OPTIMIZATION ALGORITHMS

UNIVERSITÀ DEGLI STUDI DI BRESCIA

Combinatorial Optimization

Prof. **Renata Mansini**

Research Group MAO@DII

Models and Algorithms for Optimization at Department of Information Engineering

Sito web: `https://or-dii.unibs.it/`

Optimization problem

Find a solution $\mathbf{x}$ such that

$$
\begin{aligned}
& min \quad f(\mathbf{x}) \\
\text{s. to} \quad & g_i(\mathbf{x}) \geq 0 \qquad i = 1, ..., m \\
& h_j(\mathbf{x}) = 0 \qquad j = 1, ..., p \\
& \mathbf{x} \geq 0
\end{aligned}
$$

where f, g_i and h_j are generic functions in $\mathbf{x} \in \mathbb{R}^n_+$.

Mixed Integer Linear Programming Problem (MILP)

A systematic way to study discrete optimization problems is to express them as integer programming problems:

$$
\begin{gathered}
min \quad \mathbf{c}^T\mathbf{x} + \mathbf{d}^T\mathbf{y} \\
A\mathbf{x} + B\mathbf{y} \geq \mathbf{b} \\
\mathbf{x}, \mathbf{y} \geq 0 \\
\mathbf{x} \text{ integer}
\end{gathered}
$$

Entries of **A** and **B**, **c**, **d**, and **b** are integers.

If **x** is restricted to be integer: $\rightarrow$ **Integer Linear Programming** (ILP).

If **x** is restricted to be binary: $\rightarrow$ **Binary Integer Programming** (BIP).

Combinatorial Optimization Problem

A COMBINATORIAL OPTIMIZATION PROBLEM (COP) is:

- a problem that looks for an optimal object by selecting it from a finite set of discrete objects (sets, paths, graphs, cyclic permutations, sequence of integer numbers)

DEFINITION (MINIMUM PROBLEM)

Let $N = \{1, \ldots, n\}$ be a finite set and $\mathcal{F}$ a collection of feasible subsets of N. A weight (cost) c_j is associated with each $j \in N$. We define a *COP* as the problem to determine the subset S with minimum weight (cost):

$$(COP) \qquad \min_{S \subseteq N} \left\{ \sum_{j \in S} c_j : S \in \mathcal{F} \right\}$$

A COP can be formulated as a BIP, ILP, MILP.

Some examples of COP (1/3)

EXAMPLE 1: 0-1 Knapsack Problem

Given:

- a set $N = \{1, ..., n\}$ of items, each item $i \in N$ has a weight w_i and a value v_i;
- a knapsack with capacity b.

Problem

Find the subset $I \subseteq N$ such that:

$$\max_{I \subseteq N} \left\{ \sum_{i \in I} v_i : \quad \sum_{i \in I} w_i \leq b \right\}.$$

Solution: a set $I \subseteq \{1, \cdots, n\}$.

Some examples of COP (2/3)

Example 2: Set Covering Problem

Given:

- a set of regions $M = \{1, ..., m\}$;
- a set of potential facilities (emergency centers) $N = \{1, ..., n\}$;
- a collection $S_1, S_2, ..., S_n$ of subsets of M, where $S_j \subseteq M$ represents the subset of regions covered by facility j, $j \in N$;
- an installation cost c_j for each facility $j \in N$.

Problem

Find the collection T of subsets of M such that:

$$\min_{T \subseteq N} \left\{ \sum_{j \in T} c_j : \quad \bigcup_{j \in T} S_j = M \right\}.$$

Solution: a collection T.

Some examples of COP (3/3)

Example 3: Traveling Salesman Problem (TSP)

Given:

- a set of customers (cities) $N = \{1, ..., n\}$ to visit;
- a traveling cost (time) c_{ij} for going from i to j.

Problem

Find the cyclic permutation π such that:

$$\min_{\pi} \left\{ \sum_{j=1}^{n} c_{j\pi(j)} : \pi \in T \right\}$$

where $T = \{$set of all cyclic permutations π of n items$\}$.

Solution: a cyclic permutation $\{\pi_1, \cdots, \pi_n\}$.

Guidelines for strong formulations

Linear Programming $\Rightarrow$ a good formulation has small number n, m of variables and constraints $\Rightarrow$ complexity is polynomial in n e m.

Integer Linear Programming $\Rightarrow$ choice of the formulation is crucial.

We should learn how to reply to the following questions:

- How could we compare different ILP formulations of the same problem?
- Which formulation is the strongest one?
- What does it mean that a formulation is stronger than another one?

Comparison between two different formulations

0-1 Plant Location Problem (Formulation A)

$$min \quad \sum_{j=1}^{n} k_j y_j + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1, ..., m$$

$$\sum_{i=1}^{m} x_{ij} \leq m \cdot y_j \qquad j = 1, ..., n \quad (*)$$

$$x_{ij}, y_j \in \{0, 1\} \quad i = 1, ..., m; j = 1, ..., n$$

$x_{ij} = 1$ if plant j serves customer i, 0 otherwise.

$y_j = 1$ if plant j is opened, 0 otherwise.

0-1 Plant Location Problem (Formulation B)

Same formulation as A, but instead of constraints (*) we use:

$$x_{ij} \leq y_j \quad i = 1, \ldots, m; j = 1, \ldots, n$$

We can observe that:

1. The two formulations are equivalent: they provide the same integer optimal solution.
2. The feasible regions of their continuous relaxations (convex polyhedrons) are different.
3. We use the linear programming relaxations to compare the two formulations.

Let us consider continuous variables for formulations A and B:

$$x_{ij}, y_j \in \{0, 1\} \rightarrow 0 \leq x_{ij} \leq 1, 0 \leq y_j \leq 1 \quad \forall i, j$$

The following two convex polyhedrons define the continuous relaxations of formulations A and B:

$$P_{PLA} = \{(x, y) : \textstyle\sum_{j=1}^{n} x_{ij} = 1, \forall i; \quad \sum_{i=1}^{m} x_{ij} \leq m y_j \;\; \forall j,$$
$$0 \leq x_{ij} \leq 1 \quad \forall i, j; \quad 0 \leq y_j \leq 1 \quad\; \forall j\}$$

$$P_{PLB} = \{(x, y) : \textstyle\sum_{j=1}^{n} x_{ij} = 1, \forall i; \quad x_{ij} \leq y_j \;\; \forall i, j,$$
$$0 \leq x_{ij} \leq 1 \quad \forall i, j; \quad 0 \leq y_j \leq 1 \quad\; \forall j\}$$

The polyhedrons P_{PLA} and P_{PLB} contain the same set of integer solutions, but $P_{PLB} \subset P_{PLA}$.

Let Z^* be the optimal solution value of the integer programming problem, and let Z_{PLA} and Z_{PLB} be the optimal solution values of the continuous relaxations:

$$\text{since} \quad P_{PLB} \subset P_{PLA} \quad \rightarrow \quad Z_{PLA} \leq Z_{PLB},$$
$$\text{moreover} \quad Z_{PLB} \leq Z^* \quad \rightarrow \quad Z_{PLA} \leq Z_{PLB} \leq Z^*.$$

Many methods for solving ILP problems depend on the availability of good lower (upper) bounds:

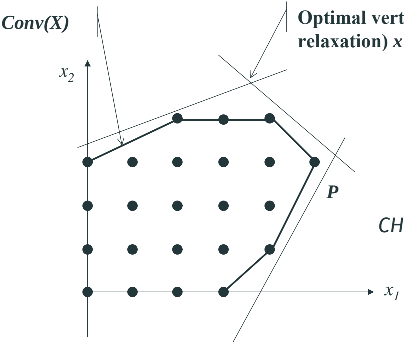
The sharper the bound (closer to Z^*)

$\downarrow$

the better the ILP methods behave.

Which is the ideal formulation of an ILP problem?

Let $X = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^k\}$ be the set of feasible integer solutions to an ILP problem. We assume that the feasible set is bounded and, therefore, X is finite.



Convex Hull of X:

$$CH(X) = \left\{ \sum_{i=1}^{k} \lambda_i \, \mathbf{x}^i \mid \sum_{i=1}^{k} \lambda_i = 1, \quad \lambda_i \geq 0, \mathbf{x}^i \in X \right\}$$

Figure 1: Definition of Convex Hull

Convex Hull

- The set $CH(X)$ is a polyhedron with all integer extreme points.
- The feasible set P of any linear programming relaxation satisfies $CH(X) \subset P$.
- If we knew $CH(X)$ explicitly, i.e. $CH(X) = \{\mathbf{x} \mid D\mathbf{x} \leq \mathbf{d}\}$, the ILP problem:
$$\begin{aligned} min \quad & \mathbf{c}^T\mathbf{x} \\ & \mathbf{x} \in X \end{aligned}$$
could be solved by finding the optimal extreme point of the linear programming problem:
$$\begin{aligned} min \quad & \mathbf{c}^T\mathbf{x} \\ & \mathbf{x} \in CH(X) \end{aligned}$$

To look for $CH(X)$ is generally a hard task.

Compromise: to find a polyhedron that strictly approximates $CH(X)$.

Quality of a formulation

QUALITY

The **quality of a formulation** of an ILP problem, with feasible solution set X, can be judged by the closeness of the feasible set of its linear programming formulation to the convex hull of X.

In particular, let A and B be two different formulations for the same integer programming problem.

Let us denote P_A and P_B as the feasible sets of the corresponding linear programming relaxations.

Formulation A is said to be **at least as strong as** formulation B if:

$$P_A \subset P_B.$$

Modeling with exponentially many constraints

Now we demonstrate through examples that strong formulations, and in particular the Convex Hull, may involve an exponential number of constraints.

Minimum Spanning Tree Problem

- Let $G = (N, E)$ a undirected graph, with node set N ($|N| = n$) and edge set E ($|E| = m$).
- A cost c_e is associated with each edge $e \in E$.

The minimum spanning tree problem looks for a spanning tree of minimum cost.

Application: design of transportation, communication and computer networks.

The spanning tree

Definition

A spanning tree of graph $G = (N, E)$:

- should have $n - 1$ edges;
- should alternatively satisfy one of the following conditions:
    a) it must not contain a cycle (Formulation A);
    b) it has to be a connected graph (Formulation B).

Minimum Spanning Tree Problem (Formulation A)

$$\min \sum_{e \in E} c_e x_e$$

$$\sum_{e \in E} x_e = n - 1$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1 \qquad S \subset N, S \neq \emptyset$$

$$x_e \in \{0, 1\} \qquad e \in E.$$

where for any $S \subset N$, we define $E(S) = \{(i, j) \in E \mid i, j \in S\}$.

- This formulation is known as *subtour elimination formulation*.
- We indicate with P_A its linear programming relaxation (we replace binary conditions with $0 \leq x_e \leq 1$).

Minimum Spanning Tree Problem (Formulation B)

$$\min \sum_{e \in E} c_e x_e$$

$$\sum_{e \in E} x_e = n - 1$$

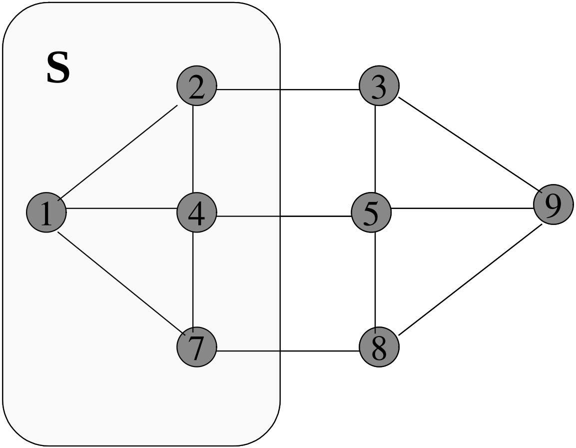
$$\sum_{e \in \delta(S)} x_e \geq 1 \qquad S \subset N, S \neq \emptyset$$

$$x_e \in \{0, 1\} \qquad e \in E.$$

where for any $S \subset N$, we define $\delta(S) = \{(i, j) \in E \mid i \in S, j \notin S\}$.

- This formulation is known as *cutset formulation*.
- We indicate with P_B its linear programming relaxation (we replace binary conditions with $0 \leq x_e \leq 1$).

Figure 2: Example: $S = \{1, 2, 4, 7\} \rightarrow E(S) = \{(1, 2), (1, 4), (1, 7), (2, 4), (4, 7)\}$ and $\delta(S) = \{(2, 3), (4, 5), (7, 8)\}$.



Which formulation is stronger?

Proposition

The following properties hold:

1. $P_A \subset P_B$, *and there exist examples for which the inclusion may be strict.*
2. *The polyhedron* P_B *can have fractional extreme points.*

The proof is interesting because it shows how we can compare alternative formulations of discrete optimization problems.

Proof of the Proposition 1/4

Proof of point 1:

For any set S of nodes, we have:

$$E = E(S) \cup \delta(S) \cup E(N \setminus S).$$

Therefore,

$$\sum_{e \in E(S)} x_e + \sum_{e \in E(N \setminus S)} x_e + \sum_{e \in \delta(S)} x_e = \sum_{e \in E} x_e.$$

Proof of the Proposition 2/4

For $\mathbf{x} \in P_A$ and for $S \subset N, S \neq \emptyset$ we have:

$$\sum_{e \in E(S)} x_e \leq |S| - 1 \text{ and } \sum_{e \in E(N \setminus S)} x_e \leq |N \setminus S| - 1.$$

Since

$$\sum_{e \in E} x_e = n - 1,$$

we obtain that

$$\sum_{e \in \delta(S)} x_e \geq 1,$$

and therefore $\mathbf{x} \in P_B$.

Proof of the Proposition 3/4

The following example shows how the inclusion may be strict.

Figure 3: The Minimum Spanning Tree $\{$(2,5), (4,5), (1,2), (3,4)$\}$ has cost 2.

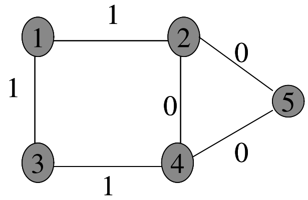
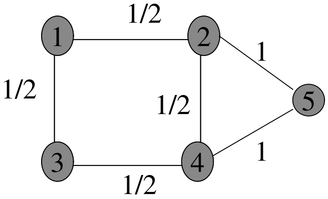


Figure 4: Optimal solution on the instance in Fig. 3 for the relaxation P_B.



Proof of the Proposition 4/4

Proof of Point 2:

To show that the polyhedron P_B may have fractional extreme points, we build an objective function under which there is a unique optimal solution that is fractional. This establishes that this unique solution is an extreme point.

Consider the example in Figure 3. The unique optimal solution to P_B is the fractional solution shown in Figure 4 with a cost of 3/2. This shows that P_B has a fractional extreme point.

Remark

It can be shown that $P_A = CH(X)$, i.e. the polyhedron P_A is a representation of the convex hull of the set of vectors corresponding to spanning trees.

The Perfect Matching Problem

Given:

- an even number n of persons to be matched into pairs to perform a job;
- a cost c_e of the edge $e = (i, j)$ for pairing person i with person j;
- an undirected graph $G = (N, E)$ with $|N| = n$ and where if edge $(i, j) \notin E$ persons i and j cannot be matched.

The Problem

Find a matching that minimizes the total cost.

Definition

Matching: a pairing of persons, so that each individual is matched with exactly another one.

Perfect Matching: Mathematical Formulations

Formulation A

$$\min \sum_{e \in E} c_e x_e$$

$$\sum_{e \in \delta(\{i\})} x_e = 1, \quad i \in N$$

$$x_e \in \{0, 1\} \quad e \in E.$$

Formulation B

Formulation A

$$+$$

$$\sum_{e \in \delta(S)} x_e \geq 1, \quad S \subset N, |S| \text{ odd}$$

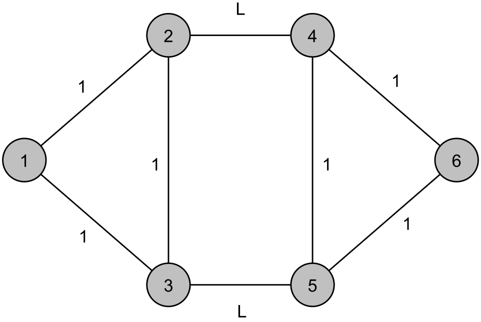
Properties

Remark

It can be shown that $P_B = CH(M)$, where M is the set of all vectors $\mathbf{x}$ corresponding to matchings.

We use example in Figure 5 to show that $P_A \neq CH(M)$

Figure 5: Instance of a Perfect Matching (costs on edges).



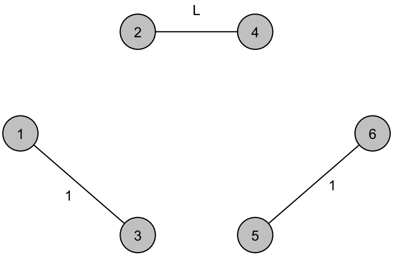


Figure 6: Optimal matching has cost $L + 2$.

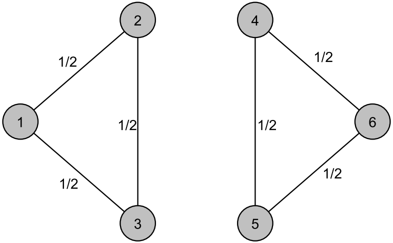


Figure 7: Optimal solution over polyhedron P_A has cost 3.