A multi-objective branch-and-cut algorithm for the multi-modal traveling salesman problem

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

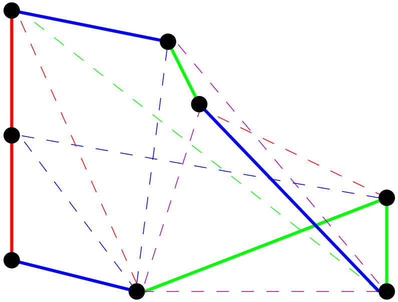
1. LAAS-CNRS, INSA, Université de Toulouse, Toulouse, France, nicolas.jozefowiez@laas.fr

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



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

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

$$\sum_{e \in \omega(S)} x_e \geq 2 \qquad \forall S \subset V, 3 \leq |S| \leq |V| - 3$$

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

$$x_e \in \{0, 1\} \qquad \forall e \in E$$

$$u_k \in \{0, 1\} \qquad \forall k \in C$$

Valid constraints

$$
\begin{aligned}
u_k &\leq \sum_{e \in \zeta(k)} x_e && \forall k \in C \\
\sum_{k \in C} \gamma_i^k u_k &\geq 2 && \forall i \in V \\
\sum_{k \in C} \lambda_k(S) u_k &\geq 2 && \forall S \in V, 3 \leq |S| \leq |V| - 3
\end{aligned}
$$

with

$$
\gamma_i^k = \begin{cases} 0 & \text{if } \nexists e \in \omega(\{i\}), e \in \zeta(k), \\ 1 & \text{if } \exists! e \in \omega(\{i\}), e \in \zeta(k), \\ 2 & \text{otherwise.} \end{cases} \quad \lambda_k(S) = \begin{cases} 0 & \text{if } \nexists e \in \omega(S), e \in \zeta(k), \\ 1 & \text{if } \exists! e \in \omega(S), e \in \zeta(k), \\ 2 & \text{otherwise.} \end{cases}
$$

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

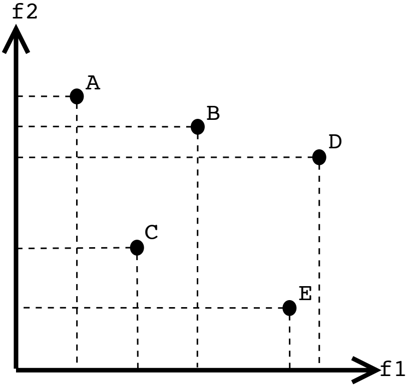
$$(PMO) = \begin{cases} min\ F(x) = (f_1(x), f_2(x), \ldots, f_n(x)) \\ s.t.\ x \in \Omega \end{cases}$$

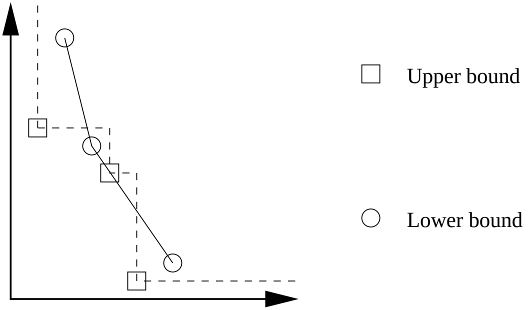
with:

- $n \geq 2$: 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

Pareto dominance relation

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 $f_i(x) < f_i(y)$.





Problem: it does not work if the agregated problem is NP-hard.

Solution: linear programming

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

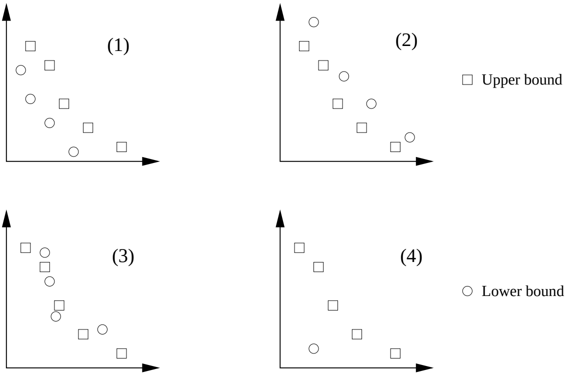
STEP 3 (Sub-problem solution)
Solve the sub-problem to obtain the lower bound lb

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.

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



Computation of the lower bound

Initial sub-problem :

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

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

$$
\begin{aligned}
\sum_{e \in \omega(\{i\})} x_e &= 2 && \forall i \in V \\
x_e &\leq u_{\delta(e)} && \forall e \in E \\
u_k &\leq \sum_{e \in \zeta(k)} x_e && \forall k \in C \\
\sum_{k \in C} \gamma_i^k u_k &\geq 2 && \forall i \in V
\end{aligned}
$$

$$
\begin{aligned}
0 \leq x_e &\leq 1 && \forall e \in E \\
0 \leq u_k &\leq 1 && \forall k \in C
\end{aligned}
$$

Computation of the lower bound

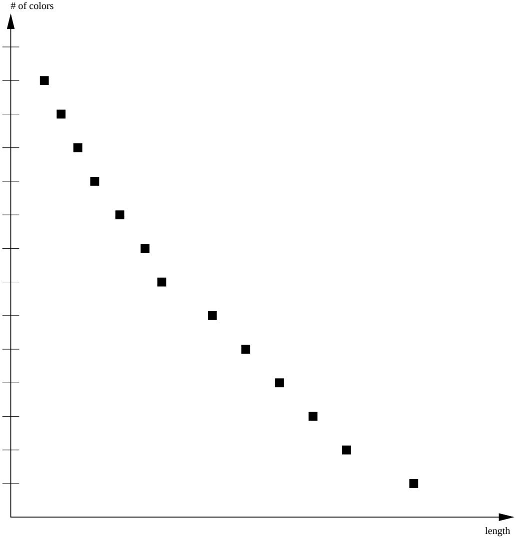
Solve the following problem for different values of ϵ

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

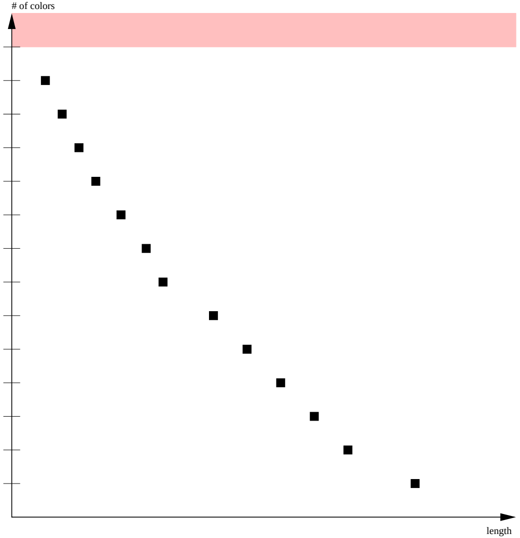
After founding non-dominated solution for a given ϵ, identify violated constraints and add them

Computation of the lower bound

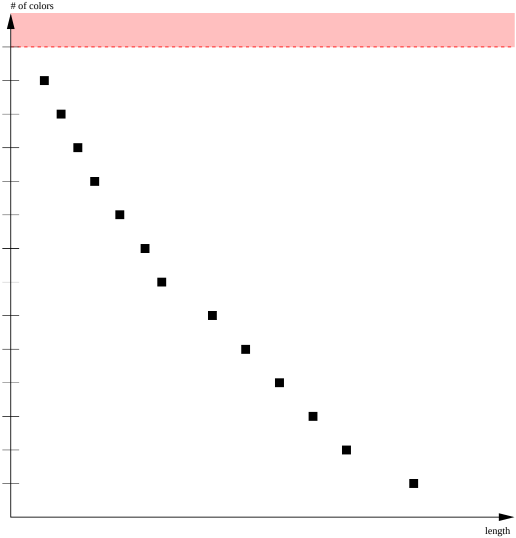
Computation of the lower bound



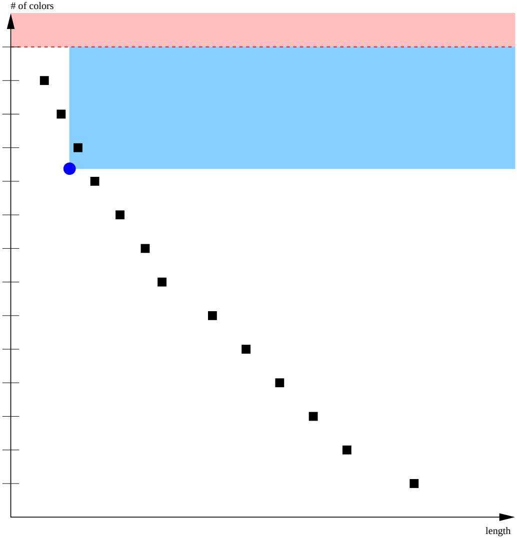
Computation of the lower bound



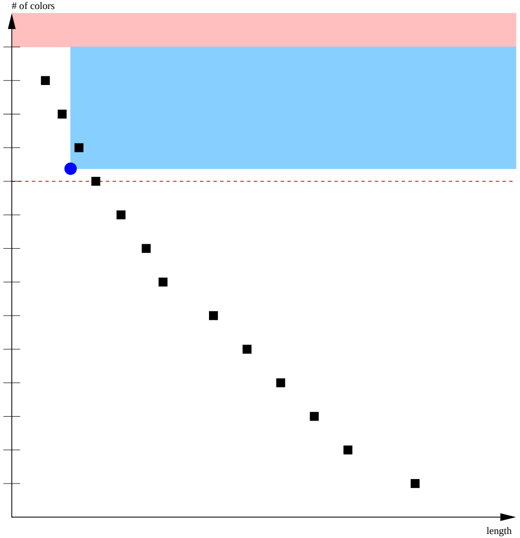
Computation of the lower bound



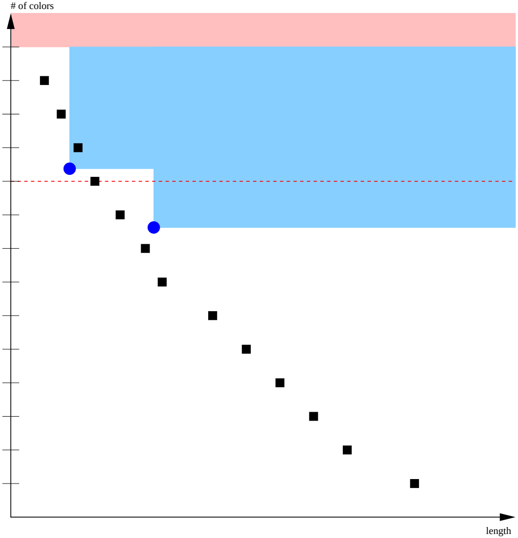
Computation of the lower bound



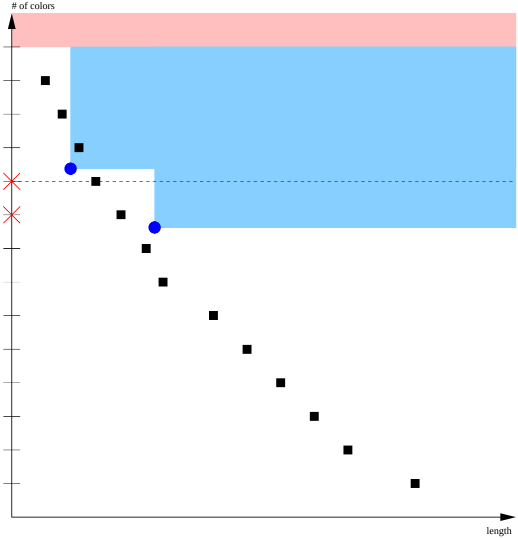
Computation of the lower bound



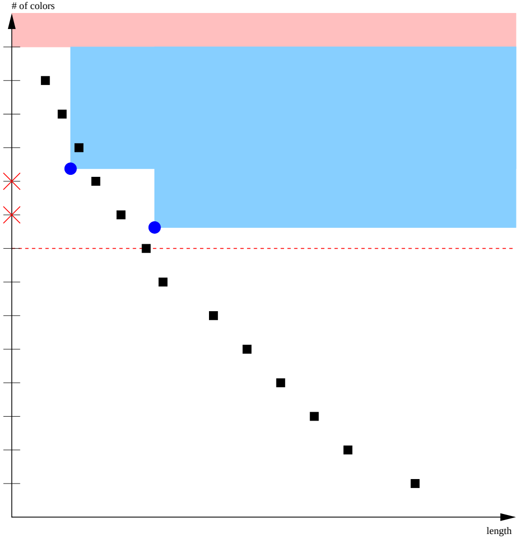
Computation of the lower bound



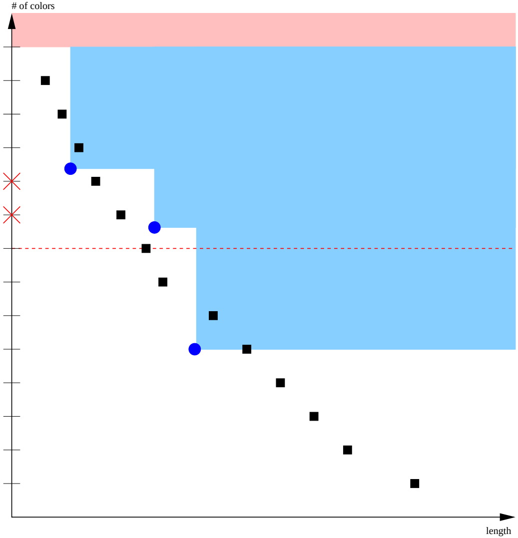
Computation of the lower bound



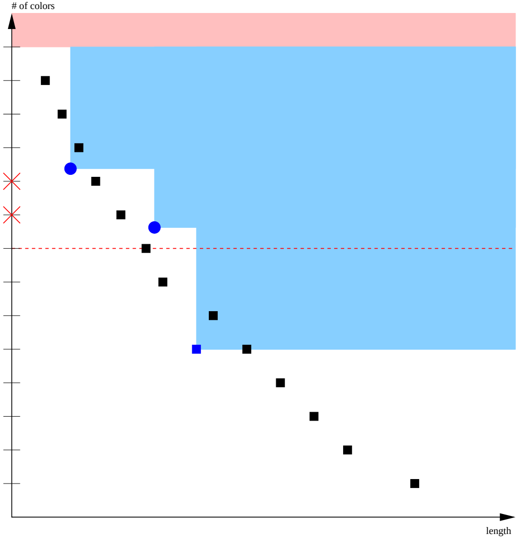
Computation of the lower bound



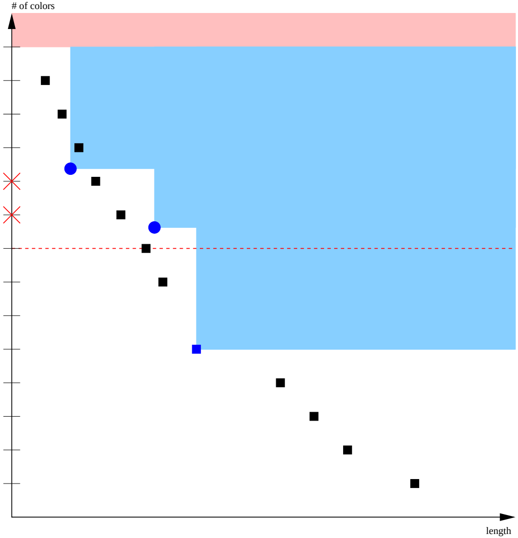
Computation of the lower bound



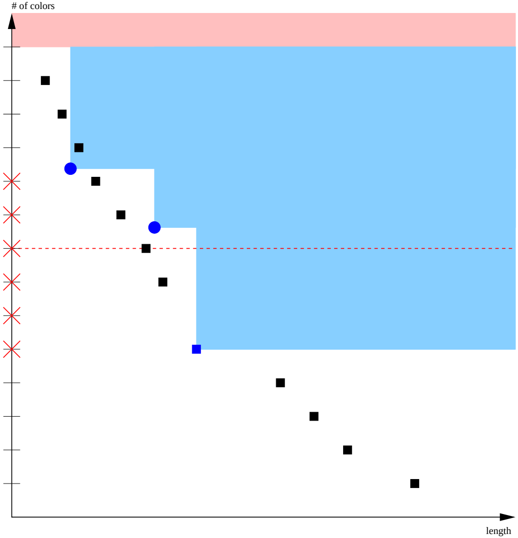
Computation of the lower bound



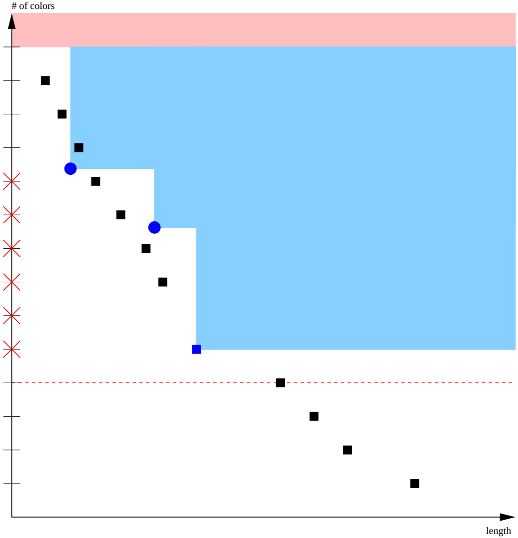
Computation of the lower bound



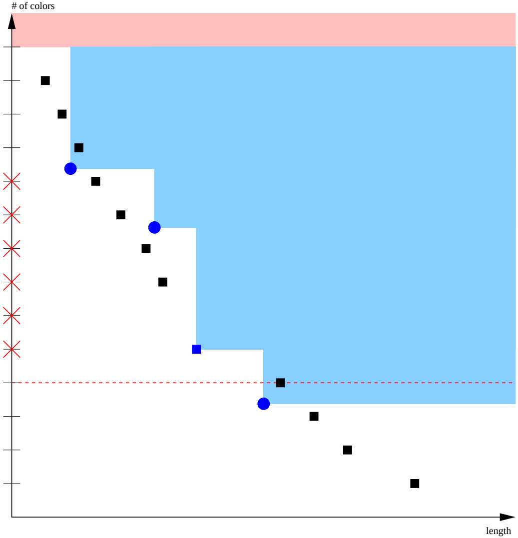
Computation of the lower bound



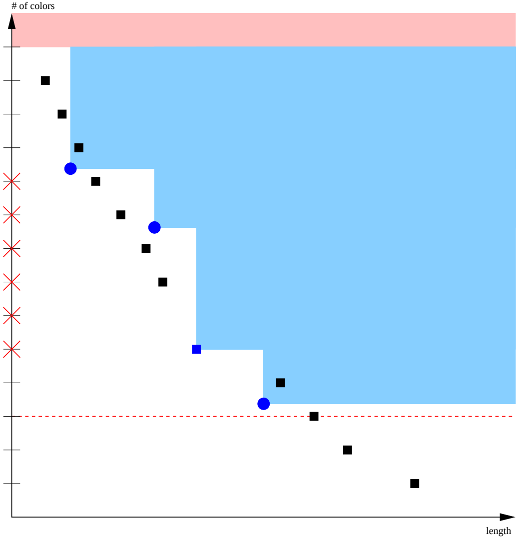
Computation of the lower bound



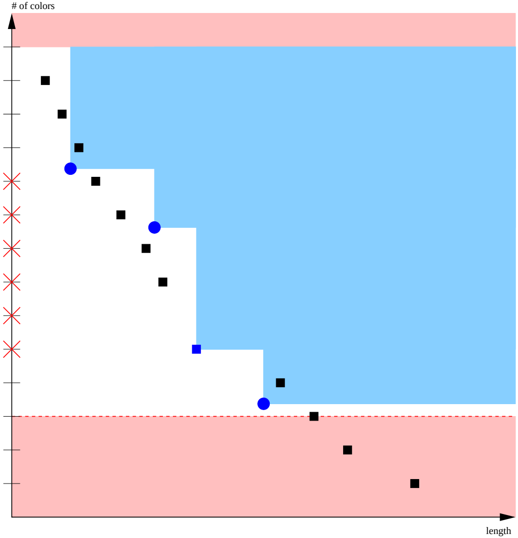
Computation of the lower bound



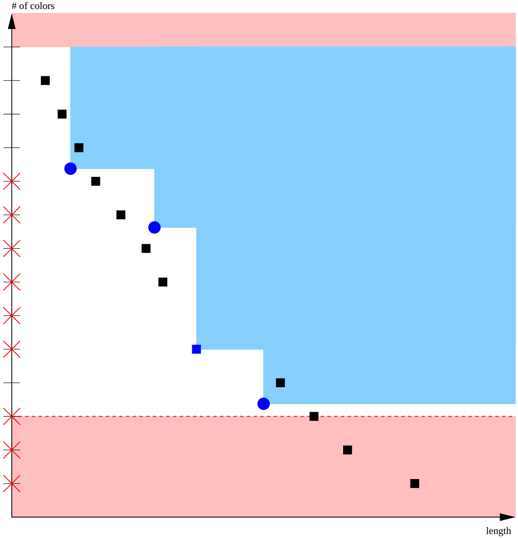
Computation of the lower bound



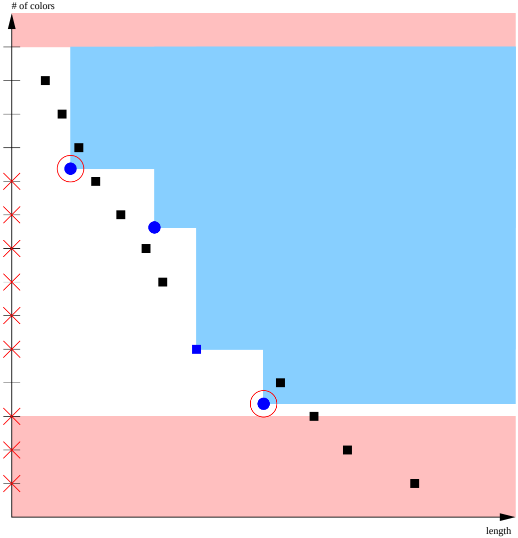
Computation of the lower bound



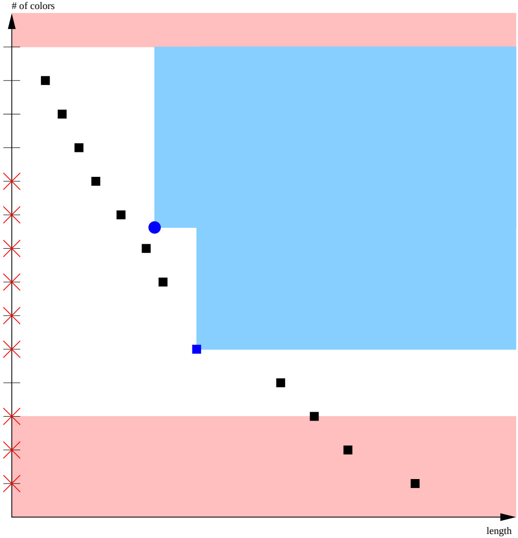
Computation of the lower bound



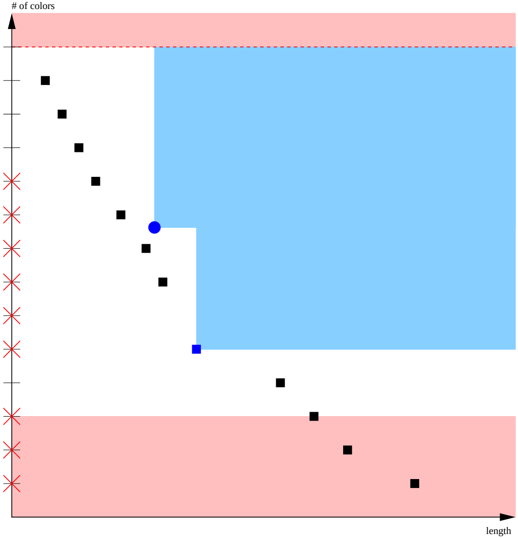
Computation of the lower bound



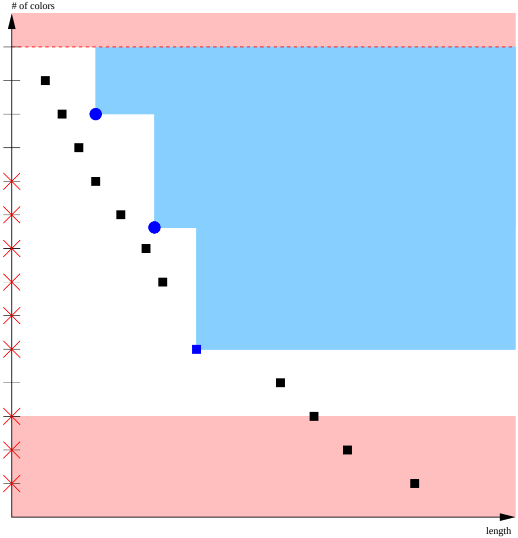
Computation of the lower bound



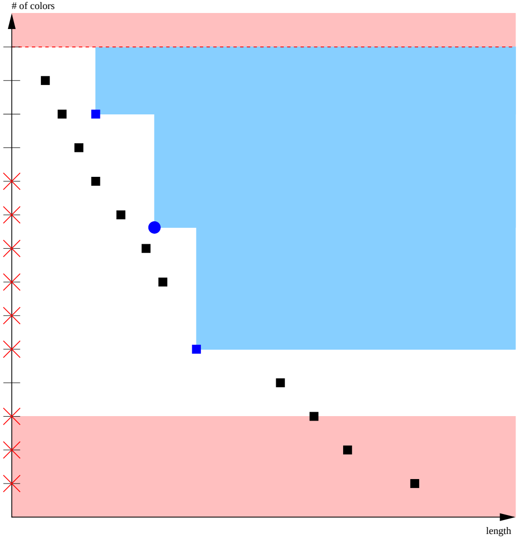
Computation of the lower bound



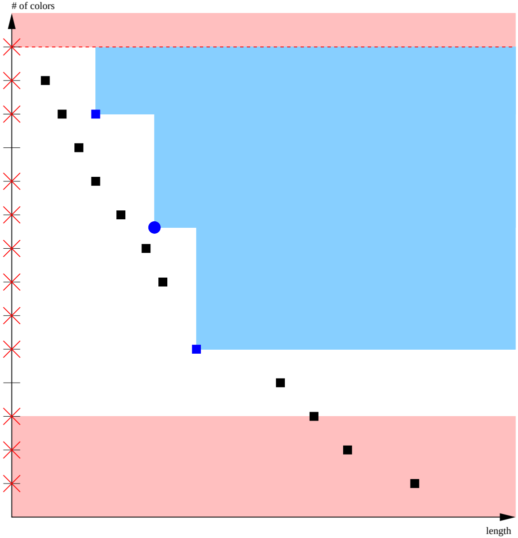
Computation of the lower bound



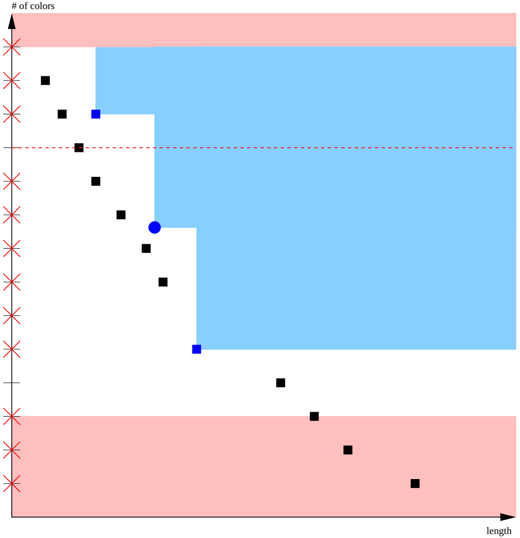
Computation of the lower bound



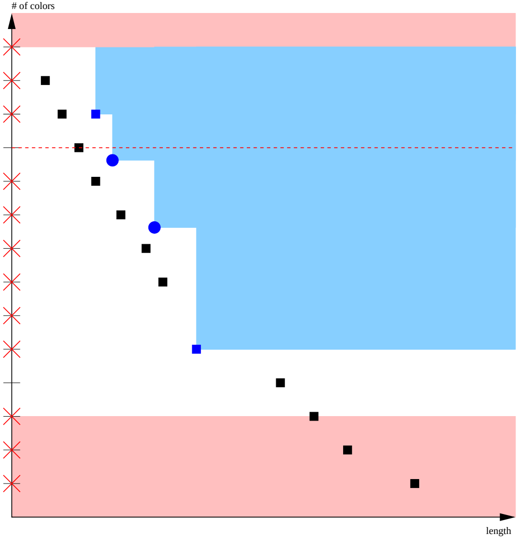
Computation of the lower bound



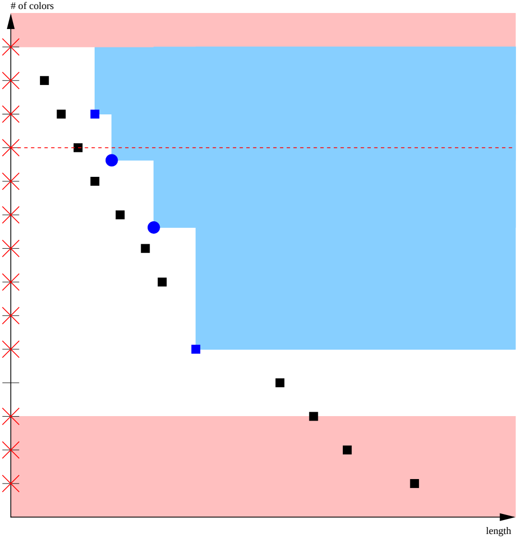
Computation of the lower bound



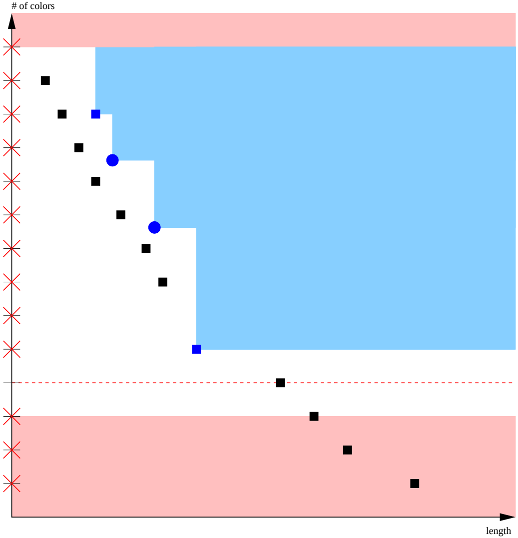
Computation of the lower bound



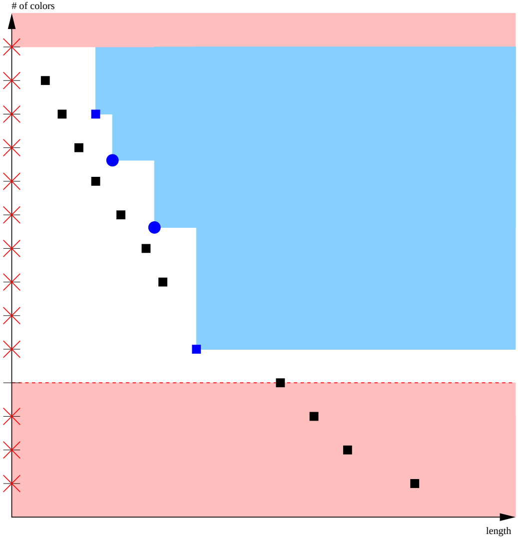
Computation of the lower bound



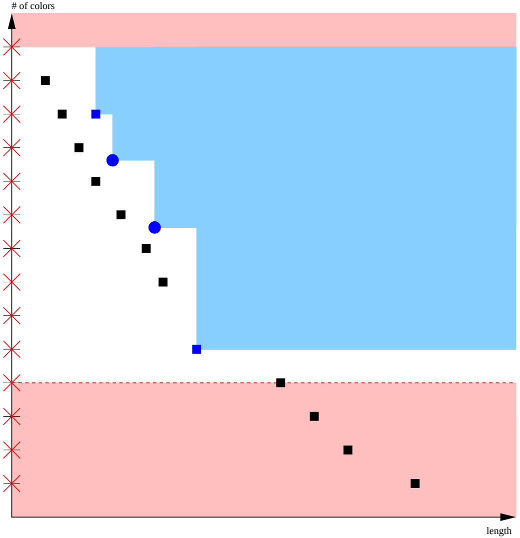
Computation of the lower bound



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]

Test if the color connectivity constraint is also violated

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

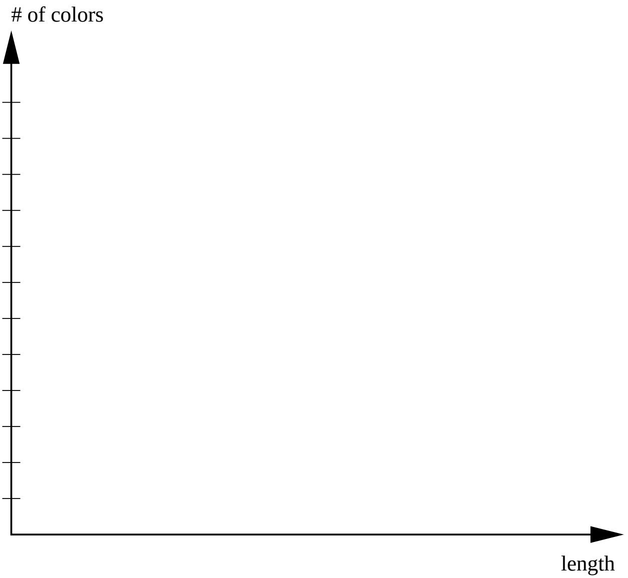
Repeat the following sequence for different values of ϵ

STEP 1: Solve the following mixed integer program :

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

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

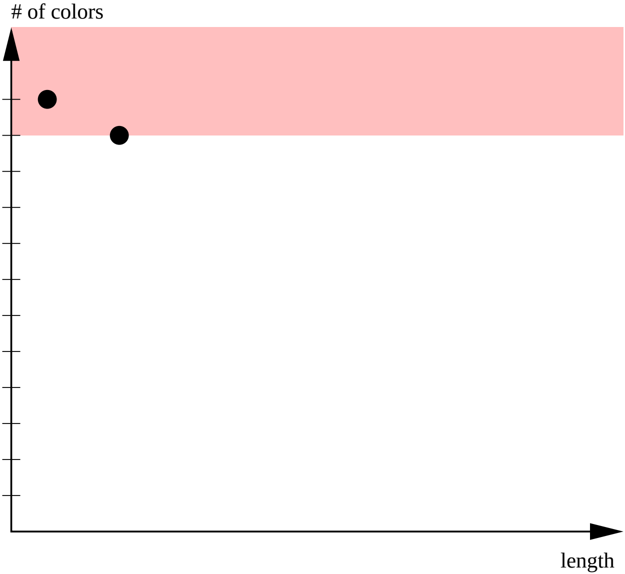
Computation of the initial upper bound

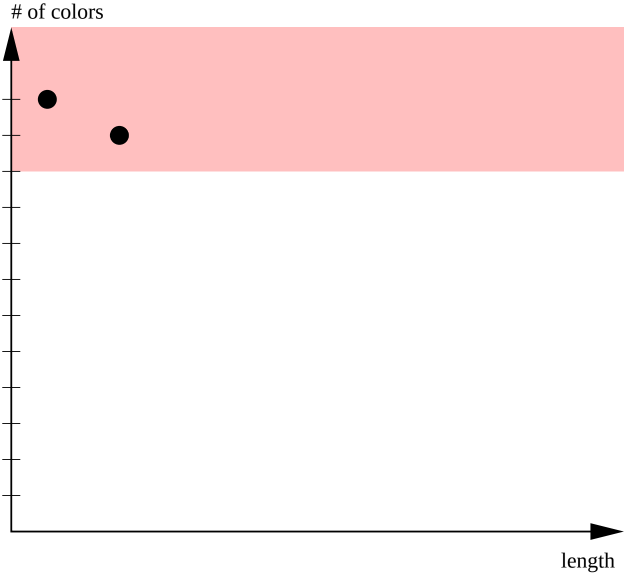



# of colors
length

Computation of the initial upper bound

Computation of the initial upper bound

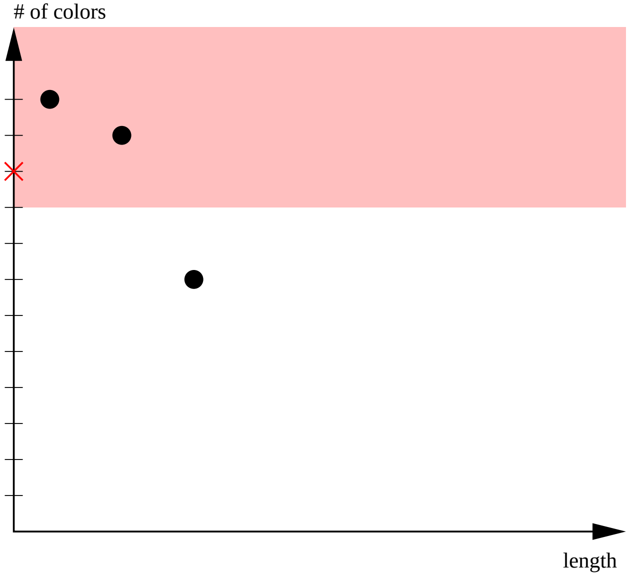



// of colors
length

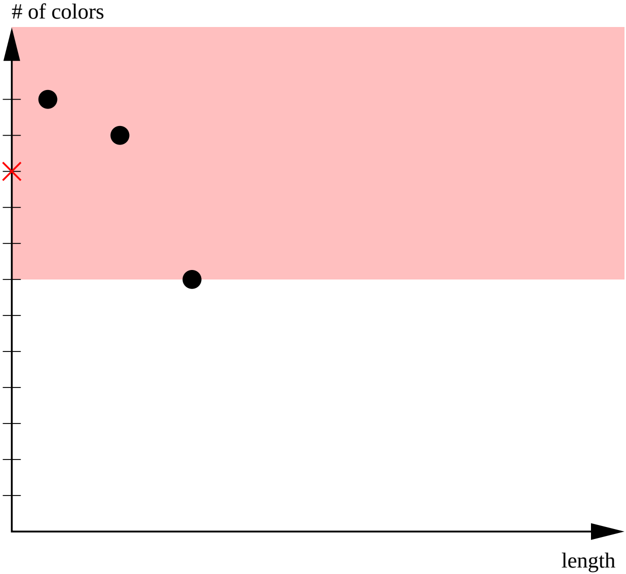
Computation of the initial upper bound

Computation of the initial upper bound

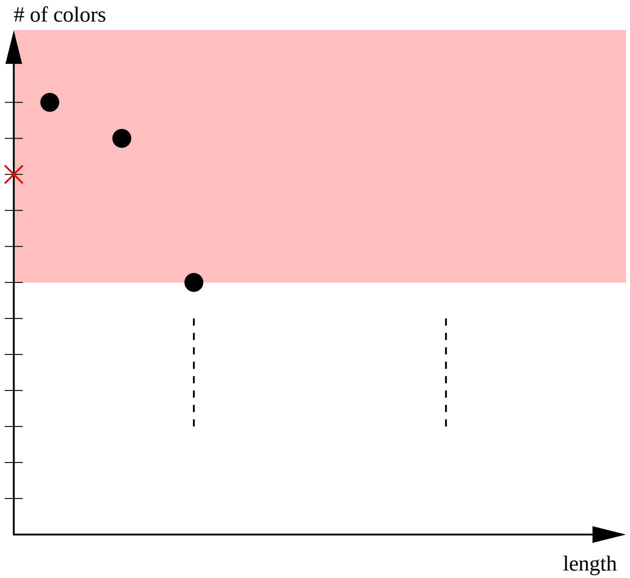
Computation of the initial upper bound



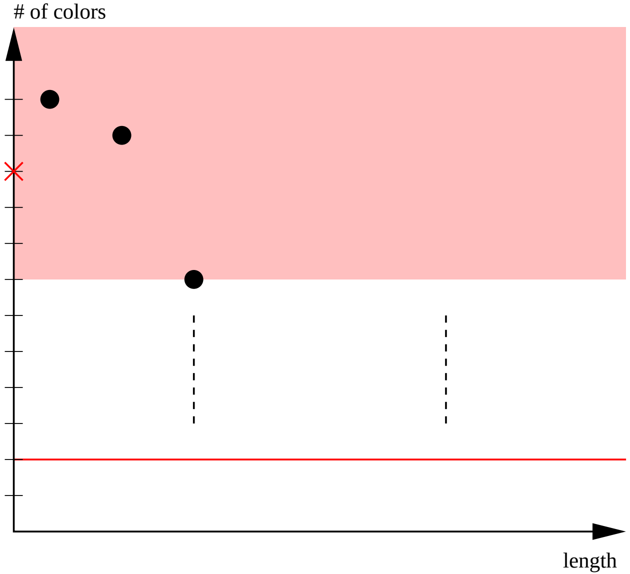
Computation of the initial upper bound



Computation of the initial upper bound



Computation of the initial upper bound



Computational results

$\lvert C \rvert$	$\lvert V \rvert$	#nodes	#u	#x	#cut	#Pareto	#Parub	#time
10	20	77.1	32.1	5.9	31.5	7.9	4.7	0.6
10	30	132.0	50.1	15.4	70.2	8.5	5.5	4.3
10	40	220.4	60.8	48.9	109.4	9.2	5.5	12.0
10	50	411.6	70.8	134.4	148.6	10.0	6.3	37.0
15	20	166.5	77.0	5.8	48.4	9.1	4.4	2.1
15	30	393.9	170.0	26.4	133.4	12.2	4.9	17.9
15	40	551.6	209.3	66.0	208.5	12.6	5.2	52.0
15	50	1880.8	362.0	577.9	340.1	13.5	5.7	207.2
20	20	350.5	169.5	5.3	78.4	10.3	4.5	6.1
20	30	781.9	357.9	32.6	194.8	13.9	4.7	45.9
20	40	1394.5	603.9	92.9	348.8	15.9	5.3	191.9
20	50	2327.0	855.2	307.8	575.8	17.4	5.8	780.0
25	20	429.5	207.2	7.0	84.4	11.1	4.5	8.1
25	30	1596.5	769.5	28.2	304.1	15.2	4.3	147.7
25	40	3200.8	1433.9	166.0	611.0	17.6	4.8	940.9
25	50	5634.5	2376.0	440.7	962.5	20.9	5.5	3915.9
30	20	792.5	391.1	4.6	137.6	12.4	4.5	20.8
30	30	2232.0	1062.5	53.0	364.2	16.4	4.6	259.8
30	40	4866.0	2247.8	184.7	757.2	18.8	4.3	2170.6
30	50	10169.1	4219.0	865.0	1370.0	21.7	5.1	9705.7

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