Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art

David A. Van Veldhuizen
Air Force Research Laboratory
Optical Radiation Branch
Brooks AFB, TX 78235, USA
david.vanveldhuizen@brooks.af.mil

Gary B. Lamont
Department of Electrical and
Computer Engineering
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433, USA
gary.lamont@afit.af.mil

Abstract

Solving optimization problems with multiple (often conflicting) objectives is, generally, a very difficult goal. Evolutionary algorithms (EAs) were initially extended and applied during the mid-eighties in an attempt to stochastically solve problems of this generic class. During the past decade, a variety of multiobjective EA (MOEA) techniques have been proposed and applied to many scientific and engineering applications. Our discussion's intent is to rigorously define multiobjective optimization problems and certain related concepts, present an MOEA classification scheme, and evaluate the variety of contemporary MOEAs. Current MOEA theoretical developments are evaluated; specific topics addressed include fitness functions, Pareto ranking, niching, fitness sharing, mating restriction, and secondary populations. Since the development and application of MOEAs is a dynamic and rapidly growing activity, we focus on key analytical insights based upon critical MOEA evaluation of current research and applications. Recommended MOEA designs are presented, along with conclusions and recommendations for future work.

Keywords

Multiobjective optimization, multiobjective evolutionary algorithms, multiobjective genetic algorithms, Pareto optimality.

1 Introduction

Solving multiobjective scientific and engineering problems is, generally, a very difficult goal. In these particular optimization problems, the objectives often conflict across a high-dimensional problem space and may also require extensive computational resources. General multiobjective optimization problem (MOP) solution methods range from linear objective function aggregation to Pareto-based techniques. In an attempt to stochastically solve problems of this generic class in an acceptable timeframe, specific multiobjective evolutionary algorithms (MOEAs) were initially developed in the mid-eighties for application to the MOP domain. Since then, a forty-fold increase in the number of MOEA publications has seen various solution techniques proposed, along with applications in numerous scientific and engineering disciplines (Van Veldhuizen, 1999; Coello, 1999b).

Our discussion's intent is to rigorously define MOPs and certain related concepts, present an MOEA classification scheme, and evaluate the variety of contemporary MOEAs. Current MOEA theoretical developments are evaluated; specific topics addressed include fitness functions, Pareto ranking, niching, fitness sharing, mating restriction, and secondary populations. We also discuss complexity models and recommend specific MOEA designs for use in new applications and performance comparisons. As space is limited, the reader desiring introductory MOEA material is directed to the following literature: Van Veldhuizen (1999), Coello (1999a), and Fonseca and Fleming (1995a). Because the development and application of MOEAs is a dynamic and rapidly growing interdisciplinary field, we focus here on evolving key analytical insights derived from critically evaluating current MOEA research and applications.

The remainder of this discussion is arranged as follows. Section 2 introduces relevant MOP concepts, and Section 3 lists the field's major literature surveys and presents an MOEA classification scheme. Sections 4 and 5 address key quantitative and qualitative analytical results from our research. Section 6 briefly describes recommended MOEA designs, and Section 7 summarizes our discussion and presents recommendations for future work.

2 MOP Definition and Overview

Neither the problem nor algorithm domains considered within this research is straightforward. Thus, we present key concepts defining and bounding both the problem class (MOPs) and algorithms selected to solve them (MOEAs). Although single-objective optimization problems may have a unique optimal solution, MOPs (as a rule) present a possibly uncountable *set* of solutions that, when evaluated, produce vectors whose components represent trade-offs in objective space. A decision maker then implicitly chooses an acceptable solution (or solutions) by selecting one or more of these vectors. MOPs are mathematically defined as follows:

DEFINITION 1 **(General MOP)**: *In general, an MOP minimizes* $F(\vec{x}) = (f_1(\vec{x}), \ldots, f_k(\vec{x}))$ *subject to* $g_i(\vec{x}) \leq 0,\ i = 1, \ldots, m,\ \vec{x} \in \Omega$. *An MOP solution minimizes the components of a vector* $F(\vec{x})$, *where* $\vec{x}$ *is an* n*-dimensional decision variable vector* ($\vec{x} = x_1, \ldots, x_n$) *from some universe* Ω.

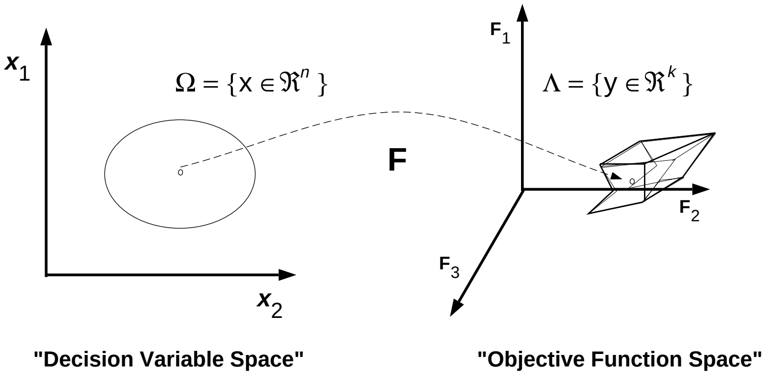

x_1
$\Omega = \{x \in \Re^n\}$
x_2
F
F_1
$\Lambda = \{y \in \Re^k\}$
F_2
F_3
"Decision Variable Space"
"Objective Function Space"

Figure 1: MOP evaluation mapping.

An MOP thus consists of n decision variables, m constraints, and k objectives of which any or all of the objective functions may be linear or nonlinear (Hwang and Masud, 1979). The MOP's evaluation function, $F : \Omega \longrightarrow \Lambda$, maps decision variables ($\vec{x} = x_1, \ldots, x_n$) to vectors ($\vec{y} = a_1, \ldots, a_k$). This situation is represented in Figure 1 for the case $n = 2$, $m = 0$, and $k = 3$. This mapping may or may not be onto some region of objective function space, depending upon the functions and constraints composing the particular MOP. Furthermore, all problems discussed in this paper are assumed to be minimization problems unless otherwise specified (since $\min\{F(x)\} = -\max\{-F(x)\}$), and to be computable.

MOPs are characterized by distinct measures of performance (the objectives) that may be (in)dependent and/or incommensurable. For example, a radar antenna's gain and input resistance may have little dependence on each other; they are also measured in different units (dB vs. ohms). The multiple objectives being optimized almost always conflict, placing a partial, rather than total, ordering on the search space. In fact, finding the global optimum of a general MOP is NP-Complete (Bäck, 1996, 56). "Perfect" MOP solutions, where all decision variables satisfy associated constraints and the objective functions attain a global minimum, may not even exist. We represent an MOP's goals or objectives as distinct mathematical functions to be achieved and use the terms *objective space* or *objective function space* to denote the coordinate space within which vectors resulting from evaluating MOP solutions are plotted.

MOPs may require specialized optimization techniques due to these characteristics (multiple, conflicting objectives and constraints). Regardless of implemented technique, a key concept many researchers use in determining a set of MOP solutions is that of *Pareto optimality*.

2.1 Pareto Concepts

Although Pareto optimality and its related concepts and terminology are frequently invoked, MOEA researchers often erroneously use them in the literature. To ensure understanding and consistency we thus define Pareto dominance, Pareto optimality, the Pareto optimal set, and the Pareto front. Examples of these concepts are found elsewhere (e.g., Van Veldhuizen (1999)). Using the MOP notation presented in Definition 1, these key Pareto concepts are mathematically defined as follows:

DEFINITION 2 **(Pareto Dominance):** *A vector* $\vec{u} = (u_1, \ldots, u_k)$ *is said to dominate* $\vec{v} = (v_1, \ldots, v_k)$ *(denoted by* $\vec{u} \preceq \vec{v}$*) if and only if* u *is partially less than* v*, i.e.,* $\forall i \in \{1, \ldots, k\}$, $u_i \leq v_i \wedge \exists i \in \{1, \ldots, k\} : u_i < v_i$.

DEFINITION 3 **(Pareto Optimality):** *A solution* $x \in \Omega$ *is said to be Pareto optimal with respect to* Ω *if and only if there is no* $x' \in \Omega$ *for which* $\vec{v} = F(x') = (f_1(x'), \ldots, f_k(x'))$ *dominates* $\vec{u} = F(x) = (f_1(x), \ldots, f_k(x))$*. The phrase "Pareto optimal" is taken to mean with respect to the entire decision variable space unless otherwise specified.*

DEFINITION 4 **(Pareto Optimal Set):** *For a given MOP* $F(x)$*, the Pareto optimal set (*$\mathcal{P}^*$*) is defined as:*

$$\mathcal{P}^* := \{x \in \Omega \mid \neg\exists\, x' \in \Omega \;:\; F(x') \preceq F(x)\} \tag{1}$$

DEFINITION 5 **(Pareto Front):** *For a given MOP* $F(x)$ *and Pareto optimal set* $\mathcal{P}^*$*, the Pareto front (*$\mathcal{PF}^*$*) is defined as:*

$$\mathcal{PF}^* := \{\vec{u} = F(x) = (f_1(x), \ldots, f_k(x)) \mid x \in \mathcal{P}^*\} \tag{2}$$

Pareto optimal solutions are also termed *non-inferior*, *admissible*, or *efficient* solutions (Horn, 1997); their corresponding vectors are termed *nondominated*. These solutions may have no clearly apparent relationship besides their membership in the Pareto optimal set. They form the set of all solutions whose corresponding vectors are nondominated with respect to all other comparison vectors; we stress here that Pareto optimal solutions are classified as such based on their evaluated functional values. When plotted in objective space, the nondominated vectors are collectively known as the Pareto front. To restate, the Pareto optimal set is a subset of all possible solutions in Ω. Its evaluated objective vectors form the Pareto front, of which each vector is nondominated with respect to all objective vectors produced by evaluating all possible solutions in Ω.

Note that the decision maker (DM) is often selecting solutions via choice of acceptable objective performance, represented by the (known) Pareto front. Choosing an MOP solution that optimizes only one objective may well ignore solutions that, from an overall standpoint, are "better." The Pareto optimal set contains those better solutions. Identifying a set of Pareto optimal solutions is thus key for a DM's selection of a "compromise" solution(s) satisfying the objectives as best possible. Of course, the accuracy of the DM's view depends on both the *true* Pareto optimal set and the set presented *as* Pareto optimal.

We note here that derived solutions of real-world MOPs often offer only a finite number of points that may or may not be truly Pareto optimal. Any time the real world is modeled (e.g., via objective functions) upon a computer (a discrete machine), there is a fidelity loss between reality's uncountable infinity and the implemented finite, discretized model. Complex MOPs do not, generally, lend themselves to analytical determination of the actual Pareto front, thus making even a computational approximation of an MOP's global optimum difficult. We have elsewhere defined an MOP's globally optimum solution set to be the Pareto optimal set (Van Veldhuizen, 1999); most MOEA researchers implicitly accept this definition as they explicitly search for an MOP's Pareto front.

2.2 Pareto Notation

An MOEA's algorithmic structure (e.g., multiple *unique* populations) can easily lead to confusion when identifying or using Pareto concepts. In fact, MOEA researchers have erroneously used Pareto terminology in the literature, suggesting a more precise notation is required.

During MOEA execution, a "current" set of Pareto optimal solutions (with respect to the *current* MOEA generational population) is determined at each MOEA generation's end and termed $P_{current}(t)$, where t represents the generation number. Many MOEA implementations also use a secondary population storing nondominated solutions found through the generations (Van Veldhuizen, 1999) (see also Section 5.2). Because a solution's classification as Pareto optimal depends upon the context within which it is evaluated (i.e., the given set Ω of which it's a member), corresponding vectors of this secondary population must be (periodically) tested and solutions whose associated vectors are dominated removed.

We term this secondary population P_{known} (t). This term is also annotated with t to reflect its possible changes in membership during MOEA execution. P_{known} (0) is defined as the empty set ($\emptyset$) and P_{known} alone as the *final* set of solutions returned by the MOEA at termination. Different secondary population storage strategies exist; the simplest is when $P_{current}$ (t) is added at each generation (i.e., $P_{current}$ $(t) \cup P_{known}$ $(t-1)$). At any given time, P_{known} (t) is thus the set of Pareto optimal solutions *yet found by the MOEA through* *generation t*. Of course, the *true* Pareto optimal set (termed P_{true}) is not explicitly known for problems of any difficulty. P_{true} is implicitly defined by the functions composing an MOP – it is fixed and does not change. Note that because of the manner in which Pareto optimality is defined, P_{true} and $P_{current}(t)$ are always nonempty solution sets (Van Veldhuizen, 1999).

$P_{current}(t)$, P_{known}, and P_{true} are sets of MOEA genotypes;[1] each set's corresponding phenotypes form a Pareto front. We term the associated Pareto front for each of these solution sets as $PF_{current}(t)$, PF_{known}, and PF_{true}. Thus, when using an MOEA to solve MOPs, the implicit assumption is that one of the following holds: $PF_{known} = PF_{true}$, $PF_{known} \subset PF_{true}$, or where distance is defined over some norm (Euclidean, RMS, etc.), $\{\vec{u}_i \in PF_{known}, \vec{u}_j \in PF_{true} \mid \forall i,\ \forall j\ \min[distance(\vec{u}_i, \vec{u}_j)] < \epsilon\}$.

3 MOEA Reviews and Classifications

MOEAs are receiving renewed interest from EA researchers. Although the first MOEA was published in the mid-eighties (Schaffer, 1985), and a substantial MOEA literature has since developed (over 450+ publications (Coello, 1999b)), there have been only four notable surveys published. The reviews by Fonseca and Fleming (1995a) and Horn (1997) quickly examine major MOEA techniques. The former additionally provides many relevant MOP issues from an MOEA perspective. Both classify existing MOEA approaches differently – Fonseca and Fleming from a broad algorithmic perspective, and Horn from a DM's. More recently, Coello (1999a) presents an MOEA review classifying implementations from a detailed algorithmic standpoint, discussing the strengths and weaknesses of each technique.

Van Veldhuizen (1999) expands upon these reviews by classifying and cataloging currently known MOEA efforts, considering more recent and related MOEA citations. Each citation therein is cataloged by recording key elements of its approach (e.g., problem domain, number and type of fitness functions, genetic representation) and classified using the structure defined in Figure 2. This database contains a major share of the currently identified MOEA-based citations from the literature. We then use this survey as the basis for an extensive analysis of key MOEA issues. The cataloged presentation highlights previously unnoticed MOEA research trends, clearly distinguishes the various implemented techniques, and identifies distinctive characteristics of each. The remainder of this section briefly describes the classification used.

Many successful MOEA approaches are predicated upon previously implemented mathematical MOP solution techniques. For example, operations researchers proposed several methods well before 1984 (Hwang and Masud, 1979; Van Veldhuizen, 1999). Their multiple objective decision making problems are closely related to design MOPs. These problems' common characteristics are a set of quantifiable objectives, a set of well-defined constraints, and a process of obtaining trade-off information between the stated objectives (and possibly also between stated or nonstated nonquantifiable objectives) (Hwang and Masud, 1979). The various multiple objective decision making techniques are commonly classified from a DM's point of view (i.e., how the DM performs search and decision making). We consider the DM to be either a single DM or a group, but a group united in its decisions.

Because the set of solutions a DM is faced with often represents "compromises" between

[1] Horn (1997) uses the terms P_{online}, $P_{offline}$, and P_{actual} instead of $P_{current}(t)$, P_{known}, and P_{true}. Our notation is more precise as it allows for generational specification. It also encompasses each set's corresponding Pareto front. Note that $P_{true} = \mathcal{P}^*$ and $PF_{true} = \mathcal{PF}^*$.

the multiple objectives, some specific compromise choice(s) must be made from the available alternatives. Thus, the final MOP solution(s) results from both *optimization* (by some method) and *decision* processes. We then choose to classify MOEA-based MOP solution techniques from a DM's perspective, defining three variants of the decision process by which the final solution(s) results from a DM's preferences being made known either before, during, or after the optimization process. This is more formally declared as follows (Hwang and Masud, 1979):

A Priori **Preference Articulation.** (*Decide* ⟶ *Search*) DM combines the differing objectives into a scalar cost function. This effectively makes the MOP single-objective prior to optimization.

Progressive **Preference Articulation.** (*Search* ⟷ *Decide*) Decision making and optimization are intertwined. Partial preference information is provided upon which optimization occurs, providing an "updated" set of solutions for the DM to consider.

A Posteriori **Preference Articulation.** (*Search* ⟶ *Decide*) DM is presented with a set of Pareto optimal candidate solutions and chooses from that set. However, note that most MOEA researchers search for and present a set of nondominated vectors (PF_{known}) to the DM.

Basic techniques below the top level of this technique hierarchy may be common to several algorithmic research fields; we limit discussion to implemented MOEA techniques. A hierarchy of the known MOEA techniques is shown in Figure 2, where each is classified by the different ways in which the fitness function and/or selection is treated. Although some may not agree with our detailed classification, perhaps preferring a simpler one (e.g., Fonseca and Fleming's (1995a) tripartite one), our intent is to formalize an algorithmic framework for the important and rapidly expanding research in MOEAs. We prefer this more detailed view as it directly reflects specific algorithmic approaches and brings to light otherwise unseen trends.

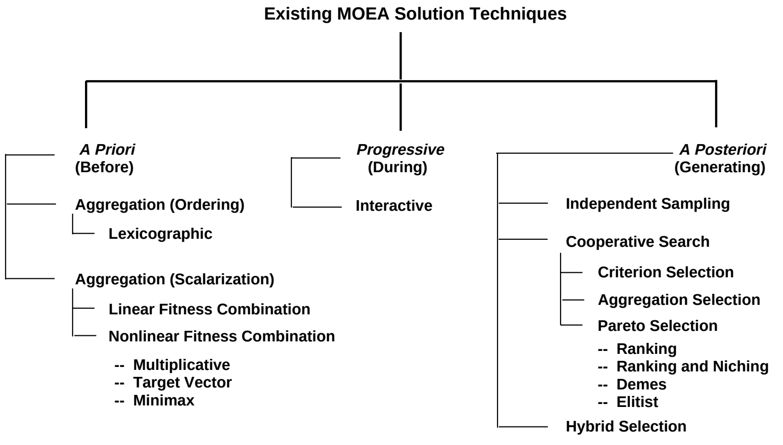


Figure 2: MOEA solution technique classification.

4 MOEA Research: A Quantitative Analysis

Freeman Dyson once said, “A good engineer is a person who makes a design that works with as few original ideas as possible.” Our extensive survey (Van Veldhuizen, 1999) and associated research is meant to help an algorithmic engineer hold those original ideas to a minimum, giving them a means to quickly identify and incorporate appropriate concepts in a new MOEA instantiation and/or solve some MOP of interest.

This section is concerned primarily with analyzing raw data from the survey, while Section 5 presents analysis of a more observational nature. This treatment shows the interested practitioner where and how the field has focused its energies. Here we present results concerning the numbers and types of MOEA publications and the solution techniques they employ. Also included are discussions of MOEA fitness functions, genetic representations, and application problem domains.

4.1 MOEA Citations

The initial transformations of EAs for the MOP domain did not spark any real interest until several years later – not until the mid 1990’s is there a noticeable increase in published MOEA research (Van Veldhuizen, 1999). However, this increase is substantial as almost three times as many MOEA citations are dated in the last six years (1994-1999) as in the first ten (1984-1993). The sheer number of recent publications indicates an active and growing research community interest in MOEAs.

Comparing citations by technique highlights the popularity of *a posteriori* techniques. Over twice as many citations occur in that category as in the *a priori* and *progressive* categories combined. When considering only these *a posteriori* techniques, almost twice as many Pareto-based selection approaches exist as the others combined. In fact, our research has shown these Pareto-based selection approaches to currently be the most popular MOEA solution technique. This is a change from three years ago when Fonseca and Fleming (1997a) stated linear fitness combination was the most popular technique.

A number of papers are primarily concerned with comparing MOEA implementations. This is a healthy sign of skepticism in that researchers are seeking to test proposed algorithms on a variety of problems. We also note that MOEA theory noticeably lags behind applications, at least in terms of published papers. This is even clearer when realizing few of these categorized papers *concentrate* on MOEA theoretical concerns. Many discuss some MOEA theory but do so only as regarding various parameters of their respective approaches. This quantitative lack of theory is not necessarily bad but indicates further theoretical development is necessary to (possibly) increase the effectiveness and efficiency of existing MOEAs.

Finally, we show that genetic algorithm-based MOEAs are the most popular implementation type *by far*, with nine times the number of citations as all other types combined (Van Veldhuizen, 1999). Also, we have to date identified only one evolutionary program-based MOEA in the literature.

4.2 MOEA Techniques

By definition, MOEAs operate on MOPs. A more theoretical discussion of the MOP domain is given elsewhere (Van Veldhuizen, 1999; Van Veldhuizen and Lamont, 1999;

Deb, 1999b); here we discuss it in more general terms. When implementing an MOEA it is (implicitly) assumed that the problem domain (fitness landscape) has been examined, and a decision made that some MOEA approach is the most appropriate solution tool for the given MOP. In general, single-objective EAs are useful search algorithms when the problem domain is multidimensional (many decision variables), and/or the search space is very large. Most cited MOEA problem domains also appear to exhibit these characteristics. An MOEA should be applied only when problem solving benefits from it. A particular problem instance may also determine MOEA performance. This is no different than is the case with single-objective EAs but bears mentioning.

Many MOEA implementations are currently available. Selecting an appropriate solution technique (e.g., *a priori*, *progressive*, *a posteriori*) and approach is dependent upon meticulous examination of the problem domain; ensuring derived solutions are the best available requires careful integration of both problem and algorithm domains. Identifying MOEA techniques and approaches that have and have not historically "worked" should improve future MOEA performance. The interested reader is referred elsewhere (Van Veldhuizen, 1999; Coello, 1999a; Fonseca and Fleming, 1997a; Horn, 1997) for an in-depth description and discussion of attendant strengths/weaknesses of the various approaches identified in Figure 2.

4.3 MOEA Comparisons and Theory

To date, most MOEA researchers' *modus operandi* is comparing some MOEA (usually the researcher's own new and improved variant) against an older MOEA (often the vector evaluated genetic algorithm (VEGA) (Schaffer, 1985), even with its identified shortfalls) and analyzing results for some MOP (often Schaffer's F2 (1985) or some other numeric example). Comparative results are then "clearly" shown in graphical form indicating which algorithm performed better, often implying the new MOEA's returned PF_{known} is a better representation of PF_{true} . To their credit, many of these publications also compare MOEA performance on real-world applications. An argument can be made along the lines of "if it works, use it" but, in general, using a test problem and/or an application's results to judge comprehensive MOEA usefulness is not conclusive.

These empirical, relative experiments are incomplete as regarding general MOEA comparisons. The literature's history of visually comparing MOEA performance on non-standard and unjustified numeric MOPs does little to determine a given MOEA's actual efficiency and effectiveness. Only recently have any researchers proposed formalized, experimental methodologies for *general* MOEA comparative analysis (Van Veldhuizen, 1999; Zitzler and Thiele, 1999; Deb, 1999a; Shaw et al., 1999). An important component of these methodologies is a validated suite of numeric functions exhibiting relevant MOP problem domain characteristics to provide a common comparative basis.

Less than one-tenth of published MOEA papers focus on underlying theoretical analyses. These papers concentrate mainly on MOEA parameters, behavior, and concepts. They attempt to further define the nature and limitations of Pareto optimality, its subsequent effects upon MOEA search, and discuss the characteristics and construction of appropriate MOEA benchmark test function suites. Although other MOEA researchers often cite these works, our detailed categorizations show their efforts to often be modifications of previously implemented approaches, or perhaps the same approach directed towards a different application. These papers add little to the body of MOEA theory.

4.4 Selected MOEA Survey Results

Our cataloged research provides various fitness function types used by MOEAs. We have identified the following generic fitness function types implemented within MOEAs: economic, electromagnetic, entropic, environmental, financial, geometrical, physical (energy and force), resource, and temporal (Van Veldhuizen, 1999). These fitness functions are not limited to MOEA applications nor are they the only types possible. However, MOEAs offer the exciting possibility of simultaneously employing different fitness function types to capture desirable characteristics of the problem domain regardless of implemented MOEA approach.

The employed fitness functions appear limited only by the practitioner's imagination and particular application. Although many implementations use only two objective functions, several approaches use three, four, seven, or more. However, a fitness function's effectiveness depends on its application in appropriate situations, i.e., it measures some *relevant* feature of the studied problem. The claim by many authors that their particular MOEA implementations are successful implies the associated fitness functions are appropriate for the given problem domains.

Our cataloged research clearly shows the incommensurability and independence of many fitness function combinations. For example, optimizing a radar antenna design may involve electromagnetic (energy transmission), geometric (antenna shape), and financial (dollar cost) objectives. The proposed antenna's shape may have no meaningful impact on its cost; the objectives may be measured in megawatts, meters, and dollars. These are the factors responsible for the partial ordering of the search space and the subsequent need to develop appropriate MOEA fitness assignment procedures.

Genetic representation is another MOEA component limited only by the implementor's imagination. The cited efforts indicate the most common representation is a binary string corresponding to some simple mapping from the problem domain. Real-valued chromosomes are also often used in this fashion. Array constructs are used, and, just as in single-objective EAs, combinatorial optimization problems often use a permutation ordering of jobs, tasks, etc.

An overwhelming majority of cited efforts are applied to nonpedagogical problems, indicating MOEA practitioners are developing and implementing MOEAs as real-world tools. In fact, almost 90% of cataloged Pareto-based MOEAs are applied to real-world scientific and engineering problems (Van Veldhuizen, 1999). These implementations span several disparate scientific and engineering research areas and give credibility to the MOEA's claim as an effective and efficient general purpose search tool.

5 MOEA Research: A Qualitative Analysis

What differentiates an MOEA from a single-objective EA? What components should be included in an MOEA? When should an MOEA be used? This section addresses these questions and presents matters of a more philosophical nature raised by the preceding discussion, considering several MOEA design issues. Although not quantitatively derived, our analytical observations are based on our catalog and substantiated with other relevant citations from the literature. This section discusses several MOEA theoretical issues, as well as MOEA secondary populations, complexity, and parallelization.

5.1 MOEA Theoretical Issues

We agree with other MOEA researchers (Horn, 1997; Fonseca and Fleming, 1995a), and indeed have shown (Van Veldhuizen, 1999) that MOEA theory is lagging behind MOEA implementations and applications. For example, until recently no proof was offered showing an MOEA is capable of converging to P_{true} or PF_{true} (Van Veldhuizen, 1999; Rudolph, 1998a, 1998b). Although the number of known MOEA implementations is significant, this fact alone does not indicate a corresponding *depth* of associated theory. This research makes absolutely clear that past effort has been mainly spent designing new or variant MOEA approaches and not in comprehensively reviewing the benefits and/or trade-offs of the various implementations.

Why is there such a lack of underlying MOEA theory? Although some mathematical foundations exist, the current situation seems akin to Goldberg's comparison of engineer and algorithmist (Goldberg, 1998). He likens algorithms to "conceptual machines" and implies computer scientists are hesitant to move forward without exact models precisely describing their situation. On the other hand, he claims a design engineer often accepts less accurate models in order to build the design. MOEA researchers certainly seem to have taken this approach!

Realizing that simple assumptions are sometimes made in order to develop limited theoretical results, the foundations of single-objective EA theory seem well-established. The *Handbook of Evolutionary Computation* (Bäck et al., 1997) devotes entire chapters to theoretical results established during the past 20–30 years. Although much of this theory is (may be?) valid when regarding MOEAs, some is not. Thus, current knowledge concerning selected MOEA theoretical issues is now discussed.

5.1.1 Fitness Functions

The general manner of fitness function implementation is two-fold. This is reflected by the work of Wienke et al. (1992) and Fonseca and Fleming (1997b), who each solved MOPs with seven fitness functions. Wienke et al., essentially, used seven copies of an identical objective function, that of meeting atomic emission intensity goals for seven different elements. Although the elements and associated goals are each different, the fitness functions are conceptually identical. This does not make the MOP "easier" but, perhaps, makes the objective space somewhat easier to understand.

On the other hand, Fonseca and Fleming's MOP's seven objectives appear both incommensurable and independent. Both solution and objective space are hard to visualize, as are their interrelationships. For example, when considering the mathematical polynomial model constructed by their MOEA, it is unclear how the number of terms affects the long-term prediction error and how that error may affect variance and model lag.

The overwhelming majority of implemented MOEAs use only two fitness functions, most probably for ease and understanding. Several use three to nine, and the currently known maximum is 23 fitness functions within a single MOEA. This approach used an MOEA to solve a heavily constrained single-objective optimization problem (Coello, 2000). Here, one objective was the fitness function and the other 22 were constraints cast as objectives. The highest number of conceptually different implemented fitness functions is found in a linkage design problem (Sandgren, 1994), where nine objectives are used.

How many fitness functions are enough? How many objectives are, generally, required to adequately capture an MOP's essential characteristics? *Can* all relevant characteristics be captured? The cataloged efforts imply most real-world MOPs are effectively solved using only two or three objectives. A practical limit to the maximum number of possible objective functions exists, as the time to compute several complex MOEA fitness functions quickly becomes unmanageable.

A theoretical limit also exists as far as Pareto optimality is concerned. As additional objectives are added to an MOP, more and more MOEA solutions meet the definition of Pareto optimality. Thus, as Fonseca and Fleming (1995a) indicate, for most Pareto MOEAs the size of $P_{current}\,(t)$, $PF_{current}\,(t)$, $P_{known}\,(t)$, and $PF_{known}\,(t)$ grows, and Pareto selective pressure decreases. However, some confusion results from both their and Horn's (1997) statements implying that the size of PF_{true} grows with additional objectives. We show that the cardinality of PF_{true} does *not* grow with the number of objectives, only (possibly) it's topological dimension (Van Veldhuizen, 1999). However, since MOEAs deal with *discretized* numerical representations, the number of possible solutions (and therefore the number of computable vectors composing PF_{known}) may increase as more objectives are added. Finally, some limit to human understanding and comprehension exists. The human mind appears to have a limited capacity for simultaneously distinguishing between multiple pieces of information or concepts.

Past MOEA implementation results imply two or three objectives are probably "satisfactory" for most problem domains. Thus, MOEA application to a given MOP should probably begin with two or three primary objectives in an effort to gain problem domain understanding. One may be able to ascertain how the different objectives interact and gain an idea of the fitness landscape's topology. Other fitness functions may then be added in order to capture other relevant problem characteristics.

5.1.2 Pareto Ranking

Two Pareto ranking methods are primarily used in MOEAs, although variations do exist. In general, all assign preferred (Pareto optimal) solutions the same rank and other solutions some less desirable rank. With the scheme proposed by Goldberg (1989), where a solution x at generation t has a corresponding objective vector x_u, and N is the population size, the solution's rank is defined by the algorithm in Figure 3.

The second technique, proposed by Fonseca and Fleming (1998), operates somewhat differently. As before, a solution x at generation t has a corresponding objective vector x_u. We also let $r_u^{(t)}$ signify the number of vectors associated with the current population dominating x_u; x's rank is then defined by:

$$\mathrm{rank}(x,t) = r_u^{(t)} \tag{3}$$

This ensures all solutions with nondominated vectors receive rank zero.

Some approaches simply split the population in two, e.g., assigning solutions with nondominated vectors rank 0 and all others rank 1 (Van Veldhuizen, 1999). Using the same notation this simple ranking scheme is defined by:

$$\mathrm{rank}(x,t) = \begin{cases} 0 & \text{if } r_u^{(t)} = 0, \\ 1 & \text{otherwise.} \end{cases} \tag{4}$$

When considering Goldberg's and Fonseca and Fleming's ranking schemes, it initially

```
curr_rank = 1
$m = N$
while $N \neq 0$ do
    For $i = 1 : m$ do
        If $x_u$ is nondominated
            $\mathrm{rank}(x, t)$ = curr_rank
    od
    For $i = 1 : m$ do
        If $\mathrm{rank}(x, t)$ = curr_rank
            Remove $x$ from population
            $N = N - 1$
    od
    curr_rank = curr_rank + 1
    $m = N$
od
```

Figure 3: Rank assignment algorithm.

appears that neither is "better" than the other, although it is mentioned in the literature that Fonseca and Fleming's method, which effectively assigns a cost value to each solution, might be easier to mathematically analyze (Fonseca and Fleming, 1997a). Horn (1997) also notes this ranking can determine more ranks (is finer-grained) than Goldberg's (assuming a fixed population size).

One last ranking method using Pareto optimality as its basis is proposed by Zitzler and Thiele (1999). Their MOEA implementation uses a secondary population whose solutions are directly incorporated into the generational population's fitness assignment procedure. Effectively, a solution in P_{known} (t) is assigned a rank equal to the proportion of the generational population's evaluated vectors that its associated vector dominates. Because of P_{known} (t)'s inclusion in the ranking process, this method's complexity may be significantly higher than the others discussed. Additionally, this method is biased in that it may result in some Pareto optimal solutions receiving preference over others in the selection process (Deb, 1999b). Given the t^{th} generational population $P(t)$, secondary population P_{known} (t), $x \in P_{known}$ (t), and $y \in P(t)$, x's rank is then defined by:

$$\mathrm{rank}(x, t) = \frac{\mid n \mid}{\mid P(t) \mid +1}, \text{ where } n = \{y \mid F(x) \preceq F(y)\} \tag{5}$$

There is currently no clear evidence as to the benefit(s) of any of these ranking schemes over another. Further clouding the issue is the fact that rank itself is often *not* directly used as a solution's fitness. For example, Fonseca and Fleming's (1998) Multi-objective Genetic Algorithm (MOGA) and Srinivas and Deb's (1994) Nondominated Sorting Genetic Algorithm (NSGA) (implementing Goldberg's scheme) both transform assigned rank before selection occurs. The MOGA sorts solutions by rank and assigns fitness via linear or exponential interpolation, while the NSGA uses "dummy" fitness assignment, ensuring only that each "wave" of identically ranked solutions has a maximum fitness smaller than the preceding wave's minimum value. Zitzler and Thiele's (1999) Strength Pareto Evolutionary Algorithm (SPEA) assigns fitness to solutions in $P(t)$ by summing the ranks of all solutions

Table 1: MOEA Pareto ranking complexities.

Technique	Best Case	Worst Case
Simple	$N^2 - N$	$N^2 - N$
Fonseca & Fleming	$N^2 - N$	$N^2 - N$
Goldberg	$N^2 - N$	$\frac{1}{3}(N^3 - N)$
Zitzler & Thiele	$(N + N_1)^2 - N - N_1$	$(N + N_1)^2 - N - N_1$

$x \in P_{known}\,(t)$ such that $F(x) \preceq F(y)$.

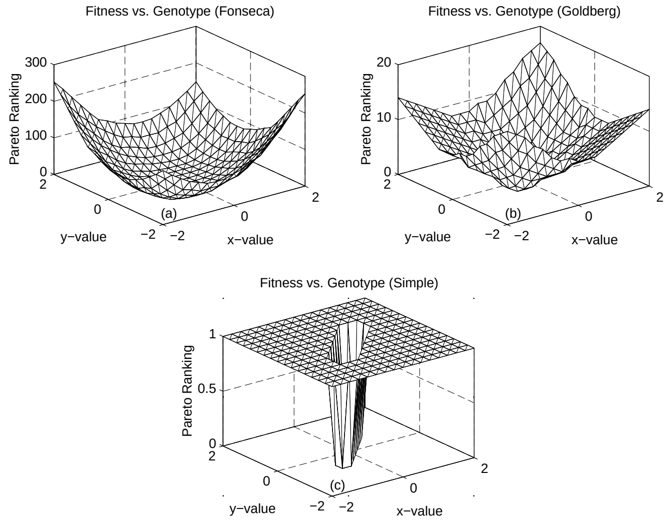
Only one experiment directly comparing any of these schemes is reported in the literature. Thomas (1998) compared Fonseca and Fleming's and Goldberg's Pareto ranking schemes in an MOEA applied to submarine stern design. He concludes that both outperformed tournament selection, and that Fonseca and Fleming's ranking appears to provide a fuller, smoother PF_{known}. However, he and we caution that this is a singular data point. On a similar note, only two citations in the known MOEA literature give data on the number of population "waves" using Goldberg's ranking, presenting graphs showing the number of waves found in each generation (Van Veldhuizen, 1999; Vedarajan et al., 1997). With a population size of 300 individuals, Vedarajan et al. show the first generation has over 40 waves. This quickly drops and from generations 10 to 100, oscillates between 20 and 25.

Analyzing these schemes' mathematical complexity is revealing. Table 1 (showing each scheme's best and worst case) and the following analysis only consider population size in computing complexity, where N is the size of $P(t)$ and N_1 the size of $P_{known}\,(t)$. Assuming that as comparisons are performed, appropriate counter or rank assignments are made or updated, the simple, Fonseca and Fleming, and Zitzler and Thiele ranking schemes require only one "pass" through the population(s) regardless of the number of nondominated solutions. Their worst and best case complexities are identical. Goldberg's scheme, however, requires at most $N - 1$ "passes" through the population if there is only one Pareto optimal solution per reduced population (or front). In addition, Zitzler and Thiele's scheme's complexity increases if $P_{known}\,(t)$'s size is significantly larger than $P(t)$'s. Thus, Goldberg's and Zitzler and Thiele's ranking schemes (potentially) involve significantly more overhead than do the others, but note that the latter scheme often limits the size of $N + N_1$ (Zitzler and Thiele, 1999).

It is also instructional to look at the possible value ranges for each ranking scheme. The simple scheme (Equation 4) offers only two values, $\Phi \in \{0, 1\}$. Both Fonseca and Fleming's (Equation 3) and Goldberg's scheme (Figure 3) offer N possible values, $\Phi \in \{0, 1, \ldots, N - 1\}$. In practice, however, Goldberg's scheme uses some subset of these values (resulting in a "coarser" ranking). Zitzler and Thiele's scheme (Equation 5) offers possibly noninteger values $\Phi \in [1, N)$. Figure 4 shows the resultant solution rankings of three Pareto ranking schemes for a particular MOP (Van Veldhuizen, 1999).

5.1.3 Pareto Niching and Fitness Sharing

Several MOEA Pareto niching and fitness sharing variants have been proposed with the same goal as in traditional single-objective optimization – that of finding and maintaining *multiple* optima. However, MOEAs use fitness sharing in an attempt to find a uniform (equidistant) distribution of vectors *representing* PF_{true}, i.e., one in which PF_{known}'s shape is a "good"


Fitness vs. Genotype (Fonseca)

Fitness vs. Genotype (Goldberg)

Fitness vs. Genotype (Simple)

Figure 4: Pareto ranking schemes.

approximation of PF_{true} . We compare selected implementations of this concept.

Fonseca and Fleming's (1998) MOGA uses restricted sharing, in the sense that fitness sharing occurs only between solutions evaluating to vectors with identical Pareto rank. They measure niching distance in phenotypic space; the distance (over some norm) between two solutions' evaluated fitness vectors is computed and compared to σ_{share} (the key sharing parameter). If the distance is less than σ_{share}, the solution's associated niche count is then adjusted. The NSGA implements a slightly different scheme. Distance (over some norm) is measured, here, in genotypic space; the distance between two solutions is compared to σ_{share}. The NSGA also shares fitness only between solutions evaluating to vectors with identical Pareto rank.

Horn et al. (1994) define niching differently in their Niched Pareto Genetic Algorithm (NPGA), which performs selection via binary Pareto domination tournaments. Solutions are selected if they dominate both the other and some small group (t_{dom}) of randomly selected solutions, but fitness sharing occurs only in the cases where both solutions are (non)dominated. Each of the two solution's niche counts is derived not by summing computed sharing values, but by simply counting the number of objective vectors within σ_{share} of their evaluated vectors in phenotype space. The solution with a smaller niche count (fewer phenotypical neighbors) is then selected. Horn et al. term this *equivalence class sharing*.

Another fitness sharing variant is NSGA-based but instead uses phenotypic sharing (Michielssen and Weile, 1995); yet another variant combines both genotypic and phenotypic distances in determining niche counts (Rowe et al., 1996). Fitness sharing may also be indiscriminately applied to all solutions regardless of associated Pareto rank.

All of these methods require setting explicit values for the key sharing parameter σ_{share}, which can affect both MOEA efficiency and effectiveness. Fitness sharing's performance is also sensitive to the population size N. Assigning appropriate values to σ_{share} is generally difficult as it usually requires some *a priori* knowledge about the shape and separation of a given problem's niches. However, as phenotypic-based niching attempts to obtain equidistantly spaced vectors along PF_{known} , both Fonseca and Fleming (1998) and Horn and Nafpliotis (1993) are able to give guidelines for determining appropriate MOEA σ_{share} values. These values are based on known phenotypical extremes (minimum and maximum) in each objective dimension. Appropriate values for the NPGA's tournament size parameter (t_{dom}) are also suggested.

To determine σ_{share} 's value using Fonseca and Fleming's method, one uses the number of individuals in the population (which implicitly determines the number of niches), scales the known attribute values, and determines the extreme attribute values in each objective dimension. These parameters are then used to derive σ_{share} . Horn and Nafpliotis' guidelines use the above parameters to define bounds for σ_{share} 's value.

How does one find each objective dimension's extreme values? Using the minimum and maximum values of either the generational or a secondary population is the easy answer. Fonseca and Fleming (1998) indicate recomputing σ_{share} at each generation (using current generational extremums) yields good results. However, we note that the MOEA's stochastic nature may not preserve these values between generations, i.e., the associated solutions may not survive. Thus, it is better to select objective extremes from the secondary population if one is incorporated in the MOEA. By definition, this population contains each objective dimension's extrema *so far*, ensuring the "ends" of PF_{known} are not lost.

As with the proposed Pareto ranking schemes, there is not yet any clear evidence as to the benefit(s) of one Pareto niching and sharing variant over another. Nor are any formal experiments reported in the literature comparing key components of these different approaches (e.g., σ_{share} value assignment).

We note the following as regarding the appropriate sharing domain. Horn et al. (1994) indicate sharing should be performed in a space we "care more about." Phenotypic-based sharing does make sense if one is attempting to obtain a "uniform" representation of PF_{true} . On the other hand, Benson and Sayin (1997) indicate many operations researchers "care more about" obtaining a uniform representation of P_{true} , in which case genotypic-based sharing seems appropriate. The end representation goal should drive the sharing domain.

5.1.4 Mating Restriction

The idea of restricted mating is not new. Goldberg (1989) first mentions its use in single-objective optimization problems to prevent or minimize "low-performance offspring (lethals)." In other words, restricted mating biases how solutions are paired for recombination in the hope of increasing algorithm effectiveness and efficiency. Goldberg presented an example using genotypic-based similarity as the mating criteria. Deb and Goldberg (1989) then implemented phenotypic-based restricted mating in their GA niching and sharing investigation. Note that these implementations only allow mating between "similar" solutions (over some metric). *Island model* GAs also implement restricted mating but in a geographic sense, where solutions mate only with neighbors residing within some restricted topology (Cantú-Paz, 1997). It is also noted (Coello, 1999a) that some researchers believe restricted mating should allow for recombination of *dissimilar* (over some metric) individuals. However defined, restricted mating is also incorporated within many MOEAs in an attempt to reduce unfit (non-Pareto optimal) offspring (Van Veldhuizen, 1999).

When considering general MOEAs, phenotypic-based restricted mating between similar solutions is of more interest to us. This is due to the fact most researchers focus on finding a uniform representation of PF_{true} and thus perform fitness sharing in the phenotype domain. Also, some state in published reports (Fonseca and Fleming, 1993, 1995b; Zitzler and Thiele, 1998): "Following the common practice of setting $\sigma_{mate} = \sigma_{share}$ ..."

This may be a common practice, but no background is cited in the literature. As σ_{share} attempts to define a region within which all vectors are "related," setting σ_{mate} equal to σ_{share} is intuitive (and the same rationale holds in genotypic-based sharing and mating restriction). We currently have only empirical explanations offered for the implementation (or lack) of restricted mating in various MOEA approaches. In fact, it has been noted that the use of mating restriction in MOEAs does not appear to be widespread (Fonseca and Fleming, 1995a; Zitzler and Thiele, 1999). Obviously, some researchers believe restricted mating is necessary or they would not have implemented it, but others indicate it is of no value!

Zitzler and Thiele (1998) state that for several different values of σ_{mate} , no improvements were noted in their test problem results (an MOP with two, three, and four objectives) when compared to those with no mating restriction. Shaw and Fleming (1996) report the same qualitative results for their application (an MOP with three objectives) whether or not mating restriction was incorporated. Horn et al. (1994) offer empirical evidence directly contradicting the basis for mating restriction. They note that recombining solutions whose associated vectors are on different portions of $PF_{known}\,(t)$ *can* produce offspring whose vectors are on $PF_{known}\,(t+1)$ but between their parent's vectors. They also claim that for a specific MOP, a constant (re)generation of vectors through recombination of "dissimilar" parents maintains PF_{known} . Finally, they believe most recombinations of solutions in P_{known} also yield solutions in P_{known} .

Just as in single-objective optimization, no clear quantitative evidence exists regarding the benefits of restricted mating. The empirical evidence presented in the literature can be interpreted as an argument either for or against this type of recombination and leaves the MOEA field in an unsatisfactory predicament. This issue clearly benefits from experiments directly comparing its algorithmic inclusion/exclusion. One must also consider the NFL theorems (Wolpert and Macready, 1997), realizing that mating restriction may not always be effective (or needed) for every problem (class).

5.2 MOEA Secondary Populations

We agree with Horn (1997) that any practical MOEA implementation *must* include a secondary population composed of all Pareto optimal solutions found so far during search ($P_{known}\,(t)$). This is due to the MOEA's stochastic nature, which does not guarantee that desirable solutions, once found, remain in the generational population until MOEA termination. This is analogous to elitism, but remember that $P_{known}\,(t)$ is a *separate* population - the question is then how best to utilize it. Is this additional population simply a repository, continually added to and periodically culled of solutions whose associated vectors are dominated? Or is it an integrated component of the MOEA? Several researchers indicate their use of secondary populations, but only a few explain its use in their implementation. As there is no consensus for its "best" use we present some of its incarnations.

A straightforward implementation stores $P_{current}\ (t)$ at the end of each MOEA generation ($P_{current}\ (t) \cup P_{known}\ (t-1)$). This set must be periodically culled since a solution's designation as Pareto optimal is *always* dependent upon the set Ω within which it is evaluated. How often the population is updated is generally a matter of choice, but as determination of Pareto optimality is an $\mathcal{O}(n^2)$ algorithm, it should probably not be performed arbitrarily. As this population's size grows, comparison time may become significant. This implementation does not feed solutions from $P_{known}\ (t)$ back into the MOEA's generational population.

Conversely, other published algorithms actively involve $P_{known}\ (t)$ in MOEA operation. For example, SPEA stores $P_{current}\ (t)$ in a secondary population, immediately culling solutions whose evaluated vectors are dominated. If the number of solutions in $P_{known}\ (t)$ exceeds a given maximum, the population is reduced by clustering in an attempt to generate a representative solution subset while maintaining the original set's ($P_{known}\ (t-1)$'s) characteristics. Solutions from both the MOEA's generational and secondary populations then participate in binary tournaments selecting the next generation. SPEA uses $P_{known}\ (t)$ in computing the fitness of solutions in the general population (effectively resulting in a larger generational population).

Solutions from $P_{known}\ (t)$ are sometimes inserted into the mating population in an attempt to maintain diversity (Todd and Sen, 1997; Ishibuchi and Murata, 1998). These implementations never reduce $P_{known}\ (t)$'s size except when removing solutions whose evaluated vectors become dominated. Although Parks and Miller (1998) implement an *archive* of Pareto optimal solutions, solutions in $P_{current}\ (t)$ are not always archived (placed in $P_{known}\ (t)$); archiving occurs only if a solution is sufficiently "dissimilar" from those already resident (clustering). If a new solution is added, any archive members no longer Pareto optimal (with respect to $P_{known}\ (t)$) are then removed. Like SPEA, the next generation's members are selected from both $P_{known}\ (t)$ and the current generational population. Finally, some researchers even use secondary populations *not* composed of Pareto optimal solutions (Van Veldhuizen, 1999).

A secondary population (of some sort) is an MOEA necessity. Because the MOEA is attempting to build up a (discrete) picture of a (possibly continuous) Pareto front, this is probably a case where, at least initially, too many solutions are better than too few. It intuitively seems that a secondary population might also be useful in adding diversity to the current generational population and in exploring "holes" in PF_{known} , although how to effectively and efficiently use P_{known} in this way is currently unknown. Again, we suggest experiments directly comparing various secondary population implementations and their effect on MOEA performance.

5.3 MOEA Complexity and "Cost"

It is well known that fitness function evaluation (for many real-world problems) may dominate EA execution time. Thus, when discussing various MOEAs' algorithmic complexity we are concerned mainly about the number of fitness evaluations, although solution comparisons and additional calculations are also considered, as this overhead is not found in simple GA implementations. We show elsewhere that MOEA complexity is generally greater than that of simple GAs and that MOEA storage requirements are problem dependent (Van Veldhuizen, 1999). Like other EAs the storage requirements are mandated by the specific data structures used. Required storage increases linearly with the number of fitness functions used and when a secondary population is brought into play.

When practically considered, fitness evaluation cost limits EA-based search. Since all algorithms must eventually terminate, the number of fitness evaluations is then often selected as the finite resource expended in search, i.e., the choice is made *a priori* for an EA to execute n fitness evaluations. The "best" solution(s) found is then returned. Assuming solutions are not evaluated more than once (no clones) a total of n possible solutions in the search space are explored.

Now consider a k-objective MOP. Here, k fitness evaluations are performed for each possible solution (one for each objective). Assuming resources are still limited to the same number of fitness evaluations and that each objective's evaluation is equally "expensive", only $\lfloor \frac{n}{k} \rfloor$ solutions are now evaluated. All else held equal, a k-objective MOP may then result in a k-fold decrease in search space exploration. Note also that in the context of MOEAs, this implies using the term "fitness function evaluations" to measure computational effort, which may be somewhat misleading. The term "solution evaluations" is clearer in this context.

This result implies an MOEA may require longer (than a single-objective EA) "wall clock" execution times for good performance. Further search is never guaranteed to return the optimal answer, but one wishes as much exploration as possible in the time allowed. This increases the sense of confidence that one has found the true, and not a local, optimum.

5.4 MOEA Parallelization

We have noted several parallel MOEA implementations executing either several MOEAs on different processors (several independent, synchronous runs) or distributing an MOEA's population among processors in a demic manner (a "master-slave" or island model) (Van Veldhuizen, 1999). However, none discuss what other parallel MOEA possibilities exist.

Affecting the ability to effectively and efficiently parallelize an MOEA is the fact it is inherently sequential. However, its fitness function evaluation task can be and has been parallelized. MOEA fitness function evaluation allows for parallelism by assigning each of k fitness function's evaluations to different processors, assigning subpopulations for all k function evaluations to different processors or, in the case of expensive fitness functions, assigning each individual's evaluation across several processors, one of the k fitness functions at a time. These options are illustrated in Figure 5.

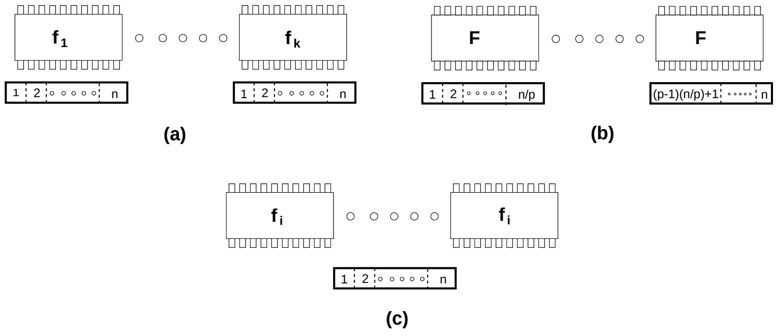


Figure 5: MOEA parallel fitness evaluation possibilities.

In broad terms, any parallel MOEA implementation should result in some speedup gains. Additionally, it offers the possibility of evaluating more candidate solutions, perhaps providing a "better" view of the fitness landscape. However, some MOEA technique modifications may be required when implemented in a parallel manner.

6 MOEA Design Recommendations

We have identified numerous MOEA approaches (Van Veldhuizen, 1999). When considering these approaches, those researchers wishing to implement an MOEA may well be asking, "Where do I begin?" We cannot specify an "all purpose" MOEA technique, nor do the NFL theorems (Wolpert and Macready, 1997) allow for one. However, we can suggest MOEAs that appear appropriate as a starting point. Interested researchers may then select one to begin their own exploration of the MOP domain.

We focus on those MOEAs employing Pareto-based selection and specifically consider a MOGA, the multiobjective messy GA (MOMGA) (Van Veldhuizen, 1999), the NPGA, the NSGA, and the SPEA.

These algorithms stand out because they incorporate known MOEA theory. The Pareto-based selection each employs explicitly seeks P_{true}. All incorporate niching and fitness sharing in an attempt to uniformly sample PF_{true}. Mating restriction may (or may not) be included in any of the five, as may a secondary population (the SPEA requires a secondary population). Finally, their general algorithmic complexity is no higher than other known MOEA techniques.

Although each MOEA's authors (and rightly so) point out deficiencies in their own and other MOEAs, any algorithmic approach is bound to have some shortfalls when applied to certain problem classes, as proved by the NFL theorems. These selected algorithms' common theme is their respect of known relevant theoretical issues, and their empirical success in both (non-)numeric MOPs and real-world applications. The MOGA, NPGA, and NSGA easily win the title "Most Often Imitated MOEAs" – this implies other researchers also see value in them. The MOGA, MOMGA, NPGA, and NSGA are used in extensive experiments supporting this research; the experiments and their results are detailed elsewhere (Van Veldhuizen, 1999). Additionally, the NPGA, NSGA, and SPEA are used in other detailed comparative experiments (Zitzler and Thiele, 1999). The five recommended MOEAs are only briefly described here.

1. **MOGA**. Implemented by Fonseca and Fleming (1998). Initially used to explore incorporation of DM goals and priorities in the multiobjective search process. Employs the Pareto ranking scheme in Equation 3 (Section 5.1.2) and incorporates fitness sharing.

2. **MOMGA**. Implemented by Van Veldhuizen (1999). Initially used to explore the relationship between MOP solution building blocks and their use in MOEA search. Incorporates fitness sharing and Horn et al.'s (1994) tournament selection.

3. **NPGA**. Implemented by Horn et al. (1994). Initially used to explore benefits of providing P_{known} as input to a decision analysis technique. Uses tournament selection based on Pareto optimality. Incorporates fitness sharing.

4. **NSGA**. Implemented by Srinivas and Deb (1994). Initially used to explore bias prevention towards certain regions of the Pareto front. Employs the Pareto ranking shown in Figure 3 (Section 5.1.2) and incorporates fitness sharing.

5. **SPEA**. Implemented by Zitzler and Thiele (1999). Initially used to explore active use of PF_{known} (t) in assigning generational fitnesses. Employs the Pareto ranking scheme shown in Equation 5 (Section 5.1.2). Incorporates fitness sharing.

Although perhaps not straightforward, many existing EA implementations are extendable into the MOEA domain. For example, GENOCOP III (Michalewicz and Nazhiyath, 1995) was readily modified to incorporate both a specialized problem domain code and linear fitness combination technique (Van Veldhuizen et al., 1998). The GEATbx for use with *MATLAB*[2] (Pohlheim, 1998) allowed us to quickly create both MOGA and NSGA variants; these codes are now being incorporated into the toolbox's baseline version. Other researchers have also provided their MOEA code upon request. Thus, initial algorithmic development should not be a barrier to solving MOPs with MOEAs.

7 Summary

As we have indicated, MOEAs continue to have substantial success across a variety of MOP applications, from pedagogical multifunction optimization to real-world engineering design. The variety of MOEAs as well as their numerous applications suggested a classification framework be developed. Thus, we have presented such a framework cataloging current MOEA research and applications, in which it is easy to include both new citations and new MOEA approaches. In concert with our consistent Pareto-based notation, this framework permitted an extensive discussion of MOEA research trends, and the multitude of contemporary MOEAs and associated key elements. Moreover, our analysis resulted in validated MOEA design recommendations for new applications and is hoped to stimulate new theoretical approaches. An integral aspect of this paper is the "points to ponder" when redesigning current MOEAs and EAs for solving MOPs; a set of references is also given for initiating this effort.

Highlighted in our discussion are many opportunities for further MOEA research. The formalized methodologies for MOEA comparative analysis require additional validated MOPs and further development to make them more effective. As indicated, no formal studies exist directly comparing the known Pareto ranking and fitness assignment schemes, determining appropriate niching/fitness sharing parameter value assignments, or recommending "best" secondary population implementations. Finally, MOEA researchers appear to have only scratched the surface concerning MOEA parallelization possibilities. Appropriate research addressing these issues may well lead to more effective and efficient MOEAs.

Acknowledgments

We thank Mr. Mike Putney for his support during code development and implementation, and the Major Shared Resource Centers belonging to the Aeronautical Systems and U.S. Army Engineer and Research Development Centers for providing the computing platforms.

[2] *MATLAB* is a Trademark of The MathWorks, Inc.

References

Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York, New York.

Bäck, T., Fogel, D. and Michalewicz, Z. (1997). *Handbook of Evolutionary Computation*, Volume 1, Oxford University Press, Oxford, England.

Benson, H. P. and Sayin, S. (1997). Towards Finding Global Representations of the Efficient Set in Multiple Objective Mathematical Programming. *Naval Research Logistics*, 44:47–67.

Cantú-Paz, E. (1997). A Survey of Parallel Genetic Algorithms. Technical Report 97003, Illinois Genetic Algorithms Laboratory, Department of General Engineering, University of Illinois, Urbana, Illinois.

Coello, C. A. C. (1999a). A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems*, 1(3):269–308.

Coello, C. A. C. (1999b). List of References on Evolutionary Multiobjective Optimization. Available: `http://www.lania.mx/~ccoello/EMOO/EMOObib.html`.

Coello, C. A. C. (2000). Treating Constraints as Objectives for Single-Objective Evolutionary Optimization. *Engineering Optimization*, 32(3):275–308.

Deb, K. (1999a). Construction of Test Problems for Multi-Objective Optimization. In Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M. and Smith, R. E., editors, *GECCO–99: Proceedings of the Genetic and Evolutionary Computation Conference*, Volume 1, pages 164–171, Morgan Kaufmann, San Francisco, California.

Deb, K. (1999b). Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. *Evolutionary Computation*, 7(3):205–230.

Deb, K. and Goldberg, D. E. (1989). An Investigation of Niche and Species Formation in Genetic Function Optimization. In Schaffer, J. D., editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 42–50, Morgan Kaufmann, San Mateo, California.

Fonseca, C. M. and Fleming, P. J. (1993). Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion, and Generalization. In Forrest, S., editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, Morgan Kaufmann, San Mateo, California.

Fonseca, C. M. and Fleming, P. J. (1995a). An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation*, 3(1):1–16.

Fonseca, C. M. and Fleming, P. J. (1995b). Multiobjective Genetic Algorithms Made Easy: Selection, Sharing, and Mating Restriction. In *Proceedings of the First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pages 45–52, IEE, London, England.

Fonseca, C. M. and Fleming, P. J. (1997a). Multiobjective Optimization. In Bäck, T., Fogel, D. and Michalewicz, Z., editors, *Handbook of Evolutionary Computation*, Volume 1, pages C4.5:1 – C4.5:9, Oxford University Press, Oxford, England.

Fonseca, C. M. and Fleming, P. J. (1997b). Non-Linear System Identification with Multiobjective Genetic Algorithms. In *Proceedings of the Thirteenth Triennial World Congress of the International Federation of Automatic Control*, pages 187–192, Pergamon, London, England.

Fonseca, C. M. and Fleming, P. J. (1998). Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms – Part I: A Unified Formulation. *IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans*, 28(1):26–37.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, Reading, Massachusetts.

Goldberg, D. E. (1998). From Genetic and Evolutionary Optimization to the Design of Conceptual Machines. Technical Report 98008, Department of General Engineering, University of Illinois, Urbana, Illinois.

Horn, J. (1997). Multicriterion Decision Making. In Bäck, T., Fogel, D. and Michalewicz, Z., editors, *Handbook of Evolutionary Computation*, Volume 1, pages F1.9:1 – F1.9:15, Oxford University Press, Oxford, England.

Horn, J. and Nafpliotis, N. (1993). Multiobjective Optimization Using the Niched Pareto Genetic Algorithm. Technical Report 930005, Illinois Genetic Algorithms Laboratory (IlliGAL), University of Illinois, Urbana, Illinois.

Horn, J., Nafpliotis, N. and Goldberg, D. E. (1994). A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In Michalewicz, Z., editor, *Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 82–87, IEEE Press, Piscataway, New Jersey.

Hwang, C.-L. and Masud, A. S. M. (1979). *Multiple Objective Decision Making - Methods and Applications*. Springer Verlag, Berlin, Germany.

Ishibuchi, H. and Murata, T. (1998). Multi-Objective Genetic Local Search for Minimizing the Number of Fuzzy Rules for Pattern Classification Problems. In Fogel, D., editor, *Proceedings of the Fifth IEEE International Conference on Evolutionary Computation*, pages 1100–1105, IEEE Press, Piscataway, New Jersey.

Michalewicz, Z. and Nazhiyath, G. (1995). Genocop III: A Co-Evolutionary Algorithm for Numerical Optimization Problems with Nonlinear Constraints. In Fogel, D., editor, *Proceedings of the Second IEEE Conference on Evolutionary Computation*, pages 647–651, IEEE Press, Piscataway, New Jersey.

Michielssen, E. and Weile, D. S. (1995). Electromagnetic System Design Using Genetic Algorithms. In Winter, G., Periaux, J., Galan, M. and Cuesta, P., editors, *Genetic Algorithms in Engineering and Computer Science*, pages 345–369, John Wiley and Sons, Chichester, England.

Parks, G. T. and Miller, I. (1998). Selective Breeding in a Multiobjective Genetic Algorithm. In Eiben, A. E., Bäck, T., Schoenauer, M. and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature - PPSN V*. Springer, Berlin, Germany.

Pohlheim, H. (1998). GEATbx: Genetic and Evolutionary Algorithm Toolbox for use with MATLAB. Available: `http://www.geatbx.com/index.html`.

Rowe, J., Vinsen, K. and Marvin, N. (1996). Parallel GAs for Multiobjective Functions. In Alander, J. T., editor, *Second Nordic Workshop on Genetic Algorithms and Their Applications (2NWGA)*, pages 61–70, University of Vaasa, Vaasa, Finland.

Rudolph, G. (1998a). Evolutionary Search for Minimal Elements in Partially Ordered Finite Sets. In Porto, V. W., Saravanan, N., Waagen, D. and Eiben, A. E., editors, *Proceedings of the Seventh Annual Conference on Evolutionary Programming*, pages 345-353, Springer, Berlin, Germany.

Rudolph, G. (1998b). On a Multi-Objective Evolutionary Algorithm and Its Convergence to the Pareto Set. In *Proceedings of the Fifth IEEE Conference on Evolutionary Computation*, pages 511–516, IEEE Press, Piscataway, New Jersey.

Sandgren, E. (1994). Multicriteria Design Optimization by Goal Programming. In Adeli, H., editor, *Advances in Design Optimization*, pages 225–265, Chapman and Hall, London, England.

Schaffer, J. D. (1985). Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In Grefenstette, J. J., editor, *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, pages 93–100, Lawrence Erlbaum, Hillsdale, New Jersey.

Shaw, K. J. and Fleming, P. J. (1996). Initial Study of Multi-Objective Genetic Algorithms for Scheduling the Production of Chilled Ready Meals. In *Proceedings of the Second International Mendel Conference on Genetic Algorithms*, Technical University of Brno, Brno, Czech Republic.

Shaw, K. J., Nortcliffe, A. L., Thompson, M., Love, J., Fonseca, C. M. and Fleming, P. J. (1999). Assessing the Performance of Multiobjective Genetic Algorithms for Optimization of a Batch Process Scheduling Problem. In Angeline, P., editor, *1999 Congress on Evolutionary Computation*, pages 37–45, IEEE Press, Piscataway, New Jersey.

Srinivas, N. and Deb, K. (1994). Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248.

Thomas, M. W. (1998). *A Pareto Frontier for Full Stern Submarines via Genetic Algorithm*. Ph.D. thesis, Ocean Engineering Department, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Todd, D. S. and Sen, P. (1997). A Multiple Criteria Genetic Algorithm for Containership Loading. In Bäck, T., editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 674–681, Morgan Kaufmann, San Francisco, California.

Van Veldhuizen, D. A. (1999). *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Ph.D. thesis, AFIT/DS/ENG/99-01, Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

Van Veldhuizen, D. A. and Lamont, G. B. (1999). Multiobjective Evolutionary Algorithm Test Suites. In Carroll, J., Haddad, H., Oppenheim, D., Bryant, B. and Lamont, G. B., editors, *Proceedings of the 1999 ACM Symposium on Applied Computing*, pages 351–357, ACM, New York, New York.

Van Veldhuizen, D. A., Sandlin, B. S., Marmelstein, R. E., Lamont, G. B. and Terzuoli, A. J. (1998). Finding Improved Wire-Antenna Geometries with Genetic Algorithms. In Fogel, D., editor, *Proceedings of the Fifth IEEE International Conference on Evolutionary Computation*, pages 102–107, IEEE Press, Piscataway, New Jersey.

Vedarajan, G., Chan, L. C. and Goldberg, D. E. (1997). Investment Portfolio Optimization using Genetic Algorithms. In Koza, J. R., editor, *Late Breaking Papers at the Genetic Programming 1997 Conference*, pages 255–263, Stanford Bookstore, Stanford University, California.

Wienke, D., Lucasius, C. and Kateman, G. (1992). Multicriteria Target Vector Optimization of Analytical Procedures Using a Genetic Algorithm. *Analytica Chimica Acta*, 265:211–225.

Wolpert, D. H. and Macready, W. G. (1997). No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.

Zitzler, E. and Thiele, L. (1998). Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In Eiben, A. E., Bäck, T., Schoenauer, M. and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature - PPSN V*, Springer, Berlin, Germany.

Zitzler, E. and Thiele, L. (1999). Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271.