STUDYING THE PARALLEL TASK SCHEDULING PROBLEM WITH CONVENTIONAL AND EVOLUTIONARY ALGORITHMS

Claudia Gatica, Susana Esquivel, Raúl Gallard

,

Laboratorio de Investigación y Desarrollo en Inteligencia Computacional,
Universidad Nacional de San Luis, Argentina
{ crgatica, esquivel, rgallard}@unsl.edu.ar

SUMMARY

This work summarizes results when facing the problem of allocating a number of non-identical tasks in a parallel system. The model assumes that the system consists of a number of identical processors and that only one task may be executed on a processor at a time. All schedules and tasks are non-preemptive. Graham's [8] well-known list scheduling algorithm (LSA) was contrasted with different evolutionary algorithms (EAs), which differ on the representations and the recombinative approach used. Regarding the representation, direct and indirect representations of schedules were used. Concerning recombination, the conventional single crossover per couple (SCPC), and multiple crossovers per couple (MCPC) [3], [4] were implemented. Latest improvements in evolutionary computation include multirecombinative variants. Multiple crossovers multiples on parents (MCMP) provides a means to exploit good features of more than two parents selected according to their fitness by repeatedly applying any crossover method: a number n_1 of crossovers is applied on a number n_2 of selected parents. Performance enhancements were clearly demonstrated in single and multicriteria optimisation [5], [6] under this approach. The use of a stud is a well-known practice in breeding by which a breeding animal due to its special features is selected more often for reproduction. This model of reproduction is being implemented for the Parallel Task Scheduling Problem.

A parallel program is a collection of tasks, some of which must be completed before than others begin. The precedence relationships between tasks are commonly delineated in a directed acyclic graph known as the *task graph.* Nodes in the graph identify task and their duration and arcs represent the precedence relationships. Factors, such as number of processors, numbers of tasks and task precedence contribute to make difficult a good assignment. The problem of how to find a schedule on $m > 2$ processors of equal capacity that minimises the whole processing time of independent tasks has been shown as belonging to the NP-complete class [9]. A schedule is an allocation of task to processors. From the whole processing time point of view, an optimal schedule is such that the total execute time is minimised. Other performance variables, such as individual processor utilisation or evenness of load distribution can be considered.

Using the LSA is possible to assign tasks to processors for a given list of tasks by always assigning each available processor to the first unassigned task on the list whose predecessor tasks have already finished execution. This heuristic presented some anomalies contrary to the intuition, for example, increasing the number of processors, decreasing the execution times of one or more tasks, or eliminating some of the precedence constraints can actually increase the makespan.

From the representation perspective many evolutionary computation approaches to the general scheduling problem exits. According to solution representation these methods can be roughly categorised as *indirect* and *direct representation* [1], [2]. In the case of indirect representation of solutions the algorithm works on a population of encoded solutions. Because the representation do not directly provides a schedule then a scheduler builder is necessary to transform a chromosome into a schedule, validate and evaluate it. The schedule builder guarantees the feasibility of a solution and its work depends of amount of information included in the representation. In direct representation [2] a completed and feasible schedule is an individual of the evolving population. The only method that performs the search is the evolutionary algorithm because the represented information comprises the whole search space.

The first trials consisted of the study of behaviour of the LSA against EAs using both indirect and direct representation on the simple set of test cases. Here we detected that the EAs are free of LSA anomalies and where possible they also find more than one optimal and alternative solution. When we considered the EAs only, they differ in the number of optimal and alternative solutions provided, being better those EAs with direct representation. Then the following study considered the multirecombinative method permitting more than one crossover on the couple (MCPC), wit an increased testing set, adding more complexes cases. Here diverse performance variables were considered to contrast the algorithms: *versatility,* to measure the ability of the algorithms in providing alternatives and optimal solutions and *quality* to measure the percentile error of the best found individual in one run when it is compared with the known, or estimated optimal value. Regarding *versatility* (the mean number of optimal solutions found in a run) MCPC provides a slight superior average behaviour than SCPC. Regarding *quality* MCPC outperforms SCPC, however there are some testing cases for which no optimal value were presently found. All the approaches and experiments explained so far, and their corresponding results can be found in [7].

Some enhancements on EAs applied to the Parallel Task Scheduling Problem were also achieved by means of MCMP. A balance between exploration and exploitation in the search space [10] was the main goal here. The results indicated that MCMP is less versatile than SCPC and MCPC but it provides higher quality solutions.

The last study included MCMP-S that is a novel variant of MCMP, which considers the inclusion of a stud individual in a pool of intervening parents. Members of this matting pool subsequently undergo multiple crossover operations having always the stud as a member of a couple. The preliminary results compared these methods. Regarding *quality* of results it is clear that MCMP-S is the best performer for each of the harder cases providing the minimum makespan. Concerning *versatility* MCMP-S is also the method providing a greater number of distinct schedules with the minimum makespan.

CONCLUSIONS AND FUTURE WOKS

In this work we approached the allocation of a number of parallel tasks in parallel supporting environments attempting to minimise the makespan. LSA is a polynomial time scheduling algorithm that provide good solutions.

Several EAs were studied. First, two variants of representations were undertaken to contrast their behaviour with the LSA, later on different approaches of recombination were considered. Preliminary results on the selected test suite showed two important facts: first, EAs provide not

a single, but a set of optimal solutions, providing for fault tolerance when system dynamics must be considered, and second, EAs are free of the LSA anomalies.
After these initial experiments, three variants of recombination were considered; SCPC, MCPC and MCMP for each representation. The behaviours of the EAs were similar and all of them showed better results than LSA.

When we compare their performance it is clear that the approaches including multirecombination behave better than the conventional ones (in both representations) but yet it would be necessary to continue experimentation with different parameter settings, self-adaptation of parameters, and to compare them with newer non-evolutionary heuristics.

Current research includes the use of a breeding individual (stud) which repeatedly mates individuals that randomly immigrate to the mating pool. Under this approach the ramdon immigrants incorporate exploration and the multi-mating operation with the stud incorporates exploitation to the search space.

REFERENCES

[1]. Bagchi S., Uckum S., Miyabe Y., Kawamura K.: Exploring Problem Specific Recombination Operators for Job Shop Scheduling. Proceedings of the 4th International Conference on Genetic Algorithms, pp 10 – 17 (1991)

[2]. Bruns R.: Direct Chromosome Representation and Advanced Genetic Operators for Production Scheduling. Proceedings of the 5th International Conference on Genetic Algorithms, pp 352-359, (1993).

[3]. Esquivel S., Leiva A., Gallard R: Multiplicity in Genetic Algorithms to Face Multicriteria Optimization. Proceedings of the Congress on Evolutionary Algorithms (IEEE), Washington DC, pp 85 – 90, (1999).

[4]. Esquivel S., Leiva A., Gallard, R.: Couple Fitness Based Selection with Multiple Crossover per Couple in Genetic Algorithms. Proceedings of the International Symposium on Engineering of Intelligent Systems (EIS'98), La Laguna, Tenerife, Spain, Vol. I, pp 235 – 241, Ed. E.Alpaydin, Published by ICSC Academic Press, Canada/Switzerland, (1998).

[5]. Esquivel S., Leiva A., Gallard R.: Multiple Crossovers between Multiple Parents to Improve Search in Evolutionary Algorithms. Proceedings of the Congress on Evolutionary Algorithms (IEEE), Washington DC, (1999).

[6]. Esquivel S., Ferrero S., Gallard R., Alfonso H. Salto C., Schütz M.:Enhanced evolutionary algorithms for single and multiobjective optimization in the job shop scheduling problem. To appear in the Knowledge Based System Journal, Elsevier 2001.

[7]. Esquivel S, Gatica C, Gallard r., Conventional and Multirecombinative Evolutionary Algorithms for the Parallel Task Scheduling Problem.To appear in LNCS, Springer, 2001

[8]. Graham R. L.: Bounds on Multiprocessing Anomalies and Packing Algorithms. Proceedings of the AFIPS 1972 Spring Joint Computer Conference, pp 205-217, (1972).

[9]. Horowitz E. and Sahni S.: Exact and Approximate Algorithms for Scheduling non Identical Processors. Journal of the ACM, vol. 23, No. 2, pp 317-327, (1976).

[10]. Michalewicz Z., Genetic Algorithms + Data Structures = Evolution Programs. Springer Verlag, Third, Extended Edition, (1996).