Metaheuristics for a bi-objective location-routing-problem in waste collection management

Hamed Farrokhi-Asl, Reza Tavakkoli-Moghaddam, Bahare Asgarian & Esmat Sangari

To cite this article: Hamed Farrokhi-Asl, Reza Tavakkoli-Moghaddam, Bahare Asgarian & Esmat Sangari (2016): Metaheuristics for a bi-objective location-routing-problem in waste collection management, Journal of Industrial and Production Engineering, DOI: 10.1080/21681015.2016.1253619

To link to this article: http://dx.doi.org/10.1080/21681015.2016.1253619

Published online: 18 Nov 2016.
Metaheuristics for a bi-objective location-routing-problem in waste collection management

Hamed Farrokhi-Asl, Reza Tavakkoli-Moghaddam, Bahare Asgarian, and Esmat Sangari

ABSTRACT
A waste collection problem is one of the most important problems in reverse logistics. This paper presents a mathematical model for waste collection problem and considers waste collection from customers’ location and waste disposal in compatible disposal facilities. The fleet of vehicles is heterogeneous and vehicles have multi-separated compartments. Vehicles have different capacity, different travel time and distance limitations, and different variable and fixed costs. Vehicles start their route from depots, collect wastes from customers’ location, and move to disposal centers to unload waste in the related disposal facility, and then return to depots. We should choose appropriate location for depots and disposal facilities from potential locations with respect to economic and social objectives. To solve the model, two well-known multi-objective evolutionary algorithms, namely non-dominated sorting genetic algorithm (NSGA-II) and multi-objective particle swarm optimization (MOPSO) are proposed. Finally, the computational results are compared and analyzed.

1. Introduction
In today’s competitive market, one of the most important problems in the supply chain management is to design the logistic system suitably so that the customers’ demand is satisfied on time and properly. A facility location problem (FLP) as a strategic policy and a vehicle routing problem (VRP) as a tactical decision are two fundamental problems that should be investigated; however, as Salhi and Rand [1] indicated, solving these two problems separately may lead to sub-optimal solutions. Hence, the location-routing problem (LRP) which considers both the mentioned problems interdependently and simultaneously is more suitable for solving an actual problem. More than 50 years have elapsed since Boventer [2] introduced this problem for the first time. A vast survey of the LRP literature was carried out by Prodhon and Prins [3]. In the classic LRP, the location of a depot, in which the vehicles start their travel, and the routes assigned to these vehicles to serve customers are determined simultaneously in order to minimize costs.

There are variants of LRPs with different assumptions and constraints. For example, one type of the LRP, called capacitated LRP (CLRP), is to assume that depots and vehicles are capacitated. Other extensions of the problem consider time window constraints, multiple periods, or various attribute of vehicles and customers. Also, the problem can be investigated in a discrete, continuous, or network environment. Lopes et al. [4] proposed a taxonomical analysis of LRPs considering physical aspects of the models, algorithmic approaches, and the perspectives of objective functions.

In this study, a novel variant of the LRP is taken into consideration. One of the most important applications of the LRP (i.e. waste collection) is considered in this paper to make the research more applicable and for real-world applications. We investigate the multi-depot LRP when vehicles are assumed to be multi-compartment and capacitated. In other words, each vehicle has different compartments for each type of waste and each compartment has a limited capacity. Also, the close routes mean that the routes in which vehicles should return to the depot, they start their route. We suppose that the fleet of vehicles is heterogeneous and vehicles have some traveling time and distance limitations. In addition to an economical objective, a social objective is addressed in this paper. The economical objective is to minimize the cost of serving customers with regards to variable and fixed costs of depots, vehicles, and routes. The social objective aims to establish disposal facilities in the locations which are far from urban areas (customers).
The capacitated multi-depot location-routing problem (CMDLRP) is a NP-hard problem, since it is a variant of VRP and FLP that are two well-known NP-hard problems. It means that exact solution methods can only be exploited for solving small size problems. For large size problems, it takes too much CPU time and this redundant time makes finding an optimal solution impossible. In these cases, as can be seen in the literature review, using heuristic and meta-heuristic algorithms is more practical. In this study, two meta-heuristics (i.e. MOPSO and NSGA-II) are developed to solve the proposed bi-objective problem and the results obtained by these two methodologies are compared with respect to comparison metrics.

The rest of the paper is organized as follows. In Section 2, the relevant literature is reviewed. The problem and mathematical model is described in Section 3. The concept of a non-dominated solution is illustrated in Section 4. In Section 5, the developed meta-heuristics are explained in detail. Comparisons and discussion on the experimental results are presented in Section 6. Finally, the study is ended by conclusions and future research in Section 7.

2. Literature review

Jacobsen and Madsen [5] introduced three heuristic procedures for solving the location-routing problem for designing a newspaper distribution system. Laporte et al. [6] presented an integer-linear programming model for the capacitated location-routing problem considering a specified capacity to any site. The objective function was defined as the summation of depot operation, vehicle acquisition, and routing cost elements that should be minimized. Srivastava and Benton [7] investigated the effects of some environmental factors (i.e. distribution of customers, cost structure, and number of depot sites) on the performance of three different heuristics in the LRP when designing a physical distribution system. They concluded that the selection of a solution procedure should be on the basis of the environmental and potential situations of the system under consideration.

Tuzun and Burke [8] developed an efficient two-phase tabu search approach to deal with the NP-hard nature of the LRP with uncapacitated depots and capacitated routes. They indicated that this architecture is capable of generating good solutions without inordinate computational efforts.

Wu et al. [9] developed a solution method for solving large-scale location-routing problems based on simulated annealing (SA). They considered the capacitated routes and depots, homogenous or heterogeneous fleet types, and restricted number of vehicles. The main problem was decomposed into location-allocation and vehicle routing sub-problems to be solved by the proposed solution algorithm. Gendreau et al. [10] investigated a heterogeneous fleet of vehicles in routing problem. Their fleet of vehicles was different in capacity, fixed, and variable cost. Also, Li et al. [11] applied record-to-record heuristic approach to solve a VRP with heterogeneous fleet of vehicles. The types of vehicles differ in capacity, fixed, and variable cost. They assumed that the number of vehicles of each type is known and a decision should be made about how to use this fleet of heterogeneous vehicles efficiently. Generally, the fleet of vehicles is considered as a single compartment, but it is more realistic to consider vehicles with several compartments and specific capacity for each type of commodity. [12,13] Recently, Rabbani et al. [14] proposed a new mathematical model for collection of waste from generation nodes with consideration of heterogeneous and multi-compartment vehicles, simultaneously. They used a simple and hybrid genetic algorithm to tackle the presented problem.

Yu and Maghfroh [15] applied a variable neighborhood search with path-relinking (VNSPR) for solving the LRP. They tested operation of their proposed heuristic approach on three sets of well-known LRP benchmark instances and results showed that proposed algorithm can reach solutions with good quality. Barreto et al. [16] formulated a discrete LRP considering two levels (i.e. capacitated distribution centers and ordered customers) when the capacity of vehicles was assumed to be limited. To solve the problem, a clustering analysis was applied as a heuristic solution method. Manzour-Al-Ajdad [17] presented a hierarchical solution approach based on some local search procedures, such as end-points of the routes, to solve the single-facility LRP in the planar environment considering capacitated vehicles and Euclidean distance. Nekooghadirli et al. [18] presented a new bi-objective location-routing-inventory model with the probabilistic traveling time among customers and customers’ uncertain demands. They proposed four multi-objective meta-heuristics, namely multi-objective imperialist competitive algorithm (MOICA), multi-objective parallel simulated annealing (MOPSA), NSGA-II and Pareto-archived evolution strategy (PAES). Jarboui et al. [19] developed variable neighborhood search (VNS) to deal with the LRP when there are multiple depots with a limited capacity and one uncappeditated vehicle can be assigned to per open depot. They incorporated the variable neighborhood descent (VDN) in the proposed VNS as local search. They showed that the presented heuristic method has a good performance in comparison to other heuristics mentioned in the literature.

Cappanera et al. [20] formulated the LRP of obnoxious activities in a discrete environment in order to find the best location for obnoxious facilities and the best routes for obnoxious materials transportation simultaneously. To solve this problem, they used the Lagrangean relaxation method to decompose the problem into location
and routing sub problems. Hazardous waste can be defined as the useless materials that have a bad influence on environment and human’s life. So, appropriate planning for management of hazardous waste can be advantageous for environment and human life. Alumur and Kara [21] stated that a waste can be characterized as hazardous if it has any of the following attributes: ignitability, corrosiveness, reactivity or toxicity. Caballero et al. [22] considered three types of objective function for transportation risk. The first one considered rejection of towns that trucks cross from it in their way. The second one was equity distribution of damage between the towns. The last one was about towns that are near to incineration plants which is undesirable facility. According to the presented literature review, the current paper has some characteristics which makes it different from other papers. The multi-depot LRP with the heterogeneous vehicles assumed to be multi-compartment and capacitated in length and loads are investigated for the first time. Also, applying this model on the managerial problem such as waste collection management with two conflicted objectives makes the paper more applicable for real-world cases.

3. Problem description

The collection of waste problem in an urban area is mathematically formulated as a bi-objective model. There are some demand points, in which customers generate wastes and request for vehicles to collect these wastes and transport them to disposal centers. A fleet of capacitated vehicles with multi compartments (e.g. capacity for each type of waste such as plastic, bottle, glass, food and paper [12]) gather wastes, separately. It is supposed that the vehicles have different capacities for each type of waste and a limited allowable operating time and traveled distance for each routes, and these limitations are different for each vehicle. Therefore, we can say that the fleet of vehicles is heterogeneous. Each vehicle starts from a depot, travels toward customers, collects wastes from customers’ location, and then takes them to all the disposal centers which are compatible with wastes. We suppose that all types of wastes are generated in each generation node and each type of waste must be disposed in compatible disposal facility. Each customer is served by only one vehicle. Capacity of depots is limited. Each customer generates all types of waste. The depots are considered capacitated meaning that the number of vehicles sent from each depot is limited.

The problem is to determine the best location of depots and disposal facilities on a planar surface and the best vehicle routes simultaneously. Two objective functions are considered in this paper. The first one is the minimization of economic cost including opening cost of depots and disposal facilities, fixed and variable costs of the routes for wastes collection. The second objective aims to maximize the distance between customers and disposal facilities. The problem considers a discrete environment meaning that each depot and disposal centers can be located in predetermined potential locations on the planar surface. The schematic figure of this problem is shown in Figure 1. To illustrate the problem addressed in this paper, a mathematical formulation is provided. The presented model aims to facilitate understanding of objectives and constraints of the problem. It should be mentioned that the presented mathematical formulation can be solved by means of some commercial software packages such as GAMS, Lingo, and CPLEX in order to find exact solutions to the problem. However, regarding NP-hard nature of the problem, applying metaheuristic algorithms to tackle the problem especially in large-scale problems is rational. Next subsections are designated to mathematical formulation.

3.1. Problem assumptions

(1) The demand of each customer is deterministic.
(2) Customers’ location are known.
(3) Vehicles’ capacity are limited.
(4) Vehicles are heterogeneous.
(5) There are some types of waste.
(6) Each type of waste must be disposed in compatible disposal facility.
(7) Each customer is served by only one vehicle.
(8) Capacity of depots is limited.
(9) Each customer generates all types of waste.

3.2. Notations

Sets:
- \( c, c' \) Index of customers’ node
- \( d, d' \) Index of depots
- \( f \) Index of potential locations for disposal centers
- \( w \) Index of waste types
- \( k \) Index of vehicles
- \( o \) Index of depot and customers node
- \( q \) Index of customers and disposal centers nodes
- \( r \) Index of disposal centers and depots
- \( ij \) Index of all nodes in the network
Parameters:
\[ d_{ij} \] Distance between node \( i \) and node \( j \)
\[ t_{ij} \] Travel time between node \( i \) and node \( j \)
\[ f_k \] Fixed cost associated with using vehicle \( k \)
\[ v_k \] Variable cost of vehicle \( k \) per unit of time
\[ q_w \] The demand of customer \( c \) for waste type \( w \) collection
\[ c_{wk} \] Loading time per unit of waste \( w \) at customer \( c \) location by vehicle \( k \)
\[ C_{dkw} \] Maximum capacity of vehicle type \( k \) for waste \( w \)
\[ q^*_k \] Maximum allowable route length for vehicle type \( k \)
\[ \eta_k \] Maximum allowable time to serve all customers in a route served by vehicle \( k \)
\[ \Omega_d \] Maximum capacity of depot for vehicles in potential location \( d \)
\[ \pi_k \] Fixed cost of opening disposal facility with technology compatible with waste type \( w \) in potential location \( f \)
\[ \alpha_d \] Fixed cost of opening depot in potential location \( d \)
\[ M^* \] Great value

Decision variables:
\[ x_{ik} \] If vehicle \( k \) travels directly from node \( i \) to \( j \), \( x_{ik} = 1 \); otherwise, it is zero.
\[ z_{ik} \] If vehicle \( k \) travels to disposal facility \( f \), \( z_{ik} = 1 \); otherwise, it is zero.
\[ y_{ijw} \] If disposal facility with technology \( w \) is opened in potential location \( f \), \( y_{ijw} = 1 \); otherwise, it is zero.
\[ o_d \] If depot is opened in potential location \( d \), \( o_d = 1 \); otherwise, it is zero.
\[ U_{dkw} \] Continuous variable that represents the load of compartment \( w \) of vehicle \( k \) just after leaving node \( i \).

3.3. Mathematical formulation

\[
\min Z_1 = \sum_{k} \sum_{c} \sum_{d} f_k x_{dck} + \sum_{k} \sum_{c} v_k t_{ij} x_{ijk} + \sum_{f} \sum_{w} \pi_{fw} y_{ijw} + \sum_{d} \pi_d o_d
\]

\[
\max Z_2 = \sum_{w} \sum_{c} \sum_{ij} d_{ij} y_{ijw}
\]

s.t.

\[
\sum_{k} \sum_{c} x_{dck} = 1 \quad \forall c
\]

\[
\sum_{k} \sum_{q} x_{cqw} = 1 \quad \forall c
\]

\[
\sum_{d} \sum_{d'} x_{dd'k} = 0 \quad \forall k
\]

\[
\sum_{c} \sum_{d} x_{dck} = z_{ik} \quad \forall f, k
\]

\[
\sum_{o} \sum_{c} f_{oc} x_{ock} + \sum_{w} \sum_{c} c_{qck} q_{cw} z_{ck} \leq \eta_k \quad \forall k
\]

\[
\sum_{q} x_{qk} = \sum_{f} x_{fck} \quad \forall f, k
\]

\[
\sum_{f} x_{fck} = \sum_{c} x_{ckk} \quad \forall d, k
\]

\[
\sum_{k} \sum_{c} x_{dck} \leq \Omega_d o_d \quad \forall d
\]

\[
U_{dkw} = 0 \quad \forall d, k, w
\]

\[
U_{dkw} + q_{cw} - M(1 - x_{cwk}) \leq U_{cwk} \quad \forall c, c', k, w
\]

\[
q_{cw} \leq \sum_{k} U_{dkw} \leq \sum_{k} \sum_{c} \sum_{o} x_{ock} C_{cwk} \quad \forall c, w
\]

\[
\sum_{k} \sum_{f} \sum_{d} x_{dck} = 0
\]

\[
\sum_{k} \sum_{d} \sum_{c} x_{cwk} = 0
\]

\[
\sum_{q} x_{qk} = 1 \quad \forall f, k
\]

\[
\sum_{f} y_{ijw} = 1 \quad \forall w
\]

\[
x_{ik} = \{0, 1\} \quad \forall i, j, k
\]

\[
z_{ck} = \{0, 1\} \quad \forall c, k
\]

\[
y_{ijw} = \{0, 1\} \quad \forall f, w
\]

\[
o_d = \{0, 1\} \quad \forall d
\]

\[
U_{dkw} \geq 0 \quad \forall i, k, w
\]

Objective function (1) refers to economic cost. In term 1 of the objective function, the fixed cost associated with vehicles is considered. In the next term, the variable cost of traveling is considered. In term 3, we consider the fixed cost of opening disposal facilities in potential locations. In term 4, costs of opening depots in potential locations are considered. In objective function (2), social negativity of building of disposal facilities is addressed. A disposal facility is an obnoxious center and really should be far away from the customers' area.

Constraints (3) and (4) ensure that every customer is allocated just one path meaning that if one vehicle enters one node, this vehicle must leave this node. Constraints (5) ban traveling between the opened depots. Constraints (6) specify the relationship between two decision variables. These constraints also satisfy that all vehicles must visit all disposal facilities opened in potential locations. Constraints (7) satisfy time restriction to serve the customers in each route. Constraints (8) satisfy that a vehicle enters disposal facilities' node...
must leave it. These limitations for depots’ nodes are considered in constraints (9). Amounts of vehicles that leave each depot should never trespass from the capacity of a depot. Constraints (10) consider the capacity of each depot. Also, these constraints guarantee that customers are assigned to depots that are opened. Constraints (11) to (13) are lifted Miller–Tucker–Zemlin (MTZ) sub-tour elimination constraints for a classical VRP. In our problem, these three constraints are modified to guarantee the sub-tour elimination. Additionally, constraints (13) consider the capacity of vehicles in each route. Each route length should never exceed from allowable route length and Constraints (14) guarantee these limitations. Constraints (15) guarantee that vehicles cross from disposal facilities that are opened. Moving directly from depots to disposal facilities is prohibited. This restriction is considered in constraints (16). Constraints (17) ban moving from customers to depots before crossing from disposal facilities. Constraints (18) and (19) represent that all vehicles enter each disposal facility only once. Constraints (20) represent that only one disposal facility for each type of waste should be established. Constraints (21) to (25) specify the ranges of the variables.

4. Non-dominated solutions

For a single-objective mathematical model, optimization methods are capable for finding solution(s) optimizing the objective function; however, many real-world problems are multi-objective and the trade-off between objectives should be considered. When there is more than one objective in conflict with each other, it is not possible to find a solution that optimizes the objectives, simultaneously. In such problems, a trade-off among solutions should be performed to determine rational solutions in the feasible area that represent best compromises among the objective functions.[23]

To understand the context of optimality in multi-objective problems, it is necessary to be familiar with some basic concepts that are brought in the following. The multi-objective mathematical model is defined by:

\[
\begin{align*}
\min_{\bar{X}} \quad & f(\bar{X}) = (\bar{f}_1(\bar{X}), \bar{f}_2(\bar{X}), \ldots, \bar{f}_m(\bar{X})) \\
\text{subject to} \quad & g_i(\bar{X}) \leq 0 \quad i = 1, 2, \ldots, m \\
& h_j(\bar{X}) = 0 \quad j = 1, 2, \ldots, p
\end{align*}
\]

where \(\bar{f}_i(\bar{X})\) is the vector of objective functions, \(\bar{X}\) is the vector of variables, \(g_i(\bar{X})\) and \(h_j(\bar{X})\) are the constraint of the problem. Then, we provide the following definitions according to [24]:

**Definition 1:** Suppose that \(\bar{X} = (x_1, x_2, \ldots, x_k)\) and \(\bar{Y} = (y_1, y_2, \ldots, y_l) \in \mathbb{R}^d\), \(\bar{X}\) dominates \(\bar{Y}\) if \(\bar{f}_i(\bar{X}) \leq \bar{f}_i(\bar{Y})\), \(i = 1, 2, \ldots, n\) and \(\bar{f}_j(\bar{X}) < \bar{f}_j(\bar{Y})\), \(\exists i = 1, 2, \ldots, n\). It is called Pareto dominance which is denoted by \(\bar{X} < \bar{Y}\).

**Definition 2:** \(\bar{X}^* \in S(\bar{X}\text{ is the feasible region})\) is Pareto optimal if \(f_i(\bar{X}^*) \leq f_i(\bar{X}), \forall i = 1, 2, \ldots, K\) and for at least one \(i, f_i(\bar{X}^*) < f_i(\bar{X})\). In other words, \(\bar{X}^*\) is Pareto optimal if there is no other feasible vector \(\bar{X}\) that would decrease one objective function without increasing at least one other objective function.

**Definition 3:** The Pareto optimal set \(P^*\) for the multi-objective problem \(f(\bar{X})\) is defined by:

\[P^* = \{X \in S|\bar{X} \text{ is Pareto optimal}\}\]

**Definition 4:** Given the multi-objective problem with the objective function \(f(\bar{X})\) and Pareto-optimal set \(P^*\), the Pareto front \(PF^*\) is defined as follows:

\[PF^* = \{f(\bar{X})|\bar{X} \in P^*\}\]

5. Methodology

A decomposition of the LRP into facility location problem (FLP) and vehicle routing problem (VRP) was the most popular approach in the early studies in the literature. Salhi and Rand [1] showed that solving these two problems independently may lead to sub-optimal solutions. This approach does not consider the trade-off between these two phases of the problem (i.e. location and routing phases). Hence, in this paper, we consider location and routing problems simultaneously. Figure 2 shows a flowchart of the heuristic algorithm for solving the problem. Each type of waste has its own disposal center, so a number of disposal centers are known and equal to a number of waste types. We should choose this number of locations \(n_f\) from potential locations for disposal centers. According to the problem, a number of disposal centers is a parameter and it is equal to number of waste types. However, the number of depots is critical and it is important for making a decision about it. This number can be varied from one solution to another one. Increase in a number of depots can decline the distance of routes; however, each depot has specific opening cost, so establishing one depot can lead to increase in the total cost. An upper bound for the number of depots is equal to minimum of the number of customer clusters and potential locations for depots. For clustering the customers, we use \(k\)-means procedure. MATLAB software has capability to cluster the customers according to their coordinates (for detailed information, the reader can refer to syntax of the \(k\)-means function in MATLAB software or refer to [16]). The lower bound for the number of depots \(n_m\) calculated as follows, where \(C\) is the number of clusters (i.e. customer groups) and \(\Omega\) is the capacity of depots.

\[n_m = \left\lfloor \frac{C}{\Omega} \right\rfloor\]

The proposed heuristic algorithm includes two stages. In the first stage, the appropriate number of depots is calculated and in the second stage, this number is used for finding Pareto solutions by means of NSGA-II and
MOPSO algorithms. Genetic algorithm and particle swarm optimization are commonly used in the literature of location routing problem such as [25–28]. This fact motivates us to apply these two algorithms to tackle the problem. In the first step of the stage one, customers are clustered into groups with respect to their coordinates. In this regard, $k$-mean method is used for clustering of customers. Interested readers for $k$-mean approach can refer to [16] which applies this method in solving a LRP. It should be noted that this method only approximates a minimum number of required depots, because the customers’ demand and capacity of vehicles are not considered. We know that the number of depots only impacts on the economical objective and this number does not impact on a social objective (depots are not undesirable facilities).

It is understandable from looking at flowchart that in the first stage of the heuristic method proposed in this paper, we want to specify the appropriate number of depots. For this purpose, the genetic algorithm (GA) is applied. After determination of $n_d$ and $n_p$, the GA tries to find a reasonable and near-optimal solution for this number of $n_d$ and $n_p$. The algorithm iterates for a different number of $n_d$ and finally we find an appropriate number of opened depot. In GA, only economical objective function is considered, because a number of depots does not influence the social objective. This number is an input data for NSGA-II and MOPSO to find Pareto-solutions. The enumeration of all combinations of $n_d$ locations for depots can take into account in a short time with today’s computing tools. In other words, after finding an appropriate number of depots, we solve the problem by means of the proposed algorithms. At the end of proposed algorithms, non-dominated solutions are selected for approximation of a Pareto-optimal set.

5.1. Proposed NSGA-II

A genetic algorithm (GA) is an intelligent population-based evolutionary meta-heuristic algorithm, which
is widely used to solve optimization problems. The GA is successfully used by a significant number of researchers to solve LRP.[29]

Different extensions of the GA for multi-objective optimization problems exist in the literature. The GA is the most popular meta-heuristic algorithm for solving multi-objective problems since it does not need the user to prioritize, scale, or weigh objectives.[30] Some advantage of NSGA-II rather that other algorithms include but not limited to explicit diversity preservation, complexity of the algorithm, and the algorithm does not omit the already found Pareto solution. Fast non-dominated sorting genetic algorithm (NSGA-II) presented by Deb et al. [31] is one of the algorithms used in this study. The steps of the proposed algorithm are as follows:

- Step 1. The initial population is randomly generated.
- Step 2. The chromosomes (i.e. solutions) are evaluated and a rank is assigned to each solution with respect to its non-domination level. Rank one is the best level, rank two is the next best level, and so on.
- Step 3. For the solutions of the same rank, a crowding distance is specified. Crowding distance provides an estimate of the density of solutions surrounding that solution. The amount of this measure for particular solution is the average distance of its two neighboring solutions. By this operator, the selection process goes toward a uniformly spread-out Pareto optimal front at different stages of the solution algorithm. In other words, applying this operator makes Pareto solutions to distribute uniformly. Among two solutions at the same front, the solution located in a lesser crowded region is preferred.
- Step 4. Genetic operators (i.e. crossover and mutation) are applied to create the new population which is called the offspring population from the current chromosomes which are called parents.
- Step 5. The new generated population and the former population are combined to make a new population. The new population is sorted using the non-domination criterion. The best solutions are selected from the sorted population as the new population with respect to elitism and crowding distance.
- Step 6. The chromosomes with the non-domination level 1 are stored in the archive as Pareto solutions.
- Step 7. The stopping criterion that is considered to be the maximum number of iterations is checked. If it is reached, the algorithm will stop; otherwise, Steps 4 to 7 are repeated.

5.1.1. Solution representation

An efficient definition of the chromosome of solutions can help us to find a better solution in a fast way. At first, the order of customers, vehicles, depots, and disposal centers are specified. Then, these chromosomes are transformed to meaningful chromosomes to show solutions of the problem.

In this case, we use a simple method to code the solutions. Each solution has four strings of chromosome to specify its attributes. String one specifies customers order, string two determines vehicles order to serve customers, and string three represents depots order and also string four represents disposal facilities order. Number of \( n_c \) and \( n_v \) primary genes are separated from string three and four, respectively, to specify open depots and disposal facilities.

Operators of GA, NSGA-II and MOPSO are applied on these raw strings to make new solutions. To transform these raw strings to intelligible solutions, we should make routes and assign each route to open depot. Vehicles and customers order are specified in chromosomes one and two. We start with a vehicle in the first gene in the vehicles' chromosome. Customers are assigned to this vehicle one by one in the order of customers' chromosome. We add customers to route as long as the capacity of a vehicle is not full. If the total demands of customers in a route exceed the capacity of a vehicle corresponding to this route by adding one customer to route, we will stop adding customers and will start with the next vehicle in vehicles order chromosome. This manner will continue in order to assign all customers to vehicles. After this, a random permutation of the open disposal facilities is added to the end of the route. Now, we should assign each route to a depot. For this purpose, the first vehicle starts its route from the first depot, the second vehicle from the second depot, and so on. This will continue until we reach the last open depot in a chromosome of the depot's order. Then, we return to the first depot, and the next vehicle will be assigned to the first depot. All the routes are assigned to opened depots by means of this method. It should be noted that we consider only a capacity limitation for making routes. Other limitations can be considered by means of a penalty function adding to both objective functions.

After doing these actions, the first part of chromosome (i.e. first gene) introduces the depot which serves customers in the specific route. The second part of chromosomes specifies the customer’s order in each route, the third part specifies disposal centers order in each route, and the last one is a depot which a vehicle starts its route. Figure 3 shows a chromosome of one route in a solution.

5.1.2. Crossover and mutation operators

There are different types of crossover operators that can be used to generate new chromosomes. In this study, an order crossover (OX) is applied. To perform this type of the crossover, two parents are selected by a roulette wheel algorithm, in which solutions with a
5.2. Proposed MOPSO

Particle swarm optimization (PSO) is a population-based meta-heuristic algorithm and it was initially proposed by Kennedy and Eberhart. This is inspired from the behavior of bird flocks when they try to find foods. The movement of each individual toward the food is affected by the best behavior in its neighborhood and the best global behavior. Some advantages of the algorithm rather than other metaheuristics are the simplicity and less required computational time, and it does not need a priori knowledge of the relative importance of the objective functions.

Also, this algorithm was successfully applied for solving many difficult locations and routing problems. Parsopoulos and Vrahatis presented multi-objective particle swarm optimization (MOPSO) and showed that it is capable to solve difficult and well-known test problems efficiently. In the study carried out by Coello and Lechuga, it was indicated that MOPSO provides the Pareto front closer to the desired Pareto fronts and it performed better than most of the other multi-objective algorithms. The structure of solutions is similar to chromosomes introduced in the previous section; however, we know that PSO is designed for continuous problems. To generate permutation for customers, vehicles, depots, and disposal facilities, we use uniform random numbers between 0 and 1. We generate random numbers and sort them in decreasing order. Figure 5 demonstrates the random numbers and the permutation are obtained from these random numbers.

The main structure of the MOPSO algorithm is illustrated in Figure 6. The steps of the proposed algorithm are as follows:

Step 1. The initial population of particles is generated.
Step 2. The speed of each particle in the population is generated and stored in the velocity vector. This vector determines the direction in which a particle should move in order to improve its current position.
Step 3. Each particle is then evaluated and its current position is stored as the particle’s memory in the personal best (pbest), which shows the best position of each individual.
where \( w \) is the inertia weight that is set at 0.5 in order to control the impact of a particle's velocity history on its current velocity. \( C_1 \) and \( C_2 \) are cognitive and social learning factors that are related to the own success of a particle and its neighborhood success, respectively. \( r_1 \) and \( r_2 \) are random numbers between 0 and 1, and \( \text{position}(i) \) presents the current position of the particle \( i \). Then the new position of each particle is calculated by:

\[
\text{position}(i) = \text{position}(i) + \text{velocity}(i)
\]

**Step 4.** The position of non-dominated particles, called leaders, is stored in an external archive called global best (\( \text{gbest} \)). Leaders are the particles that lead other particles towards better regions in the search space. For every individual, a leader should be considered.

**Step 5.** This step should be repeated until the number of iteration is equal to the maximum number of iterations. First, the velocity of each particle \( i \) is calculated by:

\[
\text{velocity}(i) = w \times \text{velocity}(i) + C_1 \times r_1 \times (pbest(i) - \text{position}(i)) + C_2 \times r_2 \times (\text{leader.position}(i) - \text{position}(i))
\]

where \( w \) is the inertia weight that is set at 0.5 in order to control the impact of a particle’s velocity history on its current velocity. \( C_1 \) and \( C_2 \) are cognitive and social learning factors that are related to the own success of a particle and its neighborhood success, respectively. \( r_1 \) and \( r_2 \) are random numbers between 0 and 1, and \( \text{position}(i) \) presents the current position of the particle \( i \). Then the new position of each particle is calculated by:

\[
\text{position}(i) = \text{position}(i) + \text{velocity}(i)
\]

**Figure 5.** Transformation of random numbers to permutation of customers.

<table>
<thead>
<tr>
<th>Random numbers</th>
<th>0.126</th>
<th>0.987</th>
<th>0.312</th>
<th>0.648</th>
<th>0.188</th>
<th>0.645</th>
<th>0.395</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related customer</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

**Sorted numbers**

<table>
<thead>
<tr>
<th>Sorted numbers</th>
<th>0.987</th>
<th>0.648</th>
<th>0.645</th>
<th>0.395</th>
<th>0.312</th>
<th>0.188</th>
<th>0.126</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permutation</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>3</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

**Figure 6.** Flow chart of the proposed MOPSO.
Population size is considered 200 for both of them.

Experiments are repeated 10 times.

NSGA-II assumptions:
(a) Crossover coefficient is set to 0.54.
(b) Mutation coefficient is set to 0.33.

MOPSO assumptions:
(a) Repository size (number of solutions banned as a Pareto solution) is set to 50.
(b) Value of the velocity coefficient ($\omega$) is set to 0.48.
(c) Damping factor for the velocity coefficient is set to 0.99 (by applying the damping factor in each iteration, the value of velocity coefficient decreases gradually).
(d) Values of $C_1$ and $C_2$ are set to 2.12.

6.2. Small and medium sized problems

6.2.1. Test problem

The initial experiment is carried out upon small-sized problems, which includes 10 sample test problems of various sizes. Attributes of the problems are shown in Tables 1 and 2. The proposed NSGA-II and MOPSO are applied for dealing with these problems and the associated results are compared with each other according to comparison metrics presented in the upcoming section. Each problem runs 10 times and the associated results are recorded.

6.2.2. Comparison metrics

In order to verify the validity of the suggested algorithms and compare these algorithms with each other, four comparison metrics are employed.[36]

(1) Number of Pareto solutions (NPS): The quantity of non-dominated solutions that every algorithm can discover. Larger number for this metric is desirable.

(2) Spacing metric one (SM1): This kind of metrics provides us details about the uniformity of the distribution of the solutions obtained by way of each algorithm. This metric is computed by:

$$SM1 = \sqrt{\frac{1}{N-1} \sum_{i=1}^{n} (d_i - \bar{d})^2}$$

where $d_i$ is the Euclidean distance between solution $i$ and the nearest solution belonged to Pareto sets of solutions. $\bar{d}$ is the average value of all $d_i$. For the multi-objective problem with $k$ objective functions, $d_i$ is calculated as follows:

$$d_i = \min_j \sqrt{(f_i^1 - f_j^1)^2 + \ldots + (f_i^k - f_j^k)^2} \quad j \neq i, j = 1, 2, \ldots, N$$

6.3. Large sized problems

6.3.1. Test problem

The initial experiment is carried out upon large-sized problems, which includes 10 sample test problems of various sizes. Attributes of the problems are shown in Tables 3 and 4. The proposed NSGA-II and MOPSO are applied for dealing with these problems and the associated results are compared with each other according to comparison metrics presented in the upcoming section. Each problem runs 10 times and the associated results are recorded.

6.3.2. Comparison metrics

In order to verify the validity of the suggested algorithms and compare these algorithms with each other, four comparison metrics are employed.[36]

(1) Number of Pareto solutions (NPS): The quantity of non-dominated solutions that every algorithm can discover. Larger number for this metric is desirable.

(2) Spacing metric one (SM1): This kind of metrics provides us details about the uniformity of the distribution of the solutions obtained by way of each algorithm. This metric is computed by:

$$SM1 = \sqrt{\frac{1}{N-1} \sum_{i=1}^{n} (d_i - \bar{d})^2}$$

where $d_i$ is the Euclidean distance between solution $i$ and the nearest solution belonged to Pareto sets of solutions. $\bar{d}$ is the average value of all $d_i$. For the multi-objective problem with $k$ objective functions, $d_i$ is calculated as follows:

$$d_i = \min_j \sqrt{(f_i^1 - f_j^1)^2 + \ldots + (f_i^k - f_j^k)^2} \quad j \neq i, j = 1, 2, \ldots, N$$

Table 1. Problem sets for small- and medium-sized problems.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Customers (n)</th>
<th>Vehicles type (m)</th>
<th>Waste type (w)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>25</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>25</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>25</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>30</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 2. Test problem generation.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customers coordinate</td>
<td>U(0100)</td>
<td>Maximum allowable route length</td>
<td>U(100100)</td>
</tr>
<tr>
<td>Capacity of depots</td>
<td>Regard to problem</td>
<td>Fixed cost</td>
<td>U(0100)</td>
</tr>
<tr>
<td>Capacity of vehicles</td>
<td>U(0200)</td>
<td>Variable cost</td>
<td>U(010)</td>
</tr>
</tbody>
</table>

If the current position calculated for an individual is better than its former position, $p_{best}(i)$ should be updated by:

$$p_{best}(i) = \text{position}(i)$$

It should be noticed that when the new position of a particle is dominated by its position in the memory, the position in the memory should be kept. Finally, at each iteration, non-dominated solutions in the population are added to the repository of non-dominated solutions. After this, the repository is rectified and dominated solutions are eliminated.

6. Experimental results

The performance of the proposed NSGA-II and MOPSO is compared with each other and the associated results are analyzed. The algorithms are coded in MATLAB R2013a and run on Intel Core i5 2.27 GHz personal computer with 4 GB RAM. The steps of both algorithms are illustrated in the previous section completely.

6.1. Algorithm assumptions

The experimental results are implemented in two sections consisting of: (1) small and medium-sized problems and (2) large-sized problems. Sets of parameters are tuned by the response surface methodology (RSM) in the Design Expert software. The assumptions are held for both small- and large-sized problems.

(1) General assumption:
(a) Number of the maximum iteration for both algorithm is considered 100.
(b) Population size is considered 200 for both of them.
(c) Experiments are repeated 10 times.

(2) NSGA-II assumptions:
(a) Crossover coefficient is set to 0.54.
(b) Mutation coefficient is set to 0.33.

(3) MOPSO assumptions:
(a) Repository size (number of solutions banned as a Pareto solution) is set to 50.
(b) Value of the velocity coefficient ($\omega$) is set to 0.48.
(c) Damping factor for the velocity coefficient is set to 0.99 (by applying the damping factor in each iteration, the value of velocity coefficient decreases gradually).
(d) Values of $C_1$ and $C_2$ are set to 2.12.
The NSGA-II and MOPSO algorithms are applied for solving the test problems and their efficiencies are compared with each other. Each test problem operates 10 times and the outcomes are summarized in Table 3. This table lists the average values for all mentioned metrics and Table 4 demonstrates the average running time for each test problem. Generally, we can claim that the NSGA-II can achieve better results in comparison with the MOPSO in far more computational times.

1) NSGA-II can achieve a greater number of Pareto-solutions than the MOPSO.
2) Spacing metrics obtained by both formula show that the NSGA-II provides non-dominated solutions that has a less average value of the spacing metrics. This data reveal that the non-dominated set obtained by the NSGA-II is more uniformly distributed in comparison with the MOPSO algorithm.
3) Diversification metric in the NSGA-II and MOPSO does not show the superiority of none of them; however, the average value for the diversification metric obtained by the MOPSO for test problems is greater than the NSGA-II.
4) Despite the fact that the number of function evaluation for the MOPSO is greater than the NSGA-II (for MOPSO and NSGA-II are 20,200 and 13,201, respectively); however, the computational time for the MOPSO algorithm is considerably less than the NSGA-II algorithm. Table 4 shows that the average computational times for both algorithms.

6.3. Large-sized problems
An additional experiment is constructed for large-sized problems. The properties associated with test problems are revealed in Table 5. Ten different sizes are implemented in this section to compare the performance of the algorithms with each other in large-sized problems. The parameters of problems are generated based on Table 2.

Table 3. Computational results for small- and medium-sized problems.

<table>
<thead>
<tr>
<th>Problem</th>
<th>NPS MOPSO</th>
<th>NSGA-II MOPSO</th>
<th>SM1 MOPSO</th>
<th>NSGA-II MOPSO</th>
<th>DM MOPSO</th>
<th>NSGA-II MOPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.4</td>
<td>7.4</td>
<td>1.849</td>
<td>0.2340</td>
<td>3.6309</td>
<td>1.46766</td>
</tr>
<tr>
<td>2</td>
<td>9.4</td>
<td>10</td>
<td>2.3985</td>
<td>1.0937</td>
<td>5.4085</td>
<td>3.1059</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>11</td>
<td>1.5562</td>
<td>0.1693</td>
<td>8.9923</td>
<td>2.1205</td>
</tr>
<tr>
<td>4</td>
<td>7.5</td>
<td>15</td>
<td>1.9684</td>
<td>0.5441</td>
<td>3.4554</td>
<td>1.6662</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>14.4</td>
<td>2.7771</td>
<td>1.2952</td>
<td>12.785</td>
<td>1.9983</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>9</td>
<td>0.96514</td>
<td>0.3923</td>
<td>16.3846</td>
<td>4.9248</td>
</tr>
<tr>
<td>7</td>
<td>9.5</td>
<td>10</td>
<td>5.8462</td>
<td>1.7662</td>
<td>13.4231</td>
<td>4.2876</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>11</td>
<td>7.8237</td>
<td>1.8879</td>
<td>7.9503</td>
<td>5.1219</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>8</td>
<td>1.8536</td>
<td>1.04989</td>
<td>2.814</td>
<td>2.7147</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>21</td>
<td>1.768</td>
<td>0.911</td>
<td>1.923</td>
<td>1.0211</td>
</tr>
<tr>
<td>Average</td>
<td>7.58</td>
<td>11.68</td>
<td>2.8745</td>
<td>0.934</td>
<td>7.6767</td>
<td>2.9210</td>
</tr>
</tbody>
</table>

Table 4. Average computational times (in seconds).

<table>
<thead>
<tr>
<th>Problem</th>
<th>NSGA-II</th>
<th>MOPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>326</td>
<td>51</td>
</tr>
<tr>
<td>2</td>
<td>840</td>
<td>52.66</td>
</tr>
<tr>
<td>3</td>
<td>957</td>
<td>70.66</td>
</tr>
<tr>
<td>4</td>
<td>765.5</td>
<td>97.33</td>
</tr>
<tr>
<td>5</td>
<td>526.5</td>
<td>48</td>
</tr>
<tr>
<td>6</td>
<td>1121</td>
<td>67.33</td>
</tr>
<tr>
<td>7</td>
<td>1149</td>
<td>113.66</td>
</tr>
<tr>
<td>8</td>
<td>969</td>
<td>238</td>
</tr>
<tr>
<td>9</td>
<td>1056</td>
<td>241</td>
</tr>
<tr>
<td>Average</td>
<td>843.9</td>
<td>103.36</td>
</tr>
</tbody>
</table>

Smaller number for this metric shows that the Pareto solutions are dispersed uniformly.

1) Spacing metric two (SM2): We employ one more spacing metric. This kind of spacing metric is presented by means of Schott.[37] The goal of this metric is to evaluate how the items in the approximation set are uniformly distributed in the objective area. Also, smaller value for this metric is desirable.

$$SM2 = \sqrt{\frac{1}{n-1} \times \sum (d_i - \bar{d})^2}$$

$$d_i = \min \left( \left| f_i^{x}(\overline{x}) - f_i^{y}(\overline{x}) \right| + \left| f_j^{x}(\overline{x}) - f_j^{y}(\overline{x}) \right| \right)_{i, j = 1, 2, \ldots, n}$$

and \( \bar{d} \) is the mean value of all \( d_i \).

1) Diversification metric (DM): This metric specifies the spread of the solution set and determined by:

$$DM = \sqrt{\sum_{i=1}^{n} \max(\|x_i^{x} - y_i^{y}\|)}$$

where \( \max(\|x_i^{x} - y_i^{y}\|) \) is the Euclidean distance between the non-dominated solutions \( x_i^{x} \) and \( y_i^{y} \). Larger number is better, because it shows the diversification of Pareto solutions and these solutions can be representative for different spaces of solutions.

6.2.3. Comparative results
The NSGA-II and MOPSO algorithms are applied for solving the test problems and their efficiencies are compared with each other. Each test problem operates 10 times and the outcomes are summarized in Table 3. This table lists the average values for all mentioned metrics and Table 4 demonstrates the average running time for each test problem. Generally, we can claim that the NSGA-II can achieve better results in comparison with the MOPSO in far more computational times.

1) NSGA-II can achieve a greater number of Pareto-solutions than the MOPSO.
2) Spacing metrics obtained by both formula show that the NSGA-II provides non-dominated solutions that has a less average value of the spacing metrics. This data reveal that the non-dominated set obtained by the NSGA-II is more uniformly distributed in comparison with the MOPSO algorithm.
3) Diversification metric in the NSGA-II and MOPSO does not show the superiority of none of them; however, the average value for the diversification metric obtained by the MOPSO for test problems is greater than the NSGA-II.
4) Despite the fact that the number of function evaluation for the MOPSO is greater than the NSGA-II (for MOPSO and NSGA-II are 20,200 and 13,201, respectively); however, the computational time for the MOPSO algorithm is considerably less than the NSGA-II algorithm. Table 4 shows that the average computational times for both algorithms.

6.3. Large-sized problems
An additional experiment is constructed for large-sized problems. The properties associated with test problems are revealed in Table 5. Ten different sizes are implemented in this section to compare the performance of the algorithms with each other in large-sized problems. The parameters of problems are generated based on Table 2.
7. Conclusion and future research

In this paper, we considered a waste collection problem, in which the location of depots and disposal facilities should be chosen from potential locations. Vehicles started their routes from depots and went on routes to serve all customers. After servicing of customers, vehicles moved to the disposal centers. Each type of the waste had the compatible disposal centers. Vehicles were heterogeneous with multiple compartments for each type of the waste. Distance, time and capacity limitations were considered in this paper. The problem was a multi-objective one, in which economic and social objectives were investigated in the presented bi-objective model. This paper presented two meta-heuristic algorithms (i.e. NSGA-II and MOPSO) for solving the bi-objective model and introduced a simple way for representation of solutions. To validate the proposed algorithms, various test problems were designed, and then two algorithms were compared with each other by means of comparison metrics (e.g. number of Pareto solutions, two pacing metrics and diversification metric). The experimental results showed that the proposed NSGA-II outperformed the MOPSO in test problems; however, the computational times for the MOPSO were smaller than the NSGA-II in all test problems. This was rational according to this fact that the NSGA-II could search more regions of solutions in this discrete type of the problem. For researchers interested in this field of logistics, we can suggest to combine the available constraints with other constraints taken from the real-world environment as the case study. Since in

<table>
<thead>
<tr>
<th>Problem</th>
<th>Customers (n)</th>
<th>Vehicles type (m)</th>
<th>Waste type (w)</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>40</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>12</td>
<td>50</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>75</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>75</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>75</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>16</td>
<td>100</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>17</td>
<td>100</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>18</td>
<td>100</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>150</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>20</td>
<td>200</td>
<td>10</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 5. Problem sets for large-sized problems.

<table>
<thead>
<tr>
<th>Problem</th>
<th>NPS</th>
<th>SM1</th>
<th>SM2</th>
<th>DM</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>10.5</td>
<td>4.2731</td>
<td>0.359</td>
<td>2.885</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>1.485</td>
<td>0.915</td>
<td>2.5308</td>
</tr>
<tr>
<td>13</td>
<td>16</td>
<td>2.121</td>
<td>1.8876</td>
<td>3.563</td>
</tr>
<tr>
<td>14</td>
<td>16</td>
<td>2.395</td>
<td>1.654</td>
<td>7.4325</td>
</tr>
<tr>
<td>15</td>
<td>18</td>
<td>2.231</td>
<td>0.9832</td>
<td>3.5632</td>
</tr>
<tr>
<td>16</td>
<td>12.6</td>
<td>1.092</td>
<td>1.231</td>
<td>2.231</td>
</tr>
<tr>
<td>17</td>
<td>19</td>
<td>3.301</td>
<td>2.573</td>
<td>4.481</td>
</tr>
<tr>
<td>18</td>
<td>22</td>
<td>1.119</td>
<td>1.008</td>
<td>2.321</td>
</tr>
<tr>
<td>19</td>
<td>23.4</td>
<td>2.312</td>
<td>0.9831</td>
<td>5.4302</td>
</tr>
<tr>
<td>20</td>
<td>21.8</td>
<td>1.185</td>
<td>1.1231</td>
<td>2.1953</td>
</tr>
</tbody>
</table>

Table 6. Computational results for large-sized problems.

<table>
<thead>
<tr>
<th>Problem</th>
<th>NSGA-II</th>
<th>MOPSO</th>
<th>NSGA-II</th>
<th>MOPSO</th>
<th>NSGA-II</th>
<th>MOPSO</th>
<th>NSGA-II</th>
<th>MOPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>7</td>
<td>10.5</td>
<td>4.2731</td>
<td>0.359</td>
<td>2.5308</td>
<td>1.9893</td>
<td>2.885</td>
<td>3.1366</td>
</tr>
<tr>
<td>12</td>
<td>9</td>
<td>12</td>
<td>1.485</td>
<td>0.915</td>
<td>3.563</td>
<td>3.323</td>
<td>2.842</td>
<td>4.4987</td>
</tr>
<tr>
<td>13</td>
<td>16</td>
<td>2.121</td>
<td>1.8876</td>
<td>7.4325</td>
<td>5.3745</td>
<td>3.5369</td>
<td>3.8330</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>16</td>
<td>2.395</td>
<td>1.654</td>
<td>3.5632</td>
<td>2.228</td>
<td>4.4963</td>
<td>5.3221</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>18</td>
<td>2.231</td>
<td>0.9832</td>
<td>2.231</td>
<td>2.012</td>
<td>3.873</td>
<td>3.881</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>19</td>
<td>3.301</td>
<td>2.573</td>
<td>4.481</td>
<td>4.006</td>
<td>4.421</td>
<td>4.582</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>22</td>
<td>1.119</td>
<td>1.008</td>
<td>2.321</td>
<td>1.842</td>
<td>1.783</td>
<td>2.226</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>23.4</td>
<td>2.312</td>
<td>0.9831</td>
<td>5.4302</td>
<td>1.489</td>
<td>3.321</td>
<td>2.981</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>21.8</td>
<td>1.185</td>
<td>1.1231</td>
<td>2.1953</td>
<td>1.9873</td>
<td>2.3947</td>
<td>2.7772</td>
<td></td>
</tr>
</tbody>
</table>

Table 7. Average computational times (in seconds).

6.3.1. Comparisons metrics

The comparison metrics are similar to small-sized problems and we employ a number of Pareto solutions (NPS), two spacing metrics (SM1, SM2), and diversification metric (DM) for comparison of the algorithms. Also, we can find out that the computational time depends on the customers and vehicles type number especially vehicles type numbers.

6.3.2. Comparative results

The NSGA-II and MOPSO are compared in large-sized problems by means of test problems. The NSGA-II almost in all factors of test problem has a better performance than the MOPSO. We can find out from Tables 6 and 7 that the NSGA-II outperforms the MOPSO in location-routing problems, especially in problem considered in this paper.

(1) The NSGA-II can find a greater number of Pareto solutions than the MOPSO.
(2) Both spacing metrics show that the Pareto-optimal set from the NSGA-II is uniformly distributed in the solution area.
(3) Diversification metric shows that the NSGA-II can explore solutions’ space better than the MOPSO in large-sized problems and the solutions in Pareto front are diversely distributed.
(4) Running time for the MOPSO is very smaller than the NSGA-II and this is the advantage of the MOPSO algorithm. However, since the NSGA-II can search more regions of solutions, this higher computational time is rational.

7. Conclusion and future research

In this paper, we considered a waste collection problem, in which the location of depots and disposal facilities should be chosen from potential locations. Vehicles started their routes from depots and went on routes to serve all customers. After servicing of customers, vehicles moved to the disposal centers. Each type of the waste had the compatible disposal centers. Vehicles were heterogeneous with multiple compartments for each type of the waste. Distance, time and capacity limitations were considered in this paper. The problem was a multi-objective one, in which economic and social objectives were investigated in the presented bi-objective model. This paper presented two meta-heuristic algorithms (i.e. NSGA-II and MOPSO) for solving the bi-objective model and introduced a simple way for representation of solutions. To validate the proposed algorithms, various test problems were designed, and then two algorithms were compared with each other by means of comparison metrics (e.g. number of Pareto solutions, two pacing metrics and diversification metric). The experimental results showed that the proposed NSGA-II outperformed the MOPSO in test problems; however, the computational times for the MOPSO were smaller than the NSGA-II in all test problems. This was rational according to this fact that the NSGA-II could search more regions of solutions in this discrete type of the problem. For researchers interested in this field of logistics, we can suggest to combine the available constraints with other constraints taken from the real-world environment as the case study. Since in
waste collection problems, time is a critical factor and the time windows constraint can be added to the given problem or a disposal facility can be seen more flexible to dispose several types of waste. Also, researchers can use other meta-heuristics and compare the obtained results with each other.

Disclosure statement

No potential conflict of interest was reported by the authors.

Notes on contributors

Hamed Farrokhi-Asl is currently a PhD candidate in industrial engineering at Iran University of Science & Technology. He finished his bachelor of Science in Industrial Engineering at University of Khaje-Nasir University of Technology, Iran in 2013. He continued his master’s studies at University of Tehran and graduated in 2014. He has published more than 20 conference and journal papers. Hamed joined Professor Rabbani’s research team as a research assistant. He initially started working on a new nature inspired algorithm at Computer Laboratory of Iran University of Science & Technology. His interests include waste management, green supply chain, and optimization algorithms.

Reza Tavakkoli-Moghaddam is a professor at Industrial Engineering at College of Engineering, University of Tehran in Iran. He obtained his BSc, MSc, and PhD from Iran University of Science and Technology in Tehran (1989), University of Melbourne in Melbourne (1994) and Swinburne University of Technology in Melbourne (1998), respectively. He serves as the editor-in-chief of two journals and the Editorial Board of five journals. He was the recipient of the 2009 and 2011 Distinguished Researcher Awards as well as the 2010 and 2014 Distinguished Applied Research Awards by University of Tehran in Iran. He was also selected as National Iranian Distinguished Researcher in 2008 and 2010 by the Ministry of Science, Research and Technology (MSRT) in Iran. He obtained the outstanding rank as the top 1% scientist and researcher in the world elite group, reported by Thomson Reuters in 2014. Professor Tavakkoli-Moghaddam has published 4 books, 17 book chapters, and more than 700 papers in reputable academic journals and conferences.

Bahare Asgarian is a master’s student in industrial engineering at Amirkabir University of Technology. She finished her bachelor of science in mathematical science at Alzahra University. Bahare attended a conference on Industrial Engineering held in Tehran, Iran. Her present research interests include inventory management and distribution of perishable goods.

Esmat Sangari recently got an acceptance for Department of Industrial Engineering and Management Sciences, Northwestern University as a PhD student. She received her MS and bachelor degrees in Department of Industrial and Systems Engineering from the University of Tehran. Her research interests include production planning and scheduling. Supply chain management, and mathematical programming.

ORCID

Reza Tavakkoli-Moghaddam http://orcid.org/0000-0002-6757-926X

References


