Preventing Premature Convergence to Local Optima in Genetic Algorithms via Random Offspring Generation

Miguel Rocha and José Neves

Departamento de Informática
Universidade do Minho
Largo do Paço
4709 Braga Codex
PORTUGAL
Voice : 351 53 604466/70
Fax: 351 53 604471
mrocha@di.uminho.pt, jneves@di.uminho.pt
http://www.di.uminho.pt/~jneves
http://www.di.uminho.pt/~mrocha

Abstract. The *Genetic Algorithms (GAs)* paradigm is being used increasingly in search and optimization problems. The method has shown to be efficient and robust in a considerable number of scientific domains, where the complexity and cardinality of the problems considered elected themselves as key factors to be taken into account. However, there are still some insufficiencies; indeed, one of the major problems usually associated with the use of *GAs* is the premature convergence to solutions coding local optima of the objective function. The problem is tightly related with the loss of genetic diversity of the *GA*'s population, being the cause of a decrease on the quality of the solutions found. Out of question, this fact has lead to the development of different techniques aiming to solve, or at least to minimize the problem; traditional methods usually work to maintain a certain degree of genetic diversity on the target populations, without affecting the convergence process of the *GA*. In one's work, some of these techniques are compared and an innovative one, the *Random Offspring Generation*, is presented and evaluated in its merits. The *Traveling Salesman Problem* is used as a benchmark.

Keywords: Genetic Algorithms, Genetic Diversity, The Traveling Salesman Problem.

1 Introduction

Since its genesis, and in particular with the work by John Holland [Hol75], the *Genetic Algorithms (GAs)* paradigm has grown either in its areas of application or in its theoretical foundations. Indeed, the number of conferences and journals devoted to the subject is remarkable, as well as the amount of commercial software already available to tackle real-world problems.

In fact, the *GAs* are not too demanding, as could be natural to expect, in terms of their needs of computational power, being applied so far to a wide range of problems, going from the fields of *Combinatorial* or *Numeric Optimization* to *Image Processing* or even *Machine Learning*, showing to be efficient and robust.

However, and in spite of these achievements, the approach still suffers from a number of insufficiencies; or rather, one of the most harmful, and object of attention in this work, is the premature convergence of the *GA* to local optima of the objective function, defined for the target problem.

The significant influence of this anathema at the problem solving level, has promoted the development of techniques in order to overcome such an handicap. In this study, one pursues the purpose to hang in balance a set of such procedures, by evaluating their performance in tasks that may lead to renew the saga of the *Traveling Salesman Problem (TSP).*

One's paper is organized as follows: it starts with a description of the problem at hand, the *TSP*. Then it endorses the *GA* paradigm in terms of its most important features. It continues addressing some techniques to avoid premature convergence to local optima and, finally, foregone conclusions and new directions for future work are reported.

2 The traveling salesman problem

The *TSP* is a classic, well known NP-hard problem in *Combinatorial Optimization*; given a set of n cities, and the costs associated with the travel between each pair, the objective is to find a roundtrip of minimal total cost (or length), visiting each city exactly once.

The problem is stated as a n-dimensional cost matrix of values d_{ij}, where the purpose of the exercise is to obtain a permutation of these values, such that the sum of the costs d_{ij}, for any i and j being i the precedent of j in the sequence, is minimal. There are problems in areas so distinct as *Computer Wiring*, *Wallpaper Cutting*, *Crystallography* or *Job Sequencing* that can be formulated and solved as instances of the *TSP*.

The *TSP* can be defined in terms of an *Integer Linear Programming* procedure and postulates (or restrictions) as follows [Lap91]:

$$
\begin{aligned}
Minimize: \quad & \textstyle\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} && (1)\\
\textit{Subject to}: \quad & \textstyle\sum_{j=1}^{n} x_{ij} = 1, \forall i && (2)\\
& \textstyle\sum_{i=1}^{n} x_{ij} = 1, \forall j && (3)\\
& x_{ij} \in \{0,1\}, \forall i,j && (4)\\
& \textstyle\sum_{i,j \in S} x_{ij} < |S|, \forall S \subset V, S \neq \emptyset && (5)
\end{aligned}
$$

The first equation defines the cost function in terms of the values d_{ij} and of the decision variables, defined as binary variables in (4). When x_{ij} is one, the edge connecting i and j is in the solution, otherwise (x_{ij} is zero) the contemplated edge is not in the solution. The equations (2) and (3) define the constraints of only one edge entering and leaving a given node. In the last equation, V stands for the set of nodes in the instance and $|S|$ for the cardinality of S, defined to be a subset of V. The intended meaning of equation (5) is, therefore, to avoid feasible solutions containing cycles with length smaller than n.

3 The genetic algorithm

In the *GA* used, each individual (or chromosome), that makes the fixed population, codes a *TSP* valid tour. The *genotype* (the genetic constitution of an organism) of the individual is built on a sequence of n integers with no repeated values. The *phenotype* (the physical constitution of an organism as fixed by the interaction of its genetic structure with the environment) is fairly obvious, once the position of any allele on the chromosome determines the order by which the node that it codes is visited. An edge is assumed to connect the nodes that are given by the chromosome's last and first values.

This kind of genotypical representation is named *Order-Based Representation (OBR)*, and is in some ways quite different from the traditional *Binary-Based Representations (BBRs)*, where the order of the genes in the chromosome is not, or should not be, important to its phenotypical interpretation. The constraint on non-duplicates in *OBR*, and the dependence on the order of the genes, justifies the development of a whole new class of operators for the *crossover* and the *mutation* operations.

As far as a *TSP* instance is considered, the evaluation function will assign to each individual in the population a fitness value, a measure of the total cost of the solution coded by its *genotype*; it should also be noticed that the best individuals are the ones with the lowest fitness values.

The major structure of the *GA* used in one's approach is outlined in the pseudo-code of Figure 1, where *ps*, *nc* and *sr* stand, respectively, for the population size, the number of offspring generated per iteration, and the number of individuals replaced when moving from a generation to the next.

3.1 Selection

The *selection* operator is used to choose parents for reproductive trials, to pick the survivors from a generation to the next one, and to decide which of the offspring will be inserted into the population. The procedure used, in this study, is based on a stochastic process using a *Roulette-Wheel* scheme. The weight assigned to each individual is calculated to be inversely proportional to its ranking, considering the fitnesses of individuals on the whole population. The ranking is ascendent when the purpose is to select the best individuals, and descendent when one is looking to the worst of it.

3.2 Crossover

The *crossover* operator is defined based on the function

```
BEGIN
  Initialize time ($t = 0$).
  Generate, at random, $ps$ individuals (initial population $P_0$) and out their evaluation.
  WHILE NOT (end test) DO
    Select from the present population ($P_t$) the individuals.
    Recombine these individuals to breed $nc$ offspring and proceed on their evaluation.
    Select $sr$ offspring to insert into the next population ($P_{t+1}$).
    Select $ps - sr$ survivors from $P_t$ to be inserted into $P_{t+1}$.
    Mutate $P_{t+1}$ and re-evaluate mutated individuals.
    Increase current time ($t = t + 1$).
  END WHILE
END
```

Fig. 1. Structure of the *GA* used

$$Crossover : Individual \times Individual \times Parameters \mapsto Individual[\times Individual]$$

where the ”×” names Cartesian product, the ”[” and ”]” stand for an optional entity, and ”↦” (without danger of ambiguity) names ”↦”.

In this study a number of different *crossover* operators were used, namely the so called *blind* operators, that recombine the genetic material of their ancestors without regard of the underlying solution; i.e., with no links to the target problem. There is also the ones known as *hybrids*, which are associated with operators that take advantage of the problem's specific knowledge; i.e., in the case of the *TSP*, the operators use the information in the cost matrix to guide the ancestors recombination process.

Below is a list of the *blind* operators implemented so far, as well as some references to their genesis [Sta91][RocNev98].

- *Uniform Order Preserving Crossover (UOPX)*
  The operator emphasizes the relative order of the genes in both ancestors, working with a randomly generated binary mask, with a size made equal to the genotype's length. It is the equivalent to the *Uniform Crossover* operator in *BBRs*, and some good results on its application to the *TSP* are being reported [Dav91].
- *EDGe Crossover (EDGX)*
  The edge family of crossover's operators is based on the principle of maintaining all possible pairs of adjacent genes (edges) on the chromosome. It was specially designed for the *TSP* problem [Sta91].
- *Maximum Preservative Crossover (MPX)*
  The *MPX* operator was designed by Mühlenbein [Muh91] with the purpose to tackle the *TSP* by preserving, in the offspring, subtours contained by both ancestors.
- *SCHleuter Crossover (SCHX)*

The *SCHX* [Sch89] is a variation of the *MPX*, with some features similar to the order preserving ones, and also contemplating the process of the inversion of partial tours.

When it comes to the *hybrid* operators, two were considered.

- *Greedy Crossover (GX)*
  The *GX* was introduced by Grenfenstette [Gre85] and it is based on a simple rule, that states that among the existing edges, when recombining information from the ancestors to the offspring, one must choose the ones that carry the minimum cost to the solution, in each step, thus believing that local minimum sub-tours should lead to better global results than the use of the more expensive ones.
- *Half Greedy Crossover (HGX)*
  The *HGX* operator was proposed in [Kur96]. The idea is to make the operator *less greedy*, delaying the convergence process with the purpose of preventing premature convergence to local optima.

3.3 Mutation

A mutation is an unary operator that can be defined as the function

$$Mutation : Individual \times Parameters \mapsto Individual$$

By analogy with what happens in nature, a mutation operation normally induces a small change to the genotype of the individual to which it applies, happening with a frequency, called *Mutation Rate (MR)*; *MR* defines the probability under which a mutation operator is applied, to a particular position of the genotype of an individual, in each iteration of the *GA*.

In this work, four different categories of mutation operators were considered, namely *adjacent swap*, *non-adjacent swap*, *sub-list scramble*, and *partial inversion.*

4 Preventing the premature convergence to local optima

In this section one aims to describe some of the techniques used to prevent the premature convergence to local optima. These methods work on to avoid the loss of genetic diversity of the whole population, and in principle, will not damage the convergence process.

4.1 Adaptive mutation rate

The mutation operator aims to introduce a random component into the search process, with the exploitation of new chunks of the solution space, thus promoting the increase of the genetic diversity of the population; i.e., it is not surprising to find out that one of the first steps to take in order to maintain the genetic diversity in a population is the *MR*'s increase.

However, a high value to this parameter introduces a certain degree of noise into the system, thus creating serious obstacles to the convergence process. Therefore, and in order to overcome this phenomenon, is was decided to change the value of the *MR*, with appeal to an adaptive strategy based on the population's genetic diversity, measured at regular spaces in time, being the standard deviation of the fitness values of the whole population used to estimate its diversity.

The process works as follows: one starts with an initial value for the *MR*, and at regular intervals in time, the value of the standard deviation is tested. If it is lower then a pre-defined limit, the *MR* is increased.

4.2 Social disasters technique

The *Social Disasters Technique (SDT)* was introduced by Kureichick and colleagues [Kur96] in order to avoid the premature convergence to local optima, when the *GAs* are applied to the *TSP*. The general idea is to diagnose the situations of loss of genetic diversity of the population, and in such a case to apply a *catastrophic operator* to it. These operators were defined with the purpose to return the population to an acceptable degree of genetic diversity, by replacing a number of selected individuals, by others, generated at random.

Two different operators were considered.

- *Packing*. Of all the individuals having the same fitness value, only one remains unchanged; all the others are fully randomized.
- *Judgment Day*. Only the individual with the best fitness value remains unchanged; all the others are fully randomized.

4.3 Random offspring generation

One of the features of a population converging to local optima is the large number of individuals sharing the same genetic material. But, when this situation occurs, there is a great probability that the crossover operator may receive as input two individuals with equal genotypes. In this case the recombination of their genetic material will be ineffective, since the offspring bred will simply be clone to their parents.

The idea behind the *Random Offspring Generation (ROG)* is to test the individual's genetic material, before the crossover operation, and if a situation as the one just referred is detected, the operation is not performed. Instead, one offspring, or even two, are randomly generated; i.e., their genotype will code a random solution on the problem's domain.

Two different strategies are possible, differing on the number of the random offspring created. With the former *(1-RO)*, the result is made of a random generated individual being the other one clonally obtained from their parents. With the latter *(2-RO)*, both descendents are randomly bred. When one uses a *hybrid* crossover operator, only the first strategy is applicable, due to the fact that it only generates one offspring per two parents.

5 Experimental results

The techniques described so far were applied to three *TSP* instances taken from the *TSPLIB95* [Rei95]. The problems are listed in Table 1, as well as the values of some of the relevant parameters for the *GA*. The instances referred can be classified as *Euclidean TSPs*, i.e., defined in a way that $\forall i, j, k \; d_{i,j} \leq d_{ik} + d_{kj}$. This is the case of all instances of the problem defined as a set of nodes characterized by their coordinates in a two-dimensional space.

Table 1. The TSP Instances

Problem	Nodes	Optimum	Population Size	Mutation Rates(%)
Eil51	51	430.0	100, 200	0.1 - 1
Eil76	76	545.4	150, 300	0.05 - 0.5
Eil101	101	642.3	200, 400	0.01 - 0.3

In the experiences conducted, several options regarding the policy used to prevent the loss of genetic diversity of the population were considered.

- *None.* No special technique was used to prevent the loss of diversity.
- *Adaptive Mutation Rate (AMR).* As described above considering the initial value of the *MR* to be 0, the value for each increment to be equal to 0.01%, 0.05% or 0.1%, the number of iterations between each test to be 25 or 50 iterations, and the minimum value for the standard deviation to be 1% or 2% of the smallest fitness value in the population.
- *Social Disasters Techniques (SDT)*, being considered two alternatives, the first one using the *Packing* operator *(SDT-P)*, and the latter using the *Judgment Day* operator *(SDT-J)*. A value of 0.5% of the smallest fitness value in the population was used as the minimum limit of the standard deviation, with tests every 100 generations.
- *Random Offspring Generation (ROG)*, being considered *1-RO* and *2-RO* strategies.

The results are given (Tables 2, 3 and 4), for each problem, in terms of the strategy applied to prevent loss of genetic diversity, and of the crossover operator used. The best result was obtained when combinations of other parameters, namely the *mutation rate*, the *population size* and the *mutation operator*, were considered. Each configuration was tested with 20 independent runs, being the result obtained as the average of the fitnesses of the best individuals in each run.

It is now possible to engage into some reflections:

Table 2. Experimental results for the problem *Eil51*

Crossover	Premature Convergence Prevention Technique				
Operator	*None*	*AMR*	*SDT-P*	*SDT-J*	*1-RO*	*2-RO*
UOPX	450.7	458.4	449.9	**441.8**	443.2	450.3
EDGX	443.9	452.6	451.5	445.6	447.0	**441.9**
MPX	438.3	448.9	442.9	442.9	**436.0**	455.9
SCHX	443.5	442.4	443.0	444.4	440.7	**437.8**
GX	434.3	436.5	436.2	435.5	**431.5**	-
HGX	434.1	**429.0**	429.5	429.5	**429.0**	-
Best	434.1	**429.0**	429.5	429.5	**429.0**	437.8

Table 3. Experimental results for the problem *Eil76*

Crossover	Premature Convergence Prevention Technique					
Operator	*None*	*AMR*	*SDT-P*	*SDT-J*	*1-RO*	*2-RO*
UOPX	592.0	**565.7**	578.6	573.5	582.3	580.6
EDGX	**569.3**	592.7	592.9	570.9	574.9	595.3
MPX	**590.1**	595.6	601.2	592.8	599.5	607.0
SCHX	577.8	565.8	569.5	591.7	**564.9**	575.8
GX	554.8	**550.6**	554.3	556.6	551.3	-
HGX	554.1	555.9	551.4	552.4	**551.1**	-
Best	554.1	**550.6**	551.4	552.4	551.1	575.8

- The *AMR* strategy leads to solutions with a similar degree of quality as the ones obtained via regular *GAs*. However, it must be stated that the use of an *AMR* strategy has an obvious advantage, in the sense that it releases the user from having to choose a specific value for the *MR's* parameter, an arduous task in many cases.
- The use of the *SDT* strategy, in either of its forms, does not seem to be an important factor to the improvement of the results so far obtained, although it may induce good solutions in some situations.
- The *ROG* strategy is, undoubtedly, the one that presents the best results, a fact that in itself is not completely surprising, since it is the approach that better takes care of the genetic diversity of the whole population.

6 Conclusions and future work

The data so far obtained, when the *GA* is applied to selected *TSP* instances, shows that the use of artifacts to prevent the loss of genetic diversity in the target population can improve significantly the quality of the results. In particular, the *ROG* strategy seems to be a simple, but powerful method to prevent premature convergence to local optima, and therefore improving the behavior of the *GA*.

Table 4. Experimental results for the problem *Eil101*

Crossover	Premature Convergence Prevention Technique					
Operator	*None*	*AMR*	*SDT-P*	*SDT-J*	*1-RO*	*2-RO*
UOPX	684.6	690.9	685.9	**669.0**	685.5	683.6
EDGX	721.5	715.5	710.4	693.5	**684.8**	708.5
MPX	708.8	716.3	719.9	718.2	711.2	**707.5**
SCHX	690.5	675.4	695.7	690.1	680.2	**674.1**
GX	652.7	664.7	**651.2**	669.8	653.2	-
HGX	646.3	653.4	653.4	654.0	**641.4**	-
Best	646.3	653.4	651.2	654.0	**641.4**	674.1

It must be mentioned that, unlike some other methods (eg. the *Crowding* scheme [DeJ75]), these techniques keep the selection procedures unchanged, and therefore their induced computational overheads may be disregarded.

Obviously, one does not intend to give the final solution to the problem of preventing the premature convergence and loss of genetic diversity that occur when using *GAs*. However, the results obtained by the *ROG's* scheme are encouraging, a reason to extend these tests to other domains, in order to evaluate the real usefulness of the method.

When one considers the way genetic diversity is created in nature, one comes to the conclusion that the spatial organization of the living species is crucial in their process of evolution. Therefore, it will not come as a surprise that the integration of these findings, as well as concepts such as diploid representations and dominance, along with the *GA's* machinery, must be object of consideration in any study that may be carried out on this arena.

References

[Dav91] Lawrence Davis ed. *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.

[DeJ75] Kenneth A. DeJong, An Analysis of the Behavior of a Class of Genetic Adaptive Systems, Doctoral Dissertation, University of Michigan, 1975.

[Gre85] John J.Grenfenstette, Rajeev Gopal, Brian Rosmaita and Dirk Van Gucht, Genetic Algorithms for the Traveling Salesman Problem. In J.Grenfenstette ed. *Proceedings of the Second International Conference on Genetic Algorithms and their Applications*, MIT, Cambridge, July 1987, Lawrence Erlbaum Associates: Hillsdale, New Jersey.

[Hol75] J.H.Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.

[Kur96] V.Kureichick, A.N.Melikhov, V.V.Miaghick, O.V.Savelev and A.P.Topchy, Some New Features in the Genetic Solution of the Traveling Salesman Problem. In Ian Parmee and M.J.Denham eds. *Adaptive Computing in Engineering Design and Control 96(ACEDC'96), 2nd International Conference of the Integration of Genetic Algorithms and Neural Network Computing and Related Adaptive Computing with Current Engineering Practice*, Plymouth, UK, March 1996.

[Lap91] Gilbert Laporte, The Traveling Salesman Problem: An Overview of Exact and Approximate Algorithms, *European Journal of Operational Research*, 59:231-247, 1992.

[Muh91] H.Muhlenbein, Evolution in Time and Space - The Parallel Genetic Algorithm. In G.Rawlins ed. *Foundations of Genetic Algorithms*, pages 316-337, San Mateo, 1991. Morgan-Kaufmann.

[Rei95] G.Reinelt, TSPLIB95. Universitat Heidelberg, 1995.

[RocNev98] M.Rocha and J.Neves, An Analysis of Genetic Algorithms applied to the Traveling Salesman Problem. Technical Report, Universidade do Minho, 1998.

[Sch89] M.G.Schleuter, ASPARAGOS - An Asynchronous Parallel Genetic Optimization Strategy. In J.D.Schafer ed. *Proceedings of the Third International Conference on Genetic Algorithms*, George-Mason University, Morgan Kaufman, 1989.

[Sta91] T.Starkweather, S.McDaniel, K.Mathias, D.Whitley and C.Whitley, A Comparison of Genetic Sequencing Algorithms. In R.Belew and L.Booker ed. *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan-Kaufmann, 1991.