A New Multiagent Algorithm for Dynamic Continuous Optimization

Julien Lepagnot, Université de Paris 12, France
Amir Nakib, Université de Paris 12, France
Hamouche Oulhadj, Université de Paris 12, France
Patrick Siarry, Université de Paris 12, France

ABSTRACT

Many real-world problems are dynamic and require an optimization algorithm that is able to continuously track a changing optimum over time. In this article, a new multiagent algorithm is proposed to solve dynamic problems. This algorithm is based on multiple trajectory searches and saving the optima found to use them when a change is detected in the environment. The proposed algorithm is analyzed using the Moving Peaks Benchmark, and its performances are compared to competing dynamic optimization algorithms on several instances of this benchmark. The obtained results show the efficiency of the proposed algorithm, even in multimodal environments.

Keywords: Dynamic, Metaheuristic, Moving Peaks Benchmark, Multiagent, Multimodal Environment, Non-Stationary, Optimization, Time-Varying

1. INTRODUCTION

Recently, optimization in dynamic environments has attracted a growing interest, due to its practical relevance. Many real-world problems are dynamic optimization problems (DOPs), i.e. their objective function changes over time: typical examples are in vehicle routing (Larsen, 2000), inventory management (Minner, 2003) and scheduling (Branke & Mattfeld, 2005). For dynamic environments, the goal is not only to locate the optimum, but to follow it as closely as possible. A dynamic optimization problem can be expressed by:

\[
\begin{align*}
\max \quad & f(\vec{x}, t) \\
\text{s.t.} \quad & h_j(\vec{x}, t) = 0 \; \text{for} \; j = 1, 2, \ldots, p \\
& g_k(\vec{x}, t) \leq 0 \; \text{for} \; k = 1, 2, \ldots, l \\
\vec{x} & = [x_1, x_2, \ldots, x_n]
\end{align*}
\]

(1)

where \( f(\vec{x}, t) \) is the objective function of the problem, \( h_j(\vec{x}, t) \) denotes the \( j^{th} \) equality constraint, and \( g_k(\vec{x}, t) \) are inequality constraints.
constraint and \( g_k(x, t) \) denotes the \( k^{th} \) inequality constraint. Both of them may change over time, denoted by \( t \). In this article, we focus on a dynamic optimization problem with time constant constraints.

The main approaches to deal with DOPs can be classified into the following five groups (Jin & Branke, 2005):

1. **Generated diversity after a change:** When a change in the environment is detected, explicit actions are taken to increase diversity and to facilitate the shift to the new optimum.

2. **Maintained diversity throughout the run:** Convergence is avoided all the time and it is hoped that a spread-out population can adapt to change more efficiently.

3. **Memory-based approaches:** These methods are supplied with a memory to be able to recall useful information from the past. In practice, they store good solutions in order to reuse them when a change is detected.

4. **Multipopulation approaches:** Dividing up the population into several subpopulations, distributed on different optima, allows tracking of multiple optima simultaneously and increases the probability to find new ones.

5. **Future prediction:** Recently, another kind of methods, trying to predict future changes, has attracted much attention (Rossi, Abderrahim, & Diaz, 2008). Another GA is described in (Yang, 2006), that is based on immune system and also devoted to DOPs. In (Huang & Rocha, 2005), an agent-based coevolutionary GA is proposed. This algorithm makes coevolve the way the genotype of an agent is read along with its genotype. Thus, the chromosomes are transcribed into their “edited” counterparts, using the “editors” which coevolve with them, then crossed-over and mutated. In (Mendes & Mohais, 2005), a multipopulation differential evolution (DE) algorithm is proposed, in which some techniques are added in order to increase diversity. DE is a population-based approach, its strategy consists in generating a new position for an individual according to the differences calculated between other randomly selected individuals. This algorithm is based on two main parameters, that must be correctly fitted. However in (Mendes & Mohais, 2005), these parameters are randomly generated in order to make DE easier to use.

A lot of existing dynamic optimization methods are EAs. EAs can be indeed well suited for optimization in changing environments, since they are inspired by the principles of natural evolution, and natural evolution deals very well with environmental changes. In (Rossi, Abderrahim, & Diaz, 2008), the algorithm incorporates a motion prediction technique based on Kalman filter, in order to improve the speed of optimum tracking. This kind of prediction could be suited for many problems, but it increases the complexity of the algorithm and the number of parameters. In (Tinos & Yang, 2007), the authors propose to maintain diversity in a genetic algorithm (GA) by replacing the worst individual and some others with randomly generated individuals, and maintaining their offspring in a subpopulation. However, this method is only designed for discrete problems (DDOPs). Another GA is described in (Yang, 2006), that is based on immune system and also devoted to DOPs. In (Huang & Rocha, 2005), an agent-based coevolutionary GA is proposed. This algorithm makes coevolve the way the genotype of an agent is read along with its genotype. Thus, the chromosomes are transcribed into their “edited” counterparts, using the “editors” which coevolve with them, then crossed-over and mutated. In (Mendes & Mohais, 2005), a multipopulation differential evolution (DE) algorithm is proposed, in which some techniques are added in order to increase diversity. DE is a population-based approach, its strategy consists in generating a new position for an individual according to the differences calculated between other randomly selected individuals. This algorithm is based on two main parameters, that must be correctly fitted. However in (Mendes & Mohais, 2005), these parameters are randomly generated in order to make DE easier to use.

Another widely used class of algorithms for dynamic optimization is PSO. PSO is a population-based approach, similar in some respects to EAs, except that potential solutions (particles) move, rather than evolve, throughout the search space (Blackwell & Branke, 2004). The move rules, or particle dynamics, are inspired by
Related Content

A Reinforced Tabu Search Approach for 2D Strip Packing
[www.irma-international.org/chapter/reinforced-tabu-search-approach-strip/63811/](www.irma-international.org/chapter/reinforced-tabu-search-approach-strip/63811/)

A Sociopsychological Perspective on Collective Intelligence in Metaheuristic Computing
[www.irma-international.org/article/sociopsychological-perspective-collective-intelligence-metaheuristic/40910/](www.irma-international.org/article/sociopsychological-perspective-collective-intelligence-metaheuristic/40910/)

Hybrid Cuckoo Search Algorithm for Optimal Placement and Sizing of Static VAR Compensator

Adaptive Non-Uniform Particle Swarm Application to Plasmonic Design
[www.irma-international.org/chapteradaptive-non-uniform-particle-swarm/69721/](www.irma-international.org/chapteradaptive-non-uniform-particle-swarm/69721/)