

Load Balancing for the Dynamic Distributed Double Guided Genetic Algorithm for MAX-CSPs

Sadok Bouamama, University of Tunis, Tunisia

Khaled Ghedira, University of Tunis, Tunisia

Nisrine Zaier, University of Tunis, Tunisia

ABSTRACT

The Dynamic Distributed Double Guided Genetic Algorithm (D³G²A) deals with Maximal Constraint Satisfaction Problems. The approach consists in creating agents cooperating together to solve problems. This paper aims to improve the D³G²A. The main purpose is to balance agent loads this distributed approach. The proposed approach will redistribute the load of Species agents more equally in order to improve the CPU time. This improvement allows not only reduction of the number of Species agents but also decreases communications agents cost. In this regard, a sub-population is composed of chromosomes violating a number of constraints in the same interval. Secondly, another proposed approach will redistribute the work load. This improvement allows diminution of inactive Species agents and it results in a balanced workloads. In fact, by analogy with social animal flocks, Species agents cooperate together to do all tasks in a reduced CPU time. Several comparisons are made about new approaches with the old version of the D³G²A. Results are promising.

Keywords: Communications, Diminution, Dynamic Distributed Double Guided Genetic Algorithm, Maximal Constraint Satisfaction Problems, Species Agents

INTRODUCTION

The constraint satisfaction problems (CSPs) formalism allow formulating and resolving many problems in artificial intelligence and operational research such as scheduling problems, resource allocation, frequency allocation, Vehicle routing problem etc.

A CSP is a triple (X, D, C) where.

- is a finite set of variables $\{x_1, x_2, \dots, x_n\}$.
- $D = \{Dx_1, Dx_2, \dots, Dx_n\}$ set of domains of possible values of variables belongs to X .
- C is a finite set of constraints of variables in X .

To resolve a CSP, we need an instantiation of all variables with values from their respective domains. This instantiation should satisfy

DOI: 10.4018/jalr.2010100105

all constraints to get a good solution. To find a CSP solution is often a hard task. In fact, with some real life problems, the solution is either too expensive to get or even nonexistent. In addition, in some cases we need to express preferences and evolution of constraints in time. For such tasks, we should find an instantiation that maximizes the number of constraints. This CSP extension, said Max-CSPs, is the topic of this paper.

Max CSPs have been dealt with by complete or incomplete methods. Complete methods such as forward checking algorithm (Freuder & Wallace, 1992) and branch and bound algorithms (Meseguer, 1997; Wallace, 1996) are shown to be able to achieve an optimal solution. Nevertheless, the combinatorial explosion is an obstacle faced by these systematic search methods. In fact, to ensure that the solution found is the optimal, systematic search algorithms would need to exhaust the entire problem space to establish that fact (Lau & Tsang, 2001).

However, incomplete ones are not able to guarantee that a solution can be found, and neither can they prove that a solution does not exist. They forgo completeness for efficiency. Many works such as (Langley, 1992) demonstrated on several large problems that what complete search algorithms fail to solve, incomplete ones efficiently conquer.

In fact, incomplete methods such as Simulated Annealing, Genetic Algorithms and Tabu Search allow escaping from local optima. Moreover, another incomplete but distributed method known as Distributed Simulated Annealing (DSA) has been shown to be efficiently applied to static Max-CSP, dynamic Max-CSP (Ghédira, 1994), Resource allocation problem (Bouamama & Ghédira, 2006). Distributing the centralized SA outperformed the solution's quality. Genetic Algorithm (GA) is a further well known incomplete method. It was frequently used in hard optimization problems (Tsang, EPK, 1993) solving multiprocessor scheduling problems (Bouamama & Ghédira, 2006). Similarly to distributing DSA, Centralized Genetic Algorithms (CGAs), known to be expensive, was also distributed to get better performances.

The result was the Distributed Guided Genetic Algorithm (DGGA) (Jlifi & Ghédira, 2003). DGGA was enhanced by a new parameter called guidance operator. The latter allowed not only diversification but also an escaping from local optima. The result is referred to as Distributed Double Guided Genetic Algorithm (D²G²A) (Bouamama & Ghédira, 2008). Moreover, a further Improvement was performed through a Dynamic aspect in D³G²A (Bouamama & Ghédira, 2006; Zaier, Bouamama, & Ghédira, 2007).

The purpose of this paper is to outperform D³G²A. The present paper describes D³G²A, its strong and weak points. Next, we set the problems and we formulate our proposition. Experimental results are presented and the performance of the two heuristics are compared and analyzed. Finally, a conclusion is given.

D³G²A

Basic Principles

D³G²A draws basically on the concept of both Species and ecological niches. The Species consists of several organisms having common characteristics whereas the ecological niche represents the task performed by a given Species (Bouamama & Ghédira, 2006). Goldberg sets that the sexual differentiation based on specialization, via both the building of Species and the exploitation of ecological niches, provides good results (Goldberg, 1989). Some methods have been settled in order to favor the building of ecological niches in GAs (Goldberg, 1989).

So, the idea here is to partition the initial population into sub-populations and to assign each one of them to an agent called Species agent. A given sub-population consists of chromosomes having their fitness values in the same range. This range, said FVR (for fitness values range), is the specificity of the Species agent Species_{FVR}. Species agents are in interaction, in order to reach an optimal solution for the problem. For this reason, each Species agent performs its own GA. The latter is guided by both template concept and min-conflict heuristic

17 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/article/load-balancing-dynamic-distributed-double/49684

Related Content

Soft-Computing-Based Optimization of Low Return Loss Multiband Microstrip Patch Antenna

Angana Sarma, Kumaresh Sarmah, Kandarpa Kumar Sarma, Sivararajan Goswami and Sunandan Baruah (2016). *Handbook of Research on Natural Computing for Optimization Problems* (pp. 583-617).

www.irma-international.org/chapter/soft-computing-based-optimization-of-low-return-loss-multiband-microstrip-patch-antenna/153830/

An Engineering Scale-Up Approach using Multi-Objective Optimization

António Gaspar-Cunha and José A. Covas (2014). *International Journal of Natural Computing Research* (pp. 17-30).

www.irma-international.org/article/an-engineering-scale-up-approach-using-multi-objective-optimization/104691/

Robust Network Services with Distributed Code Rewriting

Thomas Meyer and Christian Tschudin (2012). *Biologically Inspired Networking and Sensing: Algorithms and Architectures* (pp. 36-57).

www.irma-international.org/chapter/robust-network-services-distributed-code/58300/

Cognitively Inspired Neural Network for Recognition of Situations

Roman Ilin and Leonid Perlovsky (2010). *International Journal of Natural Computing Research* (pp. 36-55).

www.irma-international.org/article/cognitively-inspired-neural-network-recognition/41943/

Assessment of Multi-Engine Machine Translation for English to Hindi Language (MEMTEHiL): Using F&A and iBLEU Metrics

Pankaj K. Goswami, Sanjay K. Dwivedi and C. K. Jha (2016). *International Journal of Artificial Life Research* (pp. 30-45).

www.irma-international.org/article/assessment-of-multi-engine-machine-translation-for-english-to-hindi-language-memtehil/177183/