

AD-A039 164

MASSACHUSETTS INST OF TECH CAMBRIDGE OPERATIONS RESE--ETC F/G 12/2  
ON THE SOLUTION OF CONVEX KNAPSACK PROBLEMS WITH BOUNDED VARIAB--ETC(U)  
APR 77 G R BITRAN, A C HAX N00014-75-C-0556  
TR-129 NL

UNCLASSIFIED

| OF |  
ADA  
039/64



END

DATE  
FILMED  
5-77

ADA 039164

*P* *R*

ON THE SOLUTION OF  
CONVEX KNAPSACK PROBLEMS  
WITH BOUNDED VARIABLES

by  
GABRIEL R. BITRAN  
AND  
ARNALDO C. HAX

Technical Report No. 129  
OPERATIONS RESEARCH CENTER

DDC  
MAY 10 1977  
C

DDC FILE COPY



MASSACHUSETTS INSTITUTE  
OF  
TECHNOLOGY

April 1977

DISTRIBUTION STATEMENT A  
Approved for public release;  
Distribution Unlimited

Unclassified

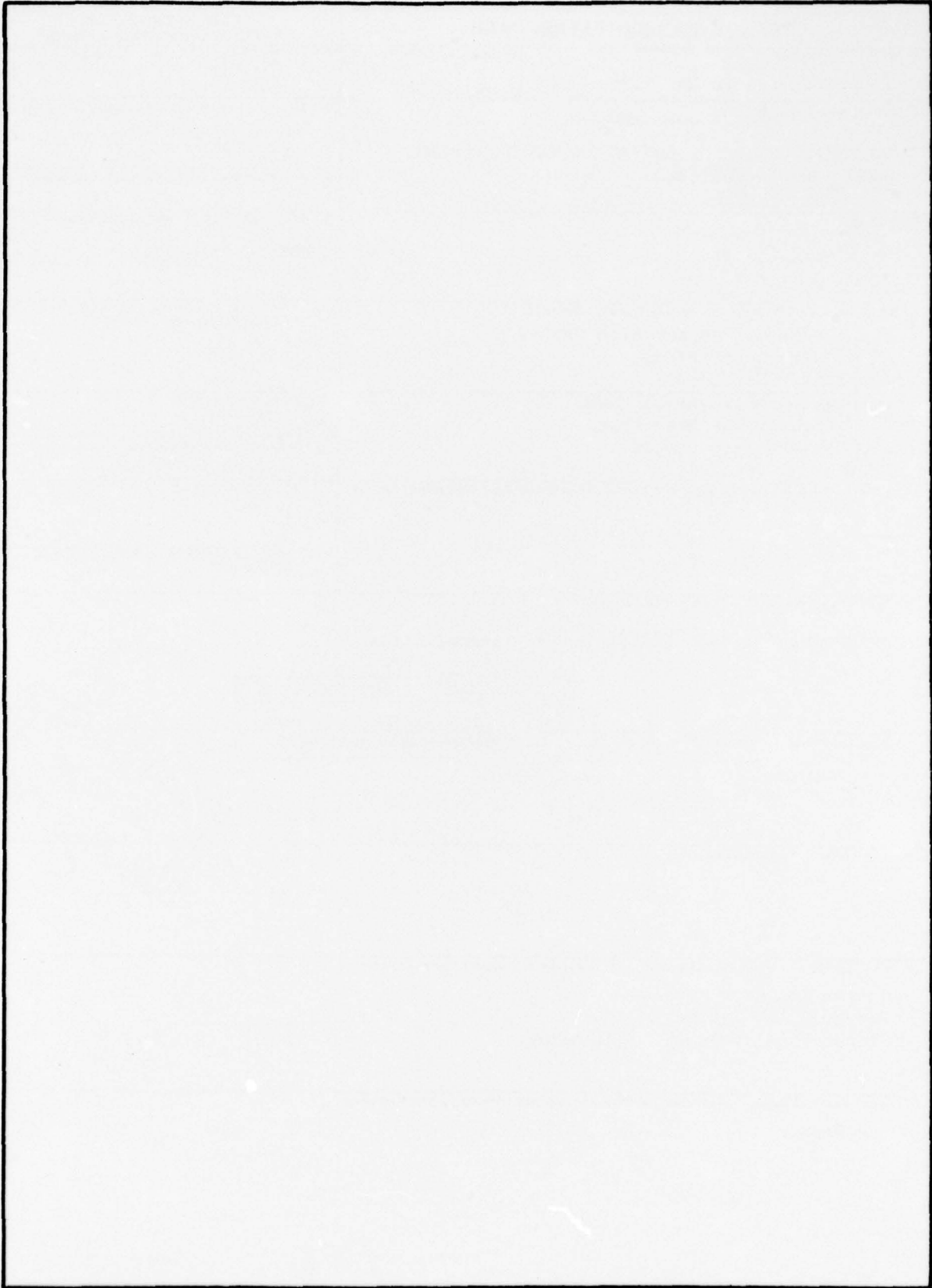
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Report No. 129	2. GOVT ACCESSION NO. (14)TR-129	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ON THE SOLUTION OF CONVEX KNAPSACK PROBLEMS WITH BOUNDED VARIABLES	5. TYPE OF REPORT & PERIOD COVERED Technical Report April 1977	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Gabriel R. Bitran Arnoldo C. Hax	8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0556	
9. PERFORMING ORGANIZATION NAME AND ADDRESS M.I.T. Operations Research Center 77 Massachusetts Avenue Cambridge, MA 02139	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 347-027	
11. CONTROLLING OFFICE NAME AND ADDRESS O.R. Branch, ONR Navy Dept. 800 North Quincy Street Arlington, VA 22217	12. REPORT DATE April 1977	13. NUMBER OF PAGES 14 pages (125) 17p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report)	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Releasable without limitation on dissemination.		
<div style="border: 1px solid black; padding: 5px; display: inline-block;">           DISTRIBUTION STATEMENT A            Approved for public release;         </div>		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 14, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Convex Knapsack Problems Resource Allocation Hierarchical Production Planning		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) See Page		

DDC  
APPROVED  
MAY 10 1977  
RECEIVED

270720 Inc

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)



SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

ON THE SOLUTION OF CONVEX KNAPSACK PROBLEMS  
WITH BOUNDED VARIABLES

by

GABRIEL R. BITRAN\*

and

ARNOLDO C. HAX

Technical Report No. 129

Work Performed Under

Contract N00014-75-C-0556, Office of Naval Research

Multilevel Logistics Organization Models

NR 347-027

M.I.T. OSP 82491

Operations Research Center

Massachusetts Institute of Technology

Cambridge, Massachusetts 02139

April 1977

ACCESSION NO.	
NTIS	White Section <input checked="" type="checkbox"/>
DIC	Buff Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	<input type="checkbox"/>
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	AVAIL. and/or SPECIAL
A	

\*Presently at "Universidade de Sao Paulo (Dept. de Engenharia de Producao)", Brazil

Reproduction in whole or in part is permitted for any purpose  
of the United States Government.

FOREWORD

The Operations Research Center at the Massachusetts Institute of Technology is an interdepartmental activity devoted to graduate education and research in the field of operations research. The work of the Center is supported, in part, by government contracts and industrial grants-in-aid. The work reported herein was supported (in part) by the Office of Naval Research under Contract N00014-75-C-0556.

Jeremy F. Shapiro  
Acting Director

ABSTRACT

✓ In this paper, ~~we present~~ a recursive method<sup>is presented</sup> to solve separable differentiable convex knapsack problems with bounded variables. The method differs from classical optimization algorithms of convex programming and determines at each iteration the optimal value of at least one variable. Applications of such problems are frequent in resource allocation and recently have shown to be useful in hierarchical production planning. Computational results are presented. ↑

ON THE SOLUTION OF CONVEX KNAPSACK PROBLEMS  
WITH BOUNDED VARIABLES

Gabriel R. Bitran and Arnoldo C. Hax  
Massachusetts Institute of Technology

In this paper we present a recursive method to solve the problem

$$(N) : z = \min \sum_{j \in J^{\circ}} G_j(x_j)$$

$$\sum_{j \in J^{\circ}} x_j = P^{\circ} \quad (1)$$

$$lb_j \leq x_j \leq ub_j \quad j \in J^{\circ} \quad (2)$$

$$x_j \in X_j \quad j \in J^{\circ}$$

where  $G_j(\cdot)$   $j \in J^{\circ}$  are differentiable convex functions on the open convex set  $X_j \subseteq R$   $j \in J^{\circ}$  respectively.  $ub_j \geq lb_j$   $j \in J^{\circ}$  and  $\sum_{j \in J^{\circ}} lb_j < P^{\circ} < \sum_{j \in J^{\circ}} ub_j$  (otherwise the problem is trivial or infeasible).

The derivative of  $G_j$  at  $x_j$  is denoted by  $DG_j(x_j)$ . The iterative method proposed to solve (N) differs from classical convex programming algorithms and at each iteration determines the optimal value of at least one variable. This problem was formerly treated, for particular objective functions, by convex programming arguments [3] and dynamic programming [6]. More recently Luss and Gupta [4] presented an iterative method for strictly convex decreasing functions and a one pass algorithm for a set of particular functions with the variables bounded from below. Luss and Gupta's method consists in relaxing the upper bounding constraints. The algorithm proposed in this paper applies for a more general set of functions. At each iteration it solves a simpler problem of the form  $\min \{ \sum G_j(x_j) \text{ st } \sum x_j = P, x_j \in X_j \}$  obtained by relaxing both bounds (2). At the end of each iteration (except possibly the last) we show that either the subset of variables  $\{x_j : x_j \geq ub_j\}$  have optimal value  $ub_j$  in (N) or the subset  $\{x_j : x_j \leq lb_j\}$  have optimal value  $lb_j$  in (N). Applications of this class of knapsack problems to resource allocation can be found in [3], [4] and to hierarchical production planning in [2].

(N) is a convex problem and is regular [1]. Thus the Kuhn-Tucker conditions (3)-(9), below, are necessary and sufficient for optimality of  $x_j^*$   $j \in J^{\circ}$  in (N).

$$DG_j(x_j^*) + \lambda + u_j - \tau_j = 0 \quad j \in J^0 \quad (3)$$

$$u_j(x_j^* - ub_j) = 0 \quad j \in J^0 \quad (4)$$

$$\tau_j(lb_j - x_j^*) = 0 \quad j \in J^0 \quad (5)$$

$$\sum_{j \in J^0} x_j^* = p^0 \quad (6)$$

$$lb_j \leq x_j^* \leq ub_j \quad j \in J^0 \quad (7)$$

$$\lambda \in R, u_j \geq 0, \tau_j \geq 0 \quad j \in J^0 \quad (8)$$

$$x_j^* \in X_j \quad j \in J^0 \quad (9)$$

We first state the algorithm and then prove that it is optimal.

Algorithm

Initialization:  $J^1 = J^0, p^1 = p^0$

Iteration  $\beta$  ( $\beta=1,2,3,\dots$ )

Step 1: Solve  $N(\beta) : \min \{ \sum_{j \in J^0} G_j(x_j) : \sum_{j \in J^0} x_j = p^\beta, x_j \in X_j, j \in J^0 \}$

and let the solution be  $x_j^\beta, j \in J^0$ . If  $lb_j \leq x_j^\beta \leq ub_j, j \in J^0$  define

$x_j^* = x_j^\beta, j \in J^0$  and stop the solution  $x_j^*, j \in J^0$  generated by the algorithm is optimal. Otherwise go to step 2.

Step 2: Compute

$$\Delta^\beta = \sum_{j \in J_+^\beta} (x_j^\beta - ub_j) \text{ where } J_+^\beta = \{j \in J^0 : x_j^\beta \geq ub_j\}$$

and

$$\nabla^\beta = \sum_{j \in J_-^\beta} (lb_j - x_j^\beta) \text{ where } J_-^\beta = \{j \in J^0 : x_j^\beta \leq lb_j\}$$

Step 3: If  $\Delta^\beta \geq \nabla^\beta$  define  $x_j^* = ub_j, j \in J_+^\beta$  and let

$$J^{\beta+1} = J^0 - J_+^\beta, p^{\beta+1} = p^\beta - \sum_{j \in J_+^\beta} ub_j.$$

If  $\Delta^\beta < \nabla^\beta$  define  $x_j^* = lb_j \quad j \in J_-^\beta$  and let

$$J^{\beta+1} = J_-^\beta - J_-^\beta, \quad p^{\beta+1} = p^\beta - \sum_{j \in J_-^\beta} lb_j.$$

If  $J^{\beta+1} = \emptyset$  stop. The solution  $x_j^* \quad j \in J^0$  generated by the algorithm is optimal. Otherwise let  $\beta = \beta+1$  and go to step 1.

Since at each iteration the set  $J^\beta$  is reduced by at least one element the algorithm is finite. To prove that  $x_j^* \quad j \in J^0$ , generated by the algorithm, solve (N) we construct a corresponding Kuhn-Tucker vector through the following results. Let  $\lambda^\beta$  be the Kuhn-Tucker multiplier associated to the knapsack constraint in  $N(\beta)$ .

Lemma 1: If at iteration  $\beta \quad \Delta^\beta \geq \nabla^\beta$  then

- a) for any  $s \in J_+^\beta$  we have  $-DG_s(ub_s) \geq -DG_i(ub_i)$  for all  $i \in J_-^\beta - J_+^\beta$   
 b)  $\lambda^{\beta+1} \leq \lambda^\beta$

Proof: a)  $\lambda^\beta = -DG_j(x_j^\beta) \quad j \in J^\beta$ . Let  $s \in J_+^\beta$  and  $i \in J_-^\beta - J_+^\beta$ . Then, since for convex functions [5]

$$[DG_j(x_j^2) - DG_j(x_j^1)](x_j^2 - x_j^1) \geq 0 \quad (10)$$

it follows that

$$-DG_s(ub_s) \geq -DG_s(x_s^\beta) = \lambda^\beta = -DG_i(x_i^\beta) \geq -DG_i(ub_i)$$

$$b) \quad \sum_{j \in J_-^\beta - J_+^\beta} x_j^\beta = p^\beta - \sum_{j \in J_+^\beta} x_j^\beta \leq p^\beta - \sum_{j \in J_+^\beta} ub_j = p^{\beta+1} = \sum_{j \in J^{\beta+1}} x_j^{\beta+1}$$

For at least one  $j \in J^{\beta+1}$  we have  $x_j^{\beta+1} \geq x_j^\beta$ . Thus

$$\lambda^{\beta+1} = -DG_j(x_j^{\beta+1}) \leq -DG_j(x_j^\beta) = \lambda^\beta$$

where the inequality follows from (10). ▀

Lemma 2: If at iteration  $\beta \quad \Delta^\beta < \nabla^\beta$  then

- a) for any  $s \in J_-^\beta$  we have  $-DG_s(lb_s) \leq -DG_i(lb_i)$  for all  $i \in J_-^\beta - J_-^\beta$   
 b)  $\lambda^{\beta+1} \geq \lambda^\beta$

The proofs of this lemma and of theorem 4 are omitted because they are similar to those of lemma 2 and theorem 3 respectively.

Theorem 3: Assume that  $\Delta^\beta \geq \nabla^\beta$ ,  $\Delta^i < \nabla^i$   $i=\beta+1, \beta+2, \dots, \gamma-1$ . Then

a)  $J^\gamma \supseteq (J^{\beta-J^{\beta-J^{\beta}}})$

b)  $\lambda^\beta \geq \lambda^\gamma$

Proof: 
$$P^{\beta+1} = \sum_{j \in J^{\beta+1}} x_j^{\beta+1} = \sum_{j \in J^{\beta-J^{\beta-J^{\beta}}}} x_j^{\beta} - \sum_{j \in J^{\beta+}} u b_j + \sum_{j \in J^{\beta+}} x_j^{\beta}$$

$$= \sum_{j \in J^{\beta+1}} x_j^{\beta+\Delta^\beta} \tag{11}$$

$$P^{\beta+2} = \sum_{j \in J^{\beta+2}} x_j^{\beta+2} = \sum_{j \in J^{\beta+1}} x_j^{\beta+1} - \sum_{j \in J^{\beta+1}} l b_j =$$

$$= \sum_{j \in J^{\beta+1-J^{\beta+1}}} x_j^{\beta} + \sum_{j \in J^{\beta+1}} x_j^{\beta} + \Delta^\beta - \sum_{j \in J^{\beta+1}} l b_j.$$

Thus, since  $J^{\beta+2} = J^{\beta+1-J^{\beta+1}}$

$$P^{\beta+2} = \sum_{j \in J^{\beta+2}} x_j^{\beta} + \Delta^\beta - \sum_{j \in J^{\beta+1}} (l b_j - x_j^{\beta}). \tag{12}$$

Similarly we obtain

$$P^\gamma = \sum_{j \in J^\gamma} x_j^{\beta} + \Delta^\beta - \sum_{s=\beta+1}^{\gamma-1} \sum_{j \in J^s} (l b_j - x_j^{\beta}) \tag{13}$$

From (11) it follows that for at least one  $j_0 \in J^{\beta+1}$

$$x_{j_0}^{\beta+1} \geq x_{j_0}^{\beta} \tag{14}$$

But, from the Kuhn-Tucker conditions for  $N(\beta+1)$ :

$$-DG_i(x_i^{\beta+1}) = \lambda^{\beta+1} \quad i \in J^{\beta+1} \tag{15}$$

Combining (10), (14), and (15) it follows that

$$x_j^{\beta+1} \geq x_j^{\beta} \quad \text{for all } j \in J^{\beta+1}$$

If all functions  $G_j$  are not strictly convex it is possible that  $-DG_{d_0}(x_{d_0}^{\beta+1}) = \lambda^{\beta+1} = \lambda^\beta$ . In this case  $N(\beta+1)$  may have more than one optimal solution. However at least one will satisfy this condition.

$$\text{Thus } J_-^{\beta+1} \subseteq J_-^\beta \text{ and } \Delta^\beta - \sum_{j \in J_-^{\beta+1}} (\ell b_j - x_j^\beta) \geq 0. \quad (16)$$

(12), (16) and the fact that  $J^{\beta+2} = J^{\beta+1} - J_-^{\beta+1}$  imply that

$$x_j^{\beta+2} \geq x_j^\beta \text{ for all } j \in J^{\beta+2}, J_-^{\beta+2} \subseteq J_-^\beta \text{ and}$$

$$\Delta^\beta - \sum_{j \in J_-^{\beta+1}} (\ell b_j - x_j^\beta) - \sum_{j \in J_-^{\beta+2}} (\ell b_j - x_j^\beta) \geq 0 \text{ thus}$$

$$P^{\beta+3} = \sum_{j \in J^{\beta+3}} x_j^{\beta+3} \geq \sum_{j \in J^{\beta+3}} x_j^\beta$$

Continuing with the same reasoning we obtain

$$J_-^i \subseteq J_-^\beta \quad i=\beta+1, \dots, \gamma-1; \Delta^\beta - \sum_{s=\beta+1}^{\gamma-1} \sum_{j \in J_-^s} (\ell b_j - x_j^\beta) \geq 0$$

and from (13)

$$P^\gamma = \sum_{j \in J^\gamma} x_j^\gamma \geq \sum_{j \in J^\gamma} x_j^\beta. \quad (17)$$

These conclusions together with  $J^{\beta+1} = J^\beta - J_+^\beta$  and  $J^{i+1} = J^i - J_-^i \quad i=\beta+1, \dots, \gamma-1$  prove part a).

From (17) it follows  $x_j^\gamma \geq x_j^\beta \quad j \in J^\gamma$ . Thus, from (10) and (15) with  $\gamma$  instead of  $\beta+1$

$$\lambda^\gamma = -DG_j(x_j^\gamma) \leq -DG_j(x_j^\beta) = \lambda^\beta \quad \blacksquare$$

**Theorem 4:** Assume that  $\Delta^\beta < \nabla^\beta$ ,  $\Delta^i \geq \nabla^i \quad i=\beta+1, \beta+2, \dots, \gamma-1$ . Then

a)  $J^\gamma \supseteq (J^\beta - J_-^\beta - J_+^\beta)$

b)  $\lambda^\beta \leq \lambda^\gamma$

Theorem 5: The set  $x_j^* \quad j \in J^0$  generated by the algorithm is optimal in (N).

Proof: By lemmas 1 and 2, theorems 3b) and 4b) the set  $x_j^* \quad j \in J^0$  generated by the algorithm has the following property:

$$\underbrace{-DG_{k_1}(ub_{k_1}) \geq \dots \geq -DG_{k_p}(ub_{k_p})}_{\text{corresponding to variables defined at upper bound}} \geq \underbrace{-DG_{v_1}(x_{v_1}^*) = \dots = -DG_{v_g}(x_{v_g}^*)}_{\text{corresponding to variables not defined at any iteration at upper or lower bound}} \geq \underbrace{-DG_{i_1}(lb_{i_1}) \geq \dots \geq -DG_{i_s}(lb_{i_s})}_{\text{corresponding to variables defined at lower bound}}$$

corresponding to variables defined at upper bound      corresponding to variables not defined at any iteration at upper or lower bound      corresponding to variables defined at lower bound

The  $x_{v_j}^* \quad j=1,2,\dots,g$  are obtained at the last iteration.

To see that conditions (3)-(9) are satisfied take

$$\lambda = -DG_{v_1}(x_{v_1}^*)$$

$$\tau_{i_j} = \lambda + DG_{i_j}(lb_{i_j}) \geq 0, \quad u_{i_j} = 0 \quad j=1,2,\dots,s$$

$$u_{k_j} = -DG_{k_j}(ub_{k_j}) - \lambda \geq 0, \quad \tau_{k_j} = 0 \quad j=1,2,\dots,p \quad \text{and}$$

$$\tau_{v_j} = u_{v_j} = 0 \quad j=1,2,\dots,g.$$

Thus, since  $x_j^* \quad j \in J^0$  also satisfies (6), (7), and (9) it follows that

$$x_{k_j} = ub_{k_j} \quad j=1,2,\dots,p; \quad x_{v_j} = x_{v_j}^* \quad j=1,2,\dots,g \quad \text{and}$$

$$x_{i_j} = lb_{i_j} \quad j=1,2,\dots,s \quad \text{solve (N).} \quad \square$$

The algorithm depends strongly on the existence of a solution to  $N(\beta)$ . However when among the  $G_j(\cdot) \quad j \in J^\beta$  there are strictly increasing and decreasing functions  $N(\beta)$  has no solution. The next theorem shows how to cope with this difficulty.

Let  $J_1 = \{j \in J^0 : G_j(\cdot) \text{ is strictly increasing}\} \neq \emptyset$  and

$J_2 = \{j \in J^0 : G_j(\cdot) \text{ is strictly decreasing}\} \neq \emptyset$

Assume that (N) has an optimal solution  $x_j^* \quad j \in J^0$ . Then

Theorem 6: The optimal solution of (N) is such that either

$$x_i^* = ub_i \quad i \in J_2 \text{ and/or } x_i^* = lb_i \quad i \in J_1.$$

Proof: The optimal solution satisfies (3)-(9) and in particular

$$\begin{aligned} -DG_j(ub_j) &= \lambda + u_j \quad j \in J(ub) \equiv \{j \in J^0 : x_j^* = ub_j\} \\ -DG_j(lb_j) &= \lambda - \tau_j \quad j \in J(lb) = \{j \in J^0 : x_j^* = lb_j\} \text{ and} \\ -DG_j(x_j^*) &= \lambda \quad j \in J^0 - J(lb) - J(ub). \end{aligned}$$

Note that the assumption  $P^0 > \sum_{j \in J^0} lb_j$  implies  $J^0 - J(lb) \neq \emptyset$ .

If  $\min \{-DG_j(x_j^*) \mid j \in J^0 - J(lb)\} \geq 0$ , since  $-DG_i(x_i^*) < 0 \quad i \in J_1$ ,

it follows that  $J_1 \subseteq J(lb)$ . Otherwise, since  $-DG_i(x_i^*) > 0 \quad i \in J_2$

and  $-DG_j(ub_j) \geq -DG_i(x_i^*) \geq -DG_k(lb_k)$  for any  $j \in J(ub) \in J^0 - J(ub) - J(lb)$

and  $k \in J(lb)$ , we have that  $J_2 \subseteq J(ub)$ . ▣

A direct consequence of theorem 6 is that the solution to (N) can be obtained by solving the following two problems

$$(N_1) : z_1 = \min \left\{ \sum_{j \in J^0 - J_2} G_j(x_j) : \sum_{j \in J^0 - J_2} x_j = P^0 - \sum_{j \in J_2} ub_j, \quad lb_j < x_j < ub_j \quad j \in J^0 - J_2 \right\}$$

and

$$(N_2) : z_2 = \min \left\{ \sum_{j \in J^0 - J_1} G_j(x_j) : \sum_{j \in J^0 - J_1} x_j = P^0 - \sum_{j \in J_1} lb_j, \quad lb_j < x_j < ub_j \quad j \in J^0 - J_1 \right\}$$

and taking  $z = \min(z_1, z_2)$ .

Let  $(N_{\leq})$  and  $(N_{\geq})$  be the versions of (N) when constraint (1) is an inequality of the type  $\leq$  and  $\geq$  respectively. For a variable  $j \in J^0$  let  $h_j$  be the value of  $x_j$  for which  $DG_j(x_j) = 0$ . If such a point does not exist we adopt  $h_j = -\infty$  ( $+\infty$ ) in Theorem 7 (Theorem 8). Let  $x_j^* \mid j \in J^0$  solve (N). Then

Theorem 7: a) If  $\lambda = -DG_{v_j}(x_{v_j}^*) \leq 0$ ,  $x_j = x_j^* \mid j \in J^0$  solve  $(N_{\geq})$ .

b) If  $\lambda = -DG_{v_j}(x_{v_j}^*) < 0$ ,  $x_j$  defined as:

$$x_{i_j} = lb_{i_j} \quad j=1,2,\dots,s;$$

$$x_{v_j} = \max(lb_{v_j}, h_{v_j}) \quad j=1,2,\dots,g;$$

$$x_{k_j} = \max(lb_{k_j}, h_{k_j}) \text{ for all } k_j \text{ such that } DG_{k_j}(ub_{k_j}) \geq 0 \text{ and}$$

$$x_{k_j} = ub_{k_j} \text{ for all } k_j \text{ such that } DT_{k_j}(ub_{k_j}) < 0$$

solve ( $N_{\leq}$ ).

Theorem 8: a) If  $\lambda = -DG_{v_j}(x_{v_j}^*) \leq 0$ ,  $x_j = x_j^* \quad j \in J^0$  solves ( $N_{\geq}$ ).

b) If  $\lambda = -DG_{v_j}(x_{v_j}^*) > 0$ ,  $x_j$  defined as:

$$x_{k_j} = ub_{k_j} \quad j=1,2,\dots,p;$$

$$x_{v_j} = \min(ub_{v_j}, h_{v_j}) \quad j=1,2,\dots,g;$$

$$x_{i_j} = \min(ub_{i_j}, h_{i_j}) \text{ for all } i_j \text{ such that } DG_{i_j}(lb_{i_j}) \leq 0 \text{ and}$$

$$x_{i_j} = lb_{i_j} \text{ for all } i_j \text{ such that } DG_{i_j}(lb_{i_j}) > 0$$

solve ( $N_{\geq}$ ).

Computational Results

In Tables 1 to 7 we present the results of 84 problems of type (N). The data were randomly generated. For identification purposes, our algorithm was denominated Bitran-Hax. In each problem the objective function was composed by functions  $G_i(x_i)$  of the same family. They are indicated in the first row of each table. Following the time in seconds to solve each problem, by both methods, appears the number of iterations required.  $n$  represents the number of variables in a problem. For a fixed  $n$  three problems were solved for each type of objective function. In Luss and Gupta's method [4] the ordering of the derivatives evaluated at the lower bound of each variable was executed by the "Quick Method". In our algorithm, problems N(8) were solved through the corresponding Kuhn-Tucker conditions. For the problems of Tables 6 and 7, Luss and Gupta's algorithm does not apply because in the former one the objective functions are strictly convex increasing and in the later one we have not imposed any condition among the values of the bounds,  $P^0$  and the point where each of the  $G_i(x_i)$  attains its minimum. The computer used is a Borroughs B6700. The programs were written in Algol. Application of problems presented in Tables 5 and 7 to hierarchical production planning can be found in [2]. The parameters ( $s_i, m_i, lb_i, ub_i, etc....$ ) corresponding to problems in Tables 1 to 5 (6 and 7) were randomly generated in intervals where the functions  $G_i(x_i)$  are strictly convex decreasing (strictly convex).

$$G_i(x_i) = s_i [1 - \exp(-m_i x_i)] \quad i=1, 2, \dots, n$$

	n = 50			n = 100			n = 150			n = 200		
	Bitran-Hax (sec.)	0.138	0.157	0.119	0.292	0.213	0.237	0.482	0.540	0.434	0.709	0.720
No. of Iterations	6	7	7	6	6	6	8	9	7	8	8	9
Luss-Gupta (sec.)	0.728	1.029	1.174	1.125	4.179	0.830	5.903	9.073	3.969	10.531	2.109	15.124
No. of Iterations	2	4	8	1	7	1	3	4	2	3	1	5

Table 1

$$G_i(x_i) = s_i \lambda n [1 + m_i x_i] \quad i=1, 2, \dots, n$$

	n = 50			n = 100			n = 150			n = 200		
	Bitran-Hax (sec.)	0.051	0.050	0.038	0.118	0.099	0.131	0.163	0.178	0.162	0.191	0.245
No. of Iterations	4	4	3	6	5	6	6	6	4	7	6	5
Luss-Gupta (sec.)	0.486	0.447	0.201	0.516	1.316	0.488	0.536	0.958	1.232	3.394	4.210	1.105
No. of Iterations	3	3	1	2	4	2	1	2	1	6	4	1

Table 2

$$G_i(x_i) = s_i \frac{x_i + c_i}{x_i + m_i} \text{ with } (m_i > c_i) \quad i=1,2,\dots,n$$

	n = 50			n = 100			n = 150			n = 200		
	Bitran-Hax (sec.)	0.113	0.095	0.124	0.208	0.221	0.178	0.308	0.307	0.298	0.400	0.421
No. of iterations	6	5	5	5	5	5	4	6	4	5	6	3
Luss-Gupta (sec.)	0.210	0.284	0.229	0.825	0.796	0.291	1.033	2.687	1.098	1.394	5.448	2.543
No. of iterations	1	1	1	3	2	1	1	4	1	1	2	1

Table 3

$$G_i(x_i) = x_i x_{i-1}^{-m_i} x_i^2 \text{ with } lb_i < k_i \text{ and } P < \sum_{i=1}^n mn(\text{ub}_i, k_i) \text{ where } k_i = \frac{s_i}{2m_i}$$

	n = 50			n = 100			n = 150			n = 200		
	Bitran-Hax (sec.)	0.063	0.059	0.049	0.105	0.108	0.101	0.152	0.161	0.157	0.223	0.261
No. of iterations	7	7	6	8	7	6	8	8	8	8	8	8
Luss-Gupta (sec.)	0.225	0.236	0.164	0.361	0.850	0.283	0.737	1.129	0.693	2.150	0.893	0.745
No. of iterations	16	12	1	5	34	1	10	13	10	18	1	7

Table 4

$$G_i(x_i) = a_i/x_i \quad i=1,2,\dots,n$$

	n = 50			n = 100			n = 150			n = 200		
	Bitran-Hax (sec.)	No. of iterations		Bitran-Hax (sec.)	No. of iterations		Bitran-Hax (sec.)	No. of iterations		Bitran-Hax (sec.)	No. of iterations	
Bitran-Hax (sec.)	0.061	0.058	0.048	0.095	0.076	0.133	0.128	0.246	0.168	0.276	0.306	0.282
No. of iterations	6	3	4	5	6	7	3	6	5	8	4	3
Luss-Gupta (sec.)	0.624	0.752	0.916	2.611	2.012	1.431	2.618	2.715	4.869	3.719	3.901	5.088
No. of iterations	2	1	3	4	2	1	2	5	5	6	2	1

Table 5

$$G_i(x_i) = \exp k_i x_i \quad k_i > 0 \quad i=1,2,\dots,n$$

	n = 50			n = 100			n = 150			n = 200		
	Bitran-Hax (sec.)	No. of iterations		Bitran-Hax (sec.)	No. of iterations		Bitran-Hax (sec.)	No. of iterations		Bitran-Hax (sec.)	No. of iterations	
Bitran-Hax (sec.)	0.163	0.151	0.135	0.237	0.267	0.261	0.445	0.443	0.381	0.596	0.486	0.536
No. of iterations	7	7	6	5	6	6	7	7	7	8	6	6

Table 6

$$G_i(x_i) = \frac{1}{2} \left( \frac{h}{s} - \frac{x_i}{s_i} \right)^2 \quad i=1,2,\dots,n$$

	n = 50			n = 100			n = 150			n = 200		
	Bitran-Hax (sec.)	No. of iterations		Bitran-Hax (sec.)	No. of iterations		Bitran-Hax (sec.)	No. of iterations		Bitran-Hax (sec.)	No. of iterations	
Bitran-Hax (sec.)	0.054	0.050	0.044	0.114	0.073	0.073	0.159	0.151	0.170	0.202	0.214	0.227
No. of iterations	6	6	5	6	3	3	5	4	5	4	5	8

Table 7

References

1. G. R. Bitran, "Admissible Points and Vector Optimization: A Unified Approach", unpublished Ph.D. thesis, Operations Research Center, M.I.T., February, 1975.
2. G. R. Bitran and A. C. Hax, "On the Design of Hierarchical Production Planning Systems", to appear in January 1977.
3. A. Charnes and W. W. Cooper, "The Theory of Search: Optimum Distribution of Search Effort", Management Science 5, 44-49 (1958).
4. H. Luss and S. K. Gupta, "Allocation of Effort Resources Among Competing Activities", Operations Research 23, 360-366 (1975).
5. O. L. Mangasarian, "Nonlinear Programming", McGraw-Hill, New York, 1969.
6. C. Wilkinson and S. K. Gupta, "Allocating Promotional Effort to Competing Activities: A Dynamic Programming Approach", IFORS Conference, Venice, 419-432 (1969).