Decision Making: Applications in Management and Engineering
Vol. 2, Issue 2, 2019, pp. 100-111.
ISSN: 2560-6018
eISSN: 2620-0104
DOI: https://doi.org/10.31181/dmame1902076r

A NOVEL MEMETIC GENETIC ALGORITHM FOR SOLVING TRAVELING SALESMAN PROBLEM BASED ON MULTI-PARENT CROSSOVER TECHNIQUE

Arindam Roy [1*], Apurba Manna [1] and Samir Maity [2]

[1] Deptartment of Computer Science, Prabhat Kumar College, Purba Medinipur, West Bengal, India
[2] Operations Management Group, Indian Institute of Management, Kolkata, India

Received: 5 January 2019;
Accepted: 4 May 2019;
Available online: 5 May 2019.

Original scientific paper

Abstract: *In the present study, a Novel Memetic Genetic Algorithm (NMGA) is developed to solve the Traveling Salesman Problem (TSP). The proposed NMGA is the combination of Boltzmann probability selection and a multi-parent crossover technique with known random mutation. In the proposed multi-parent crossover parents and common crossing point are selected randomly. After comparing the cost/distance with the adjacent nodes (genes) of participated parents, two offspring's are produced. To establish the efficiency of the developed algorithm standard benchmarks are solved from TSPLIB against classical genetic algorithm (GA) and the fruitfulness of the proposed algorithm is recognized. Some statistical test has been done and the parameters are studied.*

Key words: *TSP, Memetic GA, multi-parent crossover.*

1. Introduction

One best example of a well known intensively studied the combinatorial optimization problem is TSP. TSP is also too much related to different type of transportation problem (Kundu, 2017; Kar, 2018) with vehicle convenience. It is also an example of NP-hard problem (Lawler & Lenstra, 1985; Das et al., 2010; Das et al., 2011). Many researchers are trying to solve TSP with reasonable time and space. But still, there have lacunae to solve such kind of NP-hard problems. Presently two ways are implemented such as direct method (Lin-Kernighan Helsgaun, Scant-method, Sant-cycle method) and indirect such as heuristic or metaheuristics. The classical TSP involves finding the shortest path/minimum cost with the visit of all cities exactly ones except the starting one. The probe on the efficient algorithm for TSP is an open problem.

*Corresponding author
Email: royarindamroy@yahoo.com (A. Roy), pkcollege.contai@gmail.com (A. Manna), sumanta@iimcal.ac.in (S. Maity).

Moscato (1989) introduced the word Memetic Algorithm (MA) as a combination of population-depended global search and the heuristic local search based on every of the individuals. From the different context of view, MAs are recently used with different names like hybrid evolutionary algorithms (Martinez-Estudillo, 2005), many kinds of literature found in (Jampani, 2010; Li & Feng, 2013; Silberholz, 2013; Hiremath & Hill, 2013; Nesmachnow, 2014; Skinner, 2015), a different Lamarckian Evolutionary Algorithms are studied in (Omran, 2016), very recently a Cultural Algorithms developed found in (Reynolds & Peng, 2004). In the case of colonial optimization, the various number of instances of MA have been notifiable across a broad scope of application realm, generally merging to better-tone solutions much expeditious than the established affected counterparts. Real-world complex problems have been successfully satisfied with memetic algorithms. Although various researcher avail procedures nearly bound up to MA, through other names like hybrid genetic algorithms are also exploited. Nowadays MAs are applied in different research areas included pattern recognition (Aguilar & Colmenares, 1998), artificial neural networks (Ichimura & Kuriyama, 1998), circuit design (Harris & Ifeachor, 1998), robotic motion planning, beam orientation (Haas et al., 1998), electric service restoration (Kumar et al., 2006), medical expert systems (Wehrens, 1993), single machine scheduling (Chyu & Chang, 2010), etc. A study on multi-parent MA is found in (Wang et al., 2010) and they concluded that different combination got better results from others. Ye et al., (2014) developed a multi-parent recombination operator for solving Linear Ordering Problem but they do not restrict to chosen the parent because it increases the computational complexity. At the present investigation, only four parents are selected from the mating pool and randomly a common crossing point is chosen.

Genetic Algorithms (GAs) were first proposed by Holland (1992) whose ideas were applied and expanded on by Goldberg (1998). The classical GA has three operators, such as selection, crossover and mutation. Different kinds of selection operators (RW, Ranking, Tournament, etc.) and cyclic, partial-map, ordered based, etc. crossovers are available with a random mutation to solve the discrete optimization problem by GA.

In our proposed method (NMGA) are the combinations of probabilistic selection and adaptive four- parents crossover with the classical ergodic mutation. Now the crossover is taken from the realistic social observations. We see that some child born with legal parents but they adapted by other parents and grown up under them. In third world countries, it is very common. In the proposed crossover methods four parents are used and modified them finally comparing the costs/distance, the offspring is created.

The proposed algorithm has the following key features:

- Boltzmann probabilistic selection,
- Multi-parent adaptive crossover,
- Four parents,
- Random crossing point,
- Comparing the cost/distance genes are selected,
- Test on standard TSPLIB problems.

The remaining part of this paper is presented as follows: section 1, a short introduction is presented. In section 2, We describe mathematical pre-requisite. In section 3, the proposed modified memetic algorithm is presented. In section 4, some numerical experiments are done. Again in section 5, a brief discussion is given. Finally, in section 6, a conclusion with future scope is studied.

2. Classical Definition of TSP

The goal of a Traveling Salesman Problem (TSP) is that a salesperson would create a path. This path should be an ideal path. Ideal path means path would be the shortest while salesman completes his visit across a finite number of cities, visiting each city only once and finished at the starting city. Let G=(V, A) is a graph. G has n vertices and V is a set of this n vertices. A is also a set of arcs or edges of this G. Then G=(V, A). Let *C* = *c*(*i,j*) be a distance (or cost) matrix associated with A. The intent of TSP is determining a minimum distance or cost circuit passing through each and every vertex only once except starting node. This type of circuit is familiar as a tour or Hamiltonian circuit or cycle. In case of symmetric TSP *c*(*i,j*) = *c*(*j,i*) for all *i,j*∈ V. (n-1)! path will be generated for symmetric TSP and (n-1)!/2 path will be generated for asymmetric TSP. Now mathematically TSP defined as below.

Minimize $Z = \sum_{i \neq j} c(i,j) x_{ij}$

subject to

$$\sum_{i=1}^{N} x_{ij} = 1 \quad \text{for} \quad j = 1, 2, \ldots, N;$$

$$\sum_{j=1}^{N} x_{ij} = 1 \, \text{for} \, , i = 1, 2, \ldots, N;$$

$$\sum_{i \in S}^{N} \sum_{j \in S}^{N} x_{ij} \leq |S| - 1, \forall S \subset Q;$$

Where $x_{ij} \in \{0,1\}, i, j = 1, 2.., N$.

Now,

$Q = \{1, 2, 3,.., N\}$ set of nodes

x_{ij} = the decision variable,

$x_{ij} = 1$ if thesalesman visits from city-i to city-j,

$x_{ij} = 0$ otherwise.

Then the above TSP reduces todetermine

a complete tour $(x_1, x_2, \ldots, x_N, x_1)$;

$$Z = \sum_{i=1}^{N-1} c(x_i, x_{i+1}) + c(x_N, x_1)$$

Where, $x_i \neq x_j, i, j = 1, 2\ldots, N$.

along with sub tour elimination criteria

$$\sum_{i\in S}^{N}\sum_{j\in S}^{N} x_{ij} \le |S|-1, \forall S \subset Q \text{, where, } \quad x_{ij} \in \{0,1\}, i, j = 1,2..,N .$$

3. Proposed Memetic Genetic Algorithm

Here we propose a probabilistic selection, multi-parent crossover with the simple random mutation for solving the TSP.

3.1. Representation

Considering N cities available to make a complete tour which stands for a solution. Say an integer vector X$_i$ of N-dimensional. Where X$_i$ =(x_{i1},x_{i2},⋯ ,x_{iN}) is used as cities, and x$_{i1}$, x$_{i2}$, ⋯, x$_N$stand for N successive cities in a tour. At the beginning need a group of paths (tours) for the salesman. These paths are randomly generated for GA. These initial paths is a group of possible solutions for the GA part of this algorithm.

3.2 Selection

3.2.1. Probabilistic Selection

The main objective of TSP that minimizing the path cost/distance. So here minimum fitness value (say f$_{min}$) of the choromosome play a vital role. Matting pool is formed using the Boltzmann-Probability (Roy et al., 2018) of all chromosome from the initial population.

Now $p_B = e^{((f_{min} - f(X_j))/T)}$; T=T$_0$(1-a)^k, k=(1+100*(g/G)), g=ongoing generation number, G= maximum generation, T$_0$= rand[10,150], f(X$_i$) corresponding fitness/objectives of chromosome corresponding to X$_i$, a=rand[0,1], i=chromosome number.

3.3. Multi-Parent Crossover

Nowadays in our society adaptation is very common matter from the different practical situation. Here except original parents (father, mother), there are one more parents (father, mother) taken as a part. Inspiring this realistic happening here a new approach with four parents (first two are original parents and the other two are adoptive parents) are used to produce offspring. This urged crossover method choose four individuals or parents in an ergodic manner to create offspring. To collect optimum results of a TSP, we make a journey from one node to next node maintaining minimum traveling cost based on TSP condition. Following the above conception, we make the crossover procedure in the following condition. At first select (randomly) four individuals (parents) from the mating pool. Give an example here. PR$_1$, PR$_2$, PR$_3$, PR$_4$ are the parents and CH$_1$, CH$_2$ are the offspring.

PR$_1$:

1	2	0	3	4

PR$_2$:

0	2	4	3	1

PR$_3$:

4	0	3	1	2

PR$_4$:

3	4	1	2	0

Generate random number between 0 and 4. Suppose it is 2. Then according to our proposed algorithm it would be the starting node of a new offspring (CH_1).

CH_1:

2				

Now we have to comparing minimum traveling cost between

2	node··· (1)	(1st node of Parent1 [PR_1])
2	node··· (0)	(1st node of Parent2 [PR_2])
2	node··· (4)	(1st node of Parent3 [PR_3])
2	node··· (3)	(1st node of Parent4 [PR_4])

and if say the traveling cost node (2) to node (4) is minimum from rest three paths, then next node of the new offspring (CH_1) would be 4. So it should be like as-

CH_1:

2	4		

The above process will continue until the new offspring (CH_1) gets its all nodes maintaining the condition of TSP. Similarly, generate the second offspring but in reverse order than another. Here R_1 and R_2 are two randomly generated variable two nodes from the given set of nodes.

3.4. Random Mutation

An ergodic number r is created for every solution of P(t). Here r is generated between a range [0,1] and $r < p_m$ is a condition, if the condition is true, then a solution is selected for mutation. Two nodes are selected ergodic manner from each chromosome and they interchange their positions and replace it in the offspring set.

3.5. Proposed Novel Memetic Algorithm (NMGA)

NMGA Algorithm:

1. Input: for Crossover procedure (p_c), $Maximum_{gen}(S_0)$, (pop-size) and for Mutation procedure (p_m).
2. Output: The best solution.
3. Begin
4. Approve generation $t = 0$.
5. (Initialize) ergodic manner and generate approve population $p(t)$, here $f(x_i)$,i= 1,2,⋯,($pop - size$) state the all chromosomes.
6. All solution will be judged it's efficiency one by one from the approve population $p(t)$
7. Repeat up to (18) till ($t < S_0$)
8. Modify the current generation such as $t = t + 1$
9. Decide (p_B) for all chromosome over $p(t)$ to subsection (3.2.1)
10. Construct the mating pool on the basis $p_s$and p_B
11. For crossover parents will be chosen based on p_c over mating pool
12. According to subsection (3.3) the crossover operation will be conducted based on exclusive chromosomes/solutions
13. Produce offspring and the parents will be replaced
14. Repeat (9) to (11) based on p_c
15. Followed by the subsection (3.4) mutation process will be executed
16. Offspring will be selected for mutation depend on p_m
17. Interchange the position between selected nodes
18. Replace offspring
19. Determine the effectiveness and save the local best and near best solutions

20. Repeat (5) to (18)
21. (for best result) Store the global best and near best results
22. Stop

Proposed NMGA pseudo code:

```
Begin generation $t$ = 0;
        for (i=1 to pop-size)
         Produce chromosome ergodic manner; end for;
         for (i=1 to pop-size)
         Judge fitness;
         end for;
         for (gen=1 to max-gen)
            {
            for (j=1 to pop - size) r=rand[0,1]; $T_0$= rand[5,100];
               a=rand[0,1]; k=(1+100*(i/G));
                 T=$T_0$(1-a)$^k$;
                 $p_B = e^{((f_{min} - f(X_j))/T)}$;
            if ( $r < p_s$ )
                        select the current chromosome; j++;
              else if (r $< p_B$)
                        select f(X $_j$ ) ; j++;
              else
                       Select the corresponding chromosome of f $_{min}$ ; j++;
            end for;
            end for;
for (s=1 to ($noc * p_c$))
        $R_1$=rand[0, N-1];
        $R_2$=rand[0, N-1];
        $PR_1$=rand[0, pv-1];
        $PR_2$=rand[0, pv-1];
        $PR_3$=rand[0, pv-1];
        $PR_4$=rand[0,
        pv-1];
        $CH_1[0]=R_1$;
        i=1; do{
$CH_1$[i] = min {$c(R_1,PR_1[0]),c(R_1,PR_2[0]),c(R_1,PR_3[0]),c(R_1,PR_4[0])$}; i=i+1;}
while($CH_1$[ N-1]);
 $CH_2$[n-1]=$R_2$;
 $i = n - 2$ ;
do{
$CH_2$[i] = min {$(R_2,PR_1[N-1]),c(R_2,PR_2[N-1]),c(R_2,PR_3[N-1]),c(R_2,PR_4[N-$
$1])$};
$i = i - 1$ ;
    }
   while($CH_2$[0]) ;
  End for
 for(a=0 to noc)
{
```

```
If (rand[0,1] < $p_m$)
mutate;
}
for (i=1 to noc)
Evaluate fitness; end for
}
Stop
```

3.6. Termination Criteria

The proposed algorithm is terminated if it finishes the user-defined maximum number of generations or iterations, or the difference between consecutive iterations less than some predefined values which are earlier.

4. Numerical Experiments

4.1. Test Results of NMGA

We have taken benchmarks from TSPLIB (Reinelt, 1991) and select 53 standard instances form 7 city to 318 cities to test the performance of our proposed algorithm NMGA. Table 1 shows the comparison of performance between proposed NMGA and standard classical GA through the results presented in the form of percentage error. The total comparison held the basis of total cost. We have taken the best, average and the worst outcome of both NMGA and classical GA under 100 independent runs and the best solution is presented with relative percentage error. The parameters of the NMGA given in Table 2 for the same nodes of the benchmarks instance kora100 with 100 cities problem. We have increases pop-size, Maxgen, p_c and p_m of an instance as a parameter.

4.1.1. Sensitivity of CPU-time w.r.t. p_c and p_m

Sensitivity analysis is performed for CPU-time on the basis of concerned values of the key parameters p_m and p_c and outcomes are projected in Figure1 (three dimensions linear graph using STATISTICA). It is observed that for fixed value of p_c, as p_m increases, CPU-time increases. Also, it is observed that for a fixed value of p_m, as p_c increases, CPU-time also increases.

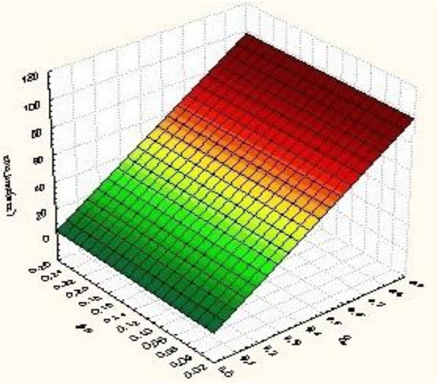

Figure 1. Sensitivity Analysis

5. Discussion

The superiority of the developed algorithm is established by solving standard benchmarks from TSPLIB which given in Table 1. This proposed algorithm NMGA is coded in C++ based on few keys like the maximum number of chromosomes (100) and a maximum number of iterations (5000). Table 1 is used to show the differences between NMGA and GA for few benchmark TSP references in TSPLIB. It is observed that the percentage of error is lesser in NMGA than the classical standard GA. Here, 53 standard instances from 7 to 318 cities are studied and most of the cases NMGA produced better results. A parameter analysis is done which is given in Table 2 for the standard benchmark kora100 with 100 cities problem.

Table 1. Performance (relative error) of benchmarks from

Instances	NMGA			Classical GA		
	Best	Worst	Average	Best	Worst	Average
sh-07	0	0	0	0	0	0
sp11	0	0	0	0	0.15	0.07
uk12	0	0	0	0.04	0.22	0.15
lau15	0	0	0	0.29	0.55	0.44
gr17	0	0	0	0.22	0.52	0.39
wg22	0	0.02	0.01	0.65	1.07	0.94
fri26	0	0.02	0.01	0.76	1.09	0.98
bay29	0	0.06	0.01	1.01	1.25	1.12
bayg29	-0.02	0.04	0.02	0.92	1.22	1.10
wi29	-0.00	0.06	0.02	1.29	1.77	1.59
ha30	0	0.08	0.02	1.14	1.53	1.37
dj38	0.00	0.04	0.01	1.90	2.24	2.12
dantzig42	0.00	0.11	0.03	1.98	2.45	2.28
swiss42	0.00	0.11	0.03	1.75	2.04	1.91
att48	2.16	2.40	2.25	9.07	10.60	10.11
eil51	0.00	0.06	0.03	1.94	2.22	2.11
berlin52	0.00	0.14	0.05	1.90	2.24	2.14
wg59	0.00	0.18	0.07	2.73	3.17	2.99
st70	0.00	0.14	0.04	3.18	3.60	3.48
eil76	0.01	0.09	0.06	3.04	3.59	3.39
pr76	0.01	0.26	0.11	3.20	3.59	3.45
rat99	0.01	0.12	0.06	14.68	16.28	15.80
kroA100	0.00	0.32	0.12	5.25	6.00	5.76
kroB100	0.02	0.27	0.10	5.05	5.64	5.44
kroC100	0.02	0.22	0.10	5.19	6.14	5.90
kroD100	0.03	0.21	0.10	5.08	5.70	5.50
kroE100	0.02	0.25	0.09	5.24	5.87	5.63
rd100	0.01	0.20	0.11	4.65	5.25	5.07
eil101	0.04	0.15	0.09	3.51	3.85	3.71
lin105	0.01	0.26	0.13	5.95	6.50	6.28
pr107	0.01	0.20	0.09	9.28	10.36	9.95
pr124	0.01	0.41	0.14	8.74	9.59	9.26
bier127	0.07	0.39	0.20	3.60	3.83	3.74
ch130	0.03	0.20	0.12	5.37	5.85	5.70
pr136	0.09	0.36	0.20	6.18	6.70	6.49
pr144	0.01	0.39	0.15	10.71	11.60	11.24

kroA150	0.06	0.37	0.18	6.91	7.80	7.52
kroB150	0.08	0.34	0.20	6.92	7.88	7.58
pr152	0.14	21.88	13.26	11.31	11.94	11.67
u159	0.03	0.27	0.16	8.13	8.80	8.45
qa194	0.09	0.27	0.17	2.28	3.20	2.23
rat195	0.06	0.25	0.16	32.79	35.82	34.81
d198	0.10	0.29	0.20	9.20	10.05	9.66
kroA200	0.13	0.45	0.26	8.95	9.73	9.37
kroB200	0.16	0.42	0.28	8.73	9.43	9.14
ts225	0.08	0.35	0.19	10.13	10.75	10.49
tsp225	0.07	0.27	0.17	8.30	8.87	8.61
pr226	0.11	0.84	0.23	16.82	18.62	18.12
gil262	0.12	0.37	0.24	9.03	9.58	9.33
pr264	0.20	0.41	0.30	17.90	20.19	19.37
a280	0.15	0.37	0.27	10.61	11.33	11.00
pr299	0.12	0.37	0.24	12.85	13.74	13.35
lin318	0.17	0.40	0.29	11.57	12.20	11.94

In Table 2, it is used to calculate the goodness of parameter of selection (p_s) in NMGA. It shows that to get the optimal solution of the standard TSP kroA100, $p_s$indicates the given space better for p_s= 0.34.

Table 2. Parameters for NMGA of kroA100 instance

Instance	p_c	pm	pop_{size}	Gen	result	cpu-time(sec)
	0.34	0.01		4673	21417	5526
		0.02		4065	21344	5733
		0.001		3407	21322	5405
		0.003		4957	21298	5373
		0.005		2427	21294	5380
		0.007		2173	21316	4810
		0.008		3376	21285	4470
		0.009		3068	21384	4214
kroA100	0.2	0.01	70	2384	21322	3331
	0.25			2787	21386	3232
	0.30			4958	21333	3916
	0.35			2612	21285	5574
	0.40			4868	21294	6164
	0.45			3883	21412	6256
	0.50			3708	21349	6149
	0.55			3505	21365	4407
	0.60			4406	21316	4438
	0.70			4467	21535	6640
	0.75			4320	21334	6874
	0.80			4975	21831	9982
	0.85			3905	21335	9840
	0.34	0.01	50	3438	21390	3175
			60	4638	21474	8382

Instance	p_c	pm	pop_{size}	Gen	result	cpu-time(sec)
			72	4453	21322	4470
			85	3460	21322	4952
			110	2706	21285	7664
			150	4700	21294	8974

It is evident from Table 2 that, for standard three parameters p_c, p_m and $pop - size$, our proposed algorithm NMGA give us optimum or near optimum result easily. Thus the importance of the parameters is discussed in Table 2.

6. Conclusion

In this paper, a novelty introduced in GA regarding selection and crossover, called Novel Memetic Genetic Algorithm (NMGA). NMGA is tested with few test references from TSPLIB and examined with classical GA. In NMGA, Boltzmann probabilistic selection and a new four parents crossover are worked with ergodic mutation. The concept of MA is not new in the area of TSPs, but the idea of multi-parent(four) crossover on the basis of the memetic concept is new, establishes our proposed algorithm as highly NP-hard combinatorial optimization problem. Realistically, it is true that multi-parent crossover especially four parent crossover may not be so much truthful than two parent crossover for a specific problem or the complexity may be high than other. But the numerical analysis proves the efficiency of our proposed algorithm. The improvement of developed NMGA is in natural form and it is also applicable to solve another discrete problem like network optimization, well-known Graph Theory, popular Standard Transportation Problems, vehicle routing problem, and electronics manufacturing units, etc. Although we have to get the much superior results by NMGA, there is also have a huge scope for improvement also.

References

Aguilar, J. & Colmenares, A. (1998). Resolution of pattern recognition problems using a hybrid genetic/random neural network learning algorithm. Pattern Analysis and Applications, 1(1), 52–61.

Baldwin, J.M. (1896). A new factor in evolution. The american naturalist, 30(3), 441–451.

Chyu, C.C., & Chang, W.S. (2010). A competitive evolution strategy memetic algorithm for unrelated parallel machine scheduling to minimize total weighted tardiness and flow time. The 40th International Conference on Computers & Indutrial Engineering, 1–6.

Das, D., Roy, A., Kar, S. (2010). Improving production policy for a deteriorating item under permissible delay in payments with stock-dependent demand rate. Computers & Mathematics with Applications, 60(7), 1973–1985.

Das, D., Roy, A., Kar, S. (2011). A volume flexible economic production lot-sizing problem with imperfect quality and random machine failure in fuzzy-stochastic environment. Computers & Mathematics with Applications, 61(9), 2388–2400.

Goldberg, D.E. & Holland, J.H. (1998). Genetic algorithms and machine learning. Machine learning, 3(2), 95–99.

Haas, O.C.L., Burnham, K.J. & Mills, J.A. (1998). Optimization of beam orientation in radiotherapy using planar geometry. Physics in Medicine & Biology, 43(8), 217-229.

Harris, S.P. & Ifeachor, E.C. (1998). Automatic design of frequency sampling filters by hybrid genetic algorithm techniques. IEEE Transactions on Signal Processing, 46(12), 3304–3314.

Hiremath, C.S. & Hill, R.R. (2013). First-level tabu search approach for solving the multiple-choice multidimensional knapsack problem. International Journal of Metaheuristics, 2(2), 174–199.

Holland, J.H. (1992). Genetic algorithms. Scientific American, 267(1), 66–73.

Ichimura, T. & Kuriyama, Y. (1998). Learning of neural networks with parallel hybrid GA using a royal road function. IEEE World Congress on Computational Intelligence in Neural Networks Proceedings, 2, 1131–1136.

Jampani, J. (2010). Integrated heuristics for scheduling multiple order jobs in a complex job shop. International Journal of Metaheuristics, 1(2), 156-180.

Kar, K. (2018). A multi-objective multi-item solid transportation problem with vehicle cost, volume and weight capacity under fuzzy environment. Journal of Intelligent & Fuzzy Systems, 35, 1–10.

Kumar, Y., Das, B., & Sharma, J. (2006). Genetic algorithm for supply restoration in distribution system with priority customers. International Conference on Probabilistic Methods Applied to Power Systems, 1–7.

Kundu, P. (2017). A solid transportation model with product blending and parameters as rough variables. Soft Computing, 21, 2297–2306.

Lawler, E.L., & Lenstra, J.K. (1985). The traveling salesman problem: a guided tour of combinatorial optimization. Wiley, 3, 1–463.

Li, W., & Feng, M. (2013). Solution attractor of local search in travelling salesman problem: concept, construction and application. International Journal of Metaheuristics, 2(3), 201–233.

Martinez-Estudillo, A.C. (2005). Hybridization of evolutionary algorithms and local search by means of a clustering method. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 36, 534–545.

Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Caltech concurrent computation program, 3, 826-839.

Nesmachnow, S. (2014). An overview of metaheuristics: accurate and efficient methods for optimisation. International Journal of Metaheuristics, 3(4), 320–347.

Omran, M.G. (2016). A novel cultural algorithm for real-parameter optimization. International Journal of Computer Mathematics, 93(9), 1541–1563.

Reinelt, G. (1991). TSPLIB-A Traveling Salesman Problem Library. ORSA Journal on Computing, 3(4), 376-384.

Reynolds, R.G., & Peng, B. (2004). Cultural algorithms: modeling of how cultures learn to solve problems. In Tools with Artificial Intelligence. IEEE International Conference, 166–172.

Roy, A., Chakraborty, G., Khan, I., Maity, S., & Maiti, M. (2018). A Hybrid Heuristic for Restricted 4-Dimensional TSP (r-4DTSP). Proceedings in Mathematics & Statistics, 285–302.

Silberholz, J. (2013). Comparison of heuristics for the colorful traveling salesman problem. International Journal of Metaheuristics, 2, 141–173.

Skinner, M.K. (2015). Environmental epigenetics and a unified theory of the molecular aspects of evolution: A neo-Lamarckian concept that facilitates neo-Darwinian evolution. Genome biology and evolution, 7(5), 1296–1302.

Wang, Y., Lu, Z., & Hao, J.K. (2010). A study of multi-parent crossover operators in a memetic algorithm. International Conference on Parallel Problem Solving from Nature, 556–565.

Wehrens, R. (1993). HIPS, a hybrid self-adapting expert system for nuclear magnetic resonance spectrum interpretation using genetic algorithms. Analytica Chimica Acta, 277(2), 313–324.

Ye, T., Lu, Z., & Hao, J.K. (2014). A Multi-parent memetic algorithm for the linear ordering problem. arXiv preprint arXiv:1405.4507.

© 2018 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).