

NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:

The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

OPTIMIZATION FOR DESIGN PROBLEMS HAVING MORE
THAN ONE OBJECTIVE

by

A.W. Westerberg & P.A. Clark

December, 1962

DRC -OG-40-32

Peter A. Clark and Arthur W. Westerberg

Department of Chemical Engineering and Design Research Center
Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213

Abstract. This paper presents a class of design optimization problems where an outer optimization is constrained by an inner one. In the first part of the paper, we show that a major subclass of this type of problem is the multiple criteria (vector-valued) decision making problem. This subclass was the requested review topic for this paper; we discuss it extensively. The paper concludes by discussing the more general problem, indicating its importance to design and exploring its potential very unpleasant nonconvex behavior. We introduce a single level solution algorithm that shows promise for finding a local optimum very quickly.

INTRODUCTION

This paper is the result of a tactical error made by the second author when writing a review paper (Westerberg, 1980) on design optimization for the FOCAPD meeting held in Henniker, NH. Included in that paper was a brief summary of some interesting Ph.D. work (Lightner, 1979) on solving multiple criteria decision making problems. That work was just completed in the Electrical Engineering Department/Design Research Center at Carnegie-Mellon University. It was particularly interesting because it gave an approach that allowed the design engineer to locate a solution to this class of problems close to a desired solution by using parameters that had intuitive meaning for him. With that demonstration of a small and probably dangerous amount of knowledge of the area, the organizers for this conference selected this topic for us to review. Seeing no reason not to explore the area, the challenge was accepted, but it was indeed a challenge.

It appeared for some time that the multiple criteria decision making problem was related to another problem we identified three years ago and have been working on for the last 18 months at CMU, the class of design optimization problems which are constrained by an inner optimization problem. Indeed the connection can be made directly because the multiple criteria decision making problem is a special case of this more general class of problems, albeit one with a "fuzzy"¹¹ outer objective function that is known only intuitively by the design engineer. Therein lies its peculiar flavor, and the task at hand is to provide a convenient means to extract the designer's optimum point from him in the most painless manner.

¹This work was funded in part by NSF grant ENG-7801809.

We will show that the inner problem, one with many competing but not comparable objectives (apples versus oranges) has its own mathematical difficulties that have been the theme of many papers. Its solution is a family of points, each with the characteristic that no objective can be improved without a loss being incurred in one of the competing objectives. Only these so called "noninferior" points are presented to the designer. This inner problem is usually recast as a single objective (that is, a traditional) optimization problem to be solved in terms of a set of "weights"¹¹ that parameterize it. The weights are varied systematically to locate the noninferior point most preferred by the designer. How to vary them based on fuzzy responses from the designer is one of the interesting aspects to this problem.

A variation on this approach is to get the designer to agree that some of the objectives need not be optimized but simply brought to a satisfactory level, in which case they can be recast as constraints and the problem size reduced accordingly.

Special problem structures, particularly linear ones, give rise to clever solution algorithms. We shall explore briefly a few of these also.

The last part of the paper will consider the more general problem of an outer optimization problem constrained by inner ones, i.e. an embedded or multilevel optimization. We shall first describe an important engineering design problem with this structure. This problem unfortunately does not have to be well-behaved. We will show that it can possess multiple local optima, in the usual tradition of believing that most real problems are usually well behaved, we present an algorithm that appears to locate a local

optimum with the efficiency of the recent successive quadratic approximation algorithms (Wilson (1963), Han (1975)).

The Embedded Optimization Problem

We shall start by defining the following general problem which is the theme of this paper.

Problem P1: $\text{Min } \hat{x} \wedge \hat{x}$
 \hat{x}_1, \hat{x}_2
 s.t. $g_1(\hat{x}_1, \hat{x}_2) \leq 0$
 $h_1(\hat{x}_1, \hat{x}_2) = 0$
 $\hat{x}_2 = \arg \min_Y \hat{x}(x_1, Y)$
 s.t. $g_2(x_1, Y) \leq 0$
 $h_2(x_1, Y) = 0$

This problem has an inner optimization problem which constrains the outer one. The variables \hat{x} are constrained to be the value or values of f that minimize $\langle L(\hat{x}) \rangle_Z$ subject to the inner problem inequality and equality constraints. They are implicitly then a function of \hat{x}_1 which parameterizes the inner problem.

We shall now show that the multiple criteria (multiple objective, vector valued, (MCDM) is a special case of (Pi). The normal statement of the MCEM problem is as follows.

Problem P2 (MCDM):

$$\text{Min } (f_1(z), f_2(z), \dots, f_q(z))$$

$$\text{s.t. } f(z) \leq$$

$$h(z) = 0$$

Clearly a dilemma exists here as we are asking that several objective functions, f_1, \dots, f_q , be simultaneously minimized. If the same point z simultaneously minimized all objectives, the problem would be solved. This event will rarely happen.

The "solution" to P2 is a family of solutions called a "pareto" optimal surface or the set of "noninferior" or "efficient" solutions. Each such solution has the property that it is not possible to improve any one of the objectives without simultaneously degrading the value of another. Figure 1 illustrates this idea for two objectives. Note the coordinates for the space are f_1 and f_2 , not z . The encircled area R is all points (f_1, f_2) reachable by all feasible choices of z . Point "a" is not noninferior since both f_1 and f_2 can be reduced, such as by moving to point "b". Point "b" on the other hand is noninferior because, to reduce either f_1 or f_2 , one would have to increase the other objective. All points from "c" to "d" along the lower edge of R are noninferior; they are the pareto optimal surface which "solves" this problem.

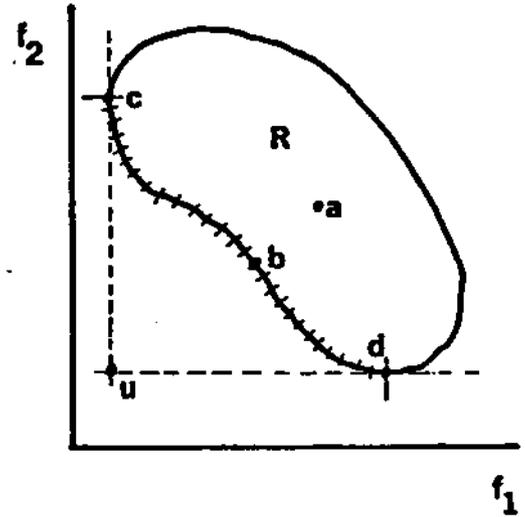


Fig. 1 Noninferior surface from point "c" to "d". "u" is the Utopia point.

The surface of noninferior solutions, (f_1, f_2, \dots, f_q) implicitly defines a function

$$G(f_1, f_2, \dots, f_q) = 0$$

in f_1, f_2, \dots, f_q space.

These points as found are shown to the decision maker, known in MCDM problems as the "decision maker" (DM), who must select his preferred point according to his internal "utility" function. The outer problem is therefore as follows.

Problem P3:

$$\text{Min } U(f_1, f_2, \dots, f_q)$$

$$\text{s.t. } G(f_1, f_2, \dots, f_q) = 0$$

As the CM (decision maker) can change his mind or forget past preferences, his utility function is a fuzzy one.

Solving problem (P2) to find a solution (f_1, f_2, \dots, f_q) satisfying

$$G(f_1, f_2, \dots, f_q) = 0$$

is always done by converting problem (P2) to a scalar optimization problem. Examples of the form of problem solved are as follows.

Problem P4:

$$\text{Min } \sum v_i f_i(z)$$

$$\text{s.t. } g(z) \leq 0$$

$$h(z) = 0$$

Problem P5 (Min Max):

$$\begin{aligned} & \text{Min} && Y \\ & Y.f && \\ & \text{s.t.} && v_1 f_1 \leq \gamma \\ & && v_2 f_2 \leq \gamma \\ & && w f \leq Y \\ & && q q \\ & && g(z) \leq 0 \\ & && h(z) = 0 \end{aligned}$$

or

Problem P6 (Variation of the c-Constraint Formulation):

$$\begin{aligned} & \text{Min} && f_1(z) \\ & \text{s.t.} && v_2 f_2 \leq 1 \\ & && v f \leq 1 \\ & && q q \end{aligned}$$

$$h(z) \leq 0$$

To find different solutions, one selects different values of $w_1, w_2, \dots, w_2 = 0$. If we use inner problem formulation (P5) as our example, the complete MCDM can be stated in the form

$$\begin{aligned} & \text{Min}_{w, z} && U(f(z)) \\ & \text{s.t.} && y \leq 0 \\ & \text{s.t.} && Y, z \text{ "ArgMin } y \\ & && Y, z \\ & \text{s.t.} && w_1 f_1(z) \leq \gamma \\ & && v_2 f_2(z) \leq \gamma \\ & && g(z) \leq 0 \\ & && h(z) = 0 \end{aligned}$$

where now we should be able to recognize this problem to be a special case of the form of problem (pi) by identifying

$$\begin{aligned} U & \leftrightarrow \phi_1 \\ w & \leftrightarrow z_1 \\ Y & \leftrightarrow z_2 \\ Y, z & \leftrightarrow x_2 \end{aligned}$$

We shall now consider the MCDM (Multiple Criteria Decision Making) problem in depth. After this review we shall return for a more detailed look at the general problem (Pi) to observe its characteristics.

The Multiple Criteria Decision Making (MCDM) Problem

We shall consider two special cases for the MCDM problem:

1. When the inner problem (P2) gives a

convex function $G(f_1, \dots, f_q)$ in the $(-1, -2, \dots, f_q)$ space.

2. When (P2) gives rise to a nonconvex function $G(f_1, \dots, f_q)$.

The main issues are three

1. How to solve problem (P2) to find different noninferior solutions. Even formulations (P4), (P5) and (P6) have their problems.
2. How to elicit from the DM (decision Baker) information on his utility function.
3. How to use the information elicited in step 2 to find the noninferior solution preferred by the DM.

To place the problem in perspective, we provide Fig. 2 which illustrates the component parts of a programming system for solving MCDM problems. We need to examine the contents of the top two "boxes" labeled 1 and 2, respectively.

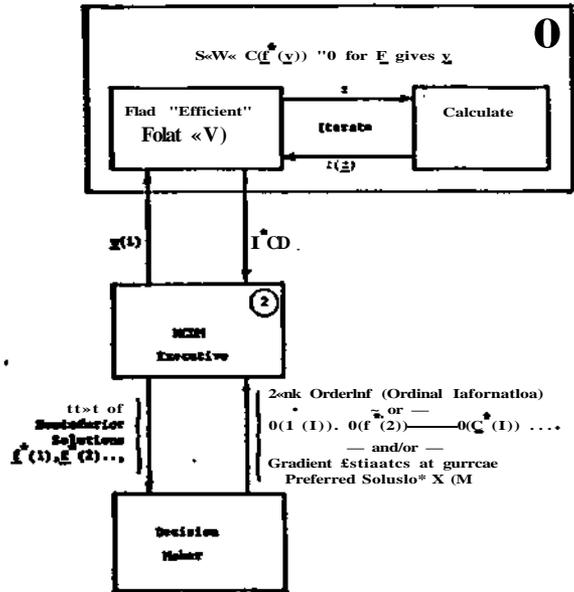


Fig. 2 Structure of a Programming System for MCDM Problems

Several excellent review articles exist to cover a literature on this subject that is already extensive. One of the easiest to read is by Hwang et al (1980), a tutorial paper referencing 61 earlier works. This paper in turn points to an earlier review by the same authors (Hwang et al (1979)) where over 400 papers and 26 books are cited.

Zlonts (1979), a name which appears repeatedly in this literature, gives his views on the subject in a paper referencing 46 articles. Lightner (1979), in Chapter 2 of his Ph.D. thesis, cites 61 references. These

three references alone will point one to much of the relevant literature and indirectly to almost all of it prior to 1979 or 1980. For the hardy among you, the review by Stadler (1979) gives a "interesting" view of the papers prior to 1970 on the subject.

Within the chemical engineering literature, the first paper found was by Seinfeld and MacBride (1970), who applied ideas developed in earlier work by Geoffrion (1967) to set up and solve a process example.

The Inner Problem (P2)

As illustrated in Fig. 1, the problem (P2) is to minimize a vector of "q" competing objectives, $f_1(z), f_2(z), \dots, f_q(z)$. As we said earlier, not all can usually be minimized at the same point, and we are thus led to the concept of noninferiority (also known as efficiency, pareto optimality, minimality, and nondominance - see Lightner (1979)). The problem is to provide an algorithm to locate noninferior points.

If the region R in Fig. 1 is convex then the most common approach taken is to formulate a scalar optimization problem of the form (P4):

$$\begin{aligned} \text{Min} \quad & \sum_{i=1}^q w_i f_i(z) \\ \text{s.t.} \quad & f(z) \leq 0 \\ & h(z) = 0 \end{aligned}$$

where w_i are selected to satisfy $w_i \geq 0$ for all i and $\sum_{i=1}^q w_i = 1$. Geoffrion (1967) has

shown that all noninferior points can be found by a suitable choice of the weights w . Figure 3 illustrates.

Even if R is not convex, this algorithm can locate points which are not noninferior (Lin (1976), Gearhart (1979), Benson (1979)) as Fig. 4 illustrates. One can avoid finding these points by requiring all $w_i \geq \epsilon > 0$ (Geoffrion (1967)).

Geoffrion (1967) defines the subtle concept he terms "proper efficiency" to exclude any point from being noninferior if one can in the limit improve one objective infinitely quickly while degrading another a finite amount. Consider the example

$$\begin{aligned} f_1(z) &= -z^2 \\ f_2(z) &= z^3 \\ g(z) &= -z \leq 0 \end{aligned}$$

We can ask if the origin $z=0$ is a point with proper efficiency. If z is small and positive, the gain in $f_1(z)$ with respect to the loss in $f_2(z)$ is in the ratio $1/z$, which in the limit as z tends to zero becomes arbitrarily large. The origin is not a point with proper efficiency. The point $z = -c > 0$

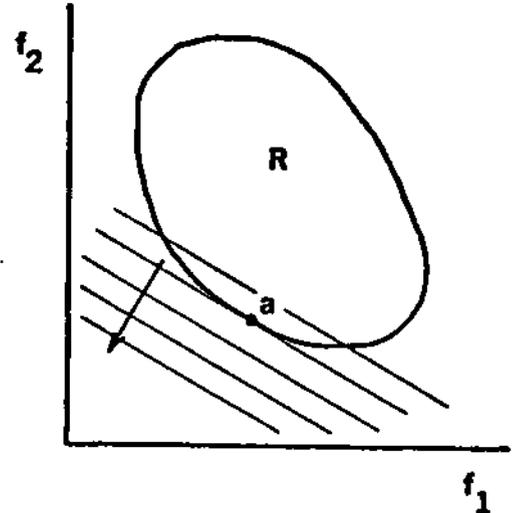


Fig. 3 Finding noninferior points using problem formulation P4. Lines of the form $v_1 f_1 + v_2 f_2 = k$ shown for different values of k . Direction of decreasing k shown. Point "a" is a noninferior point found for w_1, w_2 chosen.

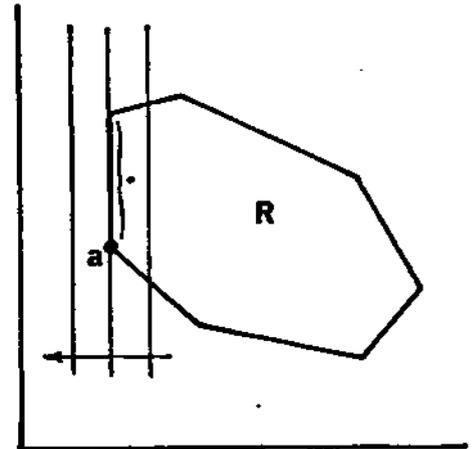


Fig. 4 A problem with formulation P4. Vertical lines are of form $f_1 + 0 f_2 = k$. Direction of decreasing k shown. All points indicated by "*" solve problem P4 with $w_1 = 1, w_2 = 0$, but only "a" is noninferior.

for any given c is however properly efficient - i.e. noninferior. The points marked with an "*" in Fig. 4 except point "a" are not properly efficient.

If region R is not convex in Fig. 1, then problem formulation (P4) can miss whole stretches of noninferior points as indicated by Fig. 5. Illustrated are curves of the form $v_1 f_1 + w_2 f_2 = k$, for different w_2/v_1 values. Problem (P4) moves each curve to

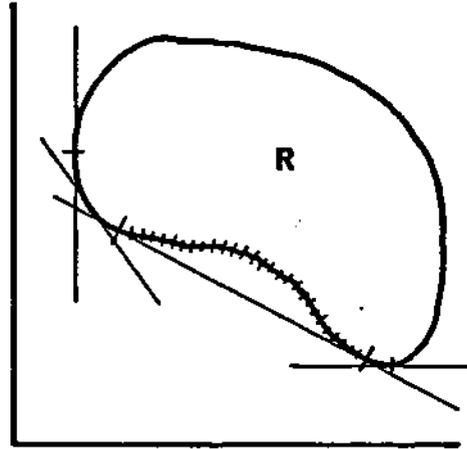


Fig. 5. Missing noninferior points using problem formulation P4. The entire cross hatched region cannot be found using formulation P4.

the left and down as far as possible, such that the curve just supports the region R. The support point is the sought after noninferior point, and clearly the crosshatched segment of the lower left edge of R can never be supported.

To locate points in the crosshatched area of region R in Fig. 5, Lightner (1979) has suggested searching using a weighted p-th norm as the objective ($1 < p < \infty$).

Problem P7:

$$\begin{aligned} \text{Min}_z & \left(\sum (w_i f_i)^p \right)^{1/p} \\ \text{s.t.} & f_i \leq \bar{f}_i \\ & h(z) = 0 \end{aligned}$$

While this norm with $p=2$ has been proposed earlier by others (e.g. Huang (1972), Salukvadze (1974)), Lightner seems to be the first to use it expressly to find the noninferior points for a nonconvex region R. Figure 6 illustrates how the choice of p affects the shape of the "support" surfaces used to locate the noninferior points. Problem (P7) is equivalent to moving the shape function selected to the left and down as far as possible until it just touches region R. Using a surface that bends as illustrated will permit more noninferior points to be found than the linear case when $p=1$. In the limit of $p \rightarrow \infty$, all noninferior points can be located.

The Utopia Point

To use the above approach all $f_i(z)$ must be guaranteed to be non-negative. We can accomplish this by first finding the so-called "utopia" point. The coordinates of the utopia point are shown in Fig. 1; the coordinates are $(f_1^*, f_2^*, \dots, f_n^*)$ where each f_i^* solves the following scalar optimization

problem.

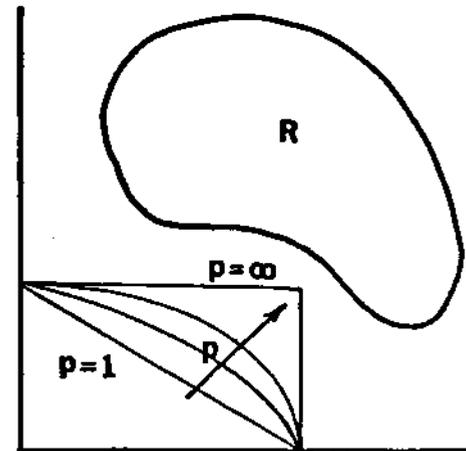


Fig. 6 The Shape of Different Support Surfaces for the p-norm objective.

Problems P8:

$$\begin{aligned} \text{Min}_z & f_i \\ \text{s.t.} & g(z) \leq 0 \\ & h(z) = 0 \end{aligned}$$

We must therefore solve q optimization problems of the form (P8), for $i=1,2,\dots,q$, to locate the coordinates of the Utopia point. The objectives can then be modified so each is always non-negative by making the Utopia point the origin of the (f_1, f_2, \dots, f_q) space, i.e. by letting each objective f_i be replaced by $C - f_i$.

The Infinite Norm

When $p \rightarrow \infty$ is used in problem (P7), the problem is a min max problem where the objective can be rewritten

$$\text{Min}_z \left\{ \max_i w_i f_i(z) \right\}$$

a problem which is usually solved by rewriting in the form of problem (P5).

Even for this problem formulation, points can be found which solve (P5) that are not "noninferior." For example the points marked with an "*" in Fig. 4 will solve (P5) for a suitable choice of weights w_i .

The major advantage to the infinite norm formulation (P5) is that the weights w_i have physical meaning for the DM. They can be set to the reciprocal of the values for each objective that the DM would like the sought after noninferior point to be most like. For example suppose the DM has discovered that maximizing profit yields a solution of \$250,000/yr at a production rate of 5000 units per year. He also discovers that

maximizing production rate yields a profit of \$150,000 when producing 6500 units per year. He would like the next solution to be a compromise most like a profit of \$225,000/yr. By choosing $w_1 < 1/225,000$ and $w_2 > 1/6000$, he will find the solution (usually noninferior) most like that sought.

He can see why this occurs by examining Fig. 7. The origin is the Utopia point. The

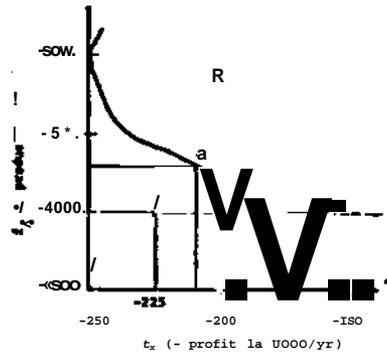


Fig. 7 Physical meaning to weights for problem formulation (P5). "a" is point found using formulation (P5). "b" is point found using formulation (P6).

rectangular shape support function will grow along the line passing from the origin through the desired compromise point. Where it just supports region R at point "a" is the solution found, which in this sense is most like the one sought.

The c Constraint Method (Haines (1970))

Formulation (P6) given earlier also allows one to generate noninferior solutions for nonconvex regions R. It too can find solutions such as those marked by an "*" in Fig. 4 which are not noninferior. Here the weights can also be chosen to locate a desired solution. Let us refer back to our example of the last two paragraphs. Suppose the DM would like the noninferior solution such that profit is maximized while keeping the production at no less than 6000 units per year. Choosing w_1 equal to $(-1/6000)$ and solving formulation (p6) will locate the solution marked "b" on Fig. 7.

Another formulation is possible which combines several earlier ideas (Wendell and Lee (1977)). By adding constraints to formulation (P4) of the form

$$f_1 \leq \bar{f}_1$$

where the point (\bar{f}_1, \bar{f}_2) is a point inside region R and solving (P4) with all

$w_i = 1/q$ (so $\sum_{i=1}^q w_i = 1$), one can also discover all noninferior points. This

particular formulation will find only noninferior points, but has the disadvantage that one must discover a sequence of points f within region R. Figure 8 illustrates the idea. The points found will be support points for the feasible region, as we argued earlier. Because all w_i are greater than zero, only noninferior points will be found.

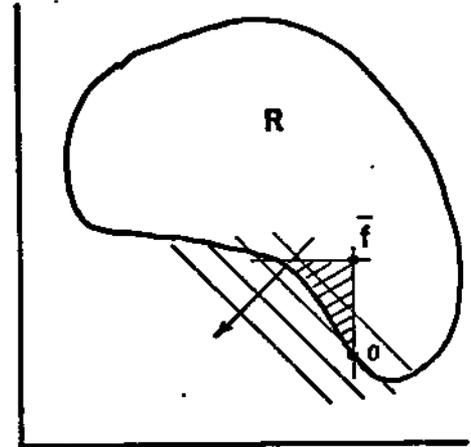


Fig. 8 Finding noninferior points using formulation (p4) with constraints and equal weights. Crosshatched area is constrained area, "a" is support point for reduced feasible region. 45° support lines are for $0.5(f_1 + f_2) - k$ for different k. Direction of decreasing k shown.

A Fix

There appears to be an expedient "fix" for preventing formulations (P5) and (P6) from locating points which are not noninferior. This "fix" was found in an article by Shimizu (-). It is to augment the objective with the term

$$c \sum_{i=1}^n v_i f_i \quad (\text{all } v_i > 0, \quad t_i \geq 0)$$

For example, the objective for (P5) becomes

$$Y + c \sum_{i=1}^n v_i f_i$$

where c is a very small number (say 10^{-8} the size expected for Y). Y will still dominate in selecting the solution point. The points marked with an "*" in Fig. 4 all result in the same Y value. However, the extra term will then come into play and select the one with least value to the sum of the weighted objectives, i.e. the noninferior point labeled "a".

While the "fix" is interesting it is not without risk. It too will find points which are not noninferior if Y can be increased while decreasing the term $c \sum_{i=1}^n v_i f_i$ by exactly the same amount or a bit more.

One could consider the odds in ones favor here, however. The exactly vertical line is to be expected; the above is unlikely. It must be the case that many problems will allow one objective to reach a minimum for a range of values for the other objective. For example the minimum heating and cooling utility usage of a process is a fixed quantity under appropriate assumptions. Several different heat exchanger configurations can usually be developed that feature this minimum usage, each differing in investment cost. Thus if one chose utility usage as one objective and investment cost as the other, then one would discover several different investment costs will yield the same minimum for utility usage. This is exactly the type of problem giving the vertical part of the curve in Fig. 4.

A Decomposition

Takama and Umeda (1980) suggest that the interior problem of locating noninferior points can be solved by decomposing it for typical engineering problems. They partition the variables \underline{z} into the disjoint sets $(z_1, z_2, \dots, z_q, \underline{v})$ such that the MCDM problem can be written:

$$\begin{aligned} \text{Min } & (f_1(\underline{z}_1, \underline{v}), f_2(\underline{z}_2, \underline{v}), \dots, f_q(\underline{z}_q, \underline{v})) \\ \text{s.t. } & \left. \begin{aligned} g_i(\underline{z}_i, \underline{v}) &\leq 0 \\ h_i(\underline{z}_i, \underline{v}) &\leq 0 \end{aligned} \right\} i=1, \dots, q \end{aligned}$$

Their approach relies on the assumption that most engineering problems will have this structure. The suggested approach to solving is illustrated in Fig. 9.

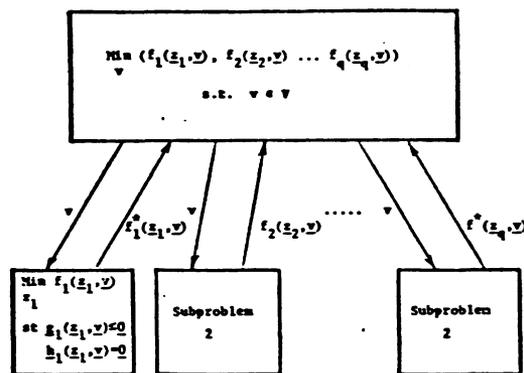


Fig. 9 Decomposition Scheme of Takama and Umeda (1980)

Being suggested is that the top level adjusts \underline{v} , passes \underline{v} to each lower level problem which finds its own $f_i^*(\underline{z}_i, \underline{v})$ as a scalar optimization, over \underline{z}_i . It adjusts \underline{v} again and continues until the DM likes the solution.

The decomposition idea is intriguing. However a problem not discussed is how to

select variables \underline{v} such that all the subproblems will remain feasible. Passing back an infinite value for $f_i^*(\underline{z}_i, \underline{v})$ when subproblem i is infeasible is possible but forces one into using rather inefficient pattern search optimization algorithms.

The MCDM Executive

So far we have been examining the problem of finding a noninferior solution given a set of weights. The next aspect of solving MCDM problems is to consider how the weights are selected, a job of the box we labeled the MCDM executive in Fig. 2. As can be seen in that figure, the executive must interact with both the subsystem that locates noninferior points and the decision maker. Several approaches exist for constructing the executive. The nature of the overall problem (is it linear, is region R convex, etc.) affects what will be in the executive.

The simplest algorithm in concept is to generate a sufficient scattering of noninferior points so as to define the entire noninferior surface completely. Clearly this approach is likely the one maximizing the work done. For more than two objectives, it also is likely far too much work because the problem size grows as n^{q-1} , where n is the number of points over which each of the objective functions is "gridded". For example if 10 different values of each f_i are desired and there are $q=4$ dimensions, one would be required to generate on the order of 10^3 noninferior points. For only two objectives, a total search is feasible and frequently used.

If problem formulation (P4) is used (objective is $\sum w_i f_i$), one difficulty is choosing the weights w_i so as to distribute the noninferior points evenly throughout the (f_1, f_2) space. Thus formulations (P5) and (P6) are likely superior for this task.

Once the entire surface is generated, it can be presented to the DM who can decide for himself which of all the points generated he prefers. Clearly an advantage is that he can change his mind as he looks at the results, or even from one day to the next.

All other approaches must in some sense seek solutions in a sequential manner, by generating a few noninferior points, asking the DM for his preferences, generating a few more points, asking again, etc. We now consider this interaction with the DM.

Interaction with the Decision Maker

In scanning through several articles, it would appear interaction with the decision maker can take several forms which are listed here.

- 1) The DM can be shown a single noninferior point (and perhaps the coordinates of the

Utopia point for reference) and asked one or more of the following

- a) which objectives must still be reduced and which may be increased in exchange
 - b) for those that can be increased, how far can they be increased
 - c) for the objective that is to be decreased, what rate of trade is the DM willing to accept in terms of decreasing f_j while increasing f_i (i.e. $-Af_i/Af_j$). An example would be to say a 5% decrease in f_i is allowed if it can be had for no more than a 1% increase in f_j .
- 2) The DM can be shown two or more noninferior points and asked to do one or more of the following:
 - a) rank order the points
 - b) suggest a desired solution
 - c) pick a "region" within which to place more points
 - d) for the best point, respond to the questions asked in (1) above.
 - 3) the DM can be asked apriori if some of the objectives take precedence over others. The purpose here would be to permit the most important objectives to be dealt with first, before considering those of lesser importance.

Alternative MCDM Executives

We shall now examine several approaches to the construction of the MCDM Executive block of Fig. 2. We shall start with the most widely covered problem type, the all linear problem. For an all linear problem (linear objective functions, linear constraints), the region R in Fig. 1 is convex and problem formulation (P4) is suitable for locating noninferior solutions.

Perhaps one of the most interesting algorithms is that of White (1980), which is a variation of the method proposed earlier by Wallenius and Zionts (1977). It is based on solving a series of linear programs of the form of (P4), specifically

$$\text{Min } \sum_{i=1}^q v_i (C_i^T z)$$

$$\text{s.t. } \begin{aligned} Az &\leq b \\ z &\geq 0 \end{aligned}$$

The MCDM executive directs the values of the weights, v_i to be used when searching for the decision maker's preferred solution. The approach is to pick arbitrarily any w which satisfies the current set of constraints on w . With each iteration a new constraint is added to the set further restricting the choice possible for the weights. Termination occurs when one can

demonstrate that all the allowed w 's yield the same solution point. The algorithm, very roughly, is as follows.

1. Define the constraint set on w to contain at least the following constraints

$$w^0 = \begin{cases} \sum v_i = 1 \\ v_i \geq 0 \end{cases}$$

White reposes this constraint set by listing the points in w space which "generate" the constraint set. For example feasible w 's subject to the above constraints are generated by the extreme points

$$w(1) = (1.0, 0, \dots, 0)$$

$$w(2) = (0, 1, 0, \dots, 0)$$

$$w(q) = (0, 0, 0, \dots, 1)$$

All feasible w 's are then convex combinations of these extreme points. Figure 10 illustrates for a problem with $q=2$ objectives.

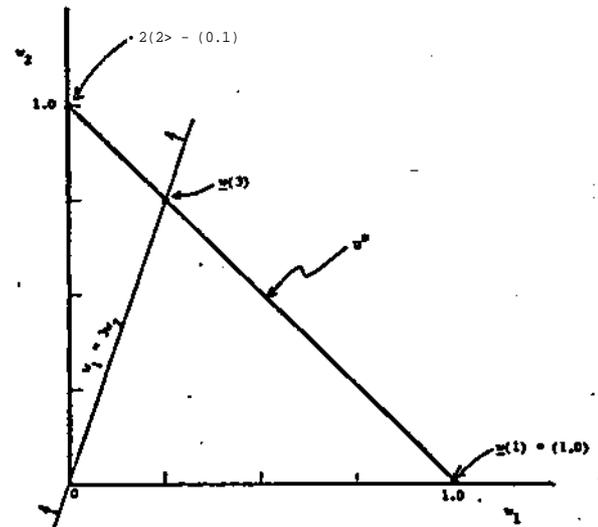


Fig. 10 Method of Wallenius and Zionts (1977) as implemented by White (1979). Initial space of feasible w 's is w^0 , the line joining $w(1)$ to $w(2)$.

2. Choose a w from the set of extreme points and generate a first noninferior point. Call this point the current best solution.
3. Choose another w and generate another noninferior point.
4. Show the latest noninferior point and the current best point to the DM.
5. If the current point is better add the

constraint

$$\left(\sum_i v_i f_i \right)_{\text{new point}} - \left(\sum_i v_i f_i \right)_{\text{Current best point}} \leq 0$$

to the set restricting the choice of w .
Rewriting this constraint gives

$$\sum_i (\Delta f_i) w_i \leq 0$$

If the current point is worse, add

$$\sum_i c_i$$

and if deemed equal by the DM, add

$$\sum_i (\Delta f_i) w_i = 0$$

Suppose in Fig. 9, the noninferior points generated by $w(1)$ give $(f_1, f_2)(1) = (10, 20)$ and by $w(2)$, $(f_1, f_2)(2) = (15, 5)$. Suppose the DM prefers point (2). Then the constraints

$$(15-10) w_1 + (5-20) w_2 \leq 0$$

is created. It is plotted in Fig. 10.

6. White shows that the new extreme point set is easily found by finding the intersection of this new constraint with the lines joining the disfavored point to the other extreme points, here generating $w(3)$. The disfavored point $w(1)$ is deleted from the set.
7. Repeat from step 3 until all extreme points for the space defining the feasible region for w can be shown to yield the same noninferior point.

Note that in this approach, the DM is asked only to compare noninferior points and rank order them.

The STEM method of Benayoun et al (1970, 1971, 1971) is another method developed for a linear problem. The RESTIM method by Shimizu (-) seems to be quite similar. The approach would seem to extend directly to nonlinear problems.

The algorithm starts by locating the Utopia point. It then uses problem formulation (P5), where the Utopia point is treated as the origin in the (f_1, f_2, \dots, f_n) space. The weights w are set as follows **

$$w_i = \frac{\alpha_i}{\sum \alpha_i}$$

where

$$\alpha_i = \frac{|\text{Range of } f_j|}{\langle \Gamma \rangle} \times \frac{1}{\left(\sum c_{ij}^2 \right)^{1/2}}$$

The range of $f_i \cdot \sum_j c_{ij} z_j$ values is the maximum

f_i value, f_i^{max} , among those found when discovering the Utopia point less the minimum f_i value, which is f_i^* , the value at the Utopia point. (While Benayoun suggests maximizing each f_i to define its range, it seems more reasonable to use the maximum of f_i as above.)

Note that a large range, small cost coefficients, and a small maximum cost, f_i^{max} , each lead to a larger weight. One detects a smattering of reasonable arbitrariness here.

Next the noninferior point $(z(m))$ is found using the weights w_i . The DM is asked how he likes each of the objective function values for that point. He can respond by saying each objective function value is or is not satisfactory. For each satisfactory one f_j , he is asked how large an increase, Δf_j , he would tolerate in f_j in order to improve the unsatisfactory objectives. For each satisfactory response, the following constraint is then added to the definition of the original problem:

$$f_j \leq f_j(z(m)) + \Delta f_j$$

For each unsatisfactory one, one adds

$$f_j \leq f_j(10)$$

Finally the weight w_j for each satisfactory objective is set to zero, and a new iteration starts with the locating of the next noninferior point using the modified weights and new constraints.

Mukai (1980) presents an algorithm for nonlinear problems which is related to the above ideas. As above he also suggests asking the user which objectives can be compromised so as to improve the others. His scheme also deletes them from the set considered, with a bound placed on the extent to which these objectives can be compromised.

The next approach to be discussed is by Srinivasan and Shocker (1973). The description here follows that given by Zionts (1979). The approach is valid for both linear and nonlinear problems.

The idea is to locate a target set of "ideal" objective function values as well as a set of weights. Each actual noninferior point, f_i , is characterized by the "distance" from the unknown ideal point, \hat{f}_i :

$$d_{ij}^2 = \sum_{i=1}^q w_i (f_{ij} - \hat{f}_i)^2$$

The noninferior points discovered so far are presented to the DM pairwise for all possible pairs. The following constraint can be written for each such comparison

$$d_{ij}^2 - d_{ik}^2 \begin{cases} \leq 0 & \text{if point } j \text{ preferred} \\ = 0 & \text{if no preference} \\ \geq 0 & \text{if point } k \text{ preferred} \end{cases} \quad (1)$$

A little algebra rewrites (1) as

$$\sum_i w_i (f_{ik}^2 - f_{ij}^2) - \sum_i (w_i \hat{f}_i) (f_{ik} - f_{ij}) \begin{cases} \leq 0 \\ = 0 \\ \geq 0 \end{cases} \quad (2)$$

which is linear both in w_i and the product $w_i \hat{f}_i$. Remember the f_{ij} , f_{jk} values are known information about noninferior points j and k .

Because the DM is likely to give inconsistent answers, each inequality has a slack variable $s_{jk} \geq 0$ added which represents the violation of the constraint. For example, an equation of the form of (2)

$$a_{jk}^T w + b_{jk}^T v \geq 0$$

would be written

$$a_{jk}^T w + b_{jk}^T v + s_{jk} \geq 0$$

The objective is then

$$\text{Min } \sum_{j,k} s_{jk}$$

subject to the constraints written. Note that variables $w \wedge f$ but v can be of any sign. This problem can be solved as a linear program. Once w and v are found, the next point to be searched for is the one that minimizes d^2 for it. It appears one would need to solve a quadratic programming problem to find that next point if the original problem is entirely linear.

An interesting variation might be to use f_i as the target set of objectives and then apply the ideas of Lightner (1979). As we noted earlier, letting $v \wedge 1/\wedge$ and formulating the problem in the form of problem (P5) would seek that solution.

Another of the approaches is called the surrogate worth trade-off (SWT) method (Halines and Hall (1974)). In this approach, useful for nonlinear problems as well as linear, one views the problem to be of the form (P3) stated earlier:

$$\text{Min } U(f_1, f_2, \dots, f_q) \\ \text{s.t. } G(f_1, f_2, \dots, f_q) = 0$$

Remember that the surface of noninferior solutions (f_1, f_2, \dots, f_q) implicitly defines the constraint function $G(f_1, \dots, f_q) = 0$ in (f_1, f_2, \dots, f_q) space. He can write the Lagrange function for (P3):

$$L(f_1, \dots, f_q, H) = U - \mu G$$

Necessary conditions for optimality of (P3) yield

$$\frac{\partial L}{\partial f_i} = \frac{\partial U}{\partial f_i} - \mu \frac{\partial G}{\partial f_i} = 0; i=1, 2, \dots, q$$

As there is only one constraint G and assuming all $\frac{\partial G}{\partial f_i}$ exist at the point of interest, we can first discover a value for μ based on f_1 :

$$\mu = \frac{\partial U / \partial f_1}{\partial G / \partial f_1}$$

Substituting μ for f_j , $j \neq 1$, gives

$$\frac{\partial L}{\partial f_j} = \frac{\partial H}{\partial f_j} - \left(\frac{\partial U}{\partial f_1} / \frac{\partial G}{\partial f_1} \right) \frac{\partial G}{\partial f_j} = 0$$

or

$$\left(\frac{\partial U}{\partial f_j} / \frac{\partial U}{\partial f_1} \right) = \frac{\partial G}{\partial f_j} / \frac{\partial G}{\partial f_1} \quad (3)$$

The term on the left is called the "marginal rate of substitution," the one on the right the "trade-off ratio."

The marginal rate of substitution is known only by the DM. It is the rate at which he is willing to trade objective f_j for objective f_1 . The trade-off ratio is a result of solving the model for noninferior solutions and gives the rate at which the model suggests one can trade f_j for f_1 . Equation (3) says that if the DM's marginal rate of substitution equals the problem's trade-off ratio for each of the objectives 2 through q against the first objective, then he has satisfied the necessary conditions and thus likely has found the optimum of his utility function. An algorithm based on this idea in Haimes and Hall (1974) (see also Nishitani et al (1980)) proceeds as follows

- 1) For each objective f_i , $i \geq 2, \dots, q$, find a sequence of noninferior solutions such that each f_j , $j \neq 1$ and i , is being held fixed and f_i is being exchanged for f_1 . Move along this "line" in (f_1, f_2, \dots, f_q) space showing each noninferior solution found to the DM. Continue until the trade of f_j for f_1 being found seems to be what he likes. At this point his marginal rate of substitution equals the problem's trade-off ratio in the direction searched.
- 2) Repeat step 1 until no moves in any direction are possible.

The above scheme is like a "univariate" search in (f_1, f_2, \dots, f_n) space. It must have similar characteristics, e.g. zigzagging in small steps if a ridge is encountered. It would seem a clever move to adopt conjugate search directions. How one deals with fuzzy DM responses is of course an open question when one attempts a fancier algorithm.

Umeda et al (1980) suggest using a pattern search method (e.g. the simplex method or complex method) to find the optimum of the DM's utility function. The major advantage is that the DM need only rank order the current set of noninferior solutions. Consistent with their paper an approach can be as follows (see Fig. 11).

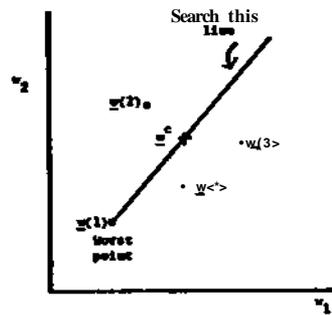


Fig. 11 Using a pattern search algorithm to adjust the weights to find the DM's optimal noninferior solution.

- 1) First find the Utopia point, $(f_1^*, f_2^*, \dots, f_q^*)$.
- 2) Choose a set of at least $q+1$ linearly independent vectors of weights, $w(i)$.
- 3) For each vector of weights $w(i)$, find the corresponding noninferior point using formulation (P4) with the Utopia point being the origin, (there appears to be no reason formulations (P5) and (P6) could not be used instead.)
- 4) Ask the DM to rank order the points, for example, using a minimal number of pairwise comparisons.
- 5) Draw a line from the worst point in w space through the centroid of the other w 's. The centroid is the point w^c with the coordinates $w_j^c = \frac{1}{n} \sum_{i=1}^n w_j(i)$ where

$$w_j^c = \frac{1}{n} \sum_{i=1}^n w_j(i)$$

ijthorst
Point

where n is the number of points $w(i)$ in the "complex" of points being used. Search along that line until a point is found which is better at least than the second worst point.

- 6) Toss out the weights $w(i)$ corresponding to the worst point from the current set and add in the new point found in step 5. (Test to be sure the set of vectors $w(i)$ stay linearly independent. If not, occasionally restart the algorithm from step 2.)
- 7) Repeat from step 3 until the DM is unable to distinguish among the noninferior points found.

The last approach we shall describe for the MCDM Executive is the one suggested by Payne and Polak (1980). It was developed for the special case of precisely two competing objectives. Figure 12 illustrates the idea. The approach is interactive and particularly attractive for interacting with the DM through a graphics display terminal.

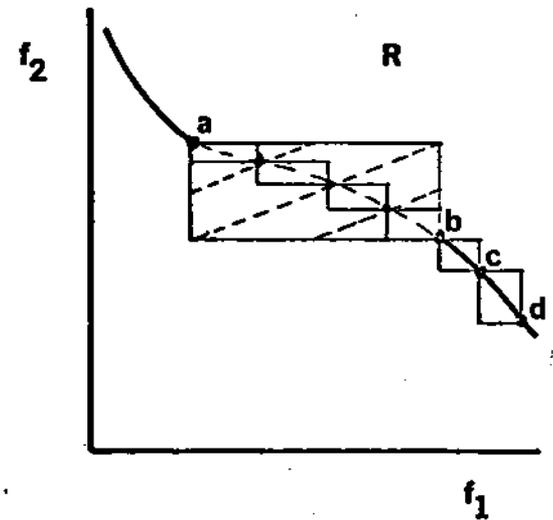


Fig. 12 Payne and Polak (1980) rectangles bounding the noninferior curve. Only the noninferior points a,b,c and d have been evaluated.

The DM is asked to establish a rectangle in (f_1, f_2) space within which his preferred solution will reside. The MCDM executive algorithm then selects where to try to find a set of a prescribed number of noninferior solutions that will leave the maximum for the least amount of area within which the optimum could exist. That twisted statement simply means they are applying an area elimination scheme in (f_1, f_2) space. See Fig. 12. Note that if points "a" through "d" are known to be noninferior solutions, then all other noninferior solutions can only exist within the rectangles shown. They look for solutions along lines passing diagonally from lower left to upper right within the chosen rectangles. When the noninferior solutions are located, new rectangles are drawn. The DM chooses which of these contains his optimum solution and the process repeats. It

terminates when the chosen rectangle is so small that the DM no longer cares to distinguish among the noninferior solutions residing within it.

Goal programming

Frequently the term goal programming appears in papers discussing multiple criteria decision making (MCDM) problems. Goal programming is where one picks a set of target values for the objectives and then seeks to find a solution that in some sense comes the closest to that target. If the target is \hat{f}_i , one can redefine the objectives to be of the form

$$f_i' = d(f_i, \hat{f}_i)$$

where the function d measures the distance between f_i and \hat{f}_i . Examples would be

$$(a) \quad d(f_i, \hat{f}_i) = |f_i - \hat{f}_i|$$

$$(b) \quad d(f_i, \hat{f}_i) = (f_i - \hat{f}_i)^2$$

Using f_i' as the i -th objective, we have converted a goal program into an MCDM problem.

Using form (a) for $d(f_i, \hat{f}_i)$, one can write

$$f_i - \hat{f}_i + u_i - v_i = 0$$

$$u_i, v_i \geq 0$$

and then

$$f_i' = d_i = u_i + v_i$$

In this form, d_i can be included within a linear programming formulation.

Chemical Engineering Applications

Within the chemical engineering literature, a number of papers exist which have applied MCDM concepts to solving example process design problems. As mentioned before the earliest found is by Seinfeld and MacBride (1970). Others include those by Nishitani and coauthors (1979, 1980, 1981), Shieh and Fan (1980), Umeda et al (1980), Grauer et al (1979a,b), and Grossmann and Jain (1981).

Other Issues to Consider in MCDM Problems

Of considerable importance for MCDM problems are a few of the following questions. First, one is faced with the issue of developing a definition and then a means to model processes so as to be able to characterize them in terms of noncomparable objectives. For example, how does one state, in terms of one or two measures each, the safety of a process, the flexibility of a process, process reliability or process controllability? These criteria open up whole new and important modeling questions, ones whose answers are the theme of much research currently

being performed. Certainly the papers in this conference by Grossmann and Morari will address this issue.

Another question which occurs is that most designers identify which variables in their problem they wish to select as objectives only as they proceed. Indeed one has to suspect significant mind changing occurs throughout the design process. That would suggest really flexible programs are needed which allow virtually any variable to be an objective or be constrained or be ignored with frequent changes in the way a variable is treated as the design proceeds.

A last point of interest is that the MCDM problem is really in the class of embedded optimization problems, (Pi), as we showed at the start of this paper. This class of problems is a difficult one which we shall now discuss.

The General Two-Level Programming Problem

The recent availability of effective techniques for solving constrained nonlinear optimization problems, particularly the Wilson-Han-Powell algorithm (Wilson (1963), Han (1975), Powell (1977)), has created considerable interest in using these methods for optimizing chemical processes. Much of this work has been oriented towards using the rigorous models in conventional, "sequential modular" simulators to evaluate the coefficients for an approximate (and perhaps reduced) process model, then applying an optimizer to this simpler model (Parker and Hughes (1978), Jirapongphan, et al (1979), Biegler and Hughes (1981)). One important problem, however, seems to have been overlooked: when evaluating the behavior of units exhibiting phase and chemical equilibrium, the simulator must make discrete decisions (based on feed composition, temperature and pressure) about the number and nature of the phases present. This causes nondifferentiabilities in the approximate model seen by the optimizer and can lead to erratic behavior - perhaps even failure to converge. We will show how this problem can be formulated naturally as an embedded optimization of the form (Pi) and discuss the inherent difficulties in solving this type of problem. We will then discuss possible algorithms for obtaining a local solution of problem (Pi) by solving a sequence of traditional "single-level" nonlinear programs.

The Embedded Equilibrium Calculation

The most general way of performing the equilibrium calculation for multiphase, reacting chemical systems is to solve the following constrained optimization problem (Gautam and Seider (1979), Castillo and Grossmann (1980)):

$$\min_{n_k} \sum_{k=1}^K J^{\wedge} - 2 \sum_{i=1}^I n_i \ln n_i - 3T \ln f \cdot$$

NC IMP

$$\text{s.t. } \sum_{i=1}^{\text{MC}} \sum_{k=1}^{\text{NP}} a_{i,k} n_{i,k} = b_j \quad j=1, \dots, \text{NE}$$

$$n_{i,k} \geq 0 \quad i=1, \dots, \text{NC}; \quad k=1, \dots, \text{NP}$$

where HC - number of components in the system

- HP • number of phases in the system
- HE * number of elements in the system
- $n_{i,k}$ m number of gram atoms of component "i" in phase "k"
- μ_i - chemical potential of "i" at system temperature and unit pressure
- $f_{i,k}$ * fugacity of component "i" in phase "k"
- $a_{i,j}$ • gram atoms of element "j" in component "i"
- b. - gram atoms of element "j" in the system

The first constraint is the conservation equation: it states that the gram atoms of element "j" in the feed must end up distributed among the phases present at equilibrium. The second constraint merely states that the mole numbers must be non-negative, and the objective function is the Gibbs Free Energy.

In the context of optimizing a chemical process, the above problem is actually a parametric program: given the feed flows, temperature and pressure for the unit at equilibrium, the minimization can be performed. The question that we wish to ask is how to adjust these "parameters" so that the overall process is optimal in some sense. Thus, the problem is of the form (P1) where we identify

- q_1 « measure of process optimality
- x_1 - feed flows, temperature and pressure for the equilibrium unit and other variables describing other processing units
- g_1, h_1 • constraints other than those for the equilibrium calculation - conservation equations for other units, design specifications, etc.
- rf_2 - Gibbs Free Energy
- x_2 • mole numbers $n_{i,k}$ for the equilibrium calculation
- g_2, h_2 • atom balances and non-negativity constraints for the equilibrium calculation

There are two important points to be noted concerning this problem. First, the "parameters" of the inner problem (i.e. feed flows, temperature and pressure) are really variables being adjusted to accommodate the

outer one. These are not adjusted by the inner problem while minimizing the Gibbs Free Energy. Second, the problem is not a conventional max-min problem, since there are two distinct objective functions. Nor is it a multiobjective optimization, because the inner problem is a constraint for the outer one and it involves minimization over a subset of the variables.

An Example Illustrating Problem Difficulties

To understand more clearly what it means for the inner minimization to be a constraint for the outer one, consider the following simple example:

$$\begin{aligned} \min_{x_1, x_2} & (x_1 - 2)^2 + (x_2 - 5)^2 \\ & \text{it } 0 \leq x_1 \leq 5 \\ & x_2 \in \arg \min_y (x_1 + 1)^2 + (y - 5)^2 \\ & \text{s.t. } -3x_1 + 2y \leq 2 \\ & x_2 \leq 3y \leq 14 \end{aligned}$$

Figure 13 shows the inner problem constraints and the contours of the outer problem objective function. Note that the inner problem essentially says "get x^* as close as possible to 5, subject to the constraints." As shown in the figure, when $x_1 \leq 2$ the first constraint is binding, while for $x_1 \geq 2$ the second constraint takes over. Thus, the x_2 that solves the inner problem for a given x_1 , which we will call $X_2(x_1)$, is

$$X_2(x_1) = \begin{cases} 1.5x_1 + 1 & \text{if } x_1 \leq 2 \\ -0.33x_1 + 4.67 & \text{if } x_1 \geq 2 \end{cases}$$

and the overall problem can be written

$$\begin{aligned} \min_{x_1, x_2} & (x_1 - 2)^2 + (x_2 - 5)^2 \\ & \text{s.t. } 0 \leq x_1 \leq 5 \\ & x_2 = X_2(x_1) \end{aligned}$$

Two difficult features of this problem are now apparent: nondifferentiability and nonconvexity. Nondifferentiability means that a gradient-based algorithm will probably have trouble solving the problem, since we cannot guarantee that the solution lies at a point where the Karush-Kuhn-Tucker conditions will hold. Nonconvexity means that we should expect multiple local optima. Both arise because we require x_2 to solve the inner problem for a given value of x_1 .

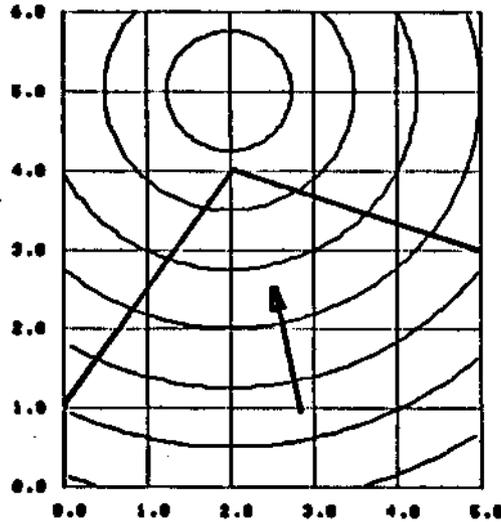


Fig. 13 The Inner problem constraints and outer problem objective contours for the example. The arrow shows the direction of decreasing g_1 .

Let us see what happens if we try to solve this example using the "most obvious" strategy. That is, given an x_1 , solve the inner problem for x_2 . Then, calculate the gradient of its solution (i.e. the gradient of $X_2(x_1)$) by differencing and the gradients of the outer problem constraints and use a gradient-based algorithm to solve the outer problem. This strategy is sketched in Fig. 14.

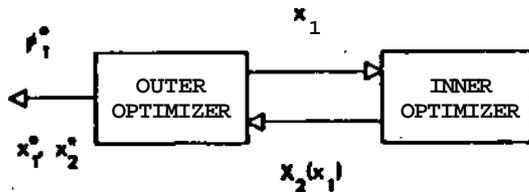


Fig. 14 The "most obvious" strategy for solving the problem: an inner optimizer is used explicitly to calculate $X_2(x_1)$.

Assume we start with an x_1 greater than 2, so the inner optimizer computes $X_2(x_1) = -0.33x_1 + 4.67$. The outer optimizer can then reduce its objective function by decreasing x_1 and moving x_2 along $X_2(x_1)$ until it reaches the point $(2,4)$. At this point, the outer optimizer will see $(x_2 - X_2(x_1))$ as shown by the vector $7X_2^*$ in Fig. 11 and the negative of its objective function gradient as $-v_1$. This suggests that further improvement can be made by reducing x_1 , since $-v_1$ does not lie along $7X_2^*$. However, reducing x_1 causes the inner optimizer to calculate $X_2(x_1) = 1.5x_1 + 1$, with gradient VX_2^* - suggesting improvement by increasing x_1 . Thus, the algorithm would bounce back and forth between the two constraints. Assuming that the outer optimizer

has a test on the gradient of the Lagrangian involved in termination, simply attempting to reduce the step size when it notices g_1 go up by moving from $(2,4)$ will not help.¹ The problem is that the equality constraint introduced by the inner problem has no gradient at that point. Note that if $-v_1$ lies anywhere within the cone defined by $7X_2^*$ and v_1 the point $(2,4)$ is optimal. This region is the cone of subgradients of $x_2 - X_2(x_1)$ at that point.

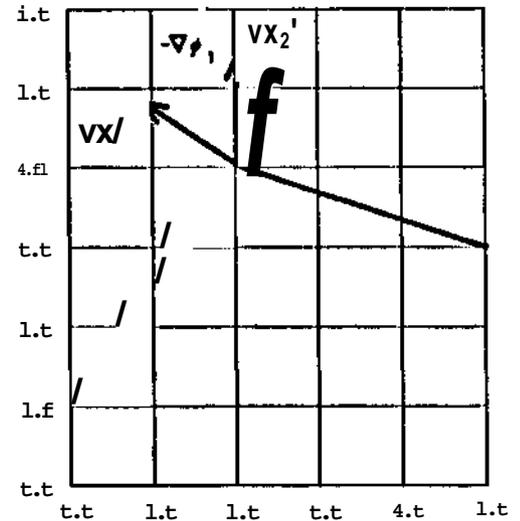


Fig. 15 The gradients at the point $(2,4)$.

An Insight into Solving

How, then, should we solve our problem? We wish to use a gradient-based method if possible, since all the truly effective algorithms for nonlinear programming fall in this class, but $X_2(x_1)$ stands in our way. Therefore let us consider X_2 more closely, with the hope of discovering a way of applying a gradient-based method to our problem.

Assuming that an appropriate constraint qualification is met, the Karush-Kuhn-Tucker conditions must hold in order that x^* solve the inner problem for a fixed x_1 (Bazaraa and Shetty (1979)):

$$\sum_{i=1}^m \lambda_i \nabla f_i(x_1, x_2) + \nabla f(x_1, x_2) = 0 \quad (4)$$

$$\lambda_i \nabla f_i(x_1, x_2) \leq 0 \quad i = 1, \dots, m \quad (5)$$

$$h_{2,j}(x_1, x_2) = 0 \quad j = 1, \dots, p \quad (6)$$

$$\lambda_i \nabla f_i(x_1, x_2) = 0 \quad i = 1, \dots, m \quad (7)$$

$$\lambda_i \nabla f_i(x_1, x_2) = 0 \quad i = 1, \dots, m \quad (8)$$

where X_1 and p_2 are the vectors of Lagrange multipliers corresponding to the constraints g_1 and g_2 . Thus, given an x_1 , equations (4) through (8) implicitly define $X_2(x_1)$. We will subsequently refer to Eq. (4) as the "stationarity relationship" and Eq. (7) as the "complementarity condition."

The equation of primary interest to us is the complementarity condition. It states that the indices of the inequality constraints g_2 can be partitioned into two sets:

$$A = \{k \mid g_2(x_1, x_2) = 0\}$$

$$I = \{k \mid g_2(x_1, x_2) < 0\}$$

such that multipliers, p_2 for the constraints with indices in I are zero, and only those corresponding to constraints with indices in A can be nonzero. Thus, if we knew what the proper partition of the inequality constraints was, $X_2(x_1)$ could be evaluated by solving Eqs. (4), (6) and the elements of (5) corresponding to constraints with indices in A . This observation is the basis of "active set strategies" for solving conventional nonlinear programs.

It is important to note that the interaction of the "on/off" nature of the complementarity constraint with the stationarity relationship is exactly the cause of the nondifferentiability of $X^*(x_1)$. That is, nondifferentiabilities in X_2 occur at those points where the partitions change, which means gradients are available if we stay within a fixed partition. Thus, one way of solving the overall problem is to examine each possible partition of the inner problem inequalities, solving the nonlinear program corresponding to each partition with a gradient-based algorithm.

Solution Algorithms

In a very recent paper, Bard and Falk (1982) also discuss the two-level programming problem. They suggest replacing the inner problem with Karush-Kuhn-Tucker conditions and solving the resulting single-level problem. Realizing that this problem is nonconvex, they propose solving it with a special-purpose algorithm for nonconvex programming (Falk (1972)). Since this algorithm requires the functions involved to be separable, the paper is devoted to solving the case where both the inner and outer problems are completely linear.

However, even when all the functions are linear, the usual forms of the complementarity conditions are not separable. Thus, the authors make use of some clever insight to convert the standard complementarity form

$$\sum_{i=1}^m X_i, g_i, W = 0$$

into the equivalent relations

$$\sum_{i=1}^m i \min(0, w_i) + ft = 0$$

$$w_i \cdot g_i(x) \cdot i = 0 \quad i = 1 \dots m$$

which makes the resulting single-level problem separable and piecewise linear. They can then apply their algorithm, which is a branch-and-bound technique, to the single-level problem and obtain its global solution. Since the problem is piecewise linear, they note that the maximum number of subproblems that will be solved for an inner problem with "n" inequalities is $2^{n+1} - 1$.

In the nonlinear case, trying the various partitions for a problem with "a" inner problem inequalities is still a combinatorial problem of size 2^a , but each element now requires solving a conventional nonlinear program. Thus, it is important to find a way of examining only a few of the possible partitions. We propose the following algorithm:

Step 1. Choose an initial partition of the inner problem inequalities into A and I .

Step 2. Solve the single-level nonlinear program:

$$\min_{x_1, x_2, \lambda} f_1(x_1, x_2)$$

$$\text{st } g_1(x_1) \leq 0$$

$$W = 0$$

$$\text{for } k \in A \quad \sum_{j=1}^p \frac{\partial h_{2j}}{\partial x_k} X_j = 0$$

$$h_2(x_1, x_2) = 0$$

$$g_{2,k}(x_1, x_2) = 0 \quad k \in A$$

$$g_{2,k}(x_1, x_2) \leq 0 \quad k \in I$$

$$f_{2,k} \geq 0 \quad k \in A$$

(Note that the last three constraints ensure that we remain in the selected partition.)

Step 3. If the solution to the above problem lies in the interior of the current partition (i.e. $\lambda_k > 0$ for all $k \in A$, and $g_{2,k} = 0$ for all $k \in I$) go to Step 5; otherwise go to Step 4.

Step 4. We are at a partition boundary. Check the adjacent partitions to

see if a feasible descent direction can be found. If one can, repartition the inner problem inequalities into A and I as appropriate and go to Step 2; otherwise go to Step 5.

Step 5. The current point is a local optimum satisfying the Karush-Kuhn-Tucker conditions for both the inner and outer problems. STOP.

Assuming a method is available for solving the conventional nonlinear program in Step 2, the difficult parts of this algorithm are choosing the initial partition in Step 1 and testing the adjacent partitions for a feasible descent direction in Step 4. We will first discuss the second point and return to the problem of choosing the initial partition later.

Testing for Descent at the Boundaries —
The Degeneracy Problem

If there were no inner problem, the information required to test for a feasible descent direction could be obtained by evaluating the Lagrange multipliers, λ_k , for the tight constraints as follows (Westérberg and DeBrosse (1973)):

Step 1. Let f be the vector of tight constraints at the current point. Perform the LU factorization

$$\left(\frac{\partial f}{\partial x}\right) = \left(\frac{\partial f}{\partial y} \cdot \frac{\partial f}{\partial z}\right) = (L)(U \cdot L^{-1} \frac{\partial f}{\partial z})$$

on its Jacobian matrix, splitting x into y and z so that $\left(\frac{\partial f}{\partial y}\right)$ is

invertible. (This is a natural consequence of the LU factorization.)

Step 2. Solve

$$(LU)^T \lambda = -\left(\frac{\partial f}{\partial y}\right)$$

for the Lagrange multipliers.

We prefer to calculate the multipliers in this way, rather than using a generalized inverse of the Jacobian, because it results in

$$\lambda_k = \left(\frac{\partial f}{\partial s_k}\right)_{s=0; j \neq k}$$

for an inequality rewritten as the pair $g_k(x) + s_k = 0$ and $s_k \geq 0$. Thus, if $\lambda_k < 0$, releasing the constraint g_k and moving into the feasible region (i.e. $\lambda_k \partial s_k > 0$) while keeping all other constraints tight ($s_j = 0$ for $j \neq k$) produces a decrease in the

objective function; this means we are testing all the constraints independently. If $\lambda_k \geq 0$ for all inequalities g_k , then the current point is optimal.

For the two-level programming problem, however, we have an additional complication associated with maintaining feasibility; the inner problem must remain at a minimum. Avoiding mathematical rigor (but see Robinson (1974), for instance), we state that if the multipliers for the tight inner problem inequalities are positive at the current point, and we hold the same constraints tight when we move, then we will remain at an inner problem minimum for a small step in the outer problem variables.

This suggests computing both the Lagrange multipliers for the inner problem (henceforth called simply "inner multipliers") and those for the nonlinear program solved in Step 2 of the algorithm ("outer multipliers"). We can then ask if there are any tight inner inequalities that the inner problem will allow us to drop (those with zero inner multipliers) and which the outer problem can release to reduce its objective function (those with negative outer multipliers). If none exist, then the current point is optimal. Otherwise, we place the indices for all the tight inner problem constraints that can not be dropped in the active set A and continue the algorithm at Step 2.

In the previous development we have assumed that the inner multipliers were uniquely defined. Unfortunately, there may be points for which this is not true: a situation referred to as "degeneracy" in the optimization literature. However, as opposed to its infrequent occurrence in conventional optimization problems, degeneracy will arise in the two-level programming problem each time the outer optimizer encounters a point with more tight inner problem constraints than there are inner problem variables.

We can see this in our simple example problem at the point (2,4). Here, both inner problem constraints are tight, yet we have only one inner problem variable. We cannot place both constraints in the active set and continue the algorithm at Step 2: to do so would mean we were fixing the value of an outer problem variable with an inner problem constraint. Thus, we are faced with a combinatorial problem of deciding which inner problem inequalities to place in the active set A.

Given an inner problem with "n" variables, "p" equalities and "m" tight inequalities, the maximum number of possible choices is

$$\frac{m!}{(m-n+p)! (n-p)!}$$

However, some of these choices may not be allowed, since we must simultaneously satisfy the stationarity constraint and

non-negativity of the inequality multipliers. To illustrate, consider the two-variable problem shown in Fig. 16. Here {1,2} and {1,3} are possible choices for the active set, but {2,3} is not, since $-p_4$ does not lie in the cone obtained from ∇g_2 and ∇g_3 .

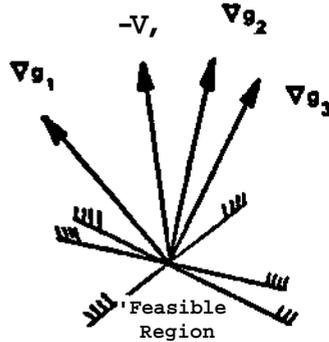


Fig. 16 Degeneracy and possible choices for the active set.

Note that seeking inner multipliers satisfying stationarity and non-negativity at a fixed point (x^*, x_2) is simply a phase 1 Simplex operation. (The solution of the nonlinear program from Step 2 provides us with one valid set of multiplier values, though, so our work is reduced.) Also note that the basis in the Simplex Tableau defines the current active set, so checking if a given inactive constraint could replace one in the active set while maintaining non-negativity is essentially a Simplex "Min Ratio test." Thus, if the number of possibilities is not too large, the work required to check the possible inner problem active sets should not be excessive.

Once the inner multipliers have been determined for the current active set, the gradients of the stationarity constraints in the associated single-level nonlinear program can be computed and the outer multipliers obtained. We can then ask the same questions as before. It is important to realize that the only portions of the Jacobian matrix used to calculate the outer multipliers which are affected by changing the inner problem active set are those rows corresponding to the inner problem stationarity constraints. Thus, if we have an inner problem with many fewer variables than the outer one (as is the case when the equilibrium calculation is embedded in a large flowsheet), considerable savings can be achieved by using techniques for modifying factorizations such as those of Gill et al (1975).

A Problem Relaxation

As mentioned previously, the difficulties associated with solving the two-level problem are caused by the discrete nature of the complementarity relationship. Thus, it was suggested in a conversation with Fiacco

(1982) to approximate the inner problem with a penalty function. This has the effect of relaxing the discrete behavior and removing the nondifferentiability. To see why this is the case, consider solving the problem

$$\begin{aligned} \text{minfix)} \\ x \\ \text{s.t } g_i(x) \leq 0 \quad i=1..m \end{aligned}$$

with the Interior penalty function

$$P(x,r) = f(x) - r \sum_{i=1}^m \ln(-g_i(x))$$

Computing an unconstrained minimum of $P(x,r)$ gives

$$\frac{\partial f}{\partial x} + \sum_{i=1}^m \frac{1}{-g_i} = 0$$

which is identical to the stationarity conditions

$$\frac{\partial f}{\partial x} + \sum_{i=1}^m \mu_i \frac{\partial g_i}{\partial x} = 0$$

for the original problem if $\mu_i g_i(x) = r$ for all i . Thus, the penalty approach works by achieving complementarity in the limit as r approaches zero.

Rather than using a penalty approach directly (attempting to avoid its ill-conditioning woes), and in the spirit of solving a single-level nonlinear program at Step 2 of our algorithm, we can solve

$$\text{min}_{x_1, x_2, \mu_2, \lambda_2} f(x_1, x_2)$$

$$\text{s.t } g_1(x_1) \leq 0$$

$$h^* x^* = 0$$

$$\frac{\partial f_2}{\partial x_2} + \sum_{j=1}^m \frac{\partial g_{2,j}}{\partial x_2} \mu_{2,j} + \sum_{j=1}^p \frac{\partial h_{2,j}}{\partial x_2} \lambda_{2,j} = 0 \quad (9)$$

$$g_{2,U}, x_2 \geq 0$$

$$h_{2,L} x_2 \geq 0$$

$$\mu_2 \geq 0$$

$$\mu_{2,i} g_{2,i}(x_1, x_2) \leq r \quad i = 1..m$$

for some "reasonably small", positive value of r to obtain a good initial estimate of a local solution to the two-level problem.

In fact, we can obtain a local solution to the overall problem (Pi) by solving the above program for a sequence of decreasing r. This is illustrated in Fig. 17, which shows the feasible region resulting from applying (9) to our previous example for various values of r: as r is decreased, the feasible region for the relaxed problem becomes a better approximation to the true feasible region of the two-level problem. Also, since it relaxes the discrete nature of the complementarity conditions, this approach avoids the combinatorial problems associated with degeneracy. In the limit as r approaches zero a valid set of inner multipliers will be found, and we are not forcing the outer problem to satisfy a selected set of inner problem inequalities as if they were equalities.

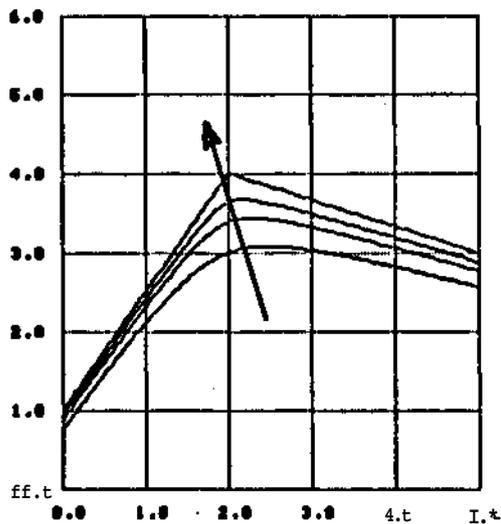


Fig. 17 The effect of r on the feasible region of (9). (The feasible region lies between the curves and the inner problem constraints.) The arrow indicates decreasing r.

While our computational experience is limited to a number of small example problems, this scheme appears to work quite well. For instance, solving the relaxed problem for our previous example at a single r value of 10^{-4} results in the proper solution - $x_1 = 2$ and $x_2 = 4$ - even though the inner problem is degenerate there. We are continuing the work by solving some examples involving the embedded equilibrium calculation.

DISCUSSION AND CONCLUSIONS

The multiple criterion decision making problem has been the subject of a large amount of literature. The difficult aspects of this problem are (1) developing effective models for different criteria, (2) finding the non-inferior points for the chosen models, and (3) Interacting with the decision maker in a way that allows him to discover his problem trade-offs without annoying or overwhelming

him.

An Issue which appears to have been given little attention, but which is a natural concern when we regard the decision making problem as an embedded optimization, is the presence of multiple local optima. For instance, consider a decision maker using problem formulation (P4) to solve the example shown in Fig. 18. Let us assume the weights have been chosen so that the solution Co (P4) gives the support plane through points "A" and "C". An increase in weight v_1 results in possible solutions at both "B" and "D", but "B" is only a local optimum. If v_1 were increased still more, the decision maker might experience a "jump" from "B" to "E", hampering his choice of an optimum.

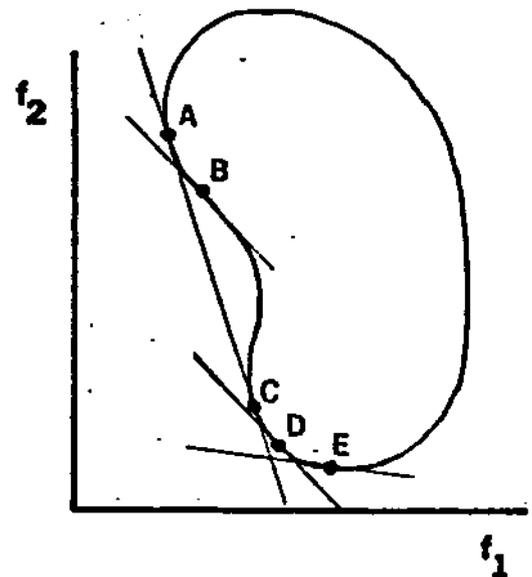


Fig. 18 Local optima in the multiple criterion decision making problem.

The general two-level programming problem has only recently begun to be studied. The difficult features of this problem are its nondifferentiability and nonconvexity; these arise from the discrete nature of the complementarity conditions for the inner problem. We have discussed two approaches that address this problem. One is based on an active set strategy for the inner problem and the other on a relaxation of the complementarity conditions. The active set idea suffers from a severe deficiency, however, which the relaxed formulation does not: deciding which constraints to place in the active set at a degenerate point is a combinatorial problem. Our limited computational experience suggests that the relaxed formulation is a very effective way to solve the embedded optimization problem.

REFERENCES

- Bard. J. F. and J. E. Falk (1982) "An Explicit Solution to the Multi-Level Programming Problem". *Computers and Operations Research*. 9(1). pp 77-100
- Bazaraa, M. S. and C. M. Shetty (1979) *Nonlinear Programming: Theory and Algorithms*. John Wiley and Sons. New York
- Benayoun. R., Tergny. J. and D. Keuneman (1970) "Mathematical Programming with Multi-objective Function: A Solution by P. O. P. (Progressive Orientation Procedure)". *METRA*. 9(2). pp 279-299
- Benayoun. R., de Montgolfier. J., Tergny, J. and O. Larichev (1971a) "Linear Programming with Multiple Objective Functions: Step Method (STEM)". *Math. Prog.*. 1(3). pp 366-375
- Benayoun. R., Larichev. O., de Montgolfier. J. and J. Tergny (1971b) "Linear Programming with Multiple Objective Functions: The Method of Constraints". *Automatic and Remote Control*. 32(8). pp 1257-1264
- Benson. K P. (1979) "Technical Note - Vector Maximization with Two Objective Functions". *J. Opt. Theory and Applic.* 28(2), pp 253-257
- Biegler, L. T. and R. R. Hughes (1981) "Infeasible Path Optimization with Sequential Modular Simulators". Paper presented at the AIChE National Meeting. New Orleans. LA. November. 1981
- Castillo. J. and I. E. Grossmann (1980) "Computation of Phase and Chemical Equilibria". Research Report DRC-06-19-80. Design Research Center, Carnegie-Mellon University. Pittsburgh. PA
- Falk, J. E. (1972) "An Algorithm for Locating Approximate Global Solutions of Nonconvex. Separable Problems", Technical Report T-262. The George Washington University. Washington. D. C.
- Fiacco. A. V. (1982) Personal Communication with Peter Clark
- Gautam. R. and W. D. Seider (1979) "Computation of Phase and Chemical Equilibrium: Part I. Local and Constrained Minima in Gibbs Free Energy". *AIChE Journal*. 25(6). pp 991-999
- Gearhart. W. B. (1979) "Technical Note - On the Characterization of Pareto-Optimal Solutions in Bicriterion Optimization". *J. Optim. Theory and Applic.* 27(2). pp 301-307
- Geoffrion. A. M. (1967) "Solving Bicriterion Mathematical Programs". *Operations Research*. 15(1). pp 39-54
- Geoffrion, A. M. (1968) "Proper Efficiency and the Theory of Vector Maximization", *J. Math. Anal. and Applic.* 22. pp 618-630
- Geoffrion. A. M., Dyer. J. S. and A. Feinberg (1972) "An Interactive Approach for Multi-criterion Optimization, with an Application to the Operation of an Academic Department". *Mgmt. Sci.*. 19(4). DP 357-368
- Gill. P. E., Murray. W. and M. A. Saunders (1975) "Methods for Computing and Modifying the LOV factors of a Matrix". *Mathematics of Computation*. 29(132). pp 1051-1077
- Grauer. M.L., Dohnal. M., and L. Pollmer (1979a) "Multiple Objective Optimization of Complex Plants - Deterministic and Random Approaches". Paper presented at the 12th Symposium on Computer Applications in Chemical Engineering. Montreaux. Switzerland. 1979a
- Grauer. M., Gruhn, G. and L. Pollmer (1979b) "A Multiple-Objective Approach to Cost Analysis of a Chemical Plant", Paper presented at the 12th Symposium on Computer Applications in Chemical Engineering, Montreaux. Switzerland, 1979b
- Grossmann. I. E., Drabbant. R. G. and R. K. Jain (1981) "Incorporating Toxicology in the Synthesis of Industrial Chemical Complexes". Paper presented at the AIChE Annual Meeting. New Orleans. LA. 1981
- Haines, Y. Y. (1970) "The Integration of System Identification and System Optimization". Ph. D. Thesis. Engineering Systems Dept.. School of Engineering and Applied Science. University of California. Los Angeles. CA
- Haines. Y. Y. and W. A. Hall (1974) "Multiobjectives in Water Resource Analysis: The Surrogate Worth Tradeoff Method", *Water Resources Research*. 10(4). pp 615-624
- Haines, Y. Y., Hall. W. A. and H. T. Freedman (1975) *Multiobjective Optimization in Water Resource Systems: The Surrogate Worth Tradeoff Method*. Elsevier Scientific Publishing CO., Amsterdam
- Han. S. P. (1975) "A Globally Convergent Method for Nonlinear Programming", Technical Report 75-257, Dept. of Computer Science. Cornell University, Ithaca. NY
- Huang. S. C. (1972) "Note on the Mean-Square Strategy for Vector-Valued Objective Functions", *J. Optim. Theory and Applic.* 9(5). pp 364-366
- Hwang, C. L., Masud, A. S. M., Paidv, S. R. and K. Yoon (1979) *Multiple Objective Decision Making - Methods and Applications: A State of the Art Survey*. Springer-Verlag. Berlin
- Hwang. C. L., Paidy. S. R., Yoon. K. and A. S. M. Masud (1980) "Mathematical Programming with Multiple Objectives: A Tutorial", *Computers and Oper. Res.*. 7. pp 5-31
- Jirapongphan, S., Boston, J. F., Britt. H. I. and L. B. Evans (1979) "A Nonlinear Simultaneous Modular Algorithm for Process Flowsheet Optimization". Technical Report, Chemical Engineering Dept., Massachusetts Institute of Technology. Cambridge, MA
- Klinger, A. (1964) "Vector-valued Performance Criteria". *IEEE Trans. Auto. Control*. AC-9. pp 117-118
- Lightner. M. R. (1979) "Multiple Criterion Optimization and Statistical Design of Electronic Circuits", Ph. D. Thesis. Electrical Engineering Department. Carnegie-Mellon University. Pittsburgh. PA

- Lin. J. G. (1976) "Multiple Objective KroDiems: Pareto-Optimal Solutions by Method of Proper Equality Constraints". *IEEE Trans. Auto. Control*. AC-2K5). pp 641-650
- Mukai. H. (1980) "Algorithms for Multicriterion Optimization". *IEEE Trans. Auto. Control*, **AC-25(2)**. pp 177-136
- Nishitani. H. and E. Kunugita (1979) "The Optimal Flow-Pattern of Multiple Effect Evaporator Systems". *Computers and Chemical Engineering J.* 3, pp 261-268
- Nishitani. K, Ogaki. Y., Kunugita. E. and L. T. Fan (1980) "Multi-Objective Optimization of an Aeration Vessel for Waste Water Treatment", *Chem. Eng. Commun.* 5. pp 135-147
- Nishtani. K and E. Kunugita (1981) "Multi-Objective Analysis for Energy and Resource Conservation in an Evaporator System". Paper presented at the Second World Congress of Chemical Engineering. Montreal. Canada. October. 1981
- Parker. A. L. and R. R. Hughes (1978) "Optimization of FLOWTRAN Models by Quadratic Approximation Programming". Research Report 2640. Engineering Experiment Station. University of Wisconsin. Madison. WI
- Payne, A. N. and E. Polak (1980) "An Interactive Rectangle Elimination Method for Biobjective Decision Making". *IEEE Trans. Auto. Control*. AC-25(3), pp 421-432
- Powell. M. J. D. (1977) "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations". Paper presented at the Dundee Conference of Numerical Analysis. 1977
- Robinson. S. M. (1974) "Perturbed Kuhn-Tucker Points and Rates of Convergence for a Class of Nonlinear Programming Algorithms", *Mathematical Programming*. 7. pp 1-16
- Salukvadze. M. (1974) "On the Existence of Solutions in Problems of Optimization Under vector-Valued Criteria", *J. Optim. Theory and Appic.* **13(2)**. pp 203-217
- Seinfeld. J. H. and W. L. MacBr. de (1970) "Optimization with Multiple Performance Criteria". *Ind. Eng. Chem. Process. Des. Devel.* 9(1). pp 53-57
- Shieh. J. H. and L. T. Fan (1980) "Multiobjective Optimal Synthesis of Methanation Process". Paper presented at the AIChE 73rd Annual Meeting. Chicago, IL, November. 1980
- Shimizu. Y. (←) "Consideration on a Rational Management of Radioactive Wastes by an Adaptation of Multiobjective Linear Programming". Technical Report. Research Reactor Institute of Kyoto University. Kyoto. Japan
- Srinivasan. V. and A. D. Shocker (1973) "Linear Programming Techniques for Multidimensional Analysis of 'Preference'", *Psychometrika*. 38, pp 337-369
- Stadler. W. (1979) "A Survey of Multicriteria Optimization or the Vector Maximization Problem. Part I: 1776 - 1960". *J. Optim. Theory and Appic.* **29(1)**. pp 1-51
- Takama. N. and T. Umeda (1980) "Multi-Level Multi-Objective Optimization in Process Engineering". *Chem. Eng. Sci.* 36. pp 129-136
- Umeda. T., Kuriyama. T., Kobayashi, S. and A. Ichikawa (1980) "Interactive Solution to Multiple Criteria Problems in Chemical Engineering Design", *Computers and Chemical Engineer/hg J.* 4, pp 157-165
- Wallenius. J. and S. Zionts (1977) "A Research Programme on Multicriterion Decision Making". In Bell. D., Keeney. R. and H. Raiffa (Eds.), *Contacting Objectives in Decisions*. John Wiley and Sons, New York, NY
- Wendell. R. E. and D. N. Lee (1977) "Efficiency in Multiple Objective Optimization Problems", *Math. Programming*. 12, pp 406-414
- Westerberg, A. W. and C. J. DeBrosse (1973) "An Optimization Algorithm for Structured Design Systems", *AIChE Journal*. 19(2), pp 335-343
- Westerberg, A. W. (1981) "Optimization in Computer Aided Design", In Mart. R. S. H. and W. D. Seider (Eds.), *Foundations of Computer-Aided Chemical Process Design*. Engineering Foundation. NY
- White. D. J. (1980) "Multi-Objective Interactive Programming". *J. Oper. Res. Soc.* 31. pp 517-523
- Wilson. R. B. (1963) "A Simplicial Algorithm for Concave Programming", Ph. D. Thesis. Graduate School of Business Administration, Harvard University. Cambridge, MA
- Zadeh. L. A. (1963) "Optimally and Non-scalar-valued Performance Criteria". *IEE Trans. Auto. Control*. **AC-8(1)**, pp
- Zionts, S. and J. Wallenius (1976) "An Interactive Programming Method for Solving the Multiple Criteria Problem". *Mejmt. Sci.* **22(6)**. pp 652-663
- Zionts, S. (1979) "Methods for Solving Management Problems Involving Multioie Objectives", Working Paper Series 400, Department of Management Science. SUNY at Buffalo, Buffalo, NY