Intelligent Automation And Soft Computing, 2020
Vol.26,no.3,501–512
DOI: 10.32604/iasc.2020.013926

Niche Genetic Algorithm for Solving Multiplicity Problems in Genetic Association Studies

Fu-I Chou[1], Wen-Hsien Ho[2,3], Chiu-Hung Chen[4,*]

[1] Department of Automation Engineering, National Formosa University, Taiwan, R.O.C.
[2] Department of Healthcare Administration and Medical Informatics, Kaohsiung Medical University, Kaohsiung, Taiwan, R.O.C.
[3] Department of Medical Research, Kaohsiung Medical University Hospital, Kaohsiung, Kaohsiung, Taiwan, R.O.C.
[4] Department of Mechanical and Computer-Aided Engineering, Feng Chia University, No. 100 Wenhwa Rd., Seatwen, Taichung, Taiwan 407, R.O.C.

ABSTRACT
This paper proposes a novel genetic algorithm (GA) that embeds a niche competition strategy (NCS) in the evolutionary flow to solve the combinational optimization problems that involve multiple loci in the search space. Unlike other niche-information based algorithms, the proposed NCSGA does not need prior knowledge to design niche parameters in the niching phase. To verify the solution capability of the new method, benchmark studies on both the travelling salesman problem (TSP) and the airline recovery scheduling problem were first made. Then, the proposed method was used to solve single nucleotide polymorphism (SNP) barcodes generation problems in a genetic association study. Experiments showed that the proposed NCS-based solver substantially improves solution quality by maintaining multiple optima.

KEY WORDS: genetic algorithm, combinational optimization, travelling salesman problem, genetic association, single nucleotide polymorphism.

1 INTRODUCTION

Combinational optimization problems that have proven to be NP-hard problems include the travelling salesman problem (TSP) (Woeginger, 2003; Odili, 2017) and single nucleotide polymorphism (SNP) selection problems (Mahdevar et al., 2010). Due to the large numbers of combinations in the search space, polynomial-time algorithms are ineffective for exploring optima (Mousavi & Zandieh, 2016). Heuristic algorithms are widely used as problem solvers in the literature. Specifically, evolutionary algorithms have proven effective for searching for global solutions (Lin et al., 2003). However, for solving real-world problems such as job-shop scheduling problems (Pérez et al., 2003) and decision systems (Addison et al., 2013) that involve multiple optima in the problem domain, capability to explore multiple solutions is essential in many applications. However, a simple genetic algorithm (SGA) maintains only a single optimum, and cannot efficiently explore multiple global solutions (Chen et al., 2014; Chen et al., 2018a; Chen et al., 2018b; Ho et al., 2018). Therefore, in problems involving many loci or optima in the search space, the SGA cannot efficiently explore multiple optima and is easily trapped in local solutions.

Various niche methods proposed in the literature can maintain multiple solutions in the evolutionary population to reduce the genetic drift effects of the replacement operator in the SGA. Generally, two solutions for drift problems in niche methods have emerged. One solution is to enhance the niche mechanism by considering conceptual niche information such as niche radius (e.g., sharing function in Goldberg, 1989; species conservation method in Li et al., 2009) or niche number (e.g., clearing method in Della Cioppa et al., 2004). However, the niche number and shape widely vary in different problems and are often difficult to track in advance. For example, the solution landscape of combinational optimization problems is particularly intractable. Another solution is to apply a parameter-free paradigm that does not require additional parameters to join the niche mechanism. The niche algorithms used in these methods usually implement dynamical niche detection or conceptual in-niche

CONTACT Chiu-Hung Chen cchung688@gmail.com; chiuhchen@fcu.edu.tw

competition. Crowding method (De Jong, 1975) implicitly treats parent individuals as niche locations and performs an in-niche competition. In a crowding GA (CGA), however, a large genetic shift can easily occur in the replacement. Deterministic crowding (DC) (Mahfoud, 1995) assumes that offspring tend to be generated near niches where their parents are located and perform an in-niche competition between the offspring and their parents. However, if the offspring generated in the DCGA loop through crossover and mutation substantially differ from their parent, the genetic shift problem remains unsolved. Cluster crowding (CC) (Ling et al., 2008) builds a parent-offspring relation tree and dynamically detects and resets niche centers in each evolutionary generation. Compared to the crowding GA, however, CCGA may obtain a poorer niche quality while maintaining the niche structures. Twin-space crowding (TC) (Chen et al., 2014) also treats parents as niche centers, and the competition between the parent and the offspring is performed within a pair of niche locations around the centers. In both CCGA and TCGA, however, a valley checking function in the algorithm steps requires a chromosome representation that supports a real-coded interpolation between the gene values. This behavior is unsuitable for solving discrete-coded problems; hence, real-coded schemas are their main targets.

To maintain the diversity of the GA without requiring prior knowledge, this study proposes a niche competition strategy (NCS) for applying parameter-free competence rules. Solution performance of the NCSGA is first characterized by using a set of combinational benchmark TSPs (Reinelt, 1994). The TSP is the NP-hard problem of optimizing the travel route to a group of cities. The problem is solved by minimizing the total distance required to travel to each city one time and then return home. The practical applications of the TSPs frequently repeat the same tours, so it will save a lot of costs in the long run after the accumulation if an improvement can be done in the standard benchmark problems. Then, the airline recovery scheduling problem (Chen et al., 2013; Liu et al., 2010) is also studied to analyze the convergence characteristics. Because of the high complexity, the two problems are highly suitable for performance verification of algorithms.

The applicability of the proposed method was also evaluated in the genetic association study of generating SNP barcodes. An SNP barcode is useful for analyzing genome-wide data to identify marker genes associated with various diseases (Roses et al., 2007). Using the SNP barcodes to analyze SNP-SNP interactions related to polygenic diseases is very time-consuming because SNP combinations are potentially very complex (Lancia et al., 2001). That is, studying barcodes generated for multiple SNPs is complicated by the many allele combinations that are possible when multiple SNPs are examined simultaneously. In a previous study by Chang et al. (2008), an odds ratio-based genetic algorithm (ORGA) was used to predict susceptibility to osteoporosis. The ORGA generated SNP barcodes for genotypes and explored the optimal SNP groups. However, the genetic flow of the ORGA resembled that of a simple GA, which limited the capability of the ORGA to explore individual optima. The NCSGA proposed here avoids this limitation. The solution capability of the proposed NCSGA is evaluated in a comparative study.

2 NICHE COMPETITION STRATEGY GENETIC ALGORITHM (NCSGA)

HOLLAND (1975) introduced the concept of GAs, in which the population is first initialized, and then a continuing evolutionary loop repeatedly selects the best fitting chromosomes for the mating pool. Crossover and mutation operators are used to generate and evaluate new offspring. Before entering a new generation, a replacement operator is used to renew the population. When a simple replacement operator is used, however, selection pressure often make the population converged into a single optimum. The niche family of GAs mainly improves the replacement operator for enhanced convergence. Figure 1 shows the evolutionary flow of the classic NGA.

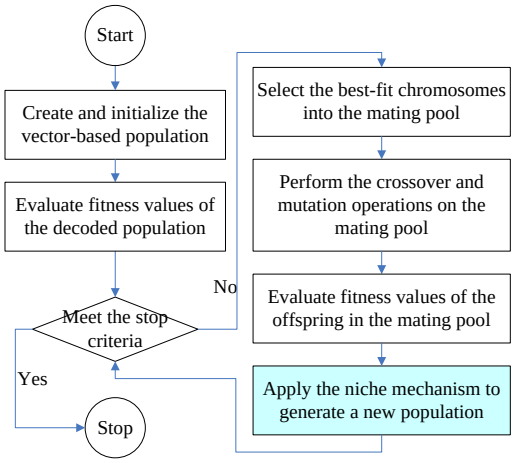


Figure 1. Flow chart of a classical niche GA.

2.1 *Niche Competition Strategy (NCS)*

In a discrete-type coding schema (Garcia-Hernandez et al., 2013), the similarity between two chromosomes is often judged in terms of hamming distance (HD). A short HD between two chromosomes is interpreted as a high similarity. Strict diversity is maintained by holding competitions to limit or even remove similar individuals. Another consideration is fitness: individuals with the highest fitness should be preserved so they can lead evolution of the overall

population. Therefore, both the similarity and fitness of the population must be considered simultaneously during evolution. This study proposes a method of controlling diversity by inserting a set of niche competition rules during the replacement phase of a GA.

When performing a replacement operation to insert offspring into a parent population, a key consideration is minimizing loss of diversity. In the following discussion of insertion rules, a dominance relation is denoted as $a \succ b$ if a has better fitness, a replacement relation is denoted as $a \leftarrow b$ when a is replaced by b, and the hamming distance evaluation of a and b is denoted as $|ab|$. In accordance with these definitions, the pseudo steps are as follows:

NCS
Input: individual x inserted into population set **P**
Output: new parent set **P**
Begin
Phase I:
Find $p \in \mathbf{P}$ that is closest to x.
If $x \succ p$, **then**
$p \leftarrow x$, and stop.
End

Phase II:
For all $p_1 \in \mathbf{P} - \{p\}$,
Find $p_2 \in \mathbf{P} - \{p_1\}$ that is closest to p_1.
If $x \succ p_1$ and $|p_1p_2| < |px|$, **then**
$p_1 \leftarrow x$, and stop.
End

In summary, the above strategy is performed in two phases. The first phase tends to replace the most similar parent by the offspring if the offspring has better fitness; this rule satisfies the condition that fitness is enhanced with minimal loss of diversity. The second phase performs an extensive search for a group of parents that have higher similarity but lower fitness; that is, this step removes parents with high similarity but lower fitness so that fitness can be improved without decreasing diversity. By applying these rules, the proposed NCSGA maintains population diversity while enhancing population fitness.

2.2 Evolutionary Flow with NCS

As new generations of the population evolve, the NCS is used as a replacement operator in the mating pool after crossover and mutation. The pseudo algorithmic steps of the NCSGA are as follows:

NCSGA Algorithm
Input:
1) (parent) population P
2) offspring size, crossover rate and mutation rate
3) stop criteria
Output: population containing the optimal solutions
Begin
Initialize the population **P**.
Evaluate the fitness values of **P**.
Repeat
Select the best-fit individuals of **P**, and enter them in the mating pool **M**.
Perform crossover and mutation operations on the individuals of **M**.
Evaluate the fitness values of **M**.
Obtain a new **P** by calling the *NCS* for all offspring of **M**.
Until the stop criterion is met.
End

Because the NCS is a parameter-free mechanism, the same optimization steps are used in different combinational problems. This feature makes it suitable for real-world applications due to the reduced number of parameters to be designed and included in the solution flow.

3 CASE STUDIES FOR PERFORMANCE VERIFICATION

BECAUSE the NCSGA is targeted at solving discrete-type optimization problems, the performance benchmark problem used in this study was the TSP. Different from the other studies only considering a single best tour of a TSP, this study focuses on exploring multiple optimal solutions.

For a TSP involving n cities $\{c_1, \cdots, c_n\}$, the computation complexity by a direct counting method is as high as O(n!). Any improvement method known to obtain a guaranteed solution still requires the complexity of exponential growth (Applegate et al., 2006). For example, the computation complexity of the Bellman–Held–Karp dynamic programming method is O(2nn2), and it takes years for a 4GHz computer to get a guaranteed best solution when n is greater than 35. Therefore, it is ineffective for exploring an exact optimum by an exhausted search. Moreover, it is much harder to solve the solution multiplicity of TSP problems. To the best of the authors' knowledge, there is no efficient algorithm to obtain exact multiple solutions in literature. Therefore, in this paper, evolutionary methods were used.

3.1 Optimization Problem and Evolutionary Methodology

Where d_{ij} denotes the Euclidean distance between cities c_i and c_j, the TSP can be mathematically formulated (Applegate et al., 2006) as follows:

$$\text{Min} \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} d_{ij} x_{ij}, \quad (1)$$

Sbj. to

$$\forall i, \sum_{j=1, i\neq j}^{n} x_{ij} = 1,$$
$$\forall j, \sum_{i=1, i\neq j}^{n} x_{ij} = 1,$$
$$\forall i, \forall j, y_i - y_j + n x_{ij} \leq n - 1, i \neq 1, j \neq 1, i \neq j, x_{ij} \in \{0,1\},$$
$$\text{and } \forall i, y_i \geq 0, y_i \in I.$$

where the binary variable x_{ij} takes the value 1 if the path from city i to j is selected and 0 otherwise.

To solve the TSP by using a GA, a designed tour sequence $\{\tilde{c}_1, \tilde{c}_2, \cdots, \tilde{c}_n\}$ is encoded into a chromosome, and fitness is evaluated according to the tour length decoded from the chromosome. When chromosomes are included in evolutionary loops, suitable genetic operators must be designed to assist the evolutionary flow.

3.2 Coding Schema

The coding schema used in this study was order-based. In the encoding phase of the schema, a discrete-type chromosome encodes the orders of cities into genes.

3.3 Selection, Crossover and Mutation

Although previous studies have designed different genetic operators to solve the TSP (Yuan et al., 2013; Rani & Kumar, 2014; Thanh et al., 2015), the focus of this study was to compare the performance of different niche methods based on the same genetic operators. Therefore, the crossover and mutation methods were kept simple to enable easy verification and validation.

Genetic operators use rank selection (Goldberg & Deb 1991) to select the best-fit chromosomes for the mating pool. A reverse crossover is performed by randomly selecting two gene positions and reversing the orders between the two positions; a shuttle mutation is performed by randomly selecting two positions and randomly swapping the orders in the selected segment.

3.4 Local Search Improvements

Studies show that, due to slow convergence, a GA usually requires a large number of evolutionary generations to obtain good results (Snyder & Daskin, 2006). Therefore, many studies have attempted to hybridize evolutionary algorithms with a local search to solve combinational optimization problems (Misevičius et al., 2015).

A local search is often performed by iteratively searching the neighborhood of one solution to explore better solutions with the minimum number of searches (Chen et al, 2015). While the genetic operator is generally used to explore the entire search space with large scale stepping, the local search mechanism is used to explore better solutions within a local area. In the case of the TSP, embedding a local search heuristic in GA can obtain a robust search. Specifically, Lin-Kernighan (LK) method (Karapetyan & Gutin, 2011) obtains the currently best heuristic solution for the TSP by repeatedly changing the number of cities visited by the travelling salesman. Therefore, this study performed a LK local search on the new born offspring. Figure 2 is a diagram of the LK-NCSGA showing the insertion point of the embedded LK local search.

3.5 Experiments and Results

The proposed NCSGA was first evaluated using a set of symmetric TSP benchmarks from TSPLIB, where $d_{ij}=d_{ji}$ for all i and j in Eq. (1). In this test, a simple SGA (SGA) and parameter-free niche algorithms, including CGA, DCGA and the proposed NCSGA, execute the entire test in 20 runs. The genetic parameters are set to a population size of 200, a crossover rate of 1.0 and a mutation rate of 0.1.

Since the complete solution space of the TSP is intractable, the actual distribution of niches is difficult to determine. Therefore, a clear comparison of the capabilities of various methods requires the use of several indices to measure convergence performance and the convergence quality of the population. The "Best" and "Mean (Std)" indices record the best, the average and the standard variation values of the optimal chromosome obtained in the executed runs and can be used to evaluate optimization performance and niche exploration performance.

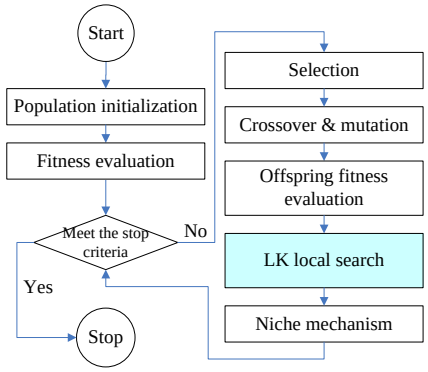


Figure 2. Summarized flow chart of a niche GA embedded with an LK local search.

Another index "Num K%" measures the average number of the solutions in the runs within K% of the optimum where K is a constant defined as follows:

$$K=\left[\frac{(\text{The tour length explored by a GA})}{(\text{The optimal tour length})}-1\right]\times 100\%. \quad (2)$$

In most evolutionary algorithms, the tour lengths obtained by the convergence procedure are more than the optimum within 2% to 4% (Snyder & Daskin, 2006; Rego et al., 2010). Therefore, the indices "Num 2%" and "Num 4%" were used to measure the solution quality in this paper.

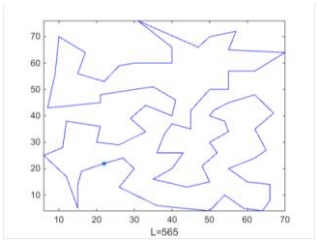
However, because the actual optima are intractable in most TSP test cases, the updated optimal values in TSPLIB were entered in the "Opt" field. Furthermore, for fair, each algorithm was executed with the same evolutionary loops (generations) in a run. The number of loops was entered in the "Loops" field. Each algorithm is directly stopped if the loop count limitation is met. Table 1 compares the test cases and related results for each index used in these experiments.

Table 1. Comparison results for various niche GAs in TSP cases.

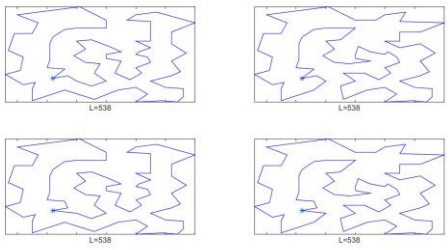
TSP Cases	Opt.	Loops	SGA		CGA		DCGA		NCSGA	
			Best	Mean (Std)	Best	Mean (Std)	Best	Mean (Std)	Best	Mean (Std)
eil51	426	30000	430	451.7 (10.95)	**426**	428.2 (1.81)	427	430.2 (2.52)	**426**	**428.0** (2.01)
berlin52	7542	30000	7623	8082.8 (254.08)	**7542**	7578.7 (88.79)	**7542**	**7556.3** (51.52)	**7542**	7602.3 (108.7)
st70	675	50000	688	719.1 (26.32)	**675**	682.7 (3.63)	678	683.7 (3.71)	**675**	**679.5** (3.85)
eil76	538	30000	568	579.9 (8.29)	**538**	545.5 (5.78)	544	552.7 (4.68)	**538**	**543.2** (4.99)
pr76	108159	60000	109265	114601.4 (2482.22)	108308	109238.1 (406.69)	**108159**	109123.3 (532.31)	**108159**	**108570.8** (538.36)
kroA100	21282	75000	22175	23305.3 (496.55)	21296	21575.0 (240.94)	**21282**	21500.9 (152.32)	**21282**	**21379.5** (124.76)
kroC100	20749	75000	21847	22947.5 (803.73)	20921	21197.7 (229.61)	20949	**21159.6** (176.28)	**20769**	21175.2 (261.8)
kroD100	21294	75000	22044	22994.2 (573.39)	21309	21595.8 (173.09)	21462	21729.5 (121.7)	**21294**	**21666.9** (240.13)
rd100	7910	75000	8195	8677.9 (261.56)	7914	**8047.9** (102.18)	7969	8105.5 (84.65)	**7910**	8108.7 (123.06)

3.5.1 Solution Multiplicity of the TSP

As mentioned in the statements above, the SGA cannot maintain multiple optima. Figure 3 demonstrates the advantage of the proposed NCSGA by comparing its optimal solutions with those obtained by SGA for test case "eil76" where the symbol 'L' is denoted as the tour length.



(a) The optimal solution explored by SGA (L=565).



(b) The optimal solutions explored by NCSGA (L=538).

Figure 3. Examples of optimal solution diagrams for SGA and NCSGA in Case eil76.

The figure shows that the SGA only explores one local solution whereas the NCSGA can explore multiple optima. More solutions in the evolutionary population explored by the NCSGA are shown in Fig. 4.

Furthermore, in the case, the SGA meets the premature condition. For reference, Fig. 5 shows the convergence diagrams for various algorithms. From the diagrams, the SGA cannot explore the optimum but although all the other algorithms can reach the optimum.

3.5.2 Solution Quality of Various Niche Methods

Table 1 also compares solution quality in different niche algorithms. In all test cases, the NCSGA outperforms the SGA and the NGA. Additionally, the results obtained by the NCSGA are better or at least comparable to those obtained by DCGA. For example, in test cases "kroC", "kroD100" and "rd100", only the NCSGA reaches the optimum for "Best". In test cases "berlin52" and "kroC100", the NCSGA obtain comparable solutions to the DCGA for "Mean"; for other indices however, NCSGA obtains a substantially higher average number and quality of solutions.

The capability to explore multiple solutions is shown in Table 2. In all test instances, the NCSGA obtains more solutions within 2% and 4% above optimal respectively. That is, the NCSGA outperforms the other algorithms in terms of diversity and number of feasible solutions.

3.5.3 Improvement Comparison by Local Search

Table 3 compares the results obtained by different niche algorithms improved by an LK heuristics search. The same performance indices are used. The use of a local search can substantially reduce the number of evolutionary generations needed to obtain good solutions. In the test cases, GAs with an LK search required only about 0.5% generations to reach the

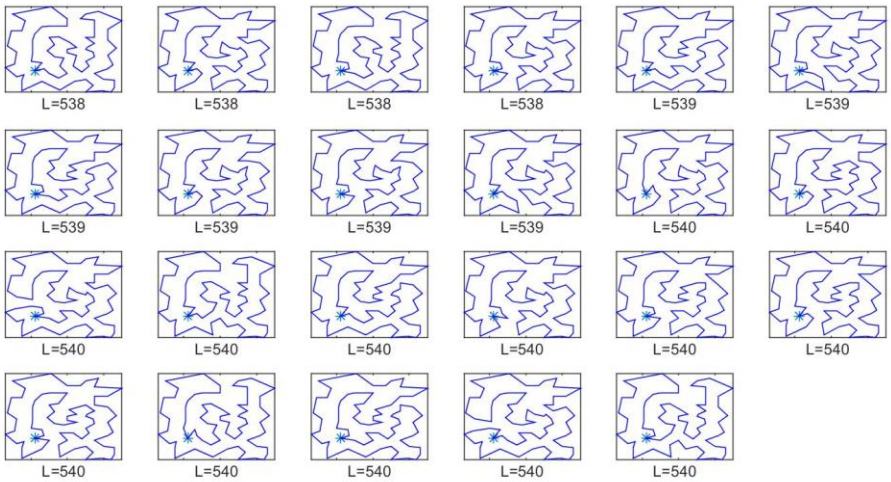


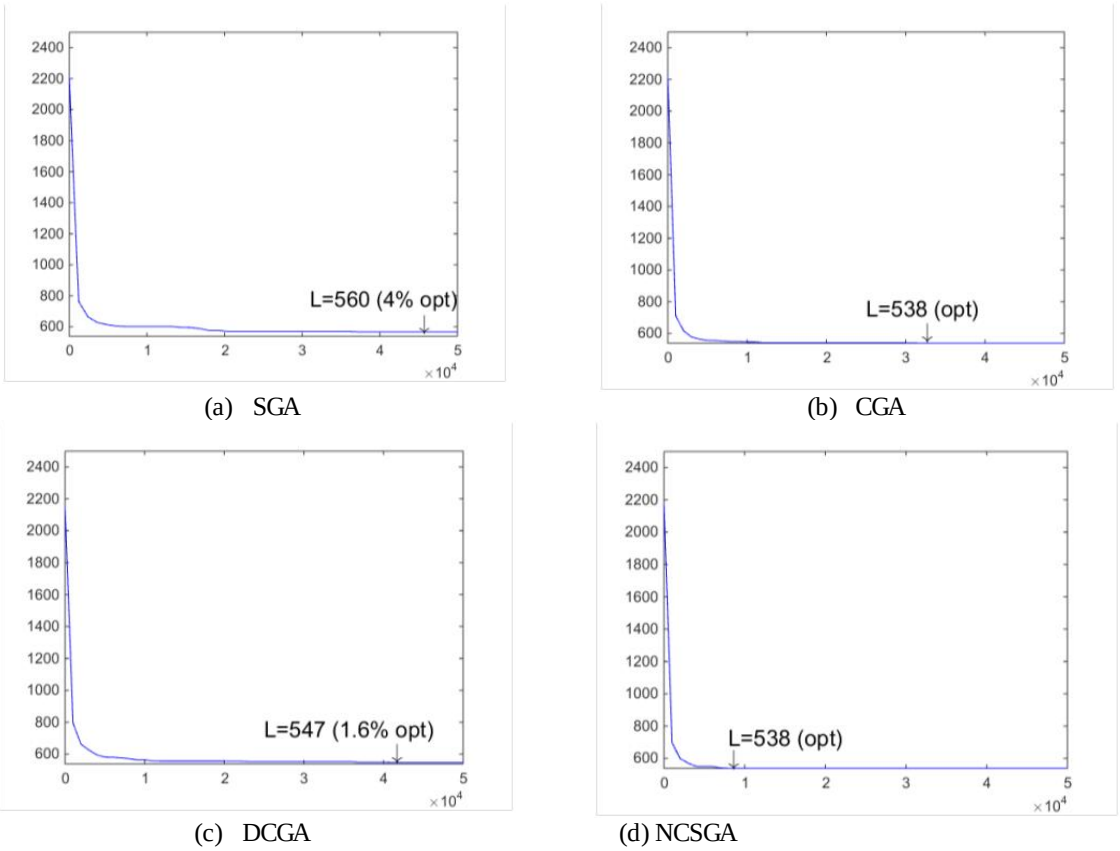
Figure 4. Solutions within 0.5% of the optimum explored by NCSGA in Case eil76.

Table 2. Solution multiplicity comparison for various niche GAs in TSP cases.

TSP Cases	Opt.	Loops	SGA		CGA		DCGA		NCSGA	
			Num(4%)	Num(2%)	Num(4%)	Num(2%)	Num(4%)	Num(2%)	Num(4%)	Num(2%)
eil51	426	30000	0.2	0.1	164.9	38.5	51.5	8.1	**196**	**54.5**
berlin52	7542	30000	0.2	0.1	77.5	11.3	35	9.9	**116.9**	**11.4**
st70	675	50000	0.3	0.1	88.8	11	46.6	15.2	**136.8**	**17.8**
eil76	538	30000	0	0	166	51.5	21.1	2.4	**198.7**	**93.7**
pr76	108159	60000	0.3	0.1	118.3	18.8	43.9	10.2	**198.3**	**126**
kroA100	21282	75000	0	0	133.7	27.8	37.9	8.6	**183.2**	**53.1**
kroC100	20749	75000	0	0	51.2	6.5	22.8	4.3	**77.7**	**7.6**
kroD100	21294	75000	0.1	0	109.1	14.6	26.5	3.4	**154.7**	**33**
rd100	7910	75000	0.1	0	67	3.6	23.1	3.1	**126.3**	**28.1**

Table 3 TSP comparison results of various niche GAs with an L-K search.

TSP Cases	Opt.	Loops	LK-SGA			LK-CGA			LK-DCGA			LK-NCSGA		
			Mean (Std)	Num (4%)	Num (2%)	Mean (Std)	Num (4%)	Num (2%)	Mean (Std)	Num (4%)	Num (2%)	Mean (Std)	Num (4%)	Num (2%)
eil51	426	150	426 (0)	40.5	11.1	426 (0)	87.5	53.5	426 (0)	71.3	47	426 (0)	**126.2**	**88.8**
berlin52	7542	150	7542 (0)	38.7	**16.1**	7542 (0)	46.9	7.5	7542 (0)	36.5	14.3	7542 (0)	**70.8**	15.3
st70	675	250	675.1 (0.30)	10.1	3.5	675 (0)	50.5	22.6	675 (0)	**153.2**	**109.1**	675 (0)	100.9	53.7
eil76	538	300	538 (0.22)	19.6	5	538 (0)	101.7	83.5	538 (0)	150.7	103.1	538 (0)	**142.8**	**133.6**
pr76	108159	300	108164.3 (22.88)	26.4	5.8	108159 (0)	62.2	36.2	108159 (0)	107.2	**82.2**	108159 (0)	**115.8**	79.5
kroA100	21282	750	21282 (0)	10.1	3.8	21282 (0)	128.7	60.3	21282 (0)	71.3	59.8	21282 (0)	**184.1**	**107.1**
kroC100	20749	750	20749 (0)	11.4	4.7	20749 (0)	94.8	32.7	20749 (0)	61	47.6	20749 (0)	**146.3**	**67**
kroD100	21294	750	21306.9 (24.95)	6.5	2	21294 (0)	119.9	63.8	21294 (0)	115	91	21294 (0)	**181.3**	**115.4**
rd100	7910	750	7913.1 (6.82)	4	2	7910 (0)	79.7	32.4	7910 (0)	31.8	20.4	7910 (0)	**134.2**	**60.1**



(a) SGA (b) CGA

(c) DCGA (d) NCSGA

Figure 5. Convergence diagrams for various niche GAs where only the CGA and NCSGA can reach the optimum.

index criterion of "Num 2%" and "Num 4%" compared to GAs without any local search.

According to the result, the LK-NCSGA can explore more solutions in most test cases. The only exception is case "st70" where the LK-DCGA performs better. In cases "berlin52" and "pr76", the LK-NCSGA obtains comparable solutions to the LK-SGA and the LK-DCGA. However, because the schema of LK-SGA is based on a simple GA, it cannot obtain optima in some runs. This case also demonstrates the importance of diversity control in an algorithm.

To the best of our knowledge, this work is a pioneer study to discuss both the search performance and the capability to explore multiple solutions in TSP problems. Therefore, the referential value of these results is considerable.

3.6 Comparison with Multiobjective Evolutionary Optimization Methods

As for the issue of multiple solutions, although multiobjective evolutionary algorithms (MOEAs) can also provide multiple choice schemes by searching the Pareto set of the solution space, it is essentially different from the multiple solutions solved by the proposed NCSGA method in this paper which aims at solving the single objective problems. Therefore, it is not easy to directly compare the solution capabilities of the two different approaches. Here we take the real-world airline recovery scheduling problem as an example problem to observe the convergence characteristics of the two kinds of solvers.

When encountering a disturbance event (such as a temporary shutdown of the airport due to climate problems), the airline recovery problem requires that the flight schedule must be rescheduled to complete the interrupted schedule. The original approach (Chen et al., 2013; Liu et al., 2010) was to solve the multiobjective recovery problem with the NSGA-II variant. However, if the problem is set to optimize only a single objective, e.g. minimizing the overall delay due to the disturbance, we can solve it by the NCSGA.

Table 4 shows the result of the two different approaches solving the two recovery cases studied in the work of Chen et al. (2013) under the same fitness call numbers. The table content shows the overall delay amount in the recovery schedules. From the result, the convergence of the single objective approach, that is, the NCSGA method is obviously better than that of the multiobjective scheme. Furthermore, the genotype density can be used to measure the diversity of the evolutionary population members. Table 5 shows the density values which are calculated by averaging the sum of genotype distances

Table 4. The total delay time values in the recovery solutions.

Study cases	NSGA-II Variant (Chen et al, 2013)			NCSGA (This work)		
	Min	Mean	Std.	Min	Mean	Std.
Case #1	595	764.5	115.24	**595**	**619.75**	**31.81**
Case #2	390	430.75	38.53	**380**	**416.75**	**15.58**

*Remark: A smaller delay is better, and the fitness function calls are set to 200000.

Table 5. The genotype density in the evolutionary population.

Study cases	NSGA-II Variant (Chen et al, 2013)			NCSGA (This work)		
	Max	Mean	Std.	Max	Mean	Std.
Case #1	22.51	16.61	2.58	**25.98**	**22.99**	**1.79**
Case #2	19.84	15.04	2.52	**29.47**	**26.73**	**2.14**

*Remark: A larger density represents a higher diversity.

from one population member to the others. Because of niching method, the NCSGA also maintain a better diversity by keeping a larger density.

To sum up, because of the niching mechanism and the single objective scheme, the NCSGA method has a unique convergence direction (objective) to obtain a better convergence and diversity. However, if Pareto solutions are required, a multiobjective approach is still preferred.

4 APPLICATION OF NCSGA TO SNP BARCODE GENERATION

THE proposed NCSGA was used here to solve the medical feature selection problem (Arif et al., 2017) which generates SNP barcodes between SNPs and osteoporosis (Chung et al., 2007). In the genetic association study, a SNP barcode is formed by combining SNPs; individuals have unique SNP barcodes for each genotype, and three SNP combinations are possible for each genotype.

In Chung et al. (2007), SNP barcodes and phenotypes were used to distinguish between a case group and a control group based on bone mineral density (BMD). When individuals were divided into high and low BMD groups, each group showed different SNP barcode patterns. After generating SNP barcodes for different genotypes, Chung et al. (2007) identified the optimal SNP pairs by comparing the occurrence of SNP pairs between the case and control groups in fitness evaluations.

4.1 Coding Schema

Chromosomes were represented by dividing them into two parts as described in the previous work. The first part is the selected number of SNPs, and the second part is the genotype associated with the SNPs. Therefore, a chromosome with eight genes would be represented by four randomly selected SNPs and their genotypes.

An example of this coding schema is {(6, 4, 7, 8), (3, 2, 2, 1)}. The (6, 4, 7, 8) denotes the selected SNPs, and (3, 2, 2, 1) denotes their genotypes. In this case, the "selected SNPs and their genotypes" field was encoded as (6, 3), (4, 2), (7, 2) and (8, 1).

4.2 Genetic Operators

For a fair comparison, the same genetic operators were used. For rank selection, the genetic schema used the fitness order as the selection probability. Crossover was performed by a two-segment single-point crossover model that separately mated the parts of two chromosomes to generate new offspring. However, a repair method was needed to repair the chromosome after the crossover. No mutation method was applied in the application flow.

4.3 Fitness Evaluation

In a previous study, the T-score for BMD was used in the fitness evaluation. Individuals with T-scores higher than -1 were classified into a high BMD (H_BMD) group. The remainders were classified into a low BMD (L_BMD) group.

Based on this classification, the fitness evaluation of each chromosome in the population was performed as follows:

$$Fitness = \frac{(\text{H\_BMD} - \text{L\_BMD})}{(\text{ALL\_H\_BMD} + \text{ALL\_L\_BMD})} \times 100\%, \quad (3)$$

where ALL_H_BMD and ALL_L_BMD denote the numbers of individuals in the high and low BMD groups respectively; H_BMD and L_BMD denote the numbers of individuals that match the selection conditions in the high and low BMD groups, respectively. According to the equation, a high percentage indicates a high probability that the SNP selection and genotype combinations are associated with osteoporosis.

4.4 Experimental Results

A dataset from a previous study of associations with osteoporosis (Chung et al., 2008) was used for experimental performance evaluations of the proposed NCSGA. The dataset included BMD, BMI, SNP, personal information, and clinical data. Table 6 shows the 11 SNP candidates that emerged in analyses of complex networks with direct or indirect crosstalk.

The case group and the control group in this analysis included 190 subjects with high BMD and 117 subjects with low BMD, respectively. The GA generated SNP barcode profiles by coupling SNP barcodes with phenotype (BMD).

Table 7 compares the results obtained by the ORGA and by the NCSGA. The table shows that the original ORGA only explored a single optimum in each combination whereas the NCSGA explored additional optima. For example, in test instance SNP #2, the test case required two SNPs to form a SNP barcode. The best SNP barcode obtained by the ORGA was "3-3" in the selected "SNP(1-5)", i.e., genotype "CC" of SNP "TNFα-857" and genotype "AA" of SNP "PTH(BstB I)". In contrast, an

Table 6. SNP Information in the Osteoporosis Association Study (Chung et al., 2008).

SNP	RS number	Geno Type1	Geno Type2	Geno Type3	Chromosome	Gene
1	rs1799724	TT	TC	CC	6	TNFα-857
2	rs1800469	TT	TC	CC	19	TGFβ1-509
3	rs1800247	CC	CT	TT	1	Osteocalcin
4	rs1800629	AA	AG	GG	6	TNFα308
5	rs6254	GG	AG	AA	11	PTH(BstB I)
6	rs6256	AA	AC	CC	11	PTH(Dra I)
7	VNTR[a]	A1A1	A1A2	A1A4	2	IL1_ra
8	rs2227956	CC	CT	TT	6	HSP70 hom
9	rs1061581	GG	AG	AA	6	HSP 70-2
10	rs1801197	CC	CT	TT	7	CTR
11	rs17563	CC	CT	TT	14	BMP-4

Table 7. Comparison SNP barcodes generated by ORGA and by NCSGA.

SNP #	Odds Ratio	**ORGA** (Chung et al., 2008)		**NCSGA** (this work)	
		Selected SNPs	SNP barcode	Selected SNPs	SNP barcode
2	2.93	SNP(1-5)	3-3	SNP(1-5) **SNP(1-7)**	3-3 **3-1**
3	2.83	SNP(1-5-7)	3-3-1	SNP(1-5-7)	3-3-1
4	3.02	SNP(1-4-5-7)	3-3-3-1	SNP(1-4-5-7)	3-3-3-1
5	2.44	SNP(1-4-5-7-10)	3-3-3-1-1	SNP(1-4-5-7-10)	3-3-3-1-1
6	2.80	SNP(1-4-5-7-9-10)	3-3-3-1-2-1	SNP(1-4-5-7-9-10)	3-3-3-1-2-1
7	3.03	SNP(1-4-5-6-7-8-10)	3-3-3-3-1-3-1	SNP(1-4-5-6-7-8-10)	3-3-3-3-1-3-1
8	2.35	SNP (1-4-5-6-7-8-9-10)	3-3-3-3-1-3-2-1	SNP (1-4-5-6-7-8-9-10)	3-3-3-3-1-3-2-1
9	3.57	SNP(1-3-4-5-6-7-8-9-10)	3-3-3-3-3-1-3-2-1	SNP(1-3-4-5-6-7-8-9-10) **SNP(1-4-5-6-7-8-9-10-11)** **SNP(1-3-4-5-6-7-8-10-11)**	3-3-3-3-3-1-3-2-1 **3-3-3-3-1-3-2-1-3** **3-2-3-3-3-1-3-1-3**
10	6.53	SNP(2-3-4-5-6-7-8-9-10-11)	2-2-3-3-3-1-2-2-1-3	SNP(2-3-4-5-6-7-8-9-10-11) **SNP(1-2-3-4-5-6-7-8-9-10)**	2-2-3-3-3-1-2-2-1-3 **1-2-2-3-3-3-1-2-2-1**

additional best barcode obtained by NCSGA in the selected "SNP (1-7)" was "3-1", i.e., genotypes "CC" of SNP "TNFα-857" and "A1A1" of SNP "IL1_ra". That is, the ORGA failed to obtain an equivalent SNP barcode in this case. Similar results were obtained for test instances SNPs #9 and #10. Specifically, SNP #10 had one additional barcode whereas case SNP #9 had two additional barcodes.

In general, gene damage or modification usually leads to an increased risk of many diseases. Therefore, it is important to obtain sensitive SNP patterns using the SNPs involved in the pathways related to genetic variants. However, obtaining an equivalent set of SNPs is also important for high solution quality because more complete solutions can help analyze the most significant pathways. For the use in real applications, an example is that the decision-makers can choose a better SNP barcode from the set to determine the impact of drug therapy according to other clinical experience.

5 CONCLUSIONS

A simple GA cannot explore multiple global solutions because it lacks capability to explore multiple loci. Therefore, this study used NCS in an evolutionary algorithm to solve multiplicity problems. Different from the other niche methods which requires prior knowledge to determine specific parameters, the NCS uses a parameter-free niche mechanism and is suitable for real-world combinational optimization applications that have intractable solution landscapes.

For solving TSP benchmark problems, experiments showed that the proposed NCSGA can explore multiple solutions and outperforms other parameter-free niche methods in most test cases with or without a local search. Further experiments in the SNP barcode generation application revealed the advantage of the proposal NCSGA. Identifying SNP barcodes is difficult and very time-consuming, and the original ORGA dealt with the problem by employing a simple GA to explore the single global solution. Our approach improved the solution capability by using the NCSGA method to explore multiple optimal solutions. That is, compared to the ORGA, the proposed method provides an effective and comprehensive approach to explore more valuable solutions for the decision-makers.

Extensions of this work include the scalability verification of more real-world cases related to genetic association studies. In addition, since the proposed NCSGA is a generalized evolutionary algorithm, it can be further used or extended to solve other combinatorial optimization problems.

6 ACKNOWLEDGEMENT

THIS work was in part supported by the Ministry of Science and Technology, Taiwan, Republic of China, under grant numbers MOST 106-2221-E-244-019, MOST 106-2218-E-327-001, MOST 106-2221-

E-037-001, MOST 106-2622-E-037-005-CC3, MOST 107-2221-E-037-006, MOST 107-2218-E-992-308, MOST 108-2221-E-035-076 and the "Intelligent Manufacturing Research Center" (iMRC) from the Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan, Republic of China.

7 REFERENCES

Addison, P. F., Rumpff, L., Bau, S. S., Carey, J. M., Chee, Y. E., Jarrad, F. C., & Burgman, M. A. (2013). Practical solutions for making models indispensable in conservation decision-making, *Diversity and Distributions*, *19*(5-6), 490-502.

Applegate, D. L., Bixby, R. E., Chvatal, V., & Cook, W. J. (2006). The traveling salesman problem: a computational study. Princeton university press.

Arif, M., Kattan, A., & Ahamed, S. I. (2017). Classification of physical activities using wearable sensors. Intelligent Automation & Soft Computing, 23(1), 21-30.

Chang, H. W., Chuang, L. Y., Ho, C. H., Chang, P. L., & Yang, C. H. (2008). Odds ratio-based genetic algorithms for generating SNP barcodes of genotypes to predict disease susceptibility. OMICS A Journal of Integrative Biology, 12(1), 71-81.

Chen, Y. J. & Chen, T. H. (2018). Fair sharing and eco-efficiency in green responsibility and green marketing. International Journal of Production Economics (in press).

Chen, Y. J., Ho, W. H., Kuo, H. W., & Kao, T. W. (2018). Repositioning conflicting partners under inventory risks. IEEE Transactions on Engineering Management (in press).

Chen, C. H., Liu, T. K., & Chou, J. H. (2013). Integrated short-haul airline crew scheduling using multiobjective optimization genetic algorithms. IEEE Transactions On Systems, Man, and Cybernetics: Systems, 43(5), 1077-1090.

Chen, C. H., Liu, T. K., & Chou, J. H. (2014). A novel crowding genetic algorithm and its applications to manufacturing robots. IEEE Transactions on Industrial Informatics,10(3), 1705-1716.

Chen, B., Zeng, W., Lin, Y., & Zhang, D. (2015). A new local search-based multiobjective optimization algorithm. IEEE Transactions on Evolutionary Computation, 19(1), 50-73.

Chung, Y., Lee, S. Y., Elston, R. C., & Park, T. (2007). Odds ratio based multifactor-dimensionality reduction method for detecting gene–gene interactions. Bioinformatics, 23(1), 71-76.

De Jong, K. A. (1975). Analysis of the behavior of a class of genetic adaptive systems, PhD Dissertation, University of Michigan, Ann Arbor.

Della Cioppa, A., De Stefano, C., & Marcelli, A. (2004). On the role of population size and niche radius in fitness sharing. IEEE Transactions on Evolutionary Computation, 8(6), 580-592.

Garcia-Hernandez, L., Arauzo-Azofra, A., Salas-Morera, L., Pierreval, H., & Corchado, E. (2013). Recycling plants layout design by means of an interactive genetic algorithm. Intelligent Automation & Soft Computing, 19(3), 457-468.

Goldberg, D. E. (1989). Genetic algorithms in search, optimization and machine learning, Addison Wesley, Massachusetts.

Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. Foundations of genetic algorithms, 1, 69-93.

Ho, W. H., Chiu, Y. H., & Chen, Y. J. (2018). Multi-objective Pareto adaptive algorithm for capacitated lot-sizing problems in glass lens production. Applied Mathematical Modelling, 53(1), 731-738.

Holland, J. H. (1975). Adaptation in Natural and Artificial Systems, the University of Michigan Press, Michigan.

Karapetyan, D., & Gutin, G. (2011). Lin–Kernighan heuristic adaptations for the generalized traveling salesman problem. European Journal of Operational Research, 208(3), 221-232.

Lancia, G., Bafna, V., Istrail, S., Lippert, R., & Schwartz, R. (2001, August). SNPs problems, complexity, and algorithms. In European symposium on algorithms (pp. 182-193). Springer Berlin Heidelberg.

Li, J. P., and Wood A. (2009). Random search with species conservation for multimodal functions. Proceedings of the Eleventh Conference on Congress on Evolutionary Computation, Trondheim, Norway, pp. 3164-3171.

Lin, F. T., Kao, C. Y., & Hsu, C. C. (1993). Applying the genetic approach to simulated annealing in solving some NP-hard problems. IEEE Transactions on systems, man, and cybernetics, 23(6), 1752-1767.

Ling, Q., Wu, G., Yang, Z., and Wang, Q. (2008). Crowding clustering genetic algorithm for multimodal function optimization. Applied Soft Computing, 8, 88-95.

Liu, T. K., Chen, C. H., & Chou, J. H. (2010). Optimization of short-haul aircraft schedule recovery problems using a hybrid multiobjective genetic algorithm. Expert Systems with Applications, 37(3), 2307-2315.

Mahdevar, G., Zahiri, J., Sadeghi, M., Nowzari-Dalini, A., & Ahrabian, H. (2010). Tag SNP selection via a genetic algorithm. Journal of Biomedical informatics, 43(5), 800-804.

Mahfoud, S. W. (1995). Niching methods for genetic algorithms. Ph.D. thesis, University of Illinois at Urbana-Champaign.

Misevičius, A., Ostreika, A., Šimaitis, A., & Žilevičius, V. (2015). Improving local search for

the traveling salesman problem. Information Technology and Control, 36(2).
Mousavi, S. M., & Zandieh, M. (2016). An efficient hybrid algorithm for a bi-objectives hybrid flow shop scheduling. Intelligent Automation & Soft Computing, 1-8.
Noraini, M. R., & Geraghty, J. (2011). Genetic algorithm performance with different selection strategies in solving TSP. Proceedings of the World Congress on Engineering 2011 Vol II.
Pérez, E., Herrera, F., & Hernández, C. (2003). Finding multiple solutions in job shop scheduling by niching genetic algorithms. Journal of Intelligent manufacturing, 14(3), 323-339.
Petrowski, A., (1996). A clearing procedure as a niching method for genetic algorithms. Proceedings of the IEEE Conference on Evolutionary Computation, Nagoya, Japan, pp. 798-803.
Rani, K., & Kumar, V. (2014). Solving travelling Salesman problem using genetic algorithm based on heuristic crossover and mutation operator. International Journal of Research in Engineering & Technology, 2(2), 27-34.
Rego, C., Gamboa, D., Glover, F., & Osterman, C. (2011). Traveling salesman problem heuristics: Leading methods, implementations and latest advances. European Journal of Operational Research, 211(3), 427-441.
Reinelt, G. (1994). The traveling salesman: computational solutions for TSP applications. Springer-Verlag.
Roses, A. D., Saunders, A. M., Huang, Y., Strum, J., Weisgraber, K. H., & Mahley, R. W. (2007). Complex disease-associated pharmacogenetics: drug efficacy, drug safety, and confirmation of a pathogenetic hypothesis (Alzheimer's disease). The pharmacogenomics journal, 7(1), 10-28.
Snyder, L. V., & Daskin, M. S. (2006). A random-key genetic algorithm for the generalized traveling salesman problem. European Journal of Operational Research, 174(1), 38-53.
Thanh, P. D., Binh, H. T. T., & Lam, B. T. (2015). New Mechanism of Combination Crossover Operators in Genetic Algorithm for Solving the Traveling Salesman Problem. In Knowledge and Systems Engineering (pp. 367-379). Springer International Publishing.
Wei, X., Zheng, X., Zhang, Q., & Zhou, C. (2015, September). Improved Niche Genetic Algorithm for Protein Structure Prediction. In Bio-Inspired Computing-Theories and Applications (pp. 475-492). Springer Berlin Heidelberg.
Woeginger, G. J. (2003). Exact algorithms for NP-hard problems: A survey. In Combinatorial Optimization—Eureka, You Shrink! (pp. 185-207). Springer Berlin Heidelberg.
Yuan, S., Skinner, B., Huang, S., & Liu, D. (2013). A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms. European Journal of Operational Research, 228(1), 72-82.

8 DISCLOSURE STATEMENT

NO potential conflict of interest was reported by the authors.

9 NOTES ON CONTRIBUTORS

Fu-I Chou received the B.S. degree from the National University of Kaohsiung, Taiwan, in 2010, the M.S. degree from the National Dong-Hwa University, Taiwan, in 2012, and the Ph.D. degree from the National Cheng-Kung University, Taiwan, in 2019, all in Electrical Engineering. He is currently an Assistant Professor with the Department of Automation Engineering, National Formosa University, Taiwan. From August 2019 to January 2020, he was an Assistant Professor at the National Chin-Yi University of Technology, Taiwan. He was a Deputy Engineer of Metal Industries Research and Development Centre, Taiwan, from September 2012 to August 2019. His research interests include state observer design, automation and control, industrial robotics, artificial intelligence applications, machine learning, quality engineering, evolutionary optimization, and machine vision. He received the 2019 Doctoral Dissertation Award from the Chinese Automatic Control Society, Taiwan. He and his colleagues proposed the research and development achievement entitled Intelligent 3D Visual Automation for Shoes Roughing and Cementing Equipment, which received the 2019 American Edison Bronze Award in the robot field as well as the 6th National Industry Innovation Award from the Taiwan Ministry of Economics.

Wen-Hsien Ho received the B.S. degree in Industrial and Information Management from National Cheng-Kung University, the M.S. degree in Mechanical and Automation Engineering from National Kaohsiung First University of Science and Technology, Taiwan, in June 1998 and June 2002, respectively, and the Ph.D. degree in Engineering Science and Technology from National Kaohsiung First University of Science and Technology, Taiwan, in January 2006. He is currently a Professor in the

Department of Healthcare Administration and Medical Informatics, Kaohsiung Medical University, Taiwan. His research interests include intelligent systems and control, computational intelligence and methods, and quality engineering.

Chiu-Hung Chen received the B. Sc. degree in computer engineering from the National Chiao Tung University, Shinchu, Taiwan, in 1990, the M. Sc. degree in computer science and information engineering from the National Taiwan University, Taipei, Taiwan, in 1992, and the Ph. D degree in engineering science and technology from National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan, in 2009. He is currently an associate professor with Feng Chia University, Taichung, Taiwan. His research interests include artificial intelligence, evolutionary computation, optimization methods, smart manufacturing, and multimedia system.