

# Using Multiple Offspring Sampling to Guide Genetic Algorithms to Solve Permutation Problems

Antonio LaTorre, José M. Peña, Víctor Robles and Santiago Muelas  
Computer Architecture Department  
Facultad de Informática  
Universidad Politécnica de Madrid, Spain  
{atorre, jmpena, vrobles, smuelas}@fi.upm.es

## ABSTRACT

The correct choice of an evolutionary algorithm, a genetic representation for the problem being solved (as well as their associated variation operators) and the appropriate values for the parameters of the algorithm is a hard task and it is often considered as an optimization problem itself.

In this contribution, we propose a new theoretical formalism, called Multiple Offspring Sampling (MOS). This new technique combines different evolutionary approaches taking advantage of the benefits provided by each of them. MOS dynamically balances the participation of different mechanisms to spawn the new offspring population, according to the benefits provided by each of them in previous generations. This approach evaluates multiple offspring generation methods (for example different coding strategies), and configures appropriate sampling sizes.

This formalism has been applied to a well-known permutation problem, the traveling salesman problem (TSP). The results on several instances of this problem show that most of the combined techniques outperform the results obtained by single ones.

## Categories and Subject Descriptors

I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods and Search—*Heuristic Methods*; I.2.m.c [Artificial Intelligence]: Miscellaneous: Evolutionary Computing and Genetic Algorithms

## General Terms

Algorithms, Experimentation

## Keywords

Genetic Algorithms, Hybrid Evolutionary Methods, Multiple Offspring Sampling, Permutation Problems, Traveling Salesman Problem

## 1. MULTIPLE OFFSPRING SAMPLING

MOS could be defined as a mechanism to create new individuals, i.e., (a) a particular evolutionary algorithm model, (b) with an appropriate coding, (c) using specific operators (if required), and (d) configured with its necessary parameters.

In our algorithm we have redefined the way the offspring and recombination of populations phases of a traditional Evolutionary Algorithm are performed. MOS proposes the definition of multiple mechanisms to generate new individuals, and makes them compete

during the evolutionary process. Each mechanism creates its own offspring  $O_i^{(j)}$  ( $i$  is the generation and  $j$  is the mechanism).

The fitness of the individuals generated is used to evaluate the quality of each recombination technique. The most obvious measure that could be used for this purpose is the average fitness of the population, but more sophisticated measures can be proposed to take into account not only the current performance of the technique but its potentiality. This quality measure dynamically determines the participation of each of the available techniques.

## 2. EXPERIMENTATION

We have tested our approach with three standard datasets for the TSP that can be downloaded from the TSPLIB repository.

Equation 1 presents the fitness function used in our experiments. It has been normalized within the interval  $[0 - 1]$  what makes it easier to interpret how good the results are regardless of the different optimal tour lengths of each instance.

$$\frac{\text{length}(\text{tour}_{\text{best}})}{\text{length}(\text{tour})} \quad (1)$$

The experimentation carried out in this study tries to confirm that MOS approach of combining different genetic codings and variation operators, under some circumstances, can improve the performance of these codings and operators when used separately.

To test our approach, we have considered two different genetic codings: path and ordinal representation which are used simultaneously through all the execution of the algorithm. Both codings allow different variation (crossover and mutation) operators. For ordinal representation, traditional 1-point crossover and uniform mutation described by [3] have been used. For path representation we have considered two crossover operators: Order Crossover (OX) by [2] and Cycle Crossover (CX) [4]; as well as two mutation operators: Repeated Exchange Mutation (REM) by [1] and Simple Inversion Mutation (SIM) by [3].

With all these alternatives we have constructed the following set of techniques: (a)  $t0$ : Integer Coding + OX + REM (b)  $t1$ : Integer Coding + CX + SIM (c)  $t2$ : Real Coding + 1-point Crossover + Uniform Mutation (d)  $t3$ : Integer Coding + OX + SIM (e)  $t4$ : Integer Coding + CX + REM.

### 2.1 Results and Discussion

The results of this experimentation are too extensive to be reported in this paper. Instead of providing a large table with the results of each combination of techniques we propose a mechanism to rank the results obtained for each combination. The procedure to deal with this analysis, for each of the problems, is the following:

1. all the combinations of techniques are considered,

**Table 1: Average Number of Wins Compared with the Number of Techniques**

Number of techniques	Average Number of Wins		
	Swiss42	Brazil58	GR120
1	-14.6	-12.8	-26
2	-3.9	-2.5	-0.8
3	6.3	4	0.4
4	7.8	8.4	21.8
5	10	7	25

**Table 2: Average Number of Wins for Each Technique**

	Number of Wins			Corrected Number of Wins		
	Swiss42	Brazil58	GR120	Swiss42	Brazil58	GR120
t0	4.44	-0.06	5.75	0.74	-3.69	0.35
t1	3.19	7.06	4.94	0.07	6.83	-0.35
t2	-0.69	-0.44	5.37	-2.17	-5.23	0.05
t3	10.56	7.50	5.12	4.02	7.39	-0.27
t4	-2.25	-1.00	5.50	-2.67	-5.23	0.22
Best combination(s) of techniques				t0t2t3	t0t1t2t3	t0t1t2t4 t0t1t3t4
Average p-value against all the other combinations				0.02	0.06	0.06 0.06

- for each combination, 20 independent executions are run until convergence,
- the fitness results obtained for each of the combinations are pairwise compared using a Wilcoxon non parametric t-test. Both fitness and p-values are stored for the following analysis,
- a global analysis is performed for all the combinations. If one combination of parameters is significantly better than another (p-value < 0.01), the winning combination is granted with +1 wins and the losing combination penalized with -1 wins. As all the combinations are compared against each other, they are ranked (depending on how many other combinations are better/worse).

In table 1, the average number of wins obtained by all the single techniques is compared with the average number obtained with 2, 3, 4 and 5 techniques. These numbers show that the higher the number of techniques is the better the average performance. The only exception is Brazil58 dataset, in which 4 techniques combinations have better performance than the 5 techniques together. In all the cases, single techniques perform worse, on average.

Table 2 presents a comparison of the performance of the different techniques. For each technique, the table shows the average number of wins of all the combinations in which it participates. A corrected version of this number of wins is also provided, in this case each technique only counts the proportional part of wins depending on how many techniques take part in the combination, for example if combination *t1t3t4* score 12 wins, each of the three technique receives 12/3 wins. The average of these corrected wins is also weighted by the number of techniques in the combination. In the previous example, the number of wins has weight 1/3 to compute the average. The sum of this corrected average of wins is zero (the sum of all the wins is 0).

The behavior of the different techniques, when they are combined with others, computed by the corrected average number of wins keeps a close relationship with the best combination, but it is

important to highlight the presence of techniques with worse performance in the best combinations. For example, in Swiss42, technique *t2* obtains poor results, but it seems to deeply improve the results of some combinations of techniques (*t0t2t3* is better than *t0t3* with p-value = 0.00431†). In Brazil58, in which *t1* and *t3* outperform other single techniques, the combination of both is improved when *t0* and *t2* are also included (*t0t1t2t3* is slightly better than *t1t3* with p-value = 0.31976).

†represents statistically significant tests (p-value < 0.01). Reader should take into account that transitivity is not guaranteed by this statistical relationship.

In all the cases, the combination of multiple techniques performs better than the best of the single techniques.

### 3. CONCLUSIONS

The general schema of an evolutionary technique carries out an iterative loop of (i) creation of new individuals (offspring spawning), (ii) evaluation of these new individuals, and (iii) selection of the new population from the individuals in the previous and offspring populations. In this contribution we have explored the possibility of using more than one mechanism to create new individuals. This possibility allows us to introduce Multiple Offspring Sampling, a theoretical framework that provides the tools to deal with multiple individual creation methods, even if different coding formats or representations are involved.

MOS is used in this paper to solve one of the most representative permutation problems, the TSP, using a parallel asynchronous genetic algorithm as the evolutionary model. From the results discussed in section 2 we can see that the combination of several offspring techniques tend to improve the results of the genetic algorithms. We can also observe that the combination of techniques reporting the best performance in different TSP instances is not always the same: different instances present different exploratory characteristics that can be exploited by different subsets of techniques. In general, the most suitable individual techniques usually appear in the best combination but, in some cases, the inclusion of an average technique can remarkably improve the overall performance.

Further work will include a deeper analysis of the influence of local search mechanisms in the performance of single techniques and combination of techniques. It would be also interesting to experiment with combinations of techniques where some of them only include one or none variation operators. Finally, we could think of a scenario where the number of available techniques is bigger than the number of techniques being used simultaneously and the algorithm could exchange the worse performing ones by some of the waiting techniques or, even, generate new techniques from the available ones.

### 4. REFERENCES

- [1] W. Banzhaf. The "molecular" traveling salesman. *Biological Cybernetics*, 64:7–14, 1990.
- [2] L. Davis. Applying adaptive algorithms to epistatic domains. In *Proceedings of the 9th IJCAI*, pages 162–164, 1985.
- [3] J. Holland. Adaptation in natural and artificial systems. *University of Michigan Press*, 1975.
- [4] I. Oliver, D. Smith, and J. Holland. A study of permutation crossover operators on the traveling salesman problem. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pages 224–230, Mahwah, NJ, USA, 1987. Lawrence Erlbaum Associates, Inc.