J Zhejiang U-Sci C manuscript No.
(will be inserted by the editor)

A Multi-Crossover and Adaptive Island Based Population Algorithm for Solving Routing Problems

E. Osaba · E. Onieva · R. Carballedo · F. Diaz · A. Perallos · X. Zhang

The final publication is available at www.springerlink.com. DOI: 10.1631/jzus.C1300184

Abstract In this letter, a new multi-crossover and adaptive island-based population algorithm (MAIPA) is proposed. This new technique divides the entire population into subpopulations, or demes, each of them with a different crossover function, which can be switched depending on their efficiency. In addition, the presented MAIPA reverses the conventional genetic algorithms philosophy. It gives priority to the autonomous improvement of the individuals (mutation phase), and introduces dynamism in the crossover probability. In the proposed MAIPA, each subpopulation begins with a very low value of crossover probability, and varies depending on two factors: the current generation number, and the search performance on recent generations. This mechanism helps to prevent premature convergence. In this first phase of the research, the quality of this technique is tested, applying it to three different well-known routing problems, and the results are compared with the ones obtained by a traditional island based genetic algorithm. The new proposal proves to be better for all problems used.

Keywords Island Model · Adaptive Algorithm · Combinatorial Optimization · Vehicle Routing problems · Intelligent Transportation Systems.

1 Introduction

Genetic algorithm (GA) is one of the most used and successful meta-heuristic to solve combinatorial optimization problems. Although its basic principles were proposed in 1975 by Holland (Holland (1992)), it was later when its

Deusto Institute of Technology (DeustoTech), University of Deusto
Av. Universidades 24, Bilbao 48007, Spain
Tel.: +34 944 139003 (ext. 2050)
E-mail: [e.osaba, enrique.onieva, roberto.carballedo, fernando.diaz, perallos, xiao.zhang] @deusto.es

practical use to solve complex problems was shown (Goldberg (1989); De Jong (1975)). Since then, GAs have been the focus of a large number of research, being applied to a wide range of fields (Moon et al. (2012); Martínez-Torres (2012)). Despite this fact, GAs have some drawbacks, such as fast convergence and the imbalance between exploration and exploitation. In order to overcome these drawbacks parallel genetic algorithms (PGA) were proposed (Whitley et al. (1999)). Analyzing the literature, PGAs can be divided into three categories: fine grain, panmitic model and island model. This last category is the most used, and it consists in multiple populations that evolve separately (most of the time) and exchange individuals occasionally. In the literature there are a lot of studies describing the main issues about this type of GAs. In Cantú-Paz (1998) can be found a comprehensive survey about PGAs.

In this letter, a multi-crossover and adaptive island-based population algorithm (MAIPA) for solving routing problems is presented. This new meta-heuristic is a variant of the classic island based GA (IGA). The proposed MAIPA divides the whole population into different subpopulations, or demes, each of them with its own crossover function and crossover probability. The migration system topology is dynamic, and each subpopulation can communicate with the others depending on the search process. In addition, the introduced technique gives priority to the local improvement of the individuals (mutation), by applying crossover operators only when they are beneficial. In the presented MAIPA, the crossover probability of each subpopulation can vary, depending on the current generation number, and the search performance on recent generations in the deme. This dynamism, accompanied by the dynamic topology (communications between subpopulations are evenly) and the multi-crossover, increases the exploration and exploitation capacity of the meta-heuristic, and helps to prevent premature convergence.

The aim of this work is to introduce this new technique, and show that it is a good alternative to solve routing problems. For this, the results obtained by the technique applied to three different routing problems have been compared with those obtained by an IGA. The objective is to demonstrate that the proposed meta-heuristic outperforms the classic IGA in terms of results quality and runtime.

The rest of the letter is structured as follows. In Section 2 a brief literature is introduced. In that section, the original contributions of the proposed MAIPA are also mentioned. In Section 3 the proposed technique is described. The experimentation is introduced in Section 4. The letter finishes with the conclusions and future work.

2 Brief literature and contribution of the presented work

As it is said in the introduction, the proposed technique is a multi-population algorithm. Some approaches have been proposed throughout the history following this same philosophy, as the parallel artificial bee colony (Tsai et al.

(2009)), parallel particle swarm optimization (Niu et al. (2007)), and the parallel genetic algorithms.

Besides this, as already explained, the meta-heuristic algorithm adapts the crossover probability depending on the performance of the algorithm. The idea of adapting the mutation and crossover probabilities (p_m and p_c) of a GA has been studied since long time ago (Schaffer & Morishima (1987)), with the aim of improving the performance of conventional genetic algorithms. Anyway, this field is subject of many studies nowadays, as in Wang & Tang (2011). About the multi-crossover feature, it has also been studied long time ago and nowadays (Spears (1995); Mukherjee et al. (2012)).

Being a variation of the IGA, the differences between the proposed technique and other multi-population techniques are the same as the IGA, and they can be found in Cantú-Paz (1998). Respect to IGAs and other adaptive techniques, the innovative aspects of the proposed MAIPA are as follows: (1) unlike the vast majority of PGA, in the presented approach each subpopulation has a different crossover function and p_c. This fact helps individuals to explore the solution space differently when they migrate to another deme. This characteristic increases the exploration capability of the technique. (2) The proposed MAIPA changes the philosophy of the conventional IGAs and GAs. It begins the execution with a very low or null value for p_c, and a high value of p_m. As shown in Osaba et al. (2013), this fact increases the exploitation capacity of the search. (3) The introduced MAIPA adapts the p_c of each deme depending on the search performance in recent iterations (i.e., if the best found solution is improved) and the current generation number, rather than using only the population fitness, as most previous studies. (4) The introduced technique combines the adaptation of the crossover probability, with a multi-population and multi-crossover system. After reviewing the literature, it can be said that this is a new approach that has not been done before. (5) The proposed MAIPA has been tested with routing problems. Usually, adaptive and multi-population techniques have not been applied to this family of well-known problems.

3 A Multi-Crossover and Adaptive Island Based Population Algorithm for Solving Routing Problems

As stated in Section 2, the proposed MAIPA is a variant of a conventional island based GA. In Algorithm 1 can be seen how the meta-heuristic works. The proposed technique gives priority to the local improvement of the individuals, provided by the mutation phase, and gives less importance to the crossovers phase. This fundament is based on the recently published work Osaba et al. (2013), which analyzes the inefficiency of the crossover phase in the optimization capacity of a basic GA, when it is used to solve routing problems. This is the reason why the proposed MAIPA gives a greater importance to the mutation phase. Spite of this, it can be considered that crossovers can be beneficial to the exploration capacity, maintaining the diversity of the population. Therefore, in the proposed MAIPA crossovers are executed only when they can be beneficial, adapting the p_c of each subpopulation to the search needs. Besides this, in the proposed MAIPA each subpopulation has its own crossover function, which can change depending on its performance. This feature increases the exploration capacity of the meta-heuristic.

Algorithm 1: Pseudocode of the proposed MAIPA

Creation of the whole population;
Subpopulations creation and crossover function assignment;
while *Termination criterion not reached* **do**
    for *each subpopulation* **do**
        Mutation process;
        Crossover process;
        Selection of survivors;
        p_c update;
    end
    Individual migration process;
end
Return of the best individual of the whole system;

Regarding the p_m and p_c parameters of the proposed technique, in the presented MAIPA all the subpopulations have a p_m equals to 100%, which means that all individuals of the whole system go through the mutation process at every generation. Furthermore, each deme has its own p_c, which start with a value close to 0%. The latter parameter is modified differently in each subpopulation, increasing or restarting its value. That modification is based on the following criteria, being $best_i^{sp}$ the best solution in subpopulation sp in the generation i:

- If $best_i^{sp}$ is better than $best_{i-1}^{sp}$: This means that the search process evolves correctly. In this case, the value of p_c is restarted, since it could be considered that it is not necessary to diversify the population.
- If $best_i^{sp} = best_{i-1}^{sp}$: In this case, it might be considered that the search process is trapped in a local optima, or that the population is concentrating in the same region of the space of solutions. For this reason, p_c is increased, trying to increase the subpopulations diversification using crossover operators.
- It must be taken into account that $best_i^{sp}$ never will be worse than $best_{i-1}^{sp}$, since the best solution of each population is always maintained throughout the generations.

This way, whenever $best_i^{sp}$ of a subpopulation has not been improved in the last generation, p_c of this deme increases following the Equation (1), where N represents the number of generations without improvements, NG the total amount of generations so far, and NMF depicts the size of the mutation operator neighborhood:

$$p_c = p_c + \frac{N^2}{NMF^2} + \frac{NG}{NMF^2} \tag{1}$$

As seen in the above formula, p_c increases proportionally to the total number of generations (NG) and the number of generations without any improvement in the best solution (N).

In relation to the multi-crossover feature, at the beginning one crossover function is randomly assigned to each subpopulation. Then, throughout the execution, these functions could be randomly replaced by another, allowing repetitions. For this, a maximum value for p_c is defined ($Max(p_c)$). This value is the same in all the demes. If over the generations the $Max(p_c)$ of any subpopulation is exceeded, the crossover function of this deme is randomly replaced, restarting p_c to its original value. This feature helps to increase the population diversification in a better way than other similar techniques.

Regarding the migration system of the proposed MAIPA, as already said in Section 1, the topology of the technique is dynamic. This means that each subpopulation will communicate with all other demes depending on the performance of the search. The communication is made as follows: whenever a deme improves its $best(sp_i)$, it shares its new best solution with all the other subpopulations. These communications help to make a greater exploration of the solution space.

4 Experimentation

In this section, results obtained by the proposed MAIPA applied to three different routing problems are presented. As has been mentioned, the outcomes got by the proposed MAIPA will be compared with the ones obtained by an IGA. For both meta-heuristics similar parameters and functions have been used. Thus, the difference between them is only their working way. This is the most reliable method to determine which meta-heuristic obtains better results. The problems used in this study are the Traveling Salesman Problem (TSP) (Lawler et al. (1985)), the Capacitated Vehicle Routing Problem (CVRP) (Laporte (1992)), and the Vehicle Routing Problem with Backhauls (VRPB) (Golden et al. (1985)). TSP is used since it is a well-known benchmarking problem, which it is simple to implement and understand. Today, there are many studies using the TSP (Bae & Rathinam (2012); Sarin et al. (2011)). In addition, CVRP and VRPB are used because they are two of the most used routing problem in the literature. Annually, a large number of studies use these two problems for their experimentations (Anbuudayasankar et al. (2012); Mattos Ribeiro & Laporte (2012); Ngueveu et al. (2010)), due to their complexity and, above all, to their applicability to real scenarios. These three problems are easily replicable, so that any reader can perform this same experiment, either to check the results, or to compare them with results obtained by other techniques.

Regarding the parameters of the algorithms, for both alternatives, and all the problems, the initial population is composed by 48 randomly created individuals, which are randomly divided in four different subpopulations of 12 individuals each. All the individuals are encoded using the *Path Encoding* (Larranaga et al. (1999)). Regarding the selection and survivor phases, the same function is used for both in all instances, which is the 50% elitist - 50% random. About the ending criteria, the execution of all the algorithms finishes in 20.000 generations. For the IGA, p_m and p_c are, respectively, 5% and 95%. In the case of the proposed MAIPA, p_c starts at 0%. When the best solution found is not improved, the p_c increases following Equation (1), shown in Section 3, otherwise, it returns to 0%. $Max(p_c)$ value is 35%.

Regarding the TSP, crossover functions implemented for this study are order crossover (Davis (1985)), half crossover (Osaba et al. (2013)), modified order crossover (Ray et al. (2004)), and order based crossover (Syswerda (1991)). For IGA, intending to make a fairer comparison, the same functions are used, assigning one of them to each subpopulation. Yet, unlike the proposed MAIPA, subpopulations not change their function during the execution. The mutation function for all both techniques is the well-known 2-opt (Lin (1965)). Migration system for the IGA is also the same as the used in MAIPA.

For the CVRP and VRPB, crossover functions implemented are the Half Route Crossover and the Half Random Route Crossover. The operation way of the first of them is the following: first of all, half of the routes (the shortest ones) of one of the parents are inserted in the child. After that, the nodes already selected are removed from the other parent, and the remaining nodes are inserted in the child in the same order, creating new routes. Half Random Route Crossover works similar as Half Route Crossover. In this case, the routes selected in the first step of the process are selected randomly, instead of choosing the best ones. Regarding the mutation function, the named vertex insertion function is used for both problems. This operator selects one random node from one randomly chosen route of the solution. This node is extracted, and inserted in another randomly selected route. With this function, the creation of new routes is possible.

4.1 Results

The experimentation has been performed on an Intel Core i5 2410 laptop, with 2.30 GHz and 4 GB of RAM. The results for the TSP, CVRP, and VRPB are shown in Table 1, Table 2, and Table 3, respectively. All instances of the TSP have been obtained from the TSPLIB Benchmark (Reinelt (1991)). For the CVRP, instances have been picked from the Christofiden and Eilon CVRP benchmark (*http://neo.lcc.uma.es/vrp* [1]). For the VRPB, 11 instances have been used. The first six have been obtained from the Benchmark of Solomon (*http://neo.lcc.uma.es/vrp*), and the remaining five from the Christofides and

[1] Last update: January 2013

Eilon CVRP benchmark. These instances are not typical of the VRPB, for this reason, in order to adapt them to the characteristics of the VRPB, demand types have been changed to have pick-ups and deliveries. This change is the reason why the optimums are not shown in Table 3.

For each run, the total average, best result, standard deviation, and average runtime (in seconds) are displayed. Each experiment is repeated 20 times. In addition, the well-known Students t-test is performed for every instance, with the aim of determining if the outcomes of the proposed MAIPA are significantly different than those obtained by IGA. 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 $\overline{X_i}$, SD_i and n_i, are the average, standard deviation and number of executions of each technique, being our MAIPA $i = 1$ and IGA $i = 2$. This way, The t values shown can be positive (+) , neutral (*), or negative (-). Positive value of t indicates that the proposed MAIPA is significantly better than IGA. If t is negative, IGA gets better solutions. Finally, if t is neutral, the difference between the two algorithms is not significant. The confidence interval has been stated at the 95% confidence level ($t_{0.05} = 2.021$).

Instance		Proposed MAIPA				IGA				t test
Name	Optima	Avg.	S. dev.	Best	Time	Avg.	S. dev.	Best	Time	t
Oliver30	420	**424.6**	6.4	420	0.10	431.0	11.4	420	0.24	+
Eilon50	425	**445.7**	8.5	434	0.32	451.9	12.8	427	0.97	+
Eil51	426	**446.0**	9.3	431	0.35	460.2	11.9	438	0.99	+
Berlin52	7542	**8004.4**	286.1	7542	0.30	8113.4	170.1	7926	1.03	+
St70	675	**714.7**	16.1	687	0.80	726.7	16.4	695	4.51	+
Eilon75	535	**570.8**	10.6	556	0.94	580.6	15.0	556	5.54	+
Eil76	538	**574.1**	12.5	553	1.01	585.8	18.3	563	6.55	+
KroA100	21282	**22349.1**	600.1	21319	2.08	22955.8	671.3	21972	15.95	+
KroB100	22140	**23350.9**	421.1	22413	2.22	23764.3	720.3	22248	16.82	+
KroC100	20749	**22133.0**	531.5	21405	1.65	22533.7	727.9	21454	17.56	+
KroD100	21294	**22281.4**	414.3	21464	1.98	22436.9	402.3	21836	18.53	+
KroE100	22068	**23397.3**	560.6	22535	1.96	23945.1	628.5	23146	17.24	+
Eil101	629	**680.8**	8.5	665	2.29	713.1	15.4	697	25.57	+
Pr107	44303	**46270.5**	975.2	44764	2.73	47664.9	1316.3	45705	32.45	+
Pr124	59030	**60995.3**	630.8	60077	3.66	63654.0	2605.1	59697	48.54	+
Pr136	96772	**102498.3**	2161.5	97759	5.78	106363.2	1562.9	104565	56.41	+
Pr144	58537	**60969.2**	1663.6	58599	5.93	63562.6	1557.6	61398	62.54	+
Pr152	73682	**76631.3**	1013.9	74745	7.30	79209.5	2622.4	76252	67.54	+

Table 1 Results of the proposed MAIPA and IGA for the TSP

4.2 Analysis of the results

Analyzing the results, some conclusions can be highlighted, being the most important that the proposed technique outperforms the classic IGA in terms

Instance		Proposed MAIPA				IGA				t test
Name	Optima	Avg.	S. dev.	Best	Time	Avg.	S. dev.	Best	Time	t
En22k4	375	**391.1**	8.54	375	1.59	401.0	15.30	375	3.80	+
En23k3	569	**599.9**	31.75	571	2.05	656.2	23.44	601	3.14	+
En30k3	534	**557.7**	16.00	544	2.16	561.6	19.41	542	4.23	+
En33k4	835	**899.5**	23.30	864	2.85	911.0	27.18	888	6.52	+
En51k5	521	**619.3**	45.15	561	4.15	628.4	31.65	572	25.10	+
En76k7	682	**799.9**	37.34	752	8.83	814.1	31.43	764	64.12	+
En76k8	735	**860.5**	21.72	829	9.27	881.3	32.40	834	66.54	+
En76k10	830	**963.0**	19.88	935	7.89	971.0	24.54	945	55.10	+
En76k14	1021	**1179.1**	33.14	1139	9.54	1183.1	52.70	1142	41.16	*
En101k8	815	**997.2**	46.92	919	14.14	1004.5	67.43	924	52.43	*
En101k14	1071	**1221.7**	34.49	1173	12.15	1246.7	56.19	1182	114.55	+

Table 2 Results of the proposed MAIPA and IGA for the CVRP

Instance	Proposed MAIPA				IGA				t test
Name	Avg.	S. dev.	Best	Time	Avg.	S. dev.	Best	Time	t
C101	**712.4**	69.53	635	6.15	739.9	56.48	656	30.91	+
C201	**740.1**	47.02	685	5.45	785.4	69.54	682	22.41	+
R101	**933.9**	22.22	901	4.87	1003.8	33.47	953	27.76	+
R201	**1103.2**	59.35	1046	6.53	1217.0	60.11	1105	54.83	+
RC101	**614.8**	36.60	544	3.16	656.0	51.26	573	7.52	+
RC201	**1215.4**	85.13	1133	10.79	1315.5	67.08	1253	52.15	+
En30k4	**597.3**	67.90	522	3.95	602.8	37.68	544	6.34	*
En33k4	**821.3**	35.62	784	3.47	835.4	44.76	746	5.80	+
En51k5	**657.4**	35.54	619	4.63	689.6	37.61	642	12.15	+
En76k8	**908.1**	56.36	831	7.13	942.1	39.21	883	24.31	+
En101k8	**1131.3**	57.00	1070	8.64	1187.5	63.21	1090	53.12	+

Table 3 Results of the proposed MAIPA and IGA for the VRPB

of solution quality and runtime. Overall, both techniques have been applied to 40 different instances, and the introduced MAIPA offers better solutions and runtimes in 100% of the cases. Besides this, thanks to the Student's t test, it can be added that these improvements in the solutions quality are significant in 92.5% (37 out of 40) of the instances. Being more specific, for the TSP, CVRP and VRPB, this significant improvement is given in the 100% (18 out of 18), 81.81% (nine out of 11) and 90.91% (10 out of 11) of the cases, respectively.

The reason because the introduced MAIPA requires lower execution time is the following: comparing the working way of the crossover and mutation operators, the first are complex operations (specially for routing problems, where all constraints have to be met) in which two individuals combine their characteristics. On the other hand, a mutation is a small modification of a chromosome, and requires considerably less time than the previous ones. The fact that the MAIPA makes fewer crossovers than the IGA is reflected in the runtime, providing an advantage to the proposed meta-heuristic.

The reason for the results improvement of the proposed technique can also be explained, and it is based in the recent published work Osaba et al. (2013). The use of crossover operators helps a broad exploration of the solution space, since they are very useful resources to make jumps inside it. However, these operators do not contribute to make a deep search of promising regions. To carry out a more exhaustive search, it is necessary to use a function that takes care of local improvement of the individuals. Mutation function can handle this objective.

This way, the introduced MAIPA can perform an intense search in the promising regions of the solution space using the mutation process. In addition, it uses crossovers when the diversity of the population is decreasing, in order to avoid local optimums. Using crossovers, subpopulation can be expanded through the entire solution space easily, and it could be more probable to find promising regions. Besides this, the multi-crossover feature enhanced this diversification, allowing a broader exploration.

In conclusion, using the IGA basic structure, the search conducted comprises a large area of the space of solutions, but it has a small intensification capacity. This means that, finally, the IGA can not get results as good as the proposed MAIPA.

5 Conclusions and further work

In this letter, a new multi-crossover and adaptive island-based population algorithm (MAIPA) is presented. It is a variation of the conventional Island based genetic algorithm (IGA). To check the quality of the proposed technique, it has been applied to three different well-known routing problems, and its outcomes have been compared with the ones obtained by a basic IGA. As a conclusion, it can be confirmed that the introduced MAIPA improves the IGA, in terms of solution quality and runtimes, being a good alternative to solve routing problems. Finally, the reasons why the proposed MAIPA obtains better results than the IGA have been explained.

Some small improvements are planned for later versions of the technique. For this reason, in the near future, these improvements will be developed and tested. Besides this, the presented technique will be applied to real life routing problems. At this time, its application to a dynamic distribution system of car windscreen repairs is planned.

References

Anbuudayasankar, S., Ganesh, K., Lenny Koh, S., & Ducq, Y. (2012). Modified savings heuristics and genetic algorithm for bi-objective vehicle routing problem with forced backhauls. *Expert Systems with Applications*, *39*, 2296–2305.

Bae, J., & Rathinam, S. (2012). Approximation algorithms for multiple terminal, hamiltonian path problems. *Optimization Letters*, *6*, 69–85.

Cantú-Paz, E. (1998). A survey of parallel genetic algorithms. *Calculateurs paralleles, reseaux et systems repartis*, *10*, 141–171.

Davis, L. (1985). Applying adaptive algorithms to epistatic domains. In *Proceedings of the international joint conference on artificial intelligence* (pp. 161–163). volume 1.

De Jong, K. (1975). *Analysis of the behavior of a class of genetic adaptive systems*. Ph.D. thesis University of Michigan, Michigan, USA.

Goldberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Professional.

Golden, B., Baker, E., Alfaro, J., & Schaffer, J. (1985). The vehicle routing problem with backhauling: two approaches. In *Proceedings of the Twenty-first Annual Meeting of SE TIMS* (pp. 90–92). South Carolina USA.

Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, *59*, 345–358.

Larranaga, P., Kuijpers, C. M. H., Murga, R. H., Inza, I., & Dizdarevic, S. (1999). Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review*, *13*, 129–170.

Lawler, E., Lenstra, J., Kan, A., & Shmoys, D. (1985). *The traveling salesman problem: a guided tour of combinatorial optimization* volume 3. Wiley New York.

Lin, S. (1965). Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, *44*, 2245–2269.

Martínez-Torres, M. (2012). A genetic search of patterns of behaviour in oss communities. *Expert Systems with Applications*, *39*, 13182–13192.

Mattos Ribeiro, G., & Laporte, G. (2012). An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, *39*, 728–735.

Moon, I., Lee, J.-H., & Seong, J. (2012). Vehicle routing problem with time windows considering overtime and outsourcing vehicles. *Expert Syst Appl*, *39*, 13202–13213.

Mukherjee, S., Ganguly, S., & Das, S. (2012). A strategy adaptive genetic algorithm for solving the travelling salesman problem. In *Swarm, Evolutionary, and Memetic Computing* (pp. 778–784). Springer.

Ngueveu, S., Prins, C., & Wolfler Calvo, R. (2010). An effective memetic algorithm for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, *37*, 1877–1885.

Niu, B., Zhu, Y., He, X., & Wu, H. (2007). Mcpso: A multi-swarm cooperative particle swarm optimizer. *Applied Mathematics and Computation*, *185*, 1050–1062.

Osaba, E., Carballedo, R., Diaz, F., & Perallos, A. (2013). 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* (pp. 17–23).

Ray, S., Bandyopadhyay, S., & Pal, S. (2004). New operators of genetic algorithms for traveling salesman problem. In *Proceedings of the 17th International Conference on Pattern Recognition* (pp. 497–500). volume 2.

Reinelt, G. (1991). Tsplib: A traveling salesman problem library. *ORSA journal on computing*, *3*, 376–384.

Sarin, S. C., Sherali, H. D., & Yao, L. (2011). New formulation for the high multiplicity asymmetric traveling salesman problem with application to the chesapeake problem. *Optimization Letters*, *5*, 259–272.

Schaffer, J. D., & Morishima, A. (1987). An adaptive crossover distribution mechanism for genetic algorithms. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application* (pp. 36–40). L. Erlbaum Associates Inc.

Spears, W. M. (1995). Adapting crossover in evolutionary algorithms. In *Proceedings of the Conference on Evolutionary Programming* (pp. 367–384).

Syswerda, G. (1991). Schedule optimization using genetic algorithms. *Handbook of genetic algorithms*, (pp. 332–349).

Tsai, P.-W., Pan, J.-S., Liao, B.-Y., & Chu, S.-C. (2009). Enhanced artificial bee colony optimization. *International Journal of Innovative Computing, Information and Control*, *5*, 5081–5092.

Wang, L., & Tang, D.-b. (2011). An improved adaptive genetic algorithm based on hormone modulation mechanism for job-shop scheduling problem. *Expert Systems with Applications*, *38*, 7243–7250.

Whitley, D., Rana, S., & Heckendorn, R. B. (1999). The island model genetic algorithm: On separability, population size and convergence. *Journal of Computing and Information Technology*, *7*, 33–48.