Multiobjective Evolutionary Algorithm Test Suites

David A. Van Veldhuizen and Gary B. Lamont
Department of Electrical and Computer Engineering
Graduate School of Engineering
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433-7765
{dvanveld,lamont}@afit.af.mil

We thank Mr. Mike Putney for his support during code development and implementation, and the ASC MSRC for providing the computing platforms.

Abstract

Multiobjective Evolutionary Algorithms (MOEAs) currently have no generic benchmark test suites. This paper provides several Multiobjective Optimization Problems (MOPs) for use as part of a standardized MOEA test suite, and proposes a methodology whereby various MOEAs can be directly compared. Supporting these contributions is a detailed discussion of MOP landscape and general test suite issues, and presentation of a new theorem defining the structural limitations of an MOP's global optimum. This paper also discusses high-performance computer software deterministically computing an MOP's Pareto front at a given computational resolution.

1 Introduction

Multiobjective Evolutionary Algorithms (MOEAs) are now a well-established field within Evolutionary Computation. They were "born" in 1985 when Schaffer [16] and Fourman [6] implemented the first MOEAs dealing with Multiobjective Optimization Problems (MOPs). Since then, over 140 published papers propose various MOEA implementations and applications, and to a much lesser extent, underlying MOEA theory [19]. Many of these efforts use numeric MOPs as examples to show algorithmic performance. Nowhere in the literature, however, is there a comprehensive discussion of MOP landscape issues; nor is there any explanation of why numeric MOPs may be appropriate MOEA test functions.

To date, most MOEA researchers' *modus operandi* is an algorithm's comparison (usually the researcher's own new and improved variant) against an older MOEA and analyzing results for some MOP (typically Schaffer's MOEA and examples [16]). Many other example numeric MOPs are also used [19]. Results are often "clearly" shown in graphical form, indicating which algorithm is more effective.

These empirical comparisons do not contribute much to a common basis for MOEA comparison. The literature's history of visually comparing MOEA performance on non-standard and unjustified numeric MOPs does little to determine a given MOEA's efficiency and effectiveness. A standard suite of numeric functions exhibiting *relevant* MOP problem domain characteristics provides a common comparative basis.

The MOEA community's limited *de facto* test suite contains many functions used only because they appear to exercise MOEA capabilities, or perhaps because other researchers used them as examples for their algorithms. Thus, a documented MOP test suite is an asset to MOEA research.

We provide various numeric MOPs for use in a standardized MOEA test suite, and propose a methodology whereby various MOEAs can be directly compared. Supporting these contributions is a detailed discussion of the MOP domain and general test suite issues; a new theorem defining one MOP characteristic and possible multiobjective NP-Complete problems are also presented.

This paper is organized as follows. Section 2 introduces necessary MOP concepts and related terminology. Section 3 discusses general test suite issues and proposes appropriate numeric examples given the MOP domain. Section 4 offers a methodology for quantitatively comparing MOEA performance. Our conclusions and future work are presented in Section 5.

2 Introducing the MOP Domain

Although a single objective optimization problem may have a unique optimal solution, MOPs (as a rule) present a possibly uncountable set of solutions, which when evaluated produce vectors whose components represent trade-offs in decision space. A decision maker then implicitly chooses an acceptable solution by selecting one of these vectors. In mathematical terms, an MOP minimizes[1] 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. Or in general,

$$\begin{array}{ll} \text{minimize} & F(\vec{x}) = (f_1(\vec{x}), \ldots, f_k(\vec{x})) \\ \text{subject to} & g_i(\vec{x}) \leq 0, i = 1, \ldots, m,\ \vec{x} \in \Omega\ . \end{array} \tag{1}$$

An MOP thus consists of n variables, m constraints, and k objectives, of which any or all of the objective functions may be linear or nonlinear [12]. The MOP's evaluation function, $F : \Omega \longrightarrow \Lambda$, maps the decision variables to vectors.

[1] Or maximizes, since $\min\{F(x)\} = -\max\{-F(x)\}$.

2.1 Pareto Optimality and Terminology

We have shown in a previous work [18] that the global optimum of an MOP is the Pareto front determined by evaluating each member of the Pareto optimal solution set. Although "Pareto optimality," and its related concepts and terminology are frequently invoked, they are sometimes used incorrectly in the literature. To ensure understanding and consistency, we define Pareto Dominance and Optimality, and introduce an associated notation. Using the MOP notation presented in Equation 1, key Pareto concepts are mathematically defined as follows [1]:

Definition 1 (Pareto Dominance): *A vector* $\mathbf{u} = (\mathbf{u_1}, \ldots, \mathbf{u_k})$ *is said to dominate* $\mathbf{v} = (\mathbf{v_1}, \ldots, \mathbf{v_k})$ *if and only if* $\mathbf{u}$ *is partially less than* $\mathbf{v}$*, 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 2 (Pareto Optimality): *A solution* $x_u \in \Omega$ *is said to be Pareto optimal if and only if there is no* $x_v \in \Omega$ *for which* $v = f(x_v) = (v_1, \ldots, v_k)$ *dominates* $u = f(x_u) = (u_1, \ldots, u_k)$. □

Pareto optimal solutions are also termed *non-inferior*, *admissible*, or *efficient* solutions. Their corresponding vectors are termed *non-dominated* [10]; selecting a vector(s) from this non-dominated vector set implicitly indicates acceptable Pareto optimal solutions (genotypes). These solutions may have no clearly apparent relationship besides their membership in the Pareto optimal set. This is the set of all solutions whose associated vectors are nondominated; we stress here that Pareto optimal solutions are classified as such based on their *phenotypical* expression. Their expression (the nondominated vectors), when plotted in criterion space, is known as the *Pareto front*. MOEA researchers have inconsistently used these terms in the literature, suggesting a more precise notation is required.

2.2 Pareto Notation

During MOEA execution, a "local" set of Pareto optimal solutions (with respect to the *current* MOEA population) is determined at each EA generation and termed $P_{current}$. Many MOEA implementations also use a secondary population, storing nondominated solutions found through the generations [19]. Because a solution's classification as Pareto optimal depends upon the context within which it is evaluated (i.e., the given set of which it's a member), corresponding vectors of this set must be (periodically) tested, removing solutions whose associated vectors are dominated.

We term this secondary population P_{known}. However, to reflect the possible changes in membership between generations we annotate our notation with the variable t representing the completion of t generations (e.g., $P_{known}(t)$). $P_{known}(0)$ is defined as $\emptyset$ and P_{known} alone as the *final* set of solutions returned by the MOEA.

Different secondary population storage strategies exist; the simplest is when $P_{current}$ is added at each generation (i.e., $P_{current} \bigcup P_{known}(t-1)$). 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 solution set (termed P_{true}) is not explicitly known for problems of any difficulty. P_{true} is defined by the functions composing an MOP; it is fixed and does not change. Because of the manner in which Pareto optimality is defined $P_{current}$ is always a non-empty solution set [18].

$P_{current}$, P_{known}, and P_{true} are sets of MOEA genotypes[2]; each set's phenotypes form a Pareto front. We term the associated Pareto front for each of these solution sets as $PF_{current}$, PF_{known}, and PF_{true}. Thus, when using an MOEA to solve MOPs, the implicit assumption is that one of the following holds: $P_{known} = P_{true}$, $P_{known} \subset P_{true}$, or $PF_{known} \in [PF_{true}, PF_{true} + \epsilon]$ over some norm (Euclidean, RMS, etc.).

2.3 MOP Domain Features

What is the nature of a given MOP's Pareto optimal set (P_{true})? Few MOEA efforts report any description of an example MOP's underlying decision variable space, i.e., the space where P_{true} resides. Since an MOP is normally composed of two or more single-objective optimization problems, the solution space is restricted by the limitations of those combined functions. Within that space, P_{true} may be connected, disconnected, a hyperarea, separate points, etc. However, in MOEA search the Pareto front is of more interest because solutions are often implicitly determined via selecting a point from PF_{known}.

What is the nature of the Pareto front (PF_{true})? It may be (dis)continuous, convex or concave, and multi-dimensional. Other researchers have noted that the structure of *any* Pareto front (independent of dimensionality) has theoretical limitations. For example, any Pareto surface must be monotonic (i.e., all first-order partial derivatives never change sign), and has asymptotic bounds [5, 11]. We have recently realized that an additional structural limitation exists, and developed a theorem describing it.

2.3.1 PF_{true}'s Structure

Theorem 1: The Pareto front of an MOP with $k = 2$ objectives is at most a (restricted) curve, and is at most a (restricted) surface when $k \geq 3$. □

Proof: By definition, all vectors of the Pareto front are *nondominated*. Given a minimization MOP, assume PF_{true} is either a polygon ($k = 2$) or a hypervolume ($k \geq 3$). Now take an imaginary line parallel to any of the objective axes passing through at least two points (represented by vectors) of the polygon or hypervolume. Since performance in each objective is to be minimized, one of these points is clearly "better" than the other and dominates it. But PF_{true} is composed only of *non*dominated vectors. Thus, the original assumption is incorrect and the Pareto front of an MOP with two objectives is at most a (restricted) curve, and that of an MOP with three or more objectives is at most a (restricted) surface. □

Horn states that in a k-objective MOP, the Pareto front *is* a $k-1$ dimensional surface [11]. We have just shown this is incorrect; PF_{true} is *at most* a surface *only when* $k \geq 3$. Although asymptotic bounds are useful, researchers must also understand the front's possible shape within those bounds.

This theorem also implies that any MOEA test suite should contain MOPs with both types of Pareto fronts: k-dimensional curves and 3-dimensional surfaces. This is necessary to fully test an MOEA's search capability.

[2] Horn [10] uses P_{online}, $P_{offline}$, and P_{actual} instead of $P_{current}$, P_{known}, and P_{true}.

3 An MOEA Test Function Suite

Test function suites have both supporters and detractors. Any algorithm successfully passing all submitted test functions has no guarantee of continual effectiveness and efficiency (i.e., examples prove nothing). Automotive passenger airbags are a prime example; not until they were widely fielded was it discovered that airbag-babyseat interactions were possibly deadly. Pattern recognition work has also recognized the problem of "testing on the training data," where an algorithm is tuned for only one or a few problem instances [3]. These analogies hold when integrating problem and algorithm domains: new and unforeseen situations may arise resulting in undesirable consequences. Thus, an MOEA test suite can be a valuable tool only if relevant issues are properly considered.

3.1 General MOEA Test Suite Issues

The "No Free Lunch" (NFL) theorem [22] implies that if problem domain knowledge is not incorporated into the algorithm domain, no formal assurances of an algorithm's general effectiveness exist. Previously, EA test suites containing various functions were proposed for testing an EA's capability to "handle" various problem domain characteristics. These suites incorporate relevant search space features which should be addressed by a particular EA instantiation. For example, De Jong [2] suggests five single-objective optimization test functions (**F1 - F5**), and Michalewicz [13] suggests five single-objective *constrained* optimization test functions(**G1 - G5**). Whitley et al. [21] and Goldberg et al. [8] offer other test suite functions. Whitley et al. also offer general test suite guidelines which include incorporating real world problems, problems ranging in difficulty from "easy" to "hard", and scalable problems.

De Jong's test bed includes functions with the following characteristics [7]: continuous and discontinuous, convex and nonconvex, unimodal and multimodal, quadratic and nonquadratic, low- and high-dimensionality, and deterministic and stochastic. Michalewicz's test bed addresses the following issues [13]: type of objective function, number of decision variables and constraints, types of constraints, number of active constraints at the function's optimum, and the ratio between the feasible and complete search space size. Particular EA instantiations are then subjected to test beds like these and judged on their performance.

Note that the NFL theorem also implies that incorporating too much problem domain knowledge into a search algorithm reduces its effectiveness on other problems. However, as long as a test suite involves only *major* problem domain characteristics, any search algorithm giving effective and efficient results over the test suite might remain broadly applicable to *problems from that domain*. Thus, we must define traits common to all (most) MOPs for test suite consideration.

When implementing an MOEA, it is (implicitly) assumed that the problem domain (fitness landscape) has been examined, and a decision made that an MOEA is the most appropriate search tool for the given MOP. We also assume the MOEA returns P_{known} , i.e., a *set* of solutions. It is not clear that all existing test functions are appropriate MOEA examples; thus, identification of appropriate functions is required to objectively determine MOEA efficiency and effectiveness.

In general, it is accepted that EAs are useful search algorithms when the problem domain is multidimensional (many decision variables), and/or the search space is very large. Many of the numerical examples used by MOEA researchers do *not* explicitly meet this criteria. Our research identifies over 25 different numerical MOPs (both constrained and unconstrained) used in published MOEA efforts [19]. All but three use at most two decision variables and the majority use only two objective functions. This implies that unless the search space is very large, MOEA performance claims and comparisons based on these functions may not be meaningful. The algorithm may be operating in a problem domain not particularly well-suited to its capabilities. Relevant MOP problem domain characteristics must be identified and considered in selecting appropriate MOEA test suite functions.

3.2 MOEA Test Suite Functions

As indicated, a *de facto* test suite exists. We have elsewhere cataloged these functions and in this paper identified several MOP characteristics which must be dealt with by effective and efficient MOEAs [19]. An extensive literature search has identified 28 numerical example MOPs; these MOPs incorporate 2-3 functions and 0-12 side constraints. We now propose initial problems for an MOP test suite drawn from the published literature, which incorporate selected characteristics and address some of the issues in Sections 3.1.

We initially propose the three MOPs listed in Table 1. **MOP1**, an unconstrained two-objective MOP [16], is selected for three primary reasons. First is its historical significance; almost all proposed MOEAs have been tested using this function. It is also an exemplar of relevant MOP concepts. Secondly, as we have noted elsewhere [18], this MOP allows us to determine an analytical expression for PF_{true} (a curve) through substitution. Third, as noted by Rudolph [15], this MOP's P_{true} is given in closed form. At any resolution, the representation of solutions composing P_{true} is thus easily determined without the necessity of exhaustively enumerating the search space. However, its one decision variable implies a large search space should be used when testing an MOEA.

MOP2 is also an unconstrained two-objective MOP, having the additional advantage of arbitrarily adding decision variables without changing PF_{true} 's structure; P_{true} is also given in closed form [4]. Figures 1 and 2 shows MOP2's Pareto optimal set and front.

Finally, we propose **MOP3** [20]. This MOP has three objective functions, and its Pareto front appears to be a k-dimensional curve following a convoluted path through objective space. This characteristic should challenge an MOEA's ability to find and maintain the entire front. Figures 3 and 4 show MOP3's Pareto optimal set and front.

Figures 1 through 4 are deterministically derived by computing all variable combinations possible at a given computational resolution. As this underlying resolution is varied the figures may slightly change.

Although previously presented in the literature, these MOPs' characteristics address some of the issues mentioned in Section 3.1. MOP1 is arguably an "easy" MOP. MOP2 is scalable, in that any number of decision variables may be used to increase the search space. MOP3's functions result in a nonlinear and asymmetric PF_{true} . Taken together they begin to form a coherent basis for MOEA comparisons. Other relevant MOP characteristics should be addressed by further MOPs selected for test suite inclusion. Desired MOPs may need to be specifically constructed to exhibit desired characteristics.

Table 1: Initial MOEA Test Suite Functions [19]

MOP	Definition	Constraints
MOP 1	$F = (f_1(x), f_2(x))$, where $f_1(x) = x^2,\; f_2(x) = (x-2)^2$	None
MOP 2	$F = (f_1(\vec{x}), f_2(\vec{x}))$, where $f_1(\vec{x}) = 1 - \exp(-\sum_{i=1}^{n}(x_i - \frac{1}{\sqrt{n}})^2),\; f_2(\vec{x}) = 1 - \exp(-\sum_{i+1}^{n}(x_i + \frac{1}{\sqrt{n}})^2)$	$-2 \leq x_i < 2$
MOP 3	$F = (f_1(x,y), f_2(x,y), f_3(x,y))$, where $f_1(x,y) = 0.5 * (x^2+y^2) + sin(x^2+y^2),\; f_2(x,y) = \frac{(3x-2y+4)^2}{8} + \frac{(x-y+1)^2}{27} + 15,\; f_3(x,y) = \frac{1}{(x^2+y^2+1)} - 1.1e^{(-x^2-y^2)}$	$-3 \leq x, y \leq 3$

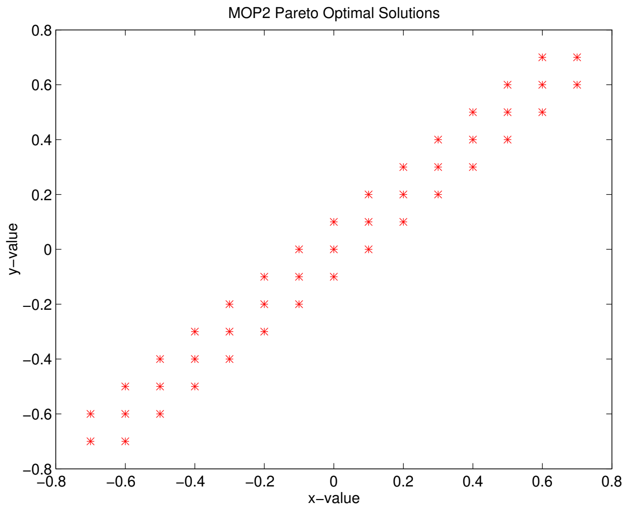

Figure 1: MOP2's Pareto Optimal Set

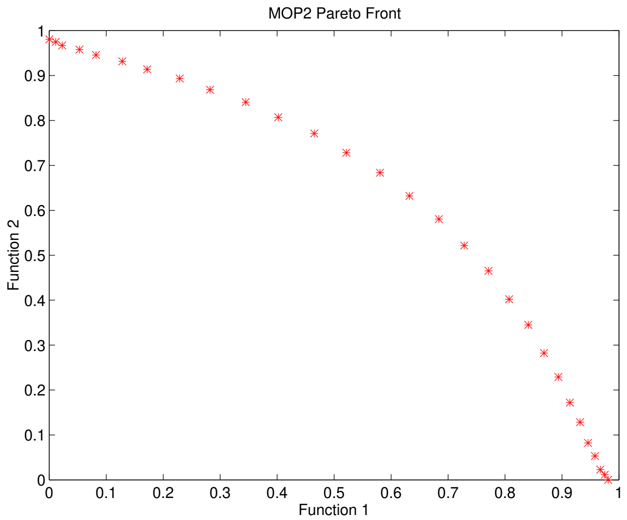

Figure 2: MOP2's Pareto Front

3.3 NP-Complete MOPs

We also consider the use of combinatorial optimization problems in the test suite. EAs often employ specialized representations and operators when solving these real-world problems. This may prevent general comparison between various MOEAs, but the problems' inherent difficulty should present the desired algorithmic challenges and complement numeric test suite MOPs. Table 2 suggests possible NP-Complete MOPs for inclusion. To date, only two non-numerical MOP examples are found in the MOEA literature: one is a multiobjective NP-Complete example (a multiobjective knapsack problem) [23], and the other is a unitation problem [11].

4 MOEA Experimental Methodology

Having investigated the MOP domain and proposed functions for an MOP test suite, we are now in a position to perform meaningful MOEA experiments. Although test suite functions provide a common basis for MOEA comparison, these comparisons are still empirical unless PF_{true} is known. We earlier intimated this is the case for most MOPs of any

Table 2: Possible Multiobjective NP-Complete Functions

NP-Complete Problem	**Examples**
0/1 Knapsack - Bin Packing	Max profit; Min weight
Traveling Salesperson	Min energy, time, and/or distance; Max expansion
Coloring	Min # colors, # of each color
Set/Vertex Covering	Min total cost, over-covering
Maximum Independent Set (Clique)	Max set size; Min geometry
Vehicle Routing	Min time, energy, and/or geometry
Scheduling	Min time, missed deadlines, waiting time, resource use
Layout	Min space, overlap, costs
NP-Complete Problem Combinations	Vehicle scheduling and routing

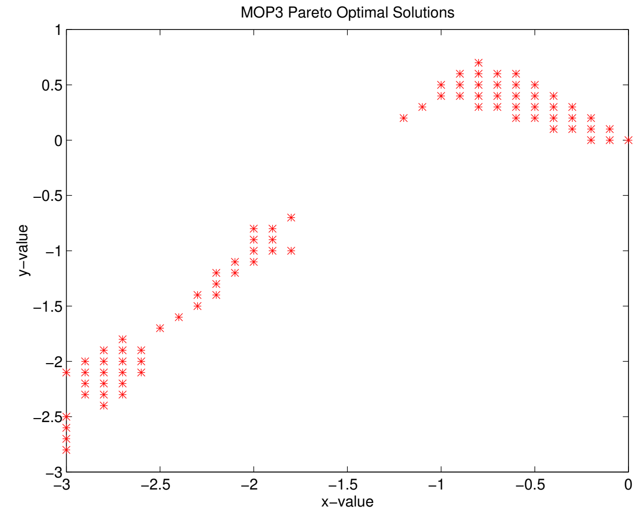

Figure 3: MOP3's Pareto Optimal Set

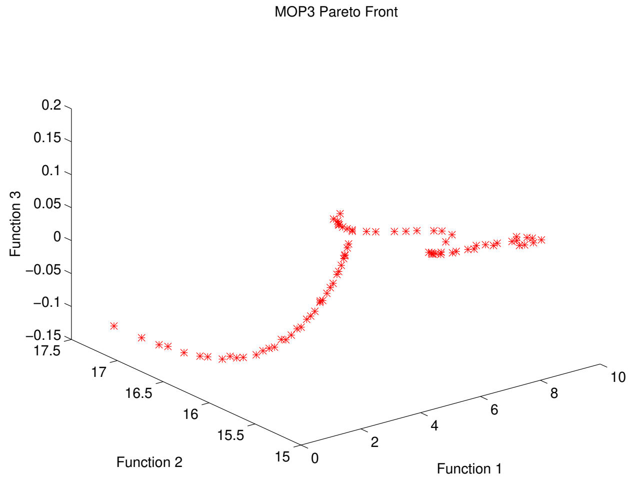

Figure 4: MOP3's Pareto Front

complexity. However, there is a way to determine PF_{true} at a given computational resolution! This section describes such a process we are currently using to construct an experimental database. It also suggests appropriate metrics when quantitatively comparing MOEAs.

4.1 Experimental Database

When the real (continuous) world is modeled (e.g., via objective functions) on a computer (a discrete machine), there is a fidelity loss between the real world and implemented model. However, at a standardized resolution and representation, MOEA results can be compared against both each other *and* PF_{true}. Thus, whether or not a given MOP's true Pareto front is actually continuous or discrete is then not a major concern, as the computed front is always composed of discrete points at a specified computational resolution.

For purposes of this discussion we determine *computational resolution* by the equidistant sampling of points in each decision variable dimension. For example, with a fixed length binary string, the associated decision variable value is determined by mapping the binary string to an integer i and solving the following:

$$l + \frac{i * (u - l)}{2^n - 1}\ , \qquad (2)$$

where l and u correspond to the lower and upper decision variable bounds, and n is the length of the binary string.

We suggest both a large search space and a binary encoding for any MOEA comparisons. The large search space challenges an MOEA's ability to find the global optimum; the binary encoding allows for both a standard chromosomal representation and a method for deterministically searching an entire space. At a given resolution, each of the 2^n binary strings is enumerated and evaluated, thus obtaining P_{true} and PF_{true} (at that resolution).

For example, assume a 30-bit binary encoding. This search space contains 2^{30} possible solutions, over one billion distinct possibilities! Published results indicate most EAs execute between thousands and tens of thousands of fitness evaluations during search; the effectiveness of any MOEA should be readily apparent in how well its results (P_{known} and PF_{known}) compare to P_{true} and PF_{true} in a search space of this large size. We have constructed an experimental database allowing this comparison for the proposed MOEA test suite functions [19].

This effort is in part due in part to a paper suggesting that exhaustive search may be the only viable approach to solving irregular or chaotic problems [14]. The authors propose harnessing ever-expanding computational capability to solve problems by exhaustive deterministic enumeration. We constructed such a program executing on the IBM SP-2.

Our program's "C" implementation uses the Message Passing Interface (MPI) to distribute function evaluations among many processors. For a given MOP, each processor initially evaluates some subset of solutions and stores the resultant vectors. These vectors are compared on the basis of Pareto optimality; nondominated solutions and their corresponding vectors are written to disk. Noting that Pareto optimality places a partial ordering on the entire search space, combining the separate solutions/vectors from different processors and again comparing vectors results in P_{true} *for that particular binary encoded resolution*. Program timing and processor loading are also recorded to determine problem scaling. To date, we have successfully extracted one MOP's P_{true} from a search space of size 2^{26} and am currently computing MOP2 and MOP3's P_{true} .

Using this database, solutions offered by various MOEAs can be compared not only against each other, but against the *true* Pareto optimal set and front. At least for selected MOPs, relativity is removed and a true quantitative comparison is possible. This methodology allows absolute, rather than relative observations.

4.2 MOEA Comparative Metrics

We propose quantitative comparisons between MOEA implementations via a rigorous, carefully designed series of experiments. However, we must first define the metrics upon which we base MOEA performance.

Deterministic enumeration provides P_{true} and PF_{true} (at a given level of resolution). After executing an MOEA on some MOP we are then able to compare the reported front (PF_{known}) against the true front and determine error measures. The following are possible metrics for this purpose.

Error Ratio. An MOEA reports a finite number of solutions; these solutions are or are not members of PF_{true} . If they are not the MOEA has erred. This metric is mathematically represented by:

$$\frac{\sum_{i=1}^{n} e_i}{n}, \tag{3}$$

where n is the number of vectors in PF_{known} ; $e_i = 0$ if vector i is a member of PF_{true} , and 1 otherwise. For example, an error of "0" means that every vector reported by the MOEA in PF_{known} is actually in PF_{true} ; an error of 100 means that none are.

Generational Distance. Used in earlier experiments [18], this metric may be effective in gauging MOEA performance. Generational distance is a value representing how "far" PF_{known} is from PF_{true} and is defined as:

$$G \triangleq \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n}, \tag{4}$$

where n is the number of vectors in PF_{known} and d_i is the distance (in objective space) between each of these and the *nearest* member of PF_{true} .

Coverage. Zitzler and Thiele propose two MOEA comparative metrics [23]. First is that of *coverage*. Coverage occurs when one solution's associated objective vector (phenotype) dominates another's (mathematically represented by $a \prec b$), or the two solutions are equal ($a = b$). Coverage defines the size of objective value space covered by PF_{known} . For example, a point on PF_{known} in the two-dimensional (minimization) case defines a rectangle bounded by the origin and $(f_1(\vec{x}), f_2(\vec{x}))$. The union of all such rectangles defined by each vector in PF_{known} is used as the comparative measure. However, if PF_{true} is not convex this metric may be misleading. Thus, for any two P_{known} sets, they compute the fraction of solutions in one set covered by solutions in the other.

Spread. Another possible metric is one measuring the spread (distribution) of vectors throughout PF_{known} . Many MOEAs perform sharing and niching [5, 11, 17], attempting to spread each population ($PF_{current}$) evenly along the front. Because the true front's "beginning" and "end" are known (at some resolution), a suitably defined metric judges how well PF_{known} conforms. Srinivas and Deb [17] define such a measure expressing how well an MOEA has distributed individuals over a nondominated region. They define this metric as:

$$\iota = \sqrt{\sum_{i=1}^{q+1} \left(\frac{n_i - \overline{n}_i}{\sigma_i}\right)^2}, \tag{5}$$

where q is the number of desired optimal points and the $(q+1)$-th subregion is the dominated region, n_i is the actual number of individuals serving the ith subregion (niche) of the nondominated region, $\overline{n}_i$ is the expected number of individuals serving the ith subregion of the nondominated region, and σ_i^2 is the variance of individuals serving the ith subregion of the nondominated region. They show that if the distribution of points is ideal with $\overline{n}_i$ number of points in the ith subregion, the performance measure $\iota = 0$. Thus, a low performance measure characterizes an algorithm with a good distribution capacity. However, this metric only measures spread uniformity.

5 Conclusions

In the tradition of providing test suites for evolutionary algorithms, we propose possible MOEA test functions for evaluation. The development of this list is based upon accepted and historic EA test suite guidelines. Specific MOP test suites can evolve from our list based upon an individual researcher's objectives, including problem characteristic classification. We have classified numeric MOPs based upon type, dimensionality, constraint structure, landscape, and Pareto front structure (curve, surface, continuous, connectiveness, symmetry, convexity). Also, we have shown that a Pareto front is at most a hyper-surface, not a hyper-volume. With a generic MOEA test suite, researchers can now compare their numeric and NP-Complete problem results (regarding effectiveness and efficiency) with others, over a spectrum of MOEAs. Using the proposed Pareto terminology and test suite functions, as well as high performance computational experiments such as ours, MOEA comparison efforts can be made more precise.

Our future efforts include extending the MOEA test suite and experimental methodology laid forth in this paper. We are constructing additional "MOEA challenging" MOPs for the test suite, and investigating other high-performance computational environments. Finally, the MOEA test suite and experiments aid our development and analysis of parallel and distributed MOEAs.

References

[1] Ben-Tal, Aharon. "Characterization of Pareto and Lexicographic Optimal Solutions." *Multiple Criteria Decision Making Theory and Application 177*. Lecture Notes in Economics and Mathematical Systems, edited by G. Fandel and T. Gal, 1–11, Berlin: Springer-Verlag, 1980.

[2] De Jong, Kenneth A. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD dissertation, The University of Michigan, Ann Arbor MI, 1975.

[3] Duda, Richard O. and Peter E. Hart. *Pattern Classification and Scene Analysis*. New York, NY: John Wiley & Sons, 1973.

[4] Fonseca, Carlos M. and Peter J. Fleming. "Multiobjective Genetic Algorithms Made Easy: Selection, Sharing, and Mating Restriction." *Proceedings of the 1st International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*. Number 414. 45–52. September: IEE, 1995.

[5] Fonseca, Carlos M. and Peter J. Fleming. "Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms – Part I: A Unified Formulation, and Part II: Application Example," *IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans*, *28*(1):26–47 (January 1998).

[6] Fourman, Michael P. "Compaction of Symbolic Layout Using Genetic Algorithms." In Grefenstette [9], 141–153.

[7] Goldberg, David E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison Wesley, 1989.

[8] Goldberg, David E., et al. "Messy Genetic Algorithms: Motivation, Analysis, and First Results," *Complex Systems*, *3*:493–530 (1989).

[9] Grefenstette, John J., editor. *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, Hillsdale, New Jersey: Lawrence Erlbaum Associates, Publishers, July 1985.

[10] Horn, Jeffrey. *Handbook of Evolutionary Computation*, *1*, chapter Multicriterion Decision Making, F1.9:1 – F1.9:15. IOP Publishing Ltd. and Oxford University Press, 1997.

[11] Horn, Jeffrey and Nicholas Nafpliotis. *Multiobjective Optimization Using the Niched Pareto Genetic Algorithm*. Technical Report 930005, University of Illinois at Urbana-Champaign, 117 Transportation Building, 104 South Matthews Avenue, Urbana, IL 61801-2996: Illinois Genetic Algorithms Laboratory (IlliGAL), July 1993.

[12] Hwang, Ching-Lai and Abu Syed Md. Masud. *Multiple Objective Decision Making - Methods and Applications*. Springer Verlag, 1979.

[13] Michalewicz, Zbigniew. "Genetic Algorithms, Numerical Optimization, and Constraints." *Proceedings of the Sixth International Conference on Genetic Algorithms*, edited by Larry J. Eshelman. 151–158. San Mateo CA: Morgan Kaufmann Publishers, Inc., 1995.

[14] Nievergelt, Jürg, et al. "All the Needles in a Haystack: Can Exhaustive Search Overcome Combinatorial Chaos?." *Computer Science Today: Lecture Notes in Computer Science 1000*, edited by J. van Leeuwen. 254–274. Berlin: Springer, 1995.

[15] Rudolph, Günter. "On a Multi-Objective Evolutionary Algorithm and Its Convergence to the Pareto Set." *Proceedings of the 1998 IEEE Conference on Evolutionary Computation*. 1998.

[16] Schaffer, J. David. "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms." In Grefenstette [9], 93–100.

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

[18] Van Veldhuizen, David A. and Gary B. Lamont. "Evolutionary Computation and Convergence to a Pareto Front." *Late Breaking Papers at the Genetic Programming 1998 Conference*, edited by John R. Koza. 221–228. Stanford, CA: Stanford University Bookstore, July 1998.

[19] Van Veldhuizen, David A. and Gary B. Lamont. *Multiobjective Evolutionary Algorithm Research: A History and Analysis*. Technical Report TR-98-03, Air Force Institute of Technology, 1998.

[20] Viennet, R., et al. "Multicriteria Optimization Using a Genetic Algorithm for Determining a Pareto Set," *International Journal of Systems Science*, *27*(2):255–260 (1996).

[21] Whitley, Darrell, et al. "Evaluating Evolutionary Algorithms," *Artificial Intelligence*, *85*:245–276 (1996).

[22] Wolpert, David H. and William G. Macready. "No Free Lunch Theorems for Optimization," *IEEE Transactions on Evolutionary Computation*, *1*(1):67–82 (April 1997).

[23] Zitzler, Eckart and Lothar Thiele. "Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study." *Parallel Problem Solving from Nature - PPSN V*, edited by A. E. Eiben, et al. Berlin: Springer, 1998.