8

MOEA Parallelization[1]

> One friend in a lifetime is much, two are many, three are hardly possible. Friendship needs a certain parallelism of life, a community of thought, a rivalry of aim.
>
> Henry B. Adams

8.1 Introduction

Successfully engineering Multiobjective Evolutionary Algorithms (MOEAs) involves thoroughly addressing many different issues. However, the performance concepts of efficiency and effectiveness are paramount. MOEAs are stochastic, population-based computational procedures mimicking evolutionary concepts and operations in attempts to find satisfactory, if not optimal, solutions of problems with multiple objectives. Evolutionary Algorithms (EAs) and MOEAs are adaptive stochastic search techniques classified under the umbrella of soft computing [1577]; generic EAs such as Genetic Algorithms, Evolution Strategies, Evolutionary Programming, and Genetic Programming are all successfully used in MOEA implementations [265].

Satisfactorily solving specific and ever-larger Multiobjective Optimization Problems (MOPs) with MOEAs is an increasingly popular goal reflected by the large number of recent research efforts attacking pedagogical and real-world problems with these algorithms [266, 361]. Such real-world MOPs typically involve highly constrained design optimization tasks with high computational cost. Through these applications, MOEA structures continue to evolve into effective (i.e., "useful") search algorithms. Once convinced of a MOEA's effectiveness (how well it solves the problem), the researcher is then often interested in increasing its efficiency (how "quickly" or "cheaply" it solves the

[1] The authors wish to thank Jesse Zydallis for his permission to include previous collaborative effort in this discussion on MOEA parallelization.

problem). The desire to reduce execution time and/or resource expenditures naturally leads to considering the use of parallel and distributed processing techniques.

A major computational bottleneck in many contemporary MOEA applications (as well as in other numerical or real-world design/optimization problems) is the calculation of complex nonlinear MOP functions, implying algorithmic parallelization may improve computational efficiency. Just as in single-objective optimization, multiple "expensive" objective function evaluations (in terms of CPU time) are often completed in less wall clock time by decomposing the computational load across two or more processors. Evaluating more solutions in the same or reduced time may then result in a larger or higher-fidelity representation of possible outcomes. This may be especially productive in MOEA applications, due primarily to the fact that identifying a (possibly large) set of "good" objective vectors is often the primary goal driving search. A parallel MOEA (pMOEA) might then be the preferred EA implementation for solving complex real-world applications where (multiple) objective function evaluations are the computational bottleneck.

Given a MOEA's obviously inherent parallelism as well as the relative ease of gaining access to contemporary multi-processor computing platforms, interest is increasing for developing pMOEAs (see [266]). However, in publications solving engineering design and numerical optimization problems with pMOEAs, very few discuss algorithmic development issues and for the most part ignore the parallel aspects of the implementation. In general, these papers lack a thorough presentation of pMOEA employment rationale, algorithmic settings and structure, test problem selection and metrics, etc. Yet, analyzing the current corpus yields several key insights into the current pMOEA state-of-the-art and suggests areas in which further research may be focused. The most significant findings from this analysis are detailed later in this chapter.

Critical engineering practice requires a disciplined approach as embodied in the following quote: "...the essence of sound engineering [lies] in clearly stating the assumptions upon which calculations are based so that they may be checked at all times for lapses in logic and other errors. It is this imperative that engineering premises be set down clearly, and that the calculations that follow be systematically and unambiguously presented, so that they may be checked by another engineer with perhaps a different perspective on the problem [1270, p. 44]." With that thought in mind, this chapter clearly presents pMOEA symbolic formulations, describes pMOEA design and implementation issues, proposes options for satisfactory issue resolution and discusses various practical considerations. Known research approaches and various insights gained from analysis are also integrated into the presentation. Throughout this discussion a template is evolved for generating a pMOEA from either an existing (MO)EA or from first principles. The result is a generic design plan with a list of parameter considerations to be considered in designing and implementing efficient and effective pMOEAs, regardless of application problem domain. Perhaps this presentation might in-