# Finding bicriteria shortest pairs of disjoint paths: algorithms and related problems[1]

Marta Pascoal, M. Eugénia Captivo, João Clímaco

INESC - Coimbra, University of Coimbra and CIO, Universidade de Lisboa
Portugal

Vienna, September 2014

[1]Partially supported by FCT under project grant PEst-OE/EEI/UI308/2014.

# Contents

# Abstract

Pairs of disjoint paths between two nodes in a network provide a primary path as well as a backup path, such that the first can be used in case one of the arcs of the second fails. In this talk we address the bicriteria shortest pair of disjoint paths problem and some related problems, and discuss exact algorithms to find their non-dominated solutions.

# Motivation

1

- Sometimes it is convenient to know an alternative to the shortest path, in case a failure occurs along the chosen primary path. They should not share physical resources.
- In telecommunications, whenever it justifies to have a backup path.
- In transportation, in order to avoid over-exposure of the same region to hazardous materials.

2

- Additionally, some applications associate each arc with a type of service, which require types not to be repeated in the main and the backup paths, in order to prevent possible failures.

# Something old

# The shortest pair of disjoint paths problem

## Definitions

- $G = (N, A)$: network, $A \subseteq N \times N$, $|N| = n$, $|A| = m$.
- $c_{ij} \in \mathbb{R}$: cost associated with $(i, j) \in A$.
- $s \in N$ ($t \in N$): source (terminal) node.
- Given a path $p$, the set of nodes (arcs) in $p$ is $V(p)$ ($A(p)$).
- $p \diamond q$ denotes the concatenation of paths $p$ and $q$.

## Assumptions

- No parallel arcs.
- No cycles with a negative cost.

# The shortest pair of disjoint paths problem

Let $s, t \in N$, the set of pairs of disjoint paths from $s$ to $t$ is

$$L = \left\{ x \in \{0,1\}^m : \sum_{(i,j) \in \Gamma^+(i)} x_{ij} - \sum_{(j,i) \in \Gamma^-(i)} x_{ji} = b_i, \forall i \in N, \forall (i,j) \in A \right\}$$

with

$$b_i = \begin{cases} 2 & \text{if } i = s \\ 0 & \text{if } i \in N - \{s,t\} \\ -2 & \text{if } i = t \end{cases}$$

and $x_{ij} = 1$ iff $(i,j) \in A$ belongs to a pair of disjoint paths.
The constraints matrix is totally unimodular and $b$ is integral, thus $L$ is

$$\left\{ x \in \mathbb{R}^m : \sum_{(i,j) \in \Gamma^+(i)} x_{ij} - \sum_{(j,i) \in \Gamma^-(i)} x_{ji} = b_i, \forall i \in N \wedge 0 \leq x_{ij} \leq 1, \forall (i,j) \in A \right\}$$

# The shortest pair of disjoint paths problem

The problem of finding a shortest pair of disjoint paths from $s$ to $t$ (SPDP) can be formulated as:

$$\begin{array}{ll} \text{minimize} & \sum_{(i,j)\in A} c_{ij} x_{ij} \\ \text{subject to} & x \in L \end{array}$$

This is a minimum cost flow problem in a network, where:

- the capacity upper bound of $(i, j)$ is $u_{ij} = 1$,
- $b_s = 2,\ b_t = -2,\ b_i = 0,\ \forall i \in N - \{s, t\}$.

# Finding the shortest pair of disjoint paths

## Suurballe's algorithm

Thus, the SPDP can be found by considering $x = 0$, and

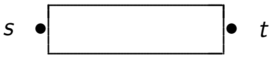
1. finding a shortest path in $G$, and
2. finding a shortest path in a residual network $G[x] = (N, A_x, c^x)$, s.t. $A_x = A_+ \cup A_-$, and
   - $A_+ = \overbrace{\{(i,j) : (i,j) \in A \wedge x_{ij} = 0\}}^{c^x_{ij} = c_{ij}}$
   - $A_- = \overbrace{\{(j,i) : (i,j) \in A \wedge x_{ij} = 1\}}^{c^x_{ji} = -c_{ij}}$

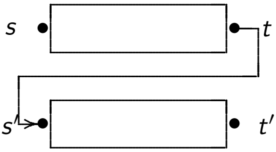
2 shortest path problems are solved $\Rightarrow$ $O(m + n \log n)$ time

# Ranking shortest pair of disjoint paths

New network $(N', A')$ [Clímaco & Pascoal, 2009]:

- Nodes are duplicated: $N' = N \cup \{i' : i \in N\}$.
- Arcs are duplicated and a new one is added, from $t$ to $s'$: $A' = A \cup \{(i', j') : (i, j) \in A\} \cup \{(t, s')\}$.
- Initial node: $s$, Terminal node: $t'$.
- Old costs are maintained. New costs: $c_{i'j'} = c_{ij}$, if $(i, j) \in A$, and $c_{ts'} = 0$.



# Ranking shortest pair of disjoint paths

## Paths in $(N', A')$

- Each path $p$ from $s$ to $t'$ has the form:

$$p = q \diamond (t, s') \diamond q',$$

  where $q$ is a path from $s$ to $t$ and $q'$ is a path from $s'$ to $t'$.
- If $q$ and $q'$ are simple and $q \cap q' = \emptyset$, then $q, q'$ correspond to a pair of disjoint paths in $(N, A)$ (maybe simple).

## Algorithm

A method for ranking simple paths can be applied to the modified topology, in order to compute disjoint pairs of paths by order of cost.

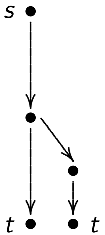
# Ranking shortest pair of disjoint paths

## Deviation algorithm [Martins et al., 1999]

Let $X$ be a set of candidates to the next $p_k$, $k \in \{1, \ldots, K\}$.

- $X$ is initialised with $p_1$.
- Repeat:
  - $p_k$ is the shortest simple path in $X$,
  - analyse $p_k$ and generate new low cost candidates from it,
  - store the candidates generated in $X$.

New candidates are deviations.



# Ranking shortest pair of disjoint paths

The shortest path from $s$ to $t$ that deviates from $p_k$ at $v_i$ is:

$$\mathrm{sub}_{p_k}(s, v_i) \diamond (v_i, j) \diamond \mathcal{T}_t(j),$$

s. t. $(v_i, j)$ doesn't belong to any of the candidates computed. Besides:

- $\mathcal{T}_t(j)$ can be computed when the algorithm starts,
- the access to $(v_i, j)$ can be immediate, and
- $j \notin \mathrm{sub}_{p_k}(s, v_i)$, $h(j) \notin \mathrm{sub}_{p_k}(s, v_i) \cap \mathcal{N}$ if $j \in \mathcal{N}'$ where $h(x') = x$ for any $x \in \mathcal{N}$.

Check if deviations correspond to disjoint pairs of paths!

**Something new**

# The shortest pair of arc/label disjoint paths problem

## Definitions

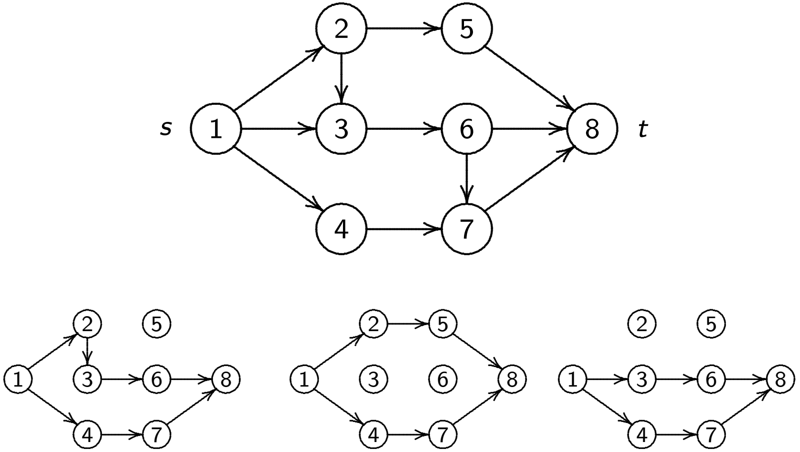
- $c_{ij} \in \mathbb{R}$: cost associated with $(i,j) \in A$;
- $l_{ij} \in \mathcal{L} = \{1, \ldots, \ell\}$: label associated with $(i,j) \in A$.

## Decision variables

- $x_{ij}^k = \begin{cases} 1 \text{ if arc } (i,j) \text{ is in the path } k \\ 0 \text{ otherwise} \end{cases}, \quad \forall (i,j) \in \mathcal{A}$

- $v_l^k = \begin{cases} 1 \text{ if label } l \text{ is in some arc of the path } k \\ 0 \text{ otherwise} \end{cases}, \quad \forall l \in \mathcal{L}$

# The shortest pair of arc/label disjoint paths problem



# The shortest pair of arc/label disjoint paths problem

$$\min \sum_{(i,j)\in\mathcal{A}} c_{ij} \sum_{k=1}^{2} x_{ij}^k$$

$$\text{s. t.} \sum_{j\in\Gamma^+(i)} x_{ij}^k - \sum_{j\in\Gamma^-(i)} x_{ji}^k = \begin{cases} 1, & i = s \\ 0, & i \in \mathcal{N} - \{s,t\} \\ -1, & i = t \end{cases} , \qquad k = 1,2$$

$$\sum_{k=1}^{2} x_{ij}^k \leq 1, \qquad (i,j) \in \mathcal{A}$$

$$\sum_{(i,j)\in\mathcal{A}_l} x_{ij}^k \leq \min\{n-1, |\mathcal{A}_l|\} v_l^k, \qquad l \in \mathcal{L},\ k = 1,2$$

$$\sum_{k=1}^{2} v_l^k \leq 1, \qquad l \in \mathcal{L}$$

$$x_{ij}^k \in \{0,1\}, \qquad (i,j) \in \mathcal{A},\ k = 1,2$$

$$v_l^k \in \{0,1\}, \qquad l \in \mathcal{L},\ k = 1,2$$

# The shortest pair of arc/label disjoint paths problem

1. $$\sum_{j \in \Gamma^+(i)} x_{ij}^k - \sum_{j \in \Gamma^-(i)} x_{ji}^k = \begin{cases} 1, & i = s \\ 0, & i \in \mathcal{N} - \{s,t\} \\ -1, & i = t \end{cases}, \quad k = 1,2$$
   ensure flow conservation for all nodes and that paths start at $s$ and end at $t$.
2. $\sum_{k=1}^{2} x_{ij}^k \leq 1, \quad (i,j) \in \mathcal{A}$ assure disjointness.
3. $$\sum_{(i,j) \in \mathcal{A}_l} x_{ij}^k \leq \min\{n-1, |\mathcal{A}_l|\} v_l^k, \quad l \in \mathcal{L}, \quad k = 1,2$$
   assure that for each label and each path, an arc with that label can only be in the solution if the associated label variable is 1.
   It also implies that the number of arcs in each path in the solution with each label cannot exceed $n-1$, neither the number of arcs with that label in the network.

# The shortest pair of arc/label disjoint paths problem

4. $\sum_{k=1}^{2} v_l^k \leq 1, l \in \mathcal{L}$
   assure that each labels appears in at most one of the two paths.
5. $x_{ij}^k \in \{0,1\}, (i,j) \in \mathcal{A},\ k = 1,2$
   $v_l^k \in \{0,1\}, l \in \mathcal{L},\ k = 1,2$
   are the integrality constraints.

# The shortest pair of arc/label disjoint paths problem

## Remarks

1. Label disjoint paths ⇒ arc disjoint paths. Therefore, we can ommit

$$\sum_{k=1}^{2} x_{ij}^{k} \leq 1, \quad (i,j) \in \mathcal{A}$$

2. It can be shown that there is always an integral optimal solution $x$, even if

$$x_{ij}^{k} \in \{0,1\}, \ (i,j) \in \mathcal{A}, \ k = 1,2$$

are ommited. Therefore, these can be replaced by

$$0 \leq x_{ij}^{k} \leq 1, \ (i,j) \in \mathcal{A}, \ k = 1,2$$

# The shortest pair of arc/label disjoint paths problem

## Comparison (CPU times in sec.s)

Computer: Dual Core AMD, 2.7 GHz, 4 Gb of RAM, Suse 10.3.

| $n$ | $d$ | $\ell$ | F | FL | S | SL |
|---|---|---|---|---|---|---|
| 100 | 5 | 5 | 0.070 | 0.057 | 0.070 | 0.043 |
| 100 | 5 | 10 | 0.057 | 0.038 | 0.052 | 0.030 |
| 100 | 5 | 20 | 0.034 | 0.023 | 0.035 | 0.021 |
| 100 | 10 | 5 | 0.129 | 0.099 | 0.096 | 0.068 |
| 100 | 10 | 10 | 0.087 | 0.057 | 0.063 | 0.046 |
| 100 | 10 | 20 | 0.052 | 0.027 | 0.083 | 0.047 |
| 100 | 20 | 5 | 0.213 | 0.187 | 0.138 | 0.123 |
| 100 | 20 | 10 | 0.135 | 0.097 | 0.098 | 0.067 |
| 100 | 20 | 20 | 0.116 | 0.101 | 0.102 | 0.086 |
| 200 | 5 | 5 | 0.116 | 0.066 | 0.090 | 0.047 |
| 200 | 5 | 10 | 0.055 | 0.051 | 0.043 | 0.035 |
| 200 | 5 | 20 | 0.052 | 0.035 | 0.053 | 0.029 |
| 200 | 10 | 5 | 0.206 | 0.146 | 0.136 | 0.103 |
| 200 | 10 | 10 | 0.205 | 0.180 | 0.142 | 0.118 |
| 200 | 10 | 20 | 0.168 | 0.134 | 0.136 | 0.117 |

| $n$ | $d$ | $\ell$ | F | FL | S | SL |
|---|---|---|---|---|---|---|
| 200 | 20 | 5 | 0.586 | 0.490 | 0.282 | 0.250 |
| 200 | 20 | 10 | 0.386 | 0.457 | 0.275 | 0.229 |
| 200 | 20 | 20 | 0.300 | 0.217 | 0.226 | 0.170 |
| 500 | 5 | 5 | 0.350 | 0.163 | 0.245 | 0.142 |
| 500 | 5 | 10 | 0.312 | 0.228 | 0.212 | 0.188 |
| 500 | 5 | 20 | 0.210 | 0.119 | 0.174 | 0.114 |
| 500 | 10 | 5 | 0.994 | 0.647 | 0.326 | 0.264 |
| 500 | 10 | 10 | 0.567 | 0.531 | 0.447 | 0.320 |
| 500 | 10 | 20 | 0.465 | 0.502 | 0.363 | 0.344 |
| 500 | 20 | 5 | 0.952 | 0.800 | 0.523 | 0.365 |
| 500 | 20 | 10 | 1.020 | 0.918 | 0.600 | 0.566 |
| 500 | 20 | 20 | 0.659 | 0.530 | 0.503 | 0.346 |

F: original model
FL: original model + relaxed $x$
S: shortened model
SL: shortened model + relaxed $x$

# The bicriteria shortest pair of arc/label disjoint paths problem

## Definitions

Under the previous conditions, but replacing $c_{ij}$ by

- $c^1_{ij}, c^2_{ij} \in \mathbb{R}$: costs associated with $(i,j) \in A$.

The bicriteria shortest pair of arc/label disjoint paths problem (BSPLDP) consists in:

$$\begin{array}{ll} \text{minimize} & c^1(x) = \sum_{(i,j)\in\mathcal{A}} c^1_{ij} \sum_{k=1}^{2} x^k_{ij}, \quad c^2(x) = \sum_{(i,j)\in\mathcal{A}} c^2_{ij} \sum_{k=1}^{2} x^k_{ij} \\ \text{subject to} & \text{the previous constraints} \end{array}$$

# The bicriteria shortest pair of arc/label disjoint paths problem

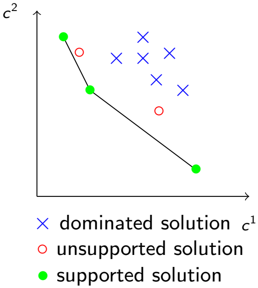
- A feasible solution $x$ is efficient *iff* there is no other $x'$ s.t. $c(x') \leq c(x)$ and $c(x') \neq c(x)$ (with $c(x) = (c^1(x), c^2(x))$).
- $c(x)$ is non-dominated *iff* $x$ is efficient.
- The Pareto frontier is formed by all efficient solutions.

Non-dominated solutions:

- supported: optimal solutions of a weighted sum problem (WSP)

$$\min_{x \in S} \left\{ w_1 c^1(x) + w_2 c^2(x) : \; w_1, w_2 \geq 0 \right\}$$

- the remaining are unsupported.



# Something “borrowed”

# Approach 1: 2-phases method

Two phases:

1. compute the supported solutions,
2. compute the unsupported solutions, by looking for solutions with images within the duality gaps formed by adjacent supported solutions.

Approaches for searching within duality gaps, [Current et al., 1990], [Ralphs, Saltzman & Wiecek, 2006]:

- using side constraints to obtain a sequence of "easier" problems,
- using an algorithm that ranks the $K$ best solutions of the problem,
- using a reference point approach, minimizing a Chebyshev metric,

. . .

# Approach 1: 2-phases method

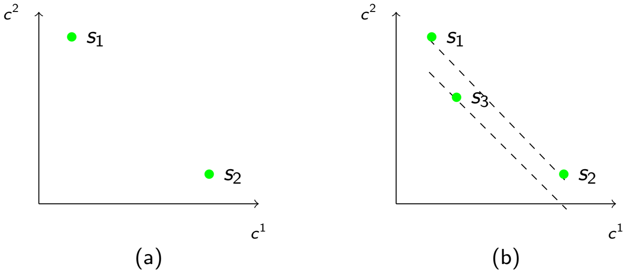
The WSM approach solves a sequence of WSPs and updates their parameters according to the solutions found (NISE method [Cohon, 1978]).

1. The lexicographic best solution wrt each criterion is calculated.
2. For each pair of consecutive solutions, $s_1$, $s_2$, a new one is calculated. The goal is to find further non-dominated solutions inside the gap defined by $c(s_1)$ and $c(s_2)$. The new solution optimizes

$$\begin{array}{ll} \min & w_1 c^1(s) + w_2 c^2(s) \\ \text{s. t.} & s \in S \end{array}$$

with $w_1 = c^2(s_1) - c^2(s_2)$, $w_2 = c^1(s_2) - c^1(s_1)$.

# Approach 1: 2-phases method




Phase 1, WSM for computing supported solutions:

(a) initial supported solutions;

(b) supported solution obtained from the previous pair of solutions.

# Approach 1: 2-phases method

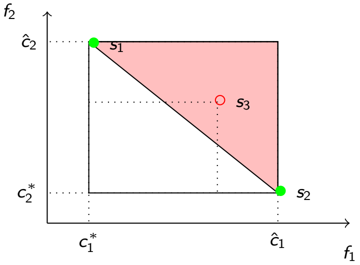
- Other solutions belong to the duality gaps formed by pairs of adjacent supported solutions.
- These unsupported solutions can be obtained by solving a new sequence of problems and updating the parameters defined by the search regions according to their solutions.
  Given $s_1, s_2$ adjacent non-dominated solutions, the method solves

$$\begin{array}{ll} \min & \max\{w_1|c^1(s) - c_1^*|, w_2|c^2(s) - c_2^*|\} \\ \text{s. t.} & s \in S \end{array}$$

with $c_i^* = c^i(s_i)$, $i = 1, 2$, which depends on the search region.
Equivalently, we have the LP

$$\begin{array}{lrcl} \min & v & & \\ \text{s. t.} & w_1 f_2(s) - v & \leq & w_1 f_2(s_2) \\ & w_2 f_1(s) - v & \leq & w_2 f_1(s_1) \\ & s \in S & & \end{array}$$

# Approach 1: 2-phases method




(a) (b)

Phase 2, computing unsupported solutions:

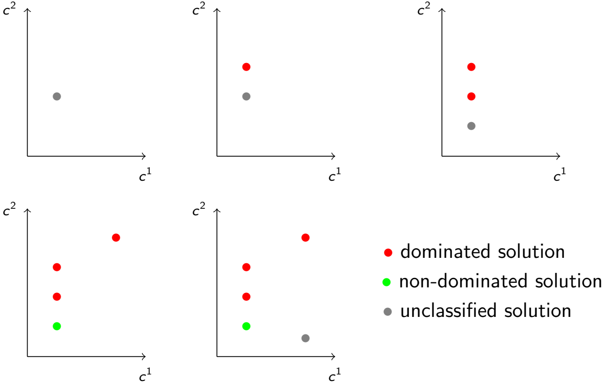
(a) initial search region;

(b) regions to scan after the unsupported solution $s_3$ is calculated in search for possible non-dominated solutions.

# Approach 2: ranking based method

Initially for the bicriteria path problem [Clímaco & Martins, 1982]:

- Solutions are ranked by order of $c^1$ (or $c^2$).
- A test filters the dominated solutions.



# Approach 2: ranking based method

## Procedure

- $\hat{c}_1 \leftarrow$ best $c_1$ value of a solution
- $c_2^* \leftarrow$ worst $c_2$ value of a solution
- $m_2 \leftarrow \hat{c}_2 \leftarrow$ best $c_2$ value of a solution
- $M_1 \leftarrow c_1^* \leftarrow$ worst $c_1$ value of a solution
- To rank solutions by order of $c_1$ and filter non-dominated solutions by means of a dominance test

This method can be extended to other problems, provided a ranking algorithm is known.

# Approach 2: ranking based method

## Ranking algorithm

The algorithm by [Clímaco & Pascoal, 2009] can be adapted.

In order to ensure that the pairs of disjoint paths are label disjoint:

- The selection of the deviation arc is restricted, so that the labels in the 2 paths are different.
- Candidates with repeated labels are discarded, after use.

# Machine and instances

Code in C language, running over Suse 10.3.

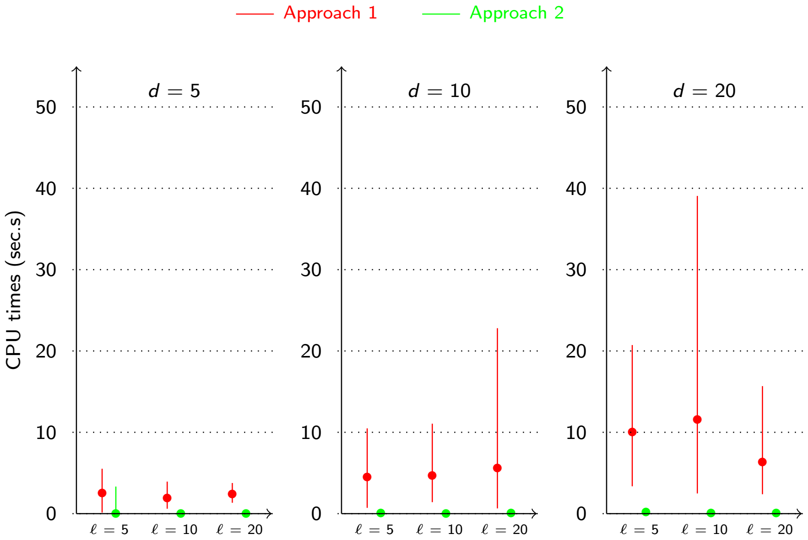
Computer characteristics:

- Dual Core AMD
- Processor 2.7 GHz
- 4 Gbytes of RAM

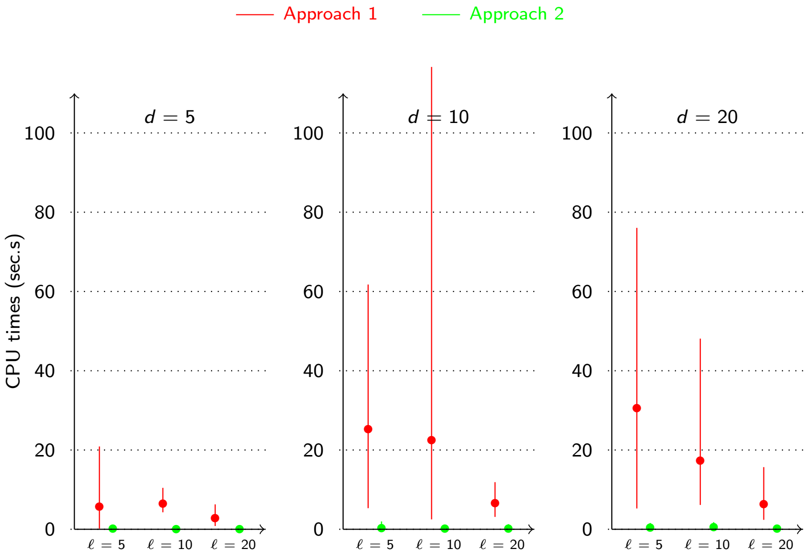
Random networks (generated with 10 seeds):

- $n \in \{100, 200, 50\}$
- $d \in \{5, 10, 20\}$
- $c_{ij}^1, c_{ij}^2$ randomly generated in $\{0, \dots, 100\}$

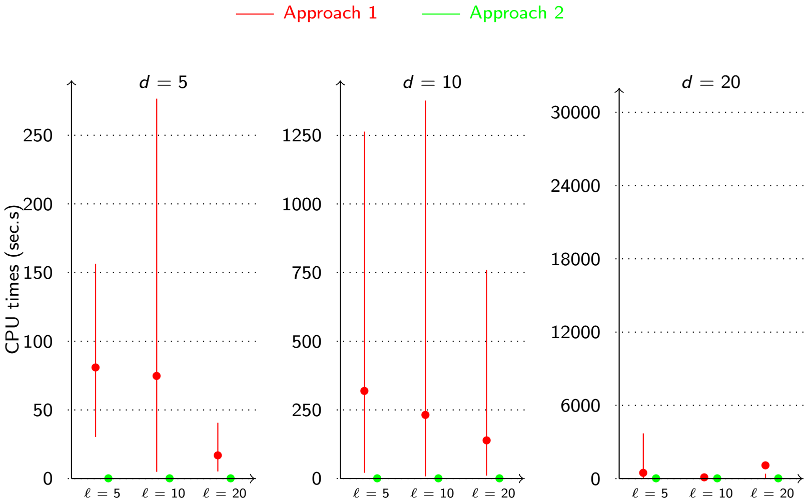
# CPU times: $n = 100$



# CPU times: $n = 200$



# CPU times: $n = 500$



## Number of non-dominated solutions

| | | | # n.-dom. sol.s | | | Succ. |
|---|---|---|---|---|---|---|
| $n$ | $d$ | $\ell$ | Mean | min | Max | (%) |
| 100 | 5 | 5 | 4.545 | 2 | 9 | 20 |
| 100 | 5 | 10 | 4.067 | 2 | 10 | 55 |
| 100 | 5 | 20 | 7.071 | 3 | 16 | 60 |
| 100 | 10 | 5 | 8.000 | 3 | 19 | 65 |
| 100 | 10 | 10 | 10.294 | 5 | 20 | 90 |
| 100 | 10 | 20 | 11.611 | 5 | 17 | 80 |
| 100 | 20 | 5 | 13.500 | 6 | 23 | 55 |
| 100 | 20 | 10 | 14.450 | 8 | 23 | 70 |
| 100 | 20 | 20 | 18.000 | 10 | 27 | 80 |
| 200 | 5 | 5 | 3.500 | 2 | 7 | 50 |
| 200 | 5 | 10 | 6.857 | 2 | 11 | 60 |
| 200 | 5 | 20 | 5.917 | 2 | 14 | 65 |
| 200 | 10 | 5 | 10.500 | 3 | 22 | 90 |
| 200 | 10 | 10 | 11.556 | 4 | 29 | 95 |
| 200 | 10 | 20 | 12.250 | 3 | 24 | 100 |

| | | | # n.-dom. sol.s | | | Succ. |
|---|---|---|---|---|---|---|
| $n$ | $d$ | $\ell$ | Mean | min | Max | (%) |
| 200 | 20 | 5 | 14.550 | 8 | 25 | 100 |
| 200 | 20 | 10 | 15.650 | 5 | 24 | 100 |
| 200 | 20 | 20 | 17.450 | 7 | 32 | 100 |
| 500 | 5 | 5 | 6.545 | 2 | 10 | 35 |
| 500 | 5 | 10 | 6.071 | 2 | 12 | 65 |
| 500 | 5 | 20 | 9.667 | 2 | 16 | 35 |
| 500 | 10 | 5 | 10.400 | 4 | 21 | 40 |
| 500 | 10 | 10 | 12.000 | 4 | 22 | 80 |
| 500 | 10 | 20 | 13.700 | 3 | 25 | 60 |
| 500 | 20 | 5 | 13.200 | 6 | 26 | 35 |
| 500 | 20 | 10 | 18.600 | 10 | 26 | 30 |
| 500 | 20 | 20 | 22.750 | 5 | 35 | 60 |

Succ.: Approach 2 success rate (problems solved till the end.)

# The end

Thank you for listening!

# References

Clímaco & Martins.
A bicriterion shortest path algorithm. *EJOR*, 11:399-404, 1982.

Clímaco & Pascoal.
Finding non-dominated shortest pairs of disjoint simple paths. *C & OR*, 36:2892-2898, 2009.

Cohon.
*Multiobjective programming and planning*. Academic Press, New York, 1978.

Current, ReVelle & Cohon.
An interactive approach to identify the best compromise solution for two objective shortest path problems. *C & OR*, 17:187-198, 1990.

Martins, Pascoal & Santos.
Deviation algorithms for ranking shortest paths. *The International Journal of Foundations of Computer Science*, 10:247-263, 1999.

Ralphs, Saltzman & Wiecek.
An improved algorithm for solving biobjective integer programs. *AofOR*, 147:43-70, 2006.

Suurballe.
Disjoint paths in a network. *Networks*, 4:125–145, 1974.