

PROBLEMA DE ROTEAMENTO DE VEÍCULOS CAPACITADOS (PRVC): SOLUÇÃO MANUAL X BUSCA DISPERSA

Gustavo Cunha de Bittencourt

COPPE/UFRJ

Stanley Rodrigues

Instituto Militar de Engenharia
Praça Gen. Tibúrcio, 80, Urca, 22290-270
Rio de Janeiro/RJ
sstanleyy@gmail.com

Paulo Oswaldo Boaventura Netto

Samuel Jurkiewicz

COPPE/UFRJ

Av. Athos da Silveira Ramos, 149,bl.F, sl.103 - Cidade Universitária, Ilha do Fundão – RJ
gustavo.bittencourt@engenharia.ufjf.br
{boaventu, jurki}@pep.ufrj.br

RESUMO

Dentre os diversos problemas de otimização combinatória estudados, o Problema de Roteamento de Veículos (PRV) é um dos que mais tem recebido atenção ultimamente, dada a sua complexidade e relevância econômica. O Problema de Roteamento de Veículos Capacitados (PRVC), versão mais clássica dos PRVs, consiste em buscar a alocação ótima de entregas para uma frota de veículos homogêneos situados em um depósito, minimizando a distância total percorrida pelos veículos de forma que todos os pedidos sejam entregues. Este trabalho apresenta a implementação de um sistema computacional para a solução de um PRVC baseado na meta-heurística Busca Dispersa (*Scatter Search*), bem como uma comparação entre os resultados obtidos com a sua utilização e os de uma solução obtida manualmente, para a determinação das rotas de entrega de um frigorífico aviário na região de Juiz de Fora - MG.

Palavras-chave: Metaheurísticas, Problema de Roteamento de Veículos, Busca Dispersa, Aplicações de Teoria dos Grafos.

Área - PO aplicada a Logística e Transportes

ABSTRACT

Among the several combinatorial optimization known problems, the Vehicle Routing Problem (VRP) is one of those who is receiving bigger attention lately due to its complexity and economic importance. The Capacitated Vehicle Routing Problem (CVRP), the classical version of Vehicle Routing Problems, consists of seek the optimal allocation of deliveries to a homogeneous fleet located in a deposit, minimizing the total distance traveled by vehicles since that all orders are delivered. This paper describes the implementation of a computational system based on the Scatter Search metaheuristic to solve a CVRP, as well as a comparison between the results obtained with its use in relation to a solution obtained manually to determine the delivery routes of a poultry slaughterhouse in the region of Juiz de Fora - MG.

Keywords: Metaheuristics, Vehicle Routing Problem, Scatter Search, Graph Theory Applications.

Area - OR applied to Logistics and Transportation

1. Introdução

Dentre as diversas áreas de gestão das organizações, a logística vem crescendo em destaque, tanto no meio acadêmico quanto no empresarial, em vista da sua importância estratégica e do impacto financeiro positivo trazido pela melhoria dos processos. A gestão eficiente dos processos logísticos pode se apresentar como vantagem competitiva, segundo Christopher [Chr97], sendo a fonte desta vantagem a diferenciação da organização aos olhos dos clientes e a redução dos custos de operação.

Vitasek [Vi10] define a logística como o processo eficiente de planejamento, implementação e controle de procedimentos para o transporte e armazenagem eficientes de bens, serviços e informações do ponto de origem ao ponto de consumo com o objetivo de atender aos requisitos do cliente.

Nos Estados Unidos e na União Europeia, os fretes absorvem até 60% do gasto logístico total, Rodrigues [Ro07]. Dados do IBGE [IBGE11] revelam a importância dos setores de transporte (que engloba transporte de carga e passageiros), armazenagem e correio para a economia brasileira, sendo responsável por 5,3% do valor adicionado a preços básicos ou 4,5% do PIB em 2010. Isto equivale a mais de 166 bilhões de reais movimentados, um aumento de 5,3% em relação ao ano de 2009. O modal rodoviário é responsável por 60% do total de cargas movimentadas no Brasil (chegando a 90% no estado de São Paulo), e a frota de caminhões é da ordem de 1.800.000 veículos, Tigerlog [Ti10].

Toth e Vigo [TV02] atribuem grandes economias ao uso de procedimentos computadorizados para o planejamento da distribuição, variando de 5% a 20% do custo total de transportes, sendo estes custos responsáveis por até 20% do preço dos bens de consumo. No caso estudado, uma técnica de pesquisa operacional se propõe substituir a determinação manual dos itinerários, o que envolve, habitualmente, sensível economia de meios e de trabalho. Esta pesquisa se justifica, portanto, pela sua relevância econômica, visto que no caso estudado, como na maior parte das indústrias, a atividade de transporte representa um dos elementos mais importantes na composição do custo logístico.

Ainda cabe salientar os benefícios colaterais resultantes da utilização de tais sistemas, como a redução nos níveis de poluição atmosférica, ruído e consumo de recursos naturais que soluções logísticas aperfeiçoadas podem trazer, diminuindo o percurso efetuado pelos veículos e aumentando a vida útil dos mesmos.

Este trabalho está estruturado da seguinte forma: a seção 2 consta de uma breve revisão bibliográfica sobre os PRVs e os sistemas de informação geográfica; na seção 3 é apresentado o problema, que é definido na seção 4; a seção 5 apresenta uma descrição do caso estudado; a seção 6 os métodos utilizados para a solução do problema e decisões de implementação; a seção 7 mostra os experimentos realizados e resultados obtidos; a seção 8 revela as conclusões oriundas do trabalho.

2. Revisão Bibliográfica

Um dos mais conhecidos problemas de otimização combinatória é o Problema do Caixeiro Viajante (PCV), que consiste em determinar o circuito mais curto para se percorrer um dado número de pontos e retornar à origem, passando uma única vez em cada um deles, Hillier e Liebermann [HL06]. Este modelo, entretanto, não reflete a realidade da maior parte das organizações, já que estas contam com veículos sujeitos a restrições diversas e que percorrem rotas distintas. Determinar a melhor alocação dos veículos disponíveis nessa situação é o objetivo do PRV, que apresenta diversas variantes que atendem às diferentes situações encontradas.

2.1 O PRV

O PRV foi proposto, inicialmente, por Dantzig e Ramser [DR59] como o problema de planejar um conjunto de entregas, utilizando uma coleção de rotas ótimas. Estas rotas são percorridas por veículos que devem partir de um ou mais depósitos para um determinado número de clientes espalhados geograficamente, sujeitas a um conjunto de restrições. Sua versão capacitada é o Problema de Roteamento de Veículos Capacitados (PRVC). Em Laporte [La92], o PRVC é definido e se mostra uma visão geral dos algoritmos exatos e das heurísticas a ele aplicadas. Os algoritmos exatos para o PRV são classificados em três grandes categorias: métodos de busca direta em árvore, programação dinâmica e programação linear inteira.

Segundo Alvarenga (2005) os PRVs estão, em sua grande maioria, na classe de problemas NP-árido, o que aponta para o uso de métodos heurísticos em instâncias de porte mais significativo., Em Dantzig e Ramser [DR59] já se encontra a descrição de um método heurístico para a solução do PRV. Posteriormente foram apresentadas outras heurísticas, como o algoritmo de economias de Clarke e Wright em 1964 [GW64] e o algoritmo de varredura (*sweep algorithm*) proposto por Wren em 1971 [Wr71], embora geralmente sua autoria seja atribuída a Gillett e Miller [GM74].

As metaheurísticas surgiram da necessidade de redução do risco de paradas em ótimos locais distantes do ótimo global (ver, por exemplo, [RH02]). São metamodelos de heurísticas que precisam apenas ser adaptados ao problema em estudo, dispensando a construção de algoritmos específicos. Usam frequentemente recursos de aleatoriedade e combinações de soluções para permitir a diversificação das regiões exploradas. Dentre elas, Laporte [La92] aponta a Busca Tabu como um dos métodos mais promissores na busca de boas soluções para os PRVs. Ochi e Tortelly [OT03] apresentaram uma nova meta-heurística híbrida baseada em conceitos do *Greedy Randomized Adaptive Search Procedure* (GRASP) e da Busca Tabu, para a solução do Problema de Roteamento Periódico de Veículos (PRPV); Galvão *et al.*[GBFH97] apresentam uma implementação baseada no *Simulated Annealing* para resolver um PRVC. A Busca Dispersa (*Scatter Search*) tem sido utilizada por diversos autores para a solução de PRVs complexos, como relatado em Rego e Leão [RL05]; Sosa, Galvão e Gandelman [SGG07]; e Belfiore *et al.* [BY06]. Trata-se da meta-heurística utilizada neste trabalho para a solução do PRV proposto.

Como os PRVs são problemas de distribuição geográfica, os vértices componentes do problema podem estar relacionados aos comprimentos de via a serem percorridos, ou então referenciados em um sistema de coordenadas, de forma que as distâncias euclidianas entre os mesmos possam ser diretamente calculadas, Câmara *et al* [CCMMH97].

2.2 Cartografia e sistemas de informações geográficas

O método padrão de localização na superfície terrestre define que todo ponto corresponde à interseção entre um paralelo e um meridiano. Os meridianos são os círculos máximos da esfera, cujo plano contém o eixo dos polos. Os paralelos são linhas que cruzam perpendicularmente o eixo dos polos. Esse é o chamado sistema de coordenadas geográficas, ou terrestres, Câmara *et al.*, [CCMMH97].

Segundo Câmara e Queiroz [CQ10], um sistema computacional pode ser denominado Sistema de Informações Geográficas (SIG) quando é capaz de tratar computacionalmente os dados geográficos e recuperar informações, baseando-se não somente nas características alfanuméricas delas, como também na sua localização espacial. Para o seu uso, é importante que seu usuário possa visualizar um ambiente que proporcione uma relação entre o conhecimento de seu domínio e sua localização geográfica. Tão variadas quanto as áreas de atuação de um SIG, são as funções e objetivos que cada um deles agrega. Ainda assim, é possível encontrar um conjunto de componentes básicos comuns a todos eles, que são: interface com o usuário; entrada e integração de dados; funções de processamento; visualização e plotagem; e armazenamento e recuperação de dados.

3. O Problema

O modelo abaixo apresentado é o de um PRVC acrescentado de uma restrição relativa ao número máximo de pedidos permitidos por caminhão. O grafo $G = (V, A)$ associado ao problema possui um conjunto de vértices $V = \{0, \dots, n\}$ que representam localidades (clientes ou cidades), estando o depósito no vértice 0 e um conjunto A de arestas associadas às possibilidades de trânsito entre os vértices. Cada arco (i, j) , $i \neq j$, é associado a uma matriz de valores $C = [c_{ij}]$ não negativa. Considera-se que o depósito dispõe de um número dado K (constante) de veículos disponíveis, todos de mesma capacidade Q , cada um podendo atender a um número máximo M_k de pedidos. As seguintes premissas e restrições devem ser respeitadas:

1. Cada vértice em $V \setminus \{0\}$ é visitado apenas uma vez e por exatamente um veículo;
2. Todas as rotas se iniciam e terminam no depósito;
3. Restrição de capacidade: a cada vértice $i > 0$ é atribuído um peso não-negativo (ou demanda) d_i e a soma dos pesos de qualquer rota do veículo não pode exceder a capacidade Q do veículo.
4. Restrição de número máximo de pedidos: cada vértice $i > 0$ possui um determinado número de pedidos p_i , cujo somatório para cada veículo k não deve exceder M_k .
5. Restrição de atribuição: o caminhão $k = 1$ deve obrigatoriamente atender aos clientes $i = 1$ e $i = 2$, por razões que serão explicadas posteriormente.

4. Modelagem Matemática

O PRVC modelado neste trabalho seguiu uma adaptação do modelo de índice triplo proposto por Toth e Vigo [TV02], contando com a adição das restrições citadas na definição do problema:

$$\text{Minimizar} \quad \sum_{i=0}^n \sum_{j=0}^n c_{ij} \sum_{k=1}^K x_{ijk} \quad 1)$$

$$\text{Sujeito a} \quad \sum_{k=1}^K y_{ik} = 1 \quad (i = 1, \dots, n) \quad 2)$$

$$\sum_{k=1}^K y_{0k} = K \quad 3)$$

$$\sum_{j=0}^n x_{ijk} = \sum_{j=0}^n x_{jik} = y_{ik} \quad (i = 0, \dots, n), (k = 1, \dots, K) \quad 4)$$

$$\sum_{i=1}^n d_i y_{ik} \leq Q \quad (k = 1, \dots, K) \quad 5)$$

$$\sum_{i=1}^n p_i y_{ik} \leq M_k \quad (k = 1, \dots, K) \quad 6)$$

$$y_{11} = y_{21} = 1 \quad 7)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1 \quad (\forall S \subseteq V \setminus \{0\}, |S| \geq 2, k = 1, \dots, K) \quad 8)$$

$$x_{ijk} \in \{0, 1\} \quad (i, j = 0, \dots, n; i \neq j), (k = 1, \dots, K) \quad 9)$$

$$y_{ik} \in \{0,1\} \quad (i = 0, \dots, n), (k = 1, \dots, K) \quad (10)$$

A função objetivo (1) busca a minimização da distância total percorrida, onde a variável binária x_{ijk} recebe 1 se o arco (i, j) é coberto pelo veículo k na solução ótima, e 0 se não. A variável binária y_{ik} recebe o valor 1 caso o vértice i seja visitado pelo veículo k na solução ótima, e 0 caso não seja. Portanto, a restrição (2) desta formulação impõe que cada vértice i seja visitado por apenas um veículo, a restrição (3) determina que K veículos devem deixar o depósito e a restrição (4) estabelece que o mesmo veículo k deve entrar e sair de um determinado vértice i . A restrição de capacidade (5) determina que a carga de qualquer veículo k não pode ultrapassar a sua capacidade Q . A restrição (6) impede que o número máximo de entregas M_k permitidas para caminhão k seja excedido, e (7) obriga o caminhão $k = 1$ a atender obrigatoriamente aos clientes $i = 1$ e $i = 2$. A restrição (8) elimina os subcircuitos ilegais (sub-rotas desconectadas do depósito), determinando que para qualquer subconjunto S de $V \setminus \{0\}$ o número total de arcos x_{ijk} ($j, i \in S$) pertencentes à solução ótima deve ser menor ou igual ao número de elementos de S menos um, conforme pode ser visto na **Figura 1**. As últimas restrições, (9) e (10), indicam que as variáveis x_{ij} e y_{ik} são binárias.

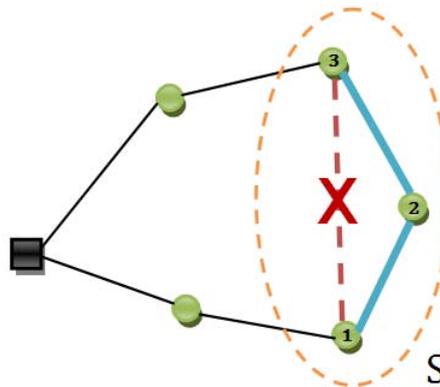


Figura 1 – Eliminação de subcircuitos ilegais

5. O estudo de caso

Este trabalho é dedicado ao estudo da distribuição, na cidade de Juiz de Fora (MG), de produtos de um frigorífico situado em Barbacena (MG). A definição das rotas de entrega não utiliza, atualmente, nenhum método de otimização computacional, valendo-se apenas da experiência prévia dos motoristas.

As entregas na cidade são feitas por três caminhões com 4.000 kg de capacidade cada, limitando-se o número de entregas por caminhão. Esta limitação foi estabelecida em vista da dificuldade em se determinar o tempo médio de entrega, tanto pela ausência de dados quanto pela grande variabilidade desses tempos (de 2 minutos em clientes menores a uma hora e meia em grandes supermercados). Entretanto, este número não é fixo: os caminhões que realizam as entregas dos bairros podem atender a até 30 entregas, enquanto o veículo (número 1) que realiza as entregas no centro da cidade pode ser carregado com até 45 entregas, em vista da proximidade geográfica e do menor volume de entregas aos clientes. Esta particularidade é modelada pela restrição (6) do modelo proposto, onde $M_1 = 45$.

Outra peculiaridade do problema estudado é a escassez de pontos de parada na região mais movimentada do centro da cidade, o que obriga o carro número 1 a estacionar em dois pontos pré-determinados, a partir dos quais as entregas são realizadas a pé pelos carregadores. Estes vértices são chamados de bolhas, e o seu número de pedidos p_i é o somatório do número de clientes abastecidos a partir de cada um deles (para os demais vértices, $p_i = 1$). A sua demanda

d_i também é dada a partir do somatório das demandas dos clientes abastecidos por cada um destes vértices. A restrição (7) impõe que os clientes pertencentes a estas bolhas sejam atendidos pelo carro 1, em consonância com a determinação da restrição (6).

O trabalho envolve a implementação de um sistema computacional baseado na meta-heurística Busca Dispersa para a solução do problema proposto. Os algoritmos aplicados ao projeto buscam determinar rotas com a menor distância total percorrida após a minimização do número de veículos a serem utilizados, com o objetivo comparar, dada a melhora obtida, o roteamento baseado apenas na experiência com o uso de métodos computacionais e, assim, verificar a viabilidade de uma mudança.

Como a métrica distância também é avaliada, o cálculo das rotas foi aliado ao georreferenciamento dos clientes atendidos pela empresa. Os endereços de cadastro de cada cliente se tornam então coordenadas em um SIG, que também foi utilizado para aferir a distância entre pares de pontos de visita da rota, formando uma matriz valorada de adjacência.

6. Métodos e Decisões de Implementação

6.1 Aspectos do Georreferenciamento

Este trabalho foi realizado com o auxílio da API do *Google Maps* [Go11], que é uma coleção de serviços que permite a incorporação de mapas e outros conteúdos do *Google* em aplicações web. A API é disponibilizada gratuitamente, desde que respeitadas algumas regras especificadas no termo de serviço; por exemplo, não se pode exceder certo número de consultas por dia. Para utilizá-la é necessário um cadastro que gera uma chave codificada, a ser inserida no código da aplicação.

A versão do serviço utilizada para o presente trabalho é a *Maps Javascript API*, que permite criar serviços utilizando mapas e imagens de satélite. Ela é estruturada em torno da classe central *Gmap2*. As demais classes associadas permitem criar interações com os mapas, tais como marcadores de posição, rotas mais curtas entre dois pontos e geocodificação de endereços. A *Google Maps API* apresenta uma série de restrições a operações de grande porte. Para a versão gratuita, o Termo de Serviço fixa um limite diário de 15 mil consultas aos seus bancos de dados para serviços como busca de rotas e conversão de endereços. Além disso, o serviço bloqueia automaticamente o acesso de um terminal que execute um grande número de consultas em um curto espaço de tempo.

Na primeira fase do projeto, todos os endereços dos clientes cadastrados para o itinerário estudado foram referenciados geograficamente com a ferramenta Google Maps. Em seguida, uma rotina foi implementada para a obtenção da distância (seguindo o traçado e orientação das ruas) entre cada par de vértices do problema.

Para a geração dos dados da matriz de pesos nas rotas, quase sempre o sistema necessitará executar um número muito grande de consultas de rotas para criar os arcos do grafo completo, já que para um número n de vértices são realizadas n^2 solicitações ao serviço da API. Este problema do bloqueio por frequência de acesso é resolvido através da implementação de um intervalo de tempo no laço de operações de consulta.

Por fim, os dados obtidos são armazenados em uma base SQL, que alimenta o algoritmo detalhado no próximo item.

6.2 A Busca Dispersa (*Scatter Search*)

Conforme mencionado na introdução, foi utilizada a Busca Dispersa (*Scatter Search*) como algoritmo de otimização de rotas, seguindo a implementação proposta por Bittencourt [Bi10]. As restrições (6) e (7) do modelo foram levadas em conta, visando a sua adequação à realidade do problema estudado.

A metodologia é baseada em populações, onde cada solução gerada para o problema é considerada um indivíduo, o que a assemelha em parte aos Algoritmos Genéticos. Um de seus aspectos importantes é, segundo [SGG07], a relação existente entre a capacidade do método de dirigir a busca a regiões promissoras e sua eficiência na exploração dessas regiões.

Esta meta-heurística proporciona o uso de estratégias flexíveis, que permitem o desenvolvimento de diferentes algoritmos com distintos graus de complexidade para as diferentes etapas da busca. Os conceitos e princípios fundamentais do método foram propostos na década de 70 por Glover [Gl77] com base em estratégias para combinar regras de decisão e restrições.

A Busca Dispersa guarda boas soluções encontradas durante o processo de busca em um conjunto de referência (*Refset*) e combina soluções a ele pertencentes com o intuito de capturar informação não contida nas soluções originais. Neste contexto, a classificação boa não se restringe apenas à qualidade da solução, mas também à sua diversidade em relação a outras soluções deste conjunto, [SGG07].

A Busca Dispersa é composta basicamente por cinco etapas, que serão detalhadas abaixo junto aos detalhes da implementação.

6.2.1. Etapa Geradora de Soluções Iniciais

Na primeira etapa é gerado um conjunto P de soluções iniciais com $Psize$ elementos, de onde é extraído um subconjunto que se denomina conjunto de referência *Refset*. Estas soluções iniciais são geradas a partir do algoritmo de varredura (*sweep algorithm*), no qual os vértices devem ser representados por suas coordenadas polares (θ_i, ρ_i) , onde θ_i é o ângulo e ρ_i é o comprimento do raio. A um vértice i^* arbitrário deve ser atribuído o valor $\theta_{i^*} = 0$, e os ângulos restantes devem ser computados a partir do arco $(0, i^*)$, como se o giro de uma semirreta varresse os demais clientes. Os vértices devem ser ordenados em ordem crescente de θ_i e um veículo k não utilizado deve ser escolhido para iniciar o preenchimento. A partir do vértice não roteado com o menor ângulo, os demais na sequência devem ser inseridos na rota k enquanto suas restrições de capacidade e número total de pedidos forem respeitadas. Quando uma das duas restrições não puder mais ser satisfeita, inicia-se o preenchimento do próximo veículo, até que todos os vértices sejam alocados.

Durante o preenchimento, quando identificado que o vértice i a ser inserido no veículo k é uma das duas bolhas, o número máximo de pedidos permitidos em k é alterado para 45. O preenchimento continua até que todos os demais vértices sejam alocados, e no final a viabilidade da solução criada é analisada, sendo aceita caso a outra bolha pertença à mesma rota que a primeira, ou descartada caso contrário.

Embora originalmente a Busca Dispersa utilize-se apenas de estratégias determinísticas, foram testadas também alternativas aleatórias para a criação do conjunto P . Na estratégia determinística, é criada uma solução com a varredura se iniciando a partir de cada vértice de cliente presente no problema, e as soluções encontradas são mais robustas e replicáveis. Já na aleatória, eram sorteados $PSize$ clientes a partir dos quais a varredura se iniciava, sendo obtidos com esta estratégia alguns resultados melhores e em tempo computacional inferior aos obtidos com a estratégia determinística, mas com grande variabilidade.

6.2.2. Etapa de Melhoria

Consiste de uma busca local para melhorar as $Psize$ soluções inicialmente obtidas, tanto através de métodos de melhoria intra-rotas quanto inter-rotas. Para qualquer desses procedimentos, dois critérios são utilizados para determinar a substituição da solução antiga pela nova solução gerada: ela deve ser viável e ter o valor da função objetivo (somatório da distância total percorrida por todos os veículos) inferior ao o da antiga.

O primeiro procedimento de melhoria realizado é a melhoria intra-rota, com a utilização da heurística 2-opt [Cr58]. Para cada solução s criada, esta heurística realiza trocas de vértices dois a

dois ($\lambda = 2$), dentro de cada uma das rotas, sendo que a cada troca a solução resultante é armazenada em uma matriz auxiliar. A distância total percorrida pelos veículos nesta matriz é calculada e, caso esta seja inferior à distância calculada previamente, a solução s é substituída pela matriz auxiliar. Em seguida, a solução é submetida sequencialmente aos três procedimentos de melhoria inter-rotas: Realocação, intercâmbio e cruzamento.

A heurística de realocação remove cada vértice v de cada rota $k = w$ (onde w é o índice de um laço que percorre todas as rotas de uma dada solução s) e tenta inseri-lo em cada posição i de cada um das demais rotas $k \neq w$ de uma matriz auxiliar criada como cópia da solução s . A distância total percorrida pelos veículos nesta matriz é calculada, bem como a sua viabilidade, e caso seja viável com a distância total inferior à distância calculada previamente a solução s é substituída pela matriz auxiliar.

Em seguida executa-se a heurística de intercâmbio que, para cada vértice v de cada rota $k = w$, tenta realizar a troca deste com cada vértice v de cada um das demais rotas $k \neq w$ (de uma matriz auxiliar criada como cópia da solução s). Novamente é avaliada a viabilidade e distância total.

O último movimento de melhoria inter-rotas executado é a heurística de cruzamento. Este procedimento seleciona as rotas de uma dada solução s duas a duas, as corta em uma determinada posição i e realiza o cruzamento, conectando a primeira parte da primeira rota na segunda parte da segunda rota, e vice versa. O índice i é percorrido em um laço que vai da segunda posição de entrega até a penúltima posição da rota com o menor número de entregas.

Por fim, executa-se novamente heurística 2-opt, e com isto se tem uma das soluções iniciais criadas. Este processo de construção se repete até que se tenha $PSize$ soluções iniciais.

6.2.3. Geração do Conjunto de Referência

Nesta etapa é construído ou atualizado o conjunto de referência. O *Refset* é composto por b soluções, sendo $b = b_1 + b_2$, onde b_1 é o número de soluções de qualidade que compõem o conjunto, isto é, as soluções com menor distância total, e b_2 é o número de soluções de diversidade que compõem o *Refset*, isto é, as soluções que mais se distinguem das outras que já o constituem. Esta distinção é calculada através da máxima diferença mínima entre uma determinada rota e todas as demais do *Refset*, sendo esta diferença entre duas rotas definida como o somatório de um fator que recebe 1 caso o vértice i não pertença à mesma rota em duas soluções, e 0 caso ele pertença.

A construção do *Refset* se dá a partir do conjunto P , sendo adotado neste trabalho $b_1 = 5$ e $b_2 = 5$, levando como base os testes realizados por [SGG07]. Como existe a possibilidade de que haja soluções idênticas apenas com o índice das rotas alterados (por exemplo, a primeira rota da solução s_1 ser igual à terceira rota da solução s_2 , e assim para todas as demais rotas), foi criado um algoritmo auxiliar que avalia a equivalência entre as rotas, e só permite que uma nova solução seja inserida no *Refset* se ela for diferente de todas as outras já presentes, evitando a redundância de combinações e esforço computacional desnecessário na etapa seguinte.

A atualização do *Refset* acontece quando novas soluções que atendem aos requisitos para ingressar no conjunto são encontradas, substituindo a pior solução de b_1 caso a nova caracterize-se como uma solução de qualidade, ou a pior solução de b_2 caso caracterize-se como uma solução de diversidade. Assim, o *Refset* mantém seu tamanho constante, mas as soluções pertencentes ao mesmo melhoram durante o processo de busca.

6.2.4. Geração de Subconjuntos

Com o *Refset* criado, são gerados os subconjuntos de soluções que serão combinadas. Estes subconjuntos são gerados dois a dois entre todas as soluções presentes no *Refset*.

6.2.5. Combinação de Soluções

Utilizando os subconjuntos de soluções gerados na etapa anterior, as duas soluções de cada subconjunto são comparadas, e um algoritmo estabelece quais são as rotas equivalentes entre elas (são classificadas rotas equivalentes nesta situação como aquelas que possuem maior número de vértices em comum nas duas soluções, em um processo semelhante ao que ocorre no cálculo da divergência entre as soluções). Para cada rota da primeira solução, os vértices que não pertencem também à rota estabelecida como sua equivalente na segunda solução são retirados e armazenados em um vetor chamado *pool_vertices*, e os que pertencem a ambas as rotas são fixados em uma nova solução criada. Um algoritmo de inserção é executado, inserindo os clientes armazenados no *pool_vertices* na rota mais próxima da nova solução, segundo o critério definido por [SGG07]: Para cada cliente *i* não fixado encontra-se a rota mais próxima, que é aquela onde a distância do último cliente alocado na rota ao cliente *i*, mais a distância do cliente *i* ao depósito, é a menor possível e satisfaz às restrições do problema. O cliente a ser inserido na rota mais próxima é aquele que apresentar o menor valor de um fator dado pelo somatório destas duas distâncias dividido pela demanda do cliente. O cliente com menor valor deste fator é inserido na rota mais próxima da nova solução, e o processo de inserção se reinicia, até que todos os clientes do *pool_vertices* sejam alocados. A nova solução passa então pelos métodos de melhoria já citados anteriormente (2-opt, realocação, intercâmbio, cruzamento, 2-opt, na respectiva ordem), e o processo de combinação de soluções se repete até que todos os subconjuntos sejam combinados.

Ao fim do procedimento anterior, o *Refset* é atualizado novamente e são gerados os subconjuntos de soluções a serem combinadas, excluindo aquelas que já foram combinadas em algum procedimento anterior.

A execução do algoritmo finaliza quando o *Refset* não puder mais ser atualizado, tanto em relação às soluções de qualidade (b_1) quanto em relação às soluções de diversidade (b_2).

7. Experimentos e Resultados

Para a realização dos testes foi utilizado uma dia típico de entregas, com 99 pedidos entregues pelos três veículos, sendo 6 pertencentes à bolha 1 e 9 pertencentes à bolha 2, configurando então 86 vértices mais o depósito. A solução manual gerada pelo frigorífico e utilizada no dia da entrega resultou em uma distância total de 827.337 metros percorridos pelos veículos, conforme imagem abaixo:

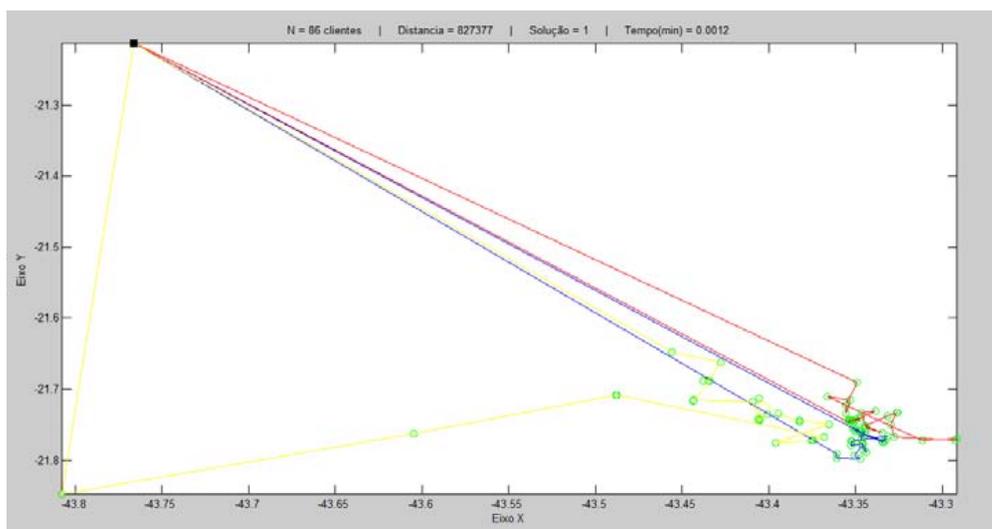


Figura 2 – Solução manual para o problema

O sistema foi implementado e executado com a utilização do software MathWorks MatLab® 7 R14. Uma primeira solução obtida com a relaxação da restrição relativa ao número total de pedidos (6) e das bolhas pertencerem à mesma rota (7) obteve a distância total percorrida de 628.918m. Após a inserção da restrição (6), com a relaxação apenas da restrição (7), obteve-se

uma solução com 769.010m de distância total, mas permitindo que a bolha 1 e a bolha 2 ficassem separadas em duas rotas de 45 entregas cada.

Dentre os testes realizados para diferentes tamanhos de *PSize* (10, 15, 20 e 86) – nos três primeiros casos utilizando a estratégia aleatória, e no último utilizando a estratégia determinística - o melhor resultado obtido foi com a utilização de *PSize* = 20. Após 88,4 minutos de processamento, a distância total percorrida pelos veículos na melhor solução viável obtida foi de 757.985 m, um ganho de 8,38% em relação à solução obtida manualmente (Figura 3).

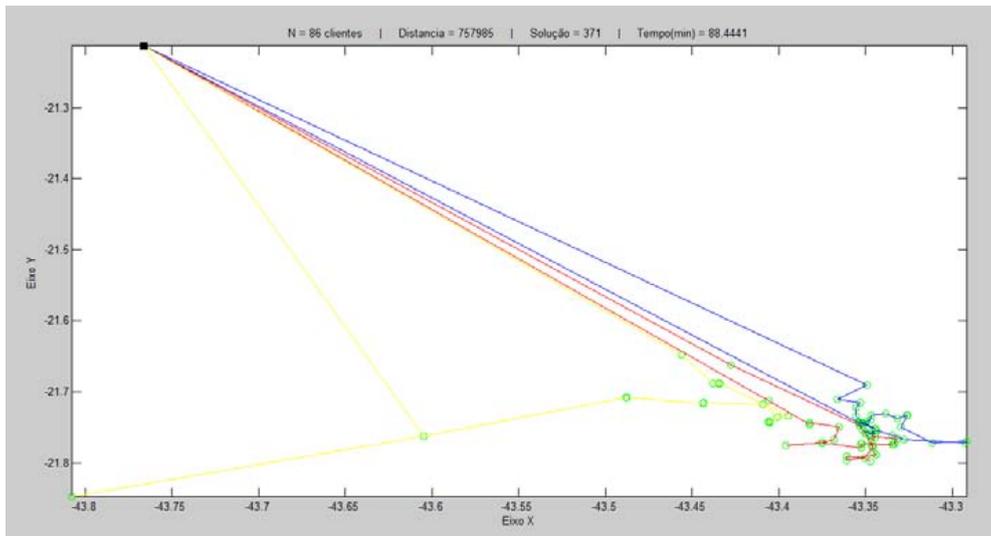


Figura 3 – Melhor solução obtida para o problema

Testes utilizando *PSize* = 10 também revelaram bons resultados em um tempo computacional inferior, sendo solução obtida como melhor solução a distância de 775.196 m (6,30% de melhoria) após 34,5 minutos (Figura 4).

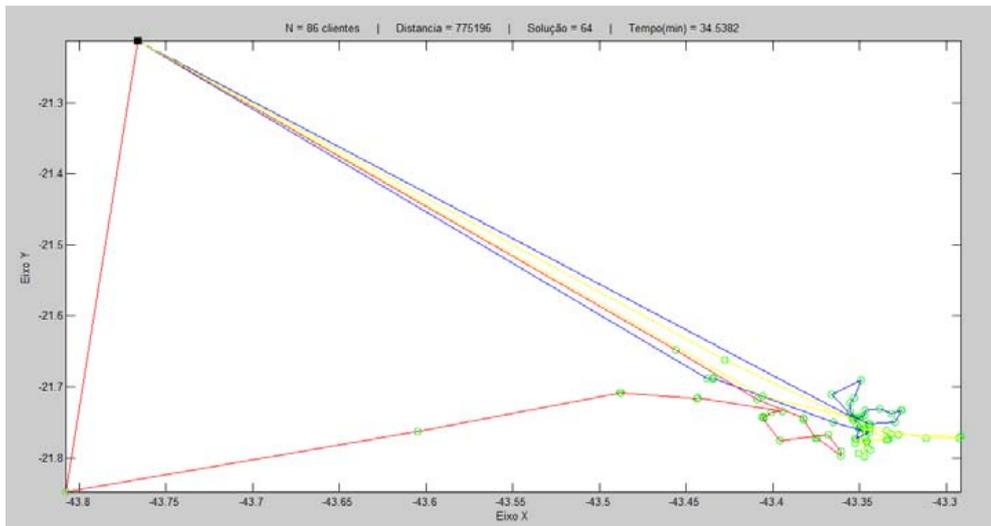


Figura 4 – Boa solução obtida com menor custo computacional.

8. Conclusões
Neste trabalho foi utilizada uma versão da meta-heurística de Busca Dispersa (Scatter Search) para otimizar um Problema de Roteamento de Veículos Capacitado representando o problema real de distribuição de mercadorias de um frigorífico aviário em Juiz de Fora-MG. Os clientes a serem abastecidos pela rota dos veículos foram georreferenciados com boa precisão e essa localização serviu de base para as distâncias entre todos os pontos, que vieram a compor a matriz de adjacência valorada do problema.

Os testes realizados contaram com rotas registradas para apenas um dia, devido à ausência de registros passados e à baixa disponibilidade de tempo dos motoristas para elaborar manualmente rotas que servissem para comparação com as otimizadas. Entretanto, os resultados obtidos já servem como um bom indicativo da eficiência do algoritmo frente às soluções encontradas manualmente até então, o que confirma a proposição de Toth e Vigo [TV02].

O georreferenciamento dos pontos garante uma boa análise do caso real, e faz com que essa metodologia seja facilmente aplicada a situações com diferentes relações de clientes, de acordo o dia de pedidos realizados.

Como trabalhos futuros, estão previstas a utilização de novas heurísticas construtivas e de melhoria, testes mais robustos para diferentes *Refsets* e *Psizes*, bem como a utilização de outras linguagens de programação e arquiteturas de processamento paralelo para a redução do custo computacional, viabilizando a sua utilização prática.

9. Referências

- [A105] **Alvarenga, G.B.** *Um Algoritmo Híbrido para o Problema de Roteamento de Veículos Estático e Dinâmico com Janela de Tempo*. Tese de Doutorado, Universidade Federal de Minas Gerais (Departamento de Ciência da Computação), Belo Horizonte, 2005.
- [BY06] **Belfiore, P. P. e Yoshizaki, H. T. Y.** *Scatter search para problemas de roteirização de veículos com frota heterogênea, janelas de tempo e entregas fracionadas*. Prod. [online], 16, n.3, p. 455-469, POLI-USP 2006 (<http://dx.doi.org/10.1590/S0103-65132006000300008>)
- [Bi10] **Bittencourt, G. C. de**, *Modelagem e Implementação de um Sistema Computacional para a Solução de um Problema de Roteamento de Veículos (PRV) com o Uso da Metaheurística Busca Dispersa (Scatter Search)*, 50f. Trabalho de Conclusão de Curso (Graduação em Engenharia de Produção) Universidade Federal de Juiz de Fora, Juiz de Fora, 2010.
- [CCMMH97] **Câmara, G.; Casanova, M.; Medeiros, C. B.; Magalhães, G. e Hemerly, A.**, *Anatomia de Sistemas de Informação Geográfica*, Vol. Único. UNICAMP- SAGRES, 1997.
- [CQ10] **Câmara, G. e Queiroz, G. R. d.**, *Introdução à Ciência da Geoinformação*, Arquitetura De Sistemas De Informação Geográfica, pp. 1–35, INPE, 2010 (<http://www.dpi.inpe.br/gilberto/livro/introd/index.html> 05/2010)
- [Chr97] **Christopher, M.** *Logística e gerenciamento da Cadeia de Suprimentos*. 1 ed. São Paulo: Pioneira, 1997.
- [CW64] **Clarke, G. e Wright, J.** (1964) *Scheduling of vehicles from a central depot to a number of delivery points*. Operations Research, 12(4):568–581.
- [Cr58] **Croes, G. A.** (1958) *A method for solving traveling salesman problems*. Operations Research, November/December 6:791-812.
- [DR59] **Dantzig, G. B. e Ramser, J. H.** (1959). *The Truck Dispatching Problem*. Management Science, Vol. 6, No. 1 (Oct., 1959), pp. 80-91 INFORMS
- [GBFH97] **Galvao, R. D.; Barros Neto, J. F.; Ferreira Filho, Virgílio J. M. ; Henriques, H B de Sousa.** *Roteamento de veículos com base em sistemas de informação geográfica*. Gest. Prod. [online], 4, n.2, p. 159-174, São Carlos, 1997. (<http://dx.doi.org/10.1590/S0104-530X1997000200005>)
- [GM74] **Gillett, B. e Miller, L.** (1974), *A heuristic algorithm for the vehicle dispatch problem*, Operations Research 22, 340-349.
- [GI77] **Glover, F.** (1977). *Heuristic for integer programming using surrogate constraints*. Decision Sciences, 8, 156-166.
- [Go11] **Google Developers**, *Google Maps JavaScript API V2 Reference*, (<https://developers.google.com/maps/documentation/javascript/v2/reference?hl=pt-BR>), 6, 2011
- [HL06] **Hillier, F. S.; Lieberman, G. J.**, *Introdução à pesquisa operacional*. 8 ed. São Paulo: McGraw-Hill, 2006.

- [IBGE11] Instituto Brasileiro de Geografia e Estatística, IBGE, (2011) *Indicadores IBGE: Contas Nacionais Trimestrais - Indicadores de Volume e Valores Correntes*. Abril / Junho 2011. (http://www.ibge.com.br/home/estatistica/indicadores/pib/pib-vol-val_201102caderno.pdf) 10-2011.
- [La92] Laporte, G. *The vehicle routing problem: an overview of exact and approximate algorithms*. European Journal of Operational Research, n 59: 345-358, 1992.
- [OT03] Ochi, L. S., Torttely Jr, A., (2003) *Uma metaheurística híbrida GRASP + TABU para o problema de roteamento periódico de uma frota de veículos*. Anais do XXXV Simpósio Brasileiro de Pesquisa Operacional (XXXV SBPO), 2003, Natal, RN.SOBRAPO, v.1. p.1 – 11
- [RL05] Rego, C. e Leão, P., *A Scatter Search Tutorial For Graph Based Permutation Problems*. Em C. Rego and B. Alidaee (Eds.) *Metaheuristic Optimization via Memory and Evolution: Tabu Search and Scatter Search*, (pp. 1-24). Kluwer Academic Publishers, 2005.
- [RH02] Ribeiro, C. C. e Hansen, P., *Essays and surveys in metaheuristics*. Kluwer, Boston, 2002.
- [Ro07] Rodrigues, P. R. A., *Introdução aos sistemas de transporte no Brasil e à logística internacional*. 4 ed. São Paulo: Aduaneiras, 2007.
- [SGG07] Sosa, N. G. M., Galvao, R. D. e Gandelman, D. A., *Algoritmo de busca dispersa aplicado ao problema clássico de roteamento de veículos*. *Pesqui. Oper.* [online]. 27, n.2, p. 293-310, 2007. (<http://dx.doi.org/10.1590/S0101-74382007000200006>) 10-2011
- [Ti10] Tigerlog – Consultoria e Treinamento em Logística, *Curiosidades em Logística, TIGERLOG – Consultoria e Treinamento em Logística*, http://www.tigerlog.com.br/logistica/curiosidades_rodoviario.asp 03-2010
- [TV02] Toth, P. e Vigo, D., *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, U.S.A, 2002.
- [Vi10] Vitasek, K., *Supply Chain Management Terms and Glossary*, Council of Supply Chain Management Professionals, 2010. (<http://cscmp.org/digital/glossary/glossary.asp>) 03-2010
- [Wr71] Wren, A., *Computers in Transport Planning and Operation*, Ian Allan, London, 1971.