

**Problema de Roteamento de Veículo Suficientemente Próximo Assimétrico:
Formulação e Heurística****Close-Enough Vehicle Routing Problem Asymmetric: Formulation and
Heuristics**

DOI:10.34117/bjdv5n12-140

Recebimento dos originais: 07/11/2019

Aceitação para publicação: 10/12/2019

Lettiery D'Lamare Portela Procópio

Doutorando em Ciências da Computação pela Universidade Federal do Rio Grande do Norte

Instituição: Universidade Federal do Rio Grande do Norte

Endereço: Rua Eugênio Carneiro Monteiro, Nº 51, AP 2020 - Bancários, João Pessoa – PB, Brasil

E-mail: lettiery@ppgsc.ufrn.br

Marco Cesar Goldberg

Pós-doutorado pela Universidade Federal de Minas Gerais.

Instituição: Universidade Federal do Rio Grande do Norte

Endereço: Departamento de Informática e Matemática Aplicada (DIMAp), Universidade Federal do Rio Grande do Norte (UFRN), Campus Universitário - Lagoa Nova, Natal - RN, Brasil.

E-mail: gold@dimap.ufrn.br

Elizabeth Ferreira Gouvêa Goldberg

Pós-doutorado em Engenharia de Sistemas e Computação, COPPE-Sistemas Universidade Federal do Rio de Janeiro

Instituição: Universidade Federal do Rio Grande do Norte

Endereço: Departamento de Informática e Matemática Aplicada (DIMAp), Universidade Federal do Rio Grande do Norte (UFRN), Campus Universitário - Lagoa Nova, Natal - RN, Brasil.

E-mail: beth@dimap.ufrn.br

RESUMO

Este trabalho propõe a versão assimétrica do Problema de Roteamento de Veículo Suficientemente Próximo, utilizado para planejamento de rotas de reconhecimento aéreo. O problema foi formulado com um modelo de programação cônica de segunda ordem e para solucioná-lo foram aplicadas técnicas de otimização heurística baseadas em uma propriedade geométrica. Também são descritos os resultados de experimentos computacionais extensivos de 240 instâncias adaptadas da literatura com até 20 pontos de observação. Destas, é possível encontrar resultados ótimos para 101 instâncias e os primeiros limites superiores para outras. Além disso, elaborou-se uma heurística baseada em clusterização, que utiliza propriedades geométricas em conjunto com o VNS. Os testes mostraram que o método proposto produz soluções de alta qualidade.

Palavras-chave: Roteamento de Veículos; Suficientemente Próximo; Heurística Geométrica.

ABSTRACT

This paper proposes the Asymmetric Version of the Close-Enough Vehicle Routing Problem, used to plan aerial reconnaissance routes. The problem was formulated with a second order conic programming model and to solve it we applied heuristic optimization techniques based on a geometric property. Also described are the results of extensive computational experiments of 240 instances adapted from the literature with up to 20 observation points. From these, you can find optimal results

for 101 instances and the first upper bounds for others. In addition, a heuristic based on clustering was developed, which uses geometric properties in conjunction with VNS. The tests showed that the proposed method produces high quality solutions.

Keywords: Vehicle Routing; Close Enough; Geometric Heuristics.

1 INTRODUÇÃO

A criação de rotas para veículos executarem demandas específicas deve ser adequada à finalidade da aplicação, e as particularidades de cada tipo de rota dão origem a variações do Problema de Roteamento de Veículo (PRV). Tendo em vista essas variações, um estudo e classificação das nomenclaturas usadas para o PRV podem ser encontrados em Goldberg et al. (2016). Com a popularização dos veículos aéreos não tripulados (VANTs), surge um novo cenário que necessita do roteamento de um ou mais veículos para alcançar um objetivo comum. A utilização de VANTs pode ser observada em diversos setores, como na vigilância e segurança de grandes terrenos, monitoramento de vias de trânsito, na entrega e coleta de produtos, no setor militar, e ainda no monitoramento de áreas agrícolas. Há casos em que a capacidade restrita de deslocamento dos veículos é um fator que necessita de uma etapa de planejamento do percurso para que se cumpram restrições inerentes à cada tipo de problema.

O Problema de Roteamento de Veículo Suficientemente Próximo (Close-Enough Vehicle Routing Problem, CEVRP), apresentado por W. K. Mennell (2009), foi inspirado na necessidade de rotear VANTs para monitoramento de áreas em um plano euclidiano. Neste caso, o veículo precisa alcançar certa distância do centro do alvo para realizar serviços, tais como fotografar, gravar vídeo, etc. W. K. Mennell (2009) formulou o CEVRP como um modelo não linear e não convexo, e mostrou que para o caso em que as distâncias das demandas ao centro do alvo são reduzidas a zero, o CEVRP é transformado no VRP no espaço euclidiano e, portanto, um problema NP-difícil.

As forças que interferem na rota dos veículos também estão presentes no ambiente de utilização dos VANTs, como por exemplo, o vento. Isto pode tornar o esforço que o veículo faz para se deslocar do ponto i a um ponto j diferente do esforço que ele tem para fazer o caminho inverso, caracterizando uma situação de assimetria. A modelagem feita no trabalho de W. K. Mennell (2009) não contempla essa característica do problema. Portanto, neste trabalho é proposto uma formulação para o Problema de Roteamento de Veículos Suficientemente Próximos Assimétrico (A-CEVRP). Também se descreve as primeiras heurísticas para esta variante do problema.

No A-CEVRP, assim como na sua versão simétrica, o roteamento dos veículos é realizado no espaço euclidiano e os pontos são dados por coordenadas cartesianas. Nos dois problemas o veículo não necessita visitar o ponto para realizar a demanda, mas deve se aproximar a uma distância suficientemente próxima. O veículo parte e retorna a um ponto base. Além disso, há uma restrição do comprimento máximo para o percurso efetuado. Quando a frota de veículo é heterogênea, o

comprimento máximo do percurso assume valores distintos de acordo com o veículo disponibilizado. Este trabalho, investiga-se o caso assimétrico com frota heterogênea.

O modelo apresentado para o A-CEVRP é implementado em um Solver e são apresentados os ótimos para 101 instâncias. Este trabalho apresenta uma heurística baseada em clusters e busca local. Como não existe registro de problemas com as mesmas variáveis do A-CEVRP, criamos um conjunto de 240 instâncias para testes. Tais instâncias possuem até 20 vértices. Os resultados obtidos com a heurística são comparados com valores do Solver.

Na Seção 2 mostra-se trabalhos que abordam problemas semelhantes ao A-CEVRP. Em seguida, na Seção 3 apresenta-se a formulação A-CEVRP. A heurística proposta é descrita na Seção 4. Os resultados dos testes computacionais são explorados na Seção 5. E por fim, a conclusão do trabalho na Seção 6.

2 PROBLEMAS CORRELATOS

Esta seção apresenta problemas de roteamento que também possuem a característica “suficientemente próximo”. Cada cliente do problema de roteamento é representado por uma figura geométrica, onde a demanda pode ser satisfeita com a visita do veículo a qualquer ponto de sua área.

2.1 TSPN

O Problema do Caixeiro Viajante com Vizinhança (Traveling Salesman Problem with Neighborhood, TSPN), introduzido por Arkin & Hassin (1994), é uma variante do clássico Problema do Caixeiro Viajante. No TSPN, os n clientes são representados por figuras geométricas. Uma solução para o problema consiste em um ciclo hamiltoniano mínimo sobre pontos presentes em todas as n regiões. O custo de deslocamento é dado pela distância euclidiana entre todos os pontos do passeio, entretanto, há casos em que o problema adota funções assimétricas não negativas para valorar o comprimento das arestas. de Berg et al. (2005) apresentaram um algoritmo aproximativo para casos em que as regiões de visitação do TSPN são disjuntas e convexas, além de um algoritmo polinomial que encontra o valor ótimo para casos em que as regiões de visitação são sobrepostas e não convexas. O problema, neste caso, é classificado como APX-difícil. No trabalho de Gentilini et al. (2013) foram mostradas duas formulações com distância dada pela norma euclidiana, seus modelos foram testados no Solver não linear COIN-OR (2016) em instâncias com até 16 pontos. Para alguns problemas testes fez-se necessário mais de quatro horas de processamento em um computador Intel Xeon 3.33 GHz com 12GB de memória RAM.

2.2CETSP

O Problema do Caixeiro Viajante Suficientemente Próximo (Close-Enough Traveling Salesman Problem, CETSP) é um caso especial do TSPN, no qual as regiões são demarcadas por

círculos ou esferas. Foi formulado por W. K. Mennell (2009) através de equações não lineares, em que as regiões de visitação são dadas por n circunferências centradas nas coordenadas (a_i, b_i) com raios r_i , $i = 1, \dots, n$. Como solução, é requisitado um ciclo hamiltoniano mínimo sobre pontos representativos (x_j, y_j) pertencentes às circunferências. W. K. Mennell (2009) mostra que para o caso em que todos os raios são iguais a zero, o CETSP é resumido ao TSP euclidiano e classificado como NP-difícil.

Gulczynski et al. (2006) propuseram diversos métodos heurísticos para casos em que as circunferências têm o mesmo raio, incluindo a Steiner Zone Heuristic posteriormente aprimorada por W. Mennell et al. (2011). Nela, três etapas são apresentadas para criação da solução. Na primeira etapa, as áreas de interseções entre os círculos de demanda (zonas de Steiner) são identificadas e enumeradas por seu grau de interseção, sendo 0 para círculos que não têm interseção, 1 para círculos com uma interseção, e assim sucessivamente. Na segunda etapa, é construída uma rota criando pontos baseados no grau de cada zona de interseção. Por fim, um rearranjo da sequência dos pontos refina a solução. Como a Steiner Zone Heuristic é baseada em interseções dos círculos, o banco de instâncias utilizado por Gulczynski et al. (2006), W. K. Mennell (2009) e W. Mennell et al. (2011) foi criado com sobreposição de círculos (Steiner Zone) em todos os casos, o que nem sempre acontece em aplicações reais.

O problema do roteamento de robôs em uma rede de sensores sem fio com variação do raio de alcance de cada cliente foi modelado por Yuan et al. (2007) como um CETSP. Eles apresentaram um método de permutação de sequência da rota que modifica a rota gulosa inicialmente criada nos centros dos círculos. Posteriormente, um algoritmo genético altera as coordenadas dos postos para minimizar o percurso realizado pelo robô.

Devido à complexidade do CETSP, Behdani & Smith (2014) propuseram duas discretizações para o problema chamadas de Grid-based Partition e Arc-based Partition. Na Grid-based Partition, a região da demanda é substituída por pequenos quadrados criando uma grade no círculo original. Os possíveis pontos de visita, são definidos pelos vértices da grade. A Arc-based Partition utiliza uma heurística de secção dos círculos para espalhar de forma arbitrária pontos sobre algumas regiões da circunferência. Tais pontos são base para criação da solução.

Coutinho et al. (2016) apresentaram um modelo cônico de segunda ordem para utilizar o método enumerativo branch-and-bound no CETSP. Os testes são realizados em 824 instâncias e comprovam a eficácia do algoritmo em cenários de duas e três dimensões.

Carrabs et al. (2017) apresentaram uma discretização para o CETSP, dividindo os círculos de demandas em regiões selecionadas por um algoritmo que analisa possíveis rotas. Essa discretização tanto apresenta limites inferiores, como superiores para o problema. Além disso, foi aplicado um

algoritmo de redução do grafo, que restringiu o tamanho do problema e acelerou a computação dos limites. Os testes computacionais foram realizados em um banco de 240 instâncias criadas para o CETSP.

2.3 CEVRP

W. K. Menell (2009) apresentou o CEVRP como uma alternativa multiveículos para o CETSP. Neste problema, k veículos são roteados por n círculos de demanda, respeitando o comprimento máximo de rota, Q . Todos os veículos devem sair e voltar para a base n_0 . W. K. Menell (2009) utilizou a Steiner Zone Heuristic para solucionar 48 problemas testes e comparar com valores obtidos pelo Solver GAMS do MATLAB.

3 A-CEVRP

Ao utilizar o CEVRP para rotear VANTs autônomos, compreendemos que a modelagem proposta por W. K. Menell (2009) não aborda algumas características primordiais para tal tarefa. A principal dificuldade foi percebida no deslocamento assimétrico do veículo, desconsiderado por sua formulação. Também foi identificada a exigência imposta naquele modelo de utilizar todos os veículos disponíveis no depósito com a mesma capacidade de deslocamento (frota homogênea). Isso não necessariamente é verdade para todos os cenários. Então, surgiu a necessidade de formular um problema que compreenda essas especificações. Alguns requisitos observados na aplicação do problema em casos reais são listados a seguir:

- Proximidade do alvo: para realizar a demanda em cada ponto, o veículo necessita aproximar-se do centro do alvo à uma distância r mínima.
- Capacidade de deslocamento do veículo: cada veículo k possui capacidade de deslocamento limitada por Q_k .
- Deslocamento assimétrico: devido à resistência do vento sobre os veículos, o esforço de deslocamento do ponto i para j , que pode ser representado como a distância entre i e j , pode ser diferente no deslocamento de j para i .

A habilidade de realizar demanda em um ponto próximo do alvo é uma particularidade de alguns problemas derivados do TSPN Arkin & Hassin (1994). A diferença do TSPN para o problema tratado por este artigo, é devido à forma de cada figura (círculos, quadrados, polígonos e outras) que delimita a área de demanda. No caso do A-CEVRP existe apenas a variação do raio das circunferências que delimitam as áreas que devem ser visitadas pelo veículo.

Em diversos problemas de roteamento existem capacidades predeterminadas para os veículos. Na maioria dos cenários esse valor é dado pela distância máxima de uma rota. Outra particularidade

a ser considerada é a capacidade heterogênea da frota de veículos envolvidos e a não obrigatoriedade de utilizar toda a frota.

O problema apresentado por W. K. Mennell (2009) foi formulado para o caso de cenários totalmente controlados, sem a interferência de forças externas. Em tais cenários, a distância percorrida no deslocamento de um veículo de um ponto i para outro j é igual no sentido contrário.

No cenário tratado neste texto existe interferência de forças no deslocamento realizado pelo veículo. Assim, no A-CEVRP a distância percorrida no deslocamento de um veículo de um ponto i para outro j pode ser diferente no sentido contrário. A gravidade, por exemplo, é uma interferência presente na natureza, porém, em princípio já é considerada no voo. Por outro lado, no caso das forças devidas ao vento é aplicada uma carga positiva (vento de cauda) ou negativa (vento de proa) ao veículo durante o deslocamento, caracterizando um percurso assimétrico.

A formulação do CEVRP apresentada por W. K. Mennell (2009) não pode ser resolvida por Solvers de programação linear, pois sua modelagem envolve o cálculo da distância euclidiana com a utilização de raiz quadrada. Uma alternativa proposta por Coutinho et al. (2016) para o CETSP transforma a modelagem não linear apresentada por W. K. Mennell (2009) em uma formulação cônica de segunda ordem. Essa proposta serviu de base para formular o A-CEVRP como um modelo cônico de segunda ordem.

3.1 FORMULAÇÃO

No A-CEVRP, V veículos partem de um único depósito e visitam n circunferências de raios r_i , centradas nas coordenadas (a_i, b_i) . Qualquer rota que toca ou passa pela circunferência c_i , atende a demanda do nó i . Vários veículos podem passar pela circunferência c_i , mas a demanda será realizada uma única vez.

Seja (x_i, y_i) um ponto da rota de um veículo que esteja na circunferência c_i . A coordenada (x_i, y_i) é dita como ponto representativo do nó i . Qualquer um dos pontos pertencentes ao deslocamento do veículo que corta a circunferência c_i pode ser selecionado como ponto representativo da área i . O deslocamento do ponto i para j do veículo k é dado por um segmento de reta que une os dois pontos.

3.1.1 Influência dos ventos

Para a inclusão da perturbação dos ventos no modelo, fez-se necessário estimar a interferência b_{ij} aplicada aos veículos (sem considerar a velocidade) no deslocamento s_{ij} do ponto i para j . b_{ij} é um parâmetro de entrada do problema que determina a influência do vento no deslocamento s_{ij} . Como não existe instâncias para o A-CEVRP, calculamos os valores de b_{ij} para servir como dado de entrada do problema.

A álgebra vetorial é utilizada para o cálculo do deslocamento (considerando vento e veículo), direção e sentido (vento de calda ou de proa). O deslocamento do veículo com a interferência do

vento, s_{ij}^{vento} , é dado na equação (I), onde s_{ij} é o vetor deslocamento do veículo sem nenhuma interferência, w é o vetor que representa a interferência do vento e β o ângulo formado entre eles.

$$s_{ij}^{\text{vento}} = s_{ij}^2 + w^2 + 2 s_{ij} w \cos \beta \quad (I)$$

Com o valor de s_{ij}^{vento} calculado é possível encontrar a interferência que o vento b_{ij} aplica no deslocamento do veículo $b_{ij} = s_{ij} - s_{ij}^{\text{vento}}$. Seu valor (b_{ij}) é normalizado em uma escala de 0,5 até 1,5, o que representa no melhor caso (0,5) a redução de 50% no impacto da distância, e no pior caso (1,5) um aumento de 50% na interferência no deslocamento. Esses valores podem ser ajustados de acordo com a característica dos veículos.

A representação de b_{ij} é apresentado por Mandle & Goudon (1989) e Slattery & Zhao (1997) como velocidade em relação ao solo (groundspeed). Para calcular seu valor são necessários os valores da direção e intensidade do vento, direção e deslocamento do veículo.

O valor de b_{ij} pode ser calculado considerando uma inferência da direção e intensidade do vento presentes no local. Mas se for possível coletar mudanças nesses valores em pontos distintos do mapa, por sensores presentes no local ou durante o voo de reconhecimento, pode-se recalculá-lo o valor de impacto b_{ij} para criar rotas mais adequadas à situação atual. Esses valores são dados como parâmetros de entrada do problema que interferem diretamente no custo da solução.

O custo do deslocamento no A-CEVRP é dado pela multiplicação da distância euclidiana d_{ij} pela interferência dos ventos b_{ij} . A exemplo de Coutinho et al. (2016), inserimos a variável de decisão auxiliar d_{ij} , que demarca a distância (sem influência do vento) do ponto i para j . O valor de d_{ij} é dado pela equação $d_{ij}^2 = (x_i - x_j)^2 + (y_i - y_j)^2$ inserida como restrição no modelo.

A variável de decisão binária e_{ijk} assume o valor 1 se o veículo k realiza o deslocamento do ponto i até o ponto j , e 0 caso contrário. Com e_{ijk} é possível garantir a continuidade do fluxo da rota para o veículo k , assim também como a proibição de subtours nas rotas. e_{ijk} ainda auxilia na verificação do cumprimento da demanda w_{ik} pelo veículo k no nó i .

Para restringir o valor do percurso total do veículo utiliza-se a variável de decisão Z_{ijk} , que demonstra o valor da distância de i para j feita pelo veículo k com influência dos ventos. Temos $Z_{ijk} = b_{ij} d_{ij}$ quando o veículo k realiza o trajeto do ponto i para j ($e_{ijk} = 1$), e 0 caso contrário. A soma de todos os valores de Z_{ijk} de um veículo k representa o custo total da rota do veículo k . O custo da rota deve ser restringido por sua capacidade Q_k .

A lista dos dados de entrada para o modelo matemático do A-CEVRP é dada a seguir:

- n quantidade de círculos de demanda.
- V quantidade de veículos disponíveis.

- (a_i, b_i) coordenadas cartesianas do ponto i .
- r_i raio do ponto i .
- Q_k distância máxima permitida para a rota do veículo k .
- b_{ij} interferência dos ventos no deslocamento do ponto i para o ponto j .

As variáveis de decisão utilizadas no modelo, são listadas a seguir:

- $w_{ik} = \begin{cases} 1, & \text{se o veículo } k \text{ atende a demanda do ponto } i. \\ 0, & \text{caso contrário.} \end{cases}$
- $e_{ijk} = \begin{cases} 1, & \text{se o veículo } k \text{ sai do ponto } i \text{ para o ponto } j. \\ 0, & \text{caso contrário.} \end{cases}$
- u_i ordem em que o ponto i é visitado (restrição de subtours de Miller et al. (1960)).
- x_i coordenada cartesiana x de um ponto suficientemente próximo de i .
- y_i coordenada cartesiana y de um ponto suficientemente próximo de i .
- d_{ij} distância entre o ponto i até o ponto j , sem a influência dos ventos.
- z_{ijk} distância que o veículo k percorre do ponto i até o ponto j com a interferência do vento (b_{ij}), 0 caso i, j não esteja na rota de k .

$$\text{Min} \sum_i^n \sum_j^n \sum_k^V z_{ijk} \quad (\text{II})$$

$$(x_i - a_i)^2 + (y_i - b_i)^2 \leq r_i^2 \quad \forall i \in N \quad (\text{III})$$

$$(x_i - x_j)^2 + (y_i - y_j)^2 = d_{ij}^2 \quad \forall i \neq j \in N \quad (\text{IV})$$

$$d_{ij} e_{ijk} b_{ij} = z_{ijk} \quad \forall i, j \in N, e \forall k \in V \quad (\text{V})$$

$$\sum_i^n \sum_j^n \sum_k^V z_{ijk} \leq Q_k \quad \forall k \in V \quad (\text{VI})$$

$$\sum_j^n e_{ijk} = \sum_j^n e_{jik} = w_{ijk} \quad \forall i, j \in N, e \forall k \in V \quad (\text{VII})$$

$$\sum_k^V w_{ik} = 1 \quad \forall i \neq 0 \in N \quad (\text{VIII})$$

$$\sum_k^V w_{0k} \geq 1 \quad \forall k \in V \quad (\text{IX})$$

$$u_i - u_j + n \cdot e_{ijk} \leq n - 1 \quad \forall i \neq j \in N e \forall k \in V \quad (\text{X})$$

$$u_i \leq n - 2 \quad \forall i \in N \quad (\text{XI})$$

$$e_{ijk}, w_{ik} \in \{0,1\}; u_i \in \mathbb{Z}^+; x_i, y_i, d_{ij}, z_{ijk} \in \mathbb{R} \quad \forall i, j \in N e \forall k \in V \quad (\text{XII})$$

A função objetivo (II) minimiza a distância percorrida por todos os veículos para o monitoramento dos n pontos. A restrição (III) garante que o ponto (x_i, y_i) seja suficientemente

próximo de (a_i, b_i) , uma vez que a distância máxima para o ponto ser considerado suficientemente próximo é definida por r_i . Em (IV) o cálculo da distância entre i e j é realizado e atribuído à variável de decisão d_{ij} .

O valor do deslocamento de i até j com a influência dos ventos b_{ij} quando o veículo k efetua esse trajeto é atribuído a z_{ijk} na restrição (V). A restrição (VI) garante que o comprimento total da rota de cada veículo não ultrapasse o limite Q_k . A restrição (VII) assegura que quando um veículo k entra em um círculo i ele deve continuar sua rota para um ponto j até retornar à base. A restrição (VIII) determina que todo ponto i deve ser atendido por um veículo k . A restrição (IX) diz que um ou mais veículos partem do depósito. As restrições (X) e (XI) são de proibição de subtours definidas conforme o trabalho de Miller et al. (1960). Por fim, a restrição (XII) estabelece os tipos das variáveis do problema.

4 CLUSTERIZAÇÃO GEOMÉTRICA

Na literatura do VRP é possível encontrar diversos trabalhos com métodos heurísticos e exatos, contudo, em casos que o tempo computacional pode levar ao sucesso ou fracasso da aplicação e soluções subótimas são aceitáveis, métodos heurísticos se apresentam como escolhas assertivas. Muitos dos métodos heurísticos aplicados ao VRP realizam agrupamento de vértices e logo em seguida criam rotas atribuídas aos veículos específicos Goldberg et al. (2016). A ordem das fases agrupar e rotear determina duas classes de algoritmos para o problema: primeiro agrupar e depois rotear e primeiro rotear e depois agrupar. A clusterização geométrica, heurística proposta por esse trabalho para solucionar o A-CEVRP, pertence à primeira classe, isto é, agrupamento como primeira fase. Uma revisão sobre alguns métodos e trabalhos que também utilizam essa estratégia pode ser encontrada no trabalho de Prins et al. (2014).

Dada a particularidade do problema, os algoritmos de clusterização necessitam de uma fase extra para encontrar os pontos representativos da circunferência (x_i, y_j) . Ao confrontar as estratégias apresentadas pelos trabalhos mencionados na seção 2, observa-se a necessidade de uma heurística que, ao clusterizar, também cria possíveis pontos representativos utilizando propriedades geométricas.

Atrelado à propriedade euclidiana de sobreposição de esferas, pode-se inferir as coordenadas dos pontos em que dois círculos se interceptam. Para alcançar esses valores utiliza-se a definição básica da circunferência em um espaço euclidiano $c = \{(x, y) \mid (x - x_c)^2 + (y - y_c)^2 \leq r^2, r > 0\}$. A partir daí, tem-se um conjunto de pontos que estão à distância r do ponto central (x_c, y_c) . A superfície f_c deste objeto é formada pelos pontos cuja distância ao centro é igual a r , $f_c = \{(x, y) \mid (x - x_c)^2 + (y - y_c)^2 = r^2\}$. O conjunto de pontos da sobreposição de dois círculos possui

infinitos pontos. Mas, ao observar a interseção das superfícies de dois círculos com centros distintos, tem-se um conjunto com no máximo dois pontos $|f_i \cap f_j| \leq 2$. Existem três casos possíveis na sobreposição de duas circunferências: o conjunto de interseção entre suas superfícies é vazio $|f_i \cap f_j| \in \{\emptyset\}$; a distância entre os centros das circunferências é igual a soma dos raios $f_c = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} = r_i + r_j$, logo suas superfícies se sobrepõem em um único ponto; e quando a distância entre elas é menor do que a soma dos seus raios, suas superfícies se cruzam em dois vértices distintos. As coordenadas dos pontos de interseção são dadas pela solução do sistema de equação do segundo grau, criado com a equação reduzida das duas circunferências $(x - x_{ci})^2 + (y - y_{ci})^2 = r^2$ e $(x - x_{cj})^2 + (y - y_{cj})^2 = r^2$. A solução deste sistema nos leva a uma equação na forma $ax^2 + bx - c = 0$. Ao encontrar $\Delta = b^2 - 4.a.c$, pode-se inferir as três distintas condições de sobreposição: se $\Delta < 0$, as superfícies não se sobrepõem; para $\Delta = 0$, as circunferências se tocam em um único ponto; e $\Delta > 0$, existe dois pontos presentes na interseção das superfícies.

São criadas h circunferências fictícias que representarão os centros do cluster (centroides). A quantidade de centroides presentes no algoritmo é limitada ao número de veículos disponíveis, pois o número de h centroides representará a quantidade de rotas criadas. Mas o valor de h é alterado aleatoriamente ao longo das interações do algoritmo. Uma estratégia de crescimento de raios dos centroides é utilizada para gerar colisões com círculos de demanda. As etapas da clusterização geométrica são apresentadas no Algoritmo I.

Algoritmo I Clusterização Geométrica

Entrada: g - quantidade de iterações do algoritmo

V - quantidade de veículos

Q - capacidades dos veículos

C - circunferências

Saída: s – solução

```

1:    $h \leftarrow \text{inicializacaoCentroides}(V, C)$ 
2:   para  $e = 0 \leq g$  faça
3:        $LC \leftarrow \text{crescimento}(h, C)$ 
4:        $s \leftarrow \text{roteamento}(LC, V, Q)$ 
5:        $\text{buscalocal}(s)$ 
6:       se  $e \neq g$  faça
7:            $h \leftarrow \text{reposiciona}(h)$ 
8:       fim se
9:   fim for

```

Para o funcionamento do Algoritmo I é necessário conhecer a quantidade V de veículos disponíveis para o roteamento e suas capacidades, dadas por Q . O mapa das solicitações de demanda com os pontos centrais e raios das circunferências estão presentes no conjunto de circunferências C .

Na primeira fase do algoritmo (linha 1), as coordenadas dos h centroides são definidas pela divisão do mapa de demandas C em quadrantes. Essa etapa utiliza as maiores distâncias das circunferências de demanda presentes em C e divide pelo número de veículos disponíveis $|V|$. Cada (x_c, y_c) do centroide c é escolhido pela coordenada central do quadrante de índice h . O raio r_c é calculado pela menor distância entre o centroide h e a superfície da circunferência mais próxima. Ao longo das iterações, a quantidade de centroides é reduzida de forma aleatória respeitando a restrição da capacidade dos veículos.

Na linha 3 os raios dos centroides são alterados para a existência de colisões com os círculos de demanda. Quando essa colisão ocorre pela primeira vez em uma circunferência de demanda i , o centroide c que a atingiu é dito dominante da circunferência i . A função de crescimento dos centroides é dada pela equação (XIII). (x_c, y_c, r_c) são dados do centroide c que estão no quadrante h , (x_i, y_i, r_i) são as coordenadas e o raio da circunferência i , e $rand$ é um fator aleatório entre 0 e 1 com distribuição normal que interfere diretamente na colisão das circunferências. Além do cliente mais próximo, pode ocorrer que o crescimento do centroide atinja outras circunferências. Isso acontece devido ao fator aleatório $rand$ estar presente no crescimento de r_c .

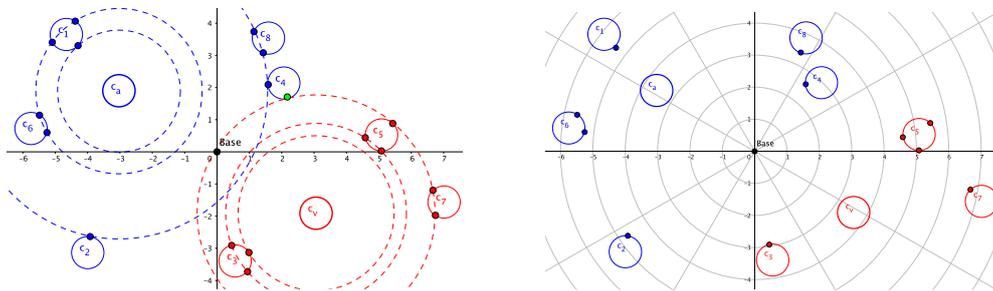
$$f(x_c, y_c, r_c) = \sqrt{(x_c - x_i)^2 + (y_c - y_i)^2} - (r_i * rand) + r_c \quad (XIII)$$

Os pontos formados pelas interseções entre centroides e círculos de demanda são armazenados em LC para a criação das futuras rotas. É possível acontecer colisões em circunferências que já estejam dominadas por outros centroides, nesses casos são desconsiderados os pontos de colisão entre os círculos já dominados. O crescimento dos centroides finaliza quando todas as circunferências estiverem relacionadas a um centroide.

A terceira etapa do algoritmo (linha 4) cria as rotas baseadas na heurística de Gillett & Miller (1974). Nela, os pontos são transformados em coordenadas polares (t, θ) , com origem na base (depósito) dos veículos. t é a distância do ponto i à origem (base) do plano, e θ é o ângulo formado entre o eixo x e o segmento de reta partindo da origem até o ponto pertencente a LC. Este valor é calculado com a função trigonométrica $\theta = \arctan\left(\frac{y}{x}\right)$, onde x e y são as diferenças das coordenadas da origem até o ponto. Posteriormente à conversão, os pontos de LC são ordenados pelos valores de θ e agrupados por proximidade de ângulos.

Figura I. Clusterização Geométrica

(a) Expansão dos raios dos centroides (b) Seleção dos pontos baseado em Gillett & Miller (1974)

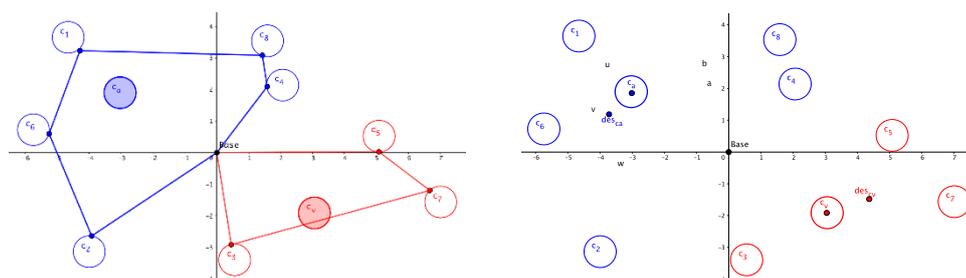


O procedimento da linha 4 cria rotas de forma iterativa partindo de todos os valores de LC para o cluster de índice h . A inserção do ponto i na a rota de índice h só será permitida se ainda não existir pontos que cubram a demanda de circunferência c_i . As duas etapas presentes nas linhas 3 e 4 do Algoritmo I são exemplificadas, respectivamente, pelas Figuras 1 (a) e (b).

Na Figura I (a) é possível observar o crescimento dos raios a partir dos centroides C_a e C_v , com as cores azul e vermelho respectivamente. Os pontos na superfície dos círculos clientes (c_1, c_2, \dots, c_8) representam as colisões do raio do centroide. O ponto verde na circunferência c_4 é ignorado devido à colisão tardia do centroide C_v . Uma transformação nas coordenadas dos pontos auxilia na aplicação da heurística de Gillett & Miller (1974), exemplificado na Figura I (b). Na última etapa a base é incluída para finalizar a criação da solução, Figura II (a).

Figura II. Reposicionamento dos Centroides

(a) Rotas Criadas (b) Atração dos centroides



Depois das etapas que criam as rotas, a solução inicial passa pelo processo de busca local descrito na seção 4.1. Em seguida acontece o deslocamento dos centroides para uma nova posição que melhor represente seu cluster. O método de reposicionamento dos centroides é baseado no cálculo do centro de massa das circunferências presentes em cada cluster, exemplificado no conjunto de equações (XIV). As novas coordenadas (x_c, y_c) do centroide de índice h são dadas pela influência das massas das circunferências pertencentes ao seu cluster.

O conjunto de equações (XIV) calcula o centro de massa de cada cluster da solução. Nesse cálculo m_i representa a massa da figura i , e (x_i, y_i) é a coordenada do centro de i . No A-CEVRP as figuras são dadas por círculos e m_i calculado com $2\pi r^2$. O conjunto de equações (XIV) é determinístico, e pode ocorrer de o centroide permanecer na mesma posição. Para esse caso, os valores de x_h e y_h são multiplicados por um valor aleatório entre 0 e 1. O processo de clusterização geométrica é repetido por g iterações. Quando o número de soluções criadas for igual ao limite (g), o algoritmo finaliza com a melhor solução encontrada.

$$x_h = \frac{\sum_i^{V_c} m_i x_i}{\sum_i^{V_c} m_i} \quad (a)$$

(XIV)

$$y_h = \frac{\sum_i^{V_c} m_i y_i}{\sum_i^{V_c} m_i} \quad (b)$$

4.1 BUSCA LOCAL

Após a criação das rotas, são utilizadas estruturas de vizinhanças de forma sistemática para explorar o espaço de soluções viáveis. Esta tarefa é aplicada para refinar a solução criada pelo algoritmo de clusterização geométrica. As estruturas de vizinhanças usadas nesse refinamento exploram duas características do problema: a sequência dos pontos e os valores das coordenadas.

A primeira estrutura de vizinhança utiliza a premissa de que, para toda reta u que atravessa uma circunferência i , existe um ou dois pontos que pertence simultaneamente à superfície da circunferência t_i e a reta u , $1 \leq |t_i \cap u| \leq 2$. A estrutura de vizinhança cria uma reta que passa pelo centro do círculo i e calcula os pontos de interseção da reta com a superfície da circunferência. As coordenadas destes pontos são dadas pela solução do sistema de equação do segundo grau constituído com a equação geral da reta $ax + by + c = 0$ e com a equação reduzida circunferência $(x - x_i)^2 + (y - y_i)^2 = r_i^2$, x_i , y_i e r_i são dados da circunferência i .

Outra estrutura de vizinhança é obtida por retas tangentes às circunferências que partem de qualquer ponto p_e externo às circunferências de demanda i . As coordenadas de p_e assumem valores de centros de outros círculos de demanda e de seus pontos representativos. Existem duas retas que passam por p_e e tangenciam a superfície do círculo i . Para encontrar os pontos tangentes da circunferência i , cria-se um círculo virtual que passa pelos pontos p_e e o centro de i . O raio do círculo virtual r_v é duas vezes a distância entre os pontos p_e e o centro de i . O centro c_v do círculo virtual é o ponto médio entre p_e e c .

Há então, novamente, um sistema de equações de segundo grau formado pelas equações paramétricas de duas circunferências centradas em (x_i, y_i) e (x_{cv}, y_{cv}) com raios r_i e r_v ,

respectivamente. A solução desse sistema nos dará as coordenadas dos pontos tangentes. Essa estrutura também é utilizada para encontrar pontos de interseção entre duas circunferências secantes s_i e s_j . Elas são identificadas quando a distância dos centros é menor do que a soma dos seus raios.

É utilizado o método 2-opt Gendreau et al. (1992) como estrutura de vizinhança para otimizar a sequência das rotas. Partindo de um ciclo hamiltoniano p_1, p_2, \dots, p_n , escolhe-se um par de arestas para remoção do ciclo, $(p_i - p_j), (p_k - p_l)$. Logo em seguida, o ciclo hamiltoniano é religado com a nova configuração $(p_i - p_l), (p_k - p_j)$. Se a nova configuração minimizar o comprimento da rota, os novos pares de arestas serão utilizados na nova solução, caso contrário, o processo é reiniciado com um novo conjunto de arestas.

A partir desses conceitos, é utilizada a heurística Variable Neighborhood Descent (VND) Mladenovic & Hansen (1997) para o refinamento da solução. VND é uma heurística de refinamento que busca melhorar a solução por uma troca sistemática de estruturas. Seu método é diferente de outras buscas locais por explorar diferentes estruturas na solução corrente.

Partindo de uma solução inicial s , que doa as características para que se encontre um melhor local, cria-se um vizinho s' para realizar uma busca local e melhorar a solução atual. Isso nos traz uma nova solução s'' , a qual torna-se a nova base para o VND se seu custo for menor do que a solução original s . Essa condição reinicia o loop das estruturas de vizinhanças, caso contrário é utilizada uma nova estrutura para realizar o processo, que só é finalizado quando a busca local para todas as vizinhanças conhecidas não melhorar a solução s base. Para o A-CEVRO utiliza-se a estrutura 2-opt primeiro, e logo em seguida as coordenadas dos pontos representativos são recalculadas com as estruturas geométricas. Essa ordem foi obtida por testes com diferentes instâncias e configurações.

5 EXPERIMENTOS

O modelo descrito na seção 3 foi implementado no software IBM ILOG CPLEX Optimization Studio 12.4. Todo código deste trabalho foi desenvolvido na linguagem de programação C++, e os testes executados em um computador com processador Intel Core i5 2,3 GHz, com 8 GB de memória RAM. Foram realizadas 20 execuções independentes para a heurística apresentada no Algoritmo I e nomeada de CG nesta seção.

Uma vez que não existem instâncias benchmark para o A-CEVRP foram adaptadas as instâncias do CEVRP disponibilizadas por Behdani & Smith (2014), seguindo metodologia análoga ao seu trabalho. No caso do A-CEVRP, duas variáveis não estão presentes nas instâncias: a capacidade Q_k dos veículos, que é gerada a partir da solução criada por um algoritmo guloso que escolhe sempre o círculo mais próximo para o veículo visitar; e a interferência dos ventos b_{ij} é calculada como descrito na seção 3, com o módulo de $w = 10$ e 180° em relação ao eixo cartesiano x .

O conjunto de instâncias, os métodos para criação dos valores de b e Q_k , estão disponíveis em LAE-UFRN (2017).

O parâmetro g de repetição da clusterização geométrica foi determinado por treinamento através da ferramenta IRACE López-Ibáñez et al. (2011), com 50 iterações em 10 instâncias excluídas dos resultados finais. Tal parâmetro recebe o valor de $g = 200$, o que representa a criação de duzentas soluções em cada execução do algoritmo.

As instâncias testes são divididas em 8 grupos, com 6, 8, 10, 12, 14, 16, 18, e 20 pontos. Cada conjunto tem 30 problemas testes, totalizando 240 instâncias para o problema. Devido à capacidade Q_k , são disponibilizados dois veículos para os conjuntos de 6, 8 e 10 pontos. Nos conjuntos 12 e 14 existem três veículos à disposição, e para as instâncias com 16 e 18 pontos estão disponíveis quatro veículos para o roteamento. Por fim, o conjunto de 20 clientes tem a disponibilidade de cinco veículos.

Os resultados apresentados pela heurística de clusterização geométrica são identificados pelas colunas demarcadas com CG. Estes valores são comparados com o Solver que tem um tempo limite de 4000 segundos. Em instâncias acima de 10 pontos o limite de tempo não foi suficiente para a finalização, e os valores de lower e upper bounds obtidos pelo Solver são apresentados respectivamente nas colunas LB e UB das Tabelas 4, 5, 6, 7 e 8. O índice de cada instância é apresentado na coluna Inst. As colunas Cost e T(s) indicam os valores da solução final e do tempo de processamento do Solver. O melhor valor encontrado pela heurística está visível na coluna Best, e a média do custo das 20 execuções de cada instância é apresentada em Av. O tempo médio de processamento em segundos da heurística é mostrado na coluna Tav(s).

Nas Tabelas I e II estão os resultados dos experimentos realizados com 6 e 8 pontos, respectivamente. No conjunto com 6 clientes, o afastamento médio das melhores soluções obtidas pela heurística (Best) em relação ao Solver foi 0,37%. No conjunto de instâncias com 8 clientes, o afastamento médio das melhores soluções obtidas pela heurística em relação Cost do Solver foi 0,22%. Ao analisar os resultados do Solver e Av, pode-se observar um afastamento de 0,63% e 1,39%, respectivamente para os conjuntos 6 e 8. O tempo dos dois algoritmos tem um afastamento de 61,06% e -42,31% nos conjuntos 6 e 8 pontos, respectivamente. Neste caso, o valor negativo representa que CG é mais rápido que o Solver.

Tabela I Resultados para instâncias com 6 pontos.

Inst	Solver		CG			Inst	Solver		CG		
	Cost	T(s)	Best	Av	T _{av} (s)		Cost	T(s)	Best	Av	T _{av} (s)
06	18,29	1,92	18,29	18,30	3,40	06	27,12	1,11	27,25	27,25	1,17

1	21,76	1,49	21,89	21,89	3,04	16	18,56	2,13	18,73	18,73	1,70
2	17,91	2,72	17,92	17,92	2,35	17	17,93	1,48	18,02	18,72	3,85
3	19,35	1,48	19,42	19,42	1,24	18	24,20	1,10	24,22	24,22	2,65
4	29,51	1,08	29,55	29,95	2,63	19	25,00	0,99	25,23	25,23	3,56
5	17,08	1,36	17,17	17,57	0,97	20	27,27	1,66	27,30	27,30	3,25
6	23,96	2,23	23,98	23,98	0,90	21	24,42	1,76	24,46	24,46	1,84
7	24,85	1,31	24,89	24,89	1,22	22	24,18	2,45	24,33	24,33	2,37
8	15,79	1,98	15,86	16,06	1,56	23	20,03	0,91	20,13	20,13	1,84
9	21,32	1,80	21,38	21,38	2,04	24	21,57	1,20	21,73	21,73	2,32
10	28,32	1,13	28,36	28,36	3,42	25	24,23	0,66	24,28	24,28	4,54
11	28,91	2,78	28,96	28,96	2,15	26	23,96	1,83	24,18	24,18	1,06
12	23,76	2,18	23,84	23,84	1,35	27	23,28	0,39	23,40	23,40	1,39
13	18,51	1,17	18,63	18,63	2,89	28	29,32	0,88	29,45	29,45	3,33
14	25,41	1,13	25,47	25,47	3,31	29	28,72	0,47	28,79	28,92	5,03

Tabela II Resultados para instâncias com 8 pontos.

Inst	Solver		CG			Inst	Solver		CG		
	Cost	T(s)	Best	Av	T _{av} (s)		08	Cost	T(s)	Best	Av
0	22,50	13,29	22,50	22,52	7,39	15	24,09	14,93	24,12	24,12	12,67
1	28,57	5,64	28,59	28,96	5,54	16	38,24	14,59	38,24	38,64	8,59
2	23,44	27,26	23,47	23,70	4,89	17	21,60	1,67	21,79	22,02	5,07
3	35,21	8,32	35,23	35,78	7,28	18	23,36	2,84	23,57	23,58	3,06
4	20,11	2,11	20,19	20,22	10,28	19	27,38	4,12	27,43	27,61	7,32
5	22,29	5,28	22,49	22,77	5,50	20	25,09	10,78	25,22	25,76	7,93
6	18,41	3,49	18,59	19,01	8,32	21	23,97	1,94	24,01	25,24	6,61
7	31,21	25,40	31,32	31,33	7,10	22	31,14	8,55	31,19	31,19	11,90
8	27,72	10,71	27,76	27,78	6,60	23	29,65	11,98	29,72	29,75	5,01
9	43,33	17,54	43,33	44,29	6,27	24	25,65	11,36	25,67	25,69	5,75
10	22,05	19,95	22,09	22,10	8,83	25	35,46	8,36	35,49	35,54	5,01
11	27,85	17,08	27,86	31,82	7,29	26	26,34	8,60	26,37	26,45	6,70
12	39,93	37,76	39,98	39,99	7,40	27	25,69	8,58	25,76	25,81	8,89
13	28,13	14,45	28,13	28,12	5,79	28	24,34	13,72	24,51	24,61	4,97
14	38,45	17,51	38,45	38,60	5,05	29	35,72	6,52	35,74	36,73	5,25

A Tabela III mostra os resultados dos testes para instâncias com 10 pontos. Existe um afastamento médio de 0,27% entre os valores do Solver e Best. Já ao compararmos Cost e a média das

execuções AV, tem-se 2,18% de afastamento médio. A heurística encontra seus resultados em média 80% mais rápido do que o Solver.

Tabela III Resultados para instâncias com 10 pontos.

Inst	Solver		CG			Inst	Solver		CG		
	Cost	T(s)	Best	Av	T _{av} (s)		10	Cost	T(s)	Best	Av
0	29,04	177,48	29,10	29,30	31,21	15	22,46	22,58	22,60	26,93	18,37
1	28,57	143,76	28,65	28,73	37,21	16	23,96	54,11	23,98	23,99	29,45
2	37,54	37,29	37,57	37,58	15,57	17	27,04	101,86	27,09	28,33	29,24
3	25,01	520,62	25,06	25,09	27,49	18	29,34	57,12	29,38	29,60	25,32
4	28,42	171,89	28,53	28,87	24,78	19	35,90	143,35	36,00	36,10	22,71
5	27,55	149,52	27,67	28,01	17,42	20	26,32	151,43	26,34	26,40	24,08
6	29,12	176,41	29,16	29,24	30,39	21	30,27	55,52	30,31	30,77	26,21
7	36,66	113,07	36,72	36,75	18,84	22	29,71	269,30	29,78	29,82	35,28
8	24,43	69,08	24,59	24,70	30,80	23	26,41	74,52	26,48	26,59	17,52
9	23,50	34,58	23,52	25,03	27,05	24	30,28	157,91	30,33	30,56	26,49
10	26,46	231,26	26,49	26,50	32,56	25	29,08	168,04	29,15	29,49	31,27
11	23,40	75,39	23,43	23,62	32,80	26	28,49	129,91	28,57	28,62	33,24
12	27,13	340,70	27,16	27,17	36,13	27	22,35	105,40	22,52	27,37	21,79
13	30,15	128,85	30,31	31,28	26,37	28	30,82	213,51	30,82	30,82	30,92
14	30,23	253,38	30,38	30,51	21,05	29	37,39	153,36	37,72	38,00	20,75

Nos conjuntos de 12, 14, 16, 18 e 20 pontos, o Solver não foi capaz de finalizar sua execução no tempo limite de 4000 segundos. Por isso, nas Tabelas IV, V, VI, VII e VIII são apresentados os limites no momento de sua interrupção (UB e LB). E nas instâncias com 16, 18 e 20 pontos, há casos onde o Solver não apresenta limites para a solução. Essas condições são demarcadas por - nas tabelas. Na Tabela IV, a heurística obteve 5 melhores valores do que o UB do Solver nas instâncias 4, 5, 9, 11 e 21. Esses valores estão em itálico na coluna Best. CG finalizou sua execução em um tempo de processamento 98% menor do que o Solver. Nesse conjunto (12 pontos) o afastamento médio calculado entre UB e LB é de 43,60%, e entre UB e Best é de -0,50%. Já para UB e Av o afastamento é de 0,58%.

Tabela IV Resultados para instâncias com 12 pontos.

Inst	Solver			CG			Inst	Solver			CG		
	LB	UB	T(s)	Best	Av	T _{av} (s)		12	LB	UB	T(s)	Best	Av
0	29,50	29,50	921,12	29,55	29,57	64,89	15	31,48	33,65	4000,00	33,88	33,98	68,08
1	30,65	30,66	4000,00	30,67	30,82	67,60	16	37,62	37,62	2979,18	37,72	36,74	50,45

2	19,88	48,51	4000,00	48,66	49,02	62,26	17	14,34	31,95	4000,00	32,13	32,54	65,17
3	26,22	32,45	4000,00	32,53	33,16	50,02	18	27,40	27,40	2672,79	30,74	30,79	55,62
4	28,51	28,51	3647,18	26,39	28,03	64,22	19	29,92	29,93	3461,22	30,00	30,65	68,92
5	19,45	39,29	4000,00	39,12	39,18	49,03	20	13,60	40,58	4000,00	41,78	41,35	47,93
6	14,87	31,94	4000,00	31,99	32,58	65,99	21	14,08	43,73	4000,00	40,65	40,79	57,87
7	19,63	43,82	4000,00	43,86	44,05	51,29	22	17,81	28,85	4000,00	29,02	30,57	64,76
8	19,59	29,72	4000,00	29,76	31,22	51,55	23	20,24	33,70	4000,00	33,81	33,10	47,85
9	19,43	35,44	4000,00	35,13	35,45	84,91	24	29,82	29,82	1515,07	30,00	30,05	61,75
10	16,93	32,64	4000,00	32,78	33,00	58,23	25	19,97	43,22	4000,00	43,24	42,24	55,53
11	14,06	36,23	4000,00	29,82	33,81	57,52	26	25,97	25,98	989,12	26,03	26,06	52,99
12	23,92	23,92	1704,94	24,07	25,21	48,34	27	13,47	25,96	4000,00	26,27	26,75	67,46
13	27,13	27,13	364,54	27,18	27,55	49,68	28	18,95	27,26	4000,00	27,29	27,39	52,55
14	35,68	35,68	3034,20	35,79	33,91	47,70	29	39,41	39,41	524,36	39,69	40,87	57,28

CG foi capaz de obter cinco novos limites para o conjunto de instâncias com 14 pontos nas instâncias 13, 14, 15, 16 e 29 (Tabela V). A heurística CG apresentou um afastamento médio de -0,20% em relação aos UB do Solver, e 1,40% de afastamento entre UB e a média das 20 execuções AV. O Solver finalizou sua execução no tempo limite de 4000 segundos em todas as instâncias dos conjuntos 14, 16, 18 e 20, por isso a coluna com esses valores foi excluída das Tabelas V, VI, VII e VIII. As soluções de CG foram obtidas em um tempo de processamento 96,78% menor do que o Solver. O afastamento entre os limites do Solver para os conjuntos 14, 16, 18 e 20, aproxima-se de 99,98%.

Tabela V Resultados para instâncias com 14 pontos.

Inst	Solver		CG			Ins t	Solver		CG		
14	LB	UB	Best	Av	Tav(s)	14	LB	UB	Best	Av	Tav(s)
0	12,27	29,55	29,69	29,69	127,38	15	12,25	33,59	32,80	32,80	125,28
1	8,30	29,32	29,52	29,52	110,36	16	9,73	54,36	48,14	48,14	121,77
2	12,36	35,45	36,36	36,36	123,94	17	7,48	31,37	35,82	35,82	98,76
3	10,47	27,37	27,62	27,62	165,35	18	10,29	32,16	32,40	32,40	111,83
4	9,49	34,37	34,61	34,61	98,77	19	7,31	27,20	27,82	27,82	132,06
5	8,13	34,19	34,61	34,61	134,03	20	7,49	29,15	29,35	29,35	144,04
6	6,55	26,85	27,73	27,73	108,3	21	7,38	32,04	32,25	32,25	117,96

					2							
7	11,10	37,66	37,94	37,94	105,4	22	14,89	35,87	37,18	37,18	149,73	
					2							
8	7,59	39,94	38,23	38,23	160,3	23	8,75	32,32	32,77	32,77	120,05	
					4							
9	9,87	29,21	30,94	30,94	118,4	24	10,86	33,77	34,04	34,04	144,62	
					5							
10	7,36	25,67	26,25	26,25	119,7	25	10,21	31,40	31,45	31,45	144,67	
					8							
11	11,04	32,66	33,02	33,02	130,6	26	9,80	37,44	39,49	39,49	132,51	
					8							
12	6,61	28,96	33,26	33,26	104,3	27	9,84	33,92	34,19	34,19	105,10	
					3							
13	10,70	35,99	35,70	35,70	132,8	28	7,94	29,08	29,85	29,85	130,05	
					1							
14	8,66	35,87	37,66	37,66	124,2	29	10,69	26,01	25,86	25,86	208,49	
					1							

No conjunto com 16 (Tabela VI) pontos o Solver não consegue limites válidos para cinco instâncias e, além desses casos, CG apresenta doze novos limites (em *itálico*) para a literatura do problema. Best possui um afastamento médio de -2,77% do limite UB em um tempo de processamento 94% menor, e 1,58% de afastamento entre os valores de Best e AV.

Tabela VI Resultados para instâncias com 16 pontos.

Inst	Solver		CG			Inst	Solver		CG		
	LB	UB	Best	Av	Tav(s)		16	LB	UB	Best	Av
0	-	-	36,37	36,48	179,19	15	2,53	44,36	37,19	37,34	162,32
1	4,40	33,83	34,10	36,80	202,78	16	-	-	36,66	36,78	199,78
2	7,82	33,29	33,84	34,23	193,59	17	2,88	25,24	25,37	25,40	347,04
3	4,17	40,54	40,60	40,64	188,08	18	7,75	40,61	35,04	35,28	152,87
4	4,94	35,20	35,33	35,39	197,50	19	7,75	42,93	41,73	41,92	259,76
5	2,39	33,18	33,24	33,35	231,71	20	4,37	40,91	40,67	40,92	206,23
6	-	-	40,14	42,51	290,21	21	3,67	41,75	37,04	38,38	199,58
7	2,27	37,37	37,19	37,58	298,36	22	5,33	32,53	32,61	32,40	275,78
8	2,97	30,09	30,33	30,72	246,28	23	4,56	44,25	42,31	42,78	276,09
9	5,22	23,59	23,88	24,75	236,66	24	6,21	29,34	30,29	31,22	284,43
10	3,67	30,01	28,55	29,00	220,09	25	5,65	35,69	33,59	33,99	149,23
11	4,79	39,26	36,92	37,71	200,88	26	4,70	37,11	36,99	37,06	280,53
12	-	-	55,24	55,92	236,67	27	-	-	41,94	41,95	155,50
13	6,57	35,84	35,95	37,71	241,05	28	8,03	37,04	39,70	40,51	191,83

14 10,04 37,99 | 36,72 36,92 143,49 | 29 5,94 45,47 | 43,08 44,21 238,72

Para os valores da Tabela VII o Solver finalizou sua execução no tempo limite sem apresentar resultados válidos em 17 instâncias do conjunto com 18 pontos. Best consegue um afastamento médio de -6,20 em relação ao UB do Solver, em um tempo de processamento médio 90% mais rápido do que o Solver. A média das execuções está AV da heurística, tem um afastamento médio de 4,72% em relação aos melhores valores encontrados em CG. O Solver apresenta 8 novos limites para as instâncias desse conjunto.

Tabela VII Resultados para instâncias com 18 pontos.

Inst	Solver		CG			Inst	Solver		CG		
	LB	UB	Best	Av	Tav(s)		LB	UB	Best	Av	Tav(s)
0	5,08	30,37	29,67	31,18	364,33	15	1,42	38,39	26,35	30,81	421,00
1	-	-	34,89	35,54	322,53	16	-	-	40,71	40,91	556,93
2	-	-	31,53	38,20	396,15	17	-	-	45,94	45,32	284,16
3	-	-	35,53	43,62	324,47	18	-	-	42,27	42,39	374,87
4	4,61	39,14	32,88	33,51	598,70	19	-	-	39,81	40,82	370,53
5	-	-	37,83	38,07	444,34	20	2,76	39,43	35,40	37,14	487,92
6	-	-	31,05	36,91	480,63	21	-	-	36,26	36,03	349,26
7	2,19	27,08	31,68	36,51	341,34	22	-	-	30,62	35,24	358,47
8	2,25	27,04	26,12	27,55	353,60	23	3,87	35,35	35,03	34,89	542,50
9	2,75	32,30	32,30	32,86	462,24	24	-	-	41,83	42,70	270,20
10	5,88	38,07	38,33	38,66	373,47	25	-	-	40,21	41,95	342,73
11	-	-	47,73	48,56	296,01	26	3,73	40,15	37,16	39,52	338,24
12	7,24	28,25	28,57	29,00	536,99	27	2,85	36,77	31,05	31,15	332,03
13	-	-	38,68	39,64	410,78	28	2,66	33,53	33,65	34,31	365,67
14	-	-	40,76	40,83	303,49	29	-	-	38,22	38,84	302,53

No último conjunto de testes com 20 pontos (Tabela VIII), o Solver apresenta limites para apenas 6 instâncias e todas são superadas por Best. UB e Best tem um afastamento médio de -10,66%, em um tempo 84% menor do que o Solver. Best ainda consegue um afastamento médio de 4,68% para a média das 20 execuções independentes AV.

Tabela VIII Resultados para instâncias com 20 pontos.

Inst	Solver		CG			Inst	Solver		CG		
	LB	UB	Best	Av	Tav(s)		LB	UB	Best	Av	Tav(s)
0	-	-	34,64	34,89	836,93	15	-	-	35,02	40,80	709,29

1	-	-	39,42	42,92	464,49	16	1,75	42,96	35,74	36,26	531,67
2	-	-	33,03	33,27	668,55	17	-	-	36,58	37,63	602,30
3	-	-	41,50	42,08	479,24	18	-	-	40,45	41,08	400,07
4	-	-	42,68	42,37	551,66	19	-	-	39,37	39,55	705,67
5	-	-	46,35	47,15	511,08	20	-	-	44,06	44,30	602,37
6	1,98	46,28	38,61	40,04	729,14	21	-	-	35,20	35,65	662,83
7	-	-	40,72	43,45	716,16	22	-	-	36,91	48,65	451,38
8	-	-	30,80	31,54	585,64	23	-	-	38,97	39,08	631,49
9	-	-	37,47	38,37	631,11	24	2,28	46,75	42,94	44,72	619,37
10	1,37	40,14	39,68	39,87	634,26	25	-	-	43,24	45,76	578,14
11	-	-	36,32	38,97	701,98	26	4,95	44,94	41,97	42,49	339,11
12	-	-	41,63	42,28	455,36	27	-	-	40,79	41,70	581,48
13	-	-	37,93	43,44	519,21	28	2,18	46,84	40,40	42,39	516,24
14	-	-	39,73	40,76	704,24	29	-	-	29,64	34,81	561,19

A clusterização geométrica mostrou-se eficiente em comparação com os valores apresentados pelo Solver, além de apresentar novos limites. A heurística apresenta um afastamento de -3,02% na média de todos os conjuntos de instâncias testadas. Devido à característica não linear do problema, o Solver não apresenta soluções para todas as instâncias. Esse cenário também foi descrito nos trabalhos relacionados Coutinho et al. (2016), Behdani & Smith (2014) e Mitchell (2007). Portanto, tem-se a clusterização geométrica como melhor método disponível na literatura para resolução do problema.

6 CONCLUSÃO

O A-CEVRP foi formulado de acordo com a necessidade de rotear uma frota heterogênea de VANTs para o monitoramento de grandes áreas. A existência da interferência dos ventos foi um dos principais diferenciais apresentados por esse problema. A primeira modelagem matemática do problema mostrada na seção 3.1, foi resolvida pelo Solver IBM CPLEX que obteve 111 soluções ótimas para instâncias de até 12 pontos. Seus valores serviram de parâmetro para comparação com o método de clusterização geométrica com busca local VND.

A utilização da clusterização geométrica como método para criar soluções do A-CEVRP apresentou um afastamento menor do que 0,4% para os valores do Solver. A heurística ainda foi capaz de mostrar 90 novos limites para nas instâncias com 14, 16, 18 e 20 pontos. No conjunto com 20 pontos, a heurística superou todos os valores apresentados pelo Solver. Além da formulação matemática, foi criado um banco de 240 instâncias testes para o problema.

A clusterização geométrica permite adaptações de acordo com a característica do problema ou a escolha da pesquisa. Pode-se facilmente substituir o método de Gillett & Miller (1974) por um segundo algoritmo. Ainda outros fatores podem ser adaptados, como o tratamento de duas colisões em uma mesma circunferência. Pode-se adaptar também a interferência dos valores (vento, proximidade de outros centroides, proximidade da base, etc.) no deslocamento dos centroides. Acredita-se que a aplicação de outras heurísticas de agrupamento podem ser utilizada para otimizações do problema em conjunto com o método de clusterização geométrica, tais como ant colony optimization, tabu search, GRASP, etc.

REFERÊNCIAS

- Arkin, E. M., & Hassin, R. (1994). Approximation algorithms for the geometric covering salesman problem. *Discrete Applied Mathematics*, 55(3), 197–218.
- Behdani, B., & Smith, J. C. (2014). An integer-programming-based approach to the close-enough traveling salesman problem. *INFORMS Journal on Computing*, 26(3), 415–432.
- Carrabs, F., Cerrone, C., Cerulli, R., & Gaudioso, M. (2017). A novel discretization scheme for the close enough traveling salesman problem. *Computers & Operations Research*, 78, 163–171.
- COIN-OR. (2016). Computational infrastructure for operations research project. Recuperado em 17 de janeiro de 2018, de <http://www.coin-or.org/>.
- Coutinho, W. P., Nascimento, R. Q. d., Pessoa, A. A., & Subramanian, A. (2016). A branch-and-bound algorithm for the close-enough traveling salesman problem. *INFORMS Journal on Computing*, 28(4), 752–765.
- de Berg, M., Gudmundsson, J., Katz, M. J., Levkopoulos, C., Overmars, M. H., & van der Stappen, A. F. (2005). Tsp with neighborhoods of varying size. *Journal of Algorithms*, 57(1), 22–36.
- Gendreau, M., Hertz, A., & Laporte, G. (1992). New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research*, 40(6), 1086–1094.
- Gentilini, I., Margot, F., & Shimada, K. (2013). The travelling salesman problem with neighbourhoods: Minlp solution. *Optimization Methods and Software*, 28(2), 364–378.
- Gillett, B. E., & Miller, L. R. (1974). A heuristic algorithm for the vehicle-dispatch problem. *Operations research*, 22(2), 340–349.
- Goldbarg, M. C., Goldbarg, E. F. G., & Luna, H. P. L. (2016). *Otimização combinatória e metaheurísticas: Algoritmos e aplicações*. Rio de Janeiro - RJ, Brasil: Editora Elsevier.
- Gulczynski, D. J., Heath, J. W., & Price, C. C. (2006). The close enough traveling salesman problem: A discussion of several heuristics. *Perspectives in Operations Research*, 271–283.

- LAE-UFRN. (2017). A-CEVRP - instances. Recuperado em 17 de janeiro de 2018, de <http://www.dimap.ufrn.br/lae/en/projects/ACEVRP.php>.
- López-Ibáñez, M., Dubois-Lacoste, J., Stützle, T., & Birattari, M. (2011). The irace package: Iterated racing for automatic algorithm configuration. IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, Tech. Rep. TR/IRIDIA/2011-004.
- Mandle, J., & Goudon, J.-C. (1989, May 9). Method for determining the air speed of a helicopter, system for carrying on this method and method for calibrating such air speed determining method and system. Google Patents. (US Patent 4,829,441).
- Mennell, W., Golden, B., & Wasil, E. (2011). A steiner-zone heuristic for solving the close-enough traveling salesman problem. In 2th INFORMS computing society conference: Operations Research.
- Mennell, W. K. (2009). Heuristics for solving three routing problems: Close-enough traveling salesman problem, close-enough vehicle routing problem, sequence-dependent team orienteering problem (Tese de doutorado) University of Maryland, Maryland.
- Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7(4), 326–329.
- Mitchell, J. S. (2007). A PTAS for TSP with neighborhoods among fat regions in the plane. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (pp. 11-18). Society for Industrial and Applied Mathematics.
- Mladenovic, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & operations research*, 24(11), 1097–1100.
- Prins, C., Lacomme, P., & Prodhon, C. (2014). Order-first split-second methods for vehicle routing problems: A review. *Transportation Research Part C: Emerging Technologies*, 40, 179–200.
- Slattery, R., & Zhao, Y. (1997). Trajectory synthesis for air traffic automation. *Journal of Guidance Control and Dynamics*, 20, 232–238.
- Yuan, B., Orłowska, M., & Sadiq, S. (2007). On the optimal robot routing problem in wireless sensor networks. *IEEE Transactions on Knowledge and Data Engineering*, 19(9), 1252–1261.