Evolutionary Computation and Convergence to a Pareto Front

David A. Van Veldhuizen
Department of Electrical and Computer Engineering
Graduate School of Engineering
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433-7765
{dvanveld}@afit.af.mil
(937) 255-3636 ext. 4704

Gary B. Lamont
Department of Electrical and Computer Engineering
Graduate School of Engineering
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433-7765
{lamont}@afit.af.mil
(937) 255-3636 ext. 4718

ABSTRACT

Research into solving multiobjective optimization problems (MOP) has as one of its an overall goals that of developing and defining foundations of an Evolutionary Computation (EC)-based MOP theory. In this paper, we introduce relevant MOP concepts, and the notion of Pareto optimality, in particular. Specific notation is defined and theorems are presented ensuring Pareto-based Evolutionary Algorithm (EA) implementations are clearly understood. Then, a specific experiment investigating the convergence of an arbitrary EA to a Pareto front is presented. This experiment gives a basis for a theorem showing a specific multiobjective EA statistically converges to the Pareto front. We conclude by using this work to justify further exploration into the theoretical foundations of EC-based MOP solution methods.

1 Introduction

Our research focuses on solving scientific and engineering multiobjective optimization problems (MOPs), contributing to the overall goal of developing and defining foundations of an Evolutionary Computation (EC)-based MOP theory. This paper addresses shortfalls recently identified by Horn (Horn 1997), focusing on MOP theory development and experimentation. Solving MOPs with EC methods presents several unique challenges. For a good introduction to the relevant issues and past EC-based approaches, see the articles by Fonseca and Fleming (Fonseca and Fleming 1995), and by Horn (Horn 1997).

This paper focuses on EC's ability to converge to the p-dimensional hypersurface defined by simultaneously optimizing p functions. For example, by optimizing a bi-objective problem where both objectives are functions of one variable, we can determine necessary analytical expressions for use in a specific EC convergence computation. We then show an arbitrary Evolutionary Algorithm (EA) statistically converges (given specific MOPs) and generalize our results.

The remainder of this paper is organized as follows. Section 2 introduces relevant MOP concepts. Section 3 describes our experiments with an example MOP and the software environment used. Section 4 presents and presents a theorem showing a multiobjective EA statistically converges to the Pareto front. Finally, Section 5 presents our conclusions and proposes promising directions for further research.

2 Key MOP Concepts

Although single-objective optimization problems may have a unique optimal solution, MOPs (as a rule) offer 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. Mathematically speaking, an MOP minimizes (or maximizes, since $\min\{F(x)\} = -\max\{-F(x)\}$) the components of a vector $f(x)$ where x is an n-dimensional decision variable vector from some universe $\mathcal{U}$. Or in general,

$$\begin{aligned} &\text{minimize} \quad f(x) = (f_1(x), \ldots, f_p(x)) \\ &\text{subject to} \quad g_i(x) \leq 0, i = 1, \ldots, m \ , \end{aligned} \tag{1}$$

An MOP then consists of n variables, m constraints, and p objectives ($p \geq 2$), of which any or all of the objective functions may be nonlinear (Hwang and Masud 1979). MOPs are often characterized by measures of performance (objectives) which may be (in)dependent and/or non-commensurable; the multiple objectives being optimized almost always conflict. These opposing objectives place a partial, rather than total, ordering on the search space. In order to successfully deal with these characteristics, several EC-based methods (Fonseca and Fleming 1995, Horn 1997) were developed to determine optimal solutions given an MOP's objectives and constraints. Key to many of these methods, however, is the use of *Pareto Optimality* in determining a (set of) MOP solutions.

2.1 Pareto Optimality

Although "Pareto optimality" and its related concepts/terminology are frequently invoked they are sometimes used incorrectly in the literature. To ensure understanding and consistency, we introduce Pareto Dominance and Optimality in this section, and introduce a consistent notation in Section 2.2. Using the MOP presented in Equation 1, key Pareto concepts are mathematically defined as follows (Ben-Tal 1980).

Definition 1 (Pareto Dominance): *A vector* $\mathbf{u} = (u_1, \ldots, u_p)$ *is said to dominate* $\mathbf{v} = (v_1, \ldots, v_p)$ *if and only if* $\mathbf{u}$ *is partially less than* $\mathbf{v}$*, i.e.,* $\forall i \in \{1, \ldots, p\}, u_i \leq v_i \wedge \exists i \in \{1, \ldots, p\} : u_i < v_i$. □

As an example of Pareto dominance we use the optimization problem:

$$\begin{aligned} &\text{minimize } f(x_u), u \in \{a, b, c, d\}, \text{where} \\ &a \triangleq f(x_a) = (3.25, 1.76, 4.67), \\ &b \triangleq f(x_b) = (3.25, 1.76, 4.67), \\ &c \triangleq f(x_c) = (3.15, 1.76, 4.67), \\ &d \triangleq f(x_d) = (3.15, 1.76, 4.22). \end{aligned} \tag{2}$$

Here, a and b are dominated by both c and d; c is dominated by d, and d dominates all other vectors.

Definition 2 (Pareto Optimality): *A solution* $x_u \in \mathcal{U}$ *is said to be Pareto optimal if and only if there is no* $x_v \in \mathcal{U}$ *for which* $v = f(x_v) = (v_1, \ldots, v_p)$ *dominates* $u = f(x_u) = (u_1, \ldots, u_p)$. □

The solution x_d in Equation 2's example is the only element of the associated *Pareto optimal set*; i.e., it is a Pareto optimal solution of the set $\{x_a, x_b, x_c, x_d\}$. Pareto optimal solutions are also termed *non-inferior*, *admissible*, or *efficient* solutions. Their corresponding vectors are termed *non-dominated* (Horn 1997); selecting a vector(s) from this non-dominated vector set implicitly indicates acceptable Pareto optimal solutions. These solutions may have no clearly apparent relationship besides their membership in the Pareto optimal set. We stress here that Pareto optimal solutions are classified as such based on their *phenotypical* expression. Their expression (the non-dominated vectors), when plotted in criterion space, is known as the *Pareto front*. Researchers have sometimes inconsistently used these terms in the literature.

As another example of Pareto optimality we present the one-variable, two-objective problem $F1$. This is the same problem used by Vincent and Grantham, Schaffer, and Srinivas and Deb for the same purpose (Srinivas and Deb 1994). The problem's two objectives are defined as:

$$\begin{aligned} \text{Minimize } f_{11} &= x^2 , \\ \text{Minimize } f_{12} &= (x-2)^2 . \end{aligned} \tag{3}$$

By looking at Figure 1 it's apparent the Pareto optimal set is $\{x \mid 0 \leq x \leq 2\}$. The solution $x = 0$ is optimal with respect to f_{11} but not f_{12}; the solution $x = 2$ is optimal with respect to f_{12} but not f_{11}. Any solution $\{x \mid x \notin 0 \leq x \leq 2\}$ is not a member of the Pareto optimal set because it is not better than a solution in the set with respect to either objective. For the general case, Rudolph (Rudolph 1998*b*) recently showed that given

$$\begin{aligned} \text{Minimize } f_{11} &= ||x||^2 , \\ \text{Minimize } f_{12} &= ||x-z||^2 \text{, with } 0 \neq z \in R , \end{aligned} \tag{4}$$

the Pareto optimal set for this general MOP is:

$$\mathcal{X}^* = \{x \in R \mid x = rz,\ r \in [0,1]\} . \tag{5}$$

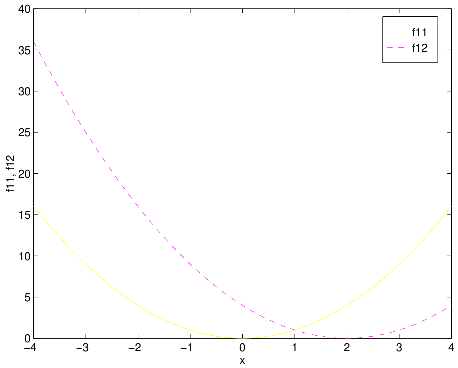

Figure 1 Function f_{11} and f_{12} Values.

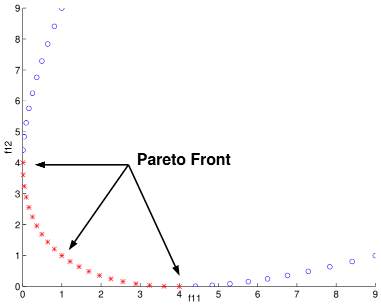

Figure 2 Function F_1's Pareto Front (f_{11} plotted against f_{12}).

We point out a significant difference between Figures 1 and 2. Figure 1 plots the values of functions f_{11} and f_{12} for different values of the independent variable. However, Figure 2 represents the values of function f_{11} plotted against those of function f_{12} *for the same value of the independent* *variable*. In other words, Figure 2 is a graph in "phenotype space." This plot graphically displays the non-dominated vectors for this problem *as points* in criterion space; this representation forms the Pareto front.

Identifying a set of Pareto optimal solutions is thus key for a decision maker to choose a "compromise" solution satisfying the objectives as "best" possible. Choosing an MOP solution optimized for only one objective may well ignore solutions, which from an overall standpoint, are better. The Pareto optimal solution set defines that overall view. Of course, the accuracy of the decision maker's view depends on both the *true* Pareto optimal set and the set presented *as* Pareto optimal.

EA-derived solutions of "real-world" MOPs offer only a finite number of points which may or may not be truly Pareto optimal. Also, complex MOPs don't generally lend themselves to analytical determination of the actual Pareto front. This raises the issue of EC convergence to the true Pareto front, which we address in Section 4.

2.2 Pareto Methodology for EAs

During EA execution, a "local" set of Pareto optimal solutions (with respect to the *current* EA population) is determined at each EA generation and termed $P_{current}$. Like other practical applications (Horn 1997), this solution set is added to a secondary population termed P_{known} (i.e., $P_{current} \bigcup P_{known}$), and the process continued until EA termination. Because a solution's classification as Pareto optimal is dependent upon the context within which it is evaluated (i.e., the given set of which it's a member), all corresponding vectors of P_{known} are tested each generation and solutions whose associated vectors are dominated are removed. The result is a final set of Pareto optimal solutions *found by the EA*. Of course, the actual Pareto optimal solution set (termed P_{true}) is not known for problems of any difficulty. This set is defined by the functions composing the MOP; P_{true} is fixed and does not change.[1]

$P_{current}$ and P_{known} are sets of EA genotypes. EA fitness is judged via phenotypes, which is a Pareto front in the MOP case. We term the associated Pareto front for each of the above sets as $PF_{current}$, PF_{known}, and PF_{true}. Thus, when using an EA to solve MOPs, the implicit assumption is that one of the following holds: $P_{known} = P_{true}$, $P_{known} \subset P_{true}$, or $PF_{known} \in [PF_{true} - \epsilon, PF_{true}]$ over some norm.

Because of the manner in which Pareto optimality is defined, $P_{current}$ is always a non-empty set. As this may be non-intuitive, and because we assume this in our EA implementation, we present the following theorem showing this for the general case.

Theorem 1: Given any non-empty solution set, at least one Pareto optimal solution exists within that set. □

[1]Horn (Horn 1997) uses P_{online}, $P_{offline}$, and P_{actual} instead of $P_{current}$, P_{known}, and P_{true}.

Proof: Assume n elements of an arbitrary solution set. When evaluated in a minimization problem n vectors result. Without loss of generality, assume a one-dimensional function which when evaluated results in n 1-D vectors. Order these vectors from "smallest" to "largest", which we can do because by definition, a total ordering exists. There are then two cases to consider:

Case I. The "smallest" vector is strictly less than or equal to all other vectors. Since the vectors are ordered this vector is Pareto dominant (by definition) and its associated solution is Pareto optimal.

Case II. Beginning with the "smallest" vector, two or more vectors are equal in value. Since the vectors are ordered, these particular vectors are guaranteed to be non-dominated since no other vector dominates them, and neither (or any other vector of equal value) dominates the other(s). Thus, since they are non-dominated, their associated solutions are also Pareto optimal.

Without loss of generality, the same reasoning holds when extended to vectors of two or more dimensions except that the ordering is now partial, and to maximization problems where the vector ordering is now "largest" to "smallest". Thus, any non-empty solution set has at least one Pareto optimal solution. □

2.3 Convergence Conjecture

The global optimum for an MOP is a *set* of vectors. Pareto optimal solutions are those in which performance in one objective *cannot* be improved without adversely affecting another objective. Thus, the Pareto front (PF_{true}) determined by evaluating the Pareto optimal solution set (P_{true}) is the global optimum of an MOP. This view forms the basis for the following conjecture (based on a theorem presented by Bäck (Bäck 1996, pg. 129)).

Conjecture 1: The global optimum of an MOP is a Pareto front $\vec{\mathcal{F}}^*$, composed of at most an uncountably infinite number of vectors $v_1^*, v_2^*, \ldots, v_{\aleph_1}^*$. An EA using Pareto-based ranking and a monotonic selection algorithm converges to the global optimum, i.e.,

$$\mathcal{P}\{\lim_{t \longrightarrow \infty} \vec{\mathcal{F}}^* \in P(t)\} = 1 \ ,$$

where $P(t) = P_{current}$. □

We show this conjecture is true in Section 4.

3 Experimentation

EC embodies the techniques of *Genetic Algorithms* (GAs), *Genetic Programming* (GP), *Evolution Strategies* (ESs), and *Evolutionary Programming* (EP), collectively known as EAs. Several research efforts successfully used Pareto optimality as the basis for EA fitness assignment in solving MOPs (Fonseca and Fleming 1995, Horn 1997). We investigate here whether using Pareto-based fitness ranking results in EA "convergence" to PF_{true}.

3.1 Software Environment

We chose the Genetic and Evolutionary Algorithm Toolbox (GEATbx) v1.91 (Pohlheim 1998) as the environment within which to perform our experiments. This toolbox operates under *MATLAB*[2] and provides a versatile tool set for implementing a wide range of EA methods. It eases development of user-specific routines for integration into GEATbx, and allows access to *MATLAB* data analysis and visualization functions for use with GEATbx-derived data. We easily modified GEATbx to solve MOPs.

3.2 Example Problem I

To experimentally determine if a specific EA exhibits convergence to PF_{true} we use the MOP example given in Equation 3. Figure 2 plots function f_{11} against f_{12} using the computed objective values for a particular input value. If the EA exhibits the desired convergence, PF_{kknown} should move closer and closer to PF_{true}.

In this case we're able to determine PF_{true}'s analytical expression because both objective functions are functions of one variable. In order to determine the polynomial equation describing PF_{true} we write f_{12} as a function of f_{11}.[3] We first define:

$$\begin{aligned} y_1 &\triangleq f_{11} = x^2 , \\ y_2 &\triangleq f_{12} = (x-2)^2 . \end{aligned} \tag{6}$$

Then, multiplying out y_2 and substituting in y_1 gives PF_{true}'s equation as

$$y_2 = (x-2)^2 = x^2 - 4x + 4 = y_1 - 4\sqrt{y_1} + 4 . \tag{7}$$

Because we've shown y_2 is a function of y_1 we can assume a standard coordinate system for the following computations. The equation to determine distance between any given point (i.e., EA MOP phenotypes) and a point on the given curve (Equation 7) is then given by

$$\begin{aligned} d_i &= \sqrt{(y_1 - y_{1_0})^2 + (y_2 - y_{2_0})^2} \\ &= \sqrt{(y_1 - y_{1_0})^2 + [f(y_1) - y_{2_0}]^2} , \end{aligned} \tag{8}$$

where y_{1_0} and y_{2_0} are EA-returned fitness values, and y_1 and y_2 are coordinates of a point on PF_{true}.

We wish to determine the distance between an arbitrary point and the *nearest* point on a given curve, where the point corresponds to a member of $PF_{current}$ and the given curve is PF_{true}. For easier mathematical manipulation Equation 8 is modified to reflect the distance squared:

$$F(x) = (\text{Distance})^2 = (y_1 - y_{1_0})^2 + [f(y_1) - y_{2_0}]^2 . \tag{9}$$

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

[3] Single-valued objective functions must be invertible in order to obtain an analytical formula for a Pareto front.

Using the Pareto front's equation (Equation 7) and setting the derivative of $F(x)$ equal to zero (since minimizing the square of the distance simultaneously minimizes the distance) results in:

$$\begin{aligned} 0 &= F'(x) \\ &= 2(y_1 - y_{1_0}) + 2[f(y_1) - y_{2_0}][f'(y_1)] \\ &= 2y_1 - 2y_{1_0} + \\ &\quad [2(y_1 - 4\sqrt{y_1} + 4) - 2y_{2_0}][1 - \frac{2\sqrt{y_1}}{y_1}] \\ &= 4y_1 - 12\sqrt{y_1} + \frac{4\sqrt{y_1}}{y_1}(y_{2_0} - 4) + \\ &\quad 24 - 2y_{2_0} - 2y_{1_0} . \end{aligned} \tag{10}$$

Finally, for an arbitrary tuple (y_{1_i}, y_{2_i}) in $P_{current}$, solving Equation 10 for y_1 allows Equation 8 to be solved and a distance d_i between (y_{1_i}, y_{2_i}) and the nearest point on PF_{true} determined. Several solutions in each generation may be elements of $P_{current}$. Thus, as a measure of distance between $PF_{current}$ and PF_{true} we define a *generational distance*:

$$G_j \triangleq \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} , \tag{11}$$

where n is the number of points in $P_{current}$ at generation j, i.e., Equation 8 is solved for each member of $P_{current}$ and the results used to derive a value for Equation 11. The generational distance value then determines the error between $PF_{current}$ and PF_{true}.

To perform our experiment we chose the default GA implementation provided by GEATbx, requiring few user-specified parameters. We set the following necessary options: a real-valued representation, 50 individuals in the GA population, terminate after 100 generations, and search space bounds of [-100,100]. Also, a secondary solution population (P_{known}) was constructed and integrated into the GA. At each generation, $P_{current} \bigcup P_{known}$ and P_{known} updated. To solve the MOP under study, the GA was augmented by the Pareto-based ranking and linear fitness assignment scheme described by Fonseca (Fonseca 1995).

Appropriate fitness values were assigned to each member of the GA's population by developing two additional *MATLAB* routines. The first determines which associated vectors of a given solution set are non-dominated and assigns them a rank of '0.' The remaining solutions (all of which are dominated) are given a rank equal to the number of associated vectors dominating theirs. The second routine transforms the rankings into fitness; all solutions of equal rank receive identical fitness values. Using the above parameters and methodology, we see the implemented GA does in fact converge using Pareto optimality as a fitness assignment criterion.

Figure 3 shows $PF_{current}$ generational distances. It's seen that at some generations, the currently known Pareto front appears to be close to or part of the true front, but at other generations is farther away from it. This clearly indicates the contextual nature of Pareto-based fitness assignment. Figure 4 shows the generational distance of the secondary population, indicating that as "better" Pareto optimal solutions are found, "worse" ones are discarded and PF_{known} moves ever closer to PF_{true}.

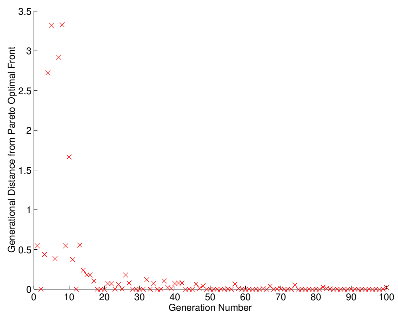

Figure 3 $PF_{current}$**'s Generational Distance (Example I).**

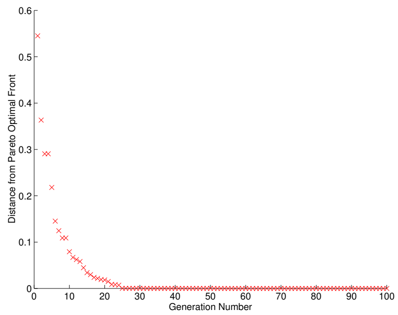

Figure 4 PF_{known}**'s Generational Distance (Example I).**

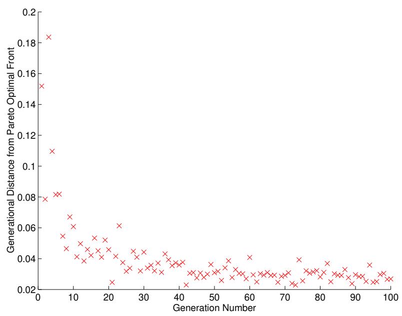

Figure 5 $PF_{current}$**'s Generational Distance (Example II).**

3.3 Example Problem II

Using the same methodology, we also applied the above GA to another bi-objective problem. Fonseca discussed this MOP (Fonseca and Fleming 1995):

$$\begin{aligned} &\text{minimize } f(x,y) = (f_1(x,y), f_2(x,y)), \text{ where} \\ &\quad f_1(x,y) \quad = 1 - \exp(-(x-1)^2 - (y+1)^2)\,, \\ &\quad f_2(x,y) \quad = 1 - \exp(-(x+1)^2 - (y-1)^2)\,. \end{aligned} \tag{12}$$

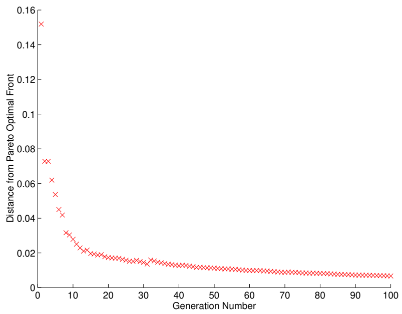

Figure 6 PF_{known}**'s Generational Distance (Example II).**

Two changes were made to the GA parameters mentioned above. First, search space bounds were set to [-10, 10]. Secondly, because this MOP involves two independent variables we used *MATLAB* to derive an approximation to the equation describing PF_{true}; the RMS error between the actual and computed solutions was 0.0224. We used this equation in determining the distance between any member of $PF_{current}$ or PF_{known} and the nearest point on the curve representing PF_{true}. Figurers 5 and 6 show the resulting data. Again, the implemented GA converges to PF_{true} using Pareto optimality as a fitness assignment criterion.

3.4 "Real World" Issues

We must consider the real world before discussing experimental results. The major issue concerns our definition of the Pareto front in the above problems. It is only because of the problems' unconstrained and bi-objective nature that we are able to determine expressions for PF_{true}. As Horn also notes (Horn 1997), in real problems we have no way of knowing when to terminate search. Unlike our example, there is no general method to determine PF_{known}'s distance from PF_{true}.

In the first example (Equation 3) we were able to derive an exact analytical expression for PF_{true} because both objective functions are functions of one variable. In the second case we deterministically computed the objective functions' values at equidistant values of x and y over a given range, thus determining which values gave rise to PF_{true}. Using those values, a *MATLAB* polynomial curve-fitting function then gave us an approximation of $f(f_1)$ for our distance computations.

We realize EAs are "overkill" for finding solutions to these particular problems. These MOPs were selected for their simplicity and ease of implementation for our experiments. We do plan to extend this experiment to MOPs of three or more functions (which are functions of both single and multiple variables), and apply results to real-world problems.

Note also that EA implementations discretize continuous models because of computational limitations. Thus, if one is searching a discrete space, PF_{true} of this space is most likely only an approximation to the continuous PF_{true}, to a particular level of resolution.

3.5 Experimental Results

Our experimental task was to determine if a specific EA exhibits convergence to PF_{true}. To do so we require metrics independent of initially measured values. Bäck defines a parameter used in assessing EA convergence velocity called a *Progress Measure* (Bäck 1996), which quantifies *relative* rather than *absolute* convergence improvement by:

$$P \triangleq \ln \sqrt{\frac{f_{max}(0)}{f_{max}(T)}}, \tag{13}$$

where $f_{max}(i)$is the best objective function value in the parent population at generation i.

We modify this definition to the following:

$$RP \triangleq \ln \sqrt{\frac{G_1}{G_T}}, \tag{14}$$

where G_1 is the generational distance at generation 1, and G_T the distance at generation T. We also select G_T as a metric showing the EA's effectiveness in converging to the Pareto front.

Initial experimental results indicate convergence occurs as conjectured, as shown in Table 1. Each reported result was derived from 10 GA executions. Niching and sharing weren't explicitly considered in these runs; we wished to see only if PF_{known} converged to PF_{true}, not "how much" of the front it covered. Each result is labeled in the form `DEB(X)` or `FNF(X)`, where `DEB` refers to (Equation 3) and `FNF` to (Equation 12). The `X` designator is interpreted as follows: `A` had a recombination probability of 0.7 and a mutation probability of 0.7, and `B`'s recombination was 0.7 and mutation was 0.35. The mean and variance are reported for each metric.

Table 1 Experimental Results.

Function	RP (Mean)	RP (Variance)	G_T (Mean)	G_T (Variance)
DEB(A)	11.0071	6.8985	1.0346e-04	1.8771e-04
DEB(B)	14.0125	11.0616	4.9123e-05	1.5534e-04
DEB(ALL)	12.5098	9.1038	7.6292e-05	1.6999e-04
FNF(A)	1.4247	0.1999	0.0056	0.0013
FNF(B)	1.3617	0.2565	0.0065	0.0022
FNF(ALL)	1.3932	0.2262	0.0061	0.0018

3.6 Experimental Analysis

RP (Equation 14) measures the relative convergence of the implemented GAs. These results cannot be generalized at this time because supporting data and analysis are lacking. Addressing this shortfall is part of of our future research. However, for both examples, after 100 generations, G_T indicates the GA was quite effective in converging PF_{known} to PF_{true}.. That is, by looking at Figures 4 and 6, we see the trend for G_T values moving towards a zero error (i.e., they are within some ϵ of PF_{true}.).

4 EA Convergence

Theorem 2: An EA using Pareto-based ranking and a monotonic selection algorithm converges to the global optimum of an MOP (a Pareto front $\vec{\mathcal{F}}^*$, composed of at most an uncountably infinite number of vectors $v_1^*, v_2^*, \ldots, v_{\aleph_1}^*$) with probability one, i.e.,

$$\mathcal{P}\{\lim_{t \longrightarrow \infty} \vec{\mathcal{F}}^* \in P(t)\} = 1 \, ,$$

where $P(t) = P_{current}$. □

Proof: Bäck proves (Bäck 1996, pg. 129) that an EA converges with probability one if it fulfills the following conditions:

$$\forall \vec{\mathcal{F}},\ \vec{\mathcal{F}}' \in I,\ \vec{\mathcal{F}}' \text{ is reachable from } \vec{\mathcal{F}} \text{ by means of mutation and recombination; and} \tag{15}$$

$$\begin{aligned} &\text{The population sequence } P(0), P(1), \ldots \text{ is monotone} \\ &\text{i.e., } \forall t:\ \min\{\Phi(\vec{\mathcal{F}}(t+1) \mid (\vec{\mathcal{F}}(t+1) \in P(t+1)\} \\ &\qquad \le \min\{\Phi(\vec{\mathcal{F}}(t) \mid (\vec{\mathcal{F}}(t) \in P(t)\} \end{aligned} \tag{16}$$

We assume an EA with infinite precision[4], a minimization MOP, an uncountably infinite population size, and appropriate mutation and recombination operators allowing every point in the search space to be visited. Thus, $\forall \vec{\mathcal{F}}$ and $\vec{\mathcal{F}}' \in I$, $\vec{\mathcal{F}}'$ is reachable from $\vec{\mathcal{F}}$. This satisfies (15), and it remains to prove that the EA's population sequence is monotone. This situation occurs when the EA used is *admissible* (both fitness function and selection algorithm are monotonic). We note here that the fitness function is *not* a combination of the objective functions (see Section 3.2).

Grefenstette defines a monotonic fitness function as one which does not reverse the sense of any pairwise solution ranking (Grefenstette 1997). When using Pareto-based fitness assignment, any given pair of Pareto optimal solutions receive identical fitness values; they also receive better fitness than dominated solutions. Any fitness function assigning fitness in this way is monotonic.

A monotonic selection algorithm also respects the fitness of pairwise solutions as regarding the expected number of offspring. Thus, we see that Equation 16 is also satisfied, since any selection algorithm selecting individuals based on fitness is monotonic. All Pareto optimal solutions present in generation $P(t)$ have the best fitness and are selected for inclusion in generation $P(t+1)$. Pareto optimal solutions whose associated vectors are non-dominated in generation $P(t+1)$ receive higher fitness than non-Pareto solutions. However, using Pareto-based ranking ensures either case satisfies Equation 16. Thus, an EA using Pareto optimality-based ranking and a monotonic selection algorithm converges to the global optimum of an MOP (i.e., PF_{true}) with probability one. □

[4]This assumption is based on our use of a real-valued EA; if the optimal solutions were rationals we could use a finite representation.

4.1 Other Convergence Proofs

Other research also addresses the desired EA convergence. Rudolph's (Rudolph 1998*a*) Corollary2 guarantees that given a countably infinite EA population and an MOP, at least one decision variable (x_k) sequence exists such that $f(x_k)$ converges in the mean to the PF_{true}, although it appears his nomenclature is inconsistent with accepted definitions.

We note his variation kernel (i.e., transition probability function) is equivalent to our reachability condition (appropriate mutation and recombination operators allowing every point in the search space to be visited). Also, his assumed elitist preservation strategy is more restrictive than our monotonic selection algorithm. Finally, he refers to at least one sequence leading to an associated point on PF_{true}, as compared to this work which indicates that through Pareto ranking *all* decision variable sequences lead towards P_{true}; likewise, these variables' phenotypical expressions lead towards PF_{true}.

Rudolph (Rudolph 1998*b*) also independently proved that a specific (1+1) EA converges with probability one to a member of the Pareto optimal set P_{true} of a specific MOP. His distance function is in the genotype domain, as compared to ours and his previous work, which is phenotypically based. Rudolph shows the desired convergence for a multiobjective (1+1)-ES, with objective functions specified by Equation 4. The evolutionary operators in his model are not able to search the entire space (in a probabilistic sense), since a step size restriction is placed upon the probabilistic mutation operator. Thus, convergence only occurs when the ES's step size is proportional to the distance to the Pareto set as shown in the elaborate proof. However, this distance is obviously unknown in problems of high complexity, which is typical of most real-world problems.

Rudolph's theorems are for a specific EA *and* MOP instantiation with constrained evolutionary operators, ours requires a less-specific EA. Both theorems show that what we seek is possible; given EAs do converge to an optimal *set*, although Rudolph defines a genotypic optimal set, and we define a phenotypic set. Using phenotypical information is more appropriate, as a decision maker's costs and profits are more accurately reflected in attribute space.

What is more important, though, is the rate at which the EAs converge to PF_{true}, and whether these nondominated vectors (PF_{known}) are uniformly distributed across PF_{true} as $t \rightarrow \infty$. These are subjects of our current research.

5 Conclusions

We have shown an EA statistically converges to a given Pareto front and presented a theorem supporting the experimental evidence. We also introduced a consistent notation not reflected in the current literature, ensuring various Pareto concepts are clearly understood. We now plan to investigate how to make the EA more efficient, i.e., how to make it converge more quickly to, and uniformly across, PF_{true}. Insight into examples of the types presented here may help in appropriate EA parameter selection. We plan to evaluate more complex multiobjective problems from this convergence perspective and apply insights to real-world MOP solving.

6 Acknowledgments

We thank Maj (Dr) Thomas Reid, and the anonymous reviewers from an earlier draft.

References

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

Bäck, Thomas, Fogel, David and Michalewicz, Zbigniew, Eds.) (1997). *Handbook of Evolutionary Computation*. Vol. 1. IOP Publishing Ltd. and Oxford University Press.

Ben-Tal, Aharon (1980). Characterization of pareto and lexicographic optimal solutions. In: *Multiple Criteria Decision Making Theory and Application* (G. Fandel and T. Gal, Eds.). Vol. 177 of *Lecture Notes in Economics and Mathematical Systems*. pp. 1–11. Springer-Verlag. Berlin.

Fonseca, Carlos M. and Peter J. Fleming (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation* **3**(1), 1–16.

Fonseca, Carlos Manuel Mira (1995). Multiobjective Genetic Algorithms with Application to Control Engineering Problems. PhD thesis. The University of Sheffield. Sheffield, UK.

Grefenstette, John (1997). *Handbook of Evolutionary Computation*. Chap. Rank-Based Selection, pp. C2.4:1 – C2.4:6. Vol. 1 of Bäck *et al.* (1997).

Horn, Jeffrey (1997). *Handbook of Evolutionary Computation*. Chap. Multicriterion Decision Making, pp. F1.9:1 – F1.9:15. Vol. 1 of Bäck *et al.* (1997).

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

Pohlheim, Hartmut (1998). Genetic and evolutionary algorithm toolbox for use with matlab. Technical report. Technical University Ilmenau.

Rudolph, Günter (1998*a*). Evolutionary search for minimal elements in partially ordered finite sets. In: *Proceedings of the Seventh Annual Conference on Evolutionary Programming*. Springer. Berlin.

Rudolph, Günter (1998*b*). On a multi-objective evolutionary algorithm and its convergence to the pareto set. In: *Proceedings of the 1998 IEEE Conference on Evolutionary Computation*. IEEE.

Srinivas, N. and Kalyanmoy Deb (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation* **2**(3), 221–248.