

OVRP_GELS: solving open vehicle routing problem using the gravitational emulation local search algorithm

Ali Asghar Rahmani Hosseinabadi¹ · Javad Vahidi² · Valentina Emilia Balas³ · Seyed Saeid Mirkamali⁴

Received: 12 January 2016 / Accepted: 7 September 2016 / Published online: 29 September 2016
© The Natural Computing Applications Forum 2016

Abstract In open vehicle routing problem (OVRP), after delivering service to the last customer, the vehicle does not necessarily return to the initial depot. This type of problem originally defined about thirty years ago and still is an open issue. In real life, the OVRP is similar to the delivering newspapers and consignments. The problem of service delivering to a set of customers is a particular open VRP with an identical fleet for transporting vehicles that do not necessarily return to the initial depot. Contractors which are not the employee of the delivery company use their own vehicles and do not return to the depot. Solving the OVRP means to optimize the number of vehicles, the traveling distance and the traveling time of a vehicle. In time, several algorithms such as tabu search, deterministic annealing and neighborhood search were used for solving the OVRP. In this paper, a new combinatorial algorithm named OVRP_GELS based on gravitational emulation local search algorithm for solving the OVRP is proposed.

We also used record-to-record algorithm to improve the results of the GELS. Several numerical experiments show a good performance of the proposed method for solving the OVRP when compared with existing techniques.

Keywords Open vehicle routing problem · Gravitational emulation local search algorithm (GELS) · Optimization · Velocity · Newton's law · Record-to-record algorithm

1 Introduction

The vehicle routing problem (VRP) is an important problem in the fields of transportation, distribution and logistics, and the open vehicle routing problem (OVRP) is an especial case of this problem. The OVRP includes finding the best route for a set of vehicles that should deliver services to a set of customers. Every route in the OVRP includes a set of customers that start from the initial depot and end in one customer [1]. OVRP usually is limited by the following constraints: all vehicles have the same capacity, every customer should be met with only one vehicle and his/her demand should be delivered, the sum of demands of customers in a route should not exceed the capacity of all the vehicles, and in some problems some constraints are considered for traveling time and distance [2, 3]. The main purpose of solving this problem is to reduce the number of vehicles, travelled time and distance by the vehicle.

The most important difference between OVRP and VRP is the existence of Hamiltonian routes and Hamiltonian cycles in these problems, respectively; Hamiltonian routes start from a point and end in another point, while Hamiltonian cycles return to the starting point finally [3, 4]. Thus, one of the main features of OVRP is that transportation

✉ Javad Vahidi
Jvahidi@iust.ac.ir

Ali Asghar Rahmani Hosseinabadi
A.R.Hosseinabadi@iaubeh.ac.ir

Valentina Emilia Balas
Valentina.balas@uav.ro

Seyed Saeid Mirkamali
S.Mirkamali@pnu.ac.ir

¹ Young Researchers and Elite Club, Ayatollah Amoli Branch, Islamic Azad University, Amol, Iran

² Iran University of Science and Technology, Tehran, Iran

³ Aurel Vlaicu University, Bd. Revolutiei 77, 310130 Arad, Romania

⁴ Department of Computer Engineering and IT, Faculty of Computer Science, Payame Noor University, Tehran, Iran

vehicles do not necessarily return to the starting depot after delivering service to the customer, and if they do so, they will meet the same customers in the delivering route [5]. OVRP is an NP-hard problem, and its solution is a scientific challenge. Conventional studies about OVRP have solved this kind of problems by assuming definite respond for the demands of all customers on the route.

Recently, several heuristic and meta-heuristic methods based on tabu Search, GA, PSO, ACO, ICA, memetic algorithm and so on have been proposed by researchers to solve the OVRP. We could cover some of them in the following paragraphs as the literature review.

In [6], a new tabu search algorithm was proposed to solve the vehicle routing problem in which vehicles do not return to the distribution depot after delivering goods to the customers or, if they do so, they must visit the same customers, for the collection of goods, in the reverse order. The authors compared the performance of their algorithm with other heuristic method designed for the same problem in terms of computation time. Erbao et al. [7] described the customer's demand as specific bounded uncertainty sets with expected demand value and nominal value and investigated the transportation costs and unsatisfied demands in the specific bounded uncertainty sets and proposed four robust strategies to cope with the uncertain demand and proposed an improved differential evolution algorithm (IDE) to solve the optimization problem. Computational experiments indicate that the robust optimization can avoid unmet demand while incurring a small extra cost. The proposed strategy balances the cost with unmet demand among different robust strategies.

In [8], a combination of GA and TS is used to solve the OVRP, in which global optimization and parallel processing features of GA and rapid local search of TS are integrated to solve the problem. A tabu search algorithm based on memory is proposed in [9] for solving OVRP in which customers' demand is provided with a fixed number of vehicles with different capacities. But this method could only solve the problems with fixed number of customers (includes 50, 75, 100, 150, 200), while our method is able to solve the problems with variable numbers of costumers (for example, 50, 71, 100, 120, 134, 150, 199,...). And also it could solve large-scale problems that contain numerous costumers (like 200, 240, 280, 320, 360, 400, 440, 480), which is one of the superiority of the proposed method. Combinatory tabu search algorithm [10] and improved tabu search [11] were implemented by Huang and Liu in order to reduce the number of transportation vehicles and traveling cost of OVRP. Based on simulation results, the algorithm is able to reduce the number of vehicles and traveling cost.

Shamshirband et al. [12] proposed a solution to solve the multi-objective commodity vehicle routing problem, in which costumers have different demands. So a new model based on integer linear programming hybrid with time windows and various demands was introduced to optimize the traveling cost and also maximize the number of demands. And also two approaches based on the NSGA-II algorithm with different mutation designs were proposed.

Hadji et al. presented a method for solving vehicle routing combinatorial optimization problem. Their method included six heuristic methods and memetic algorithm. Six heuristic algorithms were used to generate the solutions, and memetic algorithm was used to improve the obtained solutions. To increase the quality and efficiency of the method, time windows were added to 144 samples, including 15–255 customers and 15–786 data (presented by Iori et al. [13]).

Norouzi et al. [14] used multi-objective PSO for solving OVRPC (the same OVRP but in competitive locations) with two objectives: to reduce the traveling cost and increase the sales in competitive locations. The proposed method was compared with similar methods to solve the problem. In [15], PSO algorithm with other algorithms like "nearest insertion algorithm" and "GENA" has been used to solve the OVRP and improve the obtained solution; these algorithms are able to optimize the inside and outside routes. Hu and Wu [16] presented an algorithm for solving the OVRP that is based on Genetic algorithm laws and was applied to improve the performance of PSO and differential evolution algorithms. In this algorithm, the dominant and non-dominant character includes every person. In [17], a specific decoding method for implementing PSO algorithm was proposed to solve OVRP, in which a vector based on customer locations was generated in descending order and then every customer was assigned to one path based on its location. Finally, a one-step jump was applied to the generated routes and the feasibility of routes and the solution quality were checked; it is a more effective method to solve the problem. But, the feasibility of this method was not investigated for large-scale problems and it was applied only to small problems. Both of the methods [16, 17] are used for solving the vehicle routing problem with time window and multiple service workers. They were implemented based on local deterministic search.

In [18], various meta-heuristic techniques such as evolutionary approaches, ant colony optimization, simulated annealing, tabu search and other recent approaches and their applications to solve the group technology/cell formation problems have been investigated. In the studied experiments, it was shown that ACO has better performance in producing the best solutions in comparison with GRASP.

In [19], a new meta-heuristic algorithm named OVR-P_ICA based on the imperialist competition algorithm was proposed to solve the OVRP. It was a hybrid optimization method and can service the costumers by a homogeneous fleet of vehicles that do not return to the original depot necessarily. The algorithm was compared with other algorithms, and results indicated the high performance of the proposed algorithm to solve the open vehicle routing problem.

In [20], a clonal selection algorithm is proposed by Pan and Fu to solve open vehicle routing problem (OVRP) in which a new definition for continuity of antibodies and antibody variety conservation algorithm is presented.

In summary, the some of the reviewed methods could solve the OVRP for small problems [16, 17]. Some other solved OVRP with extra vehicles [18, 20] and left the OVRP as open issue. Based on our knowledge, none of the reviewed methods could solve the OVRP in optimum time with minimum number of vehicle and travel distance. In this paper, a new optimization algorithm called “OVRP_GELS” is proposed to solve OVRP with optimum number of vehicles travelled distance and travel run time for large problems.

The paper is organized as follows: in Sect. 2, gravitational emulation local search algorithm and record-to-record algorithm are explained. In Sect. 3, the proposed algorithm is explained, Sect. 4 describes the simulation results of the proposed algorithm in detail, and the conclusion is presented in Sect. 5.

2 Review of combined algorithms

Because we combined two different optimization algorithms, namely GELS and Record-to-Record, in the following subsections we try to introduce them in brief.

2.1 Gravitational emulation local search algorithm

In 1997, guided local search (GLS) algorithm was proposed by Voudouris and Tesang [21] to search and solve complex problems for the first time. In 2004, Webster [22] called this method gravitational emulation local search (GELS) and was used as a powerful local search algorithm for solving complex problems. The main idea of GELS is based on gravitational force, which causes to attract objects with each other, such a way that heavy object has the higher gravitational force and attract low weigh objects. The attraction force between two objects depends on the distance between them.

In GELS algorithm, possible solutions in the search space are divided into several categories according to their fitness's. Each of these categories named a dimension of the

solution, and there is initial velocity for them. Equation (2.1) computes the gravitational force between current solution (CU) and candidate solution (CA). This force (F) is added to the velocity vector in the path of current motion.

If velocity exceeds the maximum value (threshold), maximum velocity becomes the current velocity, and if the velocity becomes negative due to this force, the velocity is considered zero [23–29].

$$F = \frac{G(CU - CA)}{R^2} \quad (2.1)$$

The available parameters in GELS are as follows:

Maximum velocity The maximum value that can be assigned to each entry of the initial velocity vector, which prevents the further expansion of these entries.

Radius The radius (R) employed in the gravity calculation formula.

Iteration The maximum number of algorithm iterations, which ensures that the algorithm will terminate [23–29].

The pseudo-code of the GELS algorithm is described in Algorithm 1. This algorithm shows that an initial response to a problem is created, and each mass is evaluated. Next, the problem is updated as G , or *Best*, and/or *Worst*, and parameters m and a are calculated for each mass. Then *velocity* and the *location* of each mass are also updated. Finally, the algorithm ends, if the maximum number of iterations meets or all the initial velocity vector elements become zero. Otherwise, the algorithm goes back to step 2 and continues until an optimal answer is reached.

Algorithm1: The Pseudo-code of the GELS algorithm

```

Generate initial population;
while termination condition is not satisfied do
    Evaluate the fitness for each agent;
    Update the G, Best and Worst of the population;
    Calculate  $m$  and  $a$  parameters for each agent;
    Update velocity and position;
end
return best solution;

```

2.2 Record-to-record algorithm

Record-to-record algorithm (RTR) was developed by Duke in 1993 as a deterministic simulated annealing algorithm [30]. Assume that S is a current solution and $N(S)$ is a neighborhood of S including the proposed solutions that are in the neighborhood of S and a proposed solution $S \wedge' \in N(S)$ is selected and the corresponding objective function $f(S \wedge')$ is also calculated.

In record-to-record algorithm, record is the total distance which is observed so far as the best solution.

Deviation is defined by $K\% \times \text{Record}$ and if $f(S^{\wedge}) < \text{Record} + \text{Deviation}$ then S^{\wedge} is selected as a new solution and upward movements that are related to the total distance of the route, which are admissible in order to avoid trapping in a local minimum [31]. In open record-to-record (ORTR), a fixed neighborhood list with 20 customers is used, because the idea of the algorithm is that when improvement movements are carried out, a fixed number of neighborhoods is considered for every customer. In this algorithm, an initial feasible solution and also the least number of required vehicles for delivering service to the customers K_{\min} are obtained by sweeping algorithm. If none of the answers used K_{\min} number of vehicles, then a solution is selected that uses the least number of vehicles for delivering service to the customers. If the coordinates of the customers are not determined, the sweep algorithm cannot be used to generate an initial solution, because we cannot calculate the radius and angle and in this case we can employ another heuristic algorithm to generate the initial solution.

Finally, in open record-to-record (ORTR), we try to combine the paths by joining two paths together. For example, paths A and B can be combined if the demand of the combined path does not exceed the vehicle capacity and the sum of traveling distance does not exceed maximum path length. Usually finding the best path for combining two paths quickly and effectively is not possible, and in this algorithm we try to combine path A between two alternate ties with path B.

3 The proposed algorithm

We use the gravitational emulation local search algorithm (GELS) as a strategy to solve the OVRP. The objective is to find the shortest path between customers, reducing the number of vehicles and traveling time of all vehicles.

3.1 Defining the solution dimensions

In the proposed method, every dimension of solution can be considered as a customer; in fact, number of solution dimensions is equal to the number of customers. The

neighbor of the current solution (CU) in the considered dimension is the customer who has the least distance, time and most velocity with the customer of the current solution (CU) for every vehicle, and the vehicle has not met the customer so far [28].

3.2 Definition of neighborhood

In GELS algorithm unlike other algorithms, searching the neighborhood solution is not carried out randomly but every current solution (CU) has different neighbors and each of them is based on a particular change which is named moving direction toward the neighboring solution [28]. All the neighbors that are obtained based on this method are only based on this neighbor. The following procedure is used to find the neighboring solution, the customer with the least distance, time and most velocity with respect to the current solution (CU) is selected as a neighbor and candidate solution (CA) for every vehicle.

3.3 Solution method

In the proposed method, gravitational emulation local search algorithm (GELS) is applied to solve the open vehicle routing problem (OVRP). As mentioned earlier, the objective is to reduce the traveling time and distance and also reduce the number of employed vehicles. Because of the complexity of the problem, achieving to the solution is difficult even if there are a few vehicles and customers. Therefore, by considering the characteristics of GELS algorithm and its global feature, we used this algorithm to solve the OVRP.

A one-dimensional array is used to present the solution; its length is equal to the number of customers plus the number of vehicles minus 1. To initialize each solution, we put zero in the randomly selected cells of array (for the number of vehicles minus 1), and then we put the consumers randomly in remind cells.

Example 1 It is considered a particular example with three vehicles and ten customers, (3, 10). In Fig. 1, it is a solution, an array with the length 12, including the 10 customers in the order of vehicles separated by zeros. So, the vehicle 1 meets the costumers with the numbers 4, 7

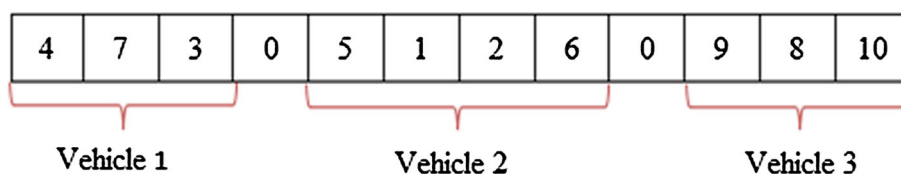


Fig. 1 The solution structure of a particular case of the open vehicle routing problem (OVRP) solved with the gravitational emulation local search algorithm (GELS). There are considered three vehicles,

separated by number 0, each cell of the solution array includes the customer's number for each of the three considered vehicles, (vehicle, [list of customers]): (1, [4, 7, 3]), (2, [5, 1, 2, 6]) and (3, [9, 8, 10])

Table 1 OVRP_GELS steps

1. To create initial response by sweep algorithm
2. To determine the perimeter of system and initialize customers who are considered as bodies
3. To arrange solutions created by customers based on each customer's weight
4. To select the first solution as the best one
5. To evaluate the weight of each customer
6. To calculate $G(t)$
7. To calculate the gravity force imposed on each customer
8. To calculate acceleration, time and speed of each customer
9. To update parameters T and V
10. To displace solutions in each dimension based on the force exerted on them in different dimensions
11. If stop condition is satisfied, run steps 12 and 13; otherwise, go to step 4
12. Run of record-to-record algorithm
13. Turn the best response

and 3, the second vehicle meets the next forth costumers 5, 1, 2 and 6, and the third vehicle meets the last three costumers 9, 8 and 10. So, for short (vehicle, [list of costumers]): (1, [4, 7, 3]), (2, [5, 1, 2, 6]) and (3, [9, 8, 10]).

Considering the objective of the problem which is reducing the traveling distance and number of vehicles, this paper introduces the gravitational emulation local search

Table 2 OVRP_GELS algorithm

```

Data:  $I, K, M, A, B, R, F, V, NBLISTSIZE$ 
Result: Distance, Time, BestFit A, BestFit B
initialization;
Parameters  $I, K, M, NBLISTSIZE, Distance, Speed, Time, A, B, R, F, V, BestFit A, BestFit B.$ 
//Generate a feasible solution using the sweep algorithm Set Record= objective
function value of the current solution
Set BestRecord= Record
Set Deviation=  $0:01 \times Record$ 
Set itr= 0
Generate a feasible solution using the GSA algorithm
Distance= Create Matrix Distance()
Speed= Create Matrix Speed()
Time= Create Matrix Time()
A= Create Parente()
B= Create Child()
BestFitA= Fitness A
BestFitB= Fitness B
while Number Of City's Region do
    Empty B
    B= Create Child()
End

```

algorithm (GELS) as a suitable strategy for solving the OVRP. Contrary to other algorithms, the GELS algorithm does not proceed with random solutions, but it proceeds by examining the current solutions (CU) and provides distinct final conditions that can complete the previous direction for a specific iteration to continue counting. Thus, although the GELS algorithm has a number of random operators-based elements, it does not proceed purely based on random operators. Although it employs the local neighborhood search method for solving the problem, it does not move in the same way between them at all. And although it includes a particular behavior of greedy algorithms, it does not find the best path for searching. The GELS algorithm uses the same law that guides the motion of objects in physical space in order to control their motion in a complex search space.

Unlike other algorithms that start from an initial population and then generate the next populations, the GELS algorithm starts from an initial or current solution (CU) and, in the next step, the secondary or candidate solution (CA) is generated based on the neighbors, customers and vehicles that can satisfy the problem constraints. In this step, the gravitational force between these two masses is calculated and it is considered as a solution and then velocity and gravitational mass are updated.

One of the advantages of this algorithm against other algorithms is the presence of velocity factor which makes this algorithm to search only for the best and optimum

Table 3 Benchmark data sets of Christofides et al. [33] and Fisher [34]

Instance	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	F11	F12
N	50	75	100	150	199	50	75	100	150	199	120	100	120	100	71	134
C	160	140	200	200	200	160	140	200	200	200	200	200	200	200	30.000	2210
L	N/A	N/A	N/A	N/A	N/A	180	144	207	180	180	N/A	N/A	648	936	N/A	N/A
e	N/A	N/A	N/A	N/A	N/A	10	10	10	10	10	N/A	N/A	10	10	N/A	N/A
in	5	10	8	12	6	5	10	8	12	16	7	10	7	10	4	7

N number of costumer, *C* vehicle capacity, *L* maximum tour length, *e* service time, *in* minimum number of used vehicles

solutions and attract them. This process leads to growth of object’s mass, and as we know, in the gravitational law, as the object mass increases, the gravitational force increases as well and thus the object attracts more optimum solutions toward itself. This procedure is repeated in each iteration until the object’s weight gets rise by attracting the optimum solutions, so the algorithm returns the best solution.

The initial solutions of OVRP can be obtained using algorithms such as sweep algorithm or in a random way. The proposed method integrates the GELS algorithm and initial solution obtained from the sweep algorithm in order to solve the OVRP. The solution procedure works as follows:

The GELS algorithm generates the conditions for OVRP by defining three parameters.

At first, sweep algorithm produces an initial solution. And then system environment and initialization to customers which are considered as masses are determined. Solutions obtained by customers are sorted based on their weight, and the first solution is considered as the best solution. In the next step, the weight of customers is evaluated. And in the next step $G(t)$, gravitational force on each costumer, acceleration, time and velocity are calculated and parameters V, T would be updated. It means that if the candidate solution (CA) obtained in each step is better than current solutions (CU), previous solution is replaced by candidate solution (CA) and then gravitational force between them is calculated using Eq. (2.1), and then parameter V is updated by Eq. (3.1).

$$V = V + F \tag{3.1}$$

In this updating formula, parameter V is increased, after that parameter T would be updated in Eq. (3.1), and it is decreased, which means that the problem is being optimized. After these steps, solutions are moved in different dimensions based on their forces and then GELS algorithm will end. Then, the algorithm performs record-to-record algorithm on produced solutions and finally the best solution obtained by the proposed algorithm would be displayed

Table 4 Benchmark data sets of Li et al. [1]

Instance	A	B	N	C	K_{min}
O1	20	10	200	900	5
O2	40	6	240	550	9
O3	28	10	280	900	7
O4	40	8	320	700	10
O5	36	10	360	900	8
O6	40	10	400	900	9
O7	44	10	440	900	10
O8	40	12	480	1000	10

(*A, B*) depot coordinate, *N* number of costumer, *C* vehicle capacity, K_{min} minimum number of used vehicles

The GELS algorithm improves distance and reduces the number of vehicles by applying the following procedure.

First, we consider three matrices: distance, initial velocity and time. The distance matrix is taken from standard problems in [32] and is calculated for every sample separately. Based on GELS algorithm, the initial velocity has an initial constant velocity at first. In this problem, the initial velocity of all masses is considered equal to 100. The initial velocity is updated in every step. As it was explained, in the initial velocity matrix, an initial velocity of 100 is assigned to each customer, which is considered as a mass, and then in the next step, velocity will change and also the time (mass) matrix based on distance. Velocity matrix is obtained from Eq. (3.2):

$$T = \frac{\sqrt{(Y_B - Y_A)^2 + (X_B - X_A)^2}}{V_{inA,B}} \tag{3.2}$$

In Equation, T indicates mass of the customer, $(Y_B - Y_A)^2 + (X_B - X_A)^2$ is the distance between two customers and $V_{inA,B}$ shows the velocity between two customers.

Using GELS algorithm in these conditions, an appropriate reserve factor should be defined. The reserve factor is equal to the number of reserved customers for cars in the future. The set of assigned customers to the cars is a

Table 5 Comparison of the proposed method with ten other methods of solving OVRP based on the solutions that minimize the number of vehicles with least distance traveled in sixteen different problems

Algorithms name	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	F11	F12
ORTR [1]	416.06	567.14	639.74	733.13	924.96	412.96	568.49	644.63	756.38	876.02	682.54	534.24	896.50	591.87	177.00	769.66
HES [36]	416.06	567.14	639.74	733.13	894.11	412.96	584.15	644.63	764.56	888.46	682.12	534.24	910.26	591.87	177.00	769.55
IVND [39]	416.06	567.14	639.74	733.13	916.39	412.96	568.49	644.63	756.38	876.38	682.12	534.24	900.32	591.87	177.00	770.17
DHPD [16]	408.5	567.14	617.0	733.13	897.93	412.96/ 6	552.87/ 11	644.63/ 9	752.0/ 14	875.67/ 17	682.12	534.24	909.8/ 11	568.87/ 12	-	-
Hybrid (1 + 1) ES [35]	412.95	564.06	639.25	733.13	868.44	412.95	566.93	640.89	741.44	871.58	678.54	534.24	836.55	552.64	-	-
BBMOOVRP [38]	416.06	567.14	639.74	735.18	895.19	412.96	583.19	644.63	759.35	879.38	682.12	535.28	904.04	592.16	-	-
BLSA [37]	416.06	567.14	639.74	733.13	893.39	-	-	-	-	-	682.12	534.24	-	-	177.00	769.55
ITS [10]	451.59	649.38	928.41	1201.95	1018.56	870.06	586.64	1254.28	-	-	-	-	-	-	-	-
ITS [11]	461.37	663.28	954.34	1247.52	1059.47	890.63	602.71	1261.39	-	-	-	-	-	-	-	-
Clonal selection algorithm [20]	426.81	574.14	691.51	850.48	1070.00	-	-	-	-	-	-	-	-	-	-	-
OVRP_GELS	406.994	543.676	630.099	729.864	841.672	418.759	549.562	667.384	765.394	873.691	672.582	526.823	851.937	549.372	159.725	694.419

Bold Indicates the solution that minimizes the number of vehicles with least distance traveled

ORTR Record-to-record travel solution from Li et al. [1]; Athlon 1 GHz computer with 256 MB of RAM using Linux

HES Hybrid evolution strategy from Repoussais et al. [36]; PIV 2.8GHz PC using C++ language

IVND Integrates a variable neighborhood descent from Chen et al. [39]; Pentium-IV 2.93G with 1G of RAM using C++ language

DHPD Diploid hybrid PSO with DE from Hu and Wu [16]; Pentium 2.8G, 1 GB of RAM using JAVA language

Hybrid (1 + 1) ES Hybrid combination out of a stochastic combined variable neighborhood search and a simple (1 + 1)-evolutionary strategy from Schneider and Reinholz [35]

BBMOOVRP Bumble bees mating optimization for the open vehicle routing problem from Marinakis and Marinaki [38]; Intel Core 2 DUO CPU T9550 at 2.66 GHz using Fortran 90 and was compiled using the Lahey f95 compiler, Linux 9.1

BLSA Broad local search algorithm from Zachariadis and Kiranoudis [37]; Intel T5500 processor 1.66 GHz using Visual C# language

ITS Improved tabu search from Huang and Liu [10]; 2.26 GHz processor with 1G of RAM using Matlab language

ITS Improved tabu search from Huang and Liu [11]; 2.26 GHz processor with 1G of RAM using Matlab language

Clonal selection algorithm Clonal selection algorithm from Pan and Fu [20]; Pentium-IV at 1.6 GHz with 256 MB of RAM using Matlab 6.5 language

OVRP_GELS open vehicle routing problem using the gravitational emulation local search algorithm; Pentium-IV at 2.06 GHz with 6 GB of RAM using C#.Net language

-, No value is calculated for this case

Table 6 Comparison of the proposed method with PSO based on the number of vehicles used in the best found solution

Instance	The input		PSO [17]		OVRP_GELS	
	C	V	Best val.	T	Best val.	T
A-n32-k5	32	5	487.306	0.82	411.563	0.45
A-n33-k5	33	5	424.543	0.89	406.457	0.64
A-n33-k6	33	6	462.433	0.90	415.349	0.78
A-n34-k5	34	5	508.516	0.92	425.538	0.72
A-n36-k5	36	5	519.455	0.88	432.863	0.73
A-n37-k5	37	5	486.243	0.81	443.957	0.67
F-n45-k4	45	4	463.986	0.90	422.323	0.69
F-n72-k4	72	4	177.453	0.96	114.118	0.81
P-n19-k2	19	5	168.569	0.97	125.548	0.86
P-n20-k2	20	5	170.278	0.97	136.563	0.72
P-n21-k2	21	2	163.877	0.93	128.860	0.73
P-n22-k2	22	5	167.191	0.962	132.436	0.75
P-n40-k5	40	5	349.552	0.88	316.659	0.71
P-n45-k5	45	5	391.809	0.92	318.347	0.68
P-n50-k7	50	5	407.727	0.91	309.348	0.65

Bold Indicates the solution that minimizes the number of vehicles with least distance traveled

C number of customer

V minimum number of used vehicle

Best val. indicates the number of vehicles used in the best found solution

T CPU time of the best run of the algorithms and proposed algorithm are presented, respectively

PSO Particle swarm optimization from Mirhassani and Abolghasemi [17]; using C programming language

OVRP_GELS open vehicle routing problem Using the gravitational emulation local search algorithm; Pentium-IV at 2.06 GHz with 6GB of RAM using C#.Net language

solution. This method can be used to represent a solution in the form of two ($n*n$) matrices which are equal to the number of cars and customers. Every row and column indicates one of the customers in the group, and the number in each row and column indicates the number of cars that belong to customers. When algorithm is completed, the reserved cars for the customers with the reservation factors are presented.

In order to improve the results of GELS, we run record-to-record algorithm in the form of one-point and two-point and then obtained solutions are saved. In the next step, the GELS algorithm would run on the obtained solution and therefore the obtained results are as an initial solution of the GELS algorithm. By applying GELS algorithm, one-point and two-point operations are carried out on the obtained solution again, which is different from the ordinary displacements. Every time after running the

algorithm, the velocity and time matrices are updated if they are improved. In each step, for every row and column of velocity matrix, the maximum value of matrix is selected and then if the assigned machine to that customer is changed, we exchange those machines. Then by considering the changes, the cost of the current solution (CU) must be calculated again. If the cost of current solution (CU) is better than the cost of candidate solution (CA), it would be replaced; otherwise, it would not be updated in the table and a new GELS algorithm run again on the candidate solution (CA), and a new solution is tested on the other candidate masses, and then mass and velocity values are updated. The previous solution is called the current solution (CU) and the newly obtained solution which is compared to the current Solution (CU) named candidate solution (CA).

The algorithm ends when the assigned speed is zero or the number of iterations reaches the maximum value determined by the problem [28]. In the following tables, we simply explain the steps of our algorithm (Table 1) and the pseudo-code of OVRP_GELS in detail (Table 2).

3.4 An overview of the main contributions

We propose GELS algorithm to solve the OVRP with minimum number of vehicles, traveling distance and travel time of the vehicle that has the following contributions:

1. The main difference between the proposed algorithm and the previous works like GA and PSO algorithms that start from an initial population and then generate the next populations is that the GELS algorithm starts from an initial or current solution (CU) and, in the next step, the secondary or candidate solution (CA) is generated based on the neighbors, customers and vehicles that can satisfy the problem constraints.
2. The GELS algorithm has high rate to reach the final solution and is a strong local and also efficient algorithm for the problem and is comparable with other algorithms.
3. To improve results, we combine the GELS and record-to-record algorithms.

4 Computational experiments

4.1 Benchmark data sets

We test the proposed on a set of well-known benchmarks by Christofide et al. [33], Fisher [34], Li et al. [1]. In the Fisher model, there are 14 problems C1–C14 [33]. The customers of both sets are in the interval of 50–199, in the

Table 7 Comparison of proposed method with eight state-of-the-art methods in this domain based on time used to solve the problem

Instance	ORTR [1]	HES [36]	IVND [39]	BBMOOVRP [38]	BLSA [37]	ITS [10]	ITS [11]	Clonal selection algorithm [20]	OVRP_GELS
	Time	Time	Time	Time	Time	Time	Time	Time	Time (s)
C1	6.2	98	5.78	0.08	25	16.1	14.9	1.769	1.769
C2	31.3	143	17.62	0.32	68	28.3	26.7	1.274	1.274
C3	39.5	330	44.74	0.55	103	212.5	218.4	2.845	2.845
C4	128.6	613	132.61	2.05	190	542.7	531.5	2.936	2.936
C5	380.6	1272	314.88	3.07	355	268.6	253.1	2.261	2.261
C6	10.3	215	8.69	0.10	–	68.3	61.4	–	1.831
C7	32.2	367	16.04	0.22	–	18.6	17.9	–	1.382
C8	53.2	510	33.40	0.45	–	562.8	553.1	–	1.568
C9	195.1	933	83.44	1.45	–	–	–	–	2.347
C10	363.5	1678	228.27	3.18	–	–	–	–	2.764
C11	121.6	318	84.89	1.01	85	–	–	–	1.271
C12	32.9	363	34.13	1.17	39	–	–	–	1.385
C13	120.3	890	87.08	1.15	–	–	–	–	2.937
C14	62.9	411	31.42	1.35	–	–	–	–	1.841
F11	19.5	264	12.24	–	93	–	–	–	2.124
F12	158.2	753	133.90	–	301	–	–	–	2.548
O1	365.3	452	–	2.51	612	–	–	–	1.342
O2	439.6	613	–	3.05	774	–	–	–	2.174
O3	492.8	736	–	3.17	681	–	–	–	2.301
O4	573.6	833	–	3.27	957	–	–	–	2.576
O5	766.5	1365	–	3.42	1491	–	–	–	3.187
O6	977.2	1213	–	4.06	1070	–	–	–	3.362
O7	935.4	1547	–	4.21	1257	–	–	–	4.127
O8	1126.8	1653	–	4.41	1512	–	–	–	6.119

ORTR Record-to-record travel solution from Li et al. [1]; Athlon 1 GHz computer with 256 MB of RAM using Linux

HES Hybrid evolution strategy from Repoussisa et al. [36]; PIV 2.8GHz PC using C++ language

IVND Integrates a variable neighborhood descent from Chen et al. [39]; Pentium-IV 2.93G with 1G of RAM using C++ language

BBMOOVRP Bumble bees mating optimization for the open vehicle routing problem from Marinakis and Marinaki [38]; Intel Core 2 DUO CPU T9550 at 2.66 GHz using Fortran 90 and was compiled using the Lahey f95 compiler, Linux 9.1

BLSA Broad local search algorithm from Zachariadis and Kiranoudis [37]; Intel T5500 processor 1.66 GHz using Visual C# language

ITS Improved tabu search from Huang and Liu [10]; 2.26 GHz processor with 1G of RAM using Matlab language

ITS Improved tabu search from Huang and Liu [11]; 2.26 GHz processor with 1G of RAM using Matlab language

Clonal selection algorithm clonal selection algorithm from Pan and Fu [20]; Pentium-IV at 1.6 GHz with 256 MB of RAM using Matlab 6.5 language

OVRP_GELS Open vehicle routing problem using the gravitational emulation local search algorithm; Pentium-IV at 2.06 GHz with 6 GB of RAM using C#.Net language

–, No value is calculated for this case

Cartesian coordination and Euclidean distance. Among them, for C6–C10, C13 and C14 the constraint on path length (L) and service time (e) is considered for all customers. Table 3 displays characteristics of 16 problems in the works of Christofide et al. [33] and Fisher [34]. Compared to VRP model, the maximum path length is a multiple of 0.9.

In addition, the set of large-scale data proposed by Li et al. [1] are also considered. The second case includes eight problems O1–O8 with 200–480 customers, and there is no constraint for path length. Every problem is geometrically symmetry, and the customers are placed in a circular manner around depot. Table 4 shows the characteristics of these eight problems.

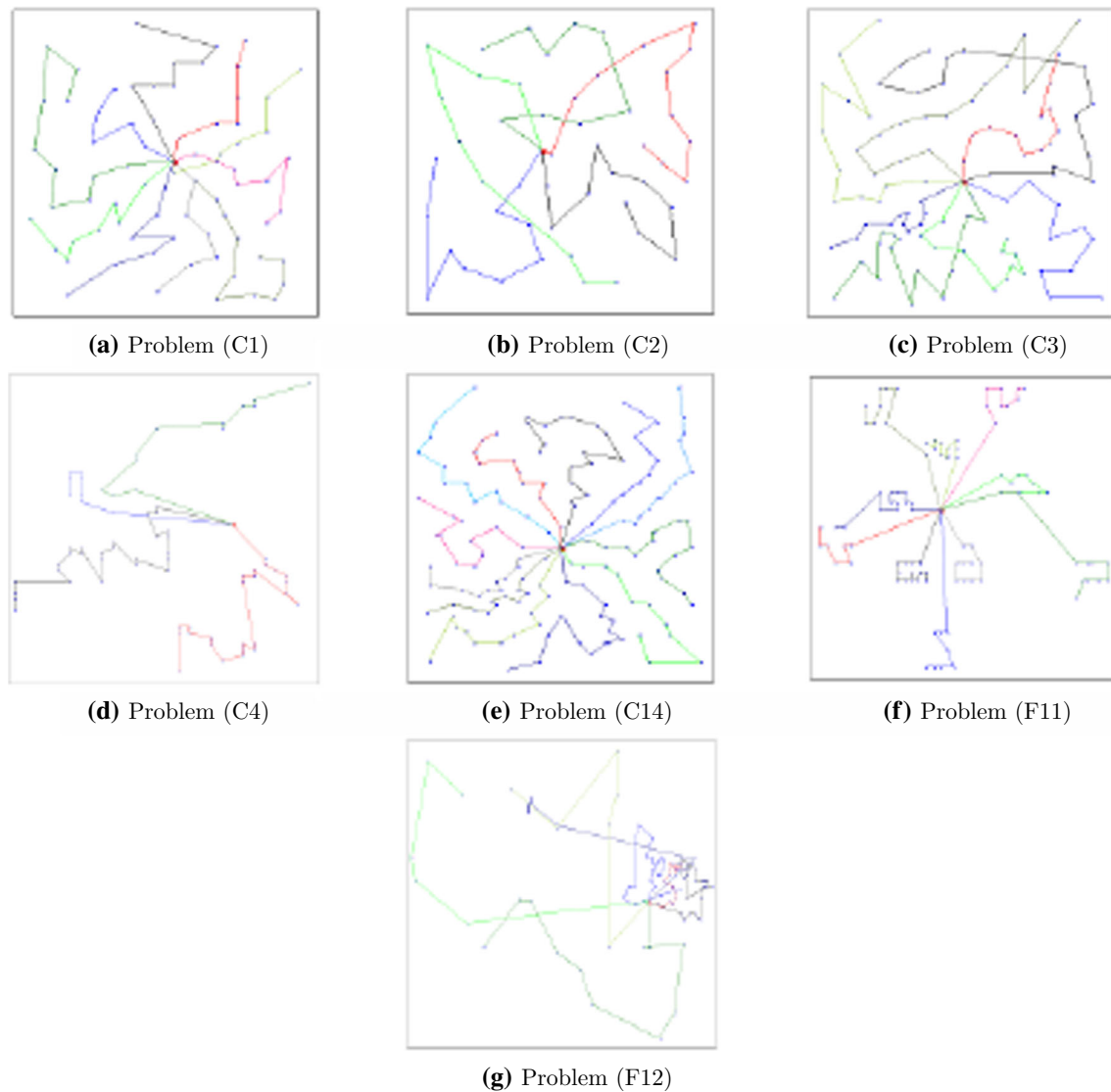


Fig. 2 The routes generated for seven problems by OVRP_GELS. With the aim of reducing the number of vehicles, the algorithm generates the best response. **a** Problem C1 ($n = 50$, solution value = **406.994**), **b** problem C2 ($n = 75$, solution value = **543.676**),

c problem C3 ($n = 100$, solution value = **630.099**), problem C4 ($n = 150$, solution value = **729.864**), problem C14 ($n = 100$, solution value = **549.372**), problem F11 ($n = 71$, solution value = **159.725**), problem F12 ($n = 134$, solution value = **694.419**)

4.2 Computational results

The results of simulation for solving the OVRP are presented in this section. C#.NET programming language is used to implement the programs, and also programs ran on a computer with 2.06 GHz Pentium-IV CPU and 6GB RAM. The proposed algorithm (OVRP_GELS) is compared to 11 different algorithms including: clonal selection algorithm [20], improved tabu search (ITS) [10], improved tabu search (ITS) [11], diploid hybrid PSO with DE (DHPD) [16], hybrid combination out of a stochastic combined variable neighborhood search and a simple (1 + 1)-evolutionary strategy ((hybrid (1 + 1) ES) [35], record-to-record travel solution (ORTR) [1], hybrid evolution

strategy (HES) [36], broad local search algorithm (BLSA) [37], bumble bees mating optimization for the open vehicle routing problem (BBMOOVRP) [38], integrates a variable neighborhood descent (IVND) [39], particle swarm optimization (PSO) [17]).

Considering the mentioned steps in the previous section, results are shown in Tables 5, 6 and 7. It is worthy to mention that solving the vehicle routing problem using the gravitational emulation local search (GELS) algorithm is completely novel and has no similar case on internet. Also the preformed simulation using the GELS algorithm is the first simulations performed so far. The performed comparisons demonstrate the advantages of the proposed algorithm (OVRP_GELS) in terms of time,

Table 8 Comparison of proposed method with four state-of-the-art methods in this domain based on their K_{min} and fitness

Instance	ORTR [1]		HES [36]		BBMOOVRP [38]		BLSA [37]	OVRP_GELS	
	K_{min}	Fitness	K_{min}	Fitness	K_{min}	Fitness	Fitness	K_{min}	Fitness
O1	5	6018.52	5	6018.52	5	6021.11	6018.52	4	6038.585
O2	9 (10)	4584.55	9	4583.70	9	4557.38	4557.38	7	4421.855
O3	7	7732.85	7	7733.77	7	7735.14	7731.00	6	7206.972
O4	10	7291.89	10	7271.24	10	7267.18	7253.20	9	7244.135
O5	8 (9)	9197.61	8	9254.15	8	9198.25	9193.15	7	9267.282
O6	9 (10)	9803.80	9	9821.09	9	9798.19	9793.72	8	9796.370
O7	10 (11)	10,374.97	10	10,363.40	10	10,351.18	10,347.70	9	9929.649
O8	10	12,429.56	10	12,428.20	10	12,418.57	12,415.36	9	12004.09

Bold Indicates the solution that minimizes the number of vehicles with least distance traveled.

N number of costumer

K_{min} minimum number of used vehicle; (), number of vehicles required if different from K_{min} .

Fitness indicates the number of vehicles used in the best found solution

ORTR Record-to-record travel solution from Li et al. [1]; Athlon 1 GHz computer with 256MB of RAM using Linux

HES Hybrid evolution strategy from Repoussisa et al. [36]; PIV 2.8 GHz PC using C++ language

BBMOOVRP Bumble bees mating optimization for the open vehicle routing problem from Marinakis and Marinaki [38]; Intel Core 2 DUO CPU T9550 at 2.66 GHz using Fortran 90 and was compiled using the Lahey f95 compiler, Linux 9.1

BLSA Broad local search algorithm from Zachariadis and Kiranoudis [37]; Intel T5500 processor 1.66 GHz using Visual C# language

OVRP_GELS Open vehicle routing problem using the gravitational emulation local search algorithm; Pentium-IV at 2.06 GHz with 6 GB of RAM using C#.Net language

minimum number of vehicles and minimum travelled distance.

Tables 5, 6 and 7 demonstrate the results of the proposed algorithm (OVRP_GELS) compared to other well-known algorithms that solved the OVRP in different kinds from easy to difficult problems. As highlighted in Table 5, the proposed algorithm (OVRP_GELS) could obtain more optimum and more acceptable solutions. OVRP_GELS has also been able to reduce the number of vehicles in difficult problems in an efficiently manner. One of the great and exclusive advantages of the proposed algorithm (OVRP_GELS) is the time factor. As it is clear from the time column in Tables 6 and 7, proposed algorithm could be able to achieve the final solution in a shorter time compared to other algorithms.

Run times by the OVRP_GELS and those of other algorithms are compared in Table 7. As it seen, the proposed algorithm could solve the problems in a reasonable time and also compete with other algorithms. OVRP_GELS has acceptable computational time for large-scale test problems which is 1.342 s for 200 customers and 6.119 seconds for 480 customers that approved the superiority of the proposed algorithm. No time value is calculated for two algorithms DHPD [16], hybrid (1 + 1) ES [35], so we did not compare them in Table 7.

Figure 2 shows the solving seven OVRPs from small to medium sizes. As you can see, the proposed algorithm (OVRP_GELS) is able to obtain desire outputs compared to other algorithms. It also can reach final answer in least time that is one of superiorities of the proposed algorithm (OVRP_GELS) than compared ones.

4.3 Large-scale test problems and computational results

In OVRP, there are eight large-scale problems that do not have any traveling distance constraint; therefore, their solution is difficult and time consuming. In this paper, we considered large-scale vehicle routing problem which is solved by Li et al. [1]. We selected eight problems with 200–480 customers and without traveling distance constraint. Every problem is geometrically symmetry, and customers are arranged in a circular way around depot. Every problem shows a geometrical symmetry that allows us to provide one solution. The obtained solutions by OVRP_GELS and solutions of other algorithms are provided in Table 8. In every problem, OVRP_GELS uses less number of vehicles (K_{min}). In Table 8, BLSA [37] has no value for K_{min} .

Figure 3 shows solving of eight large-scale OVRPs. As it is seen, the proposed algorithm (OVRP_GELS) is able to

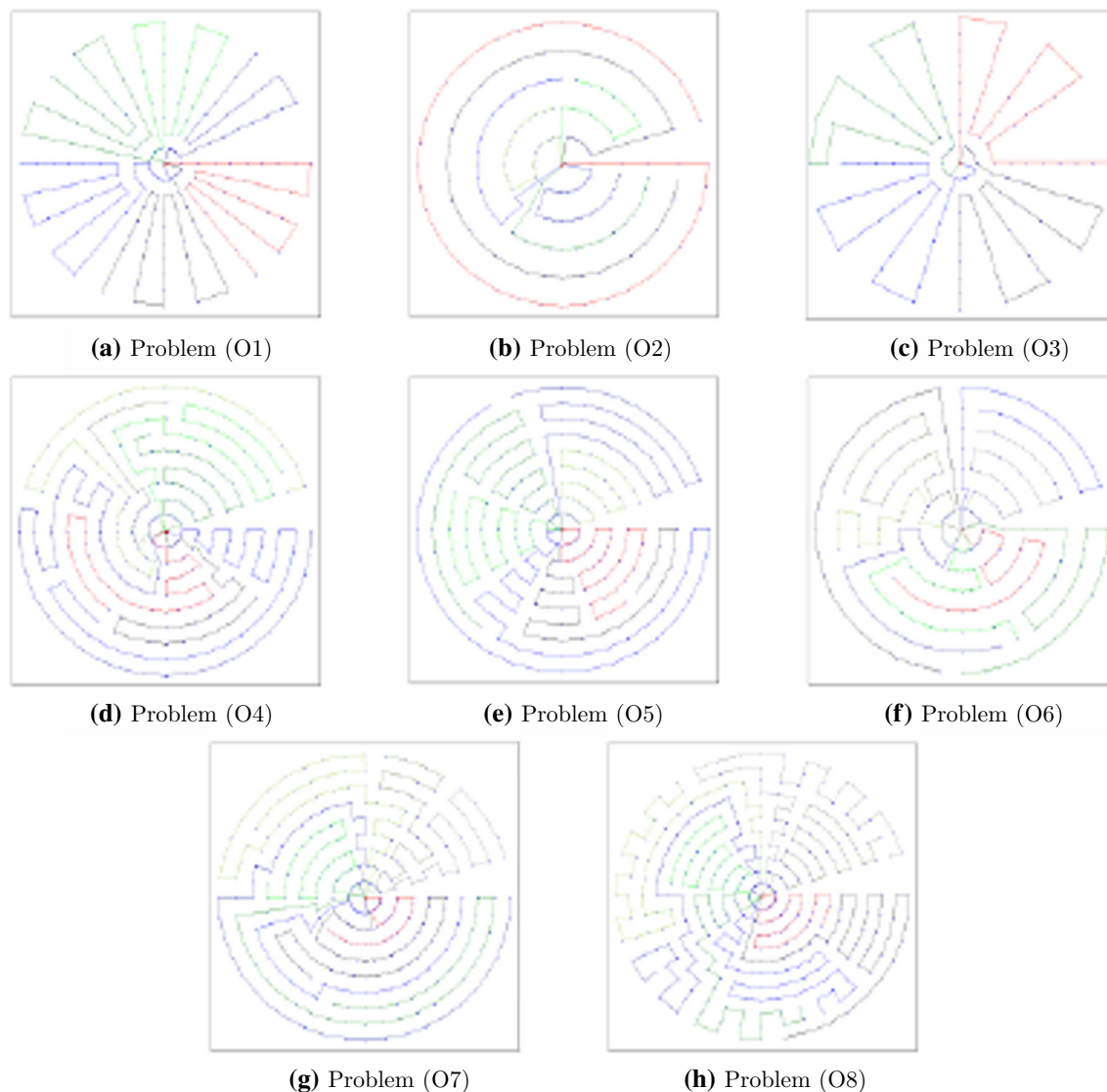


Fig. 3 The routes generated by OVRP_GELS for eight problems. With the aim of reducing the number of vehicles, the algorithm generates the best response. **a** Problem O1 ($n = 200$, solution value = **6038.585**), **b** problem O2 ($n = 240$, solution value = **4421.855**), **c** problem O3 ($n = 280$, solution value = **7206.972**), **d** problem O4 (n

= 320, solution value = **7244.135**), **e** problem O5 ($n = 360$, solution value = **9267.282**), **f** problem O6 ($n = 400$, solution value = **9796.370**), **g** problem O7 ($n = 440$, solution value = **9929.649**), **h** problem O8 ($n = 480$, solution value = **12004.09**)

obtain desire outputs, reduce the number of vehicle in each problem and reach the final response in a shorter time.

4.4 Statistical analysis: advantages and disadvantages of GELS

As mentioned earlier, the proposed algorithm (OVRP_GELS) could obtain more optimum and more acceptable solutions compare to other methods. Moreover it has been able to reduce the number of vehicles with less time consumption efficiently. As shown in Sect. 4.3, the proposed method could perform better in large-scale problems (in six out of eight problems).

Because of the nature of GELS algorithm which is a local search algorithm, this method will stock in local minima in few cases that would be the limitation of this method. We try to reduce this side effect by using record-to-record algorithm combined with GELS.

5 Conclusion

In this paper, an optimization algorithm based on gravitational emulation local search algorithm named OVRP_GELS is proposed to solve the OVRP. The advantages of this algorithm are as follows: high speed and very low run time and objective

values. The objective of the OVRP_GELS algorithm is to reduce the run time and number of vehicles and to find the shortest path between customers and vehicles. The results of GELS on several instances are compared with existing results of other eleven algorithms including the clonal selection algorithm and particle swarm optimization. The results show that OVRP_GELS algorithm is potentially more efficient when compared to other algorithms, at least on the tested benchmarks.

The running time of GELS is proposed algorithm is much lower, and it requires much less computational time to solve the problem. Since the required time for achieving the final solution, finding the shortest time and reducing the number of vehicles is an important factor in solving OVRP; therefore, less computational time is an obvious advantage of the proposed algorithm.

The main improvement of the GELS is for small and average results: the lowest value of 159.725 for problem F11, the maximum value of 873.691 for problem C10, for large-scale O2 instance the lowest value of 4421.855 and the maximum value of 12,004.09 for problem O8.

References

- Li F, Golden B, Wasil E (2007) The open vehicle routing problem: algorithms, large-scale test problems, and computational results. *Comput Oper Res* 34:2918–2930
- Fleszar K, Osman IH, Hindi KS (2009) A variable neighbourhood search algorithm for the open vehicle routing problem. *Eur J Oper Res* 195:803–809
- Csiszár S (2005) Route elimination heuristic for vehicle routing problem with time windows. *acta polytech hung* 2(2):77–89
- Hosseiniabadi AR, Kardgar M, Shojafar M, Shamshirband Sh, Abraham A (2015) Gravitational search algorithm to solve open vehicle routing problem. In: 6th International conference on innovations in bio-inspired computing and applications (IBICA 2015), chapter advances in intelligent systems and computing, Kochi, India, Springer, pp 93–103
- Sariklis D, Powell S (2000) A heuristic method for the open vehicle routing problem. *J Oper Res Soc* 51:564–573
- Brandao J (2004) A tabu search algorithm for the open vehicle routing problem. *Eur J Oper Res* 157:552–564
- Erbao C, Mingyong L, Hongming Y (2013) Open vehicle routing problem with demand uncertainty and its robust strategies. *Expert Syst Appl* 41:3569–3575
- Yu Sh, Ding Ch, Zhu K (2011) A hybrid GA-TS algorithm for open vehicle routing optimization of coal mines material. *Expert Syst Appl* 38:10568–10573
- Li X, Leung SCH, Tian P (2012) A multistart adaptive memory-based tabu search algorithm for the heterogeneous fixed fleet open vehicle, routing problem. *Expert Syst Appl* 39:365–374
- Huang F, Liu C (2010) A hybrid tabu search for open vehicle routing problem. *CCTAE* 1:132–134
- Huang F, Liu C (2010) An Improved tabu search for open vehicle routing problem. *MASS*, pp 1–4
- Shamshirband S, Shojafar M, Hosseiniabadi AR, Abraham A (2014) A solution for multi-objective commodity vehicle routing problem by NSGA-II. In: International conference on hybrid intelligent systems (HIS), pp 12–17
- Hadji SKh, Prins Ch, Yalaoui A, Reghioui M (2013) Heuristics and memetic algorithm for the two-dimensional loading capacitated vehicle routing problem with time windows. *Central Eur J Oper Res* 21(2):307–336
- Norouzi N, Tavakkoli-Moghaddam R, Salamatbakhsh A, Alinaghian M (2009) Solving a novel bi-objective open vehicle routing problem in a competitive situation by multi objective particle swarm optimization. *J Appl Oper Res* 1:15–29
- Zhen T, Zhu Y, Zhang Q (2009) A particle swarm optimization algorithm for the open vehicle routing problem. *ESIAT* 2:560–563
- Hu F, Wu F (2010) Diploid hybrid particle swarm optimization with differential evolution for open vehicle routing problem. *WCICA*, pp 2692–2697
- Mirhassani SA, Abolghasemi N (2011) A particle swarm optimization algorithm for open vehicle routing problem. *Expert Syst Appl* 38:11547–11551
- Ghosh T, Sengupta S, Chattopadhyay M, Dan PK (2011) Meta-heuristics in cellular manufacturing: a state-of-the-art review. *Int J Ind Eng Comput* 2:87–122
- Shamshirband Sh, Shojafar M, Hosseiniabadi AR, Abraham A (2015) OVRP_ICA: An Imperialist-based optimization algorithm for the open vehicle routing problem. In: International conference on hybrid artificial intelligence systems (HAIS), Springer LNCS, vol 9121, pp 221–233
- Pan L, Fu Z (2009) A clonal selection algorithm for open vehicle routing problem. In: Third international conference on genetic and evolutionary computing, pp 786–790
- Voudouris C, Tsang E (1995) Function optimization using guided local search. Technical Report CSM-249, University of Essex, Colchester
- Webster B (2004) Solving combinatorial optimization problems using a new algorithm based on gravitational attraction. Melbourne
- Bagrezaei A, Makki SVA-D, Rostami A (2013) A new energy consumption algorithm with active sensor selection using GELS in target coverage WSN. *Int J Comput Sci Issues* 10:11–18
- Balachandrar SR, Kannan K (2010) A meta-heuristic algorithm for set covering problem based on gravity. *Int J Comput Math Sci* 4:223–228
- Rezaeian J, Seidgar H, Kiani M (2013) Scheduling of a hybrid flow shop with multiprocessor tasks by a hybrid approach based on genetic and imperialist competitive algorithms. *J Optim Ind Eng* 6:1–11
- Hosseiniabadi AR, Siar H, Shamshirband S, Shojafar M, Nizam Md, Nasir MH (2015) Using the gravitational emulation local search algorithm to solve the multi-objective flexible dynamic job shop scheduling problem in Small and Medium Enterprises. *Ann Oper Res* 229(1):451–474
- Hosseiniabadi AR, Kardgar M, Shojafar M, Shamshirband S, Abraham A (2014) GELS-GA: hybrid metaheuristic algorithm for solving multiple travelling salesman problem. In: International conference on intelligent systems design and applications (ISDA), pp 76–81
- Rostami AS, Mohanna F, Keshavarz H, Hosseiniabadi AR (2015) Solving multiple traveling salesman problem using the gravitational emulation local search algorithm. *Appl Math Inf Sci* 9(2):1–11
- Hosseiniabadi AR, Farahabadi AB, Rostami MS, Lateran AF (2013) Presentation of a new and beneficial method through problem solving timing of open shop by random algorithm gravitational emulation local search. *Int J Comput Sci Issues* 10:745–752
- Dueck G (1990) New optimization heuristics : the great deluge algorithm and the record-to-record travel. *J Comput Phys* 90:161–175

31. Li F, Golden B, Wasil E (2005) Very large-scale vehicle routing: new test problems, algorithms, and results. *Comput Oper Res* 32:1165–1179
32. Site: www.branchandcut.org/VRP/data/, <http://people.brunel.ac.uk/mastjjb/jeb/info.htm>
33. Christofides N, Mingozzi A, Toth P The vehicle routing problem. In: Christofides N, Mingozzi A, Toth P, Sandi C (eds) *Combinatorial optimization*. Wiley, Chichester, pp 31538, 31979
34. Fisher M (1994) Optimal solution of vehicle routing problems using minimum k-trees. *Operations Research*, pp 626–642
35. Schneider H, Reinholz A (2008) Integrating variable neighborhood search into a hybrid evolutionary strategy for the open vehicle routing problem. *EU/MEeting 2008—Troyes, France, October 23_24*, pp 1–6
36. Repoussisa PP, Tarantilis CD, Braysy O, Ioannou G (2010) A hybrid evolution strategy for the open vehicle routing problem. *Comput Oper Res* 37:443–455
37. Zachariadis EE, Kiranoudis ChT (2010) An open vehicle routing problem metaheuristic for examining wide solution neighborhoods. *Comput Oper Res* 37:712–723
38. Marinakis Y, Marinaki M (2014) A bumble bees mating optimization algorithm for the open vehicle routing problem. *Swarm Evol Comput* 15:80–94
39. Chen P, Qu Y, Huang H, Dong X (2008) A new hybrid iterated local search for the open vehicle routing problem. *Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, pp 891–895