Column generation and Lagrangean relaxation: an example

Logistics

Giovanni Righini

Università degli Studi di Milano

The Generalized Assignment Problem

Data.

- a set $\mathcal{J}$ of jobs,
- a set $\mathcal{M}$ of machines,
- a processing time p_{jm} for each job $j \in \mathcal{J}$ on each machine $m \in \mathcal{M}$,
- a total available working time t_m on each machine $m \in \mathcal{M}$,
- an assignment cost c_{jm} for each job $j \in \mathcal{J}$ on each machine $m \in \mathcal{M}$.

Assign each job to a machine so that the available time is not exceeded on any machine and the total cost is minimum.

An ILP model

Variables.

- $x_{jm} \in \{0, 1\}$: assignment of job $j \in \mathcal{J}$ to machine $m \in \mathcal{M}$.

Constraints.

- Assignment constraints:

$$\sum_{m \in \mathcal{M}} x_{jm} = 1 \quad \forall j \in \mathcal{J}.$$

- Capacity constraints:

$$\sum_{j \in \mathcal{J}} p_{jm} x_{jm} \leq t_m \quad \forall m \in \mathcal{M}.$$

Objective.

$$\text{minimize } z = \sum_{j \in \mathcal{J}} \sum_{m \in \mathcal{M}} c_{jm} x_{jm}.$$

Compact formulation

$$\text{minimize } z = \sum_{j \in \mathcal{J}} \sum_{m \in \mathcal{M}} c_{jm} x_{jm}$$

$$\text{s.t.} \sum_{m \in \mathcal{M}} x_{jm} = 1 \qquad \forall j \in \mathcal{J}$$

$$\sum_{j \in \mathcal{J}} p_{jm} x_{jm} \leq t_m \qquad \forall m \in \mathcal{M}$$

$$x_{jm} \in \{0, 1\} \qquad \forall j \in \mathcal{J}, \forall m \in \mathcal{M}.$$

The problem is an *NP*-hard discrete optimization problem.

To solve it to optimality by implicit enumeration (branch-and-bound), we need dual bounds (lower bounds).

Dual bounds

Dual bounds are usually obtained by solving a relaxation to optimality.

A problem

$$R: \text{ minimize } z_R(x) \text{ s.t. } x \in X_R$$

is a relaxation of a problem

$$P: \text{ minimize } z_P(x) \text{ s.t. } x \in X_P$$

if and only if

1. $X_P \subseteq X_R$
2. $z_R(x) \leq z_P(x)\ \forall x \in X_P$

Conflicting requirements:

- the relaxation must be solved as efficiently as possible;
- the dual bound must be as tight as possible.

Relaxations

Most used relaxations:

- Continuous (linear) relaxation (MILP solvers);
- Lagrangean relaxation;
- Column generation.

The latter two relaxations are especially suitable to solve multi-agent problems:

- multi-facility location
- multi-bin packing
- multi-machine scheduling
- multi-vehicle routing
- etc.

Two main reasons:

- decomposition into independent single-agent sub-problems;
- convexification of each sub-problem: stronger lower bounds.

Coupling constraints

$$\text{minimize } z = \sum_{j \in \mathcal{J}} \sum_{m \in \mathcal{M}} c_{jm} x_{jm}$$

$$\text{s.t.} \sum_{m \in \mathcal{M}} x_{jm} \geq 1 \qquad \forall j \in \mathcal{J}$$

$$\sum_{j \in \mathcal{J}} p_{jm} x_{jm} \leq t_m \qquad \forall m \in \mathcal{M}$$

$$x_{jm} \in \{0, 1\} \qquad \forall j \in \mathcal{J}, \forall m \in \mathcal{M}.$$

In the GAP, assignment constraints work as $\geq$ constraints.

They are the only coupling constraints.

Lagrangean relaxation

Assignment constraints $\sum_{m \in \mathcal{M}} x_{jm} \geq 1 \ \ \forall j \in \mathcal{J}$ are relaxed and replaced by penalty terms in the objective function.

Violation of the assignment constraint for job j: $(1 - \sum_{m \in \mathcal{M}} x_{jm})$.

$$\text{minimize } z_{LR} = \sum_{j \in \mathcal{J}} \sum_{m \in \mathcal{M}} c_{jm} x_{jm} + \sum_{j \in \mathcal{J}} \lambda_j (1 - \sum_{m \in \mathcal{M}} x_{jm})$$

$$\text{s.t.} \sum_{j \in \mathcal{J}} p_{jm} x_{jm} \leq t_m \qquad \forall m \in \mathcal{M}$$

$$x_{jm} \in \{0, 1\} \qquad \forall j \in \mathcal{J}, \forall m \in \mathcal{M}.$$

Lagrangean multipliers $\lambda \geq 0$ must be suitably selected.

The objective function depends on x and λ.

Dual bounds

Lagrangean relaxation satisfies the condition for being a relaxation.

$$\begin{cases} \sum_{m\in\mathcal{M}} x_{jm} \geq 1 & \forall j \in \mathcal{J} \\ \lambda_j \geq 0 & \forall j \in \mathcal{J} \end{cases} \Rightarrow \lambda_j(1 - \sum_{m\in\mathcal{M}} x_{jm}) \leq 0 \; \forall j \in \mathcal{J} \Rightarrow$$

$$\Rightarrow z_{LR}(x, \lambda) \leq z(x).$$

Hence it provides a valid lower bound: $z^*_{LR}(\lambda) \leq z^* \; \forall \lambda \geq 0$.

Decomposition and convexification

The Lagrangean problem can be decomposed into independent sub-problems, one for each machine $m \in \mathcal{M}$:

$$\text{minimize } z_{LR}^m = \sum_{j \in \mathcal{J}} (c_{jm} - \lambda_j) x_{jm} + \sum_{j \in \mathcal{J}} \lambda_j$$

$$\text{s.t. } \sum_{j \in \mathcal{J}} p_{jm} x_{jm} \leq t_m$$

$$x_{jm} \in \{0, 1\} \qquad \forall j \in \mathcal{J}.$$

They are instances of the binary knapsack problem.

It can be solved effectively, although it is an $NP-$hard discrete optimization problem (convexification).

The Lagrangean dual problem

Maximizing the lower bound by a suitable selection of λ is the Lagrangean dual problem. As a function of λ, $z_{LR}(\lambda)$ is piecewise linear.

It may have an exponential number of linear parts.

The Lagrangean dual problem is an LP problem, but we do not know its explicit description: enumerating its constraints is the same as enumerating the integer solutions of the GAP.

The vector with components $(1 - \sum_{m \in \mathcal{M}} x_{jm})$ is the subgradient.

Multipliers λ are iteratively adjusted by subgradient optimization.

- Violated constraint: $(1 - \sum_{m \in \mathcal{M}} x_{jm}) > 0 \Rightarrow$ increase λ_j;
- Active constraint: $(1 - \sum_{m \in \mathcal{M}} x_{jm}) = 0 \Rightarrow$ keep λ_j unchanged;
- Inactive constraint: $(1 - \sum_{m \in \mathcal{M}} x_{jm}) < 0 \Rightarrow$ decrease λ_j;

Extended formulation

Theorem (Minkowsky and Weil). Every point in a polyhedron can be expressed as a convex combination of extreme points and extreme rays.

For each machine $m \in \mathcal{M}$, consider the set K_m of feasible integer extreme points defined as follows:

$$K_m = \{u \in \{0,1\}^{|\mathcal{J}|} : \sum_{j \in \mathcal{J}} p_{jm} u_j \leq t_m\}.$$

Each extreme point u^k, with $k \in K_m$, is the characteristic vector of a subset of jobs assigned to machine m, complying with the capacity constraint.

Extended formulation

With the above definition,

$$x_{jm} = \sum_{k \in K_m} u_j^k \theta_k \quad \forall j \in \mathcal{J}, \forall m \in \mathcal{M},$$

where coefficients θ define a convex combination:

$$\begin{cases} \sum_{k \in K_m} \theta_k = 1 & \forall m \in \mathcal{M} \\ 0 \leq \theta_k \leq 1 & \forall m \in \mathcal{M}, \forall k \in K_m. \end{cases}$$

Binary restrictions on x imply binary restrictions on θ:

$$x_{jm} \in \{0,1\} \ \forall j \in \mathcal{J}, \forall m \in \mathcal{M} \quad \Leftrightarrow \quad \theta_k \in \{0,1\} \ \forall k \in K_j.$$

The GAP can be reformulated so that θ are the variables, instead of x (extended formulation): the θ variables are exponential in number.

The master problem

An extended formulation of the GAP is the following:

$$\text{minimize } z_{MP} = \sum_{m \in \mathcal{M}} \sum_{j \in \mathcal{J}} c_{ij} \Big(\sum_{k \in K_m} u_i^k \theta_k \Big)$$

$$\text{s.t.} \sum_{m \in \mathcal{M}} \Big(\sum_{k \in K_m} u_i^k \theta_k \Big) \geq 1 \qquad \forall j \in \mathcal{J}$$

$$\sum_{k \in K_m} \theta_k = 1 \qquad \forall m \in \mathcal{M}$$

$$\theta_k \in \{0, 1\} \qquad \forall m \in \mathcal{M}, \forall k \in K_m$$

The original binary variables x have been replaced by convex combinations of extreme points of subsets K_m.

The new binary variables θ are exponential in number.

The set of integer solutions of the extended formulation is the same as the set of integer solutions of the compact one.

However, its continuous linear relaxation is tighter, owing to the convexification of the capacity constraints.

Linear master problem

The continuous linear relaxation (LMP) is the following:

$$\text{minimize } z_{LMP} = \sum_{m \in \mathcal{M}} \sum_{j \in \mathcal{J}} c_{ij} \Big(\sum_{k \in K_m} u_i^k \theta_k \Big)$$

$$\text{s.t.} \sum_{m \in \mathcal{M}} \Big(\sum_{k \in K_m} u_i^k \theta_k \Big) \geq 1 \qquad \forall j \in \mathcal{J} \qquad (1)$$

$$\sum_{k \in K_m} \theta_k = 1 \qquad \forall m \in \mathcal{M} \qquad (2)$$

$$0 \leq \theta_k \leq 1 \qquad \forall m \in \mathcal{M}, \forall k \in K_m$$

Constraints $\theta_k \leq 1$ are redundant.

Assignment constraints (coupling constraints) (1) are in the master problem.

Capacity constraints are not: they are now hidden in the definition of K_m for each machine.

Convexity constraints (2) have appeared.

Linear master problem

We introduce $w_k = \sum_{j \in \mathcal{J}} c_{ij} u_i^k$: the cost of each subset $k \in K_m$.

$$\text{minimize } z_{LMP} = \sum_{m \in \mathcal{M}} \sum_{k \in K_m} w_k \theta_k$$

$$\text{s.t.} \sum_{m \in \mathcal{M}} \sum_{k \in K_m} u_i^k \theta_k \geq 1 \qquad \forall j \in \mathcal{J} \qquad (3)$$

$$\sum_{k \in K_m} \theta_k = 1 \qquad \forall m \in \mathcal{M} \qquad (4)$$

$$\theta_k \geq 0 \qquad \forall m \in \mathcal{M}, \forall k \in K_m.$$

Dual variables.

- $\lambda_j \geq 0$: dual variables of covering constraints (3);
- μ_m (unrestricted): dual variables of the convexity constraint (4).

The reduced cost $\overline{w}_k$ of each column $k \in K_m$ is:

$$\overline{w}_k = w_k - \sum_{j \in \mathcal{J}} \lambda_j u_i^k - \mu_m.$$

Linear restricted master problem

Since columns θ_k are exponential in number, we consider only a (small enough) subset $\overline{K}_m$ of them for each machine m.

$$\text{minimize } z_{LRMP} = \sum_{m \in \mathcal{M}} \sum_{k \in \overline{K}_m} w_k \theta_k$$

$$\text{s.t.} \sum_{m \in \mathcal{M}} \sum_{k \in \overline{K}_m} u_i^k \theta_k \geq 1 \qquad \forall j \in \mathcal{J}$$

$$\sum_{k \in \overline{K}_m} \theta_k = 1 \qquad \forall m \in \mathcal{M}$$

$$\theta_k \geq 0 \qquad \forall m \in \mathcal{M}, \forall k \in \overline{K}_m.$$

Optimality of *LRMP*: $\overline{w}_k \geq 0 \; \forall m \in \mathcal{M}, \forall k \in \overline{K}_m$.

Optimality of *LMP*: $\overline{w}_k \geq 0 \; \forall m \in \mathcal{M}, \forall k \in K_m$.

Column generation

Iteratively, the following problems are solved:

- Master problem: the LRMP is solved to optimality; optimal dual values λ and μ are obtained;
- Pricing problem: one or more columns with negative reduced cost are searched:
  - if one or more columns with $\overline{w}_k < 0$ are found, insert them in the LRMP and solve again.
  - otherwise, the solution of the *LRMP* is optimal for the *LMP*: a valid dual bound has been achieved.

- The master problem is solved with LP algorithms (dual simplex);
- The pricing problem usually requires specialized algorithms (discrete optimization problem).

In the GAP (and in many other cases), the pricing problem

- can be solved independently for each machine (decomposition);
- it is the same discrete optimization problem as in Lagrangean relaxation (convexification).

The pricing problem

Determining a column requires determining its coefficients u_j (which job is assigned to the machine).

These are the variables of the pricing problem.

For each machine $m \in \mathcal{M}$, dropping the index k, the pricing problem is:

$$\begin{aligned} \text{minimize } \overline{w} = & w - \sum_{j \in \mathcal{J}} \lambda_j u_j - \mu_m \\ \text{s.t. } & \sum_{j \in \mathcal{J}} p_{jm} u_j \leq t_m \\ & u_j \in \{0, 1\} \qquad \forall j \in \mathcal{J}. \end{aligned}$$

It is a binary knapsack problem (*NP*-hard), but effectively solvable in practice.

Branch-and-price

For several multi-agent/multi-facility optimization problems, state-of-the-art algorithms are branch-and-bound algorithms, where the lower bound at each node is computed by column generation: branch-and-price algorithms.

There are software frameworks to make their development easier.

Many further ideas and techniques are needed:

- primal heuristics
- branching rules
- pricing algorithms
- stabilization techniques,...