Expert Systems with Applications 39 (2012) 10324–10333

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

A co-evolutionary differential evolution algorithm for solving min–max optimization problems implemented on GPU using C-CUDA

Fábio Fabris *, Renato A. Krohling *

Department of Computer Science, Federal University of Espírito Santo, Av. Fernando Ferrari, 514, CEP 29075-910, Vitória, Espírito Santo, ES, Brazil

ARTICLE INFO

Keywords:
Optimization
Differential evolution
Co-evolutionary algorithms
Graphics processing unit (GPU)
Compute unified device architecture (CUDA)
Computational performance assessment

ABSTRACT

Several areas of knowledge are being benefited with the reduction of the computing time by using the technology of graphics processing units (GPU) and the compute unified device architecture (CUDA) platform. In case of evolutionary algorithms, which are inherently parallel, this technology may be advantageous for running experiments demanding high computing time. In this paper, we provide an implementation of a co-evolutionary differential evolution (DE) algorithm in C-CUDA for solving min–max problems. The algorithm was tested on a suite of well-known benchmark optimization problems and the computing time has been compared with the same algorithm implemented in C. Results demonstrate that the computing time can significantly be reduced and scalability is improved using C-CUDA. As far as we know, this is the first implementation of a co-evolutionary DE algorithm in C-CUDA.

1. Introduction

In many design tasks in engineering, the designer is interested in robust optimal solutions. In general, optimal robust design is formulated as min–max optimization problems. Min–max optimization problems appear in many areas of science and engineering (Du & Pardalos, 1995), for instance in game theory, robust optimal control and many others. Min–max problems are considered difficult to solve. Hillis (1990), in his pioneering work, proposed a method for building *sorting networks* inspired by the co-evolution of populations. Two independent genetic algorithms (GAs) were used, whereas one for building sorting networks (host) and the other for building test cases (parasites). Both GAs evolved simultaneously and were coupled through the fitness function.

Previous studies on *co-evolutionary algorithms* (CEA) have demonstrated the suitability of the approach to solve *constrained optimization problems* (Barbosa, 1996, 1999; Krohling & dos Santos Coelho, 2006; Shi & Krohling, 2002; Tahk & Sun, 2000). A constrained optimization problem is transformed into an unconstrained optimization problem by introducing Lagrange multipliers and solving the resultant min–max problem using a CEA.

CEAs usually operate in two or more populations of individuals. In most CEAs, the fitness of an individual depends not only on the individual itself but also the individuals of other population. In this case, the fitness of an individual is evaluated by means of a competition with the members of the other population (Hillis, 1990; Paredis, 1994; Rosin & Belew, 1995, 1997). Inspired by the work of Hillis (1990), the co-evolutionary approach has been extended to solve constrained optimization problems (Barbosa, 1996, 1999; Krohling & dos Santos Coelho, 2006; Shi & Krohling, 2002; Tahk & Sun, 2000). Barbosa (1996, 1999), presented a method to solve min–max problems by using two independent populations of GA coupled by a common fitness function. Also, Tahk and Sun (2000) used a *co-evolutionary augmented Lagrangian method* to solve min–max problems by means of two populations of Evolution Strategies with an annealing scheme. The first population was made up by the vector of variables and the second one is made up of the Lagrange multiplier vector. Laskari, Parsopoulos, and Vrahatis (2002) have also presented a method using *Particle Swarm Optimization* (PSO) for solving min–max problems, but not using a co-evolutionary approach.

Krohling and dos Santos Coelho (2006) proposed a Gaussian probability distribution to generate the accelerating coefficients of PSO. Two populations of PSO using Gaussian distribution were used to solve min–max optimization problems with promising results. Cramer, Sudhoff, and Zivi (2009) proposed a co-evolutionary approach to optimize the design of a ship vessel given a set of restrictions. The initial problem was transformed to a min–max and solved by a co-evolutionary GA. Liu, Fernández, Gielen, Castro-López, and Roca (2009) used a co-evolutionary differential evolution algorithm to solve the min–max associated with a constrained non-linear optimization problem of minimizing the cost of a analog circuit design given a set of restrictions specified by

* Corresponding authors. Tel.: +55 27 3335 2136; fax: +55 27 3335 2850 (F. Fabris).

E-mail addresses: ffabris@inf.ufes.br (F. Fabris), krohling.renato@gmail.com (R.A. Krohling).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.10.015

the user. The authors reported that the design specifications were closely met, even with high constrained requirements, and the costs successfully reduced.

All these past works showed that co-evolutionary algorithms are being used with success to solve the min–max associated with constrained non-linear problems. However, the computational burden of this approach is considerable: considering two populations of n individuals each, the number of objective function calls required to evaluate one of the populations is n^2. Each individual in one population must be evaluated against each individual of the other. Evaluation of population needs to be performed many times during the optimization process and is the most relevant operation regarding computational time. So, even algorithms with fast convergence (like DE) will suffer from scalability problems if n is large or the objective function is complex. Previous works have been done in parallelizing co-evolutionary algorithms (Seredynski & Zomaya, 2002) but, as far as we know, never in context of co-evolutionary differential evolution algorithms.

The parallelization of the algorithm proposed in this work reduces the overhead of population evaluation from n^2 sequential unitary fitness evaluations to n evaluations of n functions at the same time. This is done by evaluating each individual of one population against every other in parallel. This reduces the response time and enlarges the set of tractable problems using this approach.

In this paper, two populations of independent DE are evolved: one for the variable vector, and the other for the Lagrange multiplier vector. At the end of the optimization process, the first DE provides the variable vector, and the second DE provides the Lagrange multiplier vector.

The rest of the paper is organized as follows: in Section 2, the formulation of the min–max problem is described. The standard DE is explained in Section 3. Section 4 shows the formulation of the co-evolutionary DE algorithm. In Section 5, the implementation of the co-evolutionary DE is presented to solve min–max problems. Section 6 provides simulation results and comparisons for a min–max problems and for optimization problems formulated as min–max problems followed by conclusions and directions for future research in Section 7.

2. Formulation of constrained optimization problems as min–max problems

Many problems in various scientific areas and real world applications can be formulated as constrained optimization problems. Generally, a constrained optimization problem is defined by:

$$\min_{\vec{x}\in\mathbb{R}^n} f(\vec{x}) \quad (1)$$

subject to:

$$g_i(\vec{x}) < 0, \quad i = 1,\dots,m,$$
$$h_i(\vec{x}) = 0, \quad i = 1,\dots,l,$$

where $f(\vec{x})$ is the objective function, $\vec{x} = [x_1, x_2, \dots, x_d]^T \in \mathbb{R}^d$ is the vector of variables, $\vec{g}(\vec{x})$ is the vector of inequality constraints, and $\vec{h}(\vec{x})$ is the vector of equality constraints.

The set $S \subseteq \mathbb{R}^d$ designates the search space, which is defined by the lower and upper bounds of the variables $\underline{x}_j$ and $\bar{x}_j$, respectively, with $j = 0,\dots,d$. Points in the search space, which satisfy the equality and inequality constraints, are feasible candidate solutions.

The Lagrange-based method (Du & Pardalos, 1995) is a classical approach to formulate constrained optimization problems. By introducing the Lagrangian formulation, the dual problem associated with the primal problem (1) can be written as:

$$\max_{\vec{\mu},\vec{\lambda}} L(\vec{x},\vec{\mu},\vec{\lambda}) \quad (2)$$

subject to:

$$\mu_i > 0, \quad i = 1,\dots,m,$$
$$\lambda_i > 0, \quad i = 1,\dots,l,$$

where:

$$L(\vec{x},\vec{\mu},\vec{\lambda}) = f(\vec{x}) + \vec{\mu}^T\vec{g}(\vec{x}) + \vec{\lambda}^T\vec{h}(\vec{x}) \quad (3)$$

and $\vec{\mu}$ is a $m * 1$ multiplier vector for the inequality constraints. The vector $\vec{\lambda}$ is a $l * 1$ multiplier vector for the equality constraints.

If the problem (1) satisfies the convexity conditions over S, then the solution of the primal problem (1) is the vector $\vec{x}^*$ of the saddle-point $\{\vec{x}^*, \vec{\mu}^*, \vec{\lambda}^*\}$ of $L(\vec{x}^*, \vec{\mu}^*, \vec{\lambda}^*)$ so that

$$L(\vec{x}^*,\vec{\mu},\vec{\lambda}) \leqslant L(\vec{x}^*,\vec{\mu}^*,\vec{\lambda}^*) \leqslant L(\vec{x},\vec{\mu}^*,\vec{\lambda}^*). \quad (4)$$

The saddle point can be obtained by minimizing $L(\vec{x}^*, \vec{\mu}, \vec{\lambda})$ with the optimal Lagrange multipliers $(\vec{\mu}^*, \vec{\lambda}^*)$ as a fixed vector of parameter. In general, the optimal values of the Lagrange multipliers are unknown a priori. According to the duality theorem (Du & Pardalos, 1995), the primal problem (1) subject to the inequality and equality constraints can be transformed into a dual or min–max problem.

Solving the min–max problem

$$\min_{\vec{x}} \max_{\vec{\mu},\vec{\lambda}} L(\vec{x},\vec{\mu},\vec{\lambda}) \quad (5)$$

provides the minimizer $\vec{x}^*$ as well as the Lagrange multipliers $(\vec{\mu}^*, \vec{\lambda}^*)$. However, for non-convex problems, the solution of the dual problem does not necessarily coincide with that of the primal problem. In that case, a penalty term associated with equality and inequality constraints is added to the Lagrangian function. The augmented Lagrangian is described as in Tahk and Sun (2000).

$$L_a(\vec{x},\vec{\mu},\vec{\lambda},r) = f(\vec{x}) + \sum_{i=1}^{m} p_i(\vec{x},\mu_i,r) + \vec{\lambda}^T\vec{h}(\vec{x}) + r\sum_{i=1}^{l} h_i^2(\vec{x}), \quad (6)$$

where the term p_i for the ith inequality constraint is given by

$$p_i(\vec{x},\mu_i,r) = \begin{cases} \mu_i g_i(\vec{x}) + r g_i^2(\vec{x}), & \text{if } g_i(\vec{x}) \geqslant \frac{\mu_i}{2r}, \\ -\frac{\mu_i^2}{4r}, & \text{otherwise} \end{cases} \quad (7)$$

with r being a penalty factor. It can be shown that the solution of the primal problem and the augmented Lagrangian are identical. The goal is to find the saddle-point $(\vec{x}^*, \vec{\mu}^*, \vec{\lambda}^*)$. In Section 4, co-evolutionary differential algorithm (CDE) is presented to solve min–max problems. Next, we provide some background on DE.

3. Differential evolution

The optimization procedure differential evolution (DE) was introduced in 1997 by Storn and Price (1997). Similar to other evolutionary algorithms (EAs), it is based on the idea of evolution of populations of possible candidate solutions, which undergoes very simple operations of mutation, crossover and selection.

The candidate solutions of the optimization problem in DE are represented by vectors. The components of the vectors are the parameters of the optimization problem and the set of vectors forms the population. The basic idea is the operation to generate new candidate solutions by means of a weighted difference between two vectors of the population, to which is added a third vector. All the three vectors of the population are chosen randomly. The new created vector is the trial vector. If the fitness of the trial vector is better than the fitness of a pre-chosen vector, denominated target vector, then the target vector of the population is replaced by the trial vector. Unless stated otherwise, in our study, we are considering minimization problems. Therefore, the higher