Multiobjective Evolutionary Algorithms for Job Shop Scheduling

FERRERO S.W., ESQUIVEL S. C., GALLARD R.H.
Proyecto UNSL-338403, Departamento de Informática
E-mail: { swf, esquivel, rgallard}@unsl.edu.ar

Abstract

A job shop can be seen as a multi-operation model where jobs follows fixed routes, but not necessarily the same for each job. Job Shop Scheduling (JSS) attempts to provide optimal schedules according to some criterion. Common variables to optimize are makespan, machine idleness, lateness and total weighted completion time. According to this variables different objectives can be devised.

Multiobjective optimization, also known as vector-valued criteria or multicriteria optimization, have long been used in many application areas where a problem involves multiple objectives, often conflicting, to be met or optimized.

Multistage evolution and cooperative population search (CPS), as extended evolutive models, can be applied to solve multicriteria optimization, either using a plain aggregative approach or seeking the Pareto Front.

Multirecombination and Local Search were introduced in the CPS method in order to speed up and to improve the evolution.

MULTIOBJECTIVE OPTIMIZATION

When we are faced to a multicriteria optimization problem we can work in two different ways.

- We can make multicriteria decisions before searching. Fonseca and Flemming [7] classified as plain aggregative approaches those methods where a single objective function resulting as a numerical combination of objectives values is to be optimized.
- Or, we can first search the space for a set of slightly "good" solutions and then apply multicrite-ria decisions. Vilfredo Pareto [9] established that there exists a partial ordering in the searching space of a multiobjective problem. The Pareto criterion simply states that a solution is better than another one if it is so good in all attributes, and better in at least one of these attributes. In the problem space some solutions will not be dominated by any other solution and they con-form the Pareto front, also known as the acceptable set.

Due to their implicit parallel search, evolutionary algorithms (EAs) are suitably fitted to deal with JSSP [11] [6], as well as seeking solutions in Multiobjective optimization [1], [5], [7].

MULTISTAGE OPTIMIZATION

In this case, it was defined a single objective function as a plain aggregative approach. A genetic algorithm performs as usual finding the fittest individuals for that single aggregated function.
We considered three performance parameters to minimize:

- MS = max $\{C_1,...,C_n\}$, where C_i is the completion time of Job i. It is the makespan an it is equivalent to the completion time of the last job to leave the system.
- GE = max {0, -GL} where GL = MS – Gd, is the global lateness and Gd is the global due date.
- WCT = $\sum_{i=1}^{n} w_i C_i$, where wi is the weight associated to the completion time of Job i. It is the weighted completion time.

Our multistage Genetic Algorithm (MGA) considered the 3-criteria problem optimizing three objective functions, f_1, f_2 and f_3, corresponding to MS, GE and WCT, respectively and the aggregation $f = \alpha f_1 + \beta f_2 + \gamma f_3$.

The idea is to create one sub-population for each criterion and evolve them until convergence. At this point another evolutionary process begin actuating on the whole population whose objective is an aggregation of the partial objectives. This evolution step continues until reaching convergence.

After that, the whole population is subdivided into sub-populations and the original process is started again. A final stop criterion is defined to terminate the entire process.

The multistage evolutionary approach (MGA) was contrasted against a conventional evolutionary approach (SGA). Ten instances of two types, small and big [8], with known optimal makespan were used. As optimal values of makespan were known for each instance of the test suite benchmark, the global due date to determine GE was fixed at a value 40% greater than the corresponding optimal makespan. Coefficients α, β, and γ for the aggregation f, were set at convenient values and normalised for each instance in order to give higher weights to MS and GE.

As a result, at the end of the multistage evolutionary process we accomplished two different purposes: to optimize the aggregated objective and obtain a set of good performers, which are near-optimal solutions on their corresponding partial objective.

COOPERATIVE POPULATION SEARCH (CPS) – SINGLE CROSSOVER AND MULTIRECOMBINATION

Conventional approaches to crossover, independently of the method being used, involve applying the operator only once on the selected parents. Such a procedure will be known as the Single Crossover Per Couple (SCPC) approach.

In earlier works [3], [4], we devised a different approach: to allow multiple offspring per couple, as often happens in nature. In order to deeply explore the recombination possibilities of previously found solutions, we decided to conduct several experiments in which more than one crossover operation for each mating pair was allowed.

The number of children per couple was fixed or granted as a maximum number and the process of producing offspring was controlled, for each mating pair, in order not to exceed the population size.

The idea of multiple children per couple was tested on a set of well-known testing functions (De Jong functions F1, F2 and F3 [2], Schaffer F6 [10] and other functions). A simple GA, with conventional operators and parameter values, was the basis of those initial experiments.

Basically the CPS-MCPC approach:

- Maintains a single population of solutions which is separately ranked by each criterion.
- Uses ranking selection to select one parent per criterion.
- Uses multiple crossovers per couple (MCPC), and the corresponding crossover and mutation operators to generate multiple offspring.

After each mating, for insertion in the next population, selects those offspring, which are classified so far, as globally non-dominated. If none fulfilling this condition exists then half of the newly generated offspring are inserted, selecting first those that are non-dominated within the new offspring subset and completing the insertions by random selection if necessary.

The last point above mentioned, implies to maintain the updated set of solutions found so far as belonging to the Pareto front. Let us call it $P_{current}$. This set is updated at the end of each generation cycle. Essentially the proposed CPS-MCPC,

- Augments implicit parallel search by encouraging crossbreeding among "species".
- Increases exploitation of good solutions previously found through multiple crossovers per couple.
- Favours for insertion in the next generation those solutions which are, at the present stage, non-dominated (globally, at $P_{current}$ level, or locally, at the offspring level). If none is found then genetic diversity is favored by random selection.

In this work we selected f_1 as the makespan, and f_2 as the mean absolute deviation of job completion times from a common due date d, as the conflicting criteria to minimize.

During the first studies of the CPS-MCPC approach it was observed that in many cases MCPC found better results than SCPC and best quality results were obtained allowing between 2 and 4 crossovers per couple.

MULTIRECOMBINATION AND LOCAL SEARCH (CPS-MCPC-LS)

A new set of experiments was devised attempting to enhance performance by locally perturbing the already found solutions in the final $P_{current}$ set provided by CPS-MCPC. Let us call it P_{known}. Local perturbation is achieved by adding simulated annealing (SA) as a local search heuristic. In CPS-

MCPC-LS local search was applied to each solution in P_{known} and when new non-dominated solutions were found they were added to the $P_{current}$ set in the SA algorithm.

CONCLUSIONS

This work reports experience on multiobjective optimization applied to the Job Shop Scheduling problem. We can remark:

First, results of these preliminary experiments with MGA show some enhancements when compared with the conventional evolutionary approach. In general, overall performance is slightly better in big instances also providing near optimal solutions, under each individual criterion.

Second, CPS-MCPC uses co-operative population searches with multirecombination. Experiments to contrast multiple versus single recombination were performed using three basic representations for the JSSP. In most cases CPS-MCPC builds improved, more densely and evenly distributed Pareto fronts than CPS-SCPC. Moreover, the final population obtained is grouped around compromise solutions. This fact shows that the alternative solutions provided by multirecombination attempt to balance the damage caused on the conflicting objectives of the multicriteria problem.

Finally, CPS-MCPC-LS adds local search for further improvements. This new approach slightly improves the performance of its predecessor providing better points with negligible additional computational effort.

Future work includes to start working on dynamic scheduling. First, by developing several evolutionary heuristics to solve that kind of problems and then working on multiobjective optimization.

Bibliography

1. Coello Coello C., An updated survey of evolutionary multiobjective optimization, Proceedings of the 1999 Congress on Evolutionary Computation (IEEE). Washington DC, pp 3-13.
2. De Jong K. A., Analysis of the Behavior of a Class of Genetic Adaptive Systems, PhD Dissertation, University of Michigan, 1975.
3. Esquivel S., Gallard R., Michalewicz Z., MCPC: Another Approach to Crossover in Genetic Algorithms, Proceeding of Primer Congreso Argentino de Ciencias de la Computación, pp 141 - 150, 1995.
4. 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.
5. Esquivel S., Leiva H.,.Gallard R., Multiplicity in genetic algorithms to face multicriteria optimization, Proceedings of the 1999 Congress on Evolutionary Computation (IEEE). Washington DC, pp 85-90.
6. Fang H.-L., Ross P., and D. Corne. A promising genetic algorithm approach to job-shop scheduling, rescheduling, and open-shop scheduling problems. ICGA93 , pages 375-382. Proceedings of the Fifth International Conference on Genetic Algorithms . Morgan Kaufmann Publishers, San Mateo, CA, 1993.
7. Fonseca C. M., Fleming P. J. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization Proc. of the 5th In. Conf. on Genetic Algorithms, pp 416-423, Urbana- Champaign, IL, Morgan Kaufmann, 1993.
8. Lawrence S., Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1984.
9. Pareto V., Cours d’Economie Politique, 1896, Switzerland, Lausanne: Rouge.
10. Schaffer J., Caruana R., Eshelman R., Das R., A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization, in Third International Conference on Genetic Algorithms, pages 51 - 60, 1989.
11. Yamada T. and Nakano R.. A genetic algorithm applicable to large-scale job-shop problems. In R. Maenner and B. Manderick, editors, Parallel Problem Solving from Nature 2 , pages 281-290. Elsevier, Amsterdam, 1992.