Approximation algorithms for the m-dimensional 0–1 knapsack problem: Worst-case and probabilistic analyses

A.M. FRIEZE and M.R.B. CLARKE
Department of Computer Science & Statistics, Queen Mary College, University of London, London E1 4NS, United Kingdom

Received August 1981
Revised March 1982

We describe a polynomial approximation scheme for an m-constraint 0–1 integer programming problem (m fixed) based on the use of the dual simplex algorithm for linear programming.
We also analyse the asymptotic properties of a particular random model.

1. Introduction

This paper is concerned with the m-dimensional 0–1 knapsack problem:

$$\text{Maximise} \quad z=\sum_{j=1}^{n} c_j x_j, \tag{1.1a}$$

$$\text{Subject to} \quad \sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \quad i=1,\dots,m, \tag{1.1b}$$

$$0 \leqslant x_j \leqslant 1, \quad j=1,\dots,n, \tag{1.1c}$$

$$x_j \text{ integer}, \quad j=1,\dots,n \tag{1.1d}$$

where for all i, j, $a_{ij} \geqslant 0$, b_i, $c_j > 0$. m is considered to be a *fixed constant.* The problem is known to be NP-hard even when $m=1$ (see Garey and Johnson [5]) and so there is interest in finding heuristics which work in polynomial time (polynomial in the input length) and have a guaranteed accuracy.

Given $\epsilon > 0$ we say that an algorithm is an *ϵ-approximation algorithm* if it always produces a solution $\hat{x}$ with objective value $\hat{z}$ that satisfies

$$z^* - \hat{z} \leqslant \epsilon z^* \tag{1.2}$$

where z^* is the maximum value of z in (1.1).

The aim of this paper is two-fold. In Section 2 we present a *polynomial approximation scheme* for problem (1.1), i.e. for each $\epsilon > 0$ we describe an algorithm A_ϵ which is an ϵ-approximation algorithm and runs in time polynomial in the length of the input description of (1.1).

In Section 3 we conduct a probabilistic analysis of this problem and obtain reasonably tight bounds for the asymptotic behaviour of the objective value z^*. A simple consequence of this analysis is that with probability $\to 1$ as $n \to \infty$ the solution obtained by rounding down the solution to the linear program (1.1a)–(1.1c) satisfies (1.2) with $\epsilon = n^{-a}$ provided $a < 1/(m+1)$.

North-Holland
European Journal of Operational Research 15 (1984) 100–109

Previous work

As regards polynomial approximation schemes the case $m=1$ has been studied most extensively. Sahni [12] gave the first polynomial approximation scheme generalising some work in Johnson [7]. The paper by Ibarra and Kim [6] gave a *fully* polynomial approximation scheme which was improved by Lawler [10].

The case $m \geqslant 1$ but with (1.1c) replaced by $x_j \geqslant 0$ for $j=1,\dots,n$ was analysed by Chandra, Hirschberg and Wong [2]. They could not extend their approach to (1.1) because they could not solve the linear program (1.1a)–(1.1c) in polynomial time.

With the advent of Khachian's algorithm [8] it becomes straightforward to extend their method to this case. Oguz and Magazine [11] do precisely this.

In Section 2 we give an alternative polynomial approximation scheme.

2. An approximation scheme

Let $\epsilon > 0$ be given. We shall describe an ϵ-approximation algorithm A_ϵ that runs in polynomial time assuming ϵ and m are fixed. It is not fully polynomial, i.e. polynomial in $1/\epsilon$ as well, and indeed Korte and Schrader [9] have shown that no fully polynomial approximation scheme exists for this problem unless P = NP.

Next let $k=\min(n, \lceil m(1-\epsilon)/\epsilon \rceil)$, $N=\{1,2,\dots,n\}$ and for $S \subseteq N$ let $T(S)=\{t \in S\colon c_t > \min(c_j\colon j \in S)\}$.

Let LP(S) denote the linear program obtained from (1.1) by replacing (1.1d) by

$$x_j = \begin{cases} 1, & j \in S, \\ 0, & j \in T(S). \end{cases}$$

Let $x^{\mathrm{B}}(S)$ denote an optimal *basic feasible* solution to LP(S).

A_ϵ proceeds by solving LP(S) for all $S \subseteq N$, $|S| \leqslant k$ and then rounding down $x^{\mathrm{B}}(S)$. A_ϵ outputs the best solution to (1.1) that is found in this way. Because m is fixed, the 'difference' between $x^{\mathrm{B}}(S)$ and $\lfloor x^B(S) \rfloor$ is not 'too large'.

We can now formally describe the proposed algorithm A_ϵ.

begin
  $\hat{z} := 0$
  for $S \subseteq N$ with $|S| \leqslant k$ **do**
    begin
    let $b_i(S) = b_i - \Sigma_{j \in S} a_{ij}$ for $i=1,\dots,m$;
    if $b_i(S) \geqslant 0$ for $i=1,\dots,m$
      then do $\{b_i(S) < 0 \rightarrow$ LP(S) is infeasible$\}$
      begin
      construct a solution $x^{\mathrm{B}}(S)$ to LP(S);
      round $x^{\mathrm{B}}(S)$ to an integer solution $x^{\mathrm{I}}(S)$, i.e.
      $x_j^{\mathrm{I}}(S) = \lfloor x_j^B(S) \rfloor$ for $j=1,\dots,n$;
      let $z^{\mathrm{I}}(S) = \Sigma_{j=1}^n c_j x_j^{\mathrm{I}}(S)$;
      if $\hat{z} < z^{\mathrm{I}}(S)$ **then** $\hat{z} := \hat{z}^{\mathrm{I}}(S)$; $\hat{x} := x^{\mathrm{I}}(S)$
      end
    end
end

We first show

Theorem 2.1. *$\hat{x}$ is an ϵ-approximation.*

Proof. Let x^* be an optimum solution to (1.1) and let $S^* = \{j: x_j^* = 1\}$.

If $|S^*| \leq k$ then $\hat{z} \geq z^1(S^*) \geq z^*$ implies $\hat{z} = z^*$.

Otherwise let $S^* = \{i_1, \ldots, i_r\}$ where $c_{i_1} \geq \cdots \geq c_{i_r}$. Let $S_k^* = \{i_1, \ldots, i_k\}$ and $\sigma = \sum_{t=1}^{k} c_{i_t}$. It follows that

$$j \in N - (S_k^* \cup T(S_k^*)) \text{ implies } c_j \leq \sigma/k. \tag{2.1}$$

Now

$$z^* \leq \sum_{j=1}^{n} c_j x_j^{\mathrm{B}}(S_k^*) \leq \sum_{j=1}^{n} c_j x_j^{\mathrm{I}}(S_k^*) + \delta$$

where

$$\delta = \sum_{j \in D} c_j \quad \text{and} \quad D = \{j \in N: x_j^{\mathrm{B}}(S_k^*) > x_j^{\mathrm{I}}(S_k^*)\}.$$

Now $j \in D$ implies x_j is a basic variable in $x^{\mathrm{B}}(S)$. Thus $|D| \leq m$. Also $D \cap (S_k^* \cup T(S_k^*)) = \varnothing$ and so $j \in D$ implies $c_j \leq \sigma/k$ by (2.1). Thus $\delta \leq m\sigma/k$ and

$$z^* \leq \hat{z} + m\sigma/k \leq \hat{z} + m\hat{z}/k \quad \text{as } \hat{z} \geq \sigma$$

from which (1.2) follows easily.

We now have to show that A_ε runs in polynomial time, i.e. the maximum running time must be shown to be bounded by a polynomial in the 'length of space' needed to describe the problem [5].

Let

$$L = \sum_{i=1}^{m} \sum_{j=1}^{n} l(a_{ij}) + \sum_{i=1}^{m} l(b_i) + \sum_{j=1}^{n} l(c_j)$$

where $l(\xi) = [\log_2(\xi + 1)]$. Then L is a measure of the length of space needed to describe the problem, assuming that the coefficients a_{ij}, c_j, b_i are all integers.

The main question that needs to be resolved concerns how LP(S) should be solved.

In the following description we shall for simplicity assume $S \neq \varnothing$.

If Khachian's algorithm (as described by Gács and Lovász [4]) is used the largest component in the worst case execution time is $O(n^4 L)$ multiplications of real numbers stored to $O(nL)$ precision. Each multiplication takes $O(nL \log nL \operatorname{loglog} nL)$ time (see Aho, Hopcroft and Ullman [1]) and so the time complexity of the whole approximation algorithm will be

$$O(n^{k+5} L^2 \log nL \operatorname{loglog} nL) \tag{2.2}$$

as there are $O(n^k)$ sets S with $|S| \leq k$. (Remember that m is considered to be a constant here).

Alternatively if the lexicographic dual simplex algorithm is used as described below, the solution of LP(ϕ) requires $O(n^{m+1})$ multiplications and divisions on integers with no more than $O(L)$ bits. The overall algorithm will now have time complexity

$$O(n^{m+k+1} L \log L \operatorname{loglog} L). \tag{2.3}$$

It is apparent that (2.3) is smaller than (2.2) for $m \leq 4$, and for $m \geq 5$ their relative sizes depend on the relative size of n and L. Note that L must be considered to grow without bound otherwise problem (1.1) is solvable in polynomial time using dynamic programming.

Dual simplex algorithm

We can use the dual simplex algorithm as there are no more than $\binom{n+m}{m} = O(n^m)$ dual feasible bases to (1.1) and this gives a satisfactory upper bound to the number of iterations needed. Note that we cannot use the (primal) simplex algorithm as there could be $O(2^n \binom{n+m}{m})$ feasible bases to (1.1). The nicest way to describe the dual algorithm is in conjunction with the Tucker–Beale simplex tableau—see for example Simmonard [13, pp. 88–91].

Let $N^+ = \{1,\ldots,m+n\}$. For each $I \subseteq N^+$ such that $|I| = m$ and such that the columns of coefficients of variables x_i, $i \in I$, are linearly independent ($x_{m+1},\ldots,x_{m+n}$ are slack variables) there is exactly one *tableau*

$$x = p_0 + \sum_{j \notin I} p_j(-x_j). \tag{2.4}$$

Here $x = (x_0, x_1,\ldots,x_{m+n})$ where $x_0 = z$ and (2.4) is the *unique* way of expressing x_i, $i = 0, 1,\ldots,m+n$ in terms of x_j for $j \in J = N^+ - I$. (Note that for $i \in J$, (2.4) says $x_i = x_i$).

Given I such that (2.4) exists we assign values $v = (v_0, v_1,\ldots,v_{m+n})$ to the variables by (see also (2.6))

$$v_i = \begin{cases} 0 & \text{for } i \in J \text{ and } p_i \succ 0, \\ 1 & \text{for } i \in J \text{ and } p_i \prec 0, \\ p_{i0} - \sum_{j \in J} p_{ij} v_j & \text{for } i \in I. \end{cases} \tag{2.5}$$

Here $\succ 0$ means lexicographically positive—i.e. the first non-zero component is positive.

Note that for any $i \in J$ such that $i > n$, $p_i \prec 0$ is not assigned a value by (2.5). This is because throughout the algorithm we maintain

$$i \in J \text{ and } i > n \to p_i \succ 0. \tag{2.6}$$

In this way we ensure that v is a dual feasible basic solution to (1.1).

The set I used to start the algorithm is $\{n+1,\ldots,n+m\}$ which clearly satisfies (2.6).

By examining row 0 of (2.4) it is apparent that for any I satisfying (2.6) that v_0 is an upper bound to the maximum objective value of LP(ϕ). Thus if v is feasible as well then v is optimal. If v is infeasible then there exist $l \in I$ such that

$$v_l < 0 \tag{2.7a}$$

or

$$v_l > 1 \quad \text{and} \quad l \leqslant n. \tag{2.7b}$$

If (2.7a) holds we choose k by

$$p_k/p_{lk} = \text{lex.max}(p_j/p_{lj}: p_{lj} \neq 0 \text{ and } p_j/p_{lj} \prec 0). \tag{2.8a}$$

Note that if (2.7a) holds the index k is well defined otherwise one can deduce that LP(ϕ) is infeasible which it clearly isn't. In general the calculation of $b(S)$ in A_ϵ determines whether or not LP(S) is feasible.

If (2.7a) does not hold and (2.7b) holds we choose k by

$$p_k/p_{lk} = \text{lex.min}(p_j/p_{lj}\, p_{lj} \neq 0 \text{ and } p_j/p_{lj} \succ 0). \tag{2.8b}$$

Again the index k will be well defined.

Having found k we pivot, i.e. replace I by $(I \cup \{k\}) - \{l\}$. The effect on the tableau is summarised by ($'$ denotes an updated value):

$$p'_l = -p_k/p_{lk} \quad (p'_l \text{ and not } p'_k \text{ because of the indexing convention}), \tag{2.9a}$$

$$p'_j = p_j - (p_{lj}/p_{lk})p_j, \quad j \in (J - \{k\}) \cup \{0\}, \tag{2.9b}$$

$$v' = v - (\delta/p_{lk})p_k \tag{2.9c}$$

where

$$\delta = \begin{cases} v_l & \text{in case (2.7a)}, \\ v_l - 1 & \text{in case (2.7b)}. \end{cases}$$

(2.9a), (2.9b) are standard pivot formulae and (2.9c) indicates that in case (2.7a) we 'drive' v_l up to zero and in case (2.7b) we 'drive' v_l down to 1. (See (ii) below.)

The following properties can be observed:

(i) $j \in J - \{k\}$, $\mathbf{v}'_j = \mathbf{v}_j$ or equivalently $\mathbf{p}'_j \succ 0 \leftrightarrow \mathbf{p}_j \succ 0$;

(ii) in case (2.8a) $\mathbf{p}'_l \succ 0$ and so (2.6) is maintained throughout. In case (2.8b) $\mathbf{p}'_l \prec 0$;

(iii) $\mathbf{v}' \prec \mathbf{v}$.

From (iii) and the fact that a unique $\mathbf{v}$ is associated with each I giving rise to a tableau we deduce that the algorithm does not cycle and that after $O(n^m)$ iterations a feasible and hence optimal solution to LP(S) can be found.

Assuming the coefficients p_{ij} are stored as rational numbers in which numerator and denominator are relatively prime the sizes of these integers in absolute value are bounded by subdeterminants of the original matrix and hence require $O(L)$ bits. As each pivot operation requires $O(n)$ arithmetic operations—m is constant—we have expression (2.3).

Oguz and Magazine [11] have shown that for any k there exist problems for which (1.2) can be made arbitrarily close to equality.

We can also apply these ideas to covering problems of the form

$$\begin{aligned} &\text{minimise} && \sum_{j=1}^{n} c_j x_j, \\ &\text{subject to} && \sum_{j=1}^{n} a_{ij} x_j \geqslant b_i, \quad i = 1,\dots,m, \\ &&& x_j = 0 \text{ or } 1, \qquad j = 1,\dots,n \end{aligned}$$

simply by rounding up the solutions to LP(S). (It is again assumed that all coefficients are non-negative.)

3. Random problems

In this section we study the quality of the solution to (1.1) obtained by simply rounding down the solution to LP(ϕ). We analyse this from a probabilistic viewpoint and show that for a simple model its asymptotic properties are extremely good.

We shall assume now that $b_i = 1$ for $i = 1,\dots,m$ and that all other coefficients are uniformly and independently distributed in the interval [0,1].

We take this as a model because of its simplicity and because any problem can be reduced by scaling to one with coefficients between 0 and 1.

In these circumstances let Z_{mn} be the random variate whose value is the maximum objective value to (1.1). By considering upper and lower bounds to Z_{mn} we will be able to get a good idea of its asymptotic nature.

The main tool is the following simple lemma:

Lemma 3.1. *Let $U_1,\dots,U_n$ each be independently distributed as the maximum of p independent uniform random variates in* [0,1], *i.e.* $\Pr(U_j \leqslant a) = a^p$ *for* $0 \leqslant a \leqslant 1$. *Similarly let $V_1,\dots,V_n$ be independently distributed as the maximum of q uniform* [0,1] *random variates. Let $R_j = V_j/U_j$ for $j = 1,\dots,n$. (We ignore the zero probability events $U_j = 0$) and let $R_{(k)}$ denote the value of the kth largest of $R_1,\dots,R_n$.*

Then for integers a, $b \geqslant 0$ and $1 \leqslant k < l \leqslant n$ we have

$$\exp\left(U_{(k)}^a U_{(l)}^b\right) = \left(\sum_{r=0}^{n} \binom{n}{r} q^r p^{n-r} \Theta_r \right) \Big/ \left((p+q)^{n-2} (p+q+a)(p+q+b) \right)$$

where

$$\Theta_r = \begin{cases} 1, & r = 0, 1, \dots, k-1, \\ \prod_{t=k}^{r} (tp/(a+tp)), & r = k, \dots, l-1, \\ \Theta_{l-1} \prod_{t=l}^{r} (rp/(a+b+tp)), & r = l, \dots, n. \end{cases}$$

The proof of this lemma is a long but straightforward exercise in elementary integration and is omitted.

A lower bound for Z_{mn}

Let $w_j = \max(a_{ij}: i = 1,\dots,m)$ for $j = 1,\dots,n$ and let

$$X_{mn} = \max \sum_{j=1}^{n} c_j x_j, \tag{3.1a}$$

$$\text{subject to } \sum_{j=1}^{n} w_i x_j \leq 1, \tag{3.1b}$$

$$\text{(1.1c) and (1.1d).} \tag{3.1c}$$

Clearly $X_{mn} \leq Z_{mn}$ as any x satisfying (3.1) will also satisfy (1.1).

Notation

A proposition dependent on the integer n is said to hold *almost surely* (a.s.) if the proposition holds with probability tending to 1 as $n \to \infty$.

For infinite sequences of non-negative terms u_n, v_n we write $u_n \sim v_n$ if

$$(1 - o(n))v_n \leq u_n \leq (1 + o(n))v_n \tag{3.2}$$

where $o(n)$ denotes as usual some function that tends to zero as $n \to \infty$.

We shall also write $u_n \sim v_n$ a.s. if (3.2) holds a.s.

Problem (3.1) is a one constraint linear program and so X_{mn} is easy to determine: Using Lemma 3.1 we obtain estimates of means and variances useful in computing X_{mn}. Fortunately these variances are small enough so that the Chebycheff inequality can be used to prove

Lemma 3.2.

$$X_{mn} \sim \alpha_m n^{1/(m+1)} \quad a.s. \tag{3.3}$$

where $\alpha_m = ((m+1)^m/(m^m(m+2)))^{1/(m+1)}$.

Proof. It is well known that one can solve the linear programming relaxation of (3.1) by

(a) ordering the indices as (1), (2),...(n) so that

$$c_{(1)}/w_{(1)} \geq c_{(2)}/w_{(2)} \geq \cdots \geq c_{(n)}/w_{(n)};$$

(b) finding t such that

$$w_{(1)} + \cdots + w_{(t)} \leq 1 < w_{(1)} + \cdots + w_{(t+1)}$$

(c) then putting

$$x_{(j)} = 1 \qquad \text{for } j = 1,\dots,t,$$
$$x_{(t+1)} = w_{(1)} + \cdots + w_{(t+1)} - 1$$
$$x_{(j)} = 0 \qquad \text{for } j = t+1,\dots,n$$

(of course t may not exist in which case $x_j = 1$ for $j = 1,\dots,n$ solves (3.1)).

We shall show that

$$X'_{mn} = c_{(1)} + \cdots + c_{(t)} \sim \alpha_m n^{1/(m+1)} \quad \text{a.s.} \tag{3.4}$$

and this will imply (3.3) as $X'_{mn} \leq X_{mn} \leq X'_{mn} + 1$.

For future reference we refer to the t in (3.4) as $t(m, n)$.

We shall now use Lemma 3.1 to show that if $k=k(n)\to\infty$ as $n\to\infty$ and $k<n^{1/(m+1)}$ then, where $W_k=w_{(1)}+\cdots+w_{(k)}$ and $C_k=c_{(1)}+\cdots+c_{(k)}$,

$$\mathrm{Exp}(W_k)\sim(m/(m+2))((m+1)/n)^{1/m}k^{(m+1)/m}, \tag{3.5a}$$

$$\mathrm{Var}(W_k)=\mathrm{O}(k^{(2m+1)/m}/n^{2/m}), \tag{3.5b}$$

$$\mathrm{Exp}(C_k)\sim k(m+1)/(m+2), \tag{3.5c}$$

$$\mathrm{Var}(C_k)=\mathrm{O}(k). \tag{3.5d}$$

Then putting $k^*=\beta_m n^{1/(m+1)}$ where $\beta_m=((m+2)^m/m^m(m+1))^{1/(m+1)}$ we find

$$\mathrm{Exp}(W_k^*)\sim 1, \tag{3.6a}$$

$$\mathrm{Var}(W_k^*)=\mathrm{O}(n^{-1/m(m+1)}), \tag{3.6b}$$

$$\mathrm{Exp}(C_k^*)\sim\alpha_m n^{1/(m+1)}, \tag{3.6c}$$

$$\mathrm{Var}(C_k^*)=\mathrm{O}(n^{1/(m+1)}). \tag{3.6d}$$

Using the Chebycheff inequality we can easily show from (3.5) that

$$t(m,n)\sim k^* \quad \text{a.s.} \tag{3.7}$$

Then using the same inequality in conjunction with (3.6c), (3.6d) will prove the lemma.

Proof of (3.5a). From Lemma 3.1 with $p=m$, $q=1$, $a=1$ and $b=0$ we have

$$\mathrm{Exp}(w_{(k)})=\left(\sum_{r=0}^{k-1}\binom{n}{r}m^{n-r}+\sum_{r=k}^{n}\binom{n}{r}m^{n-r}\prod_{t=k}^{r}(tm/(tm+1))\right)\Big/\left((m+1)^{n-1}(m+2)\right).$$

Now one can easily show using induction that

$$((k-1)/r)^{1/m}\leqslant\prod_{t=k}^{r}(tm/(tm+1))\leqslant(k/(r+1))^{1/m}\quad\text{for } r\geqslant k\geqslant 1.$$

Thus putting $\epsilon=1/(m+1)$ we have

$$\mathrm{Exp}(w_{(k)})\leqslant((m+1)/(m+2))k^{1/m}S$$

where

$$S=\sum_{r=0}^{n}\binom{n}{r}\epsilon^r(1-\epsilon)^{n-r}(r+1)^{-1/m}.$$

We now use the Chernoff bounds [3]: for $0<\beta<1$

$$\sum_{r=0}^{(1-\beta)\epsilon n}\binom{n}{r}\epsilon^r(1-\epsilon)^{n-r}\leqslant \mathrm{e}^{-\beta^2 n\epsilon/2},$$

$$\sum_{r=(1+\beta)\epsilon n}^{n}\binom{n}{r}\epsilon^r(1-\epsilon)^{n-r}\leqslant \mathrm{e}^{-\beta^2 n\epsilon/3}.$$

In particular putting $\beta=\frac{1}{2}$ we get

$$\sum_{r=0}^{\epsilon n/2}\binom{n}{r}\epsilon^r(1-\epsilon)^{n-r}\leqslant \mathrm{e}^{-n\epsilon/8}.$$

Now putting $\beta=(\log n/n)^{1/2}$ we obtain

$$S\leqslant \mathrm{e}^{-n\epsilon/8}+(1/\epsilon n)^{1/m}\left(2^{1/m}n^{-\epsilon/2}+n^{-\epsilon/2}+\left(1/\left(1-(\log n/n)^{1/2}\right)\right)^{1/m}\right)$$

$$\leqslant(1/\epsilon n)^{1/m}\left(1+\gamma_1(\log n/n)^{1/2}\right)\quad\text{for some }\gamma_1>0.$$

Thus

$$\operatorname{Exp}(w_{(k)}) \leqslant ((m+1)/(m+2))(k/\epsilon n)^{1/m}\left(1+\gamma_1(\log n/n)^{1/2}\right). \tag{3.8a}$$

A similar argument now gives

$$\operatorname{Exp}(w_{(k)}) \geqslant ((m+1)/(m+2))((k-1)/\epsilon n)^{1/m}\left(1-\gamma_2(\log n/n)^{1/2}\right) \tag{3.8b}$$

for some $\gamma_2 > 0$.

Using the fact

$$\sum_{i=1}^{k} i^{1/m} = (m/(m+1))k^{(m+1)/m}(1+o(k))$$

we obtain (3.5a).

Proof of (3.5c). As $w_{(n)}/c_{(n)} \geqslant \cdots \geqslant w_{(1)}/c_{(1)}$ to get $\operatorname{Exp}(c_{(k)})$ we apply Lemma 3.2 with $p=1$, $q=m$, $a=1$ and $b=0$ and $n-k+1$ in place of k to get

$$\operatorname{Exp}(c_{(k)}) = \left(\sum_{r=0}^{n-k}\binom{n}{r}m^r + \sum_{r=n-k+1}^{n}\binom{n}{r}m^r\theta_r\right)\Big/\left((m+1)^{n-1}(m+2)\right).$$

We are only interested in small k and for these values the first summation dominates the second giving $\operatorname{Exp}(c_{(k)}) \sim (m+1)/(m+2)$ and (3.5c) follows immediately.

Proof of (3.5b). Using Lemma 3.1 we obtain

$$\operatorname{Exp}\left(w_{(k)}^2\right) \leqslant \left(k(k+1)(m+1)^2/n^2\right)^{1/m}((m+1)/(m+3))\left(1+\gamma_3(\log n/n)^{1/2}\right) \tag{3.9a}$$

for some $\gamma_3 > 0$,

$$\operatorname{Exp}(w_{(k)}w_{(l)}) \leqslant \left(k(l+1)(m+1)^2/n^2\right)^{1/m}\left((m+1)^2/(m+2)^2\right)\left(1+\gamma_4(\log n/n)^{1/2}\right) \tag{3.9b}$$

for some $\gamma_4 > 0$.

We now express the variance $\operatorname{Var}(W_k)$ by

$$\operatorname{Var}(W_k) = \sum_{s=1}^{k}\lambda_s + 2\sum_{s=1}^{k-1}\sum_{t=s+1}^{k}\mu_{st}$$

where

$$\lambda_s = \operatorname{Exp}\left(w_{(s)}^2\right) - \operatorname{Exp}(w_{(s)})^2$$

and

$$\mu_{st} = \operatorname{Exp}(w_{(s)}w_{(t)}) - \operatorname{Exp}(w_{(s)})\operatorname{Exp}(w_{(t)}).$$

From (3.9a) we can deduce that

$$\sum_{s=1}^{k}\lambda_s \text{ is } O(k^{1+2/m}/n^{2/m}).$$

From (3.8b) and (3.9b) we deduce that

$$\mu_{st} \text{ is } O\left(\left(t/n^2\right)^{1/m} + \left(st/n^2\right)^{1/m}(\log n/n)^{1/2}\right).$$

For $m \geqslant 2$ it is clear that we can ignore the second term as $s < n^{1/(m+1)}$. When $m=1$ however we can replace $(\log n/n)^{1/2}$ by $1/n$ in (3.8).

Here θ_r of Lemma 3.1 is simply $k/(r+1)$. One then uses $\binom{n}{r}/(r+1) = \binom{n+1}{r+1}/(n+1)$ to give $\operatorname{Exp}(w_{(k)}) = 4k/(n+1) - O((\frac{3}{4})^n)$ when $k < n^{1/2}$.

Thus

$$\sum_{s=1}^{k-1} \sum_{t=s+1}^{k} \mu_{st} \text{ is } O\left(k^{(2m+1)/m}/n^{2/m}\right)$$

and (3.5b) follows immediately.

The proof of (3.5d) is similar (and easier).

We immediately obtain a nice tight asymptotic result for Knapsack Problems ($m = 1$):

Theorem 3.1. $Z_{1n} \sim (2n/3)^{1/2}$ *a.s.*

Proof. $X_{1n} = Z_{1n}$.

For $m \geqslant 2$ all we know at present is

Theorem 3.2.

$$(1 - o(n))\alpha_m n^{1/(m+1)} \leqslant Z_{mm} \leqslant \gamma_m n^{1/(m+1)} \quad a.s.$$

where $\gamma_m = e^{m/(m+1)}$.

Proof. The lower bound is from Lemma 3.2 and the upper bound is obtained as follows. Let

$$Y_{mn} = \max \sum_{j=1}^{n} x_j,$$

$$\text{subject to } \Sigma a_{ij} x_j \leqslant 1, \quad i = 1,\dots,m,$$

$$x_j = 0 \text{ or } 1, \quad j = 1,\dots,n.$$

Clearly $Y_{mn} \geqslant Z_{mn}$. We obtain a probabilistic upper bound for Y_{mn}. For $S \subseteq N = \{1,\dots,n\}$ let E_S be the event: $\Sigma_{j \in S} a_{ij} \leqslant 1$ for $i = 1,\dots,m$.

If $|S| = k$ the $\Pr(E_S) = (1/k!)^m$ and hence

$$\text{Pb}(Y_{mn} \geqslant k) = \Pr(S \subseteq N\text{: } E_S \text{ occurs}) \leqslant \binom{n}{k}(1/k!)^m.$$

If $k = \gamma_m n^{1/(m+1)}$ we find that $\binom{n}{k}(1/k!)^m$ is $O(n^{-1/2})$.

A simple consequence of the preceding analysis is the following: Let $\hat{Z}_{mn}$ be the objective value of the solution to (1.1) obtained by rounding down the solution to LP(ϕ).

Since $Z_{mn} - \hat{Z}_{mn} < m$ we have

$$Z_{mn} - \hat{Z}_{mn} \leqslant \epsilon Z_{mn} \quad \text{a.s.}$$

provided ϵ is $O(n^{-a})$ where $a < 1/(m+1)$.

The main questions about the above model that this paper leaves unresolved are

(i) does $Z_{mn} \sim cn^{1/(m+1)}$ a.s. for some $\alpha_m \leqslant c \leqslant \gamma_m$?

(ii) what is the effect on accuracy of applying the approximation scheme of Section 2 for a small value of k?

References

[1] A.V. Aho, J.E. Hopcroft and J.D. Ullman, *The Desing and Analysis of Computer Algorithms* (Addison-Wesley, Reading, MA, 1974).

[2] A.K. Chandra, D.S. Hirschberg and C.K. Wong, Approximation algorithms for some generalised knapsack problems, *Theoret. Comput. Sci. 3* (1976) 293–304.

[3] P. Erdös and J. Spencer, *Probabilistic Methods in Combinatorics* (Academic Press, New York, 1974).

[4] P. Gács and L. Lovász, Khachian's algorithm for linear programming, *Math. Programming Study 14* (1981) 61–68.

[5] M.R. Garey and D.S. Johnson, *Computers and Intractibility: a Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1978).

[6] O.H. Ibarra and C.E. Kim, Fast approximation algorithms for the knapsack and sum of subset problems, *J. ACM 22* (1975) 463–468.

[7] D.S. Johnson, Approximation algorithms for combinatorial problems, *J. Comput. System Sci. 9* (1974) 256–278.

[8] L.G. Khachian, A polynomial algorithm in linear programming, *Soviet Math. Dokl. 20* (1979) 191–194.

[9] B. Korte and R. Schrader, On the existence of fast approximation schemes, Report No. 80163-OR, Institut für Ökonometrie und Operations Research, Rheinische Friedrich-Wilhelms-Universität, Bonn (1980).

[10] E.L. Lawler, Fast approximation algorithms for knapsack problems, *Proc. 18th Annual Symposium on Foundations of Computer Science* (1977).

[11] O. Oguz and M.J. Magazine, A polynomial time approximation algorithm for the multidimensional 0/1 knapsack problem, Working Paper, University of Waterloo (1980).

[12] S. Sahni, Approximation algorithms for the 0/1 knapsack problem, *J. ACM 22* (1975) 115–124.

[13] M. Simmonard, *Linear Programming* (Prentice-Hall, Englewood Cliffs, NJ, 1966).