Capacitated Arc Routing Problem with Profits

Dominique Feillet
University of Avignon
(moving soon to Ecole des Mines de Saint-Etienne)

Co-authors: C. Archetti, A. Hertz, M.G. Speranza

How column generation can prove that France should have beat Italy

- The model
  - Maximize French score
  - Subject to: Italian score ≤ French score -1
- No feasible solution found so far
- Duality can help
  - French staff should price out the referee as much as the Italian staff

Capacitated Arc Routing Problem with Profits

Dominique Feillet
University of Avignon
(moving soon to Ecole des Mines de Saint-Etienne)

Co-authors: C. Archetti, A. Hertz, M.G. Speranza

A simple approach

- Take your standard basic Branch and Price solver for VRP type problems
  - send an email to the good person if needed

- Change some lines of code

Topic: how to be (almost) as simple but efficient (towards efficient generic solvers)

Outline

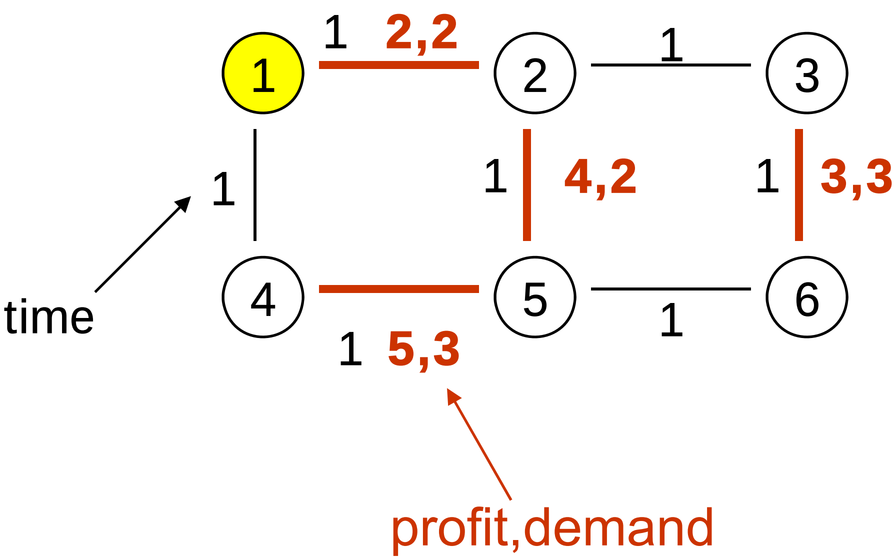
- The problem
- Model and basic algorithm
- Some robust acceleration methods
  - Initially developed for the VRPTW
- Some difficulties relative to the CARPP

The Capacitated Arc routing Problem with Profits

- Given
  - a graph G=(V,E)
  - a set Q of edges with specified demands and profits
  - a fleet of capacitated vehicles based on a depot
  - edge durations
  - a travel time limit
- Find a set of vehicle routes such that
  - the total profit is maximized
  - vehicle capacities and travel time limit are satisfied

The Capacitated Arc routing Problem with Profits

- Illustration



Number of vehicles: 3
Time limit: 5
Capacity: 10

Example of solution:
route 1=2=5=4-1

Profit: 11

Model and basic algorithm

- Model

$$\text{maximize} \sum_{r_k \in \Omega} p_k x_k$$

$$\sum_{r_k \in \Omega} a_{ek} x_k \leq 1 \qquad (e \in Q),$$

$$\sum_{r_k \in \Omega} x_k \leq m,$$

$$x_k \in \{0, 1\} \qquad (r_k \in \Omega).$$

Model and basic algorithm

- Master Problem

$$\text{maximize} \sum_{r_k \in \Omega} p_k x_k$$

$$\sum_{r_k \in \Omega} a_{ek} x_k \leq 1 \qquad (e \in Q),$$

$$\sum_{r_k \in \Omega} x_k \leq m,$$

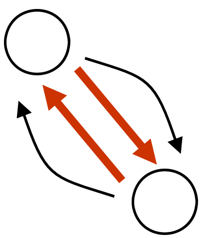
$$0 \leq x_k \qquad (r_k \in \Omega).$$

- A route has a positive reduced cost if:

$$p_k - \sum_{e \in Q} a_{ek} \lambda_e - \mu > 0$$

Model and basic algorithm

- Subproblem:
  - Shortest Path Problem with Resource Constraints
    - modeling of the edges in Q:



→ Service not performed, cost 0

→ Service performed, cost $p_e - \lambda_e$

  - Resources:
    - time
    - load
    - one binary resource for every edge in Q (a service is performed at most once)

Some robust acceleration procedures

- Heuristic solution of the subproblem
  - Limited Discrepancy Search
  - Instead of:
    - problem dependant heuristic algorithm (tabu search…)
    - scaling algorithm (consider a subgraph)

- Use of the current primal master solution in the subproblem
  - MP provides the subproblem with routes having a zero reduced cost value

Limited Discrepancy Search (LDS)

- Principle:
  - define good and bad extensions (arcs)
  - limit the number of bad extensions during the process of computing a route (discrepancies)
  - if no route of negative reduced cost is found, increase the value of the limit

- Implementation for the CARPP
  - count a discrepancy when traversing any edge unless it is in Q and
    - either you serve it
    - or it has already been served

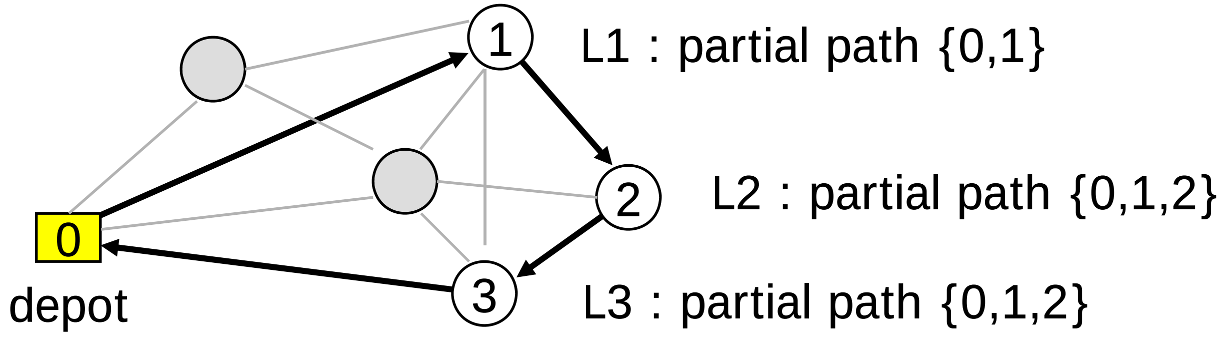
Limited Discrepancy Search (LDS)

- Interest
  - simple
  - almost problem-independent
  - efficient

- Remark:
  - the rule for counting discrepancies is here adapted to the structure of the problem
  - a standard rule would be to count a discrepancy for all outgoing arcs but the two with the most positive reduced cost

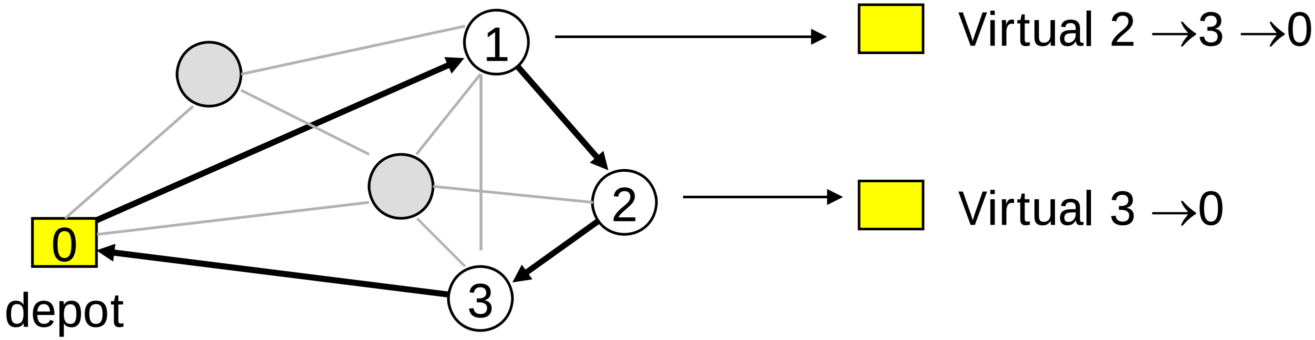
Use of the current master primal solution

- Label Loading
  - consider the set of routes with a zero reduced cost value in the current solution of the Master Problem
  - generate labels corresponding to starting parts of these routes



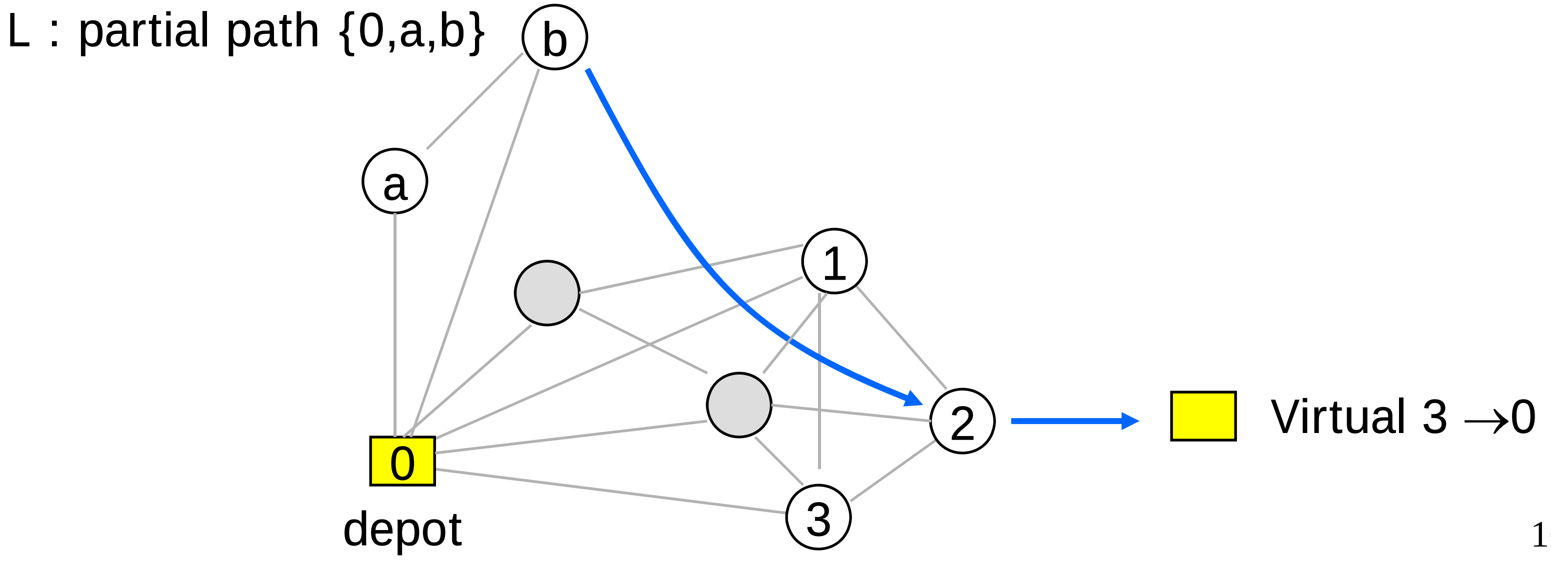
Use of the current master primal solution

- Meta Extensions
  - consider the same set of routes
  - generate virtual depots corresponding to ending parts of these routes
    - compute resource windows using resource consumption rules

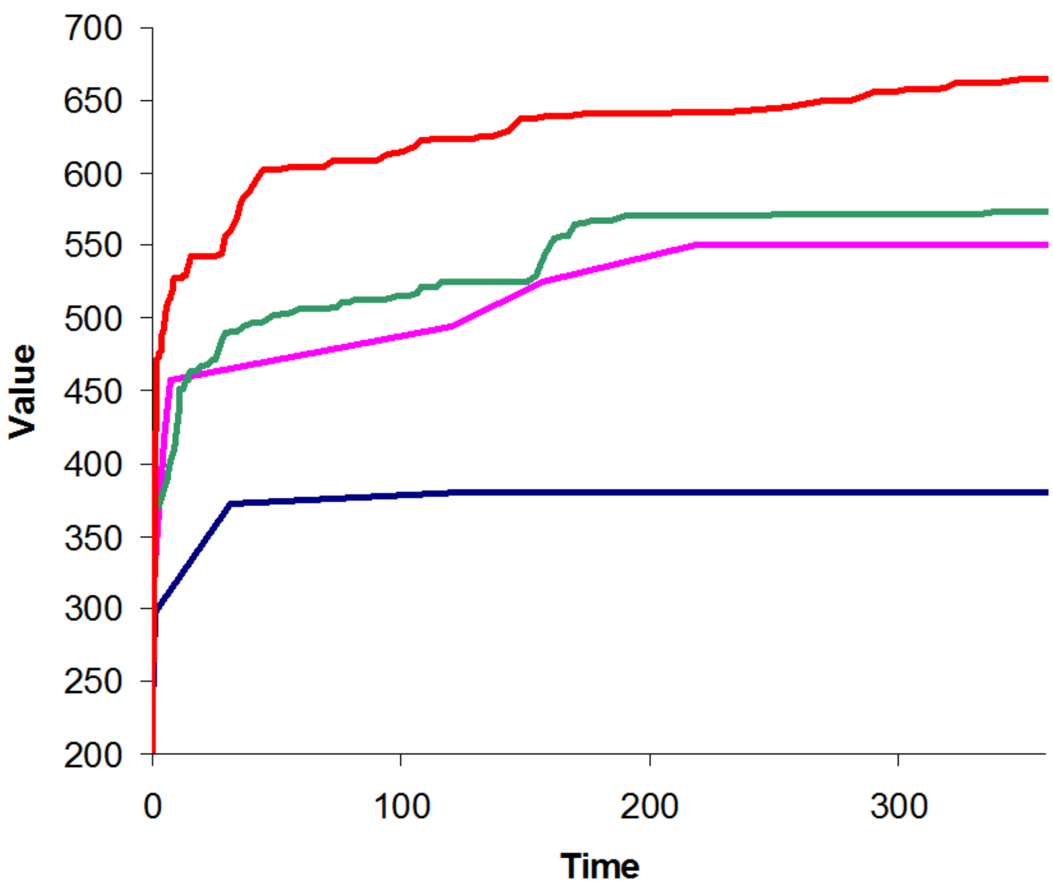


Use of the current master primal solution

- Interest
  - Amounts to a very powerful local search
  - Simple, efficient, generic



Some robust acceleration procedures



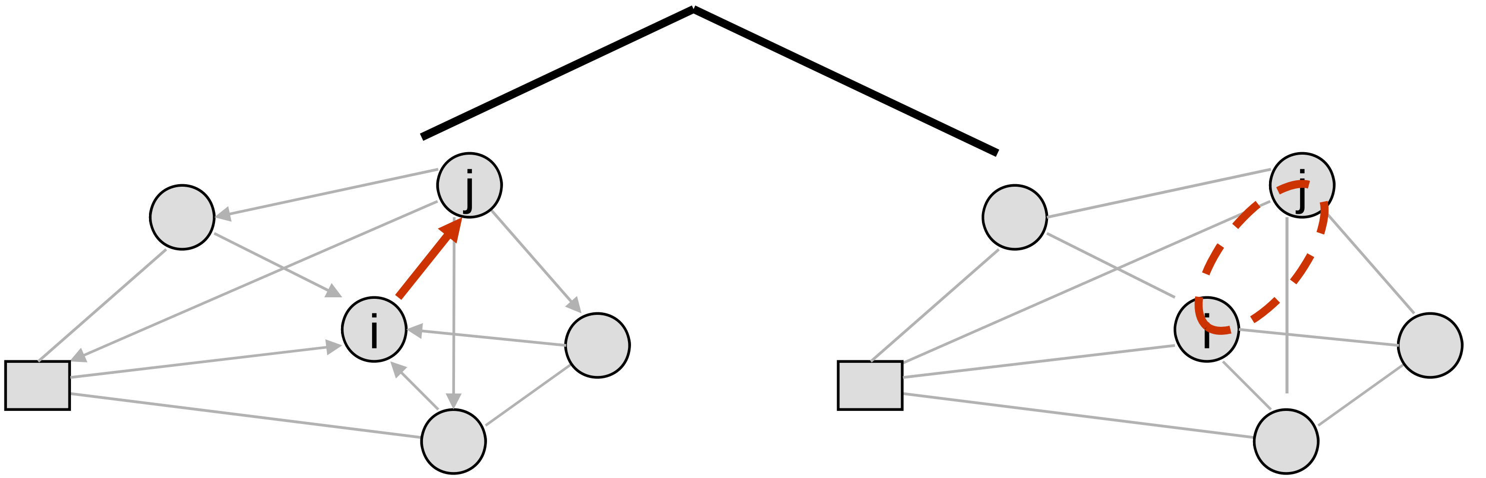
Root node
(stopped after 5 min)

Some difficulties relative to the CARPP

- Presence of long negative cost cycles
  - routes typically are about 20 edges long (in problem that we are able to solve), with vertices traversed several times
  - routes with non-elementary services of edges cannot have a value 1 in the MP
- Rounded capacity cuts are not valid
  - edges are not necessarily served

Some difficulties relative to the CARPP

- Usual branching rule when applying column generation to vehicle routing problems is not valid



  - Valid on condition that every vertex is visited exactly once in optimal solutions, which is not true here

Branching rule

- First level:
  - search for a service strictly performed between 0 and 1 times

Change ≤ to = in the MP formulation

Remove from Q (update Ω_1 and the subproblem)

Branching rule

- Second level:
  - search for an arc visited a fractional number of times

$$\sum_{r_k \in \Omega_1} a_{ijk} x_k \le \lfloor \alpha \rfloor$$

include the new dual variable in the subproblem

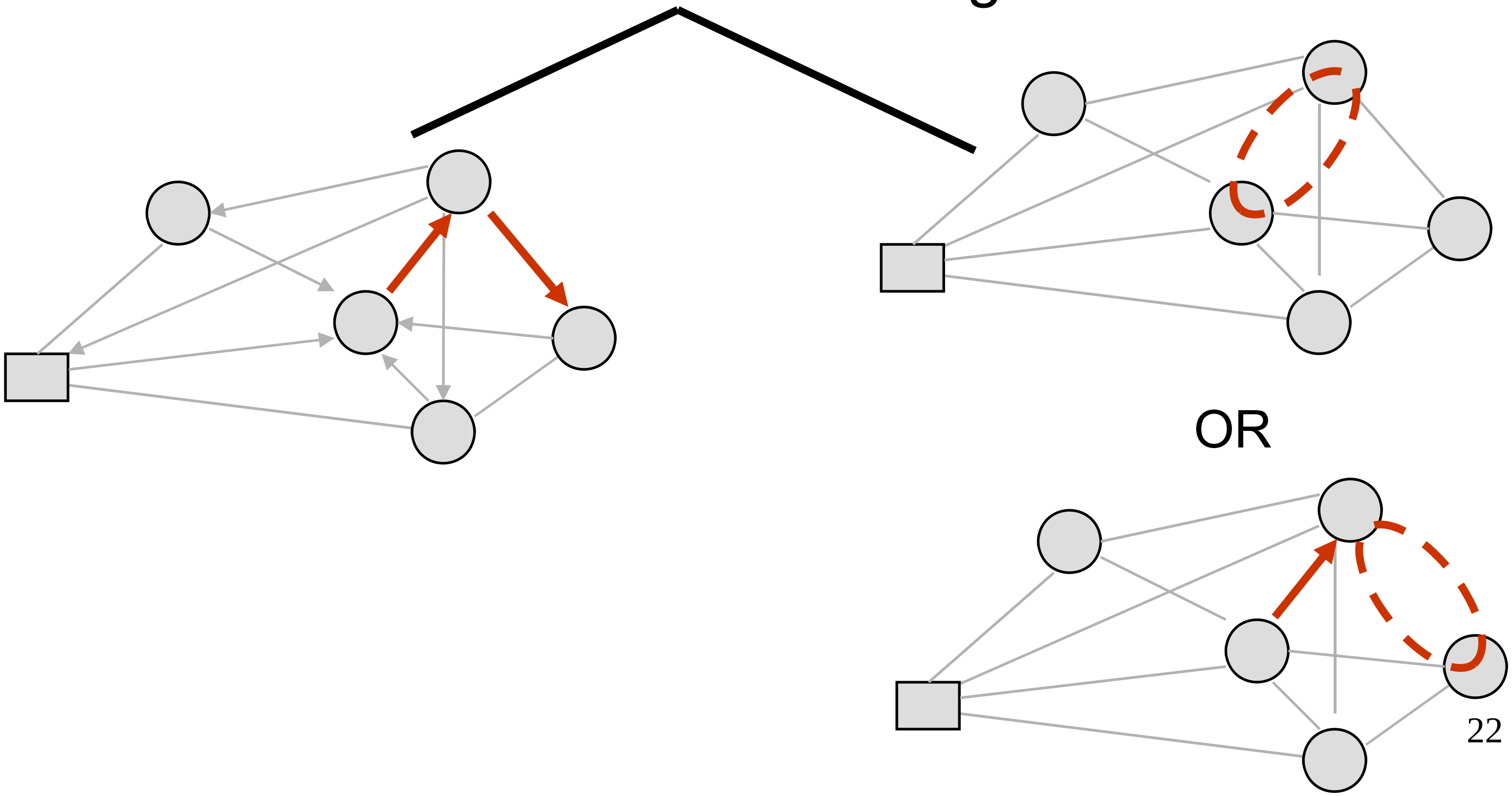
$$\sum_{r_k \in \Omega_1} a_{ijk} x_k \ge \lfloor \alpha \rfloor + 1$$

include the new dual variable in the subproblem

  - note that the time limit constraint becomes essential in case of negative cost cyles with no edge serviced

Branching rule

- Third level: Flow splitting method
  - use when the flow matrix is integer



Conclusions

- Acceleration techniques have proved to be simple to adapt and efficient
- One has to be careful with two commun tricks
  - (partial) relaxation of the condition that services are performed at most once
  - management of the branching with removal of variable (MP) and arcs (subproblem)

Questions?

- No questions on the football game allowed

Conclusions

- Perspectives
  - How the acceleration procedures could also tend to find integer solutions during the process?
    - this was often seen during the computational experiments
    - interesting for the use of column generation methods in heuristic fashions

Limited Discrepancy Search (LDS)