2

MOP Evolutionary Algorithm Approaches

An algorithm must be seen to be believed.

Donald Knuth

2.1 Introduction

Both researchers and practitioners in science, engineering, government, and industry certainly have a strong interest in knowing state-of-the-art multi-objective optimization techniques. For researchers, this is the normal procedure to trigger new and original algorithmic contributions. For practitioners, this knowledge allows them to choose the most appropriate algorithm(s) for their specific multi-objective problem (MOP) domain application. From the decision maker's (DM) perspective, it is desired that only a "few" solutions are available for ease of decision. Thus, as presented in Chapter 1, one is attempting to optimize a vector objective function possibly with constraints resulting in trade-offs between the multiple objectives. This chapter employs the various generic mathematical definitions defined in Chapter 1 for discussing multi-objective evolutionary algorithm (MOEA) design.[1] It is desired that an MOEA generates MOP solutions in P_{true} which provide a trade-off of performance (efficiency, effectiveness) for specific system model objectives (cost/profit, constraints, etc.) that may mutually conflict. For example, the classical multiobjective knapsack problem (profit and weight) and drug development (cost vs. effectiveness) represent vectors of two objectives. Maximizing one objective such as profit usually does not optimize another such as reliability. Many contemporary real-world MOP applications for the practitioner's

[1] Note that some MOEA researchers and practitioners use the phrase "Multi-Objective Optimization Problem" (*MOOP*) and "Multi-Objective Optimization" (*MOO*) to associate with the field, instead of MOP and MOEA.

and researcher's critical analysis are discussed in Chapter 7 and [277] with many examples reflected in the current MOEA literature.[2]

Since Evolutionary Algorithms and MOEAs in particular can encode individual solutions in numerous straightforward representations (chromosome data structures) as well as directly compute associated objective values, they have a considerable robust advantage over traditional MOP search techniques (see Chapter 1). That is, traditional techniques may impose restrictions or complex mappings on the problem domain or algorithm domain mathematical model in order to solve the problem. Of course, the No Free Lunch Theorem (NFL) [1708] implies that a MOEA is not a universal robust solution technique for all MOPs. But, MOEAs generally can easily be guided by problem domain information, not having to modify the problem domain model for use with MOEAs. Then, the search process is easier to develop, understand and test in its native form for a given application [1102].

Achieving the exact Pareto front of an arbitrary problem is usually quite difficult. Nevertheless, reasonably good approximations of PF_{true} are generally acceptable within limited computational time (see Chapter 1 for associated notation). MOEAs by definition attempt to find these acceptable but approximate Pareto fronts and Pareto optimal solutions within some implicit or explicit error measure (see Chapter 5).

This chapter addresses the many issues involved in MOP domain and MOEA domain integration from a design perspective. In particular, historic and generally used (MOEA) approaches such as the NSGA [1509, 374], PAES [886], SPEA [1782, 1775], and the MOMGA [1626, 1629, 1790] are detailed and analyzed. In the discussion of various MOEAs, each algorithm is catalogued by recording key elements of its approach, and classified using the structure defined in Chapter 1. The chapter also presents a generic MOEA algorithmic formulation based upon basic evolutionary operators. Related to this generic form, an analysis of currently known MOEA algorithmic design research is given. Many relevant meta-level topics are addressed, highlighting MOEA design concerns which have limited treatment in the literature. For example, discussed are dominance operator differences, diversity operator variations, population structures, impact of MOEA fitness function characteristics, lack of MOEA theory, MOEA chromosomal representations, utility of explicit vs. implicit building block approaches, and other selected topics.

Fundamental MOEA techniques and MOEA design goals along with a generic MOEA structure are presented in Section 2.2. Specific MOEA pseudo code and associated performance is discussed in Section 2.3. Constraint-handling techniques are briefly discussed in Section 2.4. Critical MOEA elements are described in Section 2.5. This leads to Section 2.6 which recapitu-

[2] For an up-to-date list of references on evolutionary multi-objective optimization, visit the EMOO repository located at: `http://delta.cs.cinvestav.mx/~ccoello/EMOO` with a mirror at: `http://www.lania.mx/~ccoello/EMOO`