



# MOCCA-II: A multi-objective co-operative co-evolutionary algorithm



Wenjing Zhao, Sameer Alam, Hussein A. Abbass\*

School of Engineering and Information Technology, University of New South Wales Australia, UNSW-Canberra, Nortcott Drive, ACT, Australia

## ARTICLE INFO

### Article history:

Received 2 September 2012

Received in revised form 28 March 2014

Accepted 9 June 2014

Available online 3 July 2014

### Keywords:

Co-operative co-evolution  
Multi-objective optimization  
Evolutionary optimization

## ABSTRACT

Most real-world problems naturally involve multiple conflicting objectives, such as in the case of attempting to maximize both efficiency and safety of a working environment. The aim of multi-objective optimization algorithms is to find those solutions that optimize several components of a vector of objective functions simultaneously. However, when objectives conflict with each other, the multi-objective problem does not have a single optimal solution for all objectives simultaneously. Instead, algorithms attempt to search for the set of efficient solutions, known as the global non-dominated set, that provides solutions that optimally trade-off the objectives. The final solution to be adopted from this set would depend on the preferences of the decision-makers involved in the process. Hence, a decision-maker is typically interested in knowing as many potential solutions as possible.

In this paper, we propose an extension to a previous piece of work on multi-objective cooperative coevolution algorithms (MOCCA). The idea was motivated with a practical problem in air traffic management to design terminal airspaces. MOCCA and a further study that was done on a distributed environment for MOCCA, were found to fit the need for the application. We systematically questioned key components of this algorithm and investigated variations to identify a better design. This paper summarizes this systematic investigation and present the resultant new algorithm: multi-objective co-operative co-evolutionary algorithm II (MOCCA-II).

© 2014 Elsevier B.V. All rights reserved.

## 1. Introduction

Most real-world problems naturally involve multiple conflicting objectives, such as in cases when attempting to maximize both efficiency and safety of a working environment. The purpose of a multi-objective (MO) optimization algorithm is to optimize a group of objective functions simultaneously. Because of the conflict among the objective functions, the multi-objective problem usually does not have a single solution that is able to optimize all objectives. Instead it has a set of trade-off solutions, known as the set of Pareto-optimal solutions (POS). Any Pareto-optimal solution is optimal in the sense that no improvement can be made in one criterion without a degradation in at least one other criterion. The choice of a solution to be adopted depends on the preferences of the decision makers involved and various other problem-related factors. Hence, a decision-maker is typically interested in knowing as many potential Pareto solutions as possible [1].

The most straightforward way to solve a MO optimization problem is to transform it into a single objective one by taking the weighted sum of its objectives. One can then use single objective optimization methods. This approach has two profound drawbacks. First, the choice of the weight vector can highly affect the resultant solutions. Second, the method can only generate the non-dominated set – through a massive number of repeated calls of the optimization algorithm with different weights – only if the Pareto-front is convex; otherwise many solutions won't be discovered.

An evolutionary algorithm (EA) is a population-based optimization heuristic that can evaluate many solutions simultaneously. This allows for the search to be done in parallel and potentially avoids convergence to a local optimum that does not coincide with the global one. Thus, it seems natural to use EA for solving MO optimization problems.

In generic terms, there are two primary goals for a MO algorithm to achieve: guiding the search toward the global Pareto front and spreading the solutions properly throughout the Pareto front. In other words, a MO optimization algorithm is expected to find a set of non-dominated solutions, which belong to the optimal Pareto front or as close as possible to it, while spreading uniformly along it. Examples of MO evolutionary algorithms include the Niche Pareto Genetic Algorithm (NPGA) [2], the strength Pareto

\* Corresponding author. Tel.: +61 402212977; fax: +61 2 6268 8276.

E-mail addresses: [h.abbass@adfa.edu.au](mailto:h.abbass@adfa.edu.au), [husein.abbass@gmail.com](mailto:husein.abbass@gmail.com) (H.A. Abbass).

evolutionary algorithm 2 (SPEA2) [3], and non-dominated sorting genetic algorithm II (NSGAI) [4]. A class of these algorithms can self-tune its own parameters such as the Self-Adaptive Pareto Differential Evolution Algorithm [5].

Multiobjective cooperative coevolutionary algorithms promise efficient mechanisms to solve MO problems. The reciprocal interaction between different populations in a coevolutionary algorithm provide an effective strategy to explore and exploit the search space simultaneously. Our aim is to design an efficient multiobjective cooperative coevolutionary algorithm to solve computational red teaming (CRT) [6] problems in the air traffic and capability planning domains. In CRT, we attempt to evaluate solutions against problems. Therefore, the optimization problem naturally falls into two populations, one that optimizes solutions while the other attempts to discover those problems that will fail to be optimized. However, before we can use cooperative coevolution for multi-objective optimization in CRT, we needed to design an algorithm that is efficient on a set of benchmark problems with known characteristics.

In this paper, we use a systematic series of investigations to design a new multiobjective cooperative coevolutionary algorithm. The proposed method is based on work done by KC Tan and colleagues [7] on multi-objective cooperative coevolution algorithms (MOCCA). This paper will demonstrate that there are opportunities to improve on this state-of-the-art multiobjective cooperative coevolutionary algorithm.

We have chosen this algorithm for a number of reasons. Our main motivation to use cooperative evolution for multiobjective optimization stems from a real-world application that we have to solve CRT problems as we discussed above. In particular, we wish to design the terminal airspace area in the domain of air traffic management. The application requires a method that can decompose the problem. MOCCA does that by assigning each variable to a sub-population. Moreover, while many other studies simply used a classical evolutionary multiobjective optimization method and adopted it for coevolution, MOCCA was designed specifically for multiobjective optimization. The analysis done on the distributed implementation of MOCCA in [8] was also assuring in that we can have a distributed environment to solve our application. When we attempted to analyze MOCCA in more details, we saw opportunities for improving it, making it even more powerful. This initiated and motivated the development of this work.

We systematically question key components of MOCCA and investigate variations to identify a better design. We call the new algorithm, multi-objective co-operative co-evolutionary algorithm II (MOCCA-II). The results demonstrate that the proposed approach is capable of evolving solutions toward the true global Pareto-front more effectively, while maintaining a high diversity of the solution set.

The rest of this paper is organized as follows. In Section 2, background materials are presented. This is followed by a brief discussion on the co-operative coevolution evolutionary algorithm (CCEA) [7] in Section 3. Two investigations are then covered in Sections 5 and 6, and the proposed algorithm is presented in Section 4. Last, but not least, conclusions are drawn in Section 7.

## 2. Background

### 2.1. Single-objective co-operative co-evolutionary algorithms

Co-operative co-evolutionary (CoC) algorithms are population-based evolution-guided stochastic search techniques which are inspired by symbiotic interactions where different species live together in a mutually beneficial relationship [9,10]. The strategy is to decompose the objective system into many smaller modules, each of which is assigned to a species evolved (crossed over and

mutated) mostly separately, and then they are combined together to form the whole system [9,10].

To implement a typical CoC algorithm, firstly, a complex system is divided into many sub-components (a decomposition method); secondly, one individual from a certain sub-population is evaluated in combination with individuals from the other sub-populations (a collaboration mechanism); and thirdly, crossover, mutation, and other evolutionary operators are designed for each sub-component evolving in its own sub-population (an optimization operation) [11]. In a conventional CoC framework, one sub-population is being evolved while all the remaining sub-populations are held fixed [12]. The optimization of all individuals in one separate sub-population is called a *phase*, while one iteration over all sub-populations is called a *cycle*. The CoC paradigm was initially proposed for single-objective (SO) optimization, the most prominent of which are listed hereafter.

The co-operative co-evolutionary genetic algorithms (CCGAs) [13] was developed to officially introduce the initial CoC model for function optimization. In order to provide computational extensions to their previous CoC model, the authors of [14] later proposed a generalized architecture for evolving co-adapted sub-components without human involvement. The authors of [15,16] made further efforts toward selecting collaborators for individual's evaluation. A blended population approach was proposed in [12] to change the search procedure in the existing CoC algorithms. In addition, there are some studies on problem decomposition strategies in the numerical optimization domain [11,17,18]. All of this research has demonstrated that co-operative co-evolution is a promising area to tackle complex systems.

### 2.2. Multi-objective co-operative co-evolutionary algorithms

Due to many issues that are caused by the interaction of objectives within the MO optimization problem, designing co-operative co-evolutionary algorithms for MO optimization is challenging for many factors including: individual evaluation (representative selection), fitness (credit) assignment, incorporation of various elitist and niching strategies, and selection mechanisms.

Elitism is able to enhance convergence and prevent loss of good solutions once they are found in order to significantly speed up algorithmic performance [4,19]. Niching acts as a diversity-preservation mechanism by ensuring diversity in a population in order to maintain a wide variety of equivalent solutions. Under the effect of niching, a population of solutions is dynamically stable under selection pressure [20,21]. Along with the fitness assignment mechanism, selection procedures guide the search toward the true global Pareto front.

A multi-objective co-operative co-evolutionary algorithm (MOCCGA) [22] was presented to integrate the co-operative co-evolutionary effect [13] and the search mechanisms utilized in multi-objective GA [23]. An extension of MOCCA [24] was proposed to incorporate elitism in the form of a fixed size archive in each sub-population to store nondominated solutions. An extension of NSGAI [4] – nondominated sorting co-operative co-evolutionary algorithm (NSCCGA) – was then presented [25].

The co-operative co-evolutionary algorithm (CCEA) [7] makes use of the co-operative co-evolutionary mechanism specially designed for multi-objective optimization problems. An individual is evaluated by co-operating with two individuals from each sub-population, and the better one is recorded as the objective value. The fitness is assigned based on an individual's rank against the nondominated solutions stored in an external archive. Tournament selection is based on the rank and niche count. This algorithm is the first co-operative co-evolutionary approach that employs a globalized perception of elitism. Later in [8], CCEA was compared with various MOEAs on more benchmark test problems, with respect to

متن کامل مقاله

دریافت فوری ←

**ISI**Articles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات