The Multiple Multi Objective Problem – Definition, Solution and Evaluation

Wolfgang Ponweiser and Markus Vincze

Vienna University of Technology
Automation and Control Institute
Gusshausstr. 27-29/E376, A-1040 Vienna, Austria
{ponweiser,vincze}@acin.tuwien.ac.at
http://www.acin.tuwien.ac.at*

Abstract. Considering external parameters during any evaluation leads to an optimization problem which has to handle several concurrent multi objective problems at once. This novel challenge, the Multiple Multi Objective Problem M-MOP, is defined and analyzed. Guidelines and metrics for the development of M-MOP optimizers are generated and exemplary demonstrated at an extended version of Deb's NSGA-II algorithm. The relationship to the classical MOPs is highlighted and the usage of performance metrics for the M-MOP is discussed. Due to the increased number of dimensions the M-MOP represents a complex optimization task that should be settled in the optimization community.

Keywords: Multiple Multi Optimization Problem M-MOP, Performance Evaluation, Genetic Optimization.

1 Introduction

Since most practical problems are characterized by contradicting targets the extension to optimize several objectives has got more and more attention. The purpose of this type of problem, the *multi objective problem MOP*, is to find the so called Pareto Set, containing all non-dominated solutions [1].

This paper presents a novel extenion to MOP, the so called *multiple multi objective problem M-MOP*. The challenge is to find the optimal solutions for several similar MOP problems at once. Since all the single problems are itself MOPs, optimality is still Pareto optimality. The similarity of the problems is, that they share exactly the same input space, the decision space.

A domain that naturally leads to a M-MOP is any kind of evaluation. A MOP results from the evaluation of an algorithm by a set of configuration parameters. Whereas these parameters represent the input (the decision space) the evaluated performance represents the output (the objective space). Due to the fact that the final preferences between the several performance metrics are usually not known

* This work is supported by projects S9101, S9103 and S9106 of the Austrian Science Foundation.

S. Obayashi et al. (Eds.): EMO 2007, LNCS 4403, pp. 877–892, 2007.

at evaluation time, this results in a classical MOP. According to Coello [2] this is called 'a posteriori' preference articulation.

In fact the resulting performance further depends on some external parameters which are properties of the test site. These properties are further called *context*. To enable the user of the evaluation data to adjust the evaluation result to fit its own application the explicit consideration of this context is required. In cases where these context parameters can be configured concurrently on the test site a single evaluation run delivers several results for each of the realized contexts (see Fig. 1). Consider that a single evaluation run has just a single input vector. Every relationship between this input and the output of a single context is the task of a single MOP. In contrary the relation between the single input and the multiple outputs is the task of a multiple MOP, the M-MOP.

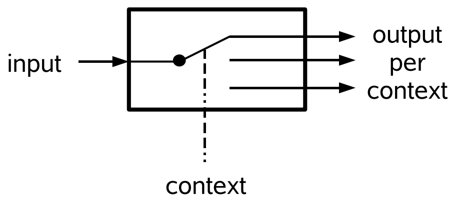

Fig. 1. The incorporation of external parameters (the context) as generator of *multiple* multi objective output

A typical example is the evaluation of object detection algorithms in the computer vision domain [3]. Such algorithms search for pre-specified objects in an image. There exist a lot of different algorithms [4,5,6] and evaluation endeavors [7,8] for this type of algorithms. Obviously their performance depends not only on the configuration of the algorithms itself, but additionally on the properties of the images and the objects used in the test databases. These properties build the context of the evaluation.

For the integration of an object detection algorithm it is important that the following requirements are specified:

- The performance characteristic that needs to be met. Typical examples for object detection algorithms are values for success rate, processing time and spatial accuracy.
- The expected context of the application. For the object detection task typical examples are the appearance and expected size of the objects to detect, lighting conditions and image noise.

In the same way as the requirements are specified the evaluation has to represent its results. Consequently a contex sensitive evaluation is required.

At first glance the context can be integrated as additional input. Due to the mechanisms of optimization methods bad context conditions will never be analyzed. Another approach is to simply expand the output by the context values. This leads to an optimization of the context itself, which is not the intention of the entire challenge. A third approach is the exhaustive calculation