9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization
4-6 September 2002, Atlanta, Georgia

AIAA 2002-5432

A SPOTLIGHT SEARCH METHOD FOR MULTI-CRITERIA OPTIMIZATION PROBLEMS

Jung-Ho Lewe*
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0150

Abstract

In many engineering applications including multidisciplinary optimization (MDO) problems, a decision maker often tends to evaluate and optimize multiple criteria at the same time to carry out a tradeoff study. Nevertheless, solving an multi-criteria optimization (MCO) problem, in general, is a difficult practice. Numerous methods have been proposed and applied to various applications over the past few decades. This paper introduces a new MCO method suitable for continuous, nonlinear MCO problems. The concept of the proposed method, a spotlight search method (SSM), is easily comprehendible and its implementation is simple and straightforward. Mathematical formulations show that this method can be considered as a variation of the goal attainment method (GAM). Five test problems are selected and numerical experiments are presented to demonstrate the usefulness of SSM with comparison to GAM. It is observed that SSM finds the Pareto front more efficiently for all test problems.

Nomenclature

$\mathbb{R}^m$	m-dimensional Euclidean space
Θ	design space
Ω	criterion (objective) space
$\partial\Omega$	boundary of Ω
X	design vector
$F(X)$, Y	criterion (objective) vector
N	dimension of a design space
n	dimension of a criterion space
$f_i(X)$	i-th criterion (objective) function
$G(X)$	inequality constraint vector function
$H(X)$	equality constraint vector function
$B(X,\delta)$	open ball centered at X
$\mathcal{P}$	Pareto front (Pareto optimal set) in Ω
$\hat{F}$	goal vector
$\|\ \|_p$	L_p–norm
ν	search (spotlight) vector

*Graduate Research Assistant, Student Member AIAA

1 Introduction

Every decision-making problem involves multiple criteria by nature. Engineering problems are no exception; they often require addressing multiple objectives which usually contradict one another. For instance, an aircraft manufacturer may want to build a new vehicle that incurs low R&D cost yet possesses good performance characteristics. However, lower R&D cost compromises performance, necessitating a design tradeoff.

While optimization with a single objective is a fundamental technique and can be found ubiquitously in practice, there are increasing demands of research on multi-criteria decision making (MCDM) techniques in academia, industry and government. A designer can make a 'better quality' decision when multiple criteria are considered simultaneously. This contributes to expanding the knowledge boundary on complex design space in or even before the conceptual design phase. For this reason, it is forecasted that MCDM techniques will be more regularly exercised on practical problems in every domain.[1, 2]

MCDM can be divided into two branches[3, 4]: multi-attribute decision making (MADM) and multi-criteria optimization (MCO). In general, MADM relates to techniques that aid a decision maker in choosing the best design from a small number of alternatives. MCO is also known as multi-objective optimization or vector optimization. Its task is to present a set of designs that are the most appealing alternatives to a decision maker. To do this task efficiently, numerous MCO methods have been developed for the past few decades.[5, 6, 7] But one cannot judge that a particular method is superior to others. This is mainly because characteristics of MCO problems are diverse depending on problem-specific situations. It is next to impossible to come up with a generic method that works evenly well for every MCO problem. Therefore, the key criteria in choosing from various methods should be based on the practicability. In other words, a user should look into not only whether the method fits their specific needs appropriately, but also the difficulty or simplicity of numerical implementation.

The purpose of the present work is to propose a new MCO method suitable for continuous, nonlinear MCO problems. The concept of the proposed method, called a spotlight search method (SSM), is very easily understandable and its implementation is simple and straightforward.

The rest of this paper is organized as follows. Section 2 briefly overviews background information on MCO. In the succeeding section, SSM is introduced in detail with a constraint handling technique. A comparison study to the most similar method, the goal attainment method (GAM) by Gembicki, is then presented. Section 4 provides the results of numerical experiments performed with comparison to those from GAM. This is followed by conclusion.

2 Multi-Criteria Optimization

2.1 Fundamentals

Envision an N-dimensional design space $\Theta \subset \mathbb{R}^N$, a design vector $X \in \Theta$, an n-dimensional criterion space $\Omega \subset \mathbb{R}^n$, a criterion vector $Y \in \Omega$, and a mapping $F : X \in \Theta \longmapsto Y \in \Omega$. This is illustrated in Figure 1 as an example taking $N = 3$ and $n = 2$. If the task is optimizing either f_1 or f_2, one simply needs to employ a single objective optimization method. Then the solution would end up to point P or P' in this particular example. However, the core of MCO is to minimize both functions at the same time. It might no longer be as simple a task as before. Mathematically, a general form of an MCO problem is stated as follows:

$$\begin{array}{l} \text{“Minimize” } F(X) = [f_1(X), f_2(X), \cdots, f_n(X)]^T \\ \text{Subject to: } G(X) \leq O,\ H(X) = O \end{array} \tag{1}$$

The scalar function $f_i(X)$ denotes an i-th criterion that is an element of the criterion vector $F(X)$. The vector function $G(X)$ and $H(X)$ indicate inequality and equality constraints respectively that bound a feasible design space Θ, i.e., $\Theta = \{X | G(X) \leq O,\ H(X) = O\}$. The symbol O simply indicates a zero vector.

The difficulty comes in due to the nature of $F(X)$. Since $F(X)$ is an n-dimensional vector function, unlike single objective optimization problems, the solution of

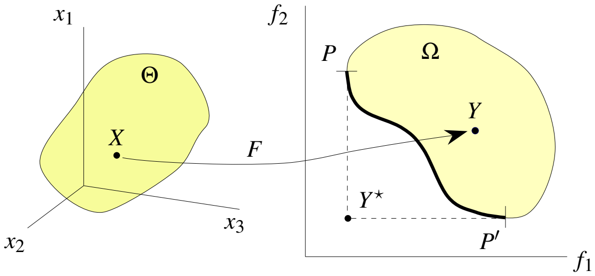


Figure 1: Design Space and Criterion Space

Problem (1) would be a set of optima rather than a single optimum. For example, any point on the thick curve $\widetilde{PP'} \subset \Omega$ in Figure 1 can be presented as the solution of the MCO problem. This is called a *non-inferior solution*, *nondominated solution*, *efficient point*, or *Pareto optimum*. The mathematical definition of a Pareto optimum is given as follows[3]:

Definition 1. *(Global) Pareto optimum*
A design vector $X^* \in \Theta$ is called a *Pareto optimum* if there does not exist another design vector $X \in \Theta$ such that $f_i(X) \leq f_i(X^*)$ for all $i = 1, 2, \cdots, n$ and $f_j(X) < f_j(X^*)$ for at least one index j.

A criterion vector $Y^* \in \Omega$ is a *Pareto optimum* if the design vector corresponding to it is a Pareto optimum.

Now, *Pareto front* $\mathcal{P}$ can be defined as a set of all Pareto optima in Ω. In a two criteria case, for example, it can be visualized by drawing a curve as in Figure 1; in a three criteria case, a surface. A very similar term, a local Pareto optimum, is defined for later use.

Definition 2. *Local Pareto optimum*
A design vector $X^* \in \Theta$ is called a *local Pareto optimum* if there exists $\delta > 0$ such that X^* is a Pareto optimum in $\Theta \cap B(X^*, \delta)$.

An objective vector $Y^* \in \Omega$ is a *local Pareto optimum* if the design vector corresponding to it is a local Pareto optimum.

It is obvious that X^* being a global Pareto optimum implies that X^* is a local Pareto optimum. The converse does not always hold, which imposes another difficulty in solving MCO problems. Lastly, another useful term is introduced.

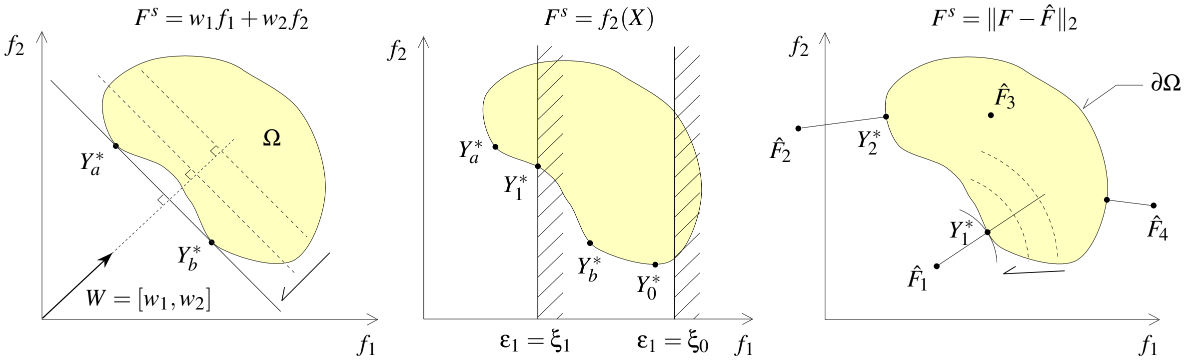
Definition 3. *Utopian vector*
A *utopian vector* $Y^\star \in \mathbb{R}^n$ is defined such that $Y^\star \triangleq [\min f_1, \min f_2, \cdots, \min f_n]^T$. Each minimization is required to satisfy the original constraints, i.e., $X \in \Theta$.

The utopian vector $Y^\star$ is marked in Figure 1. It can be perceived as a goal or an aspiration point. An MCO problem collapses to a single objective optimization problem when a utopian vector sits on $\mathcal{P}$.

Note that the names and definitions throughout this paper, including this section, may differ in literature. The reader is encouraged to check the terms whenever referring to literature related to MCO topics.

2.2 MCO Methods

The objective of MCO methods is to locate Pareto optima and from that, to generate a complete Pareto front



(a) Weighted Sum Method (b) ε-Constraint Method (c) Distant Metric Method

Figure 2: One-by-one Strategy

$\mathcal{P}$. Numerous methods have been developed to do this task efficiently, so certain classification of the methods are helpful. However, even the ways of the classification are shaky and numerous; no classification is absolute. For example, Hwang and Masud[5] categorized the methods according to the decision maker's participation in the optimization process; namely, no-preference, a posteriori, a priori and interactive methods. Carmichael[8] did so into three bases: a composite single criterion, a single criterion with constraints and many single criteria. In the present research, the classification is made based on whether a single solution or a set of solutions is generated at a single execution of the method.

2.2.1 One-by-one Strategy

This strategy begins with switching an MCO problem to a single objective optimization problem by introducing a surrogate function $F^s : X^s \times \alpha \mapsto \mathbb{R}^1$. The new design vector X^s consists of the original design vector X and an extra design vector λ if needed. The parameter vector α is a necessary input to coordinate a search procedure. Now Problem (1) will be converted as follows:

$$\begin{aligned} &\underset{X \in \Theta}{\text{Minimize}} \;\; F^s = F^s(X^s;\alpha) \\ &\text{Subject to: } G^s(X,\lambda;\alpha) \le O,\; H^s(X,\lambda;\alpha) = O \end{aligned} \quad (2)$$

where G^s and H^s are additional constraints. These extra entities, including λ, may or may not be present to complete the conversion process depending on a method. Now that Problem (1) has changed to a surrogate Problem (2), only a single Pareto optimum would be searched at a single execution if the solution converged successfully. Changing in the value of α will entail a new Pareto optimum. Through this sequential process, the Pareto front $\mathcal{P}$ would be formed by accumulating the Pareto optima. An outline of three basic approaches adopting the one-by-one strategy follows.

• *Weighted Sum Method*

This method is based on a naïve idea. F^s is simply defined as a composite of each criterion.

$$F^s \triangleq W \cdot F(X) = \sum_{i=1}^{n} w_i \, f_i(X) \quad (3)$$

Here, $W = [w_1, w_2, \cdots, w_n]$ (usually $\sum w_i = 1$) corresponds to the parameter vector α in Problem (2). By perturbing parameter vector W or weights, each optimization process will produce a different Pareto optimum. The serious drawback of the method is that the method cannot generate complete description of a Pareto front that is not convex. This situation is depicted in Figure 2(a). No matter how W is altered, the portion of the Pareto front between Y_a^* and Y_b^* can never be obtained.

• *ε-Constraint Method*

The ε-constraint method is also based on a simple idea which has become a routine for single objective optimization process. The most important criterion function is chosen and optimized taking the remaining criterion functions as constraints. Hence, F^s is defined as follows without loss of generality:

$$F^s \triangleq f_n(X) \quad (4)$$

The other criteria are incorporated into G^s. This extra constraint vector has $(n\text{–}1)$ inequality constraints and they are:

$$f_i(X) - \varepsilon_i \le 0 \;\; (i = 1, 2, \cdots, n\text{–}1) \quad (5)$$

Specific ε_i values (for $i = 1, \cdots, n\text{–}1$) need to be determined before the optimization. The optimization process is illustrated in Figure 2(b) taking $F^s = f_2(X)$. The Pareto optimum Y_0^* would be obtained with given value $\varepsilon_1 = \xi_0$. If the optimization is repeated with $\varepsilon_1 = \xi_1$, the Pareto optimum Y_1^*, which could not be obtained by the previous method, can be searched. The major drawback of this method is that it is time consuming to figure out appropriate numeric values for ε_i, especially when $n = \dim\Omega$ becomes larger.

• *Distance Metric Method*

In this method, F^s is defined as follows:

$$F^s \triangleq \|F(X) - \hat{F}\|_p = \left(\sum_{i=1}^{n} |f_i(X) - \hat{f}_i|^p\right)^{\frac{1}{p}} \qquad (6)$$

where $\hat{F} = [\hat{f}_1, \hat{f}_2, \cdots, \hat{f}_n]^T$ is a goal vector which is a predetermined vector by a user. The parameter p is usually 1, 2 or infinity. The basic idea behind this method is described in Figure 2(c). The function F^s measures distance from $F(X)$ to $\hat{F}$. If the optimizer minimizes the distance from $\hat{F}_1$, the point Y_1^* will be obtained. Moving a goal to $\hat{F}_2$ will make the optimizer search the point Y_2^*. However, this method is sensitive to the position of $\hat{F}$. For example, starting from $\hat{F}_3 \in \Omega$ will do nothing. Furthermore, if $\hat{F}_4$ is chosen, a meaningless point on $\partial\Omega$ will be the final outcome.

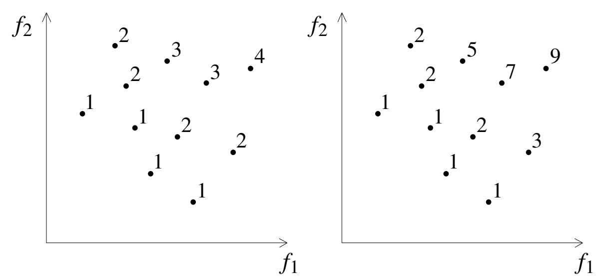
2.2.2 All-at-once Strategy

Even though various methods adopting the one-by-one strategy have been successfully applied to many applications, there exist a number of weaknesses. First, as the name implies, many repetitions are required to have the entire Pareto front. Second, some methods are very sensitive to the shape of the Pareto front. Last but not least, in order to do an effective search, the one-by-one strategy requires some degree of a priori knowledge about the criterion space. These limitations can be resolved through the all-at-once strategy also known as multiobjective evolutionary algorithms (MOEAs) or multiobjective genetic algorithms (MOGAs). These GA-based MCO techniques have gained much attention over the past decade since they have intriguing concepts and a lot of potential to tackle MCO problems.

A genetic algorithm (GA) is a stochastic optimization method mimicking the evolutionary process. Its usefulness is notably growing in accordance with the current escalating trend of computing power. Genetic algorithms utilize a set of chromosomes referred as a population. A chromosome, typically represented in the form of binary string, is a pointer to a particular design vector in a (discrete) design space. Three genetic operations (selection, crossover and mutation) play a role in generating the next population. This process continues until a user commands to stop.

Genetic algorithms (GAs) are sharply distinguished from calculus-based optimization algorithms in that they do not call for analytic information from an objective function. Thus, they can deal with objective functions that need not be differentiable or continuous. This feature makes GAs a versatile optimization method. However, GAs should work on a discrete design space (except the real GAs). Also, unlike calculus-based optimization algorithms, it is difficult to check whether the final outcome is a converged one. The most serious issue of a GA is that it requires much more function calls than any other optimization algorithms does.

In what follows, two specific ways to combine a GA and MCO will be introduced. One is called nondominated sorting procedure proposed by Goldberg.[9] Among the population, the nondominated individuals are ranked 1 then they are removed. The next nondominated individuals are ranked 2 and also removed. This process will be repeated and is illustrated in Figure 3(a). Fonseca and Flemming[10] presented a different scheme focusing on each individual, which is depicted in Figure 3(b). If an individual is dominated by R other individuals, then $(R+1)$ is assigned for its rank. Under the Darwinian principle, the survival of the fittest, top-ranking individuals are likely to be chosen to reproduce their offspring for the next generation. A hypothetical snapshot of the GA evolution is shown in Figure 4. While the initial population is scattered randomly in Ω, the final generation individuals are gathered near the Pareto front.



(a) Group Ranking (b) Individual Ranking

Figure 3: Population Ranking Methods

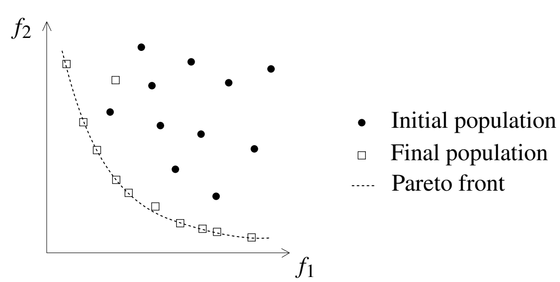


Figure 4: GA Evolution Snapshot

GA-based MCO methods rely on aforementioned strengths from GAs. It is possible to find multiple Pareto optimum at a single execution. Also, one does not need to worry about the shape of the Pareto front. More importantly, it is not necessarily required to have a priori knowledge for a given problem. The knowledge naturally grows from evolutionary process. On the other hand, GA-based MCO methods inherit the disadvantages from GAs as well. Due to the stochastic nature, repetitive executions are needed to ensure reliable solutions. It is again emphasized that GAs require a lot of function calls. In the end, a user often experiences a situation that advantages of a GA-based methods quickly become diluted after recognizing huge amount of function calls.

Therefore, a user should be familiar with the advantages and disadvantages from one-by-one and all-at-once strategy so that (s)he may choose the best strategy and the best method. In some occasions, it may be worthwhile to think of a potential benefit from combining both strategies wisely.

3 Spotlight Search Method

3.1 The Concept and Formulations

The key concept of a spotlight search method (SSM) is illustrated in Figure 5. In this 2-D example, Ω is shown with the boundary $\partial\Omega$ in an arbitrary shape. Suppose a unit vector $\nu = [\nu_1, \nu_2]^T$ with $\nu_1 \cdot \nu_2 \neq 0$ and a line aligned to ν through the origin O. It is shown that the line makes two points A and B intersecting $\partial\Omega$. Then, a criterion vector $F(X)$ lying on the line $\overline{AB}$ should satisfy the following equality constraint.

$$F(X) = t \cdot \nu \quad \text{or} \quad \frac{f_1(X)}{\nu_1} = \frac{f_2(X)}{\nu_2} = t \in \mathbb{R}^1 \tag{7}$$

Obviously, the point A with minimum distance from the origin is a Pareto optimum. From this observation, a crude formulation of SSM can be given as follows by substituting a generalized form of Equation (7) to deal with n criteria.

$$\begin{aligned} &\underset{X \in \Theta}{\text{Minimize}} \;\; F^s = \|F(X)\|_2 \\ &\text{Subject to:} \;\; \frac{f_1(X)}{\nu_1} = \frac{f_2(X)}{\nu_2} = \cdots = \frac{f_n(X)}{\nu_n} \end{aligned} \tag{8}$$

If a user rotates the spotlight vector ν from f_1 axis to f_2 axis, a subset of $\partial\Omega$ facing the origin O can be obtained. It will be called the *front* throughout this paper. In this particular example, the front is portrayed as the separate black curves, $\widetilde{Y_1Y_2}$ and $\widetilde{Y_3Y_5}$. The dashed curves are the remaining portions of $\partial\Omega$ which are invisible from the point O. It can be proven that the Pareto front is always a subset of the front. After the front is identified in a criterion space, a user is able to quickly discard a portion which is not the Pareto front such as the curve $\widetilde{Y_3Y_4}$. This is the basic mechanism of an SSM.

SSM offers another parameter vector to guide a search. It will locate a different intersecting points when the origin of the vector ν is moved from O to another point $\hat{F}$. This is equivalent to translating the reference axes. To do so, a function $\tilde{F}(X)$ is introduced with a goal vector $\hat{F} = [\hat{f}_i]^T$.

$$\tilde{F}(X) = F(X) - \hat{F} \tag{9}$$

Consequently, the surrogate function and the equality constraints in Problem (8) are substituted respectively by:

$$F^s = \|\tilde{F}(X)\|_2 \tag{10a}$$

$$\frac{\tilde{f}_1(X)}{\nu_1} = \frac{\tilde{f}_2(X)}{\nu_2} = \cdots = \frac{\tilde{f}_n(X)}{\nu_n} \tag{10b}$$

where $\tilde{f}_i(X)$ indicates the i-th element of $\tilde{F}(X)$.

A further improvement can be made. Suppose the goal is pivoted on $\hat{F}_w$ in Figure 5. Then, SSM will search the point B like the distance metric method does as explained in Section 2.2.1. To avoid this, the L_2-norm is now dropped and F^s will be substituted with $\hat{f}_k(X)$. The index k can be any number from 1 to n. Equation (10b) needs to be rearranged to fit into a computing code. Incorporating the above modifications will bring the final formulation for SSM as follows:

$$\begin{aligned} &\underset{X \in \Theta}{\text{Minimize}} \;\; \tilde{f}_k(X) \\ &\text{Subject to:} \;\; \tilde{f}_i\, \nu_{i+1} - \tilde{f}_{i+1}\, \nu_i = 0 \\ &\qquad\qquad\;\; \text{for} \quad i = 1, 2, \cdots, n\text{–}1 \end{aligned} \tag{11}$$

By using this formulation, the point A can be searched even if the goal is set mistakenly to $\hat{F}_w$. In fact, the outcome is the same as long as the goal sits on the line $\overrightarrow{OA}$.

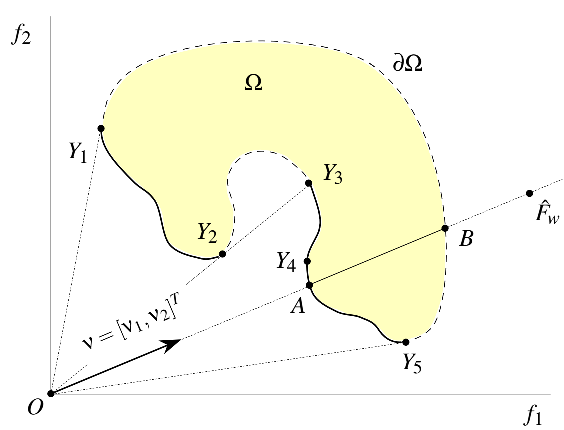


Figure 5: Spotlight Search Method

3.2 Constraint Handling

In all optimization practice, it is an important issue to represent constraints appropriately in a computing code. Obviously, better constraint handling methods result in faster and better solutions. The SSM calls for $(n-1)$ equality constraints which can be presented in a different form by algebraic manipulation. Through a pilot test, it was found that the constraint representation shown in Problem (11) worked well in most cases but numerical instability was encountered when the search vector ν was close to the reference axes. Thus, as a subproblem for this research, an empirical study in 2-D cases was carried out to find the best way to improve numerical stability. The result was simple; if the search vector is close to f_1 axis, then $F_s = f_1(X)$ and the equality constraint H^s takes $\frac{f_2(X)}{f_1(X)}\frac{\nu_1}{\nu_2} - 1 = 0$, which is described in Figure 6. When the search vector is very close to f_1 axis it follows that $\frac{\nu_1}{\nu_2} \gg 1$. The optimizer is minimizing $f_1(X)$, which forces $f_2(X)$ to decrease even more because the equality constraint requires $\frac{f_2(X)}{f_1(X)} \ll 1$. This scheme conceivably contributes to stabilize numerical calculation in a computer and was adopted for all numerical experiments. A generalized SSM problem incorporating the constraint handling technique can be induced to replace Problem (11) as follows:

$$\begin{aligned} &\underset{X\in\Theta}{\text{Minimize}} \quad \tilde{f}_k(X) \\ &\text{Subject to:} \ \frac{\tilde{f}_i}{\tilde{f}_k}\frac{\nu_k}{\nu_i} - 1 = 0 \\ &\qquad\quad \text{for} \ \ i = 1,2,\cdots,n \ (i \neq k) \end{aligned} \tag{12}$$

where k is an index for the reference axis $\tilde{f}_k$ and determined from the following simple steps.

i. Let e_j be a unit vector for the reference axis $\tilde{f}_j$.
ii. Find k such that maximize $|e_j \cdot \nu|$.

Note that special treatment is needed to use Problem (12) when any of ν_j equals to zero. The remedy is quite simple: set $f_j(X) = 0$.

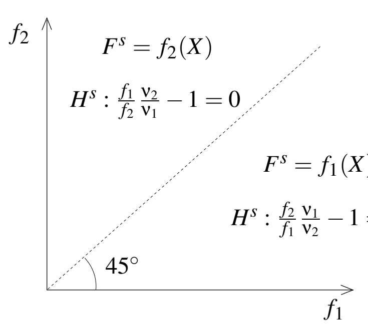


Figure 6: Constraint Handling for 2-D Cases

3.3 Comparison to GAM

The goal attainment method (GAM), proposed by Gembicki,[11] employs the very similar scheme to SSM. A weight vector $W = [w_1, w_2, \cdots, w_n]^T$ should be predetermined by a decision maker in addition to a goal vector $\hat{F}$. The following problem will be solved to generate a Pareto optimum.

$$\begin{aligned} &\underset{\lambda\in\mathbb{R}^1,\, X\in\Theta}{\text{Minimize}} \quad \lambda \\ &\text{Subject to:} \ f_i(X) - \lambda w_i \leq \hat{f}_i \\ &\qquad\quad \text{for} \ \ i = 1,2,\cdots,n \end{aligned} \tag{13}$$

where λ is a scalar variable whose sign tells the goal $\hat{F}$ is attainable or not. The initial value of λ was set to zero in the forthcoming numerical experiments described in Section 4. The vector W exactly corresponds to the spotlight vector ν. Instead of the equality constraints from SSM, GAM has inequality constraints. Even though the two methods share mathematically equivalent elements, they are not exactly the same. In Figure 7, the mechanisms by which these methods operate are portrayed. The goal is set to the origin for simplicity. While SSM will end up with the point Y_S on the front, it is possible for GAM to find the point Y_G. In addition, SSM and GAM differ mathematically in the dimension of the problem. They are contrasted as shown in Table 1.

Table 1: Dimension Comparison

	SSM	GAM
$\dim X^s$	N	$N+1$
$\dim G^s$	0	n
$\dim H^s$	$n-1$	0

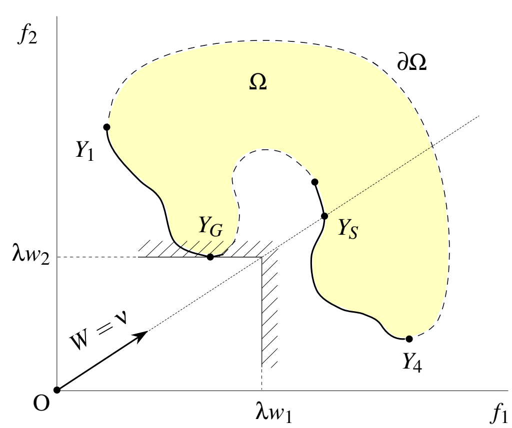


Figure 7: Goal Attainment Method in 2-D Space

Table 2: Description of Test Problems

	Criterion functions	Constraints	Feature
P_1 :	$f_1 = 10\,x_1;\;\; f_2 = \frac{10+(x_2-5)^2}{10x_1}$	$0.1 < x_1 < 1$ $0 < x_2 < 1$	Convex Pareto front
P_2 :	$f_1 = 1-\exp\{-(x_1-1)^2-(x_2+1)^2\}$ $f_2 = 1-\exp\{-(x_1+1)^2-(x_2-1)^2\}$	$-4 < x_1 < 4$ $-4 < x_2 < 4$	Concave Pareto front
P_3 :	$f_1 = b\,\cos a;\;\; f_2 = b\,\sin a$ $a = \frac{\pi}{180}\{50+40\,\sin 2\pi x_1 + 40\,\sin 2\pi x_2\}$ $b = 1+0.6\,\cos 2\pi x_1$	$0 < x_1 < 1$ $0 < x_2 < 1$	Compound Pareto front
P_4 :	$f_1 = x_1;\;\; f_2 = \frac{a}{x_1}$ $a = 2-\exp\{\frac{-(x_2-0.2)^2}{0.08}\}-0.8\exp\{\frac{-(x_2-0.6)^2}{0.4}\}$	$0.1 < x_1 < 1$ $0 < x_2 < 1$	Local Pareto front
P_5 :	$f_1 = x_1;\;\; f_2 = a\cdot b$ $a = 1+10\,x_2;\;\; b = 1-(\frac{f_1}{a})^2 - \frac{f_1}{a}\;\sin 8\pi f_1$	$0 < x_1 < 1$ $0 < x_2 < 1$	Discontinuous Pareto front

4 Numerical Experiments

4.1 Test Problems

In order to benchmark the performance of SSM, five test problems were selected for numerical experiments. These are listed on Table 2. Since the imminent task for this research is to prove the concept of SSM, all test problems were in the simplest form. They each have two criteria and two design variables with side constraints only. However, each problem has been carefully selected to represent diverse features of MCO problems.

Test problem P_1 was found in Deb.[12] It features a convex Pareto front. Test problem P_2 was introduced in Fonseca and Fleming.[13] It has a symmetric concave Pareto front bounded in a region $[0,1]\times[0,1]$. Test problem P_3 comes from Hillermeier.[14] The Pareto front of this problem is composed of a concave part in the center and convex parts at both ends. These three problems are relatively easy to solve and called Group I.

Group II consists of the problem P_4 and P_5. Test problem P_4 was used in Andersson.[15] This problem is a very deceptive one. It has an easy-to-find local Pareto front and a difficult-to-find global Pareto front since $f_2(X)$ has a local optimum at $x_2 \simeq 0.6$ and global optimum at $x_2 \simeq 0.2$ indicated in Figure 8. Test problem P_5 was formulated by Deb.[16] This problems is also difficult to solve since it has a set of discontinuous Pareto front. The feasible criterion space was shown in a shaded region in Figure 9.

The experimental setup is now being described. The optimizer used in the experiments was a built-in function fmincon in MATLAB®. From this tool, useful information on the optimization process was readily obtained such as the total number of function calls, active constraints and convergence status. The numerical experiment for each test problem went through the following steps.

First, an initial guess for fmincon was set to the mid point of each design variable. Next, a goal vector $\hat{F}$ was determined. Then, the optimizer runs SSM and GAM respectively with the vector $\nu = W$ that was given as $[\cos\theta, \sin\theta]^T,\;\; \theta = 1°, 2°, \cdots, 89°$.

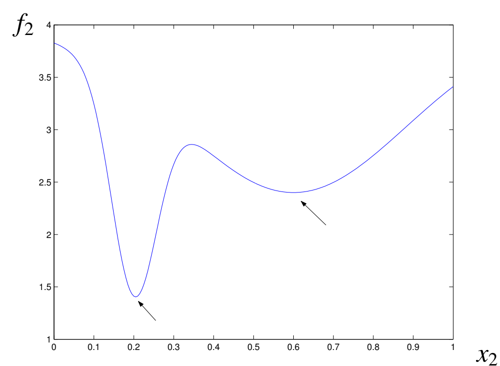


Figure 8: Plot of f_2 against x_2 for P_4

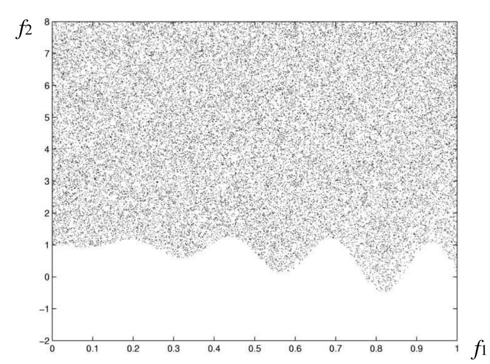
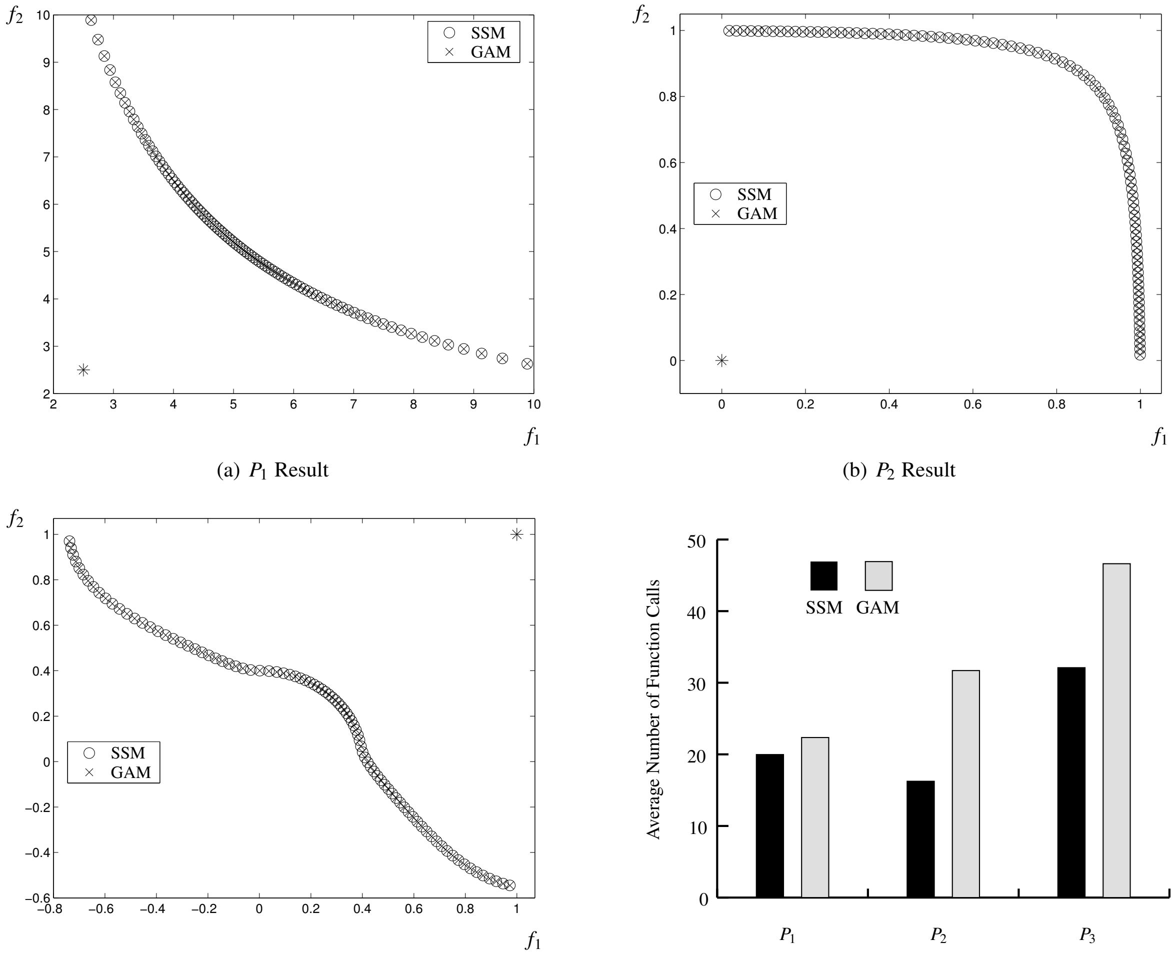


Figure 9: Feasible Criterion Space for P_5



(a) P_1 Result

(b) P_2 Result

(c) P_3 Result

(d) Comparison of Number of Function Calls

Figure 10: Test Result Summary for P_1, P_2, P_3

4.2 Results

4.2.1 Group I: P_1, P_2, and P_3

The problem P_1, P_2 and P_3 were easy to solve. SSM and GAM successfully found the corresponding Pareto fronts. The criterion spaces for all three problems are shown in Figure 10(a), 10(b) and 10(c). In the figures, the circles and the crosses denote converged data points from SSM and GAM respectively. The $\ast$ symbols were marked to indicate the location of the goals. For test problem P_1, the goal vector was located at $[2.5, 2.5]^T$. The goal vector for P_2 was set to the utopian vector, $Y^\star = [0,0]^T$. However, the goal of P_3 was intentionally given at the 'wrong' place $[1,1]^T$, simulating a decision maker's inadequate input due to insufficient knowledge about the criterion space. The result clearly shows that SSM and GAM found the same Pareto optimum and subsequently the identical Pareto fronts regardless of their shape. Under the given result, it is fair to compare the performances of both methods based on the number of function calls. The comparison data is shown in Table 3 and is plotted in Figure 10(d). Throughout all cases, SSM requires fewer number of function calls. Table 4 shows SSM has stable variations as opposed to GAM. Therefore, it is concluded that SSM outperforms GAM for the test problems P_1, P_2 and P_3, which may imply that SSM is invariant with the shapes of Pareto fronts.

Table 3: Average Function Calls

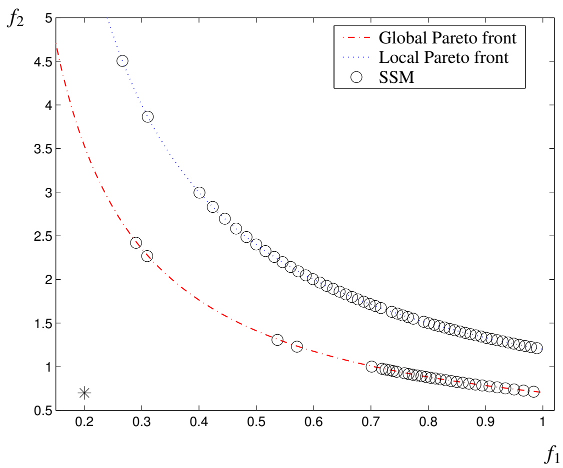
Problem	SSM	GAM	Ratio (SSM/GAM)
P_1	19.97	22.35	89.3 %
P_2	16.24	31.70	51.2 %
P_3	32.10	46.62	68.9 %

Table 4: Standard Deviations

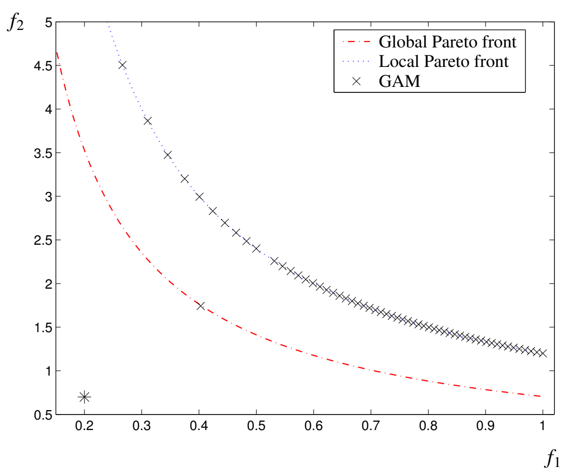
Problem	SSM	GAM	Ratio (SSM/GAM)
P_1	4.20	2.59	162.5 %
P_2	2.40	16.49	14.6 %
P_3	8.01	8.33	96.1 %

4.2.2 Group II: P_4

The test problem P_4 is more deceptive due to the presence of a local Pareto front. Here, one can start to feel a fine distinction between SSM and GAM. The result is plotted in Figure 11. For SSM test, eighty-five executions numerically converged. It is shown in Figure 11(a) that thirty-two solutions correctly indicated the true Pareto front. The remaining fifty-three solutions were located in the local Pareto front. On the other hand, although GAM had no problem in numerical convergence, there was an excessive duplicated solution point; thirty-two solutions out of eighty-nine solutions indicated the same data point $[1.0, 1.2]^T$. Furthermore, all solutions were gathered on the local Pareto front except one solution which can be found at $f_1 \simeq 0.4$ in Figure 11(b). This may mislead a decision maker. Hence, it can be concluded that the SSM is superior to GAM if one recalled the objective of MCO methods.



(a) SSM Result



(b) GAM Result

Figure 11: P_4 Result Summary

4.2.3 Group II: P_5

Another intriguing result was found in this problem. First, all the converged solutions from both method are shown in Figure 12(a). For GAM, all executions numerically converged. But an excessive duplication was identified again; forty-two solutions are on the same point $[0.568, 0.115]^T$. In contrast, seventy-six executions of SSM numerically converged. The converged solutions were located, at least, on $\partial\Omega$ for this problem.

To contrast the results even clearer, solutions on $\mathcal{P}$ were selected and plotted in Figure 12(b) and 12(c). SSM found all four segments of the Pareto front very well. The data points were evenly distributed except the top part of the second segment from the right. However, GAM's result was satisfactory only on the second segment only. More notably, it completely missed the first segment of the Pareto front from the right, which again may mislead a decision maker. Therefore, it is concluded that SSM outperforms GAM for this problem.

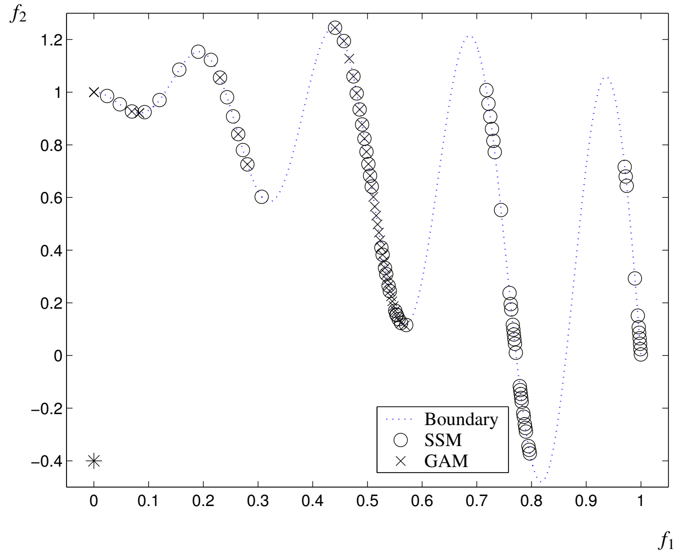
5 Conclusion

This paper presented the spotlight search method (SSM) suitable for continuous, nonlinear MCO problems. The basic concept and formulations of SSM were given with an improved technique to handle the additional equality constraints. The intrinsic advantages of the method are that SSM is not sensitive to the shape of the Pareto front and that it is partially adaptive to inadequate input from a user such as a wrong goal point. The most similar method, the goal attainment method, was identified and benchmarked against SSM. The comparison study contrasted both method in operation mechanism and in the dimension of the problem. SSM has fewer design variables and constraints. Since the equality constraints are more strict than inequality constraints, it is interpreted that GAM may be more robust in terms of numerical convergence but may mislead a decision maker. Numerical experiments concluded that SSM was superior to GAM in all five test problems. Despite the fact that the test cases were restricted in 2-D, SSM showed a lot of potential to search a design space more efficiently. Further study will attempt to deal with higher-dimensional problems. As a future work direction, it is proposed that an investigation on potential benefits from incorporating SSM within the framework of multiobjective genetic algorithms (MOGAs).

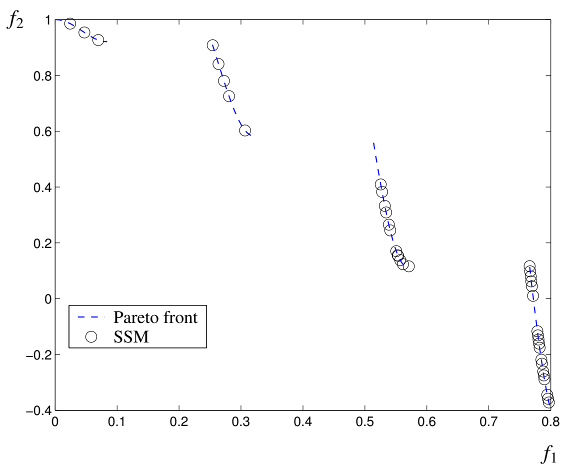
References

[1] Fishburn, P. and Lavalle, I. MCDA: theory, practice and the future. *Journal of Multi-Criteria Decision Analysis*, 8:1–2, 1999.

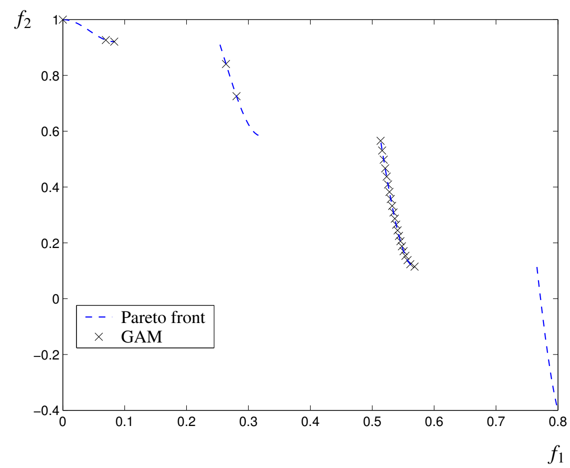
[2] Lootsma, F. The expeced future of MCDA. *Journal of Multi-Criteria Decision Analysis*, 8:59–60, 1999.



(a) Converged Data Points



(b) SSM Result



(c) GAM Result

Figure 12: P_5 Result Summary

3 Miettinen, K. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, MA, 1999.

4 Triantaphyllou, E. *Multi-criteria decision making methods: a comparative study*, vol. 44 of *Applied optimization*. Kluwer Academic Publishers, Boston, MA, 2000.

5 Hwang, C.-L. and Masud, A. *Multiple Objective Decision Making - Methods and Applications*, vol. 164 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, New York, 1979.

6 Chankong, V. and Haimes, Y. *Multiobjective decision making: theory and methodology*. North-Holland, New York, 1983.

7 Ehrgott, M. *Multicriteria Optimization*, vol. 491 of *Lecture notes in economics and mathematical systems*. Springer, Berlin, Germany, 2000.

8 Carmichael, D. G. *Structural modelling and optimization: a general methodology for engineering and control*. Ellis Horwood Limited, Chichester, England, 1981.

9 Goldberg, D. *Genetic algorithms in search, optimization and machine learning*, p. 201. Addison-Wesley, Reading, MA, 1989.

10 Fonseca, C. M. and Fleming, P. J. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Genetic Algorithms: Proceedings of the Fifth International Conference*, pp. 416–423. Morgan Kaufmann, 1993.

11 Gembicki, F. *Performance and sensitivity optimization: a vector index approach*. PhD thesis, Case Western University, 1974.

12 Deb, K. Non-Linear Goal Programming using Multi-Objective Genetic Algorithms. Technical Report CI-60/98, University of Dortmund, Germany, Oct. 1998.

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

14 Hillermeier, C. *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*, vol. 135 of *International series of numerical mathematics*. Birkhäuser Verlag, Berlin, 2001.

15 Andersson, J. and Krus, P. Metamodel representations for robustness assessment in multiobjective optimization. In *Proceedings of*

the 13^{th} *International Conferencce of Engineering Design*, Glasgow, UK, August 2001.

[16] Deb, K. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 7(3):205–230, 1999.