

## Uma abordagem meta heurística para o problema do VRPTW

Andrey Alayo Hidalgo Schulz, Cassius Scarpin, Alisson Segatto de Souza

**Resumo:** Neste artigo propõe-se uma metodologia para resolver o problema de roteamento de veículos com janelas de tempo (VRPTW). Este problema consiste em determinar as rotas de custo mínimo que devem ser executadas por uma frota de veículos homogêneos respeitando a capacidade dos veículos e os intervalos de tempo hábeis de cada cliente, chamadas de janela de tempo. Para resolver este problema foi desenvolvido um algoritmo de duas fases. A primeira fase utiliza a meta-heurística de colônia de formigas para gerar a solução inicial para o problema, já na segunda fase aplica-se a heurística *Large Neighborhood Search* junto ao algoritmo de 2-opt e o 2-opt entre rotas para melhorar a solução encontrada durante a primeira fase. Para a análise e comparação dos resultados obtidos foram utilizadas instâncias com resultados ótimos encontrados na literatura, para os quais a abordagem proposta foi capaz de obter resultados interessantes compatíveis aos da literatura.

**Palavras chave:** VRPTW, meta-heurística, colônia de formigas, *large neighborhood search*, 2-opt.

### A meta heuristic approach to the problem of VRPTW

**Abstract:** In this paper we propose a methodology for solving the time window vehicle routing problem (VRPTW). This problem is to determine the minimum cost routes that a fleet of homogeneous vehicles must follow on the vehicle's capacity and the timely intervals of each customer, called the time window. To solve this problem a two phase algorithm was developed. The first phase uses the ant colony meta-heuristic to generate the initial solution to the problem, while in the second phase we apply the *Large Neighborhood Search* heuristic along with the 2-opt algorithm and the 2-opt between routes to improve the solution found during the first phase. For the analysis and comparison of the results obtained we used instances with optimal results found in the literature, for which the proposed approach was able to obtain interesting results compatible with those in the literature.

**Key-words:** VRPTW, metaheuristic, ant colony, *large neighborhood search*, 2-opt.

#### 1. Introdução

O VRPTW é uma variação do VRP clássico, e pode ser descrito como um problema no qual uma frota de veículos que tem como origem um depósito central deve atender um conjunto de clientes que possuem suas próprias demandas. Cada cliente deve ser atendido dentro de sua janela de tempo, que são horários predefinidos pelos mesmos para receber suas cargas.

Em problemas do mundo real isso é muito comum, pois a entrega em certos horários pode interferir com a dinâmica do estabelecimento. Um exemplo é a distribuição de produtos das indústrias fabricantes/distribuidoras de bebidas para o atacado, como restaurantes que teriam dificuldades de receber suas encomendas em horário de almoço ou casas noturnas que funcionam a partir de um certo horário, não tendo funcionários para receber o produto no período da manhã. A solução do problema consiste em encontrar o número mínimo de veículos capaz de atender todos os clientes em horários válidos, em um conjunto de rotas que minimizem a distância total percorrida.

A meta-heurística de colônia de formigas foi inspirada no comportamento real de formigas.

As formigas possuem um sistema muito eficiente para organizar seus caminhos utilizando um hormônio corporal chamado de feromônio. Quando uma formiga encontra alimento ela sempre segue novamente para o formigueiro utilizando o caminho mais curto ao procurar pelo feromônio de seu formigueiro. Quanto mais formigas usam o caminho entre o formigueiro e o alimento, maior é a concentração de feromônio no caminho, de forma que com o tempo as rotas mais longas perdem o feromônio enquanto as mais usadas, por serem mais curtas, mantem o nível de feromônio. Estas características foram eficientemente adaptadas para o problema de roteamento, obtendo ótimos resultados em vários problemas combinatórios.

Este trabalho combina um método com duas fases para resolver o problema do VRPTW. Na primeira fase utilizou-se um algoritmo de colônia de formigas adaptado para contemplar as restrições de janela de tempo do problema. Nele, ao invés de minimizar a distância total percorrida pelas formigas buscou-se maximizar o lucro total da rota, penalizando rotas que não atendem as janelas de tempo. Na segunda fase, foi usado o *Large Neighborhood Search* para melhorar randomicamente a rota. Nessa heurística foram sorteados vértices para serem retirados da rota e então reinseri-los de forma gulosa, verificando arco a arco qual a posição em que o vértice retirado pode ser reinserido de forma a maximizar o lucro obtido. Nessa mesma fase foi utilizado também as heurísticas de 2-opt e 2-opt entre rotas para realizar melhorias entre os percursos.

Este artigo se encontra dividido em seis sessões. Na sessão 2, encontra-se o referencial teórico. Já na sessão 3, pode ser visto a caracterização do problema (VRPTW). Um melhor detalhamento das heurísticas utilizadas no trabalho pode ser encontrado na sessão 4. Os experimentos computacionais são apresentados na sessão 5. Por fim, a conclusão do trabalho é apresentada na sessão 6.

## 2. Referencial Teórico

Nesta sessão serão apresentados os principais conceitos necessários para o entendimento do problema, bem como algumas variações, métodos de resolução e aplicações do mesmo.

### 2.1 VRP – *Vehicle Routing Problem*

Um fator fundamental para muitos sistemas de distribuição é o roteamento de veículos que precisam ser organizados em rotas de transporte da melhor maneira para atender clientes em uma ordem apropriada (ZENG & ZHAO, 2010).

De forma genérica, o Problema de Roteamento de Veículos (mais conhecido como *Vehicle Routing Problem* –VRP) é o nome dado à toda uma classe de problemas que envolvem a definição de rotas ótimas para uma frota de veículos que devem atender um conjunto de clientes sujeitos a certas restrições (BALDACCI et al., 2012).

Segundo Erdogan (2017) o problema de VRP é um dos mais comumente encontrados em problemas logísticos de otimização e geralmente tem como objetivo minimizar os custos das operações de transporte de uma frota de veículos que operam fora de sua base, também chamada *depot*. Este problema pode originar-se de indústrias de vários setores e em contextos de nível tático ou operacional.

Solomon (2017) define o problema de VRP de forma mais detalhada ao afirmar que este envolve a elaboração de um conjunto de rotas de veículos a custo mínimo, sendo as rotas originadas e finalizadas no *depot* (base) central, para uma frota de veículos que atendem um

conjunto de clientes com demandas conhecidas. O autor ainda acrescenta que cada cliente deve ser atendido exatamente uma vez, e além disso, todos os clientes devem ser designados a um veículo sem que a capacidade deste seja excedida.

Em 1959, Dantzig e Ramaser foram os primeiros a introduzirem o problema de despacho de caminhões. Já em 1964, Clarke e Wright generalizaram o problema para um problema de otimização linear. Este problema generalizado ficou conhecido como VRP e é um dos mais amplamente estudados no campo de Pesquisa Operacional (BRAEKERS et al., 2016).

Modelos atuais do VRP estão imensamente diferentes do que foi introduzido pelos pioneiros do tema devido à crescente incorporação de restrições da vida real ao problema, o que aumentou a complexidade de resolução (BRAEKERS et al., 2016).

Os problemas de VRP são geralmente mais simples do que os problemas da vida real. Entretanto, mesmo que algumas restrições da vida real sejam deixadas de fora, os modelos matemáticos tipicamente representam as propriedades básicas do problema, e dessa forma, providenciam resultados utilizados para análise e implementação de sistemas aplicados aos problemas reais (EL-SHERBENY, 2010).

## 2.2 Variações do VRP

O problema clássico de VRP se estende de várias maneiras ao se introduzir premissas, aspectos, complexidades, características, restrições da vida real (ERDOGAN, 2017).

Segundo Baldacci et al. (2012) as variações do VRP existem devido à natureza dos produtos transportados, a qualidade dos serviços requeridos e as características tanto de clientes como dos veículos.

Existem problemas que surgiram a partir do VRP. É o caso do PRP (Production Routing Problem) que visa otimizar as decisões de produção, estoque, distribuição e roteamento ao mesmo tempo (QIU et al., 2018).

Além deste, um problema bem conhecido na literatura que surgiu do VRP é o IRP (Inventory Routing Problem) que pode ser descrito como a combinação dos problemas de roteamento de veículos e de gestão do estoque, onde um fornecedor deve entregar produtos para um número de clientes geograficamente dispersos sujeitos a restrições (COELHO et al., 2014).

Para os problemas da família do VRP, o mais simples e mais estudado é conhecido como VRP Capacitado (CVRP) onde a restrição de capacidade do veículo é adicionada. Outras variações do VRP são relacionadas a veículos heterogêneos localizados em bases diferentes, incompatibilidade de clientes com certos tipos de veículos, clientes que aceitam a entrega somente em determinado período, horizonte de planejamento de múltiplos dias e veículos realizando múltiplas rotas (BALDACCI et al., 2012).

Somadas a essas variações estão as que consideram a composição da frota (ERDOGAN, 2017), a distribuição de produtos perecíveis (LI et al., 2015) o VRP Periódico, o VRP com janelas de tempo, o VRP Dinâmico e o VRP aberto (BRAEKERS et al., 2016).

## 2.3 VRPTW – Vehicle Routing Problem with Time Window

Uma importante variação do VRP é o VRP com janelas de tempo (VRPTW) que generaliza o VRP capacitado adicionando a condição de que cada cliente seja visitado em um determinado horário, ou intervalo de tempo (BALDACCI et al., 2012) onde esse intervalo varia de cliente para cliente (BRAEKERS et al., 2016).

Essa variação acontece por imposição dos clientes que exigem restrições de prazo limite de entrega ou restrições de entrega antes do prazo. Isso indica que o VRPTW abrange aspectos espaciais e temporais das movimentações dos veículos (SOLOMON, 1987).

Ou seja, o problema VRPTW além de incorporar a restrição de capacidade, considera o intervalo de tempo em que o cliente deve ser atendido. Assim, um veículo pode chegar antes da abertura da janela de tempo e não ser recebido pelo cliente. No sentido oposto, chegadas posteriores ao fechamento da janela de tempo também não são permitidas (EL-SHERBENY, 2010).

Nesse contexto, a incorporação das janelas de tempo faz com que o custo total do roteamento inclua não somente a distância total percorrida e os custos de tempo considerados para o roteamento, mas também o custo de espera incorrido quando um veículo chega muito cedo nas instalações de um cliente (SOLOMON, 1987).

Tendo isso em vista, a definição geral do problema VRPTW, dada por El-Sherbeny (2010), consiste em elaborar um conjunto de rotas a custo mínimo, originando e terminando em uma base central, para uma frota de veículos, que servem um conjunto de clientes com demandas conhecidas. Os clientes devem ser designados exatamente a um dos veículos de forma que a capacidade do veículo não seja excedida. O atendimento a um cliente deve começar dentro de uma janela de tempo definida entre o horário mínimo e o horário máximo em que o cliente permite o início do atendimento.

#### 2.4 Métodos de Resolução para o VRPTW

O VRPTW surgiu como uma importante área para o progresso no tratamento de restrições realistas e generalizações do modelo de roteamento básico. As janelas de tempo apareceram naturalmente em problemas encarados por organizações que trabalham com agendas de horários fixos (SOLOMON, 1987).

Para lidar com esse problema, muitos trabalhos da literatura abordaram diferentes métodos de resolução como: métodos exatos, heurísticas clássicas, meta-heurísticas, simulações, métodos de solução em tempo real (BRAEKERS et al., 2016) e inteligência artificial (EL-SHERBENY, 2010).

Alguns exemplos de métodos exatos encontrados na literatura são: métodos baseados em relaxações lagrangiana, gerações de coluna, programação dinâmica (EL-SHERBENY, 2010), *branch and price*, *branch cut and price* (BALDACCI et al., 2012), *branch and bound* (SOLOMON, 1987) e *column and cut* (ERDOGAN, 2017).

Para as heurísticas, foram constatados os seguintes exemplos: heurística de *Savings*, do vizinho mais próximo, de inserção, de varredura (SOLOMON, 1987), de construção de rotas, de aprimoramento de rota e de vizinhança (EL-SHERBENY, 2010).

El-Sherbeny (2010) afirma que a meta-heurística é um conceito de solução, pois se refere a uma estratégia iterativa mestre que guia e modifica as operações de heurísticas subordinadas por meio de uma combinação inteligente de diferentes conceitos de exploração do espaço de busca. Além do mais, alega que a meta-heurística pode manipular de forma completa ou incompleta uma solução única ou coleção de soluções a cada iteração.

As meta-heurísticas mais conhecidas são os algoritmos genéticos, a busca tabu, o recozimento simulado (LI et al., 2015). Os algoritmos de meta-heurísticas considerados o estado da arte são a pesquisa adaptativa de grande vizinhança, busca local iterativa e a pesquisa genética

híbrida unificada (ERDOGAN, 2017).

A meta-heurística híbrida ACS-LS, base para este artigo, utiliza o algoritmo *Ant Colony System* (ACS) para a construção da solução inicial e busca local (*Local Search* - LS) para refinamento das soluções é proposta por Carabetti et al. (2010).

Dentre todos os métodos, a meta-heurísticas são as mais usadas. Os métodos exatos e as heurísticas são aplicados com menos frequência, enquanto que as simulações e os métodos de solução em tempo real são raramente usados (BRAEKERS et al., 2016).

## 2.5 Aplicações do VRPTW

Os problemas de VRPTW surgem em uma ampla gama de setores industriais e variados contextos onde a maioria das aplicações deste problema são encontradas no setor logístico (ERDOGAN, 2017).

Ilustram essa situação empresas de distribuição de *commodities* (LI et al., 2015), serviços de entrega à bancos, serviços de correios, serviços coleta de lixo industrial, serviços dial-a-ride, serviços de ônibus escolar (SOLOMON, 1987), serviços de entrega de comida de restaurantes e serviços de patrulha de segurança (EL-SHERBENY, 2010).

Outros exemplos são os problemas de coleta e entrega, problemas de roteamento de veículos com múltiplas bases, roteamento de veículos com carga fracionadas, roteamento de veículos com pegada ambiental, transporte de dinheiro, entrega de pequenas mercadorias, legislação social relativa as horas trabalhadas pelos motoristas, tempos de trajeto dependentes de horários– que refletem situações de congestionamento – e informações de entrada – que refletem demandas que mudam dinamicamente ao decorrer do tempo (BRAEKERS et al., 2016). Vários desses surgem da indústria da bebida quando da distribuição dos seus produtos para o próximo elo da cadeia de suprimentos, o atacado.

Entretanto, as aplicações do VRPTW não se limitam somente a empresas com foco logístico, também são encontradas no setor de assistência médica como roteamento de enfermeiras para cuidados domésticos, transporte de sangue para central de doações e serviços de transporte de amostras biológicas coletadas de pacientes para laboratórios de teste (ERDOGAN, 2017).

## 3. Caracterização do Problema

O foco deste trabalho é o roteamento de veículos com janela de tempo (VRPTW) onde os veículos devem atender todos os clientes dentro de um certo intervalo de tempo, sendo permitido que o veículo chegue na posição do cliente antes do início de sua janela de tempo, porém, deverá esperar que esta se inicie para começar o atendimento. Em contrapartida, não é permitido que o veículo se atrase, tornado o problema inviável caso aconteça. Em algumas variações do problema esta característica passa a ser tratada como uma restrição soft, ou seja, é permitida, mas recebe uma penalidade (REINA, 2012).

O modelo matemático proposto por Larsen (1999) e Ombuki et al. (2006) para o VRPTW é demonstrado abaixo:

$$\min \sum_{i=0}^n \sum_{j=0}^n \sum_{k=0}^K c_{ij} x_{ij}^k \quad \forall i \neq j \quad (1)$$

Sujeito à

$$\sum_{k=1}^K \sum_{j=1}^N x_{ij}^k = 1 \quad (i = 1, \dots, n) \quad (2)$$

$$\sum_{i=1}^N d_i \sum_{j=0}^N x_{ij}^k \leq C \quad (k = 1, \dots, nv) \quad (3)$$

$$\sum_{j=0}^N x_{0j}^k = 1 \quad (k = 1, \dots, nv) \quad (4)$$

$$\sum_{i=0}^N x_{ih}^k - \sum_{j=0}^N x_{hj}^k = 0 \quad (k = 1, \dots, nv)(h = 1, \dots, n) \quad (5)$$

$$\sum_{i=0}^N x_{i,n+1}^k = 1 \quad (k = 1, \dots, nv) \quad (6)$$

$$b_i^k + s_i + t_{ij} - M(1 - x_{ij}^k) \leq b_j^k \quad (i, j = 1, \dots, n)(k = 1, \dots, nv) \quad (7)$$

$$e_i \leq b_i^k \leq l_i \quad (8)$$

$$x_{ij}^k \in \{0,1\} \quad (9)$$

Onde:

- $e_i$  = instante inicial da janela de tempo do cliente  $i$ ;
- $l_i$  = instante final da janela de tempo do cliente  $i$ ;
- $b_i^k$  = instante do início do atendimento do cliente  $i$  pelo veículo  $k$ ;
- $s_i$  = tempo de serviço do cliente  $i$ ;
- $t_{ij}$  = tempo de viagem do cliente  $i$  para o cliente  $j$ ;
- $M$  = um número suficientemente grande;
- $x_{ij}^k$  = igual a 1 se o veículo  $k$  percorre o arco  $(i, j)$ .

A função objetivo (1) garante que o percurso de cada veículo seja minimizado. O conjunto de restrições (2) garante que apenas um veículo faça o percurso entre dois pontos. O conjunto de restrições (3) garante que a capacidade máxima do veículo seja atendida. O conjunto de restrições (4), (5) e (6) garantem que cada veículo saia do depósito, passe pelos clientes e termine a rota novamente no depósito. As restrições (7) evitam que o veículo  $k$ , que esteja viajando do cliente  $i$  ao cliente  $j$ , não chegue ao seu destino antes do tempo de início do serviço do cliente  $i$  mais o tempo de viagem entre o cliente  $i$  e  $j$  ( $s_i + t_{ij}$ ). As restrições (8) garantem que o cliente  $i$  seja atendido dentro da sua janela de tempo.

Apesar do modelo matemático ter sido apresentado acima, o foco do trabalho é apresentar um abordagem meta heurística em duas fases para resolver o problema.

#### 4. Heurísticas Aplicadas ao VRPTW

Os problemas de VRPTW são muito complexos e com um grande número de variáveis, sendo comum o uso de métodos heurísticos para resolvê-los. Neste trabalho utilizou-se de quatro heurísticas para a solução do problema: a Colônia de Formigas, para gerar a solução inicial; o *Large Neighborhood Search* (LNS) para diversificação e as heurísticas 2-opt e 2-opt entre rotas para melhorias.

#### 4.1 Algoritmo de Colônia de Formigas

O algoritmo *Ant Colony System* (ACS) foi utilizado para encontrar a solução inicial para o problema. Para cada arco de uma rota, um valor de feromônio é introduzido, sendo que cada formiga tem a tarefa de construir uma solução factível partindo do depósito, pois como consideramos janelas de tempo a formiga não pode partir de um vértice qualquer de forma aleatória, pois interferiria nos horários dos veículos. Então a formiga constrói uma rota para o grafo, movendo-se a cada etapa de construção, do seu vértice atual para o próximo ainda não visitado. Essa escolha do caminho é feita baseando-se na quantidade de feromônio associado ao arco e a sua visibilidade (o inverso do custo associado a rota até o momento), e o equilíbrio entre ambos os parâmetros é definido como a atratividade do arco. A cada passo o arco percorrido é adicionado a solução ainda em construção. Quando não restarem mais arcos, as formigas terminam a rota, movendo-se do vértice corrente novamente para o depósito.

Para adaptar o ACS para um problema com janelas de tempo adicionamos uma penalização para o caso em que o vértice é visitado fora da janela de tempo, forçando que as formigas deem preferência aos caminhos em que atendam as janelas de tempo. O valor desta penalização é calculado com a soma de todas as distâncias entre os arcos multiplicados pelo custo de usar um veículo, conforme apresentado em Erdogan (2017). A penalização está ilustrada na equação (10).

$$\text{Penalidade} = \sum \text{Distancia} * \sum \text{CustoVeiculo} \quad (10)$$

O pseudocódigo pode ser visto no quadro a seguir:

<ul style="list-style-type: none"><li>- Inicializa a Matriz de Feromônio;</li><li>- Enquanto NumeroDelterações &lt; IteraçõesMaximas<ul style="list-style-type: none"><li>- Inicializa população de formigas;</li><li>- Para cada formiga da população<ul style="list-style-type: none"><li>- Calcula a AtratividadeDoArco = Profit*Feromônio;</li><li>- Probabilidade = AtratividadeDoArco / <math>\sum</math>AtratividadeDoArco;</li><li>- Sorteia um arco utilizando a estratégia de roleta;</li></ul></li><li>- Avalia Solução</li><li>- Algoritmo de Melhoria</li><li>- Se a solução atual for melhor que a MelhorSoluçãoEncontrada<ul style="list-style-type: none"><li>- MelhorSoluçãoEncontrada = solução atual;</li></ul></li><li>- Atualiza a matriz de Feromônio;</li></ul></li><li>- NumeroDelteração = NumeroDelteração + 1;</li></ul>
---

Quadro 1 – Algoritmo de colônia de formiga

O ACS apresentado no Quadro 1 começa inicializando a matriz de feromônio com o valor de 0,5. Em seguida inicializa uma população com 50 formigas onde cada uma irá gerar uma rota aleatória usando a atratividade do arco, probabilidade e por fim o sorteio por roleta. Após a rota estar pronta ela será avaliada e em seguida melhorada usando os algoritmos de 2-opt e 2-opt entre rotas. Se alguma das formigas encontrar uma solução melhor que a já existente, está será salva, e então a matriz de feromônio será atualizada.

#### 4.2 Large Neighborhood Search (LNS)

Em seu trabalho, Erdogan (2017) criou uma adaptação do algoritmo *Adaptive Large Neighborhood Search* de Pisinger & Ropke (2007). Esta heurística diversifica a busca,

removendo aleatoriamente alguns pontos da rota e intensifica a reinserção destes pontos usando busca local. O pseudocódigo descrito em Erdogan (2017) foi adaptado e está descrito abaixo.

Recebe uma solução inicial

- Enquanto o tempo limite não for atendido;
  - Destruição: Remove vértices aleatoriamente;
  - Reparação: Adiciona heurísticamente os vértices;
  - Melhoria: Melhora a nova rota de forma heurística;
  - Se a solução encontrada é melhor que a original
    - Solução original = solução encontrada;
  - Caso contrario
    - Solução encontrada = solução original;
- Retorna a melhor solução.

Fonte: Adaptado de Erdogan (2017)

Quadro 2 – Pseudocódigo da LNS

A LNS sorteia aleatoriamente alguns vértices para serem retirados da rota e então o reinsere novamente através de algum método heurístico, que neste trabalho foi um algoritmo guloso que verifica todas as posições em que o vértice pode ser inserido na rota, e então o faz naquela posição que apresentar o maior lucro. Após todos os vértices terem sido reinseridos, aplicasse a função de melhoria que utiliza as funções 2-opt e 2-opt entre rotas. Caso a solução obtida apresentar o maior valor de lucro que a solução anterior, esta será salva. Caso contrário, o procedimento inicia-se novamente usando a mesma solução da iteração anterior até que o tempo limite de processamento seja atendido.

### 4.3 Heurística 2-opt e 2-opt entre rotas

As operações do 2-opt consistem em remover dois arcos de uma dada solução, e então reinseri-los cruzados, onde o início do primeiro arco se conecta com o início do segundo arco e o fim do segundo arco se conectando com o final do primeiro, invertendo o sentido dos arcos localizado entre eles, conforme pode ser visto na Figura 1.

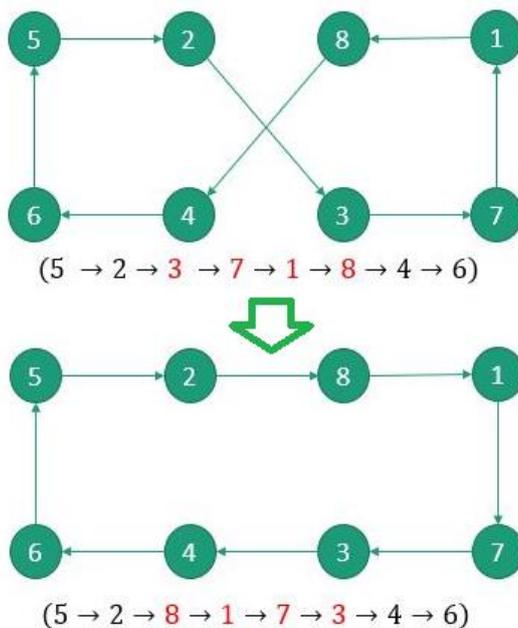


Figura 1 – Heurística 2-opt

Como pode ser visto, na Figura 1 há um cruzamento na primeira rota, então os vértices (2,3) e (8,4) são retirados da solução e reinsertidos cruzados de forma que fica (2,8) e (3,4). É possível notar que o arco entre estes dois teve seu sentido invertido de maneira que o arco (7,1) passou a ser (1,7).

O 2-opt entre rotas segue a mesma ideia vista no 2-opt simples, a grande diferença é que o cruzamento é realizado entre arcos de dois veículos diferentes, e não há inversão de sentido dos arcos, conforme a Figura 2.

Na Figura 2 podemos ver a rota de dois diferentes veículos, V1 com cinco clientes e V2 com apenas 4 clientes. Após a iteração do 2-opt parte dos clientes de V1 passa a fazer parte da rota de V2, e o contrário também acontece.

Para ter mais detalhes sobre as operações e possíveis tipos de 2-opt sugerimos que leia (GROER et al., 2010).

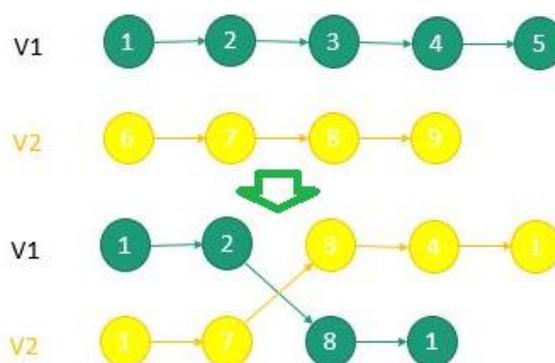


Figura 2 – Heurística 2-opt entre rotas

## 5. Experimentos Computacionais

Para realizar os experimentos computacionais utilizou-se um computador com CPU Intel Core i5-6200U e 2.4 GHz, 8 GB de RAM e com sistema operacional de 64 bits, Windows 10. Os parâmetros utilizados foram de 60 segundos para o LNS, velocidade de 50 km/h para todos os veículos, as distâncias calculadas usando distância euclidiana, um lucro de 100 unidades por cliente visitado, e um custo de 1000 unidades por veículo utilizado. As instâncias utilizadas para os testes foram as instâncias de Solomon (1987).

Essas instâncias têm servido como base de comparação entre os vários algoritmos apresentados na literatura. O conjunto é constituído por problemas com três diferentes tamanhos contendo 25, 50 e 100 clientes e um depósito central, cada um com 57 instâncias. Os problemas são separados em três grupos de acordo com a localização dos clientes. No primeiro grupo, denominado R, os clientes são distribuídos de forma aleatória. No segundo, denominado C, os clientes são agrupados em conjuntos de clusters. No último, RC, há uma mistura das duas situações anteriores. Esses grupos também são divididos com relação a duração da janela de tempo. Os grupos R1, C1 e RC1 têm janelas de tempo mais curtas, com duração entre 25% e 50% do tempo de funcionamento do depósito, já os grupos R2, C2 e RC2 possuem janelas de tempo maiores, com variação de 75% a 100% do tempo de funcionamento do depósito (SOLOMON, 1987).

Para comparar os resultados obtidos neste artigo com os resultados existentes na literatura foram escolhidos 36 instâncias apresentadas na Tabela 1.

Problema	Tamanho	Solução Ótimo		Solução Obtida	
		NV	Distância	NV	Distância
R101	25	8	617,1	8	618,46
R102	25	7	547,1	6	551,01
R201	25	4	463,3	4	464,7
R202	25	4	410,5	4	413,42
C101	25	3	191,3	3	191,81
C102	25	3	190,3	3	190,73
C201	25	2	214,7	2	215,54
C202	25	2	214,7	2	221,41
RC101	25	4	461,1	5	474,18
RC102	25	3	351,8	4	396,21
RC201	25	3	360,2	4	414,7
RC202	25	3	338	4	392,42
R101	50	12	1044	12	1030,26
R102	50	11	909	10	903,51
R201	50	6	791,9	6	813,62
R202	50	5	698,5	7	745,19
C101	50	5	362,4	5	363,24
C102	50	5	361,4	6	399,62
C201	50	3	360,2	3	388,13
C202	50	3	360,2	3	374,1
RC101	50	8	944	10	968,8
RC102	50	7	822,5	8	838,22
RC201	50	5	684,8	7	782,5
RC202	50	5	613,6	7	716,13
R101	100	20	1637,7	25	1829,70
R102	100	18	1466,6	22	1685,02
R201	100	8	1143,2	13	1315,3
R202	100	5*	1039,32*	10	1183,39
C101	100	10	827,3	10	832,63
C102	100	10	827,3	13	1068,29
C201	100	3	589,1	6	771,93
C202	100	3	589,1	5	1015,14
RC101	100	15	1619,8	17	1745,64
RC102	100	14	1457,4	17	1683,62
RC201	100	9	1261,8	11	1436,6

RC202	100	8	1092,3	10	1338,82
-------	-----	---	--------	----	---------

Tabela 1 – Resultados Computacionais

A tabela 1 está dividida nos três tamanhos das instâncias de Solomon. Para cada tamanho há 12 instâncias sendo 4 instâncias R, 4 instâncias C e 4 instâncias RC, e estas divididas em duas do tipo 1 e duas do tipo 2. Nas colunas estão representadas os números de veículos utilizados na rota (NV) e a distância percorrida.

Todas as soluções da literatura são ótimas, com exceção da instância R202 de tamanho 100 que não possui um valor ótimo, sendo o valor apresentado a melhor solução heurística apresentado na literatura.

Uma das características que dificultam a comparação das respostas é a precisão decimal. Na literatura a precisão é dada com apenas uma casa decimal, o que gera uma diferença nas distâncias encontradas. Isto também pode dar diferença na solução obtida. Como exemplo podemos citar as soluções de tamanho 25, em que as soluções da literatura possui distância um pouco inferior as obtidas neste trabalho, o que deve ser resultado desta diferença.

O oposto também acontece, nas instâncias R101 e R102 de tamanho 50 as soluções obtidas são inferiores as soluções encontradas na literatura. A justificativa para isso é que devido as distâncias não serem precisamente as mesmas o algoritmo conseguiu encontrar uma combinação de rotas diferentes que alcançassem uma solução melhor do que apresentada na literatura, mas isto só pode ser verificado com a matriz de distâncias original dos problemas.

Contudo, com exceção das instâncias de tamanho 25, o algoritmo proposto obteve soluções inferiores as da literatura, utilizando em sua maioria, mais veículos e com distâncias superiores.

## 6. Conclusão

Este trabalho propõe uma estrutura organizada em duas fases para a resolução do VRPTW. A primeira fase é composta pela meta-heurística Colônia de Formigas, que foi adaptada para englobar o problema de Janelas de Tempo com o objetivo de gerar uma solução inicial que será usada como dado de entrada para a segunda fase. Na segunda fase é utilizada a heurística *Large Neighborhood Search* para realizar melhorias na solução inicial junto das heurísticas 2-opt e 2-opt entre rotas. Para comparar a qualidade dos resultados obtidos foram testadas 36 instâncias propostas por Solomon (1987) onde as soluções encontradas ficaram muito próximas às soluções da literatura em sua grande maioria. Uma abordagem que poderia melhorar bastante a solução gerada é uma melhor adaptação dos parâmetros usados pela colônia de formigas e uma melhor padronização das distâncias euclidianas para que os resultados se tornem mais fiéis aos da literatura. Outra abordagem que poderia obter melhores soluções seria a implementação de outras heurísticas de melhoria como proposto e implementado por Erdogan (2017).

## Referências

- BALDACCI, R. et al. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. **European Journal Of Operational Research**, v. 218, n. 1, p.1-6, 2012.
- BRAEKERS, K. et al. The vehicle routing problem: State of the art classification and review. **Computers & Industrial Engineering**, v. 99, p.300-313, 2016.
- CARABETTI, E. G. et al. Metaheurística colônia de formiga aplicada ao problema de roteamento de veículos com coleta e entrega e janela de tempo. **Centro Federal de Educação Tecnológica de Minas Gerais**, 2010.
- COELHO, L. C. et al. Thirty Years of Inventory Routing. **Transportation Science**, v. 48, n. 1, p.1-19, 2014.
- EKSIOGLU, B. et al. The vehicle routing problem: A taxonomic review. **Computers & Industrial Engineering**, v. 57, n. 4, p.1472-1483, 2009.
- EL-SHERBENY, N. A. Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. **Journal Of King Saud University - Science**, v. 22, n. 3, p.123-131, 2010.
- ERDOGAN, G. An Open Source Spreadsheet for Vehicle Routing Problems. **Computers & Operation Research**, v.84, p.62-72, 2017.
- GROER, C. et al. A Library of Local Search Heuristics for the Vehicle Routing Problem. **Mathematical Programming Computation**, v.2, n.2, p.79-101, 2010.
- LARSEN, J. **Parallelization of the Vehicle Routing Problem with Time Windows**. Lyngby, 238 p. 1999. Tese (Doutorado) - Technical University of Denmark.
- LI, P. et al. Vehicle Routing Problem with Soft Time Windows Based on Improved Genetic Algorithm for Fruits and Vegetables Distribution. **Discrete Dynamics In Nature And Society**, v. 2015, p.1-8, 2015.
- OMBUKI, B. et al. Multi-objetivo Genetic Algorithm for Vehicle Routing Problem with Time Windows. **Applied Intelligence**, v.24, p.17-30, 2006.
- PISINGER, D.; ROPKE, S. A General Heuristic for Vehicle Routing Problems. **Computers & Operation Research**, v.34, n.8, p.2403-2435, 2007.
- QIU, Y. et al. Formulations and branch-and-cut algorithms for production routing problems with time windows. **Transportmetrica A: Transport Science**, v. 14, n. 8, p.669-690, 2018.
- REINA, C. D. **Roteirização de Veículos com Janelas de Tempo Utilizando Algoritmo Genético**. São Paulo, 106 p., 2012. Dissertação (Mestrado) – Universidade Federal de São Paulo.
- SOLOMON, M. M. Algorithms for Vehicle routing and Scheduling Problems with Time Window Constrains. **Operation Research**, v.35, n.2, p.254-266, 1987.
- ZENG, W.; ZHAO, Q. Study of Stochastic Demand Inventory Routing Problem with Soft Time Windows Based on MDP. **Advances in Neural Network Research & Applications**, LNEE 67, p. 193–200, 2010.