ε-Optimization Schemes and L-Bit Precision:

Alternative Perspectives in Combinatorial Optimization

James B. Orlin Andreas S. Schulz Sudipta Sengupta

Massachusetts Institute of Technology

OUTLINE

- Input with L-Bit Precision.
  - A strongly polynomial-time algorithm for the KNAPSACK PROBLEM.
  - A strongly polynomial-time algorithm for the 3-PARTITION PROBLEM.
  - The GROUP KNAPSACK PROBLEM remains NP-complete.

- Output within ε of Optimum.
  - A different notion of approximation: ε-optimality.
  - A fully polynomial-time ε-optimization scheme for a problem without (F)PTAS.

- Summary — Dealing with Imprecise Data.
  - Inverse optimization.
  - ε-optimization.
  - L-bit precision.

The L-Bit Precision Model

All numbers are of the form $a \cdot 2^t$, with $a < 2^L$.

Needs space $O(L + \log t)$.

We assume either that

- L is constant (fixed precision), or
- $L = O(\log n)$ (logarithmic precision).

Polynomial time: polynomial in n and $\log T$.

Strongly polynomial time: polynomial in n.

A First Example — The Knapsack Problem

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} c_i x_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} a_i x_i \leqslant b \\ & x \in \{0,1\}^n \end{aligned}$$

Theorem. The Knapsack Problem can be solved in strongly polynomial time under the fixed and logarithmic precision models.

A Pseudo-polynomial Time Algorithm for Knapsack

Nodes: $\langle 0,0\rangle$ and $\langle k,w\rangle$ for each $k \in \{1,2,\ldots,n\}$ and $w \in \{1,2,\ldots,nC\}$.

Arcs: From $\langle k-1,w\rangle$ to $\langle k,w\rangle$ with length 0, and from $\langle k-1,w\rangle$ to $\langle k,w+c_k\rangle$ with length a_k.

Let $s_k = \max\{w : \text{ there is a path from } \langle 0,0\rangle \text{ to } \langle k,w\rangle \text{ with length } \leqslant b\}$.

Observation. The maximum value for the knapsack problem is s_n, and a solution can be determined in $O(n^2C)$ time.

A Polynomial-Time Algorithm under L-Bit Precision

Assume $c_j = d_j \cdot 2^{t_j}$, where $d_j < 2^L$. Assume $t_1 \geqslant t_2 \geqslant \cdots \geqslant t_n$.

Let $C_i = \sum_{j=i}^{n} c_j$.

Let G^* denote the subgraph of G with node set

$$\{\langle k, w\rangle : 1 \leq k \leq n \text{ and } s_k - C_k \leq w \leq s_k\} \cup \{\langle 0, 0\rangle\}$$

.

Observation 1. The shortest path from node $\langle 0, 0\rangle$ to node $\langle k, w\rangle$ in G^* is the shortest path from node $\langle 0, 0\rangle$ to node $\langle k, w\rangle$ in G.

Observation 2. If there is a path from node $\langle 0, 0\rangle$ to node $\langle k, w\rangle$, then w is an integral multiple of 2^{t_k}.

COUNTING REACHABLE NODES IN G^*

The number of nodes $\langle k, w \rangle$ for fixed value of k is at most C_k, which is at most $n\,2^L\,2^{t_k}$.

The number of such nodes for which w is an integral multiple of 2^{t_k} is at most $n\,2^L$.

If we consider nodes $\langle k, w \rangle$ for all values of $k \in \{1, 2, \ldots, n\}$, the number of nodes in G^* reachable from node $\langle 0, 0 \rangle$ is at most $n^2 2^L$.

The knapsack problem can be solved in $O(n^2 2^L)$ time when the input values c_j are expressed with L-bit precision. This is strongly polynomial time when L is fixed or logarithmic.

THE GROUP KNAPSACK PROBLEM

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} c_i x_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} a_i x_i \leqslant b \\ & x_i = x_j \text{ for } \{i,j\} \in E \\ & x \in \{0,1\}^n \end{aligned}$$

Theorem. The 1-bit version of the GROUP KNAPSACK PROBLEM is NP-complete.

Proof.
The n-bit version of the KNAPSACK PROBLEM is known to be NP-complete.

Transform any n-bit version of the KNAPSACK PROBLEM into a 1-bit version of the GROUP KNAPSACK PROBLEM by replacing each element j of the original knapsack problem by a group of at most n elements. □

A Second Example — The 3-Partition Problem

Given $3n$ non-negative integers $a_1, a_2, \ldots, a_{3n}$ such that $\sum_{i=1}^{3n} = nb$, is there a partition of the integers into n groups $S_1, S_2, \ldots, S_n$ of three elements each, such that $\sum_{a_i \in S_j} a_i = b$ for all j?

- The 3-Partition Problem is NP-complete under logarithmic precision.
- It can be solved in strongly polynomial time in the fixed precision model.

Lemma. There are at most $(L+2)^2\,2^{2L}$ triples of L-bit precision integers which sum to b.

Proof. Let $z_i = x_i 2^{y_i}$ for $1 \le i \le 3$ with $y_1 \ge y_2 \ge y_3$ and $z_1 + z_2 + z_3 = b$. Let $2^T \le b < 2^{T+1}$. Clearly, $y_1 \le T$. We also have

$$\begin{aligned}
x_1 2^{y_1} &\ge b/3 \\
\Rightarrow x_1 2^{y_1} &> 2^{T-2} \text{ since } b \ge 2^T \\
\Rightarrow 2^{y_1} &> 2^{T-2}/x_1 \\
\Rightarrow 2^{y_1} &> 2^{T-2}/2^L \text{ since } x_1 < 2^L \\
\Rightarrow y_1 &> T-2-L
\end{aligned}$$

Thus, $T-1-L \le y_1 \le T$, and the total possible values for $z_1 = x_1 2^{y_1}$ is $(L+2)2^L$. □

AN INTEGER PROGRAM IN FIXED DIMENSION

$$\sum_{j \in S_i} \alpha_{ij} x_j = n_i \qquad \text{for } 1 \leqslant i \leqslant m$$

$$x_j \geqslant 0 \qquad \text{for all } j$$

$$x_j \text{ integer} \qquad \text{for all } j$$

x_j = number of times triple j occurs.

S_i = set of triples containing integer a'_i.

n_i = number of occurrences of a'_i.

α_{ij} = number of times that a'_i appears in triple j.

When the input integers are expressed with L-bit precision, for fixed L, the 3-PARTITION PROBLEM can be solved in strongly polynomial time.

FURTHER FACTS AND FIRST SUMMARY

- The KNAPSACK PROBLEM can be solved in strongly polynomial time under the fixed and the logarithmic precision model. A simple variant is NP-hard with one bit precision.
- 3-PARTITION is NP-complete under logarithmic precision, but it enjoys a strongly polynomial-time algorithm under the fixed precision model.
- IDENTICAL PARALLEL MACHINE SCHEDULING can be solved in strongly polynomial time when m is fixed and L is fixed or logarithmic. It is NP-hard under logarithmic precision when m is part of the input.

TRANSITION

We are given an integer objective function c. If we round c and express it using L bits of accuracy and solve the resulting problem, then the returned solution is $(1/2^{L-1})$-optimal.

ε-OPTIMIZATION

c' is an ε-perturbation of c if

$$\begin{aligned} c_j(1-\varepsilon) \leqslant c'_j \leqslant c_j(1+\varepsilon) \quad &, \text{if } c_j > 0, \\ -c_j(1-\varepsilon) \leqslant -c'_j \leqslant -c_j(1+\varepsilon) \quad &, \text{if } c_j < 0. \end{aligned}$$

A solution x' to $\min\{cx : x \in X\}$ is ε-optimal if there exists an ε-perturbation c' of c such that x' is optimal for $\min\{c'x : x \in X\}$.

An ε-optimization algorithm is an algorithm that produces an ε-optimal solution for every instance.

We will particularly be interested in (fully) polynomial-time ε-optimization schemes.

Approximation Algorithms

An ε-approximation algorithm for $\min\{cx : x \in X\}$ gives in polynomial time for any instance a solution $\bar{x}$ with

$$\frac{c\bar{x} - cx}{cx} \leqslant \varepsilon \qquad \text{for all } x \in X \quad .$$

Some drawbacks:

- If the optimal objective function value cx^* is non-positive, the relative error is an inappropriate measure of performance.
- A translation of variables has a dramatic impact; replacing x by $x - a$ will lead to very different results.

TWO GOALS

1. Investigate the relationship between ε-optimization schemes and preexisting approximation schemes.

2. Design ε-optimization schemes for specific combinatorial problems.

FPTAS vs. FPTEOS

Observation. If all cost coefficients and all feasible solutions are non-negative, and if x is ε-optimal, then x has worst-case relative error at most $2\varepsilon/(1-\varepsilon)$.

Theorem. Whenever a $0/1$-optimization problem $\min\{cx : x \in X\}$ has an FPTAS which can handle the fixing of variables, then this problem also has an FPTEOS.

FPTAS $\longrightarrow$ FPTEOS

Initialization:

Let $q := \varepsilon/2n$.
Let $I := J := \emptyset$; $K := \{1, 2, \ldots, n\}$.
Let x^* be any feasible solution.

Main Loop:

Find a q-approximate completion x' of $\langle I, J \rangle$.
if $cx' < cx^*$, **then** $x^* := x'$.
let $U := \max\{c_k : k \in K\}$.
for all indices $k \in K$ with $c_k > U/2$, **do**
  if $x^*_k = 1$, **then** $J := J \cup k$, and $K := K - k$;
  if $x^*_k = 0$, **then** $I := I \cup k$, and $K := K - k$;
if $K = \emptyset$, **then** stop; **else return to** Main Loop.

SKETCH OF PROOF

Let x be any feasible solution, $x \neq x^*$.

Among all indices k such that $x^*_k \neq x_k$, let r be the one for which c_r is maximum.

W.l.o.g., we may assume that $x^*_r = 1$ and $x_r = 0$.

Let $\langle I, J \rangle$ be the partial solution at the beginning of the main loop prior to x^*_r being fixed.

Let $K = \{1, \ldots, n\} \setminus (I \cup J)$.

Note that x^* is a q-approximate completion of $\langle I, J \rangle$. Hence,

$$\sum_{j \in K} c_j (x^*_j - x_j) \leqslant q \sum_{j \in K} c_j \leqslant q |K| 2 c_r \leqslant \varepsilon c_r \quad .$$

PROOF (CONTD.)

Moreover, we have $(c_j^* - c_j)(x_j^* - x_j) \leqslant 0$ for each $j \in K$,

and $(c_r^* - c_r)(x_r^* - x_r) - \varepsilon c_r$.

Hence,

$$c^*(x^* - x) \quad \sum_{j \in K} c_j^*(x_j^* - x_j) \quad \sum_{j \in K} c_j(x_j^* - x_j) + \sum_{j \in K} (c_j^* - c_j)(x_j^* - x_j) \leqslant 0 \ .$$

A First FPTEOS for a Problem without FPTAS

Single Machine Scheduling with Rejection

Every job has three parameters: outsourcing cost e_j, processing time p_j, and due date d_j.

We are interested in minimizing

$$\sum_{j \in R} e_j + \max_{j \in S} L_j$$

where $L_j = C_j - d_j$.

Note that the objective can be negative.

Theorem. There is an $O(n^3/\varepsilon)$ time algorithm which gives an optimal solution w.r.t. processing times p_j, due dates d_j, and ε-perturbed rejection costs e'_j.

FURTHER FACTS AND SECOND SUMMARY

- Notion of ε-optimality can be extended to the perturbation of other parameters. BIN PACKING and IDENTICAL PARALLEL MACHINE SCHEDULING have a PTEOS.
- There is a close linkage between pseudopolynomial-time algorithms, FPTAS's, and FPTEOS's.

Concluding Remarks (on Imprecise Data Optimization)

- Inverse Optimization is an alternative approach to measure distance from optimality.
- ε-Optimization is an alternative approach to produce near-optimal solutions.
- *L*-Bit Precision is an alternative way to measure complexity.