

**UNIVERSIDADE FEDERAL DO ESPIRITO SANTO
ENGENHARIA DE PRODUÇÃO**

TAMIRES NASCIMENTO PEREIRA

**VALIDAÇÃO DE UM MODELO DE ROTEIRIZAÇÃO PARA UM
PROBLEMA COM JANELAS DE TEMPO E DEPENDÊNCIA DE
TEMPO**

Vitória
2015

**UNIVERSIDADE FEDERAL DO ESPIRITO SANTO
ENGENHARIA DE PRODUÇÃO**

TAMIRES NASCIMENTO PEREIRA

**VALIDAÇÃO DE UM MODELO DE ROTEIRIZAÇÃO PARA UM
PROBLEMA COM JANELAS DE TEMPO E DEPENDÊNCIA DE
TEMPO**

Trabalho de Conclusão de Curso apresentado à
banca examinadora da Universidade Federal do
Espírito Santo para obtenção do grau de bacharel
em Engenharia de Produção.

Orientação: prof. Dr. Marta Monteiro da Costa
Cruz.

Vitória
2015

TAMIRES NASCIMENTO PEREIRA

**VALIDAÇÃO DE UM MODELO DE ROTEIRIZAÇÃO PARA UM
PROBLEMA COM JANELAS DE TEMPO E DEPENDÊNCIA DE
TEMPO**

Trabalho de Conclusão de Curso apresentado à banca examinadora da Universidade Federal do Espírito Santo para obtenção do grau de bacharel em Engenharia de Produção, sob orientação da prof. Dr. Marta Monteiro da Costa Cruz.

Aprovado em 14 de novembro de 2015

COMISSÃO EXAMINADORA

Prof^a Dr^a. Marta Monteiro da Costa Cruz
Departamento de Engenharia de Produção
Universidade Federal do Espírito Santo
Orientadora

Prof^a Dr^a. Patrícia Alcântara Cardoso
Departamento de Engenharia de Produção
Universidade Federal do Espírito Santo

Prof. Dr. Gregório Coelho de Moraes Neto
Departamento de Engenharia de Produção
Universidade Federal do Espírito Santo

DEDICATÓRIA

Dedico esse trabalho a todos que de certa forma contribuíram para a realização do meu percurso até aqui. Dedico esse resultado primeiramente à minha família, que sempre me incentivou a fazer o meu melhor em todos os meus projetos. Aos mestres que acreditaram no meu potencial e tiveram um papel fundamental na minha formação como engenheira de produção. E aos amigos que sempre me deram o suporte necessário para que eu seguisse adiante.

AGRADECIMENTOS

Registro minha profunda e eterna gratidão à minha mãe, Djanira Nascimento Pereira, que me sempre ensinou valores essenciais à superação dos desafios que vivi e que e sempre me apoiou não importa qual fosse o tamanho do meu sonho. Todas as minhas vitórias são fruto do seu amor e do seu apoio incondicionais.

À professora e orientadora Marta Cruz por me apresentar e me fazer me apaixonar pela pesquisa operacional. Agradeço pela inspiração e por me mostrar que eu era capaz de ir além de onde achava ser o meu limite, pois graças ao seu apoio hoje tenho uma formação muito mais rica.

À Professora Patrícia Cardoso por nos ensinar desde cedo valores como profissionalismo, dedicação e ousadia. Graças a ela fui capaz de quebrar paradigmas e desfrutar de experiências que nunca havia imaginado ter a chance de viver. Sou grata por cada bronca, por cada orientação e pelos momentos de descontração.

Ao professor Gregório de Moraes e a todos os demais professores da UFES que contribuíram com a minha formação, minha gratidão e minha admiração pelos seus ensinamentos e por suas dedicações.

À equipe do GERAD por me receber calorosamente em pleno inverno canadense e por me proporcionar o estudo desse trabalho.

Aos meus amigos de curso, de intercâmbios e de atividades extracurriculares, agradeço pela motivação, pela amizade e pelas valiosas experiências que tanto contribuíram para a minha formação profissional e pessoal. Agradeço em especial a Suzane Pedruzzi, Lya Lugon e Vitor de Oliveira.

A todos eles minha sincera gratidão por terem acreditado em mim e por terem me encorajado até aqui.

RESUMO

O problema de roteirização - Vehicle Routing Problem (VRP) – é amplamente estudado pela comunidade de pesquisa operacional visto que abrange problemas que levam em consideração um grande número de restrições e estão presentes em diversas situações reais. O VRP se baseia em um grafo cujos nós representam os clientes, que têm uma demanda, e os arcos representam as ruas. Todo trajeto apresenta um custo, e o objetivo do problema é visitar todos os clientes partindo e retornando ao nó principal que representa o depósito, e isso minimizando o custo total.

O planejamento e a roteirização de veículos somam custos importantes dos sistemas de transporte, e é por isso que nesse trabalho tratamos um problema que se aproxima da realidade, um problema de roteirização com dependência de tempo e janelas de tempo - TDVRPTW. Aqui a dependência de tempo admite duas premissas: (i) a velocidade depende diretamente do tipo de rua adotada. Diferenciam-se assim ruas de avenidas e autoestradas. (ii) A velocidade de uma rua varia durante o dia.

Esse projeto é uma continuação de uma dissertação de mestrado que propõe um algoritmo de resolução de um TDVRPTW usando uma metaheurística do tipo VND (Variable Neighborhood Descent) para a geração de colunas paralelizada. O objetivo desse trabalho é validar o algoritmo e propor melhorias.

Palavras chave : Roteirização com dependência de tempo e janelas de tempo (TDVRPTW), Metaheurística (VND), geração de colunas.

ABSTRACT

The Vehicle routing problem (VRP) is very studied in operational research as it respects a lot of constraints and is present in many real situations. It's based on a graph where nodes represent customers and edges represent roads between these clients. On each edge we assigned a cost that may represent distance. Each customer has a demand to satisfy. Goal is to treat all customers starting and returning to one node called deposit and minimizing the total cost.

The routing planning represents important costs in transport systems. That's why we treat a quite real routing problem witch takes in account time windows and time dependence, the TDVRPTW. The time dependency admits two premises: (i) the speed depends directly on the type of the road borrowed. We differentiate highways, boulevards and streets. (ii) The speed varies during the day.

This project gives sequence to a memory that offers a resolution for the TDVRPTW problem using a metaheuristic like VND (Variable Neighborhood Descent) for parallelized columns generation. The purpose of this project is to validate the algorithm and suggest improvements.

Keywords: Vehicle Routing, Metaheuristic, TDVRPTW, VND, Column Generation

LISTA DE FIGURAS

Figura 1 - Exemplo de uma instância de Solomon: C101 com 25 clientes.....	22
Figure 2 - Perfil de instância C101 com 25 clientes	28
Figura 4 - instância C101 com 25 clientes cenário constante Objetivo custo.....	30
Figure 5 - instância C101 com 25 clientes cenário 1 Objetivo cout	30
Figura 6 - instância C101 com 25 clientes Cenário C Objetivo Número de rotas.....	31
Figure 7 - instância C101 com 25 clientes scen1 objetivo número de rotas.....	32
Figura 8 – Distribuição das janelas de tempo dos clientes na nova versão da instância C101 com 25 clientes.....	33
Figura 9 – Resultado da resolução da nova versão da instância C101 com 25 cientes no cenário constante	34
Figura 10 - Resultado da resolução da nova versão da instância C101 com 25 cientes no cenário 2	35

LISTA DE TABELAS

Tabela 1- Perfil de velocidades do cenário 2	23
Tabela 2 – Lista de clientes mais urgentes da instância C101 com 25 clientes	29
Tabela 3 – Perfil de velocidade do cenário 1	29
Tabela 4 – Resultado da resolução da instância C101 com 25 clientes sob objetivo custo.....	31
Tabela 5 - Resultado da resolução da instância C101 com 25 clientes sob objetivo número de rotas	32

SUMÁRIO

1	INTRODUÇÃO.....	11
1.1	OBJETIVO.....	12
2	PESQUISA BIBLIOGRÁFICA	13
2.1	O TDVRPTW – TIME DEPENDENT VEHICLE ROUTING PROBLEM WITH TIME WINDOW	13
2.2	A GERAÇÃO DE COLUNAS	16
2.3	O VND - <i>VARIABLE NEIGHBORHOOD DESCENT</i>	18
3	METODOLOGIA DE PESQUISA	21
4	O ALGORITMO DE OTIMIZAÇÃO.....	22
4.1	O METODO DE RESOLUÇÃO POR GERAÇÃO DE COLUNAS.....	23
4.1.1	<i>Configuração do problema</i>	25
5	ANÁLISE DOS RESULTADOS.....	28
5.1	EVIDENCIAÇÃO DA INFLUENCIA DO TIPO DE VIA.....	29
5.2	EVIDENCIAÇÃO DA INFLUENCIA DOS DIFERENTES PERFIS DE VELOCIDADE E JANELAS DE TEMPO	32
5.3	PROPOSIÇÃO DE PONTOS DE MELHORIA NO ALGORITMO DE OTIMIZAÇÃO	35
5.3.1	<i>Erro de segmentação</i>	35
5.3.2	<i>Exibição do resultado mesmo se todos os clientes não foram visitados</i>	36

1 INTRODUÇÃO

A busca da melhor eficiência é o objetivo de toda organização e de diversos problemas do cotidiano. Essa maior eficiência pode ser representada pela redução de custos, redução de tempos gastos em tarefas ou redução de recursos empregados, enfim, busca-se a minimização dos desperdícios. A pesquisa operacional se mostra uma eficiente ferramenta para responder de forma exata problemas como esses.

Como apresentado por Figliozzi (2012), o VRP é um problema clássico de roteirização primeiramente lançado por Malandraki (1989) e Malandraki e Daskin (1992) utilizando uma formulação de programação linear mista. Ele consiste em um problema onde uma frota de veículos partindo de um depósito visita um conjunto de clientes que possuem uma demanda conhecida e fazem parte da sua rede (grafo). Nessa rede, a distância entre clientes e entre clientes e o depósito são conhecidas, o que é convertido em custo. Assim, o problema busca determinar, para cada veículo, a rota de clientes que ele deve visitar. Cada um desses itinerários deve ter como partida e fim o depósito e deve também respeitar a capacidade volumétrica dos veículos. Cada cliente deve fazer parte de um percurso.

O *Vehicle Routing Problem with Time Windows* – VRPTW é um caso especial do VRP, onde cada cliente possui uma janela de tempo. Ela representa o momento mais cedo e o momento mais tarde no qual uma visita de um veículo pode começar. Assume-se também que um veículo pode chegar mais cedo ao cliente e esperar o início de uma janela mas não pode chegar depois do fim da mesma. O número de veículos empregados é livre e as variáveis de decisão são as rotas com a lista de clientes a serem visitados em ordem cronológica.

Uma outra extensão pouco estudada do VRP, o *Time-Dependant Vehicle Routing Problem* - TDVRP, considera que a velocidade de percurso do veículo pode variar de acordo com o momento do dia e o tipo de via adotada. Em outras palavras, o momento em que o veículo entra em uma via influencia a sua velocidade e então o tempo para percorrer um trajeto. O interesse principal dessa extensão é de tomar em consideração os problemas de engarrafamento das grandes cidades devido ao aumento do tráfego em horários de pico e assim evitar que os veículos passem pelas via mais congestionadas. Dessa forma, é possível demonstrar o aspecto

dinâmico do tempo de percurso na rede e assim melhorar consideravelmente a eficiência das empresas e transportes, por exemplo empresas especializadas em entregas rápidas.

1.1 Objetivo

Esse trabalho apresenta um método de validação de um modelo de resolução de um TDVRPTW, que considera ao mesmo tempo ambas extensões apresentadas acima. Isso torna o problema consideravelmente mais realista visto que ele toma em consideração o fato que a velocidade de uma via não é constante e, principalmente, que as vias não são idênticas. Serviços de entrega de encomendas na residência do cliente e serviços de frete escolar são exemplos que ilustram aplicações reais do TDVRPTW.

Este trabalho dá sequência à dissertação de mestrado de Vincent Huart (2013), no qual ele construiu um algoritmo de resolução do TDVRPTW. Esse algoritmo foi testado apenas poucas vezes sobre poucas instâncias de Solomon, que são redes fictícias de clientes e depósito e estão disponíveis na internet sob forma de documento de texto. Essas instâncias são referência para aplicações de problemas do tipo VRPTW. Essa dúvida quanto à validação do modelo nos motivou a criar um conjunto de instâncias permitindo provar a sua eficácia e seu respeito às restrições impostas.

Este trabalho se divide da seguinte forma: As bases do TDVRPTW serão apresentadas em um primeiro momento, seguidas de uma apresentação do método de geração de colunas na segunda parte. A terceira seção será dedicada à apresentação da metodologia de pesquisa e a quarta seção discutirá os resultados obtidos. Finalmente, as conclusões serão apresentadas.

2 PESQUISA BIBLIOGRÁFICA

2.1 O TDVRPTW – Time Dependent Vehicle Routing Problem with Time Window

O problema de roteirização de veículos (VRP - *Vehicle Routing Problem*) foi criado há mais de cinco décadas por Dantzig e Ramser (1959) e é baseado no problema do cacheiro viajante, o qual busca um itinerário passante entre diversas cidades de forma a percorrer a distância mínima. Segundo os autores, ele consiste em um problema onde uma frota de veículos partindo de um depósito visita um conjunto de clientes que possuem uma demanda conhecida e fazem parte da sua rede (grafo). Nessa rede, a distância entre clientes e entre clientes o depósito são conhecidas, o que é convertido em custo. Assim, o problema busca determinar, para cada veículo, a rota de clientes que ele deve visitar. Cada um desses itinerários deve ter como partida e fim o depósito e deve também respeitar a capacidade volumétrica dos veículos. Cada cliente deve fazer parte de um e apenas um percurso. (Bodin et al. 1983)

Considerando a variável de decisão:

$$x_{ij}^k = \begin{cases} 1, & \text{se o arco } (i, j) \text{ é percorrido pelo veículo } k \\ 0, & \text{caso contrário} \end{cases}$$

O modelo matemático proposto por Bodin et al. (1983) para o problema clássico de roteirização de veículos é dado por:

$$\min \sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N \sum_{k=1}^K c_{ij} x_{ij}^k \quad (2)$$

$$\text{sujeito a: } \sum_{k=1}^K \sum_{i=1}^N x_{ij}^k = 1 \quad (j = 2, \dots, n) \quad (3)$$

$$\sum_{k=1}^K \sum_{j=1}^N x_{ij}^k = 1 \quad (i = 2, \dots, n) \quad (4)$$

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

$$\sum_{i=1}^N d_i \sum_{j=1}^N x_{ij}^k \leq C \quad (k = 1, \dots, nv) \quad (6)$$

$$\sum_{j=2}^N x_{1j}^k \leq 1 \quad (k = 1, \dots, nv) \quad (7)$$

$$\sum_{i=2}^N x_{i1}^k \leq 1 \quad (k = 1, \dots, nv) \quad (8)$$

$$x_{ij}^k \in \{0,1\} \quad (i, j = 1, \dots, n); \quad (k = 1, \dots, nv) \quad (9)$$

$$\sum_{i,j \in S} x_{ij}^k \leq |S| - 1 \quad (\forall S \subseteq \{2, \dots, n\}); \quad (k = 1, \dots, nv) \quad (10)$$

Onde:

- N = conjunto de vértices $\{0,1,\dots, n, n+1\}$ em que o subconjunto $\{1, 2, \dots, n\}$ representa os clientes e o subconjunto $\{0, n+1\}$ representa o depósito;
- K = conjunto dos veículos utilizados $\{1, \dots, nv\}$;
- C = capacidade máxima do veículo;
- S = qualquer subconjunto de vértices alocados a um veículo, excluindo-se o depósito, que não se repetem e que fazem parte de uma mesma rota;
- nv = último elemento do conjunto K de veículos;
- d_i = demanda do cliente i, e
- c_{ij} = custo da viagem do cliente i ao cliente j.

A equação (2) faz com que a distância total percorrida seja minimizada. Estamos considerando nesse problema que a frota seja homogênea. As equações

(3) e (4) garantem que um e não mais que um veículo fará a viagem entre dois pontos. A equação (5) garante que um veículo entra e sai por um mesmo vértice. A restrição (6) garante que a capacidade máxima dos veículos seja atendida. As equações (7) e (8) fazem com que cada veículo tenha uma entrada e uma saída únicas. A restrição (9) assegura que as variáveis sejam binárias e, por último, a restrição (10) assegura que o número máximo de arcos dentro de uma rota seja menor que o número de vértices de cada rota subtraído de uma unidade. Assim, evitam-se rotas que não contemplem o depósito.

Como apresentado por Figliozzi (2012), o VRP é um problema clássico de roteirização primeiramente lançado por Malandraki (1989) e Malandraki e Daskin (1992), que propuseram uma solução utilizando uma formulação de programação linear mista e uma heurística robusta para resolver o TDVRP sem janelas de tempo. Os autores também fornecem um algoritmo *branch-and-cut* para resolver pequenas instâncias.

No mesmo período, Hill e Benton (1992) iniciaram a versão do problema com dependência de tempo, focando em uma forma de representar e arquivar dados. Esse arquivamento tinha importância fundamental visto que diversos valores de distância mínima entre pares de nós deveriam ser estocados visto que essa informação variava durante o dia.

Malandraki e Dial (1996) propuseram um algoritmo com programação dinâmica para o TDTDP, que pode ser comparado a um VRP com apenas um veículo disponível. Segundo Figliozzi (2012), nesse trabalho, os autores utilizaram uma heurística do vizinho mais próximo para resolver problemas aleatórios.

O próximos importantes estudos sobre o TDVRP apareceram nos anos 2000, quando começou-se a considerar somente a dependência de tempo e não o tempo real. O primeiro artigo sobre o TDVRPTW foi publicado por Ichoua (2003), no qual os autores propunham como método de solução um *tabu-search*. Eles também impuseram a noção de FIFO entre os veículos que partem e chegam em um mesmo nó, ou seja, esses não podem se ultrapassar. Os resultados experimentais mostram que o modelo dependente do tempo fornece melhorias significativas comparado aos resultados do modelo de velocidade constante. Ichoua (2003) divide o tempo disponível em três intervalos, e por isso não é capaz de tratar a velocidade como uma função contínua.

Mais tarde, Fleischmann et al (2004) propuseram a primeira aplicação real do TDVRPTW. Seus resultados com dados da cidade de Berlim provaram que a dependência de tempo causam um significativo impacto e com isso o interesse da comunidade acadêmica sobre esse tema cresceu consideravelmente.

Em 2009, Soler (2009) propôs um algoritmo exato para resolver pequenos instâncias do TDVRPTW. Finalmente, Figliozzi (2012) apresenta um dos mais recentes artigos sobre o TDVRPTW no qual ele propõe uma construção iterativa de rotas e um algoritmo de melhoria.

No Brasil, em 2012 Reina(2012) apresentou um trabalho bastante completo utilizando o algoritmo genético para a resolução do VRPTW, também aplicando seus resultados sobre as instâncias de Solomon.

O VRP é o problema de sistemas de transporte mais conhecido em logística e tem sido largamente estudado por meios exatos e heurísticos como pode-se confirmar pelas obras de Golden (2008) e Toth (2002) por exemplo. Diversas outras variações foram propostas, como por exemplo distinguindo clientes receptores de clientes emissores, entregas com frota heterogênea de veículos, VRP com múltiplas viagens, VRP com múltiplos depósitos, etc.

2.2 A geração de colunas

A geração de colunas é uma heurística utilizada para resolver exatamente programas lineares contendo um grande número de variáveis (colunas). Ela considera todas as variáveis de maneira implícita e se baseia no princípio de decomposição matemática de Dantzig Wolfe (1960). Assim, o valor ideal para todas as variáveis de um modelo linear muito grande é obtido através da resolução sucessiva de uma série de modelos lineares contendo um pequeno subconjunto de variáveis e pela busca de colunas de melhoria. A geração de colunas aplica os passos habituais do algoritmo simplex revisado mas a busca de uma variável (ou coluna) de custo reduzido interessante se faz através da resolução de um problema de otimização separado, chamado subproblema. Dois programas são, então, associados ao programa linear original: o problema principal e o subproblema.

- O problema principal é idêntico ao programa linear original mas contém somente um número restrito de colunas.

- O problema auxiliar, ou subproblema, consiste em um outro conjunto de restrições que seleciona colunas realizáveis e as disponibiliza ao problema principal. Frequentemente, ele é formulado como um problema de otimização combinatória ou de otimização global e pode ser resolvido de maneira heurística desde que ao menos uma coluna melhoradora seja encontrada.

Assim, o valor ótimo de todas as variáveis de um modelo linear de grande ordem é obtido pela resolução sucessiva de uma série de modelos lineares contendo um conjunto restrito de variáveis e pela busca de colunas melhoradoras sob a ótica do problema principal.

A forma exata da subproblema depende do problema em consideração. Ele se formula muitas vezes como um problema de otimização combinatória ou otimização global e pode ser resolvido heurísticamente desde que ao menos uma coluna melhoradora seja encontrada. Caso contrário, um algoritmo exato deve ser utilizado para provar que nenhuma outra coluna melhoradora pode ser acrescentada ao problema principal, provando que a solução corrente para o problema principal é, na verdade, ótima para o programa linear original. Também é possível mitigar o inconveniente de nunca se ter recurso a um algoritmo exato para resolver o subproblema. Neste caso, a garantia de otimização é geralmente perdida, mas a solução resultante é muitas vezes muito boa e, por vezes, ótima.

O método de geração de colunas já foi aplicado para problemas de roteamento de veículos, por exemplo por Pessoa (2008) e Baldacci (2010) mas, a nosso conhecimento, não tem sido aplicada especificamente para TDVRPTW. A diferença fundamental de um algoritmo de geração de colunas para o problema clássico de roteirização encontra-se em formular e resolver o subproblema. Tal como acontece com o problema clássico de VRP, o subproblema do TDVRP consiste em tentar encontrar rotas mais eficazes pela resolução de um problema de minimização de distância na procura do caminho mais curto porém com limitações de recursos. No entanto, o subproblema do TDVRP também deve ter em conta a dependência no tempo. O que aumenta consideravelmente a dificuldade. As técnicas de rotulagem são as técnicas mais comuns para resolver os problemas de caminho mais curto com limitações de recursos. Trabalhos recentes sobre o caminho mais curto com a dependência do tempo de Delling (2008) e Nannicini

(2008) demonstraram que estas técnicas podem ser adaptadas de forma eficiente para ter em conta a dependência no tempo.

2.3 O VND - *Variable Neighborhood Descent*

O VND é uma metaheurística introduzida por Brinberg et al (1997) cujo objetivo é resolver problemas de otimização combinatória ou otimização global. Ele consiste uma variante do VNS - *Variable Neighborhood Search*, uma metaheurística proposta por Mladenović e Hansen (1997), que a descrevem como uma metaheurística, ou um modelo para a construção de heurísticas, com base em mudanças sistemáticas dos bairros, tanto na fase de descida para encontrar um mínimo local, como em fase de perturbação para emergir do vale correspondente (HANSEN, 2008, p.1).

A ideia geral do VNS é tentar se esquivar o máximo possível de mínimos locais obtidos por algoritmos de procura local. A busca local consiste em começar com uma solução inicial e mover-se gradualmente para as soluções vizinhas que melhoram até chegar a um ótimo local, ou seja, uma solução na qual nenhum vizinho provoca melhores resultados. Se diversos ótimos locais existem para um problema, a extensão de valores que eles cobrem pode ser grande e o ótimo global pode ser bastante diferente da média dos ótimos locais. O lançamento repetitivo de pesquisas locais não pode sozinho garantir a obtenção da solução ótima global, nem mesmo de se aproximar-se dela em termos de qualidade. A solução então é tentar se esquivar desses ótimos locais e aumentar o alcance dos ótimos explorados por meio de mudanças combinadas e apropriadas nas vizinhanças. O VNS explora as vizinhanças aproximadas e em seguida aumenta progressivamente o tamanho das vizinhanças em torno da melhor solução obtida de maneira probabilística. Assim, as características favoráveis a melhor solução são conservadas de maneira a guiar a procura em direção a soluções vizinhas em potencial.

Nesse caso em específico, alcançar uma vizinhança pode significar por exemplo trocar a ordem de visita de clientes de uma rota, trocar alguns clientes entre duas rotas ou inserir clientes de uma rota em outra rota.

O VND será utilizado no algoritmo buscando alcançar três objetivos:

1. Identificar uma boa solução inicial global para o problema, ou seja, uma distribuição aleatória de clientes em rotas.
2. Melhorar a solução inicial global encontrada até o momento.
3. Gerar colunas (rotas) potenciais para serem adicionadas ao problema principal.

De fato, essa metaheurística cria uma solução e provoca interações de pesquisa local sobre as colunas disponíveis a fim de encontrar melhores soluções.

A ordem de vizinhança possui uma grande importância pois os primeiros movimentos serão explorados mais frequentemente que os últimos. É por isso que eles são organizados em ordem de complexidade. O algoritmo para assim que todas as vizinhanças são executadas sem melhoria para o resultado atual. Além disso, para o terceiro objetivo do VND, diferentes VNDs são executados ao mesmo tempo trabalhando em paralelo em diversos servidores em diferentes máquinas.

A ideia geral do VNS consiste em tentar escapar, o máximo possível, de soluções sub-ótimas obtidas por algoritmos de pesquisa local. Para melhor compreender esse princípio, basta imaginar uma região montanhosa. Nosso objetivo é encontrar o ponto mais baixo possível, mas utilizando uma heurística comum, isso não será possível pois uma vez alcançado um mínimo local, não haveria movimentos para encontrar valores ainda menores. Para encontrar soluções ainda melhores seria necessário se permitir sair desse mínimo local, o que na analogia representaria subir uma montanha. Se vários ótimos locais existem para um mesmo problema, a extensão de valores que eles cobrem pode ser grande e o ótimo global pode então se diferenciar do valor médio de todos esses ótimos locais.

Assim, a repetição dessas pesquisas locais não garante o ótimo mas aumenta a possibilidade de achar bons resultados. Isso porque o VNS explora vizinhanças aproximadas e em seguida progressivamente o tamanho da vizinhança em torno da melhor solução obtida de maneira probabilística.

Em resumo, a implementação completa informática do algoritmo é feita em vários passos:

- (i) implementação de um algoritmo VND para encontrar um bom conjunto inicial de rotas (técnica de *warm start*);
- (ii) a implementação de uma primeira versão do algoritmo de geração de colunas para resolver a relaxação linear de TDVRP incluindo o desenvolvimento de um algoritmo VNS para resolver o subproblema;

- (iii) implantação e teste de vários métodos de ramificação;
- (iv) implementação das várias estratégias de aceleração baseadas em técnicas de estabilização.

3 METODOLOGIA DE PESQUISA

O trabalho desse projeto foi realizado durante um estágio em um centro de pesquisa, GERAD - *Groupe D'Études et de Recherche en Analyse des Décisions* - e foi dividido em duas fases :

1. Os primeiros dois meses e meio foram dedicados ao estudo do C++, da manipulação do terminal UNIX e à compreensão da dissertação de mestrado de Vincent Huart.
2. Em seguida, dois meses foram utilizados para corrigir erros de execução dos scripts até o momento em que foi possível ter certeza de ter as mesmas condições de execução que aquelas registradas no trabalho de Vincent.
3. Finalmente as seis últimas semanas foram utilizadas para testar diferentes cenários que provariam a eficácia do algoritmo de roteirização e seu respeito às restrições impostas. Para isso, nós criamos um conjunto de perfis de velocidade para as instâncias clássicas do VRPTW, as instâncias de Solomon. Um perfil de velocidade de uma instância lista a qual tipo de via pertence cada um dos arcos do grafo.

A equipe do projeto foi constituída pelo aluno de engenharia autor desse trabalho e seus dois professores tutores do laboratório, sr Sylvain Perron e sr Gilles Caporossi, professores da universidade HEC Montréal e membros do *Groupe d'Études et de Recherche en Analyse des Décisions*, GERAD. O GERAD é um centro de pesquisa interuniversitário de renome internacional reagrupando cerca de sessenta especialistas de métodos quantitativos em gestão, pesquisa operacional, informática teórica, matemática e informática matemática, principalmente vindos das instituições HEC Montreal, Polytechnique Montréal, l'Université McGill e l'Université du Québec à Montréal, UQaM.

Esse projeto foi efetuado entre 12 de fevereiro de 2015 e 05 de agosto de 2015 em sua maioria nos escritórios do GERAD situados em Montréal. O projeto igualmente contou com o suporte de Vincent Huart, o aluno que construiu o algoritmo de resolução do TDVRPTW em seu mestrado em 2013 e, uma segunda fase a partir do mês de agosto de 2015, contou com a orientação da professora Marta M. da Costa Cruz para melhorias e adaptações.

4 O ALGORITMO DE OTIMIZAÇÃO

O algoritmo criado por Vincent Huart visa otimizar problemas do tipo TDVRPTW, e isso sob dois objetivos: minimização da distância total percorrida e minimização do número de rotas. Nesse problema, inicialmente os itinerários devem respeitar as janelas de tempo de cada cliente. Assim, as instâncias de Solomon (1987) foram utilizadas como base para as simulações já que elas se aplicam perfeitamente aos problemas do tipo VRPTW. Esses arquivos são largamente conhecidos na literatura pois eles permitem construir grafos que ilustram a rede de clientes assim como suas demandas, suas janelas de tempo e seus tempos de serviço. Cada instância é representada em um arquivo de texto no qual a primeira linha exibe o número de clientes na rede e a capacidade volumétrica dos veículos disponíveis. Em seguida são listados para cada cliente o número de referência, as coordenadas, a demanda, a janela de tempo e a duração do serviço.

C101						
VEHICLE						
NUMBER	CAPACITY					
25	200					
CUSTOMER						
CUST NO.	XCOORD.	YCOORD.	DEMAND	READYTIME	DUE DATE	SERVICE
0	40	50	0	0	1236	0
1	45	68	10	912	967	90
2	45	70	30	825	870	90
3	42	66	10	65	146	90
4	42	68	10	727	782	90

Figura 1 - Exemplo de uma instância de Solomon: C101 com 25 clientes

As instâncias de Solomon são nomeadas por uma ou duas letras e uma série de três algarismos e se apresentam em três categorias:

1. CXXX cuja distribuição dos clientes na rede se dá sob a forma de concentrados;
2. RXXX cuja distribuição se apresenta de forma randômica, ou aleatória;
3. RCXXX cuja distribuição é mista.

Especificamente em um problema do TDVRPTW, as velocidades em cada via devem mudar durante o dia a fim de representar as variações de intensidade do

tráfego, e por isso 5 diferentes cenários foram criados com diferentes velocidades de acordo com o momento do dia e o tipo de via.

Em um primeiro momento, é possível considerar que a velocidade é crescente segundo o tipo de via. A tabela 1 apresenta os coeficientes de velocidade dos diferentes tipos de vias no cenário 2. As colunas dividem o dia em intervalos enquanto as linhas mostram os três tipos de vias – ruas, avenidas e estradas. Podemos concluir a partir dessa tabela que, no cenário 2, as vias do tipo 1 provocam deslocamentos lentos e constantes durante todo o dia. Além disso, nas vias dos tipos 2 e 3 existem picos de velocidade no primeiro e no último intervalos e retardamentos mais ou menos importantes durante os outros intervalos do dia. Podemos constatar que os mais importantes atrasos ocorrem entre 7h01 e 9h e entre 17h01 e 19h, justamente para representar os momentos em que há maior fluxo e mais engarrafamentos em grandes cidades.

Tabela 1- Perfil de velocidades do cenário 2

	0 – 7h [0 - 420']	7h01 – 9h [421' - 540']	9h01 – 12h [541' - 720']	12h01 – 14h [721' - 840']	14h01 – 17h [841' - 1020']	17h01 – 19h [1021' - 1140']	19h01 – 24h [1141' - 1440']
Type 1	1	1	1	1	1	1	1
Type 2	1,8	1	1,5	1,3	1,5	1	1,8
Type 3	2,5	1,5	2,0	2,2	2,0	1,5	2,5

Todos os perfis de velocidade podem ser consultados no anexo 2.

3.1 O método de resolução por geração de colunas

Nesse projeto, uma coluna representa o itinerário de um veículo da seguinte forma: uma coluna será binária do tamanho do número de clientes e conterá números 1 nos locais correspondentes aos clientes presentes no percurso.

O problema é dividido em duas partes:

- Problema principal: problema de cobertura no qual é necessário definir um conjunto de rotas que cubra todos os clientes a um custo mínimo. Ele é linear e será resolvido por CPLEX®.

- Subproblema : terá como objetivo gerar colunas realizáveis, ou seja, itinerários que respeitem as janelas de tempo dos clientes visitados, a capacidade volumétrica dos veículos empregados e as velocidades reais. Esse é um problema não linear visto que a velocidade depende do momento de partida. Assim, ele será resolvido pela metaheurística VND.

Esse projeto tem a particularidade de ter dois problemas principais, cada um para um objetivo : minimização do custo total e minimização do número de itinerários.

O algoritmo trabalha realizando os seguintes passos:

- A princípio as instâncias são transformadas em classe do C++, que recebem o número dos veículos disponíveis, capacidade desses veículos, número de clientes, vetor de clientes, vetor de arcos, matriz que reia arcos e clientes e o tempo de atendimento a esse cliente.
- Aa matriz de distâncias gera uma matriz de duração dos trajetos entre clientes e entre clientes e o depósito. Essa conversão acontece de acordo com o momento em questão e com o cenário utilizado.
- Uma heurística do tipo VND cria uma solução inicial.
- A partir desse momento os seguintes processos acontecem em paralelo rodando em diferentes processadores:
- Diversos processadores geram em paralelo colunas através de trocas entre as colunas existentes as quais são previstas da metaheurística VND, são as chamadas vizinhanças. Esse processo usa um método exato.
- Ao mesmo tempo o problema principal testa as colunas à disposição testando o valor da função objetivo.
- O algoritmo para quando a diferença entre as antigas e as novas soluções não será significativa.
- Por fim as rotas são respondidas por meio de uma classe C++, que informa os vetores com os clientes visitados em cada rota, a demanda atendida por cada rota e o numero de clientes atendidos por ela. É possível também ser informado do tempo de execução do algoritmo.

Esse algoritmo desenvolvido por Huart (2013) desenvolvido em linguagem C++, utilizou o programa CPLEX e possui mais de 15000 linhas.

3.1.1 Configuração básica do problema

As funções objetivo obedecem as seguintes premissas e restrições:

Intervalos:

$I_d \cup I_c = I$: Conjunto de vértices do grafo

I_d = Vértice depósito

I_c = Conjunto de vértices clientes

C = Quantidade de clientes na instância

Parâmetros:

D_i = Demanda de cada cliente

E_i = Quantidade entregue a cada cliente

LT_i = Lead Time de entrega a cada cliente

V = capacidade volumétrica de cada veículo

JI_i = Início da janela do cliente i

JF_i = Fim da janela do cliente i

Variáveis:

$$x_{ij} = \begin{cases} 1 & \text{se o cliente } i \text{ pertence à rota } j \\ 0 & \text{caso contrário} \end{cases}$$

VI_i = Início da visita ao cliente i

VF_i = Fim da visita ao cliente i

Condições

$$\sum_{i=1}^{I_c} \sum_{j=1}^J x_{ij} = C \quad (1)$$

$$\forall i \in I_c \quad \sum_{j=0}^J x_{ij} = 1 \quad (2)$$

$$\forall j \in J \quad \sum_{i=1}^{I_c} E_i = V \quad (3)$$

$$\forall i \in I_c \quad \sum_{j=1}^J E_i = D_i \quad (4)$$

$$\forall i \in I_c \quad VI_i \geq JI_i \quad (5)$$

$$\forall i \in I_c \quad VI_i \leq JF_i \quad (6)$$

$$\forall i \in I_c \quad JF_i = JI_i + LT_i \quad (7)$$

$$\forall i \in I_d \quad \sum_{j=0}^J x_{ij} = 2 \quad (8)$$

- (1) A soma de clientes atendidos deve ser igual à quantidade de clientes na instância.
- (2) Cada cliente deve fazer parte de uma rota.
- (3) O volume entregue em uma rota deve ser menor igual à capacidade volumétrica de cada veículo.
- (4) O volume entregue a cada cliente deve ser igual à sua demanda.
- (5) O momento de chegada do veículo ao cliente deve ser posterior ao início da sua janela de tempo.
- (6) O momento de chegada do veículo ao cliente deve ser anterior ao fim da sua janela de tempo.
- (7) O fim da visita se dá após o Lead Time do recebimento.

(8) A rota deve começar no depósito. Essa equação não atende à restrição. Ela apenas garante que o depósito pertença duas vezes a cada rota

As duas funções objetivo não puderam ser elaboradas com as variáveis apresentadas acima, mas visam os seguintes fins:

- Minimização da distância total percorrida
- Minimização do número de rotas.

4 ANÁLISE DOS RESULTADOS

Em um primeiro momento, começamos por analisar os resultados de otimização da instância C101 com 25 clientes com o objetivo de identificar na novas soluções, o efeito de mudanças em:

- tipos de vias
- perfil de velocidade e janelas de tempo

Durante a manipulação do algoritmo foi possível destacar duas grandes dificuldades de obtenção de informação: não é possível saber nem o momento no qual o cliente foi visitado nem o distinguir quando ele foi visitado ou serviu de passagem. Isso porque a resposta do algoritmo de otimização é a lista de vetores que representam as rotas a serem seguidas por cada veículo seguidas do número de clientes visitados e a demanda atendida por cada uma delas. Sendo assim, foi necessário realizar algumas manipulações, cruzamentos de dados e deduções em cima dos resultados obtidos para identificar o momento de visita de cada cliente.

As conclusões do trabalho serão comentadas em cima da otimização da instância C101 com 25 clientes. A título de ilustração, a essa instância possui seus clientes dispostos conforme descrito na figura 2:

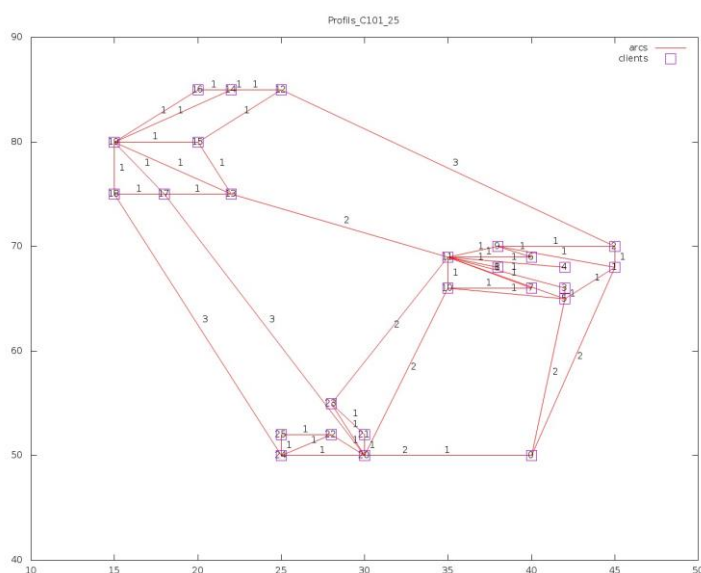


Figure 2 - Perfil de instância C101 com 25 clientes

Os clientes de maior restrição horária nessa instância são aqueles cuja janela de tempo termina logo após o seu início. Aqui, esses clientes tratados com urgência são:

Tabela 2 – Lista de clientes mais urgentes da instância C101 com 25 clientes

Cliente	Janela de tempo (em minutos)
5	[15 - 67]
13	[30 - 92]
20	[10 - 73]
24	[65 - 144]

4.1 Evidenciação da influência do tipo de via

O Cenário constante atribui a velocidade 1 a todas as vias enquanto o cenário 1 atribui a cada tipo de via velocidades constantes:

Tabela 3 – Perfil de velocidade do cenário 1

	0 – 24h [0 - 1400']
Tipo 1	1
Tipo 2	1,8
Tipo 3	2,5

Em uma primeira observação sobre as rotas obtidas pelo algoritmo e as suas especificações de instâncias, é possível confirmar que os trajetos respeitam corretamente as janelas de tempo dos clientes. Todavia, a compreensão desse evento é debilitada pelo fato que diversos clientes são visitados fora de suas janelas de tempo porque servem de passagem a outros clientes. Por exemplo esse evento foi constatado no resultado da otimização da instância C101 com 25 clientes sob objetivo minimização do custo com os clientes 1, 2, 5 e 20, como é possível observar na tabela 4.

Nesse mesmo exemplo, é possível identificar a influência da mudança de velocidade devido à mudança do cenário. De fato, como é possível se ver pela tabela 4 no cenário constante, depois da visita do cliente 25 o retorno à origem se faz pelo caminho pelos clientes 22 – 20 – 0, percorrendo-se via do tipo 1. No cenário 1 esse retorno se faz pelos clientes 24 -0 percorrendo-se vias do tipo 2. Isso mostra que a otimização dá preferência ao coeficiente de velocidade de uma via de tipo 2 em relação a um coeficiente de velocidade de um via do tipo 1. As figuras 3 e 4 ilustram as rotas apresentadas na tabela 4.

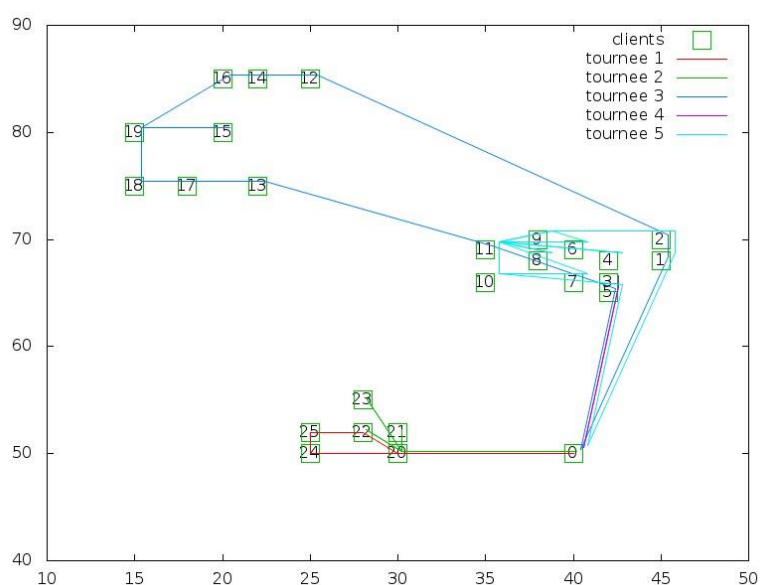


Figura 3 - instância C101 com 25 clientes cenário constante Objetivo custo

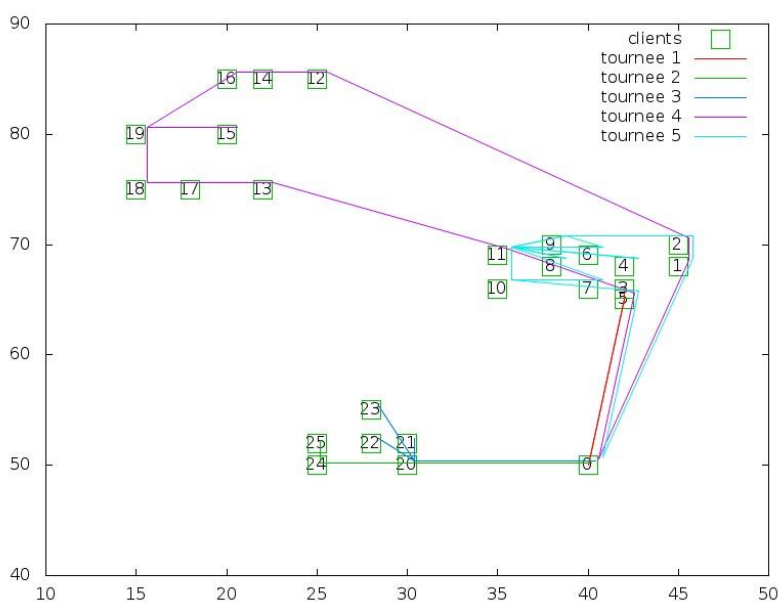


Figure 4 - instância C101 com 25 clientes cenário 1 Objetivo cout

Tabela 4 – Resultado da resolução da instância C101 com 25 clientes com objetivo minimização do custo

Cenários	Rotas
Cenário Constante	0-20-24-25-22-20-0 0-5-3-5-0 0-5-10-7-11-8-11-10-11-9-6-11-4-11-9-2-1-0 0-20-23-20-22-20-21-20-0 0-5-11-13-17-18-19-15-19-16-14-12-2-1-0
Cenário 1	0-5-3-5-0 0-20-24-25-24-0 0-20-23-20-22-20-21-20-0 0-5-11-13-17-18-19-15-19-16-14-12-2-1-0 0-5-10-7-11-8-11-10-11-9-6-11-4-11-9-2-1-0

Na otimização sob o objetivo redução do número de rotas, o resultado não é muito afetado pela mudança do cenário já que as rotas utilizam primordialmente vias do tipo 1, e essas têm o mesmo coeficiente de velocidade em ambos cenários. A partir da tabela 1, nós podemos constatar porém que os arcos do tipo 3, por exemplo os arcos 17-23 e 18-25 são mais utilizados no cenário 1 que no cenário constante. As figuras 5 e 6 mostram as rotas apresentadas na tabela 5.

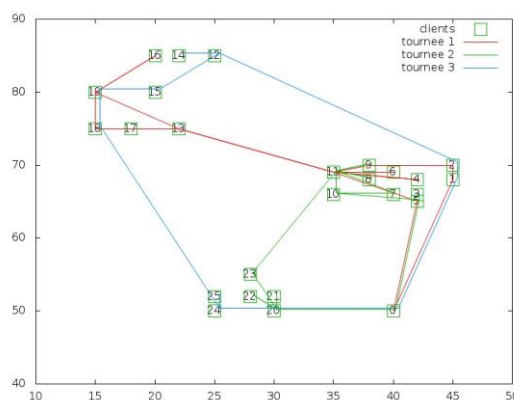


Figura 5 - instância C101 com 25 clientes Cenário C Objetivo Número de rotas

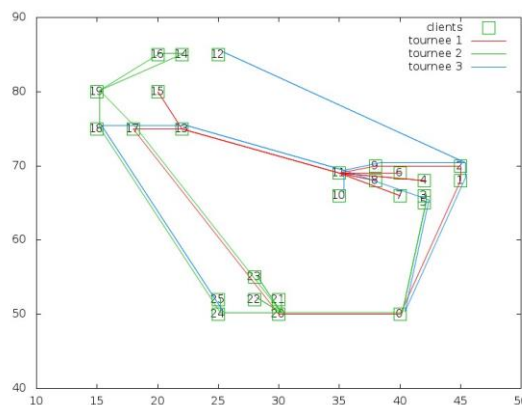


Figure 6 - instância C101 com 25 clientes scen1 objetivo número de rotas

Tabela 5 - Resultado da resolução da instância C101 com 25 clientes sob objetivo número de rotas

Cenários	Rotas
Cenário Constante	0-5-11-13-17-18-19-16-19-13-11-6-11-4-11-9-2-1-0
	0-5-3-5-10-7-11-8-11-10-11-9-11-23-20-22-20-21-20-0
	0-20-24-25-24-18-19-15-12-14-12-2-1-0
Cenário 1	0-20-17-13-11-7-11-8-11-13-15-13-11-6-11-4-11-9-2-1-0
	0-5-3-5-0-24-25-24-18-19-16-14-19-17-20-23-20-22-20-21-20-0
	0-5-11-13-17-18-24-18-17-13-11-10-11-9-2-12-2-1-0

4.2 Evidenciação da influência dos diferentes perfis de velocidade e janelas de tempo

A fim de vencer a dificuldade de obtenção de dados e provar a influência dos cenários no resultado, nós modificamos uma versão da instância C101 com 25 clientes seguindo as adaptações seguintes:

- Alguns clientes tiveram suas janelas de tempo alteradas de forma a serem localizadas no início do intervalo ao qual pertencem.
- Os clientes que funcionam como hub e que são muito utilizados como passagem receberam uma demanda diferente de um múltiplo de 10 para que sejam facilmente identificados na rota a que pertencem. Essa modificação está visível apenas no documento de texto que define a instância, não na figura abaixo.

- Todos os outros clientes tiveram suas janelas estendidas para poderem ser visitados em qualquer momento do dia.

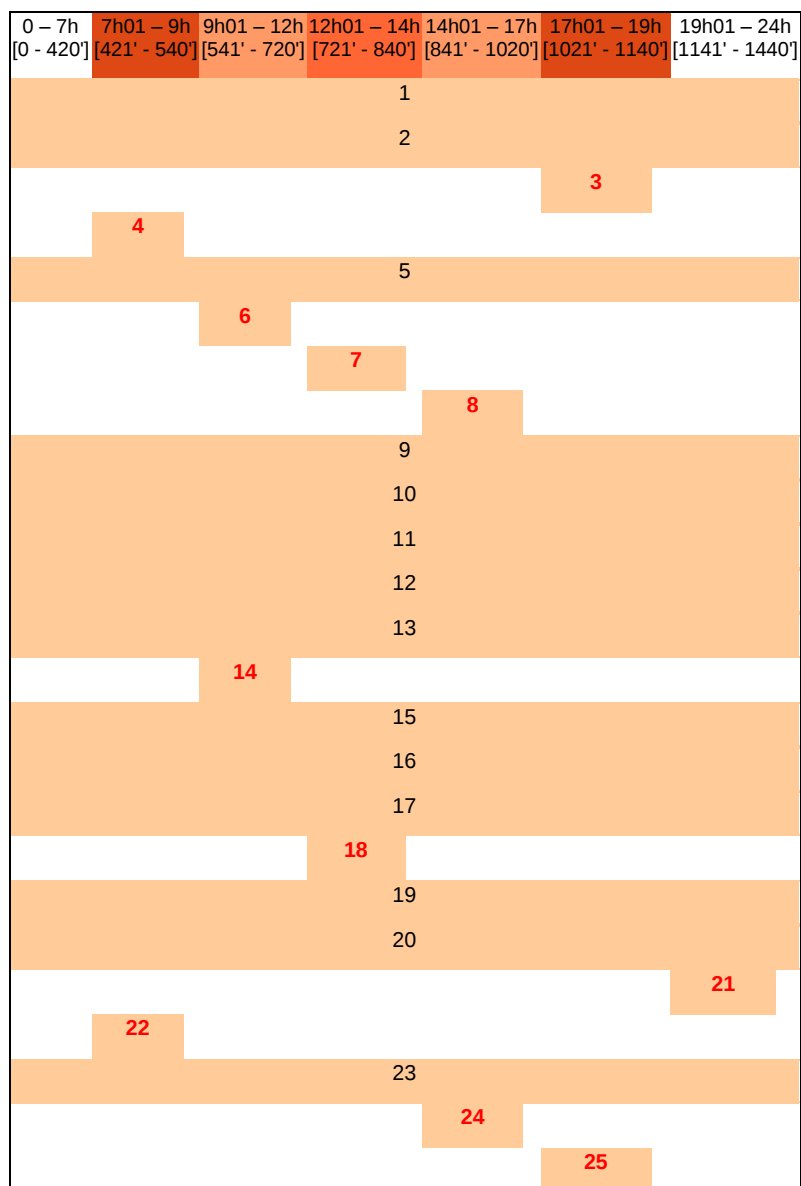


Figura 7 – Distribuição das janelas de tempo dos clientes na nova versão da instância C101 com 25 clientes

O objetivo é evidenciar o início das janelas e identificar em qual janela os clientes foram visitados. Assim, torna-se possível mostrar suas preferências por intervalos do dia onde o tráfego é menos congestionado.

As novas configurações das janelas de tempo são exibidas na figura 8.

Na primeira linha da tabela exibida nessa figura, os intervalos mais congestionados são representados com um fundo em laranja mais escuro e seguem

uma proporção de intensidade de tráfego de acordo com a cor até o branco para os intervalos menos congestionados. As linhas da tabela apresentam cada uma um cliente e as regiões coloridas de fundo apresentam suas janelas de tempo.

Os clientes cujo número encontra-se em vermelho são localizados no início de seus respectivos intervalos com o objetivo de sinalizar o início daquele intervalo. Eles foram escolhidos para essa função, depois de diversos testes e análises, porque estão localizados longe do depósito e presume-se que constarão apenas nas rotas nas quais eles foram visitados efetivamente. Isso evita erros de interpretação.

Uma vez lançadas as otimizações dessa nova instância nos cenários constante e 2, os resultados são exibidos nas figuras 8 e 9.

Cruzando as rotas obtidas e as demandas satisfeitas por cada rota, foi possível identificar em qual intervalo cada cliente foi visitado. As figuras 8 e 9 mostram as rotas obtidas nos diferentes cenários e posicionam os clientes dentro dos intervalos em que foram visitados. Por exemplo, constatamos na figura 8 que sob o cenário constante os clientes 22, 23, 4, 2 e 1 pertencem à segunda rota e foram visitados entre 7h01 e 9h.

	0 – 7h [0 - 420']	7h01 – 9h [421' - 540']	9h01 – 12h [541' - 720']	12h01 – 14h [721' - 840']	14h01 – 17h [841' - 1020']	17h01 – 19h [1021' - 1140']	19h01 – 24h [1141' - 1440']
Rota 1					24,2	25	21
Rota 2		22,23,4,2,1	6	7	8	3	
Rota 3	10,13,17,19,15,12		16	18			

Figura 8 – Resultado da resolução da nova versão da instância C101 com 25 clientes no cenário constante

Na figura 8 concluímos inicialmente no cenário constante que existe uma preferência pelo início das visitas o mais cedo possível. Como todos os intervalos promovem a mesma velocidade durante o dia, os clientes são visitados o mais cedo possível sem esperas, ou seja, no segundo intervalo pois o primeiro provocaria esperas.

No cenário 2 os intervalos mais congestionados são 7h01 às 9h e 17h01 às 19h seguida dos intervalos 12h01 às 14h e por fim 9h01 às 12h e 14h01 às 17h. A primeira e o último intervalos não causam atrasos.

Vemos pela figura 9 que no cenário 2 é possível confirmar que as diferenças de velocidade entre os intervalos influenciam o resultado. De fato, na rota número 1

os clientes 1 e 5 poderiam ter sido visitados nos intervalos vizinhos, mais isso não acontece porque eles provocam velocidades mais lentas. A rota número 2 igualmente mostra que o terceiro intervalo é privilegiado em relação a seus vizinhos.

	0 – 7h [0 - 420']	7h01 – 9h [421' - 540']	9h01 – 12h [541' - 720']	12h01 – 14h [721' - 840']	14h01 – 17h [841' - 1020']	17h01 – 19h [1021' - 1140']	19h01 – 24h [1141' - 1440']
Rota 1		22,20,23,4,9	6,1	7	8,5	3	
Rota 2			14,16,19	18	24	25	21

Figura 9 - Resultado da resolução da nova versão da instância C101 com 25 clientes no cenário 2

4.3 Proposição de pontos de melhoria no algoritmo de otimização

Durante a execução dos testes, foi possível identificar no algoritmo dois pontos passíveis de correção.

4.3.1 Erro de segmentação

Esse problema ocorre pois o algoritmo cria rotas a partir de uma lista de clientes não visitados e os retira do conjunto de pouco em pouco conforme eles são adicionados às rotas. Esse ciclo funciona até o momento em que o último cliente seja incluído em um itinerário e, caso ele não se encaixe em nenhuma rota, uma outra será criada exclusivamente para esse cliente. Porém, se esse cliente não pode de forma alguma ser visitado, esse loop vai ser executado infinitamente e gerar um erro de segmentação. Isso acontece com instâncias nas quais ao menos um cliente possui uma janela que se encerra muito cedo a ponto de que nenhum veículo tenha tempo suficiente para chegar a esse cliente a partir do depósito antes do fim de sua disponibilidade. Seria, então, interessante haver um teste com uma mensagem de erro no código a fim de informar o usuário que esse cliente não pode ser servido por conta de sua janela de tempo muito restrita.

4.3.2 Exibição do resultado mesmo se todos os clientes não foram visitados

Quando um cliente da rede possui uma janela de tempo muito curta, pode acontecer que esse cliente seja ignorado pela resolução do problema e as rotas criadas não lhe incluam. No entanto quando isso acontece o código não se encerra nem apresenta nenhum dispositivo de erro que previna o usuário que certos clientes não foram visitados. É aconselhável, então, que o algoritmo não permita a exibição dos itinerários incompletos ou então que uma mensagem de erro informe o leitor desse acontecimento.

Observou-se que a ocorrência desses dois defeitos com a instância R102. Na instância original, o cliente 14 apresenta uma janela de tempo muito curta com início e fim cedo [32,42]. Em sua otimização obteve-se o erro de segmentação para o cenário constante e, além disso, os cenários 1 e 3 no objetivo número de rotas visitaram apenas 23 dos 25 clientes da rede. Isso explica-se pelo fato que, de acordo o perfil de velocidade e objetivo do problema, o cliente pode ser visitado ou não.

Durante o projeto a equipe se concentrou sobre as instâncias com 25 clientes de três categorias de instâncias de Solomon : C, R e RC. A execução das instâncias com 50 e 100 clientes dependem da performance da máquina empregada. Assim, os resultados das instâncias otimizadas se encontram no anexo 1 a título de ilustração.

CONCLUSÃO

A confecção de rotas é um problema central no planejamento do transporte em logística. O problema de roteirização VRP é sem dúvidas o mais importante entre eles.

Esse projeto trabalhou um tema relativamente pouco abordado nas publicações em pesquisa operacional, os problemas de roteirização com janelas de tempo e dependência de tempo. Nós continuamos o projeto de mestrado de Vincent Huart (2013), que aplicou importantes bases de solução de problemas de caminho mais curto com e sem dependência de tempo. Tendo em vista a complexidade do tema, o VND se mostrou a melhor solução para a resolução de problemas não lineares e a geração de colunas a melhor metaheurística levando em conta o grande número de variáveis. Além disso, a paralelização nos permitiu alcançar diversos objetivos com diferentes formas de se obter novas colunas.

Esse primeiro trabalho tornou nosso trabalho muito enriquecedor graças ao contato com a teoria em torno do VRP e da geração de colunas abordada e também às ferramentas utilizadas. Apesar do fato de as análises não terem se aprofundado como previsto, os resultados obtidos são satisfatórios visto que foi possível provar a eficácia do algoritmo de otimização e igualmente fornecer indícios de melhoria para a sua execução.

No futuro, aconselha-se continuar essa validação sobre instâncias com 50 e 100 clientes usando máquinas com grande capacidade de processamento. Além disso, a exibição do momento da visita de cada cliente dentro da sua rota seria uma evolução no algoritmo interessante a se fazer visto que isso ajudaria bastante a validação do modelo sob novos cenários e forneceria também uma importante informação ao usuário do programa no futuro.

Aplicando-se em Vitória, não seria difícil criar instâncias como as de Solomon que representassem as principais vias de acesso da região, por exemplo ligações entre portos e centros logísticos, e assim desenvolver uma resposta eficiente que pudesse contribuir com o desenvolvimento da logística local.

ANEXOS

Anexo 1 : Resultado de execução das instâncias com 25 clientes sob dois objetivos

Em certos cenários não foi possível alcançar um resultado porque foram obtidos erros de segmentação. Como dito, esses cenários não realizáveis correspondem aos casos em que a velocidade é tão baixa que não é possível respeitar as janelas de tempo de certos clientes, e por conta disso o algoritmo para espontaneamente.

C101

Objectif Cout					Objectif Nb Tournées				
Scenario	Nb tournées	Distance	Durée	Attente	Scenario	Nb tournées	Distance	Durée	Attente
C	5	322,3	2572,3	0	CT	3	349,3	2856,9	257,6
1	5	323,5	2502,1	0	1T	3	538,3	2805,19	203,66
2	5	323,5	2515,7	0	2T	3	501,2	2816,84	207,96
3	3	332,6	2542,72	0	3T	3	547,6	2840,81	155,45
4	5	323,5	2527,97	0	4T	3	456,8	2830,48	195,2

C102

Objectif Cout					Objectif Nb Tournées				
Scenario	Nb tournées	Distance	Durée	Attente	Scenario	Nb tournées	Distance	Durée	Attente
C	4	302,3	2552,3	0	CT	3	328,1	2578,1	0
1	4	303,4	2467,71	0	1T	3	332,9	2493,31	11,9
2	4	303,4	2474,42	0,11	2T	3	328,3	2512,41	9,7
3	4	304,1	2477,45	0	3T	3	347,6	2504,4	0
4	4	303,4	2483,49	0,11	4T	3	336,3	2501,17	9,7

C103

Objectif Cout					Objectif Nb Tournées				
Scenario	Nb tournées	Distance	Durée	Attente	Scenario	Nb tournées	Distance	Durée	Attente
C	3	296,2	2366,2	0	CT	3	296,2	2366,2	0
1	3	289,6	2287,6	0	1T	3	289,6	2287,6	0
2	3	294,6	3341,23	862,9	2T	3	294,6	3341,23	862,9
3	3	369,9	2356,88	14,83	3T	3	369,9	2356,88	14,83
4	3	352,7	2339,75	6,8	4T	3	352,7	2339,75	6,8

Os resultados em vermelho representam as rotas que visitaram apenas 23 clientes.

R101

Objectif Cout					Objectif Nb Tournées				
Scenario	Nb tournées	Distance	Durée	Attente	Scenario	Nb tournées	Distance	Durée	Attente
C	Pas faisable				CT	Pas faisable			
1	9	922	789,77	17,34	1T	7	895,3	844,92	86,98
2	9	922	789,77	17,34	2T	7	937,2	879,47	102,37
3	9	922	789,77	17,34	2T	7	937,2	879,47	102,37
4	11	1141	959,32	61,92	4T	8	1137,9	1015,33	120,47

R102

Objectif Cout					Objectif Nb Tournées				
Scenario	Nb tournées	Distance	Durée	Attente	Scenario	Nb tournées	Distance	Durée	Attente
C	Pas faisable				CT	Pas faisable			
1	7	766,6	696,78	5	1T	5	785,8	702	22,5
2	7	766,6	696,78	5	2T	5	793	732,4	19,26
3	7	766,6	696,78	5	3T	5	777,4	691,97	19,5
4	7	766,6	696,78	5	4T	5	792,8	727,202	16,82

Os resultados em vermelho representam as rotas que visitaram apenas 23 clientes.

R103

Objectif Cout					Objectif Nb Tournées				
Scenario	Nb tournées	Distance	Durée	Attente	Scenario	Nb tournées	Distance	Durée	Attente
C	Pas faisable				CT	6	615,4	828,8	3,4
1	6	622,6	593,18	6,5	1T	4	665,7	617,256	17
2	Pas faisable				2T	4	619,1	563,2	11,39
3	Pas faisable				3T	4	627,9	577,4	15
4	Pas faisable				4T	4	610,4	577,4	21

Os resultados em vermelho representam as rotas que visitaram apenas 21 clientes.

RC101

Objectif Cout					Objectif Nb Tournées				
Scenario	Nb tournées	Distance	Durée	Attente	Scenario	Nb tournées	Distance	Durée	Attente
C	Pas faisable				CT	Pas faisable			
1	4	711,7	674,65	15	1T	4	711,7	674,65	15
2	4	733,9	673,98	3,22	2T	4	733,9	673,98	3,22
3	4	716,5	671,34	11,92	3T	4	716,5	671,34	11,92
4	4	733,9	673,98	3,22	4T	4	733,9	673,98	3,22

RC102

Objectif Cout					Objectif Nb Tournées				
Scenario	Nb tournées	Distance	Durée	Attente	Scenario	Nb tournées	Distance	Durée	Attente
C	Pas faisable				CT	Pas faisable			
1	3	605,3	604,45	9,68	1T	3	605,3	604,45	9,68
2	4	648,4	619,13	41,73	2T	3	589,1	631,393	41,72
3	3	605,3	604,45	9,68	3T	3	605,3	604,45	9,68
4	3	605,3	604,45	9,68	4T	3	605,3	604,45	9,68

Anexo 2 : Perfil de velocidade por cenário

0 – 24h [0 - 1400']	
Tipo 1	1
Tipo 2	1
Tipo 3	1

Cenário constante

0 – 24h [0 - 1400']	
Tipo 1	1
Tipo 2	1,8
Tipo 3	2,5

Cenário 1

	0 – 7h [0 - 420']	7h01 – 9h [421' - 540']	9h01 – 12h [541' - 720']	12h01 – 14h [721' - 840']	14h01 – 17h [841' - 1020']	17h01 – 19h [1021' - 1140']	19h01 – 24h [1141' - 1440']
Tipo 1	1	1	1	1	1	1	1
Tipo 2	1,8	1	1,5	1,3	1,5	1	1,8
Tipo 3	2,5	1,5	2,0	2,2	2,0	1,5	2,5

Cenário 2

	0 – 7h [0 - 420']	7h01 – 9h [421' - 540']	9h01 – 12h [541' - 720']	12h01 – 14h [721' - 840']	14h01 – 17h [841' - 1020']	17h01 – 19h [1021' - 1140']	19h01 – 24h [1141' - 1440']
Tipo 1	1	1	1	1	1	1	1
Tipo 2	1,8	0,5	1,5	0,5	1,5	0,5	1,8
Tipo 3	2,5	1,5	2,0	2,2	2,0	1,5	2,5

Cenário 3

	0 – 7h [0 - 420']	7h01 – 9h [421' - 540']	9h01 – 12h [541' - 720']	12h01 – 14h [721' - 840']	14h01 – 17h [841' - 1020']	17h01 – 19h [1021' - 1140']	19h01 – 24h [1141' - 1440']
Tipo 1	1	1	1	1	1	1	1
Tipo 2	1,8	1	1,5	1,3	1,5	1	1,8
Tipo 3	2,5	0,5	2,0	1	2,0	1,5	2,5

Cenário 4

REFERENCIAS BIBLIOGRAFICAS

BALDACCI R., TOTH P. e VIGO D. **Exact Algorithms for Routing Problems Under Vehicle Capacity Constraints**. Ann. Oper. Res., 175:213–245, 2010.

BODIN, L.D., et al. Routing and scheduling of vehicle and crews: The state of the art. Computers and Operations Research, v.10, p. 63-211, 1983.

BRIMBERG Jack, HANSEN P., MLADENOVIC N. et TAILLARD Eric D. **Improvements and comparison of heuristics for solving the multisource weber problem**. Operations Research, 48 :444–460, 1997.

DANTZIG G. B. e RAMSER R. H. **The truck dispatching problem**. Management Science, 6, 1959.

DANTZIG G. B. e WOLFE P. **Decomposition principle for linear programs**. Operations Research, 8 :101–111, 1960.

DELLING D., NANNICINI G.. **Bidirectional Core-Based Routing in Dynamic Time-Dependent Road Networks**. In Seok-Hee Hong, Hiroshi Nagamochi et Takuro Fukunaga, editores: Algorithms and Computation, volume 5369 de Lecture Notes in Computer Science, paginas 812–823. Springer Berlin / Heidelberg, 2008.

FLEISCHMANN B., GIETZ M. e GNUTZMANN S. **Time-varying travel times in vehicle routing**. Transportation Science, 38(2) :160–173, May 2004.

FIGLIOZZI Miguel Andres. **The time dependent vehicle routing problem with time windows : Benchmark problems, an efficient solution algorithm, and solution characteristics**. Transportation Research Part E : Logistics and Transportation Review, 48(3) :616–636, May 2012.

GOLDEN B., RAGHAVAN S., WASIL E.. **Vehicle Routing Problem: Latest Advances and New Challenges**, volume 43 em Operations Research Computer Science Interfaces. Springer, New York, United States, 2008.

HANSEN P., MLADENOVIC N., MORENO J. P. **Variable Neighbourhood Search: Methods and Applications**. 4OR: A Quarterly Journal of Operations Research, 6 :319–360, 2008.

HILL A. V. e BENTON. **Modeling intra-city time-dependent travel speeds for vehicle scheduling problems**. Journal of Operational Research Society, 43 :343–351, 1992.

HUART Vincent. **Résolution de tournées de véhicules dépendantes du temps avec fenêtre de temps (TDVRPTW) par génération de colonnes**. Dissertação de mestrado. Montréal: HEC, 2013.

ICHOUA J. P. S., GENDREAU M. **Vehicle dispatching with timedependent**

travel times. European Journal of Operational Research, 144 :379–396, 2003.

KOHL N., DESROSIERS J., MADSEN O. B. G., SOLOMON M. M. e SOUMIS F. **2-path cuts for the vehicle routing problem with time windows**. *Transportation Science*, 33(1) :101–116, 1999.

MALANDRAKI, C., 1989. **Time Dependent Vehicle Routing Problems: Formulations, Solution Algorithms and Computational Experiments**. Ph.D. Dissertation, Northwestern University, Evanston, Illinois.

MALANDRAKI, C., DASKIN, M.S., 1992. **Time-dependent vehicle-routing problems – formulations, properties and heuristic algorithms**. *Transportation Science* 26 (3), 185–200.

Malandraki, C., Dial, R.B.. **A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem**. *European Journal of Operational Research* 90 (1), 45–55. 1996.

NANNICINI G., DELLING D., LIBERTI L. e SCHULTES D. **Bidirectional A* Search for Time-Dependent Fast Paths**. In Catherine McGeoch, éditeur : *Experimental Algorithms*, volume 5038 de *Lecture Notes in Computer Science*, pages 334–346. Springer Berlin / Heidelberg, 2008.

PESSOA A., ARAGÃO M.P. de e UCHOA E. **Robust Branch-Cut-and-Price Algorithms for Vehicle Routing Problems**. In B. Golden, S. Raghavan et E. Wasil, éditeurs : *The Vehicle Routing Problem : Latest Advances and New Challenges*, volume 43 de *Operations Research/Computer Science Interfaces Series*, pages 297–325. Springer US, 2008.

SOLER E. M. D., ALBIACH J.. **A way to optimally solve a timedependent vehicle routing problem with time windows**. *OperationsResearch Letters*, 37 :37–42, 2009.

SOLOMON, M.M. **Algorithms for vehicle routing and scheduling problems with time window constraints**. *Operations Research*, V.35, n.2, p.254-266, 1987.

REINA, C.D. Roteirização de veículos com janelas de tempo utilizando algoritmo genético. Dissertação de mestrado ; USP, 2012.

TOTH P. E VIGO D. **The Vehicle Routing Problem. Monographs on Discrete Mathematics and Applications**. Society for Industrial and Applied Mathematics, 2002.