

HYBRIDIZING GENETIC ALGORITHMS WITH BRANCH AND BOUND TECHNIQUES FOR THE RESOLUTION OF THE TSP

C. Cotta*, J.F. Aldana, A.J. Nebro, J.M. Troya

Departamento de Lenguajes y Ciencias de la Computación
Complejo Politécnico (3.2.49), Campus de Teatinos
29071 - Málaga (SPAIN)
{ccottap, jfam, antonio, troya}@lcc.uma.es

* Corresponding author

Abstract - In this paper some mixed techniques are outlined in order to combine the advantages of two very different methods for the resolution of combinatorial optimization problems (*Genetic Algorithms* and *Branch and Bound Techniques*), simultaneously avoiding their drawbacks. Due to the disparity between the basic techniques it is not suitable that they work at the same level so two models have been developed in which each technique assumes the role of being a tool of the other one. Parallelism is an important issue in those techniques.

1. INTRODUCTION

Both *Genetic Algorithms* (GAs) and *Branch and Bound Techniques* (B&B) are heuristic procedures for the resolution of optimization problems [1] but they are very different in their philosophy and in the way they work. A first-look comparison between them shows the typical computational dilemma space vs. time. GAs realize a continued treatment on a finite (and small) subset of the whole work-area. This process may require a long time before acceptable solutions are found (obtaining the optimal is not guaranteed). On the contrary, B&B techniques perform an intelligent exploration of the work-area. This exploration is relatively fast (when compared with GAs) but, as the size of this area grows, requires great amounts of memory. However, these techniques do guarantee achieving the optimal solution.

A question rises from what above stands: is it possible to develop a hybrid technique which combines the advantages of both approaches and, simultaneously, avoids or minimizes their drawbacks, thus constituting a synergic system?. The existence of such a technique is the subject of this paper, in which some possibilities to achieve that integration are studied.

The *Travelling Salesman Problem* (TSP) has been chosen as training-problem.

2. COMBINATION OF GENETIC ALGORITHMS AND BRANCH & BOUND

2.1. Direct collaboration. Issues

A first possibility for GAs and B&B techniques to work together is executing both procedures independently and in parallel, that is, at the same level. Both processes will enclose the solution but coming from different directions. There are two ways of obtaining a benefit of this parallel execution (Fig 1):

- Use the upper bounds provided by the GA to purge the problem queue of the B&B algorithm, deleting those subproblems whose lower bound is bigger than the one obtained from the GA.
- Inject into the GA information about those subtours the B&B algorithm considers more promising. They would be the *building blocks* of the TSP representation.

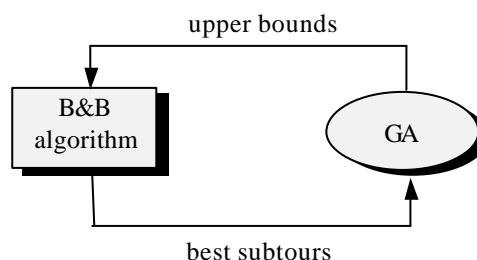


Fig 1. Direct collaboration scenario

These options are, however, difficult to carry out for many reasons:

- Execution times are quite different, so establishing an effective information exchange is very difficult. For example, injecting some high-valued building blocks in early stages of the GA could yield the

apparition of superindividuals [2], affecting diversity and polarizing further evolution.

- There is a big gap between the accuracies of both methods. Specifically for the TSP, B&B algorithms start from an initial approximation that is usually within a 2% of the optimum [3], whilst GAs do not usually achieve a high accuracy until late stages of evolution, so it is too late for a B&B algorithm to get a benefit by purging the problem queue.

Clearly, it is necessary to find an alternative approach in order to establish a productive collaboration between both techniques. According to the functioning features mentioned, there exist at least two ways of harmonizing their performances: on the one hand, it is possible to reduce the combinatorial explosion which limits the effectiveness of a B&B algorithm by imposing some restrictions to the problem. On the other hand, parallel execution of many GAs may heighten their power.

2.2. Branch and Bound working for a Genetic Algorithm

In the model described next, the GA plays the role of a master and the B&B algorithm is incorporated as a tool of it.

2.2.1. Hybridizing a Genetic Algorithm The key feature of a GA is its robustness: binary representation and simple binary crossover and mutation operators provide problem-domain independence. However, such an algorithm will be normally outperformed by a specialized algorithm.

It is possible, however, to improve a GA performance applying some techniques and knowledge used by other algorithms, although this implies a loss of robustness. In [2] some guidelines in order to incorporate those items are described.

2.2.2. A Hybrid Genetic Algorithm to solve the TSP. A promising way of hybridizing a GA incorporating B&B techniques and knowledge is to build a hybrid operator with the philosophy of B&B. The idea underlying this option is shown next (see Fig 2).

All information related to the goodness of a certain tour is contained in the edges between nodes. The classical crossover operators usually introduce a great amount of new information since they produce offsprings including edges not appearing in any ancestor. Some advanced operators like *Edge-Recombination* [4] guarantees that an offspring will inherit a 95% of its genetic information from its ancestors. The operator described next assures that a unique offspring will be created using only edges from the ancestors (i.e. without any implicit mutation).

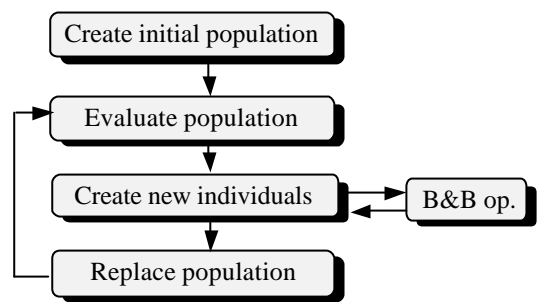


Fig 2. Outline of the Hybrid Genetic Algorithm.

The functioning of this operator is as follows: the B&B algorithm starts from a list of all edges in the graph with information related with associated weights. This operator is a restricted version of the B&B algorithm in which only the edges that appear in the ancestors are taken into account. The offspring obtained will not only 100% composed by its ancestors' edges but it will also be the best solution that can be build using those edges.

The net effect of this operator is to link low-cost subtours existing in the ancestors, or their construction. In fact, considering these elements as the building blocks of the problem the B&B algorithm would build them and glue together (injecting them into the GA as mentioned in 2.2).

2.2.3. Using the hybrid operator. Issues. It is clear that the restricted TSP solution carried out by the described operator is faster and does not have the memory limitations of the base case. However, this operator is going to be used a high number of times and its computational complexity is much higher than other operators such as edge-recombination, so execution times of a GA using it intensively might reach untenable values.

A first preventive measure in order to decrease the computational effort of the algorithm is a combined use of many crossover operators. Since the hybrid-operator complexity is determined by the number of edges taken into account, its use could be restricted to those occasions in which the ancestors share a good amount of edges, using a standard operator in any other situation.

The described model has not only the advantage of reducing execution time but it also mitigate another drawback of the operator, the greediness of its heuristic functioning. This model resembles in some way a thermodynamic system: in early stages temperature (disparity between individuals) is high and the standard operator is often used, thus grouping building blocks. As the system freezes the heuristic operator becomes more used. In those late stages the probability of an important block not being carried due to the greediness of the operator is lower.

Another less desirable side effect of this operator is the tendency to a fast convergence that induces in the GA, due to apparition of superindividuals. This effect can be relieved introducing a higher mutation rate, with the risks that come with it.

2.2.4. Improving the model. Parallelism. In spite of the use of the described model, execution times (although decreased) are still high so another strategy must be used to optimize the process. One of the most suitable options is parallelizing it. There exists some alternatives in order to carry out that parallelization. The following ones can be mentioned:

- Parallelizing the hybrid operator [5]: this option would only reduce the execution time of the GA, remaining unaltered its behaviour.
- Overlapping the hybrid operator with the rest of the GA (Fig 3): this option consists of the parallel launching of many hybrid crossover operations. The GA continues its functioning without waiting for these operations to be concluded. As they finish the generated offsprings are inserted into the population. This option do introduce qualitative behaviour differences in relation to the sequential version. The main one is the absolute disappearance of the *generation* concept since a great gapping [6] occurs (individuals resulting from crossover operations launched many iterations before are inserted into the population).

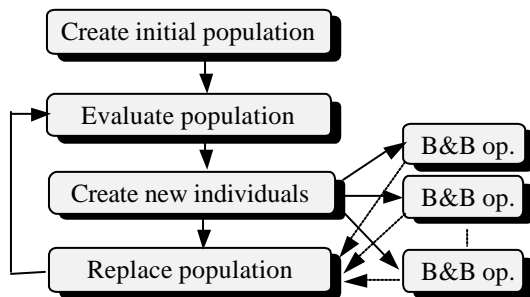


Fig 3. Outline of the Asynchronous Hybrid Genetic Algorithm (AHGA).

It is possible that this overlapping provides a positive side effect related with the problems caused by superindividuals mentioned in 2.2.3: inserting them in the population many evolutionary cycles later may allow average fitness to increase enough not to suffer (or to suffer less) because of the presence of those individuals.

2.3. Branch and bound and parallel evolution

The other way to close up characteristics of B&B techniques and GAs is to heighten the power of the later ones. This heightening can be achieved by means of parallelly executing many GAs. These GAs can independently execute or establish some kind of collaboration between them, periodically interchanging a part of their populations [7].

2.3.1. Describing the model. In the simplest scenario, a B&B algorithm executes in parallel with a certain number of GAs (Fig 4). Many GAs executing independently will generally produce solutions of similar magnitude but different structure, so the problem queue will seldom be purged.

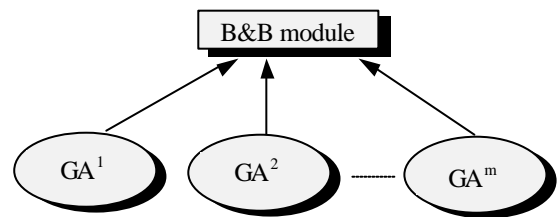


Fig 4. Outline of the parallel evolution model

Notice one of the main problems of the B&B algorithm when applied to the TSP: a high number of edges must be considered. The complexity of the algorithm can be decreased by reducing this number of edges. In order to ignore “*bad*” edges, information from the GAs can be used.

The model outlined above can be described as follows. There is a generalized hybrid operator that recombines n individuals. Such operator is placed in a master process that controls m GAs, where $n = k \cdot m$. Periodically, every GA send to the operator k individuals (e.g. the best k). The probable diversity in the structure of individuals provided by independent GAs will contribute to make that almost every edge suitable to be part of the optimal solution be included in any individual, and many non-suitable edges will not be taken into account. Achieving the optimum is anyway not guaranteed.

2.3.2. Variations and improvements. There exist many variations that take as reference the strategy above. two fairly significant ones are:

- Perform an information exchange between GAs as stated in the migratory models described in [7].
- Take $n = k \cdot m + 1$, that is, send to the hybrid operator not only the individuals provided by the GAs but also the best obtained so far. This is the B&B counterpart of GA elitism.

- Inject into the GAs the individual created by the hybrid operator. This is a dangerous action since it may cause convergence problems.

3. RESULTS AND CONCLUSIONS

3.1 Results

The set of experiments realized so far has not been very large and hence only a first impression of the behaviour of each technique has been obtained. All experiments have been carried out with randomly generated euclidean problems of size 75 and 100 running over a heterogeneous network of SUN workstations.

As a first stage, every problem was solved using *pure* GAs and B&B algorithms. The so obtained results will be used as reference for further experiments.

Next, some experiments were realized with the Synchronous Hybrid Genetic Algorithm described in 2.2.2. Although a small accuracy improvement takes place (about 10% reduction in the mean error), execution times are too large (two magnitude orders larger) so only a few experiments were finished.

Subsequently, a model based on the Asynchronous Hybrid Genetic Algorithm was tested. At this point, the results were a little bit shocking as no apparent regular behaviour was observed. In fact and depending upon the test-problem, variable accuracy improvements were achieved (including worse results as those of the *pure* GA, mainly with a 1GA+1B&B model) using the same parameter configuration.

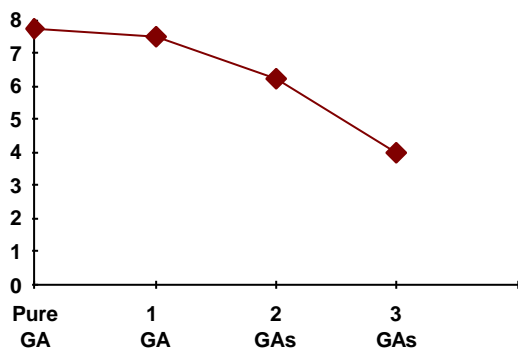


Fig 5. Mean error in some executions of the Parallel Evolution model on a 75-city problem.

Finally, the experiments focused of the model outlined in 2.3.2 including the variations described in 2.3.3 (except inter-GA communication). This model shows the best results. Moreover, its behaviour appears to be more

predictable as shown in Fig 5, where it can be seen that the mean error decreases as the number of parallel GAs increases. Specifically, the best improvements achieved were 48% for a 75-city problem (that is, halving the mean error) and 61% in a 100-city problem.

3.2 Conclusions

The results above should be carefully studied since they are only preliminary results. For example, it is possible that an AHGA exhibits a more predictable behaviour with a high number of B&B modules running in parallel. Experiments must also be done with larger problem instances in order to confirm tendencies. Anyway, if a conclusion is to be extracted of obtained results, we can say that hybridization of GAs with B&B techniques will normally improve the quality of results when compared with more classical approaches.

REFERENCES

1. Reeves C.R., *Modern Heuristic Techniques for Combinatorial Problems*, Blackwell Scientific Publications, Oxford, 1993
2. Davis L., *Handbook of genetic algorithms*, Van Nostrand Reinhold Computer Library, New York NY, 1991
3. Volgenant T. and Jonker R. "A branch and bound algorithm for the symmetric traveling salesman problem based on the 1-tree relaxation.", *European Journal of Operations Research* 9:83-89, 1982
4. Michalewicz Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Springer Verlag, Berlin, 1992
5. Nebro A.J. "Implementation and Evaluation of Parallel Schemes for Branch and Bound Techniques" (in spanish), MSc Thesis, Univ. of Málaga, 1991
6. Alba E. and Aldana J.F. "Genetic Algorithms as Heuristic in Optimization Problems. An Overview" (in spanish), Technical Report, Dept. of Lenguajes y Ciencias de la Computación (Univ. of Málaga), 1992
7. Suárez V. "Parallel Genetic Algorithms for Combinatorial Problems over Shared-Memory Mutiprocessor Systems" (in spanish), MSc Thesis, Univ. of Málaga, 1994