INSPIRING INNOVATION | INNOVANTE PAR TRADITION

Ecole Nationale Supérieure des Mines SAINT-ETIENNE

www.emse.fr

Spring School - UTT

Column generation and branch-and-price for vehicle routing problems

Introduction

Dominique Feillet – Mines Saint-Etienne and LIMOS

MINISTÈRE DE L'ÉCONOMIE DE L'INDUSTRIE ET DE L'EMPLOI

Outline: basics of column generation

1. Introduction
2. Principle
3. Implementation
4. Pricing problem
5. Branch-and-price
6. Conclusion
7. Solution of the pricing problem

Basics of column generation

INTRODUCTION

Vehicle Routing Problem with Time Windows

- Consider a directed graph **$G = (V,A)$** with **$V = \{v_0,\ldots,v_n\}$**,
  - v_0 is a depot where a fleet of **U** vehicles of capacity **Q** are based
  - v_1 to v_n are customers with demand **d_i**, time window **$[a_i,b_i]$** and service time **st_i** for all $v_i \in \{v_1,\ldots,v_n\}$
- Travel times (costs) **c_{ij}** are set on arcs $(v_i,v_j) \in A$
  - Triangle inequality is assumed
- The VRPTW aims at finding a set of U routes of minimum cost that enables satisfying the demand of customers and respects vehicle capacities and customer time windows

Compact formulation

$$\text{minimize} \sum_{1 \le u \le U} \sum_{(v_i,v_j) \in A} c_{ij} x_{ij}^u \tag{1.1}$$

subject to

$$\sum_{1 \le u \le U} \sum_{\{v_j \in V \mid (v_i,v_j) \in A\}} x_{ij}^u \ge 1 \qquad (v_i \in V \setminus \{v_0\}), \tag{1.2}$$

$$\sum_{\{v_j \in V \mid (v_i,v_j) \in A\}} x_{ij}^u - \sum_{\{v_j \in V \mid (v_j,v_i) \in A\}} x_{ji}^u = 0 \qquad (v_i \in V, 1 \le u \le U), \tag{1.3}$$

$$\sum_{\{v_i \in V \mid (v_0,v_i) \in A\}} x_{0i}^u \le 1 \qquad (1 \le u \le U), \tag{1.4}$$

$$\sum_{(v_i,v_j) \in A} d_i x_{ij}^u \le Q \qquad (1 \le u \le U), \tag{1.5}$$

$$s_i^u + st_i + c_{ij} - s_j^u + M x_{ij}^u \le M \qquad ((v_i,v_j) \in A, v_j \ne v_0, 1 \le u \le U), \tag{1.6}$$

$$s_i^u + st_i + c_{i0} - b_0 + M x_{i0}^u \le M \qquad ((v_i,v_0) \in A, 1 \le u \le U), \tag{1.7}$$

$$a_i \le s_i^u \le b_i \qquad (v_i \in V, 1 \le u \le U), \tag{1.8}$$

$$x_{ij}^u \in \{0,1\} \qquad ((v_i,v_j) \in A, 1 \le u \le U), \tag{1.9}$$

Extended formulation

- Additional notation
  - $\Omega = \{r_1,\ldots,r_{|\Omega|}\}$: set of feasible vehicle routes
  - c_k: cost of route r_k
  - $a_{ik} = 1$ if route r_k visits customer v_i, 0 otherwise

$$\text{minimize} \sum_{r_k \in \Omega} c_k \theta_k$$

subject to

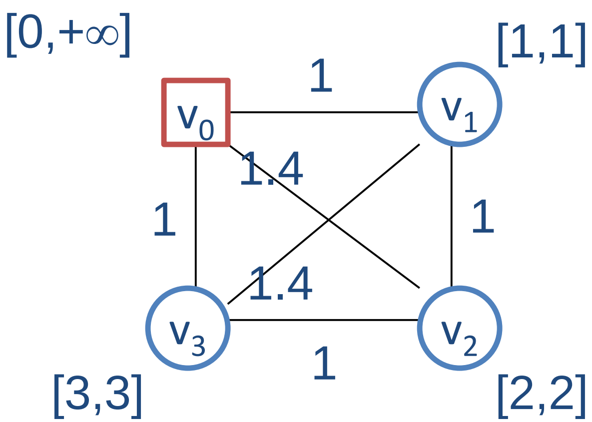
$$\sum_{r_k \in \Omega} a_{ik}\theta_k \geq 1 \qquad (v_i \in V \setminus \{v_0\}),$$

$$\sum_{r_k \in \Omega} \theta_k \leq U,$$

$$\theta_k \in \mathbb{N} \qquad (r_k \in \Omega).$$

Extended formulation: illustration

Instance



$d_i = 1$, $st_i = 0$
$Q = +\infty$
$U = +\infty$

Model

$\Omega = \{r_1,\ldots,r_7\}$ with:

$r_1 = (v_0,v_1,v_0)$ $\quad r_5 = (v_0,v_1,v_3,v_0)$
$r_2 = (v_0,v_2,v_0)$ $\quad r_6 = (v_0,v_2,v_3,v_0)$
$r_3 = (v_0,v_3,v_0)$ $\quad r_7 = (v_0,v_1,v_2,v_3,v_0)$
$r_4 = (v_0,v_1,v_2,v_0)$

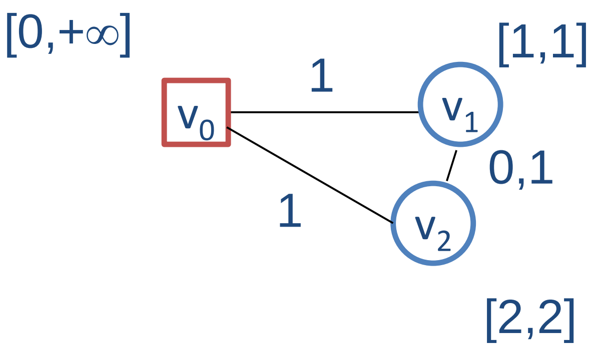
Min $2\theta_1 + 2.8\theta_2 + 2\theta_3 + 3.4\theta_4 + 3.4\theta_5 + 3.4\theta_6 + 4\theta_7$
subject to

$$\begin{cases} \theta_1 + \theta_4 + \theta_5 + \theta_7 \geq 1 \\ \theta_2 + \theta_4 + \theta_6 + \theta_7 \geq 1 \\ \theta_3 + \theta_5 + \theta_6 + \theta_7 \geq 1 \\ \theta_1,\ldots,\theta_7 \text{ integer} \end{cases}$$

Optimal solution: $\theta = \{0,0,0,0,0,0,1\}$, value = 4

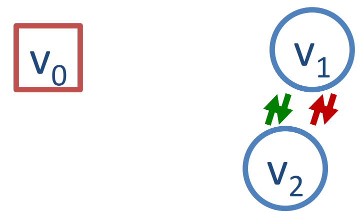
Motivation for using the extended formulation

Instance



$d_i = 1$, $st_i = 0$
$Q = +\infty$
$U = +\infty$

Compact formulation: linear relaxation



Vehicle 1 (flow 0.5)
- $x^1_{12}=x^1_{21}=0.5$
- $s^1_1=1$, $s^1_2=2$

Vehicle 2 (flow 0.5)
- $x^2_{12}=x^2_{21}=0.5$
- $s^2_1=1$, $s^2_2=2$

Example of a feasible solution
(value 0.2)

Motivation for using the extended formulation

$$\text{minimize} \sum_{1\le u\le U} \sum_{(v_i,v_j)\in A} c_{ij} x_{ij}^u \qquad (1.1)$$

subject to

$$\sum_{1\le u\le U} \sum_{\{v_j\in V \mid (v_i,v_j)\in A\}} x_{ij}^u \ge 1 \qquad (v_i \in V\setminus\{v_0\}),$$

$$\sum_{\{v_j\in V \mid (v_i,v_j)\in A\}} x_{ij}^u - \sum_{\{v_j\in V \mid (v_j,v_i)\in A\}} x_{ji}^u = 0 \qquad (v_i \in V, 1 \le u \le$$

$$\sum_{\{v_i\in V \mid (v_0,v_i)\in A\}} x_{0i}^u \le 1 \qquad (1 \le u \le U), \qquad (1.4)$$

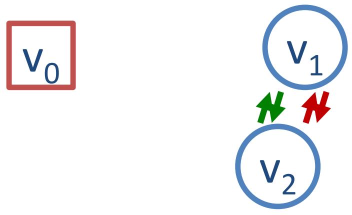
$$\sum_{(v_i,v_j)\in A} d_i x_{ij}^u \le Q \qquad (1 \le u \le U), \qquad (1.5)$$

$$s_i^u + st_i + c_{ij} - s_j^u + M x_{ij}^u \le M \qquad ((v_i,v_j)\in A, v_j \ne v_0, 1 \le u \le U), \qquad (1.6)$$

$$s_i^u + st_i + c_{i0} - b_0 + M x_{i0}^u \le M \qquad ((v_i,v_0)\in A, 1 \le u \le U), \qquad (1.7)$$

$$a_i \le s_i^u \le b_i \qquad (v_i \in V, 1 \le u \le U), \qquad (1.8)$$

$$x_{ij}^u \in \{0,1\} \qquad ((v_i,v_j)\in A, 1 \le u \le U), \qquad (1.9)$$

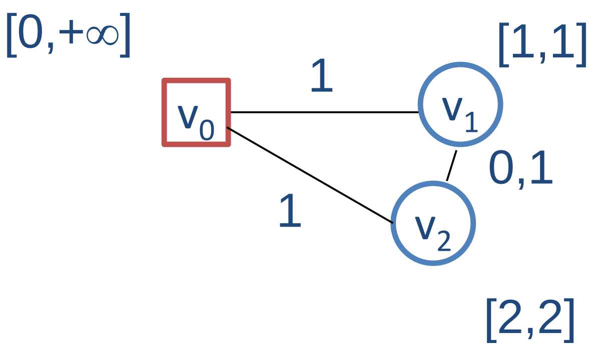


$x^1_{12}=x^1_{21}=0.5$ $\quad$ $x^2_{12}=x^2_{21}=0.5$

$s^1_1=1,\ s^1_2=2$ $\quad$ $s^2_1=1,\ s^2_2=2$

Motivation for using the extended formulation

Instance



$d_i = 1$, $st_i = 0$

$Q = +\infty$

$U = +\infty$

Extended formulation: linear relaxation

Min $2\theta_1 + 2\theta_2 + 2.1\theta_3$

subject to

$$\begin{cases} \theta_1 + \theta_3 \geq 1 \\ \theta_2 + \theta_3 \geq 1 \\ \theta_1, \ldots, \theta_3 \geq 0 \end{cases}$$

Optimal solution: $\theta = \{0,0,1\}$, value = 2.1

A few words about Dantzig-Wolfe decomposition…

$$\text{minimize} \sum_{1\le u\le U} \sum_{(v_i,v_j)\in A} c_{ij}x_{ij}^u \tag{1.1}$$

subject to

$$\sum_{1\le u\le U} \sum_{\{v_j\in V|(v_i,v_j)\in A\}} x_{ij}^u \ge 1 \qquad (v_i \in V\setminus\{v_0\}), \tag{1.2}$$

$$\sum_{\{v_j\in V|(v_i,v_j)\in A\}} x_{ij}^u - \sum_{\{v_j\in V|(v_j,v_i)\in A\}} x_{ji}^u = 0 \qquad (v_i\in V, 1\le u\le U), \tag{1.3}$$

$$\sum_{\{v_i\in V|(v_0,v_i)\in A\}} x_{0i}^u \le 1 \qquad (1\le u\le U), \tag{1.4}$$

$$\sum_{(v_i,v_j)\in A} d_i x_{ij}^u \le Q \qquad (1\le u\le U), \tag{1.5}$$

$$s_i^u + st_i + c_{ij} - s_j^u + Mx_{ij}^u \le M \qquad ((v_i,v_j)\in A, v_j\ne v_0, 1\le u\le U), \tag{1.6}$$

$$s_i^u + st_i + c_{i0} - b_0 + Mx_{i0}^u \le M \qquad ((v_i,v_0)\in A, 1\le u\le U), \tag{1.7}$$

$$a_i \le s_i^u \le b_i \qquad (v_i\in V, 1\le u\le U), \tag{1.8}$$

$$x_{ij}^u \in \{0,1\} \qquad ((v_i,v_j)\in A, 1\le u\le U), \tag{1.9}$$

= Ω^U

DW decomposition

$$\text{minimize} \sum_{r_k\in\Omega} c_k\theta_k$$

subject to

$$\sum_{r_k\in\Omega} a_{ik}\theta_k \ge 1 \qquad (v_i\in V\setminus\{v_0\}),$$

$$\sum_{r_k\in\Omega} \theta_k \le U,$$

$$\theta_k \in \mathbb{N} \qquad (r_k\in\Omega).$$

Basics of column generation

PRINCIPLE

Master Problem and Restricted Master Problems

- **Master Problem (MP)**: linear relaxation of the extended formulation

- **Restricted Master Problem (MP(Ω_t))**: restrict the variable set to a subset Ω_t of Ω

$$(MP(\Omega_1)) \qquad \text{minimize} \sum_{r_k \in \Omega_1} c_k \theta_k$$

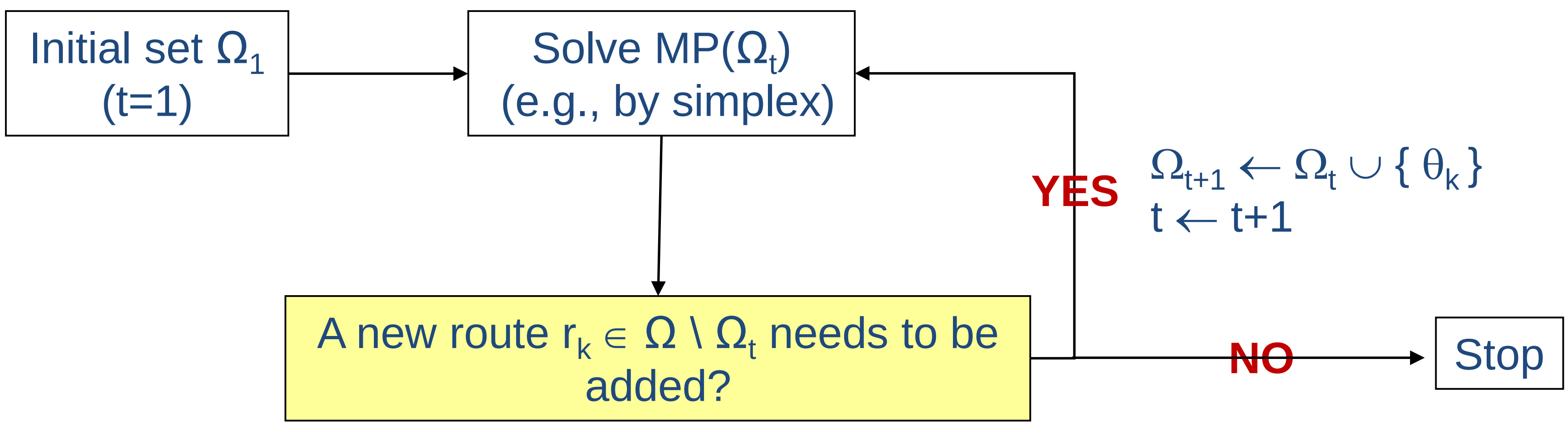
subject to

$$\sum_{r_k \in \Omega_1} a_{ik} \theta_k \geq 1 \qquad (v_i \in V \setminus \{v_0\}),$$
$$\sum_{r_k \in \Omega_1} \theta_k \leq U,$$
$$\theta_k \geq 0 \qquad (r_k \in \Omega_1).$$

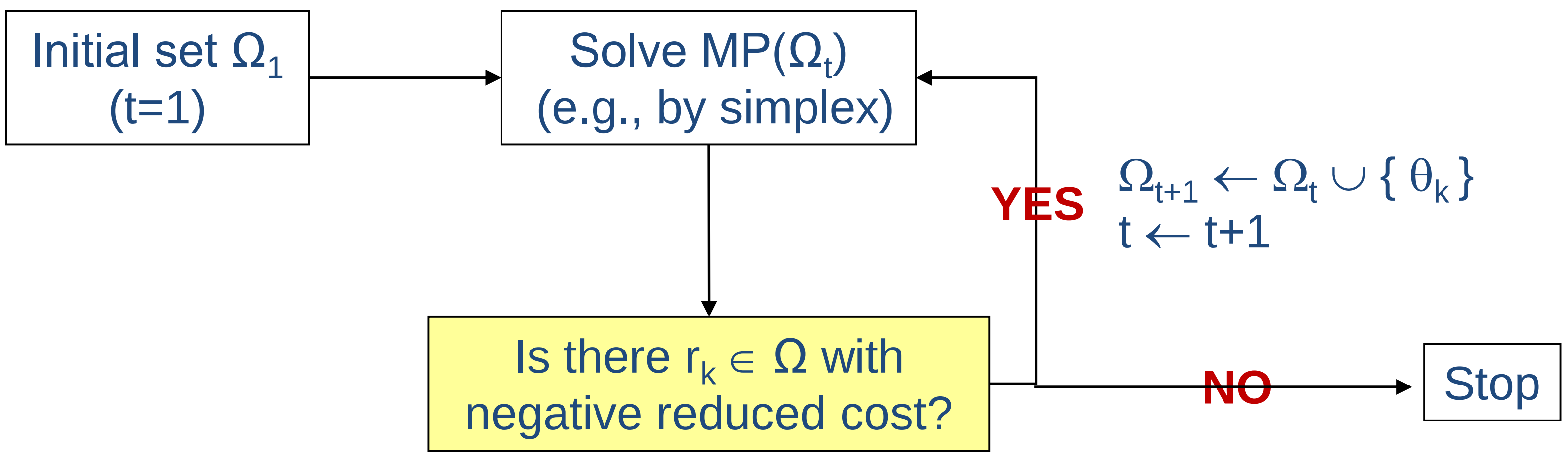
General scheme

- The aim of column generation is to solve MP

- The principle is to find a subset Ω_t such that solving MP(Ω_t) also solves MP



More detailed scheme



Computation of variable reduced costs

- Reduced cost is computed from optimal dual values

$$(MP(\Omega_1)) \qquad \text{minimize} \sum_{r_k \in \Omega_1} c_k \theta_k$$

subject to

$$\sum_{r_k \in \Omega_1} a_{ik} \theta_k \geq 1 \qquad (v_i \in V \setminus \{v_0\}),$$

$$\sum_{r_k \in \Omega_1} \theta_k \leq U,$$

$$\theta_k \geq 0 \qquad (r_k \in \Omega_1).$$

Dual variables

$\lambda_i \geq 0$

$\lambda_0 \leq 0$

- Reduced cost of variable $\theta_k \in \Omega$: $$c_k - \sum_{v_i \in V \setminus \{v_0\}} a_i^k \lambda_i - \lambda_0$$

Remarks

- In what follows terms ***variables*** / ***columns*** / ***routes*** will be used indifferently

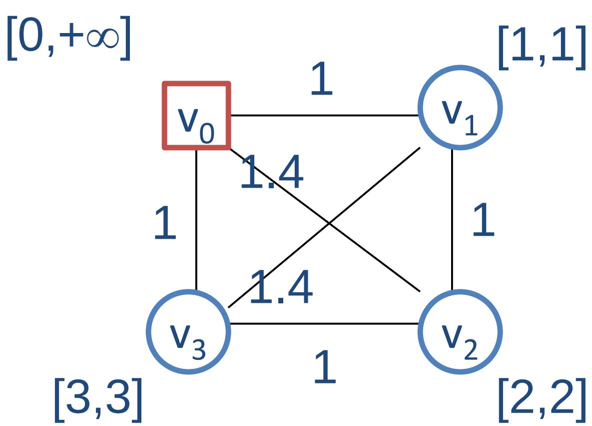
- A column is never generated more than once
  - Every column in Ω_t has a nonnegative reduced cost when MP(Ω_t) is solved

- The algorithm is finite
  - The number of columns in Ω is finite

Illustration on the previous example

- Initialization and iteration 1

Data



$d_i = 1$, $st_i = 0$

$Q = +\infty$

$U = +\infty$

$\Omega_1 = \{r_1, r_2, r_3\}$ with $r_1 = (v_0, v_1, v_0)$, $r_2 = (v_0, v_2, v_0)$, $r_3 = (v_0, v_3, v_0)$

Min $2\theta_1 + 2{,}8\theta_2 + 2\theta_3$

subject to

$$\begin{cases} \theta_1 & & & \geq 1 \\ & \theta_2 & & \geq 1 \\ & & \theta_3 & \geq 1 \\ \theta_1, \ldots, \theta_3 \geq 0 \end{cases}$$

Max $\lambda_1 + \lambda_2 + \lambda_3$

subject to

$$\begin{cases} \lambda_1 & & & \leq 2 \\ & \lambda_2 & & \leq 2.8 \\ & & \lambda_3 & \leq 2 \\ \lambda_1, \ldots, \lambda_3 \geq 0 \end{cases}$$

Optimal solution (cost = 6.8)

$\theta = (1; 1; 1)$

$\lambda = (2; 2.8; 2)$

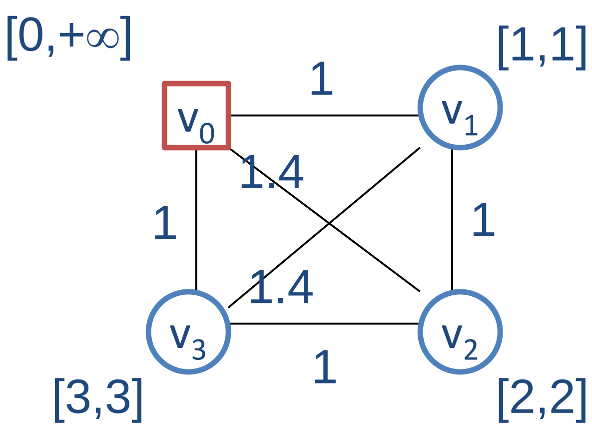
Route $r_4 = (v_0, v_1, v_2, v_0)$ has a reduced cost -1.4

(reduced cost = 3.4 - 2 – 2.8 = -1.4)

Illustration on the previous example

- Iteration 2

Data



$d_i = 1$, $st_i = 0$
$Q = +\infty$
$U = +\infty$

$\Omega_2 = \{r_1, r_2, r_3, r_4\}$ with $r_4 = (v_0, v_1, v_2, v_0)$

Min $2\theta_1 + 2{,}8\theta_2 + 2\theta_3 + \mathbf{3.4\theta_4}$
subject to

$$\begin{cases} \theta_1 + \theta_4 \geq 1 \\ \theta_2 + \theta_4 \geq 1 \\ \theta_3 \geq 1 \\ \theta_1, \dots, \theta_3, \theta_4 \geq 0 \end{cases}$$

Max $\lambda_1 + \lambda_2 + \lambda_3$
subject to

$$\begin{cases} \lambda_1 \leq 2 \\ \lambda_2 \leq 2.8 \\ \lambda_3 \leq 2 \\ \lambda_1 + \lambda_2 \leq 3.4 \\ \lambda_1, \dots, \lambda_3 \geq 0 \end{cases}$$

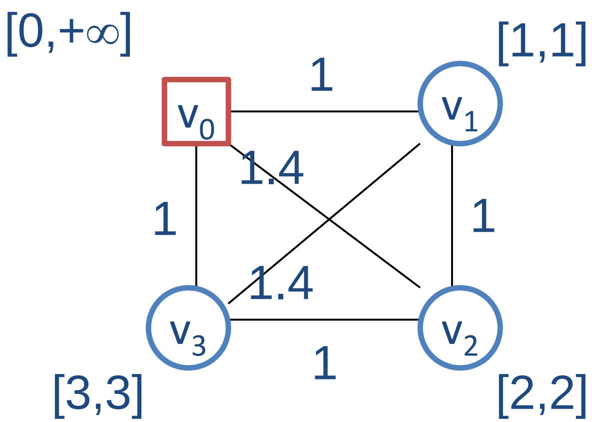
Optimal solution (cost = 5.4)
$\theta = (0;\ 0;\ 1;\ 1)$
$\lambda = (2;\ 1.4\ ;\ 2)$

Route $r_7 = (v_0, v_1, v_2, v_3, v_0)$ has a reduced cost -1.4
(reduced cost = 4 – 2 – 1.4 - 2 = -1.4)

Illustration on the previous example

- Iteration 3

Data



$d_i = 1$, $st_i = 0$
$Q = +\infty$
$U = +\infty$

$\Omega_3 = \{r_1, r_2, r_3, r_4, r_7\}$ with $r_7 = (v_0, v_1, v_2, v_3, v_0)$

Min $2\theta_1 + 2{,}8\theta_2 + 2\theta_3 + 3{,}4\theta_4$ $\mathbf{+4\theta_7}$
subject to

$$\begin{cases} \theta_1 + \theta_4 + \theta_7 \geq 1 \\ \theta_2 + \theta_4 + \theta_7 \geq 1 \\ \theta_3 + \theta_7 \geq 1 \\ \theta_1, \ldots, \theta_4, \theta_7 \geq 0 \end{cases}$$

Max $\lambda_1 + \lambda_2 + \lambda_3$
subject to

$$\begin{cases} \lambda_1 \leq 2 \\ \lambda_2 \leq 2{,}8 \\ \lambda_3 \leq 2 \\ \lambda_1 + \lambda_2 \leq 3{,}4 \\ \lambda_1 + \lambda_2 + \lambda_3 \leq 4 \\ \lambda_1, \ldots, \lambda_3 \geq 0 \end{cases}$$

Optimal solution (cost = 4)
$\theta = (0; 0; 0; 0; 1)$
$\lambda = (1; 2; 1)$

No route with a negative reduced cost exists: solution θ is also optimal for MP

Remarks

- Equivalently, a variable with negative reduced cost is associated with a violated constraint in the dual program of MP(Ω_t)

$$
\begin{aligned}
&(D(\Omega_1)) \qquad \text{maximize} \sum_{v_i \in V\setminus\{v_0\}} \lambda_i + U \times \lambda_0 \\
&\text{subject to} \\
&\qquad \sum_{v_i \in V\setminus\{v_0\}} a_i^k \lambda_i + \lambda_0 \le c_k \qquad (r_k \in \Omega_1), \\
&\qquad \lambda_i \ge 0 \qquad (v_i \in V \setminus \{v_0\}), \\
&\qquad \lambda_0 \le 0.
\end{aligned}
$$

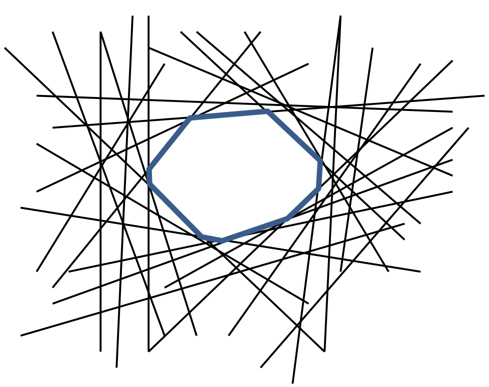
- Adding a column amounts to adding a violated constraint in the restricted dual program

Remarks

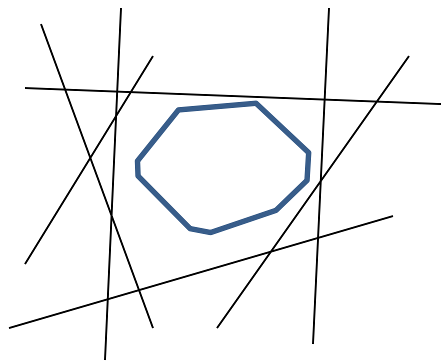
- At each iteration, the solution of MP(Ω_t) provides a feasible primal solution and non-necessarily feasible dual solution (with the same cost). If the dual solution is feasible, they are both optimal (weak duality theorem)

Remarks

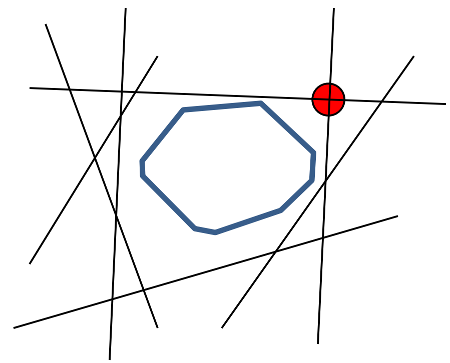
- Dual point of view



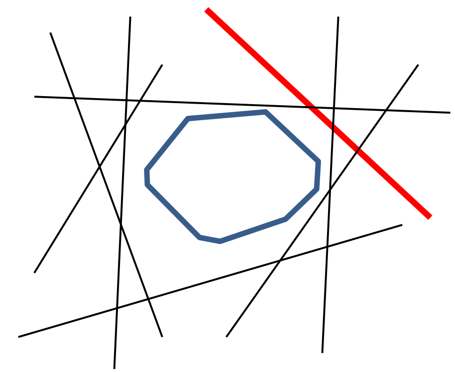
Dual polyhedron for MP



Dual polyhedron for MP(Ω_t)



Optimal dual solution at iteration t



Dual polyhedron for MP(Ω_{t+1})

- Equivalent to Kelley's algorithm for convex nonlinear programming

Remark

- Having generated the columns of the optimal solution is not necessarily sufficient for the algorithm to stop

- Illustration (initial set of column)

$\Omega_1 = \{r_7\}$ with $r_7 = (v_0, v_1, v_2, v_3, v_0)$

Min $4\theta_7$
subject to

$$\begin{cases} \theta_7 \geq 1 \\ \theta_7 \geq 1 \\ \theta_7 \geq 1 \\ \theta_7 \geq 0 \end{cases}$$

Max $\lambda_1 + \lambda_2 + \lambda_3$
subject to

$$\begin{cases} \lambda_1 + \lambda_2 + \lambda_3 \leq 4 \\ \lambda_1, \ldots, \lambda_3 \geq 0 \end{cases}$$

Remark

- Illustration (first iteration)

$\Omega_1 = \{r_7\}$ with $r_7 = (v_0, v_1, v_2, v_3, v_0)$

Min $4\theta_7$
subject to

$$\begin{cases} \theta_7 \geq 1 \\ \theta_7 \geq 1 \\ \theta_7 \geq 1 \\ \theta_7 \geq 0 \end{cases}$$

Max $\lambda_1 + \lambda_2 + \lambda_3$
subject to

$$\begin{cases} \lambda_1 + \lambda_2 + \lambda_3 \leq 4 \\ \lambda_1, \ldots, \lambda_3 \geq 0 \end{cases}$$

Optimal solution (cost = 4)
$\theta = (1)$
$\lambda = (4;0;0)$

Route $r_1 = (v_0, v_1, v_0)$ has a reduced cost -2

Remark

- Illustration (second iteration…)

$\Omega_2 = \{r_7, r_1\}$ with $r_1 = (v_0, v_1, v_0)$

Min $4\theta_7$
subject to

$$\begin{cases} \theta_7 + \theta_1 \geq 1 \\ \theta_7 \geq 1 \\ \theta_7 \geq 1 \\ \theta_7, \theta_1 \geq 0 \end{cases}$$

Max $\lambda_1 + \lambda_2 + \lambda_3$
subject to

$$\begin{cases} \lambda_1 + \lambda_2 + \lambda_3 \leq 4 \\ \lambda_1 \leq 2 \end{cases}$$
$$\lambda_1, \ldots, \lambda_3 \geq 0$$

Optimal solution (cost = 4)
$\theta = (1;0)$
$\lambda = (0;4;0)$

Route (v_0, v_2, v_0) has a negative reduced cost -1.2

Basics of column generation

IMPLEMENTATION

Initial set of columns: efficiency

- A good initial set of columns is a set of columns that help limiting oscillations of dual variables
  - However, the initial set of columns doesn't necessarily have a strong impact on the efficiency of the method
  - Actually, in first iterations, "good" columns can usually be found quickly

- Example of initial sets
  - Columns obtained from a heuristic solution
  - $\{ (v_0,v_1,v_0) , \dots , (v_0,v_n,v_0) \}$

- Preventing from dual oscillations is also the subject of stabilization techniques (see later)

Initial set of columns: feasibility

- A feasible linear program is needed to start the algorithm
- If it is difficult to obtain a set of columns that ensures feasibility, artificial variables can be added
- Artificial variables can be viewed as subcontracted services
- Examples:
  - a high cost route that serves all customers
  - High cost routes that visit each a single customer and that do not appear in the fleet size constraint

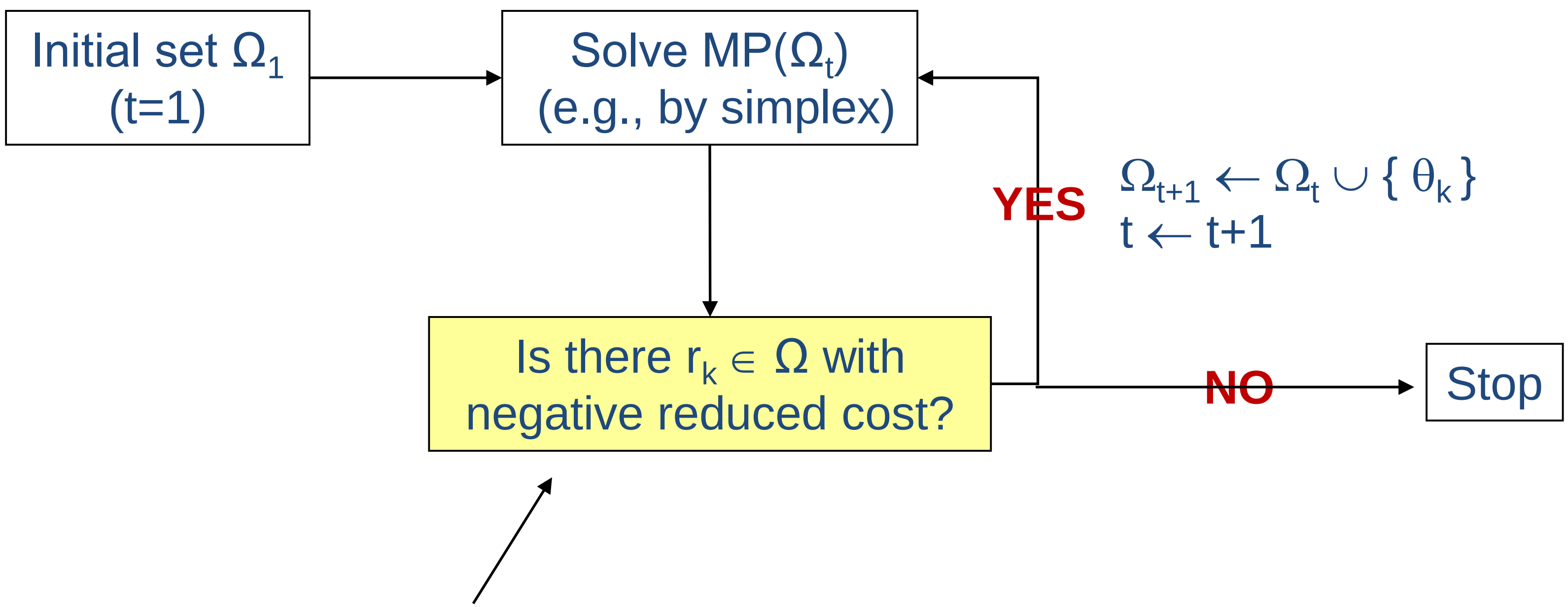
Other remarks

- A usual practice is to generate several columns at each iteration
  - Save iterations
- If too many columns have been generated, columns that never enter the basis can be removed (with the sole risk that they may be generated again later)
  - Save solution time for MP(Ω_t)
  - Usually useless for vehicle routing problems
- Be careful with numerical imprecision
  - Risk of being trapped in the repeated generation of the same column with reduced cost that looks like -0.00000001

Basics of column generation

PRICING PROBLEM

Recall of the column generation scheme

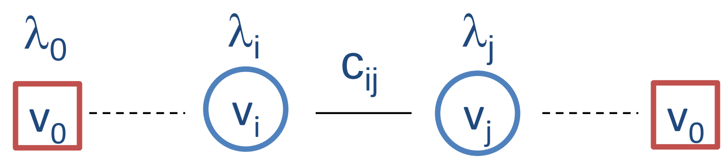


Pricing problem
(or subproblem, or slave problem or oracle)

Reformulation of the pricing problem

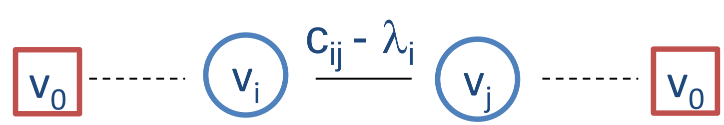
- Find $r_k \in \Omega$ such that

$$c_k - \sum_{v_i \in V \setminus \{v_0\}} a_i^k \lambda_i - \lambda_0 < 0.$$



- Equivalently, find $r_k \in \Omega$ such that

$$\sum_{(v_i, v_j) \in A} b_{ij}^k (c_{ij} - \lambda_i) < 0.$$

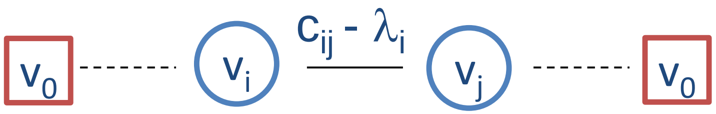


with b^k_{ij} =1 when arc (v_i, v_j) belongs to route r_k

Reformulation of the pricing problem

- Can be expressed as the following combinatorial optimization problem:

  Find the shortest path in the graph G, with arc costs $c_{ij} - \lambda_i$, from v_0 to v_0, subject to capacity and time windows constraints, such that no vertex is traversed more than once

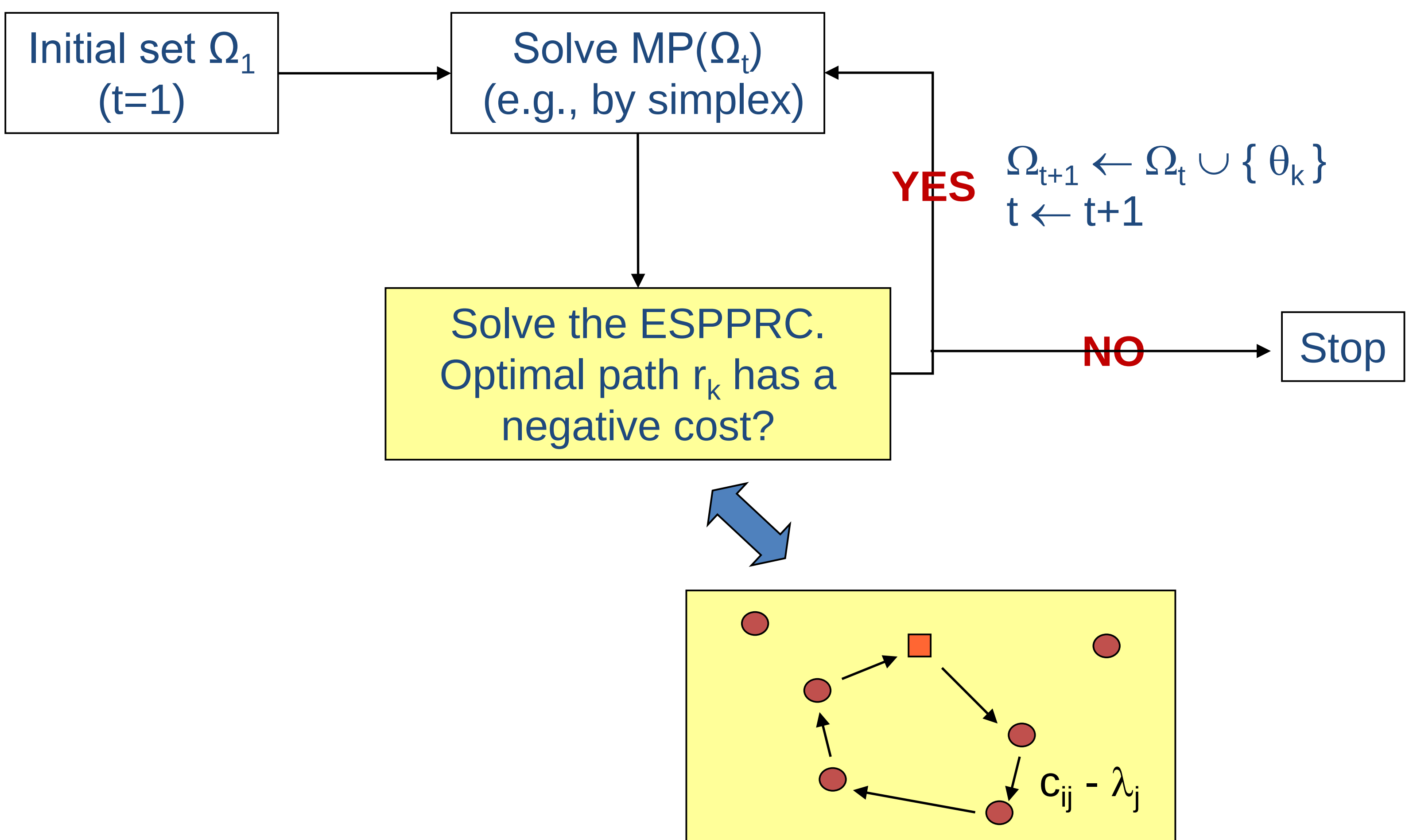


- This problem is known as the **Elementary Shortest Path Problem with Resource Constraint** (ESPPRC)

Remarks

- The ESPPRC is NP-hard in the strong sense
- It is usually solved with Dynamic Programming
  - Other possibilities: branch-and-cut, Constraint Programming…
- The optimal solution is not needed; one can stops as soon as one (or a “sufficient” number) of paths with negative costs are found
- One can first search for good solutions with a heuristic (e.g., tabu search)
- One can exploit the fact that paths from the current basis have a cost equal to zero

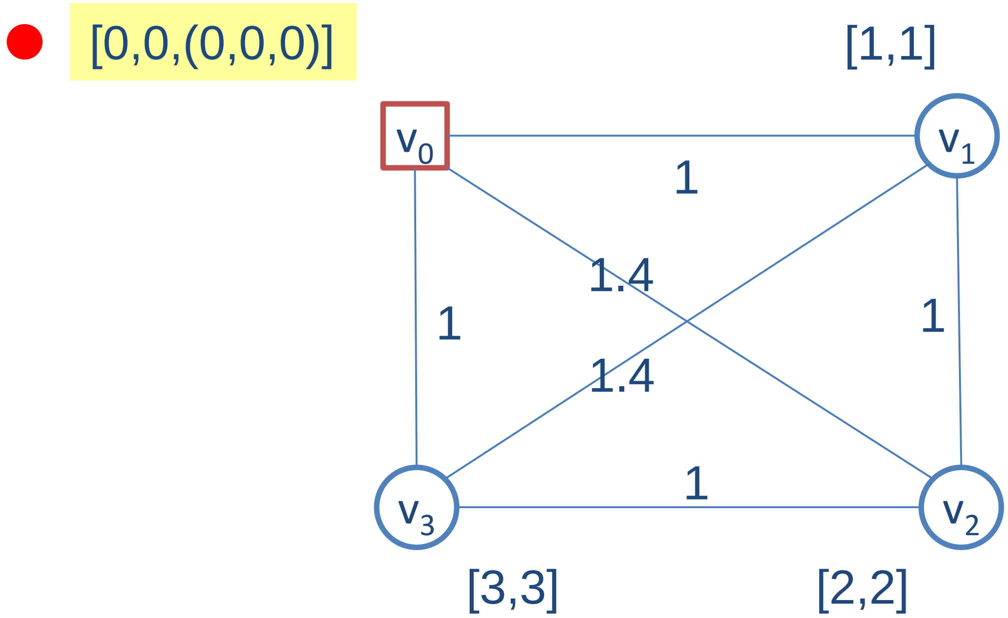
Complete column generation scheme



SOLUTION OF THE ESPPRC

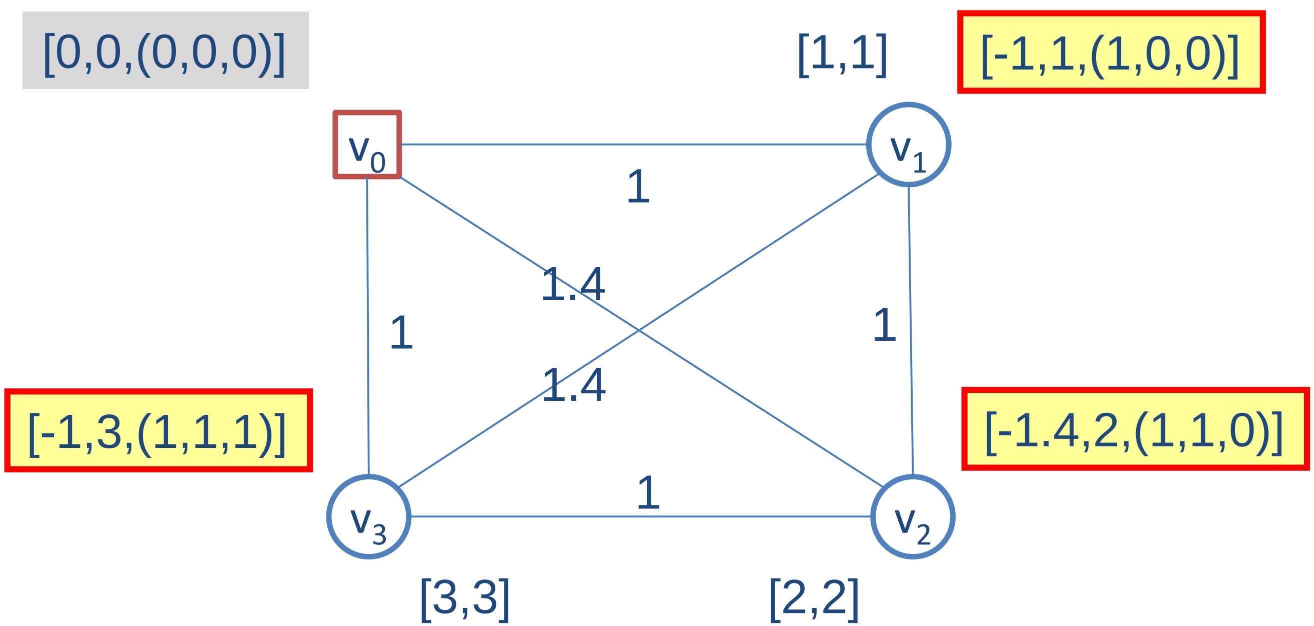
Dynamic programming algorithm

No capacity constraints
λ=(2; 2.8; 2)
Labels: [cost,time,(v_1,v_2,v_3)]



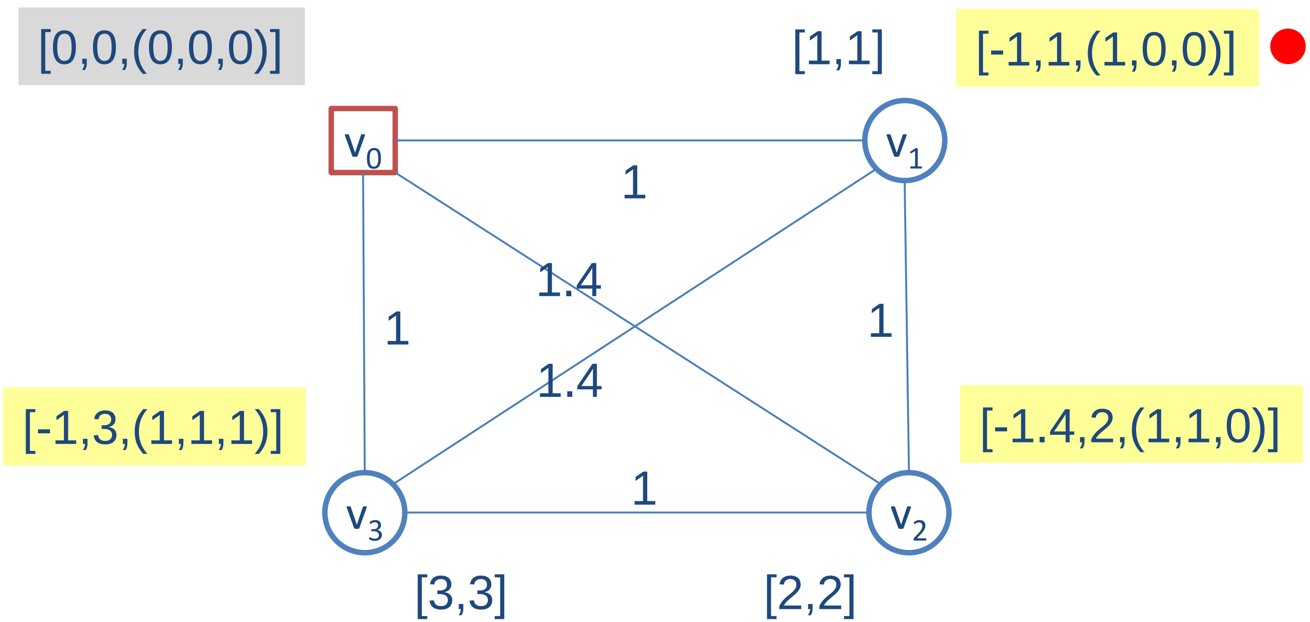
Dynamic programming algorithm

No capacity constraints
λ=(2; 2.8; 2)
Labels: [cost,time,(v_1,v_2,v_3)]



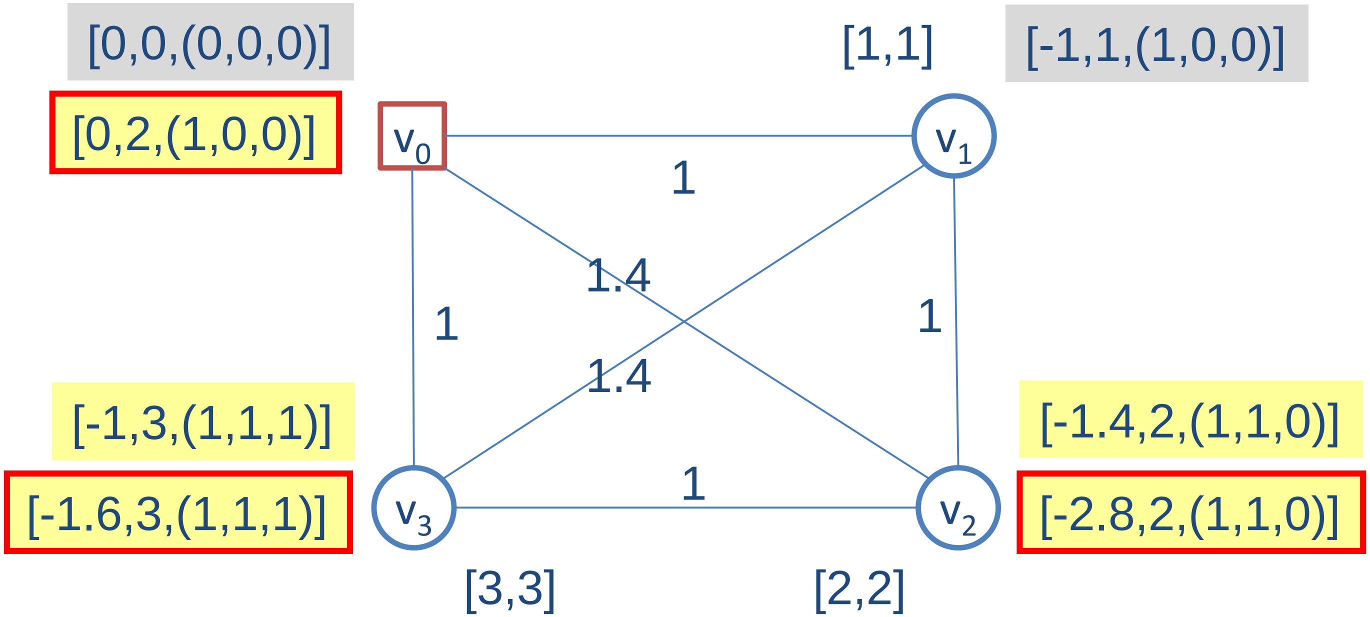
Dynamic programming algorithm

No capacity constraints
λ=(2; 2.8; 2)
Labels: [cost,time,(v_1,v_2,v_3)]



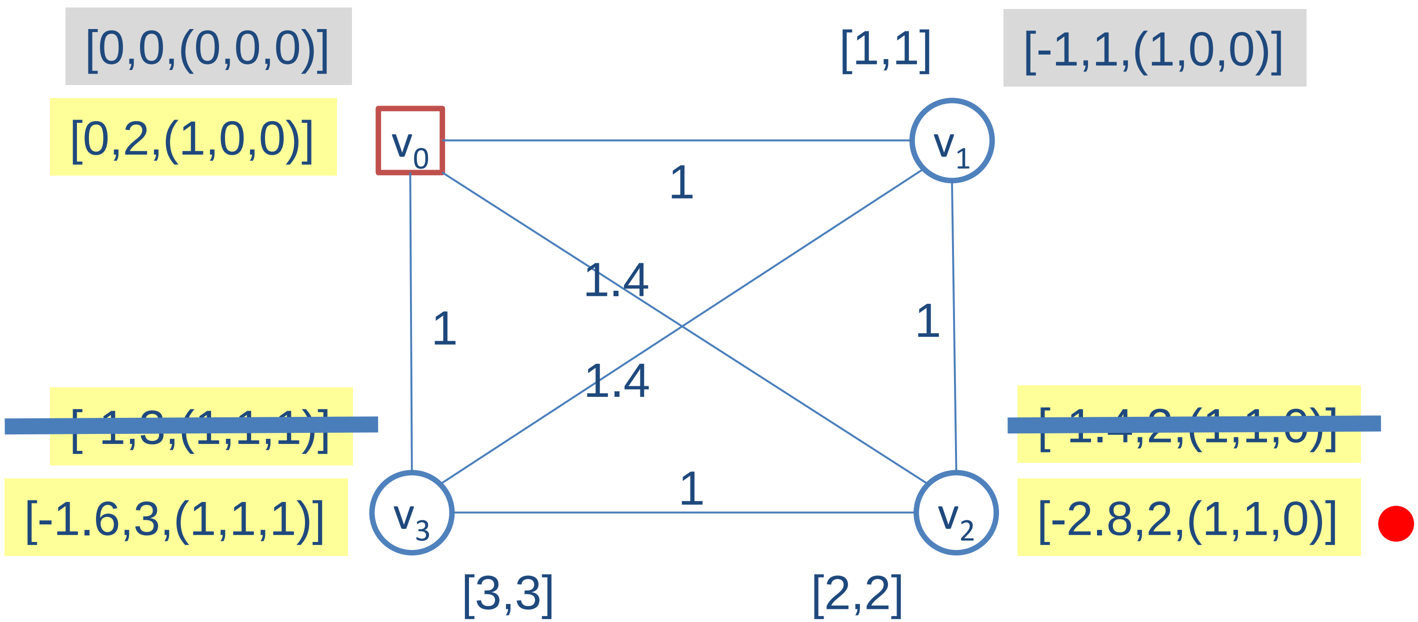
Dynamic programming algorithm

No capacity constraints
λ=(2; 2.8; 2)
Labels: [cost,time,(v_1,v_2,v_3)]



Dynamic programming algorithm

No capacity constraints
λ=(2; 2.8; 2)
Labels: [cost,time,(v_1,v_2,v_3)]



Basics of column generation

BRANCH-AND-PRICE

Why branch-and-price?

- Recall that column generation is just a method to solve a linear program

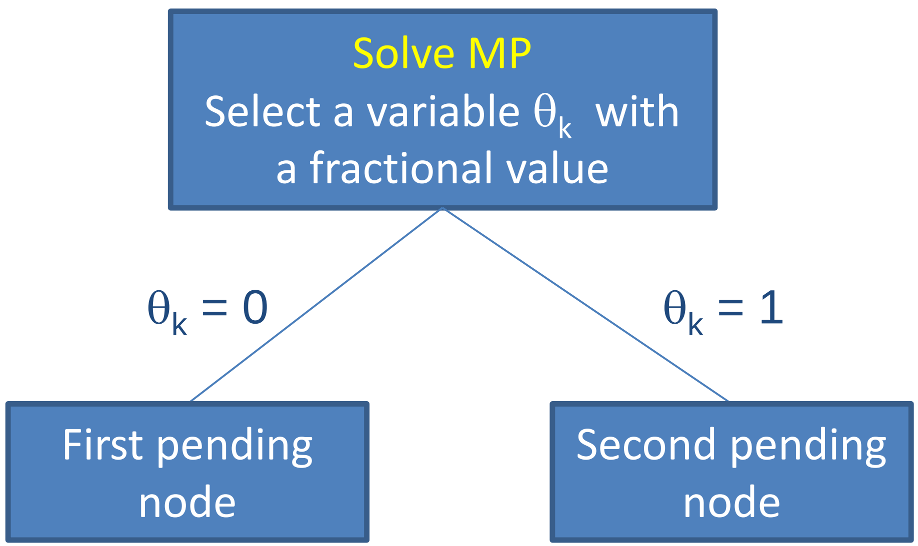
- It is embedded in branch-and-bound to solve the integer program
  - At each node of the search tree (including the root node), column generation is used to compute the LP relaxation

- The name branch-and-price just emphasizes the fact that column generation is applied at each node

- The main issue with branch-and-price is that one has to be careful about the way separation is applied

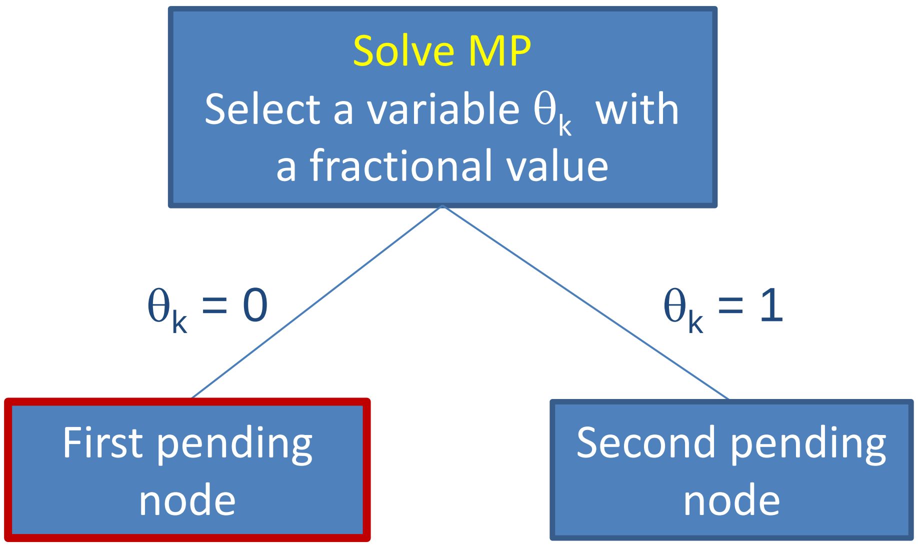
Separation rule

- Standard separation rule in branch-and-bound



Separation rule

- Standard separation rule in branch-and-bound



Master problem: remove θ_k (or fix $\theta_k = 0$)
Pricing problem: forbid path r_k (ESPPRC with forbidden paths)

Separation rule

- Standard separation rule in branch-and-bound

Solve MP
Select a variable θ_k with a fractional value

$\theta_k = 0$ → First pending node

$\theta_k = 1$ → Second pending node

Master problem: fix θ_k =1, remove (or fix to 0) all other columns where a customer from route r_k is visited
Pricing problem: remove customers from route r_k from the graph

Separation rule

- Standard separation rule in branch-and-bound

Solve MP
Select a variable θ_k with a fractional value

$\theta_k = 0$ | $\theta_k = 1$

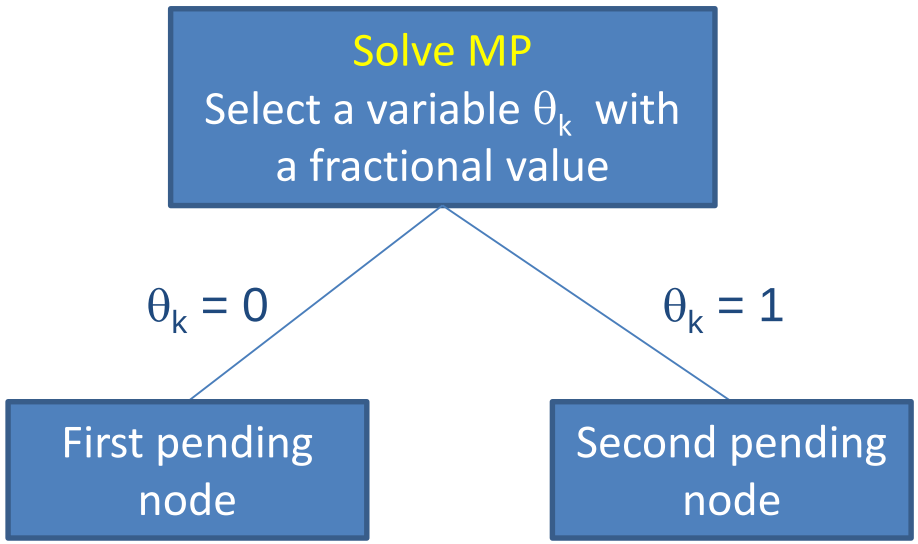
First pending node | Second pending node

HARD: solution of ESPPRC with forbidden paths

EASY: amounts to removing vertices from the instance

Separation rule

- Standard separation rule in branch-and-bound



+ Inefficient : strong imbalance of the search tree

Separation rule

- Usual separation rule

Solve MP
Select an arc (v_i,v_j) with a fractional flow

forbid the use of arc (v_i,v_j) → First pending node

impose to use arc (v_i,v_j) → Second pending node

- It can be shown that in any fractional solution an arc with a fractional flow exists

Separation rule

- Usual separation rule

Solve MP
Select an arc (v_i,v_j) with a fractional flow

forbid the use of arc (v_i,v_j)

impose to use arc (v_i,v_j)

First pending node

Second pending node

Master problem: remove (or fix to 0) all columns that use arc (v_i,v_j)
Pricing problem: remove arc (v_i,v_j) from the graph

Separation rule

- Usual separation rule

Solve MP
Select an arc (v_i,v_j) with a fractional flow

forbid the use of arc (v_i,v_j) | impose to use arc (v_i,v_j)

First pending node | Second pending node

Master problem: remove (or fix to 0) all columns that use an arc (v_i,v_k) with $k \neq j$ or an arc (v_k,v_j) with $k \neq i$
Pricing problem: remove arcs (v_i,v_k) with $k \neq j$ and arcs (v_k,v_j) with $k \neq i$ from the graph

Separation rule

- Usual separation rule

Solve MP
Select an arc (v_i,v_j) with a fractional flow

forbid the use of arc (v_i,v_j)

impose to use arc (v_i,v_j)

First pending node

Second pending node

Easy + efficient

Remarks

- It is generally admitted that in branch-and-price one should branch on the variables of the compact formulation
- It is possible to add constraints in the Master Problem when branching
    - The new dual variables then have to be considered when computing the reduced costs in the pricing problem
    - (dumb) Example

Impose arc (v_i,v_j) → Add constraint $\sum b^k_{ij}\ \theta_k \geq 1$ in the master problem **Imply new dual variable λ_{ij}** → Set cost of arc (v_i,v_j) to $\mathbf{c_{ij} - \lambda_l - \lambda_{ij}}$ in the pricing problem

Remarks

- One can start by branching on the number of vehicles (if it is fractional)
  - Usually, the impact on the lower bound is very strong
  - The risk is to impose a maximal value that is too small (unfeasible solution) and that the algorithm spends a long time to close the node

Basics of column generation

CONCLUSION

Summary

- The extended formulation gives a far better lower bound than the compact formulation
- It is theoretically obtained from the compact formulation through Dantzig-Wolfe decomposition
- Column generation is needed to compute its linear relaxation
- It implies solving repeatedly NP-hard pricing problems (ESPPRC)

Other comments

- Branch-and-price is very **generic**
  - Application to different Vehicle Routing Problems only imply different resource constraints in the pricing problem

- Most of the computing time is spent when solving the pricing problem
  - **Way of improvement 1**: accelerate solution time of the pricing problem,
  - **Way of improvement 2**: reduce the number of iterations
    - Number of iterations per node: generation of good sets of columns at each iteration, stabilization techniques
    - Number of nodes: add valid inequalities (branch-price-and-cut)

Accelerate solution time for the pricing problem

- Accept non-elementary routes
  - The pricing problem becomes weakly NP-hard
  - The quality of the LP bound may decrease a lot…
- Accept some non-elementary routes
  - Accept routes without 2-cycles, 3-cycles…
  - Ng-routes
    - Ng-set(i): subset of customers that vertex i is able to “remember”
    - Memory(L): memory of label L
    - A label cannot be extended to a vertex in its memory
  - Dynamic relaxation

Improve the relaxation with valid inequalities

- Subset-row inequalities
  - Use a set-partitioning formulation
  - Find r_1, r_2, r_3 such that
    - r_1 visits i_1 and i_2
    - r_2 visits i_1 and i_3
    - r_3 visits i_2 and i_3
    - $\theta_1 + \theta_2 + \theta_3 \geq 1$
  - Add valid inequality $\theta_1 + \theta_2 + \theta_3 \leq 1$
  - But the new dual variable complicates the pricing problem…

Other comments

- Typical statistics for column generation applied to the VRPTW:
  - solve instances with less than 100 customers (up to to 200 for advanced implementations)
  - a few nodes in the search tree
  - several thousand columns generated
  - several hundred iterations

Some references

R. Baldacci, P. Toth, and D. Vigo. Recent advances in vehicle routing exact algorithms. *4OR*, 5(4):269–298, 2007.

C. Barnhart, C.A. Hane, and P.H. Vance. Using Branch-and-Price-and-Cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, 48(2):318–326, 2000.

G. Desaulniers, J. Desrosiers, and M.M. Solomon, editors. *Column generation*. GERAD 25th Anniversary Series. Springer, 2005.

G. Desaulniers, F. Lessard, and A. Hadjar. Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows. *Transportation Science*, 42(3):387–404, 2008.

M. Desrochers, J. Desrosiers, and M.M. Solomon. A new optimization algorithm for the Vehicle Routing Problem with Time Windows. *Operations Research*, 40(2):342–354, 1992.

D. Feillet. A tutorial on column generation and branch-and-price for vehicle routing problems. *4*, 8(4):407–424, 2010.

M.E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.