

Algoritmo genético para o problema de rotas de cobertura multiveículo

Washington Alves de Oliveira

8 de Julho de 2009

Abstract

This work consider an adaptation of the model for the *multi-vehicle covering tour problem* (*m-CTP*). In this model a set of locations must be visited in the planned routes, whereas another subset must be close enough to the locations visited. The planning is restricted to the availability of vehicles and strives to achieve balanced and short routes. It constitutes an integer program whose size and complexity makes the use of an exact method impractical. Suboptimal solutions are obtained with application of two proposed genetic algorithm (GA), that to construct short routes.

Keywords: Integer and combinatorial programming, genetic algorithm, heuristics, vehicle routing and set covering.

Resumo

Este trabalho considera uma adaptação do modelo para o *Problema de Rotas de Cobertura multi-veículo* (*m-PRC*). Neste modelo alguns pontos geográficos devem ser obrigatoriamente visitados em rotas traçadas, enquanto outros devem ficar suficientemente próximos dos pontos visitados. O planejamento deve levar em conta os recursos disponíveis, normalmente o número de veículos, visar agilidade para os veículos e uma distribuição equânime nas rotas. Este modelo consiste em um programa linear inteiro cujo tamanho e complexidade torna inviável a aplicação de métodos exatos para sua solução. Soluções subótimas são obtidas aplicando-se duas propostas de algoritmo genético (AG), os quais geram rotas de tamanho menor possível.

Palavras-chave: Programação inteira e combinatória, algoritmo genético, heurísticas, roteamento de veículos e cobertura de conjuntos.

1 Introdução

O objetivo desta trabalho é propor uma forma de construção de rotas dentro de uma região mapeada, a fim de encontrar uma boa solução para a difícil tarefa de distribuição adequada de viaturas. Tais rotas poderiam ser usadas como rotas de patrulhamento urbano preventivo por alguns departamentos responsáveis em promover segurança, como as Polícias, as Guardas Civis Municipais e empresas de segurança privada. Temos como proposta de trabalho construir tais rotas usando um modelo de programação linear inteira, que será chamado de Problema de Rotas de Cobertura multiveículo (m -PRC), originalmente conhecido como *multi-vehicle Covering Tour Problem* (m -CTP). Contudo algumas restrições do problema original serão ignoradas e outras acrescentadas de acordo com a nossa proposta.

Este modelo é um problema combinatório classificado como NP-difícil, cuja solução, através de algoritmos exatos de otimização demandaria grande esforço computacional. Para que o tempo computacional na obtenção de uma boa solução seja satisfatório, vamos considerar na busca desta solução dois modelos de algoritmo genético juntamente com uma combinação de técnicas heurísticas. Para validar a nossa proposta de construção de rotas, foram gerados aleatoriamente alguns exemplares para m -PRC e a partir de dados reais, aplicamos testes na construção de rotas de patrulhamento preventivo para a Guarda Civil Municipal da cidade de Vinhedo-SP.

Na Seção 2 descrevemos o m -PRC para rotas preventivas e a formulação proposta neste trabalho. Na Seção 3 apresentamos com detalhes as heurísticas que são usadas hibridamente com os algoritmos genéticos. Na Seção 4 apresentamos com detalhes os dois algoritmos genéticos propostos neste trabalho. Os resultados computacionais são apresentados na Seção 5 e as conclusões na Seção 6.

2 O m -PRC para rotas preventivas

A formulação para o m -PRC se baseia na apresentada por Hachicha et al. [9], porém com algumas alterações. É definido sobre um grafo $G = (V \cup W, E)$ não direcionado, onde $V \cup W$ compõem o conjunto de vértices, $V = \{v_0, \dots, v_n\}$, $W = \{v_{n+1}, \dots, v_l\}$ e $E = \{(v_i, v_j) : v_i, v_j \in V \cup W, i < j\}$ o conjunto de arestas, ou seja, o subgrafo induzido por E é um grafo completo cujo conjunto de nós é V . O vértice v_0 é a base, onde estão disponíveis m veículos idênticos. O conjunto V é composto dos vértices que **podem** ser visitados e inclui um subconjunto T de vertices que **devem** ser visitados ($v_0 \in T$); W é um conjunto de vértices que **devem** ser cobertos coletivamente pelos m veículos. Uma matriz de distância ou tempo de viagem $C = (c_{ij})$ satisfazendo a desigualdade triangular é definida em E . Interpretamos c_{ji} como c_{ij} se $j > i$. O m -PRC consiste em atribuir um

conjunto de m rotas de veículo de tamanho total mínimo, satisfazendo um certo conjunto de restrições:

1. Existem exatamente m rotas de veículo, e cada uma delas inicia e termina na base;
2. Cada vértice de V pertence a no máximo uma rota, enquanto cada vértice de T , com exceção da base, pertence a exatamente uma rota;
3. Cada vértice de W deve ser coberto por uma rota, no sentido de que deva situar-se dentro de uma distância c de um vértice de V visitado (assumimos que v_0 não cobre todos os vértices de W);
4. O módulo da diferença entre o número de vértices de diferentes rotas não pode exceder um determinado valor r .

Vemos na Figura 1 um exemplo do grafo $G = (V \cup W, E)$ não direcionado. Note que este grafo é completo apenas no conjunto V , e apesar da matriz $C = (c_{ij})$ estar definida para todos os pares de nós em $V \cup W$, os vértices de W não são visitados.

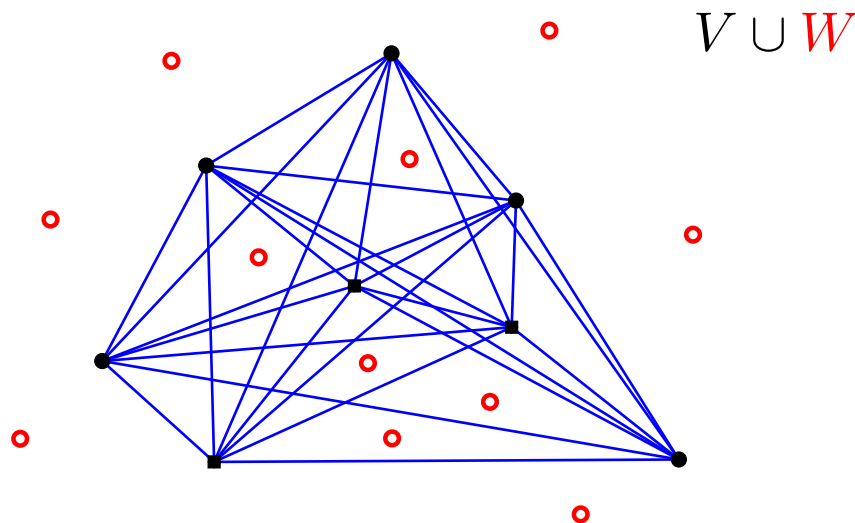


Figura 1: Exemplo de grafo não direcionado para o conjunto $V \cup W$

Aplicações para o m -PRC surgem em algumas situações. Um exemplo é o problema de localização de caixas de correios (Labbé e Laporte [11]), onde deve-se alocar simultaneamente caixas de correios e uma coleção de rotas ótimas devem ser construídas dentro um conjunto de cidades candidatas, de tal maneira que usuários estejam localizados dentro de uma distância razoável de uma caixa de correio. Um outro exemplo é o desenho de rotas para equipes médicas móveis de atendimento em países em desenvolvimento, onde os serviços são realizados pelas equipes médicas em um número selecionado de vilarejos, então

veículos viajam através deste número limitado de vilarejos e toda localidade não visitada deve estar dentro de uma curta distância de uma visitada, tal que as pessoas possam ir, por exemplo, a pé a este local para serem atendidas (Foord [5]; Hodgson et al., [10]; Oppong e Hodgson [15]; Swaddiwudhipong et al. [17]). Problemas similares são encontrados em vários países ocidentais, em prevenção médica por equipes de saúde (Brown e Fintor[3]), na indústria leiteira (Simms [16]) e livrarias ou sistemas bancários móveis. O m -PRC com $T = V$ reduz-se para um Problema de Roteamento de Veículos (PRV) com unidades de demanda (Fisher [4], Laporte [12]).

2.1 Formulação do m -PRC para rotas preventivas

Para melhor entendimento, formulamos o m -PRC como um programa linear inteiro. Para cada vértice $v_l \in W$ definimos seu conjunto de cobertura $S_l = \{v_h \in V : c_{hl} \leq c\}$, $|S_l| \geq 2$. Seja y_{hk} uma variável binária igual a 1 se o vértice $v_h \in V$ é visitado pelo veículo k na solução, e seja x_{ijk} representando o número de vezes que o veículo k usa a aresta (v_i, v_j) em seu trajeto. Se $i = 0$ esta variável toma valores 0,1 ou 2 (2 no caso de uma viagem de retorno), se $i > 0$, x_{ijk} será binária.

$$\begin{aligned}
& \text{Min} \quad \sum_{k=1}^m \sum_{i=0}^{n-1} \sum_{j=i+1}^n c_{ij} x_{ijk} \\
& \text{S.a} \quad \sum_{k=1}^m \sum_{v_h \in S_l} y_{hk} \geq 1, \quad (v_l \in W) \tag{1} \\
& \sum_{k=1}^m y_{hk} \leq 1 \quad (v_h \in V \setminus \{v_0\}) \tag{2} \\
& \sum_{i=0}^{h-1} x_{ihk} + \sum_{j=h+1}^n x_{hjk} = 2y_{hk} \quad (v_h \in V \setminus \{v_0\}, k = 1, \dots, m) \tag{3} \\
& \sum_{k=1}^m \sum_{\substack{v_i \in S, v_j \in V \setminus S \\ \text{ou} \\ v_j \in S, v_i \in V \setminus S}} x_{ijk} \geq 2 \sum_{k=1}^m y_{hk} \quad (S \subset V, \quad 2 \leq |S| \leq n-2, \quad T \setminus S \neq \emptyset, \quad v_h \in S) \tag{4} \\
& \sum_{j=1}^n x_{0jk} = 2 \quad (k = 1, \dots, m) \tag{5} \\
& \left| \sum_{h=1}^n y_{hp} - \sum_{h=1}^n y_{hq} \right| \leq r \quad (p = 1, \dots, m; \quad q = 1, \dots, m; \quad p \neq q; \quad r \in \mathbb{N}) \tag{6} \\
& \sum_{k=1}^m y_{hk} = 1 \quad (v_h \in T \setminus \{v_0\}), \\
& y_{0k} = 1 \quad k = 1, \dots, m, \\
& y_{hk} \in \{0, 1\} \quad (v_h \in V \setminus \{v_0\}, \quad k = 1, \dots, m), \tag{7} \\
& x_{0jk} \in \{0, 1, 2\} \quad (v_j \in V \setminus \{v_0\}, \quad k = 1, \dots, m), \\
& x_{ijk} \in \{0, 1\} \quad (v_i, v_j \in V \setminus \{v_0\}, \quad i < j, \quad k = 1, \dots, m)
\end{aligned}$$

Nesta formulação as restrições (1) garantem que cada vértice v_l de W é coberto pelo menos uma vez. As restrições (2) garantem que cada vértice v_h de $V \setminus \{v_0\}$ aparece no máximo uma vez na solução. As restrições (3) garantem que a solução conterà duas arestas usadas pelo veículo k e incidentes ao vértice v_h , ou nenhuma. As restrições (4) impedem soluções com subrotas desconectadas. Se o vértice v_h não aparece na solução, então o lado direito é igual a zero, ou seja, redundante. Caso contrário, pelo menos duas arestas irão conectar S e seu complemento $V \setminus S$, para todo subconjunto próprio de V , isto é válido desde que um vértice v_h visitado pertença a S e existam vértices de T fora de S . As restrições

(5) garantem que exatamente m veículos entram e deixam o depósito. As restrições (6) garantem o equilíbrio entre o número de vértices nas diferentes rotas. Por fim, as restrições (7) são todas de integralidade padrão do problema.

3 Heurísticas usadas

Durante a execução dos algoritmos genéticos algumas heurísticas foram usadas, nesta seção apresentamos tais heurísticas para facilitar o entendimento dos algoritmos genéticos. Essas heurísticas são encontradas com riqueza de detalhes em [14].

3.1 Heurística GENIUS

Este procedimento é usado para resolver o PCV e se divide em duas fases, conhecidas como *Generalized Insertion Procedure* (**GENI**), que é a inserção de vértices na rota. E *Unstringing and Stringing* (**US**), um procedimento de pós-otimização, que tenta melhorar o custo de uma rota, removendo vértices e inserindo-os novamente. As formas de inserções e remoções de vértices não permitem em nenhum momento criação de subrotas. Com relação ao m -PRC, após decidido quais vértices entram em cada rota, temos exatamente que construir um ciclo Hamiltoniano de tamanho mínimo para cada rota, assim torna-se importante conhecer uma boa heurística para resolver o PCV, devido a sua complexidade.

3.1.1 Heurística GENI

Criada por Gendreau et al. [6], a heurística *Generalized Insertion Procedure* (GENI) demonstra ser igualmente eficiente para problemas simétricos e assimétricos. A característica principal desta heurística é que a inserção de um vértice v , não necessariamente ocorre entre dois vértices adjacentes da rota. No entanto, após sua inserção, aqueles vértices escolhidos tornam-se adjacentes a v . Os vértices são escolhidos um a um de forma arbitrária para inserção a partir de uma rota inicial. Suponha que desejamos inserir v entre algum par v_i e v_j de vértices da rota. Para uma dada orientação da rota, seja v_k um vértice no caminho de v_j para v_i , e v_l um vértice no caminho de v_i para v_j . Se um vértice v_h está na rota, vamos chamar os seus dois vértices adjacentes por v_{h-1} , referindo-se ao seu predecessor e v_{h+1} , ao seu sucessor. A inserção de v entre v_i e v_j pode ser feita de duas maneiras. Vamos classificá-las de Tipo 1 e Tipo 2. A Figura 2 ilustra essas duas possibilidades. Na sua parte superior temos um exemplo de inserção do Tipo 1, e na parte inferior uma do Tipo 2. Nela observamos configurações que são dadas a partir de uma escolha de orientação da rota. As linhas contínuas representam caminhos entre dois extremos, enquanto as pontilhadas representam vértices consecutivos.

Inserção Tipo 1

Neste primeiro tipo temos que tomar $v_k \neq v_i$ e $v_k \neq v_j$. Inserir v na rota resulta em eliminar os arcos (v_i, v_{i+1}) , (v_j, v_{j+1}) , e (v_k, v_{k+1}) e em substituí-los pelos arcos (v_i, v) , (v, v_j) , (v_{i+1}, v_k) e (v_{j+1}, v_{k+1}) . Isto significa que os dois caminhos (v_{i+1}, \dots, v_j) e (v_{j+1}, \dots, v_k) são invertidos. Note que tomando $j = i + 1$ e $k = j + 1$ leva a um procedimento de inserção entre vértices adjacentes da rota. O custo deste último também é testado e comparado com os demais.

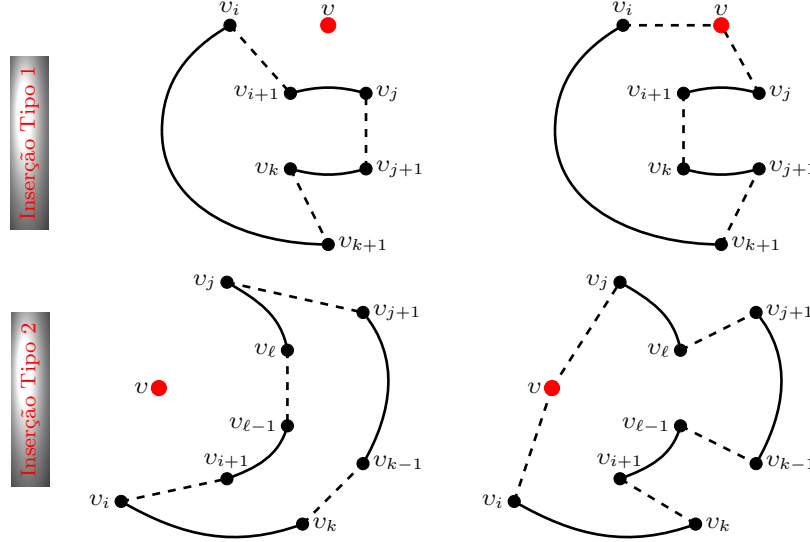


Figura 2: Inserção Tipo 1 e Tipo 2 de v entre v_i e v_j

Inserção Tipo 2

Neste segundo tipo temos que tomar $v_k \neq v_j$ e $v_k \neq v_{j+1}$; $v_l \neq v_i$ e $v_l \neq v_{i+1}$. Inserir v na rota resulta em eliminar os arcos (v_i, v_{i+1}) , (v_{l-1}, v_l) , (v_j, v_{j+1}) e (v_{k-1}, v_k) e em substituí-los pelos arcos (v_i, v) , (v, v_j) , (v_l, v_{j+1}) , (v_{k-1}, v_{l-1}) e (v_{i+1}, v_k) . Com isso os dois caminhos $(v_{i+1}, \dots, v_{l-1})$ e (v_l, \dots, v_j) são invertidos.

A heurística GENI considera as duas possibilidades de orientações da rota para cada possibilidade de inserção de vértices. Temos que o número potencial de escolhas para v_i, v_j, v_k e v_l é grande, então são limitadas as formas de escolhas desses vértices, como segue.

Para algum vértice $v \in V$, define-se sua p -vizinhança $N_p(v)$ como sendo o conjunto dos p vértices na rota mais próximos de v , com respeito as distâncias c_{ij} 's; se v tem menos que p vizinhos na rota, eles todos pertencem a $N_p(v)$. Assim, para um dado parâmetro p , primeiro selecionamos $v_i, v_j \in N_p(v)$, $v_k \in N_p(v_{i+1})$, e $v_l \in N_p(v_{j+1})$. Na prática,

p é um número relativamente pequeno. Agora os passos da heurística GENI podem ser descritos.

Passos da heurística GENI

Passo 1. Selecione um subconjunto de três vértices arbitrariamente e crie uma rota inicial. Inicialize a p -vizinhança de todos os vértices.

Passo 2. Escolha arbitrariamente um vértice v fora da rota e calcule o menor custo de inserção de v considerando as duas possibilidades de orientação da rota e os dois tipos de inserção. Refaça a p -vizinhança de todos os vértices, considerando o fato de que v está agora na rota.

Passo 3. Se todos os vértices já fazem parte da rota, pare. Caso contrário vá para o Passo 2.

3.1.2 Heurística US

Esta é uma heurística de pós-otimização que busca melhorar a solução de um PCV. Desenvolvida também por Gendreau et al. [6], consiste em remover um vértice de uma rota viável e inseri-lo novamente. Esses processos são chamados de *Unstringing* e *Stringing* (US) e podem ser aplicados em rotas produzidas por qualquer heurística. O procedimento de inserção do vértice é feito de maneira idêntica ao Passo 2 da heurística GENI. O procedimento de remoção é o inverso. Quando um vértice v_i é removido da rota, temos que considerar duas maneiras de reconectá-la. Vamos classificá-las de Tipo 1 e Tipo 2. A Figura 3 ilustra essas duas possibilidades. Na sua parte superior temos um exemplo de remoção do Tipo 1, e na parte inferior uma do Tipo 2. Nela, observamos configurações que são dadas a partir de uma escolha de orientação da rota. As linhas contínuas representam caminhos entre dois extremos, enquanto as pontilhadas representam vértices consecutivos.

Remoção Tipo 1

Seja $v_j \in N_p(v_{i+1})$, e, para uma dada orientação da rota seja $v_k \in N_p(v_{i-1})$ um vértice no caminho $(v_{i+1}, \dots, v_{j-1})$. Então os arcos (v_{i-1}, v_i) , (v_i, v_{i+1}) , (v_k, v_{k+1}) e (v_j, v_{j+1}) são eliminados, enquanto os arcos (v_{i-1}, v_k) , (v_{i+1}, v_j) e (v_{k+1}, v_{j+1}) são inseridos. Os dois caminhos (v_{i+1}, \dots, v_k) e (v_{k+1}, \dots, v_j) mudam de orientação.

Remoção Tipo 2

Seja $v_j \in N_p(v_{i+1})$, para uma dada orientação da rota, seja $v_k \in N_p(v_{i-1})$ um vértice no caminho $(v_{j+1}, \dots, v_{i-2})$. Também seja $v_l \in N_p(v_{k+1})$ no caminho (v_j, \dots, v_{k-1}) .

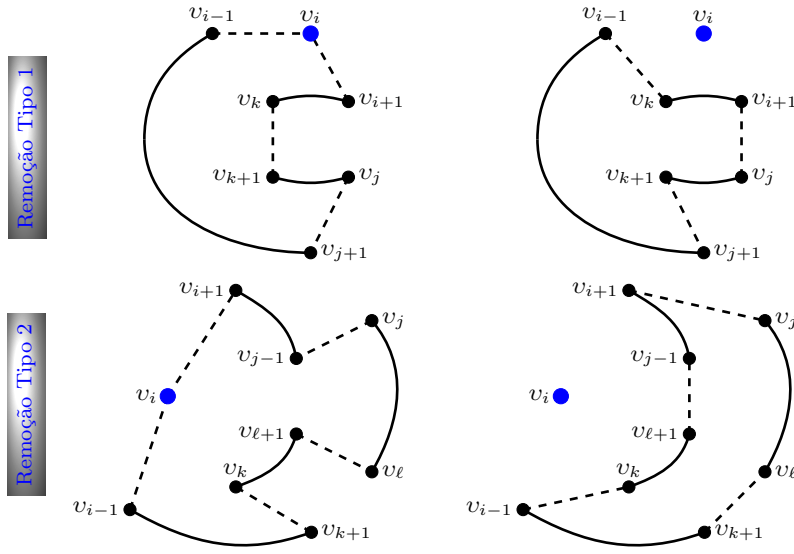


Figura 3: Remoção Tipo 1 e Tipo 2 do vértice v_i da rota

Então os arcos (v_{i-1}, v_i) , (v_i, v_{i+1}) , (v_{j-1}, v_j) , (v_l, v_{l+1}) e (v_k, v_{k+1}) são eliminados, enquanto os arcos (v_{i-1}, v_k) , (v_{l+1}, v_{j-1}) , (v_{i+1}, v_j) e (v_l, v_{k+1}) são inseridos. Os dois caminhos $(v_{i+1}, \dots, v_{j-1})$ e (v_{l+1}, \dots, v_k) mudam de orientação. Considerando um PCV com n cidades, os passos da heurística US podem ser descritos.

Passos da heurística US

Passo 1. Considere uma rota inicial τ de custo z . Faça $\tau^* := \tau$, $z^* := z$ e $t := 1$.

Passo 2. Iniciando da rota τ , aplique o procedimento de remoção e inserção com o vértice v_t , considerando em cada caso os dois tipos possíveis de operação e as duas orientações da rota. Seja $\tilde{\tau}$ a rota obtida e \tilde{z} seu custo. Faça $\tau := \tilde{\tau}$ e $z := \tilde{z}$.

- Se $z < z^*$, faça $\tau^* := \tau$, $z^* := z$ e $t := 1$; repita o Passo 2.
- Se $z \geq z^*$, faça $t := t + 1$.
- Se $t = n + 1$, Pare: a melhor rota disponível é τ^* e seu custo é z^* . Caso contrário, repita o Passo 2.

Realmente, um vértice para ser removido de uma rota não precisa necessariamente estar localizado entre dois vértices pertencentes à sua p -vizinhança. Assim, reinserir um vértice

na posição que ele ocupava antes de sua remoção da rota, pode não ser permitido. Porém, a melhor rota conhecida sempre é armazenada.

3.1.3 Passos da heurística GENIUS

Passo 1. Construa um ciclo Hamiltoniano τ de custo z usando a heurística **GENI**.

Passo 2. Aplique a heurística **US** sobre τ .

3.2 H-1-PRC-MOD

Esta é uma modificação (Oliveira, [14]) da heurística descrita por Gendreau et al. [7] e combina a heurística Primal1 (Balas e Ho, [1]) para o problema de cobertura com a heurística GENIUS (Gendreau et al., [6]) para o PCV. Ela é usada para resolver cada 1-PRC no processo de resolução do m -PRC. Durante a aplicação da heurística H-1-PRC-MOD, temos que garantir a cobertura do conjunto W associado ao seguinte problema de cobertura

$$\begin{aligned}
& \text{minimizar} && \sum_{v_h \in V} c_h y_h \\
& \text{sujeito a} && \\
& && \sum_{v_h \in S_l} y_h \geq 1, \quad \forall v_l \in W, \quad S_l = \{v_h \in V : c_{hl} \leq c\} \\
& && y_h = 1 \quad \forall v_h \in T \\
& && y_h \in \{0, 1\}, \quad \forall v_h \in V \setminus T.
\end{aligned} \tag{1}$$

onde c_h é definido como o custo de inserção de um vértice v_h na rota usando as configurações da heurística GENIUS. A heurística Primal1 constrói uma solução para o problema acima por um método ganancioso, incluindo gradualmente uma variável y_h na solução ao minimizar uma certa função $f(c_h, b_h)$, onde b_h é o número de vértices $v_l \in W$, satisfazendo $c_{hl} \leq c$, mas ainda não cobertos. As funções f usadas e sugeridas em [1] são: (1) $f(c_h, b_h) = c_h$, (2) $f(c_h, b_h) = c_h/b_h$ e (3) $f(c_h, b_h) = c_h/\log_2 b_h$. Ela constrói uma solução para o Problema (1) usando a definição (3) de f , e todos vértices de V cobrindo um vértice de W mais de uma vez são removidos e uma nova solução é construída com outra definição de f . Na H-1-PRC-MOD utilizamos a sequência (3)-(2)-(1).

A heurística H-1-PRC-MOD constrói primeiramente uma rota utilizando a GENIUS com todos os vértices de T . Depois seleciona gradualmente vértices em $V \setminus T$ utilizando cada uma das três definições de f com a heurística Primall e adiciona tais vértices na rota através dos tipos de inserções da GENIUS até que todos os vértices de W sejam cobertos. Os passos da heurística H-1-PRC-MOD são os seguintes.

3.2.1 Passos da heurística H-1-PRC-MOD

Passo 1 Faça $H := T$, $\bar{z} := \infty$. A definição atual de f é (3). Com o conjunto H , resolva um PCV, de custo z , usando a heurística GENIUS. Se todos os vértices de W já estão cobertos, pare com a melhor solução construída com os vértices de H . Caso contrário vá para o Passo 3.

Passo 2 Se $z \leq \bar{z}$, faça $\bar{z} := z$ e $\bar{H} := H$. Se a definição atual de f é a última, pare com a melhor solução dada pela rota \bar{H} de custo \bar{z} . Caso contrário, remova de $H \setminus T$ todos os vértices associados com multicobertura de vértices em W utilizando os dois tipos de remoção de vértices e as duas orientações da rota, como descrito na heurística US. Vá para o Passo 3 com a próxima definição de f .

Passo 3 Dentre os vértices em $V \setminus H$ selecione gradualmente v_{h^*} utilizando a heurística Primall de acordo com a definição atual de f . Os custos para a escolha são dados pelos dois tipos de inserção de vértices e orientações da rota de acordo com a heurística GENI. Inclua cada v_{h^*} na rota H usando a GENI até que todos vértices de W sejam cobertos. Denote por z o custo da rota e vá para o Passo 2.

3.3 Heurística 2-opt Balanceada

Esta é uma adaptação (Oliveira, [14]) da heurística 2-opt* descrita por Hachicha et al. [9] para o m -PRC. Neste caso, o objetivo maior, além de melhorar uma solução atual de um m -PRC, é manter o equilíbrio entre o número de vértices nas diferentes rotas. O Passo 1 desta heurística faz m cópias da base como no múltiplo PCV descrito por Lenstra e Rinnooy Kan [13] e os Passos de 2 e 3 são adaptações da 2-opt*. Esta heurística executa os seguintes passos.

3.3.1 *Passos da heurística 2-opt Balanceada*

Passo 1 Chame de ρ o número de vértices na menor rota da solução viável inicial. Crie uma única rota, fazendo m cópias da base e considere todas as combinações de pares de arestas nas rotas atuais. Veja descrição completa em [14].

Passo 2 Dado o par de arestas, $\{(v_r, v_s), (v_t, v_u)\}$, remova (v_r, v_s) , (v_t, v_u) da rota e considere sucessivamente as duas opções de novas arestas: (i) (v_r, v_t) e (v_s, v_u) ; (ii) (v_r, v_u) e (v_s, v_t) . Isto cria um ou dois ciclos (veja [14]). Cada ciclo só é viável se (i) ele contém no mínimo uma cópia da base e (ii) todo ciclo iniciando e terminando em uma base, com relação ao número de vértices, deve ter no mínimo ρ vértices.

- Se ambas opções derem soluções inviáveis ou não melhorar a melhor solução conhecida, repita o Passo 2 com o próximo par de arestas, ou vá para o Passo 4 se todos já foram considerados.
- Caso contrário, guarde a melhor opção viável e vá para o Passo 3.

Passo 3 Construa uma solução do m -PRC usando apenas uma base, onde cada ciclo ligando duas bases ou a mesma base define uma rota. Vá para o Passo 1.

Passo 4 Considere todas as combinações de pares de vértices $\{v_r, v_s\}$ das rotas.

- Dado um par de vértices, considere a possível troca desses vértices de posição. Removendo cada vértice da sua posição original e incluindo na posição do outro (veja [14]).
 - Se não houver melhora da melhor solução conhecida, repita o Passo 2.1 com o próximo par de vértices ou pare se todos já foram considerados. Caso contrário, efetue a troca de vértices, guarde o melhor custo obtido e retorne ao Passo 4.
-

3.4 Processo de varredura

Este procedimento é baseado no processo de construção de rotas para o PRV descrito por Gillet e Miller [8]. Porém, uma adaptação (Hachica et al. [9]) no processo de varredura é feita para ser aplicado nos vértices de $T \cup W$, tal que, um local de visita é selecionado de acordo com um incremento no ângulo polar à partir de um ponto fixo até sua posição no espaço de busca. Uma varredura no sentido anti-horário é feita, e as k rotas são selecionadas de acordo com o conjunto $T \cup W$. Pela sua característica vamos chamar este processo de Procedimento de Abertura de um Leque (Oliveira [14]).

3.4.1 *Passos do Procedimento de Abertura de um Leque*

-
- Escolha $v \in (T \cup W) \setminus \{v_0\}$ o mais próximo da base.
 - Considere uma linha tendo uma extremidade na base e passando por v .
 - Construa um vetor G ordenado com todos os vértices $v_h \in T \cup W$ em ordem crescente do ângulo $\theta_h = \widehat{vv_0v_h}$.
 - Sejam $C_r = \{v_l \in W : c_{rl} \leq c\}$, $D := \emptyset$, $z := |G|$, então $G = \{v_1, v_2, \dots, v_z\}$. Faça $h := 1$.
 - Enquanto $h \leq z$
 - Se $v_h \in T$, $D = [D \ v_h]$, $h := h + 1$.
 - Se $v_h \in W$, determine v_r de S_h que cobre o maior número de vértice de W , faça $D = [D \ v_r]$, $T \cup W := (T \cup W) \setminus C_r$ e $h := h + 1$.
 - Se $v_h \notin W$, $h := h + 1$.
 - D deve ser considerado como um vetor ordenado por construção.
-

Vamos definir N_k como sendo o número de vértices permitido inicialmente na rota k . Esses valores serão obtidos de acordo com a quantidade de vértices incluídos no vetor D a partir do Procedimento de Abertura de um Leque. Assim, vamos obter os N_k iniciais da seguinte maneira.

- Seja $N_k = \left\lfloor \frac{|D|}{m} \right\rfloor$, o número de vértices permitidos para a rota k no Passo 1 do AG, e $\beta = |D| - m \cdot N_k$. Temos que $\beta < m$. Faça $k := 1$.
- Enquanto $\beta \neq 0$, faça:
 - $N_k := N_k + 1$.
 - $k := k + 1$ e $\beta := \beta - 1$.

Este critério é escolhido para manter um equilíbrio entre o número de vértices nas diferentes rotas. Note que assim, a diferença é de apenas um vértice de uma rota para outra.

4 Algoritmo genético

Sugerimos duas propostas de algoritmo para resolução de um m -PRC. Um deles, chamaremos de Algoritmo Genético Sequencial (A.G.S.), e o outro, de Algoritmo Genético H-1-PRC (A.G.H.). No A.G.S. usamos as heurísticas GENIUS e 2-opt Balanceada apenas no final do algoritmo para tentar melhorar a solução final. Enquanto que no A.G.H. utilizamos hibridamente as heurísticas H-1-PRC-MOD e 2-opt Balanceada em todo seu processo de resolução. Nesta seção apresentamos os algoritmos e suas características.

4.1 Base para o algoritmo genético

Nos AG uma população de soluções é mantida e um processo reprodutivo permite selecionar soluções à partir desta população. As primeiras soluções são as soluções pais. Soluções filhos (*proles*) são produzidas exibindo algumas características de cada solução pai. Deve-se medir a qualidade de cada solução, por exemplo, podemos avaliar o valor da função objetivo nesta solução, no caso o tamanho total das rotas, e também o nível de violação de alguma restrição. Esta medida é conhecida como o aptidão *fitness* do indivíduo. Indivíduos com melhores níveis de aptidão têm maior chance de sobreviverem e de se reproduzirem, com isso, a expectativa de que o nível de aptidão de toda a população irá melhorar quando estes indivíduos sobrevivem e participam da reprodução (Barrie et al. [2]).

Normalmente a representação de cada solução (indivíduo da população) é feita a partir de um vetor (cromossomo). As entradas de cada cromossomo é conhecida como genes. O uso de vetores binários nas representações de soluções em AG facilita bastante em muitos problemas de programação combinatória, porém representações com entradas decimais também podem ser usadas. No nosso caso, achamos que o uso de vetores com representação binária e inteira é o mais apropriado. Assim, dado n locais de visitas obrigatórias e m veículos à serem usados, o cromossomo para cada individual solução tem a forma de um vetor de tamanho $n+p$, onde cada gene poderá tomar valores entre 1 e m , sendo que p representa o número de visitas opcionais que poderam ser incluídas nas rotas. Nesta representação não temos de forma explícita em qual rota cada veículo se encontra, porém conhecida a atribuição de cada local de visita a seu veículo, podemos facilmente gerar um vetor binário e associá-lo a rotas individuais com o seu custo total de viagem e o seu nível de violação das restrições.

4.2 Gerando a população inicial

Para ambos os casos, A.G.S. e A.G.H., a população inicial é gerada utilizando o processo de varredura apresentado na Seção 3.4. Ela é gerada até atingir um número X de indivíduos. Para isso, seguimos os passos abaixo.

4.2.1 *Passos de construção da população inicial*

Passo 1 Determine o vetor D através do Procedimento de Abertura de um Leque, considere D como um ciclo e obtenha os valores N_k , como descrito na Seção 3.4.1.

Repita o Passos 2 iniciando com cada vértice v incluído na primeira rota construída ao se processar o Passo 2 pela primeira vez, até atingir X indivíduos

Passo 2 Iniciando de um vértice v arbitrário em D , construa uma rota k fazendo um passeio em D no sentido anti-horário e incluindo os vértices visitados neste passeio até atingir o valor N_k , complete tal rota unindo seu último vértice a v_0 . Então considere o próximo vértice neste passeio como v , e repita este passo para a próxima rota até todos os vértices de D serem incluídos em alguma rota. Isto produz um indivíduo da população.

Passo 3 Se o número X de indivíduos ainda não foi atingido. Crie novos indivíduos usando 2 _{τ} -crossover até atingí-lo.

4.2.2 Representação da população inicial

Abaixo temos a representação da população inicial.

vértice	4	5	3	7	6	2	43	44
varredura	1	2	3	4	5	6	7	8
conjunto	0	0	0	0	0	0	1	1
cromossomo 1	1	1	1	2	2	2	3	3
cromossomo 2	3	3	1	1	1	2	2	3
cromossomo i
cromossomo (X-1)	3	1	1	2	2	2	2	3
cromossomo (X)	3	1	1	2	1	2	3	2

Na primeira linha representamos o número que identifica o vértice no conjunto V da formulação do m -PRC. Na segunda linha, temos a sequência que os vértices de V foram selecionados no processo de varredura. Na terceira linha denotamos em que conjunto o vértice selecionada está, se a posição na linha tem entrada 0 ou 1, o vértice pertence ao conjunto T ou $V \setminus T$ respectivamente. Este último serve para contarmos quantos vértices

de visitas opcionais uma determinada solução contém. Todas as outras linhas representam soluções determinadas durante os passos apresentados em 4.2.1.

4.3 Aptidão

Um indivíduo será considerado apto se atender a dois pontos principais: respeita a restrição de equilíbrio nos números de vértices nas diferentes rotas e se sua representação cromossômica é distinta daqueles indivíduos já existentes. Atendido a esses dois requisitos podemos computar o *fitness* de um indivíduo, que será o valor da função objetivo. No caso do A.G.S. computamos exatamente o custo das rotas de acordo com a sua sequência cromossômica. No A.G.H. a sequência cromossômica só indica os possíveis vértices que farão parte de cada rota individualmente, porém após a aplicação das heurísticas H-1-PRC-MOD e 2-opt Balanceada, a sequência cromossômica poderá ser alterada, de acordo com a decisão de qual vértice em $V \setminus T$ ficará em cada rota. Após isso, o custo total de um indivíduo é computado.

4.4 O processo reprodutivo (2_r -crossover)

Selecionamos dois indivíduos pais da população de forma aleatória. As soluções filhos são produzidas a partir dos dois pais utilizando o procedimento crossover padrão. Vamos utilizar o crossover 2-pontos, no qual dois pontos no cromossomo são escolhidos aleatoriamente. Os genes de uma das proles consiste dos genes do pai 1 que estão na esquerda do primeiro ponto e na direita do segundo ponto, juntamente com os genes do pai 2 que estão entre os dois pontos escolhidos. A segunda prole é obtida com o mesmo procedimento, porém invertendo a posição dos pais. Proles com existência duplicada na população são descartadas. Vamos chamar este procedimento de 2_r -crossover, porque se a prole não respeita o valor r pré determinado na formulação do m -PRC para rotas preventivas, ela também é descartada.

Um dos objetivos é definir as rotas, de modo que, para todos os indivíduos da população, cada rota seja traçada aproximadamente na mesma região. Para exemplares com o depósito centralizado e uma distribuição uniforme de pontos de visita, o crossover 2-ponto equivale a dividir o plano em dois setores por duas linhas determinadas aleatoriamente a partir do depósito e usa-se os veículos alocados do pai 1 em um setor e do outro pai 2 no outro setor.

A Figura¹ 5 ilustra a aplicação do crossover 2-pontos para um problema com 20 pontos de visitas e quatro veículos. Neste exemplo, os pontos de visitas estão enumerados de acordo com sua seleção no processo de varredura. A parte superior da Figura 5 apresenta

¹Figura retirada da referência [2]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1:	1	1	1	1	2	2	2	3	3	2	3	3	3	3	4	4	4	4	4	1
P2:	1	1	2	2	1	2	2	2	2	3	3	3	3	3	4	4	4	4	1	4

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
F1:	1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	1
F2:	1	1	2	2	1	2	2	3	3	2	3	3	3	3	4	4	4	4	1	4

Figura 4: Cromossomo dos Pais 1, 2 e Proles

a estrutura de dois indivíduos pais, e suas representações em cromossomos estão ilustradas na parte superior da Figura² 4.

Dois pontos foram escolhidos entre os pontos de visitas 6 e 7, e entre os pontos 15 e 16. Na Figura 5, essas escolhas são representadas por linhas pontilhadas, que divide a região em dois setores. A aplicação do crossover 2-pontos fornece as duas soluções filhos ilustradas na parte inferior da Figura 4. Na parte inferior da Figura 5 temos a representação da Prole 1, que é a melhor entre as duas. Note que ela tem os veículos designados do Pai 1 em um setor, e do Pai 2 no outro setor da região, após a sua divisão.

4.5 Troca de indivíduos na população

Sempre que um indivíduo estiver apto para entrar na população, ele entrará no lugar do indivíduo com a pior função objetivo no momento, e o indivíduo selecionado para sair será descartado.

4.6 Mutação

A mutação será feita em indivíduos selecionados aleatoriamente da população. Se baseia em testar trocas de vértices entre as diferentes rotas até que se consiga uma melhora no custo final das rotas ou até que todas as trocas possíveis sejam testadas. No último caso, o indivíduo retorna para a população sem sofrer alteração. Os seguintes passos de mutação será usado apenas no A.G.S., pois no A.G.H. passos parecidos são realizados na aplicação da heurística 2-opt Balanceada.

²Figura retirada da referência [2]

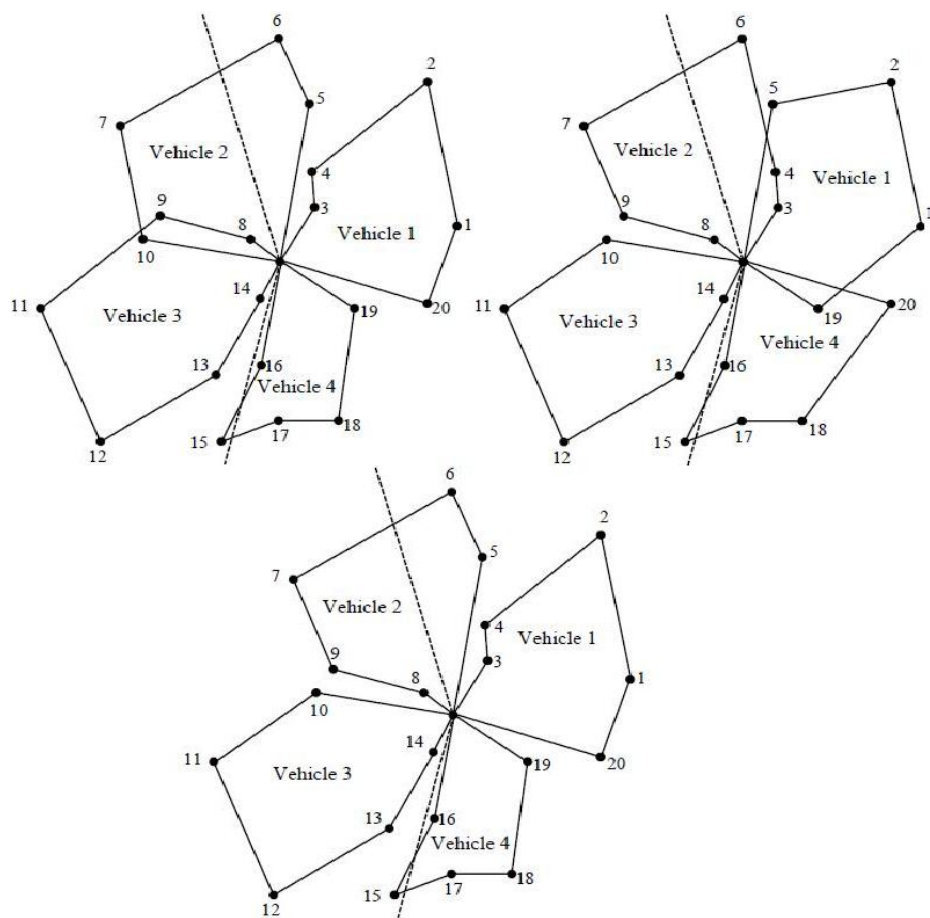


Figura 5: Pais 1, 2 e prole

4.6.1 Passos da mutação

Dada uma solução, faça m cópias da base e considere todas as combinações de pares de vértices $\{v_r, v_s\}$ das rotas.

Passo 1. Dado um par de vértices, considere a possível troca desses vértices de posição. Removendo cada vértice da sua posição original e incluindo na posição do outro (veja a Figura 6).

Passo 2. Se não houver melhoria no custo, repita o Passo 1 com o próximo par de vértices. Pare assim que houver melhora, efetue a troca de vértices e guarde o custo obtido. Se todas as combinações já foram considerados o indivíduo não é alterado.

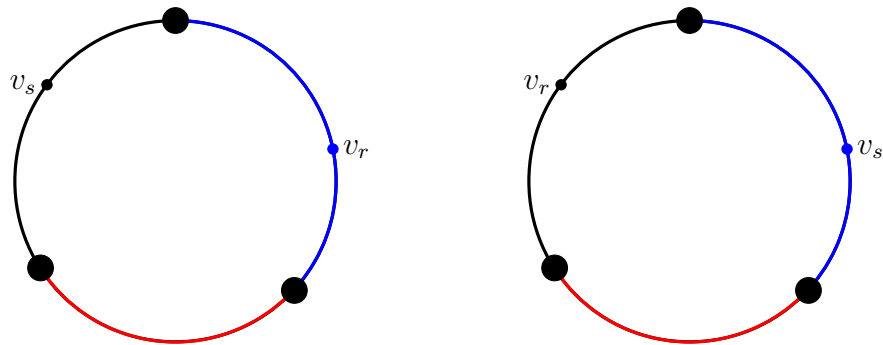


Figura 6: Troca de vértices entre rotas opostas na mutação

4.7 Algoritmo genético sequencial

A melhor solução é sempre guardada durante todo processo deste algoritmo.

4.7.1 *Passos do A.G.S.*

Gere a população inicial até X indivíduos, como em 4.2.1

Avalie a aptidão de cada indivíduo da população, conforme 4.3

ENQUANTO o número de gerações (Y) não é atingido

REPRODUÇÃO

ENQUANTO o critério de entrada não for satisfeito (conforme 4.3)

Selecione dois pais da população usando um sorteio aleatório

Produza duas proles e avalie a aptidão das proles, conforme 4.4 e 4.3

Escolha a prole favorita

Escolha um membro da população e efetue a substituição, conforme 4.5

FIM ENQUANTO

MUTAÇÃO

ENQUANTO o número de mutações (Z) não é atingido

Sorteie um indivíduo da população aleatoriamente

Produza mutação conforme 4.6.1 e avalie a aptidão

Escolha um membro da população e efetue a troca

FIM ENQUANTO

FIM ENQUANTO

Aplique a GENIUS e depois a 2-opt Balanceada na melhor solução

4.8 Algoritmo genético H-1-PRC

A melhor solução é sempre guardada durante todo processo deste algoritmo. Neste caso, não temos fase de mutação.

4.8.1 *Passos do A.G.H.*

Gere a população inicial até X indivíduos, como em 4.2.1

Avalie a aptidão (*aplicação das heurísticas H-1-PRC-MOD e 2-opt Balanceada*) de cada indivíduo da população, conforme 4.3

ENQUANTO o número de gerações (Y) não é atingido

REPRODUÇÃO

ENQUANTO o critério de entrada não for satisfeito (conforme 4.3)

Selecione dois pais da população usando um sorteio aleatório

Produza duas proles e avalie a aptidão das proles, conforme 4.4 e 4.3

Escolha a prole favorita

Escolha um membro da população e efetue a substituição, conforme 4.5

FIM ENQUANTO

FIM ENQUANTO

5 Resultados computacionais

As implementações foram feitas em um código MATLAB. Comparamos os algoritmos genéticos com outras heurísticas descritas em [14]. A validação do código foi feita executando alguns exemplares gerados aleatoriamente e um exemplar com dados reais. Utilizamos um Pentium 4, 2G RAM e 3.40 GHz em Linux..

5.1 Exemplos gerados aleatoriamente

Semelhante ao apresentado em [9] construímos exemplos para o m -PRC gerados aleatoriamente com distribuição uniforme. Foram criados 300 exemplos. Para cada exemplo foram gerados $|V| + |W|$ pontos numa região quadrada $[0, 100] \times [0, 100]$. A localização da base ficou restrita ao quadrado $[35, 65] \times [35, 65]$. Os conjuntos T e V foram definidos ao tomar os primeiros $|T|$ e $|V|$ pontos, respectivamente. Todos os outros pontos foram atribuídos a W . Testamos as combinações dos seguintes valores: $|V| = 50, 100, 200$; $|W| = 50, 100, 200, 400$ e $|T| = \left\lceil \frac{|V|}{8} \right\rceil, \left\lceil \frac{|V|}{4} \right\rceil, \left\lceil \frac{|V|}{2} \right\rceil$. Dividimos 300 exemplos em classes, como podemos ver na Tabela 1. Nela observamos que os exemplos se dividem em 5 classes principais com dimensões iguais a 100, 150, 200, 300 e 400. Em cada uma dessas classes principais fizemos outras três subdivisões de acordo com o tamanho do conjunto T escolhido. Foram gerados 10 exemplos em cada subdivisão.

$ T = \left\lceil \frac{ V }{8} \right\rceil, \left\lceil \frac{ V }{4} \right\rceil, \left\lceil \frac{ V }{2} \right\rceil$					
Exemplar	Quantidade	Dimensão	$ V $	$ W $	$ T $
Classe 100-1	10	100	50	50	7
Classe 100-2	10	100	50	50	13
Classe 100-3	10	100	50	50	25
Classe 150-1	10	150	50	100	7
Classe 150-2	10	150	50	100	13
Classe 150-3	10	150	50	100	25
Classe 200-1	10	200	100	100	13
Classe 200-2	10	200	100	100	25
Classe 200-3	10	200	100	100	50
Classe 300-1	10	300	100	200	13
Classe 300-2	10	300	100	200	25
Classe 300-3	10	300	100	200	50
Classe 400-1	10	400	200	200	25
Classe 400-2	10	400	200	200	50
Classe 400-3	10	400	200	200	100

Tabela 1: Exemplos gerados aleatoriamente com distribuição uniforme

5.2 Exemplar do problema em estudo com dados reais

Foram disponibilizados pela Guarda Civil Municipal da cidade de Vinhedo-SP dados referentes a segurança da cidade. O administrador providenciou uma lista com 102 endereços de visitas obrigatórias, entre esses, estão todas as escolas, creches, postos de saúde, hospitais públicos, etc. (conjunto T). Outra lista contendo 133 endereços importantes para ações preventivas de segurança, entre esses, estão várias praças, semáforos, etc. (conjunto W). O conjunto V foi formado com os pontos geográficos referentes a todos cruzamentos de ruas da cidade, conforme a Tabela 2.

Exemplar	Quantidade	Dimensão	$ V $	$ W $	$ T $
Dados Reais	1	2731	2496	133	102

Tabela 2: Exemplar gerado com dados reais

5.3 Definição da constante c

Definimos $c = \max\{\max_{v_h \in V \setminus T} \min_{v_l \in W} \{c_{lh}\}, \max_{v_l \in W} \{c_{l, h(l)}\}\}$ para os exemplares gerados aleatoriamente, onde $h(l)$ é o índice do vértice em $V \setminus T$ que é o segundo mais próximo a v_l . Com esta definição, cada vértice de $V \setminus T$ cobre pelo menos um vértice de W e cada vértice de W é coberto por pelo menos dois vértices de $V \setminus T$. Para o exemplar com dados reais o valor de c foi escolhido pelo administrador responsável pela região, que estipulou 150m como uma distância adequada.

5.4 Análise dos resultados

Para melhor entendimento dos resultados nas tabelas, temos algumas definições de parâmetros:

- T : tempo de execução em segundos
- Dif. N_k : diferença entre o número de vértices nas diferentes rotas encontrado
- t : número de vezes que um algoritmo é executado no mesmo exemplar
- R : Relação: valor obtido ao se dividir o custo de cada heurística pelo melhor custo obtido entre todas as heurísticas por exemplar

Para todos os exemplares definimos o tamanho da população, $X=30$, o número de gerações, $Y=15$, e para o A.G.S. fixamos o número de mutações $Z=5$.

A parte superior da Tabela 3 compara os valores médios da função objetivo para três classes de exemplares. Na sua parte inferior temos para as mesmas classes a comparação quanto

Classe	A.G.S.	A.G.H.	1-PRC, m -PRC
100-3	566.58	521.30	524.18
200-2	589.20	538.25	543.17
300-1	574.41	504.62	520.11
Classe	A.G.S.	A.G.H.	1-PRC, m -PRC
100-3	1.0868	1	1.0055
200-2	1.0946	1	1.0091
300-1	1.1383	1	1.0306

Tabela 3: Compara o valor da função objetivo

ao valor R. Note que comparamos os algoritmos genéticos com a heurística primeiro 1-PRC/segunda m -PRC, cuja descrição é encontrada em [14]. Percebemos que o desempenho do A.G.H. é melhor para todos os casos.

A Tabela 4 compara os valores médios dos tempos de execução para todas as classes de exemplares. Onde não é apresentado valores indica que o algoritmo superou o tempo tolerável. Note que novamente comparamos os algoritmos genéticos com a heurística primeiro 1-PRC/segunda m -PRC. Percebemos que o desempenho do A.G.S. é melhor para todos os casos, e a diferença entre os tempos de execução é bem significativa.

Na Tabela 5 encontramos os resultados para o exemplar com dados reais. Ela compara o valor da função objetivo e o tempo de execução entre o A.G.S. e a Heurística da Varredura descrita em [14]. Neste caso o A.G.H. novamente superou o tempo tolerável. Percebemos que o A.G.S. encontra um valor para função objetivo muito parecido com a da Heurística da Varredura, porém com um tempo muito abaixo.

Os resultados mostram que o A.G.S. é bastante competitivo, pois encontra valores para função objetivo muito próximos do A.G.H., porém em um menor tempo. O A.G.H. se destacou em comparação com as melhores heurísticas apresentadas em [14], porém efetua muitas operações, tornando-se um algoritmo lento. Ao contrário do A.G.S. que é uma versão muito mais simples e rápida.

Classe	A.G.S.	A.G.H.	1-PRC, m -PRC
100-1	2.63	63.44	6.62
100-2	3.44	72.83	13.71
100-3	12.49	495.20	108.79
150-1	4.29	138.55	11.55
150-2	4.35	146.22	20.14
150-3	11.27	489.81	105.26
200-1	15.02	1070.34	97.21
200-2	22.10	1001.98	328.41
200-3	71.50	4507.84	3041.96
300-1	22.5	1243.57	259.49
300-2	27.2	—	395.56
300-3	72.5	—	4377.25
400-1	73.25	—	3689.24
400-2	122.25	—	6630.70
400-3	457.50	—	13796.70

Tabela 4: Compara o tempo de execução

Heurística	R	(Km)	T(seg)	Dif. N_k	tentativas	viável
Varredura	1,0141	94,801	77831	2	37	sim
A.G.S.	1	93,486	701	4	$t \times 30 \times 15 \times 5$	sim
A.G.H.	—	—	—	—	—	—

Heurística	R	(Km)	T(seg)	Dif. N_k	tentativas	viável
Varredura	1,0159	104,800	63352	3	31	sim
A.G.S.	1	103,159	796	4	$t \times 30 \times 15 \times 5$	sim
A.G.H.	—	—	—	—	—	—

Tabela 5: Compara tempo e função objetivo (dados reais)

6 Conclusões

Desenvolvemos dois tipos de algoritmos genéticos para resolução de uma adaptação do problema de rotas de cobertura multiveículo. Concluimos que as implementações para construções de rotas utilizando os critérios de algoritmo genético são bem relevantes e de fácil manipulação. Também percebemos que o A.G.S. é um algoritmo competitivo dentre os critérios de comparação adotados, pois resolve os problemas de forma rápida e é de simples implementação. Por outro lado, o A.G.H. encontra custos melhores, porém de

forma lenta, devido as heurísticas que são usados por ele hibridamente na construção das rotas, e por conta disto, ele não resolveu alguns exemplares. Os testes mostram que os algoritmos genéticos contribuídos neste trabalho podem ser usadas para encontrar boas soluções para as diversas aplicações do m -PRC.

Referências

- [1] E. Balas, A. Ho, *Set covering algorithms using cutting planes, heuristics, and sub-gradient optimization: A computational study*, Mathematical Programming 12 (1980) 37–60.
- [2] M. B. Barrie, M. A. Ayechev, *A genetic algorithm for the vehicle routing problem*. Computers and Operations Research 30 (2003) 787–800.
- [3] M. L. Brown, L. U. S. Fintor, *Screening mammography services with mobile units: results from the National Survey of Mammography Facilities*, Radiology 1995; 195: 529–32.
- [4] M. L. Fisher, *Vehicle routing*, In: M.O. Ball, T.L. Magnanti, C.L. Monma, G.L. Nemhauser, editors. Network routing, Handbooks in Operations Research and Management Science, Amsterdam: North-Holland, 1995: 1–33.
- [5] F. Foord, *Gambia: evaluation of the mobile health care service in West Kiang district*, World Health Statistics Quarterly 1995; 48: 18–22.
- [6] M. Gendreau, A. Hertz, G. Laporte, *New insertion and postoptimization procedures for the traveling salesman problem*, Operations Research, Vol. 40, No. 6. (Nov. – Dec., 1992), pp. 1086–1094.
- [7] M. Gendreau, G. Laporte, F. Semet, *The covering tour problem*, Operations Research, Vol. 45, No. 4. (Jul. – Aug., 1997), pp. 568–576.
- [8] B. E. Gillet, L. R. Miller, *A heuristic algorithm for the vehicle dispatch problem*, Operations Research, Vol. 22, No. 2. (Mar. – Apr., 1974), pp. 340–349.
- [9] M. Hachicha, M. J. Hodgson, G. Laporte, F. Semet, *Heuristics for the multi-vehicle covering tour problem*, Computers and Operations Research 27 (2000) 29–42.
- [10] M. J. Hodgson, G. Laporte, F. Semet, *A covering tour model for planning mobile health care facilities in Suhum district, Ghana*, Journal of Regional Science 1998; 38: 621–38.

- [11] M. Labbé, G. Laporte, *Maximizing user convenience and postal service efficiency in post box location*, Belgian Journal of Operations Research, Statistics and Computer Science 1986; 26: 21–35.
- [12] G. Laporte, *The vehicle routing problem: an overview of exact and approximate algorithms*, European Journal of Operational Research 1992; 59: 345–58.
- [13] J. K. Lenstra, A. H. G. Rinnooy Kan, *Some simple applications of the travelling salesman problem*, Operational Research Quarterly (1970–1977), Vol. 26, N°4, Part 1.(Nov., 1975), pp. 717–733.
- [14] W. A. Oliveira, *Construção de rotas para patrulhamento urbano preventivo*, Tese de Mestrado - Universidade Estadual de Campinas 2008.
- [15] J. R. Oppong, M.J. Hodgson, *Spatial accessibility to health care facilities in Suhum district, Ghana*, The Professional Geographer 1994; 46: 199–209.
- [16] J. C. Simms, *Fixed and mobile facilities in dairy practice*, Veterinary Clinics of North America - Food Animal Practice 1989; 5: 591–601.
- [17] W. Swaddiwudhipong, C. Chaovakiratipong, P. Nguntra, P. Lerdlukanavongse, S. Konchote, *Effect of a mobile unit on changes in knowledge and use of cervical cancer screening among rural Thai women*, International Journal of Epidemiology 1995; 24: 493–8.