

Accelerated Evolutionary Algorithm with Second-order Enhancement

Joel Hewlett, Bogdan M. Wilamowski
Electrical and Computer Engineering
Auburn University
Auburn, AL 36849 United States
hewlejd@auburn.edu, wilam@ieee.org

Okyay Kaynak
Electrical and Electronic Engineering
Bogazici University
Istanbul, Turkey
okyay.kaynak@boun.edu.tr

Abstract—The paper describes a multiobjective optimization method which combines a second order algorithm with an enhanced evolutionary search in order to obtain a set of points which lie on the Pareto-optimal front. The second order portion of the method makes use of a “quasi-Jacobian” modification of the Levenberg-Marquardt algorithm, while the evolutionary portion involves a population based search which follows a Pareto defined “gradient.” Both methods are described in detail, and experimental results are included.

I. INTRODUCTION

Genetic algorithms (GA's) currently constitute the prevailing paradigm in the field of multiobjective optimization [1][2][3][5], however, despite their relative power, genetic algorithms are known to be significantly slower than gradient based methods. For this reason, single objective problems are most often solved using gradient methods, while GA's are used for only the most complex of surfaces. Unfortunately, this trend does not carry over to the multiobjective realm. The added complexity of the multiobjective problem (MOP) greatly reduces the effectiveness of the gradient approach. Furthermore, MOP's introduce the concept of Pareto efficiency which allows for, and in fact requires, an entire set of feasible solutions to be found, a task for which the gradient methods are simply not equipped. In this paper, a method is introduced by which the evolutionary process may be accelerated when used in tandem with a second order approach. Though this method has been shown to be effective for single-objective problems [7][9], it remains largely unexplored for multiobjective applications (a similar approach has been explored in [6], however the approach presented here differs in a number of ways including the methods of evolutionary computation and second order approximation. The interested reader is encouraged to explore both methods since each one offers a unique set of tools.)

The paper is arranged as follows: Section two offers an overview of basic concepts and terminology related to multiobjective optimization, and is intended primarily for the uninitiated reader. Those who are already familiar with the subject may skip this section, however, the notation presented here will be used throughout the remainder of the paper.

Section three discusses the specifics of the second order method, which includes a derivation of the “quasi-Jacobian.”

The details of the pseudo-evolutionary process are presented in section four, followed by the experimental results and conclusion.

II. MULTIOBJECTIVE OVERVIEW

The fundamental difference between single-objective and multi-objective optimization problems is the concept of Pareto efficiency. While Single-objective problems generally give rise to a single most efficient solution, this is not the case for MOP's. Instead, MOP's generally give rise to an entire set of solutions based on the Pareto efficiency. For this reason it is important to offer a formal definition of this concept, and to establish some useful notation.

A. Definitions and Notation

Definition 1 Consider a multiobjective minimization problem with m decision variables and n objectives:

$$\mathbf{y} = f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x})) \quad (1)$$

Where $\mathbf{x} = (x_1, \dots, x_m) \in X$ is a decision vector in the parameter space X , and $\mathbf{y} = (y_1, \dots, y_n) \in Y$ is an objective vector in the objective space Y . A decision vector $\mathbf{a} \in X$ is said to dominate a decision vector $\mathbf{b} \in X$ (also written $\mathbf{a} \prec \mathbf{b}$) if and only if

$$\begin{aligned} \forall i \in \{1, \dots, n\} : f_i(\mathbf{a}) \leq f_i(\mathbf{b}) & \quad \wedge \\ \exists j \in \{1, \dots, n\} : f_j(\mathbf{a}) < f_j(\mathbf{b}) & \end{aligned} \quad (2)$$

Definition 2 Let $\mathbf{a} \in X$ be an arbitrary decision vector.

1. The decision vector \mathbf{a} is said to be nondominated regarding a set $X' \subseteq X$ if and only if there is no vector in X' which dominates \mathbf{a} ; formally

$$\nexists \mathbf{a}' \in X' : \mathbf{a}' \prec \mathbf{a} \quad (3)$$

2. The decision vector \mathbf{a} is Pareto-optimal if and only if \mathbf{a} is nondominated regarding X .

Definition 3 Consider the set of decision vectors $X' \subseteq X$.

1. The set X' is denoted as a local Pareto-optimal set if and only if

$$\forall \mathbf{a}' \in X': \exists \mathbf{a} \in X : \mathbf{a} \prec \mathbf{a}' \quad \wedge \quad \|\mathbf{a} - \mathbf{a}'\| < \varepsilon \wedge \|f(\mathbf{a}) - f(\mathbf{a}')\| < \delta \quad (4)$$

Where $\|\cdot\|$ is a corresponding distance metric and $\varepsilon > 0, \delta > 0$.

2. The set X' is called a global Pareto-optimal set if and only if

$$\forall \mathbf{a}' \in X': \exists \mathbf{a} \in X : \mathbf{a} \prec \mathbf{a}' \quad (5)$$

It is important to note that a globally Pareto-optimal set does not necessarily contain all Pareto-optimal solutions. The set of all globally Pareto-optimal solutions is commonly referred to as the *Pareto-optimal front*.

B. A Graphical Representation of Pareto Efficiency

Each of the definitions above may be understood graphically via the objective space, and in fact, this technique is quite useful in the comparison of multiobjective algorithms. Using figure 1, the definitions from the previous section can be restated to offer a graphical representation of Pareto efficiency.

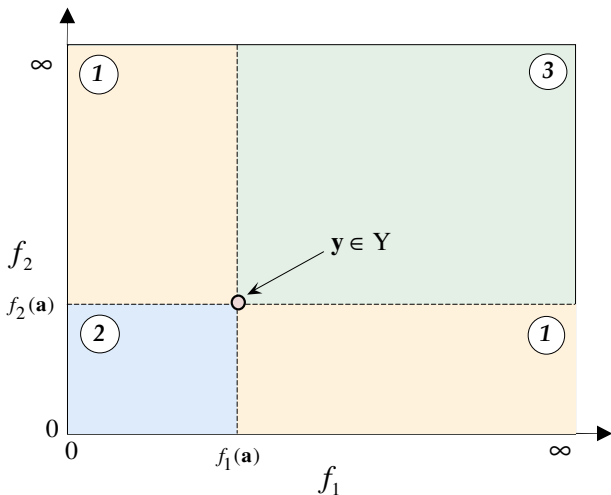


Fig. 1 Geometric view of Pareto-efficiency

Definition 1 For the arbitrarily chosen point \mathbf{y} in figure 1, any point which lies in the green shaded region is said to be

dominated by \mathbf{y} , whereas any point which lies in the blue shaded region is said to dominate \mathbf{y} .

Definition 2 For a point $\mathbf{a} \in X'$ which corresponds to the objective point \mathbf{y} , if there is no point $\mathbf{a}' \in X'$ such that \mathbf{y}' lies in the blue shaded region, \mathbf{a} is said to be non-dominated regarding X' .

Definition 3 For every point $\mathbf{a} \in X'$ with objective \mathbf{y} , there is no point $\mathbf{a}' \notin X'$ in the neighborhood of \mathbf{a} for which \mathbf{y}' lies in the blue shaded region, then X' represents a local Pareto optimal set. For the case in which (9) holds for any arbitrary ε and δ , X' is said to represent a globally Pareto optimal set.

A 2-dimensional example of an objective space is shown in figure 2. The blue region represents the set of all objective vectors $(f_1(\mathbf{x}), f_2(\mathbf{x}))$ such that $\mathbf{x} \in X$. It should be evident that each of the three bold lines on the border of the blue region represents a global Pareto-optimal set. It should also be evident that together, these three segments represent the Pareto-optimal front. There are, however, some less obvious observations which should also be noted. Firstly, the global minimum of each objective function will lie on the Pareto-frontier. (see \mathbf{a}, \mathbf{b}). Also, The minimum of one objective with respect to the Pareto-front will correspond to the maximum of the other. (see \mathbf{a}, \mathbf{b})

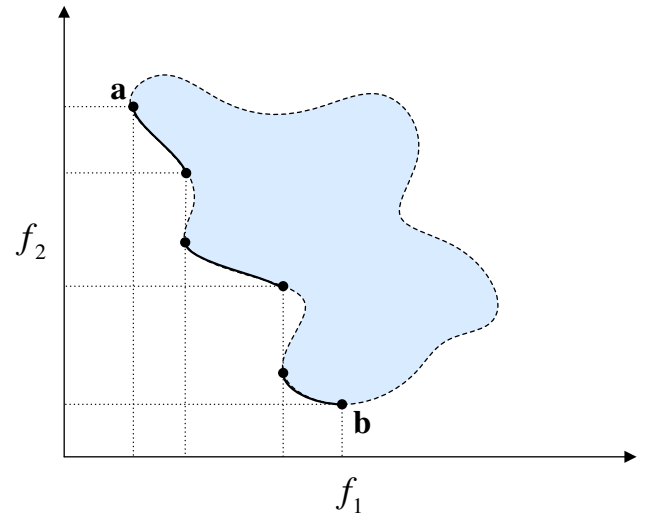


Fig. 2 Example of objective space in two dimensions

The importance of these observations will become evident in the following section.

III. DESCRIPTION OF SECOND ORDER METHOD

It will be shown in section 4 that the correct placement of the

initial population (*i.e.* near the Pareto-optimal front) greatly enhances the performance of the evolutionary method. Thus, the first step in the proposed method involves searching for a set of suitable starting points from which the evolutionary process may proceed. This is where the two observations made at the end of the last section become especially useful. Taken together, the two observations imply that minimization of each individual objective will result in decision vectors which lie at opposite ends of the Pareto optimal front. In other words, *a set of suitable starting points may be found by breaking the multi-objective problem up into a set of single objective problems.* In this way, suitable starting points may be found using the gradient methods mentioned earlier, which will greatly accelerate the process. For this purpose, a powerful and relatively fast second order method known as the Levenberg-Marquardt algorithm (LMA) was selected. LMA interpolates between the Gauss-Newton algorithm and the method of gradient descent. Although the Gauss-Newton Algorithm generally converges faster than LMA when applied to well-behaved functions, LMA is more robust, allowing it to solve more difficult problems. LMA employs the following update rule:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\mathbf{J}_k^T \mathbf{J}_k + \mu \mathbf{I})^{-1} \mathbf{J}_k^T \mathbf{f}$$

where μ is an adjustable damping factor, \mathbf{J} is the Jacobian matrix, and \mathbf{f} is a vector containing the values of the functions to be minimized. The key to the computation of LMA lies in the Jacobian matrix. For this, the proposed method makes use of a numerical approximation of the Jacobian matrix, which is described below.

A. Quasi-Jacobian

Given a function f , a linear approximation of f at the point $\mathbf{x} = (x_1, x_2, \dots, x_m)$ may be found using the first two terms of the Taylor series expansion.

$$\begin{aligned} f(\mathbf{x}) &\approx f(\mathbf{x}_0) + \frac{\partial f}{\partial x_1} \cdot \Delta x_1 + \frac{\partial f}{\partial x_2} \cdot \Delta x_2 + \dots + \frac{\partial f}{\partial x_m} \cdot \Delta x_m \\ &= f(\mathbf{x}_0) + \nabla_x f(\mathbf{x}) \cdot \Delta \mathbf{x} \end{aligned} \quad (9)$$

Solving this equation for the gradient yields

$$\nabla_x f(\mathbf{x}) \approx \frac{f(\mathbf{x}) - f(\mathbf{x}_0)}{\Delta \mathbf{x}} = \frac{f(\mathbf{x}) - f(\mathbf{x}_0)}{\mathbf{x} - \mathbf{x}_0} \quad (10)$$

The goal here is that (10) might be used to obtain a numerical approximation of f at \mathbf{x} , but for $\mathbf{x} = (x_1, x_2, \dots, x_m)$, solving for $\nabla_x f$ results in a system of one equation with m unknowns. This means that for any case where $m > 1$, (10) is underdetermined and $\nabla_x f$ cannot be found, however if (10) is applied to not just one, but to a whole set of decision vectors, any number of equations can be found. More specifically, for an m -dimensional function f , finding the

gradient of f at some point \mathbf{x} requires that the value of f be known at a minimum of m linearly independent points. This may be represented by the following matrix equation.

$$\nabla_x f(\mathbf{x})_{1 \times m} \approx \left(\begin{bmatrix} \Delta x_{11} & \dots & \Delta x_{m1} \\ \vdots & \ddots & \vdots \\ \Delta x_{1m} & \dots & \Delta x_{mm} \end{bmatrix}_{m \times m}^{-1} \begin{bmatrix} \Delta f_1 \\ \vdots \\ \Delta f_m \end{bmatrix}_{m \times 1} \right)^T \quad (11)$$

This method may then be extended to the multiobjective case by expressing the Jacobian matrix in terms of (11). For a set of n functions, the Jacobian matrix is expressed

$$\mathbf{J}_x(f_1, \dots, f_n) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_m} \end{bmatrix}_{n \times m} = \begin{bmatrix} \nabla_x f_1(\mathbf{x})_{1 \times m} \\ \vdots \\ \nabla_x f_n(\mathbf{x})_{1 \times m} \end{bmatrix}_{n \times m} \quad (12)$$

As is evident from (12), the Jacobian is simply a matrix of row vectors which correspond to the gradients of each of the functions, and can therefore be rewritten using (11), which yields the following expression,

$$\mathbf{J}_x \approx \left(\begin{bmatrix} \Delta x_{11} & \dots & \Delta x_{m1} \\ \vdots & \ddots & \vdots \\ \Delta x_{1m} & \dots & \Delta x_{mm} \end{bmatrix}_{m \times m}^{-1} \begin{bmatrix} \Delta f_{11} & \dots & \Delta f_{n1} \\ \vdots & \ddots & \vdots \\ \Delta f_{1m} & \dots & \Delta f_{nm} \end{bmatrix}_{m \times n} \right)^T \quad (13)$$

It is important to note that for a population whose size exceeds the number of dimensions of f , *i.e.* $\text{Rank}(\Delta \mathbf{X}) > m$, (11) and (13) may still be applied using the pseudo-inverse of $\Delta \mathbf{X}$ [4]. Note also that the accuracy of the approximation in (10), and therefore (11) and (13), depends entirely on the size of $\Delta \mathbf{x}$. This would imply that the accuracy of the approximation is inversely related to the geometric size of the population.

IV. ENHANCED-EVOLUTIONARY METHOD

The second portion of the proposed method involves the implementation of an enhanced-evolutionary process. This process is evolutionary in the sense that it uses the combination of desirable attributes within a given population to produce an ever improving set of successive populations. What sets the method apart from more traditional approaches is the way in which the successive populations are generated. Assuming the initial population is centered about a point which lies on the Pareto-front, an issue which was addressed in the previous section, successive populations may be generated by determining the Pareto-efficiency of the existing population and simply shifting the center point accordingly. Furthermore, if the initial starting point is located at the minimum of the first objective, then the non-dominated member of the current population for which the second objective is lowest will lie on or near the Pareto-front in the direction of the opposite end. Repeating this process will

result in a series of partially overlapping populations which traverse the solution set using a sort of Pareto-defined gradient.

A. A Pareto Based Gradient

Given an MOP with n objectives, for two points $\mathbf{a}, \mathbf{b} \in X$, the Pareto-efficiency of \mathbf{a} with respect to \mathbf{b} may be determined using the following rule,

$$S = \frac{1}{n} \cdot \left[\sum_{i=1}^n \text{sign}(f_i(\mathbf{a}) - f_i(\mathbf{b})) \right] \quad (14)$$

$$P = \prod_{i=1}^n \text{sign}(f_i(\mathbf{a}) - f_i(\mathbf{b}))$$

If $S = 0$ then

$$S > 0 \Rightarrow \mathbf{a} < \mathbf{b}$$

$$S < 0 \Rightarrow \mathbf{a} > \mathbf{b}$$

$$S = 0 \Rightarrow \mathbf{a} = \mathbf{b}$$

If $S \neq 0$ then

$$S = -1 \Rightarrow \mathbf{a} < \mathbf{b}$$

$$S = 1 \Rightarrow \mathbf{a} > \mathbf{b}$$

$$|S| < 1 \Rightarrow \mathbf{a} = \mathbf{b}$$

For a given population P and solution set X , (14) may be used to determine the non-dominated subset of P

$$P' = \{\mathbf{a} \in P : \forall \mathbf{b} \in P \cup X, \mathbf{a} \preceq \mathbf{b}\} \subseteq P$$

Finally, given a population of decision vectors P , centered about \mathbf{c} , the “Pareto-gradient” for P with respect to objective f is defined in the following manner:

$$G = \mathbf{c} - \mathbf{a}_{\min}$$

where

$$\mathbf{a}_{\min} = \mathbf{a} \in P' : (\forall \mathbf{a}' \in P'), f(\mathbf{a}) < f(\mathbf{a}')$$

V. EXPERIMENTAL RESULTS

The proposed method was tested for a number of different MOP's in order to determine performance. two of the tested cases are presented below in order of increasing complexity. An overview is offered for each problem, followed by the corresponding test results. Each Problem was also tested using the non-dominated sorting genetic algorithm 2 (NSGA-II) [8], a well known evolutionary method which has seen widespread use. The second test case offers a unique visual example of the algorithm's behavior, and is therefore presented in slightly greater detail.

A. Overview of the first test case

The first test case is a simple single variable MOP which was originally presented in [5], and briefly discussed in [3]. The problem is comprised of a pair of parabola's which are slightly offset. The problem may be formally stated:

$$\text{minimize} \begin{cases} f_1(x) = x^2 \\ f_2(x) = (x-2)^2 \end{cases}$$

where $x \in [-10^5, 10^5]$

The major obstacle posed by this problem is the large set of variable bounds which may pose some problems for evolutionary approaches if the initial starting point is selected at a great distance from the desired solution set. The proposed method, on the other hand, is suited quite well for such situations.

B. Results for the first test case

For testing, a random point was chosen and the algorithm was applied. A near worst case starting point of $-9.1e5$ was generated. From here, the second order method was run for a total of 20 iterations and value of $x = 0.9455$ was selected, about which the first population was generated. A total of 5 populations were used, each having 20 points. In all, 140 points were evaluated (including calculation of the quasi-Jacobian) over the range $[-9.1e5, 2.2]$, of which 80 were optimal solutions on the interval $[0, 2]$. The solution set for NSGA-II was also found for the sake of comparison. For NSGA, 7 populations were used in order to maintain an equal number of total points, and the boundary conditions were decreased to $[-100, 100]$. Out of 140 total points generated over the range $[-80, 97]$, 20 were Pareto-optimal solutions. The final solutions for both methods are plotted in figure.

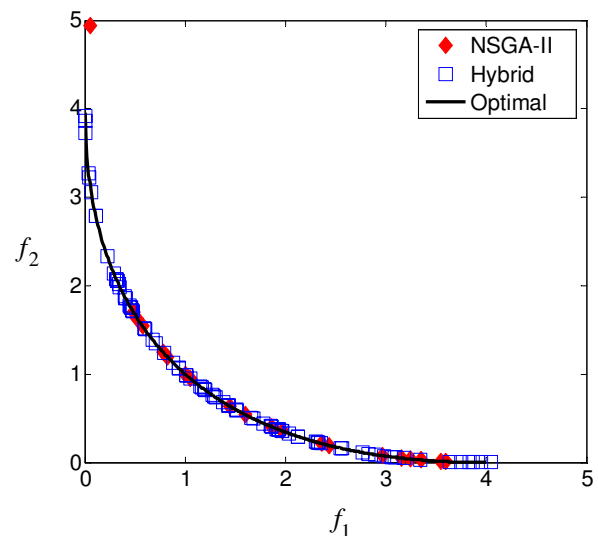


Fig. 3 Results for the first test case

The squares in figure 3 represent the members of the solution set produced by the proposed method, while the solution set found using NSGA-II is represented by the diamonds. The black curve represents the Pareto-optimal front. It is clear from figure 3 that the both algorithm performed well in terms of convergence, however in terms of diversity, the hybrid method is shown to be superior. Despite the significant difference in the sizes of the search areas, the hybrid method still produced four times as many solutions.

C. Overview of the second test case

The second MOP tested is a 2 variable problem first presented in [2], and later in [3]. Here, the problem was chosen in order to offer a visual representation of how the algorithm performs in the objective space as well as how it behaves in parametric space. The ability to plot the decision vectors in 3 dimensions will offer a great deal of insight into how the algorithm narrows its search, whereas the objective plot only gauges the algorithm's performance. Ultimately, performance is the key issue, however, understanding the algorithm's behavior will allow greater appreciation of its effectiveness.

The first problem seeks to minimize the two following objectives:

Let

$$A_1 = 0.5 \sin(1) - 2 \cos(1) + \sin(2) - 1.5 \cos(2)$$

$$A_2 = 1.5 \sin(1) - \cos(1) + 2 \sin(2) - 0.5 \cos(2)$$

$$B_1 = 0.5 \sin(x_1) - 2 \cos(x_1) + \sin(x_2) - 1.5 \cos(x_2)$$

$$B_2 = 1.5 \sin(x_1) - \cos(x_1) + 2 \sin(x_2) - 0.5 \cos(x_2)$$

$$1. f_1(\mathbf{x}) = 1 + (A_1 + B_1)^2 + (A_2 + B_2)^2$$

$$2. f_2(\mathbf{x}) = (x_1 + 3)^2 + (x_2 + 1)^2$$

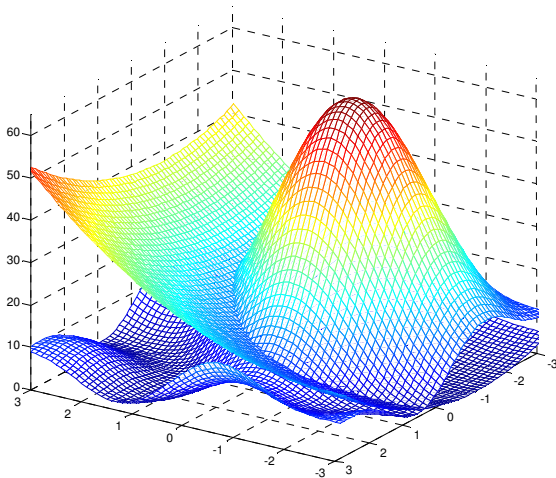


Fig. 4 Sum of the objective functions

In order to gain a basic understanding of the general nature of the parameter space, f_1 and f_2 are shown in Fig 4. It is clear from figure 4 that while f_2 is well behaved and has only one minimum, f_1 is significantly more complex, having multiple local minima. This behavior will ultimately result in a Pareto optimal set which is disconnected in both the parameter space and in the objective space. To see this more clearly, a parameter space representation of the Pareto-optimal front is shown in Figure 5.

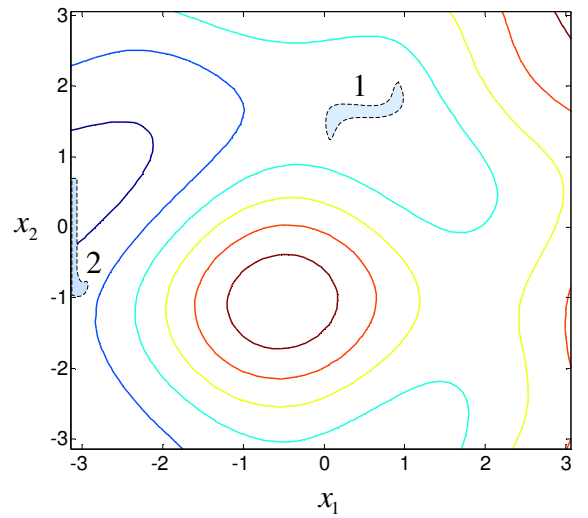


Fig. 5 Pareto-front as viewed from the parameter space

The blue shaded regions in figure 5 represent the Pareto-optimal front. Region 1 contains the minimum of f_1 , and includes all surrounding decision vectors for which the gradients of f_1 and f_2 are opposite. Region 2 contains a global and local minimum of f_2 and f_1 respectively, which are connected by the behavior of the boundary region.

D. Results for the second test case

For the first test case, starting points were sought using f_1 , f_2 , and $(f_1 + f_2)$. Each function was minimized using “quasi-Jacobian” modification of the Levenberg-Marquardt algorithm described in section 2, in order to find the corresponding starting points. The pseudo-evolutionary method described in section 3 was then applied for each of the starting points, and a Pareto-optimal solution set was formed using the efficiency test from (14). 50 populations of 25 decision vectors were generated for each of the three starting points. The results are presented in figures 6 and 7.

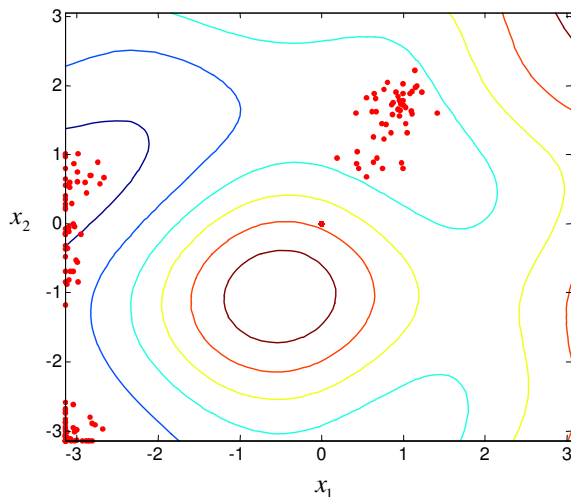


Fig. 6 Population centers in parametric space

Figure 6 shows how the algorithm directs its search of the parameter space. The center of each population of decision vectors is represented by a red dot, while the objective vectors themselves have been omitted for the sake of clarity. From figures 5 and 6, the efficiency of the search method should be evident by comparison. By narrowing the search space using the second order method, the proposed method has greatly increased the performance of the evolutionary process. It is clear from figure 6 that the large majority of the populations were generated in very close proximity to the Pareto-front.

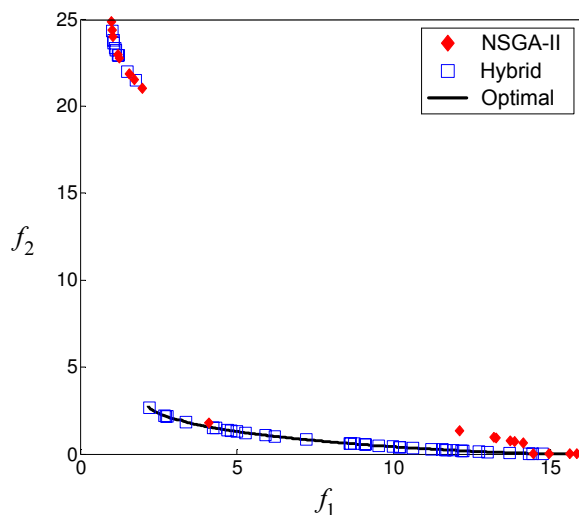


Fig. 7 Pareto-optimal set plotted in objective space

A visual representation of the algorithm's performance is offered in Figure 7. The boxes represent members of the solution set obtained by the modified algorithm, while the diamonds show those solutions found using NSGA-II (again, for the sake of clarity, only a sample of the solution set is shown here). The Pareto-optimal front is represented by the black curve. It is evident from the figure that the proposed method has outperformed NSGA-II in terms of both convergence and diversity. The algorithm has converged to a

solution set which is nearly indistinguishable from the Pareto-front, with a relatively even distribution. Out of a total of 2000 points generated, 324 were selected as Pareto optimal solutions using the modified method, while NSGA-II only yields a solution set of 100 points with a lesser degree of convergence.

VI. CONCLUSION

A new approach for multiobjective optimization was proposed. The proposed method combines a second order gradient search with an evolutionary algorithm in order to increase the rate of convergence, while also improving the diversity of the solution set. The resulting algorithm was tested using a set of problems with varying degrees of difficulty. The results of these tests were presented, and the proposed method was shown to outperform its unmodified counterpart in each and every case. The enhanced algorithm showed significant improvement in both solution set diversity and rate of convergence, which confirms the authors' initial hypothesis.

VII. REFERENCES

- [1] Zitzler, E.; Deb, K.; Thiele, L.; "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, vol.8, no.2, pp. 173-195, Summer 2000
- [2] C. Poloni, "Hybrid GA for Multiobjective Aerodynamic Shape Optimization," *Genetic Algorithms in Engineering and Computer Science*, G. Winter, Ed. New York: Wiley, 1995, pp.397-416.
- [3] Chan, T.M.; Man, K.F.; Kwong, S.; Tang, K.S.; "A Jumping Gene Paradigm for Evolutionary Multiobjective Optimization," *IEEE Transaction on Evolutionary Computation*, Accepted for future publication
- [4] B.M. Wilamowski and M. Manic, "Random Weights Search in Compressed Neural Networks Using Overdetermined Pseudoinverse," *ISIE 03 IEEE International Symposium on Industrial Electronics*, vol. 2, pp. 678 – 683
- [5] J.D. Schaffer, "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms," *Proc. 1st International Conference on Genetic Algorithms*, J.J. Grefenstette, Ed., 1987, pp. 93-100.
- [6] X. Hu; Z. Huang; Z. Wang "Hybridization of the multi-objective evolutionary algorithms and the gradient-based algorithms," *Congress on Evolutionary Computation - CEC'03*, Volume 2, 8-12 Dec. 2003 pp. 870 - 877 Vol.2
- [7] M. Bundzel and P. Sincak, "Combining Gradient and Evolutionary Approaches to the Artificial Neural Networks Training According to Principles of Support Vector Machines," *IEEE International Joint Conference on Neural Networks - IJCNN'06*, 16-21 July 2006, pp. 2068 – 2074
- [8] Deb, K.; Agrawal, S.; Pratap, A.; Meyarivan, T.; "A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multiobjective Optimization: NSGA-II," *IEEE Transaction on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, Apr. 2002
- [9] J. Hewlett, B. Wilamowski, and G. Dundar "Merge of Evolutionary Computations with Gradient Based Method for Optimization Problems" *ISIE 07 IEEE International Symposium on Industrial Electronics*, Vigo, Spain, June 4-7, 2007.