Adaptive Fireworks Algorithm Based on Two-Master Sub-population and New Selection Strategy

Xiguang Li, Shoufei Han(✉), Liang Zhao, and Changqing Gong

School of Computer, Shenyang Aerospace University, Shenyang 110136, China
hanshoufei@gmail.com

Abstract. Adaptive Fireworks Algorithm (AFWA) is an effective algorithm for solving optimization problems. However, AFWA is easy to fall into local optimal solutions prematurely and it also provides a slow convergence rate. In order to improve these problems, the purpose of this paper is to apply two-master sub-population (TMS) and new selection strategy to AFWA with the goal of further boosting performance and achieving global optimization. Our simulation compares the proposed algorithm (TMSFWA) with the FWA-Based algorithms and other swarm intelligence algorithms. The results show that the proposed algorithm achieves better overall performance on the standard test functions.

Keywords: Adaptive fireworks algorithm · Two-master Sub-population · Selection strategy · Swarm intelligence algorithm · Standard test functions

1 Introduction

Fireworks Algorithm (FWA) [1] is a new group of intelligent algorithms developed in recent years based on the natural phenomenon of simulating fireworks sparking, and can solve some optimization problems effectively. Compared with other intelligent algorithms such as particle swarm optimization and genetic algorithm, the FWA algorithm adopts a new type of explosive search mechanism, which is explosive. In addition, to calculate the explosion amplitude and the number of explosive sparks through the interaction mechanism between fireworks.

However, many researchers quickly find that traditional FWA has some disadvantages in solving optimization problems, the main disadvantages include slow convergence speed and low accuracy, thus, many improved algorithms have been proposed. So far, research on the FWA has concentrated on improving the operators. One of the most important improvements of the FWA, the enhanced fireworks algorithm (EFWA) [2], the operators of the conventional FWA were thoroughly analyzed and revised. Based on the EFWA, an adaptive fireworks algorithm (AFWA) [3] was proposed, which was the first attempt to control the explosion amplitude without preset parameter by detecting the results of the search process. In [4], a dynamic search fireworks algorithm (dynFWA) was proposed in which divided the fireworks into core firework and non-core fireworks according to the fitness value and adaptive adjustment of explosion amplitude for the core firework. In addition, since the FWA was proposed,

D. Liu et al. (Eds.): ICONIP 2017, Part IV, LNCS 10637, pp. 70–79, 2017.
https://doi.org/10.1007/978-3-319-70093-9_8

it has been applied to many areas [5], including digital filters design [6], nonnegative matrix factorization [7], spam detection [8], etc.

Aforementioned AFWA variants can improve the performance of FWA to some extent. However, inhibition of premature convergence and solution accuracy improvement is still a challenge issue for further research on AFWA.

In order to improve the above problems, in this paper, the searching range of AFWA is expanded by searching the mutual cooperation between the two groups of master sub-populations, to accelerate the convergence rate and improve the search ability of the algorithm. In addition, a new selection strategy is proposed to keep the diversity of the population. Based on this, an improved fireworks optimization algorithm (TMSFWA) is proposed to improve the convergence speed and precision.

The paper is organized as follows. In Sect. 2, the adaptive fireworks algorithm is introduced. The TMSFWA algorithm is presented in Sect. 3. The simulation experiments and results analysis are given in details in Sect. 4. Finally, the conclusion summarizes in final part.

2 Adaptive Fireworks Algorithm

The TMSFWA is based on the AFWA because its ideal is very simple and it works stably. In this section, we will briefly introduce the framework and the operators of the AFWA for further discussion.

In AFWA, there are two important components: the explosion operator (the sparks generated by the explosion) and the selection strategy.

2.1 Explosion Operator

Each firework explodes and generates a certain number of explosion sparks within a certain range (explosion amplitude). The numbers of explosion sparks (Eq. (1)) calculated according to the qualities of the fireworks.

For each firework X_i, its explosion sparks' number is calculated as follows:

$$S_i = m \times \frac{y_{\max} - f(X_i) + \varepsilon}{\sum_{i=1}^{N} (y_{\max} - f(X_i)) + \varepsilon} \tag{1}$$

where $y_{max} = max(f(X_i))$, m is a constant to control the number of explosion sparks, and ε is the machine epsilon to avoid S_i is equal to 0.

In AFWA, the calculation of the amplitude of the normal fireworks and the optimal firework (the value of the objective function is smallest) are different. The normal fireworks' explosion amplitudes are calculated just as in the previous versions of FWA:

$$A_i = A \times \frac{f(X_i) - y_{\min} + \varepsilon}{\sum_{i=1}^{N} (f(X_i) - y_{\min}) + \varepsilon} \tag{2}$$

where y_{min} = *minf(X$_i$))*, *A* is a constant to control the explosion amplitude, and ε is the machine epsilon to avoid A_i is equal to 0.

But for the optimal firework, its explosion amplitude is adjusted according to the search results in the last generation:

$$A_i(t+1) = \begin{cases} UB - LB & t = 0 \ or \ f(s_i) < f(x) \\ 0.5 \times (\lambda \times ||s_i - s^*||_\infty + A_i(t)) & otherwise \end{cases} \tag{3}$$

where *UB* and *LB* stand for the upper bound and lower bound of the search space respectively, $s_1 ... s_n$ denote all sparks generated in generation *t*, *s** denotes the best spark and *x* stands for fireworks in generation *t*, and the parameter λ is suggested to be fixed value of 1.3 empirically.

2.2 Selection Strategy

In AFWA, it applies a selection method, which is referred to as elitism-random selection method. In this selection process, the optima of the set will be selected first. Then, the other individuals are selected randomly.

3 The Proposed Algorithm (TMSFWA)

The proposed algorithm (TMSFWA) is a simple and easy to implement AFWA based on two-master sub-populations and new selection strategy.

3.1 Two-Master Sub-population

The realization of the two-master sub-populations' idea is: in a random initialization of a group of fireworks, it is divided into two independent sub-populations, one is the master sub-population, and the other is the assistant sub-population. The definitions are as follows.

Definition 1. Master sub-population is the optimal firework in the current fireworks population.

Definition 2. Assistant sub-population is the fireworks except the optimal in the current fireworks population.

For the assistant sub-populations are iteratively searched by the AFWA, but displacement operation of the master sub-population is calculated by two methods. Displacement operation of a master sub-population is calculated as the AFWA (Eq. (4)). As we known, the position of the master sub-population is the best information in the population, therefore, the other master sub-population is added into the AFWA, and its displacement operation is calculated by Eq. (5).

$$\Delta x_i^k = x_i^k + rand(0, A_i) \tag{4}$$

$$\Delta x_i^k = x_i^k - rand(0, A_i) \tag{5}$$

where rand(0, A_i) represents a uniform random number within the amplitude A_i.

At the end of each iteration, the fitness values corresponding to the optimal positions of the two master sub-populations are compared, and the optimal fireworks are retained. The two master sub-populations complement each other and co-evolve to fully extend the search range and mine the useful information in the search domain to reduce the risk of the AFWA falling into the local optimal.

The Algorithm 1 is proposed to generate the explosion sparks with two-master sub-population.

Algorithm 1: Generating explosion sparks with two-master sub-population

```
Initialize the location of the explosion sparks: $X_j = X_i$
Calculate the number of explosion sparks $S_i$
Calculate the explosion amplitude $A_i$
Set $z = rand(1,d)$
For $k = 1:d$ do
  If $k \in z$ then
    If $X_j^k$ is the optimal firework
      $X_j^k = X_j^k + rand(0,A_i)$
      $X_j^k = X_j^k - rand(0,A_i)$
    Else
      $X_j^k = X_j^k + rand(0,A_i)$
    End if
    If $X_j^k$ out of bounds
      $X_j^k = X_{min}^k + |X_j^k| \% (X_{max}^k - X_{min}^k)$
    End if
  End if
End for
```

3.2 New Selection Strategy

In AFWA, it applies a selection method, which is referred to as elitism-random selection method. In this selection process, the optima of the set will be selected firstly. Then, the other individuals are selected randomly. Obviously, this method cannot ensure the diversity of the population. Based on this, this paper proposes a new selection strategy: elitism-tournament selection strategy.

The same as AFWA, elitism-tournament selection also requires that the current best location is always kept for the next iterations. And then, two individuals were randomly selected from the remaining individuals in the population. Each time the individual with the best fitness is placed in the next generation group, the next generation was obtained by repeating $N-1$ times [9].

From the above, we know that the elitism-tournament selection not only maintaining the competitive advantage, but also considering the diversity of the population. This method can maintain the diversity of the population, reflect the better global searching ability.

4 Experiments

4.1 Experiment Settings

Similar to AFWA, the number of fireworks in TMSFWA is set to 5, and the number of mutation sparks is also set to 5, the maximum number of sparks is set to 200 each generation.

In the experiment, the function of each algorithm is repeated 51 times, and the final results after the 300000 function evaluations are presented. In order to verify the performance of the algorithm proposed in this paper, we use the CEC2013 test set [10], including 28 different types of test functions. All experimental test functions dimensions are set to 30, d = 30.

Finally, we use Matlab R2014a software on a PC with a 3.2 GHz CPU (Intel Core i5-3470), and 4 GB RAM, and Windows 7 (64 bit).

4.2 Simulation Results and Analysis

Comparison of TMSFWA with FWA-Based algorithms. To assess the performance of TMSFWA, TMSFWA is compared with EFWA, dynFWA and AFWA, and EFWA parameters set in accordance with [2], AFWA parameters set in accordance with [3], dynFWA parameters set in accordance with [4].

For each test problems, each algorithm runs 51 times, all experimental test functions dimensions are set as 30, and their mean errors and total number of rank 1 are reported in Table 1. The best results among the comparisons are shown in bold. It can be seen that the proposed TMSFWA clearly outperforms among EFWA, AFWA and dynFWA on the test functions.

To clear show the advantages of TMSFWA, the convergence curves of mean objective function value which have great difference in evolution speed are plotted in Fig. 1. Evidently, TMSFWA has better solution accuracy, convergence rate and robustness than all the competitors on majority of cases.

Comparison of TMSFWA with other swarm intelligence algorithms. In order to measure the relative performance of the TMSFWA, a comparison among the TMSFWA and other swarm intelligence algorithms is conducted on the CEC 2013 single objective benchmark suite. The algorithms compared here are described as follows.

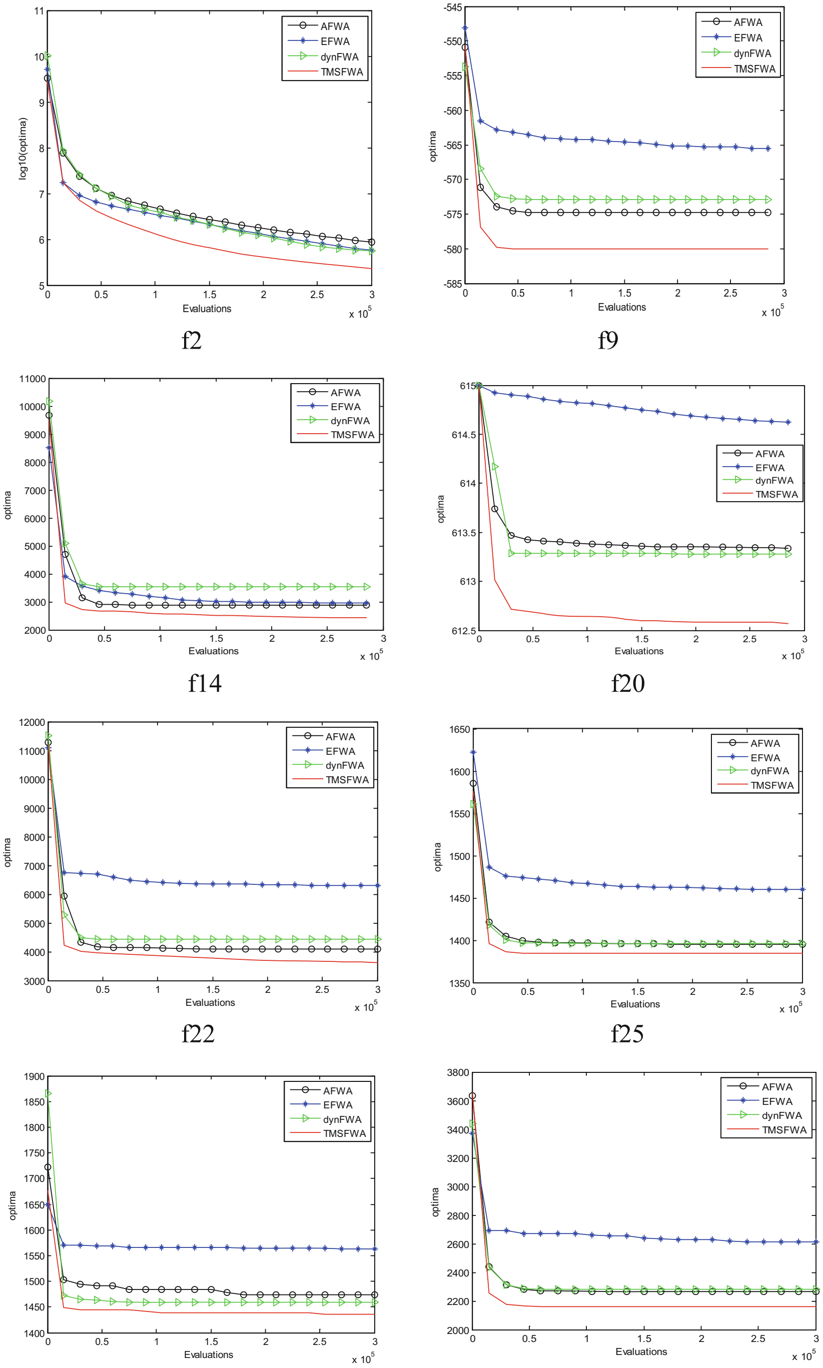
- Artificial bee colony (ABC) [11]: A powerful swarm intelligence algorithm. The results were reported in [12].
- Standard particle swarm optimization (SPSO2011) [13]: The most recent standard version of the famous swarm intelligence algorithm PSO. The results were reported in [5].
- Differential evolution (DE) [14]: One of the best evolutionary algorithms for optimization. The results were reported in [15].

Table 1. Mean errors and total number of rank 1 achieved by EFWA, AFWA, dynFWA and TMSFWA.

Functions	EFWA	AFWA	dynFWA	TMSFWA
	Mean error	Mean error	Mean error	Mean error
f1	7.82E−02	**00E+00**	**00E+00**	**00E+00**
f2	5.43E+05	8.93E+05	7.87E+05	**2.33E+05**
f3	1.26E+08	1.26E+08	1.57E+08	**6.86E+07**
f4	1.09E+00	1.15E+01	1.28E+01	**00E+00**
f5	7.9E−02	6.04E−04	5.42E−04	**00E+00**
f6	3.49E+01	2.99E+01	3.15E+01	**1.2E+01**
f7	1.33E+02	9.19E+01	1.03E+02	**7.7E+01**
f8	2.10E+01	2.09E+01	2.09E+01	**2.09E+01**
f9	3.19E+01	2.48E+01	2.56E+01	**1.99E+01**
f10	8.29E−01	4.73E−02	4.20E−02	**3.00E−02**
f11	4.22E+02	1.05E+02	1.07E+02	**8.45E+01**
f12	6.33E+02	1.52E+02	1.56E+02	**1.22E+02**
f13	4.51E+02	2.36E+02	2.44E+02	**1.95E+02**
f14	4.16E+03	2.97E+03	2.95E+03	**2.53E+03**
f15	4.13E+03	3.81E+03	3.9E+03	**3.77E+03**
f16	5.92E−01	4.97E−01	4.77E−01	**2.9E−01**
f17	3.10E+02	1.45E+02	1.48E+02	**1.19E+02**
f18	1.75E+02	1.75E+02	1.89E+02	**1.68E+02**
f19	1.23E+01	6.92E+00	6.87E+00	**5.68E+00**
f20	1.46E+01	1.30E+01	1.30E+01	**1.25E+01**
f21	3.24E+02	3.16E+02	2.92E+02	**2.97E+2**
f22	5.75E+03	3.45E+03	3.41E+03	**2.82E+03**
f23	5.74E+03	4.70E+03	4.85E+03	**4.6E+03**
f24	3.37E+02	2.70E+02	2.72E+02	**2.54E+02**
f25	3.56E+02	2.99E+02	2.97E+02	**2.84E+02**
f26	3.21E+02	2.73E+02	2.62E+02	**2.35E+02**
f27	1.28E+03	9.72E+02	9.92E+02	**8.6E+02**
f28	4.34E+02	4.37E+02	3.40E+02	**3.1E+02**
	Total number of rank 1			
	0	1	1	28

- Covariance matrix adaptation evolution strategy (CMA-ES) [16]: A developed evolutionary algorithm. The results are based on the code from (https://www.lri.fr/∼hansen/purecmaes.m) using default settings.

The above four algorithms are using the default settings. The comparison results of among ABC, DE, CMA-ES, SPSO2011, and TMSFWA are presented in Table 2, where ‘Mean error’ is the mean error of best fitness value. The best results among



f2 f9

f14 f20

f22 f25

f26 f27

Fig. 1. Convergence curves of the TMSFWA, the AFWA, the EFWA and the dynFWA.

the comparisons are shown in bold. ABC beats other algorithms on 12 functions (some differences are not significant), which is the most, but performs poorly on other functions. CMA-ES performs extremely well on unimodal functions, but suffers from premature convergence on some complex functions. From the Table 3, the TMSFWA ranked the top three (25/28) more than the other four algorithms, and in terms of average ranking, the TMSFWA performs the best among these 5 algorithms on this benchmark suite due to its stability. DE and ABC take the second place and the third place respectively. The performances of CMA-ES and the SPSO2011 are comparable.

Table 2. Mean errors achieved by ABC, DE, CMA-ES, SPSO2011 and TMSFWA.

	ABC	DE	CMA-ES	SPSO2011	TMSFWA
f1	0.00E+00	1.89E−03	**0.00E+00**	**0.00E+00**	**0.00E+00**
f2	6.20E+06	5.52E+04	**0.00E+00**	3.38E+05	2.33E+05
f3	5.74E+08	2.16E+06	**1.41E+01**	2.88E+08	6.86E+07
f4	8.75E+04	1.32E−01	**0.00E+00**	3.86E+04	0.00E+00
f5	0.00E+00	2.48E−03	**0.00E+00**	5.42E−04	0.00E+00
f6	1.46E+01	7.82E+00	**7.82E−02**	3.79E+01	1.2E+01
f7	1.25E+02	4.89E+01	**1.91E+01**	8.79E+01	7.7E+01
f8	2.09E+01	**2.09E+01**	2.14E+01	2.09E+01	2.09E+01
f9	3.01E+01	**1.59E+01**	4.81E+01	2.88E+01	1.99E+01
f10	2.27E−01	3.42E−02	**1.78E−02**	3.40E−01	3.00E−02
f11	**00E+00**	7.88E+01	4.00E+02	1.05E+02	8.45E+01
f12	3.19E+02	**8.14E+01**	9.42E+02	1.04E+02	1.22E+02
f13	3.29E+02	**1.61E+02**	1.08E+03	1.94E+02	1.95E+02
f14	**3.58E−01**	2.38E+03	4.94E+03	3.99E+03	2.53E+03
f15	3.88E+03	5.19E+03	5.02E+03	3.81E+03	**3.77E+03**
f16	1.07E+00	1.97E+00	**5.42E−02**	1.31E+00	2.9E−01
f17	**3.04E+01**	9.29E+01	7.44E+02	1.16E+02	1.19E+02
f18	3.04E+02	2.34E+02	5.17E+02	**1.21E+02**	1.68E+02
f19	**2.62E−01**	4.51E+00	3.54E+00	9.51E+00	5.68E+00
f20	1.44E+01	1.43E+01	1.49E+01	1.35E+01	**1.25E+01**
f21	**1.65E+02**	3.20E+02	3.44E+02	3.09E+02	2.97E+02
f22	**2.41E+01**	1.72E+03	7.97E+03	4.30E+03	2.82E+03
f23	4.95E+03	5.28E+03	6.95E+03	4.83E+03	**4.6E+03**
f24	2.90E+02	**2.47E+02**	6.62E+02	2.67E+02	2.54E+02
f25	3.06E+02	2.89E+02	4.41E+02	2.99E+02	**2.84E+02**
f26	**2.01E+02**	2.52E+02	3.29E+02	2.86E+02	2.35E+02
f27	**4.16E+02**	7.64E+02	5.39E+02	1.00E+03	8.6E+02
f28	2.58E+02	4.02E+02	4.78E+03	4.01E+02	3.1E+02

Table 3. Total number of rank and average rankings.

	ABC	DE	CMA-ES	SPSO2011	TMSFWA
Total number of rank 1	12	5	9	3	8
Total number of rank 2	0	11	3	5	9
Total number of rank 3	3	7	0	8	8
Total number of rank 4	10	3	1	8	3
Total number of rank 5	3	2	15	4	0
Total number of rank	76	70	94	88	62
Average ranking	2.71	2.5	3.36	3.14	2.21

5 Conclusions

TMSFWA was developed by applying two-master sub-population and a new selection strategy to AFWA. We apply the CEC2013 standard functions to examine and compare the proposed algorithm TMSFWA with ABC, DE, SPSO2011, CMA-ES, AFWA, EFWA and dynFWA. The results clearly indicate that TMSFWA can perform significantly better than other seven algorithms in terms of solution accuracy. Overall, the research demonstrates that TMSFWA performed the best for solution accuracy.

References

1. Tan, Y., Zhu, Y.: Fireworks algorithm for optimization. In: Tan, Y., Shi, Y., Tan, K.C. (eds.) ICSI 2010. LNCS, vol. 6145, pp. 355–364. Springer, Heidelberg (2010)
2. Zheng, S., Janecek, A., Tan, Y.: Enhanced fireworks algorithm. In: Proceedings of 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, pp. 2069–2077 (2013)
3. Zheng, S., Li, J., Tan, Y.: Adaptive fireworks algorithm. In: Proceedings of 2014 IEEE Congress on Evolutionary Computation, Beijing, China, pp. 3214–3221 (2014)
4. Zheng, S., Tan, Y.: Dynamic search in fireworks algorithm. In: Proceedings of 2014 IEEE Congress on Evolutionary Computation, Beijing, China, pp. 3222–3229 (2014)
5. Tan, Y.: Fireworks Algorithm Introduction, 1st edn. Science press, Beijing (2015)
6. Gao, H.Y., Diao, M.: Cultural firework algorithm and its application for digital filters design. Int. J. Model. Ident. Control **14**(4), 324–331 (2011)
7. Janecek, A., Tan, Y.: Using population based algorithms for initializing nonnegative matrix factorization. In: Tan, Y., Shi, Y., Chai, Y., Wang, G. (eds.) ICSI 2011. LNCS, vol. 6729, pp. 307–316. Springer, Heidelberg (2011). doi:10.1007/978-3-642-21524-7_37
8. Wen, R., Mi, G.Y., Tan, Y.: Parameter optimization of local-concentration model for spam detection by using fireworks algorithm. In: Proceedings of 4th International Conference on Swarm Intelligence, Harbin, China, pp. 439–450 (2013)
9. Chen, T.: On the Computational Complexity of Evolutionary Algorithms. University of Science and Technology of China, Anhui, China (2010, in Chinese)
10. Liang, J., Qu, B., Suganthan, P., et al.: Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization (2013)
11. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Glob. Optim. **39**(3), 459–471 (2007)

12. M, El-Abd.: Testing a particle swarm optimization and artificial bee colony hybrid algorithm on the CEC13 benchmarks. In: Proceedings of 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, pp. 2215–2220 (2013)
13. Zambrano-Bigiarini, M., Clerc, M., Rojas, R.: Standard particle swarm optimization 2011 at CEC2013: a baseline for future PSO improvements. In: Proceedings of 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, pp. 2337–2344 (2013)
14. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. **11**(4), 341–359 (1997)
15. Padhye, N., Mittal, P., Deb, K.: Differential evolution: performances and analyses. In: Proceedings of 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, pp. 1960–1967 (2013)
16. Hansen, N., Ostermeier, A.: Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In: Proceedings of 1996 IEEE International Conference on Evolutionary Computation, Nagoya, Japan, pp. 312–317 (1996)