Information Sciences 179 (2009) 4046–4056

Contents lists available at ScienceDirect

Information Sciences

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

# Periodical switching between related goals for improving evolvability to a fixed goal in multi-objective problems

Seppo J. Ovaska [a,*], Bernhard Sick [b], Alden H. Wright [c]

[a] Faculty of Electronics, Communications, and Automation, Helsinki University of Technology, Otakaari 5 A, FI-02150 Espoo, Finland
[b] Faculty of Informatics and Mathematics, University of Passau, Passau, Germany
[c] Department of Computer Science, University of Montana, Missoula, MT, USA

ARTICLE INFO

*Article history:*
Received 1 October 2008
Received in revised form 22 April 2009
Accepted 11 August 2009

*Keywords:*
Evolutionary computation
Evolutionary programming
Evolvability
Robustness
Varying environments
Modularity
Multi-objective optimization
Adaptive filtering

ABSTRACT

Evolutionary computation plays a principal role in intelligent design automation. Evolutionary approaches have discovered novel and patentable designs. Nonetheless, evolutionary techniques may lead to designs that lack robustness. This critical issue is strongly connected to the concept of evolvability. In nature, highly evolvable species tend to be found in rapidly changing environments. Such species can be considered robust against environmental changes. Consequently, to create robust engineering designs it could be beneficial to use variable, rather than fixed, fitness criteria. In this paper, we study the performance of an evolutionary programming algorithm with periodical switching between goals, which are selected randomly from a set of related goals. It is shown by a dual-objective filter optimization example that altering goals may improve evolvability to a fixed goal by enhancing the dynamics of solution population, and guiding the search to areas where improved solutions are likely to be found. Our reference algorithm with a single goal is able to find solutions with competitive fitness, but these solutions are results of premature convergence, because they are poorly evolvable. By using the same algorithm with switching goals, we can extend the productive search length considerably; both the fitness and robustness of such designs are improved.



## 1. Introduction

Evolutionary computation [3,6] is used increasingly as an "innovation engine" in intelligent design automation. Those methods have already created novel solutions to difficult engineering problems. Patentable innovations include a longer durable flash memory cell, an optical fiber with doubled bandwidth, and a high-performance Wi-Fi antenna [11]. The intelligent design approach is becoming more and more popular and accepted in all fields of engineering. However, it has been observed that evolutionary techniques may converge to designs that lack robustness. This is partly a consequence of using traditional ways for evaluating the fitness of individual solution candidates. Another reason for the lack of robustness is often an inefficient genotype–phenotype mapping. In many cases with a *fixed goal*, the best fitness scores may correspond to solutions that are poorly evolvable – further search is impeded, as the effects of recombination and random perturbation become increasingly unfavorable (in exploitation with a small mutation rate), or the run times turn out to be impractically long (in exploration with a large mutation rate) [15]. Earl and Deem have even verified that evolvability is a selectable trait among other desired traits [7]. The lack of evolvability leads to a lack of robustness against manufacturing and implementation tolerances, varying operation conditions, as well as disturbances and noise.

* Corresponding author. Tel.: +358 9 451 2468; fax: +358 9 451 2432.
*E-mail address:* seppo.ovaska@tkk.fi (S.J. Ovaska).

0020-0255/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2009.08.019

In nature, highly evolvable species tend to be found in rapidly and radically changing environments. Motivated by that observation, Kashtan et al. demonstrated that in evolutionary computation, *temporally varying goals* contribute, in certain cases, remarkably to speed up of evolution [9]; such varying goals help by pushing the population in a random direction, thereby rescuing it effectively from local maxima or fitness plateaus. The highest speedup was found when switching goals are related – or modular [8] – in such a way that each new goal shares some of the subproblems with the previous goal. This kind of goal switching may first appear confusing in the context of evolution, since natural evolution is usually considered as long-term, sustained directional change in species. However, Thompson emphasizes that most of evolution and coevolution is a local and geographic process that continually shifts populations in one direction and then another in a constantly changing world; and this may take place in just a few generations [19]. Therefore, periodical switching between randomly selected goals in evolutionary computation mimics the short-term evolutionary and coevolutionary processes going on in nature.

This paper is organized as follows. Temporal switching between goals in multi-objective optimization problems is first discussed in Section 2. Section 3 provides a concise introduction to the filter design problem used as a practical testbed. The applied goal-switching schemes are introduced in Section 4 with a comprehensive collection of optimization results. Section 5 closes this paper, provides a generalizing discussion, and suggests some topics for future research.

## 2. Temporal switching between goals

Many real-world design problems have multiple objectives and, thus, the ultimate fitness function is often formulated as a reciprocal of a product of individual cost functions or as a reciprocal of a sum of weighted individual cost functions:

$$F(\mathbf{x}) = \frac{1}{\prod_i C_i(\mathbf{x})} \tag{1}$$

$$F(\mathbf{x}) = \frac{1}{\sum_i w_i C_i(\mathbf{x})} \tag{2}$$

In the above equations, $\mathbf{x}$ is a variable vector in the feasible solution space and $w_i$ is a weighting factor indicating the importance of the $i$th cost function. Without a loss of generality, we consider the maximization of $F(\mathbf{x})$ and, correspondingly, minimization of $\prod C_i(\mathbf{x})$ and $\sum w_i C_i(\mathbf{x})$ throughout this paper. The individual cost functions, $C_i(\mathbf{x})$, are often conflicting, i.e., there is no simultaneous optimal solution to all of them. A Pareto optimal (also known as nondominated or noninferior) solution is one where any improvement of one cost function can be achieved only at the expense of another. It is commonly known that the traditional weighted-sum method of Eq. (2) can only solve multi-objective problems with convex Pareto fronts, and it does not work effectively with concave or discontinuous fronts.

In recent years, multi-objective evolutionary algorithms (MOEAs) has been an active field of research and development. Coello Coello presents an insightful overview of this dynamic field and gives a representative bibliography with 93 references in [5]. In MOEAs, the approximated Pareto front supplies a set of candidate solutions for decision making, and usually the user of such algorithms chooses the best-compromise solution according to his preferences. Blasco et al. proposed an illustrative graphical representation of multi-dimensional Pareto fronts to aid in identifying the best-compromise solution [1], while Sanchis et al. integrated the procedure of multi-objective optimization with *a priori* preferences [16].

Nevertheless, the motivation of our work is not to compete with the state-of-the-art multi-objective evolutionary algorithms in solutions' flexibility, but to introduce an easy-to-use and computationally efficient extension to evolutionary algorithms for improving evolvability in multimodal environments, and, simultaneously, validate the simple nature-inspired scheme of temporal goal switching in a challenging real-world application. This goal-switching extension should be seen as a fresh option that could be taken into use if problems with evolvability are experienced when single-objective algorithms are used for solving multi-objective problems with basic fitness functions of type Eq. (1) or (2). Besides, it is obvious that a majority of multi-objective real-world problems are still solved by single-objective algorithms. Thus, there is a considerable group of practicing engineers and scientists who could be interested in the following goal-switching extension. From the users' point of view, our work should not be seen as competing with the MOEAs; they both are *complementary* instead and could even be integrated together in certain applications [4].

The fitness functions of Eqs. (1) and (2) can easily be decomposed to a variety of goals that are *related* to the ultimate goal, i.e., they share some subproblems of $F(\mathbf{x})$, for example

$$F_A(\mathbf{x}) = \frac{1}{\prod_{i=1}^{K} C_i(\mathbf{x})} \text{ and } F_B(\mathbf{x}) = \frac{1}{\prod_{i=K+1}^{L} C_i(\mathbf{x})} \tag{3}$$

$$F_A(\mathbf{x}) = \frac{1}{\sum_{i=1}^{K} w_i C_i(\mathbf{x})} \text{ and } F_B(\mathbf{x}) = \frac{1}{\sum_{i=K+1}^{L} w_i C_i(\mathbf{x})} \tag{4}$$

Such related or modular goals offer potential for advantageous goal switching in which goals change temporally as suggested by Kashtan et al. [9,8]. This evolution of altering subproblems is linked to the evolutionary origin of complex features in biology, too. Already Charles Darwin reasoned that "organs of extreme perfection and complication," like the eye, are much too complex to appear *de novo* and, for that reason, they must evolve by incremental transitions through many intermediate states, sometimes undergoing changes in function [10]. Goal switching is also practiced in job rotating programs used widely