1911, nov. 2010. ISSN 0305-0548. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0305054809002779>>.
- TSITSIKLIS, J. N. Special cases of traveling salesman and repairman problems with time windows. *Networks*, Wiley, v. 22, n. 3, p. 263–282, maio 1992. ISSN 1097-0037. Disponível em: <<http://dx.doi.org/10.1002/net.3230220305>>.

UZUN, G. Önder; KARA, New formulations for the traveling repairman problem with time windows. *Expert Systems with Applications*, Elsevier BV, v. 176, p. 114863, ago. 2021. ISSN 0957-4174. Disponível em: <<http://dx.doi.org/10.1016/j.eswa.2021.114863>>.

VIDAL, T. et al. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, v. 40, n. 1, p. 475–489, 2013. ISSN 0305-0548. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0305054812001645>>.

WU, B. Y.; HUANG, Z.-N.; ZHAN, F.-J. Exact algorithms for the minimum latency problem. *Information Processing Letters*, Elsevier BV, v. 92, n. 6, p. 303–309, dez. 2004. ISSN 0020-0190. Disponível em: <<http://dx.doi.org/10.1016/j.ipl.2004.09.009>>.