Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Science Letters:

A multi-crossover and adaptive island based population algorithm for solving routing problems

Eneko OSABA, Enrique ONIEVA, Roberto CARBALLEDO, Fernando DIAZ, Asier PERALLOS, Xiao ZHANG

(*Deusto Institute of Technology (DeustoTech), University of Deusto, Bilbao 48007, Spain*)

E-mail: {e.osaba, enrique.onieva, roberto.carballedo, fernando.diaz, perallos, xiao.zhang}@deusto.es

Received July 9, 2013; Revision accepted Oct. 3, 2013; Crosschecked Oct. 15, 2013

Abstract: We propose a multi-crossover and adaptive island based population algorithm (MAIPA). This technique divides the entire population into subpopulations, or demes, each with a different crossover function, which can be switched according to the efficiency. In addition, MAIPA reverses the philosophy of conventional genetic algorithms. It gives priority to the autonomous improvement of the individuals (at the mutation phase), and introduces dynamism in the crossover probability. Each subpopulation begins with a very low value of crossover probability, and then varies with the change of the current generation number and the search performance on recent generations. This mechanism helps prevent premature convergence. In this research, the effectiveness of this technique is tested using three well-known routing problems, i.e., the traveling salesman problem (TSP), capacitated vehicle routing problem (CVRP), and vehicle routing problem with backhauls (VRPB). MAIPA proves to be better than a traditional island based genetic algorithm for all these three problems.

Key words: Island model, Adaptive algorithm, Combinatorial optimization, Vehicle routing problems, Intelligent transportation systems

doi:10.1631/jzus.C1300184 **Document code:** A **CLC number:** TP273

1 Introduction

Genetic algorithm (GA) is one of the most commonly used and successful meta-heuristic to solve combinatorial optimization problems. Its basic principles were proposed by Holland (1975); however, its practical use in solving complex problems was shown later (de Jong, 1975; Goldberg, 1989). Since then, GAs have been the focus of a large amount of research, applied to a wide range of fields (Martínez-Torres, 2012; Moon *et al*., 2012). Traditional GAs have some drawbacks, however, such as premature convergence and imbalance between exploration and exploitation. To overcome these drawbacks, parallel genetic algorithms (PGAs) were proposed (Whitley *et al*., 1999). PGAs can be divided into three categories: fine grain, panmictic model, and island model. The island model is the most often used, consisting of multiple populations that evolve separately during most of the time and exchange individuals occasionally. There are a lot of studies describing the main issues about this type of GA. In Cantú-Paz (1998) you can find a comprehensive survey about PGAs.

In this paper, a multi-crossover and adaptive island based population algorithm (MAIPA) is presented to solve routing problems. This new meta-heuristic is a variant of the classic island based GA (IGA). In MAIPA the whole population is divided into different subpopulations, or demes, each with its own crossover function and crossover probability. The migration system topology is dynamic, and each subpopulation can communicate with 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.

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 occasional) and the multi-crossover, increases the exploration and exploitation capacity of the meta-heuristic, and helps prevent premature convergence.

This work is aimed to introduce this new technique, and show that it is a good alternative to solving routing problems. The technique has been applied to three routing problems, i.e., the traveling salesman problem (TSP), capacitated vehicle routing problem (CVRP), and vehicle routing problem with backhauls (VRPB), and compared with a classic island based GA.

2 Brief review of existing literature and the contribution of the presented work

As noted in the introduction, the proposed technique is a multi-population algorithm. Some approaches have been proposed following this philosophy, including parallel artificial bee colony (Tsai *et al*., 2009), parallel particle swarm optimization (Niu *et al*., 2007), and parallel genetic algorithms.

As already explained, the meta-heuristic adapts the crossover probability according to the performance of the algorithm. The idea of adapting the mutation and crossover probabilities (p_m and p_c) of a GA has been proposed by Schaffer and Morishima (1987) to improve the performance of conventional genetic algorithms. This has been the subject of many studies, e.g., Wang and Tang (2011). The multi-crossover feature has also been studied for a long time (Spears, 1995; Mukherjee *et al*., 2012).

The differences between the proposed technique, a variation of the IGA, and other multi-population techniques can be referred to Cantú-Paz (1998). Compared with IGAs and other adaptive techniques, the innovative aspects of MAIPA are as follows: (1) Unlike the vast majority of PGAs, in the presented approach each subpopulation has a different crossover function and crossover probability. This helps individuals to explore the solution space differently when they migrate to other demes. This characteristic increases the exploration capability of the technique. (2) MAIPA changes the philosophy of conventional IGAs and GAs. It begins the execution with a very low or null value of crossover probability, and a high value of mutation probability. As shown in Osaba *et al*. (2013), this increases the exploitation capacity of the search. (3) MAIPA adapts the crossover probability 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 in most previous studies. (4) The introduced technique combines the adaptation of the crossover probability with a multi-population and multi-crossover system. To the best of our knowledge, this is a new approach. (5) 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 Multi-crossover and adaptive island based population algorithm for solving routing problems

As stated in Section 2, MAIPA is a variation of the conventional island based GA. Algorithm 1 shows how the meta-heuristic works. The proposed technique gives priority to the local improvement of the individuals at the mutation phase, and gives less importance to the crossover phase. This behavior is based on Osaba *et al*. (2013), who analyzed the inefficiency of the crossover phase in terms of the optimization capacity of basic GAs applied to routing problems. This is why MAIPA gives a greater

Algorithm 1 Multi-crossover and adaptive island based population algorithm (MAIPA)

```
Create the whole population
Create subpopulation
Assign the crossover function
while the termination criterion is not reached do
  for each subpopulation do
     The mutation process
     The crossover process
     Select survivors
     Update the crossover probability
  end
  The individual migration process
end
Return the best individual of the system
```

importance to the mutation phase. In spite of this, crossovers can still be beneficial to the exploration capacity, maintaining the diversity of the population. Therefore, in MAIPA crossovers are executed only when they are beneficial, adapting the crossover probability of each subpopulation to the search needs. In addition, in MAIPA each subpopulation has its own crossover function, which can change according to its performance. This feature increases the exploration capacity of the meta-heuristic.

Regarding the p_m and p_c of the proposed technique, in MAIPA all the subpopulations have a p_m equal 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 starts with a value close to 0. The crossover probability is modified differently in each subpopulation, increasing or resetting its value. The modification is based on the following criteria, with $\text{best}_i^{\text{sp}}$ being the best solution in subpopulation sp and generation i:

$\text{best}_i^{\text{sp}} > \text{best}_{i-1}^{\text{sp}}$: This means the search process evolves correctly. In this case, p_c is reset, since it is unnecessary to diversify the population.

$\text{best}_i^{\text{sp}} = \text{best}_{i-1}^{\text{sp}}$: In this case, the search process may be considered to be trapped in a local optimum, or the population is concentrating in the same region of the space of solutions. For this reason, p_c is increased, in order to increase the subpopulation diversification using crossover operators.

Note that $\text{best}_i^{\text{sp}}$ will never be worse than $\text{best}_{i-1}^{\text{sp}}$, since the best solution of each population is always maintained throughout the generations.

In this way, whenever $\text{best}_i^{\text{sp}}$ of a subpopulation is not improved in the last generation, p_c of this deme increases following Eq. (1):

$$p_c = p_c + \frac{N^2}{\text{NMF}^2} + \frac{N_G}{\text{NMF}^2}, \tag{1}$$

where N is the number of generations without improvement, N_G is the total number of generations so far, and NMF is the size of the mutation operator neighborhood.

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, allowing repetitions. A maximum value of p_c, $\max(p_c)$, is defined. This value is the same for all the demes. After all the generations, if the $\max(p_c)$ of a subpopulation is exceeded, the crossover function of this deme is randomly replaced, resetting p_c to its original value. This feature helps increase the population diversification in a better way than other state-of-the-art techniques.

Regarding the migration system of MAIPA, as mentioned 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 as follows: whenever a deme improves its $\text{best}_i^{\text{sp}}$, it shares its new best solution with all the other subpopulations. These communications help make a greater exploration of the solution space.

4 Experiments

4.1 Experiment description

MAIPA has been applied to three different routing problems, and compared with IGA. For these two meta-heuristics, similar parameters and functions are used. The difference between them lies only in their structure and flowchart. This is the most reliable setting to determine which meta-heuristic obtains better results. The three problems used in this study are the traveling salesman problem (TSP) (Lawler *et al.*, 1985), capacitated vehicle routing problem (CVRP) (Laporte, 1992), and vehicle routing problem with backhauls (VRPB) (Golden *et al.*, 1985). TSP is a well-known benchmarking problem, simple to implement and understand. There have been many studies using TSP (Sarin *et al.*, 2011; Bae and Rathinam, 2012). CVRP and VRPB are two of the most widely used routing problems. Many studies have used these two problems (Ngueveu *et al.*, 2010; Anbuudayasankar *et al.*, 2012; Mattos Ribeiro and Laporte, 2012), due to low complexity and, above all, applicability to real scenarios. These three problems are easily replicable, and thus interested readers can easily perform the same experiment, either to check the results or to compare with other techniques.

For both MAIPA and IGA, and all the three problems, the initial population is composed of 48 randomly created individuals, which are randomly divided into four subpopulations, each with 12 individuals. All the individuals are encoded using path encoding (Larranaga *et al.*, 1999). The same function is used in the selection and survivor phases for all instances, which is 50% elitist+50% random. The execution of all the algorithms finishes in 20000 generations. For IGA, p_m and p_c are set to 0.05 and 0.95, respectively. For MAIPA, p_c starts at 0. When the best solution found is not improved, p_c increases following Eq. (1); otherwise, it returns to 0. The $\max(p_c)$ is set to 0.35.

For TSP, the crossover functions implemented in 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, to make a fair comparison, the same functions are used, assigning one of them to each subpopulation. Yet, unlike MAIPA, subpopulations do not change their functions during the execution. The mutation function used for both techniques is the well-known 2-Opt (Lin, 1965). The migration system for IGA is the same as the one used in MAIPA.

For CVRP and VRPB, the crossover functions implemented are the half route crossover and half random route crossover. The half route crossover operates as follows: first, half of the routes (the shortest ones) of one of the parents are inserted into the child. Then, the nodes selected are removed from the other parent, and the remaining nodes are inserted into the child in the same order, creating new routes. The half random route crossover operates similarly. In this case, the routes selected in the first step of the process are selected randomly. The vertex insertion function is used for both problems. This operator selects a random node from a randomly chosen route of the solution. This node is extracted and inserted into another randomly selected route. With this function, the creation of new routes is possible.

4.2 Experiment results

Experiments have been performed on an Intel Core i5-2410 laptop, with 2.30 GHz CPU and 4 GB RAM. The results for TSP, CVRP, and VRPB are shown in Tables 1–3, respectively. All instances of TSP were obtained from the TSPLIB benchmark (Reinelt, 1991). For CVRP, instances were picked from the Christofides and Eilon CVRP benchmark (http://neo.lcc.uma.es/vrp). For VRPB, 11 instances were used. The first six were obtained from the Benchmark of Solomon (http://neo.lcc.uma.es/vrp), and the remaining five from the Christofides and Eilon CVRP benchmark. These instances have been adapted to VRPB; for this purpose, demand types have been changed to have pick-ups and deliveries. This change is why the optima are not shown in Table 3.

For each run, the total average, best result, standard deviation, and average runtime are shown. Each experiment is repeated 20 times. The well-known *t*-test is performed for every instance, to determine if the outcomes of MAIPA are significantly different from those of IGA. The *t* statistic has the following form:

$$t = \frac{X_1 - X_2}{\sqrt{\dfrac{(n_1-1)\mathrm{SD}_1^2 + (n_2-1)\mathrm{SD}_2^2}{n_1 n_2 - 2} \dfrac{n_1+n_2}{n_1 n_2}}}, \tag{2}$$

where X_i, SD_i, and n_i (i=1 represents MAIPA, and i=2 represents IGA) are the average, standard deviation, and number of executions of each technique, respectively. The *t* value can be positive (+), neutral (*), or negative (−). A positive value of *t* indicates that MAIPA is better than IGA; a negative value of *t* indicates that IGA is better than MAIPA; a neutral *t* indicates that the difference between the two algorithms is not significant. A 95% confidence level is used ($t_{0.05}$=2.021).

4.3 Analysis

According to Tables 1–3, some conclusions can be drawn. The most important is that the proposed technique outperforms the classic IGA in terms of solution quality and runtime. Overall, both techniques have been applied to 40 different instances. MAIPA offers better solutions and runtimes in all the cases. Due to the *t* test, these improvements in solution quality are significant in 92.5% (37 out of 40) of the instances. To be more specific, for TSP, CVRP, and VRPB, this significant improvement is given in 100% (18 out of 18), 81.82% (9 out of 11), and 90.91% (10 out of 11) of the cases, respectively.

Table 1 Results of MAIPA and IGA for the traveling salesman problem (TSP)

Instance	Optimum	Average		Standard deviation		Best		Time (s)		$t^{\#}$
		MAIPA	IGA	MAIPA	IGA	MAIPA	IGA	MAIPA	IGA	
Oliver30	420	424.6	431.0	6.41	11.44	420	420	0.10	0.24	+
Eilon50	425	445.7	451.9	8.55	12.85	434	427	0.32	0.97	+
Eil50	426	446.0	460.2	9.38	11.98	431	438	0.35	0.99	+
Berlin52	7542	8004.4	8113.4	286.11	170.17	7542	7926	0.30	1.03	+
St70	675	714.7	726.7	16.16	16.46	687	695	0.80	4.51	+
Eilon75	535	570.8	580.6	10.63	15.01	556	556	0.94	5.54	+
Eil76	538	574.1	585.8	12.57	18.39	553	563	1.01	6.55	+
KroA100	21 282	22 349.1	22 955.8	600.15	671.35	21 319	21 972	2.08	15.95	+
KroB100	22 140	23 350.9	23 764.3	421.18	720.37	22 413	22 248	2.22	16.82	+
KroC100	20 749	22 133.0	22 533.7	531.55	727.93	21 405	21 454	1.65	17.56	+
KroD100	21 294	22 281.4	22 436.9	414.32	402.37	21 464	21 836	1.98	18.53	+
KroE100	22 068	23 397.3	23 945.1	560.69	628.56	22 535	23 146	1.96	17.24	+
Eil101	629	680.8	713.1	8.51	15.46	665	697	2.29	25.57	+
Pr107	44 303	46 270.5	47 664.9	975.23	1316.34	44 764	45 705	2.73	32.45	+
Pr124	59 030	60 995.3	63 654.0	630.87	2605.16	60 077	59 697	3.66	48.54	+
Pr136	96 772	102 498.3	106 363.2	2161.56	1562.95	97 759	104 565	5.78	56.41	+
Pr144	58 537	60 969.2	63 562.6	1663.61	1557.68	58 599	61 398	5.93	62.54	+
Pr152	73 692	76 631.3	79 209.5	1013.92	2622.49	74 745	76 252	7.30	67.54	+

'+' indicates MAIPA is better than IGA

Table 2 Results of MAIPA and IGA for the capacitated vehicle routing problem (CVRP)

Instance	Optimum	Average		Standard deviation		Best		Time (s)		$t^{\#}$
		MAIPA	IGA	MAIPA	IGA	MAIPA	IGA	MAIPA	IGA	
En22k4	375	391.1	401.0	8.54	15.30	375	375	1.59	3.80	+
En23k3	569	599.9	656.2	31.75	23.44	571	601	2.05	3.14	+
En30k3	534	557.7	561.6	16.00	19.41	544	542	2.16	4.23	+
En33k4	835	899.5	911.0	23.30	27.18	864	888	2.85	6.52	+
En51k5	521	619.3	628.4	45.15	31.65	561	572	4.15	25.10	+
En76k7	682	799.9	814.1	37.34	31.43	752	764	8.83	64.12	+
En76k8	735	860.5	881.3	21.72	32.40	829	834	9.27	66.54	+
En76k10	830	963.0	971.0	19.88	24.54	935	945	7.89	55.10	+
En76k14	1021	1179.1	1183.1	33.15	52.70	1139	1142	9.54	41.16	*
En101k8	815	997.2	1004.5	46.92	67.43	919	924	14.14	52.43	*
En101k14	1071	1221.7	1246.7	34.49	56.19	1173	1182	12.15	114.55	+

'+' indicates MAIPA is better than IGA; '*' indicates the difference between the two algorithms is not significant (at 95% confidence level)

Table 3 Results of MAIPA and IGA for the vehicle routing problem with backhauls (VRPB)

Instance	Average		Standard deviation		Best		Time (s)		$t^{\#}$
	MAIPA	IGA	MAIPA	IGA	MAIPA	IGA	MAIPA	IGA	
C101	712.4	739.9	69.53	56.48	635	656	6.15	30.91	+
C201	740.1	785.4	47.02	69.54	685	682	5.45	22.41	+
R101	933.9	1003.8	22.22	33.47	901	963	4.87	27.76	+
R201	1103.2	1217.0	59.35	60.11	1046	1105	6.53	54.83	+
RC101	614.8	656.0	36.60	51.26	544	573	3.16	7.52	+
RC201	1215.4	1315.5	85.13	67.08	1133	1253	10.79	52.15	+
En30k4	597.3	602.8	67.90	37.68	522	544	3.95	6.34	*
En33k4	821.3	835.4	35.62	44.76	784	746	3.47	5.80	+
En51k5	657.4	689.6	35.54	37.61	619	642	4.63	12.15	+
En76k8	908.1	942.1	56.36	39.21	831	883	7.13	24.31	+
En101k8	1131.3	1187.5	57.00	63.21	1070	1090	8.64	53.12	+

'+' indicates MAIPA is better than IGA; '*' indicates the difference between the two algorithms is not significant (at 95% confidence level)

Crossovers are complex operations (especially for routing problems, where all constraints have to be met) in which two individuals combine their characteristics; in contrast, a mutation is a small modification of a chromosome, and requires considerably less time than crossovers. MAIPA makes fewer crossovers than IGA, and thus requires less execution time. This is reflected in the runtime, providing an advantage to the proposed meta-heuristic.

The improvement of the proposed technique can also be explained based on Osaba *et al.* (2013). The use of crossover operators enables a broad exploration of the solution space, since crossover operators are very useful resources to make jumps inside the solution space. These operators, however, do not contribute to a deep search of promising regions. For a more exhaustive search, it is necessary to use a function in charge of the local improvement of the individuals. Mutation functions satisfy this objective.

In this way, 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, to avoid local optima. Using crossovers, subpopulation can be expanded easily through the entire solution space, and it could be more probable to find promising regions. Also, the multi-crossover feature enhances this diversification, allowing a broader exploration.

In conclusion, using the basic structure of IGA, the search conducted comprises a large area of the space of solutions, but it has a small intensification capacity. This means IGA cannot obtain as good results as MAIPA.

5 Conclusions

In this paper, 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 well-known routing problems, and the outcomes have been compared with those obtained using a basic IGA. As a conclusion, MAIPA is superior to IGA, in terms of solution quality and runtime, being a good alternative to solving routing problems. Finally, the reasons why MAIPA obtains better results than IGA have been explained.

Some small improvements are planned for later versions of this technique. In the near future, these improvements will be developed and tested. The presented technique will also be applied to real-life routing problems. Its application to a dynamic distribution system of car windscreen repairs is being planned.

References

Anbuudayasankar, S., Ganesh, K., Lenny Koh, S.C., Ducq, Y., 2012. Modified savings heuristics and genetic algorithm for bi-objective vehicle routing problem with forced backhauls. *Expert Syst. Appl.*, **39**(3):2296-2305. [doi:10.1016/j.eswa.2011.08.009]

Bae, J., Rathinam, S., 2012. Approximation algorithms for multiple terminal, Hamiltonian path problems. *Optim. Lett.*, **6**(1):69-85. [doi:10.1007/s11590-010-0252-4]

Cantú-Paz, E., 1998. A survey of parallel genetic algorithms. *Calcul. Parall. Res. Syst. Rep.*, **10**(2):141-171.

Davis, L., 1985. Applying Adaptive Algorithms to Epistatic Domains. Proc. Int. Joint Conf. on Artificial Intelligence, p.161-163.

de Jong, K., 1975. Analysis of the Behavior of a Class of Genetic Adaptive Systems. MS Thesis, University of Michigan, Michigan, USA.

Goldberg, D., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Professional, Boston, USA.

Golden, B., Baker, E., Alfaro, J., Schaffer, J., 1985. The Vehicle Routing Problem with Backhauling: Two Approaches. Proc. 21st Annual Meeting of SETIMS, p.90-92.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. MIT Press, Michigan, USA.

Laporte, G., 1992. The vehicle routing problem: an overview of exact and approximate algorithms. *Eur. J. Oper. Res.*, **59**(3):345-358. [doi:10.1016/0377-2217(92)90192-C]

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. *Artif. Intell. Rev.*, **13**(2):129-170. [doi:10.1023/A:1006529012972]

Lawler, E., Lenstra, J., Kan, A., Shmoys, D., 1985. The Traveling Salesman Problem: a Guided Tour of Combinatorial Optimization, Volume 3. Wiley, New York, USA.

Lin, S., 1965. Computer solutions of the traveling salesman problem. *Bell Syst. Techn. J.*, **44**(10):2245-2269. [doi:10.1002/j.1538-7305.1965.tb04146.x]

Martínez-Torres, M., 2012. A genetic search of patterns of behaviour in OSS communities. *Expert Syst. Appl.*, **39**(18):13182-13192. [doi:10.1016/j.eswa.2012.05.083]

Mattos Ribeiro, G., Laporte, G., 2012. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Comput. Oper. Res.*, **39**(3):728-735. [doi:10.1016/j.cor.2011.05.005]

Moon, I., Lee, J.H., Seong, J., 2012. Vehicle routing problem with time windows considering overtime and outsourcing vehicles. *Expert Syst. Appl.*, **39**(18):13202-13213. [doi:10.1016/j.eswa.2012.05.081]

Mukherjee, S., Ganguly, S., Das, S., 2012. A strategy adaptive genetic algorithm for solving the travelling salesman problem. *LNCS*, **7677**:778-784. [doi:10.1007/978-3-642-35380-2_91]

Ngueveu, S., Prins, C., Wolfler Calvo, R., 2010. An effective memetic algorithm for the cumulative capacitated vehicle routing problem. *Comput. Oper. Res.*, **37**(11):1877-1885. [doi:10.1016/j.cor.2009.06.014]

Niu, B., Zhu, Y., He, X., Wu, H., 2007. MCPSO: a multi-swarm cooperative particle swarm optimizer. *Appl. Math. Comput.*, **185**(2):1050-1062. [doi:10.1016/j.amc.2006.07.026]

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. Proc. 8th Int. Symp. on Applied Computational Intelligence and Informatics, p.17-23.

Ray, S., Bandyopadhyay, S., Pal, S., 2004. New Operators of Genetic Algorithms for Traveling Salesman Problem. Proc. 17th Int. Conf. on Pattern Recognition, p.497-500.

Reinelt, G., 1991. TSPLIB: a traveling salesman problem library. *ORSA J. Comput.*, **3**(4):376-384. [doi:10.1287/ijoc.3.4.376]

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. *Optim. Lett.*, **5**(2):259-272. [doi:10.1007/s11590-010-0205-y]

Schaffer, J.D., Morishima, A., 1987. An Adaptive Crossover Distribution Mechanism for Genetic Algorithms. Proc. 2nd Int. Conf. on Genetic Algorithms and Their Application, p.36-40.

Spears, W.M., 1995. Adapting Crossover in Evolutionary Algorithms. Proc. Conf. on Evolutionary Programming, p.367-384.

Syswerda, G., 1991. Schedule Optimization Using Genetic Algorithms. *In*: Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York, USA, p.332-349.

Tsai, P.W., Pan, J.S., Liao, B.Y., Chu, S.C., 2009. Enhanced artificial bee colony optimization. *Int. J. Innov. Comput. Inf. Control*, **5**(12):5081-5092.

Wang, L., Tang, D.B., 2011. An improved adaptive genetic algorithm based on hormone modulation mechanism for job-shop scheduling problem. *Expert Syst. Appl.*, **38**(6): 7243-7250. [doi:10.1016/j.eswa.2010.12.027]

Whitley, D., Rana, S., Heckendorn, R.B., 1999. The island model genetic algorithm: on separability, population size and convergence. *J. Comput. Inf. Technol.*, **7**:33-48.