

Uma Heurística Composta para a Determinação de Rotas para Veículos em Problemas com Janelas Temporais e Entregas e Recolhas

Ana Moura * ‡

José F. Oliveira † ‡

* ESTiG - Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Bragança
anamoura@fe.up.pt

† FEUP - Faculdade de Engenharia da Universidade do Porto
jfo@fe.up.pt

‡ INESC Porto – Instituto de Engenharia de Sistemas e Computadores do Porto

Abstract

In this paper a new heuristic for the vehicle routing problem is presented. This algorithm was applied to a problem originated from a Portuguese alimentary products distribution company. This company has many clients and more than 130 deliveries per day for all products. When considering these figures, the vehicle routing problem becomes too much complex to be manually solved. The necessity of automatization naturally arises.

This new multi-phase heuristic has a constructive phase, a local optimisation phase and a pos-optimisation phase, and aims the minimization of the sum of the routes total time. Additional constraints to the vehicle routing problem, driven by the particular company that motivated this work, are considered. In particular time-windows both for the drivers and for the clients and pick-up together with deliveries are considered.

Resumo

Neste artigo apresenta-se uma nova heurística para a determinação de rotas para veículos, aplicada a uma empresa portuguesa de distribuição de produtos alimentares. Esta empresa é detentora de uma grande carteira de clientes, realizando perto de cento e trinta entregas diárias de vários tipos de produtos. A necessidade da automatização do processo de determinação das rotas dos veículos surge naturalmente neste contexto.

Esta nova heurística composta tem uma fase construtiva, uma fase de optimização local e uma fase de pós-optimização, sendo o objectivo a minimização do tempo total dos percursos. São ainda incorporadas restrições adicionais derivadas da aplicação concreta que motivou o presente trabalho. Em particular são consideradas janelas temporais, quer para os condutores/veículos quer para os clientes, e recolhas de vasilhame em simultâneo com as entregas de mercadorias.

Keywords: Vehicle routing problem, Time-windows, Pick-up and delivery, Multiple travelling salesman problem, heuristics

Title: A multi-phase heuristic for the delivery pick-up vehicle-routing problem with time-windows

1 Introdução

O problema de distribuição apresentado pela empresa, e que esteve na base do desenvolvimento deste algoritmo, pode ser formulado como um problema de planeamento de rotas para veículos com janelas temporais (VRPTW). Este problema é uma generalização do problema de rotas para veículo (VRP) e consiste na determinação de n rotas para n veículos, onde uma rota é definida como um percurso que tem início num armazém, visita por uma determinada ordem um sub-conjunto de clientes, cada um dentro de um intervalo de tempo específico (janelas temporais), retornando finalmente ao mesmo armazém.

Existem ainda várias restrições adicionais, tanto da parte dos clientes como da parte da empresa e dos veículos, que tornam este problema mais complexo. Existem diversos artigos publicados nesta área, e variadas formas de obter soluções muito perto do óptimo, no entanto poucos consideram simultaneamente o leque alargado de restrições com que neste problema específico se lida. Referências relevantes são os trabalhos de Laporte, onde se apresenta algoritmos exactos e aproximados para o VRP [6], e onde se faz um resumo das várias heurísticas para o VRP, divididas em heurísticas clássicas e heurísticas modernas [7].

Uma abordagem clássica à resolução do VRP passa pela sua modelização como um problema de caixeiro viajante múltiplo (m-TSP), onde cada um dos veículos é um caixeiro viajante que tem de visitar um determinado número de clientes com diferentes localizações geográficas, diferentes pedidos de entrega e recolha e ainda diferentes janelas temporais. Na literatura encontram-se vários métodos de abordagem para estes problemas. Por exemplo, Laporte [8] apresenta alguns dos algoritmos exactos e aproximados até então desenvolvidos para aplicação ao TSP, considerando TSP simétricos e que satisfazem a desigualdade triangular. Outros autores ([1],[9] e [10]) apresentam heurísticas de melhoramentos aplicadas a uma solução inicial admissível. Estas heurísticas baseadas nos princípios do r-OPT e Or-OPT (procedimentos de melhoria aplicados a soluções já existentes) melhoram a solução inicial através de movimentos de trocas ou inserções em rotas ou entre rotas. Helsgaun [4] descreve uma implementação da heurística Lin-Kernighan, defendendo que é um dos métodos com mais sucesso para o TSP simétrico.

Quando se consideram entregas e recolhas, Gendreau et al [3] descreve uma nova heurística para o problema do caixeiro viajante com entregas e recolhas (TSPPD), baseada em pesquisa tabu. No domínio das heurísticas de construção de percursos, Johnson e McGeoch [5] obtiveram resultados teóricos bastante interessantes. Relativamente às heurísticas de pesquisa local 2-Opt e 3-Opt, os mesmos autores analisam o seu comportamento, apresentando uma descrição e resultados computacionais.

A consideração conjunta de janelas temporais e entregas e recolhas é feita em [2]. Contudo, neste caso, e ao contrário do problema abordado neste artigo, as entregas são feitas numa primeira metade do percurso e as recolhas na segunda.

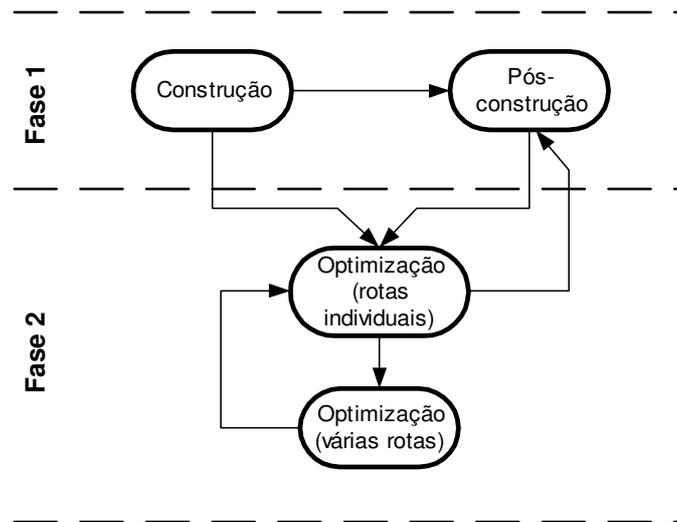


Figura 1: Fases da Heurística Composta

No que diz respeito à organização do presente artigo, na secção 2 é descrito o problema de uma forma sucinta, passando-se de seguida à sua modelização e formulação matemática (secção 3). Posteriormente é descrita a heurística composta desenvolvida (secção 4). Esta heurística está dividida em duas fases (Figura 1). A primeira fase corresponde à construção das rotas e divide-se em duas subfases, uma construtiva propriamente dita e outra de pós-construção. A segunda fase corresponde a uma heurística de melhoramento, que por sua vez também se encontra dividida em duas subfases, uma de optimização local, executada em cada uma das rotas individualmente, e outra com trocas entre as várias rotas, incluindo uma componente aleatória.

Por fim (secção 5 e 6), são apresentados resultados computacionais obtidos pela heurística composta quando aplicada a vários exemplos e retiradas algumas conclusões.

2 Descrição do Problema

A empresa faz a distribuição de produtos alimentares, tendo a recepção e expedição de mercadorias centralizada em duas plataformas, uma no Porto e outra em Lisboa. A sua distribuição é efectuada por veículos subcontratados, de acordo com as suas necessidades diárias e de forma a satisfazer as encomendas.

O planeamento das rotas dos veículos está obrigatoriamente dependente das encomendas existentes, que naturalmente condicionam o número de veículos necessários para a distribuição diária e os percursos correspondentes. Estes veículos são escolhidos de acordo com quatro factores: os tipos de produtos a entregar, as localizações dos clientes relativamente aos tipos de acessos, as capacidades máximas e as dimensões físicas de cada veículo.

Existem três tipos de veículos para o transporte que variam em tamanho e características. O mais pequeno tem uma capacidade de 10 Toneladas (transporta os produtos em contentores de pequena dimensão), o seguinte de 15 Toneladas (com sistema de frio e que transporta os produtos em paletes) e o de maior dimensão de 19 Toneladas (onde a carga de uma maneira

geral é transportada a granel). No que diz respeito aos tipos de produtos distribuídos pela empresa em questão, podemos dizer que estão divididos em cinco categorias: congelados, frescos, mercearias (não alimentares e alimentares), bebidas e batatas. O tipo de produto restringe à partida os tipos de veículos que podem ser utilizados para um dado transporte. Conjugando o tipo de produtos a entregar com a localização geográfica do cliente, que define a sua acessibilidade por cada tipo de veículo, fica univocamente definido o tipo de veículo a usar para fazer as entregas a esse cliente, uma vez que é política da empresa utilizar sempre para cada cliente o tipo de veículo admissível com maior capacidade (i.e. sendo possível usar um veículo de 15 Toneladas ou um de 19 Toneladas, opta-se pelo veículo de 19 Toneladas). Desta forma os clientes ficam divididos em escalões, correspondendo cada escalão a um tipo de veículo.

Uma vez que às diferentes áreas geográficas correspondem diferentes velocidades médias dos veículos, torna-se necessário trabalhar com tempos de deslocação em vez de distâncias. Consequentemente, a cada cliente é atribuída uma velocidade de acesso que corresponde à velocidade média com que os veículos se deslocam na respectiva área geográfica. Esta característica faz com que neste problema não se verifique a desigualdade triangular.

Restrições existentes para a definição das rotas:

Existem várias restrições que têm que ser consideradas na elaboração das rotas:

1. Capacidade da frota, tendo em consideração que existem entregas de encomendas e recolhas de vasilhames nos clientes;
2. Dimensão e tonelagem dos veículos, que condicionam a acessibilidade dos mesmos;
3. Janelas temporais relativas aos horários dos motoristas, sendo estas janelas duplas por efeito da hora para almoço;
4. Janelas temporais relativas aos clientes, para a recepção das encomendas;
5. Velocidades médias diferentes, para os veículos, conforme a zona geográfica.

3 Modelização e Formulação Matemática do Problema

Conforme foi descrito anteriormente, o problema que se pretende modelizar considerará os vários pedidos de entrega de um determinado tipo de produto, para vários clientes de diferentes (ou da mesma) área geográfica. Assim ter-se-á uma lista de clientes para serem visitados uma e uma só vez, cada um por um único veículo. Garantindo que todos os pedidos desses clientes não excedem a capacidade de um determinado veículo, deve-se escolher uma sequência para visitar esses clientes, fazendo as entregas e recolhas necessárias, retornando ao ponto de origem (armazém) e percorrendo o caminho total no menor tempo possível. São também conhecidas as distâncias geométricas entre o armazém e cada um dos clientes e entre os vários clientes, sabendo-se assim todos os tempos de percurso entre o armazém e clientes e entre cada par de clientes.

O problema é formulado sobre um grafo $G(N, T)$, sendo N o conjunto dos clientes mais o armazém e T as ligações entre os vários clientes e entre estes e o armazém. Estas arestas são caracterizadas pelos tempos de deslocação.

Será considerado um conjunto de veículos diferentes $V = \{1, 2, \dots, v\}$, um conjunto de clientes de um dado escalão $C = \{1, 2, \dots, c\}$ e o armazém A . O armazém é representado por dois nós $A = \{0, n + 1\}$. Nenhum arco termina no nó 0 e nenhum arco tem origem no nó $n + 1$. Todos os percursos terão início no nó 0 e terminam no nó $n + 1$. O conjunto de todos os nós $N = \{0, 1, 2, \dots, c, n + 1\}$, corresponde à união dos conjuntos clientes C e armazém A . A cada arco do grafo (i, j) , $i \neq j$, é associada uma distância d_{ij} , que corresponde à distância entre clientes, e um tempo de percurso t_{ijv} , que corresponde ao tempo que um determinado veículo v leva a deslocar-se entre os clientes e entre clientes e armazém. Existe ainda um tempo de serviço ts_{iv} que corresponde ao tempo que o veículo v demora a fazer a entrega no cliente i , e um tempo de deslocação td_{ijv} que corresponde à soma do tempo de percurso entre dois clientes (ou entre o último cliente da rota e o armazém) com o tempo de serviço no cliente i , tudo relativamente ao veículo v . O tempo de deslocação e o tempo de percurso são obtidos através das seguintes expressões seguintes:

- Entre vários clientes, clientes e armazém e armazém e clientes, o tempo de deslocação, é dado por:

$$td_{ijv} = ts_{iv} + t_{ijv} \text{ [minutos]} \quad i, j \in N, v \in V$$

- O tempo de percurso t_{ijv} , é calculado pela seguinte expressão:

$$t_{ijv} = \frac{d_{ij} * 60}{VelMed_v} \text{ [minutos]} \quad i, j \in N, v \in V$$

onde $VelMed$ é a velocidade de deslocação do veículo em km/h, para a área geográfica dos clientes. Nos casos em que os clientes i e j se encontram em áreas geográficas diferentes, a velocidade de deslocação do veículo entre eles é a média das velocidades de deslocação nas duas zonas geográficas. No caso de $i = 0$ (partindo do armazém) toma-se $ts_{0v} = 0$.

Cada veículo tem uma determinada capacidade q_v (com $v \in V$), que não pode ser excedida, e a cada cliente corresponde um determinado pedido p_i (com $i \in C$). Cada veículo tem ainda associada uma janela temporal $[0, k_v]$, $v \in V$, sendo K_v as horas de trabalho dos motoristas.

Para cada cliente, o início da entrega deve estar dentro de uma janela temporal, $[a_i, b_i]$, $i \in C$. Um veículo pode chegar a um cliente antes do início da sua janela temporal e esperar até que seja possível efectuar as entregas, mas não pode chegar depois do fim da janela temporal.

Como estamos perante um caso de entregas e recolhas, é necessário ter em atenção quando um determinado cliente tem vasilhame para ser recolhido. Assim, r_i com $i \in C$ é o vasilhame a ser recolhido num cliente.

Assume-se que todos os dados, $q_v, r_i, p_i, d_{ij}, t_{ijv}, ts_{iv}, a_i, b_i$, têm valores positivos. Finalmente assume-se também que este modelo não satisfaz a desigualdade triangular, i.e.,

$$\exists_{h,i,j \in N} \exists_{v \in V} : d_{ij} > d_{ih} + d_{hj} \vee t_{ijv} > t_{ihv} + t_{hvj}.$$

Como variáveis de decisão, ter-se-á:

1. As variáveis x_{ijv} , definidas $\forall i, j \in N, \forall v \in V$ com $i \neq j, i \neq n + 1, j \neq 0$, que tomam o valor 1 se o veículo v se deslocar do nó i para o nó j , e o valor 0 se não se deslocar.

2. As variáveis s_{iv} , definidas $\forall i \in N, \forall v \in V$, que nos indicam o instante de tempo em que um veículo $v, v \in V$, inicia o serviço no cliente $i, i \in C$. Assume-se que $s_{0v} = 0, \forall v \in V$, e $s_{n+1,v}$ que nos indica o tempo de chegada do veículo v ao armazém.

Para um determinado número de veículos, o objectivo é definir rotas cujo tempo total de percurso seja mínimo. Matematicamente podemos modelizar este problema da seguinte forma:

$$\min \sum_{v \in V} s_{n+1,v} \quad (1)$$

$$\sum_{v \in V} \sum_{j \in N} x_{ijv} = 1 \quad \forall i \in C \quad (2)$$

$$\sum_{i,j \in S} x_{ijv} \leq |S| - 1 \quad S \subset C, 2 \leq |S| \leq n + 2 \quad \forall v \in V \quad (3)$$

$$\sum_{i \in C} \left[\max(p_i, r_i) \times \sum_{j \in N} x_{ijv} \right] \leq q_v \quad \forall v \in V \quad (4)$$

$$\sum_{j \in N} x_{0jv} = 1 \quad \forall v \in V \quad (5)$$

$$\sum_{i \in N} x_{ihv} - \sum_{j \in N} x_{h j v} = 0 \quad \forall i \in C, \forall v \in V \quad (6)$$

$$\sum_{i \in N} x_{i,n+1,v} = 1 \quad \forall v \in V \quad (7)$$

$$s_{iv} + td_{ijv} - M(1 - x_{ijv}) \leq s_{jv}, \forall i \in N, \forall j \in N, \forall v \in V, M \rightarrow +\infty \quad (8)$$

$$a_i \leq s_{iv} \leq b_i \quad \forall i \in N, \forall v \in V \quad (9)$$

$$s_{n+1,v} \leq k_v \quad \forall v \in V \quad (10)$$

$$x_{ijv} \in \{0, 1\} \quad \forall i \in N, \forall j \in N, \forall v \in V \quad (11)$$

$$s_{iv} \geq 0 \quad \forall i \in N, \forall v \in V \quad (12)$$

A restrição (2) garante que a cada cliente é atribuído um e um só veículo e que de cada cliente só se sai para um outro cliente ou para o armazém. A restrição (3), impede a formação de ciclos. Note-se que a separação do armazém num nó origem 0 e num nó destino $n+1$, leva a que não sejam admissíveis ciclos de qualquer comprimento. A restrição (4) é relativa à capacidade do veículo, garantindo que nenhum veículo visita mais clientes do que os permitidos pela sua capacidade. Esta restrição não só tem em consideração os pedidos dos clientes, como também considera os vasilhames existentes nos clientes e que é necessário recolher. As restrições (5, 6 e 7), são restrições de fluxo que garantem que cada veículo parte do nó 0, só sai de um nó h se lá tiver entrado previamente e que termina o percurso no nó $n+1$. A restrição (8) garante-nos que se a ligação de i para j for escolhida para o veículo v ($x_{ijv} = 1$) então não se inicia o serviço em j antes de lá se chegar. Se $x_{ijv} = 0$ a restrição fica não activa. A restrição (9) garante-nos que todas as janelas temporais são respeitadas. A restrição (10) impõe uma duração máxima para a rota de cada veículo, k_v , motivada pelo horário de trabalho dos motoristas.

Este modelo tem como base o modelo do problema do caixeiro viajante (TSP), embora com várias restrições adicionais. Por outro lado, a consideração de vários veículos em simultâneo

leva à consideração de m caixeiros (correspondendo a cada um dos veículos), que se localizam numa determinada cidade (o Armazém) – esta assunção é importante por causa das janelas temporais – e têm de visitar um determinado conjunto de cidades C (que corresponde aos clientes). Partindo do armazém, o objectivo é visitar todas as cidades e retornar à cidade de origem, percorrendo o menor caminho possível, aqui medido em termos de tempos de percurso.

4 Heurística Composta para a Determinação de Rotas para Veículos

Foi desenvolvida uma heurística composta com duas fases. A primeira fase consiste numa heurística construtiva. Este algoritmo baseia-se no algoritmo do vizinho mais próximo, sendo a noção de proximidade dada pela ordenação dos clientes na lista de candidatos. Vários critérios de ordenação foram implementados e testados. Por sua vez este algoritmo construtivo está dividido em duas subfases. A primeira subfase (Tabela 2) constrói rotas completas para cada escalão, considerando todas as restrições adicionais mencionadas na secção 2. Para cada um dos escalões, o início da determinação de uma rota toma em consideração as ligações do armazém a todos os clientes do escalão. Encontrando o primeiro cliente que satisfaça todas as restrições insere-o em primeiro lugar no conjunto C^* (conjunto correspondente a uma rota) e retira-o do conjunto C . Em seguida, para os clientes seguintes candidatos à rota, considera sempre as ligações do último cliente (L_n) inserido em C^* com todos os outros do conjunto C . Estas listas de ligações correspondem aos tempos de ligação entre cada um dos clientes com todos os outros, incluindo o armazém.

Sempre que se insere um cliente na rota são verificadas as restrições de capacidade e das janelas temporais, além de se verificar também as condições de paragem do algoritmo, nomeadamente: o tempo máximo de trabalho dos motoristas e $C = \{\}$, i.e., já não existirem mais clientes no conjunto C .

Para esta fase construtiva foram considerados quatro métodos distintos de ordenação dos dados, i.e., ordenação dos clientes nas listas de ligação:

1. Tempos de deslocação;
2. Tempos de percurso;
3. Tamanho das janelas temporais.
4. Tamanho das janelas temporais agrupadas por intervalos de tempo.

Para se validarem as restrições de capacidade dos veículos e as restrições das janelas temporais dos clientes, foram também desenvolvidas duas rotinas específicas que são invocadas pelo algoritmo construtivo sempre que um cliente é inserido na rota. A sua função é, respectivamente:

1. Verificar se a capacidade máxima dos veículos foi atingida ou ultrapassada.
2. Verificar se foram violadas: as janelas temporais dos clientes, a hora de saída dos veículos do armazém, o tempo máximo que um veículo pode esperar em cada um dos clientes (30 minutos) antes do início da sua janela temporal, a hora de almoço dos motoristas.

```

// Algoritmo construtivo
Início:
Para cada escalão fazer
    Enquanto (Elementos escalão != 0)
        Elimina Elementos da Melhor Rota;
        Condição verdadeira = Calcula Rota;
        Se(Melhor Rota > 0)
            Guarda Melhor Rota;
            Retira Elementos Lista Escalão;
        Senão
            Guarda Elementos não Inseridos;
        Fim Se
    Fim Enquanto
Fim Para
Fim
//Calcula Rota
Início:
Se (rota Vazia)
    Para cada ligação do armazém fazer
        Se (Cliente pertence ao escalão)
            Acrescenta Cliente à Rota;
            Condição verdadeira = Calcula Rota;
            Se (condição verdadeira)
                Retorna (verdadeiro);
            Senão
                Retira Cliente da Rota;
            Fim Se
        Fim Se
    Fim Para
Senão
    Se (validaCapacidade &&
        validaJanelasTemporais)
        Retorna (falso);
    Fim Se
    Se (Rota é melhor que a melhor guardada)
        Guarda Rota como a melhor;
    Fim Se
    Se (tamRota==numClientes ou
        Tempo trabalho motorista>12horas)
        Retorna (verdadeiro);
    Fim Se
    Para cada ligação do último Cliente na Rota fazer
        Se (Cliente pertence ao escalão e Cliente não
            pertence à Rota)
            Acrescenta Cliente à rota;
            Condição verdadeira = CalculaRota;
            Se (verdadeiro)
                Retorna (verdadeiro);
            Senão
                Retira Cliente da Rota;
            Fim Se
        Fim Se
    Fim Para
Fim Se
Retorna (falso);
Fim

```

Figura 2: Primeira subfase do algoritmo construtivo.

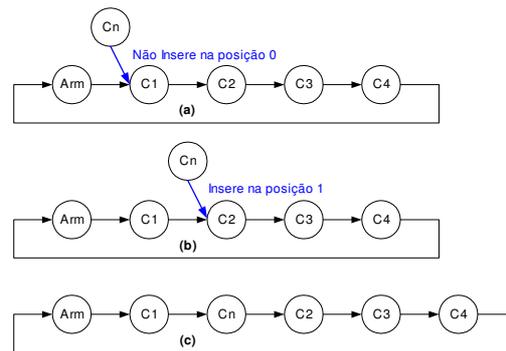


Figura 3: Inserção de um elemento.

Uma vez que nesta primeira subfase, não é garantida a inserção de todos os clientes nas rotas, foi necessário considerar uma segunda subfase na heurística construtiva. Assim foi criado um algoritmo pós-construtivo (Figura 3) que, no caso de existirem ainda clientes não inseridos, tenta um a um inseri-los em todas as posições de todas as rotas, testando sempre todas as restrições associadas à rota respectiva.

Depois de construídas as rotas passa-se para a segunda fase da heurística composta. Esta é uma fase de melhoramentos das rotas obtidas, i.e. diminuição dos tempos totais de percurso. Assim, com base na heurística 2-OPT para optimização local, foram desenvolvidos dois algoritmos que executam trocas de nós vizinhos em cada uma das rotas previamente construídas.

Para cada uma das trocas efectuadas é necessário preceder à verificação das seguintes restrições: capacidade dos veículos, janelas temporais dos clientes e horário de trabalho dos motoristas. Esta optimização local é também executada em duas subfases, i.e. em primeiro lugar é aplicada a cada uma das rotas individualmente (Figura 4) e só depois efectuando trocas de clientes entre as várias rotas, agora incorporando uma componente aleatória.

Sempre que no fim da primeira subfase se consegue efectuar alguma troca entre clientes, é de imediato aplicado o algoritmo pós-construtivo anteriormente mencionado. Caso contrário avança-se para a segunda subfase do algoritmo de melhoramentos.

Como se pode verificar no pseudocódigo da Figura 4, em cada uma das rotas selecciona-se o primeiro cliente e tenta-se trocá-lo com todos os outros (Tabela 4). Sempre que se tenta uma troca, é necessário verificar se as restrições não são violadas. Caso se verifiquem todas as restrições e o novo tempo total da rota obtido seja inferior ao tempo total da rota inicial, então a troca é considerada. Caso contrário, i.e., se alguma das restrições não for validada ou se o tempo novo for superior ou igual ao tempo inicial, então o cliente volta à sua posição inicial passando-se ao cliente seguinte.

Novamente, sempre que uma troca é aceite e se ainda existirem clientes não inseridos do mesmo escalão da rota que está a ser melhorada, é chamado o algoritmo pós-construtivo que tenta inserir mais algum cliente. De facto, tendo a ordem de visita aos clientes sido alterada poderá ser possível inserir mais um cliente na rota.

Neste segundo passo do algoritmo de melhoramentos são efectuadas trocas de vários clientes de uma rota com os restantes clientes de rotas diferentes, embora do mesmo escalão. O número de clientes de cada rota, que vão sofrer alterações nas suas posições, é aleatório. Contudo a

```

// Algoritmo de melhoramento individual
Início:
Para cada escalão fazer
  Para cada rota fazer
    Fazer
      Condição verdadeira =Optimiza Uma Rota;
      Enquanto (Verdadeiro)
        Retira Listas Vazias de Elementos NãoInseridos;
    Fim Para
  Fim Para
Fim Para
// Optimiza Uma Rota
Início:
Para cada elemento da rota fazer
  Inserir Elementos Não Inseridos;
Fim Para
Para cada elemento da rota fazer
  Cliente i;
  Cliente sucessor de i;
  Troca i com sucessor de i;
  Se(validaCapacidade &&validaJanelasTemporais)
    Se (tempoPercursoTotalNovo<tempoPercursoTotal)
      retorna (verdadeiro);
  Fim Se
Fim Se
Troca sucessor de i com i;
Fim Para
retorna (falso);
Fim

```

Figura 4: Algoritmo de otimização local para rotas individuais.

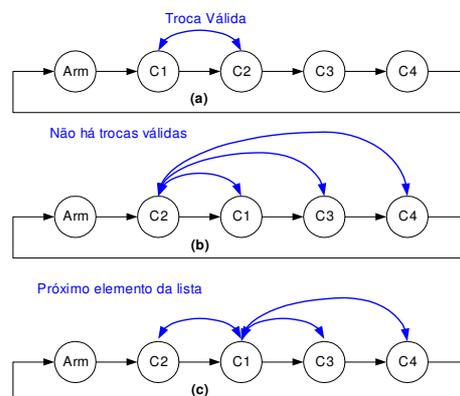


Figura 5: Troca entre clientes.

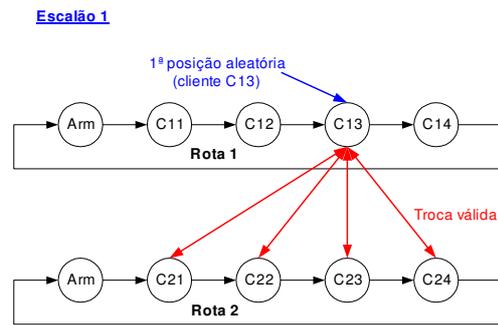


Figura 6: Troca do primeiro cliente.

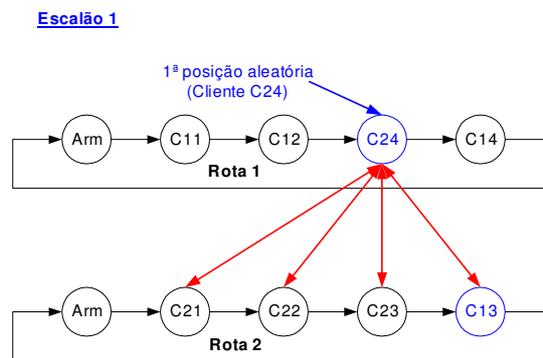


Figura 7: Troca aceite.

distribuição de probabilidade não é uniforme: há uma probabilidade de 0.45 de ser trocado um número de clientes que se situa entre $\frac{1}{4}$ e $\frac{1}{2}$ do número total de clientes da rota, enquanto a probabilidade de serem trocados menos de $\frac{1}{4}$ dos clientes ou mais de metade é de 0.55. Depois de determinado o número de clientes a ser trocados por rota, define-se também, de uma forma aleatória e dentro da própria rota, quais os clientes que vão estar sujeitos às trocas.

Para cada um desses clientes, escolhido ao acaso, são efectuadas trocas com todos os outros clientes das restantes rotas do mesmo escalão (Figura 5). Para cada uma das trocas efectuadas são sempre validadas as restrições, para que a troca seja considerada admissível. A troca será aceite se, para além de admissível, verificar ainda a seguinte condição: *a soma dos tempos totais das duas rotas em questão, depois da troca efectuada, deve ser inferior à soma dos tempos totais das duas mesmas rotas antes da troca.*

Se esta condição for verificada então a troca é efectuada (Figura 6). Caso contrário, os clientes são mantidos nas suas posições iniciais nas respectivas rotas e a execução do algoritmo continua até que não existam mais posições nem rotas para executar trocas.

De acordo com o exemplo das Figura 5 e 7, verifica-se que sendo uma troca aceite o algoritmo reinicia-se tentando-se trocar esse cliente (no exemplo o cliente C24 que pertence agora à Rota 1), com todos os outros clientes das outras rotas do escalão, tentando desta forma obter uma rota melhor que a anterior. Quanto o cliente seleccionado já não puder ser efectivamente trocado, esta rotina termina com a melhor rota até então encontrada e passa ao passo seguinte. No passo seguinte executa-se o algoritmo de optimização (Figura 4) para uma rota, tentando desta forma optimizar as duas rotas alteradas. Além disso tenta também inserir

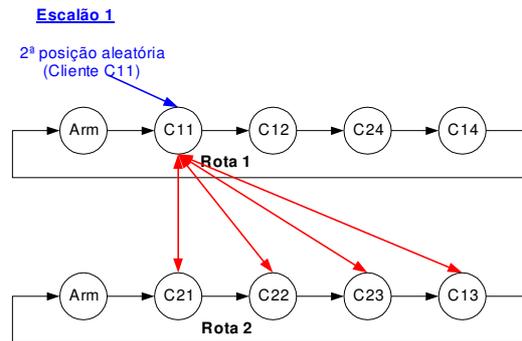


Figura 8: Novo cliente.

quaisquer elementos não inseridos que eventualmente ainda existam (Figura 8). Seguidamente escolhe aleatoriamente um outro cliente e retoma todo o processo (Figura 7) até que já não existam mais clientes em nenhuma rota para efectuar trocas.

Verifica-se que no final da execução deste algoritmo se obtêm sempre reduções significativas dos tempos totais dos percursos.

Se no final da execução global do algoritmo ainda existir algum cliente não inserido, duas medidas poderão ser tomadas. A primeira é a re-execução do algoritmo, que tem tempos de execução muito pequenos, e tentar desta forma obter um resultado em que todos os clientes sejam inseridos, uma vez que a segunda subfase da segunda fase do algoritmo tem uma componente aleatório na selecção dos clientes a serem trocados entre as várias rotas. Alternativamente, poderá ser criada uma nova rota (com um novo veículo) para esse ou esses clientes.

5 Resultados obtidos

Para cada um dos métodos de ordenação dos dados apresentados neste artigo, foram efectuados testes utilizando instâncias com 50, 100 e 150 clientes. Mesmo quando o número de clientes é igual, todas as instâncias diferem no que diz respeito às localizações dos clientes, às encomendas, aos tipos de acesso, etc. Estas diferenças são fundamentais para retirar conclusões acerca dos resultados obtidos pelos quatro métodos de ordenação quando aplicados na heurística composta desenvolvida. Para cada um destes exemplos e para cada um dos métodos de ordenação dos dados a heurística foi executada 20 vezes, obtendo-se assim resultados diferentes para o total dos tempos de percurso. Os valores apresentados correspondem às médias obtidas para cada uma das execuções da heurística composta e para cada um dos exemplos.

Os resultados obtidos são analisados segundo quatro vectores diferentes:

1. Tempo total dos percursos

Os melhores tempos obtidos (Tabela 9) são relativos ao método de ordenação de dados pelo tamanho das janelas temporais agrupadas em intervalos de tempo e o pior dos casos para o método de ordenação pelos tempos de deslocação.

2. Número de clientes não inseridos nas rotas

```

// Algoritmo de Pós-Optimização
Início:
Para cada escalão fazer
  Para cada rota fazer
    Gera Número de Clientes Aleatórios;
    Para cada número de clientes aleatórios fazer
      Gera Um Cliente Aleatório;
      Para cada rota fazer
        Se rota é diferente da rota do cliente aleatório
          Fazer
            Condição verdadeira = Optimização
              entre Rotas;
            Enquanto (verdadeiro)
              Fazer
                Condição verdadeira = Optimização Uma
                  Rota (do Cliente Aleatório);
            Enquanto (verdadeiro)
              Retira Listas Vazias de Elementos Não
                Inseridos;
          Fazer
            Condição verdadeira = Optimização
              Uma Rota (outra Rota);
            Enquanto (verdadeiro)
              Retira Listas Vazias de Elementos Não
                Inseridos;
        Fim Se
      Fim Para
    Fim Para
  Fim Para
Fim Para
Fim
// Optimização entre Rotas
Início:
TempoTrabalho=TtrabalhoRota(do cliente aleatório)
  +TtrabalhoRota;
Para cada elemento da rota fazer
  Troca Cliente Aleatório com Cliente da rota;
  TempoTrabalhoNovo=TtrabalhoNovo da rota do
    cliente aleatório+TtrabalhoNovo da rota;
  Se (validaCapacidade & ÉvalidaJanelasTemporais)
    Se (TempoTrabalhoNovo< TempoTrabalho)
      retorna (verdadeiro);
  Fim Se
  Fim Se
Troca Cliente da rota com Cliente Aleatório;
Fim Para
retorna (falso);
Fim

```

Figura 9: Algoritmo para troca de clientes aleatórios entre rotas.

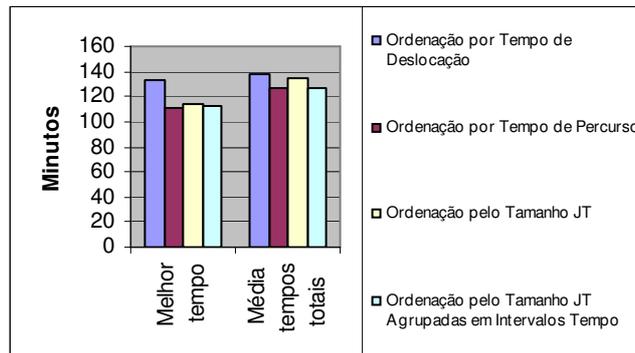


Figura 10: Tempos médios e Melhor dos tempos totais.

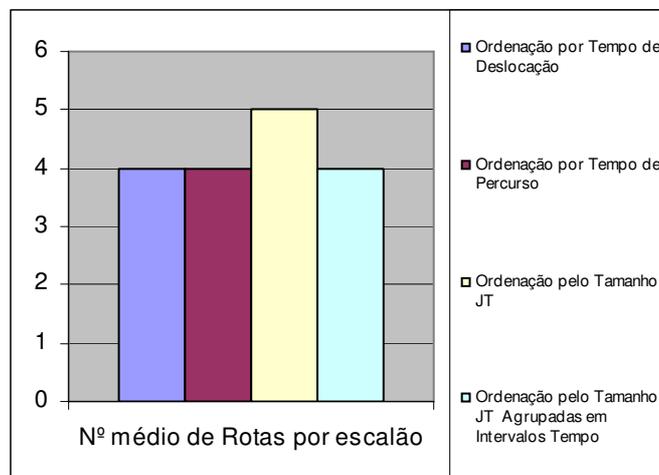


Figura 11: Número de rotas / Número veículos.

Para algumas instâncias, e independentemente do método de ordenação de dados escolhido, verificou-se a existência de clientes não inseridos na fase construtiva da heurística, sendo estes sempre inseridos posteriormente na fase de melhoramento.

3. Número de rotas (número de veículos)

Os piores resultados obtiveram-se para o método tamanho das janelas temporais ordenadas (Figura 10), obtendo-se com os outros três métodos valores muito próximos ou mesmo, na maior parte dos casos, iguais.

4. Tempos de execução do algoritmo

Os tempos de execução dependem bastante das localizações dos clientes, associadas às respectivas janelas temporais. Em média são necessários 50 segundos para os casos com 50 clientes e 1min30seg para os casos com 150 clientes, não sendo no entanto esta relação linear. Comparando os tempos de processamento entre os quatro métodos de ordenação (Figura 11), verifica-se que no caso do tempo de deslocação ordenado e no tempo de percurso ordenado, obtêm-se resultados mais rapidamente que no caso dos outros dois métodos.

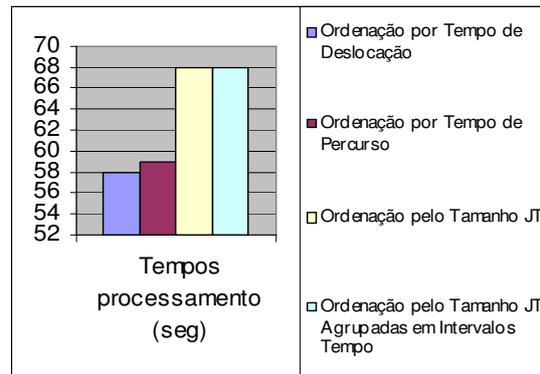


Figura 12: Tempos de processamento.

6 Conclusões

Neste artigo apresenta-se uma heurística composta para a determinação de rotas para veículos, que foi desenvolvida para uma empresa de distribuição de produtos alimentares. A primeira fase da heurística é construtiva, baseia-se no algoritmo do vizinho mais próximo e utiliza uma métrica de acordo com quatro métodos de ordenação de dados: tempos de percurso, tempos de serviço, janelas temporais e janelas temporais por intervalos de tempo.

A segunda fase da heurística está dividida em duas subfases baseadas no algoritmo 2-OPT. Na primeira fase faz-se uma optimização local considerando individualmente cada rota, enquanto na segunda subfase se procede a uma optimização local com trocas entre as várias rotas. Esta última subfase incorpora uma componente aleatória.

O principal objectivo da empresa era acomodar um crescente número de clientes sem alterar o número de veículos utilizados na distribuição diária, uma vez que recorria a veículos subcontratados e o contrato permitia flexibilidade na gestão das rotas. Por outro lado a empresa procurava responder melhor às alterações de última hora nas encomendas dos clientes. O sistema informático onde a heurística descrita neste artigo foi implementada tornou estes objectivos possíveis pela conjugação dos bons resultados obtidos com esta heurística composta (melhores que o anterior planeamento manual) com os baixos tempos de processamento. Por comparação com os resultados dos planeamentos manuais anteriores podemos afirmar que a heurística composta produz resultados bastante satisfatórios, quer em termos de tempo total dos percursos quer em termos de tempo de processamento.

7 Referências

- [1] Croes, A., "A method for solving travelling salesman problem", *Operations Research* (1958), 6, 791-812.
- [2] Derigs, U., and Metz, A., "A matching-based approach for solving a delivery pick-up vehicle-routing problem with time constraints", *OR Spektrum*, (1992) 14 (2), 91-106.
- [3] Gendreau, M., Laporte, G., and Vigo, D., "Heuristics for the travelling salesman problem with pickup and delivery", *Computers & Operations Research*, (1999) 26, 699-714.

- [4] Helsgaun, K., “An effective implementation of the Lin-Kernighan travelling Salesman heuristic”, *European Journal of Operational Research*, (2000) 126(1), 106-130.
- [5] Johnson, D., and McGeoch, L., “The Traveling Salesman Problem: a case study in local optimization”, *Local Search in Combinatorial Optimization*, E.H.L. Aarts and J.K. Lenstra (eds.), Wiley, N.Y., (1995), 215-310.
- [6] Laporte, G., “The vehicle routing problem: An overview of exact and approximated algorithms”, *European Journal of Operational Research*, (1992), 59(2), 345-358.
- [7] Laporte, G., Gendreau, M., Potvin, J., and Semet, F., “Classical and modern heuristics for the Vehicle routing problem”, *International Transactions in Operational Research*, (2000) 7, 285-300.
- [8] Laporte, G., “The travelling salesman problem: An overview of exact and approximated algorithms”, *European Journal of Operational Research*, (1992), 59(2), 231-248.
- [9] Lin, S., “Computer solutions to the travelling salesman problem”, *Bell System Technical Journal*, (1965) 44, 2245-69.
- [10] Lin, S., and Kernighan, B., “An effective heuristic algorithm for the travelling salesman problem”, *Operations Research*, (1973) 21, 498-516.

Anexo

Neste exemplo existem 50 clientes, com encomendas de vários tipos de produtos, e três tipos de veículos diferentes. Os resultados apresentados são relativos a dois dos quatro métodos de ordenação de dados. O primeiro método - ordenação pelo tempo de deslocação, é o método que se obteve piores resultados, em termos do total dos tempos de percurso dos veículos. O segundo método apresentado - ordenação pelo tamanho das janelas temporais agrupadas por intervalos de tempo, é o método que apresentou melhores resultados.

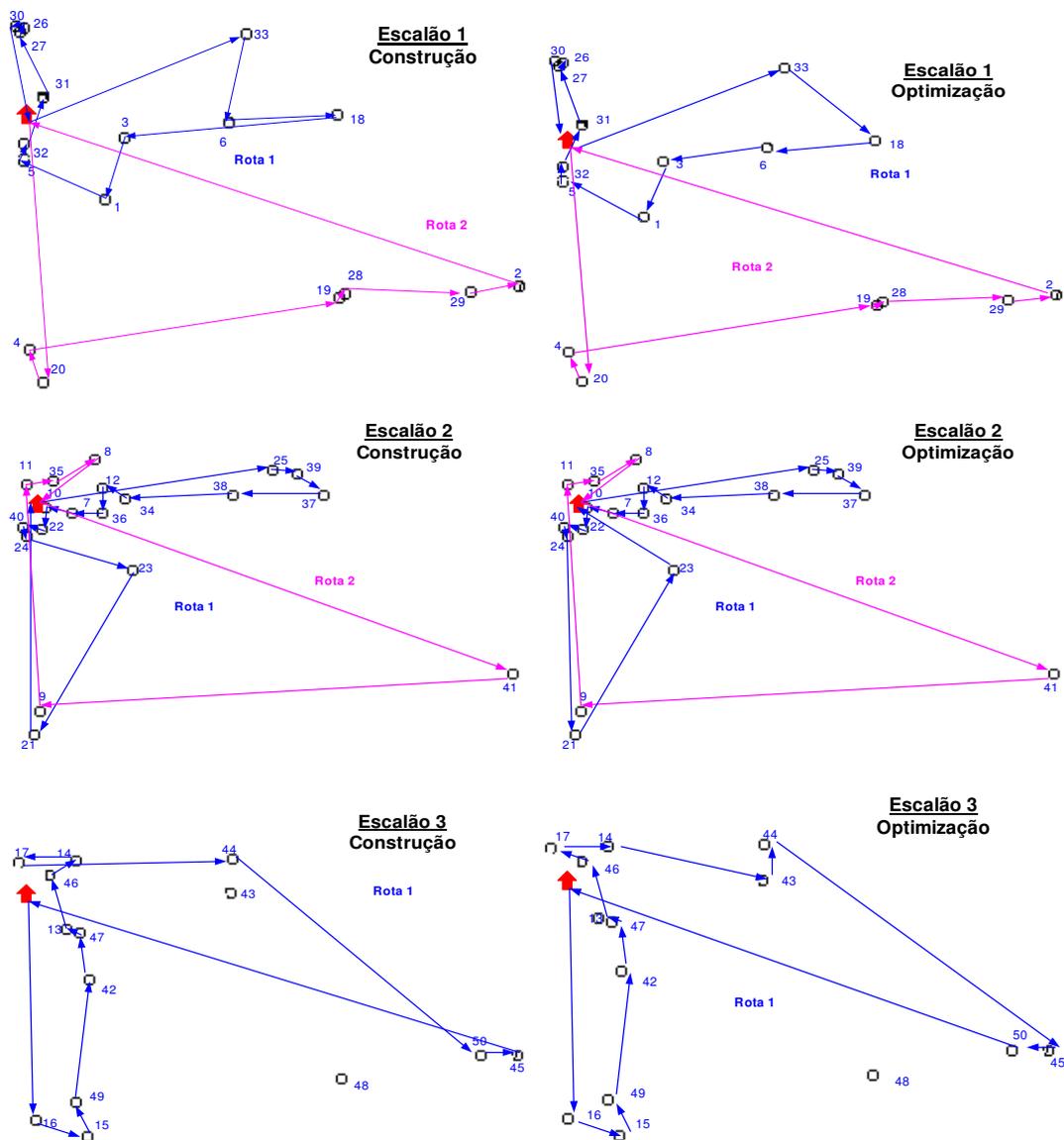
É de salientar, que nos resultados da fase construtiva do primeiro método existe um cliente pertencente ao escalão 3 que não foi inserido na rota. Sendo posteriormente inserido na fase de optimização.

1. Ordenação por tempo de deslocação

Fase 1 – Construtiva	Escalão 1		Escalão 2		Escalão 3
	Rota 1	Rota 2	Rota 1	Rota 2	Rota 1
Número Clientes	11	6	14	5	12
Hora de Saída do Armazém	6h50	6h42	6h56	5h35	6h48
Hora de Chegada ao Armazém	17h19	16h39	18h59	14h12	20h06
Hora Almoço	12h40 às 13h40	12h14 às 13h14	12h18 às 13h18	12h06 às 13h06	12h18 às 13h18
Tempo de Percurso Total	9h23	8h57	11h03	7h37	12h17
Carga Total (Kg)	12170	5290	9930	3931	15050

Fase 2 – Optimização local	Escalão 1		Escalão 2		Escalão 3
	Rota 1	Rota 2	Rota 1	Rota 2	Rota 1
Número Clientes	11	6	14	5	13
Hora de Saída do Armazém	6h56	6h42	6h56	5h35	6h48
Hora de Chegada ao Armazém	16h48	16h39	18h46	14h12	20h35
Hora Almoço	12h09 às 13h09	12h14 às 13h14	12h18 às 13h18	12h06 às 13h06	12h18 às 13h18
Tempo de Percurso Total	8h52	8h57	10h50	7h37	12h46
Carga Total(Kg)	12170	5290	9930	3931	15210

Ordem de visita aos clientes:



2. Ordenação pelo tamanho das Janelas Temporais agrupadas em intervalos de tempo:

Fase 1 - Construtiva	Escalão 1		Escalão 2		Escalão 3	
	Rota 1	Rota 2	Rota 1	Rota 2	Rota 1	Rota 2
Número Clientes	11	6	14	5	11	3
Hora de Saída do Armazém	5h56	6h42	6h56	6h00	6h48	6h11
Hora de Chegada ao Armazém	17h19	16h39	18h59	14h37	19h26	11h58
Hora Almoço	12h40 às 13h40	12h14 às 13h14	12h18 às 13h18	12h04 às 13h04	12h18 às 13h18	-
Tempo de Percurso Total	9h23	8h57	11h03	7h37	11h37	4h46
Carga Total (Kg)	1217	5290	9930	3931	14550	760

Fase 2 – Optimização Local	Escalão 1		Escalão 2		Escalão 3	
	Rota 1	Rota 2	Rota 1	Rota 2	Rota 1	Rota 2
Número Clientes	11	6	14	5	11	3
Hora de Saída do Armazém	6h39	6h42	6h00	6h56	6h48	6h17
Hora de Chegada ao Armazém	16h16	14h02	18h20	11h28	17h24	10h33
Hora Almoço	12h47 às	12h03 às	12h10 às	-	12h14 às	-
Hora Almoço	13h47	13h03	13h10		13h14	
Tempo de Percurso Total	8h37	6h20	11h20	3h32	9h36	3h15
Carga Total (Kg)	11005	6455	9862	3999	14520	790

Ordem de visita aos clientes:

