ParaDisEO-Based Design of Parallel and Distributed Evolutionary Algorithms

S. Cahon¹, N. Melab¹, E.-G. Talbi¹, and M. Schoenauer²

¹ Laboratoire d’Informatique Fondamentale de Lille
Université des Sciences et Technologies de Lille
59655 - Villeneuve d’Ascq Cedex
² Projet Fractales – INRIA Rocquencourt – 78153 Le Chesnay Cedex
{cahon, melab, talbi}@lifl.fr, marc.schoenauer@inria.fr

Abstract. ParaDisEO is a framework dedicated to the design of parallel and distributed metaheuristics including local search methods and evolutionary algorithms. This paper focuses on the latter aspect. We present the three parallel and distributed models implemented in ParaDisEO and show how these can be exploited in a user-friendly, flexible and transparent way. These models can be deployed on distributed memory machines as well as on shared memory multi-processors, taking advantage of the shared memory in the latter case. In addition, we illustrate the instantiation of the models through two applications demonstrating the efficiency and robustness of the framework.

1 Introduction

There are (at least) two broad categories of optimization problems that requires heavy computational resources: large combinatorial problems, and complex numerical engineering problems.

Large combinatorial problems are continuously evolving in terms of requirements, constraints, etc. Therefore, one needs flexible and adaptable algorithms to solve them. Furthermore, these problems are often NP-hard, characterized by a complex landscape and large instances with many decision variables.

Combinatorial optimization algorithms fall into two categories: exact methods and metaheuristics. Exact methods allow to find optimal solutions but they are often inefficient and unpractical. On the opposite, the metaheuristics aim at finding efficiently near-optimal solutions. Metaheuristics include Evolutionary Algorithms (EAs), local search methods, and the like.

Similarly, the advances of both modelization techniques and numerical simulation algorithms have increased the demand for numerical optimization algorithms. And the practitioner is faced with the following dilemma: either use some deterministic method to precisely optimize a simplified model, or use a stochastic method to approximately optimize a complex model. Evolutionary Algorithms are a good choice in the latter situation.

Unfortunately, EAs applied to real-world problems, either large combinatorial problem or complex numerical optimization applications, are known to
be time consuming. On the other hand, the proliferation of networks/clusters of workstations and parallel machines offers more and more facilities to the users: EAs are also known to allow efficient Parallelization/distribution and multi-threading, achieving high performance and robustness in reasonable time . . . at the cost of a sometimes painful apprenticeship of parallelization techniques.

Many implementations of EAs have been proposed and some of them are available on the Web. Code developers are often tempted to reuse them to save time. However, understanding and reusing a third party code is generally a long, tedious and error prone task. Indeed, one needs to examine the internal working of the code and make several modifications to adapt it to a new problem. A better reuse approach consists in using a framework, such as ParaDisEO [4] or MALLBA [1], dedicated to the design and development of EAs. Good frameworks should be well documented and tested. But their success strongly depends on their ability to allow flexible and adaptable design. For instance, adaptation should simply require to parametrize some existing EAs and at least to write the fitness function. The same is true for the parallelization/distribution of EAs: In order to facilitate this step for those who are unfamiliar with parallel mechanisms, the frameworks must integrate the up-to-date parallelization techniques and allow their exploitation and deployment in a transparent manner.

This paper focuses on the ParaDisEO framework [1], a flexible approach for the transparent design and deployment of parallel and distributed evolutionary algorithms. ParaDisEO is basically an extension of Evolving Objects (EO) [9], an open source framework based on C++ templates allowing a flexible design of EAs. The extensions include local search methods (descent search, simulated annealing and tabu search for combinatorial optimization, gradient-based search for numerical optimization), hybridization mechanisms (coupling local search and global evolutionary search) and parallel/distributed models. This paper will concentrate on the latter topic: the different models supported by ParaDisEO will be presented, together with their flexible and transparent parameterization. The user is relieved from the burden to explicitly manage the communication and concurrency. Furthermore, the models can be efficiently deployed both on shared memory multi-processors and on distributed networks.

The reader is referred to [2] for a state of the art about parallel and distributed evolutionary algorithms, their design, many techniques of hybridization, and a full taxonomy of applications done until now. In this paper, the use of ParaDisEO will be illustrated on two test cases: the spectroscopic data mining [10] and the network design optimization [11]. The results show that they allow an efficient and robust deployment.

1 This work is a part of a current French joint grid computing project ACI-GRID DOC-G (Challenges in Combinatorial Optimization on Grids)