Genetic Algorithms with Substitution and Re-entry of Individuals

Ashish Ghosh[*1], **Shigeyoshi Tsutsui**[*2], **Hideo Tanaka**[*3], **and David Corne**[*4]

*1 Machine Intelligence Unit, Indian Statistical Institute, 203 B. T. Road, Calcutta 700 035, INDIA, ash@isical.ac.in

*2 Dept. of Management and Information Science, Hannan University, 5-4-33 Amamihigashi, Matsubara, Osaka 580 JAPAN, tsutsui@hannan-u.ac.jp

*3 Dept. of Industrial Engineering,, College of Engineering, Osaka Prefecture University, 1-1 Gakuen-cho, Sakai , Osaka 593, JAPAN, tanaka@ie.osakafu-u.ac.jp

*4 Dept. of Computer Science, University of Reading, Whiteknights, PO Box 225, Reading RG6 2AY UK, W.Corne@reading.ac.uk

Abstract In this article we introduce two concepts, *substitution* and *re-entry* in Genetic Algorithms (GAs). Substitution replaces the worst individual of the existing population in a generation by a new individual (chosen randomly from the outside of the population) whose fitness value is more than, at least, that of the worst individual. Re-entry, on the other hand, replaces the existing worst individual by an old individual which was dropped out at an earlier generation. Both methods enhance the GA's maintenance of population diversity. The effectiveness of these concepts is demonstrated via experiments on a collection of function optimization problems, and a sequencing problem. The performance of both techniques is found to be significantly superior to that of a standard genetic algorithm. Optimal solutions are found significantly more often, and with significantly fewer evaluations. In particular, the re-entry model is found to be more promising than the substitution method.

Keywords: Genetic algorithms, substitution, re-entry

1. Motivation

Several methods have been examined to enhance the performance of standard Genetic Algorithms (GAs) [1, 2], but there seems to have been little published research on methods which enhance performance by way of the following techniques: *introducing better individuals from outside the population* or *preserving and reintroducing individuals which were dropped out at earlier generations*.

Here we investigate a technique called *substitution*, which introduces fit new individuals from outside the current population at various stages during the search process, and a technique called *re-entry*, which involves re-introducing into the population individuals which were discarded at an earlier stage.

To some extent, the re-entry concept can be viewed as extending the "memory" of the genetic algorithm, whereas the substitution concept can be seen as a rather more direct diversity-injection measure. Both promise to enhance the GA's exploration of the search space. Re-entry *reminds* the GA of previously *lost* individuals which, despite being rejected at an earlier stage, might contain genetic material useful again in the current, possibly quite different areas of the search space currently represented by the population. In some ways, this technique

promises to have a similar effect to *multiploid* methods [3, 4, 5] or *structured* methods [6]. Multiploid methods effectively contain extra unexpressed genes within a single genotype, and a *mask* chromosome which decides which particular genes are expressed at each locus. A multiploid method can thus *hide* recessive genes within a genotype, until they perhaps become useful at later stages of the search process, and are then perhaps rediscovered by fortunate mutations of the mask chromosome. Structured methods amount to a more elaborate way of doing a similar thing, whereby the genotype is a tree-like structure whose operators guide the chromosome interpretation process to a particular selection of the tree's leaves, which then make up the expressed chromosome. Both methods have been found to outperform standard GAs on a range of problems.

In contrast, the substitution concept is partly related to the idea of `re-initialization', which is often employed when a GA has converged. In re-initialization, a common technique is to restart the population largely at random, but seeded by either direct copies of certain chromosomes in the converged generation, or mutants of these. Thus the population is restarted with a mixture of entirely new chromosomes and old ones.

The re-entry method promises to yield the benefits associated with multiploid or structured methods, but without the considerable additional baggage associated with chromosome interpretation and memory usage in these techniques. The substitution method, in contrast, promises, to some extent, to offset the potential need for re-initialization, by way of adequate exploration of the search space *during* search, rather than waiting to inject diversity when convergence has already occurred.

Substitution and re-entry are performed after selection and before crossover & mutation operations. A preliminary version of this work was reported in [7].

2. Substitution and re-entry of individuals

In this section we describe the proposed concepts in more detail.

2.1 Substitution

By *substitution* (*SUB*) we mean replacing the worst individual of the existing population in a generation by a new individual chosen randomly from outside of the existing population. We perform substitution if the fitness value of the new individual is more than at least the fitness value of the (existing) worst one. The concept of substitution owes its origin to natural systems where recruitment of better individuals is always preferred. Substitution itself is performed after selection and before crossover & mutation operations.

It is intuitively evident that the number of substitutions will be more at the beginning of the algorithm, and will decrease with time. Also, the substitution operation is expected to work better for small population sizes where the convergence rate is very high. Since the new individual can only be substituted if it has a fitness value better than the worst individual of the existing population, average population fitness following a successful substitution will obviously increase, and we would also expect performance to benefit from this injection of new and fit individual.

A number of related but different published concepts exist, in addition to the aforementioned `re-initialization' techniques. For example, Cobb and Grefenstette proposed a technique [8] to replace a fraction of the existing population chosen randomly in order to improve the tractability of GAs for changing environment; similarly, multiploid techniques are also usually proposed in the context of changing fitness environments [3]. However, in the present work we concentrate on stationary fitness environments and a particular well-specified *minimal* substitution method, whereby a single worst fit individual is replaced by a better one, taken from the outside of the population, if such can be found with one random trial.

2.2 Re-entry

By *re-entry* we mean replacing the worst individual of the existing population by another individual which was dropped out at an earlier generation. In GAs, popularly used selection techniques are stochastic in nature and the population size is finite; as a result at each generation a fraction of the existing individuals is selected for the next generation. Thus some of the individuals containing "good genetic material" may be lost at any time. In this work we introduce

a concept which `stores' discarded genetic material for later use. It's later re-introduction into the population promises to *kick-start* the GA towards more fruitful exploration of its current environment. Let an individual be dropped (due to selection) at generation g ($g \geq 1$) and then re-inserted into the population at generation $g + gg$

($gg \geq 1$); we define gg to be the *generation gap* for the re-entry model. Let all the individuals that are dropped out at generation g be collected in a temporary pool $tp(g)$. Some of these individuals may be re-introduced at generation $g+gg$ in two ways. We can pick up the best individual of the temporary pool (we call this *best re-entry*) or we can pick up one individual randomly from the temporary pool $tp(g)$ (we call this *random re-entry*).

So as to keep the population size fixed, the re-entered individual replaces the existing worst individual. Replacement also involves two different strategies. In the first strategy, like *substitution*, re-entry is only allowed if the individual which is seeking re-entry is better than at least the existing worst individual. In the second strategy, the re-entry is always permitted. We call these *re-entry with checking* and *re-entry without checking*, respectively.

3. Simulation results and analysis

In the present study the parameters for the GAs were chosen as follows. Population size was kept fixed at 50 and crossover probability as 0.8. We adopted Grefenstette's *scaling window* (=10) concept [9] for fitness scaling, and used two point crossover. The number of simulations performed was 50 in each case. We used Baker's stochastic universal sampling [10] for selection and adopted the elitist strategy (by copying the best member of each generation to replace the worst one of the next generation). The value for the generation gap gg was tuned for each problem, and the best set are as shown in the tables. For the function optimization problems, Gray coded strings were used for chromosomal representation. The number of bits representing a parameter and the maximum n umber of generations were decided depending on the complexity of the problem. Mutation probability was varied so as to optimize the performance of the standard GA (SGA), and this set is reproduced in the tables.

3.1 Test suit

To demonstrate the effectiveness of the proposed concepts the following test suit was considered. The functions tested have a large number of local optima and a single global optimum and have various degrees of complexity. In Fig. 1 we depict a three dimensional view of the functions, so as to have a rough idea of the complexity of the problems being tried.

Davis function : The first function tested was defined by Davis [11] as

$$f_6 = 0.5 - \frac{\{\sin(\sqrt{x^2+y^2})\}^2 - 0.5}{(1.0 + 0.001(x^2+y^2))^2}, \quad x, y \in [-100.0, 100.0].$$

The function f_6 describes a hilly two-dimensional landscape, with an optimal region that occupies a tiny fraction of the total area. The maximum value of this function is 1.0. For the function f_6 mutation probability was 0.008, each parameter was represented by 22 bit; and a maximum of 250000 trials were performed.

Griewank function : The second function is a modified version of Griewank function [12] and is defined as

$$f_g = \frac{1}{4000}\sum_{i=1}^{5} x_i^2 - \prod_{i=1}^{5}\cos(\frac{x_i}{\sqrt{i}}) + 1, \quad x_i \in [-51.2, 51.1].$$

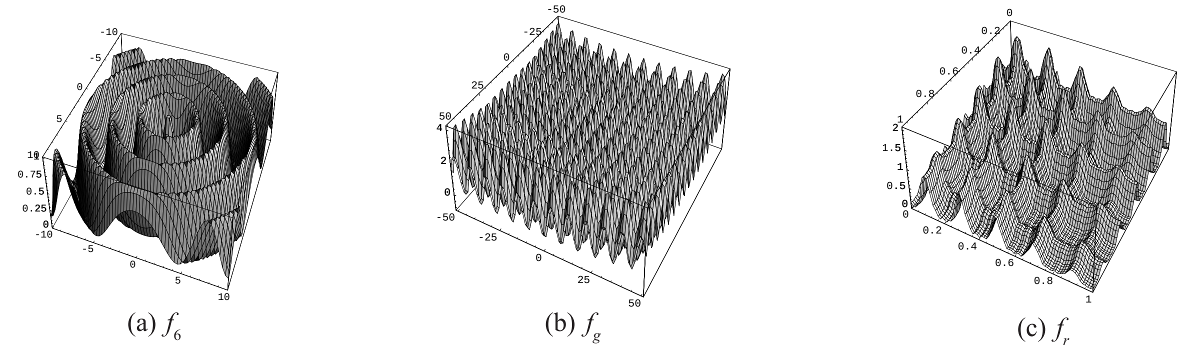
This is also a highly complex multi-modal function with the minimum value 0.0. For the function f_g mutation probability was 0.015 and 10 bits for each parameter, and 1000000 trials in each simulation.

Ripple function : The other function tested is a ripple function [12] and is defined as

$$f_r = \sum_{i=1}^{5} e^{-2\ln 2(\frac{x_i - 0.1}{0.8})^2}(\sin^6(5\pi x_i) + 0.1\cos^2(500\pi x_i)),$$
$$x_i \in [0.0, 100.0].$$

This function f_r has many important sharp peaks of different heights (with one global peak having value 5.5) occupying a small region of the search space; and each of them is surrounded by high frequency of small peaks. Average (over the whole search space) functional value is very small. Here we used a mutation probability of 0.008, coded each parameter with 22 bits, and allowed 500000 trials for each simulation.

Traveling salesperson problem : The traveling salesperson problem (**TSP**) involves finding the shortest Hamil-



(a) f_6 (b) f_g (c) f_r

Fig. 1 A three dimensional view of the functions

tonian cycle in a complete graph of n nodes. The Euclidean distance between any two nodes is computed from their coordinates. An instance of TSP is specified by n, the number of cities, and the coordinates of the n cities. We present here the results for the well known Oliver's 30-city TSP [13], for which the known shortest tour length is 420. We choose this problem as an example of sequencing problems. For this problem we used ordered-crossover (OX) [13] and no mutation operator was used, and the maximum number of trials was set at 1000000.

3.2 Analysis of results

We evaluated the models by measuring *OPT* (number of runs in which the algorithm succeeded in finding the global optimum solution) over the 50 trials; this indicates the *success rate*. Also, we find *MNT*, which is the mean number of trials to find the global optimum for those runs in which it did find the optimum. This reflects the convergence rate for detecting the global optimum. These are recorded in Tables 1 and 2 respectively. In the tables *b-ren* means *best re-entry* & *r-ren* means *random re-entry*, WC means *without checking* & C *with checking*; and *SUB* stands for substitution.

Let us first examine the results of function optimization; and first choose the function f_6. For this case *OPT = 44* (88%) and *MNT = 20627* by the SGA. For the *substitution* technique the results were *OPT = 50* (100%) and *MNT = 19643* (order of 95% of that required by the SGA). From the Tables we notice that different types of re-entry models produced similar results, which are generally comparable to those of the substitution model and better than the SGA. For re-entry, however, *MNT* is sometimes as low as 67% of that of SGA or *SUB*.

Statistical analysis via a standard comparison of population means establishes with better than 95% confidence that the re-entry and substitution models are each superior to the SGA in terms of success rate. We can also report with 95% confidence that the re-entry models are usually superior to the SGA in terms of convergence rate to the global optimum, except in the best-re-entry, with-checking, generation gap 3 case, and the best re-entry, without checking, generation gap 1 case.

Next we consider f_g, *OPT = 5* (10%) and *MNT = 75866* by the SGA. The corresponding results for *substitution* are *OPT =* 12 (24%) and *MNT* = 129972. From Table 1 we notice that the maximum number of success is 18 (36%).

For this function also we see that the success rate is increased significantly, although the *MNT* required is more. This shows that the SGA normally gets stuck in local optima, and even if we continue the algorithm it does not help much. On the contrary, the *re-entry* and *substitution* of individuals helps to escape from these local optima, to a large extent.

Statistical analysis shows with 95% confidence that all re-entry methods and the substitution method are superior to the SGA in terms of success rate, although there are very mixed results in terms of convergence speed.

Now we concentrate on the function f_r. For this function we see that the SGA and the substitution model

Table 1 Number of simulations succeeded in detecting global optima (OPT)

Fu.	*gg*	b-ren		r-ren		SGA	*SUB*
		C	WC	C	WC		
f_6	1	50	50	49	50	44	50
	2	49	50	49	50		
	3	50	50	50	49		
f_g	2	11	11	11	10	5	12
	3	12	18	14	10		
	4	15	15	8	13		
f_r	4	42	41	38	33	33	34
	5	39	39	36	38		
	6	37	38	36	41		
TSP	4	37	36	26	30	13	20
	5	30	31	26	27		

Table 2 Mean number of trials for successful completions (MNT)

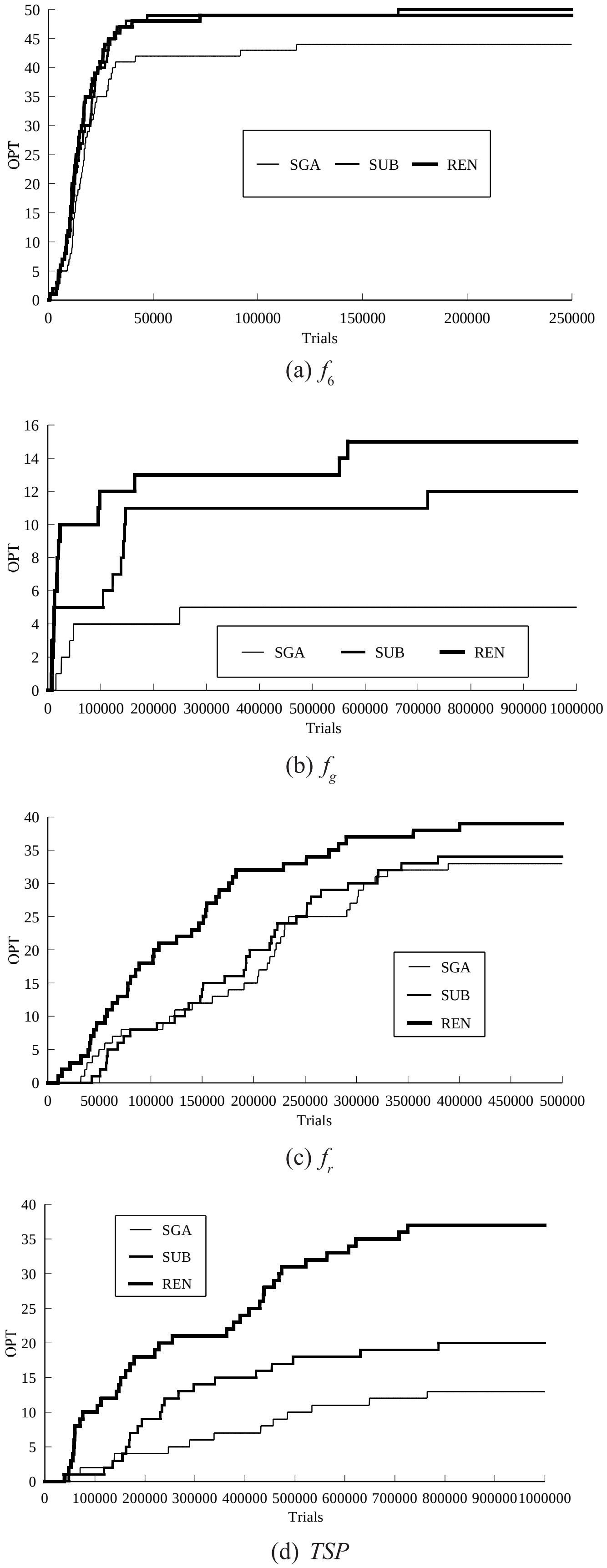
Fu.	*gg*	b-ren		r-ren		SGA	*SUB*
		C	WC	C	WC		
f_6	1	16442	19852	17716	13941	20627	19643
	2	16296	16650	17654	15825		
	3	19094	14433	14670	16032		
f_g	2	196513	137859	52038	176842	75866	129972
	3	238571	142290	212235	136755		
	4	107785	141553	120567	74347		
f_r	4	182455	161627	179940	132692	186993	181673
	5	130150	151133	177539	150913		
	6	171511	165598	173114	162976		
TSP	4	282142	281935	197613	254294	352764	286722
	5	239184	268260	235744	239069		

produced very similar results, in terms of OPT (Table 1) and MNT (Table 2) both. This is possibly due to the following fact. Since the average functional value is small over the whole search space and the global optimum occupies a tiny part of it, probability of introducing better individuals into the existing population becomes small and thus the performance is not enhanced much. The re-entry model on the other hand improves the results as and when it succeeds to re-introduce "good genetic material" which was dropped out due to stochastic selection at an earlier generation. For some other cases the improvement is not so much. It can thus be inferred that the present techniques (specially the substitution concept) may not be of much help for a "needle in the haystack" type problem.

Statistical analysis in this case shows with 90% confidence that the re-entry methods are superior to the SGA or substitution methods, but little can be conclusively said concerning convergence speeds.

Overall, it appears that the substitution and re-entry models generally outperform the SGA for certain complex function optimization problems, while re-entry in particular seems significantly superior in success rate, and occasionally also superior in terms of convergence speed.

Let us now consider the *TSP*. For this case the SGA



(a) f_6

(b) f_g

(c) f_r

(d) *TSP*

Fig. 2 The OPT with restricted numbers of trials and the three algorithmms

succeeded in detecting the best known solution 13 times (26%) only; and the *MNT* was 352764. Under the same environment the *substitution* technique succeeded 20 times (40%) with the *MNT* as 286722; which is much better than the SGA with respect to both success rate and convergence rate. Required number of trials for this case is order of 80% of that of the SGA. Results by the *re-entry* model with a similar environment are much better than the previous two techniques. Success rate in this case is sometimes as high as 74%; and the required mean number of trials is sometimes as low as 56%. This shows that the proposed techniques succeeded much more number of times and converged faster. Scrutiny of the results show that *best re-entry* model showed higher success rate over the *random re-entry* model. Results by *re-entry with checking* and *re-entry without checking* are not that distinguishable. Once again, we notice that the re-entry model performs better than the substitution technique.

Statistical analysis (using T-tests in the convergence rate case due to reduced sample size) confirms these observations, establishing with 95% confidence that the re-entry and substitution models outperform the SGA in terms of both success rate and convergence time.

In Fig. 2 we depict the graphs showing the *OPT* with restricted numbers of trials and the three algorithms. For the *re-entry* model, *best re-entry* with generation gap *gg* at 2 for the Davis function, 4 for the Griewank function, 5 for the Ripple function, and 4 for the TSP problem were chosen for this purpose. From the figures, we can see that for a given number of trials, *OPT* is generally better with the proposed techniques than with the SGA.

From Figure 2(a) we see that the curves are similar for lesser numbers of trials by all the algorithms, but the curves for the proposed algorithms reach a higher level with increasing numbers of trials. This indicates that the proposed techniques detect the global optimum more quickly. This claim is clearly evident from Figures 2(b) and (d). Figure 2(c) shows that *OPT* is better for SGA than the *substitution* model for small numbers of trials, but the situation is reversed as time goes on. However, *re-entry* clearly maintains its upper hand in this case also.

Generally, the tables, statistical analysis, and figures help illustrate that success rate of the proposed algorithms is generally better than that of a standard GA for any given number of trials. In particular, the re-entry model seems particularly strong.

4. Conclusion

We have presented two modifications to standard GAs; the idea of introducing new, relatively fit individuals from the outside world (*substitution*) and re-introducing old individuals which were discarded at an earlier generation (*re-entry*). These methods promise to increase the general degree of exploration of the search space, but without certain overhead associated with related techniques such as multiploidy, structured methods, or reinitialization.

The effectiveness of the proposed techniques has been demonstrated by experiments on some function optimization problems and a sequencing problem. Empirically, we find that GAs with substitution and re-entry operations lead to generally better performance with respect to success rate and convergence rate for these problems. The re-entry model in particular shows much promise, and seems statistically significantly superior to SGA in most cases, and often also to the substitution model. Although the re-entry model has a number of parameters which require tuning, it appears to show robust performance over their variation.

Future work on this topic will concentrate on testing the suitability of such models for non-stationary environments, and further investigation of general design for re-entry based strategies, especially ones which adaptively tune the generation gap, choice-from-store method, and replacement method.

Acknowledgment

A part of this work was done when Dr. Ashish Ghosh held a research fellowship of the Ministry of Education, Science, Sports and Culture, Govt. of Japan (MONBUSHO) at the Osaka Prefecture University. This research is also partially supported by MONBUSHO under Grant-in-Aid for Scientific Research number 10680396.

References

[1] Z. Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs*, Springer-Verlag, Berlin, 1994.

[2] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, MA, 1996.

[3] J. D. Bagley, T*he Behaviour of Adaptive Systems Which Employ Genetic and Correlation Algorithms*, PhD thesis, University of Michigan, 1967.

[4] Y. Yoshida and N. Adachi, A diploid Genetic Algorithm for Preserving Population Diversity - pseudomeiosis GA, In Manner Davidor, Schwefel, editor, *Parallel Problem Solving from Nature: PPSN III*, pp. 36-45, Springer-erlag, 1994.

[5] D. W. Corne, E. Collingwood, and P. M. Ross, Investigating Multiploidy's Niche, *Proceedings of the Evolutionary Computing: AISB Workshop 1996*, Selected Papers, Lecture Notes in Computer Science 1143, pp. 189-197, Springer-Verlag, 1996.

[6] D. Dasgupta and D. R. McGregor, Using Structured Genetic Algorithms for Solving Deceptive Problems, Technical report, University of Strathclyde Department of Computer Science, 1993.

[7] A. Ghosh, S. Tsutsui, H. Tanaka, and D. Corne, Substitution and Re-entry of Individuals in Genetic Algorithms, *Proceedings of the First International Workshop on Frontiers of Evolutionary Algorithms*, pp. 43-46, 1997.

[8] H. G. Cobb and J. J. Grefenstette, Genetic Algorithms for Tracking Changing Environments, *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 523-530, 1993.

[9] J. J. Grefenstette, Optimization of Control Parameters for Genetic Algorithms, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 16, pp.122-128, 1986.

[10] J. E. Baker, Reducing Bias and Inefficiency in the Selection Algorithm, *Proceedings of the Second International Conference on Genetic Algorithms*, pp. 14-21, 1987.

[11] L. Davis (ed.), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.

[12] S. Tsutsui, Y. Fujimoto, and A. Ghosh, Forking GAs: GAs with Search Space Division Scheme, *Evolutionary Computation*, Vol. 5, No. 1, pp. 61-80, 1997.

[13] I. M. Oliver, D. J. Smith, and J. R. C. Hollan, A Study of Permutation Crossover Operators on the Travelling Salesman Problem, *Proceedings of the Second International Conference on Genetic Algorithms*, pp. 224-230, 1987.