# A Study of Branching Heuristics for the Car-Sequencing Problem

M. Siala, E.Hebrard, M.J, Huguet

**Laboratory of Analysis and Architecture of Systems (LAAS-CNRS)**

MOGISA Team http://www.laas.fr/MOGISA

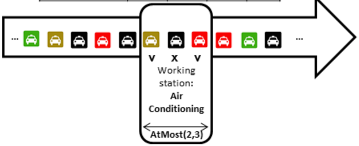
*May* 2012

LAAS-CNRS

# Outline

# The Car-sequencing Problem

| | | | | |
|---|---|---|---|---|
| Air Conditioning | V | X | V | X |
| Sunroof | X | V | X | V |
| MP3 Device | V | V | X | X |




- $n$ vehicles, $k$ classes, $m$ options.
- Demand constraints : Each class $c \in \{1, \ldots, k\}$ is associated with a demand $D_c$.
- Capacity constraints : for each option, we associate two integers $p$ and $q$, such that no subsequence of size $q$ may contain more than $p$ vehicles requiring this option (i.e. a chain of Atmost(p,q) constraints).

## Example

$n = 10, m = 5, k = 6$

ATMost(1,2), ATMost(2,3), ATMost(1,3), ATMost(2,5), ATMost(1,5)

| Class's id | # card | Class's specification |
|---|---|---|
| 0 | #1 | 1 0 1 1 0 |
| 1 | #1 | 0 0 0 1 0 |
| 2 | #2 | 0 1 0 0 1 |
| 3 | #2 | 0 1 0 1 0 |
| 4 | #2 | 1 0 1 0 0 |
| 5 | #2 | 1 1 0 0 0 |

## Example

$n = 10, m = 5, k = 6$

ATMost(1,2), ATMost(2,3), ATMost(1,3), ATMost(2,5), ATMost(1,5)

| Class's id | # card | Class's specification |
|---|---|---|
| 0 | #1 | 1 0 1 1 0 |
| 1 | #1 | 0 0 0 1 0 |
| 2 | #2 | 0 1 0 0 1 |
| 3 | #2 | 0 1 0 1 0 |
| 4 | #2 | 1 0 1 0 0 |
| 5 | #2 | 1 1 0 0 0 |

$\rightarrow$ A possible Solution: 0, 1, 5, 2, 4, 3, 3, 4, 2, 5

# Model description

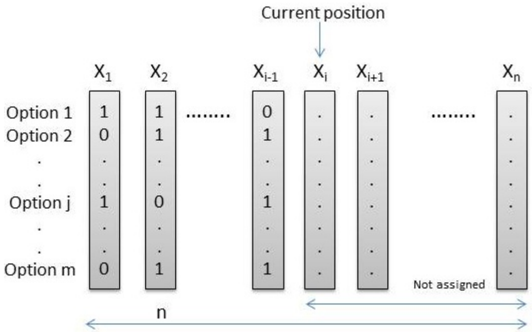
## Variables

- $n$ integer variables $\{x_1, \ldots, x_n\}$ taking values in $\{1, \ldots, k\}$
- $nm$ Boolean variables $\{y_1^1, \ldots, y_n^m\}$, where $y_i^j$ stands for whether the vehicle in the $i^{th}$ slot requires option $j$.

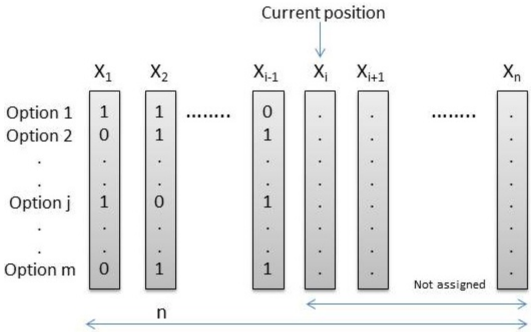
## Constraints

- Demand constraints: Sum, GCC.
- Capacity constraints: Sum, GSC.

# Resolution process

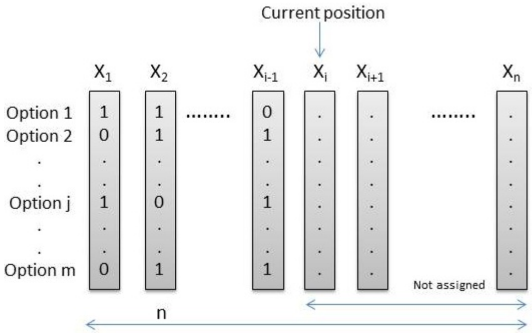


# Resolution process




→ Fail first strategy, [Smith, 96]

# Resolution process




→ Fail first strategy, [Smith, 96]

→ How to choose the most constrained class or option?

# How to choose the most constrained class or option?

# How to choose the most constrained class or option?

- *max option*

# How to choose the most constrained class or option?

- *max option*
- It's much harder to satisfy AtMost(1,7) than AtMost(1,4)

# How to choose the most constrained class or option?

- *max option*
- It's much harder to satisfy AtMost(1,7) than AtMost(1,4)
  $\rightarrow$The capacity $\frac{q}{p}$

# How to choose the most constrained class or option?

- *max option*
- It's much harder to satisfy AtMost(1,7) than AtMost(1,4)
  $\rightarrow$The capacity $\frac{q}{p}$
- The demand

# How to choose the most constrained class or option?

- *max option*
- It's much harder to satisfy AtMost(1,7) than AtMost(1,4)
  →The capacity $\frac{q}{p}$
- The demand
- Example: $d_1 = 3$, AtMost(1,7) and $d_2 = 5$, AtMost(6,8).

# How to choose the most constrained class or option?

- *max option*
- It's much harder to satisfy AtMost(1,7) than AtMost(1,4)
  →The capacity $\frac{q}{p}$
- The demand
- Example: $d_1 = 3$, AtMost(1,7) and $d_2 = 5$, AtMost(6,8).
  To satisfy $d_1$, we need more than 14 slots.

# How to choose the most constrained class or option?

- *max option*
- It's much harder to satisfy AtMost(1,7) than AtMost(1,4)
  $\rightarrow$The capacity $\frac{q}{p}$
- The demand
- Example: $d_1 = 3$, AtMost(1,7) and $d_2 = 5$, AtMost(6,8).
  To satisfy $d_1$, we need more than 14 slots.
  To satisfy $d_2$, 5 slots could be enough.

# How to choose the most constrained class or option?

- *max option*
- It's much harder to satisfy AtMost(1,7) than AtMost(1,4)
  →The capacity $\frac{q}{p}$
- The demand
- Example: $d_1 = 3$, AtMost(1,7) and $d_2 = 5$, AtMost(6,8).
  To satisfy $d_1$, we need more than 14 slots.
  To satisfy $d_2$, 5 slots could be enough.
  It's much harder to satisfy option 1's demand although $d_1 < d_2$.

# How to choose the most constrained class or option?

- *max option*
- It's much harder to satisfy AtMost(1,7) than AtMost(1,4)
  $\rightarrow$The capacity $\frac{q}{p}$
- The demand
- Example: $d_1 = 3$, AtMost(1,7) and $d_2 = 5$, AtMost(6,8).
  To satisfy $d_1$, we need more than 14 slots.
  To satisfy $d_2$, 5 slots could be enough.
  It's much harder to satisfy option 1's demand although $d_1 < d_2$.
  The *load* $\delta$ combines the demand with the capacity $\frac{q}{p}$.
  $\rightarrow \delta = \frac{dq}{p}$.

# How to choose the most constrained class or option?

- *max option*
- It's much harder to satisfy AtMost(1,7) than AtMost(1,4)
  $\rightarrow$The capacity $\frac{q}{p}$
- The demand
- Example: $d_1 = 3$, AtMost(1,7) and $d_2 = 5$, AtMost(6,8).
  To satisfy $d_1$, we need more than 14 slots.
  To satisfy $d_2$, 5 slots could be enough.
  It's much harder to satisfy option 1's demand although $d_1 < d_2$.
  The *load* $\delta$ combines the demand with the capacity $\frac{q}{p}$.
  $\rightarrow \delta = \frac{dq}{p}$.
  The load represents approximately to the number of slots required to mount $d$ times the specific option

# How to choose the most constrained class or option?

- *max option*
- It's much harder to satisfy AtMost(1,7) than AtMost(1,4)
  $\rightarrow$The capacity $\frac{q}{p}$
- The demand
- Example: $d_1 = 3$, AtMost(1,7) and $d_2 = 5$, AtMost(6,8).
  To satisfy $d_1$, we need more than 14 slots.
  To satisfy $d_2$, 5 slots could be enough.
  It's much harder to satisfy option 1's demand although $d_1 < d_2$.
  The *load* $\delta$ combines the demand with the capacity $\frac{q}{p}$.
  $\rightarrow \delta = \frac{dq}{p}$.
  The load represents approximately to the number of slots required to mount $d$ times the specific option
  $\rightarrow \delta_1 = 21$, $\delta_2 = 6.66$.

# New organisation

## New organisation

- Exploration: lexicographical order (*lex*), from middle to sides (*mid*).

## New organisation

- Exploration: lexicographical order (*lex*), from middle to sides (*mid*).
- Branching: *class*, *opt*.

## New organisation

- Exploration: lexicographical order (*lex*), from middle to sides (*mid*).
- Branching: *class*, *opt*.
- Selection:
  - $p/q$
  - $d$
  - $\delta = \frac{dq}{p}$
  - $\rho = \frac{\delta}{n}$
  - $\sigma = n - \delta$.

## New organisation

- Exploration: lexicographical order (*lex*), from middle to sides (*mid*).
- Branching: *class*, *opt*.
- Selection:
  - $p/q$
  - $d$
  - $\delta = \frac{dq}{p}$
  - $\rho = \frac{\delta}{n}$
  - $\sigma = n - \delta$.
- Aggregation:
  - $\leq_{\sum}$
  - $\leq_{Euc}$
  - $\leq_{lex}$

## New organisation

- Exploration: lexicographical order (*lex*), from middle to sides (*mid*).
- Branching: *class*, *opt*.
- Selection:
  - $p/q$
  - $d$
  - $\delta = \frac{dq}{p}$
  - $\rho = \frac{\delta}{n}$
  - $\sigma = n - \delta$.
- Aggregation:
  - $\leq_{\sum}$
  - $\leq_{Euc}$
  - $\leq_{lex}$

$\rightarrow \langle \{lex, mid\}, \{class, opt\}, \{q/p, d, \delta, n{-}\sigma, \rho\}, \{\leq_{\sum}, \leq_{Euc}, \leq_{lex}\} \rangle$

# Aggregation

# Aggregation

$c_1$

$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

# Aggregation

$c_1$ Evaluation

$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

# Aggregation

$c_1$ **Evaluation**

$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} Evaluate\ opt_1 \\ 0 \\ Evaluate\ opt_3 \\ 0 \\ Evaluate\ opt_5 \end{pmatrix}$$

# Aggregation

$c_1$ Evaluation $c_2$ Evaluation

$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} Evaluate\ opt_1 \\ 0 \\ Evaluate\ opt_3 \\ 0 \\ Evaluate\ opt_5 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ Evaluate\ opt_2 \\ 0 \\ Evaluate\ opt_4 \\ Evaluate\ opt_5 \end{pmatrix}$$

# Aggregation

$$c_1 \quad \textcolor{red}{\text{Evaluation}} \qquad c_2 \quad \textcolor{red}{\text{Evaluation}}$$

$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} Evaluate\ opt_1 \\ 0 \\ Evaluate\ opt_3 \\ 0 \\ Evaluate\ opt_5 \end{pmatrix} \qquad \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ Evaluate\ opt_2 \\ 0 \\ Evaluate\ opt_4 \\ Evaluate\ opt_5 \end{pmatrix}$$

$\rightarrow$ How to compare $c_1$ and $c_2$

# Aggregation

$$c_1 \quad \text{Evaluation} \qquad c_2 \quad \text{Evaluation}$$

$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} Evaluate\ opt_1 \\ 0 \\ Evaluate\ opt_3 \\ 0 \\ Evaluate\ opt_5 \end{pmatrix} \qquad \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ Evaluate\ opt_2 \\ 0 \\ Evaluate\ opt_4 \\ Evaluate\ opt_5 \end{pmatrix}$$

$\rightarrow$ How to compare $c_1$ and $c_2$

$\rightarrow \leq_{\sum}$

# Aggregation

$$c_1 \quad \text{Evaluation} \qquad c_2 \quad \text{Evaluation}$$

$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} Evaluate\ opt_1 \\ 0 \\ Evaluate\ opt_3 \\ 0 \\ Evaluate\ opt_5 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ Evaluate\ opt_2 \\ 0 \\ Evaluate\ opt_4 \\ Evaluate\ opt_5 \end{pmatrix}$$

$\rightarrow$ How to compare $c_1$ and $c_2$

$\rightarrow \leq_{\sum}$ , $\leq_{Euc}$

# Aggregation

$c_1$ Evaluation $c_2$ Evaluation

$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} Evaluate\ opt_1 \\ 0 \\ Evaluate\ opt_3 \\ 0 \\ Evaluate\ opt_5 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ Evaluate\ opt_2 \\ 0 \\ Evaluate\ opt_4 \\ Evaluate\ opt_5 \end{pmatrix}$$

$\rightarrow$ How to compare $c_1$ and $c_2$

$\rightarrow \leq_{\sum}$ , $\leq_{Euc}$ and $\leq_{lex}$.

# Load

The load $\delta_j = \frac{d_j q_j}{p_j}$ is tied to the number of slots required to mount $d_j$ times option $j$.

Example: $d_j = 3; p_j = 1; q_j = 3; \rightarrow \delta_j = \frac{d_j q_j}{p_j} = 9$.

# Load

The load $\delta_j = \frac{d_j q_j}{p_j}$ is tied to the number of slots required to mount $d_j$ times option $j$.

Example: $d_j = 3; p_j = 1; q_j = 3; \rightarrow \delta_j = \frac{d_j q_j}{p_j} = 9$.

The exact formula of the load?

# Load

The load $\delta_j = \frac{d_j q_j}{p_j}$ is tied to the number of slots required to mount $d_j$ times option $j$.

Example: $d_j = 3; p_j = 1; q_j = 3; \rightarrow \delta_j = \frac{d_j q_j}{p_j} = 9$.

The exact formula of the load?

$\rightarrow$ Alternative definition for the load of an option:

$$\delta'_j = q_j(\lceil d_j/p_j \rceil - 1) + \begin{cases} p_j & \text{if } d_j \bmod p_j = 0 \\ d_j \bmod p_j & \text{otherwise} \end{cases}$$

# Load

The load $\delta_j = \frac{d_j q_j}{p_j}$ is tied to the number of slots required to mount $d_j$ times option $j$.

Example: $d_j = 3; p_j = 1; q_j = 3; \rightarrow \delta_j = \frac{d_j q_j}{p_j} = 9$.

The exact formula of the load?

$\rightarrow$ Alternative definition for the load of an option:

$$\delta'_j = q_j(\lceil d_j/p_j \rceil - 1) + \begin{cases} p_j & \text{if } d_j \bmod p_j = 0 \\ d_j \bmod p_j & \text{otherwise} \end{cases}$$

For each option $j$, $\delta'_j$ is a lower bound on the number of required slots

# Load

The load $\delta_j = \frac{d_j q_j}{p_j}$ is tied to the number of slots required to mount $d_j$ times option $j$.

Example: $d_j = 3; p_j = 1; q_j = 3; \rightarrow \delta_j = \frac{d_j q_j}{p_j} = 9$.

The exact formula of the load?

$\rightarrow$ Alternative definition for the load of an option:

$$\delta'_j = q_j(\lceil d_j/p_j \rceil - 1) + \begin{cases} p_j & \text{if } d_j \bmod p_j = 0 \\ d_j \bmod p_j & \text{otherwise} \end{cases}$$

For each option $j$, $\delta'_j$ is a lower bound on the number of required slots

Example: $d_j = 3; p_j = 1; q_j = 3; \rightarrow \delta'_j = 7$.

# Load

The load $\delta_j = \frac{d_j q_j}{p_j}$ is tied to the number of slots required to mount $d_j$ times option $j$.

Example: $d_j = 3; p_j = 1; q_j = 3; \rightarrow \delta_j = \frac{d_j q_j}{p_j} = 9$.

The exact formula of the load?

$\rightarrow$ Alternative definition for the load of an option:

$$\delta'_j = q_j(\lceil d_j/p_j \rceil - 1) + \begin{cases} p_j & \text{if } d_j \bmod p_j = 0 \\ d_j \bmod p_j & \text{otherwise} \end{cases}$$

For each option $j$, $\delta'_j$ is a lower bound on the number of required slots

Example: $d_j = 3; p_j = 1; q_j = 3; \rightarrow \delta'_j = 7$.

$\rightarrow$ new slack $\sigma'$ and new usage rate $\rho'$.

# New filtering rule

Suppose that all variables up to a rank $i - 1$ are ground.

**Theorem**

In the case of lexicographical branching :

- If $\delta' > n - i + 1$, then we should fail.
- When $\delta' = n - i + 1$, we can filter out some values.
  - If $d$ mod $p = 0$, we impose $y_i = 1$ for all $i$ such that $i$ mod $q < p$.
  - If $d$ mod $p \neq 0$, we impose $y_i = 1$ for all $i$ such that $i$ mod $q < (d$ mod $p)$.

# Pruning rule

Figure: Filtering when $d \bmod p = 0$

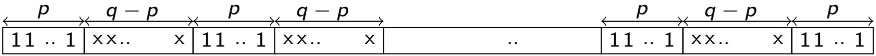
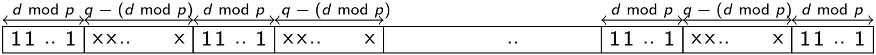



Figure: Filtering when $d \bmod p \neq 0$



# Filtering Rule

## Mistral results on the first set (70 sat)

| | | Basic model | | Filtering rule | |
|---|---|---|---|---|---|
| Eval. | Aggr. | % sol | time | % sol | time |
| 1 | - | 52 | 41.15 | 94 | 23.46 |
| $q/p$ | $\leq_{\sum}$ | 34 | 7 | 100 | 0.01 |
| | $\leq_{Euc}$ | 38 | 43.67 | 97 | 0.01 |
| | $\leq_{lex}$ | 38 | 55.38 | 97 | 0.01 |
| $d$ | $\leq_{\sum}$ | 85 | 17.38 | 100 | 0.01 |
| | $\leq_{Euc}$ | 81 | 0.01 | 100 | 0.01 |
| | $\leq_{lex}$ | 75 | 0.03 | 100 | 0.03 |
| $\rho$ | $\leq_{\sum}$ | 100 | 0.01 | 100 | 0.01 |
| | $\leq_{Euc}$ | 100 | 0.01 | 100 | 0.01 |
| | $\leq_{lex}$ | 100 | 0.03 | 100 | 0.02 |
| $\rho'$ | $\leq_{\sum}$ | 100 | 0.03 | 100 | 0.01 |
| | $\leq_{Euc}$ | 100 | 0.01 | 100 | 0.01 |
| | $\leq_{lex}$ | 100 | 0.03 | 100 | 0.03 |

# Filtering Rule

## Mistral results on the first set (70 sat)

| | | Basic model | | Filtering rule | |
|---|---|---|---|---|---|
| Eval. | Aggr. | % sol | time | % sol | time |
| 1 | - | 52 | 41.15 | 94 | 23.46 |
| $q/p$ | $\leq_{\sum}$ | 34 | 7 | 100 | 0.01 |
| | $\leq_{Euc}$ | 38 | 43.67 | 97 | 0.01 |
| | $\leq_{lex}$ | 38 | 55.38 | 97 | 0.01 |
| $d$ | $\leq_{\sum}$ | 85 | 17.38 | 100 | 0.01 |
| | $\leq_{Euc}$ | 81 | 0.01 | 100 | 0.01 |
| | $\leq_{lex}$ | 75 | 0.03 | 100 | 0.03 |
| $\rho$ | $\leq_{\sum}$ | 100 | 0.01 | 100 | 0.01 |
| | $\leq_{Euc}$ | 100 | 0.01 | 100 | 0.01 |
| | $\leq_{lex}$ | 100 | 0.03 | 100 | 0.02 |
| $\rho'$ | $\leq_{\sum}$ | 100 | 0.03 | 100 | 0.01 |
| | $\leq_{Euc}$ | 100 | 0.01 | 100 | 0.01 |
| | $\leq_{lex}$ | 100 | 0.03 | 100 | 0.03 |

# Discrepancy Search methods

## Mistral results on the second set (4 sat)

| | | Basic model | | | | | Filtering rule | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Eval. | Aggr. | c.b | lds | lds-b | prlds | prlds-b | c.b | lds | lds-b | prlds | prlds-b |
| 1 | – | 0 | 25 | 25 | 25 | 25 | 25 | 100 | 100 | 100 | 100 |
| $q/p$ | $\leq_{\sum}$ | 0 | 50 | 25 | 50 | 25 | 25 | 100 | 100 | 100 | 100 |
| | $\leq_{Euc}$ | 0 | 50 | 25 | 50 | 25 | 25 | 100 | 100 | 100 | 100 |
| | $\leq_{lex}$ | 0 | 25 | 25 | 25 | 25 | 0 | 100 | 100 | 100 | 100 |
| $d$ | $\leq_{\sum}$ | 25 | 100 | 75 | 100 | 75 | 50 | 100 | 100 | 100 | 100 |
| | $\leq_{Euc}$ | 25 | 100 | 75 | 75 | 75 | 50 | 100 | 100 | 100 | 100 |
| | $\leq_{lex}$ | 0 | 75 | 75 | 75 | 75 | 50 | 100 | 100 | 100 | 100 |
| $\rho$ | $\leq_{\sum}$ | 25 | 100 | 100 | 100 | 100 | 50 | 100 | 100 | 100 | 100 |
| | $\leq_{Euc}$ | 25 | 100 | 100 | 100 | 100 | 50 | 100 | 100 | 100 | 100 |
| | $\leq_{lex}$ | 50 | 100 | 100 | 100 | 100 | 50 | 100 | 100 | 100 | 100 |
| $\rho'$ | $\leq_{\sum}$ | 25 | 100 | 100 | 100 | 100 | 25 | 100 | 100 | 100 | 100 |
| | $\leq_{Euc}$ | 25 | 100 | 100 | 100 | 100 | 25 | 100 | 100 | 100 | 100 |
| | $\leq_{lex}$ | 75 | 100 | 100 | 100 | 100 | 75 | 100 | 100 | 100 | 100 |

# Discrepancy Search methods

## Mistral results on the second set (4 sat)

| | | Basic model | | | | | Filtering rule | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Eval. | Aggr. | c.b | lds | lds-b | prlds | prlds-b | c.b | lds | lds-b | prlds | prlds-b |
| 1 | – | 0 | 25 | 25 | 25 | 25 | 25 | 100 | 100 | 100 | 100 |
| $q/p$ | $\leq_{\sum}$ | 0 | 50 | 25 | 50 | 25 | 25 | 100 | 100 | 100 | 100 |
| | $\leq_{Euc}$ | 0 | 50 | 25 | 50 | 25 | 25 | 100 | 100 | 100 | 100 |
| | $\leq_{lex}$ | 0 | 25 | 25 | 25 | 25 | 0 | 100 | 100 | 100 | 100 |
| $d$ | $\leq_{\sum}$ | 25 | 100 | 75 | 100 | 75 | 50 | 100 | 100 | 100 | 100 |
| | $\leq_{Euc}$ | 25 | 100 | 75 | 75 | 75 | 50 | 100 | 100 | 100 | 100 |
| | $\leq_{lex}$ | 0 | 75 | 75 | 75 | 75 | 50 | 100 | 100 | 100 | 100 |
| $\rho$ | $\leq_{\sum}$ | 25 | 100 | 100 | 100 | 100 | 50 | 100 | 100 | 100 | 100 |
| | $\leq_{Euc}$ | 25 | 100 | 100 | 100 | 100 | 50 | 100 | 100 | 100 | 100 |
| | $\leq_{lex}$ | 50 | 100 | 100 | 100 | 100 | 50 | 100 | 100 | 100 | 100 |
| $\rho'$ | $\leq_{\sum}$ | 25 | 100 | 100 | 100 | 100 | 25 | 100 | 100 | 100 | 100 |
| | $\leq_{Euc}$ | 25 | 100 | 100 | 100 | 100 | 25 | 100 | 100 | 100 | 100 |
| | $\leq_{lex}$ | 75 | 100 | 100 | 100 | 100 | 75 | 100 | 100 | 100 | 100 |

# Discrepancy Search methods

## Mistral results on the second set (4 sat)

| | | Basic model | | | | | Filtering rule | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Eval. | Aggr. | c.b | lds | lds-b | prlds | prlds-b | c.b | lds | lds-b | prlds | prlds-b |
| 1 | – | 0 | 25 | 25 | 25 | 25 | 25 | 100 | 100 | 100 | 100 |
| $q/p$ | $\leq_{\sum}$ | 0 | 50 | 25 | 50 | 25 | 25 | 100 | 100 | 100 | 100 |
| | $\leq_{Euc}$ | 0 | 50 | 25 | 50 | 25 | 25 | 100 | 100 | 100 | 100 |
| | $\leq_{lex}$ | 0 | 25 | 25 | 25 | 25 | 0 | 100 | 100 | 100 | 100 |
| $d$ | $\leq_{\sum}$ | 25 | 100 | 75 | 100 | 75 | 50 | 100 | 100 | 100 | 100 |
| | $\leq_{Euc}$ | 25 | 100 | 75 | 75 | 75 | 50 | 100 | 100 | 100 | 100 |
| | $\leq_{lex}$ | 0 | 75 | 75 | 75 | 75 | 50 | 100 | 100 | 100 | 100 |
| $\rho$ | $\leq_{\sum}$ | 25 | 100 | 100 | 100 | 100 | 50 | 100 | 100 | 100 | 100 |
| | $\leq_{Euc}$ | 25 | 100 | 100 | 100 | 100 | 50 | 100 | 100 | 100 | 100 |
| | $\leq_{lex}$ | 50 | 100 | 100 | 100 | 100 | 50 | 100 | 100 | 100 | 100 |
| $\rho'$ | $\leq_{\sum}$ | 25 | 100 | 100 | 100 | 100 | 25 | 100 | 100 | 100 | 100 |
| | $\leq_{Euc}$ | 25 | 100 | 100 | 100 | 100 | 25 | 100 | 100 | 100 | 100 |
| | $\leq_{lex}$ | 75 | 100 | 100 | 100 | 100 | 75 | 100 | 100 | 100 | 100 |

# Discrepancy Search methods

## Mistral results on the second set (4 sat)

| | | Basic model | | | | | Filtering rule | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Eval. | Aggr. | c.b | lds | lds-b | prlds | prlds-b | c.b | lds | lds-b | prlds | prlds-b |
| 1 | – | 0 | 25 | 25 | 25 | 25 | 25 | 100 | 100 | 100 | 100 |
| $q/p$ | $\leq_{\sum}$ | 0 | 50 | 25 | 50 | 25 | 25 | 100 | 100 | 100 | 100 |
| | $\leq_{Euc}$ | 0 | 50 | 25 | 50 | 25 | 25 | 100 | 100 | 100 | 100 |
| | $\leq_{lex}$ | 0 | 25 | 25 | 25 | 25 | 0 | 100 | 100 | 100 | 100 |
| $d$ | $\leq_{\sum}$ | 25 | 100 | 75 | 100 | 75 | 50 | 100 | 100 | 100 | 100 |
| | $\leq_{Euc}$ | 25 | 100 | 75 | 75 | 75 | 50 | 100 | 100 | 100 | 100 |
| | $\leq_{lex}$ | 0 | 75 | 75 | 75 | 75 | 50 | 100 | 100 | 100 | 100 |
| $\rho$ | $\leq_{\sum}$ | 25 | 100 | 100 | 100 | 100 | 50 | 100 | 100 | 100 | 100 |
| | $\leq_{Euc}$ | 25 | 100 | 100 | 100 | 100 | 50 | 100 | 100 | 100 | 100 |
| | $\leq_{lex}$ | 50 | 100 | 100 | 100 | 100 | 50 | 100 | 100 | 100 | 100 |
| $\rho'$ | $\leq_{\sum}$ | 25 | 100 | 100 | 100 | 100 | 25 | 100 | 100 | 100 | 100 |
| | $\leq_{Euc}$ | 25 | 100 | 100 | 100 | 100 | 25 | 100 | 100 | 100 | 100 |
| | $\leq_{lex}$ | 75 | 100 | 100 | 100 | 100 | 75 | 100 | 100 | 100 | 100 |

# Evaluation of the Exploration & Branching Strategies

| Eval. Ord. | Aggr. Br. | Aggr. Exp. | set 1 (70 sat) %sol | set 1 (70 sat) time | set 2 (4 sat) %sol | set 2 (4 sat) time | set 3 (7 sat) %sol | set 3 (7 sat) time |
|---|---|---|---|---|---|---|---|---|
| 1 | Clas | Lex. | 100 | 17.08 | 100 | 211.55 | 0 | - |
| | | Mid. | 98 | 52.55 | 25 | 0.20 | 0 | - |
| $q/p$ | Opt | Lex. | 75 | 44.93 | 0 | - | 0 | - |
| | | Mid. | 80 | 9.35 | 0 | - | 0 | - |
| | Clas | Lex. | 98 | 15.45 | 0 | - | 0 | - |
| | | Mid. | 98 | 1.01 | 25 | 182.12 | 0 | - |
| $d$ | Opt | Lex. | 88 | 2.78 | 75 | 128.32 | 0 | - |
| | | Mid. | 90 | 17.74 | 25 | 0.22 | 14 | 962.88 |
| | Clas | Lex. | 100 | 1.20 | 50 | 64.96 | 57 | 630.76 |
| | | Mid. | 100 | 1.08 | 100 | 129.07 | 57 | 606.10 |
| $n-\sigma$ | Opt | Lex. | 44 | 29.47 | 75 | 803.88 | 0 | - |
| | | Mid. | 51 | 57.95 | 25 | 262.18 | 0 | - |
| | Clas | Lex. | 100 | 1.19 | 50 | 31.98 | 42 | 484.37 |
| | | Mid. | 100 | 1.05 | 75 | 263.61 | 42 | 642.65 |
| $\rho$ | Opt | Lex. | 98 | 16.60 | 75 | 58.16 | 14 | 875.62 |
| | | Mid. | 100 | 28.26 | 25 | 0.81 | 14 | 267.53 |
| | Clas | Lex. | 100 | 1.19 | 50 | 31.98 | 42 | 484.56 |
| | | Mid. | 100 | 1.05 | 75 | 218.22 | 42 | 607.49 |

## Evaluation of the Exploration & Branching Strategies

| Eval. Ord. | Aggr. Br. | Exp. | set 1 (70 sat) %sol | set 1 (70 sat) time | set 2 (4 sat) %sol | set 2 (4 sat) time | set 3 (7 sat) %sol | set 3 (7 sat) time |
|---|---|---|---|---|---|---|---|---|
| 1 | Clas | Lex. | 100 | 17.08 | 100 | 211.55 | 0 | - |
| | | Mid. | 98 | 52.55 | 25 | 0.20 | 0 | - |
| $q/p$ | Opt | Lex. | 75 | 44.93 | 0 | - | 0 | - |
| | | Mid. | 80 | 9.35 | 0 | - | 0 | - |
| | Clas | Lex. | 98 | 15.45 | 0 | - | 0 | - |
| | | Mid. | 98 | 1.01 | 25 | 182.12 | 0 | - |
| $d$ | Opt | Lex. | 88 | 2.78 | 75 | 128.32 | 0 | - |
| | | Mid. | 90 | 17.74 | 25 | 0.22 | 14 | 962.88 |
| | Clas | Lex. | 100 | 1.20 | 50 | 64.96 | 57 | 630.76 |
| | | Mid. | 100 | 1.08 | 100 | 129.07 | 57 | 606.10 |
| $n - \sigma$ | Opt | Lex. | 44 | 29.47 | 75 | 803.88 | 0 | - |
| | | Mid. | 51 | 57.95 | 25 | 262.18 | 0 | - |
| | Clas | Lex. | 100 | 1.19 | 50 | 31.98 | 42 | 484.37 |
| | | Mid. | 100 | 1.05 | 75 | 263.61 | 42 | 642.65 |
| $\rho$ | Opt | Lex. | 98 | 16.60 | 75 | 58.16 | 14 | 875.62 |
| | | Mid. | 100 | 28.26 | 25 | 0.81 | 14 | 267.53 |
| | Clas | Lex. | 100 | 1.19 | 50 | 31.98 | 42 | 484.56 |
| | | Mid. | 100 | 1.05 | 75 | 218.22 | 42 | 607.49 |

# Evaluation of the Exploration & Branching Strategies

| Eval. Ord. | Aggr. Br. | Aggr. Exp. | set 1 (70 sat) %sol | set 1 (70 sat) time | set 2 (4 sat) %sol | set 2 (4 sat) time | set 3 (7 sat) %sol | set 3 (7 sat) time |
|---|---|---|---|---|---|---|---|---|
| 1 | Clas | Lex. | 100 | 17.08 | 100 | 211.55 | 0 | - |
| | | Mid. | 98 | 52.55 | 25 | 0.20 | 0 | - |
| $q/p$ | Opt | Lex. | 75 | 44.93 | 0 | - | 0 | - |
| | | Mid. | 80 | 9.35 | 0 | - | 0 | - |
| | Clas | Lex. | 98 | 15.45 | 0 | - | 0 | - |
| | | Mid. | 98 | 1.01 | 25 | 182.12 | 0 | - |
| $d$ | Opt | Lex. | 88 | 2.78 | 75 | 128.32 | 0 | - |
| | | Mid. | 90 | 17.74 | 25 | 0.22 | 14 | 962.88 |
| | Clas | Lex. | 100 | 1.20 | 50 | 64.96 | 57 | 630.76 |
| | | Mid. | 100 | 1.08 | 100 | 129.07 | 57 | 606.10 |
| $n - \sigma$ | Opt | Lex. | 44 | 29.47 | 75 | 803.88 | 0 | - |
| | | Mid. | 51 | 57.95 | 25 | 262.18 | 0 | - |
| | Clas | Lex. | 100 | 1.19 | 50 | 31.98 | 42 | 484.37 |
| | | Mid. | 100 | 1.05 | 75 | 263.61 | 42 | 642.65 |
| $\rho$ | Opt | Lex. | 98 | 16.60 | 75 | 58.16 | 14 | 875.62 |
| | | Mid. | 100 | 28.26 | 25 | 0.81 | 14 | 267.53 |
| | Clas | Lex. | 100 | 1.19 | 50 | 31.98 | 42 | 484.56 |
| | | Mid. | 100 | 1.05 | 75 | 218.22 | 42 | 607.49 |

# Evaluation of the Exploration & Branching Strategies

| Eval. Ord. | Aggr. Br. | Exp. | set 1 (70 sat) %sol | set 1 (70 sat) time | set 2 (4 sat) %sol | set 2 (4 sat) time | set 3 (7 sat) %sol | set 3 (7 sat) time |
|---|---|---|---|---|---|---|---|---|
| 1 | Clas | Lex. | 100 | 17.08 | 100 | 211.55 | 0 | - |
| | | Mid. | 98 | 52.55 | 25 | 0.20 | 0 | - |
| $q/p$ | Opt | Lex. | 75 | 44.93 | 0 | - | 0 | - |
| | | Mid. | 80 | 9.35 | 0 | - | 0 | - |
| | Clas | Lex. | 98 | 15.45 | 0 | - | 0 | - |
| | | Mid. | 98 | 1.01 | 25 | 182.12 | 0 | - |
| $d$ | Opt | Lex. | 88 | 2.78 | 75 | 128.32 | 0 | - |
| | | Mid. | 90 | 17.74 | 25 | 0.22 | 14 | 962.88 |
| | Clas | Lex. | 100 | 1.20 | 50 | 64.96 | 57 | 630.76 |
| | | Mid. | 100 | 1.08 | 100 | 129.07 | 57 | 606.10 |
| $n-\sigma$ | Opt | Lex. | 44 | 29.47 | 75 | 803.88 | 0 | - |
| | | Mid. | 51 | 57.95 | 25 | 262.18 | 0 | - |
| | Clas | Lex. | 100 | 1.19 | 50 | 31.98 | 42 | 484.37 |
| | | Mid. | 100 | 1.05 | 75 | 263.61 | 42 | 642.65 |
| $\rho$ | Opt | Lex. | 98 | 16.60 | 75 | 58.16 | 14 | 875.62 |
| | | Mid. | 100 | 28.26 | 25 | 0.81 | 14 | 267.53 |
| | Clas | Lex. | 100 | 1.19 | 50 | 31.98 | 42 | 484.56 |
| | | Mid. | 100 | 1.05 | 75 | 218.22 | 42 | 607.49 |

# Evaluation of the Exploration & Branching Strategies

| Eval. Ord. | Aggr. Br. | Exp. | set 1 (70 sat) %sol | set 1 (70 sat) time | set 2 (4 sat) %sol | set 2 (4 sat) time | set 3 (7 sat) %sol | set 3 (7 sat) time |
|---|---|---|---|---|---|---|---|---|
| 1 | Clas | Lex. | 100 | 17.08 | 100 | 211.55 | 0 | - |
| | | Mid. | 98 | 52.55 | 25 | 0.20 | 0 | - |
| $q/p$ | Opt | Lex. | 75 | 44.93 | 0 | - | 0 | - |
| | | Mid. | 80 | 9.35 | 0 | - | 0 | - |
| | Clas | Lex. | 98 | 15.45 | 0 | - | 0 | - |
| | | Mid. | 98 | 1.01 | 25 | 182.12 | 0 | - |
| $d$ | Opt | Lex. | 88 | 2.78 | 75 | 128.32 | 0 | - |
| | | Mid. | 90 | 17.74 | 25 | 0.22 | 14 | 962.88 |
| | Clas | Lex. | 100 | 1.20 | 50 | 64.96 | 57 | 630.76 |
| | | Mid. | 100 | 1.08 | 100 | 129.07 | 57 | 606.10 |
| $n - \sigma$ | Opt | Lex. | 44 | 29.47 | 75 | 803.88 | 0 | - |
| | | Mid. | 51 | 57.95 | 25 | 262.18 | 0 | - |
| | Clas | Lex. | 100 | 1.19 | 50 | 31.98 | 42 | 484.37 |
| | | Mid. | 100 | 1.05 | 75 | 263.61 | 42 | 642.65 |
| $\rho$ | Opt | Lex. | 98 | 16.60 | 75 | 58.16 | 14 | 875.62 |
| | | Mid. | 100 | 28.26 | 25 | 0.81 | 14 | 267.53 |
| | Clas | Lex. | 100 | 1.19 | 50 | 31.98 | 42 | 484.56 |
| | | Mid. | 100 | 1.05 | 75 | 218.22 | 42 | 607.49 |

## Comparison with existing methods

| | FC | | cREG | | LNS | | ILOG | | Mistral | |
|---|---|---|---|---|---|---|---|---|---|---|
| | %sol | time | %sol | time | %sol | time | %sol | time | %sol | time |
| Set 1 (70 sat) | 91 | few s | 58 | $> 30s$ | 98 | few s | 100 | 1.01s | 100 | $0,01s$ |
| Set 2 (4 sat) | 75 | $12,66s$ | 100 | $0,55s$ | 100 | 206s | 100 | 129.07s | 100 | $< 1s$ |
| Set 3 (7 sat) | N.A | N.A | N.A | N.A | N.A | N.A | 57 | 606.10s | 42 | $0,03s$ |

## Conclusion

- We revisited the most common heuristics for this problem.
- New structure for the heuristics.
- A simple but very useful filtering rule
- Interested results with untested combination of heuristics.
- The new filtering algorithm, combined with discrepancy search methods, outperforms existing CP approaches.

## Future Research

- [Submitted paper] An Optimal Arc Consistency Algorithm for a Chain of Atmost Constraints with Cardinality, CP'12
- Using filtering techniques in SMT-Solvers.

# Thank you for your attention!

Questions?