New results in the suitability analisys of using blind crossover operators in genetic algorithms for solving routing problems

E. Osaba, R. Carballedo, F. Diaz, E. Onieva, A. Perallos

***Abstract*—This paper aims to be an extension of the previously published work "Analisys of the suitability of using blind crossover operators in genetic algorithms for solving routing problems". In that paper is shown that the crossover operator offers no advantage in the optimization process of the genetic algorithms when they are applied to routing problem. In this next step of the research we reinforce that hypothesis. For this purpose, we have conducted a new analysis of the results, and we make a new experimentation with two different vehicle routing problems, the Capacitated Vehicle Routing Problem and the Vehicle Routing Problem with Backhauls.**

***Keywords*—Genetic Algorithm, Crossover Operator, Combinatorial Optimization, Routing Problems.**

I. INTRODUCTION

GENETIC algorithms (GA) are one of the most successful meta-heuristics for solving combinatorial optimization problems. GAs were proposed in an attempt to imitate the genetic process of living organisms and the law of the evolution of species. The basic principles of GAs were proposed by Holland [1], even though its practical use for solving complex problems was shown by De Jong [2] and Goldberg [3]. Every year, GAs are the focus of a large number of books and papers [4, 5], and they have been applied in a wide range of fields, as industry [6], transport [7], and software engineering [8].

Although the conventional structure of a basic genetic algorithm is well defined, there are some works in the literature which have analyzed the suitability of the different parts of the GAs. The paper presented by Osaba et al. [9] is one example of this kind of works. In that contribution, authors demonstrate the inefficiency of one of the central steps of a GA, when it is applied to routing problems. This step is the crossover (CX). To prove their hypothesis, authors perform an experiment in which they compare the results and times obtained by using the same GA, but with different CX, with the results obtained by an evolutionary algorithm which performs no CX phase. The problem used in this work is the well-known Traveling Salesman Problem (TSP) [10].

This present paper is a continuation of that work, and our objective is to reinforce the hypothesis proposed by Osaba et al. in [9]. To achieve this objective, we conduct a new analysis of the results presented in [9], performing a statistical analysis of them. Furthermore, we present a new experimention with two additional routing problems, in order to demonstrate that the hypothesis is also satisfied for them. These problems are the Capatitated Routing Problem (CVRP) [11] and the Vehicle Routing Problem with Backhauls (VRPB) [12].

The rest of the paper is structured as follows. In Section 2 we perform a new statistical analysis of the results presented in [9]. In Section 3 we show the new experimentation conducted. Finally, we finish this work with the conclusions of the study.

II. STATISTICAL ANALYSIS OF THE EXISTING RESULTS

First of all, the hypothesis proposed by Osaba et al., which we try to reinforce, is the following:

"Crossover phase of the genetic algorithms is not efficient for the search process and the capacity of optimization of the technique when it is applied to routing problems using path encoding".

In [9], authors verify this hypothesis, conducting an experimentation with the TSP and different implementations of the classical GA. In that experimentation, six different versions of the GA are used, using different CX functions such as the Order Crossover (OX) [13], Modified Order Crossover (MOX) [14], Half Crossover (HX) [9] and Order Based Crossover (OBX) [15], and different survivors selection functions, such as the 100% elitist or the 50% elitist - 50% random. The results obtained by these six GAs are compared with an Evolutionary Algorithm (EA) which focuses its execution only in the process of mutation and survivors selection. The characteristics of the algorithms used are as follows, listed by their CX function, crossover and mutation probability (p_c and m_p), and survivor selection function:

- GA_{V1}: OX function, p_c= 100%, p_m= 100%, 100% elitist.
- GA_{V2}: MOX function, p_c= 100%, p_m= 100%, 100% elitist.
- GA_{V3}: OBX function, p_c= 100%, p_m= 100%, 100% elitist.
- GA_{V4}: HX function, p_c= 100%, p_m= 100%, 100% elitist.
- GA_{V5}: OX function, p_c= 90%, p_m= 5%, 100% elitist.
- GA_{V6}: OX function, p_c= 90%, p_m= 5%, 50% elitist - 50% random.
- *EA:* No Cx function, p_c= 0%, p_m= 100%, 100% elitist.

Deusto Instute of Technology, University of Deusto,
Av. Universidades, 24, Bilbao, Spain
Email: {e.osaba, roberto.carballedo, fernando.diaz}@deusto.es
{enrique.onieva, perallos}@deusto.es

The mutation function for every approach is the well-known 2-opt [16], which has been very used since its formulation [17–19].

In [9], the results between different approached are compared by their average and the deviation between these averages. In this work, we compare these same results statistically, performing the Student's t-test. This way, we could determine if the results shown are significantly different.

In the Table I, we show the results obtained by Osaba et al. in [9]. We display the total average and the standard deviation for each instance. All the tests were performed on an Intel Core i5 2410 laptop, with 2.30 GHz and a RAM of 4 GB. The number of executions for each instance is 20. Instances were obtained from the TSP Benchmark TSPLIB. The name of each instance has a number that displays the number of nodes it has.

On the other hand, in Table II, we show a direct comparison between EA and each of the other GA versions, using the Student's t-test. This comparison is made for each of the instances used in Table I. The t statistic has the following form:

$$t = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{(n_1-1)SD_1^2+(n_2-1)SD_2^2}{n_1 n_2-2}\frac{n_1+n_2}{n_1 n_2}}}$$

where:

$\overline{X_1}$: Average of EA
SD_1: Standard deviation of EA,
$\overline{X_2}$: Average of the other technique,
SD_2: Standard deviation of the other technique,
n_1: EA size,
n_2: Size of the other technique,

The t value can be positive, neutral, or negative. The positive value of t indicates that the EA is signiffcantly better than the technique with which it is facing. In the opposite case, EA obtains signiffcant worse solutions. If t is neutral, the difference between the two algorithms is not signiffcant. We stated confidence interval at the 95% confidence level ($t_{0.05}$ = 2.021).

A. Analisys of the results

Viewing the results presented in Table I and Table II, the conclusions that can be drawn contribute to those obtained in [9]. In the first four columns of the Table I the results obtained by the different GA versions with the different functions of CX are shown, being all results very similar to each other. This way, we cannot be able to conclude roundly which one gets better results and which is the better. The next two columns show the result obtained by those version of GA with more conventional mutation and crossover probabilities, and in which different functions are used for the survivors selection. The results of these new experiments lead to the same conclusions as the above, there is no perceptible improvement in the results of the different GAs.

In the last column, the results of the EA are shown. If we look at the percentages of the deviations of the means, the vast majority are less than 2%, and rarely exceeds the 2.5%, so that the differences between them are minimal. In this article, we have performed a deeper analysis of these results, applying the Student's t-test between the EA and the GAs. Although the differences between the means are not very wide, Table II shows that in the 61'53% of cases, these differences are significant in favor the EA. On the other hand, in the 17'95% of cases, the results obtained by the EA are significantly worse than one of the exposed GAs. Finally, in 20'52% these differences are not appreciable. The conclusions that can be drawn having made this new analysis reinforce the hypothesis we want to validate. In addition, this new analysis adds credibility to the conclusions mentioned by Osaba et al. in [9], which the most important are the following:

- *Conclusion 1:* The use of CX functions do not give any improvement to the process of optimizing of a GA, being the other functions (mutation and selection of survivors) which provide the ability of optimization to the GA.
- *Conclusion 2:* The crossover phase increases the complexity of the GA algorithm without providing any visible improvement.

III. New experimentation

In this section the new experimentation performed is shown. With it, we will try to validate the hypothesis displayed in Section II, demonstrating that not only holds for the TSP. The new tests are conducted with two different vehicles routing problems, which are the CVRP and the VRPB. Both problems are well-known in combinatorial optimization and they are used in many studies annually [20–23].

For each problem, the performance of an EA is compared with that of 4 different versions of the classic GA. For all algorithms, the population is composed by 48 individuals, which are created randomly. The parents and survival selection criterias are 100% elitist for all experiments, with the exception of both GA_{V4}, which uses a 50% elitist-random criterion for the survival selection. About the ending criteria, the execution of all the algorithms finishes when there are a generation number proportional to the size of the neighborhood (obtained by the mutation operator) without improvements in the best solution found.

The crossover functions used are the *Half Route Crossover* (HRX) and *Half Random route Crossover* (HRRX) [19]. With HRX, first, the 50% of the best routes in one randomly chosen parent are selected and inserted in the child. Then, the nodes already inserted are removed from the other parent. Finally, the remaining nodes are inserted in the same order in the final solution, creating new routes. The HRRX working way is similar to HRX. In this case, in the first step, the routes selected from one of the parents are chosen randomly, instead of selecting the best ones.

Regarding the mutation function, For both techniques the called *Vertex Insertion Routes* is used. This function selects and extracts one random node from a random route. Then, the node is re-inserted in a random position in another randomly selected route. New routes creation is possible with this function.

TABLE I. EXPERIMENTATION WITH THE TSP

Instance	GA_{V1}		GA_{V2}		GA_{V3}		GA_{V4}		GA_{V5}		GA_{V6}		EA	
Name	Avg.	St dev.	Avg.	St dev.	Avg.	St dev.	Avg.	St dev.	Avg.	St dev.	Avg.	St dev.	Avg.	St dev.
Oliver30	425.3	9.86	423.5	5.97	426.1	8.57	427.2	8.99	428.4	8.03	427.0	12.94	423.1	2.81
Eilon50	442.9	7.43	445.9	7.94	445.4	5.87	451.3	7.42	456.2	8.89	457.0	11.58	448.0	8.00
Eil51	448.3	8.69	450.3	6.62	448.7	7.12	451.1	8.57	459.6	19.25	451.7	14.79	449.1	7.49
Berlin52	7945.2	262.71	8062.4	168.4	8027.6	267.1	7988.4	251.72	7912.8	227.81	7843.6	229.64	8023.5	357.31
St70	709.4	13.53	721.4	17.73	718.5	23.98	714.2	12.03	738.2	22.00	720.6	17.00	712.6	13.98
Eilon75	578.2	5.37	582.0	5.85	579.6	16.74	575.9	7.62	590.4	12.83	586.6	15.88	583.4	13.49
Eil76	581.5	12.39	585.5	14.98	577.5	8.73	584.6	14.59	606.1	18.77	588.2	11.14	577.4	8.73
KroA100	22265.7	581.74	22445.3	687.87	22690.0	577.76	22195.8	381.69	44366.1	512.87	22316.7	614.86	21856.9	309.93
KroB100	23602.9	413.89	23599.5	878.01	23548.0	577.06	23366.6	533.15	23252.5	487.89	23026.5	500.14	23194.6	304.66
KroC100	21850.1	465.54	221456.4	241.32	22710.0	875.78	21995.8	593.22	22077.9	851.97	21591.2	724.96	21680.2	447.95
Eil101	693.7	10.45	692.9	11.18	683.0	9.68	696.0	12.35	726.9	26.27	708.4	12.19	689.8	14.83
Pr107	46676.5	1406.5	46076.5	1043.54	46682.2	1105.01	46866.9	964.11	47220.4	1225.64	46481.0	1332.4	45584.0	802.83
Pr124	60852.6	1288.63	61399.0	1263.75	61282.6	1832.66	61002.2	1361.51	59414.4	6941.5	60428.3	911.87	61040.8	1389.97

TABLE II. STUDENT'S t-TEST FOR THE TSP PROBLEM

Instance	EA vs. GA_{V1}	EA vs. GA_{V2}	EA vs. GA_{V3}	EA vs. GA_{V4}	EA vs. GA_{V5}	EA vs. GA_{V6}
Oliver30	+	*	+	+	+	+
Eilon50	-	-	-	+	+	+
Eil51	*	*	*	+	+	+
Berlin52	-	*	*	*	-	-
St70	-	+	+	*	+	+
Eilon75	-	*	-	-	+	*
Eil76	+	+	*	+	+	+
KroA100	+	+	+	+	+	+
KroB100	+	+	+	+	*	-
KroC100	+	+	+	+	+	*
Eil101	+	+	-	+	+	+
Pr107	+	+	+	+	+	+
Pr124	*	+	*	*	-	-

Thus, the characteristics of the algorithms are as follows, listed by their CX function, p_c and m_p, and survivor selection function:

- GA_{V1}*:* HRX function, p_c= 100%, p_m= 100%, 100% elitist.
- GA_{V2}*:* HRRX function, p_c= 100%, p_m= 100%, 100% elitist.
- GA_{V3}*:* HRX function, p_c= 90%, p_m= 5%, 100% elitist.
- GA_{V4}*:* HRX function, p_c= 90%, p_m= 5%, 50% elitist - 50% random.
- *EA:* No Cx function, p_c= 0%, p_m= 100%, 100% elitist.

The tests have been made in the same computer mentioned in Section II. Results are shown in Table III for the CVRP and in Table V for the VRPB. For each instance we display the total average, the standard deviation and the average runtime. The objetive function used in both problems is the total traveled distance. In order to determine if the outcomes are is significantly different, we perform Students t-test in the same way as in Section II. In Table IV and Table VI results of these statistical tests are shown.

Each experiment is repeated 20 times. For the CVRP, the instances were picked from the CVRP set of Christofides and Eilon (*http://neo.lcc.uma.es/vrp* [1]). The name of each CVRP instance has a number that displays the number of nodes it has. For the VRPB we have used 10 instances. The first 6 were obtained from the VRPTW Benchmark of Solomon (*http://neo.lcc.uma.es/vrp*). In this case, the time constraints have been removed, but vehicle capacities and the amount of customer demands are retained. Apart from this, we also have been modified the demands nature with the aim of creating pickup and deliveries. The remaining 4 instances were obtained from the CVRP set of Christofides and Eilon. In these instances, the vehicle capacities and the number of nodes have been maintained, but the demand types have been also changed to have pickups and deliveries.

A. Analisys of the results

The first conclusion that we can extract seeing the results of the Table III and Table V is the following: There is no GA

[1]Last update: January 2013

TABLE III. EXPERIMENTATION WITH THE CVRP

Instance	GA_{V1}			GA_{V2}			GA_{V3}			GA_{V4}			EA		
Name	Avg.	St dev.	Time	Avg.	St dev.	Time	Avg.	St dev.	Time	Avg.	St dev.	Time	Avg.	St dev.	Time
En22k4	**400.5**	11.68	2.76	410.1	20.81	3.10	413.9	14.23	2.91	403.9	24.69	3.81	413.7	14.78	1.05
En23k3	647.0	45.21	2.17	650.9	29.23	2.83	665.3	18.54	2.66	645.9	37.98	3.39	**645.5**	28.13	1.27
En30k3	598.6	26.43	3.58	590.4	25.06	3.81	597.1	15.52	5.34	**579.6**	29.32	6.82	588.9	36.32	1.48
En33k4	922.1	24.03	5.88	944.2	47.42	5.71	942.5	29.57	11.72	**916.8**	35.45	8.48	921.0	25.29	2.85
En51k5	661.6	35.85	10.44	675.4	68.13	10.77	679.2	31.23	24.76	681.0	39.15	20.94	**652.8**	42.64	3.09
En76k7	843.0	65.47	24.85	867.5	63.25	30.95	880.1	26.17	67.01	897.0	33.70	54.25	**834.4**	22.41	6.51
En76k8	**882.8**	39.52	29.52	926.6	56.12	30.95	977.3	28.24	55.20	968.1	61.11	51.36	898.5	40.49	9.06
En76k10	1014.7	42.48	28.57	1026.4	44.21	30.45	1031.2	27.88	55.20	1032.6	26.56	54.52	**999.8**	46.42	6.14
En76k14	**1176.3**	38.99	26.63	1185.6	39.05	31.25	1206.1	11.15	46.44	1212.4	29.14	42.52	1191.5	40.43	7.31
En101k8	1057.3	71.89	67.54	1021.6	44.84	65.24	1131.4	37.65	134.21	1104.6	83.84	130.25	**985.7**	57.02	13.57
En101k14	**1261.9**	27.45	78.54	1274.0	36.41	81.25	1322.2	62.35	134.41	1277.8	84.23	132.54	1276.9	52.47	15.32

TABLE IV. STUDENT'S t-TEST FOR THE CVRP PROBLEM

Instance	EA vs. GA_{V1}	EA vs. GA_{V2}	EA vs. GA_{V3}	EA vs. GA_{V4}
En22k4	-	-	*	-
En23k3	*	*	+	*
En30k3	+	*	+	-
En33k4	*	+	+	*
En51k5	+	+	+	+
En76k7	*	+	+	+
En76k8	-	+	+	+
En76k10	+	+	+	+
En76k14	-	*	+	+
En101k8	+	+	+	+
En101k14	-	*	+	*

version that provides a visible advantage over EA results. In addition, if we look at Tables IV and VI we can conclude that no technique gives significantly better results than the EA. For the CVRP, in the 59'1% of cases, the differences are significant in favor the EA. On the other hand, in the 15'9% of cases, the results obtained by the EA are significantly worse than one of the GAs. Finally, in 25% these differences are not appreciable. For the VRPB, these percentages are, respectively, 62'5%, 5%, and 32'5%.

In addition, regarding the run times, these are much lower for the EA than for the other techniques. This difference is not very noticeable in problems with few nodes, But when the nodes increases, the differences become larger. This happens because routing problems are classified as NP-Hard problems [24].

These results support the Conclusions 1 and 2 presented in Section II-A. Furthermore, we can add one more important finding:

- *Conclusion 3:* Using CX functions increases considerably the execution time. Furthermore, as the number of nodes grows, the time rises exponentially. This is especially important for real-time applications, where the execution time becomes decisive.

Conclusions 2 and 3 can be obtained easily, since the more steps a meta-heuristic has, more time needs for execution and more complex is to design and develop. On the other hand, Conclusion 1, shown in Section II-A, can be based on several arguments. The main purpose of the CX functions is to obtain new individuals making combinations of the characteristics of the individuals of the population. Thus, it is logical to think that it is very improbable that the resulting offspring from neutral crosses improve their parents. It is for this reason that the main utility of CX might be to increase the capability of the search process of the algorithm. The use of this kind of functions or mechanisms can be very beneficial for a meta-heuristic, as can be seen in other techniques such as simulated annealing [25], which uses a cooling process to allow this type of jumps inside the space of solutions, or the tabu search [26], with the mechanisms of memory at medium and long term. Despite this, the use of this type of functions must not be excessive, since the jumps in the solution space are beneficial only in determined times, for example, to avoid local optima.

In GA, mutations can handle this type of jumps in the solution space. In addition the mutation can also make small jumps on the solution space, which are very positive for the optimization process. This fact can free the GA of using CX functions, and it can reduce its running time, achieving similar results, as we can see in the experimentation conducted.

With all this, we can say that the hypothesis has been verified for two additional problems, the CVRP and the VRPB.

TABLE V. EXPERIMENTATION WITH THE VRPB

Instance	GA_{V1}			GA_{V2}			GA_{V3}			GA_{V4}			EA		
Name	Avg.	St dev.	Time	Avg.	St dev.	Time	Avg.	St dev.	Time	Avg.	St dev.	Time	Avg.	St dev.	Time
C101	775.8	47.21	15.75	**769.7**	46.88	20.18	813.0	58.17	30.38	813.4	53.46	28.22	776.4	51.53	5.21
C201	903.9	125.77	4.60	934.7	66.37	3.77	924.7	117.12	7.54	942.3	136.62	4.60	**902.1**	101.21	0.91
R101	984.2	25.24	18.51	**979.2**	52.36	24.58	1085.3	56.76	39.37	1060.8	66.32	33.92	987.5	53.65	6.24
R201	1170.6	61.95	37.95	1231.2	63.11	42.15	1358.0	123.31	42.53	1386.4	54.06	52.50	**1168.4**	41.13	6.24
RC101	668.0	82.34	3.49	669.4	54.63	4.05	712.3	84.76	6.54	691.8	33.09	6.87	**631.5**	46.31	1.10
RC201	**1304.9**	90.09	33.28	1364.6	104.22	44.62	1516.4	90.91	40.25	1485.2	113.79	44.51	1310.4	52.29	5.08
En30k4	606.0	44.75	1.69	620.9	82.29	1.74	627.7	48.69	2.98	**594.7**	99.26	3.16	626.9	58.79	0.62
En33k4	844.4	28.46	2.50	855.3	28.31	2.42	860.5	37.16	4.53	847.6	30.62	5.76	**841.7**	42.96	0.87
En51k5	**696.4**	46.82	5.27	713.3	49.37	3.83	754.0	56.75	13.36	725.6	38.13	13.14	701.7	34.41	2.12
En76k8	**915.8**	38.43	17.56	987.7	81.80	15.64	1027.4	66.55	25.53	1009.0	46.59	27.33	917.7	62.35	3.27

TABLE VI. STUDENT'S t-TEST FOR THE VRPB PROBLEM

Instance	EA vs. GA_{V1}	EA vs. GA_{V2}	EA vs. GA_{V3}	EA vs. GA_{V4}
C101	*	*	+	+
C201	*	+	+	+
R101	*	*	+	+
R201	*	+	+	+
RC101	+	+	+	+
RC201	*	+	+	+
En30k4	-	*	*	-
En33k4	*	+	+	*
En51k5	*	+	+	+
En76k8	*	+	+	+

Finally, as a last conclusion we could enter the next reasoning which is supported by the results shown in the experiments.

- *Conclusion 4:* The most efficient way to implement an evolutionary algorithm to solve routing problems is basing it only on the functions of mutation and selection of survivors.

B. Conclusions

The aim of this paper is to extend the study made by Osaba et al. in [9], which analizes efficacy of using blind crossover operators in genetic algorithms for solving routing problems. To do this, we have performed a more deeper study of the results presented in that work, by applying a statistical study of the existing results. Apart from this, we have conducted a new experimentation with two different routing problems, the Capacitated Vehicle Routing Problem and the Vehicle Routing Problem with Backhauls, both widely used in the literature.

The conclusions drawn from the new experimentation reinforce the hypothesis proposed by Osaba et al. in their work, and it can be ensured that not only holds for the TSP, but also for the CVRP and VRPB. This hypothesis says that the crossover phase of the genetic algorithms is not efficient for the search process and the capacity of optimization it is applied to routing problems.

REFERENCES

[1] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1975.

[2] K. De Jong, "Analysis of the behavior of a class of genetic adaptive systems," Ph.D. dissertation, University of Michigan, Michigan, USA, 1975.

[3] D. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Professional, 1989.

[4] I. Harvey, "The microbial genetic algorithm," *Advances in Artificial Life. Darwin Meets von Neumann*, vol. 5778, pp. 126–133, 2011.

[5] M. Affenzeller, S. Wagner, and S. Winkler, *Genetic algorithms and genetic programming: modern concepts and practical applications*. Chapman & Hall/CRC, 2009, vol. 6.

[6] J. Gao, M. Gen, L. Sun, and X. Zhao, "A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems," *Computers & Industrial Engineering*, vol. 53, no. 1, pp. 149–162, 2007.

[7] I. Moon, J.-H. Lee, and J. Seong, "Vehicle routing problem with time windows considering overtime and outsourcing vehicles," *Expert Systems with Applications*, vol. 39, no. 18, pp. 13 202–13 213, 2012.

[8] M. Martínez-Torres, "A genetic search of patterns of behaviour in oss communities," *Expert Systems with Applications*, vol. 39, no. 18, pp. 13 182–13 192, 2012.

[9] E. Osaba, R. Carballedo, F. Diaz, and A. Perallos, "Analysis of the suitability of using blind crossover operators in genetic algorithms for solving routing problems," in *Proceedings of the 8th International Symposium on Applied Computational Intelligence and Informatics*. IEEE, 2013, pp. 17–23.

[10] E. Lawler, J. Lenstra, A. Kan, and D. Shmoys, *The traveling salesman problem: a guided tour of combinatorial optimization*. Wiley New York, 1985, vol. 3.

[11] G. Laporte, "The vehicle routing problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, no. 3, pp. 345–358, 1992.

[12] B. Golden, E. Baker, J. Alfaro, and J. Schaffer, "The vehicle routing problem with backhauling: two approaches," in *Proceedings of the Twenty-first Annual Meeting of SE TIMS*. South Carolina USA, 1985, pp. 90–92.

[13] L. Davis, "Applying adaptive algorithms to epistatic domains," in *Proceedings of the international joint conference on artificial intelligence*, vol. 1, 1985, pp. 161–163.

[14] C. Wang, J. Zhang, J. Yang, C. Hu, and J. Liu, "A modified particle swarm optimization algorithm and its application for solving traveling salesman problem," in *Proceedings of the International Conference on Neural Networks and Brain*, vol. 2. IEEE, 2005, pp. 689–694.

[15] M. Albayrak and N. Allahverdi, "Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms," *Expert Systems with Applications*, vol. 38, no. 3, pp. 1313–1320, 2011.

[16] S. Lin, "Computer solutions of the traveling salesman problem," *Bell System Technical Journal*, vol. 44, no. 10, pp. 2245–2269, 1965.

[17] C. Tarantilis and C. Kiranoudis, "A flexible adaptive memory-based algorithm for real-life transportation operations: Two case studies from dairy and construction sector," *European Journal of Operational Research*, vol. 179, no. 3, pp. 806–822, 2007.

[18] N. Bianchessi and G. Righini, "Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery," *Computers & Operations Research*, vol. 34, no. 2, pp. 578–594, 2007.

[19] E. Osaba, E. Onieva, R. Carballedo, F. Diaz, and A. Perallos, "An adaptive multi-crossover population algorithm for solving routing problems," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2013)*. Springer, 2014, pp. 113–124.

[20] G. Mattos Ribeiro and G. Laporte, "An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem," *Computers & Operations Research*, vol. 39, no. 3, pp. 728–735, 2012.

[21] S. Ngueveu, C. Prins, and R. Wolfler Calvo, "An effective memetic algorithm for the cumulative capacitated vehicle routing problem," *Computers & Operations Research*, vol. 37, no. 11, pp. 1877–1885, 2010.

[22] Y. Gajpal and P. Abad, "Multi-ant colony system (macs) for a vehicle routing problem with backhauls," *European Journal of Operational Research*, vol. 196, no. 1, pp. 102–117, 2009.

[23] S. Anbuudayasankar, K. Ganesh, S. Lenny Koh, and Y. Ducq, "Modified savings heuristics and genetic algorithm for bi-objective vehicle routing problem with forced backhauls," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2296–2305, 2012.

[24] J. K. Lenstra and A. Kan, "Complexity of vehicle routing and scheduling problems," *Networks*, vol. 11, no. 2, pp. 221–227, 1981.

[25] P. J. Van Laarhoven and E. H. Aarts, *Simulated annealing*. Springer, 1987.

[26] F. Glover, M. Laguna *et al.*, *Tabu search*. Springer, 1997, vol. 22.