AoS4 - Session #2

MultiObjective Optimization with LP

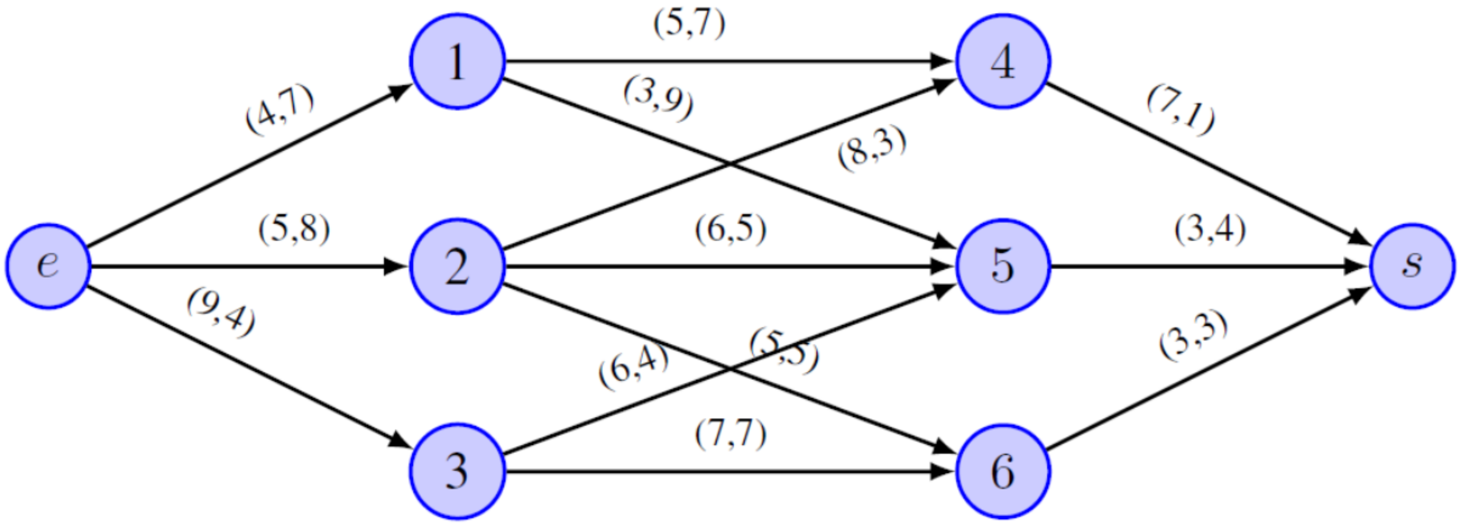
Problem : Solve a MOO problem over a combinatorial domain

1. to get a clear view, we solve it **brutally** and **manually** on a small instance
2. we devise approaches to **automatize** the resolution

The problem : multiobjective shortest path

Here : #objectives = 2 and #nodes = 8

Instance

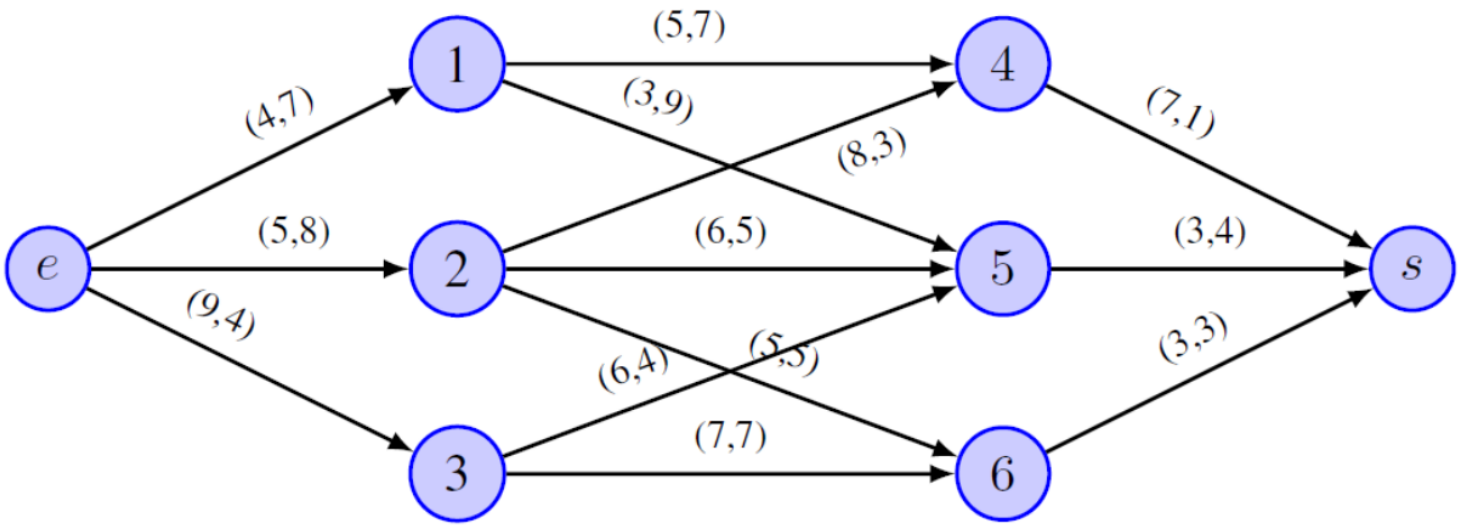


> 1. *On this graph, compute the shortest path according to criterion 1. Same question for criterion 2. For these two optimal paths, what is the value of the criterion which is not optimized?*

The problem : multiobjective shortest path

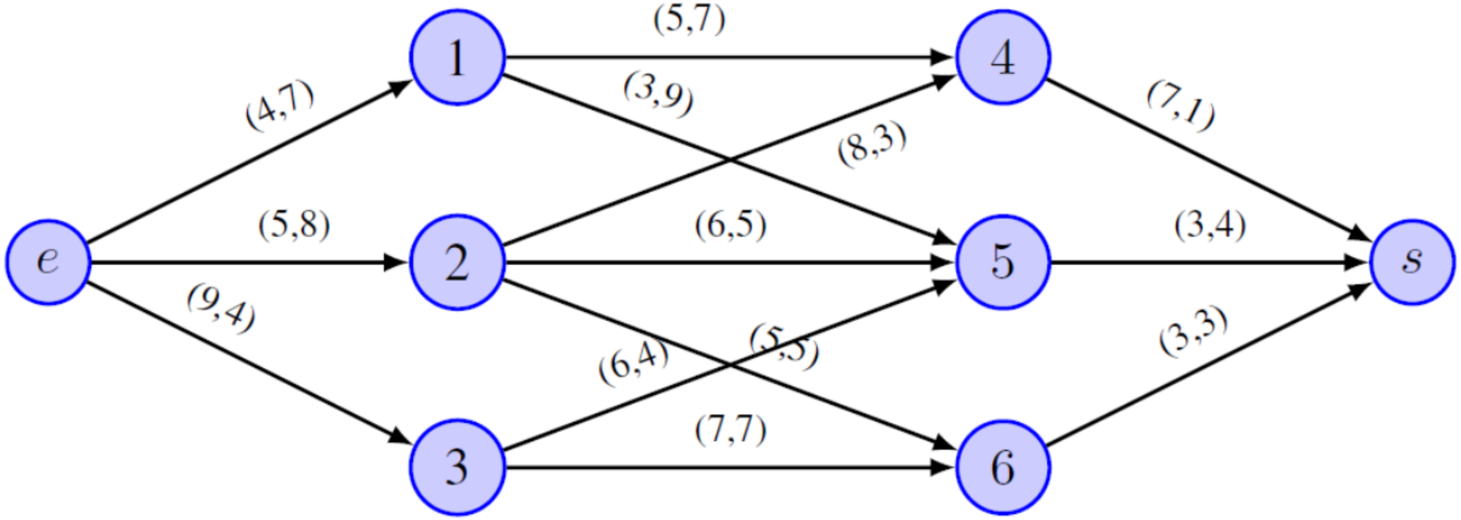
Here : #objectives = 2 and #nodes = 8

Instance



> 1. *On this graph, compute the shortest path according to criterion 1. Same question for criterion 2. For these two optimal paths, what is the value of the criterion which is not optimized?*
> 2. *How many path are there from e to s ? Enumerate them, represent them in the criterion space and identify the set of efficient paths.*

The problem : multiobjective shortest path



name	path	crit. 1	crit. 2
A	$e \rightarrow 1 \rightarrow 4 \rightarrow s$	16	15

The problem : multiobjective shortest path

name	**path**	**crit. 1**	**crit. 2**
A	$e \to 1 \to 4 \to s$	16	15
B	$e \to 1 \to 5 \to s$	10	20
C	$e \to 2 \to 4 \to s$	20	12
D	$e \to 2 \to 5 \to s$	14	17
E	$e \to 2 \to 6 \to s$	13	16
F	$e \to 3 \to 5 \to s$	18	12
G	$e \to 3 \to 6 \to s$	19	14

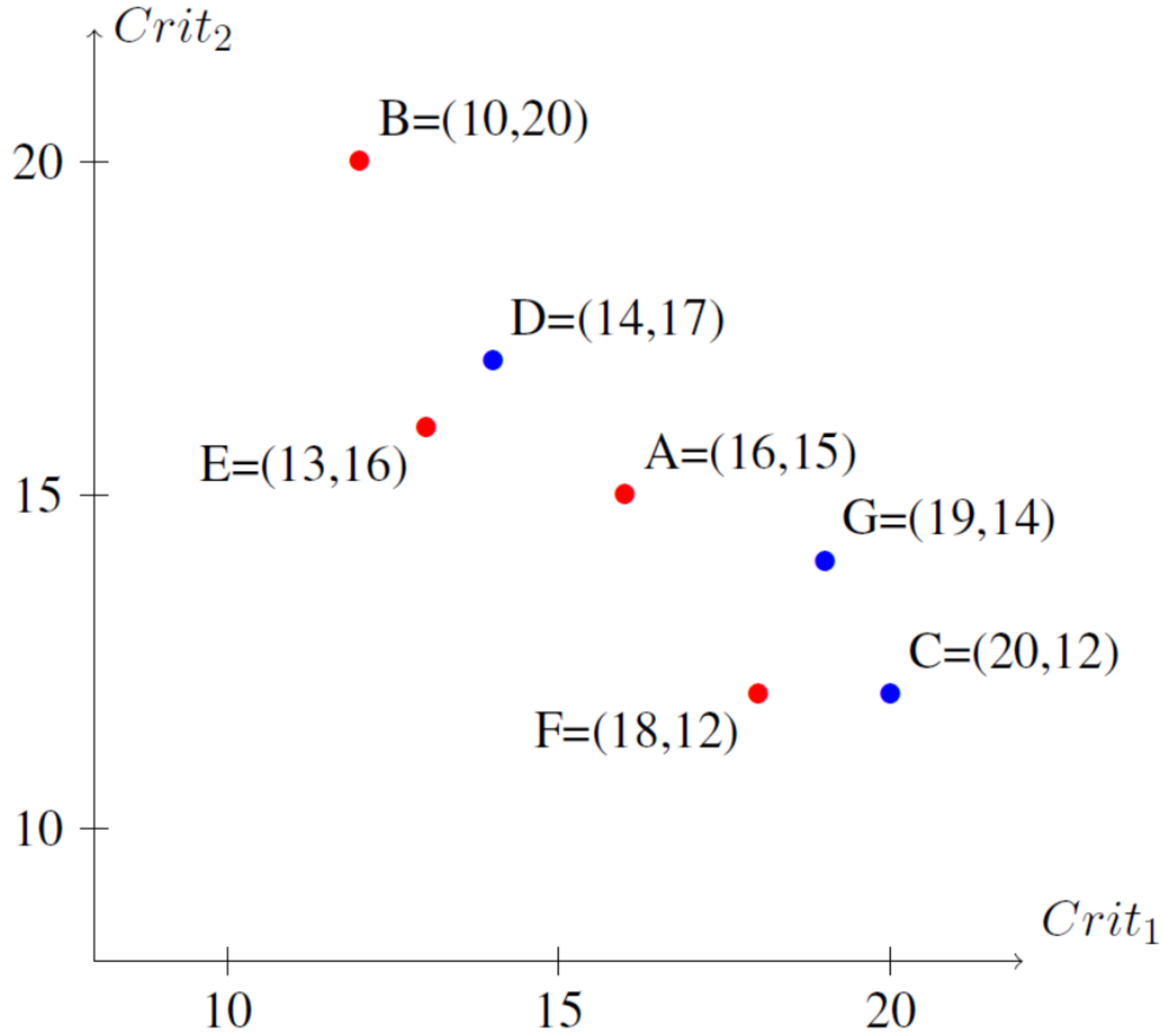
> *Efficient paths? (a.k.a. Pareto front)*

The problem : multiobjective shortest path

name	path	crit. 1	crit. 2
A	$e \to 1 \to 4 \to s$	16	15
B	$e \to 1 \to 5 \to s$	10	20
C	$e \to 2 \to 4 \to s$	20	12
D	$e \to 2 \to 5 \to s$	14	17
E	$e \to 2 \to 6 \to s$	13	16
F	$e \to 3 \to 5 \to s$	18	12
G	$e \to 3 \to 6 \to s$	19	14

> *Efficient paths? (a.k.a. Pareto front)*

Draw alternatives in the (c^1, c^2) plane



$Crit_2$
B=(10,20)
20
D=(14,17)
E=(13,16)
A=(16,15)
15
G=(19,14)
C=(20,12)
F=(18,12)
10
$Crit_1$
10
15
20

Solving single-objective Shortest Path with LP

> *Formulate a linear program that determines the shortest path from e to s according to the first criterion.*

Solving single-objective Shortest Path with LP

> *Formulate a linear program that determines the shortest path from e to s according to the first criterion.*

- *Decision variables*: 1/edge (i, j), x_{ij}
- Linear *objective function*: minimize $\sum x_{ij} c_{ij}^1$
- Linear *constraints*: flow conservation at each node

Solving single-objective Shortest Path with LP

> *Formulate a linear program that determines the shortest path from e to s according to the first criterion.*

- *Decision variables*: 1/edge (i, j), x_{ij}
- Linear *objective function*: minimize $\sum x_{ij} c^1_{ij}$
- Linear *constraints*: flow conservation at each node

> *Which difficulty appears when formulating the same program to optimize the second criterion? How can we tweak the objective function so that the optimal solution of the linear program corresponds to an efficient solution of the bi-criteria problem?*

Solving single-objective Shortest Path with LP

> *Formulate a linear program that determines the shortest path from e to s according to the first criterion.*

- *Decision variables*: 1/edge (i, j), x_{ij}
- Linear *objective function*: minimize $\sum x_{ij}c^1_{ij}$
- Linear *constraints*: flow conservation at each node

> *Which difficulty appears when formulating the same program to optimize the second criterion? How can we tweak the objective function so that the optimal solution of the linear program corresponds to an efficient solution of the bi-criteria problem?*

- C and F are equivalent w.r.t. criterion 2
- We need to devise an objective function that *break ties* in favor of F

Solving single-objective Shortest Path with LP

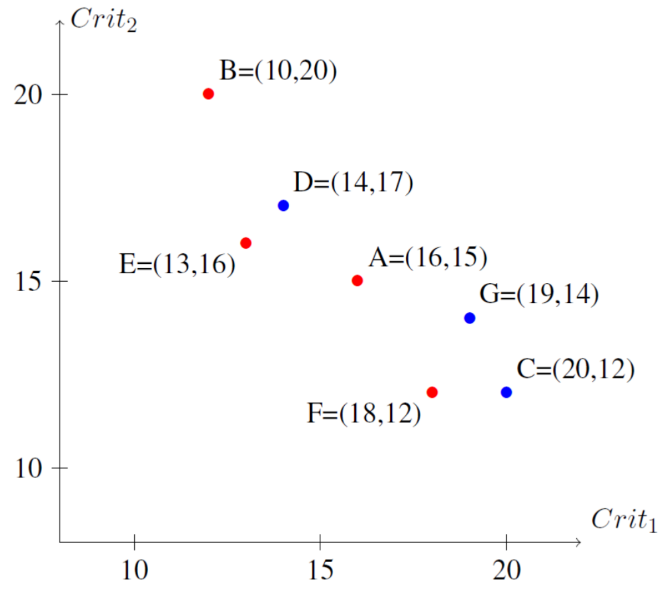
> *Formulate a linear program that determines the shortest path from e to s according to the first criterion.*

- *Decision variables*: 1/edge (i, j), x_{ij}
- Linear *objective function*: minimize $\sum x_{ij} c^1_{ij}$
- Linear *constraints*: flow conservation at each node

> *Which difficulty appears when formulating the same program to optimize the second criterion? How can we tweak the objective function so that the optimal solution of the linear program corresponds to an efficient solution of the bi-criteria problem?*

- C and F are equivalent w.r.t. criterion 2
- We need to devise an objective function that *break ties* in favor of F
- *lexicographically* minimize crit. 2, then crit. 1: use $c^2 + \alpha c^1$ with α "small enough"

Using LP to solve the bi-objective problem



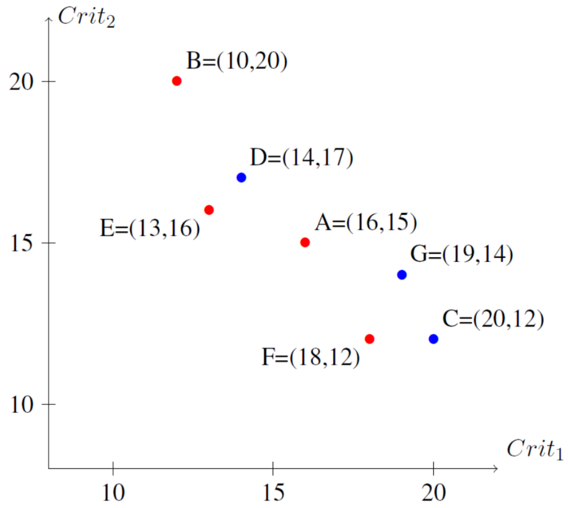
First approach: using a convex combination of the objectives

Suppose we want to solve the bi-objective problem by forming a single objective, and that we use a WS of the two objectives

$$C_{\lambda}^{MO} := \lambda c_1 + (1 - \lambda) c_2$$

> 1. *Given a value of λ, can we use LP ?*
> 2. *Find values of the parameter λ governing the WS aggregation, such that the optimal solution is the path B (resp. E, A, F).*

Using LP to solve the bi-objective problem



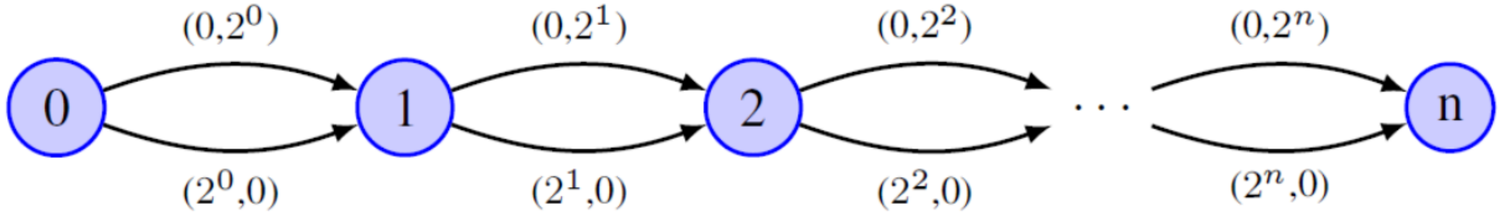
Second approach: ϵ-constraint

(Enumerating the Pareto front via iteratively solving LP problems)

> *Propose an algorithm that identifies all efficient solutions of the bi-criteria problem. This algorithm will proceed by the resolution of a sequence of linear programs that you should specify.*

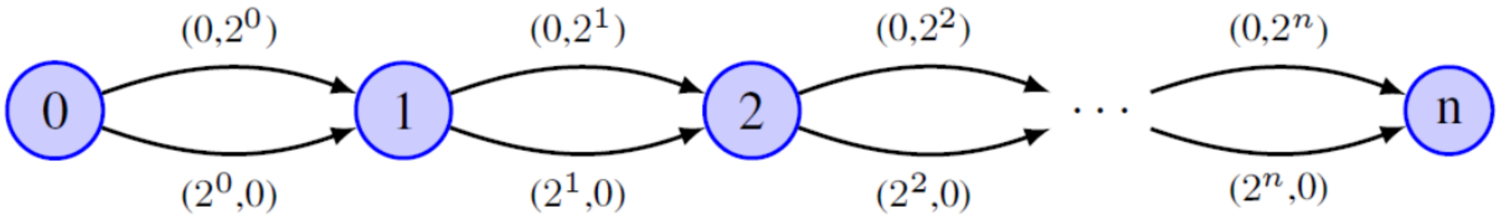
Combinatorics explosion!

We now consider the bi-criteria shortest path problem on the following graph.



Combinatorics explosion!

We now consider the bi-criteria shortest path problem on the following graph.



> *Which new difficulty occurs? What conclusion can be drawn concerning the complexity of the ϵ-constraint algorithm?*

Implement!

- Using e.g. *Python 3.9* and a LP module (scipy.optimize.linprog or *preferably PuLP* or …)
- Decide on a *representation for multi-valuated graphs*
- write a function mapping an input graph and valuation to a *LP formulation of Shortest Path*
- write a function allowing to solve *Shortest Path for a convex combination of the objectives*
- write a function performing ϵ*-constraint on a bi-objective LP formulation*
- extend to a *multi-objective* formulation!