A Multi-Agent based Algorithm for Continuous Dynamic Optimization

J. Lepagnot, A. Nakib, H. Oulhadj, P. Siarry

Laboratoire Images, Signaux et Systèmes Intelligents, LISSI, E.A. 3956
Université Paris-Est - Créteil - France
julien.lepagnot@univ-paris12.fr

1 Introduction

Many real-world optimization problems are dynamic and require an algorithm that is able to continuously track a changing optimum over time. A dynamic optimization problem can be expressed as in (1), where $f(\vec{x},t)$ is the objective function of a minimization problem, $h_j(\vec{x},t)$ denotes the j^{th} equality constraint and $g_k(\vec{x},t)$ denotes the k^{th} inequality constraint. Both of them may change over time, denoted by t.

$$\begin{array}{ll} \min & f(\vec{x},t) \\ \text{s.t.} & h_j(\vec{x},t) = 0 \text{ for } j = 1,2,...,p \\ & g_k(\vec{x},t) \leq 0 \text{ for } k = 1,2,...,l \end{array} \tag{1}$$

In this paper, we present an enhanced version of the multi-agent dynamic optimization algorithm called MADO, introduced in [1, 2], for unconstrained dynamic continuous optimization. MADO is based on several specialized local searches and on archiving the optima found, to use them when the environment changes. The performances of the algorithm are evaluated through the Generalized Dynamic Benchmark Generator (GDBG) [3]. Then, we compare them to those of competing dynamic optimization algorithms available in the literature.

2 Overall scheme

The overall scheme of MADO is illustrated in Figure 1, where the agents are depicted by numbered black-filled circles in the search space S, and the neighborhood of the i^{th} agent is denoted by N_i.

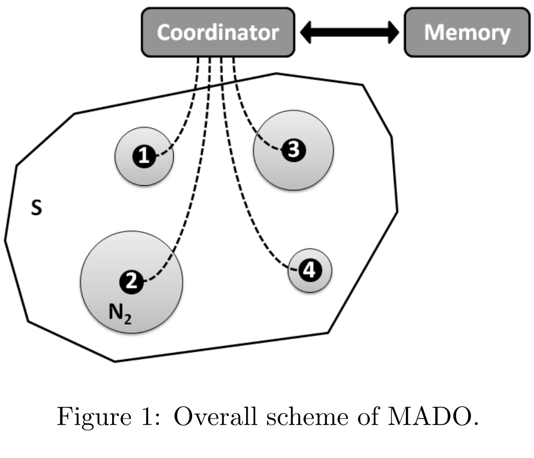

Figure 1: Overall scheme of MADO.

MADO is a multi-agent algorithm, that makes use of a population of agents to explore the search space, and to track the optima when a change occurs in the objective function. Agents perform local searches, moving from their current solution to a better one in their neighborhood, until they cannot improve their current solution, reaching thus a local optimum. The agents are coordinated by a dedicated module (the coordinator) to prevent them from searching in unpromising zones, and to force them to focus on unexplored ones of the search space, thus increasing the diversity of the solutions. Each agent performs its local search in an exclusive zone of the search space. Hence, when several agents converge to the same zone, only the one with the best current solution continues its search in this zone. Furthermore, when an agent terminates its local search (when it stagnates on its current solution), the local optimum found is transmitted to another module, dedicated to the archiving of the found optima. Then, the agent starts a new local search elsewhere, and the initial solution is given by the coordinator. The archive of local optima is used when a change is detected in the objective function, to track the archived optima.

3 Experimental results

The comparison, on GDBG, of this enhanced version of MADO with the other leading optimization algorithms in dynamic environments is summarized in Figure 2. GDBG was used during the CEC 2009 competition on dynamic optimization. It is composed of 49 test cases that correspond to the combinations of six problems with seven change scenarios. These change scenarios have different degrees of difficulty, from a small displacement of the optima to a change of the dimension of the problem. A mark is calculated for each test case in a way that favors early discovering, after each change in the objective function, of the global optimum. The sum of all marks gives a score that corresponds to the overall performance of the tested algorithm. The maximum value of this score is 100. In Figure 2, algorithms are ranked according to their overall performance. The obtained results show the efficiency of the proposed algorithm.

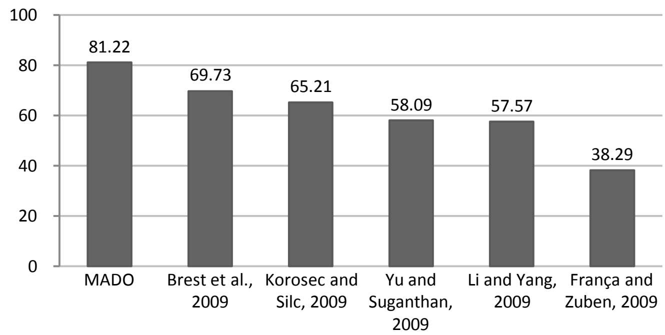

Figure 2: Comparison with competing algorithms on GDBG.

References

[1] J. Lepagnot, A. Nakib, H. Oulhadj, and P. Siarry. Performance analysis of MADO dynamic optimization algorithm, in *Proc. IEEE Int. Conf. on Intelligent Systems Design and Applications*, pages 37–42. IEEE, Pisa, Italy, 2009.

[2] J. Lepagnot, A. Nakib, H. Oulhadj, and P. Siarry. A new multiagent algorithm for dynamic continuous optimization. *International Journal of Applied Metaheuristic Computing*, 1(1):16–38, 2010.

[3] C. Li, S. Yang, T. T. Nguyen, E. L. Yu, X. Yao, Y. Jin, H.-G. Beyer, and P. N. Suganthan. Benchmark generator for CEC 2009 competition on dynamic optimization. University of Leicester, University of Birmingham, Nanyang Technological University, Tech. Rep., 2008.