A NONCONVEX MAX-MIN PROBLEM*

James E. Falk
The George Washington University
Washington, D.C.

Karla Hoffman
National Bureau of Standards
Washington, D.C.

ABSTRACT

An algorithm designed to solve a large class of nonconvex max-min problems is described. Its usefulness and applicability is demonstrated by solving an extension of a recently introduced model which optimally allocates strategic weapon systems. The extended model is shown to be equivalent to a nonconvex mathematical program with an infinite number of constraints, and hence is not solvable by conventional procedures. An example is worked out in detail to illustrate the algorithm.

1. INTRODUCTION

In [4], Shere and Wingate give the details of a model which they designed to optimally, allocate strategic weapon systems. Mathematically, their model is a max-min problem with independent constraints on the players. The inside problem is a convex program with a single complicating constraint.

In this paper, we extend the above model by allowing a more general objective function and more general constraints. The resulting model is characterized by having a nonconvex inside problem and cannot be solved by extending the technique suggested in [4].

The main purpose of this paper is to demonstrate a global solution procedure for general "separable" max-min problems. The technique is actually a composition of two optimization algorithms, one for nonconvex, separable programs, and the other for problems with an infinite number of constraints. It applies to any max-min problem of the form

$$\max_{x \in X} \min_{y \in Y} F(x, y) \quad \text{Problem P},$$

where $F(x, y) = \sum_{i} \sum_{j} F_{ij}(x_i, y_j)$ and where the sets $X$ and $Y$ are described by separable inequalities, e.g.,

$$X: f(x) = \sum_{i} f_i(x_i) \leq 0$$

---

*This work was sponsored by the Office of Naval Research under Program in Logistics Contract N00014-67-A-0214, Task 0001, Project NR 347 020.
\[ l \leq x \leq L \]
\[ Y: g(y) = \sum_{i} g_i(y_i) \leq 0. \]
\[ m \leq y \leq M \]

Note that the functional form of \( F \) results from the assumption that \( F(\cdot, y) \) is separable as a function of \( x \) for each \( y \in Y \) and \( F(x, \cdot) \) is separable in \( y \) for each \( x \in X \).

To ensure that the above problem has a solution, we shall require

(a) that \( l, L, m, M \) are finite vectors,

(b) that each component of the vector values functions \( f \) and \( g \) and the functions \( F(x, \cdot) \) and \( -F(\cdot, y) \) are at least lower semi-continuous.

With these assumptions, Problem \( P \) must have at least one globally optimal point \((x^*, y^*)\).

Section 2 contains a brief description of the Shere-Wingate model and indicates a number of extensions that increase its applicability. In particular, we allow their search effort to apply to more than one type of system and allow for changes in the unit procurement costs of a system which would be reasonable if quantity discounts are available.

Section 3 contains the details of the algorithm and Section 4 contains small example of the solution of an extended Shere-Wingate problem.

2. HISTORICAL DEVELOPMENT OF THE PROBLEM AND EXTENSIONS

A brief description of the Shere-Wingate model follows for readers unfamiliar with the original paper. Since our emphasis in the present paper is to show that difficult max-min problems can be solved, we do not concentrate on the details of this particular model. Interested readers are referred to [4].

Shere and Wingate discuss both missile and bomber systems. The missile systems are divided into two basic types. The first class is comprised of systems that are difficult to locate but easy to destroy once located (e.g., POLARIS). These systems are called "percentage vulnerable" systems. If \( y_i \) is the amount of search effort employed by an attacker in dollars, then the survivability of attack given an original force of \( z_i \) dollars of missile system \( i \) is measured by the surviving fraction of weapons

\[ x_iz_i(e^{-a_iz_i}) \]

where \( a_i \) is the vulnerability of the system.

The second class consists of systems which are easy to locate but difficult to destroy (e.g., MINUTEMEN). These systems are called "numerically vulnerable" systems. The amount of weapons surviving a barrage attack is \( x_i(z_i(e^{-a_iz_i})) \) where, again, \( y_i \) is the attacker's level of effort and \( z_i \) is the retaliator's level of effort, both measured in dollars.

Thus Wingate and Shere conclude that for a mix of \( M \) percentage vulnerable systems and \( N-M \) numerically vulnerable systems, the surviving "value" of the mix is

\[ F(x, y) = \sum_{i=1}^{M} v_ix_i(e^{-a_iz_i}) + \sum_{i=M+1}^{N} v_ix_i(e^{-a_iz_i}) \]

where the \( v_i \)'s are values assigned to the systems and measured, say, in megatonnage per dollar.
The above model does not consider the possibility that the search effort \( y_i \) might be able to locate more than one type of system. This paper has extended the model to include this possibility. If there are \( M \) percentage vulnerable systems and \( P - M \) numerically vulnerable systems, and if the attacker has \( N \) search systems which attempt to locate any of the first class of systems and \( Q - N \) search systems which have the capability to attack any of the second class of missile systems, then the surviving value of an attack would be

\[
F(x, y) = \sum_{i=1}^{M} v_i x_i \left( \sum_{j=1}^{N} e^{-a_i y_j} \right) + \sum_{i=M+1}^{P} v_i x_i \left( \sum_{j=N+1}^{Q} e^{-a_i y_j} \right).
\]

Modifications to include combinations of attack systems capable of searching for only certain missile systems of either class would not involve additional computational difficulties.

Another extension of the above model allows changes in vulnerability as the attacker's force increases. This change would require \( a_i \) to be a function of \( y_i \), and is easily handled by the method to be described.

In addition to the above modifications in missile systems, our model has also included bomber systems using the formula for the fraction of surviving bombers as \((1 - b_i \arctan a_i y_i)\) where \( y_i \) is the number of reentry vehicles attacking the bomber bases, and \( a_i \) and \( b_i \) are positive numbers depending on the number of bases, reaction time and density of attack. Thus, when including bomber systems, the surviving value of an attack becomes

\[
(1)
\]

\[
F(x, y) = \left[ \sum_{i=1}^{M} v_i x_i \left( \sum_{j=1}^{N} e^{-a_i y_j} \right) \right] + \left[ \sum_{i=M+1}^{P} v_i x_i \left( \sum_{j=N+1}^{Q} e^{-a_i y_j} \right) \right] + \left[ \sum_{i=P+1}^{P+R} v_i x_i \left( 1 - b_i \arctan a_i y_i \right) \right].
\]

It should be noted that when bomber systems are added to the model, the function \( F(x, \cdot) \) is no longer convex in \( y \).

Applying a conservative strategy, the objective of this model is to determine the optimal allocations \( x^* \) and \( y^* \) such that \( F(x^*, y^*) = \max \min F(x, y) \) subject to the fiscal, political and arms limitation constraints imposed on the model.

If \( R_x \) is the total funds available to the retaliator over a specified time frame and if \( R_y \) is the funds available to the attacker, then Wingate and Shere considered the cost for the retaliator to be

\[
\sum_{i=1}^{P+R} \left\{ o_i (x_i + \tilde{x}_i)/2 + p_i \max (0, x_i - \tilde{x}_i) + q_i \sgn x_i \right\} \leq R_x
\]

where \( o_i, p_i, \) and \( q_i \), respectively, denote the operating, procurement and buy-in costs per weapon; \( x_i \) is the number of weapons at the end of the time period and \( \tilde{x}_i \) is the existing or initial number of weapons. Note that \( q_i = 0 \) if either \( \tilde{x}_i > 0 \) or \( x_i = 0 \); otherwise it is positive. This paper extends the cost constraint developed by Wingate and Shere to allow not only jumps in the cost curve due to an initial procurement cost but also shifts in the slope of the cost curve, since it is reasonable to have the cost per item decrease due to economies of scale. Figure 1 illustrates the types of cost curves that may be considered.
The cost constraint for the retaliator therefore has the form

\[ \sum_{i=1}^{P+R} f_i(x_i) \leq R_2. \] (2)

A similar constraint for the attacker is

\[ \sum_{i=1}^{Q+R} g_i(y_i) \leq R_2, \] (3)

where \( g_i \) has a form similar to (2).

In addition to these budgetary constraints, there are arms limitations constraints imposed by international treaties. The constraints relating to these treaties given in the Wingate and Shere article are included here, name y

\[ \sum_{i=1}^{N} g_i x_i \leq \delta \] (4)

\[ \sum_{i=M_1} g_i x_i \leq \delta_1 \] (5)

\[ \sum_{i=M_2} g_i x_i \leq \delta_2 \] (6)

when the three constraints relate to the total number of launchers, sea-based launchers \( (M_1) \) and land-based launchers \( (M_2) \), respectively.

Another extension of their model included in this paper is the imposition of an additional constraint on the attacker. Since the arms limitations agreements are so loose as to never be a binding constraint on the attacker, we have added the more likely political constraint which states that the attacker can only use a proportion of his total force on any one attack, i.e.,

\[ \sum_{i=1}^{Q+R} y_i \leq cY, \]

where \( Y \) is the total armament storage of the attacker and \( c \) is some positive number between zero and one.

Any of the sample political constraints presented in the Wingate and Shere article can also be included in this model. These constraints together with the constraints on all the \( x \) and \( y \) variables make up the total constraints of the model.
To solve their model, Wingate and Shere used the fact that the inner problem
\[
\min_{y \in Y} F(x, y)
\]
was a convex programming problem with one complicating constraint. Our extensions have made
the inner problem nonconvex and have introduced more than one complicating constraint so that
the algorithm employed is no longer applicable. The next section of this paper will describe how
such problems can be solved.

3. THE ALGORITHM

Any max-min problem of the form of Problem P
\[
\max_{x \in X} \min_{y \in Y} F(x, y)
\]
is equivalent to an optimization problem of the form
\[
\max_{x_0 \in X} \quad \text{subject to } F(x, y) \geq x_0 \text{ for all } y \in Y.
\]
Note that there is a constraint for each point \(y \in Y\) so that, in effect, there are an infinite number of
constraints on the problem.

Recently, a cutting plane algorithm was introduced [1] which applies to problems of the
general form
\[
\max_{x \in X} f(x) \quad \text{subject to } g(x, y) \geq 0 \text{ for all } y \in Y
\]
and hence applies to the problem which we are considering. The most interesting feature of the
algorithm is that it generates a sequence of (potentially) nonconvex programs whose solutions will
converge to a solution of Problem Q if the functions \(f\) and \(g\) are continuous over \(X\) and \(X \times Y\),
respectively, and if these sets are compact. While the results of [1] are contingent on the continuity
of \(f\) and \(g\), an examination of the proofs contained therein shows that the results are still valid
with continuity replaced by upper semi-continuity of \(f\) and \(g\) in \(x\) and lower semi-continuity of \(g\)
in \(y\). Thus the algorithm will apply to problems of our form if we add lower and upper bounds \(l_0\)
and \(L_0\) on the introduced variable \(x_0\).

The cutting plane algorithm may be summarized as follows:

1. Select \(y^0 \in Y\) and set \(Y^0 = \{y^0\}, i = 0\).
2. Solve the problem
\[
\max_{x \in X} f(x) \quad \text{subject to } g(x, y^i) \geq 0 \text{ for each } y^i \in Y^i
\]
Problem \(Q^i\)

Denote a solution by \(x^i\).
3. Solve the problem
\[
\min_{y \in Y} g(x^i, y) \Big|_{\text{Problem R}^i}
\]

Denote a solution by \(y^{i+1}\).

(4) If \(g(x^i, y^{i+1}) \geq 0\), stop. Otherwise set \(Y^{i+1} = Y^i \cup \{y^{i+1}\}\) and continue.

Note that Problem \(Q^{i+1}\) has one more constraint that does \(Q^i\). Under certain conditions which, unfortunately, do not apply in our case, all constraints \(g(x, y^i) \geq 0\) \((j = 1, \ldots, i)\) for which \(g(x^i, y^i) > 0\) may be dropped from Problem \(Q^{i+1}\) without affecting convergence. We conjecture that such constraint dropping is possible in our case as well, but we have not tested this. Indeed, in the problems which we have run, convergence of the \(x^i\)'s was obtained so quickly that none of the added constraints became nonbinding.

It is important to point out that the problems \(Q^i\) and \(R^i\) generated by the above scheme may be nonconvex problems, and hence may have proper local solutions. There are a few schemes now available which guarantee global solutions of problems of the form of Problem \(Q^i\) and \(R^i\) provided that the constituents of these problems are separable functions. In our case, Problem \(Q^i\) has the form

\[
\max x_0 \\
\text{subject to } x_0 \leq \sum_j \sum_i F_{ij}(x_i, y_i) \text{ for all } k \in Y^i \\
\sum_i f_i(x_i) \leq 0 \\
l_0 \leq x_0 \leq L_0 \\
l \leq x \leq L
\]

and Problem \(R^i\) has the form

\[
\min \sum_i \sum_j F_{ij}(x^i, y_j) \\
\text{subject to } \sum_j g_j(y_j) \leq 0 \\
m \leq y \leq M,
\]

so that all problems involved are separable. The particular scheme which we applied is described in detail in [3]. Briefly, a continuous separable problem of the form

\[
\min \sum_{j=1}^n h_{ij}(x_i) \\
\text{subject to } \sum_{j=1}^n h_{ij}(x_i) \leq b_i \text{ (i = 1, \ldots, m)} \\
l_j \leq x_j \leq L_j
\]

is replaced by its "piecewise linear approximating problem"

\[
\min \sum_{j=1}^n \sum_{k=0}^{K_j} \theta_{jk} h_{ij}(x_{jk}) \\
\text{subject to } \sum_{j=1}^n \sum_{k=0}^{K_j} \theta_{jk} h_{ij}(x_{jk}) \leq b_i \text{ (i = 1, \ldots, m)}
\]
NONCONVEX MAX-MIN PROBLEM

\[
\sum_{k=0}^{K_j} \theta_{jk} = 1 \quad (j = 1, \ldots, n)
\]

\[
\theta_{jk} \geq 0 \quad (k = 0, \ldots, K_j; j = 1, \ldots, n)
\]

where the \(x_{jk}\)'s are fixed grid points lying between \(l_j\) and \(L_j\). The solution of this linear program offers a lower bound on the desired solution value. However, unless each set \(\theta_j = \{\theta_{j0}, \ldots, \theta_{jK_j}\}\) consists of at most two nonzero and adjacent variables, the point

\[
x_j = \sum_{k=0}^{K_j} \theta_{jk} x_{jk} \quad (j = 1, \ldots, n)
\]

is not an approximate solution of the desired solution. A branch-and-bound scheme is then invoked which eventually insures that the sets \(\theta_j\) do satisfy the adjacency conditions.

The modification necessary to handle lower semi-continuous functions \(h_{ij}\) \((x_i)\) is explained in [3].

The number of grid points that one selects is a user input. Increasing the number of grid points will, of course, increase the accuracy of the approximation, but will also increase the number of columns of the linear programs which are solved as subproblems. The total number of columns (exclusive of slack or artificial columns) of the subproblems is

\[
\sum_{j=1}^{n} (K_j + 1),
\]

so that the column size of the L.P.'s is a linear function of the number of grid points chosen.

In practice, we have usually taken equally spaced grid points of about 50/original problem variable. To test the accuracy of our solutions, we then double or triple this number to test the sensitivity of the global value.

It should be emphasized that this global optimizing scheme will generate points which are only approximately equal to the desired solution, with the degree of approximation dependent on the number of grid points \(x_{jk}\) selected in \([l_j, L_j]\). An exact globally optimizing solution could usually be found by using any of the locally convergent algorithms (e.g., SUMT) by starting at the approximate solution.

4. SAMPLE PROBLEM

The problem considered here includes all of the extensions covered in Section 2. There is a search system \(y_1\) capable of attacking percentage vulnerable systems \(x_1\) and \(x_2\) and a search system \(y_2\) capable of attacking numerically vulnerable systems \(x_3\) and \(x_4\). In addition, there is a search system \(y_3\) capable of attacking a bomber system \(x_5\). The vulnerabilities \(a_1\) and \(a_2\) of missile systems \(x_1\) and \(x_2\) are allowed to be functions of the search effort \(y_1\). The cost curves used here have jumps (discontinuities) due to initial procurement costs and also have shifts in their slopes reflecting large quantity cost discounts. An additional constraint on the \(y\) variable limits the total first attack force. There are constraints on the \(z\) variables that would be imposed by SALT agreements. The specific numbers used here were chosen to be fairly representative in size, but not in value, to actual numbers that might arise in a real application. The specific cost functions used do not reflect any actual cost functions, but were chosen to reflect the versatility of the model.
The sample problem is
\[
\max_{x} \min_{y} \{5x_1e^{-a_1(y)\cdot y_1}+2x_2e^{-a_2(y)\cdot y_1}+2x_3e^{-a_3(y)\cdot y_1}+x_4e^{-a_4(y)\cdot y_1}+0.5x_5(1-0.3 \arctan 0.02y_5)\}
\]
where
\[
a_1(y_1) = \begin{cases} 
0.02, & \text{if } 0 \leq y_1 \leq 4 \\
0.2, & \text{if } y_1 > 4,
\end{cases}
\]
\[
a_2(y_1) = \begin{cases} 
0.04, & \text{if } 0 \leq y_1 \leq 4 \\
0.10, & \text{if } y_1 > 4.
\end{cases}
\]
Here \(x\) must satisfy
\[
\sum_{i=1}^{5} f_i(x_i) \leq 2500
\]
\[
35x_1+30x_2 \leq 360
\]
\[
10x_3+24x_4 \leq 240
\]
when
\[
f_1(x_1) = \begin{cases} 
0, & \text{if } x_1 = 0 \\
200+60x_1, & \text{if } 0 < x_1 \leq 8 \\
320+45x_1, & \text{if } 8 < x_1,
\end{cases}
\]
\[
f_2(x_2) = \begin{cases} 
10x_2, & \text{if } 0 \leq x_2 \leq 5 \\
50x_2-200, & \text{if } 5 < x_2,
\end{cases}
\]
\[
f_3(x_3) = \begin{cases} 
8x_3, & \text{if } 0 \leq x_3 \leq 25 \\
38x_3-750, & \text{if } 25 < x_3,
\end{cases}
\]
\[
f_4(x_4) = \begin{cases} 
0, & \text{if } x_4 = 0 \\
37x_4+100, & \text{if } 0 < x_4,
\end{cases}
\]
\[
f_5(x_5) = \begin{cases} 
5x_5, & \text{if } 0 \leq x_5 \leq 10 \\
25x_5-200, & \text{if } 10 < x_5.
\end{cases}
\]
The variables \(y_i\) must satisfy
\[
\sum_{j=1}^{3} g_j(y_j) \leq 250
\]
\[
y_1+y_2+y_3 \leq 55,
\]
where
\[
g_1(y_1) = 17.5y_1,
\]
\[
g_2(y_2) = \begin{cases} 
5y_2/2, & \text{if } 0 \leq y_2 \leq 20 \\
25y_2/2-200, & \text{if } 20 < y_2,
\end{cases}
\]
\[
g_3(y_3) = \begin{cases} 
3y_3, & \text{if } 0 \leq y_3 \leq 10 \\
11y_3-80, & \text{if } 10 < y_3 \leq 15 \\
6y_3+10 & \text{if } 15 < y_3.
\end{cases}
\]
These cost functions \(f_i\) and \(g_j\) indicate the various different possible slopes that can be assumed. For example, \(g_1\) has a jump at zero reflecting a fixed or buy-in cost and a shift in slope at 8, reflecting per unit quantity discounts. The cost \(g_3\) reflects only operating costs up to 10 units, after which procurement costs are added, again with quantity discounts allowed after 15.
The algorithm then sets up Problem $Q^0$:

$$\max_{x \in X} x_0$$

subject to $x_0 \leq F(x, y^0)$,

where the set $X$ indicates those above constraints on $x$ together with upper bounds 12, 15, 40, 40, 50 on $x_1$, . . . , $x_5$ (their lower bounds are 0), and bounds $0 \leq x_0 \leq 200$. GLÔBAL was then applied by requesting 20 subintervals on each of the variables $x_0$, $x_1$, . . . , $x_5$. GLÔBAL terminated after solving four linear programming problems with a global solution

$$(x_0^0, x_1^0, \ldots, x_5^0) = (103.86, 10.28, 0.0, 24.0, 0.0, 50.0).$$

The next step in the algorithm involves setting up the Problem $R^0$:

$$\min_{y \in Y} F(x^0, y) - x_0^0$$

where $Y$ includes all of the previous constraints on $y$ together with upper bounds of 50 on $y_1$, $y_2$ and $y_3$. On this problem, GLÔBAL terminated after a single LP solution with the global solution $y = (10, 22, 0)$ and objective function value $-68.49$.

To form Problem $Q^1$, we now add the constraint

$$x_0 \leq F(x, y^1)$$

to those of Problem $Q^0$ and resolve. After solving four LP problems, GLÔBAL terminated with the solution $(x_0^1, \ldots, x_5^1) = (65.040, 0.000, 13.167, 24.000, 0.000, 50.000)$.

Problem $R^1$ is then

$$\min_{y \in Y} \{F(x^1, y) - x_0^1\}$$

whose solution was again found after solving a single LP problem to be $y^2 = (10, 22, 0)$. Since the optimal objective function value is 0 this time, the algorithm terminates.

The computational times for Problems $Q^0$, $R^0$, $Q^1$ and $R^1$ were 26.83, 14.99, 28.30 and 14.68 seconds for a total solution time of 84.80 seconds.

Several similar problems were run with different values of the coefficients. In none of these cases did the sequence $Q^0, R^0, \ldots$ proceed past $R^1$, even though the convergence theorem can, at best, guarantee limiting convergence.

It is difficult to make generalizations about the running times that one would expect on problems of this form, since such times are functions of:

(a) the number of original problem variables
(b) the number of grid points chosen
(c) the specific nature of the problem.

Item (c) is the most troublesome to address, since it dictates both the number of cuts that must be adjoined to the master problem and the number of nodes that are necessary to explore on the branch and bound tree.

A recent sequence of runs with NUGÔBAL on a single, badly behaved non-convex optimization problem with 14 original problem variables produced solutions requiring from 7.6 seconds to 41.6 seconds CPU on the university’s IBM 370, model 145. A total of 15 problems (differing
only in the values of their input data, and not their functional forms) were solved in an average of 17.2 seconds of CPU time. Details of this study are found in reference [2].

REFERENCES


