The multi-modal traveling salesman problem

Nicolas Jozefowiez[1], Gilbert Laporte[2], Frédéric Semet[3]

1. LAAS-CNRS, INSA, Université de Toulouse, Toulouse, France, nicolas.jozefowiez@laas.fr

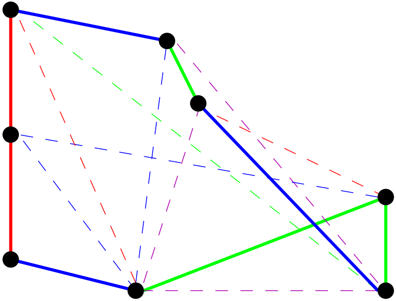
2. CIRRELT, HEC, Montréal, Canada, gilbert@crt.umontreal.ca

3. LAGIS, Ecole Centrale de Lille, Villeneuve d'Ascq, France, frederic.semet@ec-lille.fr

Outlines

- The multi-modal traveling salesman problem
- Multi-objective optimization
- A branch-and-cut algorithm
- Computational results
- Conclusions and perspectives

The multi-modal traveling salesman problem



Data:

$G = (V, E)$: an undirected valuated graph
C is a set of colors
Each $e \in E$ has a color $k \in C$

Goal:

Find a Hamiltonian cycle
Two objectives:

1. Minimize the total length of the cycle
2. Minimize the number of colors appearing on the cycle

Integer program

Variables

$$x_e = \begin{cases} 1 & \text{if } e \in E \text{ is used,} \\ 0 & \text{otherwise.} \end{cases}$$

$$u_k = \begin{cases} 1 & \text{if } k \in C \text{ is used,} \\ 0 & \text{otherwise.} \end{cases}$$

Constants and notations

$$\forall e \in E,\ \delta(e) = k \in C \text{ the color of } e$$

$$\forall k \in C,\ \zeta(k) = \{e \in E | \delta(e) = k\}$$

$$\forall S \subset V,\ \omega(S) = \{e = (i,j) \in E | i \in S \text{ and } j \in V \setminus S\}$$

Integer program

Objective functions

$$\min \sum_{e \in E} c_e x_e$$

$$\min \sum_{k \in C} u_k$$

Constraints

$$
\begin{aligned}
\sum_{e \in \omega(\{i\})} x_e &= 2 && \forall i \in V \\
\sum_{e \in \omega(S)} x_e &\geq 2 && \forall S \subset V, 3 \leq |S| \leq |V| - 3 \\
x_e &\leq u_{\delta(e)} && \forall e \in E \\
x_e &\in \{0,1\} && \forall e \in E \\
u_k &\in \{0,1\} && \forall k \in C
\end{aligned}
$$

Valid constraints

$$u_k \leq \sum_{e \in \zeta(k)} x_e \quad \forall k \in C$$

State-of-the-art

Minimum labelling hamiltonian cycle problem (Tabu search) [Cerulli, Dell'Olmo, Gentili, Raiconi, 2006]

Colorful traveling salesman problem (heuristic, GA) [Xiong, Golden, Wasil, 2007]

Traveling salesman problem with labels (approximation algorithm) [Gourvès, Monnot, Telelis, 2008]

Minimum labelling spanning tree problem

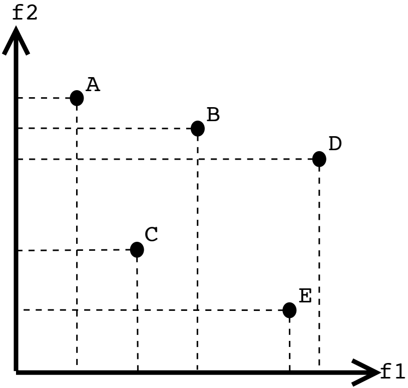
Multi-objective optimization problem

$$(PMO) = \begin{cases} min\ F(x) = (f_1(x), f_2(x), \ldots, f_n(x)) \\ s.t.\ x \in \Omega \end{cases}$$

with:

- $n \geq 2$: number of objectives
- $F = (f_1, f_2, \ldots, f_n)$: vector of functions to optimize
- $\Omega \subseteq \mathbb{R}^m$: set of feasible solutions
- $x = (x_1, x_2, \ldots, x_m) \in \Omega$: a feasible solution
- $\mathcal{Y} = F(\Omega)$: objective space
- $y = (y_1, y_2, \ldots, y_n) \in \mathcal{Y}$ avec $y_i = f_i(x)$: a point in the objective space

A solution x dominates ($\preceq$) a solution y if and only if
$\forall i \in \{1, \ldots, n\}, f_i(x) \leq f_i(y)$ and $\exists i \in \{1, \ldots, n\}$ such that $f_i(x) < f_i(y)$.



A multi-objective branch-and-cut algorithm

STEP 1 (Root of the tree)
Generate an initial upper bound ub
Define a first sub-problem
Insert the sub-problem in a list L

STEP 2 (Stopping criterion)
If $L = \emptyset$ then STOP, else choose a sub-problem from L and remove it from L

STEP 3 (Sub-problem solution)
Solve the sub-problem to obtain the lower bound lb

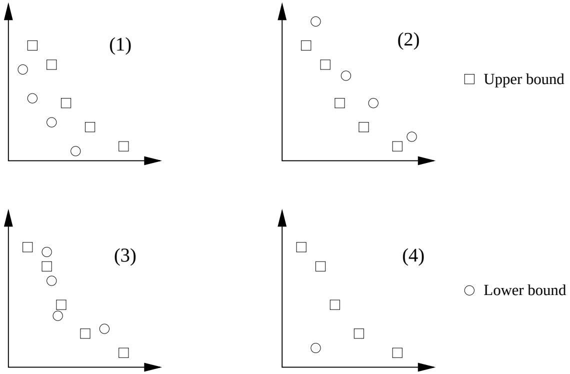
STEP 4 (Constraint generation)
If some integer solutions have been found, try to insert them in ub
if $ub \preceq lb$ **then**
  Go to STEP 2.
else
  if violated constraints are identified **then**
    Add them to the model and go to STEP 3.
  else
    Go to STEP 5.
  end if
end if

STEP 5 (Branching)
Branch on variable and introduce 2 new sub-problems in L. Go to STEP 2.

Adaptations to a multi-objective problem

Upper bound = set of non-dominated solutions found during the search

Lower bound = set of non-dominated points in the objective space such that all feasible solutions are dominated by these points



[Sourd et Spanjaard, 2008]

Computation of the lower bound

Initial sub-problem :

$$\min \quad \sum_{e \in E} c_e x_e$$

$$\min \quad \sum_{k \in C} u_k$$

$$\sum_{e \in \omega(\{i\})} x_e = 2 \qquad \forall i \in V$$

$$x_e \leq u_{\delta(e)} \qquad \forall e \in E$$

$$u_k \leq \sum_{e \in \zeta(k)} x_e \qquad \forall k \in C$$

$$0 \leq x_e \leq 1 \qquad \forall e \in E$$

$$0 \leq u_k \leq 1 \qquad \forall k \in C$$

Computation of the lower bound

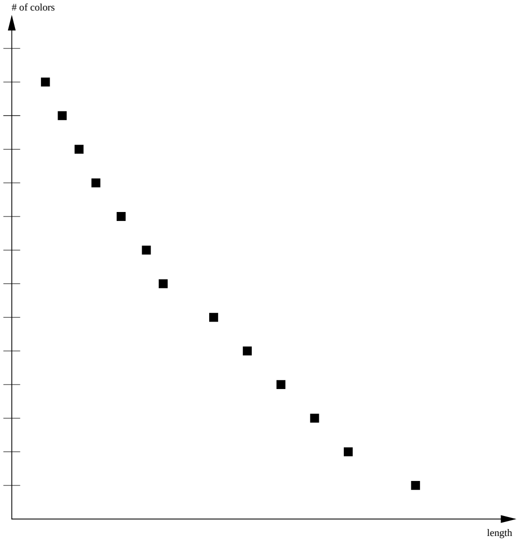
Solve the following problem for different values of ϵ

$$
\begin{aligned}
\min \sum_{e \in E} c_e x_e &+ m \sum_{k \in C} u_k & \\
\sum_{e \in \omega(\{i\})} x_e &= 2 & \forall i \in V \\
x_e &\leq u_{\delta(e)} & \forall e \in E \\
u_k &\leq \sum_{e \in \zeta(k)} x_e & \forall k \in C \\
\sum_{k \in C} u_k &\leq \epsilon & \\
0 \leq x_e &\leq 1 & \forall e \in E \\
0 \leq u_k &\leq 1 & \forall k \in C
\end{aligned}
$$

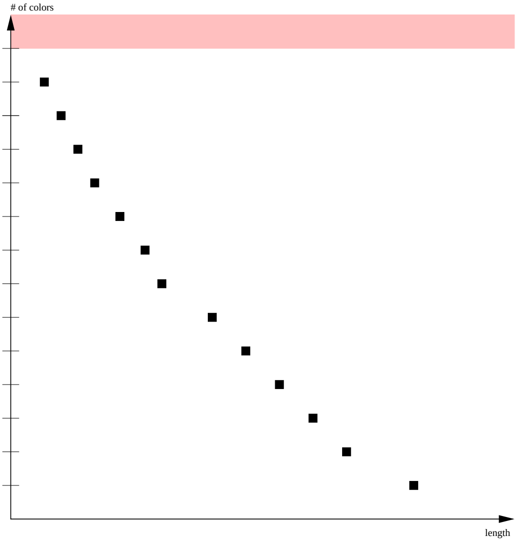
After founding non-dominated solution for a given ϵ, identify violated constraints and add them

Computation of the lower bound

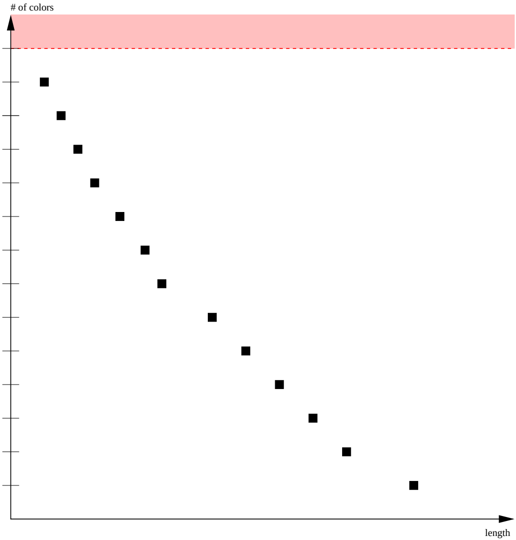
Computation of the lower bound



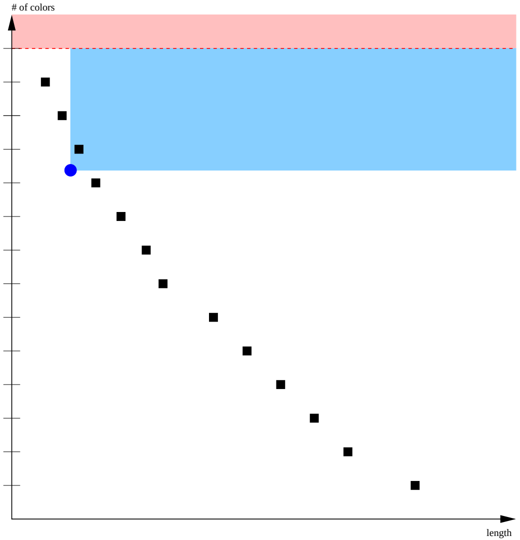
Computation of the lower bound



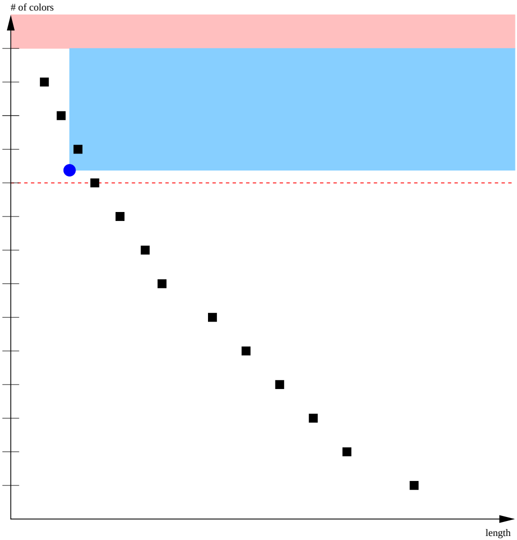
Computation of the lower bound



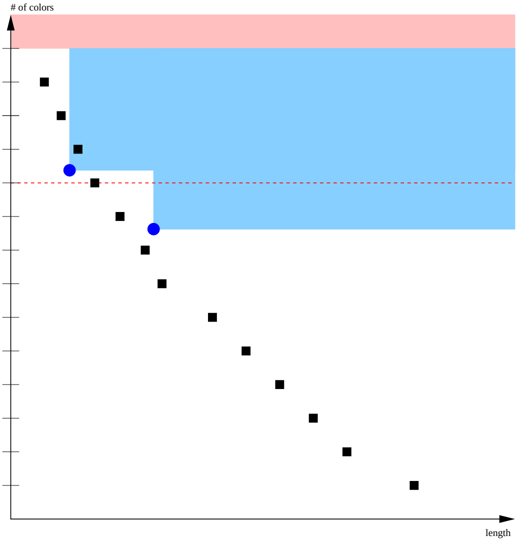
Computation of the lower bound



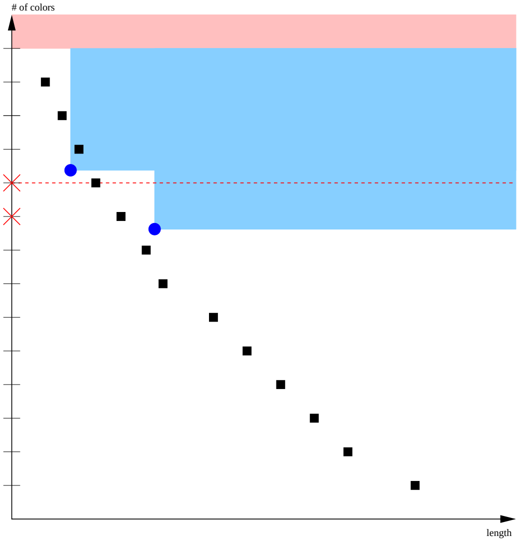
Computation of the lower bound



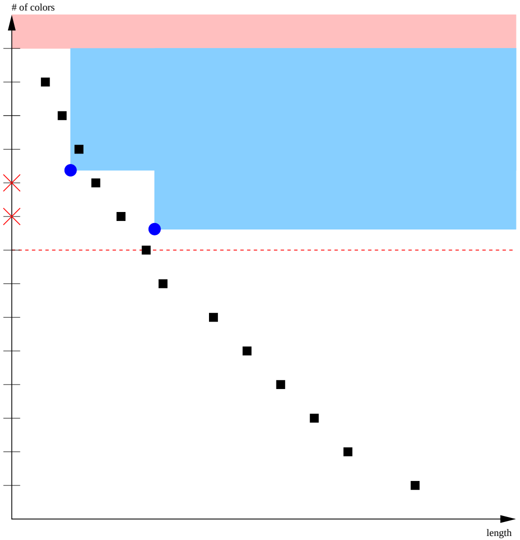
Computation of the lower bound



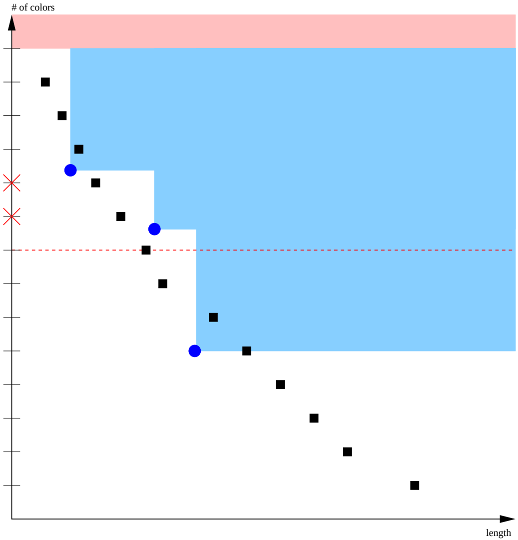
Computation of the lower bound



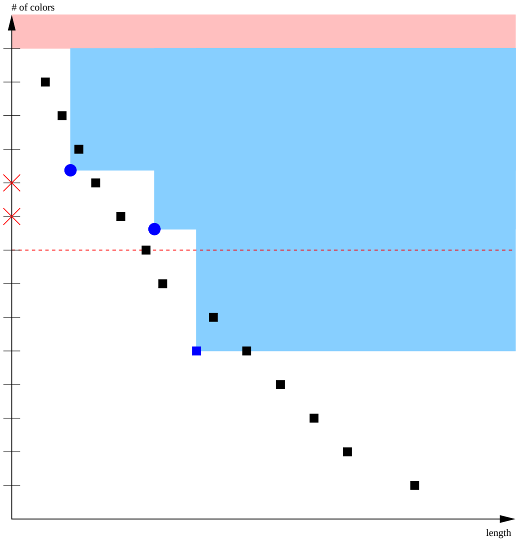
Computation of the lower bound



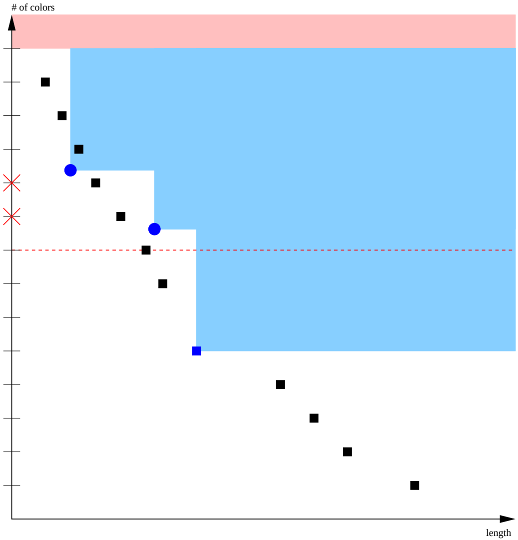
Computation of the lower bound



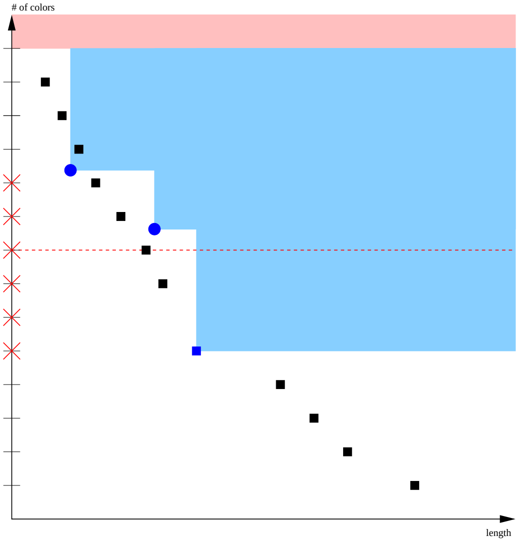
Computation of the lower bound



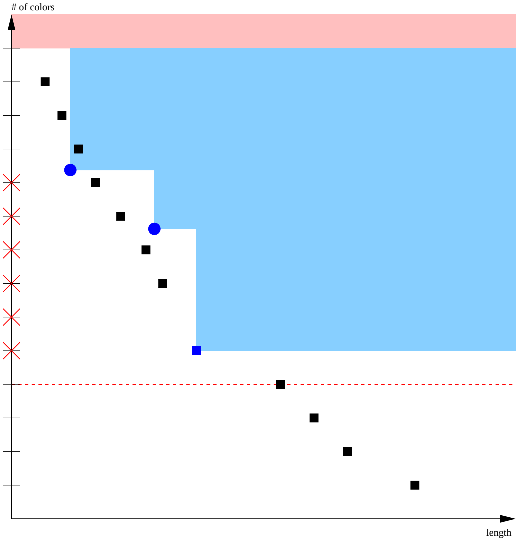
Computation of the lower bound



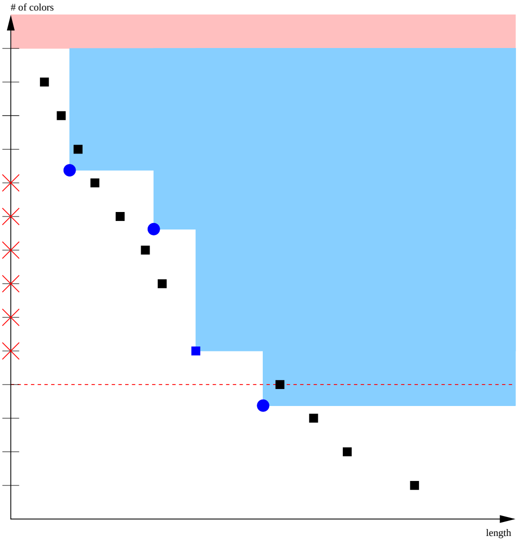
Computation of the lower bound



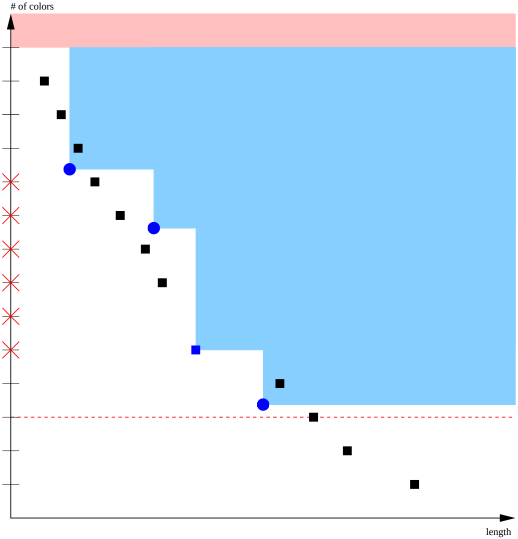
Computation of the lower bound



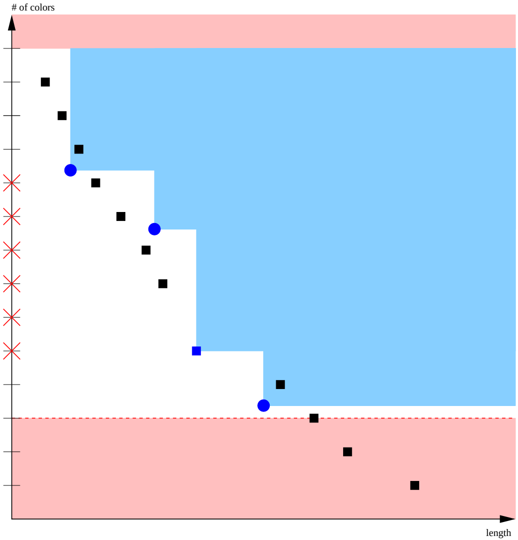
Computation of the lower bound



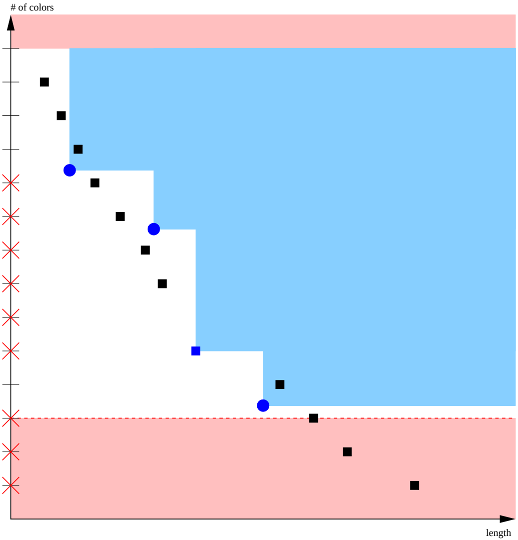
Computation of the lower bound



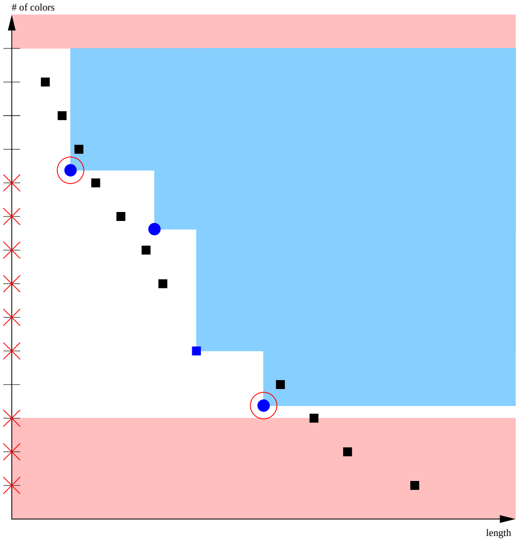
Computation of the lower bound



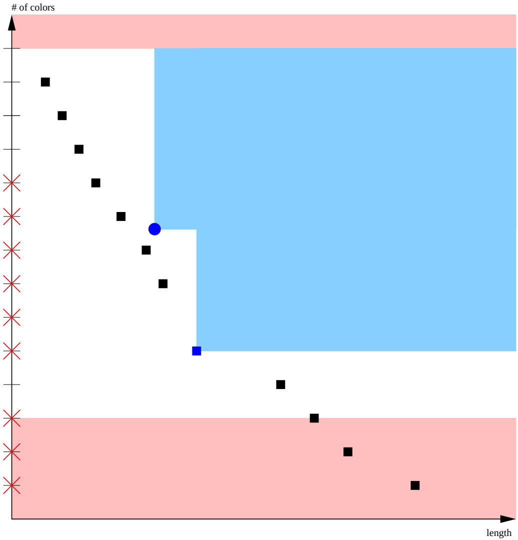
Computation of the lower bound



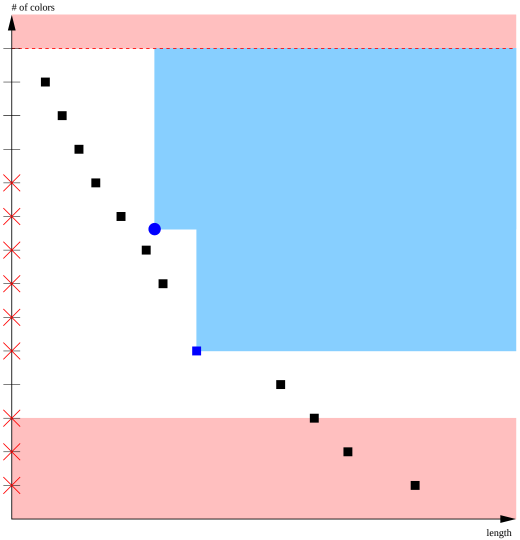
Computation of the lower bound



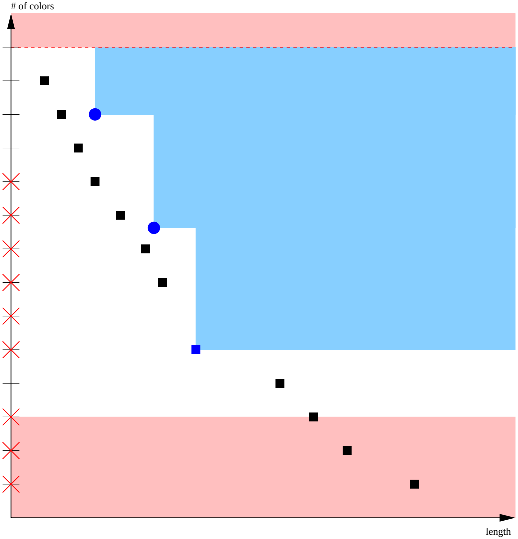
Computation of the lower bound



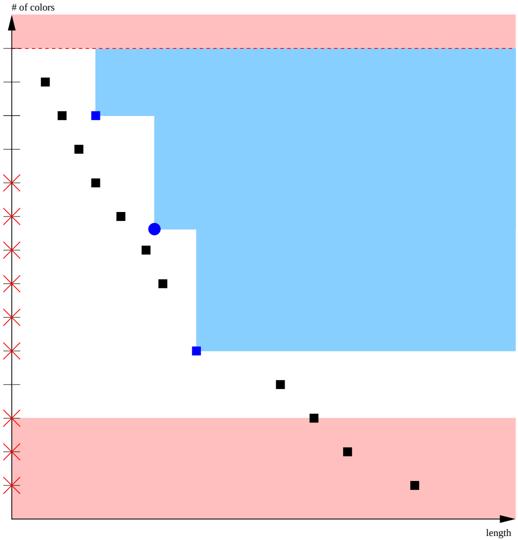
Computation of the lower bound



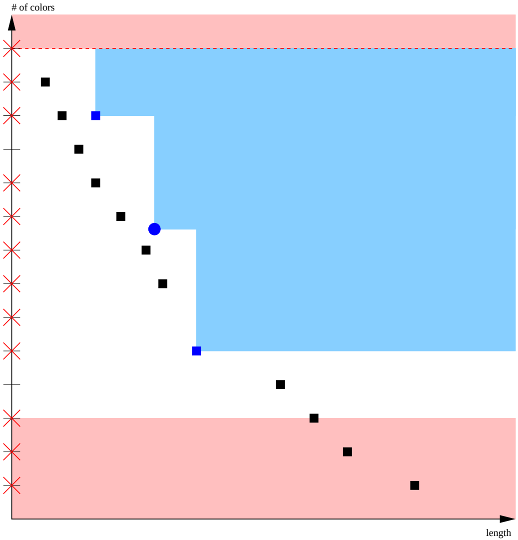
Computation of the lower bound



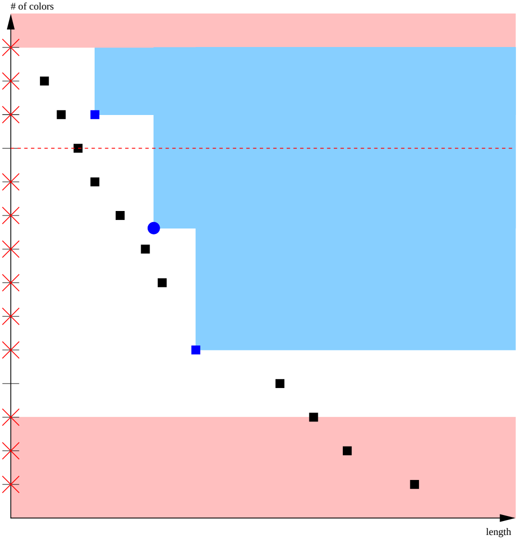
Computation of the lower bound



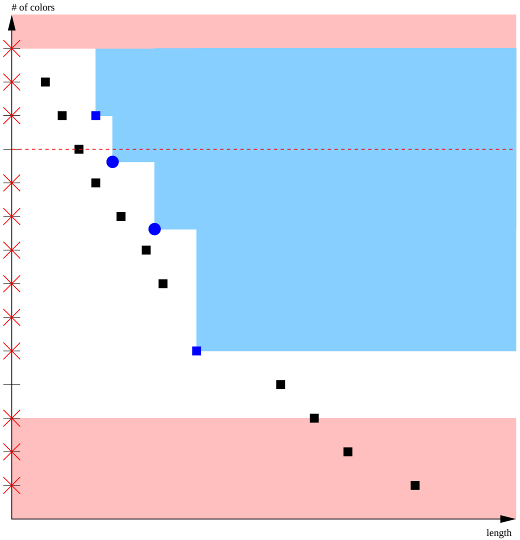
Computation of the lower bound



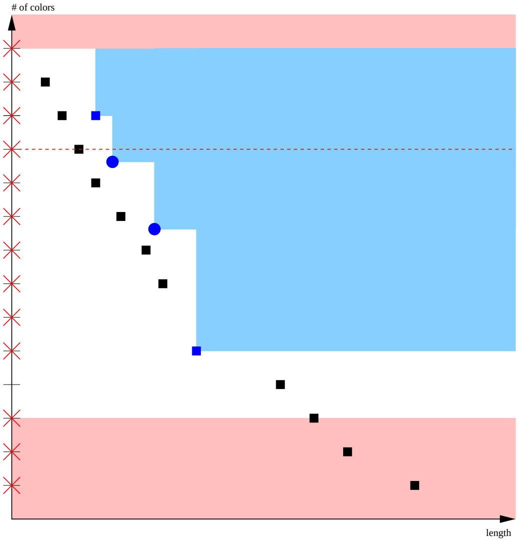
Computation of the lower bound



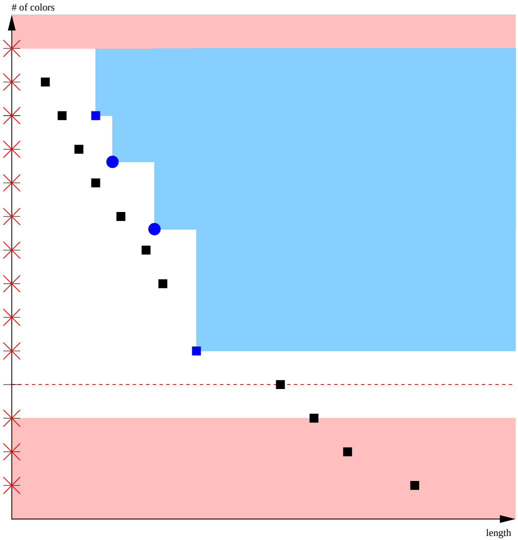
Computation of the lower bound



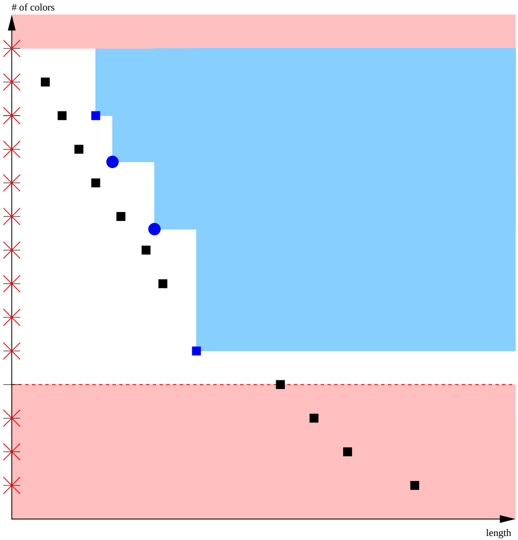
Computation of the lower bound



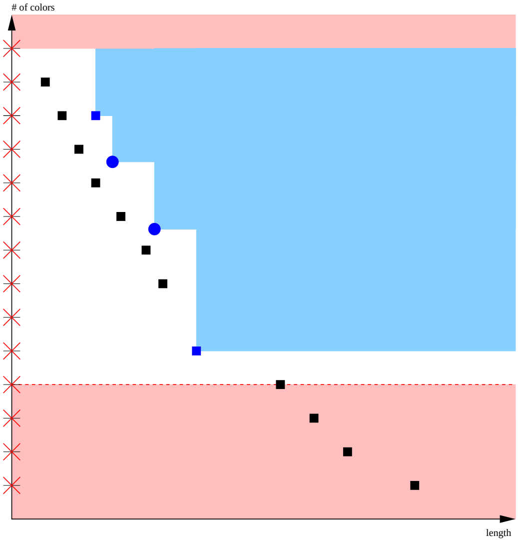
Computation of the lower bound



Computation of the lower bound



Computation of the lower bound



Constraint generation, cutting, and branching

Constraint generation

Connectivity constraints → min-cut problem

Call to a CONCORDE function [Padberg & Rinaldi, 1990]

Cutting

$\forall \epsilon$, the sub-problem is unfeasible

$\forall \epsilon$, the solution is either feasible or dominated by ub

Branching

First on the u_k variables then on the x_e

Priority on the variable that is non integral for the most values of ϵ

Computation of the initial upper bound

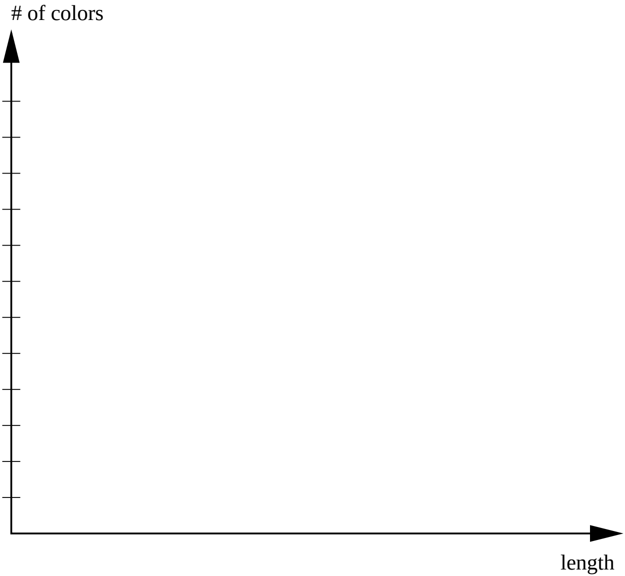
Repeat the following sequence for different values of ϵ

STEP 1: Solve the following mixed integer program :

$$
\begin{array}{rcll}
\min \displaystyle\sum_{e \in E} c_e x_e & + & m \displaystyle\sum_{k \in C} u_k & \\
\displaystyle\sum_{e \in \omega(\{i\})} x_e & = & 2 & \forall i \in V \\
x_e & \leq & u_{\delta(e)} & \forall e \in E \\
u_k & \leq & \displaystyle\sum_{e \in \zeta(k)} x_e & \forall k \in C \\
\displaystyle\sum_{k \in C} u_k & \leq & \epsilon & \\
0 \leq & x_e & \leq 1 & \forall e \in E \\
u_k & \in & \{0,1\} & \forall k \in C
\end{array}
$$

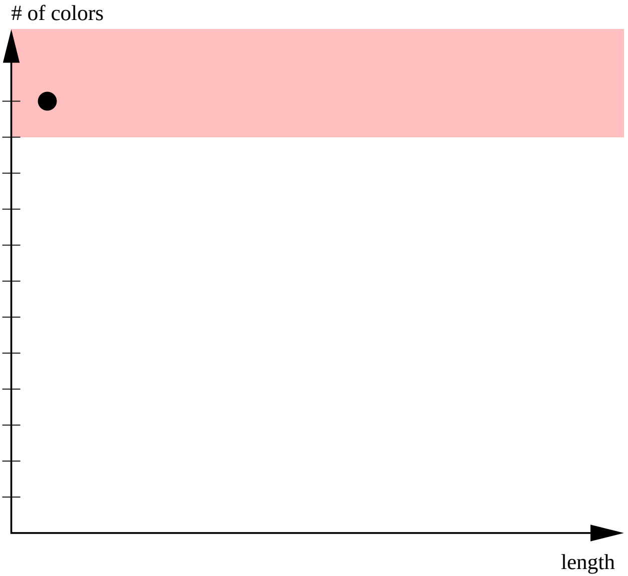
STEP 2: Solve a TSP on $G' = (V, E')$ with $E' = \{e \in E | u_{\delta(e)} = 1\}$

Computation of the initial upper bound

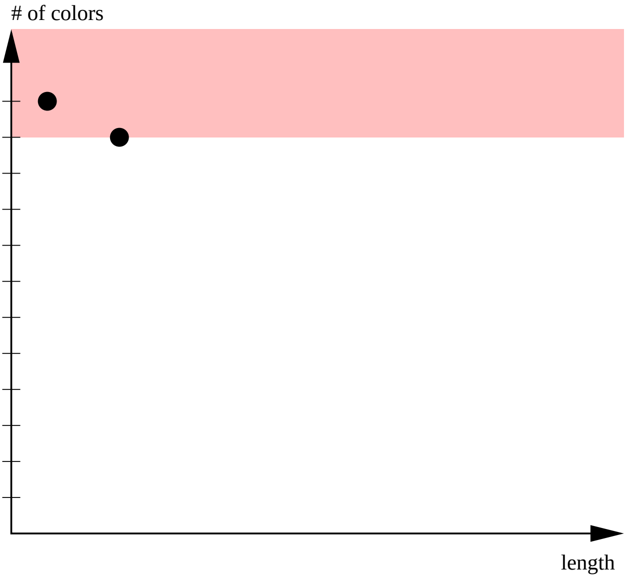


Computation of the initial upper bound

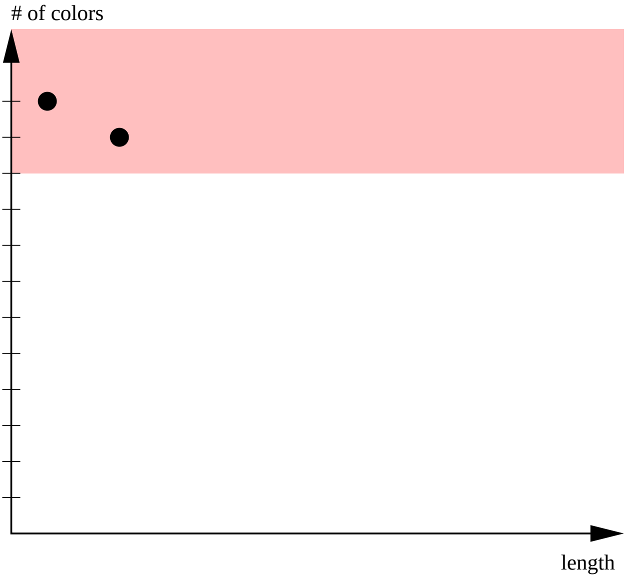
Computation of the initial upper bound



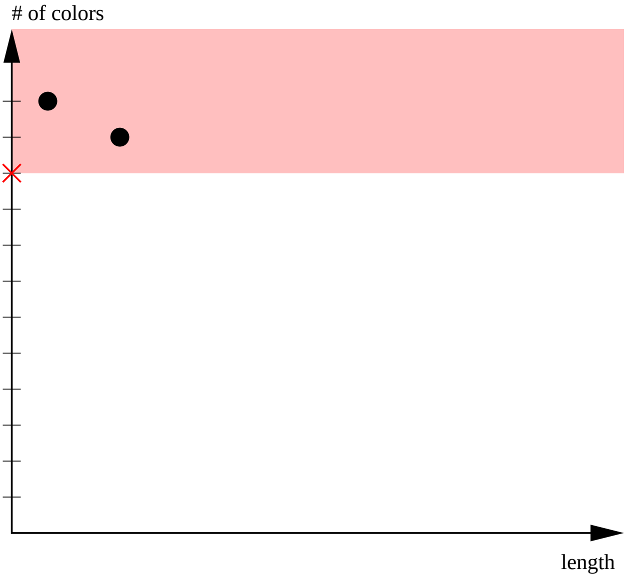
Computation of the initial upper bound



Computation of the initial upper bound

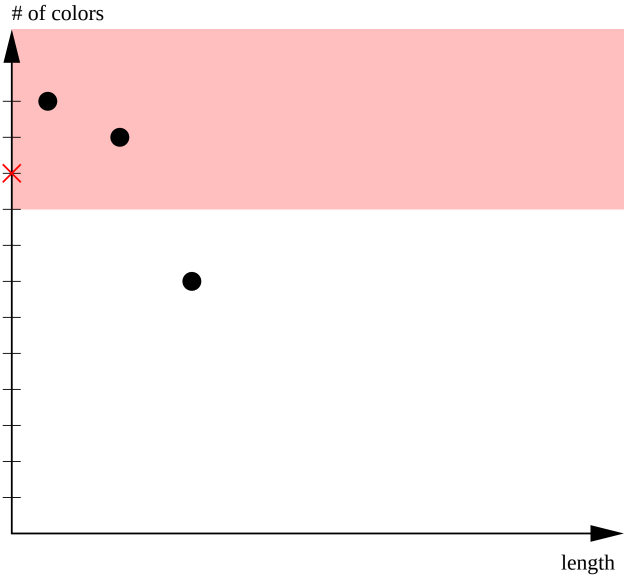


Computation of the initial upper bound

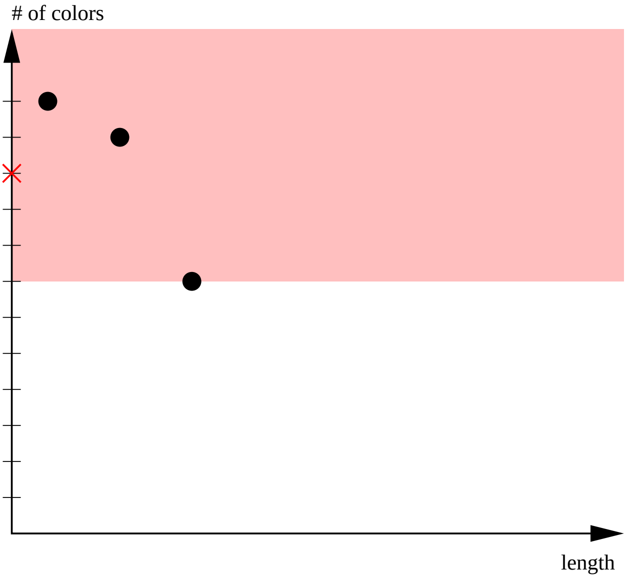


Computation of the initial upper bound

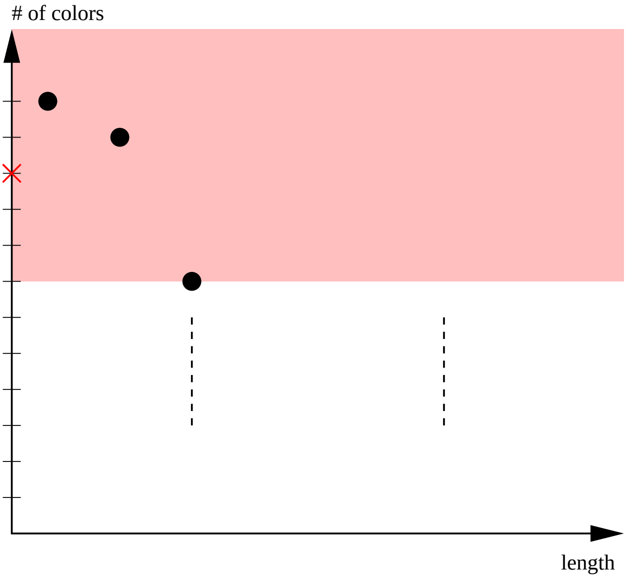
Computation of the initial upper bound



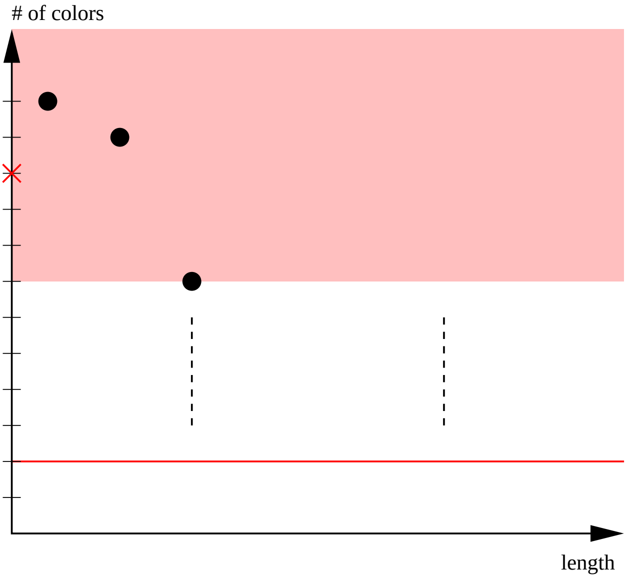
Computation of the initial upper bound



Computation of the initial upper bound



Computation of the initial upper bound



Computational results

$\lvert C \rvert$	$\lvert V \rvert$	$\times$	#nodes	#u	#x	#cut	#Pareto	#ub	#Parub
10	20	25	84	36	5	49	7	7	4
10	30	25	134	51	14	105	8	8	5
10	40	25	231	65	50	156	9	9	5
10	50	25	371	73	111	214	9	9	6
15	20	25	178	82	6	73	9	8	4
15	30	17	428	186	26	193	12	11	4
15	40	20	573	217	68	286	12	12	5
15	50	13	1102	298	252	432	13	13	5
20	20	25	372	180	5	123	10	9	4
20	30	25	815	376	31	274	13	12	4
20	40	25	1489	653	91	477	15	14	5
20	50	25	2468	898	335	759	17	16	5
25	20	25	466	226	5	130	11	9	4
25	30	25	1564	755	26	412	15	12	4
25	40	25	3244	1455	166	788	17	15	4
25	50	15	3756	1627	249	988	20	18	5
30	20	25	814	401	4	199	12	9	4
30	30	25	2274	1089	47	490	16	13	4
30	40	25	4875	2254	182	966	18	16	4
30	50	6	7752	3659	215	1563	20	17	4

Computational results

$\lvert C \rvert$	$\lvert V \rvert$	$\times$	time(bc)	time(ub)	time
10	20	25	1	0	1
10	30	25	11	0	12
10	40	25	39	2	41
10	50	25	112	5	117
15	20	25	4	0	5
15	30	17	50	1	52
15	40	20	165	5	171
15	50	13	527	12	538
20	20	25	12	0	13
20	30	25	142	3	146
20	40	25	731	8	739
20	50	25	3051	20	3071
25	20	25	17	0	17
25	30	25	392	4	397
25	40	25	3362	13	3376
25	50	15	8243	31	8268
30	20	25	36	0	37
30	30	25	749	5	754
30	40	25	5212	17	5229
30	50	6	35001	43	35045

Conclusions and perspectives

- Branch-and-cut algorithm able to solve a multi-objective problem in one run
- Identify new valid constraints $\rightarrow$ variables u_k
- Rules to choose on which variables to branch
- Progressive partition of the objective space