Discrete Applied Mathematics 4 (1982) 329–334
North-Holland Publishing Company

NOTE

PROBABILISTIC ANALYSIS OF THE SUBSET-SUM PROBLEM

Gianfranco D'ATRI

Dipartimento di Matematica, Università della Calabria, Arcavacata di Rende, Cosenza, Italy

Claude PUECH

Laboratoire de Recherche en Informatique, Université Paris Sud, Bât. 490, 91405 Orsay, France and Université René Descartes, Sorbonne, 75005 Paris, France

Received 31 March 1981
Revised 28 September 1981

Two linear time algorithms are shown to solve the n-object SUBSET-SUM problem with probability approaching 1 as n gets large, for a uniform instance distribution. Precise evaluations of the probabilities involved are given.

1. The SUBSET-SUM problem

In recent years, many algorithms have been proposed for the knapsack problem (KP); though (KP) is NP-complete all these algorithms have a good performance on test-problems. For example Laurière [3] has obtained practical linear time for problems with up to 60.000 0/1-variables.

To explain such experimental behaviour, one of the authors analyzed in [1] a very simple algorithm under an appropriate probabilistic hypothesis on input data. This assumption - i.e. uniform distribution of independent coefficients - reflects usual ways of generating pseudo-random data for test-problems.

In this paper we consider a special case of (KP): the so-called Subset-Sum (SS) for which we have obtained precise evaluation of probabilities involved rather than asymptotic or approximate expressions.

An instance of SUBSET-SUM consists of a set of n objects, each with some integer weight v_j, and a capacity u. The objective is to choose a subset, S, of objects with maximum total weight not exceeding the capacity. The problem can be formulated as the integer linear program:

$$(\mathrm{SS})\qquad \begin{array}{ll} \mathrm{Max} & \sum v_j x_j, \\ & \sum v_j x_j \le u, \\ & x_j = 0 \text{ or } 1 \quad \text{for } j = 1, \ldots, n. \end{array}$$

It is related to the 0/1-diophantine equation:

$$(\mathrm{E})\qquad \sum v_j x_j = u.$$

Indeed, (E) has a feasible solution if and only if (SS) has an optimal solution of value u.

The best theoretical bound on the number of steps required to solve (SS) is $O(n^2 \cdot c(n))$ [2], where $c(n)$ is an upper bound on the weights; however we show that two *linear time* algorithms give good 'approximate' solutions of (SS).

2. Algorithm GOLOSONE(T)

These algorithms are variants of the following simple modification of the greedy algorithm:

```
GOLOSONE($T$):
    Step 0: $\Pi \leftarrow T(\upsilon_1, \ldots, \upsilon_n)$
    Step 1: If $\sum_1^n \upsilon_j < u$ then ($\bar{x} \leftarrow (1, 1, \ldots, 1)$; Stop)
    Step 2: $S \leftarrow 0$; $i \leftarrow 1$; $\bar{x} \leftarrow (0, 0, \ldots, 0)$
    Step 3: While $S + \upsilon_{\Pi(i)} \leq u$ do ($S \leftarrow S + \upsilon_{\Pi(i)}$; $\bar{x}_{\Pi(i)} \leftarrow 1$; $i \leftarrow i+1$)
    Step 4: $\alpha \leftarrow u - S$
    Step 5: $\bar{\upsilon} \leftarrow 0$
    Step 6: For$j = i+1, \ldots, n$ do (if $\bar{\upsilon} < \upsilon_{\Pi(j)} \leq \alpha$ then ($\kappa \leftarrow \Pi(j)$; $\upsilon \leftarrow \upsilon_{\Pi(j)}$))
    Step 7: If $\bar{\upsilon} \neq 0$ then $\bar{x}_\kappa \leftarrow 1$
    Step 8: $\beta \leftarrow \alpha - \bar{\upsilon}$; Stop
```

Step 0 sorts the objects according to some criterion ($T(\upsilon_1, \ldots, \upsilon_n)$ is some permutation of $1, \ldots, n$ depending on the weights). Step 1 deals with the 'trivial case'. Then GOLOSONE(T), shortly GL, consists of two phases. In phase I (Steps 2, 3) GL sequentially (according to T) places objects in S, until the next object, if placed in S, would cause the total weight of S to exceed u; this leaves a *residual* capacity α. In phase II (Steps 5 to 7) GL selects one object κ from the remaining ones and places it in S: the selected object has the highest weight (among the remaining objects) less or equal to α; this leaves a *gap* β between the total weight of S and the aimed capacity u.

In the sequel we consider two variants of GL:GL1, when $T = T1$, retains the initial ordering of the objects ($\Pi(1) = 1, \ldots, \Pi(n) = n$); GL2, when $T = T2$, sorts the objects in non-increasing weight order ($\upsilon_{\Pi(1)} \geq \upsilon_{\Pi(2)} \geq \cdots \geq \upsilon_{\Pi(n)}$).

GL1 trivially requires linear time.

GL2 requires $O(n \log n)$ steps but can be made linear by the following remark: the rounded (continuous) solution of the continuous knapsack problem

$$\text{"maximize } \sum \upsilon_j^2 \text{ subject to } \sum x_j \upsilon_j \leq u\text{"}$$

is the vector $\bar{x}$ obtained by phase 1 of GL2; so that any linear time algorithm for the knapsack [2] may be used to find $\bar{x}$ and related parameters α, β and κ, without sorting.

3. Probabilistic analysis

In order to analyse the behaviour of GL1 and GL2 we make the following assumptions on the distribution of input data: $v_1, \ldots, v_n$, u are random integer variables such that:

(a) $v_1, \ldots, v_n$ are uniformly distributed over $\{1, \ldots, c(n)\}$;

(b) u is uniformly distributed over $\{1, \ldots, nc(n)\}$;

(c) $v_1, \ldots v_n$ and u are mutually independent.

(The previous model is equivalent to uniform distribution over the set of possible problem instances: there are $nc^{n+1} = c^n(nc)$ different sequences of $n+1$ integers $(v_1, \ldots, v_n, u)$ in the specified range).

All quantities defined by algorithm GL, as α, β or $\bar{x}$, are random variables whose distribution is related to that of weights and capacity.

We derive, first, the distribution of the residual α, and of $\gamma = \Pi(i^\circ)$ with $i^\circ = \min\{i \mid \sum_1^i v_{\Pi(j)} \geq u\}$. If $\alpha \neq 0$, γ is the first object not placed in S during phase 1 of GL; if $\alpha = 0$, it is the last object placed in S during that phase.

Lemma 1. *Let Q_{kp} be the event $\{\alpha = k \cap \gamma = p\}$. Then*

$$\text{Prob}(Q_{kp}) = \frac{1}{nc}\left(1 - \frac{k}{c}\right) \quad \textit{for } k = 0, \ldots, c-1 \textit{ and } p = 1, \ldots, n.$$

(This lemma does not depend on permutation T).

Proof. Let $S_1 = 0$, $S_h = \sum_1^{h-1} v_{\Pi(i)}$ for $h = 2, \ldots, n$ and $q = \Pi^{-1}(p)$.

(a) $\text{Prob}(\alpha = 0 \cap \gamma = p) = \text{Prob}(S_{q+1} = u) = 1/nc$, as S_h and u are independent for all h, and u is uniformly distributed.

(b) For $k \geq 1$,

$$\begin{aligned}
\text{Prob}(\alpha = k \cap \gamma = p) &= \text{Prob}(S_q + k = u \cap v_p > k) \\
&= \sum_{r=1}^{nc} \text{Prob}(S_q + k = r \cap v_p > k \cap u = r) \\
&= \sum_{r=1}^{nc} \text{Prob}(S_q = r - k \cap v_p > k)\, \text{Prob}(u = r) \\
&= \frac{1}{nc} \sum_{r=0}^{nc-k} \text{Prob}(S_q = r \cap v_p > k) \; {}^{1} \\
&= \frac{1}{nc} \text{Prob}(v_p > k),
\end{aligned}$$

because r spans all the possible values of S_q; indeed $S_q \leq (n-1) \cdot c$ for any q. □

[1] If u is not uniformly distributed one may derive from this equality an upperbound for the left-hand side probability.

Let

$$E_{kp}^{h} = \{\nexists j \mid j > p,\, \upsilon_j \in \{k, k-1, \ldots, k-h+1\}\} \quad \text{for GL1},$$
$$E_{kp}^{h} = \{\nexists j \mid j \neq p,\, \upsilon_j \in \{k, k-1, \ldots, k-h+1\}\} \quad \text{for GL2},$$

Lemma 2.

$$\mathrm{Prob}(E_{kp}^{h}) = \left(1 - \frac{h}{c}\right)^{n-p} \quad \textit{for } \text{GL1},$$

$$\mathrm{Prob}(E_{kp}^{h}) = \left(1 - \frac{h}{c}\right)^{n-1} \quad \textit{for } \text{GL2},$$

Proof. Trivial. □

Lemma 3. *Q_{kp} and E_{kp}^{h} are independent*

Proof. In the GL1 case this property is trivial as Q_{kp} and E_{kp}^{h} are defined by disjoint sets of random variables.

In the GL2 case, as E_{kp}^{h} is defined via the random variables υ_j, $j \neq p$, it is $\bigcup_{w \in W} \{(\upsilon_2, \ldots, \upsilon_n) = w\}$ where W is a subset of $\{1, \ldots, c\}^{n-1}$ (we suppose w.l.o.g. that $p = 1$). As

$$Q_{k1} = \bigcup_{h > k} \{\upsilon_1 = h \cap \alpha = k \cap \gamma = 1\},$$

then

$$E_{k1}^{h} \cap Q_{k1} = \bigcup_{W} \bigcup_{h > k} \{(\upsilon_2, \ldots, \upsilon_n) = w \cap \upsilon_1 = h \cap u = f(h, w)\}$$

where $f(h, w)$ is the only value for u such that $\alpha = k$ and $\gamma = 1$ when the coefficients of the problem are (in that order) h, $w_1, \ldots, w_{n-1}$.

Hence

$$\mathrm{Prob}(E_{k1}^{h} \cap Q_{k1}) = \left(\sum_{W} \mathrm{Prob}((\upsilon_2, \ldots, \upsilon_n) = w)\right) \cdot \left(\sum_{h > k} \mathrm{Prob}(\upsilon_1 = h) \cdot \frac{1}{nc}\right)^{1}$$

and, by Lemma 1, $\mathrm{Prob}(E_{k1}^{h} \cap Q_{k1}) = \mathrm{Prob}(E_{k1}^{h}) \cdot \mathrm{Prob}(Q_{k1})$. □

Let $S_{kp}^{h} = \{\nexists j \mid j < \Pi^{-1}(p),\, \upsilon_{\Pi(j)} \in \{k, k-1, \ldots, k-h+1\}\}$.

Lemma 4. $S_{kp}^{h} \cap Q_{kp} = E_{kp}^{h} \cap Q_{kp}$.

Proof. Trivial in the GL1 case as $E_{kp}^{h} = S_{kp}^{h}$.

In the GL2 case, if $\alpha = k$ and $\gamma = p$, then $\upsilon_p > k$, so that $\upsilon_{\Pi(j)} \geq \upsilon_p > k$ for all $j < \Pi^{-1}(p)$ which proves that the elements of $S_{kp}^{h} \cap Q_{kp}$ are elements of $E_{kp}^{h} \cap Q_{kp}$; the converse is trivial. □

Remark that it is also true that $S_{kp}^{h} \cap Q_{kp} = \{\nexists j \mid \upsilon_j \in \{k, k-1, \ldots, k-h+1\}\}$, but the last two events are not independent.

We can now estimate the gap β.

Theorem. *Let*

$$p(c,h)=1+\frac{1}{c}\left(\frac{c-2ch+h^2-h}{c}\right).$$

Then for $h=1,\ldots,c$:

$$\operatorname{Prob}(\beta\geq h)=\begin{cases}\dfrac{c}{2n}\,\dfrac{1-(1-h/c)^n}{h}\cdot p(c,h), & \textit{for}\ \mathrm{GL1},\\[2mm] \tfrac{1}{2}(1-h/c)^{n-1}\cdot p(c,h), & \textit{for}\ \mathrm{GL2}.\end{cases}$$

Proof. Noting that $\beta\leq\alpha$ and decomposing, we obtain:

$$\operatorname{Prob}(\beta\geq h)=\sum_{k=h}^{c}\sum_{p=1}^{n}\operatorname{Prob}(\beta\geq h\cap Q_{kp})=\sum_{k=h}^{c}\sum_{p=1}^{n}\operatorname{Prob}(S_{kp}^{h}\cap Q_{kp}).$$

This is $\sum_{k=h}^{c}\sum_{p=1}^{n}\operatorname{Prob}(E_{kp}^{h})\operatorname{Prob}(Q_{kp})$ by Lemmas 3 and 4, and can be evaluated by Lemmas 1, 2. □

As $\operatorname{Prob}(\beta\neq 0)=\operatorname{Prob}(\beta\geq 1)$:

Corollary 1. *For large c and n, with* $c\ll n$

$$\operatorname{Prob}(\beta\neq 0)\sim\begin{cases}\tfrac{1}{2}c/n & \textit{for}\ \mathrm{GL1},\\ \tfrac{1}{2}\mathrm{e}^{-n/c} & \textit{for}\ \mathrm{GL2}.\end{cases}$$

Remark that, asymptotically $\operatorname{Prob}(\beta\neq 0)\to 0$ when $c(n)=o(n)$, thus:

Corollary 2. *For* $c=o(n)$, Prob(GOLOSONE *solves* SS) → 1 *and* Prob(*Equation* (E) *has a feasible solution*) → 1.

Moreover Prob(GL2 solves SS) approaches 1 much more rapidly than Prob(GL1 solves SS) does.

4. Conclusion

We have obtained the exact distribution of the output parameter of an algorithm for the subset-sum problem. The results rely on the hypothesis of uniform and independent distribution of coefficients and may be generalized to the general knapsack problem under similar probabilistic assumptions, but, in this case, only approximate values of probabilities involved may be given [1]. Furthermore, a similar analysis may be performed under different probabilistic assumptions, namely for non-independent or non-uniform distribution of coefficients; the principle on which the results of Corollary 2 depends (phase 1 of GL ends with so many objects not included in S, relative to the maximum weight, that, with high probability, at least one of these has a weight equal to the gap β) applies to other combinatorial optimization problems. Some of these extensions are announced in [4] and will be analyzed in a forthcoming companion paper.

References

[1] G. d'Atri, Analyse probabiliste du problème du sac-à-dos, Rapport No. 18, Equipe Graphes et Optimisation Combinatoire, Université Paris VI (1979).

[2] E.L. Lawler, Fast approximation algorithms for knapsack problems, Maths. of Operations Research 4(4) (1979) 339–356.

[3] M. Lauriere, An algorithm for the 0/1 knapsack problem, Mathematical Programming 14(1) (1978) 1–10.

[4] G. d'Atri, Probabilistic analysis of knapsack-type problems, Proceedings of the Vth Symposium on Operations research, Köhn, August 1980 (to appear).