Local Search for Multiobjective Function Optimization: Pareto Descent Method

Ken Harada
Tokyo Institute of Technology
4259 Nagatsuta-cho, Midori-ku
Yokohama-shi, Kanagawa, Japan
ken@fe.dis.titech.ac.jp

Jun Sakuma
Tokyo Institute of Technology
4259 Nagatsuta-cho, Midori-ku
Yokohama-shi, Kanagawa, Japan
jun@fe.dis.titech.ac.jp

Shigenobu Kobayashi
Tokyo Institute of Technology
4259 Nagatsuta-cho, Midori-ku
Yokohama-shi, Kanagawa, Japan
kobayasi@dis.titech.ac.jp

ABSTRACT

Genetic Algorithm (GA) is known as a potent multiobjective optimization method, and the effectiveness of hybridizing it with local search (LS) has recently been reported in the literature. However, there is a relatively small number of studies on LS methods for multiobjective function optimization. Although each of the existing LS methods has some strong points, they have respective drawbacks such as high computational cost and inefficiency in improving objective functions. Hence, a more effective and efficient LS method is being sought, which can be used to enhance the performance of the hybridization.

Defining Pareto descent directions as descent directions to which no other descent directions are superior in improving all objective functions, this paper proposes a new LS method, Pareto Descent Method (PDM), which finds Pareto descent directions and moves solutions in such directions thereby improving all objective functions simultaneously. In the case part or all of them are infeasible, it finds feasible Pareto descent directions or descent directions as appropriate. PDM finds these directions by solving linear programming problems, which is computationally inexpensive. Experiments have shown PDM's superiority over existing methods.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*Gradient methods*; I.2 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms, Performance, Experimentation, Theory

Keywords

Multi-objective optimization, Local search, Constraint handling

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.

1. INTRODUCTION

The problem of simultaneously optimizing multiple conflicting objective functions is called multiobjective optimization and is encountered in many applications such as design, control, and systems modeling. If the variables of the objective functions are real-valued, it is called multiobjective function optimization, which is dealt with herein.

The objective of multiobjective optimization is to find the set of Pareto-optimal solutions, to which no other feasible solutions are superior in all objective functions. One approach to multiobjective optimization is scalarizing methods. Scalarizing methods combine objective functions, as specified by a parameter vector, to form a scalar function, which is then optimized to give the Pareto-optimal solution corresponding to that particular parameter vector. Min-max method and ε-constraint method are a few of such methods [3]. A set of Pareto-optimal solutions can be obtained by solving the scalar function optimization problems for various parameter vectors. However, solving constrained nonlinear scalar optimization problems for obtaining many Pareto-optimal solutions is computationally prohibitive.

Much attention has recently been paid to Genetic Algorithms (GA) as a promising alternative. It maintains a set of solutions and converges it progressively toward Pareto-optimal solutions with relatively small computational cost [3]. However, there is a report [10] from which it can be inferred that GA may not be suitable for obtaining solutions of high precision.

In order to avoid such a problem, GA can be hybridized with local search (LS), and the effectiveness of the hybridization has been demonstrated in the literature [6, 7, 9]. LS methods for multiobjective function optimization such as Evolution Strategies (ES) [8], Multiobjective Steepest Descent Method (MSDM) [4], Combined-Objectives Repeated Line-search (CORL) [1] have been proposed. However, they have limitations such as inefficiency (ES), high computational cost (MSDM), and inability to improve objective function for solutions on feasible region boundaries (CORL). Hence, a new LS method that resolves these problems is been sought.

This paper, assuming that objective functions are differentiable, defines Pareto descent directions as descent directions to which no other descent directions are superior in improving all objective functions and proposes a new LS method, Pareto Descent Method (PDM), which efficiently finds feasible Pareto descent directions or descent directions as appropriate in which solutions can be moved to simultaneously improve all objective functions for problems with arbitrary numbers of variables, objective functions, and constraints.

In the ensuing sections, the basics of multiobjective function optimization are reviewed and some of the existing LS methods are surveyed, followed by a description of PDM and experimental results that show PDM's superiority over the existing LS methods.

2. MULTIOBJECTIVE OPTIMIZATION AND LS METHODS

2.1 Multiobjective Optimization

2.1.1 Multiobjective Optimization Problem and Pareto-optimal Solutions

Let the dimensions of the real-valued variable space and the objective space be N and M, respectively. Denote a solution by $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)^T \in \mathbb{R}^N$, the vector of objective functions by $\boldsymbol{f} = (f_1, f_2, \ldots, f_M)^T$, the feasible region by $S \subset \mathbb{R}^N$, and the image of $\boldsymbol{x}$ in the objective space by $\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^M$. *Multiobjective function optimization problems* can be formulated as:

$$\text{Minimize } f_i(\boldsymbol{x}) \ (i = 1, 2, \ldots, M), \text{ subject to } \boldsymbol{x} \in S.$$

Its feasible region S is the region which satisfies L constraints such as:

$$g_j(\boldsymbol{x}) \geq 0 \ (j = 1, 2, \ldots, L).$$

f_i and g_j are assumed to be differentiable herein.

If the following holds for some solutions $\boldsymbol{x_1}, \boldsymbol{x_2} \in S$, $\boldsymbol{x_1}$ is said to be *superior* to $\boldsymbol{x_2}$, which is denoted by $\boldsymbol{x_1} \succ \boldsymbol{x_2}$:

$$\forall i \in \{1, \ldots, M\},\ f_i(\boldsymbol{x}_1) \leq f_i(\boldsymbol{x}_2)$$
$$\wedge \exists i \in \{1, \ldots, M\},\ f_i(\boldsymbol{x}_1) < f_i(\boldsymbol{x}_2).$$

If there is no feasible solution $\boldsymbol{x}'$ such that $\boldsymbol{x}' \succ \boldsymbol{x}$, the solution $\boldsymbol{x}$ is called a *Pareto-optimal solution*. There are often multiple Pareto-optimal solutions. If there is no solution $\boldsymbol{x}'$ such that $\boldsymbol{x}' \succ \boldsymbol{x}$ in the feasible ε-vicinity of a solution $\boldsymbol{x}$, $\boldsymbol{x}$ is called a *local Pareto-optimal solution*. Since this paper focuses on LS methods, it is henceforth assumed, for the brevity of discussion, that local Pareto-optimal solutions that are not Pareto-optimal do not exist.

2.1.2 Descent Directions

Denote by $\nabla f_i(\boldsymbol{x})$ $(i = 1, 2, \ldots, M)$ the gradients of objective functions at a solution $\boldsymbol{x}$. If the following holds for a direction $\boldsymbol{d} = (d_1, d_2, \ldots, d_N)^T \in \mathbb{R}^N$, all objective functions can be improved simultaneously by moving $\boldsymbol{x}$ in the direction $\boldsymbol{d}$:

$$\boldsymbol{d} \cdot (-\nabla f_i(\boldsymbol{x})) \geq 0 \ (i = 1, 2, \ldots, M). \quad (1)$$

Descent directions [4] for multiobjective function optimization are defined as directions that satisfy Eq. (1). There are often multiple descent directions. Note that scaling objective functions does not inherently change multiobjective optimization problems. In order to remove the influence of scaling, Eq. (1) can be rewritten as:

$$\boldsymbol{d} \cdot (-\bar{\nabla} f_i(\boldsymbol{x})) \geq 0 \ (i = 1, 2, \ldots, M), \quad (2)$$

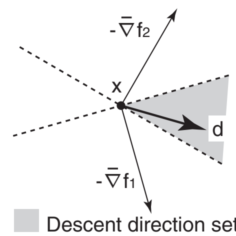


Figure 1: A descent direction: ***at a solution x of a 2-variable-2-objective problem***

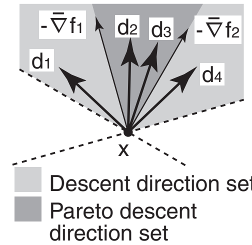


Figure 2: Pareto descent directions: ***at a solution x of a 2-variable-2-objective problem***

where $\bar{\nabla} f_i(\boldsymbol{x}) = \nabla f_i(\boldsymbol{x}) / ||\nabla f_i(\boldsymbol{x})||$. Descent directions, herein, are defined as directions that satisfy Eq. (2) which is equivalent to Eq. (1). A descent direction for a 2-variable-2-objective problem is shown in Fig. 1. Note that, since Eq. (2) is a simultaneous linear inequality, the complete set of descent directions forms a convex cone pointed at the origin in the vector space.

2.1.3 Pareto Descent Directions

Not all descent directions are similarly capable of improving all objective functions. Consider descent directions $\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3$, and $\boldsymbol{d}_4$ of Euclidean norm 1 at a solution $\boldsymbol{x}$ of a 2-variable-2-objective problem shown in Fig. 2. The following holds for $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$:

$$\boldsymbol{d}_1 \cdot (-\bar{\nabla} f_1(\boldsymbol{x})) < \boldsymbol{d}_2 \cdot (-\bar{\nabla} f_1(\boldsymbol{x})),$$
$$\boldsymbol{d}_1 \cdot (-\bar{\nabla} f_2(\boldsymbol{x})) < \boldsymbol{d}_2 \cdot (-\bar{\nabla} f_2(\boldsymbol{x})).$$

Hence, $\boldsymbol{d}_2$ improves each objective function more than $\boldsymbol{d}_1$ does. Similarly, $\boldsymbol{d}_3$ improves each objective function more than $\boldsymbol{d}_4$ does. Furthermore, there is no descent direction which can improve each objective function more than $\boldsymbol{d}_2$ or $\boldsymbol{d}_3$ does. *Pareto descent directions*[1] are defined as descent directions to which no other descent direction is superior in improving all objective functions. There are often multiple Pareto descent directions.

None of Pareto descent directions are better than one another. For example, the following holds for $\boldsymbol{d}_2$ and $\boldsymbol{d}_3$:

$$\boldsymbol{d}_2 \cdot (-\bar{\nabla} f_1(\boldsymbol{x})) > \boldsymbol{d}_3 \cdot (-\bar{\nabla} f_1(\boldsymbol{x})),$$
$$\boldsymbol{d}_2 \cdot (-\bar{\nabla} f_2(\boldsymbol{x})) < \boldsymbol{d}_3 \cdot (-\bar{\nabla} f_2(\boldsymbol{x})).$$

This means that $\boldsymbol{d}_2$ improves f_1 more than $\boldsymbol{d}_3$ does while $\boldsymbol{d}_3$ improves f_2 more than $\boldsymbol{d}_2$ does.

A descent direction $\boldsymbol{d}$ is a Pareto descent direction iff $\boldsymbol{d}$ can be expressed as a convex combination of the steepest descent directions of objective functions, i.e. there exist $\alpha_i \geq 0$ $(i = 1, 2, \ldots, M)$ such that:

$$\boldsymbol{d} = \sum_{i=1}^{M} \alpha_i (-\bar{\nabla} f_i(\boldsymbol{x})). \quad (3)$$

Since both the complete set of descent directions and all convex combinations of the steepest descent directions form convex cones, the union of the two, namely, the complete set of Pareto descent directions, also forms a convex cone.

[1]A similar concept has been proposed in [1] as non-dominated improving directions.

2.2 Local Search Methods

This section briefly reviews some of the existing LS methods and mentions their respective advantages and disadvantages.

Evolution Strategies (ES): ES samples a new solution according to the normal distribution centered at the previous solution and replaces it with the new solution if all the objective functions are improved at the new one. ES may or may not maintain the archive of the previously visited solutions. ES without the archive, which is assumed throughout this paper, can be thought of a local search method. Although the cost of generating new solutions in ES is small, its performance is sensitive to the standard deviation of the normal distribution [10], and randomly generating an improving solution becomes more and more inefficient as the dimension of the variable space increases.

Note that, if a new solution randomly sampled over the surface of the small hypersphere centered at the previous solution is superior to the previous one in all objective functions, the direction from the previous solution to the new one is a descent direction. Based on this consideration, an ES-like simple LS method that improves a solution in a descent direction found in this manner can easily be conceived, which herein is called *Random Direction Search* (*RDS*).

Weighted Steepest Descent Method (WSDM): WSDM combines objective functions as specified by some convex combination weight $\boldsymbol{w}$ to form a scalar function and optimizes it with steepest descent method to move the solution progressively toward the Pareto-optimal solution corresponding to the weight $\boldsymbol{w}$. Since the steepest descent direction of the scalar function is equivalent to the convex combination of the steepest descent directions of objective functions with the same combination weight, WSDM can be thought of as one of the methods that specify the direction in which a solution is moved based on the gradients of objective functions at the solution.

If the Pareto-optimal solutions are non-convex in the objective space, WSDM is unable to find the solutions in the non-convex part. Furthermore, WSDM is unable to move solutions on a feasible region boundary if all convex combinations of the steepest descent directions are infeasible.

Multiobjective Steepest Descent Method (MSDM): MSDM [4] defines the degree of improvement in each objective function when a solution is moved in a direction as the inner product of the direction and the steepest descent direction of respective objective function. MSDM finds the direction that maximizes the minimum degree of improvement of all objective functions by solving a quadratic programming problem and moves the solution in that direction [4]. The direction it finds is a Pareto descent direction. When a solution is on a feasible region boundary, it incorporates the boundary information into the quadratic programming problem to exclude infeasible directions.

MSDM finds only a single Pareto descent direction although there are many other such directions. In addition, when a solution is on a feasible region boundary and there exists a solution in its infeasible vicinity which further improves all objective functions, MSDM finds a direction parallel to the boundary, which may not be properly feasible. Furthermore, MSDM is computationally expensive since a quadratic programming problem has to be solved to find a single direction. Two less computationally expensive variants of MSDM are suggested in [4]. However, they do not necessarily find Pareto descent directions.

Combined Objectives Repeated Line-search (CORL): CORL utilizes the gradients of objective functions and analytically calculates the vectors that generate the convex cone of Pareto descent directions or descent directions [1]. It is computationally inexpensive.

Although CORL is guaranteed to find the complete set of Pareto descent directions on 2-objective problems, it may not find these directions on problems with more than 2 objectives, as explained in the appendix. In that case, CORL is expected to perform poorly. Furthermore, CORL may not be able to move a solution on a feasible region boundary when all Pareto descent directions are infeasible yet some descent directions are feasible.

Hence, it can be concluded that, although each of the existing methods has certain strong points, they have their respective drawbacks.

3. PARETO DESCENT METHOD

This section presents a new LS method, Pareto Descent Method (PDM), that finds, at a relatively small computational cost, a set of Pareto descent directions for solutions inside feasible regions and a set of feasible Pareto descent directions or descent directions as appropriate for solutions on feasible region boundaries and moves the solutions in these directions, thereby efficiently improving all objective functions simultaneously.

3.1 Overview

Consider the problem of finding a direction in a convex cone which is defined by a simultaneous linear inequality and pointed at the origin in the vector space, as shown in Fig. 3. Suppose that imposing a linear constraint on the convex cone gives a convex polyhedron which has the origin as one of its vertices. The vertices of the convex polyhedron can be obtained by solving the linear programming problems corresponding to the convex polyhedron. Note that the vectors from the origin to the vertices of the convex polyhedron other than the origin represent directions in the convex cone. The property of convex cone ensures that any convex combinations of these vectors are also in the convex cone.

The set of convex combination weight with which the steepest descent directions of objective functions are combined to give a Pareto descent direction forms a convex cone pointed at the origin as explained in Subsection 3.2. The set of descent directions also forms a convex cone pointed at the origin. Hence, the above-mentioned method can be used for finding both a set of Pareto descent directions and a set of descent directions.

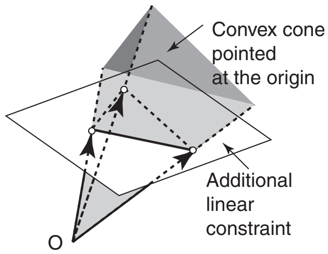


Figure 3: Convex polyhedron defined by a convex cone pointing at the origin and an additional linear constraint: ***The circles denote the vertices of the convex polyhedron other than the origin, and the arrows the vectors from the origin to these vertices.***

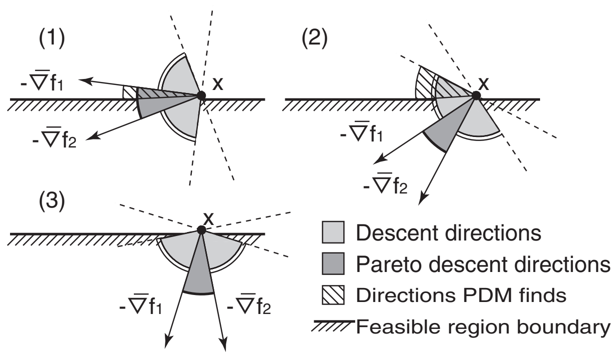


Figure 4: Directions that PDM finds: ***PDM finds appropriate feasible directions for line search in accordance with the feasibility of Pareto descent directions and descent directions***

When a solution is on a feasible region boundary, part of Pareto descent directions or descent directions may be infeasible, as in the cases shown in Fig. 4(1) and Fig. 4(2). These infeasible directions can be excluded by incorporating the boundary information into the direction calculation as done in [4].

Since finding a direction requires some computation, it is reasonable to move the solution in that direction until just before any of the objective functions deteriorate or any of the constraints are violated.

If no feasible descent directions can be found as in the case shown in Fig. 4(3), the above-mentioned method can detect it, thereby providing a test of (local) Pareto-optimality.

The details of the proposed algorithm are elucidated in the following subsections, followed by the flowchart of the whole algorithm.

3.2 Finding Pareto Descent Directions at Solutions inside Feasible Regions

Recall that Pareto descent directions satisfy Eq. (2) and can be expressed as Eq. (3). Substituting Eq. (3) into Eq. (2) gives:

$$\sum_{i=1}^{M} \alpha_i \beta_{ij} \geq 0 \ (j = 1, 2, \ldots, M), \tag{4}$$

where $\beta_{ij} = \bar{\nabla} f_i(\boldsymbol{x}) \cdot \bar{\nabla} f_j(\boldsymbol{x})$. This is a simultaneous linear inequality of combination weights $\alpha_i \geq 0$ $(i = 1, 2, \ldots, M)$. Since all constant terms are 0, the weight vector $\boldsymbol{\alpha}$ satisfying the inequality forms a convex cone pointed at the origin. Suppose imposing the following linear constraint:

$$\sum_{i=1}^{M} \alpha_i \leq 1. \tag{5}$$

The set of $\boldsymbol{\alpha}$ satisfying all linear inequalities forms a convex polyhedron having the origin as one of its vertices. For each $k \in \{1, 2, \ldots, M\}$, consider solving the following linear programming problem of finding $\boldsymbol{\alpha}$ that maximizes α_k:

$$\begin{array}{ll} \text{Maximize} & \alpha_k \\ \text{subject to} & \sum_{i=1}^{M} \alpha_i \beta_{ij} \geq 0 \ (j = 1, 2, \ldots, M), \\ & \sum_{i=1}^{M} \alpha_i \leq 1, \text{ and } \alpha_i \geq 0 \ (i = 1, 2, \ldots, M). \end{array} \tag{6}$$

This linear programming problem has an obvious feasible solution $\boldsymbol{\alpha} = \boldsymbol{0}$ for artibrary β_{ij}. If $\boldsymbol{\alpha} = \boldsymbol{0}$, the convex combination of the steepest descent directions of objective functions does not represent a direction. Hence the obvious solution $\boldsymbol{\alpha} = \boldsymbol{0}$ has to be separated. Since α_k is maximized, $\sum_{i=1}^{M} \alpha_i = 1$ holds if some $\boldsymbol{\alpha} \neq \boldsymbol{0}$ exist. Therefore, if the solution to the linear programming problem is $\boldsymbol{\alpha} = \boldsymbol{0}$, Pareto descent directions do not exist, and neither do descent directions. On the other hand, if the solution is $\boldsymbol{\alpha} \neq \boldsymbol{0}$, the steepest descent directions combined with the weight $\boldsymbol{\alpha}$ gives a Pareto descent direction. Since Pareto descent directions form a convex cone, convex combinations of thus found M Pareto descent directions are also Pareto descent directions.

For solutions inside feasible regions, there exist Pareto descent directions unless they are (locally) Pareto-optimal. The above-mentioned method can test the existence of such directions when solving the linear programming problem. Thus, it provides a test of (local) Pareto-optimality.

3.3 Finding Pareto Descent Directions at Solutions on Feasible Region Boundaries

If a solution is on a feasible region boundary and part of its Pareto descent directions are infeasible as shown in Fig. 4(1), some of the directions found with the method described in the previous subsection may be infeasible, which may undermine the efficiency of searching. Therefore, for such a solution, it is desirable to exclude infeasible directions.

Suppose that the active constraint $g_j(\boldsymbol{x}) \geq 0$ is linear. The inequality can be transformed into the Hessian normal form $\boldsymbol{n}_j \cdot \boldsymbol{x} \geq l_j$, where $\boldsymbol{n}_j$ is a unit vector and l_j is a real value, which reveals that $\boldsymbol{n}_j$ is the normal vector to the boundary [5]. Infeasible directions can be excluded by adding the following linear constraint to the linear programming problem in Eq. (6):

$$\boldsymbol{d} \cdot \boldsymbol{n}_j \geq 0.$$

If the solution is on the boundary specified by multiple linear constraints, they can be incorporated all together.

If the constraint $g_j(\boldsymbol{x}) \geq 0$ is not linear, Taylor expansion can approximate it with a linear constraint in the vicinity of $\boldsymbol{x}$. Hence, nonlinear constraints can also be handled.

3.4 Finding Descent Directions at Solutions on Feasible Region Boundaries

When a solution is on a feasible region boundary and the method described in Subsection 3.3 does not find Pareto descent directions, feasible descent directions may still exist, as in the case shown in Fig. 4(2). In such a case, it is desirable to find feasible descent directions in which the solution can be moved to improve all objective functions.

Recall that the complete set of descent directions forms a convex cone pointed at the origin as noted in Subsection 2.1.2. Imposing the following constraint on the set of descent directions gives a convex polyhedron having the origin as one of its vertices:

$$-1 \leq d_i \leq 1 \ (i = 1, 2, \ldots, N). \tag{7}$$

Solving the following linear programming problem for some weight vector $\boldsymbol{w} = (w_1, w_2, \ldots, w_N)$ gives one of the vertices of the convex polyhedron:

$$\begin{array}{ll} \text{Maximize} & \boldsymbol{w} \cdot \boldsymbol{d} \\ \text{subject to} & \boldsymbol{d} \cdot (-\bar{\nabla} f_i(\boldsymbol{x})) \geq 0 \ (i = 1, 2, \ldots, M), \\ & -1 \leq d_i \leq 1 \ (i = 1, 2, \ldots, N). \end{array} \tag{8}$$

Since a vertex of the convex polyhedron is either the origin or a point on the hypercube defined by Eq. (7), whether $\boldsymbol{d} = \boldsymbol{0}$ can easily be tested. If $\boldsymbol{d} \neq \boldsymbol{0}$, the direction from the origin to the vertex is a descent direction. If $\boldsymbol{d} = \boldsymbol{0}$, it does not represent a direction and has to be separated. Solving Eq. (7) for many randomly generated $\boldsymbol{w}$ gives multiple descent directions. Since descent directions form a convex cone, convex combinations of thus found directions are also descent directions. The information of the active feasible region boundary can also be incorporated into the calculation as done for finding feasible Pareto descent directions.

Suppose that the solution of the linear programming problem is $\boldsymbol{d} = \boldsymbol{0}$ for some weight $\boldsymbol{w}$. If there are any vertices of the convex polyhedron other than the origin, solving the linear problem with the weight $-\boldsymbol{w}$ gives one of such vertices. Its contraposition allows for the test of existence of descent directions, hence the test of (local) Pareto-optimality.

3.5 Stepsize Calculation

Having found a feasible Pareto descent or descent direction at a solution as described in Subsections 3.2 through 3.4, the distance at which the solution is moved from its original position has to be determined. Since finding directions entails some computational cost, it is reasonable to move the solution until just before any of the objective functions deteriorate or any of the constraints are violated. The same strategy is also employed in [1]. Assuming the local unimodality of objective functions, golden section method can be used to determine the distance.

3.6 PDM Algorithm

The proposed LS method, PDM, comprises these components described thus far. The algorithm flowchart of PDM is shown in Fig. 5, in which it is assumed that a solution can exist either inside the feasible region or on the boundary of it. The algorithm comes to an end when the number of times direction calculation and line search are conducted reaches T.

3.7 Computational Complexity

Since the most computationally intense part of PDM is that of solving linear programming problems for finding directions, the computational complexity of PDM primarily depends on that of the linear programming solver employed in PDM. The most commonly used linear programming solver is simplex method. Although simplex method can require

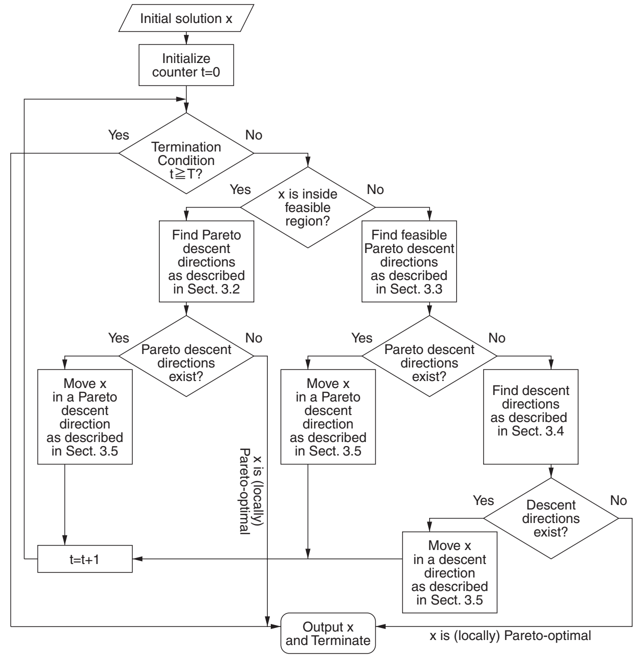


Figure 5: Algorithm flowchart of PDM

exponential time in the worst case, it often solves general linear programming problems quickly in practice [2]. In addition, there is a class of polynomial-time solvers which is known as interior-point methods [2]. When such methods are employed, it can be said that PDM is a polynomial-time algorithm.

Since PDM finds feasible Pareto descent directions or descent directions as appropriate, the factors that influence computational complexity of direction calculation in PDM change accordingly. When PDM finds feasible Pareto descent directions, the computational complexity depends only on the number of objective functions and the number of active constraints, *not* on the number of variables. However, when PDM finds feasible descent directions, it depends on the number of variables as well.

4. EXPERIMENTS

4.1 Overview

This section shows the results of the experiments comparing the performances of PDM, RDS, WSDM, and CORL and verifying that PDM exhibits favorable behaviors as described thus far. MSDM was not included in the comparison because of its notably high computational cost.

For problems without local Pareto-optimal solutions, it is desirable for LS methods to move an arbitrary initial solution in the feasible region to one of the Pareto-optimal solutions. How well this is achieved can be measured by the squared distance, or error, between the solution and its nearest Pareto-optimal solution in the variable space. Hence, the performances of LS methods can be compared by examining the transition of the mean squared error (MSE) of many solutions that are initially distributed uniformly at random across the feasible region as the LS methods are applied to them.

In Experiment 1, the LS methods are compared on a 30-variable-3-objective benchmark problem MED1(e) in order to investigate the effects of the convexity of the Pareto-optimal solutions in the objective space. The Pareto-optimal solutions of MED1(e) can be made convex or non-convex by adjusting the parameter e. RDS is expected to achieve limited improvements in objective functions since MED1(e) is of high dimension and RDS can hardly find even descent directions. If the Pareto-optimal solutions of MED1(e) are non-convex in the objective space, WSDM is unable to move solutions to the Pareto-optimal solutions in the non-convex part. CORL is expected to achieve limited improvements in objective functions since it often finds descent directions on MED1(e), as described in the appendix. PDM is expected to improve all objective functions efficiently by moving solutions in Pareto descent directions, regardless of the convexity of the Pareto-optimal solutions in the objective space.

In Experiment 2, the LS methods are compared on a 2-variable-2-objective benchmark problem MED2 in order to investigate the effects of feasible region boundaries. When a solution is on some part of MED2's feasible region boundary, all steepest descent directions and all Pareto descent directions are infeasible. Such a solution cannot be moved any closer to the Pareto-optimal solutions with WSDM and CORL since the directions they find are all infeasible. However, PDM can move such solution closer to the Pareto-optimal solutions by moving it in feasible Pareto descent direction or descent direction as appropriate. RDS is expected to perform poorly since it relies on finding descent directions solely by chance.

4.2 Performance Metrics and Experiment Setups

The LS methods are applied to 10^4 initial solutions distributed uniformly at random across the feasible region, and their performances are compared by examining the transition of the MSE as the LS methods are applied to them.

Gradients are approximated by forward difference with the difference of 10^{-4}. Simplex method is used in direction calculation of PDM. In order for PDM to sufficiently approximate the complete convex cone of feasible descent directions, 40 combination weights are randomly drawn for direction calculation.

All the LS methods employ the same line search method, golden section method, with the basic search segment length of 10^{-2}, the maximum number of extension of the segment of 20, and the number of iteration of 20. A solution is assumed to be on a feasible region boundary if the distance between them is less than $10^{-2} \times \tau^{20}$, where τ is the golden ratio. In order to offset RDS's inefficiency of finding descent directions, up to 10^4 directions are drawn for one iteration of RDS to the advantage of the number of function evaluations for finding a direction. If it finds a descent direction, line search is conducted in that direction. Otherwise, that particular iteration of RDS is declared unsuccessful. Note that the numbers of function evaluations PDM, WSDM, and CORL require for finding a direction are the same.

4.3 Experiment 1: Comparison on a 3-Objective Problem

4.3.1 Benchmark Problem

The objective functions of the 30-variable-3-objective benchmark problem MED(Multiple Euclidean Distances)1(e) are:

$$f^e_{11}(\boldsymbol{x}) = ||\boldsymbol{x}-\boldsymbol{c}_{11}||^e,\ f^e_{12}(\boldsymbol{x}) = ||\boldsymbol{x}-\boldsymbol{c}_{12}||^e, f^e_{13}(\boldsymbol{x}) = ||\boldsymbol{x}-\boldsymbol{c}_{13}||^e,$$

where $\boldsymbol{c}_{11} = (1, 1, 0, \dots, 0), \boldsymbol{c}_{12} = (0.1, 0, 0, \dots, 0)$, and $\boldsymbol{c}_{13} = (0, 0.1, 0, \dots, 0)$, and the feasible region is $[0, 1]^2 \times [-0.5, 0.5]^{28}$. Its Pareto-optimal solutions form a triangle whose vertices are at $\boldsymbol{c}_{11}, \boldsymbol{c}_{12}$, and $\boldsymbol{c}_{13}$. Parameter values $e = 2,\ 0.5$ are used, for which the Pareto-optimal solutions are convex and non-convex, respectively, in the objective space.

4.3.2 Results and Consideration

Fig. 6 shows the transition of the MSE against the number of times a direction is found and line search is conducted for each solution.

On MED1(2), WSDM gave an excellent value and speed of convergence. Although the convergence speed of PDM was slower than that of WSDM, its convergence value was comparable to that of WSDM. CORL performed poorly as it often finds descent directions on MED1(e). RDS performed worst, since its direction search is inefficient on high dimensional problems.

On MED1(0.5), WSDM again performed best with respect to both the value and speed of convergence, whereas the convergence of PDM was slightly slower than that of WSDM. RDS and CORL performed poorly for the same reasons as they did in MED1(2). Note, however, that WSDM is not necessarily a suitable LS method. Fig. 7 shows the distribution of the solutions obtained with PDM and WSDM projected onto the x_1-x_2 coordinate. The figure shows that PDM moved solutions to the whole Pareto-optimal solutions while WSDM moved most of the solutions to either of $\boldsymbol{c}_{11}, \boldsymbol{c}_{12}$, or $\boldsymbol{c}_{13}$ and the remainder to a small number of Pareto-optimal solutions. This bias is undesirable for the purpose of multiobjective optimization.

The above consideration reveals that the LS method which has a good value and speed of convergence and which does not bias solutions to a limited portion of the Pareto-optimal solutions is PDM.

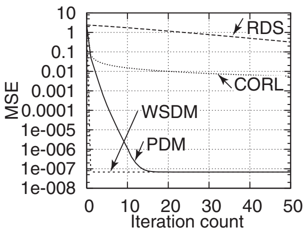
4.4 Experiment 2: Comparison on a Problem Whose Pareto-Optimal Solutions are on its Feasible Region Boundary

4.4.1 Benchmark Problem

The objective functions of the 2-variable-2-objective benchmark problem MED2 are:

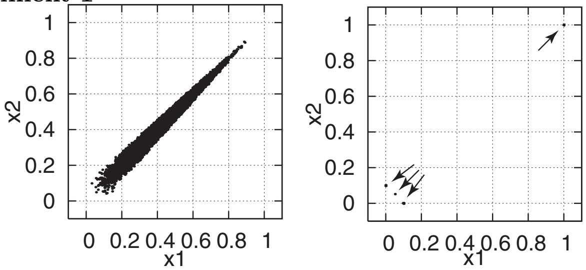
$$f_{21} = ||\boldsymbol{x} - \boldsymbol{c}_{21}||, f_{22} = ||\boldsymbol{x} - \boldsymbol{c}_{22}||,$$

where $\boldsymbol{c}_{21} = (0, -1), \boldsymbol{c}_{22} = (1, -1)$, and the feasible region is $[-1, 2] \times [0, 1]$. Its Pareto-optimal solutions form a straight line segment connecting $(0, 0)$ and $(1, 0)$, which is on the feasible region boundary. For solutions in the feasible region, all convex combinations of the steepest descent directions are Pareto descent directions. These directions are infeasible for solutions on the x_1 axis. Initial solutions in the region $[0, 1]^2$ can be moved to the Pareto-optimal solutions by moving them in Pareto descent directions. However, those in the other feasible region may have to be moved in feasible descent directions as well in order for them to reach the Pareto-optimal solutions. These two regions are named Region A and Region B, respectively, and are shown in Fig. 8. PDM, WSDM, and CORL exhibit identical behaviors for



MED1(2) MED1(0.5)

Figure 6: Transition of MSE on MED1(e) in Experiment 1



PDM WSDM

Figure 7: Distribution of the solutions obtained with PDM and WSDM on MED(0.5) in Experiment 1: ***The solutions are projected onto the x_1-x_2 coordinate.***

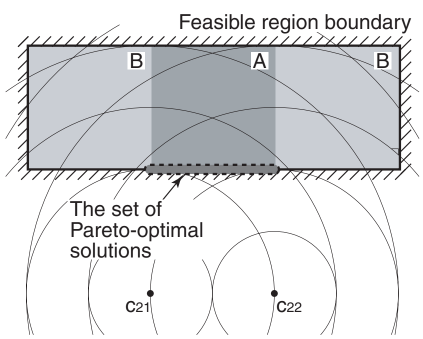


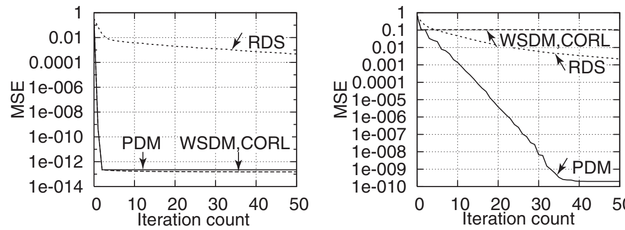
Figure 8: Pareto-optimal solutions, Region A, and Region B of MED2: ***The thin lines are the contours of the objective functions***

initial solutions in Region A. However, they behave differently for initial solutions in Region B: PDM moves solutions in feasible descent directions when all Pareto descent directions become infeasible, whereas WSDM and CORL stagnate once the solutions reach the x_1 axis.

4.4.2 Results and Consideration

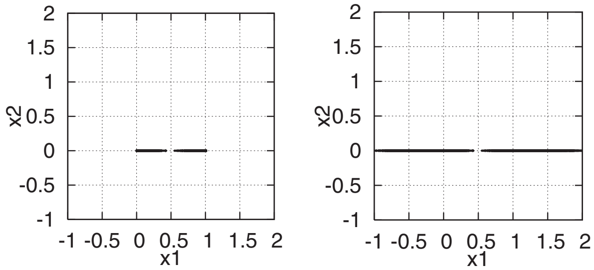
Fig. 9 shows the transition of the MSE for initial solutions in Region A and Region B against the number of times a direction is found and line search is conducted for each solution.

For initial solutions in Region A, PDM, WSDM, and CORL performed equally well regarding the value and speed of convergence. For those in Region B, PDM improved MSE steadily by moving solutions in feasible descent directions when all Pareto descent directions are infeasible. On the contrary, WSDM and CORL stagnated due to their inability to move solutions once they reach the x_1 axis and all Pareto



Region A Region B

Figure 9: Transition of MSE on MED2 in Experiment 2



PDM WSDM

Figure 10: Distribution of the solutions obtained by applying PDM and WSDM to initial solutions in Region B of MED2 in Experiment 2

descent directions become infeasible. RDS performed poorly in both regions due to the inefficiency of its direction search.

Fig. 10 shows the distribution of the solutions obtained with PDM and WSDM that were initially located in Region B. The figure shows that PDM moved the solutions to the Pareto-optimal solutions, whereas WSDM moved them to the x_1 axis but not any closer to the Pareto-optimal solutions.

Hence, it has been verified that PDM efficiently improve all objective functions simultaneously by moving them in feasible Pareto descent directions or descent directions as appropriate for solutions inside feasible regions and on the boundaries of them.

5. CONCLUSION

Based on the observation of the LS methods known in the multiobjective function optimization literature and on the consideration of Pareto descent directions, this paper clarified the requirements for LS methods to achieve high efficiency and effectiveness: the ability to find Pareto descent directions for solutions inside feasible regions and feasible Pareto descent directions or descent directions for those on feasible region boundaries on problems with arbitrary numbers of variables, objectives, and constraints at a relatively small computational cost. This paper proposed a new LS method, PDM, which meets these requirements. It also provides a test of (local) Pareto-optimality, which is another advantage of PDM.

Two experiments were conducted, which verified that PDM exhibits the expected favorable behaviors and achieves efficient improvements of all objective functions for solutions both inside the feasible region and on the boundary of it on 2- and 3-objective problems.

GA is known as a potent global optimization method for its ability to overcome local Pareto-optimal solutions by maintaining a set of solutions during the course of its search. However, a recently published paper [10] showed some experimental results from which it can be inferred that GA may not be suitable for obtaining solutions of high precision for multiobjective function optimization. One remedy for this is its hybridization with an LS method, such as PDM. We are planning to investigate how the performance of the hybridization is affected by the particular LS method chosen for the hybridization and the way in which GA and LS are hybridized.

6. ADDITIONAL AUTHORS

Additional authors: Kokolo Ikeda (Kyoto University, email: kokolo@media.kyoto-u.ac.jp) and Isao Ono (Tokyo Institute of Technology, email: isao@dis.titech.ac.jp).

7. REFERENCES

[1] P. A. N. Bosman and E. D. de Jong. Exploiting gradient information in numerical multi-objective evolutionary optimization. In H.-G. Beyer, U.-M. O'Reilly, D. V. Arnold, W. Banzhaf, C. Blum, E. W. Bonabeau, E. Cantú-Paz, D. Dasgupta, K. Deb, J. A. Foster, E. D. de Jong, H. Lipson, X. Llora, S. Mancoridis, M. Pelikan, G. R. Raidl, T. Soule, A. M. Tyrrell, J.-P. Watson, and E. Zitzler, editors, *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO-2005*, pages 755–762, New York, NY, USA, 2005. ACM Press.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, second edition, 2001.

[3] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Chichester, 2001.

[4] J. Fliege and B. F. Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.

[5] W. Gellert, S. Gottwald, M. Hellwich, H. Kästner, and H. Künstner, editors. *VNR Concise Encyclopedia of Mathematics*. Van Nostrand Reinhold, New York, 1989.

[6] T. Goel and K. Deb. Hybrid methods for multi-objective evolutionary algorithms. In *Proceedings of the Fourth Asia-Pacific Conference on Simulated Evolution and Learning (SEAL'02)*, pages 188–192, 2002.

[7] H. Ishibuchi and K. Narukawa. Some issues on the implementation of local search in evolutionary multiobjective optimization. In *Proceedings of Genetic and Evolutionary Computation Conference LNCS*, pages 1246–1258, 2004.

[8] J. D. Knowles and D. W. Corne. M-PAES: a memetic algorithm for multiobjective optimization. In *Proceedings of the Congress on Evolutionary Computation (CEC00)*, pages 325–332, Piscataway, NJ, 2000. IEEE Press.

[9] J. D. Knowles and D. W. Corne. Memetic algorithms for multiobjective optimization: issues, methods and prospects. In N. Krasnogor, J. E. Smith, and W. E. Hart, editors, *Recent Advances in Memetic Algorithms*, pages 313–352. Springer, 2004.

[10] M. Laumanns, G. Rudolph, and H.-P. Schwefel. Mutation control and convergence in evolutionary multi-objective optimization. In Matousek and P. Osmera, editors, *Proceedings of the 7th International Mendel Conference on Soft Computing (MENDEL 2001)*, pages 24–29, 2001.

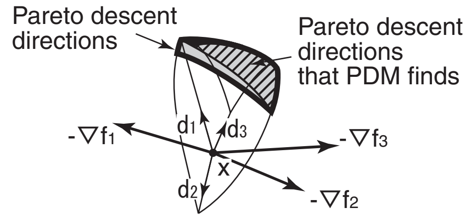


Figure 11: A 3-variable-3-objective problem for which CORL may not find Pareto descent directions: ***The long arrows denote the normalized steepest descent directions of the objective functions, the thin lines denote the cross-section of the descent direction set and the unit sphere centered at x, the thick lines denote that of the Pareto descent direction set and the unit sphere, and the short arrows denote the directions that CORL calculates for the solution x. The cross-hatched area denotes the Pareto descent directions that PDM finds.***

APPENDIX

CORL was proposed as a method that analytically finds the vectors that generate the complete set of Pareto descent directions and moves solutions in these directions [1]. Although CORL is guaranteed to find Pareto descent directions on 2-objective problems, it may not find them on problems with more than 2 objectives. For example, at a solution $\boldsymbol{x}$ of a 3-variable-3-objective problem shown in Fig. 11, the vectors that CORL calculates are $\boldsymbol{d}_1, \boldsymbol{d}_2$, and $\boldsymbol{d}_3$. However, $\boldsymbol{d}_2$ and $\boldsymbol{d}_3$ are not Pareto descent directions and their convex combination may not be a Pareto descent direction.

In this situation, PDM finds the Pareto descent directions denoted by the cross-hatched area in Fig. 11. Note that PDM does not necessarily find the entire Pareto descent directions for problems with more than 2 objectives. However, PDM is guaranteed to find a multiple of Pareto descent directions with relatively small computational cost, as opposed to CORL which may not find such directions and MSDM which finds only a single such direction by solving a quadratic programming problem.

The situation shown in Fig. 11 often occurs on the 3-objective benchmark problem MED1(e), and CORL is expected to perform poorly when applied to this problem, which was confirmed by Experiment 1 in Section 4.3.

The frequency with which the situation shown in Fig. 11 occurs increases in general as the number of objective functions increases. Therefore, the larger the number of objective functions, the more likely CORL finds descent directions instead of Pareto descent directions and, hence, poorer CORL's performance. We have verified this through experiments conducted on the benchmark problem SPH [10] with various number of objective functions.