The average behaviour of greedy algorithms for the knapsack problem: Computational experiments

B.Bank [*], G.Diubin [†], A.Korbut[‡], I.Sigal [§]

Abstract

We describe primal and dual greedy algorithms for the one-dimensional knapsack problem with Boolean variables. A theorem concerning their average behaviour is formulated. It is supposed that all coefficients of the problem are independent random variables uniformly distributed on [0,1], and $b = \lambda n$. The theorem asserts that for λ exceeding the "critical" value $1/2 - t/3$ both algorithms have asymptotical tolerance t. The main goal of the experiments was clarifying the behaviour of the algorithms for pre-critical value of λ. A brief characterization of a computer program implementing these methods together with preliminary results of the experiments is given. These results confirm the good behaviour of both methods and suggest some interesting theoretical problems.

1 Introduction

Our main object is the classical knapsack problem with Boolean variables. It consists in finding

$$f^* = \max\{\sum_{j=1}^{n} c_j x_j | \sum_{j=1}^{n} a_j x_j \leq b,\ x_j \in \{0,1\},\ j = 1,\ldots,n\}. \tag{1}$$

All coefficients in (1) are positive. The standard interpretation of the problem (1) is the following: we have to fill a knapsack of capacity b with the most profitable subset of items from $\{1,\ldots,n\}$, where each item j is characterized by its utility c_j and weight a_j. The Boolean variables x_j equal 1 if the item j is chosen, and 0 otherwise.

Without loss of generality, we can suppose that $a_j < b$ for all j and that $\sum_{j=1}^{n} a_j > b$. Besides, we shall suppose that

$$\frac{c_1}{a_1} \geq \frac{c_2}{a_2} \geq \cdots \geq \frac{c_n}{a_n}, \tag{2}$$

i.e., the variables x_j are numbered in the non-increasing order of their "weight densities" c_j/a_j. The condition (2) is often called the *regularity condition*.

The problem (1) has numerous applications, and it is one of the main models of combinatorial optimization (see [7] and the new monograph [4]). From the viewpoint of the general complexity theory, it is NP-hard. This means that exact algorithms with polynomial complexity can only exist in the case $P = NP$. Therefore, the main research efforts are now concentrated around

[*]Humboldt-Universität zu Berlin, Institut für Mathematik
[†]Institute for Economics and Mathematics, Russian Academy of Sciences, St.Petersburg
[‡]Institute for Economics and Mathematics, Russian Academy of Sciences, St.Petersburg. Partially supported by the DFG, Grant 436 RUS 17/59/99
[§]Computing Center, Russian Academy of Sciences, Moscow

approximate methods for the problem (1), and this tendency is characteristic for the entire combinatorial optimization.

Among these approximate methods, the so-called *greedy* methods play a major role. They can be interpreted as discrete analogs of gradient (or steepest-ascent) methods in continuous optimization. Their undoubted advantage is that for the problem (1) they work in linear time (if the regularity condition (2) is fulfilled). The greedy methods do not guarantee optimality; however, theoretical estimations of their worst-case performance can be given. Details can be found in the survey paper [6].

The idea of the greedy algorithm for the problem (1) consists in a consecutive selection of items with the largest weight density c_j/a_j until the knapsack capacity admits it. More formally, the algorithm starts with a feasible solution $x = (0, \ldots, 0)$ and consecutively replaces zeroes by ones in the order of decreasing ratios c_j/a_j (i.e., from the left to the right); every time the feasibility of the corresponding solution is checked. The process terminates after obtaining the last feasible solution. This solution x^G is called the *greedy solution*; the corresponding objective function value is denoted by f^G. Thus, for a greedy solutiuon x^G we have $x_1^G = 1$ (since $a_1 < b$) and for $k = 2, \ldots, n$

$$x_k^G = \begin{cases} 1 & \text{if} \quad \sum_{j=1}^{k-1} a_j x_j + a_k \leq b, \\ 0 & \text{if} \quad \sum_{j=1}^{k-1} a_j x_j + a_k > b. \end{cases} \tag{3}$$

An idea which is in some sense opposite, consists in a consecutive rejecting the least profitable items (again, in the sense of the ratios c_j/a_j) until the remaining ones fit in the knapsack. In accordance with the usual terminology, such algorithms can be called *dual* algorithms. Therefore the greedy algorithm described above will be termed *primal*. More formally, the dual greedy algorithm starts with an infeasible solution $x = (1, \ldots, 1)$ and consecutively replaces ones by zeroes in the order of increasing ratios c_j/a_j (i.e., from the right to the left). Every time the feasibility of the current solution is checked. The process terminates when the first feasible solution is obtained. This solution x^{DG} is called the *dual greedy* solution; the corresponding objective function value is denoted by f^{DG}. Thus, the dual greedy solution x^{DG} has the following from:

$$x_k^{DG} = \begin{cases} 1 & k = 1, \ldots, s \\ 0 & k = s+1, \ldots, n, \end{cases} \tag{4}$$

where the index s is defined as

$$s = \max \left\{ r \,\middle|\, \sum_{j=1}^{r} a_j \leq b \right\}. \tag{5}$$

Up to now, practically no attention to the analysis of dual greedy algorithms was paid. The reason was probably the following "folklore theorem".

Proposition 1 *The dual greedy algorithm for the problem (1) can be arbitrarily bad.*

It is natural to estimate the performance of the dual greedy algorithm by the ratio $R_{DG} = f^{DG}/f^*$. The assertion means that R_{DG} can take arbitrarily small values. To prove this, we consider the following one-parametric family of instances of (1):

$$\max\left\{3x_1 + 2\lambda x_2 | x_1 + \lambda x_2 \le \lambda,\ x_1, x_2 \in \{0,1\}\right\}, \tag{6}$$

where $\lambda > 3/2$. We have $x^* = (0,1)$ and $f^* = 2\lambda$. At the same time, $x^{DG} = (1,0)$, $f^{DG} = 3$. Thus, $R_{DG} = 3/2\lambda$ tends to zero when $\lambda \to \infty$.

Along with the problem (1) we consider its linear relaxation which consists in finding

$$f^{LR} = \max\left\{\sum_{j=1}^{n} c_j x_j \,\middle|\, \sum_{j=1}^{n} a_j x_j \le b, \quad 0 \le x_j \le 1, \quad j = 1,\dots,n\right\}. \tag{7}$$

It is well-known (cf., e.g., [7],[8]) that its optimal solution x^{LR} can be determined explicity:

$$x_k^{LR} = \begin{cases} 1, & k = 1,\dots,s \\ \alpha, & k = s+1, \\ 0, & k = s+2,\dots,n, \end{cases} \tag{8}$$

where $0 \le \alpha < 1$ and the index s is defined by (5). The value of α can be easily computed:

$$\alpha = \frac{b - \sum_{j=1}^{s} a_j}{a_s}. \tag{9}$$

From (4), (5) and (8) we see that the dual greedy solution x^{DG} contains the first block of consecutive ones from the primal greedy solution x^G (and only this block), and x^{LR} differs from x^{DG} at most by the component α.

Proposition 2 *The following inequalities hold:*

$$f^{DG} \le f^G \le f^* \le f^{LR}. \tag{10}$$

Note that non-trivial is only the first inequality which was proved in [2], [3] (and, for the general case of independence systems in [5]). This inequality means that the dual greedy algorithm is always not better than the primal one. We showed above (cf. (6)) that in some cases it can be arbitrarily bad. However, in computations for some applied problems such bad behaviour was never observed: the results were invariably rather good. This contradiction between theoretically bad and practically good behaviour can be resolved by analyzing the behaviour of algorithms not in the worst-case but in average. This analysis in average leads in most cases to substantially better results which completely agree with the computational practice.

The analysis of algorithms behaviour in average requires defining some probabilistic structure on the set of data. We make the following assumptions concerning the problem (1):

1) The coefficients c_j, a_j, $j = 1,\dots,n$ are independent random variables uniformly distributed on [0,1];

2) The right-hand side b of the constraint is proportional to the number of variables n, i.e. $b = \lambda n$ where $0 < \lambda < 1$.

We will study the behaviour of approximate algorithms for problems with n variables when n is growing. Consider an approximate algorithm $\mathcal{A}$, which, in order to stress its dependence on the number of variables, we shall denote by $\mathcal{A}_n$. Let $f^{\mathcal{A}_n}$ be the objective function value obtained by $\mathcal{A}_n$. We say that the algorithm $\mathcal{A}_n$ has asymptotic tolerance $t > 0$ if

$$\mathbf{P}(f^* - f^{\mathcal{A}_n} \le t) \underset{n\to\infty}{\longrightarrow} 1. \tag{11}$$

In the sequel we consider the problem (1), and $\mathcal{A}_n$ will be the primal and the dual greedy algorithms.

In [2], [3] the following theorem has been established.

Theorem 1 *Let the assumptions 1), 2) be fulfilled and*

$$\lambda > \frac{1}{2} - \frac{t}{3}. \tag{12}$$

The both the primal and the dual greedy algorithms have asymptotic tolerance t.

We call the value $1/2 - t/3$ in the right-hand side of (12) the *critical value* of the parameter λ and denote it by λ_0. The theorem guarantees that for $\lambda > \lambda_0$ both methods are in a certain sense "good". One of the goals of our computational experiment was clarifying and comparison of the behaviour of primal and dual algorithms for various n and for "pre-critical" values of the parameter λ.

2 A brief characterization of the program

For carrying out the experiments, a computer program was compiled. This program generates series of N random instances and implements primal and dual greedy algorithms for the solution of each instance. The results are then averaged for each series of N instances. The approximate objective function values f^G, f^{DG} are compared not with the optimal value f^* (which is unknown) but with its upper bound f^{LR} which can be easily computed (cf. (8), (9)).

The program includes the following main procedures:

1^o. Generating the random coefficients c_j, a_j, $j = 1, \ldots, n$ which are uniformly distributed on the intervals $[c_{\min}, c_{\max}]$, $[a_{\min}, a_{\max}]$ respectively (in accordance with the assumption 1) made above, both these intervals were always [0,1]).

2^o. Solving the linear relaxation (7) yielding its optimal solution x^{LR} and the optimal objective function value f^{LR}.

3^o. Finding the primal greedy solution x^G and the corresponding objective function value f^G.

4^o. Finding the dual greedy solution x^{DG} and the corresponding objective function value f^{DG}.

5^o. Regression analysis (see below).

At the initializing stage the program requires from the user the input of the following parameters.

1. The initial $t_{\min}$ and the final $t_{\max}$ values of the tolerance t where $t_{\min} \le t_{\max}$.

1a. The step size h_t of the parameter t (only in the case $t_{\min} < t_{\max}$).

2. The initial $n_{\min}$ and the final $n_{\max}$ values of the number of variables, where $n_{\min} \le n_{\max} \le 3700$.

2a. The step size h_n of the parameter n (only in the case $n_{\min} < n_{\max}$).

3. The minimal $c_{\min}$ and maximal $c_{\max}$ values of the coefficients c_j, $j = 1, \ldots, n$.

4. The minimal $a_{\min}$ and maximal $a_{\max}$ values of the coefficients a_j, $j = 1, \ldots, n$.
5. The number N of instances in a series for all values of n.
6. The minimal $\lambda_{\min}$ and maximal $\lambda_{\max}$ values of the parameter λ where $\lambda_{\min} \leq \lambda_{\max}$.
6a. The step size h_λ of the parameter λ (only in the case $\lambda_{\min} < \lambda_{\max}$).

After termination, the program yields, for each value of n, the average values of the following parameters:

$$f^{DG},\ f^{G},\ f^{LR},$$

$$\begin{aligned}
\varepsilon_1 &= f^G - f^{DG}, \\
\varepsilon_2 &= \frac{f^G - f^{DG}}{f^G} \cdot 100\%, \\
\varepsilon_3 &= \frac{f^G - f^{DG}}{f^G}, \\
\varepsilon_4 &= f^{LR} - f^{DG}, \\
\varepsilon_5 &= f^{LR} - f^G.
\end{aligned}$$

Besides, the program output contains two frequencies p_4 and p_5 where

$$p_i = \frac{\text{number of problems with } \varepsilon_i \leq t}{N},$$

$i = 4, 5$. Thus, for each λ, the value $p_4 = p_4(\lambda)$ characterizes the behaviour of the dual greedy algorithm, and $p_5 = p_5(\lambda)$ – that of the primal greedy algorithm. These values are approximations to the probabilities that the respective algorithms have an asymptotic tolerance t (cf. (11)).

Besides, for the regime with $n_{\min} = n_{\max}$, $t_{\min} = t_{\max}$, $\lambda_{\min} < \lambda_{\max}$, a regression procedure can be included which approximates the points in the plane $(\lambda,\ p_i(\lambda))$, $i = 4, 5$ by algebraic polynomials of degree s, $s = 1, 2, 3, 4$. The step size in λ is subject to the constraint that the number of points in the interval $[\lambda_{\min}, \lambda_{\max}]$ should not exceed 100. The output of this procedure is a matrix with the number of rows equal to the number of points in the interval $[\lambda_{\min}, \lambda_{\max}]$. In each row, the following values are printed: the row number, the value of λ, the frequency and the values of approximating polynomials for all degrees $s = 1, 2, 3, 4$. After that, for each degree s, the sum of squared deviations together with the maximal and minimal alsolute deviations of the approximating curve from the respective frequency are given.

In the sequel, we suppose to complement this procedure with a graphical block which will draw the graphs of the frequencies $p_4(\lambda)$, $p_5(\lambda)$ as well as the approximating curves.

3 Preliminary results of experiments

With the program described above a series of numerical experiments was made for instances of various sizes (from $n = 200$ to $n = 3700$). The tolerance t varied from 0.01 to 0.03, the sample size N - from 100 to 500. Several dozens of instances were solved. It is important to stress that, from the qualitative point of view, the situation was in all cases very much the same.

As a typical example, we demostrate now selected results for one problem with the following parameter values: $n = 3700$, $\lambda \in [0.1, 0.5]$, $h_\lambda = 0.01$, $t = 0.03$, $N = 500$. In the table only the objective function values f^{DG}, f^G, f^{LR} and the frequencies $p_4(\lambda)$, $p_5(\lambda)$ are given.

Table 1

$\boldsymbol{\lambda}$	$\boldsymbol{f^{DG}}$	$\boldsymbol{f^{G}}$	$\boldsymbol{f^{LR}}$	$\boldsymbol{p_4(\lambda)}$	$\boldsymbol{p_5(\lambda)}$
0.10	955.630	955.968	955.991	0.056	0.730
0.11	1001.212	1001.583	1001.605	0.044	0.756
0.12	1047.006	1047.363	1047.385	0.040	0.750
0.13	1088.763	1089.113	1089.134	0.034	0.792
0.14	1130.797	1131.152	1131.173	0.046	0.804
0.15	1169.341	1169.690	1169.711	0.048	0.804
0.16	1207.871	1208.212	1208.232	0.050	0.816
0.17	1244.879	1245.226	1245.246	0.050	0.818
0.18	1281.665	1282.009	1282.029	0.046	0.836
0.19	1314.992	1315.335	1315.354	0.034	0.852
0.20	1350.547	1350.882	1350.901	0.052	0.832
0.21	1382.400	1382.713	1382.732	0.046	0.874
0.22	1414.556	1414.839	1414.858	0.056	0.908
0.23	1444.194	1444.492	1444.511	0.060	0.862
0.24	1472.609	1472.893	1472.910	0.050	0.884
0.25	1502.490	1502.772	1502.790	0.044	0.876
0.26	1529.362	1529.604	1529.621	0.044	0.894
0.27	1556.290	1556.536	1556.552	0.062	0.926
0.28	1581.779	1582.012	1582.028	0.074	0.926
0.29	1605.844	1606.072	1606.088	0.062	0.942
0.30	1628.057	1628.265	1628.281	0.086	0.944
0.31	1648.486	1648.701	1648.716	0.062	0.966
0.32	1671.298	1671.484	1671.498	0.096	0.968
0.33	1688.639	1688.811	1688.825	0.082	0.964
0.34	1708.411	1708.568	1708.582	0.098	0.968
0.35	1725.531	1725.686	1725.700	0.118	0.974
0.36	1740.865	1741.011	1741.024	0.090	0.980
0.37	1755.928	1756.056	1756.068	0.128	0.984
0.38	1770.315	1770.442	1770.455	0.124	0.988
0.39	1782.183	1782.293	1782.305	0.124	0.998
0.40	1796.022	1796.128	1796.139	0.166	0.996
0.41	1804.029	1804.122	1804.132	0.170	0.998
0.42	1814.066	1814.147	1814.157	0.204	1.000
0.43	1820.748	1820.818	1820.827	0.196	1.000
0.44	1830.653	1830.716	1830.725	0.216	1.000
0.45	1835.905	1835.952	1835.960	0.272	1.000
0.46	1841.644	1841.682	1841.688	0.414	1.000
0.47	1843.612	1843.641	1843.646	0.496	1.000
0.48	1847.867	1847.884	1847.889	0.718	1.000
0.49	1849.987	1849.996	1850.000	0.940	1.000
0.50	1851.641	1851.643	1851.644	1.000	1.000

From Table 1 some qualitative conclusions can be drawn. First of all, we observe that the values f^{DG}, f^G, f^{LR} are rather close for all λ, and they become closer as λ grows. Recall that we compare the approximate objective function values not with the exact optimum f^* but with its upper bound f^{LR}; therefore we can conclude that the approximate solutions (especially for the primal greedy algorithm) are in average very close to the optimal ones. Moreover, note that our condition $f^* - f^{\mathcal{A}_n} \leq t$ (cf. (11)) is in fact very strong. If we replace our absolute error $f^* - f^{\mathcal{A}_n}$ by the widely used relative error $(f^* - f^{\mathcal{A}_n})/f^*$, we'll see that the actual average behaviuor of greedy algorithms is really excellent with respect to the last criterion.

We can consider the problem (1) with the additional assumption $b = \lambda n$ as a one-parametric integer program. It is well-known (cf., e.g., [1]) that its optimal value f^* (along with the approximate values f^{DG}, f^G) is an increasing function of λ. We see from Table 1 that the growth of objective function values for small values of λ is rather rapid, and this growth is decelerating when λ is approaching the critical value. This empirical fact requires a theoretical explanation.

As for the frequencies of obtaining approximate solutions with a given tolerance t, the behaviour of primal and dual algorithms differs dramatically. We see from Table 1 that $p_4(\lambda)$ (the frequency for the dual algorithm) for small λ is very small (e.g., for $\lambda \in [0.1, 0.34]$ it does not exceed 0.1). An intensive growth of this frequency is observed only for "pre-critical" values of λ (after 0.45). On the contrary, the frequency $p_5(\lambda)$ for the primal algorithm is rather large (say, for $\lambda \geq 0.27$ it exceeds 0.9, and for $\lambda \geq 0.42$ it becomes 1). This sheds additional light on the fact that the dual greedy algorithm is always not better that the primal one (cf. the first inequality in (10)), giving, in a certain sense, some probabilistic characterization of this fact. Of course, this deserves a deeper theoretical explanation.

Of a certain interest is also the behaviour of both algorithms in dependence on the tolerance t. It is clear that the larger t, the better both algorithms must behave. One example: for an instance with $n = 2000$, $N = 100$, $\lambda \in [0.17, 0.50]$, $h_\lambda = 0.01$ we took the tolerance $t = 0.2$ and observed that the frequency $p_5(\lambda)$ for the primal algorithm was 1 for all λ; the frequency $p_4(\lambda)$ for the dual algorithm exceeded 0.3 from the beginning, and became 1 for $\lambda \geq 0.43$.

4 References

[1] Bank B., Mandel R. Parametric integer optimization. Berlin: Akademie-Verlag, 1988.

[2] Diubin G., Korbut A. On the average behaviour of greedy algorithms for the knapsack problems. Preprint 99-14, Humboldt-Universität zu Berlin, Institut für Mathematik, 1999.

[3] Diubin G.N., Korbut A.A. Greedy algorithms for the knapsack problem: The average behaviour (in Russian). Sibirski J. Industr. Math. - 1999.- 2, N 2(4). - pp. 68-93.

[4] Kellerer H., Pferschy U., Pisinger D. Knapsack problems. Springer, 2004.

[5] Korbut, A.A. The characterization of dual greedy algorithms for independence systems (in Russian). In: Ekonomiko-matematicheskie issledovania: matematicheskoe modelirovanie i informazionnie technologii. St.Petersburg: Nauka, 2000, pp. 42-51.

[6] Korte B. Kombinatorische Optimierung und algorithmische Prinzipien. In: W.Krelle, ed. Ökonomische Prognose-, Entscheidungs- und Gleichgewichtsmodelle, VCH Verlagsgesellschaft, Weinheim, 1986, 286-341.

[7] Martello S., Toth P. Knapsack problems. Algorithms and computer implementation. Chichester: Wiley, 1990.

[8] Sigal I.H., Ivanova A.P. Introduction to applied discrete programming (in Russian). Moscow: Fizmatlit, 2002.