Applied Soft Computing 12 (2012) 255–265

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

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

Multi-objective Distinct Candidates Optimization: Locating a few highly different solutions in a circuit component sizing problem

Rasmus K. Ursem*, Peter Dueholm Justesen

Structural & Fluid Mechanics, Grundfos Management A/S, Poul Due Jensens Vej 7, DK-8850 Bjerringbro, Denmark

ARTICLE INFO

Article history:
Received 15 July 2010
Received in revised form 4 May 2011
Accepted 18 August 2011
Available online 10 September 2011

Keywords:
Multi-objective optimization
Many-objective optimization
Multi-criterion decision making
Distinct candidates
Diversity management

ABSTRACT

Traditional multi-objective optimization algorithms typically return several hundred non-dominated solutions. From a practical point of view, a small set of 5–10 distinct candidates is often preferred because post-processing many solutions may be too costly, too time-consuming, or it may be too difficult to compare design differences.

In this paper, we introduce Multi-objective Distinct Candidates Optimization (MODCO) as an approach to find a user-defined low number of clearly different solutions wrt. performance and design. To demonstrate the potential of the MODCO approach, we suggest the General Cluster-Forming Differential Evolution (GCFDE) algorithm and test it on five well-known mechanical engineering problems and a new five-objective constrained problem from electrical engineering – the circuit component sizing problem of the Alpha Pro pump.

The experiments showed that GCFDE significantly outperformed all competing MOEAs on the many-objective circuit problem and had slightly better performance on the mechanical problems. Furthermore, our algorithm was able to return result sets in accordance with the user's settings for result set cardinality as well as performance and design distinctiveness.

1. Introduction

Successful application of multi-objective optimization to a real-world problem typically consists of two steps. First, the optimization step where the problem is set up, the chosen algorithm is executed, and all non-dominated solutions are gathered. Second, the decision making step where the single solution to implement is chosen among the non-dominated solutions found in step 1, see Fig. 1 and Deb [[1], pp. 5].

In this process, the decision maker (DM) has to apply his preferences among the objectives to select the final solution. Veldhuizen and Lamont categorize the point in the process where the DM applies his preferences into three categories; (1) *a priori*– before the optimization is initiated, (2) *progressive*– during the optimization, and (3) *a posteriori*– after the optimization is finished [3]. Algorithms in category 1 typically transform the multi-objective problem into a single objective by specifying a utility function combining the multiple objectives. The weighted sum approach is the most widely known algorithm in this category. The progressive algorithms in category 2 usually incorporate the DM's preferences in the form of decision support systems, see [4] for a survey. Finally, category 3 algorithms exclude the DM's preferences from the search. Instead, they typically produce a large set of Pareto-optimal solutions for the DM to choose from in step 2.

The drawback of approaches 1 and 2 is that the DM has to make a choice regarding the importance of the involved objectives *prior* to the actual optimization, which may be difficult before the DM has seen any solutions. In addition, such choices are highly domain-specific and problem dependent, and algorithms are thus hard to generalize for a broader range of applications. In contrast, the traditional MO algorithms in category 3 are generally applicable. However, these algorithms produce hundreds or thousands of solutions and leave it to the DM to gather the "higher-level information" in step 2 on this set and choose the actual solution to implement. The often large set returned by *a posteriori* algorithms pose a serious problem because it may be *impossible* to gather "higher-level information" on such a large set. In short, time, money, and other reasons may prevent the application of the higher-level information gathering methods (further simulation, prototype construction, testing, etc.) on a set of more than 5–10 solutions. Consequently, we consider the current algorithms as either too domain-specific (category 1 or 2) or too general (category 3), because a huge set of candidate solutions is returned. Naturally, pruning the set using the DM's preferences is the obvious remedy for this drawback. However, this approach poses another problem because it may be difficult for the DM to state his preferences as explicit decision making rules. In our view, selection from a huge set or pruning the set tend to make the DM focus on the performance (objective space) and neglect

* Corresponding author.
E-mail addresses: ursem@cs.au.dk (R.K. Ursem), juste@cs.au.dk (P.D. Justesen).

1568-4946/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2011.08.048

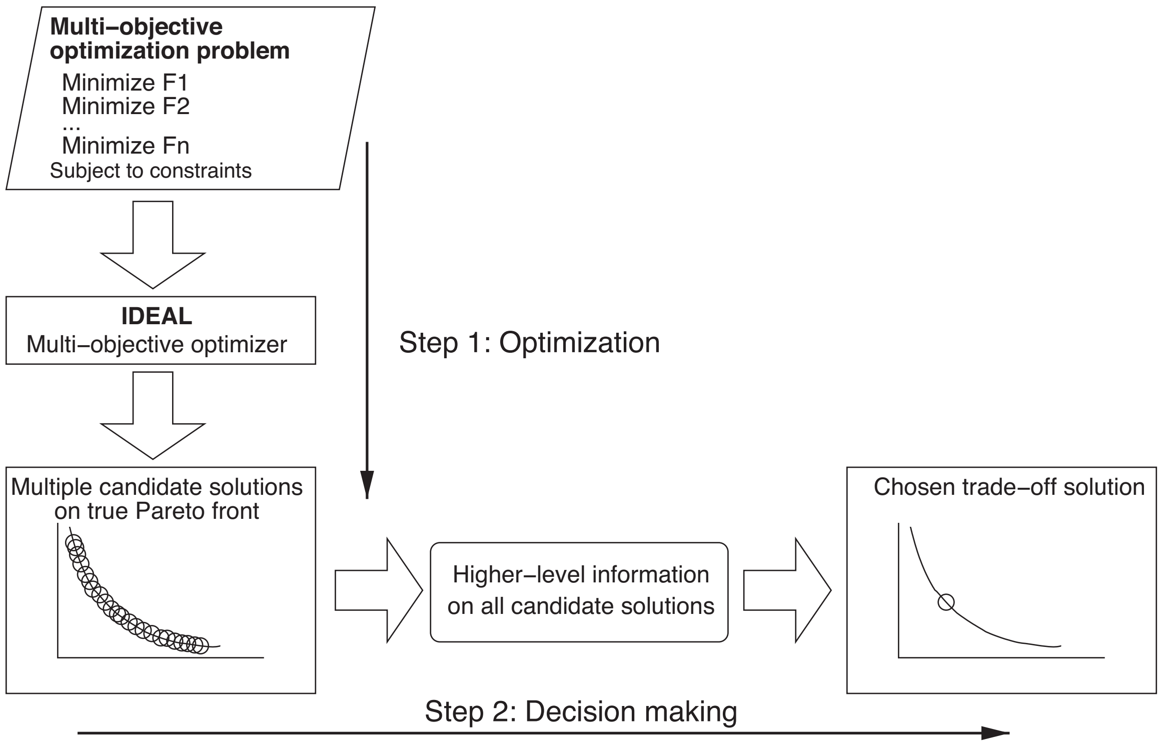

Fig. 1. Process for application of multi-objective optimization.

the design differences (search space). In contrast, only investigating a few solutions promotes a better balance between the two. Furthermore, the low number of solutions allows the DM to apply preferences, use decision rules, and evaluate objectives not stated explicitly.

The MODCO approach addresses these challenges by incorporating *generalized preferences* with the goal of finding a small set of 5–10 distinct candidates to make step 2 manageable *without* stating explicit preferences. In MODCO, the concept *generalized preferences* covers *a priori* considerations that are relevant to most if not all real-world applications. This analysis answers the following questions (further elaborated in Sections 2 and 3):

1. *Number of candidates:* $K_{NC} \in [1:\infty] \subseteq \mathbb{N}$
   How many candidates is it practically and economically feasible to inspect, analyze, and compare in post-processing?
2. *Performance distinctiveness:* $K_{PD} \in [0:1] \subset \mathbb{R}$ or $[0:1]^M \subset \mathbb{R}^M$
   How different should the candidates be performance-wise? Is it an overall distinctiveness or on certain objectives?
3. *Design distinctiveness:* $K_{DD} \in [0:1] \subset \mathbb{R}$
   How different should the candidates be design-wise?
4. *Simulator accuracy:* $K_{SA} \in [0:1]^M \subset \mathbb{R}^M$
   What is the accuracy of the involved simulators?

In MODCO, the parameters K_{NC}, K_{PD}, K_{DD}, and K_{SA} constitute the generalized preferences, and they may be implemented as the secondary selection criterion in an algorithm. Hence, MODCO algorithms aim at reducing the DM's task in step 2 by dividing the higher-level information into two groups, generalized preferences as an a priori analysis to step 1 and the *domain-specific information gathering* as a precursor to the decision making in step 2. In this approach, the domain-specific information gathering includes further investigations such as visual inspection, detailed simulation, and prototype testing on the distinct candidates followed by evaluation of the DM's implicit or explicit preferences regarding the objectives. Thus, a MODCO algorithm combines categories 1 and 3 by integrating the generalized preferences a priori and leaving the domain-specific part to a *manageable* second step a *posteriori*.

In contrast to methods with explicitly stated preferences, the MODCO approach allows incorporation of rather vague statements from the DM or domain expert. For example, a domain expert may say *"For this problem, I know that many somewhat different solutions have roughly the same performance."* In MODCO, such a statement can be transformed into $K_{PD} = 0.0$ meaning "roughly same performance" and $K_{DD} = 1.0$ or perhaps $K_{DD} = 0.5$ representing a desire for highly or somewhat different solutions.

In relation to published research, a simple classification would be to categorize multi-criterion decision making (MCDM) research as producing categories 1 and 2 optimization algorithms, and evolutionary MO research as introducing category 3 algorithms. In specific relation to the MODCO approach, the "modeling to generate alternatives (MGA)" suggested by Brill [5] address the challenge of handling non-modeled/implicit objectives by finding a small set of distinct candidates to present to the DM (for applications see [6–9]). A more elaborate survey is provided in Section 4 as it is necessary to describe the MODCO ideas before related research can be discussed.

The paper is structured as follows: Section 2 provides the motivation for the MODCO approach by summarizing 6 years of observations from real-world industrial MO problems. Section 3 lists the features of the ideal MODCO algorithm and the goals of the MODCO approach. After having introduced the MODCO ideas, we provide a survey of related research in Section 4. Section 5 introduces our novel MODCO algorithm. In Section 6 we demonstrate its usefulness on a set of well-known mechanical engineering problems and a real-world circuit component sizing problem provided by Grundfos. Finally, Section 7 concludes the paper.

2. Motivation for MODCO algorithms

The application of multi-objective optimization in an industrial context raises a number of interesting challenges, dilemmas,