Parallel Memetic Algorithms for Independent Job Scheduling in Computational Grids

Fatos Xhafa and Bernat Duran

Polytechnic University of Catalonia, Department of Languages and Informatics Systems, C/Jordi Girona 1-3, 08034 Barcelona, Spain
{fatos,bduran}@lsi.upc.edu

Summary. In this chapter we present parallel implementations of Memetic Algorithms (MAs) for the problem of scheduling independent jobs in computational grids. The problem of scheduling in computational grids is known for its high demanding computational time. In this work we exploit the intrinsic parallel nature of MAs as well as the fact that computational grids offer large amount of resources, a part of which could be used to compute the efficient allocation of jobs to grid resources.

The parallel models exploited in this work for MAs include both fine-grained and coarse-grained parallelization and their hybridization. The resulting schedulers have been tested through different grid scenarios generated by a grid simulator to match different possible configurations of computational grids in terms of size (number of jobs and resources) and computational characteristics of resources. All in all, the result of this work showed that Parallel MAs are very good alternatives in order to match different performance requirement on fast scheduling of jobs to grid resources.

Keywords: Memetic Algorithms, Independent Job Scheduling, Computational Grids, Parallelisation, Two Level Granulation.

14.1 Introduction

In this chapter we present several parallel implementations of Memetic Algorithms (MAs), both unstructured and structured, for the problem of independent job scheduling to grid resources. The scheduling problem is at the heart of any Computational Grid (CG) [415, 416]. Due to this, the job scheduling problem is increasingly receiving the attention of researchers from the grid computing community with the objective of designing schedulers for high performance grid-enabled applications. The independent job scheduling on computational grids is computationally hard. Therefore the use of heuristics is the de facto approach in order to cope in practice with its difficulty. Thus, the evolutionary computing research community has already started to examine this problem [407, 411, 420, 423, 413]. Yet, the parallelization of meta-heuristics, in particular of MAs [422], for the resolution of the problem has not been explored.

This work builds upon previous work on MAs for independent job scheduling [424]. One of the most advantageous characteristics of the family of Evolutionary Algorithms (EAs) is the intrinsic parallel nature of their structure.
Holland [418] in his early works introduced the first ideas for defining a parallel structure for EAs. The main observation here is that the algorithms based on populations of individuals could have a very complex structure, yet easily decomposable in smaller structures. This decomposition could be very useful to distribute the work of the algorithm to different processors. The objective of parallelizing EAs is essentially to distribute the burden of work during the search to different processors in such a way that the overall search time within the same (sequential) exploration is reduced. Therefore, the parallelization of EAs, in general, and MAs, in particular, could be beneficial for the resolution of the independent job scheduling in computational grids. Moreover, parallelizing EAs could imply not only the reduction of resolution time or better quality of solutions; it is also a source for new ideas in re-structuring the EA algorithm, differently from its original sequential structure, which could eventually lead to better performance of the algorithm.

In this chapter three different models of parallelizing unstructured MAs and Cellular MAs (here after refereed to as MAs and cMAs, resp.) are studied: (a) the model of independent searches, referred to as Independent Runs (IR) model for MAs; (b) the Master-Slave (MS) for MAs; and, (c) parallel hybridization among the coarse-grained and fine-grained models (also referred to as Two Level Granulation) for cMAs. The IR model consists of simultaneous execution of independent searches. In the MS model, the search algorithm is executed by a master processor, which delegates independent sub-tasks of high computational cost to the rest of processors (slaves). In the context of MAs, different slave processors could apply different local search procedures [421, 419] on the individuals of the population. Finally, hybrid parallel implementation is done for cMAs by combining the coarse-grained model, at a first hierarchical level, and the fine-grained model, at a second hierarchical level.

The proposed parallel MAs are implemented in C++ and MPI using a skeleton for MAs [409], extended for the purposes of this work. The implementations are extensively tested, on the one hand, to identify a set of appropriate values for the search parameters and, on the other, to compare the results for the makespan parameter obtained by different parallel implementations and the corresponding sequential versions. To this end we have used different grid scenarios generated using a grid simulator [425] to match different possible configurations of computational grids in terms of size (number of jobs and resources) and computational characteristics of resources.

The remainder of the chapter is organized as follows. We give in Section 14.2 the description of independent job scheduling problem. An overview on the taxonomy of parallel models for meta-heuristics and their application to MAs and cMAs is given in Section 14.3. The implementation of the parallel MAs is given in Section 14.4. The experimental study and some relevant computational results are presented in Section 14.5. We conclude in Section 14.6 with most important aspects of this work and indicate directions for future work.