INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2015; 5(1):1-20

CAPACITATED VEHICLE ROUTING PROBLEM WITH VEHICLES HIRE OR PURCHASE DECISION: MODELING AND SOLUTION APPROACHES

Z. Hajishafee, S.H. Mirmohammadi[*,†] and S.R. Hejazi
Department of Industrial and Systems Engineering, Iran Isfahan University of Technology

ABSTRACT

The overall cost of companies dealing with the distribution tasks is considerably affected by the way that distributing vehicles are procured. In this paper, a more practical version of capacitated vehicle routing problem (CVRP) in which the decision of purchase or hire of vehicles is simultaneously considered is investigated. In CVRP model capacitated vehicles start from a single depot simultaneously and deliver the demanded items of several costumers with known demands where each costumer must be met once. Since the optimal vehicle procurement cost is a function of total distance it traverses during the planning horizon, the model is modified in a way that the decision of purchasing or hiring of each vehicle is made simultaneously. The problem is formulated as a mixed integer programming (MIP) model in which the sum of net present value (NPV) of procurement and traveling costs is minimized. To solve the problem, a hybrid electromagnetism and parallel simulated annealing (PSA-EM) algorithm and a Shuffled Frog Leaping Algorithm (SFLA) are presented. Finally, the presented methods are compared experimentally. Although in some cases the SFLA algorithm yields better solutions, experimental results show the competitiveness of PSA-EM algorithm from the computational time and performance points of view.

Received: 15 March 2014; Accepted: 10 November 2014

KEY WORDS: capacitated vehicle routing problem; hire; purchase; parallel simulated annealing; electromagnetism-like algorithm; shuffled frog leaping.

---

[*]Corresponding author: Department of Industrial and Systems Engineering, Iran Isfahan University of Technology
[†]E-mail address: h_mirmohammadi@cc.iut.ac.ir (S.H. Mirmohammadi)

1. INTRODUCTION

Vehicle Routing Problem (VRP) is one of the most widely used problems in all industries. This problem was first raised by Dantzing and Ramser [1]. The problem belongs to NP-hard problems. The complexity of the problem pertains to two problems namely traveling sale man problem (TSP) and bin packing problem ([2]). In this paper, the basic model of capacitated vehicle routing problem (CVRP) is modified in a way that vehicles hire or purchase decision is entered into the model such that total transportation costs including fleet procurement cost and traveling cost are simultaneously considered. In addition to modeling the problem, we determine to purchase vehicle for which routs and hire for which ones so that they totally burden lower costs. Since the cost of purchasing incurs at the present time (decision time) and the cost of hiring arise within various periods in planning horizon, therefore, it is necessary to enter the time value of money in the problem and consider the net present value (NPV) of costs as the objective function which is to be minimized.

Many researchers have studied CVRP problem and presented different solution methods for that and also, they could manage to create new models by adding different assumptions to the issue. Augerat et al [3] applied an exact branch-and-cut method for CVRP and used tabu search (TS) algorithm in cutting plane program. Contardoa and Martinelli [4] presented an exact algorithm for the more general case of CVRP in which more than one depot is available and the tour length is limited named multi-depot vehicle routing problem (MDVRP). They formulated MDVRP using a vehicle-flow and a set-partitioning formulation. They validate their approach by conducting extensive computational experiments on several instances on CVRP as a particular case of the MDVRP.

A wide range of researchers used heuristic or meta heuristic algorithms to achieve a near-optimal solution for the CVRP. Using a multi-phase model of improved shuffled frog leaping algorithm (SFLA) Luo and Chen [5] presented a meta heuristic algorithm to solve the multi-depots vehicle routing problems (MDVRPs).They used a power law extremal optimization neighborhood search (PLEONS) to further improve the local search ability of SFLA and speed up convergence. Chen et al. [6] presented a hybrid heuristic method named iterated variable neighborhood search (IVND) with variable neighborhood descent based on multi-operator optimization for solving the CVRP. They have designed a perturbation strategy by cross-exchange operator as an approach for escaping local minima. Lin et al [7] presented a hybrid algorithm of simulated annealing and tabu search to solve CVRP. Via simulation results on classical instances, they have shown that their algorithm is competitive with other existing algorithms for solving CVRP. Wang and Lu [8] presented a hybrid genetic algorithm consisting of three stages for this problem. In first stage, they used the nearest addition method (NAM) into sweep algorithm (SA) to generate an initial chromosome population. Secondly, the applied response surface methodology (RSM) to optimize crossover probability and mutation probability. Finally, the authors incorporated an improved sweep algorithm into to enhance the exploration diversity of their GA. Ai and Kachitvichyanukul [9] applied particle swarm optimization algorithm (PSO) structure to solve the CVRP. They presented two solution representations for CVRP in PSO. They analyzed the effect of two solution representations and their decoding methods on the algorithm efficiency and via the computational results they concluded that one of these representations is more competitive than other one and other methods for solving CVRP. Yurtkuran and Emell [10] presented a hybrid Electromagnetism-like algorithm (EM) and an iterated swap procedure (ISP) as a local search for solving this CVRP. According to the experiments, their hybrid algorithm act better in comparison with Electromagnetism algorithm but due to the mentioned local search procedure, the computational time of their algorithm became longer. This hybrid EM and ISP algorithm was compared with five methods named tabu search algorithm (TS), simulated annealing algorithm (SA), genetic algorithm (GA), particle warm optimization algorithm and ant colony optimization (ACO) and according to the experimental results it obtained better solutions or the obtained solution had no significant difference with the best solution. In 2010, Garaix et al [11] provided a new model of CVRP. They stated that in solving the VRP problems, only one attribute is usually considered e.g. minimizing the travel time or minimizing the route length, while several attributes can be defined for one arc connecting the origin to destination in the graph model of VRP. They proposed considering several alternative routes via considering several attributes for arcs by a multi-graph representation of the road network and they analyzed their impact on solution algorithms and solution values. They used an accurate dynamic programming method and a heuristic algorithm to solve this problem. Ngueveu et al [12] also presented a new model of CVRP named cumulative capacitated vehicle routing problem (CCVRP). This model seeks to minimize the total times of reaching to customers by considering capacitated vehicle. In the objective function, the total times of reaching to customers has been considered instead of route length or travel cost. This occurs in situations such as rescuing individuals after natural disasters or in supply chain of vital goods. For this problem (CCVRP) recently, Lysgaard and Wøhlk [13] have presented an optimal solution approach based on branch-and-cut-and-price approach.

The remainder of the paper is as follows. In Section 2, the mathematical model of the problem is introduced. Section 3, contains the two presented algorithms; the parallel simulated annealing-electromagnetism algorithm and the shuffled frog leaping algorithm and their elements in detail. Computational experiments and the results of comparison of the presented algorithms are presented in Section 4. Concluding remarks are appeared in Section 5.

2. PROBLEM FORMULATION

2.1 Fundamentals of the developed model

Various evaluation criteria have been considered for Vehicle Routing Problem variants so far. In most problems, it is assumed that the required vehicle must be purchased, while some companies are capable of outsourcing this issue or hire the required vehicles. Since the purchase cost of vehicles considerably differs with their hire cost, the important decision is that according to the planning horizon, how many vehicles are purchased and how many ones are hired so that the total present value of costs is minimized. In fact, in the present paper, the CVRP has been modified to a periodic CVRP whose objective function is the total net present value of costs. In this case, customers demand is assumed as constant during various periods. Two different approaches can be considered for hiring and purchasing mode, including:

The CVRP is solved without considering hire and purchase decision and then, according to the costs, the hiring and purchasing decision is made for each vehicle.

The CVRP and hiring and purchasing decision are integrated and solved simultaneously.

Since the routing problem and the procurement problem are not independent, the first approach has no warranty to obtain the optimal solution of the problem.

2.2 The mathematical model

The mathematical model of the problem includes the following assumptions:

Maximum K vehicle can be purchased or hired.

The source and destination (depot) of all vehicles are the same.

The capacity of all vehicles is the same and equals to C.

The demand of each customer is smaller than the capacity of vehicles.

The demand of each customer is constant and equal in different periods.

Distribution Operation is done in N year and M times per year.

The associated costs of purchased vehicles include purchasing cost, fuel cost and maintenance cost and for the hired ones it is the per unit of time hiring cost.

For the purchased vehicles, salvage value has also been considered at the end of planning horizon.

The notations used in the model are given in Table 1.

Table 1: Notation of the mathematical model

Parameter	**Definition**
K	The number of maximum available vehicles
C	The capacity of each vehicle
r	The nominal annual interest rate
r_{num}	The number of annual compounding periods
V	Set of customers aggregated with depot
N	The length of planning horizon in year
M	The number of required distribution task per year
CP	The vehicles purchasing cost
SV	The vehicles salvage value
FC	The rate of fuel cost
HC	The rate of hiring cost
NC	The rate of maintenance cost of vehicles
t_{ij}	The traveling time from costumer i to costumer j

The decision variables of the presented model are defined in Table 2.

Table 2: Definition of the decision variables

Notation	**Definition**
x_{ijk}	Equals 1 if vehicle k traverses edge i-j; otherwise it is 0
y_{ik}	Equals 1 if vehicle k meets customer i; otherwise it is 0
u_{ik}	The amount of load discharged by vehicle k after meeting customer i
p_k	Equals 1 if vehicle k is purchased; otherwise it is 0
h_k	Equals 1 if vehicle k is hired; otherwise it is 0

The model of the problem is as follows.

$$Min\ Z = \sum_{k=1}^{K} CP \times p_k + \left({}^{P}/_{A}, \frac{r}{r_{num}}, r_{num}N\right)\left(\frac{M}{r_{num}}\right)\left[\sum_i \sum_j t_{ij}\left(\sum_k HCx_{ijk}\,h_k + FCx_{ijk}p_k + NC\ x_{ijk}p_k\right)\right] - \left({}^{P}/_{F}, \frac{r}{r_{num}}, r_{num}N\right)\sum_k SV \times p_k \quad (1)$$

Subject to

$$\sum_{k=1}^{K} y_{ik} = 1 \qquad \forall\, i \in V\backslash\{0\}. \quad (2)$$

$$\sum_{j\in V} x_{ijk} = \sum_{j\in V} x_{jik} = y_{ik}, \ \ \forall\, i \in V\,, k = 1, \dots, K. \quad (3)$$

$$u_{ik} - u_{jk} + Cx_{ijk} \le C - d_j, \qquad \forall\, i,j \in V\backslash\{0\}, i \ne j, k = 1, \dots, K. \quad (4)$$

$$d_j \le u_{jk} \le C, \forall\, j \in V\backslash\{0\}, k = 1, \dots, K. \quad (5)$$

$$p_k + h_k \le 1 \quad , k = 1, \dots, K. \quad (6)$$

$$\sum_{i\in V} \sum_{j\ne i\in V} x_{ijk} \quad \le |V|^2(h_k + p_k),, k = 1, \dots, K. \quad (7)$$

$$x_{ijk} \in \{0,1\}.\ \forall\, i,j \in V\,, k = 1, \dots, K. \quad (8)$$

$$y_{ik} \in \{0,1\}. \forall\, i \in V\,, k = 1, \dots, K. \quad (9)$$

$$p_k \in \{0,1\}. k = 1, \dots, K. \quad (10)$$

$$h_k \in \{0,1\}\,, k = 1, \dots, K. \quad (11)$$

$$u_{ik} \ge 0, \forall\, i \in V\,, k = 1, \dots, K. \quad (12)$$

In equation (1), the present value of costs, as the objective function, is minimized. The first term of (1) is the total cost of purchased vehicles; the second term calculates the present value of the equal annual costs considering the number of compounding periods in a year and the third terms is the present salvage value of the purchased vehicles which is deduced of the total present cost. Equation (2) ensures each customer is met by a vehicle. Equation (3) provides the flow balance of the network (i.e. the number of entering vehicles must be equal to the number of departing ones for each node). Eliminating the sub tours and satisfying the capacity constraint of vehicles are provided by Equations (4) and (5) simultaneously. Equation (6) shows that each vehicle can be either hired or purchased. Equation (7) reveals that vehicle k meets customer i provided that it has been hired or purchased. Equations (8) to (11) are the 0-1 constraints for related decision variables. Equation (12) is the sign constraint of decision variables u.

3. SOLUTION ALGORITHMS

The above model is an MINLP[1] model in which in addition to route detection, the decision

[1] Mixed integer nonlinear programming

of hire/purchase vehicles is made. Therefore, the basic CVRP model is a special case of the above model in which the hire/purchase decision is not consider. Since CVRP belongs to the NP-hard class of problem, it is obvious that the above problem is also an NP-Hard problem and hence there is no an optimal algorithm with polynomial time complexity function for it (unless P=NP).

Thus, it is reasonable to find a near optimal solution in large scale instances of the problem via meta heuristic algorithms. Here, we present two meta heuristic algorithms for this problem: first, a hybrid algorithm based on parallel simulated annealing and electro magnetism algorithms (PSA-EM) and second, shuffled frog leaping algorithm (SFLA).

3.1 PSA-EM Algorithm

Parallel simulated annealing algorithm is the extended version of simulated annealing algorithm is a probabilistic search method that imitates physical melt of solids to find the problem solution of combinatorial optimization problems. Since this algorithm is not population-based and starts to search from one point of the solution space, a few simulated annealing processes have been used to search the solution space in parallel.

EM-type algorithms were first proposed by Birbil and Fang [14]. This algorithm has been used either as a stand-alone approach or as an accompanying procedure for other methods. Chang et. al [15] presented a meta-heuristic that applies the electro magnetism methodology to the single machine scheduling problem also Debels et al [16] used hybrid Electromagnetism algorithm to solve a resource constrained project scheduling problem.

In fact, Electromagnetism algorithm is a population based meta heuristic method. This approach starts with random selection of the points from the feasible space. Each point (particle) is a solution and has some charge. The value of the charge depends on the quality of the objective function in a way that the better the amount of the objective function is, the more the charge will be. Equation (13) shows the relation between the charge amount and the fitness of particle *i* in which q^i is the charge of particle i, $f(x^i)$, $f(x^{best})$ and $f(x^k)$ are the objective values of particle *i*, the best particle and particle *k*, respectively and *m* is the population size. *n* indicates the dimension of solution space

$$q^i = exp\,(-n\frac{f(x^i) - f(x^{best})}{\sum_{k=1}^{m}(f(x^k) - f(x^{best}))}) \tag{13}$$

Based on the value of this charge, an attraction-repulsion mechanism is performed to move the points to the optimum solution in a way that a better particle attracts other particles to itself and a bad particle repulses others from itself.

Electrostatic force between two points is directly related to the magnitude of the charge of two particles, while it is reversely related to the square of the distance between two points.

The resultant force acting on particle *i* from other particles is obtained by (14).

$$F^i = \sum_{j \neq i} |x^i - x^j| \frac{q^i q^j}{\|x^i - x^j\|^2} \tag{14}$$

3.1.1 The framework of PSA-EM algorithm

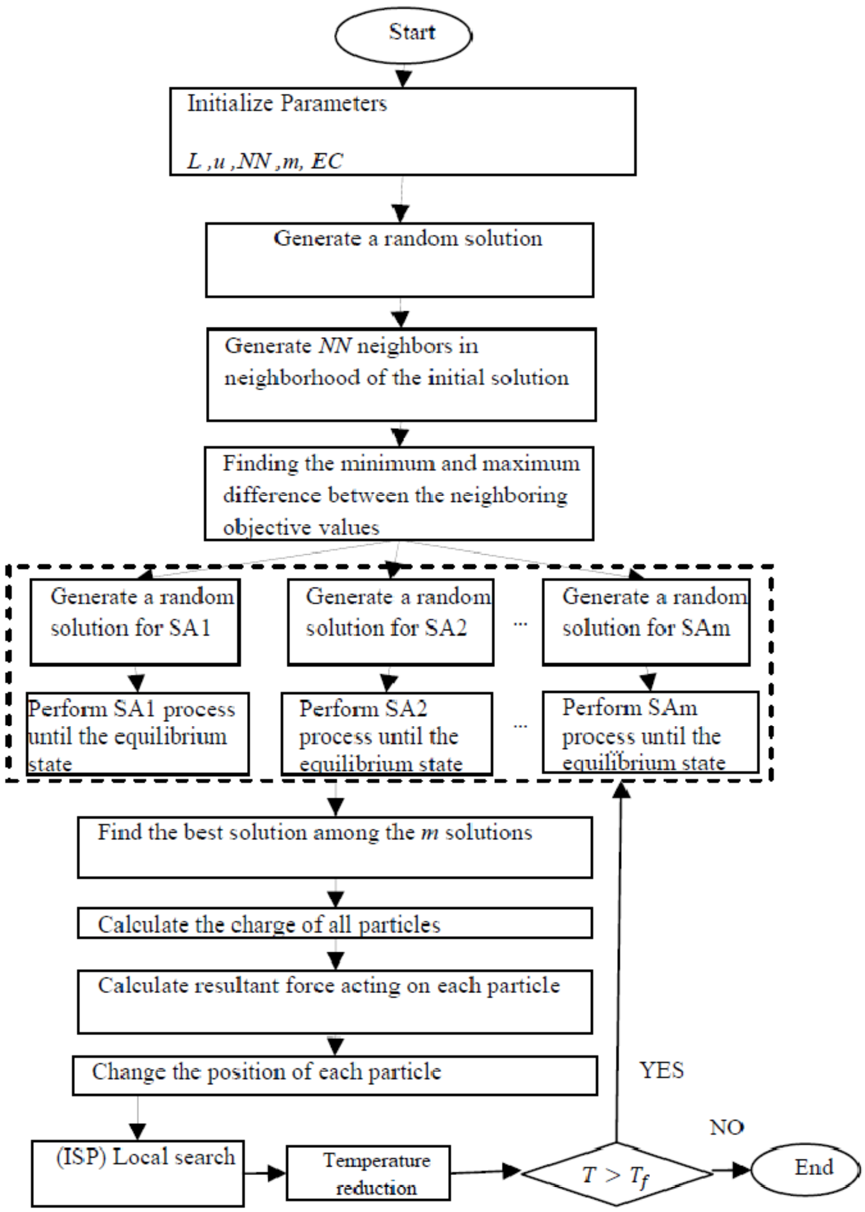

Figure 1. The flowchart of PSA-EM

Fig. 1 depicts the flowchart of the hybrid PSA-EM algorithm. In this algorithm, *m* parallel SA processes with the same initial temperature start to search the solution space from random points. Then, in each process, the search is performed based on the simulated annealing algorithm until the equilibrium condition is met. This process which is shown in Fig. 2 performs as a sub algorithm in *m* parallel section to search the feasible region. Then, the *m* solutions are mapped to *m* separate particles which are the input of the EM algorithm. These particles are moved based on EM mechanism and after that a local search around each of them is performed. Then, the process of temperature reduction is carried out, if the final temperature of the SA processes is met the algorithm stops, otherwise the *m* SA processes are run with the new *m* solutions. This procedure is shown in Fig. 1. The dashed line in this figure shows the *m* parallel SA's. The rest of this section has been devoted to define the basic features of the presented algorithm

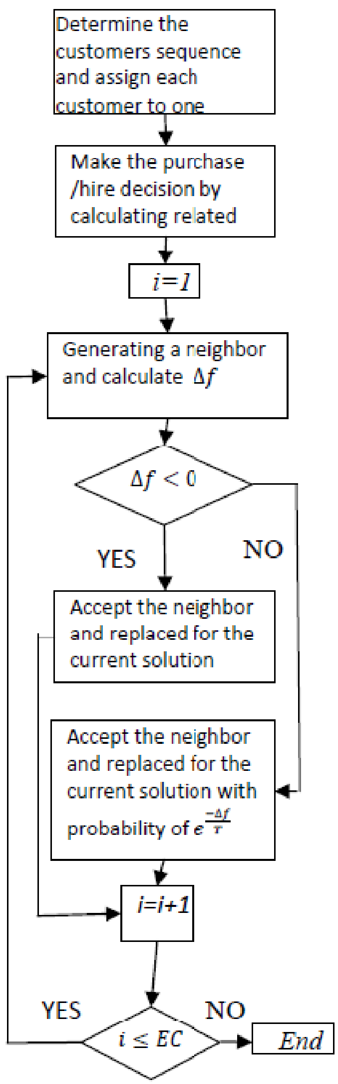

Figure 2. The flow chart of SA_i process

3.1.2 Solution representation

The code representing the problem solution in this algorithm is composed of three parts. The first part contains the visiting order of customers. The second part reveals the purchase/hire decision of vehicles and the third part determines the number of customers that each vehicle meets. Hence, Part I is an array with $|V|-1$ cells where each cell contains a random integer between 1 to $|V|-1$. Part II is an array with K cells where cell k, $k=1, 2, \ldots, K$, contains 1, 2 or 0 if vehicle k is purchased, hired or not purchased nor hired, respectively. Part III is an array with K cells where cell k, $k=1, 2, \ldots, K$, contains an integer representing the number of customers visited by vehicle k.

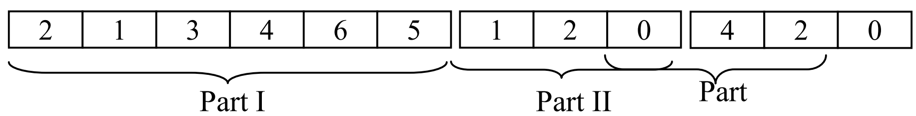

Figure 3. An example of solution coding in PSA-EM

Fig. 3 shows an example of coding in which there are 6 customers and 3 vehicles. It comes from this code that vehicle 1 is purchased and it meets customers 2, 1, 3 and 4, respectively; vehicle 2 is hired and it meets customers 6 and 5, respectively and vehicle 3 is not purchased nor hired. Since the coordinates of the particles and the basic equations in EM algorithm are defined as real number, we used the R-K[2] for discretization of real numbers in arrays.

3.1.3 Initial solutions

All the initial solutions are randomly generated based on uniform distribution through the feasible integer intervals. The first, second and third part of the code for the initial population are randomly generated through sets $\{1,2,\ldots,|V|-1\}$, $\{0,1,2\}$ and $\{1,2,\ldots,|V|-1\}$, respectively. After generating each solution randomly, it may be not feasible. A procedure is applied to make sure that each solution is feasible. This procedure checks the logical constraints such that the number of customers visited by every vehicle that is not purchased nor hired must be zero or the sum of elements in Part III of the solution must be $|V|-1$. The other feasibility condition that is check by this procedure is the functional constraints like capacity constraint, i.e. the total demand of customers met by each vehicle should not exceeds the vehicle capacity.

3.1.4 Initial and final temperature of the SA processes

Initial and final temperatures have an important effect on the quality of the SA processes. The more they match with the problem specification (i.e. the dimensions of the problem, the size of the feasible region and etc.) the more SA processes search the region efficiently. Hence we set them according to the mechanism adopted by Connolly [17]. According to this procedure, a solution is first generated randomly. Then, a number of its neighborhoods are generated and the objective function of them is calculated. (i.e $f_i, \forall i$). The initial and final

---

[2] Random-Key

temperatures are determined by Equations (15) to (18).

$$T_0 = \Delta_{min} + 0.1(\Delta_{max} - \Delta_{min}) \quad (15)$$

$$T_f = 0.08T_0 \quad (16)$$

$$\Delta_{min} = \min_{i,j}\{|f_i - f_j|\} \quad (17)$$

$$\Delta_{max} = \max_{i,j}\{|f_i - f_j|\} \quad (18)$$

3.1.5 Cooling schedule program

In order to guide the search process and converge it to a final solution it is necessary to reduce the temperature after equilibrium state in each temperature. Accepting a bad solution becomes harder while the temperature is reduced and the diversification of the search process is then reduced. There are several cooling schedule program for SA in the literature, but the best known of them is the one presented by Lundy and Mees [18] which has been adapted in our algorithm. The related equations of this cooling schedule program are Equations (19) to (21).

$$T_c = \frac{T_{c-1}}{1 + \beta T_{c-1}} \quad (19)$$

$$\beta = \frac{T_0 - T_f}{M * T_0 * T_f} \quad (20)$$

$$M = \frac{|V|(|V| - 1)}{2} \quad (21)$$

3.1.6 Equilibrium conditions

A certain number of iteration in each temperature (EC) has been considered as the condition for equilibrium. Like cooling schedule program, there are several equilibrium tests for SA. We test some of best known ones, but considering the algorithm efficiency as decision criteria, this test outperformed the others in our algorithm.

3.1.7 Neighborhood Generation

To generate a neighbor of current solution we adopt the swap operator. But it is performed in three different dimensions named two, three and four dimensional swap. In other words, the neighborhood generation is performed in one of the following three ways with equal probability:

A- Two elements of solution array are randomly selected and their values are swapped

B- Three elements of solution array are randomly selected and their values are swapped randomly.

C- Four elements of solution array are randomly selected and their values are swapped randomly.

3.1.8. Local search: iterated swap procedure (ISP)

One of the local search methods used in this type of solution coding is ISP method that is relatively more rapid compared to other methods like 2 or 3-Opt (Ho et al [19]). This method was first used to improve solutions in a framework of GA in a scheduling problem by Ho and Ji [20]. The procedure of this method consists of five steps described as follows.

Step 1: Two elements of the parent solution (the solution its neighborhoods is to be searched) are randomly selected.

Step 2: The locus (location in the array) of these two elements are swapped with each other.

Step 3: The locus of these two elements in the recent solution are swapped with their neighbors, hence four different new solutions are generated.

Step 4: Calculate the objective function of the five generated solutions.

Step 5: If the best solution among the generated solutions was better than the parent solution, that solution will be replaced for the parent solution and return to step 1; otherwise, stop.

The steps of this procedure are illustrated by an example in Table 3.

Table 3: An example of ISP method

Parent	0.15	5.2	11.1	3.2	0.2	1.9	9.2	8.1	6.5	4.8
child										
C1:	0.15	5.2	109	3.2	0.2	11.1	9.2	8.1	6.5	4.8
C2:	0.15	5.2	1.9	3.2	11.1	0.2	9.2	8.1	6.5	4.8
C3:	0.15	5.2	1.9	3.2	0.2	9.2	11.1	8.1	6.5	4.8
C4:	0.15	1.9	5.2	3.2	0.2	11.1	9.2	8.1	6.5	4.8
C5:	0.15	5.2	5.2	1.9	0.2	11.1	9.2	8.1	6.5	4.8

3.2 Shuffled frog leaping algorithm

Shuffled frog leaping algorithm (SFLA) was first introduced by Eusuff et al [21] as a memetic meta-heuristic which has been developed for solving combinatorial optimization problems. The SFLA is a population-based cooperative search metaphor inspired by natural memetics and has been applied to various fields of combinatorial optimization such as incapacitated single level lot-sizing by Liping et al [22] and vehicle routing problem by Lou [23]. This algorithm applies three algorithms of memetic algorithm (MA), particle swarm optimization (PSO) and shuffled complex evolution (SCE). Each frog contains memo and in any collection of frogs, say memplex, a local search is performed based on PSO algorithm. Frogs move to explore and search in various directions of the search region. After a few iterations, they are reorganized in new groups by a technique similar to what in SCE (Eusuff et al [21]).

The presented algorithm initiates by selecting a random population of frogs which covered the whole swamp as possible, In other words, an initial population of solutions is elected. The population is divided into a number of subpopulation, say memplex, with equal numbers of members that independently search the feasible region in various directions. Based on the performance of each frog, a sub-memplex is selected from each memplex. In other words, the frog with better performance is more probable to be selected and vice versa.

The best and worst frog are determined in each sub-memplex and during the evolutionary process, the worst frog of sub-memplex tries to improve itself. After a certain number of iterations, sub-memplexes are combined with each other and new populations are built. This process continues until the stop criterion is met. The diagram of this algorithm has been given in Fig. 4. The remainder of this section is devoted to describe the detail of the presented algorithm.

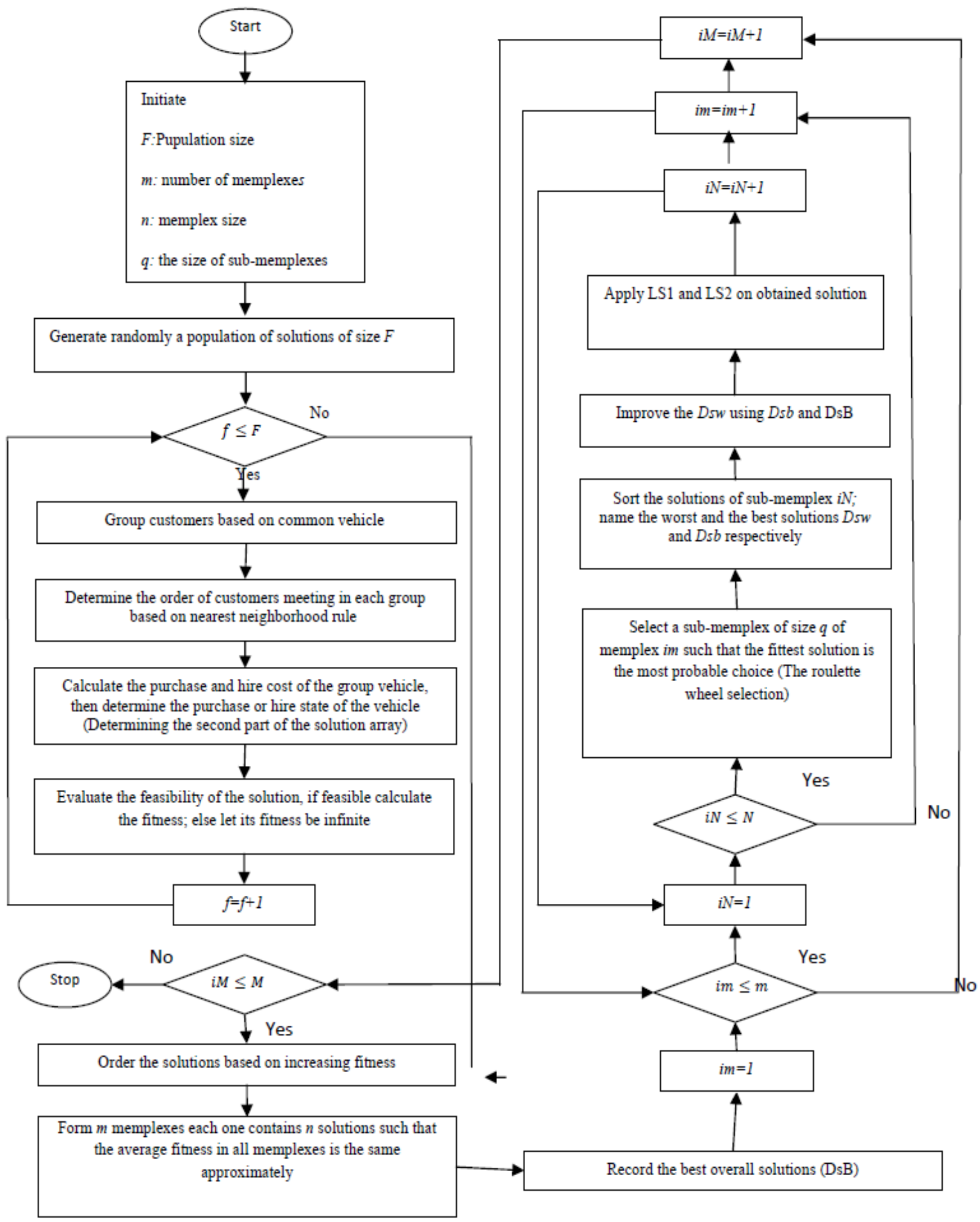

Figure 4. Flowchart of the proposed shuffled frog leaping algorithm

3.2.1 Solution representation

The code of solution consists of three parts; the first part determines the sequence of customers in meeting, the second part indicates that each customer has been met by which vehicle and the third part shows that which vehicle is purchased and which one is hired. Part I is an array of $(|V|-1)$ cells where the value of cell i shows the customer that has priority i in meeting sequence and V is defined in Table 1. Part II of the solution code is an $(|V|-1)$-dimensional array. Each element value of the array is a real number in the interval of [*1, K*] where K is the number of available vehicles. Therefore, the integer part of each element shows the vehicle by which the related customer is met. Part III of the code is an array of K elements where the value of element i is 0, 1 and 2 if vehicle i is not used, is purchased and is hired, respectively.

3.2.2 Generating the initial population

At the start of the SFLA an initial population of size F is necessary. This population is generated randomly in which each solution is an array that its values are randomly generated in related intervals and after that it become feasible.

3.2.3 Sequence of customers in meetin

In the presented algorithm, the order by which the customers, assigned to a vehicle, are met is based on nearest not met neighbor rule. In other words, each vehicle selects the nearest not met neighbor of current customer from its assigned customers as the next destination.

3.2.4 Improvement the worst solution of each sub- memplex

The worst solution of each sub-memplex is improved using the best solution of the same sub-memplex (*Dsb*). If at this stage, the improvement of the worst solution is not successful, it will be improved using the best overall solution (*DsB*). If also, at this stage, we fail to improve the solution, a random solution will be produced and is replaced with the worst solution of the sub-memplex. In order to improve the worst solution, an evolution coefficient is first calculated for each vehicle based on the framework proposed by Luo [23].

$$\alpha_k = \left|1 - \frac{l_k}{C}\right| \quad , \qquad k = 1,2,\dots,K \tag{22}$$

Where l_k is the load of vehicle k and C is the capacity of vehicle k. Secondly, corresponding to the Part II of *Dsw* and *Dsb*, two $(|V|-1)$-dimensional vector of evolution coefficient are made as α^{worst} and α^{best} for the worst and best solution of sub-memplex using Equation (23) and (24).

$$\alpha_i^{best(worst)} = \alpha_k, \quad i = 1,2,\dots,|V|-1 \tag{23}$$

$$k = \lfloor Dsb(w)_i \rfloor, \quad i = 1,2,\dots,|V|-1 \tag{24}$$

Where $\alpha_i^{best(worst)}$ is the value of i^{th} element of α^{best} or α^{worst} and $\lfloor Dsb(w)_i \rfloor$ is the integer part of i^{th}element of Dsb or Dsw. At third, the leaping step vector is obtained from the following equation:

$$S = \alpha^{best} Dsb - \alpha^{worst} Dsw \tag{25}$$

Finally, the worst solution of the memplex is changed as follows.

$$Dsw = Dsw + S \tag{26}$$

If the new solution is feasible, the purchase or hire mode will be obtained for vehicles used in this solution by calculation of the related costs. If the objective function is better than the previous solution, the new solution will be replaced by the worst solution of the memplex. If the new solution is infeasible or feasible and its objective function is not better than the previous solution, in this case, the improvement process is done by the best solution among the whole population (*DsB*). If the new solution is again infeasible or its objective function is not better than the previous solution, a solution will randomly be produced and replaced for the worst frog in the memplex.

3.2.5 Local search

After improving the worst solution of each sub-memplex, two types of local search are applied on the obtained solution named LS1 and LS2 which are described as follows.

3.2.5.1. Local search LS1

In LS1, the sequence of customers in meeting is changed for each vehicle. We test three methods for this goal that already are in the literature named iterated swap procedure (ISP) (Ho et al [19]), insertion and 2-opt swap. Our experimental results showed that the ISP method increases the computational time while it does not improve the best solution found by the algorithm compared with two other local searched. Hence, we applied the swap and insert local search for changing the order of customers in meeting in fifty-fifty frequency. In swap method, two customers of the first part of the solution code are randomly selected and their location are changed, but in the insert method a customer is selected randomly and it is inserted between two other customers which are randomly selected.

3.2.5.2 Local search LS2

The second local search which is applied to improve the worst solution in each memplex, changes the second part of the solution code by changing the assignment of customers to the vehicles. In detail, a customer (an element of Part II of the code) is randomly selected and its value is changed to another value that already exist in Part II of the solution code. In fact by this modification, a node of a vehicle tour is omitted and it is added to the tour of another vehicle. Fig. 5 shows the Part II of a solution code before and after exertion of this operator.

Before	1	1	2	2	1	1	3	2	3	2
After	1	1	3	2	1	1	3	2	3	2

Figure 5. Part II of a solution code before and after LS2

4. COMPUTATIONAL EXPERIMENTS

In order to demonstrate the efficiency of the presented algorithms some test problems are solved by both of them and the results are compared. The basic data of the test problems such as the distance matrix is based on Solomon data [24] which are designed for problems with homogeneous vehicles. To evaluate the effect of vehicle capacity on the performance of algorithms, three different levels of vehicle capacity named low, medium and high capacities are considered and in each level a number of test problems are solved.

4.1 Parameters setting

Parameters raised in the model are economic factors parameters varying through the vehicles type. The values of these parameters which are obtained by inquiry from related resources are listed in Table 4. The capacities of the vehicles for low, medium and high are 2, 5 and 8 tons respectively.

Table 4: Cost parameters of the three types of vehicles

Costs \ The type of vehicles	Low	Medium	High
Purchase cost	21600	68000	90000
Fuel cost per hour	5.2	1.6	3
Maintenance and repair cost per hour	4	5.8	6.4
Hire cost per hour	10	15	20
Salvage value	11000	30000	40000

The factor of annual interest rate has been considered as 22% which is monthly compounded. Also the planning horizon has been considered as 5 years that in each year the task of supply and distribution is performed 360 times.

In order to do experiments for selecting a group of parameters for SFLA, instances with various dimensions in the number of customers and vehicles have been considered. In so doing, 12 classes of the sample problem with the number of customers of 5, 6, 7, 8, 9 and 10 in small and large cases and 8 classes of the sample problems with the number of customers of 5 and 12 in medium case were evaluated. In other words, in general, 20 classes of sample problems were tested and the results obtained from the algorithm were compared with the optimal values obtained from GAMS software. Finally, parameters of SFLA were set as listed in Table 5.

Table 5: The values of SFLA parametrs in the experiments

Parameter	F	M	N	M
Value	50	10	50	20

In order to set the parameters of PSA-EM the test problems used for setting SFLA have been applied. The results obtained from running the algorithm were compared with the optimal values Based on our observations in this experiment, the best level of parameters of PSA-EM is 10 for the number of initial population, 20 for the number of iteration in each temperature and [-10, 10] for the interval which the array elements get values.

To evaluate and compare the presented algorithms, they have been programmed in C #.net software and they applied to solve the problem instances in various environments by a machine with CPU Intel Core i5-2450M, 2.5GHZ and 4GB RAM. Several environments, in which the problem instances are generated and solved, are designed based on the framework presented by Rodríguez and Ruiz [25]. In this experiment the varying factors and their different levels of them are listed in Table 6.

Table 6: The varying parameters and their values in several environments

Control parameters	Various modes
The number of customers	Small (10 nodes), Large (100 nodes)
The distribution pattern of customers	Random, group
The distance matrix	Low, high
The location of depot	Between customers, out of customers scope
The capacity of vehicles	Small (200), medium (700), large (1000)

The distribution pattern of customers is considered in grouped and random style. In grouped pattern the customers are clustered in several groups which are far from each other. The example of this pattern is the districts of a city in which the customers are located.

4.2 Experimental results

According to Table 6, there exists 48 different environments that in each one 5 problem instances are randomly generated and solved. The results of solving these 240 instances are gathered in Table 7 and 8. Comparison of two algorithms in tested categories has been reported as the difference percentage between the mean values of the objective functions calculated by Equation (27).

$$\frac{f_{SFLA} - f_{PSA-EM}}{f_{PSA-EM}} * 100 \tag{27}$$

Table 7: The effect of distribution pattern on the performance of the algorithms

Problem type \ Problem size			Small	Large
Distribution pattern	Depot location	Vehicle capacity	Difference percent (%)	Difference percent (%)
Grouped	Center	Small	0	36.7
Random	Center	Small	1.8-	20.5
Grouped	Out	Small	0	31.9
Random	Out	Small	-1.8	23.5
Grouped	Center	Medium	0.8	11.1
Random	Center	Medium	1.3-	4.1-
Grouped	Out	Medium	3.4	22.5
Random	Out	Medium	2.4-	7.7
Grouped	Center	Large	0	26.5
Random	Center	Large	-0.5	7.7
Grouped	Out	Large	0	19.6
Random	Out	Large	11	18.2

Table 8: The effect of distance matrix on the performance of the algorithms

Problem type \ Problem size			Small	Large
Distance Matrix	Depot location	Vehicle capacity	Difference percent (%)	Difference percent (%)
Low	Center	Small	2	21.5
High	Center	Small	0	40.2
Low	Out	Small	0.7	20.9
High	Out	Small	0.3	28.5
Low	Center	Medium	1	6.7
High	Center	Medium	0	16.1
Low	Out	Medium	2.9	15
High	Out	Medium	0.5	16.8
Low	Center	Large	4	10
High	Center	Large	0	20.3
Low	Out	Large	8.3	14
High	Out	Large	0.3	21.4

The following results about tested problem instances can be concluded from Table 7 and Table 8.

- In the small size instances with random distribution pattern and center depot, SFLA algorithm leads to better results specially when the vehicles have medium capacity.
- The PSA-EM outperforms the SFLA algorithm in all environments except the ones mentioned above.

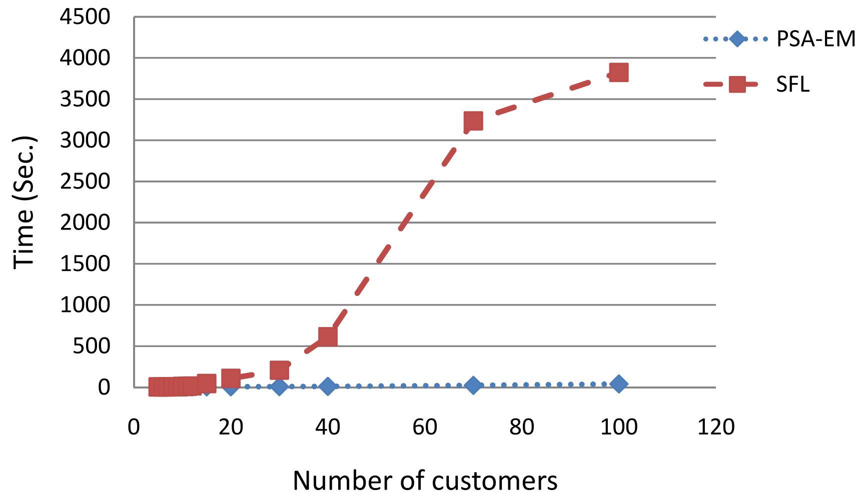

Figure 6. The computational time of the PSA-EM and SFLA algorithms

- When the size of the instances increases, the relative advantage of PSA-EM increases in all environments.
- In the category of small problems that the extent of the area of the customers is high and depot is located in the center of customers, the results obtained from two algorithms are equal.
- In the category of small problems that the extent of the area of the customers is small, the results of PSA-EM is better.

As depicted in Fig. 3, the PSA-EM is considerably faster than SFLA.

5. CONCLUSION

The route length each vehicle traveled in an optimal solution of the capacitated vehicle routing problem (CVRP) is dependent to the way that the vehicle is procured. Hence, in this paper, the problem of CVRP integrated with the decision of purchase or hire of vehicles has been studied. Considering the time value of the money, the problem is formulated as an ILP model in which the total costs (including the procurement and traveling costs of vehicles) is minimized. Since the problem belongs to the class of NP-Hard problems, two meta heuristic algorithms are presented for solving it. The first one is a hybrid algorithm of simulated annealing and electromagnetism algorithms (PSA-EM) and the second one is based on Shuffled Frog Leaping Algorithm (SFLA). Experimental results show that the PSA-EM algorithm outperforms the SFLA in most environments from the efficiency point of view.

REFERENCES

1. Dantzig GB, Ramser JH. The truck dispatching problem, *Management science* 1959; **6**(1): 80-91.
2. Prins C. A simple and effective evolutionary algorithm for the vehicle routing problem, *Computers & Operations Research* 2004; **31**(12): 1985-2002.
3. Augerat P, et al. Separating capacity constraints in the CVRP using tabu search, *European Journal of Operational Research* 1998; **106**(2): 546-57.
4. Contardo C, Martinelli R. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints, *Discrete Optimization* 2014; **12**: 129-46.
5. Luo J, Chen MR. Improved Shuffled Frog Leaping Algorithm and its multi-phase model for multi-depot vehicle routing problem, *Expert Systems with Applications* 2014; **41**(5): 2535-45.
6. Chen P, Huang Hk, Dong XY. Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem, *Expert Systems with Applications* 2010; **37**(2): 1620-7.
7. Lin SW, et al. Applying hybrid meta-heuristics for capacitated vehicle routing problem. *Expert Systems with Applications* 2009; **36**(2): 1505-12.
8. Wang CH, Lu JZ. A hybrid genetic algorithm that optimizes capacitated vehicle routing problems, *Expert Systems with Applications* 2009; **36**(2): 2921-36.
9. Ai TJ, Kachitvichyanukul V. Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem, *Computers & Industrial Engineering* 2009; **56**(1): 380-7.
10. Yurtkuran A, Emel E. A new hybrid electromagnetism-like algorithm for capacitated vehicle routing problems, *Expert Systems with Applications* 2010; **37**(4): 3427-33.
11. Garaix T, et al. Vehicle routing problems with alternative paths: An application to on-demand transportation, *European Journal of Operational Research* 2010; **204**(1): 62-75.
12. Ngueveu SU, Prins C, Wolfler Calvo R. An effective memetic algorithm for the cumulative capacitated vehicle routing problem, *Computers & Operations Research* 2010; **37**(11): 1877-85.
13. Lysgaard J, Wøhlk S. A branch-and-cut-and-price algorithm for the cumulative capacitated vehicle routing problem, *European Journal of Operational Research* 2014; **236**(3): 800-10.
14. Birbil Şİ, Fang SC. An electromagnetism-like mechanism for global optimization, *Journal of global optimization* 2003; **25**(3): 263-82.
15. Chang PC, Chen SH, Fan CY. A hybrid electromagnetism-like algorithm for single machine scheduling problem, *Expert Systems with Applications* 2009; **36**(2): 1259-67.
16. Debels D, et al. A hybrid scatter search/electromagnetism meta-heuristic for project scheduling, *European Journal of Operational Research* 2006; **169**(2): 638-53.
17. Connolly DT. An improved annealing scheme for the QAP, *European Journal of Operational Research* 1990; **46**(1): 93-100.
18. Lundy M, Mees A. Convergence of an annealing algorithm, *Mathematical programming* 1986; **34**(1): 111-24.
19. Ho W, et al. A hybrid genetic algorithm for the multi-depot vehicle routing problem, *Engineering Applications of Artificial Intelligence* 2008; **21**(4): 548-57.

20. Ho W, P Ji. Component scheduling for chip shooter machines: a hybrid genetic algorithm approach, *Computers & Operations Research* 2003; **30**(14): 2175-89.
21. Eusuff M, Lansey K, Pasha F. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization, *Engineering Optimization* 2006; **38**(2): 129-54.
22. Liping Z, et al. Application of Shuffled Frog Leaping Algorithm to an Uncapacitated SLLS Problem, *AASRI Procedia* 2012; **1**: 226-31.
23. Luo KP. A Shuffled Frog Leaping Algorithm for Solving Vehicle Routing Problem, *Applied Mechanics and Materials* 2012; **197**: 529-33.
24. Solomon MM. *Vehicle Routing and Scheduling with Time Window Constraints: Models and Algorithms*, 1984.
25. Rodríguez A, Ruiz R. A study on the effect of the asymmetry on real capacitated vehicle routing problems, *Computers & Operations Research* 2012; **39**(9): 2142-51.