Control
and Cybernetics
VOL. 12 (1983) No. 3-4

Probabilistic analysis of simple algorithms for binary knapsack problem

by

KRZYSZTOF SZKATUŁA

MAREK LIBURA

Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warszawa, Poland

In the paper two simple algorithms for binary knapsack problem are considered: the simplest algorithm (greedy algorithm without ordering of variables) and typical greedy algorithm. The probabilistic properties of the simplest algorithm are investigated. Recurrence equations describing the weight of knapsack in each iteration are obtained. These equations are then solved in the case of uniformly distributed weights of items packed into the knapsack and the distribution function of the knapsack weight is found. Next the mean value and variance of the knapsack weight as well as the knapsack value are calculated and asymptotic properties of them are investigated. Finaly, probabilistic behaviour of greedy algorithm is investigated using simulation approach. A numerical experiment is described and the mean value of the knapsack value is estimated. Both algorithms are then compared.

1. Introduction

The quality of approximate algorithms can be characterized by their accuracy and computational complexity. Usually these parameters of an algorithm are analysed in the worst case. Such analysis gives frequently too pessimistic conclusions. Such is for example the case with greedy algorithm for binary knapsack problem. One can practically observe pretty good behaviour of this algorithm although in the worst case its accuracy can be arbitrarily bad.

This disadvantage of the worst case analysis is partially overcome by the probabilistic analysis of algorithms. Unfortunately the latter is usually more complicated and results published concern mainly simple probabilistic models (see [1], [2], [3], [9]).

Most of the results are obtained for asymptotic case (when the size of problem tends to infinity). Usually only average (in probabilistic sense) properties of algorithms are analysed.

Recently there have been some attempts to perform more detailed probabilistic analysis for simple algorithms (see for example [7], [8], where the bin packing problem is considered). The goal of these attempts is to obtain not only some moments but also the distribution function for important parameters of algorithms. This is also approach presented in this paper.

Two simple algorithms for binary knapsack problem are considered. The first one is the simplest algorithm (sometime called greedy without ordering); the second is the greedy algorithm.

Most of analytical results concern the simplest algorithm. Recurrence equations for probabilistic distribution function describing the weight of knapsack in any iteration of the simplest algorithm are obtained (section 3). These equations are then solved in the case when weights of items packed into the knapsack are independent, uniformly distributed random variables (section 4). Some properties of this solution are then analysed (section 5), which allows to calculate mean value and variance of the objective function value generated by the simplest algorithm (section 6). Asymptotic properties of the algorithm are also considered (section 6).

It seems that it would be too difficult to repeat the same analysis for greedy algorithm. To have the possibility of comparing the behaviour of both algorithms the parameters of greedy algorithm are obtained by computer simulation and some analogues of analytical results presented earlier for the simplest algorithm are described.

In the paper the following notation is used. Random variables are denotes by capital Latin letters, their realizations by small letters. Symbol $P\{\cdot\}$ denotes the probability of event $\{\cdot\}$. For random variables X by $E(X)$ — their mean value and by $\mathrm{Var}(X)$ their variance is denoted.
R is the set of real numbers.
N denotes the set of natural numbers.

2. The simplest algorithm for binary knapsack problem

Consider the binary knapsack problem

$$\max z = \sum_{i=1}^{n} c_i \cdot x_i$$
$$\sum_{i=1}^{n} a_i \cdot x_i \leqslant b \tag{1}$$

where $a_i, c_i, b \in R$, $i=1, \ldots, n$.

An interpretation of (1) is the following: One has to pack a knapsack which can hold a maximum weight b. Each of n packed items has a weight a_i and a value c_i. The goal is to fill the knapsack in such a way, that their weight does not exceed b and the total value of choosen items is maximized.

In probabilistic analysis of an algorithm one needs a family π of problems. These problems are created by assuming that a_i, c_i, $i=1, \ldots, n$, are the realizations of random variables A_i and C_i, $i=1, \ldots, n$. Each possible collection of realizations of A_i and C_i, $i=1, \ldots, n$, defines an element of family π. Let $H_i(x)=P\{C_i \leqslant x\}$ be the distribution function of random variable C_i and

$$G_i(x)=P\{A_i \leqslant x\}.$$

Assume that A_i, $i=1, \ldots, n$, are independent random variables, $0<A_i \leqslant b$, and that b is a fixed constant. Let X_i, $i=1, \ldots, n$, be random variables corresponding to the decision variables of problem (1). The realizations of these variables are determined by an algorithm used to solve the problem (1). For $k=1, \ldots, n$, define $S_k=\sum_{i=1}^{k} A_i \cdot X_i$, $Z_k=\sum_{i=1}^{k} C_i \cdot X_i$ and let s_k and z_k be the realizations of S_k and Z_k.

The simplest algorithm for problem (1) proceeds in the following way:

The simplest algorithm

1° $x_1 := 1$
$s_1 := a_1$
$z_1 := c_1$
2° **for** $k=2, \ldots, n$ do
if $s_{k-1}+a_k \leqslant b$ then
$x_k := 1$
$s_k := s_{k-1}+a_k$
$z_k := z_{k-1}+c_k$
else
$x_k := 0$
$s_k := s_{k-1}$
$z_k := z_{k-1}$

It is easy to see that the computational complexity of above algorithm is $O(n)$.

In the following section the distribution function of random variable S_k, $k=1, \ldots, n$, (which is interpreted as a weight of knapsack after k-th iteration of the simplest algorithm) is considered.

3. Recurrence relations

Assume that $a_i \in R$, $i=1, \ldots, n$. Without loss of generality we can take $b=1$. Let $F_k(x)=P\{S_k \leqslant x\}$, $k=1, \ldots, n$, $x \in [0, 1]$.

LEMMA 1. *The following equations hold*

$$F_1(x)=G_1(x), \quad x \in [0, 1]$$

$$F_k(x)=\int_0^x [1+G_k(x-t)-G_k(1-t)]\, dF_{k-1}(t) \tag{2}$$

$$\text{for } k=2, \ldots, n, \quad x \in [0, 1].$$

Proof. The initial condition $F_1(x)=G_1(x)$, $x\in[0,1]$ is an immediate consequence of Step 1° of the algorithm.

For $2\leqslant k\leqslant n$ there are two possibilities:

(i) $S_{k-1}+A_k\leqslant 1$ and then $X_k=1$, $S_k=S_{k-1}+A_k$;

(ii) $S_{k-1}+A_k>1$ and then $X_k=0$, $S_k=S_{k-1}$

This implies that

$$P\{S_k\leqslant x\}=P\{S_k\leqslant x \text{ and } X_k=1\}+P\{S_k\leqslant x \text{ and } X_k=0\}=$$
$$=P\{S_{k-1}+A_k\leqslant x\}+P\{S_{k-1}+A_k>1 \text{ and } S_{k-1}\leqslant x\} \tag{3}$$

Random variables A_i, $i<k$, and A_k are independent. Thus

$$P\{S_k\leqslant x \text{ and } X_k=1\}=P\{S_{k-1}+A_k\leqslant x\}=\int_0^x G_k(x-t)dF_{k-1}(t) \tag{4}$$

An event $S_{k-1}+A_k>1$ and $S_{k-1}\leqslant x$ is equal to the sum of events $\{t+A_k>1\}$ and $\{S_{k-1}=t\}$ for all $t\leqslant x$, so

$$P\{S_k\leqslant x \text{ and } X_k=0\}=P\{S_{k-1}+A_k>1 \text{ and } S_{k-1}\leqslant x\}=$$
$$=\int_0^x P\{t+A_k>1\}\,dF_{k-1}(t) \tag{5}$$

From (3), (4), (5) we obtain for $k=2,\ldots,n$.

$$P\{S_k\leqslant x\}=\int_0^x [G_k(x-t)+1-G_k(1-t)]\,dF_{k-1}(t)$$ ■

In the case of discrete variables A_i similar recurrence relations can be obtained.

Assume $a_i\in N$, $i=1,\ldots,n$, and $b\in N$.

Denote $p_i^k=P\{S_k=i\}$

$$p^k(i)=P\{S_k\leqslant i\}=\sum_{j=1}^{i} p_j^k$$

$g_i^k=P\{A_k=i\}$

$G^k(i)=P\{A_k\leqslant i\}$

In this case the following relations hold (see [10]):

$$P^1(i)=G^1(i), \quad i=1,\ldots,b$$
$$P^k(i)=\sum_{j=1}^{i}[1+G^k(i-j)-G^k(b-j)]p_j^{k-1}, \quad k=2,\ldots,n, \quad i=1,\ldots,b \tag{6}$$

Moreover,

$$P\{S_k\leqslant i \text{ and } X_k=1\}=\sum_{j=1}^{i} G^k(i-j)p_j^{k-1} \tag{7}$$

$$P\{S_k\leqslant i \text{ and } X_k=0\}=\sum_{j=1}^{i}[1-G^k(b-j)]p_j^{k-1} \tag{8}$$

4. Solving of recurrence relations in the case of uniform distribution of A_i, $i=1, \ldots, n$

Consider at first a continuous case. Assume that

$$G_i(x)=G(x)=\begin{cases}0 & \text{for } x<0\\ x & \text{for } x\in[0,1],\\ 1 & \text{for } x>1\end{cases} \qquad i=1,\ldots,n \tag{9}$$

LEMMA 2. *If A_i, $i=1, \ldots, n$, are independent and* (9) *holds then*

$$F_k(x)=x^k, \qquad k=1,\ldots,n, \qquad x\in[0,1] \tag{10}$$

solves the recurrence relations (2).

Proof. Substituting in (2) $G_k(x)=x$ for $x\in[0,1]$ we obtain

$$F_1(x)=x$$

$$F_k(x)=\int_0^x x\,dF_{k-1}(t), \qquad k=2,\ldots,n$$

which immediately implies that $F_k(x)=x^k$, $k=1, \ldots, n$, $x\in[0,1]$. Observe that the density function of S_k is

$$f_k(x)=k\cdot x^{k-1}, \qquad x\in[0,1], \qquad k=1,\ldots,n \tag{11}$$

In the case when A_i, $i=1, \ldots, n$, are discrete, independent and uniformly distributed random variables a solution of (6) is the following: Assume

$$g_j^i=g_j=1/b, \qquad j=1,\ldots,b, \qquad i=1,\ldots,n$$

Then

$$P^k(i)=(i/b)^k, \qquad i=1,\ldots,b, \qquad k=1,\ldots,n \tag{12}$$

$$p_i^k=\frac{i^k-(i-1)^k}{b^k}, \qquad i=1,\ldots,b, \qquad k=1,\ldots,n \tag{13}$$

5. On the properties of solutions of (2)

In the following we analyse some properties of obtained solutions of recurrence relations. At first we prove a theorem from which as a corollary we obtain the independence of events $\{S_k\leqslant x\}$, $\{X_i=\delta_i\}$, $\delta_i=0$ or 1, $i=1, \ldots, k$, and expressions for $P\{X_i=\delta_i\}$, $\delta_i=0$ or 1, $i=1, \ldots, n$. This allows us to find a mean value and variance of the value of knapsack and analyse asymptotic properties of these moments.

Theorem. *Let A_i, $i=1, \ldots, k$, be independent random variables uniformly distributed on* $[0, 1]$ *and $\delta_i=0$ or 1, $i=1, \ldots, k$. Then*

$$P\{S_k\leqslant x \quad \text{and} \quad X_i=\delta_i,\ i=1,\ldots,k\}=x^k\prod_{i=1}^{j}\frac{i-1-(1-2)\delta_i}{i} \tag{14}$$

Proof. The proof is by induction. Denote

$$d_j(\delta_1, \ldots, \delta_j) = \prod_{i=1}^{j} \frac{i-1-(i-2)\delta_i}{i}$$

Observe that according to (4), (5) for $i=1$

$$P\{S_1 \leqslant x \text{ and } X_1=\delta_1\} = x\delta_1$$

which coincides with (14).

Assume that for some j, $1 \leqslant j \leqslant k-1$, the formula (14) holds, i.e.,

$$P\{S_j \leqslant x \text{ and } X_i=\delta_i,\ i=1,\ldots,j\} = x^j d(\delta_1, \ldots, \delta_j) \tag{15}$$

We will show that this implies that (14) is fulfilled also for $j+1$, i.e.,

$$P\{S_j \leqslant x \text{ and } x_i=\delta_i,\ i=1,\ldots,j,j+1\} = x^{j+1} d_{j+1}(\delta_1, \ldots, \delta_j, \delta_{j+1}).$$

Two cases are considered:

(i) If $\delta_{j+1}=1$ (it means that the item $(j+1)$ can be packed) then

$$P\{S_{j+1} \leqslant x \text{ and } X_i=\delta_i,\ i=1,\ldots,j,\ X_{j+1}=1\} =$$

$$= P\{S_j + A_{j+1} \leqslant x \text{ and } X_i=\delta_i,\ i=1,\ldots,j\} =$$

$$= \int_0^x G_{j+1}(x-t)\, d_t P\{S_j \leqslant t \text{ and } X_i=\delta_i,\ i=1,\ldots,j\} =$$

$$= \int_0^x (x-t) j t^{j-1} d_j(\delta_1, \ldots, \delta_j)\, dt =$$

$$= x^{j+1} d_j(\delta_1, \ldots, \delta_j) \frac{1}{j+1} = x^{j+1} d_{j+1}(\delta_1, \ldots, \delta_j, \delta_{j+1}). \quad \text{(by (15) and (9))}$$

(ii) Similarly, for $\delta_{j+1}=0$

$$P\{S_{j+1} \leqslant x \wedge X_i=\delta_i,\ i=1,\ldots,j,\ X_{j+1}=0\} =$$

$$= P\{S_j + A_{j+1} > 1 \quad \text{and } S_j \leqslant x \quad \text{and } X_i=\delta_i,\ i=1,\ldots,j\} =$$

$$= \int_0^x [1 - G_{j+1}(1-t)\, d_t P\{S_j \leqslant t \quad \text{and } X_i=\delta_i,\ i=1,\ldots,j\}] =$$

$$= x^{j+1} d_j(\delta_1, \ldots, \delta_j) \frac{j}{j+1} = x^{j+1} d_{j+1}(\delta_1, \ldots, \delta_j, \delta_{j+1}). \quad \text{(by (15) and (19))}$$ ■

Corollary 1.

$$P\{X_k=1\} = 1/k, \quad k=1, \ldots, n \tag{16}$$

$$P\{X_k=0\} = (k-1)/k, \quad k=1, \ldots, n. \tag{17}$$

Proof. Taking $x=1$ in (14) and summing over all δ_i, $i \neq k$, we obtain (16) and (17). ■

COROLLARY 2. *The events* $\{S_k \leqslant x\}$, $\{X_i=\delta_i\}$, $i=1, \ldots, k$, *are independent.*

Proof. From (14) and (10), (16), (17) we have

$$P\{S_k \leqslant x \text{ and } X_i=\delta_i,\ i=1, \ldots, k\}=P\{S_k \leqslant x\} \prod_{i=1}^{k} P\{X_i=\delta_i\} \qquad \blacksquare$$

6. Mean value and variance of the knapsack weight and value for the simplest algorithm

The results obtained in section 5 allows us to calculate mean value and variance of the weight (S_k) and the value (Z_k) of the knapsack generated by the simplest algorithm in k-th step. Assume that

A_i, $i=1, \ldots, n$, are independent random variables uniformly distributed on $[0, 1]$. (18)

From Lemma 2 we have $F_k(x)=x^k$, $k=1, \ldots, n$, $x \in [0, 1]$, hence for $k=1, \ldots, n$

$$E(S_k)=\int_0^1 x dF_k(x)=\frac{1}{k+1} \tag{19}$$

$$\operatorname{Var}(S_k)=E(S_k^2)-E(S_k)^2=\frac{k}{(k+2)(k+1)^2} \tag{20}$$

Using (19) and (20) one can now evaluate the probability that the random variable S_k differs from its mean value by given $\varepsilon>0$. Substituting in well known inequality (21) (see [5]) $Y=(S_k-E(S_k))^2$, $R=\varepsilon^2$

$$P\{Y \geqslant R\} \leqslant E(Y)/R \tag{21}$$

we obtain

$$P\{|S_k-E(S_k)| \geqslant \varepsilon\} \leqslant \frac{\operatorname{Var}(S_k)}{\varepsilon^2}=\frac{k}{(k+2)(k+1)^2 \varepsilon^2} \tag{22}$$

Assume now that random variables A_i, $i=1, \ldots, n$, and C_j, $j=1, \ldots, n$, are mutually independent, (18) holds and, moreover, that

$$E(C_j)=V=\text{const}, \quad j=1, \ldots, n. \tag{23}$$

Then the mean value of the objective value (i.e., the value of knapsack) generated by the simplest algorithm in k-th step can be calculated as follows

$$E(Z_k)=E\Big(\sum_{i=1}^{k} C_i X_i\Big)=\sum_{i=1}^{k} E(C_i X_i)=$$

$$=\sum_{i=1}^{k} E(C_i) E(X_i)=(\text{by Corollary 1})=V \sum_{i=1}^{k} \frac{1}{i} \tag{24}$$

This way we have proved the following corollary:

COROLLARY 3. *If* A_i, $i=1, \ldots, n$, *and* C_j, $j=1, \ldots, n$, *are mutually independent and* (18), (23) *hold then the mean value of the objective function obtained by the simplest algorithm is given by the following formula*

$$E(Z_n)=V\cdot\sum_{i=1}^{n}\frac{1}{i}$$

Observe that for large $nE(Z_n)$ behaves as logarithm of n because $\sum_{i=1}^{n}\frac{1}{i}=\ln n+$ $+\gamma+o(n)$, where $o(n)\to 0$ if $n\to\infty$ and $\gamma\approx 0.5772$ is Euler's constant. Thus we have for large n

$$E(Z_n)=V(\ln n+\gamma)+o(n) \tag{25}$$

In a similar way a variance of the objective value generated by the simplest algorithm can be calculated.

Assume that

$$E(C_i^2)=W=\text{const} \tag{26}$$

and, moreover, the assumptions of Corollary 3 hold. Then for $k=1, \ldots, n$

$$\begin{aligned}\operatorname{Var}(Z_k)=E(Z_k^2)-(E(Z_k))^2&=E\Big(\Big(\sum_{i=1}^{k}C_iX_i\Big)^2\Big)-\Big(E\Big(\sum_{i=1}^{k}C_iX_i\Big)\Big)^2=\\&=\sum_{i=1}^{k}E(C_i^2X_i^2)+\sum_{i=1}^{k}\sum_{\substack{j=1\\j\neq i}}^{k}E(C_iC_jX_iX_j)+\\&-\sum_{i=1}^{k}(E(C_iX_i))^2-\sum_{i=1}^{k}\sum_{\substack{j=1\\j\neq i}}^{k}E(C_iX_i)E(C_jX_j)=\\&\text{(by Corollary 2 and above assumptions)}\\&=\sum_{i=1}^{k}E(C_i^2)E(X_i^2)-\sum_{i=1}^{k}(E(C_i)E(X_i))^2\end{aligned}$$

From Corollary 1 it is easy to calculate that $E(X_i^2)=\frac{1}{i}$. Using (16), (17), (25), (26) and above formula for $\operatorname{Var}(Z_k)$ we obtain for $k=n$ the following corollary:

COROLLARY 4. *If* A_i, $i=1, \ldots, n$, *and* C_j, $j=1, \ldots, n$, *are independent random variables and* (18), (25), (26) *hold then*

$$\operatorname{Var}(Z_n)=W\sum_{i=1}^{n}\frac{1}{i}-V^2\sum_{i=1}^{n}\frac{1}{i^2} \tag{27}$$

Observe that for large n from the fact that $\lim_{n\to\infty}\sum_{i=1}^{n}\frac{1}{i^2}=\frac{\pi^2}{6}$ it follows that

$$\operatorname{Var}(Z_n)=W(\ln n+\gamma)-V^2\frac{\pi^2}{6}+o(n)$$

7. Analysis of greedy algorithm for the knapsack problem

The simplest algorithm is a very useful theoretical model but it can be hardly regarded as a practical method for solving binary knapsack problem.

On the other hand, the greedy algorithm which differs from the simplest algorithm by additional initial phase, is of great theoretical and practical importance. This initial phase consists in ordering of items to satisfy the following condition

$$\frac{c_{i+1}}{a_{i+1}} \leqslant \frac{c_i}{a_i}, \quad i=1, \ldots, n-1 \tag{28}$$

After completing this initial phase greedy algorithm performs exactly as the simplest algorithm. In the worst case greedy algorithm has time complexity $O(n \cdot \ln n)$. Observe that after initial phase random variables C_i, A_i, $i=1, \ldots, n$, are not independent anymore, which significantly complicates the analysis. For this reason we were not able to obtain for greedy algorithm theoretical results like presented above. Some equivalents of these theoretical results for greedy algorithm were obtained in the experimental way.

Computational experiment was organized in the following way: For each size $n=100, 200, \ldots, 1000, 2000, 3000, 5000$, 50 random problems were generated assuming A_i, C_i, $i=1, \ldots, n$, to be uniformly distributed on $[0, 1]$ independent random variables. Then mean value and variance of knapsack value were estimated. Let V_n denote a random variable, whose realizations are values of knapsack generated by greedy algorithm. Results of experiment suggested the following dependence of the mean value of V_n on n:

$$L(n)=A+Bn^T\left(\ln^R(n)+C\right) \tag{29}$$

where A, B, C, R, T ($T \leqslant 1$) are constants.

Using methods of nonlinear regression these constants are determinated which allows to formulate the following conclusions.

1° If A_i, C_i, $i=1, \ldots, n$, are independent, uniformly distributed on $[0, 1]$ random variables, then $E(V_n)$ can be approximated by the following formula

$$E(V_n) \approx 0.552+0.012 \cdot \sqrt{n} \cdot (\ln n+61.8) \tag{30}$$

Table 1 compares estimated mean values of $E(V_n)$ with values given by (30). High quality of approximation is observed for considered problem dimensions (the standard deviation is equal to 0.114).

Another observation derived from the results of experiment is the following:

2° Under the assumptions of Hypothesis 1 there exist a constant $D>0$ such that

$$\operatorname{Var}(V_n) < D \cdot \ln n \tag{31}$$

For obtained data it is enough to take $D=1.15$

Observe that if we assume that facts 1°, 2° hold for large n, then

$$\lim_{n \to \infty} \frac{\operatorname{Var}(V_n)}{E(V_n)} = 0 \tag{32}$$

and for $\varepsilon>0$ (from (21))

$$P\left\{\left|\frac{V_n-E(V_n)}{\ln n}\right|\geqslant\varepsilon\right\}\leqslant\frac{D}{\varepsilon^2\ln n} \tag{33}$$

3° If (32) holds then the mean value of knapsack value Z_n can be regarded as representative for solutions which can appear. Compare that for the simplest algorithm we have

$$\lim_{n\to\infty}\frac{\mathrm{Var}(Z_n)}{E(Z_n)}=\mathrm{const} \tag{34}$$

Table 1

$n=$	estimated mean values of V_n	values of $L(n)$	absolute value of difference
100	7.537	8.020	0.483
200	11.546	11.438	0.107
300	14.016	14.033	0.017
400	16.305	16.185	0.120
500	18.004	18.088	0.084
600	19.845	19.797	0.047
700	21.214	21.402	0.188
800	22.958	22.886	0.072
900	24.166	24.282	0.116
1000	25.524	25.604	0.080
2000	36.838	36.596	0,241
3000	44.779	44.636	0.143
5000	57.793	57.881	0.080

4° We can observe that the greedy algorithm is much better than the simplest algorithm. Formally this fact can be expressed as follows

$$\lim_{n\to\infty}\frac{E(Z_n)}{E(V_n)}=0 \tag{35}$$

According to (25) and above observation the ratio $\frac{E(Z_n)}{E(V_n)}$ tends to 0 as fast as $\frac{1}{\sqrt{n}}$.

Conclusions

The results presented in the paper concern very simple algorithms and probabilistic models. Only for the simplest algorithm analytical results are obtained. Although it would be desired to have results of this type for more complex models there are some doubts whether similar approach can be significantly extended.

The main difficulty lies in the dependence of random variables appearing in the model after ordering decision variables. The step of ordering is pretty common in greedy-like procedures and these algorithms seem to be intractable with the approach presented. On the other hand the algorithms which do not destroy initial independence of random data could be regarded as good candidates for performing similar probabilistic analysis.

Bibliography

[1] d'Atri G. Probabilistic analysis or the knapsack problem. Rapport de Recherche No. 7, Octobre 1978. Groupe de Recherche 22, Univ. Paris VI.

[2] d'Atri G. Outline of probabilistic framework for combinatorial optimization. Numerical Techniques for Stochastic Systems. North-Holland Publishing Company, 1980.

[3] Ausiello G., Marchetti-Spaccamela A., Protasi M. Probabilistic analysis of the performance of greedy strategies over some combinatorial problems. Istituto di Analisi dei Sistemi ed. Informatica R. 21, Novembre 1981.

[4] Fisher M. L. Worst-Case analysis of heuristic algorithms. Management Science, vol. 26, No. 1, 1980.

[5] Fisz M. Probability mathematics and mathematical statistics (in Polish) PWN, Warszawa, 1969.

[6] Garey M. R., Johnson D. J. Computers and intractability. Guide to the theory of NP — completeness. Freeman and Co., 1979.

[7] Ong H. L. Analysis of the expected performance of bin packing heuristics. Working Paper No. 139 Department of Management Science University of Waterloo, 1981.

[8] Ong H. L., Wee T. S., Magazine M. J. Analysis of the probabilistic performance of bin packing heuristics. Working Paper No. 144 Department of Management Science, University of Waterloo, 1981.

[9] Słomiński L. Probabilistic analysis of combinatorial algorithms: A bibliography with selected annotations. Computing 28, 257—267, 1982.

[10] Szkatuła K., Libura M. Probabilistic analysis of simplest algorithm for binary knapsack problem (in Polish) Working Paper, ZPM — 7/82, September 1982.

[11] Szkatuła K. Probabilistic analysis and experimental comparison of simple algorithms for binary knapsack problem (in Polish) Working Paper, ZPM — 11/82. November 1982.

Received, March 1983

Analiza probabilistyczna prostych algorytmów dla binarnego zadania załadunku

W pracy rozważone są dwa algorytmy przybliżone dla binarnego zadania załadunku (tzw. zadania pakowania plecaka): algorytm najprostszy (algorytm zachłanny bez porządkowania zmiennych) oraz typowy algorytm zachłanny. Wyprowadzone są zależności rekurencyjne opisujące zapełnienie plecaka w kolejnych krokach algorytmu najprostszego i uzyskiwane są ich rozwiązania dla równomiernych rozkładów prawdopodobieństwa wag elementów. Wyznaczane są: wartość średnia i wariancja funkcji celu generowane przez algorytm. Odpowiednie wielkości dla algorytmu zachłannego są uzyskiwane w eksperymencie obliczeniowym a następnie oba algorytmy są porównane.

Вероятностный анализ простых алгоритмов для булевой задачи о рюкзаке

В работе рассмотрены два приближённые алгоритмы для решения булевой задачи о рюкзаке на максимум: простейший алгоритм (гриди алгоритм без переупорядочивания переменных) и обыкновенный гриди алгоритм. Для простейшего алгоритма получена вероятностная характеристика результатов очередных его шагов. Соответствующая характеристика гриди алгоритма получена путём численного эксперимента. В заключительной части работы сравнивается поведение обоих алгоритмов для булевой задачи о рюкзаке на максимум.