Integer Linear Programming

NP-Completeness

Christos Litsas

January 26, 2012

Outline

The Problem

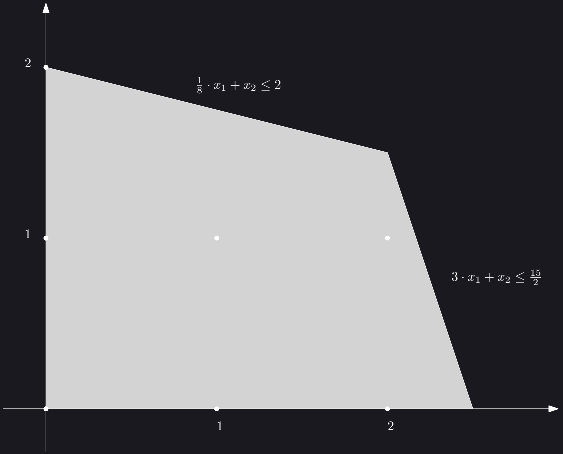

Figure: A, linear, bounded set.

- Maximize some function $f(x_1, x_2)$
- Subject to $\frac{1}{8} \cdot x_1 + x_2 \leq 2$, $3 \cdot x_1 + x_2 \leq \frac{15}{2}$, $x_1, x_2 \in \mathbb{N}$

The Problem

Primal Problem

- Maximize $c' \cdot x$
- Subject to $Ax \leq 0$, $x \in \mathbb{N}^n$

Questions

- How hard is ILP (any connection with general LP)?
- Are there efficient algorithms that solve at least some special instances of the problem?

The Problem

Integer Programming

Decide whether, for given $m \times n$ integer matrix A and m-vector b, the conditions

$$Ax = b, x \geq O,$$

are satisfied by some $x \in \mathbb{N}$

Connection with Other Famous NP-Complete Problems

Set Cover

- $A \cdot x \geq ones(n)$ (A: has rows the bit vectors of the sets)
- $\sum x_i \leq b$ (b: the budget)
- $x_i \in \{0, 1\}$

Knapsack

- $\sum a_i \cdot x_i = B$ (B: the budget)
- $x_i \in \mathbb{N}$

NP-Complete Instances

Reduction

From 3 SAT:

- ϕ : *a formula in CNF.*
- *introduce a variable* x_i *for every atomic variable* y_i.
- x_i *can only be zero (false) or one (true).*
- *replace* $\wedge$ *by* $\times$, $\vee$ *by* $+$, *literals* y_i *by* x_i *and* $\neg y_i$ *by* $1 - x_i$.
- *Let* Φ : *be the arithmetic formula obtained*
- *The ILP is* $\Phi \geq 1$ *subject to* x_i *in*$\{0, 1\}$

NP-Complete Instances

Corollary

0-1 ILP is NP-Hard.

NP-Complete Instances

Corollary

0-1 ILP is NP-Hard.

Corollary

0-1 ILP is NP-Complete (since it is NP-Hard but we can check a solution in Poly Time).

Some maths...

Theorem

Let A be an $m \times n$ integer matrix and b an m-vector, both with entrtes from $\{0, \pm 1, ..., \pm a\}$. Then if $Ax = b$ has a solution $x \in \mathbb{N}^n$, it also has one in $\{0, 1, ..., n(ma)^{2m+1}\}^n$.

Some maths...

Theorem

Let A be an $m \times n$ integer matrix and b an m-vector, both with entrtes from $\{0, \pm 1, ..., \pm a\}$. Then if $Ax = b$ has a solution $x \in \mathbb{N}^n$, it also has one in $\{0, 1, ..., n(ma)^{2m+1}\}^n$.

Corollary

There is a pseudopolynomial algorithm for solving $m \times n$ integer programs, with fixed m.

Dynamic Programming Algorithm

Algorithm

Solve the $m \times n$ integer program $Ax = b$ by dynamic programming, proceeding in stages.
j-th stage: compute the set S_j of all vectors v that can be written as $\sum_{i=1}^{j} v_i x_i$, with v_i the i-th column of A and with the x_i in the range $0 \leq x_i \leq B$, where $B = n(ma)^{2m+1}$.

Dynamic Programming Algorithm

Algorithm

Solve the $m \times n$ *integer program* $Ax = b$ *by dynamic programming, proceeding in stages.*
j*-th stage: compute the set* S_j *of all vectors* v *that can be written as* $\sum_{i=1}^{j} v_i x_i$*, with* v_i *the* i*-th column of* A *and with the* x_i *in the range* $0 \leq x_i \leq B$*, where* $B = n(ma)^{2m+1}$*.*

Theorem

Since the S_j *cannot become larger than* $(nB)^m$*, the whole algorithm can be carried out in time* $O((nB)^{m+1}) = O(n^{2m+2}(ma)^{(m+1)(2m+1)})$*, a polynomial in* n *and* a *if* m *is fixed.*

More maths...

Theorem

Consider the following linear programming relaxation:

- *Maximize* $c' \cdot x$
- *Subject to* $Ax \leq 0$, $x \in {\mathbb{R}^{+}}^{n}$

If the original ILP is feasible and the above problem is unbounded, then original ILP is also unbounded.

More maths...

Theorem

Consider the following linear programming relaxation:

- *Maximize* $c' \cdot x$
- *Subject to* $Ax \leq 0$, $x \in {\mathbb{R}^+}^n$

If the original ILP is feasible and the above problem is unbounded, then original ILP is also unbounded.

Theorem

Suppose that the original ILP is feasible and bounded, and let z *be its optimal cost. Then* $|z| \leq (\sum_{j=1}^{n} |c_j|) \cdot M$, *where* $M = n^2(ma^2)^{2m+3}$.

Pseudopolynomial Algorithm

Theorem

There is a pseudopolynomial algorithm for finding the optimum in any $m \times n$ optimization integer program, for m fixed.

Pseudopolynomial Algorithm

Theorem

There is a pseudopolynomial algorithm for finding the optimum in any $m \times n$ optimization integer program, for m fixed.

Algorithm

Solve the problem:

- *Maximize $c' \cdot x = z$*
- *Subject to $Ax \leq 0$, $x \in \mathbb{N}^n$*

for each value of z in the range $[-(\sum_{j=1}^{n} |c_j|) \cdot M - 1, (\sum_{j=1}^{n} |c_j|) \cdot M]$ using the pseudopolynomial algorithm described above.
Binary search would yield a better bound.