

Improved Exponential Time Lower Bound of Knapsack Problem under BT model

Xin Li, Tian Liu, Han Peng, Hongtao Sun*, Jiaqi Zhu
School of EECS, Peking University, P.R.China

August 5, 2013

Abstract

M.Alekhovich et al. recently have proposed a model of algorithms, called BT model, which covers Greedy, Backtrack and Simple Dynamic Programming methods and can be further divided into fixed, adaptive and fully adaptive three kinds, and have proved exponential time lower bounds of exact and approximation algorithms under adaptive BT model for Knapsack problem which are $\Omega(2^{n/2}/\sqrt{n}) = \Omega(2^{0.5n}/\sqrt{n})$ and $\Omega((1/\epsilon)^{1/3.17}) \approx \Omega((1/\epsilon)^{0.315})$ (for approximation ratio $1-\epsilon$) respectively (M. Alekhovich, A. Borodin, J. Buresh-Oppenheim, R. Impagliazzo, A. Magen, and T. Pitassi, Toward a Model for Backtracking and Dynamic Programming, *Proceedings of Twentieth Annual IEEE Conference on Computational Complexity*, pp308-322, 2005). In this note, we slightly improved their lower bounds to $\Omega(2^{(2-\epsilon)n/3}/\sqrt{n}) \approx \Omega(2^{0.66n}/\sqrt{n})$ and $\Omega((1/\epsilon)^{1/2.38}) \approx \Omega((1/\epsilon)^{0.420})$, and proposed as an open question what is the best achievable lower bounds for knapsack under adaptive BT models.

Key Words: Knapsack Problem, Exponential Time Lower Bound, Restricted Models of Algorithms

1 introduction

Many combinatorial optimization problems are NP-complete and probably have no polynomial time algorithms [1]. It is presumed that there are only exponential time algorithms for these problems, however, this has not been

*Corresponding author, sunht_pku@tcs.pku.edu.cn

proved. It is very hard to prove exponential time lower bound for these problems under universal model of algorithms, unless prove $P \neq NP$. Nevertheless, under some restricted model of algorithms, it is possible to prove exponential lower bounds for NP-complete problems. Alekhovich et al have proposed a restricted model of algorithms, called BT model, which covered greedy algorithm, backtrack and simple dynamic programming [2], and have proved exponential lower bound for Knapsack, Vertex Cover and SAT under this model. In this note, we slightly improved their lower bounds of exact and approximation algorithms for Knapsack problem under the adaptive BT model.

2 A Glimpse on BT model

In this section we quickly recall the relevant notions from BT models [2].

2.1 A description of BT model

A combinatorial optimization problem P is represented by a set of data items D and a set of choices H . Each data item represents a partial structure of an instance. The set of choices contains all the possible choices which can be applied to data items. For example, in Knapsack problem, each item can be a data item and "chosen to be in the knapsack" and "chosen not to be in the knapsack" can be the set of choices. In the following, $O(S)$ denotes the set of all the orderings of all the elements in S .

A BT algorithm A to a problem P is comprised of an ordering function r_A^k and a choice function c_A^k :

$$r_A^k : D^k \times H^k \mapsto O(D)$$

is the ordering of unprocessed data items made by algorithm after the first k data items has been processed;

$$c_A^k : D^{k+1} \times H^k \mapsto O(H \cup \{\perp\})$$

represents the constraints made by algorithm when it makes choice for the $k + 1$ th item according to the processed k data items and the choices for them. The algorithm consider only the choices before \perp . If r_A^k is a constant function and does not depend on D^k or H^k , then it's called fixed. If r_A^k depends on D^k but not on H^k , then it's called adaptive. If r_A^k depends on both D^k and H^k , then it's called fully adaptive. In this note, only adaptive BT algorithm is considered.

2.2 Lower bound strategies for BT model

The lower bound strategies for BT model takes the form of a game between the adversary and the solver(the BT algorithm) [3]. In the game, since at the beginning the solver cannot see all the input data items, so the adversary always tries to produce a difficult problem instance for the solver. The game can be viewed as a series of rounds. The i th round is composed of three parts: P_i , PI_i and T_i ($i = 0, 1, 2, \dots$). P_i is the finite set of data items owned by the adversary which cannot be seen by the solver in the i th pattern. PI_i is the set of data items representing a partial instance of the problem which have been seen by the solver in the i th pattern. And T_i is the set of partial solutions to PI_i . See figure 1.

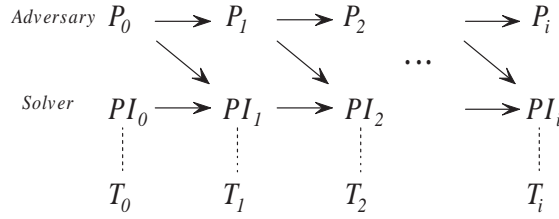


Figure 1: Model of solver-adversary

At the beginning of the game, the solver gives an ordering rule on the input data items. And the adversary constructs rule of deleting data items according to the solver's rule and gives P_0 . In this pattern, PI_0 is empty and T_0 is also empty. In the i th ($i \geq 1$) pattern, solver picks a data item a from P_{i-1} and add it too PI_{i-1} , and gets $PI_i = PI_{i-1} \cup a$. Then computes T_i in which every solution is an extension of some solution in T_{i-1} . Then adversary deletes a from P_{i-1} and some other data items and gets P_i . This process continues until P_i is empty. In the last pattern, if PI_i is not a valid instance or T_i contains the optimum solution of PI_i , then solver wins, or else adversary wins.

If the set of all the solutions to PI_i can be classified to some equivalent classes and for any partial solution PS to any equivalent class, there exists an instance $A \subseteq PI_i \cup P_i$ so that every optimum solution of A contains PS , then PS is called "indispensable". If there exists a pattern in which the number of all the indispensable solutions is exponential with the scale of the problem, then the exponential lower bound of the problem can be achieved under the fixed or adaptive BT model.

3 Improved lower bound of Knapsack

In this section we give the improved exponential lower bounds of exact algorithm and approximation algorithm on Knapsack under adaptive BT model. This improvement is obtained by setting parameters differently in original proofs in the work of M.Alekhovich et. al [2].

Knapsack Problem

Input: n pairs of non-negative integers, $(x_1, p_1), \dots, (x_n, p_n)$ and a positive integer N . x_i represents the weight of the i th item and p_i represents the value of the i th item. N is the volume of the knapsack.

Output:

$$\{i | \max\{\sum_{i \in S} p_i | S \subseteq 1, 2, \dots, n, \sum_{i \in S} x_i \leq N\}\}$$

Simple Knapsack Problem

Input: n non-negative integers $\{x_1, \dots, x_n\}$ and a positive integer N . x_i is the weight and value of the i th item and N is the volume of the knapsack.

Output:

$$\{i | \max\{\sum_{i \in S} x_i | S \subseteq [n], \sum_{i \in S} x_i \leq N\}\}$$

Simple Knapsack problem is also NP complete [1]. So it is only needed to prove the exponential lower bound for simple Knapsack. In the following, we denote a simple Knapsack problem with n items and knapsack volume of N with (n, N) .

3.1 Lower Bound of Exact Algorithm

M.Alekhovich et. al proved the following theorem in [2].

Theorem 1. For simple Knapsack problem (n, N) , the time complexity of any adaptive BT algorithm is at least

$$\binom{n/2}{n/4} = \Omega(2^{0.5n}/\sqrt{n}).$$

Next we prove the following theorem.

Theorem 2. For simple Knapsack problem (n, N) , $\forall 0 < \epsilon \leq 1/2, \exists N_0, s.t.$ when $N > N_0$, the time complexity of any adaptive BT algorithm is at least

$$\binom{(2-\epsilon)n/3}{(2-\epsilon)n/6} = \Omega(2^{(2-\epsilon)n/3}/\sqrt{n}) \approx \Omega(2^{0.66n}/\sqrt{n}).$$

Proof: Let $I = \{0, 1, \dots, \frac{3N}{n}\}$, all the weights of the items are from I .

This is an constructive proof containing three steps.

Step 1.

Solver chooses the first $(2 - \epsilon)n/3$ items. After each, adversary deletes some items from the remaining by the following two rules. Let S be the set of the items which have been seen by the solver.

- (1) If $S_1, S_2 \subseteq S$, then delete items with weight of $|\sum S_1 - \sum S_2|$;
- (2) If $S_1 \subseteq S$, then delete items with weight of $|N - \sum S_1|$;

Step 2.

Let P be the set of the first $(2 - \epsilon)n/3$ items chosen by solver. Now we prove: $\forall Q \subset P$ and $|Q| = (2 - \epsilon)n/6, \exists R \subset I - P, |R| = (1 + \epsilon)n/3$ and $\sum_{x \in Q} x + \sum_{x \in R} x = N$. See figure 2:

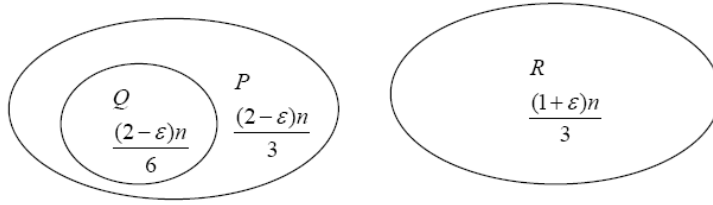


Figure 2: Constructing an indispensable partial solution

$\forall Q \subset P$, let $U = 3^n, a = \frac{3}{(1+\epsilon)n}(N - \sum_{x \in Q} x), J = \{i | i \in \mathbb{Z} \wedge i \in [a - U, a + U]\}$ (later we will prove $J \subset I$). Adversary chooses $\frac{(1+\epsilon)n}{3} - 2$ items in J . After each, adversary will delete some items from the remaining by the following two rules. Let S be the set of items chosen by solver and adversary.

- (1) If $S_1, S_2 \subseteq S$, then delete items with weight of $|\sum S_1 - \sum S_2|$;
- (2) If $S_1 \subseteq S$, then delete items with weight of $|N - \sum S_1|$;

Let the sum of the weights of these $\frac{(1+\epsilon)n}{3} - 2$ items be w . Then the adversary can choose items with weight bigger or smaller than a so that

$$|w - a(n/2 - 2)| \leq U.$$

Step 3.

Adversary choose a pair of items from the following $U + 1$ pairs (later we will prove they are in I), such that the weight of these two items and the weight of the previously chosen items sum to N :

$$(\frac{v}{2} - i, \frac{v}{2} + i), i = 1, 2, \dots, U + 1$$

in which $v = N - \sum Q - w$.

Since in the process of choosing items, the number of all the items does not exceed n , let this number be n_t . Order these n_t items by a fixed order, then every item corresponds to a single bit of a n_t -bit 0-1-(-1) string. And every such string corresponds to a deleted item in the previous process. In detail, let the sum of the weights of items corresponding to 1 in the string be s_1 , and let the sum of the weights of items corresponding to -1 in the string be s_{-1} . If $s_1 - s_{-1} > 0$, then delete item with weight of $s_1 - s_{-1} > 0$; if $s_1 = 0$, then delete item with weight of $N - s_{-1} > 0$. So the number of the deleted items does not exceed U . So in these $U + 1$ pairs of items, there must be one pair which is not deleted. Then we can choose this pair in the third step.

Next we prove all the weights of the chosen items are in I .

Because

$$a = \frac{3}{(1 + \epsilon)n}(N - \sum Q)$$

and

$$0 \leq \sum Q \leq \frac{(2 - \epsilon)n}{6} \cdot \frac{3N}{n} = \frac{(2 - \epsilon)3N}{6},$$

so

$$\frac{3\epsilon N}{2(1 + \epsilon)n} \leq a \leq \frac{3N}{(1 + \epsilon)n},$$

clearly,

$$0 \leq \frac{3\epsilon N}{2(1 + \epsilon)n} \leq a \leq \frac{3N}{(1 + \epsilon)n} \leq \frac{3N}{n}.$$

Since

$$|w - a(n/2 - 2)| \leq U,$$

$$a = \frac{3}{(1 + \epsilon)n}(N - \sum Q)$$

and

$$v = N - \sum Q - w,$$

so

$$|v - 2a| \leq U,$$

that is

$$a - \frac{U}{2} \leq \frac{v}{2} \leq a + \frac{U}{2}.$$

So in the third step, the possible lightest and heaviest items are $a - \frac{U}{2} - U - 1$ and $a + \frac{U}{2} + U + 1$. So in order that all the items chosen in the third step are from I , it is only needed that

$$\begin{cases} a - \frac{U}{2} - U - 1 \geq 0 \\ a + \frac{U}{2} + U + 1 \leq \frac{3N}{n} \end{cases},$$

that is

$$\begin{cases} N \geq \frac{3(1+\epsilon)n3^n+2(1+\epsilon)n}{3\epsilon} \\ N \geq \frac{3(1+\epsilon)n3^n+2(1+\epsilon)n}{6\epsilon} \end{cases},$$

So it is only need that $N \geq \lceil \frac{1+\epsilon}{\epsilon} \rceil n3^n$. In fact, $N \geq \lceil \frac{1+\epsilon}{\epsilon} \rceil n3^n$ is also sufficient to $J \subset I$.

Next we prove that solver must keep all Q in P as partial solution in the computation tree, or else adversary can let the solver cannot find the optimum solution.

We use the reduction to absurdity. Let $Q_1, Q_2 \subset P$ and $|Q_1| = |Q_2| = (2 - \epsilon)n/6$. R_1 and R_2 correspond to Q_1 and Q_2 respectively. If solver does not keep Q_1 , then adversary deletes all the items except R_1 . If solver keeps Q_2 to get the optimum solution, then there are only three cases(see figure 3):

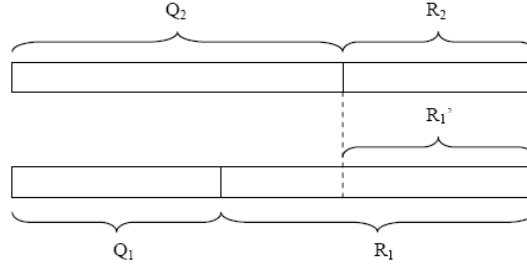


Figure 3: Proof of indispensable of partial solution of Knapsack

1) $\sum Q_2 + \sum R_1 < N$

The optimum solution N cannot be achieved in this case.

2) $\sum Q_2 + \sum R_1 = N$

So $\sum Q_1 = \sum Q_2$. This is impossible. Since according to the first rule in the first step, the last chosen item in the first step which is contained in Q_1 or Q_2 but not both cannot be chosen in the first step.

3) $\sum Q_2 + \sum R_1 > N$, that is $\sum Q_2 > \sum Q_1$

Assume R_1' is a subset of R_1 and it leads to the optimum solution with Q_2 . Then there are two cases:

3.1) there is only one item in $R_1 - R_1'$

According to the first rule in the first step, this item should be deleted.

3.2) there are two or more than two items in $R_1 - R_1'$

If there is only one item in $R_1 - R'_1$ which is chosen in the third step, then according to the first rule in the second step, it should be deleted.

If there are two items which are chosen in the third step, then all the items in R'_1 are chosen in the second step. Let the choosing order of these items be i_1, \dots, i_s , then according to the second rule in the second step, the last item i_s should be deleted.

So every Q in P is indispensable. So the time complexity is at least

$$\binom{(2-\epsilon)n/3}{(2-\epsilon)n/6} = \Omega(2^{(2-\epsilon)n/3}/\sqrt{n}) \approx \Omega(2^{0.66n}/\sqrt{n}).$$

■

3.2 Lower bound of approximation algorithm

M.Alekhovich et. al proved the following theorem in [2].

Theorem 3. For simple Knapsack problem, using adaptive BT algorithm, to achieve $1-\epsilon$ approximation ratio, the time complexity is at least $\Omega((1/\epsilon)^{1/3.17})$.

Next we prove the following theorem:

Theorem 4. For simple Knapsack problem, using adaptive BT algorithm, to achieve $1-\epsilon$ approximation ratio, the time complexity is at least $\Omega((1/\epsilon)^{1/2.38})$.

Proof: In theorem 2 we proved that for given $n, N = \lceil \frac{1+\delta}{\delta} \rceil n 3^n, 0 < \delta < 0.5$, the time complexity of simple Knapsack problem is $\Omega(2^{\frac{(2-\delta)}{3}n}/\sqrt{n})$. So the optimum solution cannot be achieved by algorithm with complexity of $\gamma = o(2^{\frac{(2-\delta)}{3}n}/\sqrt{n})$. Since the weight of item is integer, so the upper bound of approximation ratio is

$$\frac{N-1}{N} \sim 1 - \frac{1}{\lceil \frac{1+\delta}{\delta} \rceil n 3^n} \sim 1 - \tilde{O}(\gamma^{-\frac{3}{2-\delta} \log_2 3}) \sim 1 - \epsilon,$$

so

$$\gamma = \Omega((1/\epsilon)^{\frac{2-\delta}{4.75}}).$$

For $n, N, 0 < \delta < 0.5 (N > \lceil \frac{1+\delta}{\delta} \rceil n 3^n)$, let $N = \lceil \frac{1+\delta}{\delta} \rceil n_0 3^{n_0}$. So we can set the weights of $n - n_0$ items in the n items to 0.

■

4 Discussion

In this paper, we slightly improved the exponential time lower bounds of exact and approximation algorithms under adaptive BT model for Knapsack

problem which are $\Omega(2^{(2-\epsilon)n/3}/\sqrt{n}) \approx \Omega(2^{0.66n}/\sqrt{n})$ and $\Omega((1/\epsilon)^{1/2.38}) \approx \Omega((1/\epsilon)^{0.420})$.

We do not know whether our lower bounds are optimal. An interesting question is: what are the best achievable lower bounds for Knapsack problem under adaptive BT model?

References

- [1] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., 1979.
- [2] M. Alekhnovich, A. Borodin, J. Buresh-Oppenheim, R. Impagliazzo, A. Magen, and T. Pitassi. Toward a model for backtracking and dynamic programming. *Proceedings of Twentieth Annual IEEE Conference on Computational Complexity*, pages 308–322, 2005.
- [3] P. Pudlak. Proofs as games. *American Math. Monthly*, June-July 2000.