The Simplex Method as a Global Optimizer: A C-Programming Perspective

MOSHE SNIEDOVICH, EMMANUEL MACALALAG, and SUZANNE FINDLAY
Department of Mathematics, The University of Melbourne, Parkville VIC 3052, Australia
(*E-mail*: *moshe@mundoe.maths.mu.oz.au*)

(Received: 16 June 1992; accepted: 18 March 1993)

Abstract. In this paper we give a brief account of the important role that the conventional simplex method of linear programming can play in global optimization, focusing on its collaboration with composite concave programming techniques. In particular, we demonstrate how rich and powerful the c-programming format is in cases where its parametric problem is a standard linear programming problem.

Key words. Linear programming, simplex method, c-programming, composite functions, global optimization, dc problems.

1. Introduction

For over forty years the simplex method has been one of the most powerful and useful tools of optimization theory. Although it has been used mostly to solve linear programming problems, it has been extended in several ways to facilitate the solution of non-linear optimization problems, e.g. quadratic programming, convex programming and fractional programming problems. Each of these extensions is "classical" in the sense that it exploits convexity properties of the objective function which ensures that any local optimum is a global one. Recently, Horst *et al.* [7], Konno *et al.* [11], Yajima and Konno [21] and Konno and Kuno [10] discussed some applications of linear programming techniques in the solution of global optimization problems. In our discussion we continue this line of investigation, examining the simplex method as a *global optimizer*, namely as a method for recovering global optimal solutions to problems that may have more than one local optimal solution and where, in general, there is no guarantee that a local optimum is a global one. Obviously, this implies that we shall use the simplex method in the context of problems whose objective functions are not pseudoconvex (assuming opt is min). The following naive example illustrates the sort of difficulty addressed in this paper:

EXAMPLE 1.

$$\min_x -(x_1-4)^2 - x_2$$

subject to

$$x_1 + x_2 \leqslant 14; x_1 + x_2 \geqslant 4; x_1 - x_2 \leqslant 4; -x_1 + x_2 \leqslant 4; x_1, x_2 \geqslant 0\,.$$

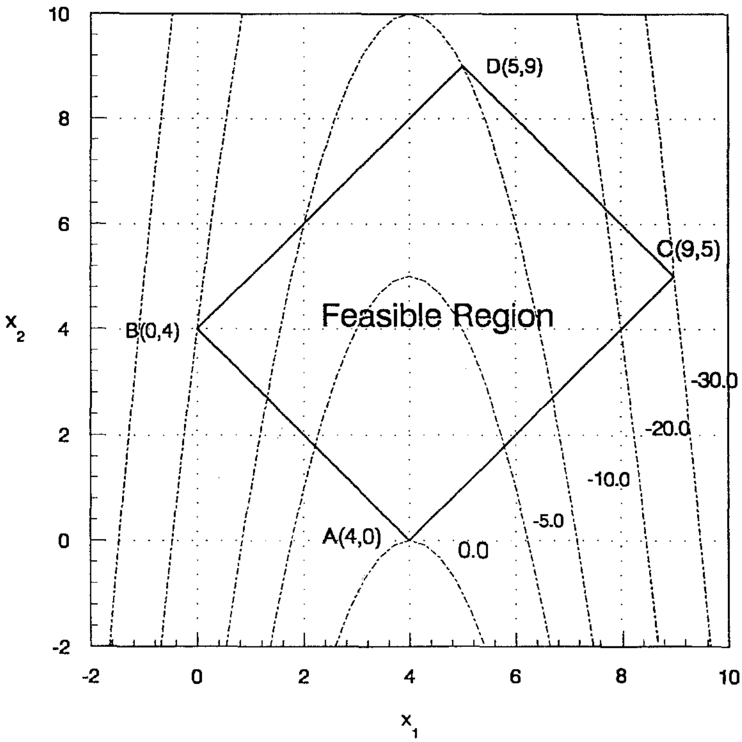

Fig. 1. Geometry of Example 1.

Figure 1 illustrates the geometry of the problem. Observe that there are two local minima, namely $x = (0, 4)$ and $x = (9, 5)$. Only the latter is a global one. So if, for instance, we use the GINO System (see Liebman *et al.* [13]), which is based on the generalized reduced gradient method, we recover the point $x = (0, 4)$. Using the GUESS command to instruct GINO to use $x_1 = 8$ as an initial guess, we recover the global optimum, $x = (9, 5)$. □

The difficulty is that in cases where the objective function is not pseudoconvex it is often difficult to check whether a given local optimum is a global one. Nevertheless, in this discussion we illustrate how c-programming [18, 20] resolves this difficulty for a certain class of problems. Technical details concerning the theoretical idiom of c-programming, its techniques, algorithms and potential applications can be found in the references cited in the discussion and in Section 9. The main purpose of the discussion is to illustrate the expressive power of c-programming and the role that it can play in global optimization when it is used in conjunction with the simplex method. In other words, the paper as a whole is expository in nature.

With this in mind, consider now the following class of composite linear programming problems:

PROBLEM P.

$$\tau := \min_{x \in X} g(x) := \Phi(Cx + d) \tag{1}$$

where

$$X := \{x \in \boldsymbol{R}^n : Ax = b, x \geqslant 0\} , \tag{2}$$

$\boldsymbol{R}$ denotes the real line, A is an m by n matrix, b is an m vector, C is a k by n matrix, d is a k vector and Φ is a real valued function on $\boldsymbol{R}^k$. Let X^* denote the set of (global) optimal solutions to Problem P.

Observe that the solution set $X \subseteq \boldsymbol{R}^n$ has the usual form associated with standard linear programming problems. On the other hand, the objective function g is not linear. It is expressed as a composite function, Φ, of k linear functions $C_1x + d_1$, $C_2x + d_2, \ldots, C_kx + d_k$, where C_i denotes the i-th row of the matrix C and d_i denotes the i-th component of the vector d. For example, in the case of

$$g(x) = (3x_1 + 6x_2 + 1)(5x_1 + 2x_2 + 3) , \quad x \in \boldsymbol{R}^2 \tag{3}$$

we can set $k = 2$, $d = (1, 3)$, $C = [\begin{smallmatrix} 3 & 6 \\ 5 & 2 \end{smallmatrix}]$ and

$$\Phi(v, w) = v \times w , \quad v, w \in \boldsymbol{R} , \tag{4}$$

whereas in the case of the function defined by

$$g(x) = 3x_1 + 6x_2 + 1 - (5x_1 + 2x_2 + 3)^2 , \quad x \in \boldsymbol{R}^2 \tag{5}$$

we can set k, d, and C as above, and define Φ as follows:

$$\Phi(v, w) = v - w^2 , \quad v, w \in \boldsymbol{R} . \tag{6}$$

Note that in both cases the function g is not pseudoconvex with respect to x. The difficulty is then that a local optimum is not necessarily a global one.

In the next section we present the parametric problem deployed by c-programming as a framework for solving Problem P, and specify the conditions imposed on the composite function Φ to ensure the recovery of a global optimal solution for Problem P. As we shall see, if $k \leqslant 2$, then the parametric simplex method can be used very effectively to solve Problem P. For larger values of k it would be necessary to use other techniques, e.g. linear multicriteria methods.

2. c-Programming's Approach

Before we examine how c-programming tackles Problem P, it is instructive to outline c-programming's basic approach in general, namely its basic approach for instances where its parametric problem is not a standard linear programming problem. Consider then the following generalization of Problem P:

PROBLEM Q.

$$\pi^* := \min_{y \in Y} q(y) := \psi(u(y)) \tag{7}$$

where Y is some non-empty set, q is a real valued function on Y, u is a function on Y with values in $\boldsymbol{R}^k$, and ψ is a real valued function on the set $u(Y) := \{u(y): y \in Y\}$. We shall denote by u_i the i-th component of u, thus $u(y) = (u_1(y), \ldots, u_k(y))$, where u_i is a real valued function on Y. Let Y^* denote the set of (global) optimal solutions to Problem Q.

To derive the parametric problem deployed by c-programming for the solution of Problem Q, we linearize the composite function Ψ with respect to u. That is, the parametric problem induced by Problem Q takes the following form:

PROBLEM Q(λ).

$$\pi(\lambda) := \min_{y \in Y} q(y; \lambda) := \lambda u(y) := \sum_{i=1}^{k} \lambda_i u_i(x) , \quad \lambda \in \boldsymbol{R}^k . \tag{8}$$

Let $Y^*(\lambda)$ denote the set of (global) optimal solutions of Problem Q(λ).
So the idea is to obtain an optimal solution for Problem Q by solving Problem Q(λ) for an appropriate value of λ, namely we seek a vector $\lambda^* \in \boldsymbol{R}^k$ such that any optimal solution to Problem $Q(\lambda^*)$ is also an optimal solution for Problem Q. Such a λ is referred to as an *optimal* λ. Let Λ^* denote the set of optimal values of λ. The following result spells out simple conditions under which an optimal λ exists, and its relationship to the constructs of Problem Q.

THEOREM 1 [18]. *Assume that the composite function* ψ is differentiable and pseudoconcave on some open convex set $U \subseteq \boldsymbol{R}^k$ *such that* $u(y) \in U$ *for all* $y \in Y$. *Then,*

$$Y^*(\nabla\psi(u(y))) \subseteq Y^* , \quad \forall y \in Y^* \tag{9}$$

where $\nabla\psi(u(y))$ *denotes the gradient of* ψ *with respect to* u *at* $u(y)$, *namely*

$$\nabla\psi(u(y)) := \left(\frac{\partial}{\partial\xi_1}\psi(\xi), \ldots, \frac{\partial}{\partial\xi_k}\psi(\xi)\right)\Big|_{\xi=u(y)} , \quad y \in Y . \ \blacksquare \tag{10}$$

A similar result can be obtained for the case where ψ is quasiconcave by imposing certain restrictions on the gradient of ψ (see [9]). So if we define

$$\Lambda := \{\nabla(\psi(u(y))): y \in Y^*\} , \tag{11}$$

it follows from Theorem 1 that all the elements of Λ are optimal. Now, turning back to Problem P and viewing it as an instance of Problem Q, we can set

$$u(x) := Cx + d\,, \quad x \in X \tag{12}$$

and consequently the objective function of the parametric problem of c-programming has the form $q(x; \lambda) = \lambda u(x) = \lambda Cx + \lambda d$. For the purpose of optimizing this function with respect to x, the constant λd can be dropped. Thus the problem involves optimizing the expression λCx. In short, in the context of Problem P the parametric problem of c-programming is:

PROBLEM P(λ).

$$\tau(\lambda) := \min_{x \in X} g(x; \lambda) := \lambda Cx\,, \quad \lambda \in \boldsymbol{R}^k\,. \tag{13}$$

Let $X^*(\lambda)$ denote the set of optimal solutions of Problem P(λ). Observe that for every $\lambda \in \boldsymbol{R}^k$, Problem P($\lambda$) is a standard linear programming problem. Thus, in the case of the function g specified in (3) we would have

$$g(x; \lambda) = \lambda_1(3x_1 + 6x_2) + \lambda_2(5x_1 + 2x_2)\,, \quad \lambda \in \boldsymbol{R}^2 \tag{14}$$

which would also be the form of the parametric function induced by the function g specified in (5).

Because this parametric function is linear with respect to x, for any given value of the parameter λ, it can be optimized by standard linear programming methods.

And so, applying Theorem 1 in the context of Problem P, we obtain the following result.

COROLLARY 1. *Assume that the composite function Φ is differentiable and pseudoconcave on some open convex set $U \subseteq \boldsymbol{R}^k$ such that $Cx + d \in U$ for all $x \in X$. Then,*

$$X^*(\nabla\Phi(Cx + d)) \subseteq X^*\,, \quad \forall x \in X^* \tag{15}$$

where

$$\nabla\Phi(Cx + d) := \left(\frac{\partial}{\partial \xi_1}\Phi(\xi), \ldots, \frac{\partial}{\partial \xi_k}\Phi(\xi)\right)\Bigg|_{\xi = Cx + d}, \quad x \in X\,. \quad \blacksquare \tag{16}$$

Note that by definition, $\nabla\Phi(Cx + d)$ denotes the gradient of Φ with respect to u defined in (12) at $u(x) = Cx + d$. For example, in the case of (5)–(6),

$$\nabla\Phi(v, w) = (1, -2w)\,, \tag{17}$$

thus

$$\nabla\Phi(Cx + d) = (1, -2(C_2x + d_2)) \tag{18}$$

$$= (1, -2(5x_1 + 2x_2 + 3)) \tag{19}$$

$$= (1, -10x_1 - 4x_2 - 6)\,. \tag{20}$$

Two interrelated questions arise naturally. First, what kind of linear c-programming problems satisfy the conditions required by Corollary 1? Second, how well can linear programming techniques cope with the task of recovering optimal λ's for Problem P(λ)? To address these important questions it is convenient to focus on the case where $k = 2$, which is discussed next.

3. Two-Dimensional Linear c-Programming Problems

Suppose that $k = 2$. Then Problem P can be written as follows:

PROBLEM P2.

$$\tau := \min_{x \in X} g(x) := \Phi(C_1 x + d_1, C_2 x + d_2) \tag{21}$$

in which case the parametric problem of c-programming has the following form:

$$\tau(\lambda) := \min_{x \in X} g(x; \lambda) := \lambda_1 C_1 x + \lambda_2 C_2 x\,, \quad \lambda \in \boldsymbol{R}^2\,. \tag{22}$$

Ignoring momentarily the instance where $\lambda_1 = 0$, upon dividing the right-hand side of (22) by the absolute value of λ_1, we obtain the following parametric problem:

PROBLEM P2(β).

$$\tau(\beta) := \min_{x \in X} g(x; \beta) := sC_1 x + \beta C_2 x\,, \quad \beta \in \boldsymbol{R}, s \in \{1, 0, -1\} \tag{23}$$

$$= (sC_1 + \beta C_2)x \tag{24}$$

where β is equal to λ_2 divided by the absolute value of λ_1. Note that formally we do not regard the scalar s as a parameter, rather it is treated merely as an indicator that in some instances it may be necessary to multiply C_1 by -1 and in some instances by 0. The latter case corresponds to the instance where $\lambda_1 = 0$. In particular, in cases where the first component of the gradient of Φ with respect to u is strictly positive, we can set $s = 1$. In short, for all practical purposes we can regard s in (23) as a nuisance *constant*.

Of course, the significance of the form of the parametric problem of c-programming given in (23)–(24) is that it is readily amenable to the conventional *parametric analysis* of the simplex method ([2] pp. 294–298, [6] pp. 307–309). This means that solving Problem P2(β) for a range of values of β is not much more expensive than solving a single linear programming problem of the same size (see for example the numerical results in Konno *et al.* [11], Macalalag and

Sniedovich [14] and Konno and Kuno [10]). The algorithm suggested by the foregoing analysis consists of three logical steps.

(1) First, it might be necessary to determine the interval, call it B, over which β is varied. This is usually accomplished by inspection. Alternatively, if no obvious lower and upper bounds can be computed for β, one may set $B = [-M, M]$ where M is a sufficiently large number. In many cases the parametric simplex procedure itself will take care of this matter.

(2) Using conventional parametric linear programming techniques, Problem P2(β) is solved for a finite sequence of values of $\beta \in B$, say $\{\beta^{(i)}\}$. Let $x(\beta^{(i)})$ denote the optimal solution recovered for Problem P2($\beta^{(i)}$). The parametric analysis techniques guarantee that for any $\beta \in B$ there is some $\beta^{(i)}$ such that $x(\beta^{(i)})$ is an optimal solution for Problem P2(β).

(3) The optimal solution for Problem P2 is recovered by selecting an i^* such that $x(\beta^{(i^*)})$ minimizes $g(x(\beta^{(i)}))$ over $\{x(\beta^{(i)})\}$.

So the overall conclusion when $k = 2$ is that if the composite function Φ is differentiable and pseudoconcave then Problem P can be solved efficiently by conventional linear programming techniques. Extensive experiments conducted with algorithms based on the foregoing analysis are reported on in Macalalag and Sniedovich [14].

4. Scope of Operation

What needs to be examined are the types of composite functions that readily lend themselves to the two-dimensional linear c-programming format discussed above. For obvious reasons we shall not provide a complete list. Rather, we shall present three major classes of problems.

4.1. RATIO FUNCTIONS

As is well known, e.g. Avriel [1], Bazaraa and Shetty [3], Sniedovich [18], the function Φ defined by

$$\Phi(v, w) := \frac{v}{w}, \quad v \in \boldsymbol{R}, w \in \boldsymbol{R}^+ := \{r \in \boldsymbol{R} : r > 0\} \tag{25}$$

is differentiable and pseudolinear on $\boldsymbol{R} \times \boldsymbol{R}^+$. Thus, fractional programming problems fall under the format of c-programming. But linear fractional programming problems can easily be transformed into standard linear programming problems by means of a simple transformation of variables (see [4, 15]). For this reason, and in this case, the c-programming format is primarily of methodological and theoretical interest.

However, the c-programming format significantly extends the scope of operation of the conventional parametric method of fractional programming. For example, it encompasses objective composite functions of the form

$$\Phi(v, w) := \frac{\varphi(v)}{w}, \quad v, w \in \boldsymbol{R}, w > 0 \tag{26}$$

where φ is a differentiable concave function, in which case Φ itself is pseudoconcave with (v, w). The same situation will be encountered in cases where

$$\Phi(v, w) := \frac{v}{\varphi(w)}, \quad v, w \in \boldsymbol{R}, v > 0 \tag{27}$$

and φ is differentiable, convex and strictly positive. For example, the function

$$\Phi(v, w) := \frac{v}{\sqrt{w}}, \quad v, w \in \boldsymbol{R}, w > 0, v \leq 0 \tag{28}$$

surfaces quite naturally in optimization problems involving deterministic equivalents of stochastic problems associated with normally distributed random variables (see [19]). More generally, c-programming can handle cases where

$$\Phi(v, w) := \frac{\sigma(v)}{\varphi(w)}, \quad v, w \in \boldsymbol{R} \tag{29}$$

where σ is differentiable concave and non-negative and φ is differentiable convex and strictly positive. We refer the reader to Avriel ([1], pp. 154–156) for a detailed analysis of convexity properties of composite ratio functions. The reader is also reminded that in our discussion opt = min.

4.2. MULTIPLICATIVE FUNCTIONS

Under this heading we consider cases where the composite function is of the form

$$\Phi(v, w) := \sigma(v)\varphi(w) \tag{30}$$

where both σ and φ are real-valued differentiable functions. Observe that ϕ is pseudoconcave if either one of the following conditions holds ([1], p. 156):

1. σ is nonnegative and concave and φ is positive and concave.
2. σ is nonpositive and convex and φ is negative and convex.

Thus, the c-programming format offers a substantial extension to the multiplicative case considered recently in Konno *et al.* [11]. Of course, the multiplicative functions and the ratio functions are essentially of the same structure, namely (29) can be rewritten in a product form

$$\Phi(v, w) := \sigma(v)\frac{1}{\varphi(w)} \tag{31}$$

and (30) can be rewritten in a fractional from

$$\Phi(v, w) := \frac{\sigma(v)}{1/\varphi(w)} . \tag{32}$$

Nevertheless, for our purposes it is instructive to consider these two classes separately, as this allows a more direct reference to the properties of the functions σ and φ.

4.3. ADDITIVE FUNCTIONS

It should be noted that because a function defined as the sum of two pseudoconcave functions is not necessarily pseudoconcave, the composite function

$$\Phi(v, w) := \sigma(v) + \varphi(w) , \quad v, w \in \boldsymbol{R} \tag{33}$$

where both σ and φ are pseudoconcave, is not necessarily pseudoconcave. But since concavity entails pseudoconcavity and furthermore concavity is preserved under addition, the c-programming format will accept the additive form given in (33) if both σ and φ are differentiable and concave. Obviously, this covers the degenerate case

$$\Phi(v, w) := v + \varphi(w) , \quad v, w \in \boldsymbol{R} \tag{34}$$

where φ is differentiable and concave. Observe that in this case

$$\nabla(\Phi(v, w)) = (1, \varphi'(w)) , \quad v, w \in \boldsymbol{R} \tag{35}$$

where φ' denotes the derivative of φ. Therefore, in the framework of Problem P2, we are interested only in $\lambda_1 = 1$, in which case the parametric problem is of the form given by (23)–(24) with $s = 1$. Note that in this case β represents the derivative of φ at some point $w \in \boldsymbol{R}$.

So in short, considering that there are many possible choices for the functions σ and φ, we conclude that the class of two-dimensional linear c-programming problems is rich indeed.

5. Beyond the 2-Dimensional Case

Needless to say, the most attractive feature of the two-dimensional case is that problems belonging to this class can be solve directly by conventional parametric analysis techniques of the simplex method. Furthermore, this involves only a moderate increase in the number of pivot operations compared to the number of pivot operations involved in solving a single conventional linear programming problem of the same size (see [10, 11, 14]).

Naturally, the computational effort is expected to increase as k is increased. But this does not mean that it is impossible to handle problems where $k > 2$. To assess the complications that would normally result when $k > 2$ it is instructive to consider the case when $k = 3$.

Upon dividing the objective function of Problem P(λ) by λ_1, assuming for simplicity that $\lambda_1 > 0$, we obtain the following problem:

PROBLEM P3(β).

$$\tau(\beta) := \min_{x \in X} g(x; \beta) := C_1 x + \beta_1 C_2 x + \beta_2 C_3 x,\ \beta \in \boldsymbol{R}^2 \tag{36}$$

$$= (C_1 + \beta_1 C_2 + \beta_2 C_3)x \tag{37}$$

where C_i denotes the i-th row of C.

Our plan is to solve Problem P3(β) for a finite number of β's, say $\{\beta^{(i)};\ i = 1, 2, \ldots, t\}$ such that for every $x \in X$ there is some i such that $x(\beta^{(i)})$ is an optimal solution of Problem P3($\beta(x)$), where $x(\beta)$ denotes the optimal solution for Problem P3(β) and

$$\beta(x) := \left(\frac{z_2}{z_1}, \frac{z_3}{z_1}\right),\ z_j := j\text{-th component of } \nabla(\Phi(u(x)))\,, \quad j = 1, 2, 3, x \in X\,. \tag{38}$$

If we let β_{pu} and β_{pl} denote upper and lower bounds of β_p, respectively, and set

$$B := \{(\beta_1, \beta_2) \in \boldsymbol{R}^2\colon \beta_{1l} \leqslant \beta_1 \leqslant \beta_{1u}, \beta_{2l} \leqslant \beta_2 \leqslant \beta_{2u}\} \tag{39}$$

then the search for an optimal β involves solving Problem P3(β) parametrically over B.

Obviously, the computational requirements of such a search can be much more demanding than those of the search associated with the one-dimensional case. But it should be noted that things are not as bad as they may appear. Firstly, the search very naturally lends itself to a divide and conquer approach, namely the overall search can be broken down into many *independent* searches each conducted over a small subset of B. In short, the search is amenable to *parallel processing*.

Secondly, it should be noted that the two components of the vector β are not completely independent of one another. In effect, the search need not be conducted on the entire rectangle B. In other words, it is often possible to exploit the fact that the two components of any optimal β are related to each other as stipulated by (38). In the extreme case, this relationship is so close that the two dimensional search degenerates into a one-dimensional search. In the context of (36)–(37) this happens when C_2 equals C_3, in which case the parametric objective function would be of the form

$$g(x; \beta_1, \beta_2) = C_1x + (\beta_1 + \beta_2)C_2x \,. \tag{40}$$

So we can set $\lambda = (\beta_1 + \beta_2)$ and treat the two-dimensional case as a one-dimensional case. The following example illustrates this point. Consider the case where

$$g(x) = cx - e^{(hx+r)} - (hx - r)^2 \tag{41}$$

where c and h are n-vectors and r is a scalar. We can then set $k = 3$, $C_1 = c$, $C_2 = C_3 = h$, $d = (0, r, -r)$ and

$$\Phi(\xi) = \xi_1 - e^{\xi_2} - \xi_3^2 \,, \quad \xi \in \boldsymbol{R}^3 \tag{42}$$

so that

$$\nabla\Phi(\xi) = (1, -e^{\xi_2}, -2\xi_3), \quad \xi \in \boldsymbol{R}^3 \tag{43}$$

and therefore

$$\beta(x) = (-e^{hx+r}, -2(hx - r)) \,, \quad x \in X \,. \tag{44}$$

Thus, we are interested only in pairs $(\beta_1, \beta_2) \in \boldsymbol{R}^2$ such that $\beta_1 = -e^{(-\beta_2/2)+2r}$. In this case the search is a one-dimensional one, namely the parametric problem is of the form

$$\tau(\lambda) := \min_{x \in X} g(x; \lambda) := cx + \lambda hx \,, \quad \lambda \in \boldsymbol{R} \,. \tag{45}$$

In short, although formally $k = 3$, the search for an optimal value of λ is conducted, in this case, in $\boldsymbol{R}$.

6. Composite Linear DC Programming Problems

Here we consider two-dimensional additive linear c-programming problems of the form

$$g(x) = \Phi(v, w) = \sigma(v) + \varphi(w) \,, \quad v = C_1x + d_1 \,, \quad w = C_2x + d_2 \tag{46}$$

where σ is *convex*, φ is *concave* and both are differentiable. In this case, Φ is a *dc function* (8), namely it is the difference between two convex functions, σ and $-\varphi$. The c-programming problem is then as follows:

PROBLEM DC.

$$\min_{x \in X} g(x) := \Phi(C_1x + d_1, C_2x + d_2) := \sigma(C_1x + d_1) + \varphi(C_2x + d_2) \,. \tag{47}$$

As usual, we linearize Φ with respect to σ and φ and consider the parametric problem

PROBLEM DC(λ).

$$\min_{x \in X} g(x; \lambda) := \lambda_1 C_1 x + \lambda_2 C_2 x \,, \quad \lambda \in \boldsymbol{R}^2 \,. \tag{48}$$

The difference between this format and the two-dimensional additive c-programming format discussed above is that here the function σ is convex rather than concave. This modification is reflected in the following result.

THEOREM 2. *Assume that Φ is a dc function as described above, and let x^* be any optimal solution of Problem DC. Then x^* is also an optimal solution of Problem DC(λ^*), $\lambda^* = \nabla\Phi(Cx^* + d)$.* ■

Although this result can be deduced directly from the analysis in Sniedovich [17], it is instructive to prove it formally here. We do this in two stages, the first merely invokes Theorem 1 to establish that if x^* is an optimal solution of Problem DC, then it must also be an optimal solution for

$$\min_{x \in X} \sigma(C_1 x + d_1) + \beta^* C_2 x \tag{49}$$

where $\beta^* = \varphi'(C_2 x^* + d_2)$ and φ' denotes the derivative of φ with respect to $C_2 x + d_2$. This is due to the fact that φ is concave. Thus,

$$\sigma(C_1 x^* + d_1) + \beta^* C_2 x^* \leq \sigma(C_1 x + d_1) + \beta^* C_2 x \,, \quad \forall x \in X \,. \tag{50}$$

Now, the classical first order necessary condition for x^* to be a local minimum point of (49) is

$$(x - x^*)[C_1 \sigma'(C_1 x^* + d_1) + \beta^* C_2] \geq 0 \,, \qquad \forall x \in X \tag{51}$$

where σ' denotes the derivative of σ with respect to $C_1 x + d_1$. Rearranging the terms in (51) yields

$$\alpha^* C_1 x^* + \beta^* C_2 x^* \leq \alpha^* C_1 x + \beta^* C_2 x \,, \quad \forall x \in X \tag{52}$$

where $\alpha^* = \sigma'(C_1 x^* + d_1)$. This implies that x^* is an optimal solution of Problem DC(λ^*), where

$$\lambda^* = (\alpha^*, \beta^*) = \nabla\Phi(Cx^* + d) = (\sigma'(C_1 x^* + d_1), \varphi'(C_2 x^* + d_2)) \,. \quad ■ \tag{53}$$

Observe, however, that methodologically it might be advantageous to disregard the fact that σ is a composite function. Namely, we can use the one dimensional format

$$g(x) = t(x) + \varphi(w(x)) , \quad w(x) = cx + d \tag{54}$$

where t is a convex function of x and φ is a differentiable concave function of $w(x)$. In this case the parametric problem of c-programming would be of the form

PROBLEM DC(λ).

$$\min_{x \in X} g(x; \lambda) := t(x) + \lambda cx , \quad \lambda \in \boldsymbol{R} . \tag{55}$$

Thus, for each value of λ, the parametric problem involves minimizing a convex function subject to linear constraints. This means that to solve the target problem we have to solve a number of "easy" convex problems. Needless to say, the simplex method can play a central role in such scheme, for example see the Frank–Wolfe method [5, 6].

In summary, what emerges is that in cases where the constraints of the target dc problem (47) are linear, c-programming offers a straightforward solution strategy for the target problem. Needless to say, a similar conclusion applies to situations where the solution set X is defined by inequalities of the form $h_i(x) \leqslant 0$, $1 \leqslant i \leqslant p$, where the h_i's are functions satisfying the usual constraints qualification requirements, in which case the parametric problem is solved by appropriate *classical optimization* methods.

7. Illustrative Examples

In this section, we consider a number of illustrative examples which are by design very small and naive. Extensive computational experiments with c-programming/ linear programming schemes are reported on in Macalalag and Sniedovich [14], where problems of up to 100 constraints in 1000 variables are considered.

EXAMPLE 2.

$$\min_{x} - 4x_1 + x_2 + 2\sqrt{10x_1 + x_2 + 1} \tag{56}$$

subject to:

$$x_1 + 3x_2 \leqslant 18 \tag{57}$$

$$3x_1 + x_2 \leqslant 14 \tag{58}$$

$$x_1 - x_2 \leqslant 2 \tag{59}$$

$$x_1 \geqslant 0, x_2 \geqslant 0 . \tag{60}$$

If we cast the above problem in the format of (21), we have,

$$\Phi(v, w) = v + 2\sqrt{w}$$

where $v(x) = -4x_1 + x_2$, and $w(x) = 10x_1 + x_2 + 1$. The gradient of Φ is given by

$$\nabla\Phi(v, w) = \left(1, \frac{1}{\sqrt{w}}\right) \tag{61}$$

so that the associated parametric problem in the format of (24) where $s = 1$, and $\beta = \lambda_2$, is

$$\min_x \, (-4x_1 + x_2) + \beta(10x_1 + x_2) \tag{62}$$

subject to (57)–(60)

Using (61) we can safely restrict the value of β to the closed interval

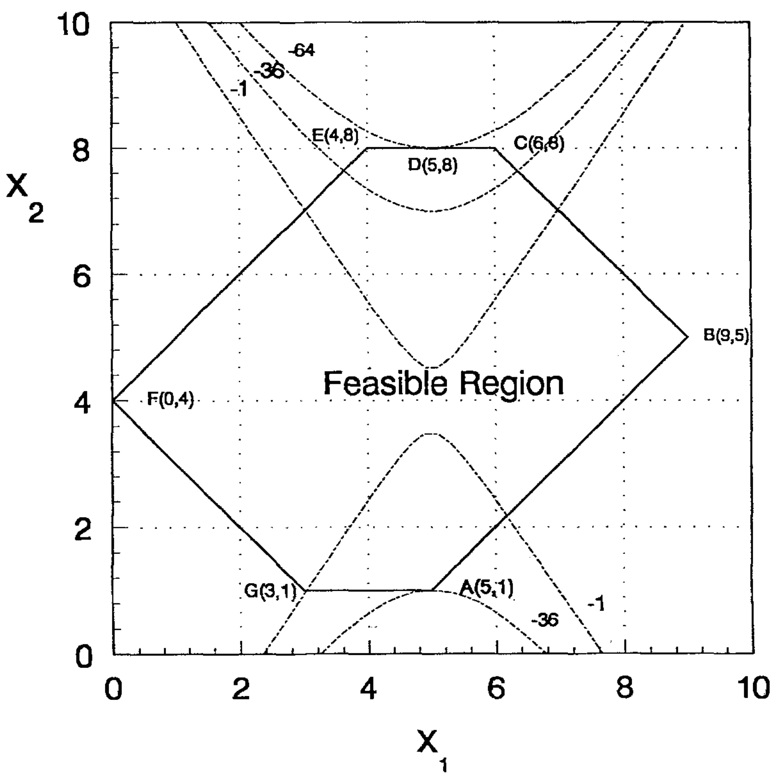

Fig. 2. Geometry of Example 2.

Table I. Summary of parametric analysis procedure applied to Example 2

β	Optimal solution to parametric problem	Objective value of target problem
0.000000	(4, 2)	−0.885123
0.272727	(2, 0)	1.165151
0.400000	(0, 0)	2.000000

$B=[0,.\,1]$. Figure 2 shows the feasible region and some level curves of the objective function given in (56).

The results obtained using the algorithm described in Section 3 are summarized in Table I.

Notice that we did not solve the parametric problem (62) for $\beta=1.000000$ because for $\beta>0.400000$ the solution $x=(0,0)$ remains optimal. The above procedure yields the global optimal solution $x=(4,2)$ with objective value equal to -0.885123. A more elaborate illustrative example is given in [14].

In the following examples, we demonstrate the capability of the c-programming approach used in conjunction with nonsimplex optimization procedures, in the solution of *dc programming problems*.

EXAMPLE 3.

$$\min_x 9(x_1-5)^2-4(x_2-4)^2 \tag{63}$$

subject to:

$$x_1+x_2\leqslant 14 \tag{64}$$

$$x_1+x_2\geqslant 4 \tag{65}$$

$$-x_1+x_2\geqslant 4 \tag{66}$$

$$x_1-x_2\leqslant 4 \tag{67}$$

$$1\leqslant x_2\leqslant 8,\, x_1\geqslant 0\,. \tag{68}$$

Hence, in compliance with the format of (54), we have,

$$t(x)=9(x_1-5)^2$$
$$\varphi(w)=-4w^21\,,\quad w(x)=x_2-4\,.$$

Thus, by (55), the associated c-programming parametric problem is given by

$$\min_x 9(x_1-5)^2+\lambda x_2 \tag{69}$$

subject to the same set of constraints (64)–(68). Note that $\varphi'(w(x))=-8(x_2-4)$ and thus λ can be restricted to the set $B=[-32,24]$.

For any $\lambda\in B$, the parametric problem given by (69) can be solved by standard quadratic programming methods (e.g. Frank–Wolfe), or by any one of a host of known constrained nonlinear programming methods (eg. reduced gradient methods). Therefore, given these various available solution methods, the c-programming approach to the solution of Example 3 is straightforward.

Table II. Summary of results for Example 3

Iteration	λ	Optimal solution	Number of line searches	Objective value
1	-32	(5, 8)	5	-128
2	24	(5, 1)	7	-72
3	0	(5, 1)	1	-72

We applied the c-programming algorithm [20], pp. 370–371 using the GINO system with its default settings to solve the required parametric problems. The results are summarized in Table II.

Note that a total of 3 iterations comprising 13 lines searches were needed to find the optimal solution $x = (5, 8)$ with objective value of -128. A direct solution by GINO using the default settings yields the nonoptimal solution $x = (5, 1)$. Figure 3 shows some level curves and the feasible region associated with Example 3.

EXAMPLE 4.

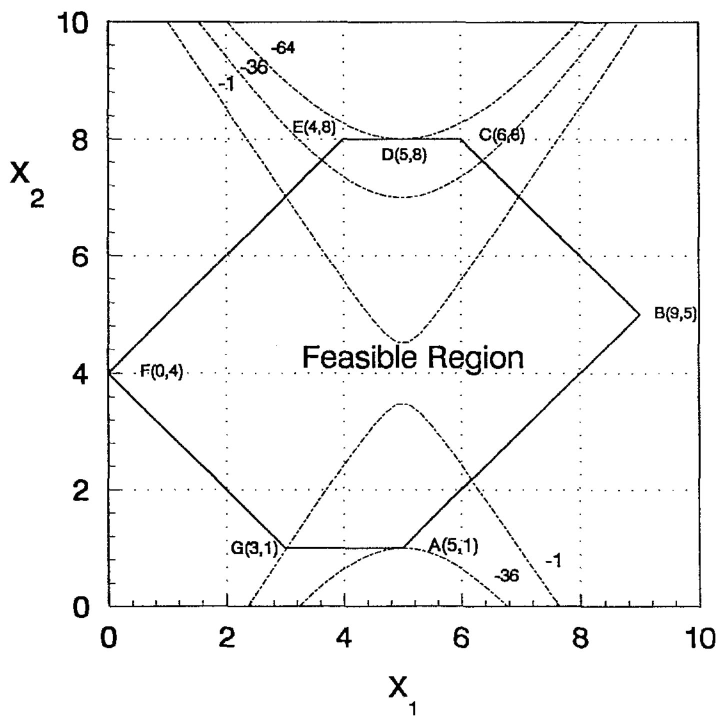

Fig. 3. Geometry of Example 3.

$$\min_x \left(\frac{x_1}{2} - 3\right)e^{(x_1/2-3)} - (x_2 - 5)^2 e^{(x_2-5)} \tag{70}$$

subject to:

$$x_1 + x_2 \leqslant 10 \tag{71}$$

$$-x_1 + 2x_2 \leqslant 8 \tag{72}$$

$$x_1 \geqslant 2\,, \quad x_2 \geqslant 3 + \sqrt{2}\,. \tag{73}$$

Hence, in compliance with the format of (55), we have

$$t(x) = \left(\frac{x_1}{2} - 3\right)e^{(x_1/2-3)}$$
$$\varphi(w) = -w^2 e^w$$
$$w(x) = x_2 - 5$$

observing that $t(x)$ is convex for $x_1 \geqslant 2$ and φ is concave for $w \geqslant 2 + \sqrt{2}$. The associated c-programming parametric problem is then given by

$$\min_x \left(\frac{x_1}{2} - 3\right)e^{(x_1/2-3)} + \lambda x_2 \tag{74}$$

subject to (71)–(73). Note that from the definition of φ it follows that

$$\varphi'(w(x)) = -(x_2 - 3)(x_2 - 5)e^{(x_2-5)}\,.$$

And since (71)–(73) entail that $3 + \sqrt{2} \leqslant x_2 \leqslant 6$, it follows that

$$\varphi'(w(x)) \in B := [-8.154845, 0.461159]$$

for all feasible x. Thus, the search for optimal values of λ can be restricted to the interval B. The results obtained by the c-programming procedure using GINO to solve the parametric problem are summarized in Table III.

The procedure terminated after 3 iterations applying a total of 8 line searches to find the optimal solution $x = (4.0000, 60000)$ with objective value equal to -3.0861. A direct solution by GINO yields the nonoptimal solution

Table III. Summary of results for Example 4

Iteration	λ	Optimal solution	Number of line searches	Objective value
1	0.461159	(3.999984, 4.414214)	4	−0.558897
2	−8.154845	(3.999984, 5.999992)	2	−3.086095
3	0	(4.000001, 6.000001)	2	−3.086169

(3.999723, 4.414214). All applications of GINO in this example used a tolerance level equal to 10^{-4}.

8. Beyond Linear Programming

It should be noted, as illustrated above in Example 3 and Example 4, that the c-programming format does not require the parametric problem to be a linear programming problem. That is, c-programming can collaborate with optimization methods other than the Simplex Method. For example, Sniedovich [20] describes collaboration schemes between c-programming and dynamic programming. In the following example we illustrate a collaboration scheme between c-programming and nonlinear optimization methods in the solution of global optimization problems having nonlinear constraints.

EXAMPLE 5.
Here we look at the two illustrative examples studied by Horst *et al.* [7].

EXAMPLE 5A.

$$\min_x 4x_1^4 + 2x_2^2 - 4x_1^2 \tag{75}$$

subject to

$$x_1^2 - 2x_1 - 2x_2 - 1 \leq 0 \tag{76}$$

$$-1 \leq x_1 , \quad x_2 \leq 1 . \tag{77}$$

The c-programming format given by (55) entails the following:

$$t(x) = 4x_1^4 + 2x_2^2$$
$$\varphi(w) = -4w^2 , \quad w(x) = x_1 .$$

Since $\varphi'(w(x)) = -8x_1$, we can set $B = [-8, 8]$. The parametric problem is given by

$$\min_x 4x_1^4 + 2x_2^2 + \lambda x_1 \tag{78}$$

subject to (76)–(77).

EXAMPLE 5B

$$\min_x (x_1^4 + x_2 + x_3) - (x_1 + x_2^2 - x_3) \tag{79}$$

subject to

$$(x_1 - x_2 - 1.2)^2 + x_2 \leqslant 4.4 \tag{80}$$

$$x_1 + x_2 + x_3 \leqslant 6.5 \tag{81}$$

$$x_1 \geqslant 1.4\,, \quad x_2 \geqslant 1.6\,, \quad x_3 \geqslant 1.8\,. \tag{82}$$

Again, a c-programming formulation of this problem can assume the format given by (55) where,

$$t(x) = x_1^4 + x_2 + 2x_3 - x_1$$
$$\varphi(w) = -w^2\,, \quad w(x) = x_2\,.$$

Thus $\varphi'(w(x)) = -2x_2$, and consequently using (81)–(82) we can set $B = [-6.6, -3.2]$. The parametric problem is given by

$$\min(x_1^4 + x_2 + 2x_3 - x_1) + \lambda x_2 \tag{83}$$

subject to (80)–(82).

The results obtained by a c-programming algorithm using GINO to solve the parametric problem are summarized in Table IV. Examples 5A and 5B were solved by applying GINO directly to the target problems. To comply with the settings used by Horst *et al.* [7], the c-programming tolerance levels based on the parametric objective function values were set to 0.05 and 0.01 for Example 5A and Example 5B, respectively.

In comparing the performance of the procedures, it should be noted that each iteration of Horst *et al.* [7]'s procedure requires solving a number of linear programming problems, and each iteration of the c-programming algorithm involves an application of GINO to a convex programming problem.

Table IV. Results for Examples 5A and 5B

	Example 5A	Example 5B
Horst *et al.* [1991]		
Optimal solution	(0.7197, 0.0000)	(1.400, 1.8128, 1.800)
Optimal objective value	−0.9987	4.568156
Number of iterations	34	18
GINO		
Optimal solution	(0.707105, −0.000004)	(1.400000, 1.809500, 1.800000)
Optimal objective value	−1.000000	4.576800
No. of line searches	7	4
C-programming		
Optimal solution	(0.685152, −0.000009)	(1.400000, 1.809509, 1.800000)
Optimal objective value	−0.996270	4.576809
No. of iterations	25	2
No. of line searches	135	4

9. Bibliographic Notes

In view of the expository nature of this paper we provide the following bibliographic guide. Details concerning the origin of c-programming can be found in [17]. A summary of the main results appear in [18, 20]. The general profile of c-programming algorithms and examples illustrating how they work can be found in [18–20] and [14]. The latter specializes in hybrid algorithms involving c-programming and linear programming techniques.

A slightly different class of composite problems and an approximation algorithms for treating them is discussed by Katoh and Ibaraki [3].

Numerical experiments with the parametric simplex method in the context of nonlinear programming problems of the type discussed in this paper are reported on in Konno *et al.* [11], Macalalag and Sniedovich [14], Yajima and Konno [21] and Konno and Kuno [10].

10. Conclusions

The simplex method of linear programming offers a number of possibilities for the solution of difficult global optimization problems. In this discussion we focused on collaborative schemes between the simplex method and c-programming, and showed that such schemes are readily available, and in fact can be easily implemented on available commercial optimization software such as LINDO (see Schrage [16]) and GINO. We believe that this is an indication that much progress in global optimization can be achieved by *specialization*, namely by focusing on subclasses of problems whose features make it possible to circumvent the difficulties posed by the general format of global optimization. So, while research into the possibility of formulating general purpose global optimization algorithms should not be discouraged, it would appear that there is much scope for progress in certain subclasses of problems which are complications of otherwise classical linear or convex optimization problems.

Acknowledgment

This research was supported in part by ARC Grant SG1911733 and by the Cooperative Research Centre for Intelligent Decision Systems, Melbourne, Australia.

References

1. Avriel, M. (1976), *Nonlinear Programming: Analysis and Methods*, Prentice-Hall, Englewood Cliffs, New Jersey.
2. Bazaraa, M. S., J. J. Jarvis, and H. D. Sherali (1990), *Linear Programming and Network Flows*, Wiley, New York.

3. Bazaraa, M. S. and C. M. Shetty (1979), *Nonlinear Programming, Theory and Algorithms*, John Wiley, New Jersey.
4. Craven, B.D. (1988), *Fractional Programming*, Heldermann Verlag, Berlin.
5. Frank, M. and Wolfe, P. (1956), An Algorithm for Quadratic Programming, *Naval Research Logistics Quarterly* **3**, 95–110.
6. Hillier, F. S. and G. J. Lieberman (1990), *Introduction to Operations Research* (5th ed.), McGraw-Hill, New York.
7. Horst, R. T. Q. Phong, Ng. V Thoai, and J. Vries (1991), On Solving a d.c. Programming Problem by a Sequence of Linear Programs, *Journal of Global Optimization* **1**, 183–203.
8. Horst, R. and H. Tuy (1990), *Global Optimization: Deterministic Approaches*, Springer-Verlag, Berlin.
9. Katoh, N. and T. Ibaraki (1987), A Parametric Characterization of an ε-Approximation Scheme for the Minimization of a Quasiconcave Program, *Discrete Applied Mathematics* **17**, 39–66.
10. Konno, H. and T. Kuno (1992), Linear Multiplicative Programming, *Mathematical Programming A* **56**, 51–65.
11. Konno, H., Y. Yajima, and T. Matsui (1991), Parametric Simplex Algorithms for Solving a Special Class of Nonconvex Minimization Problems, *Journal of Global Optimization* **1**, 65–81.
12. Kuno, T. and H. Konno (1991), A Parametric Successive Underestimation Method for Convex Multiplicative Programming Problems, *Journal of Global Optimization* **1**, 267–285.
13. Liebman J., L. Lasdon, L. Schrage, and A. Warren (1986), *Modelling and Optimization with GINO*, The Scientific Press, San Francisco.
14. Macalalag, E. and M. Sniedovich (1991), On the Importance of Being a Sensitive lp Package, Preprint Series No. 3-1991, Department of Mathematics, The University of Melbourne.
15. Schaible, S. (1976), Fractional Programming.I, Duality, *Management Science* **22**, 858–867.
16. Schrage, L., (1991), *LINDO: An Optimization Modelling System* (4th ed.), The Scientific Press, San Francisco.
17. Sniedovich, M. (1985), C-Programming: An Outline, *Operations Research Letters* **4**(1), 19–21.
18. Sniedovich, M. (1986), C-Programming and the Minimization of Pseudolinear and Additive Concave Functions, *Operations Research Letters* **5**(4), 185–189.
19. Sniedovich, M. (1989), Analysis of a Class of Fractional Programming Problems, *Mathematical Programming* **43**(3), A, 329–347.
20. Sniedovich, M. (1992), *Dynamic Programming*, Marcel Dekker, New York.
21. Yajima, Y. and H. Konno (1991), Efficient Algorithms for Solving Rank Two and Rank Three Bilinear Programming Problems, *Journal of Global Optimization* **1**, 155–171.