

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

DISOPT

MASTER SEMESTER PROJECT

AUTUMN SEMESTER 2019

---

# A combinatorial approach to the trains routing problem

---

*Student:*  
Charles Dufour

*Supervisors:*  
Prof. Friedrich Eisenbrand  
Jonas Racine

**EPFL**

# Contents

<b>Introduction</b>	<b>1</b>
<b>1 Literature review</b>	<b>2</b>
1.1 Double vertex graph . . . . .	2
1.2 MAPF . . . . .	2
<b>2 Modelization of the environment</b>	<b>3</b>
2.1 Transition network . . . . .	3
2.2 Time expanded network . . . . .	3
2.3 Constraints . . . . .	4
<b>3 Minimum cost multicommodity flow</b>	<b>5</b>
3.1 Arc flow formulation . . . . .	5
3.1.1 Relaxation and approximation . . . . .	5
3.2 Column generation method . . . . .	5
3.2.1 Reformulation as a path flow problem . . . . .	6
3.2.2 Column generation method . . . . .	8
3.2.3 Relaxation and approximation . . . . .	9
<b>4 Experimental results</b>	<b>10</b>
4.1 Arc formulation and flow formulation . . . . .	10
4.2 Approximation comparison . . . . .	10
4.3 Comparison with reinforcement learning approach . . . . .	10
<b>5 On the mixed RL-CO approach</b>	<b>11</b>
5.1 Ideas . . . . .	11
5.2 Theoretical approach . . . . .	11
5.3 Results . . . . .	11
<b>References</b>	<b>12</b>

## **Introduction**

Presentation of the challenge that serves as experimental ground.

## 1 Literature review

### 1.1 Double vertex graph

### 1.2 MAPF

## 2 Modelization of the environment

### 2.1 Transition network

The transition network is an oriented graph  $G = (V, A)$  that represents the original 2D grid world.



Figure 1: Flatland environment example. Randomly generated  $7 \times 7$  grid with 3 trains.

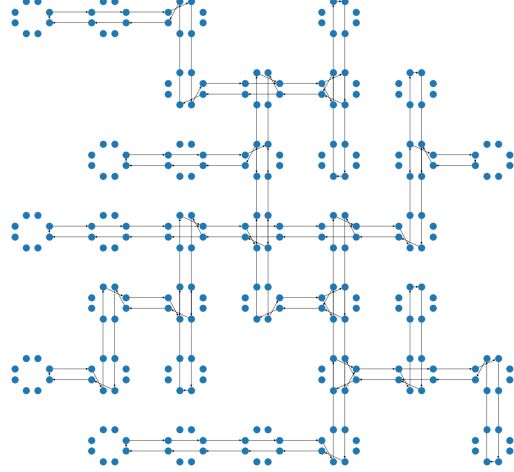


Figure 2: transition graph extracted from the flatland environment on the left.

### 2.2 Time expanded network

We give a definition inspired from (Skutella, 2008, p. 19):

**Definition 2.1** (Time-expanded network). Let  $G = (V, E)$  be a network with capacities  $u$  and costs  $c$  on the arcs. For a given time horizon  $T \in \mathbb{Z}_{>0}$ , the corresponding *time-expanded network*  $G^T = (V^T, E^T)$  with capacities and costs on the arcs is defined as follows. For each node  $v \in V$  we create  $T$  copies  $v_0, v_1, \dots, v_{T-1}$ , that is,

$$V^T := \{v_\theta | v \in V, \theta = 0, 1, \dots, T-1\}.$$

For each arc  $e = (v, w) \in E$ , there are  $T-1$  copies  $e_0, e_1, \dots, e_{T-2}$  where arc  $e_\theta$  connects node  $v_\theta$  to node  $w_{\theta+1}$ . Arc  $e_\theta$  has capacity  $u_{e_\theta} := u_e$  and cost  $c_{e_\theta} := c_e$ . Moreover,  $E^T$  contains *waiting* arcs  $(v_\theta, v_{\theta+1})$  for  $v \in V$  and  $\theta = 0, \dots, T-2$ . The capacity of waiting arcs is 1 and they have cost 1. Summarizing, the set of arcs  $E^T$  is given by

$$E^T := \{e_\theta = (v_\theta, w_{\theta+1}) | e = (u, v) \in E, \theta = 0, 1, \dots, T-2\} \\ \cup \{(v_\theta, v_{\theta+1}) | v \in V, \theta = 0, 1, \dots, T-2\}$$

An example of a time-expanded network is given in Figure ???. Notice that the size of the time-expanded network  $G^T$  is linear in  $T$  and therefore only pseudo-polynomial in the input size.

We then proceed to define a time step, which represent what would be added to a time expanded graph, were we to expand the horizon by one.

**Definition 2.2** (Time step in time expanded network). A *time step* at time  $\theta$  is the set of all edges  $(v_\theta, w)$  for all  $v \in V$ ,  $w \in V^T$  such that  $(v_\theta, w) \in E^T$ . It represents the set of all edges starting at time  $\theta$ .

**Remark.** There are no cycles in the time expanded network since  $\nexists (v_\theta, w_{\tilde{\theta}}) \in E^T$  with  $\tilde{\theta} < \theta$ .

### 2.3 Constraints

Needs to define  $C_S$  in term of trains movements and then in term of arcs flow and explain why there only span one time step.

### 3 Minimum cost multicommodity flow

We now consider the time expanded network defined in section 2.2 and denote it by  $G = (V, A)$ . The trains will be the commodities, we suppose that we have  $K$  of them.

We now formalize the definition of constraint as described in section 2.3.

**Definition 3.1.** A constraint  $S$  over a time expanded network  $G = (V, A)$  is a set of arcs:  $S \subset A$  over which we want to impose certain specification (e.g. global capacity).

We define  $C_S$  the set of all constraints over the time expanded network  $G$ .

We also introduce a notation we will be heavily using in this section:

**Definition 3.2.** Given  $\alpha$  a set,

$$\delta_\alpha(\beta) = \begin{cases} 1\{\beta \cap \alpha \neq \emptyset\} & \text{if } \beta \text{ is a set} \\ 1\{\beta \in \alpha\} & \text{otherwise} \end{cases}$$

Informally this represents the fact that  $\alpha$  and  $\beta$  are not disjoint or that  $\beta$  is contained in  $\alpha$ .

#### 3.1 Arc flow formulation

The minimum cost multicommodity flow can be formulate as an arc-flow integer program:

$$\begin{aligned} & \text{minimize} && \sum_{k=1}^K \sum_{(i,j) \in A} x_{ij}^k \\ & \text{subject to} && \sum_{k=1}^K x_{ij}^k \leq 1, && \forall (i,j) \in A \\ & && \sum_{k=1}^K \sum_{(i,j) \in S} x_{ij}^k \leq 1, && \forall S \in C_S \\ & && \mathcal{N}x^k = b^k, && \forall k \in \{1, \dots, K\} \\ & && x_{ij}^k \in \{0, 1\}, && \forall (i,j) \in A \end{aligned} \tag{3.1.1}$$

This formulation is clear and intuitive: for each commodity we decide whether or not it will use a specific arc at a certain time by setting  $x_{ij}^k$  to 1 or to 0. But this formulation (3.1.1) is not scalable due to its high number of constraints and its integrability (see section 4.1 for a more detailed explanation). We then proceed to find new ways to solve this problem.

##### 3.1.1 Relaxation and approximation

We first relax (3.1.1) by allowing  $x_{ij}^k \geq 0$  while letting the other constraints untouched.

#### 3.2 Column generation method

In this section, we give a column generation solution procedure that works with arbitrary constraints, as long as the constraints do not span more than one time step (see section 2.3 for an explanation). We directly consider the relaxation problem.

### 3.2.1 Reformulation as a path flow problem

For this section we will restate our problem using path flows instead of arc flows as before. The problem then becomes:

$$\begin{aligned}
 & \text{minimize} && \sum_{P \in \mathcal{P}} f(P) |P| \\
 & \text{subject to} && \sum_{P \in \mathcal{P}_S} f(P) \leq 1, \quad \forall S \in C_S \\
 & && \sum_{P \in P^k} f(P) = 1, \quad \forall k \in \{1, \dots, K\} \\
 & && f(P) \geq 0, \quad P \in \mathcal{P}
 \end{aligned} \tag{3.2.1}$$

With:

- $\mathcal{P} = \{\mathcal{P}_1, \dots, \mathcal{P}_K\}$  and  $\mathcal{P}_i$  is all the possible distinct paths between  $s_i$  and  $t_i$ .
- $\mathcal{P}_S = \{P \in \mathcal{P} : |P \cap S| > 0\}$ . This represents the paths that "go through" the constraint  $S$ .

The second constraint  $\sum_{P \in P^k} f(P) = 1$  contains in our case the fact that we need to have exactly one train going from  $s_k$  to  $t_k$  for each commodity.

The dual of the primal formulation (3.2.1) is:

$$\begin{aligned}
 & \text{maximize} && - \sum_{S \in C_S} y_S + \sum_{i=1}^K \sigma_i \\
 & \text{subject to} && \sum_{S \in C_S} \delta_P(S) \cdot y_S + \sigma_k \leq |P|, \quad \forall P \in \mathcal{P} \\
 & && y \geq 0, \quad y \in \mathbb{R}^{|C_S|} \\
 & && \sigma \in \mathbb{R}^K
 \end{aligned} \tag{3.2.2}$$

With respect of the dual variables, the reduced cost  $c_P^{\sigma, y}$  for each path flow variable  $f(P)$  which belongs to commodity  $k$  is :

$$c_P^{\sigma, y} = |P| - \sum_{S \in C_S} \delta_P(S) \cdot y_S - \sigma_k \tag{3.2.3}$$

We can then derive the same complementary slackness conditions.

**Theorem 3.1** (Path flow complementary slackness conditions). *The commodity path flow  $f(P)$  are optimal in the path flow formulation (3.2.1) of the multicommodity flow problem if and only if for some constraint prices  $y_S$  and commodity prices  $\sigma_k$ , the reduced cost and arc flows satisfy the following complementary slackness conditions:*

$$y_S \left[ \sum_{k \in [K]} \sum_{P \in P^k} \delta_P(S) \cdot f(P) - 1 \right] = 0 \text{ for all } S \in C_S. \tag{3.2.4}$$

$$c_P^{\sigma, y} \geq 0 \text{ for all } k \in [K] \text{ and all } P \in P^k. \tag{3.2.5}$$

$$c_P^{\sigma, y} \cdot f(P) = 0 \text{ for all } k \in [K] \text{ and all } P \in P^k. \tag{3.2.6}$$



*Proof of theorem 3.1.*

We show that optimality of the primal (3.2.1) implies the complementarity slackness conditions from theorem 3.1.

Denote  $f^*(\mathcal{P})$ ,  $y_S^*$  and  $\sigma_k^*$  the optimal solution from (3.2.1) and (3.2.2). Since  $y^*$  and  $\sigma^*$  are solution of the dual formulation we get condition (3.2.5) directly.

To show the other two conditions, we will write the primal and dual expression in matrix form.

**Primal**

$$\begin{aligned} & \text{minimize} && b^T f(\mathcal{P}) \\ & \text{subject to} && \begin{pmatrix} A_{C_S} \\ A_K \\ -A_K \end{pmatrix} \cdot f(\mathcal{P}) \geq \begin{pmatrix} -1_{|C_S|} \\ 1_K \\ -1_K \end{pmatrix} \\ & && f(\mathcal{P}) \geq 0 \end{aligned}$$

With:

- $b \in \mathbb{R}^{|\mathcal{P}|}$ ,  $b_P = |P|$  is the vector containing the length of the paths.
- $A_{C_S} \in \mathbb{R}^{|C_S| \times |\mathcal{P}|}$  where  $(A_{C_S})_{ij}$  is 1 if the  $i^{th}$  constraint contains an edge that belongs to the  $j^{th}$  path and 0 else.
- $A_K \in \mathbb{R}^{K \times |\mathcal{P}|}$  where  $(A_K)_{ij}$  is 1 if the  $j^{th}$  path belongs to  $\mathcal{P}_i$  (i.e. belongs to the  $i^{th}$  commodity) and 0 else.

**Dual**

$$\begin{aligned} & \text{maximize} && \begin{pmatrix} -1_{|C_S|}^T, 1_K^T, -1_K^T \end{pmatrix} \cdot \tilde{y} \\ & \text{subject to} && \begin{pmatrix} A_{C_S}^T, A_K^T, -A_K^T \end{pmatrix} \cdot \tilde{y} \leq b \\ & && \tilde{y} \geq 0 \end{aligned}$$

Where  $\tilde{y} = \begin{pmatrix} y \\ \tilde{\sigma}_1 \\ \tilde{\sigma}_2 \end{pmatrix}$  with  $y \in \mathbb{R}^{|C_S|}$  and  $\tilde{\sigma}_1 + \tilde{\sigma}_2 = \sigma \in \mathbb{R}^K$ , with  $y, \sigma$  being as (3.2.2).

Then we rewrite the conditions (3.2.4) and (3.2.6) in matrix form:

$$\begin{aligned} & \left[ \left( \begin{pmatrix} -1_{|C_S|}^T, 1_K^T, -1_K^T \end{pmatrix} - f(\mathcal{P})^T \begin{pmatrix} A_{C_S}^T, A_K^T, -A_K^T \end{pmatrix} \right) \cdot \tilde{y} \right]_i = 0 \quad \text{for all } i \in \{0, 1, \dots, |C_S|\} \\ & f(\mathcal{P})^T (b - \begin{pmatrix} A_{C_S}^T, A_K^T, -A_K^T \end{pmatrix} \tilde{y}) = 0 \end{aligned}$$

By the weak duality theorem we have:

$$\begin{pmatrix} -1_{|C_S|}^T, 1_K^T, -1_K^T \end{pmatrix} \cdot \tilde{y} \leq f(\mathcal{P})^T \begin{pmatrix} A_{C_S}^T, A_K^T, -A_K^T \end{pmatrix} \tilde{y} \leq f(\mathcal{P})^T b$$

Using the strong duality theorem, we also have

$$\begin{pmatrix} -1_{|C_S|}^T, 1_K^T, -1_K^T \end{pmatrix} \cdot \tilde{y} = f(\mathcal{P})^T b$$

Combining the two results from strong and weak duality we get the following two equalities:

$$\begin{aligned} \left(-1_{|C_S|}^T, 1_K^T, -1_K^T\right) \cdot \tilde{y} &= f(\mathcal{P})^T (A_{C_S}^T, A_K^T, -A_K^T) \tilde{y} \\ f(\mathcal{P})^T (A_{C_S}^T, A_K^T, -A_K^T) \tilde{y} &= f(\mathcal{P})^T b \end{aligned}$$

Which is exactly what we aimed to obtain.

The other direction is quite similar but works in the opposite way we just did.  $\square$

### 3.2.2 Column generation method

With the above formulation, we reduced the number of constraints but increased the number of variables.

We then proceed to define a *reduced master LP* on a reduced number of paths variables that can be increased if necessary to attain an optimal solution.

**Initial Solution** We produce an initial feasible solution using a greedy algorithm.

---

**Algorithm 1:** Finding and initial feasible solution for the column generation method

---

**Result:**  $\{p_1^0, \dots, p_K^0\}$ , where  $p_k^0$  is a path from  $s_k$  to  $t_k$

```

1 pathsFeasible = {};
2 for  $k \in \{1, \dots, K\}$  do
3     find shortest path candidate  $\tilde{p}$  from  $s_k$  to  $t_k$  using Dijkstra's algorithm.
4     if  $\tilde{p}$  does not have any conflicts with pathsFeasible then
5         add  $\tilde{p}$  to pathsFeasible.
6     end
7     else
8         Delete the edges of  $\tilde{p}$  from the graph.
9         Goto 3.
10    end
11 end
```

---

Where we say that a path  $p^*$  does not have conflicts with a set of paths  $P$  if using formulation (3.2.1) and putting all  $f(p) = 1 \quad \forall p \in P \cup \{p^*\}$  all the constraints are respected.

**Pricing problem** Following (Ahuja et al., 1993, p. 669), we consider only (3.2.5).

**Solving the pricing problem efficiently** Because of the particular nature of the constraints in our problem, one can see that no constraint can span more than one time step in the time expanded network, and that a path cannot have two edges in the same time step (see section 2.2).

Based on these observations, we define the cost of an arc in the graph as:

$$w_{ij} = - \sum_{S \in C_S} \delta_S((i, j)) \cdot y_S \in \mathbb{R}$$

One can notice that

$$\sum_{(i,j) \in P} \left( \sum_{S \in C_S} \delta_S((i,j)) \cdot y_S \right) = \sum_{S \in C_S} \delta_P(S) \cdot y_S$$

We can then rewrite the reduced cost as:

$$c_P^{\sigma,y} = |P| + \sum_{(i,j) \in P} w_{ij} - \sigma_k$$

So equation (3.2.5) becomes:

$$\begin{aligned} c_P^{\sigma,y} &\geq 0 & \forall P \in P^k \\ |P| + \sum_{(i,j) \in P} w_{ij} - \sigma_k &\geq 0 & \forall P \in P^k \\ |P| + \sum_{(i,j) \in P} w_{ij} &\geq \sigma_k & \forall P \in P^k \\ \sum_{(i,j) \in P} (w_{ij} + 1) &\geq \sigma_k & \forall P \in P^k \\ \min_{P \in P^k} \sum_{(i,j) \in P} (w_{ij} + 1) &\geq \sigma_k \end{aligned}$$

This shows that the pricing problem can efficiently be solved by searching for a weighted shortest paths  $p_k^*$  between  $s_k$  and  $t_k$  for all commodities  $k \in [K]$ . The weight for each arc  $(i,j) \in A$  is given by  $\tilde{w}_{ij} = w_{ij} + 1$ .

But we have to keep in mind that the weights  $\tilde{w}_{ij}$  can be negative! We then use Belman-Ford algorithm since in the time expanded graph there are no cycles.

### 3.2.3 Relaxation and approximation

Once we solved the problem above, we do not have an integral multicommodity flow, hence the solution is not feasible. In order to get a feasible solution from the fractional flow, we use the following approximations:

## 4 Experimental results

### 4.1 Arc formulation and flow formulation

Is the branch and bound process already implemented in Gurobi ? in this case just ask to solve the reduced master LP first as an IP, if it fails do one step of the column generation method and then ask again after updating the variables if the IP is feasible.

### 4.2 Approximation comparison

### 4.3 Comparison with reinforcement learning approach

## 5 On the mixed RL-CO approach

### 5.1 Ideas

### 5.2 Theoretical approach

### 5.3 Results

## References

- Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice hall.
- Dai, W., Zhang, J., & Sun, X. (2017). On solving multi-commodity flow problems: An experimental evaluation. *Chinese Journal of Aeronautics*, 30(4), 1481 – 1492.  
URL <http://www.sciencedirect.com/science/article/pii/S100093611730122X>
- Dufour Charles, G. E. (2019). Comparison of standard combinatorial optimization with reinforcement learning techniques on rescheduling problems. <https://github.com/dufourc1/SemesterProjectMA3>.
- Skutella, M. (2008). An introduction to network flows over time. In *Bonn Workshop of Combinatorial Optimization*.