An Hybrid NSGA-II Algorithm for the Bi-objective Mobile Mammography Unit Routing Problem

Thiago Giachetto de Araujo(✉), Puca Huachi Vaz Penna, and Marcone Jamilson Freitas Souza

Departamento de Computação, Universidade Federal de Ouro Preto (UFOP), Campus Universitário, Morro do Cruzeiro, Ouro Preto, MG 35.400-000, Brazil
{thiago.giachetto}@aluno.ufop.edu.br, {puca,marcone}@ufop.edu.br

Abstract. This work deals with the Mobile Mammography Unit Routing Problem in Brazil. The problem is a Multi-depot Open Vehicle Routing Problem variant. In this problem, there are a fixed number of depots, each with a limited number of Mobile Mammography Units (MMUs). Each MMU has a known screening capacity and a set of candidate cities it can serve with known demands for screening. The objective is to define the cities visiting order for each MMU, maximizing the served screening demand and minimizing the total travel distance. We introduce a mathematical programming formulation and two algorithms based on Non-dominated Sorting Genetic Algorithm II (NSGA-II). They differ from each other by the use of a local search. One version has a Local Search as a mutation operator, and the other does not. Both algorithms were tested on benchmark based on real data from Minas Gerais state, Brazil. We used the hypervolume metric to analyze the performance of the proposed algorithms considering different scenarios. The results indicate that using multiple crossover operators and adding a local search as a mutation operator to the algorithm brings better results.

Keywords: Mobile Mammography Unit Routing · Multi-objective Optimization · NSGA-II · Variable Neighborhood Descent · Vehicle Routing

1 Introduction

According to the World Health Organization (WHO), cancer is the leading cause of death in the world. It was responsible for 10 million deaths in 2020. Among all types of cancer, breast cancer is the most common, responsible for 2.26 million new cases in 2020. Nevertheless, it was the fifth in mortality, accounting for 685 thousand deaths in 2020 [8,28]. In Brazil, the reality is the same observed worldwide. Breast cancer has the highest incidence among women and causes the highest mortality (except for non-melanoma skin cancer) [13]. In 2020, it was estimated that 29.7% of cases and 16.5% of deaths in women caused by

L. Rutkowski et al. (Eds.): ICAISC 2023, LNAI 14125, pp. 339–351, 2023.
https://doi.org/10.1007/978-3-031-42505-9_29

cancer were due to breast cancer (when excluding cases of non-melanoma skin cancer) [14].

The Brazilian situation is worsened by the inequality allocation of mammography equipment. The total number of equipment would be sufficient to attend the entire country screening demand. However, the Ministry of Health imposes on women a maximum travel distance of 60 km (or 60 min) to carry out the [26] screening. Consequently, some regions of the country are not served by any mammography unit, and others have more equipment than necessary [1]. Faced with this problem of unbalanced allocation of mammography equipment, several authors have studied this problem. They proposed mathematical formulations and heuristic resolution methods that can help decision-makers allocate equipment in the most efficient way [6,22,23].

Furthermore, to install a mammography unit is necessary a minimum screening demand. Added to the continental dimensions of Brazil, some regions will never be their screening demand completely met. Given this problem, some authors propose using mammography trucks to deal with this residual screening demand [4,22,23]. However, the possibility of using Mobile Mammography Units (MMUs) requires the public manager to make new decisions. For example, how many MMUs are needed to meet all screenings? Which routes will each vehicle take? With this new problem in mind, [20] introduced the Mobile Mammography Unit Routing Problem (MMURP). They treated the problem at two hierarchical levels. At the first level, the objective is to maximize the demand for screenings, and the second is to minimize the total traveled distance by the MMUs.

In this paper, we work with the MMURP as a bi-objective problem, aiming to maximize the demand for screenings and minimize the total traveled distance by the MMUs. The MMURP is a Multi-depot Open Vehicle Routing Problem (MDOVRP) variant. As the vehicle routing problem is NP-hard [16], this work proposes a heuristic for its resolution and uses it to solve the MMURP problem in the state of Minas Gerais, which is the Brazilian state with the largest number of cities and the fourth in area [12]. We run several scenarios that differ from each other by the number of MMUs available in each depot.

The rest of this article is organized as follows. Section 2 introduces the formulation for the MMURP. Section 3 presents the proposed algorithm and its components. Section 4 reports the computational results. Finally, Sect. 5 concludes the paper.

2 Formulation

The MMURP uses the following notation. Let $G = (V, A)$ be a complete and directed graph, where V represents the set of vertices and A the sets of arcs. Let D be the set of m depots and U the set of n candidate cities to be visited. Then, $V = U \cup D = \{1, \ldots, n, n+1, \ldots, n+m\}$, where $U = \{1, \ldots, n\}$ and $D = \{n+1, \ldots, n+m\}$. Each depot $k \in D$ has r_k MMUs, each one with the capacity to perform Q annual screenings. Given two cities $i, j \in V$, the distance between them is given by $c_{ij} \geq 0$. The female population of each city $i \in U$ demands q_i screenings per year. The MMU travel distance between two candidate cities should be lower than $distMax$.

The proposed model adapts the MDOVRP$_{2i-flv}$ formulation for the Multi-depot Open Vehicle Routing Problem (MDOVRP) presented in [15]. A new parcel to the objective function that aims at maximizing the screening demand served was added, and changes in the constraints that require visiting all cities to allow the non-attendance of some of them. Moreover, the number of vehicles per depot and the maximum travel distance allowed between two candidate cities were added. The following are the decision variables:

$x_{ij} = 1$, if a vehicle travels from node $i \in V$ to a node $j \in V$, 0 otherwise.
$w_j = 1$, if the screening demand of city $j \in U$ is served, 0 otherwise.
$u_{ij} > 0$, represents an upper bound on the MMU service capacity available when leaving a node $i \in V$ to $j \in V$.

The mathematical model proposed in this work for the bi-objective MMURP is defined as follows:

$$\min_{x} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \tag{1}$$

$$\max_{w} \sum_{j \in U} q_j w_j \tag{2}$$

subject to:

$$\sum_{i \in V, i \neq j} x_{ij} = w_j, \qquad \forall j \in U \tag{3}$$

$$\sum_{j \in U} x_{kj} \leq r_k, \qquad \forall k \in D \tag{4}$$

$$\sum_{j \in U} q_j \cdot w_j \leq \sum_{k \in D} r_k \cdot Q \tag{5}$$

$$c_{ij} \cdot x_{ij} \leq dist_{max}, \qquad \forall i \in U, j \in V, i \neq j \tag{6}$$

$$\sum_{i \in V, i \neq j} x_{ij} - \sum_{i \in U, i \neq j} x_{ji} \geq 0, \qquad \forall j \in U \tag{7}$$

$$x_{ij} + x_{ji} \leq 1, \qquad \forall i, j \in V, i \neq j \tag{8}$$

$$\sum_{j \in V} \sum_{k \in D} x_{jk} = 0 \tag{9}$$

$$\left(\sum_{i \in V, i \neq j} u_{ij} - \sum_{i \in V, i \neq j} u_{ji}\right) - q_j \geq -Q \cdot (1 - w_j), \qquad \forall j \in U \tag{10}$$

$$(Q - q_i) \cdot x_{ij} \geq u_{ij}, \qquad \forall i, j \in U \tag{11}$$

$$Q \cdot x_{kj} \geq u_{kj}, \qquad \forall k \in D, j \in U \tag{12}$$

$$x_{ij} \in \{0, 1\}, \qquad \forall i, j \in V \tag{13}$$

$$u_{ij} \geq 0, \qquad \forall i, j \in V \tag{14}$$

$$w_j \in \{0, 1\}, \qquad \forall j \in U \tag{15}$$

The objective function (1) aims to minimize the total traveled distance, and (2) aims to maximize the total screening demand served. Constraints (3) guarantee that the demand of a city j will be served only if it is visited once. Constraints (4) and (5) establish an upper bound for the number of MMUs leaving each depot and the demand for screenings that can be served. Constraints (6) bounds the maximum travel distance for an MMU between two cities, except for the first city, leaving a depot. Constraints (7) establish that when a location is visited, it can be the end of the route or go to another city, as long as it is not a depot. Constraints (8) forbid the use of the path ij and ji at the same time, and Constraint (9) prevents the return of any vehicle to a depot. Constraints (10) can be divided into two cases: if w_j is equal to 0, they are redundant; if w_j is equal to 1, they require a sufficient capacity to meet the demand at city j. Constraints (11) and (12) are upper bounds for the variables u. Constraints (13) to (15) define the domain of the decision variables.

3 An Hybrid NSGA-II Approach for the MMURP

The Non-dominated Sorting Genetic Algorithm II (NSGA-II) is a population-based meta-heuristic proposed in [7]. The method is an improved version of NSGA [24]. In this new version, the non-dominated sorting algorithm was improved. NSGA-II has an algorithmic complexity of $O(MN^2)$, whereas, in the NSGA, it was $O(MN^3)$ (where M is the number of objectives and N is the population size). Otherwise, NSGA-II is elitist, combining the parents and children of one generation and choosing the best ones for the next generation; and it is no longer necessary to define a sharing parameter to maintain the diversity that is maintained considering the fitness function and a crowded-comparison operator [7].

The NSGA-II used here was implemented as proposed in [7]. The improvements made are inside the *make_new_pop* procedure. Algorithm 1 presents the pseudo-code of the proposed *make_new_pop* procedure. It works with a list of crossover operators (CXL), a crossover probability (p_c), a list of mutation operators (MOL), a mutation probability (p_m), and the size of population (N). These operators are described in Subsect. 3.4.

A new population C and the lists of operators are initialized at lines 2–4. New individuals are added to C, (lines 6–17) while the new population is not full (line 5). With probability pc (line 6), a crossover operator is randomly chosen from the CXL (line 8) and applied to parents p_1 and p_2 (line 9), generating two children c_1 and c_2. For each child, a random mutation operator is chosen from MOL (line 12) and it is applied with probability pm (line 11). At line 13, the child is added to C. Finally, the new population C is returned (line 19).

3.1 Representation

The problem solution, named complete solution, is represented by a list of routes. Each route is denoted by a list, where the first position is the depot, and the others are the candidate cities in the visit order. However, in our algorithm, an

Algorithm 1. Algorithm to create a new population

function MAKE-NEW-POP(P)
  $C \leftarrow \emptyset$
  Initialize the Crossover Operator List (CXL)
  Initialize the Mutation Operator List (MOL)
  while $|C| \leq |P|$ **do**
    $p_1, p_2 \leftarrow choose\_parents(P)$
    if $random(0,1) \leq pc$ **then**
      Choose randomly a crossover operator $\mathcal{C}^{\varsigma} \in CXL$
      $c_1, c_2 \leftarrow \mathcal{C}^{\varsigma}(p_1, p_2)$
      for each $c \in \{c_1, c_2\}$ **do**
        if $random(0,1) \leq pm$ **then**
          Choose randomly a mutation operator $\mathcal{M}^{\varsigma} \in MOL$
          $c \leftarrow \mathcal{M}^{\varsigma}(c)$
        end if
        $C \leftarrow C \cup \{c\}$
      end for
    end if
  end while
  return C
end function

individual is represented as a visiting order, known as a giant tour. The giant tour is a permutation of candidate cities without the use of delimiters (depots).

The procedure to transform a giant tour into a complete solution is called *Split*, which will be presented in Subsect. 3.3. The inverse process, of transforming a complete solution in the giant tour, is called $Split^{-1}$ [19].

Let a complete solution with L different routes, where each route $\mathcal{R}^l = (d_l, i_1^l, i_2^l, \ldots, i_{b_l}^l)$, $\forall l \in L$, $b_l \in \mathbb{N}$, $i_j^l \in N$, $\forall j \in \{1, 2, \ldots, b_l\}$, and $d_l \in D$. A transformed individual is represented as a giant tour given by:

$$g = (i_1^1, i_2^1, \ldots, i_{b_1}^1, i_1^2, i_2^2, \ldots, i_{b_2}^2, \ldots, i_1^L, i_2^L, \ldots, i_{b_L}^L)$$

3.2 Initial Population

The initial population contains one solution given by a greedy constructive heuristic for the traveling salesman problem (TSP) that uses an insertion by the nearest neighbor [3], and the other solutions are random permutations of candidate cities.

3.3 Individual Evaluation

The individuals are evaluated using the Split procedure. This procedure consists of building a route that is not dominated by any other route with the same visiting order. On MMURP, a solution A will be dominated by a solution B, if $dem_B \geq dem_A$ and $dist_B < dist_A$ or $dem_B > dem_A$.

In the MMURP, each depot has a limited number of MMUs, which must be allocated to the routes. As the MMUs in each depot are resources required by other routes, the route construction problem becomes a resource-constrained shortest path problem (RCSPP). To solve this problem, we adapted a multi-label extension of Bellman's algorithm [19]. The label for a path turns a vector $L = (\phi, \pi | a_1, a_2, \ldots, a_p)$, where ϕ is the total distance, π is the screening demand met, and a_k is the number of MMUs of depot k used by this path. The splitting procedure for the MMURP is detailed in Algorithm 2.

Algorithm 2. Splitting algorithm for the MMURP

$\Lambda(0) \leftarrow \{(0, 0|0, \ldots, 0)\}$
for $i \leftarrow 1$ *to* n **do**
  $\Lambda(i) = \emptyset$
end for
for $i \leftarrow 1$ *to* n **do**
  for all depot d_k **do**
    for all label $L = (\phi, \pi | a_1, a_2, \ldots, a_p) \in \Lambda(i-1)$ **do**
      if $a_k + 1 \leq r_k$ **then**
        $tour\_distance \leftarrow c(d_k, T_i)$
        $tour\_demand \leftarrow q(T_i)$
        $W \leftarrow (\phi + tour\_distance, \pi + tour\_demand | a_1, \ldots, a_k + 1, \ldots, a_p)$
        if no label in $\Lambda(i)$ dominates W **then**
          delete in $\Lambda(i)$ all labels dominated by W
          $\Lambda(i) \leftarrow \Lambda(i) \cup \{W\}$
        end if
        $j \leftarrow i + 1$
        $stop \leftarrow false$
        repeat
          $tour\_demand \leftarrow tour\_demand + q(T_j)$
          if $c(T_{j-1}, T_j) \leq dist_{Max} \wedge (tour\_demand \leq Q)$ **then**
            $tour\_distance \leftarrow tour\_distance + c(T_{j-1}, T_j)$
            $W \leftarrow (\phi + tour\_distance, \pi + tour\_demand | a_1, \ldots, a_p)$
            if no label in $\Lambda(j)$ dominates W **then**
              delete in $\Lambda(j)$ all labels dominated by W
              $\Lambda(j) \leftarrow \Lambda(j) \cup \{W\}$
            end if
          else
            $stop \leftarrow true$
          end if
          $j \leftarrow j + 1$
        until $(j \geq n) \vee (stop = true)$
      end if
    end for
  end for
end for

Given a sequence of cities T_i, $\forall i \in \{1, \ldots, n\}$ to be served, with size n, the Algorithm 2 creates an auxiliary graph H with $n+1$ nodes. At each node i, we obtain a set of non-dominated solutions ($\Lambda(i)$), which met the screening demand of the first $i-1$ cities of the sequence. Many of these solutions are discarded using a dominance rule.

For example, consider that we have 2 depots, with $r = (2, 3)$, and a label $L = (150, 2500|1, 2)$. The label L represents a path with a distance of 150, meets the demand of 2500, and uses 1 and 2 MMUs of each respective depot. L dominates $(160, 2500|1, 2)$ and $(140, 2400|1, 2)$, which have, respectively, a lower total distance or meet a bigger demand with the same number of MMUs. L also dominates $(150, 2500|2, 2)$: the travel distance and demand met are equal, but this uses more MMUs than L. Finally, L is not comparable with $(160, 2500|0, 2)$: the travel distance of L is better but needs more MMUs on depot 1.

Initially, only node 0 has a label, which is equivalent to an empty path (lines 1–4). For each node i of H and each depot d_k, we iterate through each label of non-dominated solutions at node $i-1$, that is feasible to use one more vehicle from depot d_k (lines 5–8). At lines 9–11, a new route W is started. If W is not dominated by any solution of $\Lambda(i)$ (line 12), all solutions of $\Lambda(i)$ that is dominated by W are removed and W is inserted in $\Lambda(i)$ (lines 13–14). In the lines 16 and 17 the variables j and *stop* are initialized. In the loop from 18 to 31, routes that goes from city T_i to T_j, passing through $T_{i+1}, \ldots, T_{j-1}$ are built. A city j is added to the path until two consecutive cities have a distance greater than $dist_{Max}$ or the route screening demand is greater than Q (line 20). In lines 21–22, a new route W is started. If W is not dominated by any solution of $\Lambda(j)$ (line 23), all solutions of $\Lambda(j)$ that are dominated by W are removed and W is inserted in it (lines 24-25). If the distance between two cities exceeds the maximum distance or the screening demand met exceeds the maximum capacity of an MMU (line 20), the variable $stop \leftarrow true$ (line 28), which terminates the loop before j reaches n.

3.4 Genetic Operators

The crossover and mutation operators are defined as follows:

Crossover

As we have used the giant tour to represent the individuals and this representation is the same commonly used in TSP, we can take advantage of this, using the same recombination operators of TSP [18]. Thus, in our algorithm, 3 recombination strategies are used, which are chosen at random every time that parents need to be recombined: OX (Order Crossover), CX (Cycle Crossover) and PMX (Partial Mapped Crossover) [11].

Mutation

We use a list of mutation operators (MOL). Every time a mutation operation occurs, an operator is randomly chosen in MOL. The mutations used are 1, 2, or 3 exchanges between any two positions in the giant tour or a local search in intra-route or inter-route neighborhoods defined at the end of this section.

The local search algorithm used is the Randomized Variable Neighborhood Descent (RVND) [21,25], which is a VND [10] variant, where the order of use of neighborhoods in the search process is random, rather than predefined, as in VND.

The search performed by VND on a function f uses a predefined set of neighborhoods, $\mathcal{N} = \{\mathcal{N}_1, \mathcal{N}_2, \ldots, |\mathcal{N}|\}$. $\mathcal{N}_j(s)$ is the set of neighboring solutions of s in the jth neighborhood ($\mathcal{N}_j$). If s' is a local optimum concerning the neighborhood $\mathcal{N}_j(s)$, then $f(s') \leq f(s)$ for all $s' \in \mathcal{N}_j(s)$.

VND performs the search with a systematic change of neighborhoods until the current solution cannot be improved. The search starts using the neighborhood $\mathcal{N}_1$ and remains there as long as there is an improvement. If there is no improvement, the neighborhood is replaced by the next one following the pre-established order. The neighborhood is reset to the first one whenever there is an improving solution. The search continues until there are no improvements in any of the neighborhoods. Therefore, the solution returned by the method is a local optimum concerning all neighborhoods.

In RVND, the order in which the neighborhoods are used is random. Every time that a better solution is found, the list of neighborhoods is reset and a new neighborhood is chosen at random. However, the search ends in the same way as VND, when the solution is a local optimum for all neighborhoods [25].

The neighborhoods used to explore the solution space of the problem during the local search accept only feasible solutions and are defined below.

Intra-route neighborhoods: This set of neighborhoods is defined through the following moves:
- **Shift -** $\mathcal{N}'^{(1)}$: a candidate city is removed from its position and reinserted into another in the same route.
- **Swap -** $\mathcal{N}'^{(2)}$: two candidate cities i and j from a route r are swapped.

Inter-route neighborhoods: Five moves define this set of neighborhoods:
- **Shift(1,0) -** $\mathcal{N}^{(1)}$: a candidate city i is transferred from route r_1 to route r_2.
- **Swap(1,1) -** $\mathcal{N}^{(2)}$: a candidate city i of a route r_1 is exchanged with the city j of a route r_2.
- **Shift_Depot -** $\mathcal{N}^{(3)}$: a route r is moved from a depot D_1 to a depot D_2.
- **Swap_Depot -** $\mathcal{N}^{(4)}$: the depot D_1 of a route r_1 is exchanged with the depot D_2 of a route r_2.
- **Insert_Not_Visited -** $\mathcal{N}^{(5)}$: a candidate city not visited i is inserted into a route r.

4 Computational Experiments

The proposed NSGA-II algorithm was implemented in C++ and compiled using GNU c++ 11.3.0 with options -O3 and -march=native. All experiments were performed on a computer with an Intel Core i7-4790 CPU 3.60 GHz x 4 running Ubuntu 22.04.1 operating system. A single thread was used for all tests.

Two versions of the NSGA-II were implemented. The first version (V1) is the full version described in Sect. 3. The second version (V2) differs from the first by not applying local search.

The hypervolume metric was used to evaluate the performance of the two versions of the NSGA-II algorithm. This metric has the characteristic of being strictly monotonic, i.e., a better Pareto front approximation will have a greater hypervolume value, assuming that all points of both fronts dominate the reference point [2]. To calculate the hypervolume, the algorithm proposed in [9] was used, and it is available at https://lopez-ibanez.eu/hypervolume.

In order to normalize the results, we transform the second objective into a minimization problem and sum $maxDem = Q \sum_{k \in D} r_k$, which is the maximum demand that can be met. Thus, the second objective value is always positive and ranges from 0 to $maxDem$. This way, the hypervolume was calculated and normalized by the product of $maxDem$ and $maxDist$. The calculation of $maxDist$ was done by a constructive heuristic, which draws routes with maximum distance, obeying the capacity restrictions of the MMUs.

To evaluate the two versions of the implemented algorithms, 13 instances based on real data were generated[1]. The number of mammography units used was obtained in [27] and it is related to data from the state of Minas Gerais on September/2019. The travel distances between the cities were obtained from the work of [5]. As the data on where each woman did the mammogram is not available, we adapted the models present in the studies [4,22] to be possible to simulate the assistance provided in different cities in the state of Minas Gerais. We used the number of mammography units present in each municipality and simulated which city with a mammography unit installed covers the cities without one, aiming to maximize the number of served women. We added the constraint that cities with a mammography unit should first meet their own demand and, if they have idle capacity, they will cover a neighboring city; cities can only serve neighbors that are within their health region; and one city can cover a percentage of another city's demand.

Table 1 shows the summary of instance characteristics. The first column brings the instance id, the second column the number of cities, and the third column shows the number of depots. The fourth column shows the maximum service capacity of each MMU, and the fifth column shows the number of MMUs per depot. Columns 6 and 7 bring the reference point used to calculate the hypervolume.

Our hybrid NSGA-II implementation requires a few parameters to be defined in advance. For the experiments, these parameters were determined using the irace [17] tool. To use the irace, it is necessary to define a set of instances and a set of values for each parameter. Table 2 shows the range of values considered by irace and the returned parameter values. Table 3 shows the results obtained in the experiments, where each version of the algorithm ran 30 times. The first column

[1] All used instances are available at https://bit.ly/3mxnIbl.

Table 1. Instance characteristics

ID	#cities	#depots	MMU Capacity	#vehicles per depot	Max distance	Max demand
i01	350	2	5069	1, 1	3758	10138
i02	350	2	5069	8, 8	28262	81104
i03	350	2	5069	16, 16	47531	162208
i04	350	2	5069	24, 24	66271	243312
i05	350	2	5069	32, 32	81868	324416
i06	350	2	5069	40, 40	91329	405520
i07	350	2	5069	48, 48	94308	486624
i08	350	2	5069	56, 56	94308	567728
i09	350	2	10138	1, 1	4132	20276
i10	350	2	10138	8, 8	35072	162208
i11	350	2	10138	16, 16	57341	324416
i12	350	2	10138	24, 24	73380	486624
i13	350	2	10138	32, 32	78944	648832

Table 2. Parameter values considered and returned by irace for the hybrid NSGA-II.

Param.	Range	Returned values
N	{25, 30, 35, 40, 45, 50}	50
Gen	{500, 1000, 1500, 2000}	1500
p_m	{0.05, 0.075, 0.10, 0.125, 0.15, 0.20}	0.15
p_c	{0.85, 0.90, 0.95}	0.90
CXL	{{OX}, {CX}, {PMX}, {OX, CX, PMX}}	{OX, CX, PMX}
MXL	{{1 Swap}, {2 Swaps}, {3 Swaps}, {1, 2, 3 Swaps, LS}}	{1, 2, 3 Swaps, LS}

shows the instance id. Columns 2–3 show the values obtained for algorithm V1 and columns 4-5 for algorithm V2. For each algorithm, we have the average value and standard deviation of hypervolume and the average value of execution time for each instance.

Observing the results of Table 3, we can notice that the local search has an important role in providing a better quality set of solutions when using hypervolume as a metric.

Table 3. Results of NSGA-II with LS (V1) and without LS (V2)

ID	V1		V2	
	HV	Time (s)	HV	Time (s)
i01	**0.969 (4.8E-04)**	9.58	0.955 (8.3E-03)	2.89
i02	**0.919 (2.4E-03)**	68.64	0.857 (1.2E-02)	9.88
i03	**0.857 (2.6E-03)**	219.31	0.785 (8.6E-03)	42.79
i04	**0.804 (4.5E-03)**	471.49	0.728 (7.2E-03)	126.92
i05	**0.743 (7.0E-03)**	944.94	0.690 (5.0E-03)	301.56
i06	**0.630 (4.8E-03)**	1432.96	0.619 (4.3E-03)	499.90
i07	0.509 (6.4E-03)	2108.69	**0.509 (3.1E-03)**	593.28
i08	0.405 (1.9E-02)	2318.49	**0.413 (4.5E-03)**	572.47
i09	**0.963 (3.9E-04)**	11.14	0.952 (5.7E-03)	2.89
i10	**0.911 (3.2E-03)**	107.06	0.847 (1.5E-02)	12.70
i11	**0.832 (4.1E-03)**	377.21	0.740 (1.4E-02)	73.89
i12	**0.692 (2.0E-03)**	916.06	0.653 (3.3E-03)	255.33
i13	0.510 (4.8E-03)	1381.72	**0.511 (1.8E-03)**	440.57

5 Conclusions

This paper addressed a variant of the Multi-depot Open Vehicle Routing Problem, the Mobile Mammography Unit Routing Problem (MMURP), with the objectives of maximizing the demand for screenings and minimizing the total travel distance by the MMUs. We introduced a mathematical programming formulation for the problem and developed a hybrid algorithm based on the Non-dominated Sorting Genetic Algorithm II (NSGA-II) for treating it. The NSGA-II uses three crossover operators with a Randomized Variable Neighborhood Descent (RVND) as one of the mutation operators. For testing the hybrid NSGA-II, we used 13 instances based on real data and compared its results with those of a version of NSGA-II without the local search as a mutation operator.

Using the hypervolume metric to compare the sets of non-dominated solutions, on average, the NSGA-II with the local search yielded better results in 10 of 13 instances. On the other hand, the NGSA-II without the local search required less than 30% of the computational time compared to its complete version.

Acknowledgments. The authors are grateful for the support provided by the Universidade Federal de Ouro Preto, and by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES, Finance Code 001), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, grants 428817/2018-1 and 303266/2019-8), and Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG, grant PPM CEX 676/17).

References

1. Amaral, P., Luz, L., Cardoso, F., Freitas, R.: Spatial distribution of mammography equipment in brazil. Revista Brasileira de Estudos Urbanos e Regionais **19**(2), 326 (2017). https://doi.org/10.22296/2317-1529.2017v19n2p326
2. Audet, C., Bigeon, J., Cartier, D., Le Digabel, S., Salomon, L.: Performance indicators in multiobjective optimization. Eur. J. Oper. Res. **292**(2), 397–422 (2021). https://doi.org/10.1016/j.ejor.2020.11.016
3. Bellmore, M., Nemhauser, G.L.: The traveling salesman problem: a survey. Oper. Res. **16**(3), 538–558 (1968). https://doi.org/10.1287/opre.16.3.538
4. de Campos, M.V.A., de Sá, M.V.S.M., Rosa, P.M., Penna, P.H.V., de Souza, S.R., Souza, M.J.F.: A mixed linear integer programming formulation and a simulated annealing algorithm for the mammography unit location problem. In: ICEIS, pp. 428–439 (2020). https://doi.org/10.5220/0009420704280439
5. de Carvalho, L.R., do Amaral, P.V.M., Mendes, P.S.: Matrizes de distâncias e tempo de deslocamento rodoviário entre os municípios brasileiros : uma atualização metodológica para 2020. Textos para Discussão Cedeplar-UFMG 630, Cedeplar, Universidade Federal de Minas Gerais (2021). https://ideas.repec.org/p/cdp/texdis/td630.html
6. Corrêa, V.H.V., Costa Lima, B.J., Silva-e-Souza, P.H., Penna, P.H.V., Souza, M.J.F.: Localização de mamógrafos: um estudo de caso na rede pública de saúde. In: Anais do Simpósio Brasileiro de Pesquisa Operacional, p. 84874. Rio de Janeiro, Brazil (2018). http://bit.ly/3ZDmusx
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002). https://doi.org/10.1109/4235.996017
8. Ferlay, J., et al.: International agency for research on cancer (2020). https://gco.iarc.fr/today. Accessed 18 Mar 2023
9. Fonseca, C., Paquete, L., Lopez-Ibanez, M.: An improved dimension-sweep algorithm for the hypervolume indicator. In: 2006 IEEE International Conference on Evolutionary Computation, pp. 1157–1163 (2006). https://doi.org/10.1109/CEC.2006.1688440
10. Hansen, P., Mladenović, N., Todosijević, R., Hanafi, S.: Variable neighborhood search: basics and variants. EURO J. Comput. Opt. **5**(3), 423–454 (2017). https://doi.org/10.1007/s13675-016-0075-x
11. Hussain, A., Muhammad, Y., Nauman Sajid, M., Hussain, I., Mohamd Shoukry, A., Gani, S.: Genetic algorithm for traveling salesman problem with modified cycle crossover operator. Computational Intelligence and Neuroscience 2017, pp. 1–7 (2017). https://doi.org/10.1155/2017/7430125
12. IBGE: Áreas territoriais (2021). https://bit.ly/3Mh1TqO. Accessed 23 Mar 2023
13. INCA: A situação do câncer de mama no brasil: síntese de dados dos sistemas de informação (2019). https://bit.ly/3ocLOZt
14. INCA: Estatísticas de câncer (2022). https://www.inca.gov.br/numeros-de-cancer. Accessed 3 Jul 2022
15. Lalla-Ruiz, E., Mes, M.: Mathematical formulations and improvements for the multi-depot open vehicle routing problem. Optim. Lett. **15**(1), 271–286 (2020). https://doi.org/10.1007/s11590-020-01594-z
16. Lenstra, J.K., Kan, A.R.: Complexity of vehicle routing and scheduling problems. Networks **11**(2), 221–227 (1981). https://doi.org/10.1002/net.3230110211

17. López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L.P., Birattari, M., Stützle, T.: The Irace package: iterated racing for automatic algorithm configuration. Oper. Res. Perspect. **3**, 43–58 (2016). https://doi.org/10.1016/j.orp.2016.09.002
18. Prins, C.: A simple and effective evolutionary algorithm for the vehicle routing problem. Comput. Oper. Res. **31**(12), 1985–2002 (2004). https://doi.org/10.1016/S0305-0548(03)00158-8
19. Prins, C., Lacomme, P., Prodhon, C.: Order-first split-second methods for vehicle routing problems: a review. Transp. Res. Part C Emer. Technol. **40**, 179–200 (2014). https://doi.org/10.1016/j.trc.2014.01.011
20. Rosa, O.A.S., Rosa, P.M., Penna, P.H.V., Souza, M.J.F.: Um algoritmo construtivo para o problema de roteamento de unidades móveis de mamografia. Simpósio Brasileiro de Pesquisa Operacional (2020). http://bit.ly/3lVa5Ck
21. Souza, M.J.F., Coelho, I.M., Ribas, S., Santos, H.G., Merschmann, L.H.C.: A hybrid heuristic algorithm for the open-pit-mining operational planning problem. Eur. J. Oper. Res. **207**(2), 1041–1051 (2010). https://doi.org/10.1016/j.ejor.2010.05.031
22. Souza, M.J.F., Penna, P.H.V., Moreira de Sá, M.V.S., Rosa, P.M.: A vns-based algorithm for the mammography unit location problem. In: International Conference on Variable Neighborhood Search. pp. 37–52. Springer (2020). DOI: 10.1007/978-3-030-44932-2_3
23. Souza, M., Penna, P., Sá, M., Rosa, P., Monteiro, J., Lisboa, M.: Localização de mamógrafos: formulações e estudo preliminar de caso de Rondônia. In: Anais do LI Simpósio Brasileiro de Pesquisa Operacional, vol. 51, p. 107698. SOBRAPO, Galoá, Limeira (SP) (2019). https://bityli.com/HAEBCtEkv. Accessed 31 Aug 2022
24. Srinivas, N., Deb, K.: Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evol. Comput. **2**(3), 221–248 (1994). https://doi.org/10.1162/evco.1994.2.3.221
25. Subramanian, A., Drummond, L., Bentes, C., Ochi, L., Farias, R.: A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. Comput. Oper. Res. **37**(11), 1899–1911 (2010). https://doi.org/10.1016/j.cor.2009.10.011
26. SUS: Série parâmetros sus - volume 1 - caderno 1 - republicado 1 (2017). https://bit.ly/41tfmjD
27. SUS: Cnes - recursos físicos (2019). http://bit.ly/40BKfm9. Accessed 28 Mar 2023
28. WHO: Cancer, February 2022. https://www.who.int/news-room/fact-sheets/detail/cancer. Accessed 18 Mar 2023