Smart column generation solver for multi-commodity flow problems

Enrico Bettiol, Pierre Fouilhoux, Roland Grappe, Mathieu Lacroix, Joseph Le Roux, Lucas Létocart, Sébastien Martin, Youcef Magnouche, Alexandre Schulz, Emiliano Traversi, Roberto Wolfler Calvo

10/11/2022

Outline

Problem definition

Formulations

Baseline algorithm

Project

- 1 - Dual Ascent
- 3 - Cutting Planes
- 2 - Smart Pricing
- 4 - Smart Matheuristic

Input

- Directed graph $G = (V, A)$,
- Arc capacity $c_a \in \mathbb{Z}_+$ (MB/s),
- Arc unitary routing cost $r_a \in \mathbb{Z}_+$,
- Set K of demands with for each $k \in K$:
  - $s_k \in V$: source of the demand k,
  - $t_k \in V$: target of the demand k,
  - $b_k \in V$: bandwith of the demand k.

Unsplittable Multicommodity Flow Problem (UMFP)

Find for each demand $k \in K$ an $s_k t_k$-path p_k^* such that:

- the capacities are satisfied (the total bandwith of the demands routed through an arc is no more than its capacity):

$$\sum_{k \in K : a \in p_k^*} b_k \leq c_a$$

- the sum of the costs of the paths is minimum:

$$\sum_{k \in K} \left(b_k \times \sum_{a \in p_k^*} r_a \right)$$

Compact Formulation

Variables

$$x_a^k = \begin{cases} 1 & \text{if } k \text{ is routed through arc } a, \\ 0 & \text{otherwise,} \end{cases}$$

for all arcs $a \in A$ and all demands $k \in K$.

$$\min \sum_{a \in A} r_a \sum_{k \in K} b_k x_a^k$$

$$\sum_{a \in \delta^{\text{out}}(v)} x_a^k - \sum_{a \in \delta^{\text{in}}(v)} x_a^k = \begin{cases} b_k & \text{if } v = s_k, \\ -b_k & \text{if } v = t_k, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall v \in V, \forall k \in K, \tag{1}$$

$$\sum_{k \in K} b_k x_a^k \leq c_a \qquad \forall a \in A, \tag{2}$$

$$x_a^k \in \{0, 1\} \qquad \forall k \in K, a \in A. \tag{3}$$

Extended Formulation

Notations

- For a demand $k \in K$, let P_k denote the set of $s_k t_k$-paths,
- For a path p and an arc a, let

$$\chi_{a,p} = \begin{cases} 1 & \text{if } a \text{ belongs to } p, \\ 0 & \text{otherwise.} \end{cases}$$

Extended Formulation

Variables

$$x_p = \begin{cases} 1 & \text{if } k \text{ is routed through path } p, \\ 0 & \text{otherwise,} \end{cases}$$

for all demands $k \in K$ and all paths $p \in P_k$.

$$\min \sum_{a \in A} r_a \sum_{k \in K} b_k \sum_{p \in P_k} \chi_{a,p} x_p$$

$$\sum_{p \in P_k} x_p = 1 \qquad \forall k \in K, \qquad (4)$$

$$\sum_{k \in K} b_k \sum_{p \in P_k} \chi_{a,p} x_p \leq c_a \qquad \forall a \in A, \qquad (5)$$

$$x_p \in \{0, 1\} \qquad \forall k \in K, p \in P_k. \qquad (6)$$

Column generation and rounding based approach

Heuristic algorithm

1. Compute the linear relaxation of the extended formulation
2. Apply rounding procedure to obtain a solution

Linear relaxation

Solve the linear relaxation with column generation.

- λ: dual variables associated with inequalities (4),
- μ: dual variables associated with inequalities (5).

Pricing problem

Looking for a path $\bar{p} \in P_{\bar{k}}$ with negative reduced cost:

$$b_{\bar{k}} \sum_{a \in \bar{p}} (r_a + \mu_a) - \lambda_{\bar{k}} < 0$$

It reduces to compute a shortest path with nonnegative costs (polynomial).

Let x^* denote the computed optimal solution of the linear relaxation.

Rounding procedure

Try several rounding attempts and return the best solution found.

Rounding attempt

1. Randomly order the demands
2. For each demand $k \in K$ (following the order):
   2.1 Let $\bar{P}_k \subseteq P_k$ be the set of paths p such that:
      - $x^*_p > 0$,
      - the remaining capacities are enough to route k through p.

   2.2 If $\bar{P}_k = \emptyset$:
      2.2.1 Compute an $s_k t_k$-shortest path p^*_k considering only arcs having remaining capacity no less than b_k. If no such a path exists, STOP.

   2.3 Else:
      2.3.1 Sample a path p^*_k in $\bar{P}_k$ with probability

$$\frac{x^*_p}{\sum_{p \in \bar{P}_k} x^*_p} \qquad \text{for all paths } p \in \bar{P}_k$$

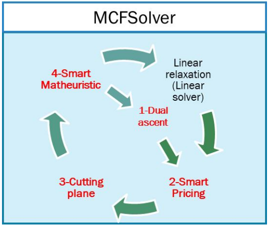
   2.4 Route k through p^*_k and decrease arc capacities accordingly.

Algorithm improvements

Final objectives

Using Machine learning and Combinatorial Optimization, improve the baseline algorithm by:

- decreasing its running time, especially the time necessary to solve the linear relaxation,
- improving the gap:
  - finding better solutions,
  - improving the quality of the lower bound (linear relaxation).



1 - Dual Ascent

The Dual Ascent method:

- generates *dual feasible solutions* for the *linear relaxation* of the problem;
- is faster than simplex-based methods;
- is based on a *parametric relaxation* of the original SP problem;
- is based on a *Lagrangian relaxation* of the problem combined with sub-gradient optimization;
- is suitable to solve the *restricted master problem* in a column generation framework.

The generated dual feasible solution:

- provides a *strong* bound;
- is obtained by solving simpler and smaller independent subproblems.

Comparison with classic Lagrangian relaxation

Given a generic problem P:

$$\begin{aligned} \min \quad & c^\top x \\ \text{s. t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

(Classic) Lagrangian relaxation $\forall \lambda \in \mathbb{R}^m$:

$$\phi(\lambda) = \min_{x \geq 0} \quad (c - A^\top \lambda)^\top x + \lambda^\top b$$

The problem:

$$\max_{\lambda \in \mathbb{R}^m} \phi(\lambda)$$

is the dual of P.

Formulation

Notation:

- $M = \{1, \dots m\}$: set of objects (e.g. customers) (rows)
- $N = \{1, \dots n\}$: set of subsets of objects (e.g. feasible routes) (columns)
- $R_j \subset M$, $j \in N$: subsets of objects (e.g. customers visited in route j)
- c_j: cost of subset j
- $N_i \subset N := \{j \in N | i \in R_j\}$: set of columns *covering* row i

Find a minimum-cost family of subsets R_j, $j \in N$ which is a partition of M.

Formulation

$$\begin{aligned} \min \quad & \sum_{j \in N} c_j x_j \\ \text{s. t.} \quad & \sum_{j \in N_i} x_j = 1 \qquad \forall i \in M \\ & x_j \in \{0,1\} \qquad \forall j \in N \end{aligned}$$

Example

$$\begin{aligned} \min \quad & c_1 x_1 + c_2 x_2 + c_3 x_3 \\ \text{s. t.} \quad & \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \\ & x_1, x_2, x_3 \in \{0,1\}. \end{aligned}$$

Formulation (Primal, linear relaxation)

$$(P) \qquad \min \quad z_P = \sum_{j \in N} c_j x_j$$

$$\text{s. t.} \quad \sum_{j \in N_i} x_j = 1 \qquad \forall i \in M \qquad [u_i]$$

$$x_j \geq 0 \qquad \forall j \in N$$

Dual problem formulation

$$(D) \qquad \max \ z_D = \sum_{i \in M} u_i$$

$$\text{s. t.} \quad \sum_{i \in R_j} u_i \leq c_j \qquad \forall j \in N$$

$$u_i \in \mathbb{R} \qquad \forall i \in M$$

Introducing new variables

Replace variables

Replace variable x_j by $|R_j|$ variables $y_j^i \in \{0, 1\}$ for $i \in R_j$:

- $y_j^i = 1 \iff$ row i is covered by column j;

Associate positive weights $q_i > 0$ with rows:

$$x_j = \sum_{i \in R_j} \frac{q_i}{q(R_j)} y_j^i$$

where

$$q(R_j) := \sum_{i \in R_j} q_i$$

If $x_j = 1$ then $y_j^i = 1 \, \forall i \in R_j$
If $x_j = 0$ then $y_j^i = 0 \, \forall i \in R_j$

$$\begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Reformulation

Parametric relaxation (PR) of SP problem

$$\min \quad z_{RP}(q) = \sum_{j\in N}\sum_{i\in R_j} c_j \frac{q_i}{q(R_j)} y_j^i \tag{7}$$

$$\text{s. t.} \quad \sum_{j\in N_i}\sum_{k\in R_j} \frac{q_k}{q(R_j)} y_j^k = 1 \qquad \forall i \in M \tag{8}$$

$$\sum_{j\in N_i} y_j^i = 1 \qquad \forall i \in M \tag{9}$$

$$y_j^i \geq 0 \qquad \forall i \in M,\ j \in N \tag{10}$$

Lagrangian relaxation

We relax constraints (8) with vector of multipliers $\lambda \in \mathbb{R}^m$:

$$\begin{aligned}
\min \quad & z_{LRP}(\lambda, q) = \sum_{i \in M} \Big(\sum_{j \in N_i} q_i \frac{(c_j - \lambda(R_j))}{q(R_j)} y_j^i + \lambda_i \Big) & \\
\text{s. t.} \quad & \sum_{j \in N_i} y_j^i = 1 & \forall i \in M \\
& y_j^i \geq 0 & \forall i \in M,\ j \in N
\end{aligned}$$

where $\lambda(R_j) = \sum_{i \in R_j} \lambda_i$.

This is *separable* into m independent problems.

Let

$$\tilde{c}_j := \frac{(c_j - \lambda(R_j))}{q(R_j)}.$$

Solutions of a subproblem

For each $i \in M$, each subproblem is:

$$\begin{aligned} \min \ & z^i_{LRP}(\lambda, q) = \sum_{j \in N_i} q_i \tilde{c}_j y^i_j + \lambda_i \\ \text{s. t.} \ & \sum_{j \in N_i} y^i_j = 1 \\ & y^i_j \geq 0 \qquad \forall j \in N. \end{aligned}$$

Let $j_i \in N_i$ such that

$$\tilde{c}_{j_i} = \min_{j \in N_i} \tilde{c}_j$$

The solution is given by:

$$y^i_j = \begin{cases} 1 \text{ if } j = j_i, \\ 0 \text{ otherwise} \end{cases}$$

$z^i_{LRP}(\lambda, q) = \tilde{c}_{j_i} q_i + \lambda_i.$

Dual feasible solution

$$z_{LRP}(\lambda, q) = \sum_{i \in M} (\tilde{c}_{j_i}(\lambda, q) q_i + \lambda_i)$$

Theorem The optimal solution of the decomposed problem provides a feasible dual solution:

$$u_i := \tilde{c}_{j_i}(\lambda, q) q_i + \lambda_i$$

where j_i is defined above, with dual value equal to $z_{LRP}(\lambda, q)$.

Idea of the proof It is easy to check that $\sum_{i \in R_j} u_i \leq c_j \quad \forall j \in N$.

Corollary For all $\lambda \in \mathbb{R}^m$, $q > 0$:

$$z_{LRP}(\lambda, q) \leq z_D^*$$

where z_D^* is the optimal value of the dual problem (D).

The Classical Lagrangian Relaxation (CLR)

$$\min \; z_{CLR}(\lambda) = \sum_{j \in N} (c_j - \lambda(R_j)) x_j + \sum_{i \in M} \lambda_i \quad s.t. \quad 0 \leq x_j \leq 1 \quad \forall j \in N$$

Let $\hat{c_j}(\lambda) := c_j - \lambda(R_j)$

Solution

Let $H \subset N, H := \{j \in N \,|\, \hat{c_j}(\lambda) < 0\}$. Therefore: $x_j = 1$ if $j \in H$ and 0 otherwise. $z_{CLR}(\lambda) = \sum_{j \in H} \hat{c_j}(\lambda) + \sum_{i \in M} \lambda_i$

Theorem

$z_{LRP}(\lambda, q) \geq z_{CLR}(\lambda)$

What we are doing in the project

- Adapt the DA to our problem
  - inequalities instead of equalities
  - coefficients greater than one everywhere
- Compare with other dual ascent approaches
- Use DA for removing LP solver

3 - Cutting planes

Capacity inequalities

- Volumes of objects ⇒ demand bandwiths,
- Capacity of the knapsack ⇒ arc capacity.

Inequalities valid for the knapsack polytope can be used to strengthen the linear relaxation of UMFP

Cover inequalities

For any arc $a \in A$, a cover C of a is a subset of K satisfying $\sum_{k \in C} b^k > c_a$.

$$\sum_{k \in C} \Big(\sum_{p \in P_k} \chi_{a,p} x_p \Big) \leq |C| - 1 \qquad \begin{array}{l} \text{for all arcs } a \in A \\ \text{and for all minimal covers } C \text{ of } a \end{array} \tag{11}$$

Lagrangian Decomposition based formulation

- Duplicate variables in the compact formulation,
- Dualize the linking equations,
- Apply Dantzig-Wolfe reformulation for both types of variables

$$\min \sum_{a \in A} r_a \sum_{k \in K} b_k x_a^k$$

$$\sum_{a \in \delta^{\text{out}}(v)} x_a^k - \sum_{a \in \delta^{\text{in}}(v)} x_a^k = \begin{cases} 1 & \text{if } v = s_k, \\ -1 & \text{if } v = t_k, \\ 0 & \text{otherwise}, \end{cases} \qquad \forall v \in V, \forall k \in K, \tag{12}$$

$$\sum_{k \in K} b_k y_a^k \leq c_a \qquad \forall a \in A, \tag{13}$$

$$x_a^k = y_a^k \qquad \forall k \in K, a \in A, \tag{14}$$

$$x_a^k \in \{0, 1\} \qquad \forall k \in K, a \in A. \tag{15}$$

Lagrangian Decomposition based formulation

- P_k: set of $s_k t_k$-paths for $k \in K$
- B_a: set of patterns, ie sets of demands which can all be routed through arc a at the same time, for $a \in A$

$$\min \sum_{a \in A} r_a \sum_{k \in K} b_k \sum_{p \in P_k} \chi_{a,p} x_p$$

$$\sum_{p \in P_k} x_p = 1 \qquad \forall k \in K, \tag{16}$$

$$\sum_{b \in B_a} y_b = 1 \qquad \forall a \in A, \tag{17}$$

$$\sum_{p \in P_k} \chi_{a,p} x_p = \sum_{b \in B_a} \chi_{a,b} y_b \qquad \forall a \in A, \forall k \in K, \tag{18}$$

$$x_p \in \{0,1\} \qquad \forall k \in K, p \in P_k, \tag{19}$$

$$y_b \in \{0,1\} \qquad \forall a \in A, b \in B_a. \tag{20}$$

Lagrangian Decomposition based formulation

- P_k: set of $s_k t_k$-paths for $k \in K$
- B_a: set of patterns, ie sets of demands which can all be routed through arc a at the same time, for $a \in A$

$$\min \sum_{a \in A} r_a \sum_{k \in K} b_k \sum_{p \in P_k} \chi_{a,p} x_p$$

$$\sum_{p \in P_k} x_p = 1 \qquad \forall k \in K, \tag{21}$$

$$\sum_{b \in B_a} y_b \leq 1 \qquad \forall a \in A, \tag{22}$$

$$\sum_{p \in P_k} \chi_{a,p} x_p \leq \sum_{b \in B_a} \chi_{a,b} y_b \qquad \forall a \in A, \forall k \in K, \tag{23}$$

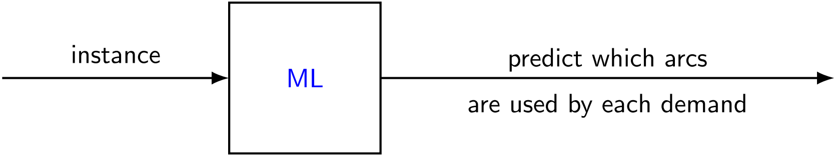
$$x_p \in \{0,1\} \qquad \forall k \in K, p \in P_k, \tag{24}$$

$$y_b \in \{0,1\} \qquad \forall a \in A, b \in B_a. \tag{25}$$

What we are doing in the project

- Adapt the DA to lagrangean decomposition
  - decompose path variables . . . straightforward
  - decompose configuration variables . . . less trivial
- Add reduntant constraints
- Build lagrangean relaxation
- Identify expressions for build dual variables

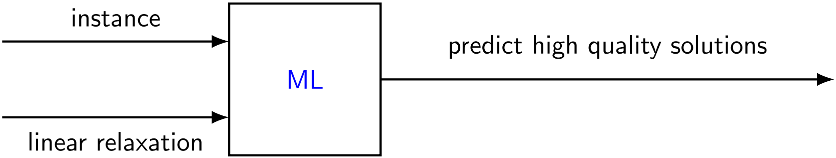
Smart Pricing



Use to

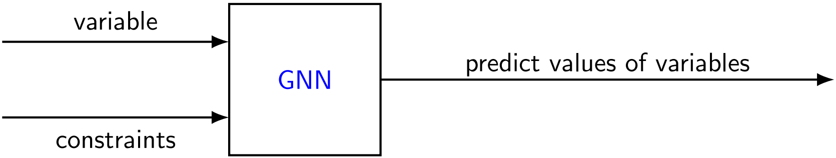
- Inizialize restricted master problem
- sparcify graph on which pricing is performed

Smart Matheuristic



ML - models for 2 and 4

Based on a bipartite graph representations of a MIP



Conclusions

- Ongoing project
- Many different works in progress
- Split in several different working groups
- Open software
- Real life instances