Managing flow problems defined on time-expanded networks through a project/lift decomposition

Aurélien Mombelli[1], Alain Quilliot[1], Mourad Baiou[1]

LIMOS, Clermont-Auvergne INP, UCA, France

{prenom.nom}@uca.fr

Mots-clés : *time-expanded, mixed integer linear program, branch and cut.*

1 The Time-Expanded 2 Flow Problem (TE2FP)

When trying to solve problems with arcs whose status depend on the time, a straightforward way is to search for the optimal solution in a Time-Expanded Network (TEN) as did Krumke and al.[1]. A connection between two nodes in this network represents the crossing of an arc at a given time. Those kind of networks are used, among other applications, for evacuation routing problems as did Park and al.[2].

Using a TEN has a lot of advantages because it does not contain cycles and Mixed Integer Linear Programs are simpler. However, TENs depend on a time discretization. Unfortunately, the number of arcs of a TEN grows very quickly depending on the time discretization used. Of course, one can have an idea of the solution of their problem by using a coarse discretization, but the rounding errors are multiplied with every edge used in the solution.

We are working here on a *Dial A Ride Problem* defined on a network G with a time horizon T_{max}. We are required to transport multiple commodities. For every commodity k, q^k units of commodity must travel from their origin o^k to their destination d^k. They can be pickup after the date t^k and must be delivered at most δ^k time units after. Also, the number of vehicles is not fixed beforehand and the length of an arc is time dependant (i.e. L_e is a function of the time $L_e : t \mapsto L_e(t)$). Lastly, we add an important hypothesis: preemption is allowed. It means that one commodity may be handled by several vehicles.

A time-expanded network G^T is constructed by copying every node of G for every possible date of the time discretization used. Therefore, a node in this network corresponds to a node of G at a certain date. Then, let X and $Y = (Y^k, k = 1 \ldots, K)$ be two flows that satisfy Kirchhoff laws on every node of the time-expanded network G^T. Their meaning is that X is integer and represents the number of vehicles on each edge and that Y^k is real and represents the quantity of commodity k travelling on each edge. Those two vectors are linked by the fact that if commodities are transported through an arc, they must be transported by enough vehicles.

Coupling constraints

On any edge $e = ((v_1, t_1), (v_2, t_2))$ with $v_1 \neq v_2$ and $t_2 = t_1 + L_{(v_1,v_2)}(t_1)$:

$$C.X_e \geq \sum_k Y_e^k \tag{1}$$

But the number of variables is way too high and this model cannot give a precise solution in a reasonable amount of time. Therefore, we want to project the TE2FP on the graph G, solve this projected problem and lift its solution back into a TE2FP solution.

2 The projected problem

In the projected problem, we search for two flows $\bar{X}$ (integer) and $\bar{Y} = (\bar{Y}^k, k = 1\ldots, K)$ (real) defined on the network G that satisfy Kirchhoff laws and the Coupling Constraints. But they now need to satisfy sub-tour constraints.

Extended no sub-tour constraints
The time dimension may be implicitly reintroduced in the projected model by noticing that if the vehicles spend more than $T_{max}.Q$ time in a given area S, then at least Q vehicles must enter into S. Therefore, no sub-tour constraints can be extended as:

$$T_{max}. \sum_{e\in\delta^-(S)} x_e \geq \sum_{e\in\bar{\delta}(S)} l_e.x_e, \quad S \subset N\backslash\{Depot\} \tag{2}$$

where N is the set of nodes of G, $\delta^-(S)$ is the set of arcs starting from outside S and ending inside S and $\bar{\delta}(S)$ is the set of arcs which either start or end (or both) in S.

Commodities must follow acceptable paths
There is a risk that the solution $(\bar{X}, \bar{Y} = (\bar{Y}^k, k = 1\ldots, K))$ of the projected problem does not correspond to a feasible solution $(X, Y = (Y^k, k = 1\ldots, K))$ of the original problem. We ensure that $\bar{Y}$ can be lifted as a feasible vector Y by requiring from $\bar{Y}$ to be decomposable into a collection of acceptable paths:

Définition 1 *A path γ from o^k to d^k is acceptable if and only if there is γ_{o^k} (resp. γ_{d^k}) a path from the depot to o^k (resp. from d^k to the depot) such that:*

$$L(\gamma_{o^k}) + L(\gamma) + L(\gamma_{d^k}) \leq T_{max} \tag{3}$$

To verify if all those constraints are respected, the following problem is written, with Γ the set of acceptable paths γ and $y_\gamma = \left\{ \begin{array}{l} 1 \; if \;\; e \in \gamma \\ 0 \; otherwise \end{array} , e \in E \right\}$:

$$(P) \quad \begin{array}{l} Does\ it\ exist\ (\lambda_\gamma)_{\gamma\in\Gamma}, \\ s.t. \sum_k \bar{Y}^k = \sum_{\gamma\in\Gamma} \lambda_\gamma.y_\gamma \\ \lambda_\gamma \geq 0, \quad \forall \gamma \in \Gamma \end{array}$$

However, Γ is the set of all possible paths and not just the set of shortest paths. If T_{max} is large enough, Γ contains all possible paths. Therefore, this decomposition problem will be solved by column generation or, in its dual version, by cuts generation (i.e. path generation).

3 The lift issue

It consists in turning the projected solution $\bar{X}$, $\bar{Y} = (\bar{Y}^k, k = 1\ldots, K)$ into a solution X, $Y = (Y^k, k = 1\ldots, K)$ of the original problem. We deal with it in a heuristic way by solving a sequence of min cost flow problems defined on specific small-size sub-networks of the time-expanded network G^T.

References

[1] Sven O Krumke, Alain Quilliot, Annegret K Wagler, and Jan-Thierry Wegener. *Relocation in carsharing systems using flows in time-expanded networks.* Springer, 2014.

[2] Inhye Park, Gun Up Jang, Seho Park, and Jiyeong Lee. *Time-Dependent Optimal Routing in Micro-scale Emergency Situation.* IEEE, 2009.