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

Um algoritmo heurístico baseado em *Iterated Greedy* para o Problema de Roteamento de Unidades Móveis de Mamografia

Otávio Augusto Souza Rosa, Patrick Moreira Rosa, Jéssica Natália Miranda Paiva, Puca Huachi Vaz Penna, Marcone Jamilson Freitas Souza
Departamento de Computação, Universidade Federal de Ouro Preto (UFOP)
Campus Universitário, Morro do Cruzeiro, CEP 35400-000, Ouro Preto (MG) Brasil
{otavio.rosa, patrick.moreira, jessica.paiva}@aluno.ufop.edu.br
{puca, marcone}@ufop.edu.br

RESUMO

Este trabalho trata do Problema de Roteamento de Unidades Móveis de Mamografia (MMURP), que consiste em roteirizar um conjunto de Unidades Móveis de Mamografia (MMU) para atender a demanda de localidades desprovidas de mamógrafos fixos ou com número insuficiente deles. O objetivo é maximizar a demanda atendida e minimizar a distância total percorrida pelas MMUs. Para resolvê-lo, propomos o algoritmo *Smart* IG, baseado na metaheurística *Iterated Greedy* (IG). Neste algoritmo, uma solução inicial é gerada por meio de um procedimento de três passos. Para refinar uma solução, usamos o procedimento *Randomized Variable Neighborhood Descent* (RVND). Para testá-lo, usamos instâncias com 579 localidades, dois depósitos, até 56 MMUs e até 180km entre dois locais. Os resultados mostraram que o algoritmo retorna o número necessário de MMUs e suas roteirizações, de forma a atender a toda a demanda e minimizar a distância total percorrida por elas.

PALAVRAS CHAVE. Unidades Móveis de Mamografia, Roteamento de Veículos, Roteamento de Veículos Multi-depósito, Logística na Área de Saúde.

Tópicos: PO na Área de Saúde, Meta-heurísticas

ABSTRACT

This work deals with the Mobile Mammography Unit Routing Problem (MMURP). This problem consists of defining routes for a set of Mobile Mammography Units (MMU) to cover the demand of locations without fixed mammography units or with an insufficient number of them. The objective is to maximize the demand attended and minimize the total distance traveled by the MMUs. To treat the problem, we propose the Smart IG algorithm, which is based on the Iterated Greedy metaheuristic. In this algorithm, we generated an initial solution through a three-step procedure. To refine a solution, we use the Randomized Variable Neighborhood Descent (RVND) method. To test the proposed algorithm, we used instances with 579 locations, two depots, up to 56 MMUs, and 180km between two locations at maximum. The results showed that the algorithm returns the required number of MMUs and their routings to meet all the demand and minimize the total distance traveled by them.

KEYWORDS. Mobile Mammography Units, Vehicle Routing, Multi-depot Vehicle Routing, Logistics in Health Area.

Paper topics: OR in Health Area, Metaheuristics

1. Introdução

O câncer de mama é o tipo de câncer mais comum em mulheres e a maior causa de mortes por câncer nesse grupo. O Instituto Nacional de Câncer [INCA, 2019] constatou um aumento de 33,6% das taxas de mortalidade por câncer de mama, padronizadas por idade, entre 1980 e 2016. Ainda de acordo com o INCA, a taxa de mortalidade e a morbidade associada ao tratamento podem ser reduzidas pelo diagnóstico precoce da doença.

O exame de mamografia é um dos serviços gratuitos de diagnósticos ofertados pelo Sistema Único de Saúde (SUS). De acordo com da Cunha et al. [2019], muitos fatores dificultam o acesso ao exame de mamografia, dentre eles podemos destacar: a qualidade dos equipamentos, a falta de especialistas para a solicitação de exames, as desigualdades socioeconômicas e a distância entre a localidade onde o mamógrafo fixo está instalado e a localidade em que a mulher reside.

Algumas localidades podem não ter hospitais com infraestrutura adequada e pode não ser justificável economicamente a instalação de um mamógrafo fixo. Este último caso ocorre quando, por exemplo, há localidades que não têm estrutura para sediar um equipamento, têm uma demanda baixa e estão distantes de localidades com infraestrutura. As localidades nessas condições ficam desassistidas e a alternativa para atendê-las são as unidades móveis de mamografia.

Os serviços de mamografia móvel tornam o rastreamento mamográfico acessível para a população que esteja distante dos centros onde os equipamentos de mamografia estão localizados. Além de reduzir a desigualdade geográfica, o mamógrafo móvel possibilita diminuir a desigualdade social na medida em que torna o exame acessível aos indivíduos em situação de carência [Mil et al., 2019]. Assim, as Unidades Móveis de Mamografia (MMU, da sigla em inglês *Mobile Mammography Unit*) podem servir como estratégia complementar para o atendimento da demanda feminina por exames de mamografia. Por outro lado, roteirizar as MMUs não é uma tarefa trivial e exige algoritmos especializados.

Rosa et al. [2020] introduziram o Problema de Roteamento de Unidades Móveis de Mamografia (MMURP, da sigla em inglês *Mobile Mammography Unit Routing Problem*) e apresentaram um algoritmo construtivo para resolver o problema. Não foi encontrada solução manual ou automatizada para resolver este problema e não há na literatura outros trabalhos que apresentem soluções para o MMURP. Este problema é uma variante de um problema de roteamento de veículos, que é NP-difícil [Lenstra e Rinnooy-Kan, 1981].

No MMURP, pretende-se roteirizar uma frota de MMUs sediadas em diferentes hospitais (os quais servem como depósitos desses veículos) para atender a um conjunto de localidades desprovidas de equipamentos fixos ou em número insuficiente destes. O objetivo é maximizar a cobertura de exames e reduzir a distância total percorrida pelas MMUs. O MMURP tem características próprias que o diferenciam de outros problemas de roteamento encontrados na literatura. Dada a dificuldade de resolvê-lo na otimalidade por métodos exatos, nós desenvolvemos um algoritmo heurístico, baseado na metaheurística *Iterated Greedy* (IG) [Ruiz e Stützle, 2008]. A versão que propomos é a *Smart IG*, baseada no trabalho de Reinsma et al. [2018].

O restante deste trabalho está organizado como segue. A Seção 2 caracteriza o MMURP, e a Seção 3 introduz o algoritmo proposto para resolver este problema. Na Seção 4 são apresentados os experimentos computacionais e os resultados alcançados. Por fim, a Seção 5 conclui o trabalho e apresenta as perspectivas de trabalhos futuros.

2. Caracterização do Problema

O MMURP possui as seguintes características. Seja $G = (V, A)$ um grafo direcionado onde $V = N \cup M$ sendo N um conjunto de localidades e M um conjunto de depósitos, e $A = \{(i, j) : i, j \in V, i \neq j\}$ é o conjunto de arestas, sendo que para cada aresta $(i,j) \in A$ está

associado uma distância d_{ij} entre as localidades i e j. O vértice 0 representa o depósito, onde se encontra a frota de veículos, enquanto o conjunto $V' = V \backslash \{0\}$ dos vértices restantes representam as n localidades. Cada localidade $i \in V'$ possui uma demanda dem_i não negativa (assume-se que $dem_0 = 0$).

Cada MMU $k \in K$ possui uma capacidade cap_k definida por: cap_k = *Quantidade de mamógrafos transportado pela MMU* $k \times$ *capacidade do mamógrafo* (isto é, a capacidade equivale à quantidade de exames que podem ser realizados em cada veículo). Há uma distância máxima, dada por *distMax*, permitida para uma MMU realizar o percurso de uma localidade a outra. Apenas a viagem inicial partindo do depósito não tem limite de quilometragem.

Cada localidade deve ser visitada uma única vez e por apenas uma MMU. A demanda da localidade visitada deve ser totalmente atendida. O roteamento é aberto, isto é, as MMUs não retornam aos depósitos de origem, elas seguem um percurso enquanto for possível atender as demandas por exames e finalizam a rota na última localidade visitada.

A Figura 1 ilustra um exemplo de solução viável com 12 localidades, representadas pelo conjunto $N = \{1, 2, \cdots, 12\}$, e dois depósitos, representados pelo conjunto $M = \{13, 14\}$. No depósito 13 há uma única MMU, enquanto no depósito 14 há duas MMUs. Cada localidade possui uma demanda a ser atendida que está representada pelo valor entre colchetes. Como pode ser observado no exemplo, a demanda da localidade 1 é indicada por $dem_1 = 100$ exames. A distância entre cada par de vértices é representada pelo valor acima dos arcos. No exemplo, a distância entre as localidades 1 e 6 é indicada por $d_{1,6} = 20$ unidades de distância.

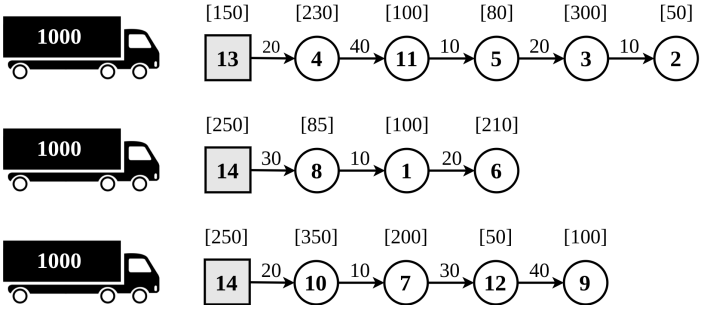


Figura 1: Exemplo de solução com doze localidades e dois depósitos

Na solução apresentada, as localidades 13, 4, 11, 5, 3 e 2 são atendidas pela MMU que inicia sua rota no depósito situado na localidade 13, nesta mesma ordem de visita. No depósito 14, duas MMUs foram utilizadas: a primeira atende as localidades 14, 8, 1 e 6, respectivamente, e já a segunda atende as localidades 10, 7, 12 e 9.

A demanda total coberta neste exemplo foi de 2.255 exames, e a distância total percorrida pelas MMUs foi de 260km. As MMUs apresentadas na Figura 1, possuem a capacidade de realizar 1.000 exames. A MMU do depósito 13 atendeu a uma demanda de 910 exames, restando assim uma capacidade residual de 90 exames. A primeira MMU do depósito 14 possui capacidade residual de 355 exames, enquanto que a segunda MMU ainda pode realizar 50 exames.

3. Algoritmo Proposto

O algoritmo proposto para resolver o MMURP é o *Smart* IG. Esse algoritmo é uma variação do IG [Ruiz e Stützle, 2008], que possui como característica a exploração do espaço de soluções do MMURP por meio de componentes de destruição e construção da solução. Ao contrário do algoritmo IG, no *Smart* IG o nível de perturbação aumenta somente após um certo número de

tentativas de melhoria sem sucesso, visto que a região pode não ter sido explorada adequadamente. Essa abordagem faz parte do estudo de Reinsma et al. [2018], os quais nomearam *Smart* ILS o algoritmo ILS desenvolvido com essa característica.

Esta seção está assim organizada. A Subseção 3.1 mostra a representação e a avaliação de uma solução. Na Subseção 3.2 o procedimento para gerar uma solução inicial é apresentado. As vizinhanças, o procedimento de busca local, e os procedimentos de destruição e construção que modificam a solução corrente são descritos nas Subseções 3.3, 3.4 e 3.5, respectivamente.

3.1. Representação e Avaliação da Solução

Representa-se uma solução s do MMURP por meio de até $|K|$ vetores que armazenam em cada posição as localidades percorridas pelas MMUs na ordem em que aparecem na rota. Uma solução deste problema é avaliada de modo hierárquico pela função de avaliação a seguir: $f(s) = (f_{dem}(s), f_{dist}(s))$. A solução é avaliada inicialmente pela demanda total atendida, conforme apresentado na Equação (1). Caso as duas soluções tenham a mesma demanda total, a melhor é aquela cuja soma das distâncias percorridas seja a menor, como apresentado na Equação (2):

$$f_{dem}(s) = \sum_{(j,k)\in A_1} dem_j \tag{1}$$

$$f_{dist}(s) = \sum_{(i,j)\in A_2} d_{ij} \tag{2}$$

em que A_1 é o conjunto das associações entre uma localidade j e uma MMU k que a atende na solução s, dem_j é a demanda da localidade j, A_2 é o conjunto das arestas ligando as localidades i e j atendidas por uma mesma MMU, e d_{ij} é a distância entre as localidades i e j.

3.2. Geração da Solução Inicial

Uma solução inicial para o MMURP é gerada por meio de um procedimento definido pelos três passos seguintes:

Passo 1 - Ordenação: Seja LC a lista de localidades candidatas a serem visitadas pelas MMUs, ordene-a decrescentemente de acordo com a demanda de cada localidade;

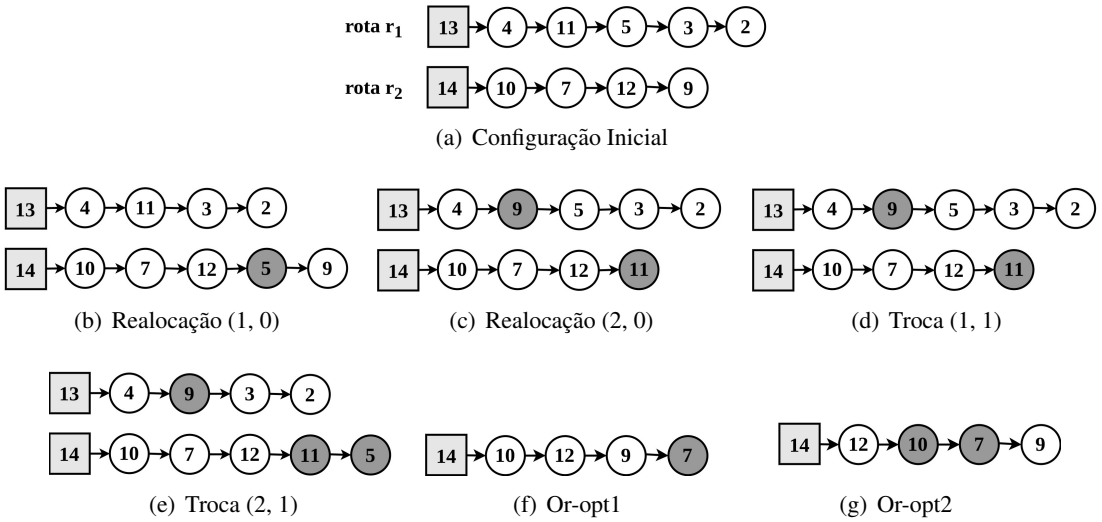
Passo 2 - Alocação: Para cada localidade i da LC ordenada, aloque a localidade i ao depósito mais próximo e elimine-a de LC. Passe para a próxima localidade da lista. Ao final deste passo tem-se uma lista de localidades candidatas para cada depósito;

Passo 3 - Heurística do vizinho mais próximo: Para cada depósito, enquanto houver MMU disponível e localidades ainda não visitadas da lista de candidatas do depósito:

- 3.1: Inicie a rota de uma nova MMU com a localidade onde se encontra o depósito, atenda a demanda do depósito e atualize a capacidade da MMU que foi utilizada neste atendimento;
- 3.2: Determine a localidade mais próxima da última localidade inserida na rota da MMU que ainda não foi visitada;
- 3.3: Verifique se é possível atender a demanda da localidade mais próxima:
  - (a) Se a demanda da localidade mais próxima for menor ou igual à capacidade da MMU e a distância até a localidade mais próxima for menor ou igual a *distMax*, atenda-a e insira-a na rota da MMU. Atualize a capacidade da MMU e remova a localidade atendida da lista de candidatas do depósito. Volte ao item 3 do Passo 3;
  - (b) Caso contrário, finalize a rota da MMU na última localidade por ela visitada;
- 3.4: Se houver MMU disponível, retorne ao item 1 do Passo 3; caso contrário, encerre o método e retorne as rotas feitas pelas MMUs.

3.3. Estruturas de Vizinhança

O conjunto de vizinhanças utilizadas para explorar o espaço de soluções do MMURP é formado a partir de quatro movimentos inter-rotas (Realocação (1, 0), Realocação (2, 0), Troca (1, 1) e Troca (2, 1)) e por dois movimentos intrarrotas (Or-opt1 e Or-opt2). Esses movimentos podem ocorrer entre rotas de um mesmo depósito ou entre rotas de depósitos diferentes. A Figura 2 ilustra as vizinhanças utilizadas.



(a) Configuração Inicial

(b) Realocação (1, 0)

(c) Realocação (2, 0)

(d) Troca (1, 1)

(e) Troca (2, 1)

(f) Or-opt1

(g) Or-opt2

Figura 2: Estruturas de vizinhanca

Realocação (1, 0): Uma localidade i é removida de uma rota r_1 e inserida em outra rota r_2. Observa-se na Figura 2(b) que a localidade 5 é retirada da rota r_1 e reinserida na rota r_2.

Realocação (2, 0): Duas localidades adjacentes i e $i+1$ são removidas de uma rota r_1 e inseridas em outra rota r_2. Na Figura 2(c), as localidades 5 e 3 são transferidas para a rota r_2.

Troca (1, 1): É realizada uma troca entre duas localidades i e j de duas rotas distintas. Na Figura 2(d), a localidade 11 da rota r_1 é trocada com a localidade 9 da rota r_2.

Troca (2, 1): Duas localidades adjacentes i e $i+1$, são trocadas com uma localidade j de uma outra rota. Na Figura 2(e), as localidades 11 e 5 foram trocadas com a localidade 9.

Or-opt1: Uma localidade i é reinserida em uma outra posição da mesma rota. Na Figura 2(f), a localidade 7 é retirada de sua posição e reinserida após a localidade 9.

Or-opt2: Duas localidades adjacentes i e $i+1$ são realocadas em um outro ponto da mesma rota. Na Figura 2(g), as localidades 10 e 7 foram realocadas entre as localidades 12 e 9.

3.4. Busca Local

O algoritmo *Smart* IG utiliza na fase de busca local o procedimento *Randomized Variable Neighborhood Descent* (RVND) [Subramanian et al., 2010; Souza et al., 2010], no qual as vizinhanças são selecionadas em ordem aleatória para explorar o espaço de soluções do problema.

O pseudocódigo do RVND é apresentado no Algoritmo 1. A princípio, inicializamos o conjunto V_{inter} das quatro vizinhanças inter-rotas apresentadas na Seção 3.3. A cada iteração, uma vizinhança v desse conjunto é selecionada aleatoriamente (linha 3); em seguida, é realizada uma

descida na solução usando a estratégia *First Improvement* (linha 4) para encontrar um vizinho de primeira melhora s', dentro da vizinhança escolhida anteriormente.

A solução ótima local s' é, então, avaliada conforme a função objetivo (linha 5). Caso essa solução ótima local com relação à vizinhança inter-rota v seja melhor do que a solução corrente, a busca continua a partir de s' (linha 6), porém aplicando-se o procedimento *Busca_Intrarrota*. Este procedimento consiste também em um RVND, porém utilizando-se movimentos intrarrotas (linha 7). Na linha 8, o conjunto V_{inter} é reiniciado com todas as vizinhanças definidas inicialmente. No caso de a solução s' não ser de melhora, a vizinhança selecionada anteriormente é removida do conjunto V_{inter} (linha 10), e a busca continua a partir de uma outra estrutura de vizinhança inter-rota. A busca termina quando o conjunto de vizinhanças V_{inter} estiver vazio.

Algoritmo 1: RVND

Entrada: s
Saída: s

1 Inicialize V_{inter} com as vizinhanças inter-rotas e V_{intra} com as vizinhanças intra-rotas;
2 **enquanto** $V_{inter} \neq \emptyset$ **faça**
3 $v \leftarrow$ Escolha uma vizinhança inter-rota $v \in V_{inter}$ aleatoriamente;
4 $s' \leftarrow$ Encontre o vizinho de primeira melhora de s, de acordo com a vizinhança v escolhida;
5 **se** $f_{dem}(s') > f_{dem}(s)$ ***ou*** $(f_{dem}(s') = f_{dem}(s)$ ***e*** $f_{dist}(s') < f_{dist}(s))$ **então**
6 $s \leftarrow s'$;
7 $s \leftarrow Busca_Intrarrota(s, V_{intra})$;
8 Reinicialize V_{inter};
9 **senão**
10 Remova v de V_{inter};
11 **fim**
12 **fim**

3.5. Destruição e Construção

Os procedimentos de destruição e construção modificam a solução corrente para que a busca local explore diferentes regiões, mantendo características importantes do ótimo local. O procedimento de destruição é descrito no Algoritmo 2. A princípio, as MMUs são ordenadas de maneira decrescente de acordo com a sua capacidade residual (linha 1), isto é, as MMUs com maior capacidade disponível para realizar exames estarão nas primeiras posições. Em seguida, são selecionadas aleatoriamente *kPercentPerturb* MMUs dentre as *nivel* MMUs de maior capacidade residual (linha 2). Por fim, as rotas das MMUs selecionadas anteriormente são esvaziadas e as localidades removidas são reinseridas no conjunto de localidades candidatas LC' (linha 3).

O parâmetro *kPercentPerturb* equivale ao percentual de MMUs que serão escolhidas aleatoriamente dentre o conjunto ordenado de MMUs com as maiores capacidades residuais. Esse percentual pode estar entre 0, caso nenhuma MMU for selecionada, e 1, caso todas as $nivel$ MMUs de maior capacidade residual forem escolhidas. O parâmetro $nivel$ equivale ao nível de perturbação do algoritmo *Smart* IG, sendo incrementado sempre que o algoritmo não conseguir encontrar uma solução de melhora. Após aplicado o procedimento de destruição, a solução s' é então reconstruída conforme mostrado no Algoritmo 3. No procedimento de construção, parte-se da inicialização de uma lista auxiliar com as localidades candidatas LC_{aux}. Enquanto essa lista não estiver vazia, a localidade c de maior demanda é selecionada (linha 3). Caso exista uma MMU com capacidade suficiente para atender a demanda dessa localidade (linha 4), ela será inserida na posição cuja soma das distâncias percorridas pelas MMUs seja a menor e, por isso, será removida da lista de localida-

des candidatas LC'. Por fim, a localidade c é removida da lista LC_{aux}.

Algoritmo 2: Destruição

Entrada: s, $nivel$, $kPercentPerturb$, LC'
Saída: s, LC'
1 Ordene as *MMUs* em ordem decrescente de acordo com sua capacidade residual;
2 Dentre as $nivel$ MMUs de maior capacidade residual, selecione aleatoriamente $kPercentPerturb$ MMUs;
3 Retire todas as localidades das $kPercentPerturb$ MMUs selecionadas anteriormente e as insira na lista de localidades candidatas LC';

Algoritmo 3: Construção

Entrada: s, LC'
Saída: s,
1 $LC_{aux} \leftarrow LC'$;
2 **enquanto** $LC_{aux} \neq \emptyset$ **faça**
3 $c \leftarrow$ Selecione a localidade de maior demanda $\in LC_{aux}$;
4 **se** *exista uma MMU com capacidade para atender a demanda de* c **então**
5 Insira c na posição de menor custo de inserção de s;
6 $LC' \leftarrow LC' \setminus \{c\}$;
7 **fim**
8 $LC_{aux} \leftarrow LC_{aux} \setminus \{c\}$;
9 **fim**

3.6. *Smart Iterated Greedy*

O funcionamento do *Smart IG* é apresentado pelo Algoritmo 4. Neste algoritmo, parte-se da inicialização de uma lista LC com as localidades que são candidatas a serem inseridas na solução. Em seguida, é gerada uma solução inicial s_0 e a partir dela é realizada uma busca por um ótimo local corrente (linhas 2 e 3), gerando uma solução intermediária s. A cada iteração (linhas 8 a 30) a solução corrente s é parcialmente destruída de acordo com um nível de intensidade (linha 11) e reconstruída pelo procedimento de construção (linha 12). A solução reconstruída s'' é então submetida ao procedimento de busca local (linha 13) e avaliada de acordo com a função hierárquica (linha 14). Caso a solução ótima local, s'', seja melhor que a solução s corrente, esta passará a ser então a solução corrente (linha 15). Dessa forma, a lista de candidatas é atualizada com as localidades que foram removidas da solução e não foram reinseridas na fase de construção (linha 16), e o índice de iteração em que houve melhora *melhorIter*, o nível de perturbação *nivel*, e o número de tentativas antes de aumentar o nível de perturbação *tentativas*, são reiniciados (linha 17 a 19). Caso não haja melhorias na solução e o número máximo de tentativas de melhora for alcançado, o número de tentativas é reiniciado (linhas 21 e 22), e o nível de perturbação é incrementado se o seu valor for menor que o nível máximo de perturbação *nivelMaxPerturb* (linhas 23 e 24). Senão, caso o número máximo de tentativas de melhora não tenha sido alcançado, o número de tentativas é incrementado (linha 27). O algoritmo é executado enquanto o número máximo de iterações sem melhora no valor da função de avaliação, $maxIter$, não for atingido.

4. Experimentos Computacionais

O algoritmo proposto foi desenvolvido na linguagem C++, com o compilador gcc 9.3.0. Os testes foram realizados em um computador com processador Intel Core i7-9750, CPU @2.60GHz, com 16 GB de memória RAM sob o sistema operacional Linux Ubuntu, versão 20.04.1 LTS.

Algoritmo 4: Smart Iterated Greedy

Entrada: $maxIter$, $maxTentativasMelhora$
Saída: s

Inicializa LC com as localidades candidatas;
$s_0 \leftarrow$*GeraSolucaoInicial*(LC);
$s \leftarrow$ BuscaLocal(s_0);
$iter \leftarrow 0$;
$melhorIter \leftarrow iter$;
$nivel \leftarrow 1$;
$tentativas \leftarrow 1$;
enquanto $iter - melhorIter < maxIter$ **faça**
    $iter \leftarrow iter + 1$;
    $LC' \leftarrow LC$;
    $s' \leftarrow$ *Destruicao*(s, $nivel$, LC');
    $s'' \leftarrow$ *Construcao*(s', LC');
    $s'' \leftarrow$ *BuscaLocal*(s'');
    se $f_{dem}(s'') > f_{dem}(s)$ ***ou*** $(f_{dem}(s'') = f_{dem}(s)$ ***e*** $f_{dist}(s'') < f_{dist}(s))$ **então**
        $s \leftarrow s''$;
        $LC \leftarrow LC'$;
        $melhorIter \leftarrow iter$;
        $nivel \leftarrow 1$;
        $tentativas \leftarrow 1$;
    senão
        se $tentativas \geq maxTentativasMelhora$ **então**
            $tentativas \leftarrow 1$;
            se $nivel < nivelMaxPerturb$ **então**
                $nivel \leftarrow nivel + 1$;
            fim
        senão
            $tentativas \leftarrow tentativa + 1$;
        fim
    fim
fim

As instâncias utilizadas para testar os métodos computacionais propostos são referentes ao Estado de Minas Gerais, composto por 853 municípios. As demandas de cada localidade foram obtidas do Departamento de Informática do SUS (DATASUS) e foram projetadas para 2020 com base nos dados do censo de 2010 realizado pelo IBGE. Nessa projeção, a demanda total no Estado foi de 1.727.487 exames de mamografia por ano. As distâncias entre cada par de localidades foram obtidas por meio da API Google Maps, considerando-se o deslocamento por carro. A capacidade de realização de mamografias foi calculada conforme a nota técnica divulgada pelo Instituto Nacional de Câncer [INCA, 2015]. Sendo assim, uma MMU com um mamógrafo instalado é capaz de realizar 5.069 exames anualmente.

O modelo para localização de mamógrafos fixos desenvolvido em Souza et al. [2020] foi aplicado inicialmente para simular a atual distribuição dos mamógrafos já existentes no Estado de Minas Gerais e identificar localidades desprovidas desses equipamentos ou em número insuficiente para atendê-las. De acordo com as informações obtidas pelo DATASUS, em setembro de 2019, Minas Gerais possuía 326 mamógrafos. Com essa quantidade de mamógrafos disponíveis no Es-

tado, 274 localidades tiveram a sua demanda atendida por eles, representando um atendimento de 1.146.490 exames. O modelo de Souza et al. [2020] foi utilizado pois não existem informações sobre quais localidades são atendidas por cada mamógrafo do Estado. É importante destacar que cada mamógrafo pode atender a demanda de outras localidades além do local onde ele está fixado.

Portanto, 579 localidades não foram atendidas parcial ou integralmente pelos mamógrafos fixos do Estado. A demanda total das localidades sem atendimento é de 580.997 exames, o que é 33,6% da demanda total do Estado. Dentre essas 579 localidades, verifica-se que 13 delas possuem a demanda superior à capacidade de um mamógrafo. Sendo assim, foram instalados nessas localidades quantos mamógrafos fixos forem necessários para que a sua demanda restante fosse menor do que a capacidade de um mamógrafo. Esse pré-processamento é feito para que as localidades candidatas a serem atendidas pelas MMUs possuam pouca demanda, pois não justifica usar uma MMU para deixá-la fixa em uma localidade.

Após a identificação das demandas dos municípios sem atendimento, foram formulados dois cenários com o objetivo de analisar a cobertura da demanda da região estudada. Foram considerados dois depósitos sendo o primeiro em Patrocínio, por sediar o Hospital do Câncer de Patrocínio Dr. José Figueiredo e que nos forneceu as informações para caracterizar o problema abordado. A sede do segundo depósito foi a cidade de Belo Horizonte, por ser a capital do Estado e estar próximo de uma grande parte das localidades candidatas.

A Tabela 1 apresenta as características de cada instância em cada cenário. A coluna *Instância* mostra o identificador da instância; *MMUs Disp.* mostra a quantidade de MMUs disponíveis nos dois depósitos (como exemplo, 1 / 1 indica que foram usadas 1 MMU do depósito de Patrocínio e 1 do depósito de Belo Horizonte, respectivamente); e *Cap. MMUs*, define a capacidade de cada MMU. As instâncias estão divididas de acordo com o cenário em que foram utilizadas. Os cenários diferenciam-se entre si pela capacidade das MMUs utilizadas. Para o cenário 2, foi considerado que as MMUs possuem dois mamógrafos instalados, por isso a capacidade delas é o dobro que a capacidades dos veículos do cenário 1.

Tabela 1: Características das Instâncias

	Instância	MMUs Disp.	Cap. MMUs
Cenário 1	i1	1 / 1	5069
	i2	8 / 8	5069
	i3	16 / 16	5069
	i4	24 / 24	5069
	i5	32 / 32	5069
	i6	40 / 40	5069
	i7	48 / 48	5069
	i8	56 / 56	5069
Cenário 2	i9	1 / 1	10138
	i10	8 / 8	10138
	i11	16 / 16	10138
	i12	24 / 24	10138
	i13	32 / 32	10138

Para a aplicação do algoritmo *Smart* IG foi necessária a definição de quatro parâmetros. A calibragem dos parâmetros foi feita pela ferramenta $irace$ [López-Ibáñez et al., 2016]. Os valores sugeridos ao $irace$ para cada um desses parâmetros foram *maxIter* $\in \{50, 100, 200, 400\}$; *nivelMaxPerturb* $\in [1, 112]$; *maxTentativasMelhora* $\in \{1, 3, 5, 7, 9\}$; e por fim, *kPercentPerturb* $\in \{0{,}2; 0{,}3; 0{,}4; 0{,}5; 0{,}6; 0{,}7; 0{,}8; 0{,}9; 1{,}0\}$. Foram consideradas para a calibração as instâncias i2, i4, i6, i10 e i12. O *irace* retornou como resultado os valores *maxIter* = 200, *nivelMaxPerturb* = 75,

maxTentativasSemMelhora = 7 e *kPercentPerturb* = 0,7.

Os resultados obtidos nos cenários 1 e 2 são mostrados nas Tabelas 2 e 3, respectivamente. Em todas essas tabelas, a coluna *Inst.* mostra o identificador da instância utilizada; *MMUs Usadas* mostra a quantidade de MMUs que foram usadas; *Loc. Atend.* apresenta a quantidade de localidades atendidas pelos veículos dos dois depósitos; *Dem. Atend.* mostra a demanda total atendida pelas MMUs de ambos os depósitos; *% Cob.* equivale ao percentual de cobertura em relação à demanda total de 464.410 exames, *Dist.* (km) é a distância total percorrida pelos veículos de ambos os depósitos e, por fim, *Tempo* (s) indica o tempo gasto, em segundos, para encontrar a solução. Os resultados estão divididos entre a média e o melhor resultado encontrado em 30 execuções. O tempo de execução do algoritmo foi limitado em 1.200 segundos.

A Tabela 2 mostra os resultados obtidos no cenário 1. As soluções que cobriram 100% da demanda da região estudada estão destacadas em negrito. Neste cenário, foram necessárias em média 99,6 MMUs, cada uma com um mamógrafo instalado, para atender a toda a demanda das 579 localidades candidatas. Nessa solução, as distâncias totais que as MMUs percorreram, em média, foi de 54.597km. Analisando a melhor solução encontrada, foram necessárias 99 MMUs para atender toda a demanda e a distância total percorrida de 53.511km. Além disso, na instância $i7$ deste cenário o atendimento de 100% da demanda total foi alcançado usando 96 MMUs, valor abaixo da média, com uma distância total de 54.040km. Como esperado, analisando esses resultados é possível verificar que nas instâncias em que o número de MMUs utilizadas aumenta, a quantidade de locais atendidos e a distância total percorrida para atender esses locais também aumentam. Além disso, o tempo gasto pelo algoritmo também aumenta.

Tabela 2: Resultados alcançados no Cenário 1

	Média						**Melhor**					
Inst.	MMUs Usadas	Loc. Atend.	Dem. Atend.	% Cob.	Dist. (km)	Tempo (s)	MMUs Usadas	Loc. Atend.	Dem. Atend.	% Cob.	Dist. (km)	Tempo (s)
i1	2,0	4,0	10000,0	2,1	597,0	0,1	2	4	10000	2,1	597	0,1
i2	16,0	53,2	80723,5	16,8	9080,1	1,4	16	57	80777	17,3	8918	2
i3	32,0	131,3	161431,1	34,8	17546,5	14,8	32	122	161677	34,8	18673	24
i4	48,0	210,4	241846,8	52,1	28711,6	65,5	48	206	242231	52,1	29079	113
i5	64,0	305,5	322210,2	69,4	39356,1	102,5	64	290	322825	69,5	40552	104
i6	80,0	426,5	403663,7	86,9	58779,8	706,6	80	430	404354	87,0	61265	1093
i7	96,0	577,3	463796,0	99,9	55008,1	549,0	**96**	**579**	**464410**	**100**	**54040**	**1033**
i8	**99,6**	**579,0**	**464410,0**	**100,0**	**54597,0**	**1116,4**	**99**	**579**	**464410**	**100**	**53511**	**1200**

A Tabela 3 apresenta os resultados obtidos no cenário 2. Foram necessárias, em média, 51,6 MMUs para atender toda a demanda percorrendo 35.709,6km. Já na melhor solução encontrada, foram usadas 51 MMUs. Nessa solução, a distância total foi de 34.410km. Na instância $i12$, foi encontrada uma solução que atendeu toda a demanda fazendo uso de 48 MMUs, e neste caso, os veículos percorreram uma distância total de 36.276km.

Tabela 3: Resultados alcançados no Cenário 2

	Média						**Melhor**					
Inst.	MMUs Usadas	Loc. Atend.	Dem. Atend.	% Cob.	Dist. (km)	Tempo (s)	MMUs Usadas	Loc. Atend.	Dem. Atend.	% Cob.	Dist. (km)	Tempo (s)
i9	2,0	11,0	20259,0	5,6	678,0	0,1	2	11	20259	5,6	678	0,1
i10	16,0	114,5	161966,7	34,9	14268,6	4,0	16	104	162016	34,8	14084	5
i11	32,0	286,8	324015,9	69,8	33170,4	47,6	32	279	324177	69,8	33636	75
i12	48,0	572,9	461811,3	99,4	37230,2	106,8	**48**	**579**	**464410**	**100**	**36276**	**457**
i13	**51,6**	**579,0**	**464410,0**	**100,0**	**35709,6**	**892,6**	**51**	**579**	**464410**	**100**	**34410**	**749**

Ao comparar as soluções encontradas nas instâncias $i8$ do primeiro cenário e $i13$ do se-

gundo, é possível observar que ambas as soluções obtiveram uma cobertura de 100% da demanda da região estudada. No entanto, na solução da instância $i8$ foram utilizadas, em média, 99,6 MMUs com um mamógrafo instalado. Já na solução da instância $i13$ foram utilizadas 51,6 MMUs com dois mamógrafos, ou seja, 48 unidades a menos que na instância $i8$.

Finalmente, a Tabela 4 mostra os resultados da comparação entre os algoritmos *Smart* IG, proposto neste trabalho e o algoritmo proposto por Rosa et al. [2020]. Nesta tabela, apresentamos nas duas primeiras colunas o cenário e a instância. A terceira e quarta colunas apresentam a demanda atendida pelos métodos construtivo e *Smart* IG, respectivamente. A última coluna mostra o percentual de melhora do algoritmo proposto em relação ao algoritmo construtivo.

Tabela 4: Comparativo entre o Algoritmo Construtivo de Rosa et al. [2020] e o *Smart* IG

		Algoritmo Construtivo	***Smart* IG**	
Cenário	Instância	Demanda Atendida	Demanda Atendida	Melhora (%)
1	i1	8201	10000	21,9
	i2	70786	80777	14,1
	i3	131909	161677	22,5
	i4	167578	242231	44,5
	i5	204066	322825	58,1
	i6	237537	404354	70,2
	i7	268875	464410	72,7
	i8	305861	464410	51,8
2	i9	19019	20259	6,5
	i10	137049	162016	18,2
	i11	212899	324177	52,2
	i12	285811	464410	62,4
	i13	358131	464410	29,6

5. Conclusões e Trabalhos Futuros

Neste trabalho, propomos um algoritmo *Smart* IG baseado na metaheurística *Iterated Greedy* para resolver o Problema de Roteamento de Unidades Móveis de Mamografia (MMURP). Neste algoritmo, geramos uma solução inicial por meio de um procedimento de três passos. Para refinar uma solução, usamos o método *Randomized Variable Neighborhood Descent* (RVND). Para não ficar preso em ótimos locais e explorar diferentes regiões do espaço de soluções do problema, aplicamos um procedimento para destruir a solução atual e outro para construí-la de forma gulosa.

Para testar esse algoritmo proposto, usamos instâncias com 579 localidades, dois depósitos, até 56 MMUs e 180km entre dois locais no máximo. Realizamos testes considerando dois cenários diferentes. Os resultados mostraram que foi possível atender a toda a demanda de 464.410 exames da região estudada nos dois cenários, com uma média de 99,6 e 51,6 MMUs, respectivamente.

Analisando os resultados nos dois cenários, observa-se que ao aumentar a quantidade de MMUs utilizadas em cada instância, aumenta-se a quantidade de municípios atendidos por elas, a demanda total atendida e, em consequência, a porcentagem de cobertura da demanda total. Além disso, a distância total percorrida pelos veículos na solução também é maior. Comparando os resultados entre os dois cenários, verifica-se que no cenário 2, na instância $i13$ são utilizadas 48 MMUs a menos que na instância $i8$. Vale ressaltar que no segundo cenário, os veículos possuem dois mamógrafos instalados. O uso das MMUs com dois mamógrafos diminui os gastos da aquisição de novos veículos. Ademais, o número de profissionais necessários para o deslocamento e atendimentos das MMUs também diminui. Assim, concluímos que a aquisição de mamógrafos fixos em conjunto com a implementação do serviço de mamografia móvel permite expandir a oferta das mamografias e facilitar o acesso da população na realização de mamografias.

Como trabalhos futuros, propõe-se um estudo para conhecer como é o serviço de roteamento de MMUs em outras regiões do Brasil, um estudo para identificar quais as localidades mais adequadas para serem os depósitos das MMUs e o desenvolvimento de um aplicativo para apoiar os gestores da área de saúde na definição das rotas das MMUs.

Agradecimentos

Os autores agradecem à CAPES, CNPq, FAPEMIG e UFOP pelo apoio ao desenvolvimento deste trabalho.

Referências

da Cunha, G. N., Vianna, C. M. d. M., Mosegui, G. B. G., da Silva, M. P. R., e Jardim, F. N. (2019). Rastreamento do câncer de mama: modelo de melhoria do acesso pelo uso de mamógrafos móveis. *Revista Panamericana de Salud Publica*, 43:e19. PMC6459392.

INCA (2015). Revisão do parâmetro para cálculo da capacidade de produção do mamógrafo simples. Instituto Nacional do Câncer. Disponível em `https://bit.ly/3t8J37W`.

INCA (2019). A situação do câncer de mama no Brasil: síntese de dados dos sistemas de informação. Instituto Nacional do Câncer. Disponível em `https://bit.ly/3eEbplu`.

Lenstra, J. K. e Rinnooy-Kan, A. H. G. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, 11:221–227.

López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Birattari, M., e Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58.

Mil, R. D., Guillaume, E., Launay, L., Guittet, L., Dejardin, O., Bouvier, V., Notari, A., Launoy, G., e Berchi, C. (2019). Cost-effectiveness analysis of a mobile mammography unit for breast cancer screening to reduce geographic and social health inequalities. *Value in Health*, 22(10): 1111 – 1118.

Reinsma, J., Penna, P. H. V., e Souza, M. J. F. (2018). Um algoritmo simples e eficiente para resolução do problema do caixeiro viajante generalizado. In *L Simpósio Brasileiro de Pesquisa Operacional*, Rio de Janeiro. Disponível em `https://bit.ly/3aUfe4V`.

Rosa, O. A. S., Souza, M. J. F., Penna, P. H. V., e Rosa, P. M. (2020). Um algoritmo construtivo para o problema de roteamento das unidades móveis de mamografia. In *LII Simpósio Brasileiro de Pesquisa Operacional*, João Pessoa. Disponível em `https://bit.ly/3gSGzbq`.

Ruiz, R. e Stützle, T. (2008). An iterated greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. *European Journal of Operational Research*, 187(3):1143 – 1159.

Souza, M. J. F., Coelho, I. M., Ribas, S., Santos, H. G., e Merschmann, L. (2010). A hybrid heuristic algorithm for the open-pit-mining operational planning problem. *European Journal of Operational Research*, 207(2):1041–1051.

Souza, M. J. F., Penna, P. H. V., Moreira de Sá, M. V. S., e Rosa, P. M. (2020). A vns-based algorithm for the mammography unit location problem. In Benmansour, R., Sifaleras, A., e Mladenović, N., editors, *Variable Neighborhood Search: 7th International Conference, ICVNS 2019, Rabat, Morocco, October 3–5, 2019, Revised Selected Papers*, volume 12010 of *Lecture Notes in Computer Science*, chapter 3, p. 37–52. Springer, Cham, Switzerland.

Subramanian, A., Drummond, L., Bentes, C., Ochi, L., e Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 37(11):1899 – 1911.