LIII Simpósio Brasileiro de Pesquisa Operacional
João Pessoa - PB, 3 a 5 de novembro de 2021

Uma heurística baseada em GRASP-VND para o Problema de Roteamento de Mula de Dados em Redes Mistas

Igor Morais
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca - CEFET/RJ
igor.morais@eic.cefet-rj.br

Pablo Luiz Araújo Munhoz
Universidade Federal de Viçosa - UFV
Campus Rio Paranaíba
pablo.munhoz@ufv.br

Marcos A. Guerine
Instituto Federal de Educação, Ciência e Tecnologia do Rio de Janeiro - IFRJ
Campus Arraial do Cabo
marcos.ribeiro@ifrj.edu.br

Pedro Henrique González
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca - CEFET/RJ
pegonzalez@eic.cefet-rj.br

RESUMO

As redes de sensores sem fio são usadas em muitas aplicações devido ao seu baixo custo e fácil implantação. Em redes de sensores esparsas, é impossível que os sensores se comuniquem apenas usando mensagens *multi-hop*. Para esse tipo de problema, um sensor específico, chamado mula de dados, geralmente é usado para coletar dados dos sensores e levá-los até uma estação base. O presente trabalho analisa a estrutura do problema de roteamento de mulas de dados e propõe tanto um modelo de programação linear inteira, quanto uma meta-heurística GRASP-VND. Experimentos computacionais foram realizados com a abordagem proposta e os resultados mostram que o GRASP-VND pode ser usado para obter soluções viáveis em um tempo computacional aceitável, quando comparado com o modelo proposto.

PALAVRAS CHAVE. **Mula de Dados. Redes de Sensores Esparsas. GRASP-VND.**
OC – Otimização Combinatória. POI – PO na Indústria. MH – Metaheurísticas.

ABSTRACT

Wireless sensor networks are used in many applications due to their low cost and easy deployment. In sparse sensor networks, it is impossible for sensors to communicate using multiple hop messages only. For this type of problem, a specific sensor, called a data mule, is usually used to collect data from the sensors and take them to a base station. The present work analyzes the structure of the data mule routing problem and proposes a integer programming formulation and a GRASP-VND metaheuristic. Experiments are performed with the proposed approach, and the results show that GRASP-VND can be used to obtain feasible solutions in an acceptable computational time, when compared to the proposed formulation.

KEYWORDS. **Data Mule. Sparse Sensor Networks. Smart Cities. GRASP-VND.**
CO – Combinatorial Optimization. ORI – OR in Industry. MH – Metaheuristics.

1. Introdução

Problemas de otimização são encontrados em diversos setores, em geral procurando minimizar custos e maximizar lucros. No contexto das Cidades Inteligentes e Indústrias 4.0, se torna prioridade a eficiência de serviços prestados ao cidadão, uma logística efetiva de transporte de pessoas/bens de consumo e monitoramento inteligente de cenários de difícil monitoramento, o que motiva uma série de propostas acadêmicas para resolução desses problemas [Meijer e Bolívar, 2016; Li et al., 2017; Kamble et al., 2018]. Nesse cenário, cada vez mais, Redes de Sensores Sem Fio (RSF) têm se mostrado uma poderosa ferramenta para monitoramentos ambientais, militares e de regiões pós-desastre [Erdelj et al., 2017; Munhoz et al., 2019; Naser e Kodur, 2020]. Esse tipo de rede vem sendo utilizada devido ao seu baixo custo, fácil implantação e manutenção [Sugihara e Gupta, 2011a]. Em uma rede multi-ponto, devido à faixa de comunicação dos sensores da rede, os sensores são capazes de se comunicar entre si. No entanto, quando os sensores são esparsos, esse tipo de conexão é impossível [Sugihara e Gupta, 2011a]. Uma maneira de reunir os dados em uma RSF esparsa é através do uso de uma agente especial denominado mula de dados, que pode ser, dentre outros, um Veículo Aéreo Não-tripulado (VANT). O VANT costuma ser uma escolha interessante para assumir o papel de mula de dados devido a seus recursos computacionais e de armazenamento aprimorados [Cohen e Jones, 2020].

Diversos trabalhos apresentam o conceito de mula de dados [Sugihara e Gupta, 2011a; Munhoz et al., 2018, 2019], e alguns deles mostram maneiras de usar a mula de dados para coletar dados na interseção desses sensores, enquanto outros trabalhos usam mulas de dados para melhorar o alcance de redes ad-hoc. Essas redes dependem de que todos os sensores tenham interseções para que a comunicação seja feita ponto a ponto. Em Munhoz et al. [2018], duas meta-heurísticas são usadas para resolver o problema de agendamento de mulas de dados, onde a coleta de dados pode ser feita na interseção dos sensores: o GRASP com descida aleatória de vizinhança variável (RVND) e a busca em vizinhança variável (GVNS). Ainda em Munhoz et al. [2018], é importante ressaltar que o trabalho buscou minimizar o tempo gasto para coletar os dados de todos os sensores, mas outros objetivos podem ser facilmente incorporados. Por fim, em Munhoz et al. [2019], os autores propuseram uma formulação matemática usando programação inteira e uma heurística distribuída para resolver uma variante do problema de roteamento de mula de dados (PRMD), onde a mula de dados deve dar o menor número de passos possíveis para atender todos os nós da rede. Além disso, os autores também provaram que tal variante do PRMD pertence à classe de problemas NP-Difícil.

Para coletar todos os dados em uma rede de sensores esparsos de maneira eficiente, é necessário um planejamento cuidadoso da rota a ser executada pela mula de dados, o que pode ser feito de diversas formas. Para determinar tais rotas, define-se, neste trabalho, uma nova variante do Problema de Roteamento de Mula de Dados, onde apenas parte dos sensores precisariam ser visitados, uma vez que é permitido aos sensores se comunicarem. Neste trabalho, é feita uma definição matemática formal para o problema, e, para resolvê-lo, propõe-se tanto um modelo de programação inteira, quanto uma heurística baseada em GRASP-VND [Gendreau e Potvin, 2010].

O restante deste artigo está organizado da seguinte maneira. A Seção 2 apresenta a definição matemática da nova variante proposta e suas diferenças em relação as demais variantes na literatura. Na Seção 3, o método GRASP-VND proposto é descrito. Na Seção 4, o ambiente computacional, os problemas testes e os experimentos computacionais são apresentados, e os resultados obtidos são discutidos. Por fim, na Seção 5, as conclusões são feitas e alguns trabalhos futuros são apontados.

2. Problema de Roteamento de Mula de Dados em Redes Mistas

Existem diversas abordagens na literatura que tratam de problemas de coleta de dados em redes de sensores ad-hoc esparsas. A Tabela 1 destaca os principais trabalhos da literatura e as

características das variantes tratadas.

Tabela 1: PRMD na Literatura

Autores	Sensores Movimento	Mula de dados	Caracterísitcas	
			Dest.	Comunicação
Zhao e Ammar [2003]	Estático	Única	Sensores	Ponto a ponto
Zhao et al. [2005]	Estático	Múltipla	Sensores	Ponto a ponto
Bin Tariq et al. [2006]	Dinâmico	Única	Sensores	Ponto a ponto
Yuan et al. [2007]	Estático	Única	Base	Ponto a ponto
Ma e Yang [2007]	Estático	Única	Base	Ponto a ponto
Xing et al. [2008]	Estático	Única	Nós centrais	Multi-ponto
Rao et al. [2008]	Dinâmico	Única	Base	Multi-ponto
Ngai et al. [2009]	Estático	Múltipla	Base	Ponto a ponto
Sugihara e Gupta [2010]	Estático	Single	Base	Ponto a ponto
Sugihara e Gupta [2011b]	Estático	Única e Múltipla	Base	Multi-ponto
Wichmann et al. [2012]	Estático	Única	Nós centrais	Ponto a ponto
Ma et al. [2013]	Estático	Múltipla	Base	Ponto a ponto

Nas abordagens apresentadas em [Zhao e Ammar, 2003; Zhao et al., 2005], a mula necessita de visitar os nós para finalizar a comunicação. Em [Bin Tariq et al., 2006; Ma e Yang, 2007], a mula é utilizada para aumentar a comunicação em redes esparsas e, nesse caso, a mula necessita de visitar uma área intermediária e transmitir os dados para a uma parte conexa de sensores. Em [Xing et al., 2008; Rao et al., 2008; Wichmann et al., 2012], a coleta de dados é feito em nós que são nós centrais em redes de tipo multi-ponto e, nesse caso, a mula precisa visitar apenas o nó central desse subconjunto de nós centrais. O problema de [Sugihara e Gupta, 2011b] apresenta uma ideia de coletar os dados no raio de acesso daquele sensor, seja visitando o nó ou a aresta, o que faz com que o problema de visita aos nós seja reduzido. Neste trabalho, a Rede Mista possui a ideia de esparsidade na comunicação e em alguns sensores a comunicação pode ser feita via multi ponto. Isso faz com que possamos unir a ideia de Zhao e Ammar [2003] e Xing et al. [2008], utilizando assim uma estrutura que busca soluções em redes que possuem características muti ponto e ponto a ponto.

Assim, este trabalho introduz o Problema de Roteamento de Mula de Dados em Redes Mistas (PRMD_RM), que consiste em uma variante do PRMD onde os sensores da rede são capazes de se comunicar com parte dos sensores da rede e a coleta de dados ocorre apenas quando a mula de dados visita um sensor, ou quando visita um sensor que esteja no raio de comunicação do sensor do qual quer-se coletar os dados. Considerando o descrito, podemos dizer que a rede abordada nesta variante é uma rede mista onde parte dos sensores consegue se comunicar, e a outra parte, devido a esparsidade da rede, não. Assim sendo, dadas as peculiaridades da rede, busca-se encontrar o roteamento de menor custo (*i.e.*, menor distância) que colete todos os dados dos sensores. Essa variante difere da apresentada por [Sugihara e Gupta, 2011b], uma vez que nela a coleta dos dados é feita apenas ao se visitar um nó, diferentemente da de [Sugihara e Gupta, 2011b], onde a coleta pode ser feita tanto no nó, quanto na aresta, contanto que esteja dentro do raio do sensor.

Pode-se representar matematicamente uma instância do PRMD_RM como um grafo completo $G(V \cup \{b\}, A)$, onde V representa o conjunto de sensores, b representa o nó base de onde a mula de dados parte e para onde retorna e A representa o conjunto arestas que representam os possíveis caminhos entre pares de $v \in V' = V \cup \{b\}$. Além disso, cada sensor $v \in V$ possui um atributo raio que representa o alcance de sua comunicação. Considera-se que dois sensores são capazes de se comunicar caso exista interseção entre as áreas cobertas por seus raios de comunicação. Para cada aresta $(i, j) \in A$, é associada uma distância d_{ij} usada para calcular o custo do deslocamento entre (i, j) e, consequentemente, da rota final que a mula terá de percorrer caso essa aresta esteja

presente na solução. O PRMD_RM é representado por um grafo completo, pois a mula de dados é capaz de acessar qualquer sensor a partir de qualquer sensor. Considerando as características de uma instância do PRMD_RM, seja $C = \cup C_i$ o conjunto de todas as coberturas (C_i) de sensores do grafo de entrada e A_{C_i} o conjunto das arestas pertencentes ao menor ciclo que passa por todos os sensores contidos em $C_i \cup \{b\}$. Pode-se definir o PRMD_RM matematicamente como:

$$\text{PRMD_RM} = \{E_{C_i} \subseteq E \mid \sum_{(i,j)\in E_{C_i}} d_{ij} \leq \sum_{(i,j)\in E_{C_j}} d_{ij}; \forall C_j \subset C, \text{tal que } C_i \neq C_j\} \quad (1)$$

Com a compreensão matemática do problema, pode-se representa-lo através de um modelo de programação linear inteira. As variáveis $x_{ij}\ \forall (i,j) \in A$ e $y_i\ \forall i \in V$ representam, respectivamente, se uma aresta está na solução e se o nó i é visitado. A variável auxiliar $z_{ij}\ \forall (i,j) \in A$ é usada para restrições de fluxo com intuito de evitar os subciclos. O modelo é definido como apresentado abaixo:

$$\min \quad \sum_{(i,j)\in A} d_{ij}x_{ij} \quad (2)$$

$$\text{s.a.} \quad \sum_{j\in\bar{C}(i)\cup\{i\}} y_j \geq 1, \qquad \forall i \in V \setminus \{0\} \quad (3)$$

$$\sum_{j\in\delta^+(i)} x_{ij} = y_i, \qquad \forall i \in V \quad (4)$$

$$\sum_{j\in\delta^-(i)} x_{ji} = y_i, \qquad \forall i \in V \quad (5)$$

$$y_b = 1, \quad (6)$$

$$\sum_{j\in\delta^+(i)} z_{ij} = \sum_{j\in\delta^-(i)} z_{ji} + y_i, \qquad \forall i \in V \setminus \{0\} \quad (7)$$

$$z_{ij} \leq \sum_{l\in V} y_l, \qquad \forall (i,j) \in A \quad (8)$$

$$z_{ij} \leq |V| x_{ij}, \qquad \forall (i,j) \in A \quad (9)$$

$$\sum_{j\in\delta^+(0)} z_{0j} = y_0, \quad (10)$$

$$\sum_{j\in\delta^-(0)} z_{j0} = \sum_{l\in V} y_l \quad (11)$$

$$x_{ij} \in \{0,1\}, \qquad \forall (i,j) \in A \quad (12)$$

$$z_{ij} \in \mathbb{Z}^+, \qquad \forall (i,j) \in A \quad (13)$$

$$y_i \in \{0,1\}, \qquad \forall i \in V \quad (14)$$

A Função Objetivo (2) busca minimizar o custo total do ciclo usado para vistar os nós. As Restrições (3) representam que um sensor precisa ser visitado ou acessado pelos nós que estão no seu raio de comunicação, onde $\bar{C}$ representa o conjunto de sensores que . Já as Restrições (4) e (5) definem que se um sensor foi visitado ele precisa ter uma aresta incidente e uma aresta saindo do nó. As Restrições (6) garantem que a base é sempre visitada na rota. Já as Restrições (7 - 11) definem o fluxo de cada aresta e geram uma ordem em que os sensores devem ser visitados, e também são usadas para eliminação de sub-rotas. O domínio das variáveis x e y é binário, representando respectivamente as arestas utilizadas e se o nó foi visitado , e a variável z possui domínio nos números inteiros positivos, pois define a enumeração da ordem de uso das arestas.

Considerando as características do PRMD_RM, é possível fazer um redução usando o

Problema do Caixeiro Viajante [Laporte, 1992], o que indica que o PRMD_RM pertence a classe de problemas $\mathcal{NP}$-difícil.

3. Método de Solução: GRASP-VND

Considerando os trabalhos presentes na literatura listados anteriormente, o trabalho aqui proposto se diferencia dos demais uma vez que resolve uma versão diferente do PRMD, adicionando novas restrições ao contexto de estruturação do problema. A metodologia proposta usa uma terminologia de grafos para representar o problema e propõe uma heurística baseada na metaheurística GRASP, usando como componente de intensificação um VND [Gendreau e Potvin, 2010] para resolver o PRMD_RM.

Metaheurísticas podem ser definidas como abordagens genéricas que combinam heurísticas desenvolvidas especificamente para resolver problemas de otimização combinatória e possuem mecanismos que tentam evitar que a busca por novas soluções fique presa em ótimos locais, o que resulta em uma melhor exploração do espaço de busca. *Greedy Randomized Adaptive Search Procedure* (GRASP) é uma metaheurística multipartida, baseada em busca local, proposta por Feo e Resende [1995], e que já foi aplicada com sucesso a diversos problemas de otimização combinatória. O GRASP é composto por um processo iterativo, no qual cada iteração é dividida em duas fases: construção e busca local. A cada iteração, uma solução viável s é construída por um procedimento guloso e aleatório e avaliada por uma função objetivo $f(s)$. Em seguida, a solução s é submetida a uma busca local no espaço vizinho de soluções, com o objetivo de melhorá-la até encontrar um ótimo local. A melhor solução geral é mantida ao longo das iterações e retornada como resultado final. Nas seções a seguir, serão apresentadas as principais componentes usadas no desenvolvimento da heurística proposta.

3.1. Heurística Construtiva

No presente trabalho, a etapa de construção de uma solução viável para o PRMD_RM pode ser dividida em duas etapas. A primeira etapa busca encontrar um subconjunto de nós de V cujos elementos façam a cobertura completa de todos os nós. A segunda etapa visa encontrar a melhor (mais curta) rota que, saindo da base, passe uma única vez por todos os nós e retorne à base.

A primeira fase do construtivo é iterativa e baseada no procedimento guloso e aleatório clássico da metaheurística GRASP. Para o PRMD_RM, a ideia central da primeira etapa do construtivo é escolher os elementos que possuem mais nós cobertos para fazerem parte da rota inicialmente, diminuindo assim a quantidade de nós necessários para montar a rota de atendimento na segunda etapa. Dessa maneira, visa-se diminuir o esforço para construção da rota, uma vez que encontrar a rota de custo mínimo significa resolver um problema do caixeiro viajante, que é reconhecidamente difícil.

Inicialmente, todos os nós $i \in V$ que podem ser inseridos na solução são avaliados e ordenados de maneira decrescente de acordo com uma função gulosa $c(i)$, que basicamente retorna o grau daquele nó. A decisão de avaliar os nós pelo seu grau se deve ao fato de que quanto maior o grau de um nó, mais nós vizinhos a ele são atendidos pelo seu raio de atuação. Logo, escolher um nó de grau mais alto diminui a quantidade de elementos restantes para a criação da rota de atendimento. Após essa avaliação gulosa, uma lista denominada Lista Restrita de Candidatos (LRC) é criada com todos os elementos i que atendem ao seguinte critério:

$$c(i) \in \left[c^{max}, c^{min} + \alpha\left(c^{max} - c^{min}\right)\right], \tag{15}$$

com $\alpha \in [0, 1]$ e considerando c^{min} e c^{max}, respectivamente, os valores máximo e mínimo de grau dentre os elementos avaliados.

Após a composição da LRC, um dos elementos é escolhido aleatoriamente e inserido na solução. A cada nova inserção, é possível que existam nós já cobertos pelos nós já inseridos na solução. Esses nós já cobertos são excluídos do conjunto de possibilidades e, embora não façam parte da rota de atendimento, serão atendidos pelos nós da rota. Esse procedimento é repetido até que não existam elementos possíveis para serem inseridos e todos os nós estejam cobertos.

Com os elementos escolhidos na primeira fase, inicia-se a segunda fase da heurística construtiva. Nesse momento, deve-se encontrar a rota de menor custo saindo da base e passando por todos os elementos escolhidos na primeira fase. Para isso, optou-se por encontrar uma rota gulosa simples, baseada na heurística de vizinho mais próximo do TSP [Laporte, 1992]. O Algoritmo 1

Algoritmo 1: Heurística Construtiva

1: **ConstruçãoGulosaAleatória**(α, V)
2: $s \leftarrow \emptyset$
3: Avaliar $c(i)\ \forall\ i \in V$
4: **Enquanto** Todos os nós não forem atendidos **faça**
5: $LRC \leftarrow$ ConstroiLRC(α)
6: $i \leftarrow$ EscolhaAleatória(LRC)
7: $s \leftarrow s \cup \{i\}$
8: Avaliar $c(i)\ \forall\ i \in E \backslash s$
9: **Fim-enquanto**
10: $s \leftarrow$ ConstroiRotaVizinhoMaisPróximo(s)
11: **Retorne** s

representa o pseudocódigo da fase de construção do GRASP. A solução a ser construída s é inicializada como uma solução vazia (linha 2) e, após avaliar cada elemento i, $\forall\ i \in V$ (linha 3), constrói-se a LRC (linha 5) de acordo com a função $c(i)$ e o parâmetro α, presentes na Equação 15. Dentre os elementos da LRC, um é escolhido aleatoriamente (linha 6) e inserido na solução (linha 7). Esses passos (linhas 5, 6, 7 e 8) se repetem até que todos os nós sejam atendidos pelos nós presentes na solução. Finalmente, na linha 10, constrói-se uma rota com os nós da solução usando a política de inserção do vizinho mais próximo.

Após a construção, a solução s é submetida a uma fase de intensificação, constituída pela aplicação de estruturas de vizinhanças com o objetivo de melhorar a solução corrente. Todas essas estruturas de vizinhanças e o procedimento de intensificação serão apresentados na próxima seção.

3.2. Variable Neighborhood Descent (VND)

As vizinhanças combinadas no VND foram três vizinhanças clássicas da literatura [Munhoz et al., 2018]: Swap, Shift e 2-Opt. A estrutura de vizinhança N^{Swap} aplica um movimento no qual dois nós são selecionados em uma solução base (s), e suas posições são trocadas, gerando uma solução vizinha à s (s'). Já estrutura de vizinhança N^{Shift} aplica um movimento no qual um nó é selecionado em uma solução base (s), e ele é inserido entre outros dois nós, gerando uma solução vizinha à s (s'). Por último, a estrutura de vizinhança $N^{2\text{-}opt}$ aplica um movimento no qual dois nós são selecionados em uma solução base (s), e toda a rota é invertida à partir dos nós selecionados, gerando uma solução vizinha à s (s').

A heurística VND explora as vizinhanças em uma ordem predefinida (Swap, Shift e 2-Opt). Toda vez que uma exploração não é bem sucedida, a próxima vizinhança é explorada. Por outro lado, toda vez que uma vizinhança encontrar melhoria, todo o procedimento é reiniciado e a busca retorna à primeira vizinhança. Ao final do VND, uma solução é retornada como ótimo local com respeito à todas as vizinhanças [Gendreau e Potvin, 2010].

A Figura 1 exemplifica cada uma das vizinhanças na ordem de execução.

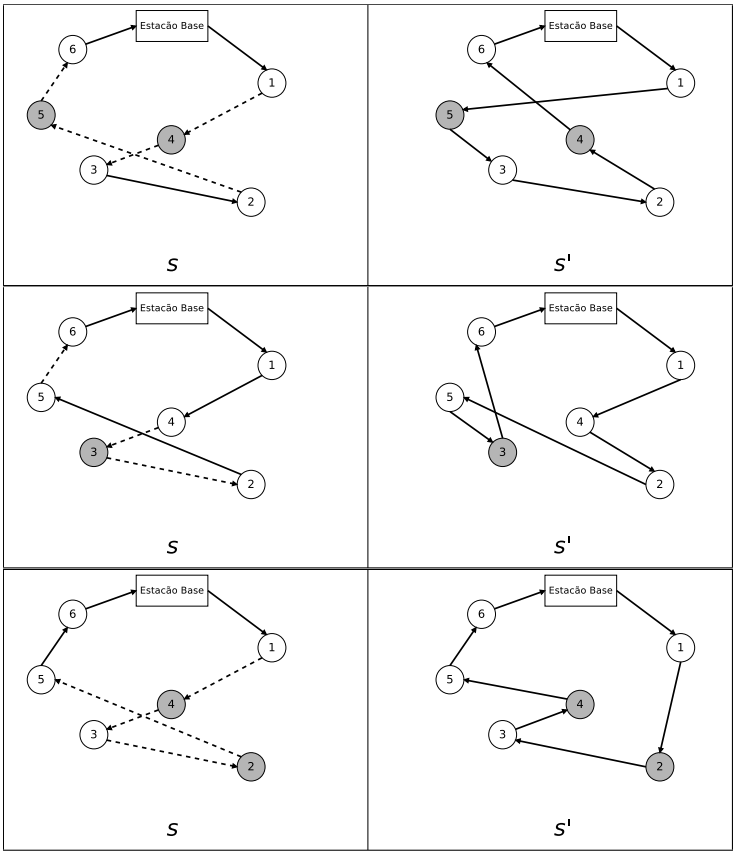


Figura 1: Exemplo de cada uma das estruturas de vizinhança utilizadas na ordem Swap, Shift e 2-Opt.

3.3. GRASP-VND

Nesta seção, apresenta-se a heurística proposta neste trabalho, denominada GRASP-VND. O GRASP-VND é basicamente composto pela heurística construtiva e pelas estruturas de vizinhanças descrita nas seções anteriores, dentro de um processo iterativo. O pseudocódigo do GRASP-VND para o PRMD_RM é apresentado no Algoritmo 2. O primeiro passo do algoritmo é inicializar as soluções s^* (melhor solução) e s (solução corrente) como vazias (linha 2) e o valor do custo de s^*, representado por $f(s^*)$, com valor infinito (linha 3). Assim, o laço principal começa e uma solução s é construída (linha 5). Essa solução é submetida à heurística de busca local (linha 6) e, caso o custo da solução s seja melhor (no caso, menor) que o custo da melhor solução s^*, então s^* é atualizada com a solução corrente s (linha 8). Esses passos se repetem até que se completem $maxIter$ iterações.

4. Resultados Computacionais

Nesta seção, são apresentados os experimentos computacionais feitos com o modelo matemático e com a heurística GRASP-VND para o PRMD_RM. Inicialmente, são exibidas as características das instâncias resolvidas e, em seguida, o ambiente computacional utilizado nos testes. Na sequência, são apresentados os resultados computacionais obtidos com a execução do CPLEX e do GRASP-VND para as instâncias descritas.

Algoritmo 2: GRASP-VND para o PRMD_RM

1: **GRASP-VND**(α, V, *maxIter*)
2: $s^* \leftarrow s \leftarrow \emptyset$
3: $f(s^*) \leftarrow \infty$
4: **Para** $iter = 1$ **até** *maxIter* **faça**
5: $s \leftarrow$ ConstruçãoGulosaAleatória(α, V)
6: $s \leftarrow$ VND(s)
7: **Se** $f(s) < f(s^*)$ **então**
8: $s^* \leftarrow s$
9: **Fim-se**
10: **Fim-para**
11: **Retorne** s^*

As instâncias desse problema foram geradas aleatoriamente e possuem 3 identificadores (X_Y_Z) numéricos que as caracterizam: tamanho da grade (X), densidade de sensores (Y) e posicionamento da base (Z). A grade é quadrada e define os pontos que os sensores podem ocupar; para esse trabalho foram definidos 3 tamanhos de grade de 20x20, 40x40, 60x60. A densidade (Y) representa a quantidade de pontos da grade que são preenchidos por sensores; as densidades podem ser de 20%, 50% e 80% dos pontos da grade preenchidos. O último identificador da instância (Z) indica o posicionamento da base, que pode ser de 1 a 6. As posições podem ser nos cantos da grade (1–4), no centro (5), ou, ainda, uma posição aleatória (6). A instância ainda possui como característica as coordenadas de cada sensor e um raio de comunicação gerado aleatoriamente. É possível visualizar melhor o posicionamento das bases na Figura 8. Para saber exatamente quantos sensores cada instância possui é possível fazer o cálculo $Qp = Tg \times Tg$, onde Tg é o tamanho da grade e Qp é a quantidade de pontos, para depois realizarmos o cálculo de densidade $Qp \times densidade$. Logo, as instâncias com identificadores 20_20 possuem 80 sensores. As instâncias foram geradas dessa maneira para avaliar se alguma dessas características descritas acima dificultam ou facilitam a resolução do PRMD_RM pelos métodos propostos. Essas características e suas consequências são analisadas nos experimentos computacionais.

O modelo matemático de fluxo apresentado na seção 2 foi executado em uma máquina com processador Intel i7-4770 3,40GHz e 16 GB de memória RAM e o CPLEX versão 20.1 sem preprocessamentos e cortes. A Tabela 4 apresenta os resultados encontrados para cada instância, reportando a solução obtida, o tempo em segundos e o GAP de otimalidade. Essa tabela reporta somente os resultados computacionais das instâncias de classe PRMD_20_20_*, com exceção da instância PRMD_20_20_3. Isso acontece pois não foi possível encontrar soluções inteiras viáveis utilizando 3600 segundos como tempo limite para execução dos experimentos do modelo matemático. Isso demonstra e reforça a importância da utilização de métodos heurísticos para encontrar soluções do PRMD_RM para as instâncias em que o modelo matemático não foi capaz de encontrar.

Tabela 2: Resultados computacionais do modelo de fluxo para o PRMD_RM.

Instância	Solução	Tempo(s)	GAP
20_20_1	346,265	3566,66	0,911692
20_20_2	315,464	3570,05	0,912489
20_20_3	-	3600,01	-
20_20_4	94,589	3567,26	0,644912
20_20_5	275,29	3570,63	0,885999
20_20_6	212,608	3571,67	0,863107

A heurística GRASP-VND foi desenvolvida em linguagem C++ e o método foi executado 10 vezes para cada instância no mesmo computador que o modelo matemático. A Tabela 3 mostra os resultados computacionais obtidos pela heurística GRASP-VND. Nela, pode-se observar os identificadores das instâncias, junto com os valores de melhor custo de solução, a média de custo e a média de tempo seguidas por seus respectivos desvios padrão (DesvPad). Esses resultados numéricos podem ser analisados de acordo com as métricas consideradas na criação das instâncias, observando a variação: da quantidade de nós, da densidade e do posicionamento da base.

Analisando a quantidade de nós na instância é possível observar pela Tabela 3, que da grade de 20x20 para as de 40x40 o aumento é de aproximadamente 100 vezes no tempo de execução. Esse mesmo crescimento não pode ser observado quando passamos de 40x40 para 60x60. Quanto ao custo de solução, é possível avaliar também um aumento de aproximadamente 3,5 vezes de 20x20 para 40x40. De 40x40 para 60x60, esse aumento foi de 2,6 vezes. As duas análises indicam que a dificuldade de passar de um tamanho de grade para outro gera barreiras grandes no custo computacional, que podem ser causadas tanto pelo processo de geração da instância, quanto pelas heurísticas. Em relação ao modelo de fluxo podemos observar que a meta-heurística conseguiu encontrar soluções mais rápido e de melhor qualidade.

Considerando o indicador de densidade para as instâncias nota-se que o aumento na quantidade de nós entre as densidades possui aumento de custo entre 14% e 20% indicando que para os tamanhos de instâncias geradas a dificuldade não aumenta muito. Quanto ao tempo computacional a variação das densidades apresenta um aumento de aproximadamente 3 vezes para instâncias de mesmo tamanho de grade e densidades maiores. Essa característica pode ser devido à construção das instâncias que foram randomicamente geradas.

Na Figura 8, é possível observar, para cada instância do conjunto de 20_20, a representação gráfica da melhor solução encontrada pelo GRASP-VND. A mula sai da base (representada por um quadrado preto) e visita um conjunto de nós (representados por pontos cinzas) que juntos fazem a cobertura de todos os demais nós que não se encontram nas rotas. Ainda na Figura 8, vale ressaltar que os nós representados pelas coordenadas (3,14), (1,2) e (17,8) aparecem em todas as soluções, indicando nós que possuem boa cobertura . Quanto ao posicionamento da base é possível apontar que a diferença entre as bases não corresponde em variação importante dos custos de solução e de tempo computacional na heurística. No entanto, o posicionamento da base na posição 3 dificultou encontrar soluções para instância DMRP_20_20_3 no modelo de fluxo, demonstrando que existe uma dificuldade associada ao posicionamento da base, quando o intuito é encontrar solução viável.

Os experimentos realizados avaliaram as características das instâncias geradas e mostraram que o GRASP-VND prosposto é efetivo para a solução do PRMD_RM, encontrando soluções viáveis para todas as instâncias consideradas, com um tempo computacional de no máximo 11 minutos (tempo necessário para resolver instâncias com até 2880 sensores). Além disso, verificou-se que a execução do modelo matemático não conseguiu resolver um conjunto grande de instâncias,

principalmente quando esse conjunto tinha um número maior de nós no grafo.

Tabela 3: Resultados computacionais da heurística GRASP-VND para o PRMD_RM.

Instância	Melhor	Média Sol	DesvPad Sol	Média Tempo	DesvPad Tempo	Instância	Melhor	Média Sol	DesvPad Sol	Média Tempo	DesvPad Tempo	Instância	Melhor	Média Sol	DesvPad Sol	Média Tempo	DesvPad Tempo
20_20_1	66,79	72,06	2,20	0,04	0,00	40_20_1	248,30	262,44	6,87	3,76	0,11	60_20_1	549,39	567,67	6,77	61,08	2,02
20_20_2	71,54	74,92	2,14	0,04	0,00	40_20_2	258,99	263,71	2,87	3,68	0,14	60_20_2	558,35	569,69	6,07	60,30	2,00
20_20_3	69,44	73,00	1,83	0,04	0,00	40_20_3	241,99	255,58	7,74	3,78	0,14	60_20_3	555,44	568,90	6,44	59,50	1,09
20_20_4	71,11	77,74	2,84	0,04	0,00	40_20_4	251,59	264,28	7,46	3,70	0,15	60_20_4	559,15	565,65	5,14	60,67	1,31
20_20_5	69,10	72,46	2,19	0,04	0,00	40_20_5	238,16	253,73	7,24	3,47	0,13	60_20_5	552,40	563,51	7,07	56,26	1,48
20_20_6	64,04	69,06	2,22	0,04	0,00	40_20_6	237,14	253,39	8,63	3,57	0,13	60_20_6	546,82	562,23	5,63	60,34	2,14
Média	68,67	73,21	2,24	0,04	0,00	Média	246,03	258,86	6,80	3,66	0,13	Média	553,59	566,28	6,19	59,69	1,67

Instância	Melhor	Média Sol	DesvPad Sol	Média Tempo	DesvPad Tempo	Instância	Melhor	Média Sol	DesvPad Sol	Média Tempo	DesvPad Tempo	Instância	Melhor	Média Sol	DesvPad Sol	Média Tempo	DesvPad Tempo
20_50_1	80,69	84,68	2,45	0,17	0,01	40_50_1	307,57	315,16	5,99	16,69	0,45	60_50_1	667,93	679,20	7,27	298,64	6,35
20_50_2	78,80	85,64	2,56	0,17	0,01	40_50_2	309,24	318,97	4,07	16,64	0,42	60_50_2	662,28	676,88	6,83	302,43	5,51
20_50_3	82,07	85,83	1,85	0,17	0,01	40_50_3	304,56	316,30	7,08	16,53	0,43	60_50_3	669,53	677,16	3,81	296,41	5,89
20_50_4	80,55	83,35	1,81	0,17	0,01	40_50_4	311,88	317,49	5,18	16,40	0,55	60_50_4	666,14	676,44	6,53	299,19	8,35
20_50_5	76,83	80,74	1,81	0,16	0,01	40_50_5	304,97	314,39	6,03	15,20	0,49	60_50_5	663,39	676,30	7,86	280,48	5,29
20_50_6	78,86	81,51	1,50	0,17	0,01	40_50_6	308,01	314,47	3,99	15,73	0,54	60_50_6	658,17	672,65	7,24	293,28	6,09
Média	79,63	83,63	2,00	0,17	0,01	Média	307,71	316,13	5,39	16,20	0,48	Média	664,57	676,44	6,59	295,07	6,25

Instância	Melhor	Média Sol	DesvPad Sol	Média Tempo	DesvPad Tempo	Instância	Melhor	Média Sol	DesvPad Sol	Média Tempo	DesvPad Tempo	Instância	Melhor	Média Sol	DesvPad Sol	Média Tempo	DesvPad Tempo
20_80_1	84,68	88,05	1,92	0,33	0,01	40_80_1	317,03	333,96	7,56	37,27	1,90	60_80_1	733,16	742,10	6,42	693,11	18,07
20_80_2	82,05	88,73	2,62	0,33	0,02	40_80_2	318,77	332,23	7,73	36,98	1,39	60_80_2	728,64	742,30	7,29	696,18	15,70
20_80_3	83,95	89,76	2,62	0,33	0,01	40_80_3	325,96	331,87	3,73	37,09	1,10	60_80_3	728,18	744,09	9,18	698,87	20,74
20_80_4	85,76	90,83	2,47	0,33	0,01	40_80_4	322,10	333,82	6,69	36,83	1,19	60_80_4	718,93	737,58	8,13	701,03	20,54
20_80_5	80,08	85,49	2,62	0,30	0,01	40_80_5	322,51	330,77	5,60	34,40	1,32	60_80_5	715,75	737,21	10,23	659,73	22,06
20_80_6	78,84	85,45	2,70	0,32	0,01	40_80_6	321,28	331,01	4,53	35,01	1,44	60_80_6	718,18	736,92	8,31	686,39	16,85
Média	82,56	88,05	2,49	0,32	0,01	Média	321,28	332,28	5,97	36,26	1,39	Média	723,81	740,03	8,26	689,22	19,00

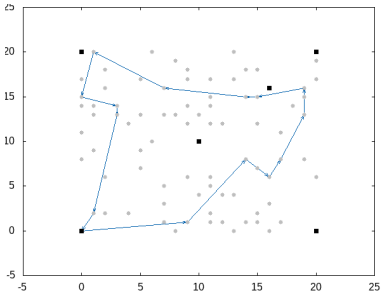


Figura 2: Inst. 20_20_1

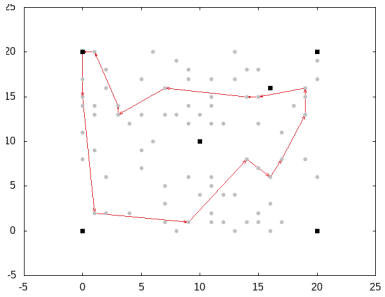


Figura 3: Inst. 20_20_2

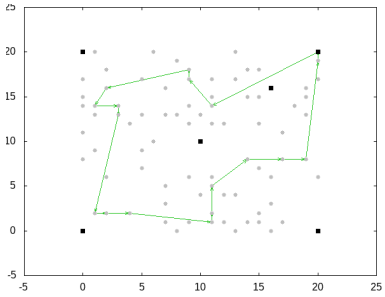


Figura 4: Inst. 20_20_3

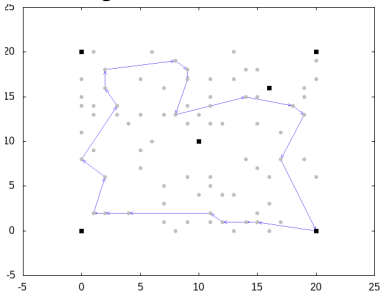


Figura 5: Inst. 20_20_4

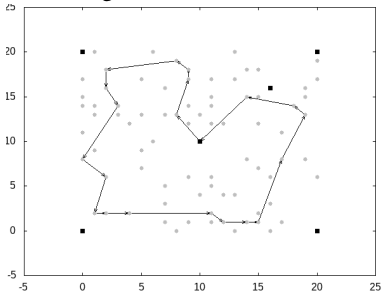


Figura 6: Inst. 20_20_5

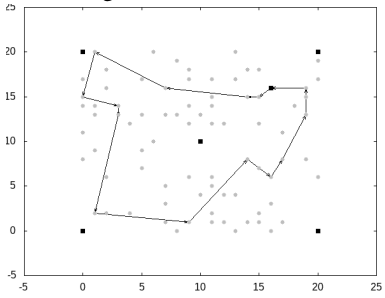


Figura 7: Inst. 20_20_6

Figura 8: Melhores soluções encontradas para as menores instâncias 20_20, para diferentes bases

5. Conclusão e Trabalhos Futuros

Este trabalho apresentou uma abordagem para o problema de roteamento de mula dados onde os sensores são capazes de se comunicar, reduzindo assim a necessidade da mula de dados percorrer todos os sensores. Além da definição formal e do modelo matemático desta aplicação, foi proposta uma heurística GRASP-VND para buscar soluções de qualidade em baixo tempo computacional. A nível de experimentos computacionais, como o modelo matemático e o GRASP-VND propostos são as primeiras técnicas para resolução do PRMD_RM, foi feita uma análise do impacto do crescimento da instância na execução dos métodos e também do posicionamento da base em relação ao sensores. Mostrou-se aqui que o aumento de custo computacional das instâncias de grade 20x20 para as de grade 40x40 foi muito mais significativo do que as demais variações no tamanho da grade. Quanto ao posicionamento da base, verificou-se que este gera pouco impacto na solução obtida e no tempo computacional da heurística GRASP-VND, embora ainda possua relevância na execução do modelo matemático.

Considerando o apresentado neste trabalho, um caminho natural de pesquisa a ser seguido no futuro consiste em desenvolver outras formulações de programação linear inteira e outros métodos exatos para que seja possível avaliar a qualidade das soluções obtidas em relação a solução ótima ou a um limitante inferior, já que com modelo de fluxo utilizado não foi possível encontrar soluções para a maior parte das instâncias. É possível também adotar novas estratégias para construção de soluções, além de realizar estudos no sentido de determinar uma ordem mais adequada para as vizinhanças do VND, ou substituí-lo por um RVND.

Referências

Bin Tariq, M. M., Ammar, M., e Zegura, E. (2006). Message ferry route design for sparse ad hoc networks with mobile nodes. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '06, p. 37–48, New York, NY, USA. ACM.

Cohen, T. e Jones, P. (2020). Technological advances relevant to transport–understanding what drives them. *Transportation Research Part A: Policy and Practice*, 135:80–95.

Erdelj, M., Król, M., e Natalizio, E. (2017). Wireless sensor networks and multi-uav systems for natural disaster management. *Computer Networks*, 124:72–86.

Feo, T. A. e Resende, M. G. C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133.

Gendreau, M. e Potvin, J.-Y. (2010). *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*. Springer, $2nd$ edition.

Kamble, S. S., Gunasekaran, A., e Gawankar, S. A. (2018). Sustainable industry 4.0 framework: A systematic literature review identifying the current trends and future perspectives. *Process Safety and Environmental Protection*, 117:408–425.

Laporte, G. (1992). The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(2):231–247.

Li, X., Li, D., Wan, J., Vasilakos, A. V., Lai, C.-F., e Wang, S. (2017). A review of industrial wireless networks in the context of industry 4.0. *Wireless networks*, 23(1):23–41.

Ma, M. e Yang, Y. (2007). SenCar: An Energy-Efficient Data Gathering Mechanism for Large-Scale Multihop Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems*, 18 (10):1476–1488.

Ma, M., Yang, Y., e Zhao, M. (2013). Tour planning for mobile data gathering mechanisms in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 62(4):1472–1483.

Meijer, A. e Bolívar, M. P. R. (2016). Governing the smart city: a review of the literature on smart urban governance. *international review of administrative sciences*, 82(2):392–408.

Munhoz, P. L., do Carmo, F. P., Souza, U. S., Drummond, L. M., González, P. H., Ochi, L. S., e Michelon, P. (2019). Locality sensitive algotrithms for data mule routing problem. In *International Conference on Algorithmic Applications in Management*, p. 236–248. Springer.

Munhoz, P. L. A., González, P. H., dos Santos Souza, U., Ochi, L. S., Michelon, P., e Drummond, L. M. d. A. (2018). General variable neighborhood search for the data mule scheduling problem. *Electronic Notes in Discrete Mathematics*, 66:71–78.

Naser, M. e Kodur, V. (2020). Concepts and applications for integrating unmanned aerial vehicles (uav's) in disaster management. *Advances in Computational Design*, 5(1):91–109.

Ngai, E. C. H., Liu, J., e Lyu, M. R. (2009). An adaptive delay-minimized route design for wireless sensor-actuator networks. *IEEE Trans. Vehicular Technology*, 58(9):5083–5094.

Rao, J., Wu, T., e Biswas, S. (2008). Network-Assisted Sink Navigation Protocols for Data Harvesting in Sensor Networks. *2008 IEEE Wireless Communications and Networking Conference*, p. 2887–2892.

Sugihara, R. e Gupta, R. K. (2010). Speed control and scheduling of data mules in sensor networks. *ACM Transactions on Sensor Networks*, 7(1):4:1–4:29.

Sugihara, R. e Gupta, R. K. (2011a). Path planning of data mules in sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 8(1):1–27.

Sugihara, R. e Gupta, R. K. (2011b). Path planning of data mules in sensor networks. *ACM Transactions on Sensor Networks*, 8(1):1:1–1:27.

Wichmann, A., Chester, J., e Korkmaz, T. (2012). Smooth path construction for data mule tours in wireless sensor networks. In *GLOBECOM*, p. 86–92.

Xing, G., Wang, T., Xie, Z., e Jia, W. (2008). Rendezvous Planning in Wireless Sensor Networks with Mobile Elements. *IEEE Transactions on Mobile Computing*, 7(12):1430–1443.

Yuan, B., Orlowska, M., e Sadiq, S. (2007). On the Optimal Robot Routing Problem in Wireless Sensor Networks. *IEEE Transactions on Knowledge and Data Engineering*, 19(9):1252–1261.

Zhao, W. e Ammar, M. (2003). Message ferrying: Proactive routing in highly-partitioned wireless ad hoc networks. *The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems, 2003. FTDCS 2003. Proceedings.*, p. 308–314.

Zhao, W., Ammar, M., e Zegura, E. (2005). Controlling the mobility of multiple data transport ferries in a delay-tolerant network. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 2, p. 1407–1418. IEEE.