Evolutionary Computing within Grid Environment

Ashutosh Tiwari, Gokop Goteng, and Rajkumar Roy

Decision Engineering Centre, Manufacturing Department, School of Applied Sciences, Cranfield University, Cranfield, Bedfordshire, MK43 0AL, UK
\{a.tiwari, g.l.goteng, r.roy\}@cranfield.ac.uk
http://www.cranfield.ac.uk/sas/mem/decisionengineering/

Summary. Evolutionary computing (EC) techniques such as genetic algorithms (GA), genetic programming (GP), evolutionary programming (EP) and evolution strategies (ES) mimic nature through natural selection to perform complex optimisation processes. Grid-enabled environment provides suitable framework for EC techniques due to its computational and data capabilities. In addition, the semantic and knowledge Grids aid in the design search and exploration for multi-objective optimisation tasks. This chapter explores some problem solving environments such as Geodise (Grid-Enabled Optimisation Design Search for Engineering), FIPER (Federated Intelligent Product Environment), SOCER (Service-Oriented Concurrent Environment), DAME (Distributed Aircraft Maintenance Environment) and Globus toolkit to demonstrate how EC techniques can be performed more efficiently within a Grid environment. Service-oriented and autonomic computing features of Grid are discussed to highlight how EC algorithms can be published as services by service providers and used by service requestors dynamically. Grid computational steering and visualisation are features that can be used for real-time tuning of parameters and visual display of optimal solutions. This chapter demonstrates that grid-enabled evolutionary computing marks the future of optimisation techniques.

10.1 Introduction

The use of Evolutionary computing (EC) to solve many optimisation problems is still hindered by the inadequate computational power and data manipulation platform for computational and data intensive problems. Examples of such computational and data intensive applications are engineering design optimisation, bioinformatics, pharmaceutical and particle physics simulations and many others. Additionally, since EC techniques mimic nature through natural selection, knowledge-driven problem solving environments (PSEs) are ideal frameworks for optimisation problems using EC. Grid computing (GC) promises to provide computational resources and services for optimisation
processes using EC techniques. Besides, the service-oriented architecture (SOA) in Grid-based deployments provide easy and seamless access to heterogeneous distributed optimisation codes to users in a dynamic and coordinated manner. Service-oriented problem solving environments (SO-PSEs) such as Geodise (Grid-Enabled Optimisation Design Search for Engineering), FIPER (Federated Intelligent Product Environment) and SORCER (Service-Oriented Concurrent Environment) are good examples of Grid-based PSEs that use semantic Grid and ontology to intelligently drive optimisation processes. Globus toolkit which serves both as a middleware as well as a PSE has been the major innovative contribution to the second generation Grid by the Global Grid Forum (GGF) which has now been merged with the Enterprise Grid Alliance (AGA) to become the Open Grid Forum (OGF).

This chapter will focus on how evolutionary computing techniques can exploit Grid-based framework to provide seamless access to optimisation resources and services for users involved in multidisciplinary optimisation applications. The first part of this chapter will discuss evolutionary computing and related application areas. The second part will look at intelligent Grid environments such as semantic Grid, knowledge Grid and autonomic computing. The third part will concentrate on service-oriented and problem solving environments for Grid-based problems. The fourth part will delve on problem solving environment case studies. The last part of this chapter will talk about collaborative evolutionary computing.

10.2 Evolutionary Computing

Evolutionary computing uses techniques such as genetic algorithms (GAs), genetic programming (GP), evolutionary programming (EP) and many others. These techniques work on the principles of natural selection based on Darwin’s theory of evolution. This principle works on the composition of genetic traits called chromosomes, in which successive operations through crossover or mutation give rise to better performing off-springs (population) due to successive refinement of these hereditary traits. In the same way, evolutionary computing techniques mimic this phenomenon of natural selection to improve upon classical methods of optimisation such as preference-based and generating methods that work well only on single-objective optimisation problems [10]. With evolutionary computing techniques, multi-objective optimisation problems can be solved, producing multiple optimal solutions in a single simulation run. This is not possible with classical optimisation methods. However, the evolutionary computation can be computationally expensive for complex optimisation problems. Imagine an optimisation problem of an underground system with hundred of thousands of optimal solutions. This is where Grid-based framework comes in. Grid computing is a distributed computing infrastructure which aims to overcome large-scale computation within the