Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 4, No. 1, pp. 63 – 73, January 2008

COMPUTATIONAL STUDY OF SURROGATE DUAL METHOD FOR MULTI-DIMENSIONAL NONLINEAR KNAPSACK PROBLEMS [*]

Xiaoling Sun[†], Shanshan Kong and Duan Li

Dedicated to Professor Liansheng Zhang on the occasion of his 70th birthday.

Abstract: Multi-dimensional nonlinear knapsack problems are often encountered in real-world applications such as in resource allocation, industrial planning and reliability networks. In this paper, we propose a new exact method for solving this class of problems . The method is based on surrogate dual search and domain cut technique. The optimal surrogate multipliers are computed by the cutting plane method, where the surrogate relaxation problem is solved by the 0-1 linearization method in convex cases. Numerical results are reported for medium-size multi-dimensional nonlinear convex knapsack problems.

Key words: *nonlinear knapsack problem, surrogate dual, Lagrangian dual, branch-and-bound method*

Mathematics Subject Classification: *90C10, 90C46*

1 Introduction

Consider the following multi-dimensional nonlinear knapsack problem:

$$
\begin{aligned}
(P) \qquad & \max\ f(x) = \sum_{j=1}^{n} f_j(x_j) \\
& \text{s.t. } g_i(x) = \sum_{j=1}^{n} g_{ij}(x_j) \le b_i,\ i = 1,\dots,m, \\
& \qquad x \in X = \{x \mid\ 0 \le x_j \le u_j,\ x_j \text{ integer},\ j = 1,\dots,n\},
\end{aligned}
$$

where all f_j and all g_{ij} are nondecreasing functions on $[0, u_j]$ for $j = 1,\dots,n$, $i = 1,\dots,m$, and u_j is the upper bound for x_j, $j = 1,\dots,n$. The constraints, $g_i(x) \le b_i$, $i = 1,\dots,m$, are called *resource constraints*. Problem (P) is called *convex* multi-dimensional knapsack problem if f is concave and all g_i's are convex on $conv(X)$. Without loss of generality, we assume that $f_j(0) = g_{ij}(0) = 0$ for all i and j. When $m = 1$, problem (P) reduces to the classical nonlinear knapsack problem. It is noticed that the general nonlinear knapsack

[*] Research supported by the National Natural Science Foundation of China under Grants 70671064 and 10571116, and by Research Grants Council of Hong Kong, Ref. No. 413606.

[†] Corresponding author.

problem is NP-hard since the classical linear knapsack problem is NP-hard. Therefore, developing exact method for (P) is a computational challenge.

Nonlinear knapsack problem and its continuous version have numerous applications in various fields, such as production planning [30], capital budgeting [21], marketing [17], stratified sampling [1] and reliability networks [26, 29]. Existing methods in the literature for (P) have been mainly developed for nonlinear knapsack problems with a single resource constraint, for example, continuous relaxation based branch-and-bound methods [2, 3] and dynamic programming methods [10, 14]. In real-world applications, multiple resource constraints are often encountered and are always indispensable for modeling optimization problems. For example, in reliability network optimization, three types of resource constraints: cost, volume and weight, are necessary to be considered simultaneously. However, only a few exact methods have been proposed for solving multi-dimensional nonlinear knapsack problems. In [4], a surrogate relaxation based method was proposed for multi-dimensional quadratic 0-1 problem and certain optimality criteria were derived for the branch-and-bound method. A dynamic programming method, combined with a branch-and-bound strategy, was proposed in [18] for the general case of (P). This method recursively generates efficient feasible solutions of the problem and removes inefficient feasible solutions by certain dominance rules. We point out that the dynamic programming method is inefficient for (P) when there are more than two resource constraints, due to "the curse of dimensionality."

In the context of linear knapsack problems, multi-dimensional cases have been considered by various authors. Exact methods for linear knapsack problems with two resource constraints were proposed in [7, 20, 28]. Surrogate relaxation techniques, combined with cutting and variable fixation, were investigated in [6, 8, 22, 23] for solving multi-dimensional 0-1 linear knapsack problems. One can refer to [12] for a survey on theory and methodologies for general linear knapsack problems. Recently, a convergent Lagrangian and domain cut method has been proposed in [16] for solving problem (P). In [16], the multiple resource constraints in (P) were surrogated into a single constraint using the optimal Lagrangian multipliers. Domain cut technique was then used to partition the domain into subdomains, thus reducing the duality gap and ensuring the convergence of the branch-and-bound method.

In this paper, we propose a new exact method for solving multi-dimensional nonlinear knapsack problems. The method is of a branch-and-bound framework that computes the upper bounds by surrogate dual search and eliminates the duality gap by a special domain cut technique. The cutting plane method is employed to search for the optimal surrogate multipliers, where the surrogate relaxation problems are solved by 0-1 linearization method. Our computational results show that the proposed algorithm is capable of solving medium-size multi-dimensional nonlinear knapsack problems. Favorable comparison results with the subgradient Lagrangian dual search is also reported. To our knowledge, this is the first computational implementation of the surrogate dual method for multi-dimensional nonlinear knapsack problems.

2 Surrogate Relaxation and Dual Search

The idea of surrogate constraint was first introduced by Glover [9] for 0-1 integer programming. We now extend the surrogate method to deal with multi-dimensional nonlinear knapsack problems.

Let $g(x) = (g_1(x), g_2(x), \cdots, g_m(x))^T$ and $b = (b_1, b_2, \cdots, b_m)^T$. The surrogate relaxation of (P) is defined as follows:

$$\begin{array}{lll}(S_\mu) & \max & f(x)\\ & \text{s.t.} & \mu^T(g(x)-b)\le 0,\\ & & x\in X,\end{array}$$

where $\mu=(\mu_1,\mu_2,\cdots,\mu_m)^T\in\mathbb{R}^m_+$ is the surrogate multiplier vector.

Define F and $F(\mu)$ the feasible regions of (P) and (S_μ), respectively.

$$\begin{aligned}&F=\{x\in X\mid g(x)\le b\},\\ &F(\mu)=\{x\in X\mid \mu^T(g(x)-b)\le 0\}.\end{aligned}$$

Obviously, we have $F\subseteq F(\mu)$. Thus, (S_μ) is a relaxation of (P). Denote by $v(\cdot)$ the optimal value of problem $(\cdot)$. Then

$$v(S_\mu)\ge v(P),\qquad \forall\mu\in\mathbb{R}^m_+.$$

The surrogate dual is to search for the minimum upper bound generated by (S_μ):

$$\begin{array}{lll}(D_S) & \min & v(S_\mu)\\ & \text{s.t.} & \mu\in\mathbb{R}^m_+.\end{array}$$

Recall that the conventional Lagrangian dual problem of (P) is

$$\begin{array}{lll}(D_L) & \min & v(L_\mu)\\ & \text{s.t.} & \mu\in\mathbb{R}^m_+,\end{array}$$

where the Lagrangian relaxation problem is defined as

$$(L_\mu)\qquad \max_{x\in X} f(x)+\sum_{i=1}^m \mu_i(g_i(x)-b_i).$$

The following theorem shows that the surrogate dual problem (D_S) provides a tighter bound than the Lagrangian dual problem (D_L).

Theorem 2.1 ([15, 24]). *$v(D_S)\le v(D_L)$.*

Various surrogate dual search procedures have been proposed for solving (D_S) where the original problem is linear (see [5, 11, 13, 25]). In the following, we will discuss a cutting plane method for solving the surrogate dual problem. It is clear that $v(S_\mu)=v(S_{\theta\mu})$ for any $\theta>0$. Thus, the surrogate dual problem (D_S) can be normalized to an equivalent problem with a compact feasible region:

$$\begin{array}{lll}(D_S^n) & \min & v(S_\mu)\\ & \text{s.t.} & \mu\in\Lambda,\end{array}$$

where $\Lambda=\{\mu\in\mathbb{R}^m_+\mid e^T\mu=1\}$ and $e=(1,1,\cdots,1)^T$.

For $\alpha\in\mathbb{R}^n$, let $X(\alpha)$ denote the level set of $f(x)$, $X(\alpha)=\{x\in X\mid f(x)\ge\alpha\}$. For any given $\mu\in\Lambda$ and $\alpha\in\mathbb{R}$, $v(S_\mu)\ge\alpha$ if and only if

$$F(\mu)\cap X(\alpha)\ne\emptyset. \tag{2.1}$$

Consider the following problem

$$(S(\alpha,\mu)) \qquad \begin{aligned} &\min \ \ \mu^T(g(x)-b) \\ &\text{s.t.} \ \ x\in X(\alpha). \end{aligned}$$

It is easy to see that (2.1) holds if and only if $v(S(\alpha,\mu))\leq 0$. Since $v(D_S^n)=\min\{v(S_\mu)\mid \mu\in\Lambda\}$, it follows that $v(D_S^n)\geq\alpha$ if and only if $v(S(\alpha,\mu))\leq 0$ for all $\mu\in\Lambda$. Define the following dual problem:

$$(D(\alpha)) \qquad \begin{aligned} &\max \ \ v(S(\alpha,\mu)) \\ &\text{s.t.} \ \ \mu\in\Lambda. \end{aligned}$$

The following theorem is evident.

Theorem 2.2 ([15, 24]). *For any given $\alpha\in\mathbb{R}$, $v(D_S^n)\geq\alpha$ if and only if $v(D(\alpha))\leq 0$.*

From the above theorem, we imply that the optimal surrogate dual value $v(D_S^n)$ is the maximum $\alpha\in\mathbb{R}$ such that $v(D(\alpha))\leq 0$. This motivates the use of a cutting plane method to solve $(D(\alpha))$. Notice that $(D(\alpha))$ is equivalent to the following linear programming:

$$\begin{aligned} &v(D(\alpha)) = \max_{(\beta,\mu)} \ \beta \\ &\text{s.t.} \ \ \beta\leq\mu^T(g(x)-b), \ \forall x\in X(\alpha), \\ &\qquad \mu\in\Lambda. \end{aligned}$$

In general, there is a huge number of integer points in set $X(\alpha)$. Thus, it is impossible to solve the above linear programming directly in practice. Nevertheless, we can replace $X(\alpha)$ by a much smaller set T^k and update and enlarge T^k iteratively in the course of the solution process for solving $D(\alpha)$. At the k-th iteration, consider the following linear programming:

$$(LP_k) \qquad \begin{aligned} &\max_{(\beta,\mu)} \ \beta \\ &\text{s.t.} \ \ \beta\leq\mu^T(g(x)-b), \ \forall x\in T^k\subset X(\alpha), \\ &\qquad \mu\in\Lambda. \end{aligned}$$

The cutting plane method for (D_S^n) can then be described as follows.

Procedure 2.3 (Cutting Plane Procedure for (D_S^n)).

Step 0 (Initialization). Set $\alpha^0=+\infty$, $T^0=\emptyset$. Choose any $\mu^1\in\Lambda$. Set $k=1$.

Step 1 (Surrogate relaxation) Solve the surrogate relaxation problem (S_{μ^k}) and obtain an optimal solution x^k. If $g(x^k)\leq b$, stop and x^k is an optimal solution to (P) and $v(D_S^n)=v(P)$.

Step 2 (Updating upper bound). If $f(x^k)<\alpha^{k-1}$, then set $\alpha^k=f(x^k)$. Otherwise, set $\alpha^k=\alpha^{k-1}$.

Step 3 (Updating multiplier). Set $T^k=T^{k-1}\cup\{x^k\}$. Solve the linear program (LP_k) and obtain an optimal solution (β^k,μ^k). If $\beta^k\leq 0$, stop and $\alpha^k=v(D_S^n)$. Otherwise, set $\mu^{k+1}=\mu^k$ and $k:=k+1$, goto Step 1.

Similar to the linear case, it can be shown that Procedure 2.3 finds an optimal value of (D_S^n) within a finite number of iterations (see Theorem 4.4, [15]). In our implementation of the procedure, the initial surrogate multiplier vector is taken as $\mu^1 = (1/m, 1/m, \cdots, 1/m)^T$. Since a new point x^k is added to T^k at each iteration, the linear programming (LP_{k+1}) is formed by adding a new constraint $\beta \le \mu^T(g(x^k) - b)$. Thus, the dual simplex method is suitable to solve (LP_k), as we did in our implementation of Procedure 2.3 in the numerical experiments.

Since the surrogate relaxation problem (S_{μ^k}) has to be solved many times during the surrogate dual search, the efficiency of Algorithm 3.1 depends on how fast the singly constrained problem (S_{μ^k}) can be solved. In the following, we use the linearization scheme proposed in [10] to convert the surrogated problem (S_{μ^k}) into a 0-1 linear integer programming problem.

Notice that (S_μ) can be written as

$$\begin{aligned}(S_\mu) \qquad & \max\ f(x) = \sum_{j=1}^n f_j(x_j) \\ & \text{s.t. } \sum_{j=1}^n h_j(x_j, \mu) \le B_\mu, \\ & \qquad x_j = 0, 1, \ldots, u_j,\ j = 1, \ldots, n,\end{aligned}$$

where $h_j(x_j, \mu) = \sum_{i=1}^m \mu_i g_{ij}(x_j)$ and $B_\mu = \sum_{i=1}^m \mu_i b_i$. Let $x_j = \sum_{i=1}^{u_j} y_{ij}$, where $y_{ij} \in \{0, 1\}$. Define

$$c_{ij} = f_j(i) - f_j(i-1), \qquad i = 1, \ldots, u_j, \tag{2.2}$$

$$d_{ij}(\mu) = h_j(i, \mu) - h_j(i-1, \mu), \qquad i = 1, \ldots, u_j,\ j = 1, \ldots, n. \tag{2.3}$$

Assume that $f_j(x_j)$ is concave on $[0, u_j]$ and $g_{ij}(x_j)$ is convex on $[0, u_j]$ for each i, $j = 1, \ldots, n$. By assumption of (P), f_j and g_{ij} are nondecreasing functions on $[0, u_j]$ for each j. Thus,

$$\begin{aligned}& c_{1j} \ge c_{2j} \ge \cdots \ge c_{u_j,j}, \qquad j = 1, \ldots, n, \\ & d_{1j}(\mu) \le d_{2j}(\mu) \le \cdots \le d_{u_j,j}(\mu), \qquad j = 1, \ldots, n.\end{aligned}$$

Consider the following 0-1 linear knapsack problem:

$$\begin{aligned}(0\text{-}1LP) \qquad & \max\ \psi(y) = \sum_{j=1}^n \sum_{i=1}^{u_j} c_{ij} y_{ij} \\ & \text{s.t. } \phi(y, \mu) = \sum_{j=1}^n \sum_{i=1}^{u_j} d_{ij}(\mu) y_{ij} \le B_\mu, \\ & \qquad y \in Y = \{y_{ij} \in \{0, 1\} \mid i = 1, \ldots, u_j,\ j = 1, \ldots, n\}.\end{aligned}$$

Theorem 2.4. *Assume that $f_j(x_j)$ is concave on $[0, u_j]$ and $g_{ij}(x_j)$ is convex on $[0, u_j]$ for each i, $j = 1, \ldots, n$. Problems (S_μ) and (0-1LP) are equivalent under the transformation $x_j = \sum_{i=1}^{u_j} y_{ij}$.*

Proof. Notice that under transformation $x_j = \sum_{i=1}^{u_j} y_{ij}$, $\psi(y)$ and $\phi(y, \mu)$ take the same value as $f(x)$ and $h(x, \mu)$ on X and Y, respectively, if for each j, there is no 1 after $0's$ in the 0-1 sequence $\{y_{1j}, y_{2j}, \ldots, y_{u_j,j}\}$. By (2.2) and (2.3), for the optimal solution $y^* = \{y_{ij}^*\}$

of (0-1LP), taking $y^*_{kj} = 1$ results in more "profit" (c_{kj}) and less resource (d_{kj}) than does by taking $y^*_{k+1,j} = 1$. Thus, there must be no 1 after 0's in the sequence $\{y^*_{1j}, \ldots, y^*_{u_j j}\}$ for $j = 1, \ldots, n$. Therefore, (S_μ) and (0-1LP) are equivalent. □

Various implicit enumerative methods have been proposed for solving the 0-1 linear knapsack problem, for example, branch-and-bound method [19] and dynamic programming [10].

3 The Main Algorithm

In this section, we develop an exact algorithm for (P). This algorithm is based on the cutting plane method for the surrogate dual search discussed in the previous section.

Let $\gamma, \delta \in \mathbb{R}^n$ be integer vectors. Denote by $\langle \gamma, \delta \rangle$ the set of integer points in $[\gamma, \delta]$. We will call $\langle \gamma, \delta \rangle$ an integer subbox. Denote by $(P(\tilde{X}))$ the subproblem of (P) with X replaced by an integer subbox $\tilde{X} \subseteq X$. For each subproblem $(P(\tilde{X}))$ of (P), applying Procedure 2.3 to its surrogate dual problem gives rise to an optimal multiplier vector μ^* and an upper bound $\alpha = v(D^n_S)$. An optimal solution x^* to the surrogate relaxation problem $(S_{\mu^*}(\tilde{X}))$ can be also computed. Suppose we have an incumbent feasible solution x_{opt} to (P). If x^* is feasible to (P), then, by the monotonicity of f and g_i's, we can cut the integer box $\langle \gamma, x^* \rangle$ from $\langle \gamma, \delta \rangle$ and update the incumbent x_{opt} if $f(x^*) > f(x_{opt})$. Otherwise, we can cut the integer box $\langle x^*, \delta \rangle$ from $\langle \gamma, \delta \rangle$ without missing any feasible solution.

The main algorithm can be described as follows.

Algorithm 3.1.

Step 0 (Initialization). If $x = 0$ is infeasible, then the problem has no feasible solution, or if $x = u$ is feasible, then u is the optimal solution, stop. Applying Procedure 2.3 to (P), we obtain an optimal multiplier vector μ^0, upper bound $\alpha(X)$, and an optimal solution to the surrogate relaxation problem $(S_{\mu^0}(X))$. Set $x_{opt} = 0$, $f_{opt} = f(0)$, $\Omega^1 = \{X\}$, $k = 1$.

Step 1 (Sub-Domain Selection). Select an integer subbox $X^k = \langle \gamma^k, \delta^k \rangle$ from Ω^k with the maximum upper bound:

$$X^k \in \arg \max_{\tilde{X} \in \Omega^k} \alpha(\tilde{X}).$$

where $\alpha(\tilde{X})$ is the upper bound on $\tilde{X}$ produced by Procedure 2.3.

Step 2 (Partition). Let x^k be the optimal solution to $(S_{\mu^k}(X^k))$, where μ^k is the optimal surrogate multiplier vector to $(P(X^k))$.

(i) If x^k is a feasible solution to (P), then cut $\langle \gamma^k, x^k \rangle$ from $\langle \gamma^k, \delta^k \rangle$ and partition $\langle \gamma^k, \delta^k \rangle \setminus \langle \gamma^k, x^k \rangle$ into a union of integer subboxes. Update x_{opt} and f_{opt} if $f(x^k) > f(x_{opt})$.

(ii) If x^k is an infeasible solution to (P), then cut $\langle x^k, \delta^k \rangle$ from $\langle \gamma^k, \delta^k \rangle$ and partition $\langle \gamma^k, \delta^k \rangle \setminus \langle x^k, \delta^k \rangle$ into a union of integer subboxes.

Let Y^k denote the set of integer subboxes generated by the above cut-and-partition process. Remove X^k from Ω^k.

Step 3 (Dual Search and Fathoming). For each new integer subbox $\tilde{X} = \langle \xi, \eta \rangle \in Y^k$, do the following:

(i) Apply Procedure 2.3 to subproblem $(P(\tilde{X}))$ to generate an optimal surrogate multiplier vector $\tilde{\mu}$, a surrogate dual bound $\alpha(\tilde{X})$, and an optimal solution to $(S_{\tilde{\mu}}(\tilde{X}))$.

(ii) Update x_{opt} and f_{opt} if a better feasible solution to (P) is found.

(iii) Remove $\tilde{X}$ from Y^k if $\alpha(\tilde{X}) \leq f_{opt}$.

Set $\Omega^{k+1} = \Omega^k \cup Y^k$.

Step 4 (Termination). If Ω^{k+1} is empty, stop, and x_{opt} is an optimal solution. Otherwise, set $k := k + 1$. Go to Step 1.

In Step 2, the partition schemes proposed in [15, 27] can be used to decompose the complement sets $\langle \gamma^k, \delta^k \rangle \setminus \langle \gamma^k, x^k \rangle$ or $\langle \gamma^k, \delta^k \rangle \setminus \langle x^k, \delta^k \rangle$ into a union of integer subboxes. Since no feasible solution better than the incumbent is removed during the cutting process in Step 2, the incumbent solution must be an optimal solution to (P) when the algorithm terminates. The finite termination of the algorithm is obvious by noting that the finiteness of X and the fact that at least one nonempty integer subbox is cut from X at each iteration.

$\boxed{4}$ Computational Results

In this section, we report some computational results of Algorithm 3.1. Three classes of nonlinear knapsack problems with multiple constraints are tested for the proposed algorithm.

As we need to solve the singly constrained surrogated problem (S_{μ^k}) during the surrogate dual search, we select concave objective functions for testing the algorithm so that the surrogated problem can be solved efficiently by using 0-1 linearization technique. In our testing, the objective functions of the tested problems are selected as follows:

- Quadratic knapsack problems (QP): $f(x) = \sum_{j=1}^{n}(c_j x_j - d_j x_j^2)$, where $c_j \in [100, 300]$, $d_j \in (0, 10]$.
- Constrained redundancy problems in reliability systems $(RELI)$: $f(x) = \prod_{j=1}^{n}[1-(1-r_j)^{x_j}]$, where $r_j \in (0, 1)$.
- Optimal sample allocation in stratified sampling $(SAMP)$: $f(x) = -\sum_{j=1}^{n} d_j/x_j$, $d_j \in [1, 20]$ for $j = 1, \ldots, n$.

The constraints are linear functions: $g_i(x) = \sum_{j=1}^{n} a_{ij}x_j$, $i = 1, \ldots, m$, where $a_{ij} \in [1, 50]$ for $i = 1, \ldots, m$, $j = 1, \ldots, n$.

All the coefficients in the test problems are randomly generated from a uniform distribution. The right-hand side b is determined last to ensure the feasibility of the test problems. In our implementation, we take $b_i = 0.7 \times \sum_{j=1}^{n} a_{ij}u_j$, where $u_j = 5$, $j = 1, \ldots, n$.

The algorithm was programmed by Fortran 90 and ran on a Pentium IV PC (2GHz and 256Mb RAM). To compare the surrogate dual with the Lagrangian dual, we also programmed a branch-and-bound method using the subgradient Lagrangian dual search in computing the upper bounds (see [16]). The computational results are reported in Tables 1-3, where

- $n \times m$ is the size of the problem;

- T_{CPU} denotes the average CPU time for 5 test problems;
- N_{ISB} denotes the average number of integer subboxes generated in the solution process (after rounding off) for 5 test problems;
- "SUB" stands for the method using subgradient Lagrangian dual search.
- "NS" denotes the situation where 5 test problems were not solved in 10 CPU hours.

From Tables 1-3, we can conclude that Algorithm 1 can solve medium-scale multi-dimensional nonlinear knapsack problems within reasonable computation time. Comparing Tables 1-3, we can also conclude that the number of subproblems solved in Algorithm 1 is significant less than the number solved by the algorithm using the subgradient Lagrangian dual search. This is due to tighter upper bounds obtained by the surrogate dual search. Tables 1-3 indicate that the surrogate dual search outperforms the subgradient Lagrangian dual search both in terms of the number of subproblems solved and the CPU time consumed. We point out that the most time-consuming part of the method is to solve the surrogate relaxation using the 0-1 linearization procedure. Our numerical results reveal that the surrogate dual procedure is a promising method in solving multi-dimensional nonlinear knapsack problems.

Table 1: Numerical results for (QP)

$n \times m$	T_{CPU}		N_{ISB}	
	Algorithm 1	SUB	Algorithm 1	SUB
30×5	24.5	67.9	559	5624
30×10	116.9	871.2	3150	54419
30×20	441.6	NS	7941	NS
40×5	75.2	644.5	2600	42286
40×10	654.8	NS	11955	NS
50×5	357.0	NS	8593	NS

Table 2: Numerical results for $(RELI)$

$n \times m$	T_{CPU}		N_{ISB}	
	Algorithm 1	SUB	Algorithm 1	SUB
80×5	23.4	34.9	1054	9814
80×10	48.5	110.2	959	12910
80×20	127.8	1566.3	2570	117291
100×5	242.5	944.8	5567	88273
100×10	458.9	NS	5787	NS
100×20	556.7	NS	6567	NS

References

[1] K.M. Bretthauer, A. Ross and B. Shetty, Nonlinear integer programming for optimal allocation in stratified sampling, *European J. Oper. Res.* 116 (1999) 667–680.

Table 3: Numerical results for $(SAMP)$

$n \times m$	T_{CPU}		N_{ISB}	
	Algorithm 1	SUB	Algorithm 1	SUB
30×3	4.2	5.4	759	6710
30×5	10.8	28.9	1035	12800
30×10	55.9	220.9	2522	20331
40×3	10.2	30.7	1195	9147
40×5	81.0	249.2	4618	50571
40×10	977.4	NS	24218	NS

[2] K.M. Bretthauer and B. Shetty, The nonlinear resource allocation problem, *Oper. Res.* 43 (1995) 670–683.

[3] K.M. Bretthauer and B. Shetty, The nonlinear knapsack problem–Algorithms and applications, *European J. Oper. Res.* 138 (2002) 459–472.

[4] M. Djerdjour, K. Mathur and H. M. Salkin, A surrogate relaxation based algorithm for a general quadratic multi-dimensional knapsack problem, *Oper. Res. Lett.* 7 (1988) 253–258.

[5] M.E. Dyer, Calculating surrogate constraints, *Math. Program.* 19 (1980) 255–278.

[6] A. Fréville and S. Hanafi, The multidimensional 0-1 knapsack problem–Bounds and computational aspects, *Ann. Oper. Res.* 139 (2005) 195–227.

[7] A. Fréville and G. Plateau, An exact search for the solution of the surrogate dual of the 0-1 bidimensional knapsack problem, *European J. Oper. Res.* 68 (1993) 413–421.

[8] B. Gavish and H. Pirkul, Efficient algorithms for solving multiconstraint zero-one knapsacks to optimality, *Math. Program.* 31 (1985) 78–105.

[9] F. Glover, A multiphase-dual algorithm for the zero-one integer programming problem, *Oper. Res.* 13 (1965) 879–919.

[10] D.S. Hochbaum, A nonlinear knapsack problem, *Oper. Res. Lett.* 17 (1995) 103–110.

[11] M.H. Karwan and R.L. Rardin, Searchability of the composite and multiple surrogate dual functions, *Oper. Res.* 28 (1980) 1251–1257.

[12] H. Kellerer, U. Pferschy and D.D. Pisinger, *Knapsack Problems*, Springer-Verlag, Berlin, 2004.

[13] S.L. Kim and S. Kim, Exact algorithm for the surrogate dual of an integer programming problem: Subgradient method approach., *J. Optim. Theory Appl.* 96 (1998) 363–375.

[14] F. Korner, A hybrid method for solving nonlinear knapsack problems, *European J. Oper. Res.* 38 (1989) 238–427.

[15] D. Li and X.L. Sun, *Nonlinear Integer Programming*, Springer, New York, 2006.

[16] D. Li, X.L. Sun, J. Wang and K.I.M. McKinnon, Convergent Lagrangian and domain cut method for nonlinear knapsack problems, *Comput. Optim. Appl.* (2007) to appear.

[17] H. Luss and S.K. Gupta, Allocation of effort resources among competing activities, *Oper. Res.* 23 (1975) 360–366.

[18] R.E. Marsten and T.L. Morin, A hybrid approach to discrete mathematical programming, *Math. Program.* 14 (1978) 21–40.

[19] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*, John Wiley & Sons, New York, 1990.

[20] S. Martello and P. Toth, An exact algorithm for the two-constraint 0-1 knapsack problem, *Oper. Res.* 51 (2003) 826–835.

[21] K. Mathur, H.M. Salkin and S. Morito, A branch and search algorithm for a class of nonlinear knapsack problems, *Oper. Res. Letters* 2 (1983) 155–160.

[22] M.A. Osorio and F. Glover, Exploiting surrogate constraint analysis for fixing variables in both bounds for multidimensional knapsack problems, in *Proceedings of the Fourth Mexican International Conference on Computer Science, ENC 2003*, 2003, pp. 263–267.

[23] M.A. Osorio, F. Glover and P. Hammer, Cutting and surrogate constraint analysis for improved multidimensional knapsack solutions, *Ann. Oper. Res.* 117 (2002) 71–93.

[24] R.G. Parker and R.L. Rardin, *Discrete Optimization*, Academic Press, Boston, 1988.

[25] S. Sarin, M.H. Karwan and R.L. Rardin, A new surrogate dual multiplier search procedure, *Naval Res. Logist.* 34 (1987) 431–450.

[26] X.L. Sun and D. Li, Optimality condition and branch and bound algorithm for constrained redundancy optimization in series systems, *Optim. Eng.* 3 (2002) 53–65.

[27] X.L. Sun, F.L. Wang and D. Li, Exact algorithm for concave knapsack problems: Linear underestimation and partition method, *J. Global Optim.* 33 (2005) 15–30.

[28] B. Thiongane, A. Nagih and G. Plateau, Lagrangean heuristics combined with reoptimization for the 0-1 bidimensional knapsack problem, *Discrete Appl. Math.* 154 (2006) 2200–2211.

[29] F.A. Tillman, C.L. Hwuang and W.Kuo, *Optimization of System Reliability*, Marcel Dekker, 1980.

[30] H. Ziegler, Solving certain singly constrained convex optimization problems in production planning, *Oper. Res. Letters* 1 (1982) 246–252.

Manuscript received 27 June 2007
revised 19 September 2007
accepted for publication 20 September 2007

Xiaoling Sun
Department of Management Science, School of Management
Fudan University, Shanghai 200433, P. R. China
E-mail address: xls@fudan.edu.cn

Shanshan Kong
Department of Computer Science, Jining College
Jining, Shandong, P.R. China
E-mail address: `kongshanshan@126.com`

Duan Li
Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong, Shatin, N. T., Hong Kong
E-mail address: `dli@se.cuhk.edu.hk`