

Minimização da Dispersão em um Problema de Roteamento de Veículos

Guilherme Dhein^a

gdhein@gmail.com

Marcelo Serrano Zanetti^b

marcelo.zanetti@ufsm.br

Felipe Martins Müller^d

felipe@inf.ufsm.br

Olinto César Bassi de Araújo^c

olinto@ctism.ufsm.br

Ghendy Cardoso Júnior^a

ghendy.ufsm.br

Universidade Federal de Santa Maria

Av. Roraima nº 1000 - Cidade Universitária - Bairro Camobi - Santa Maria - RS - 97105-900

^aPrograma de Pós-graduação em Engenharia Elétrica

^cDepartamento de Eletrônica e Computação

^cColégio Técnico Industrial de Santa Maria

^dPrograma de Pós-graduação em Informática

RESUMO

Neste trabalho é apresentado um problema de roteamento de veículos no qual as rotas são construídas considerando o posicionamento relativo dos veículos, de modo que estes se desloquem mantendo uma proximidade espacial e temporal. O objetivo é a minimização de uma métrica de dispersão não linear que promove um tipo de sincronização que permite um veículo receber auxílio dos demais em caso de emergência ou necessidade. Resultados computacionais obtidos com um Algoritmo Genético com Busca Local são apresentados para um conjunto de 24 instâncias derivadas do clássico Problema de Múltiplos Caixeiros Viajantes. A análise dos resultados demonstra que as soluções apresentam características de dispersão mínima desejada.

PALAVRAS CHAVE. Rotamento de veículos, Algoritmo Genético com Busca Local, Dispersão.

Área Principal: Logística e Transporte, Otimização Combinatória, Metaheurísticas.

ABSTRACT

We present a vehicle routing problem where routes are built considering the relative position of the vehicles, such a way to keep a spatial and temporal proximity. The objective is to minimize a nonlinear dispersion metric to support a kind of synchronization that allows a vehicle to be rescue by other vehicles in case of emergency or need. Computational results obtained with a Local Search Genetic Algorithm are presented for a set of 24 instances derived from the classical Multiple Traveling Salesman Problem. Analysing the results, we can see that the routes exhibit the desired characteristics of minimal dispersion.

KEYWORDS. Vehicle Routing, Local Search Genetic Algorithm, Dispersion.

Main Area: Logistic and Transportation, Combinatorial Optimization, Metaheuristics.

1. Introdução

Problemas de roteamento de veículos consideram situações nas quais se deseja calcular rotas para uma frota de veículos de forma que cada elemento de um conjunto de consumidores seja visitado uma única vez. Entre os objetivos mais considerados é possível citar minimização do custo total de operação, minimização do tempo total de deslocamento e maximização da qualidade do serviço ao cliente. No que se refere ao conjunto de restrições, especificamente serão abordados problemas que consideram tão somente atendimento de solicitações de serviços sem restrições secundárias, tais como capacidade ou janelas de tempo. Um recente estudo sobre a classificação e revisão destes problemas pode ser encontrado em Braekers et al. (2015).

Neste trabalho é apresentado um problema de roteamento de veículos (PRV) no qual os veículos devem visitar os consumidores mantendo uma configuração tal que permita um veículo receber reforço rapidamente dos demais caso necessário. O objetivo consiste em minimizar uma métrica não linear de dispersão que considera o posicionamento relativo entre os veículos e promove um tipo de sincronização espacial e temporal. O problema é denominado Problema de Roteamento de Veículos com Minimização da Dispersão (PRV-MD). Como método de resolução é proposto um Algoritmo Genético com Busca Local que incorpora estruturas e procedimentos projetados especificamente para o PRV-MD.

O posicionamento relativo dos veículos ou equipes tem sido abordado em diferentes contextos. Um exemplo é a busca por caminhos dissimilares, útil no transporte de cargas perigosas [Talarico et al., 2015] ou na redução da possibilidade de detecção em missões militares [Thyagarajan et al., 2005]. Outro exemplo são os roteamentos colaborativos, em que duas ou mais equipes são direcionadas para o mesmo ponto, seja para tarefas síncronas [Wex et al., 2013] ou assíncronas, eventualmente com restrições de precedência [Mankowska et al., 2014]. Nesses casos, a interdependência entre equipes no atendimento é uma informação conhecida já no momento da construção do roteamento, e as rotas são planejadas para que as visitas ocorram de acordo com as necessidades de colaboração.

Existem ainda situações reais em que a necessidade de colaboração só pode ser descoberta durante a execução do roteamento, pela percepção de uma situação emergencial quando da chegada a um consumidor. A necessidade de reforço, especialmente em atividade policial, é considerada com o posicionamento de helicópteros acionados em caso de emergência em van Urk et al. (2013), e inclusão da redução da distância entre distritos de patrulha como parte da função objetivo multicritério em problema de distritamento em Braekers et al. (2015), ou pelo conceito de cobertura de *backup* em Curtin et al. (2010), que se refere à existência de mais de uma equipe capaz de cobrir um determinado incidente

Aspectos relacionados a sincronização são discutidos em Drexl (2012). A sincronização leva a uma interdependência entre os veículos que impacta no projeto dos métodos de solução. Quando existe interdependência entre as rotas, o efeito de uma alteração em uma ou mais rotas usualmente demanda a reavaliação de toda a solução, e não somente a consideração dos arcos envolvidos. Este processo demanda um esforço computacional significativo e requer que os métodos de solução sejam especializados de forma a amenizar essa dificuldade.

Os estudos citados que tratam a questão do reforço apresentam como solução o cálculo do posicionamento de unidades de apoio ou definição de áreas de atuação com proximidade ou sobreposição. No melhor do conhecimento dos autores, o roteamento focado na disposição das rotas de modo a garantir a proximidade espacial e temporal dos veículos ainda não foi abordado na literatura especializada. É justamente nessa característica que reside distinção entre o PRV-MD e os demais problemas de roteamento.

O trabalho está organizado como segue. Na Seção 2, o problema é descrito de forma detalhada, sendo explicitada a métrica de dispersão usada para avaliar o posicionamento relativo dos veículos. Na Seção 3 é descrito o método de solução, com ênfase nos aspectos do Algoritmo

Genético e da Busca Local específicos para o novo problema. A seguir, na Seção 4 são apresentados os resultados computacionais, e na Seção 5 as conclusões.

2. Descrição do Problema

Considere um grafo conexo $G = (V, A)$ em que $V = \{0, 1, \dots, n\}$ representa o conjunto de $n + 1$ consumidores ou vértices e $A = \{a_{ij} : 0 \leq i \leq n, 0 \leq j \leq n\}$ é o conjunto de arestas que conectam pares de vértices. O vértice 0 representa a garagem de um conjunto $K = \{1, 2, \dots, m\}$ de veículos idênticos, enquanto os demais vértices representam consumidores a serem visitados. A cada vértice $i \in V$ é associado um tempo de serviço s_i e um par ordenado de coordenadas (x_i, y_i) . Um custo c_{ij} é associado a aresta $a_{ij} \in A$, que representa o tempo necessário para percorrer a distância euclidiana entre os vértices que a aresta conecta.

O PRV-MD pode ser definido como a busca por rotas para todos os m veículos, iniciadas e terminadas na garagem, que contemplem a visita a todos os vértices exatamente uma vez, minimizando uma métrica de dispersão.

A definição da métrica de dispersão exige a definição de equações de deslocamento dos veículos. Dados os vértices $i, j \in V$ e as respectivas coordenadas (x_i, y_i) e (x_j, y_j) , o deslocamento de um veículo $k \in K$ pela aresta a_{ij} com velocidade $\vec{v}_k = \langle \nu_{xk}, \nu_{yk} \rangle$ pode ser representado pela seguinte equação paramétrica, em que t representa um instante de tempo

$$X_k(t) = x_i + t \nu_{xk}, \quad Y_k(t) = y_i + t \nu_{yk}, \quad t \in [0, c_{ij}].$$

Observe que o vetor \vec{v}_k tem a mesma direção e sentido do vetor $\langle x_j - x_i, y_j - y_i \rangle$.

Considerando a posição instantânea de dois veículos k e k' em um instante t , a distância entre eles é dada por

$$d_{kk'} = \sqrt{[X_k(t) - X_{k'}(t)]^2 + [Y_k(t) - Y_{k'}(t)]^2}$$

A equação de deslocamento de um veículo precisa ser atualizada cada vez que este chega ou parte de um vértice. Cada período durante o qual as equações referentes a todos os veículos se mantêm inalteradas é denominada *fatia de tempo*. Dentro de uma fatia de tempo, o veículo só pode estar se movendo entre dois vértices ou parado servindo um vértice, já que a alternância entre esses estados representa o fechamento de uma fatia.

Para determinar o início e o fim de cada fatia, é considerada a sequência de todos os instantes de tempos em que ocorrem chegadas e partidas de veículos dispostos em ordem crescente, conforme ilustrado na Figura 1. Na figura, as linhas horizontais representam deslocamentos, as caixas representam visitas a vértices, e as linhas verticais demarcam fatias de tempo. A partida do vértice 5, por exemplo, é o 12º momento em que um veículo muda da velocidade, e é seguido na sequência pela chegada de outro veículo ao vértice 8.

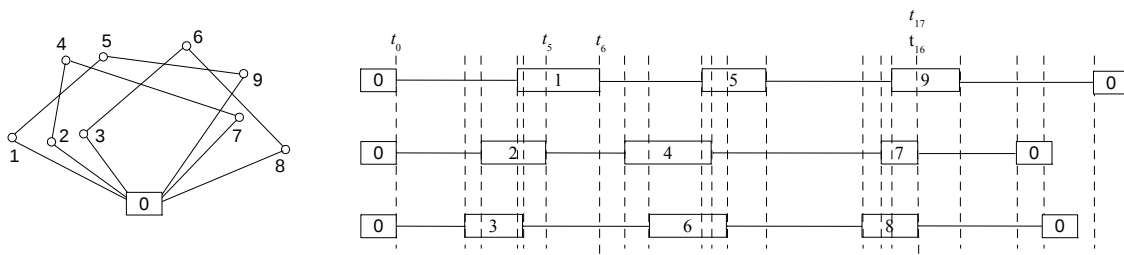


Figura 1: Conjunto de rotas as correspondentes fatias de tempo.

Para n vértices e m veículos, a sequência contém $2n + m + 1$ tempos, em que 1 indica a partida simultânea da base, $2n$ representa as chegadas e partidas nos demais vértices, e m representa as chegadas dos veículos na base, encerrando as rotas.

Seja $\mathcal{S} = \{t_\tau : \tau = 0, \dots, 2n + m\}$ a referida sequência, existem $2n + m$ fatias de tempo, já que cada fatia é delimitada por dois instantes consecutivos $[t_{\tau-1}, t_\tau]$ na sequência \mathcal{S} . Portanto, a duração de uma fatia τ é dada por $T_\tau = t_\tau - t_{\tau-1}$. A duração de uma fatia pode ser 0, desde que dois instantes consecutivos de \mathcal{S} sejam idênticos, como os instantes t_{16} e t_{17} na Figura 1.

A avaliação da dispersão total de uma solução do PRV-MD se dá a partir da verificação do distanciamento entre os veículos, tomados dois a dois, durante cada fatia de tempo. Considera-se distanciamento como sendo a média aritmética das distâncias dos veículos no início e no final de uma fatia. Para um par de veículo k e k' , dadas as distâncias no início e no final da fatia τ , respectivamente $d_{kk'}(t_{\tau-1})$ e $d_{kk'}(t_\tau)$, o distanciamento é dado por

$$D_{kk'\tau} = \frac{d_{kk'}(t_{\tau-1}) + d_{kk'}(t_\tau)}{2} \quad (1)$$

Observando a equação (1), percebe-se que para calcular as distâncias entre k e k' no início e no final da fatia é necessário conhecer suas posições nesses instantes. Note-se que a posição inicial da primeira fatia é a posição do vértice 0, uma vez que todos os veículos partem juntos da base. Já a posição inicial de cada veículo nas demais fatias pode ser obtida de forma recursiva, pois é idêntica à posição final na fatia anterior. A posição de cada veículo ao final de uma fatia pode ser obtida a partir da sua posição inicial, da duração da fatia e do vetor de velocidade, que tem magnitude idêntica para todos os veículos quando em movimento, pois a frota é homogênea, e direção e sentido dados pela aresta que o veículo deve percorrer.

O valor de dispersão atribuído à fatia τ é o maior valor de dispersão encontrado entre pares de veículos, ou seja,

$$C_\tau = \max_{\substack{k=1, \dots, m \\ k'=k+1, \dots, m}} (D_{kk'\tau}) \quad (2)$$

A dispersão total é dada pela soma das dispersões das fatias, ponderadas pela duração de cada fatia. A ponderação é necessária uma vez que a dispersão de fatias mais longas é mais representativa do posicionamento relativo das equipes durante todo o roteamento do que a dispersão de fatias curtas. Dessa forma, a função objetivo não linear para a dispersão total é dada por

$$\min \sum_{\tau=1}^{2n+m} T_\tau C_\tau \quad (3)$$

Proposição 1. *O PRV-MD é NP-Hard.*

Demonstração. É demonstrado que uma instância do Problema de Caixeiro Viajante Euclidiano (PCVE) pode ser reduzida a uma instância do PRV-MD.

Inicialmente, considere uma instância do PCVE dada pelo grafo $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, em que $\mathcal{V} = \{1, \dots, n\}$ é o conjunto de vértices, $\mathcal{A} = \{\alpha_{ij} : i, j \in \mathcal{V}, i \neq j\}$ é o conjunto de arestas e cada vértice $i \in \mathcal{V}$ tem coordenadas (x_i, y_i) .

O grafo $\mathcal{G}' = (\mathcal{V}', \mathcal{A}')$ é criado a partir de \mathcal{G} , em que $\mathcal{V}' = \{i + n : i \in \mathcal{V}\}$ e cada vértice tem coordenadas $(x_i + \delta, y_i)$, ou seja, os vértices de \mathcal{V}' são posicionados no plano deslocados δ unidades no eixo x em relação aos vértices em \mathcal{V} . Cada nó em $i + n \in \mathcal{V}'$ é denominado cópia de um nó original $i \in \mathcal{V}$. Esta denominação também é válida para as arestas em \mathcal{A}' e \mathcal{A} .

Considere também dois vértices $2n+1$ e $2n+2$ posicionados nas coordenadas dos vértices 1 e $1+n$, respectivamente, e o nó 0 posicionado no ponto médio entre os vértices 1 e $1+n$ definido como garagem.

O grafo referente a uma instância do PRV-MD é dado por $G = (V, A)$, em que $V = \mathcal{V} \cup \mathcal{V}' \cup \{0, 2n + 1, 2n + 2\}$ e $A = \mathcal{A} \cup \mathcal{A}' \cup \{a_{0,1}, a_{0,n+1}, a_{2n+1,0}, a_{2n+2,0}\} \cup \{a_{i,2n+1} : i \in \mathcal{V}\} \cup \{a_{i,2n+2} : i \in \mathcal{V}'\}$. O tempo de serviço é igual a zero para todos os nós ($s_i = 0, \forall i \in V$) e o conjunto de veículos é definido como $K = \{1, 2\}$. O arco que incide na garagem são definidos de tal forma que obrigatoriamente os veículos iniciam as rotas pelos nós 1 e $n + 1$ e retornam para a garagem após visitarem os nós $2n + 1$ e $2n + 2$. A Figura 2 ilustra uma instância do PRV-MD.

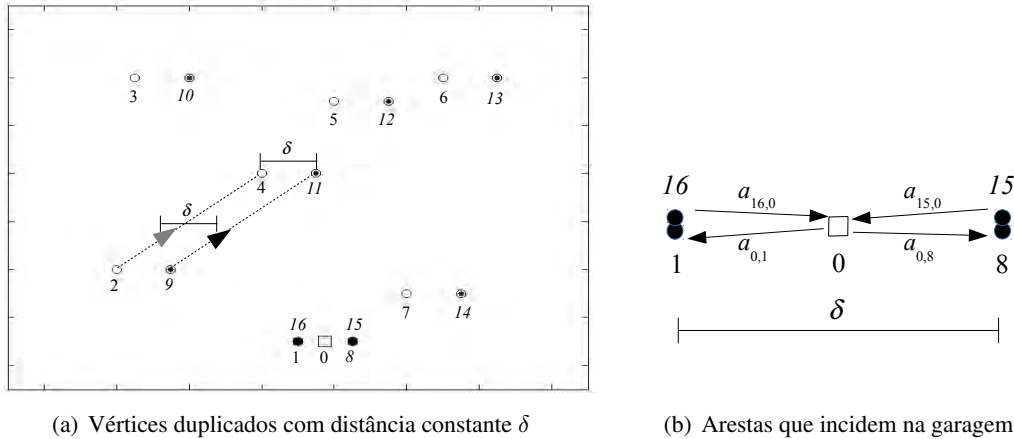


Figura 2: Ilustração de uma instância PRV-MD criada a partir de uma instância do PCVE com sete vértices.

Um valor de δ igual a zero faz com que os vértices dos grafos \mathcal{G} e \mathcal{G}' coincidam no plano. Nesta configuração os veículos se deslocam em conjunto quando a solução é ótima, ou seja, no instante que um veículo visita um nó original o outro veículo visita o nó cópia. Nesta configuração a solução ótima é trivial, pois qualquer ordem de visita aos vértices resulta em dispersão igual a zero.

Observe que se os veículos não se deslocarem em conjunto, então obrigatoriamente irão se afastar a partir de dois pontos coincidentes i e $i + n$. Considere que o veículo em i se desloca para o vértice l e o veículo em $i + n$ se desloca para o vértice $j + n$ ($j \neq l$), o que acarreta em uma dispersão não nula na primeira fatia após a separação que pode ser calculada conforme descrito na Seção 2. Um limitante inferior LB para a dispersão total pode ser definido considerando a menor dispersão de todas as combinações possíveis de $i, l, i + n$ e j .

Se o δ for definido como $LB / (n \cdot a_{max})$, em que a_{max} corresponde a maior aresta em A e $n \cdot a_{max}$ é um limitante superior para o comprimento da rota do PCVE, a dispersão de uma solução na qual os veículos se deslocam em conjunto nunca será superior a LB . No entanto, a solução ótima não é mais trivial. Uma vez que a separação dos veículos, embora constante, passa a ser não nula o comprimento da rota é determinante para definir o valor da dispersão total.

Dado que a dispersão das fatias passa a ser constante quando os veículos se deslocam em conjunto, a equação (3) assume a forma $\hat{C} \sum_{\tau}^{2n+m} T_{\tau}$, em que \hat{C} é a dispersão local e $\sum_{\tau}^{2n+m} T_{\tau}$ corresponde ao comprimento da rota de cada veículo. Desta forma, a única maneira de minimizar a função objetivo é encontrar a rota com menor tempo total, o que corresponde a solução ótima do PCVE a menos do tempo de deslocamento que corresponde as arestas que têm como origem ou destino a garagem.

□

3. Algoritmo Genético com Busca Local

Na busca por soluções para o problema, foi implementado um Algoritmo Genético com Busca Local (LSGA, *Local Search Genetic Algorithm*), cujo pseudocódigo está representado no

Algoritmo 1. A população $pop = \{I_1, I_2, \dots, I_{ps}\}$ tem tamanho fixo ps , e, conforme a linha 2 do código, inicialmente contém indivíduos gerados através do método construtivo guloso, que está descrito em detalhes na Seção 3.1. Após a geração, é feita uma busca local sobre uma parcela da população inicial, com processo que será descrito nas seções 3.3 e 3.4. A partir da população resultante da busca local inicia-se o laço de avaliação-seleção-reprodução, encerrado quando o tempo de total de execução atingir o tempo limite $tempoMax$, conforme a linha 5.

Algorithm 1 Pseudocódigo LSGA.

```

inicializaParametros( $es, tempoMax, \beta, ps, cp$ )
1:  $novaPop \leftarrow \emptyset$ 
2:  $pop \leftarrow PopulacaoInicial()$ 
3: BuscaLocal( $pop$ )
4:  $t \leftarrow 1$ 
5: Enquanto tempo de CPU <  $tempoMax$  faça
6:   AvaliaPop( $pop$ )
7:    $elite \leftarrow SeleccionaElite(es, pop)$ 
8:    $novaPop \leftarrow novaPop \cup elite$ 
9:   Equanto  $|novaPop| < ps$  faça
10:     $individuo1 \leftarrow Roleta(pop)$ 
11:    Se NumAleat(  $\in [0, 1)$  ) <  $cp$  E  $|novaPop| < ps - 1$  então
12:       $individuo2 \leftarrow Roleta(pop)$ 
13:       $filhos \leftarrow Cruzamento(individuo1, individuo2)$ 
14:       $novaPop \leftarrow novaPop \cup filhos$ 
15:    Senão
16:       $individuo2 \leftarrow Mutacao(individuo1)$ 
17:       $novaPop \leftarrow novaPop \cup \{individuo2\}$ 
18:    FimSe
19:    FimEnquanto
20:     $pop \leftarrow novaPop$ 
21:     $novaPop \leftarrow \emptyset$ 
22:    Se  $t \bmod 1000 = 1$  então
23:      BuscaLocal( $pop$ )
24:    FimSe
25:     $t \leftarrow t + 1$ 
26: FimEnquanto
27: Retorna melhor individuo em  $pop$ 

```

A avaliação dos membros da população corrente está representada no código pela linha 6. Por se tratar de problema de minimização, o valor de função objetivo não pode ser utilizado como *fitness* de um indivíduo. Desta forma, o *fitness* de um indivíduo l é definido como o quadrado do quociente entre o maior valor de função objetivo da população e o valor de função objetivo do indivíduo, $fo(I_l)$.

$$fitness(I_l) = \left(\frac{\max_{h=1, \dots, ps} fo(I_h)}{fo(I_l)} \right)^2$$

A construção da nova população ($novaPop$) é iniciada pela cópia direta dos es indivíduos de melhor avaliação, em um processo de elitismo (linhas 7 a 9). O processo continua com a seleção de indivíduos para reprodução, através de um processo de roleta. Selecionado um indivíduo (linha 10), é definida a operação a ser feita sobre ele, de acordo com um parâmetro cp que estabelece a

probabilidade de cruzamento ou de mutação ($1 - cp$). Se for definida a execução de cruzamento, um segundo indivíduo é selecionado (linha 12), com a restrição de que seja diferente do escolhido anteriormente. O operador de cruzamento a que se refere a linha 13 é descrito em detalhes na Seção 3.2. Os dois indivíduos resultantes do cruzamento são então inseridos na nova população (linha 14). O processo de mutação é executado na linha 16 e resulta em um indivíduo que é colocado na nova população (linha 17).

O procedimento de seleção de indivíduos e aplicação dos operadores genético é repetida até que a nova população esteja completa, conforme laço de repetição da linha 9. Após a população é atualizada, conforme linha 20, e verifica-se se a geração corrente deve ser submetida à busca local (linhas 22 a 24). Ao final o melhor indivíduo da última geração é retornado.

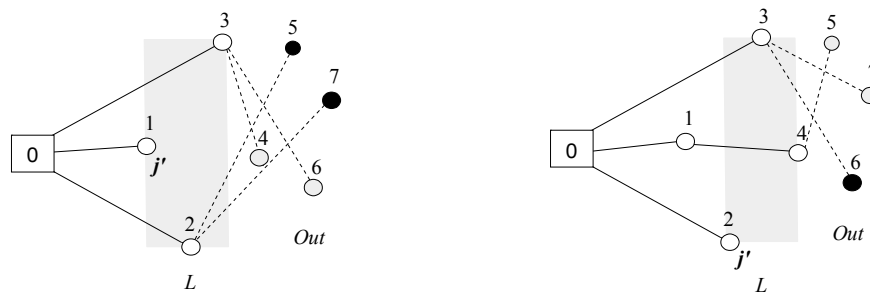
3.1. Construção da População Inicial

A população inicial é gerada através de método construtivo com aleatoriedade. Inicialmente, é definido um conjunto $Nn(i)$ associado a cada vértice $i \in V$ que contém os $numNn$ vértices de $V \setminus \{0, i\}$ mais próximos (vizinhos) de i . É necessário também definir o conjunto Out , que contém os vértices ainda não incluídos em rotas, e o conjunto L , que contém os m últimos vértices inseridos nas rotas parciais.

Inicialmente, um vértice i é selecionado de forma aleatória e posicionado como vértice inicial da rota do primeiro veículo. A seguir, os vértices iniciais das demais rotas são selecionados dentre os vértices de $Nn(i) \cap Out$, ou seja, próximos a i e ainda não incluídos em rotas. Cada vértice selecionado é removido de Out e inserido em L .

A partir de então, um processo iterativo é iniciado para designação dos vértices em Out às rotas. A cada iteração, é identificado o vértice j' com menor tempo de partida entre os vértices em L , que indica a rota parcial mais curta até então. Cada vértice em Out é avaliado para ser inserido na rota de j' , sendo o valor de avaliação a máxima distância entre o vértice em Out e os demais vértices em L .

A Figura 2 ilustra um caso em que as rotas parciais são compostas por um vértice cada. O vértice j' em L define a rota mais curta até então. As maiores distâncias de cada vértice de Out aos outros vértices de L são identificadas por linhas pontilhadas.



(a) j' corresponde ao vértice 1

(b) o vértice 4 é inserido na rota mais curta

Figura 3: Ilustração de uma iteração do método construtivo guloso

O uso da máxima distância para avaliar os vértices de Out decorre do fato de que é o máximo distanciamento entre um par de veículos que determina a avaliação de uma fatia. Uma lista de candidatos é então construída com os $nLRC$ vértices de Out com menor valor de avaliação, já que se quer aproximar os mais distantes, e dentre estes é escolhido aleatoriamente o que entrará na rota mais curta. O valor $nLRC$ é gerado aleatoriamente entre 2 e $nNeigh$, sempre que se inicia a construção de um roteamento. No exemplo ilustrado na Figura 3(a), é considerado um $nLRC$ igual

a 2, e os vértices candidatos são os destacados no conjunto *Out*. O processo de escolha de vértices para a rota mais curta é repetido até que o indivíduo esteja completo.

3.2. Formato do cromossomo e operador genéticos

Representações de soluções de roteamento através de *arrays* têm a característica de privilegiar as informações sobre a disposição espacial (proximidade) dos vértices em suas rotas e favorecer a transferência dessas informações para os descendentes. Para o PRV-MD, o sincronismo das rotas fica mais evidente e é melhor propagado pelas operações genéticas se cada rota for representada separadamente, e os pontos de cruzamento e mutação forem definidos pelo tempo de andamento do roteamento.

Foi adotada uma representação em que cada indivíduo é composto por sequências encadeadas, em que cada elemento possui informações sobre o vértice representado, os tempos de chegada e partida deste vértice na solução, e referências aos vértices sucessor e antecessor. Os operadores genéticos são aplicados diretamente sobre a sequência encadeada, o que permite afirmar que não existe a representação de uma solução através de genótipo que exija codificação e decodificação.

O cruzamento combina indivíduos pais, p_1 e p_2 , na criação de indivíduos filhos, o_1 e o_2 . Trata-se de um cruzamento de dois pontos, definidos por dois instantes de tempo t_1 e t_2 que dividem as rotas em 3 segmentos: o inicial, contendo vértices com tempo de chegada anterior ou igual a t_1 ; o central, contendo os vértices com tempo de chegada posterior a t_1 e anterior ou igual a t_2 ; e o final, contendo os vértices com tempo de chegada posterior a t_2 .

As rotas do filho o_1 são formadas pelos segmentos inicial e final das rotas do pai p_1 e segmento central das rotas do pai p_2 . A combinação das rotas de p_2 com as rotas do filho o_1 oriundas de p_1 é definida de forma aleatória a cada novo cruzamento. O procedimento para o filho o_2 é semelhante, necessitando apenas alterar os índices que definem os pais.

É importante notar que os segmentos de pais diferentes podem gerar vértices duplicados no filho. Ocorrências assim são contornadas ignorando a inclusão do vértice pela segunda vez, e colocando-o em uma lista para inserção ao final do roteamento do outro filho, já que a ocorrência de vértices duplicados em um indica ausência do mesmo vértice no outro. A inclusão dos vértices é feita através de procedimento guloso.

Por fim, optou-se por possibilitar a inversão das rotas envolvidas antes de aplicar o operador de cruzamento. Selecionados dois indivíduos para a recombinação, existe uma probabilidade de 30% de que o operador seja aplicado após a inversão no sentido de todas as rotas dos pais.

A mutação é aplicado considerando quatro possíveis procedimentos, todos com igual probabilidade. As três primeiras formas de mutação têm como ponto de partida a escolha aleatória de um instante de tempo entre 0 e o tempo de chegada na base da rota mais curta. Em cada rota é selecionado o vértice em atendimento no momento definido ou, em caso de deslocamento, o último vértice visitado. Na primeira forma, cada vértice selecionado troca de posição com um de seus vizinhos do respectivo conjunto Nn . Na segunda forma, cada vértice selecionado é recolocado em posição aleatória de outra rota. E na terceira forma, os vértices são removidos de suas rotas, e outro instante de tempo é definido aleatoriamente para a reinserção em rotas também definidas aleatoriamente. Na quarta forma de mutação, são selecionados m vértices, um por rota, para serem reposicionados, também um por rota, em posições escolhidas também de forma aleatória. A diferença neste caso é que não há qualquer relação temporal entre os vértices envolvidos.

3.3. Estrutura da Vizinhança

De acordo com a métrica de dispersão utilizada, a avaliação de uma fatia é dada pelo par de rotas mais afastadas considerando as médias das posições iniciais e finais. Portanto, a redução na distância entre as rotas mais afastadas reduz a avaliação da fatia, com impacto positivo na avaliação global. A vizinhança implementada utiliza informação sobre o par de rotas determinante do valor de cada fatia de tempo.

Dada uma fatia e encontradas as rotas mais afastadas, é verificado se as rotas se aproximam ou distanciam, de modo a determinar se o par de vértices que mais influencia no afastamento

é o que determina as posições iniciais ou finais dos veículos. Se os veículos mantêm distância constante, é adotado o par de vértices determinante da posição final.

É possível a ocorrência do mesmo par de vértices como determinante da dispersão em diversas fatias subsequentes, já que duas equipes distantes podem estar realizando longos deslocamentos. A ocorrência de tais repetições é observada, para evitar reavaliações de movimentos. O de conjunto de pares de vértices responsáveis pela avaliação de uma fatia tempo orientam a busca local.

A Figura 4 a estrutura de vizinhança. Selecionado do conjunto um par formado pelos vértices i e j , inicialmente o vértice i é mantido em sua posição e os vértices em $Nn(i)$ são envolvidos em movimentos que causam alterações na rota de j . O primeiro movimento é o movimento de troca, cada um dos vértices $i' \in Nn(i)$ é trocado de lugar no roteamento com o vértice j . O segundo movimento é caracterizado pela retirada de cada vértice $i' \in Nn(i)$ de sua rota e a inserção na posição imediatamente anterior a j . O terceiro movimento se assemelha ao segundo, mas a inserção de cada vértice i' é feita em posição posterior a j .

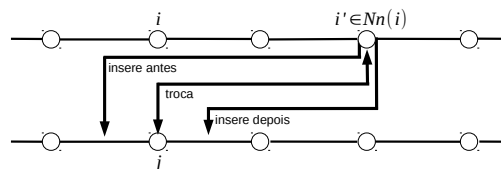


Figura 4: Ilustração da estrutura de vizinhança.

Cada um dos movimentos tem objetivo de melhorar a avaliação da fatia através da colocação de vértices de $Nn(i)$ na rota de j , em posições que aumentem a probabilidade de execução concomitante com i . É claro ainda que os mesmos movimentos devem ser aplicados mantendo a rota do vértice j inalterada, fazendo as alterações em que todos os possíveis $j' \in Nn(j)$ são colocados na rota do vértice i .

Como a busca local utiliza uma estratégia do tipo primeira melhora (*first-improvement*) nem todos os possíveis movimentos são avaliados. Devido a isso, todas as decisões que impactam na ordem de avaliação dos movimentos são tomadas de forma aleatória, como a ordem em que os pares de vértices i e j são retirados do conjunto de pares de vértices a serem tratados, qual deles terá seu respectivo conjunto Nn explorado primeiro, a ordem em que cada conjunto $Nn(i)$ é percorrido, como também a própria ordem em que os movimentos de troca e inserção são aplicados.

3.4. Busca Local

A busca local é aplicada sobre as duas primeiras populações, a população inicial e a primeira gerada por operadores genéticos, e então a cada β gerações. A busca considera um percentual do tamanho total da população, composto pelos melhores indivíduos. O uso de um subconjunto da população foi adotado para equilibrar o *trade-off* entre o número total gerações em um determinado tempo e a diversidade populacional. A busca local sobre toda a população requer populações menores, sob pena do LSGA avaliar poucas gerações. Por outro lado, uma população pequena reduz diversidade e favorece a convergência prematura.

Dada uma solução corrente x , a busca local pode ser resumida como a sucessiva aplicação dos movimentos da vizinhança N descrita na Seção 3.3 para localizar outra solução $x' \in N(x)$ de melhor qualidade, que passa então a ser a solução corrente. Como critério de parada é adotado o número máximo de iterações $nIter$. Na prática, o valor é definido de forma a não influenciar a busca local executada nos melhores indivíduos da população.

Conforme já descrito anteriormente, uma solução é composta por $2n + m$ fatias de tempo, e a avaliação da função objetivo exige que se realize o cálculo apresentado na equação (2) para

cada par de veículos em cada fatia. Esta característica inerente ao problema representa um alto custo computacional para a avaliação da função objetivo de uma solução. Para reduzir o custo é utilizada uma estratégia que evita a repetição da varredura da vizinhança de indivíduos para os quais a varredura já foi realizada e resultou inócua. Se uma solução x não possui vizinho x' com função de avaliação melhor, ou seja, $f_o(x) \leq f_o(x')$ para todo $x' \in N(x)$, a solução x é incluída em uma lista LN contendo as soluções para as quais a vizinhança já é sabidamente infrutífera.

O Algoritmo 2 representa o funcionamento do método de busca local, que é aplicado aos indivíduos selecionados. A variável que indica a iteração (it) é inicializada com zero na linha 1. Inicia-se a seguir um processo iterativo que começa na linha 2, com a verificação da presença ou não de x na lista das soluções já exploradas na vizinhança corrente. Se x for encontrado em LN , não há vizinho melhor e a busca é encerrada. Caso contrário, é feita a busca por um vizinho com avaliação melhor que x , utilizando a vizinhança descrita na Seção 3.3. Se um vizinho com tal característica não for encontrado, x é adicionado à lista LN e a busca encerrada. Caso contrário, o vizinho é adotado como solução corrente. O encerramento da busca por qualquer situação implica no retorno da solução corrente x .

Algorithm 2 Pseudocódigo Busca Local.

```

1:  $it \leftarrow 0$ 
2: Enquanto  $it < nIter$  faça
3:   Se  $x \in LN$  então
4:     Retorna  $x$ 
5:   Senão
6:     Gera  $x' \in N(x)$ 
7:     Se  $f_o(x) \leq f_o(x')$  então
8:        $LN \leftarrow LN \cup x$ 
9:     Retorna  $x$ 
10:   Senão
11:      $x \leftarrow x'$ 
12:   FimSe
13: FimSe
14:  $it \leftarrow it + 1$ 
15: FimEnquanto
16: Retorna  $x$ 

```

4. Resultados Computacionais

A partir das instâncias de *benchmark* para o Problema de Múltiplos Caixeiros Viajantes de 51, 52 e 76 vértices propostas em Necula et al. (2015), foram criadas 24 instâncias pela adição de tempos de serviço. Cada uma das 12 instâncias originais (4 por número de vértices) recebeu duas configurações de tempos de serviço: uma uniforme, de 10 minutos para cada vértice (identificada por "U"), e outra aleatória, com os valores gerados entre 1 e 10 minutos (identificada por "A"). O conjunto de instâncias pode ser obtido a partir de solicitação aos autores.

A parametrização do Algoritmo Genético foi feita de forma empírica. Adotou-se uma população de tamanho $ps = 1500$ indivíduos, um tempo máximo de execução dado em minutos por $tempoMax = \lceil n \cdot m / 15 \rceil$, elitismo usando 5% da população e probabilidade de cruzamento de 95%. O tamanho dos conjuntos Nn é definido como o maior valor entre o dobro do número de rotas ($2m$) e 10% do tamanho da população ($n/10$).

A busca local tem sua duração balizada por $nIter$, que foi fixado em 50. O parâmetro β , que define o número de gerações entre duas buscas locais sucessivas, foi fixado em 1000, e a busca foi executada sobre os 33% melhores indivíduos da população.

Os algoritmos foram desenvolvidos na linguagem de programação Java e os experimentos realizados em uma máquina dual chip equipada com dois processadores Intel E5-2687 2.7 GHz com 64 GB de RAM. Não foi implementada qualquer forma de paralelismo, sendo cada execução realizada utilizando apenas um dos processadores.

A Tabela 1 resume os resultados computacionais. A primeira coluna identifica nome original da instância TSPLIB. O número de vértices é informado junto ao nome, e as demais informações relevantes para identificar a instância são apresentadas nas duas colunas seguintes, que indicam o número de veículos e o tipo de tempo de serviço. Cada linha da tabela compreende 10 execuções do LSGA, e os resultados são resumidos pela apresentação do valor de função objetivo da melhor solução encontrada, do valor médio, do desvio padrão e do desvio padrão relativo. Os baixos valores para este último indicador, que variam entre 0,91% e 3,62%, indicam a consistência do método de solução.

Tabela 1: Resultados de 10 execuções do LSGA para um conjunto de 24 instâncias.

Instância	m	Serviço	Melhor resultado	Média	Desvio padrão (DP)	DP relativo
eil51	2	U	3333,7	3570,5	111,5	3,02
eil51	2	A	2395,1	2512,1	91,0	1,82
eil51	3	U	3796,2	3915,3	67,2	3,23
eil51	3	A	3082,8	3152,9	49,0	2,32
eil51	5	U	4306,4	4378,5	42,5	1,44
eil51	5	A	3380,7	3463,8	73,2	1,18
eil51	7	U	4516,5	4662,3	58,6	1,14
eil51	7	A	3978,7	4046,5	55,2	0,91
berlin52	2	U	621800,7	674312,6	20340,7	3,12
berlin52	2	A	670396,4	684361,2	12479,2	3,62
berlin52	3	U	827535,3	856040,8	27669,3	1,72
berlin52	3	A	801942,5	830325,2	19234,3	1,55
berlin52	5	U	1091086,1	1107409,9	15893,1	0,97
berlin52	5	A	1080126,7	1089450,4	12864,6	2,11
berlin52	7	U	1335659,7	1354078,2	15392,9	1,26
berlin52	7	A	1326437,9	1342134,1	12197,7	1,37
eil76	2	U	4430,9	4568,5	121,5	2,66
eil76	2	A	3380,1	3508,0	125,8	3,59
eil76	3	U	4882,1	5037,6	148,1	2,94
eil76	3	A	3629,9	3786,1	72,3	1,91
eil76	5	U	5050,0	5183,3	96,6	1,86
eil76	5	A	4028,3	4198,2	133,3	3,18
eil76	7	U	4935,1	5156,5	166,9	3,24
eil76	7	A	4211,1	4454,8	109,5	2,46

A Figura 5 ilustra o roteamento representado na melhor solução encontrada para a instância "eil51" com 3 veículos e tempo uniforme de serviço. Além da proximidade espacial, a proximidade temporal também é evidente, com os veículos mantendo-se agrupados durante o percurso das rotas.

5. Conclusões

Neste trabalho é apresentado o PRV-MD, uma variação não linear de problemas de roteamento no qual as rotas são avaliadas em função do posicionamento relativo dos veículos. O objetivo consiste em minimizar a dispersão dos veículos de modo a propiciar suporte mútuo e atuação colaborativa aumentando a segurança da frota.

Como método de resolução foi proposto um Algoritmo Genético com Busca Local com estruturas especializadas para o PRV-MD. O avaliação das soluções obtidas demonstrou que a mé-

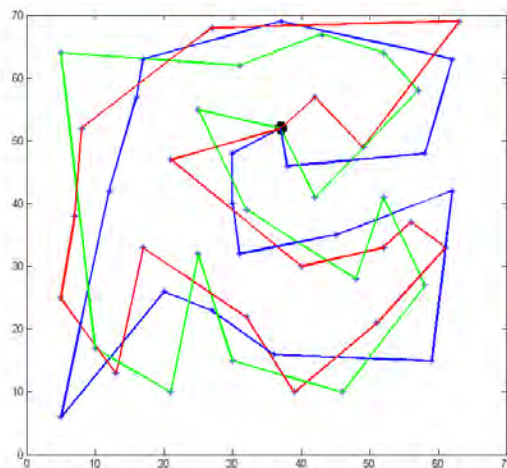


Figura 5: Representação da solução da instância eil51 com 3 veículos.

trica de dispersão proporciona a obtenção de rotas que apresentam as características de proximidade temporal e espacial desejadas.

Referências

- Braekers, K., Ramaekers, K. and Van Nieuwenhuysse, I.** (2015). The vehicle routing problem: State of the art classification and review, *Computers & Industrial Engineering*, Publicado online em 21 December 2015, <http://www.sciencedirect.com/science/article/pii/S0360835215004775>.
- Camacho-Collados, M., Liberatore, F. and Angulo, J.M.** (2015). A multi-criteria Police Districting Problem for the efficient and effective design of patrol sector, *European Journal of Operational Research*, 246, 674-684.
- Curtin, K. M. and Hayslett-McCall, K. and Qiu, F.** (2010). Determining Optimal Police Patrol Areas with Maximal Covering and Backup Covering Location Models, *Networks and Spatial Economics*, 10, 125-145.
- Drexel, M.** (2012). Synchronization in vehicle routing - A survey of vrps with multiple synchronization constraints, *Transportation Science*, 46(3), 297-316.
- Mankowska, D. S., Meisel, F. and Bierwirth, C.** (2014). The home health care routing and scheduling problem with interdependent services. *Health Care Management Science*, 17, 15-30.
- Necula, R., Breaban, M., Raschip, M.** (2015). Performance Evaluation of Ant Colony Systems for the Single-Depot Multiple Traveling Salesman Problem, *10th International Conference on Hybrid Artificial Intelligence Systems*, 22-24 June, Bilbao, Spain, 9121, p257-268.
- Talarico, L., Sörensen, K. and Springaël, J.** (2015). The k-dissimilar vehicle routing problem. *European Journal of Operational Research*, 244(1), 129-140.
- Thyagarajan, K., Batta, R., Karwan, M. H. and Szczerba, R. J.** (2005). Planning dissimilar paths for military units. *Military Operations Research*, 10(1), 25-42.
- van Urk, R., Mes, M. R.K. and Hans, E. W.** (2013). Anticipatory routing of police helicopters, *Expert Systems with Applications*, 40(17), 6938-6947.
- Wex, F., Schryen, G. and Neumann, D.** (2013). Assignments of Collaborative Rescue Units during Emergency Response. *International Journal of Information Systems for Crisis Response and Management*, 5(4), 63-80.