PERFORMANCE EVALUATION OF SELECTION METHODS TO SOLVE THE JOB SHOP SCHEDULING PROBLEM

Stark N., Salto C., Alfonso H.
Proyecto UNLPAM-09/F015[1]
Departamento de Informática - Facultad de Ingeniería
Universidad Nacional de La Pampa
Calle 110 esq. 9
(6360) General Pico – La Pampa – Rep. Argentina
e-mail: { stark, saltoc, alfonsoh }@ing.unlpam.edu.ar
Phone: +54 2302 422780/422372, Ext. 6302

Gallard R.
Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)[2]
Departamento de Informática
Universidad Nacional de San Luis
Ejército de los Andes 950 - Local 106
(5700) - San Luis -Argentina
e-mail: rgallard@unsl.edu.ar
Phone: +54 2652 420823
Fax : +54 2652 430224

ABSTRACT

In evolutionary algorithms selection mechanisms aim to favour reproduction of better individuals imposing a direction on the search process. It does not create new individuals; instead it selects comparatively good individuals from a population and typically does it according to their fitness. The idea is that interacting with other individuals (competition), those with higher fitness have a higher probability to be selected for mating. In that manner, because the fitness of an individual gives a measure of its “goodness”, selection introduces the influence of the fitness function to the evolutionary process. Moreover, selection is the only operator of genetic algorithm where the fitness of an individual affects the evolution process. In such a process two important, strongly related, issues exist: selective pressure and population diversity.

In this work we are showing the effect of applying different selection mechanisms to a set of instances of the Job Shop Scheduling Problem, with different degrees of complexity. For these experiments we are using multiplicity features in the selection of parents for the reproduction with the possibility to generate multiple number of children too, because the results using these approaches outperform to those obtained under traditional evolutionary algorithms. This was shown in our previous works.

A description of each method, experiments and preliminary results under different combinations are reported.

[1] The Research Group is supported by the Universidad Nacional de La Pampa.

[2] The LIDIC is supported by the Universidad Nacional de San Luis and the ANPCYT (National Agency to Promote Science and Technology)

1. INTRODUCTION

By simulating evolution, an Evolutionary Algorithm (EA) maintains a population of individuals (*chromosomes*) which evolve throughout generations by reproduction of the fittest individuals. After initialisation, to create the original population of individuals, an EA consists of a selection-recombination-mutation cycle until a termination criterion holds.

Selection, crossover and *mutation* are the main operators repeatedly applied throughout the EA execution used to modify individual features. So, it is expected that evolved generations provide better and better individuals (searchers in the problem space).

A well known property of a selection operator is *selective pressure* which can be defined as the probability of the best individual being selected relative to the average probability of selection of all individuals.

During the selection step of an EA, copies of better ones replace worst individuals. Consequently, part of the genetic material contained in these worst individuals disappears forever. This *loss of diversity* is defined as the proportion of the population that is not selected for the next generation [3].

When the selection mechanism imposes a strong selective pressure then the loss of diversity can be high and, to prevent a premature convergence to a local optimum then, either a larger population size or adequate crossover and mutation operators are needed. On the other side of the coin a small selective pressure can excessively slow the convergence rate.

In the last years a variety of evolutionary schedulers based on genetic algorithms have been reported in the literature. In general, the task of scheduling is the allocation of jobs over time when limited resources are available, where a number of objectives should be optimized, and several constraints must be satisfied. A job is determined by a predefined set of operations, and the result of a scheduling algorithm is a schedule that contains the start times and allocation of resources to each operation [2].

Improvements in evolutionary algorithms have been found by using a multiplicity feature [6, 7, 8, 9], which allows multiple recombination on multiple parents. A multiplicity approach is M*ultiple crossovers on multiple parents* (MCMP) [8,9] which supplies a balance in exploitation and exploration because the searching space is efficiently exploited (by the multiple application of crossovers) and explored (by a greater number of samples provided by multiple parents). MCMP provides a means to exploit good features of more than two parents selected according to their fitness by repeatedly applying one of the *scanning crossover* (SX) variants [4,5]: a number n_1 of crossovers is applied on a number n_2 of selected parents. From the n_2 produced offspring a number n_3 of them are selected, according to some criterion, to be inserted in the next generation. This is the method used in all our experiments.

MCMP was successfully applied to single and multicriteria optimisation and to the *job shop scheduling problem* (JSSP) under various representations, such as: decoders, dispatching rule representation and operation based representation [11,12, 13, 14, 15]. It was shown that a greater number of crossovers for a given number of parents provide better results. Another representation is *job based representation,* which consists of a list of jobs and a schedule is constructed according to the sequence of jobs. Here we deal with permutations, and then adequate genetic operators should be used. Due to the use of permutations in the chromosome representation and aiming to apply MCMP, we considered the modified SX methods [5] in order to produce feasible offspring. Furthermore, another crossover method, specially designed for this kind of representation, is the *Adjacency Based Crossover* (ABC). For the two crossover methods we used the variants *Uniform* (U-XX) (all the parents have the same chance to be donors) and *Occurrence-Based* (OB-XX) (the gene values is selected according to the value occurrence).

In next sections, the selection mechanisms used are described, and finally implementation details and some previous results are presented.

2. COMMONLY USED SAMPLING MECHANISMS

For the following discussion we concentrate on EAs applied to search (optimization) problems. Here it is convenient to adopt the notation used by Bäck [1]. Let us call *I* the space of individuals $a \in I$ and $f : I$

$\to \Re$ a real-valued fitness function (maximization and minimization are equivalent). Let be μ the population size and $P(t) = (a_1^t ..., a_\mu^t) \in I^\mu$ a population at generation t.

2.1.Proportional Selection

In proportional selection, an individual a_i is chosen at random for mating from a population of size μ according to the following probability:

$$P_{sel}(a_i) = \frac{f(a_i)}{\sum_{j=1}^{\mu} f(a_j)}$$

This is the simplest selection scheme also known as *roulette-wheel selection* or *stochastic sampling with replacement.*Here, individuals are mapped to contiguous segments in the real interval [0,1] in such a way that a segment corresponding to an individual has a size equal to the individual fitness. Then a random number in such interval is generated and the individual whose segment encompasses the random number is selected.

2.2.Rank-based selection

A first approach was called *linear ranking,* where the selective pressure can be controlled more directly than using the proportional selection and consequently the search process can be accelerated remarkably. Whitley [16] pointed out that ranking acts as a function transformation assigning a new fitness value to an individual based on its performance relative to other individuals. The Baker's original linear ranking method assigns a selection probability that is proportional to the individual's rank. Here, according to Bäck [1] the mapping *rank:* $I \to \{1,...,\mu\}$ is given by:

$$\forall i \in \{1,...,\mu\}: rank(a_i) = i \Leftrightarrow$$
$$\forall j \in \{1,...,\mu-1\}: f(a_j) \leq\geq f(a_{j+1})$$

where $\leq\geq$ denotes the $\leq$ relation or the $\geq$ relation for minimization or maximization problems, respectively. Consequently the index i of an individual a_i denotes its rank. Hence, individuals are sorted according to their fitness resulting a_1 the best individual and a_μ the worst one. Assuming that the expected value for the number of offspring to be allocated to the best individual is $\eta_{max} = \mu P(a_1)$ and that to be allocated to the worst one is $\eta_{min} = \mu P(a_\mu)$ then

$$P_{sel}(a_i) = \frac{1}{\mu}\left(\eta_{max} - (\eta_{max} - \eta_{min}) \cdot \frac{i-1}{\mu-1}\right)$$

As the following constraints must hold

$$P_{sel}(a_i) \geq 0 \quad \forall i$$
$$\sum_{i=1}^{\mu} P_{sel}(a_i) = 1$$

it is required that: $1 \leq \eta_{max} \leq 2$ and $\eta_{min} = 2 - \eta_{max}$

The selective pressure can be adjusted by varying η_{max}. As remarked by Baker [2] if $\eta_{max} = 2.0$ then all individuals would be within 10% of the mean and the population is driven to convergence during every generation. To restrain selective pressure, Baker recommended a value of $\eta_{max} = 1.1$. This value for η_{max} close to 1 leads to $P_{sel}(a_i) \cong 1/\mu$, almost the case of random selection.

2.3.Tournament Selection

In tournament selection q individuals are randomly chosen from the population and then the best fitted individual, designated as the winner, is selected for the matting pool. The parameter q is known as the tournament size and usually it is fixed to $q = 2$ (binary tournament). If $q = 1$ then there is no selection at

all: each individual has the same probability to be selected. As long as q increases the selective pressure is augmented.
As Blickle [3] affirms, tournament selection can be implemented efficiently having the time complexity $O(\mu)$ because no sorting of the population is necessary but, as a counterpart, this also leads to high variance in the expected number of offspring resultant from μ independent trials.
As showed by Bäck [1], the selection probability for individual a_i , $(i \in \{1,...,\mu\}$) for *q-tournament* selection is given by

$$P_{sel}(a_i) = \frac{1}{\mu^q}\left((\mu - i + 1)^q - (\mu - i)^q\right)$$

2.4.Stochastic Universal Sampling (SUS)

The idea, introduced by Baker [2], is to make a single draw from a uniform distribution and use it for determining the exact number of copies from each parent.
In this method the individuals are mapped to contiguous segments of a line, such that each individual's segment is equal in size to its fitness exactly as in proportional selection. Here equally spaced pointers are placed over the line as many as individuals have to be selected. Consider n the numbers of the individuals to be selected, then the distance between the pointers are $1/n$ and the position of the first pointer is given by a randomly generated number in the range [0, $1/n$].

3. EXPERIMENT DESCRIPTIONS

The experiments were developed for a set of selected instances of the Job Shop Scheduling Problem (JSSP), under permutation representation. For each experiment series of fifty runs was performed. Experiments corresponded to different (n_1,n_2) combinations, crossover methods and selection mechanisms. Elitism to retain the best-valued individual was implemented. The population size was fixed at 100 individuals. USX, OBSX, OBABC and UABC were implemented and for insertion in the next generation the best child was chosen ($n_3 = 1$). Number of crossovers and parents were set to: $1 \leq n_1 \leq 4$ and $3 \leq n_2 \leq 5$, respectively. For mutation an *interchange* operator was used. The algorithms evolved for a minimum of 500 generations, after that a control of the mean fitness population progress began: if this value remained within a determined range for 20 consecutive generations the algorithm stops. Probabilities for crossover and mutation were fixed at 0.7 and 0.2, respectively. These values were determined as the best combination of probabilities after many initial trials. Proportional selection and SUS were applied in the conventional way. In the case of raking selection, setting value of η_{max} was 1.1. In the case of tournament selection, the size q of the set of competing individuals was fixed to 2. Four instances [10], with known optimal makespan were used: *la01*, *la06*, *la12* and *la15*.

4. PRELIMINARY RESULTS AND CONCLUSIONS

This contribution explained the behaviour of multirecombinative approaches (MCMP) and selection methods used to solve the JSSP for a set of instances of distinct complexity using a permutation representation.
After a long series of experiments a general overview indicates, independently of the crossover method used, a better performance of uniform approaches than occurrence approaches for any (n_1,n_2) association. Particularly, comparing results obtained on both USX and UABC, the first one exhibits a better quality of results.
Contrasting the different selection method the values achieved, show a better performance for both SUS and Proportional Selection over the other methods (without a clear conclusion on which one of these outperform the other).
In general, when 4 crossovers are applied better results than when a single crossover is applied are obtained.

The promising results on the smallest instances encourage us to deep investigation. To improve results in larger instances, further work include self adaptation of parameters such as (n_1,n_2) associations and study the adjustment of selection method according to the evolution.

5. REFERENCES

[1] Bäck T: *Evolutionary algorithms in theory and practice*. Oxford University Press, 1996.

[2] Baker J. E.: *Reducing bias and inefficiency in the selection algorithm*. Proc. of the 2nd International Conf. on Genetic Algorithms, pp. 14-21. J.J Grefenstette Ed, 1987.

[3] Blickle T., Thiele L.: *A comparison of selection schemes used in genetic algorithms*. Technical Report 11 TIK, Swiss Federal Institute of Technology, December 1995.

[4] Eiben A.E., Van Kemenade CHM, Kok J.N.: *Orgy in the computer: multi-parent reproduction in genetic algorithms*, in Moran F., Moreno A., Merlo J.J., and Chacon P., editors, Proc. of the 3rd European conference on Artificial Life, number 929 in LNAI, pp. 934-945. Springer Verlag, 1995.

[5] Eiben A.E., Bäck T.: *An empirical investigation of multi-parent recombination operators in evolution strategies*, Evolutionary Computation, 5(3), pp.347-365, 1997.

[6] Esquivel S., Leiva A., Gallard R.: *Multiple crossover per couple in genetic algorithms*. Proc. of the 4th IEEE International Conf. on Evolutionary Computation (ICEC'97), pp 103-106, Indianapolis, USA, April 1997.

[7] Esquivel S., Leiva A., Gallard R.: *Couple Fitness Based Selection with Multiple Crossover Per Couple in Genetic Algorithms.* Proc. of the International Symposium on Engineering of Intelligent Systems, University of La Laguna, España, Tenerife, Vol. 1, pp. 235-241, ISBN 3-906454-12-6, Feb. 1998.

[8] Esquivel S., Leiva A., Gallard R.: *Multiple Crossovers between Multiple Parents to Improve Search in Evolutionary Algorithms*, en Proc. of the 1999 Congress on Evolutionary Computation (IEEE), vol. 2, pp. 1589-1594, Washington DC., 1999.

[9] Esquivel S., Leiva A., Gallard R.: *Multiplicity in Genetic Algorithms to Face Multicriteria Optimization*, Proc. of the 1999 Congress on E. C., IEEE Service Center, pp. 85-90, Washington, D.C., 1999.

[10] Lawrence S.: *Resource Constrained Project Scheduling: an Experimental Investigation of Heuristic Scheduling Techniques*, (Supplement), Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1984.

[11] Minetti G., Salto C., Alfonso H., Gallard R.: *A Comparison of two Multirecombinated Evolutionary Algorithms for the Job Shop Scheduling Problem*, Proc. VI Congreso Argentino de Ciencias de la Computación, CACIC 2000 Universidad Nacional de la Patagonia San Juan Bosco, 2000.

[12] Salto C., Alfonso H., Gallard R.: *Multiplicity and Incest Prevention in Evolutionary Algorithms to Deal with de Job Shop Problem*, Proc. of the Second ICSC Symposium on Engineering of Intelligent Systems, University of Paisley, Scotland, pp. 451-457, 2000.

[13] Salto C., Minetti G., Alfonso H., Gallard R.: *A Hybrid Evolutionary Algorithm: Multirecombination with Priority Rule Base Representation for the Job Shop Scheduling Problem*, Proc. VI Congreso Argentino de Ciencias de la Computación, CACIC 2000, Universidad Nacional de la Patagonia San Juan Bosco, pp. 1365-1376, 2000.

[14] Salto C., Alfonso H., Gallard R.: *Breeding in multirecombinated evolutionary algorithms with permutation based representation for the job shop scheduling problem*, Proc. of 4th International ICSC Symposium on soft Computing and Intelligent Systems for Industry (SOCO/ISFI2001), pp. 130, 2001.

[15] Salto C., Alfonso H., Gallard R.: *Performance Evaluation of Evolutionary Algorithms under Different Representations when Solving the JSS problem*, Proc. of 5th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2001), pp. 424-428, 2001.

[16] Whitley D.: *The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best*. Proc. of the 3rd International Conf. on Genetic Algorithms, pp. 116-121. Morgan Kaufmann 1989.