Computers & Operations Research 40 (2013) 1552–1563

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

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

Achieving super-linear performance in parallel multi-objective evolutionary algorithms by means of cooperative coevolution

Bernabé Dorronsoro [a,*], Grégoire Danoy [b], Antonio J. Nebro [c], Pascal Bouvry [b]

[a] Interdisciplinary Centre for Security, Reliability, and Trust, University of Luxembourg, 6 rue R. Coudenhove-Kalergi, L-1359, Luxembourg
[b] Faculty of Sciences, Technology and Communication, University of Luxembourg, 6 rue R. Coudenhove-Kalergi, L-1359, Luxembourg
[c] E.T.S.I. Informática, University of Málaga, Campus de Teatinos, 29071 Málaga, Spain

ARTICLE INFO

Available online 25 November 2011

Keywords:
Cooperative coevolutionary EAs
Multi-objective optimization
Parallel algorithms

ABSTRACT

This article introduces three new multi-objective cooperative coevolutionary variants of three state-of-the-art multi-objective evolutionary algorithms, namely, Non-dominated Sorting Genetic Algorithm II (NSGA-II), Strength Pareto Evolutionary Algorithm 2 (SPEA2) and Multi-objective Cellular Genetic Algorithm (MOCell). In such a coevolutionary architecture, the population is split into several subpopulations or islands, each of them being in charge of optimizing a subset of the global solution by using the original multi-objective algorithm. Evaluation of complete solutions is achieved through cooperation, i.e., all subpopulations share a subset of their current partial solutions. Our purpose is to study how the performance of the cooperative coevolutionary multi-objective approaches can be drastically increased with respect to their corresponding original versions. This is specially interesting for solving complex problems involving a large number of variables, since the problem decomposition performed by the model at the island level allows for much faster executions (the number of variables to handle in every island is divided by the number of islands). We conduct a study on a real-world problem related to grid computing, the bi-objective robust scheduling problem of independent tasks. The goal in this problem is to minimize makespan (i.e., the time when the latest machine finishes its assigned tasks) and to maximize the robustness of the schedule (i.e., its tolerance to unexpected changes on the estimated time to complete the tasks). We propose a parallel, multithreaded implementation of the coevolutionary algorithms and we have analyzed the results obtained in terms of both the quality of the Pareto front approximations yielded by the techniques as well as the resulting speedups when running them on a multicore machine.

1. Introduction

Multi-objective optimization is a discipline that deals with the simultaneous optimization of a number of functions or objectives which usually are in conflict. This means that improving one of the functions leads to decreasing the quality of (some of) the others. Therefore, solving a multi-objective problem involves the search for a set of optimal solutions, known as Pareto optimal set, instead of a unique solution as in single-objective optimization. The correspondence of this set in the objective space is known as the *Pareto front*. All the solutions in the Pareto front are *non-dominated*, meaning that none is better than the others for all the objectives (a solution *dominates* another one if the former is better than the latter for all the considered objectives). In practice, a multi-objective algorithm is expected to find a limited number of solutions, belonging to the optimal Pareto front (or as close as possible to it), and being uniformly spread throughout it. The books by Coello et al. [1] and Deb [2] are excellent references for multi-objective optimization using evolutionary algorithms (EAs), the most widely used techniques in this field.

In order to tackle problems of increasing size where "classical" EAs tend to perform poorly or are difficult to apply, researchers referred again to a nature inspired process so as to extend EAs, i.e., coevolution. In such an approach, instead of evolving one homogeneous population of individuals that represent a global solution, several subpopulations representing specific parts of the global solution either compete or cooperate during the evolution. In this paper, we focused on the Cooperative Coevolutionary model as introduced by Potter and De Jong [3], where each subpopulation evolves a part of the decision variables using a standard EA and all subpopulations evaluate complete solutions through a cooperative exchange of individuals. Performance improvements brought by such a division of the search space

* Corresponding author. Tel.: +352 4666445619; fax: +352 4666445500.
E-mail addresses: bernabe.dorronsoro@uni.lu (B. Dorronsoro), gregoire.danoy@uni.lu (G. Danoy), antonio@lcc.uma.es (A.J. Nebro), pascal.bouvry@uni.lu (P. Bouvry).

0305-0548/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2011.11.014

was already demonstrated on various single-objective problems such as [4,5]. Although Cooperative Coevolutionary EAs (CCEAs) were introduced for single-objective optimization, the cooperative coevolutionary paradigm can also be applied to the multi-objective domain.

The main contribution of this work is the design of three new CCEAs for multi-objective optimization (CCMOEA) based on three multi-objective EAs which are representative of the state-of-the-art, namely NSGA-II [6], MOCell [7], and SPEA2 [8]. The new CCMOEAs will be compared versus the original techniques on a real world combinatorial problem, i.e., the robust static mapping problem of independent tasks on Grids (RSMP). Additionally, our proposed CCMOEAs are parallelized, and their performance is also analyzed in terms of the speedup.

The paper structure is detailed next. The following section contains a brief survey on the main existing CCMOEAs. After that, Sections 3 and 4 present the reference state-of-the-art multi-objective algorithms considered in this paper and their cooperative coevolutionary design, respectively. The combinatorial optimization problem selected for comparing the performance of all the algorithms is described in Section 5. The results of all the considered algorithms, as well as their configurations, are summarized and discussed in Section 6. Finally, we conclude our work in Section 7.

2. Related work

The main difference between CCEAs [9] and EAs comes from the adaptive nature of fitness evaluation in coevolutionary systems: the fitness of an individual is based on its interactions with other individuals from other so-called subpopulations. These interactions can be either negative or positive, leading, respectively, to competitive or cooperative coevolution, this work being focused on the latter. Cooperative coevolution, as defined in the framework proposed by Potter et al. in [3], consists in splitting the solution vector into different subcomponents that are evolved in different subpopulations which periodically exchange their partial solutions. Coevolutionary EA models have been initially developed for single-objective problems optimization but more recently several multi-objective versions proved to be efficient [1], the most prominent ones being listed hereafter.

2.1. Competitive models

First, Parmee and Watson [10] proposed a competitive CCMOEA that evolves simultaneously one population per objective function to optimize. Contrary to other MOEAs, this approach aims at finding a unique trade-off solution, which could be used for preliminary design. Parmee's CCMOEA was applied on a real-world problem, i.e., air frame design, and a parallel version was implemented using the PVM (Parallel Virtual Machine) software.

In [11], Barbosa et al. proposed the interactive genetic algorithm with coevolving weighting factors (IGACW). It is based on two populations, one population of solutions and the other of weights, which evolve in a round-robin process and exchange information through a shared fitness function. Applied on a graph layout problem, IGACW is highly dependent on the user's solution ranking inputs it requires during the search, which makes it hardly comparable to any other CCMOEA.

Finally, Lohn et al. [12] proposed the Lohn's CGA, a model using two populations, one evolving solutions and the other evolving fitness functions (named objective vectors). Applied on six ZDT functions, this coevolutionary model provided better results than four standard MOEAs.

2.2. Cooperative models

Mao et al. [13] proposed an extension of the Genetic Symbiosis Algorithm (GSA) for multi-objective optimization problems. The MOGSA algorithm still uses a single population, but differs from the standard GSA with a second symbiotic parameter, which represents the interactions of the objective functions. One main drawback of this algorithm is that it requires knowledge of the search space, which highly reduces its application possibilities.

In [14], Keerativuttitumrong et al. proposed the Multi-Objective Co-operative Co-evolutionary Genetic Algorithm (MOCCGA), which combines the multi-Objective GA (MOGA) and Potter's CCGA. As defined in the CCGA framework, one subpopulation is created per decision variable and requires the best individual from each other subpopulations to complete a solution. Each subpopulation evolves using Fonseca and Flemings MOGA [20] and assigns a fitness to its individuals based on their rank in the subpopulation local Pareto front. This local Pareto optimality perception, however, leads to a limited performance of the MOCCGA demonstrated on ZDT functions. A parallel implementation of MOCCGA was empirically validated using 1, 2, 4 and 8 cores.

Maneeratana et al. proposed an extension of the MOCCGA in [15,21] which used other MOEAs (i.e., Niched Pareto GA, MOGA, NSGA and Controlled NSGA) and added a central fixed size archive, named Preserved Non-Dominated Solutions Set (PNSS), to store the non-dominated solutions. Individuals are here evaluated by assembling them with random individuals from the PNSS. Similarly, Maneeratana's MOCGA was validated on ZDT functions.

Another variant named cooperative-coevolution evolutionary algorithm was introduced by Tan et al. [16]. It introduces a coarser-grained model in which populations are grouped in so-called peer groups. Within peer groups, populations exchange two representatives (best + random) which implies twice more fitness evaluations per generations. It also uses a different archive system (one per peer group) plus a novel adaptive niching mechanism. Similar to Keerativuttitumrong, a parallel version of CCEA is proposed, but assigning one peer per computing node. Finally, [17] presented a non-dominated sorting cooperative coevolutionary algorithm (NSCCGA), which is essentially the coevolutionary extension of NSGAII. It also uses a different representatives selection by choosing a random individual from the best non-domination level.

Coello and Reyes Sierra [18] proposed a different cooperative coevolutionary approach (CO-MOEA) that subdivides the decision variables space. Each population evolves all decision variables, but on different intervals. CO-MOEA includes a dynamic population management that eliminates populations not contributing to the search. CO-MOEA provided results similar to state-of-the-art MOEAs on benchmark problems; however, its elitism strategy induces a high selection pressure which could degrade its performance on deceptive problems.

Finally, most lately Goh et al. proposed a new co-evolution model (COEA) using both competitive and cooperative methods [19,22] in which individuals from different species collaborate to solve the problem and compete at regular intervals. Its performance was assessed on various static benchmark functions and a dynamic variant, dCOEA, using a diversity scheme based on a competitive mechanism, was evaluated on dynamic problems.

Table 1 lists the aforementioned CMOEA by coevolutionary scheme employed, by EA used, by collaboration mechanism, if an archive was used (centralized or distributed), by optimization problem tackled (test function or real-world problem), and if a parallel version was implemented.

The CCMOEA framework proposed in this paper also extends Potter and De Jong's CCGA framework for multi-objective optimization, as already proposed by Keerativuttitumrong's MOCCGA