DIFFERENT EVOLUTIONARY APPROACHES TO SOLVE THE FLOW SHOP SCHEDULING PROBLEM

Federico Zuppa, Susana C. Esquivel, Raúl H. Gallard
{fede, esquivel, rgallard}@unsl.edu.ar
Laboratorio de Investigación y desarrollo en Inteligencia Computacional (LIDIC)
Phone: +54 2652 420823

Abstract

Over the past three decades extensive search have been done on pure m-machine flow shop problems. Many researchers faced the Flow Shop Scheduling Problem (FSSP) by means of well-known heuristics which, are successfully used for certain instances of the problem providing a single acceptable solution. Current trends involve distinct evolutionary computation approaches.
This work shows [5, 6, 7] implementations of diverse evolutionary approaches on a set of flow shop scheduling instances, including latest approaches using a multirecombination feature, Multiple Crossovers per Couple (MCPC), and partial replacement of the population when possible stagnation is detected. A discussion on implementation details, analysis and a comparison of evolutionary and conventional approaches to the problem are shown.

Introduction

The permutation Flow Shop Problem is a discrete optimization problem, that belongs to the field of scheduling. In the simplest situation, each job is processed by a series of machines in exactly the same order. The objective is to optimize some variable of performance, that in this case is the makespan. This problem, when there are more than two jobs, is NP-hard.

Many researchers [1, 8, 9, 10] faced the Flow Shop Scheduling Problem (FSSP) by means of well-known heuristics, which are successfully used for certain instances of the problem providing a single acceptable solution. Current trends involve distinct evolutionary computation approaches, that besides obtaining quality results, provide with a number of solutions. Therefore, the system is more fault tolerant.

The most preeminent problem that must be faced, in order to use these Evolutionary Algorithms (EA's), is the trade-off between exploration and explotation. Too much explotation can cause the algorithm to be mislead towards a local optimum. On the contrary, if the selective pressure were not enough, the algorithm would make a random walk without reaching the expected solutions. This balance may also be affected by other componentes, such as the complexity of the instance being analyzed. In this case, if the instance were simple, it wouldn't be necessary to put much selective pressure. On the other side, as the instances become more difficult, it is necessary to add more selective presure. This could lead to think that every instance of the problem requires a special EA and, unfortunately, this is to some extent true. Although it is not possible, if we tuned the EA in every instance, results would be much better. However, research has shown that there are certain patterns that can be followed to obtain a better performance.

Experiments

The first experiments we conduct by means of Evolutionary Algorithms used a single population, that evolved by means of simple crossovers between two parents and mutation. The first one used Proportional Selection (PS), Partially Mapped Crossover (PMX) and Random Exchange Mutation (RXM), while the second, used a selective operator created by Reeves, One-Cut-Point-Crossover (OCPX) and Shift Mutation (SM). Both EA's used a population of 100 individuals, 2000 generation and probabilities of crossover and mutation of 0.65 and 0.1 respectively. These EA's, were compared with heuristics created specifically for this problem, such as: Palmer's, Gupta's, Campbel's, Dudeck's and Smith's and Nawaz's, Enscore's and Ham's. The results obtained were very encouraging, showing that these simple algorithms outperformed almost all the conventional heuristics in the quality of results. The conventional heuristic that obtained the best results, was NEH, showing mean *Ebest* (percentile error of the best found individual) values ranging from 0.346% (in 100x5-problem size) to 5.002% (in 50x10-problem size). The first EA showed mean Ebest values ranging from 0.62% (in 100x5-problem size) to 3.585% (in 50x10-problem size), while the second one showed mean Ebest values ranging from 0.802% (in 100x5-problem size) to 4.775% (in 50x10-problem size) . EAs outperformed every heuristic, with the exception of NEH's heuristics, which got the best results in the 100x5 instances. It can also be seen that the first EA outperformed the second one. Further studies determined that the reason of this, layed in the mutation operator, which in the second case is very disruptive.

The next step was trying to reach better quality results. The first technique introduced with this purpose was the inoculation technique. In this technique, individuals generated by the given heuristics are inserted into the initial population. Two experiments were conducted, in order to check how the behavior of the EA is modified. In both experiments, 1 to 4 individuals generated by the already mention heuristics were inserted in the initial population of the earlier algorithms. This technique slightly improved the results in both EA's. In the first EA (with inoculation), the improvement was very little, showing mean *Ebest* values ranging from 0.642% (in 100x5-problem size) to 3.088% (in 50x10-problem size). In the second EA, results were much better using inoculation, and mean Ebest vaules reached 0.473% (in 100x5-problem size) to 3.334% (in the 50x10-problem size).

Recent improvements on EA's include a multiplicity feature [2, 3, 4], which allows multiple recombination on a pair of parents (MCPC). In order to deeply explore the recombination possibilities of previously found solutions, we decided to conduct two experiments in which a number n_1 of crossover operations for each mating pair was allowed. Each time a couple generated a number of children only two of them, best or random selected, were inserted in the next population The main advantage of this approach is a greater exploitation of the good, previously found, solutions. But on the other hand, because better parents produce a greater number of offspring, experiments showed that in some cases, the method increased the risk of premature convergence due to a loss of genetic diversity. This method obtained very poor results in the small instances, but as the instances became bigger, it obtained better results outperforming the single crossover EA's. For instance, the EA with MCPC and random selection of individuals, outperform every other EA in the 20x10 and 20x20 instances (with minimum Ebest mean values of 0.911 and 0.741 respectively), while the other MCPC EA outperform every other EA in the 100x5 instances (with a minimum ebest value of 0.156%), beating also the NEH heuristic.

The later method provides a better exploitation of the best individuals obtained until the moment. Watching closely to the graphs, generated by the best-so-far individuals, we noted that the problem of all the later EA's was premature convergence. In order to deal with this problem, we deploy a technique called "Shaking", that replaces a percentage of the population with new individuals created randomly. This is done during the evolutionary process when possible search improvements are not perceived through the measure of the mean population fitness. The decision criterion establish to apply partial replacement each time the mean population fitness remains unchanged within a threshold ε, along a predefined number n_2 of consecutive generations. The evolutionary process unconditionally finish when the maximum number of generations is achieved. These two last EAs were the best performers. The first one showed mean *Ebest* values ranging from 0.32% (in 100x5-problem size) to 2.81% (in 50x10-problem size) and the second one obtained mean *Ebest* values ranging from 0.51% (in 100x5-problem size) to 3.30% (in 50x10-problem size).

Conclusions

In the past few years, the Flow Shop Scheduling Problem was faced by means of different heuristics, including different Evolutionary approaches. These works have shown that Evolutionary Algorithms, combined with different techniques such as the inoculation technique, the multirecombinative approaches and shaking, obtained very good results, outperforming in most cases the conventional heuristics. Moreover the EA's, unlike the heuristics, provide a population of alternative optimal or near-optimal solutions.

Bibliography

- Campbell H., Dudek R., Smith M., *A heuristic algorithm for the n job m machine sequencing problem*, Management Science 16, 1970, 630-637.
- Esquivel S., Gallard R. and Michalewicz Z., *MCPC: Another Approach to Crossover in Genetic Algorithms*, Proceeding of Primer Congreso Argentino de Ciencias de la Computación, pp 141 - 150, 1995.
- Esquivel S., Leiva A., Gallard R., - *Multiple Crossover per Couple in Genetic Algorithms*, Proceedings of the Fourth IEEE Conference on Evolutionary Computation (ICEC'97), pp 103-106, ISBN 0-7803-3949-5, Indianapolis, USA, April 1997.
- Esquivel S., Leiva H., Gallard R., *Multiple crossovers between multiple parents to improve search in evolutionary algorithms*, Proceedings of the 1999 Congress on Evolutionary Computation (IEEE). Washington DC, pp 1589-1594.
- Esquivel S., Zuppa F., Gallard R., Multirecombinated Evolutionary Algorithms for the Flow Shop Scheduling Problem. PPSN IV. P. 263, 272.
- Esquivel S.C., Zuppa F., Gallard R., *Contrasting Conventional and Evolutionary Approaches for the Flow Shop Scheduling Problem, published in EIS 2000.*
- Esquivel S.C., Zuppa F., Gallard R., *Parallel and Sequential Evolutionary Algorithms for the Flow Shop Scheduling Problem*, published in CACIC 2000.
- Gupta J., A functional heuristic algorithm for the flowshop scheduling problem, Operational Research Quarterly 22, 1971,39-48,.
- Johnson S., Optimal two and three stage production schedule with setup times included, Naval Research Logistic Quarterly, vol 1, pp 61-68.1954.
- Nawaz M., Enscore E., Ham I., *A heuristic algorithm for the m-machine n-job flow shop sequencing problem*. Omeea vol II, pp 11-95, 1983.