in: Proceedings of 9th International Symposium on Systems - Modelling – Control, Zakopane, Poland, 27/04-01/05/1995, [CD-ROM]

On using parallel genetic algorithms for large-scale optimization problems: a case report

Heitor Silvério Lopes
Centro Federal de Educação Tecnológica do Paraná
Av. 7 de setembro, 3165 - 80230-901 - Curitiba (PR) - Brazil
hslopes@cpgei.cefetpr.br

Abstract
This paper presents a case report of the implementation of a parallel genetic algorithm (PGA) used to optimize the performance of an intelligent medical diagnostic system called DoToR. The search space involved in such optimization is of high dimensionality and complexity, which could not be accomplished using traditional mathematical approaches. The implementation of PGA in a network of computers using parallel virtual machine is discussed. The general ideas here presented could be employed to deal with other classes of problems where large-scale complex optimization is required.

1. Introduction

There are several mathematical methods that are traditionally used to solve optimization problems. However, these methods can not be applied to some classes of problems that include, besides extremely large search space, high multimodality and multiobjective optimization.

This paper describes the implementation of a parallel genetic algorithm system, based on a parallel virtual machine over a set of networked workstations. This system was successfully employed optimize a large-scale medical diagnostic system, for which other strategies were not applicable.

2. Background: PGAs and PVM

2.1 Parallel Genetic Algorithms

Genetic Algorithms - GAs (Goldberg, 1989) is an implicitly distributed search method. The motivation to develop Parallel Genetic Algorithms - PGAs (Tanese, 1989) comes from the appealing idea of separately evolve populations in parallel, and sometimes share genetic material between. This approach is intuitively acceptable considering its biological plausibility, since that in nature, the evolution of species is a highly parallel process. The parallel evolution of populations in PGAs is explicit, and should not be confused with the implicit parallelism of GAs. This last is related to the property of GAs to simultaneously process a large number of schemata, while processing a small number of individuals.

Several approaches to PGAs have been proposed, but their implementation is a function of the available hardware. According to Pit (1995), PGAs implementations are possible in three different forms: global population with parallelism, island genetic algorithm - IGA and cellular genetic algorithm - CGA. IGA implementations require either a set of networked computers or a computer that runs multiple processes. CGA implementations require a massively parallel computer.

This work is based in the IGA approach because it can be implemented in network of heterogeneous computers. In an IGA implementation, each processor of the network controls and evolves a whole population, independently of the others. A single central control is necessary in order to allow file sharing and manage migration between populations. Both the implementation and the performance of an IGA include the fine-tuning of some parameters that can be viewed as a migration policy, which defines:

- The selection scheme of immigrants of a population;
- The rate and frequency of immigration and emigration for each population;
- Which populations are allowed to receive and/or send migrants to which.
- The substitution schema of the immigrants in the population.

IGAs have demonstrated a better performance over serial GAs for complex problems, justifying its practical implementation. This is particularly true for large-scale and highly multimodal problems, where the simultaneous exploration of several regions of the search space is mandatory. A class of problems that clearly require the use of PGAs is when it is required the formation of located subpopulations (niches) with small or no competition between then, minimizing the effects of selective pressure.

2.2 Parallel Virtual Machine

The Parallel Virtual Machine system - PVM (Geist, Beguelin, Dongarra et al, 1994) is a joint project that has been conducted in order to become a standard for parallel computation in heterogeneous networks. Bakic and Enbody (1995) have developed a PGA system that uses real parallelism over a network using PVM. This system was used as the base of the GADoToR system development, further described.

3. Problem setting

The DoToR system (Lopes, Coutinho, Heinisch et al, 1997) is a knowledge-based system for decision-support in chest pain diagnosis. This system is based on analogical reasoning with disease prototypes and a similarity measure, and its methodology was detailed elsewhere (Lopes, Coutinho, Lima, 1995). This system is able to diagnose 12 different diseases and uses 161 variables to fully describe each one of the disease prototypes. These variables are related to patient information regarding the pain characteristics, accompanying symptoms, other signals perceived by the clinician, clinical history and available results of laboratory tests. To each variable a weight is attached representing its "diagnostic power" for a given diagnosis. These weights were elicited from an expert clinician and demanded heavy efforts from him. Furthermore, tests on the first version of the DoToR system have suggested that further adjustments in the weights sets could improve its performance. In fact, a previous work has already demonstrated the feasibility of these ideas (Lopes, Coutinho, Lima, 1997). This has motivated the development of GA-based system to optimize the 12 sets of weights of the DoToR system.

In order to optimize the whole system, each one of the 12 sets of weights has to be optimized separately. Using 4 bits to achieve an acceptable precision, yields a search space of $2^{161 \times 4} = 2^{644} \approx 10^{194}$ discrete points for each one of the 12 disease prototypes. These figures are exceedingly large to be coped with any current computer system. This huge search space is one of the main reasons to use a GA-based optimization system. The second reason is the nature of what is being optimized, since there is no mathematical function to be manipulated. Instead, the set of weights is indirectly evaluated based on the performance of the diagnostic system (DoToR) to diagnose a database of 90 clinical cases. Empirical knowledge has demonstrated that such evaluation may include severe noise and non-linearity.

4. The GADoToR system

The GADoToR system is a PGA implementation of the island type (IGA), running over a PVM platform. Experiments of the GADoToR system were run in a Sun Sparcstation-20 as master controller and four Sun Classic as slave processors, although other heterogeneous UNIX workstations could be used as well.

The system has three different kinds of processes: master, monitor and slave. For each run there is one master process, N monitor processes and N slave processes, where N represents the number of processors (and actually, the number of populations, in this version), as shown in figure 1. The master process is responsible for the user interface, as well as setting running parameters and computing global statistics. Moreover, it spawns and controls the termination of all other processes. Monitor processes are the link between the master and the slave processes, control the synchronized migration between populations and manage the communication between the local slave and the master. Finally, slave processes execute a sophisticated GA code. Also, they compute statistics for the local population and request emigrants from other neighbor monitor processes.

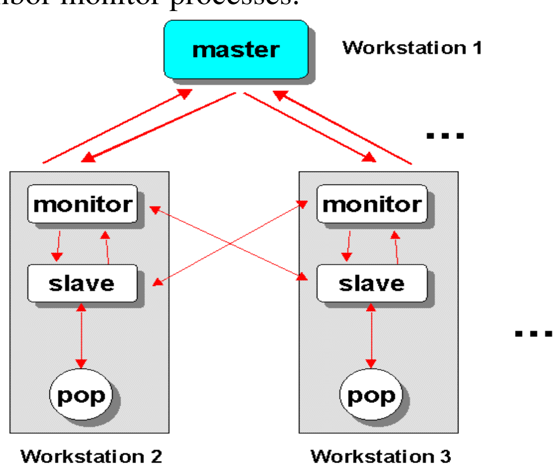

Figure 1: Block diagram of processes in the GADoToR

The current version of GADoToR uses PVM version 3 (Geist, Beguelin, Dongarra et al, 1994), which is a very powerful platform for parallel computing. Despite of its flexibility, PVM was not used full explored, and advanced capabilities like dynamic load balancing and mechanisms for task migration were not implemented. These capabilities were not important in this implementation because the network of slave workstations used were based on the same hardware and not running other processes. Thus, the main concern was the static scheduling, provided that the same computational load were assigned to all processors (except the master). Notwithstanding, those advanced features as well others would be of great value when exploiting a network of UNIX-based heterogeneous workstations (Shirazi, Hurson, Kavi, 1995).

5. Results and conclusions

The GADoToR system was used to optimize the weights set of DoToR system as previously mentioned. The performance was measured as the ability to correctly classify a database of 90 clinical cases (including all 12 diseases). The system was submitted to a supervised learning, starting from scratch. The final average performance of the trained system was 17% higher when compared to the system when it used the set of weights elicited from the expert. This means that the elicitation procedure, well known as the bottleneck of knowledge-based systems, could be significantly facilitated (or even avoided) using this methodology and a high-quality database. After trained, the diagnostic system achieved 100% of sensibility (all positive cases were detected) and 89% of specificity (only 11% of negative cases were wrongly classified). The use of GAs has proven to be efficient to tackle this difficult, large-scale complex problem. The convergence of training was greatly accelerated using PGAs, notably due to the co-evolution effect caused by migration. This observation confirms the capability of PGAs to control selective pressure, while preserving genetic diversity in order to efficiently explore huge search spaces.

References

BAKIC V., ENBODY R., (1995) *The PVM GALOPP System User's Guide*, Lansing, Michigan State University.

GEIST A., BEGUELIN A., DONGARRA J., et al., (1994) *PVM - Parallel Virtual Machine: a user's guide and tutorial for networked parallel computing*, Cambridge, MIT Press.

GOLDBERG D., (1989) *Genetic algorithms in search, optimization and machine learning*, Reading, Addison-Wesley.

LOPES H.S., COUTINHO M.S., LIMA W.C., (1995) A model of similarity-triggered analogical reasoning diagnosis in medicine, *8th Int. Symp. on System, Modelling and Control*, Lodz, PSMI, **2**, 13-17.

LOPES H.S., COUTINHO M.S., HEINISH R., et al, (1997) A knowledge-based system for decision-support in chest pain diagnosis, *Medical & Biological Engineering & Computing* **35**, suppl., part I, 514.

LOPES H.S., COUTINHO M.S., LIMA W.C., (1997) An evolutionary approach to simulate cognitive feedback in medical domain, in: E. Sanchez, T. Shibata and L. A. Zadeh (eds.), *Genetic Algorithms and Fuzzy Logic Systems*, Singapore, World Publishing, 193-207.

PIT L.J., (1995) *Parallel Genetic Algorithms*, MSc. Dissertation, Dept. of Computer Science, Leiden University.

SHIRAZI, B.A., HURSON, A.R., KAVI, K.M., (1995) *Scheduling and Load Balancing in Parallel and Distributed Systems.* IEEE Computer Press.

TANESE R., (1989) Distributed genetic algorithms, in: I. Schaffer, J. David (eds.), *3rd Int. Conf. on Genetic Algorithms*, San Mateo, Morgan Kauffmann, 434-439.