Applied Soft Computing 46 (2016) 187–203

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Parallel extremal optimization in processor load balancing for distributed applications☆

Ivanoe De Falco[a], Eryk Laskowski[b,*], Richard Olejnik[c], Umberto Scafuri[a], Ernesto Tarantino[a], Marek Tudruj[b,d]

[a] Institute of High Performance Computing and Networking, CNR, Naples, Italy
[b] Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
[c] University of Lille, CNRS, Centrale Lille, UMR 9189 – CRIStAL, F-59000 Lille, France
[d] Polish-Japanese Academy of Information Technology, Warsaw, Poland

ARTICLE INFO

Article history:
Received 15 July 2015
Received in revised form 25 April 2016
Accepted 26 April 2016
Available online 6 May 2016

Keywords:
Distributed programs
Load balancing
Extremal optimization

ABSTRACT

The paper concerns parallel methods for extremal optimization (EO) applied in processor load balancing in execution of distributed programs. In these methods EO algorithms detect an optimized strategy of tasks migration leading to reduction of program execution time. We use an improved EO algorithm with guided state changes (EO-GS) that provides parallel search for next solution state during solution improvement based on some knowledge of the problem. The search is based on two-step stochastic selection using two fitness functions which account for computation and communication assessment of migration targets. Based on the improved EO-GS approach we propose and evaluate several versions of the parallelization methods of EO algorithms in the context of processor load balancing. Some of them use the crossover operation known in genetic algorithms. The quality of the proposed algorithms is evaluated by experiments with simulated load balancing in execution of distributed programs represented as macro data flow graphs. Load balancing based on so parallelized improved EO provides better convergence of the algorithm, smaller number of task migrations to be done and reduced execution time of applications.

1. Introduction

Dynamic load balancing in parallel and distributed systems is a very important problem of computer engineering. It has accumulated a very rich bibliography, too numerous to be detailed in this paper, including a large number of survey papers [1–8].

Load balancing has been recently supported by many nature-inspired optimization methods for which some representative papers have been outlined in Section 2. Among nature inspired methods for load balancing no attention has been paid so far to extremal optimization (EO) [9] which is a fairly new optimization technique with very interesting properties. Main elements of this technique are stochastic improvements of the worst components of problem's solution represented in a way similar to a chromosome in a genetic algorithm. EO follows the approach of self-organized criticality [10], which means that the quality of a problem solution improves after a number of iterative improvements due to global auto-breeding effect in response to actions on solution worst components. Very important features of EO are low operational and memory complexities, which make it a good candidate for on-line dynamic load balancing.

The idea of modifying the worst solution elements is not unique to EO, but it is also present in the shuffled frog leaping algorithm (SFLA) [11]. However, the complexity of solution improving actions is lower in EO than in SFLA. This is because SFLA is based on a population of solution representations, structured into groups and subgroups in which sets of worst components are modified in each iteration. In EO, instead, just one solution individual exists at each generation, and only one of its components is modified at that generation. Another difference is that in EO a global fitness function and a local fitness function exist, whereas in SFLA fitness is computed for frogs (solutions) only, not for their components. Also mechanisms for acceptance of new individuals are different.

☆ This paper is an extended, improved version of the paper "Parallel Extremal Optimization with Guided State Changes Applied to Load Balancing" presented at EvoCOMNET 2015 and published in: Applications of Evolutionary Computing, Proc. of 18th European Conference, EvoApplications 2015, Copenhagen, Denmark, April 8–10, 2015, LNCS Vol. 9028, pp. 79–90, Springer 2015.

* Corresponding author. Tel.: +48 223800517.
E-mail addresses: ivanoe.defalco@na.icar.cnr.it (I. De Falco), laskowsk@ipipan.waw.pl (E. Laskowski), richard.olejnik@lifl.fr (R. Olejnik), umberto.scafuri@na.icar.cnr.it (U. Scafuri), ernesto.tarantino@na.icar.cnr.it (E. Tarantino), tudruj@ipipan.waw.pl (M. Tudruj).

http://dx.doi.org/10.1016/j.asoc.2016.04.033
1568-4946/

Classical mathematical programming approaches can be used to face the load balancing tasks. Given the non-linearity of the problem, non-linear programming procedures should be used. They have the advantage of being able to quickly detect optima. The drawback with them is that many of the nonlinear-programming solution procedures that have been developed so far do not solve the problem of optimizing a function in its generality, rather this can be achieved in a set of special cases [12]. Otherwise said, most general-purpose nonlinear programming procedures are near-sighted and can do no better than determine local maxima, apart from some special cases such as when for example the function under evaluation is convex [12].

In our previous papers [13–15], we have shown how to use EO sequential algorithms to support load balancing of processors in execution of distributed programs. In these papers, we have additionally modified the EO algorithm to improve its convergence and better deal with the time constraints posed by this real-time problem, for larger load balancing problems, for which we have noticed unsatisfactory EO behaviour. In the modified EO, we have replaced a fully random processor selection as a target of task migration in load balancing actions by stochastic selection in which the probability used in the selection mechanism is guided by some knowledge of the problem. It constitutes the essence of the EO algorithm with guided state changes (EO-GS). The guidance is based on a formula which examines how a migrated task fits a given target processor in terms of the global computational balance in the system and the processor communication loads. In [16], which is the basis of this publication, we have presented an initial approach to parallelization of EO. The algorithm was evaluated by simulation experiments in the discrete event simulation (DEVS) model [17]. The experiments have assessed the new algorithm against different parameters of the application program graphs and the kinds of load balancing algorithms.

Within the current paper we are interested in parallelized versions of the EO-based processor load balancing algorithms applied in execution of distributed programs, thus extending our previous results from [16]. Parallelization is an interesting aspect of EO, since it additionally adapts this technique towards on line dynamic optimization. In this respect, this research direction is fully convergent with this technique's low complexity. Generally speaking, parallelization of EO can be considered using two methods. The first method is to intensify the actions aiming at a possibly stronger improvement of the current EO solution, frequently with the introduction of a population-based representation or a multipoint strategy during solution improvement. It can be done using a really parallel system or modelling concurrency in a sequential system in which some components of an EO solution are identified based on multipoint selection and improved in a possibly concurrent way. The second method consists in using a population-based approach for solutions with parallel component improvement. Both approaches have accumulated already some non-negligible bibliography (see Section 2). In the mentioned papers, different parallel versions of EO were proposed based on the population-based and distributed approach including the island model. However, in these papers EO has not been applied in the context of load balancing and they do not address the use of problem knowledge-guided search viewed from a theoretical point of view.

In this paper, we use a population-based EO approach in solving a processor load balancing problem for programs represented as layered graphs of tasks. In our approach, we first identify load imbalance in the functioning of the executive system. Then, we apply a parallel (population-based) EO-GS algorithm to select tasks which are to be migrated among processors to improve the general balance of processor loads. The EO-GS algorithm is performed a given number of times in the background of the application program to find a number of best logical migrations of tasks. When the iterations are over, the physical migrations worked out by EO-GS take place. In the parallel EO-GS algorithm we define and use an additional local fitness function as a base for a stochastic selection of the best solution state in the neighbourhood of the one chosen for improvement. Additionally, we verify the use of the crossover operation (known in genetic algorithms) in the algorithm steps leading to selection of the next solution state used in further EO iterations. Performance features of the proposed approach have been assessed by experiments performed with load balancing of applications represented by layered graphs. The experiments include mutual comparisons of the proposed variants of EO-based algorithms and the algorithms tests related to scalability.

The paper is organized as follows: Section 2 surveys and discusses the state of the art in the domain of the EO-based parallel optimization methods. Section 3 presents EO principles including the guided state changes. Section 4 reports theoretical foundations for the load balancing method based on EO that we propose. Section 5 outlines the way in which we parallelize EO algorithms. Section 6 shows the experimental results which evaluate the proposed approach. Section 7 compares the proposed algorithms using a classical statistical technique based on non-parametric statistical tests.

2. State of the art

Processor load balancing is one of the most important research domains in the methodology of parallel and distributed systems. The number of papers covering this domain is beyond the possibility to be cited and discussed here, so we identify only leading relevant survey papers. Good surveys and classifications of general load balancing methods are presented in [1–3]. In [4] a survey on energy efficient load balancing algorithms for multicore processors is presented. Surveys of load balancing techniques for application in cloud computing can be found in [6,7]. In our paper we address application of a specific nature-inspired algorithm to load balancing. Nature-inspired algorithms applied to load balancing, including genetic algorithms [18–21], simulated annealing [22], swarm intelligence methods [23,24], ant colonies [25–27] and similar, have received attention in many papers. Good surveys of this subject can be found in [8,15]. Among relevant earlier papers enumerated in the surveys we have not spotted any reports on research on application of EO to processor load balancing. The only papers on this subject are [13–15], however they propose only sequential EO algorithms for these purposes.

Our current paper concerns parallel EO-based methods, hence in the rest of this section we will discuss so-far works on EO algorithms suitable for parallel implementation, presenting new methodology which has already accumulated some non-negligible bibliography. In [28] the authors propose an extended EO model called a population-based EO in which a single EO solution is replaced by a set of EO solutions which are improved using the general EO strategy. The set of these solutions is subject to parallel component selection and mutation that provides a number of solution vectors next processed in parallel.

In [29] population-based EO was applied for solving numerical constrained optimization problems. This was done on a set of six benchmark problems in the domain of constrained nonlinear programming. Both coarse and fine grain search methods were tested. The local fitness function evaluated for the components of a solution was the mutation cost measured as the sum of the deviation of the new solution after mutation from the currently known best solution plus the sum of penalties due to constraints violation. The experiments have shown a competitive performance of the proposed approach in comparison to other state of the art approaches.