Crew Assignment via Constraint Programming: Integrating Column Generation and Heuristic Tree Search

Kyriakos Zervoudakis
Panagiotis Stamatopoulos
UNIVERSITY OF ATHENS

Meinolf Selmann
Torsten Fahle
UNIVERSITY OF PADERBORN

EU Esprit-Project Parrot

Content

1. The Airline Crew Assignment Problem
2. Quick Tour: Constraint Programming
3. Column Generation for the ACA
4. Heuristic Tree Search for the ACA
5. Integrating the Approaches: $1+1 > 2$
6. Conclusion

Airline Crew Assignment I

Market Modeling

Network Design

Fleet Assignment

Aircraft Rotation

Crew Assignment

Crew Rostering

Day of Operation

$\approx$ 2 years | $\approx$ 8 month | $\approx$ 3 month | $\approx$ 2 month | $\approx$ 2 weeks

Airline Crew Assignment II

Let $k, m, n \in \mathbb{N}$ and let $C := \{1, \ldots, m\}$ the *set of crew members* and $T := \{1, \ldots, n\}$ the *set of activities*.

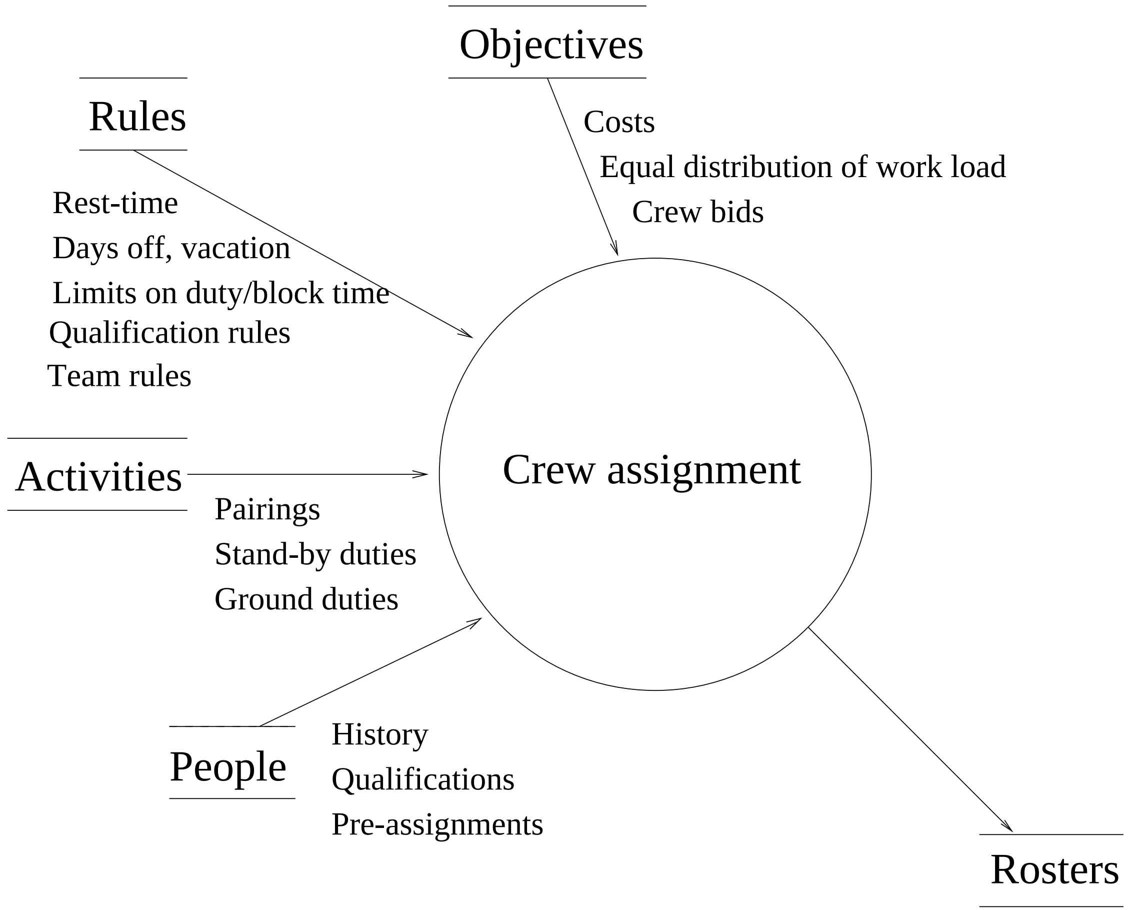
1. Assign to each crew member c_i a *legal* subset of activities $t_i \subseteq 2^T$, $R := \{(c_i, t_i), \ldots, (c_m, t_m)\}$ is the *set of rosters*.

2. $f : R \to \mathbb{Q}^+$ is called a *cost function*.

3. The *Crew Assignment Problem (ACA)* is to minimize $\sum_{1 \leq i \leq m} f((c_i, t_i))$ s.t.

$$\{c_1, \ldots, c_m\} = C \qquad \text{and} \qquad \bigcup_{1 \leq i \leq m} t_i = T$$

Airline Crew Assignment III



The Need for Flexibility and Efficiency

one software to be designed for three airline companies

- There is a need for a flexible modeling of rather complex European airline rules and regulations.
- There is a need for modeling e.g. nonlinear constraints, etc.
- There is a need for fast/efficient algorithms . . .

Our approach:

Combine the flexibility of Constraint Programming with the efficiency of Operations Research

Constraint Programming – A Quick Tour

Given a CSP:

- A finite set of variables $X = \{x_1, \ldots, x_n\}$
- With each variable $x \in \mathcal{X}$ a *finite domain* $D(x)$ is associated,
- *Constraints* $C \in \mathcal{C}$
  - couple one or more variables $C(x_{i_1}, \ldots, x_{i_k})$
  - define a relation $R(C) \subseteq D(x_{i_1}) \times \cdots \times D(x_{i_k})$.

We ask for a value assignment $v(x)\ \forall\, x \in X$ such that

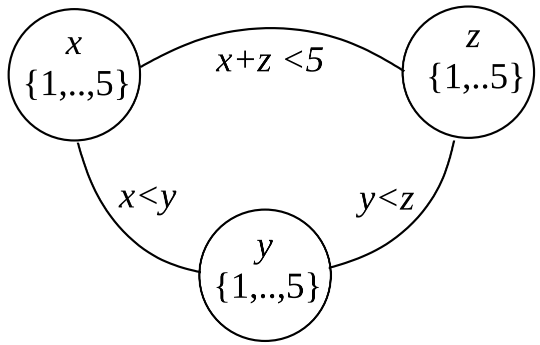
- all constraints are satisfied, i.e. $\forall\, C : (v(x_1), \ldots, v(x_n)) \in R(C)$
- the values respect the corresponding domains, i.e. $v(x) \in D(x)\ \forall\, x \in X$.

CSP is $\mathcal{NP}$-complete

Constraint Programming – A Simple Example

We need a solution for:

$$\begin{cases} x < y \\ y < z \\ x + z < 5 \\ x, y, z \in \{1, 2, 3, 4, 5\} \end{cases}$$

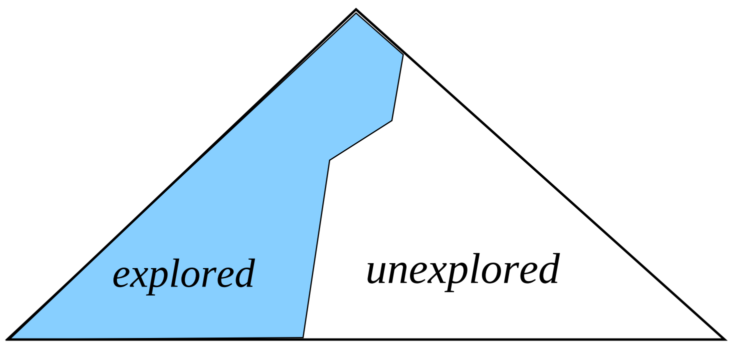


Domain reduction:

C		$\Rightarrow$		x	y	z
$x < y$	$\Rightarrow$	$x \neq 5,\ y \neq 1$	$\Rightarrow$	$\{1,2,3,4\}$,	$\{2,3,4,5\}$	
$y < z$	$\Rightarrow$	$y \neq 5,\ z \neq 1$	$\Rightarrow$		$\{2,3,4\}$,	$\{2,3,4,5\}$
$x + z < 5$	$\Rightarrow$	$x \notin \{3,4\}, z \notin \{4,5\}$,	$\Rightarrow$	$\{1,2\}$,		$\{2,3\}$
$y < z$	$\Rightarrow$	$y \notin \{3,4\}$, $z \neq 2$	$\Rightarrow$		$\{2\}$	$\{3\}$
$x < y$	$\Rightarrow$	$x \neq 2$	$\Rightarrow$	$\{1\}$		

Constraint Programming – Solution Techniques

- *Domain reduction* removes inconsistent values from the set of possible variable assignments.
- Branching on subproblems where domains are not unique after domain reduction (*tree search*)



Various notions of consistency, various tree search variants, . . .

Example: Constraint Programming for Airline Crew Assignment

Rules:

- Earliest Start time at A is 10:00
- IF arriving at B after 16:00 THEN earliestStart := 9:00 ELSE earliestStart := 6:00
- Arrival at C before 15:00
- $A \to B$ takes 12h, $A \to C$ takes 10h, $B \to C$ takes 8h

Decision: Fly $B \to C$

$\Rightarrow$ Arrival at C before 15:00 (Rule 3)
$\Rightarrow$ Departure at B before 7:00 (time)
$\Rightarrow$ Arrival at B before 16:00 (Rule 2)
$\Rightarrow$ ***A* cannot be successor of *B*** (as start at A + flight time $>$ 16:00)
$\Rightarrow$ remove edge (A, B) and propagate. . .

Approach 1: Heuristic tree Search

ACA can be regarded as a complex assignment problem:

- Try “all” possible assignments
- (with the help of some heuristics: We’ll find a good one early)

Approach: use Constraint Programming alone

Heuristic Tree Search

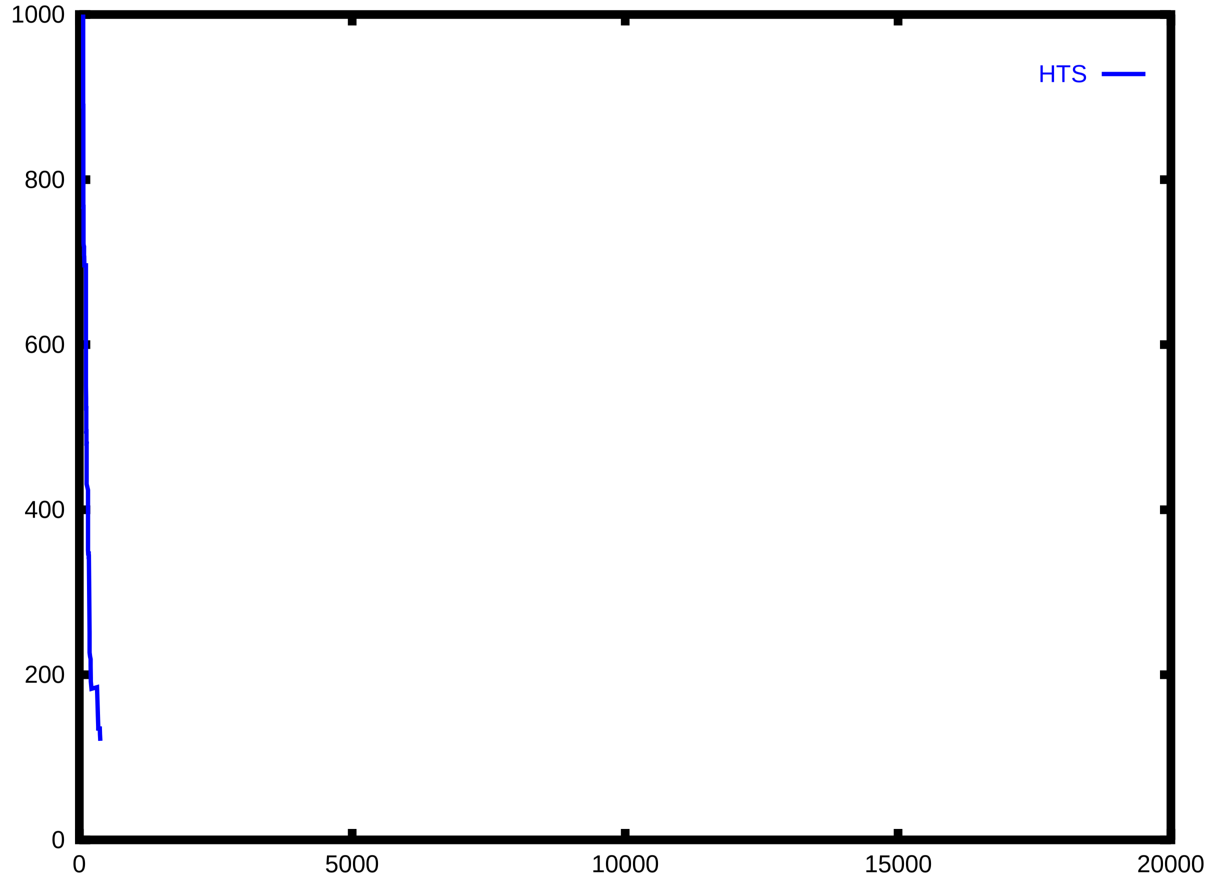
- ACA formulation
  - Pairings are represented by constrained variables
  - Crew members are represented by values in the variable domains

- Problem solving as search in the tree of possible values for variables

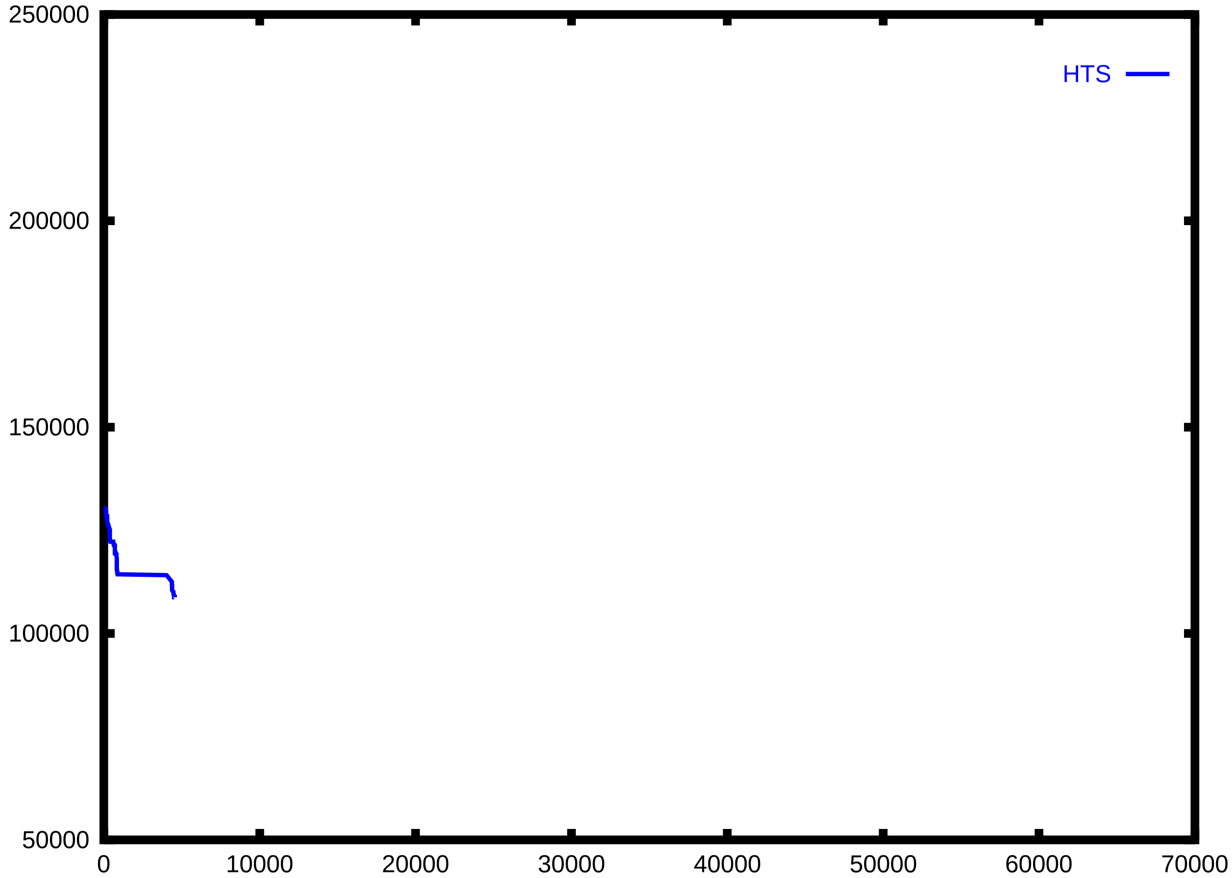
- Heuristics and search methods tailored for a specific airline case control search tree traversal
  - LDS, DDS
  - cost and feasibility oriented variable selection

- Incremental solution construction

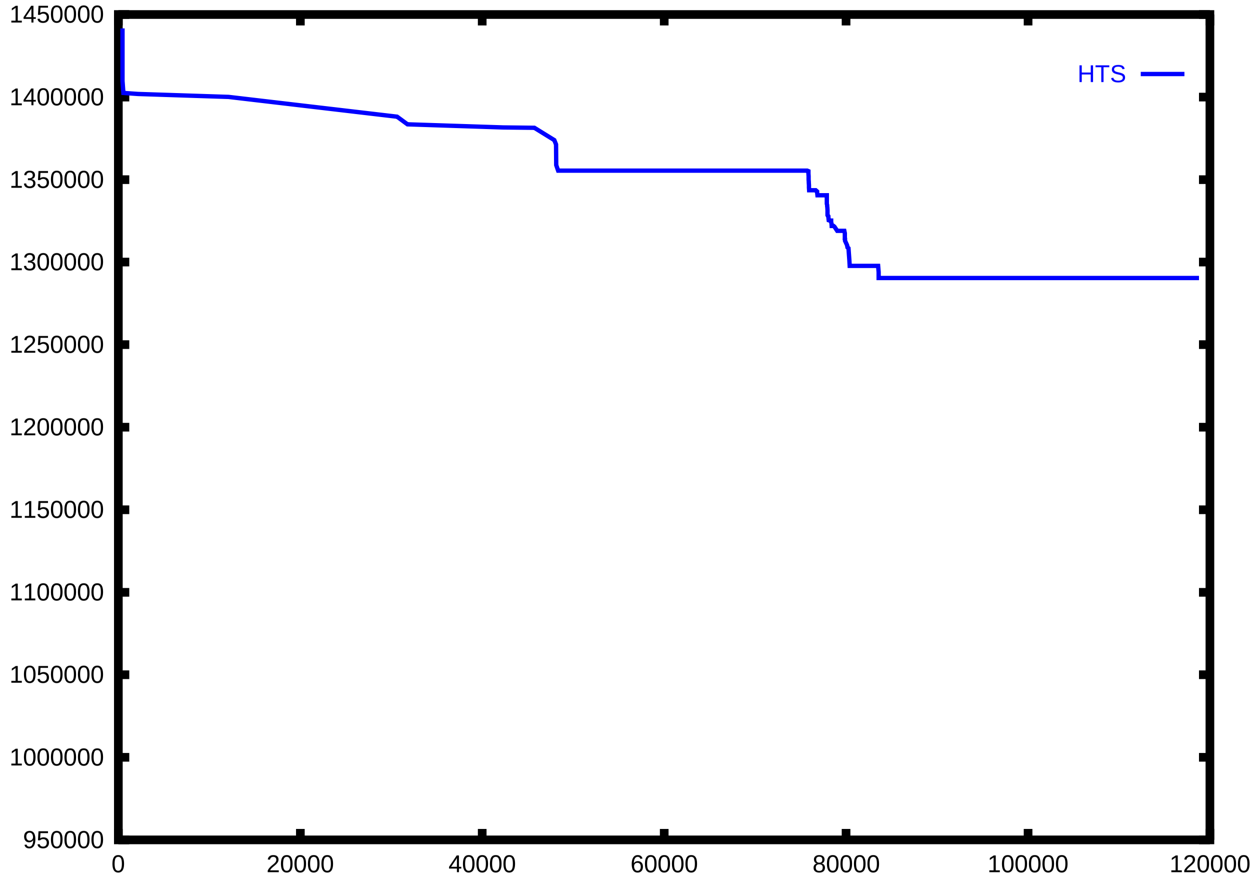
Results: 7 Crew Member, 129 Pairings



Results: 30 Crew Member, 279 Pairings



Results: 65 Crew Member, 959 Pairings



Observation: Heuristic Tree Search

- Feasible solutions can be found quickly
- Heuristic information and search methods guide the search towards good initial solutions
- Search gets stuck in local minima
- Convergence "stops" after initial bunch of solutions

Approach 2: Column Generation

ACA decomposes naturally into:

- Masterproblem: combine a selection of rosters to generate feasible schedule → Set Partitioning
- Subproblem: generate feasible *rosters* for each person → constrained shortest path

Approach:

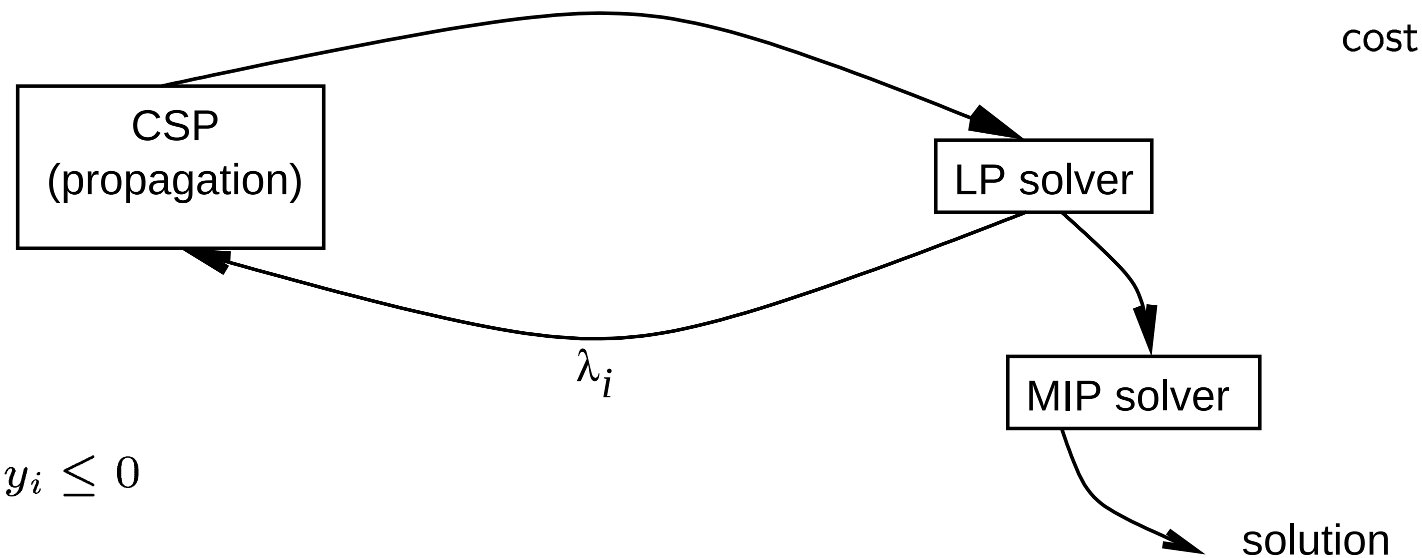
- use linear programming for combining rosters
- use Constraint Programming for the complex European rule sets

Constraint Programming based Column Generation

Integrate OR and CP

solution y_i, z
(= path)

Variable x_j,
coefficient $a_{i,j}$,
cost c_j



$$z - \sum_{i \in \mathcal{N}} \lambda_i y_i \leq 0$$

(NRC constraint)

⇒ Constraint Programming based Column Generation [Junker et al 1999]

The IP Masterproblem – Basic Model

	$x_1 \dots x_{n_1}$	$x_{n_1+1} \dots x_{n_2}$	$\dots$	$x_{n_{p-1}+1} \dots x_n$	
person 1	1 . . . 1				$= 1$
person 2		1 . . . 1			$= 1$
$\vdots$			$\ddots$		$\vdots$
person p				1 . . . 1	$= 1$
activity 1	1	1 1		1	$= b_1$
$\vdots$					$\vdots$
activity k	1		1	1 1	$= b_k$
	additional multi crew member constraints				

find : solution $x \in \{0,1\}^n$ with minimal costs $\sum_j c_j\, x_j$, where c_j is the cost value of roster j.

Constraints

assignment constraints

$$\underbrace{x_{n_a} + \cdots + x_{n_{a'}}}_{\text{person } a} = 1, \quad \underbrace{x_{n_b} + \cdots + x_{n_{b'}}}_{\text{person } b} = 1, \quad \ldots$$

SCP/SPP constraints

$$x_{j_1} + \cdots + x_{j_{n_\ell}} = b_\ell$$

(cockpit crew usually $b_\ell = 1$, cabin crew often $b_\ell \geq 1$)

additional constraints

- person a is *not allowed to* fly together with persons b, c:

$$\underbrace{3x_{n_a} + \cdots + 3x_{n_{a'}}}_{\text{person } a} + \underbrace{x_{n_b} + \cdots + x_{n_{b'}}}_{\text{person } b} + \underbrace{x_{n_c} + \cdots + x_{n_{c'}}}_{\text{person } c} \leq 3$$

- person a *has to* fly together with persons b:

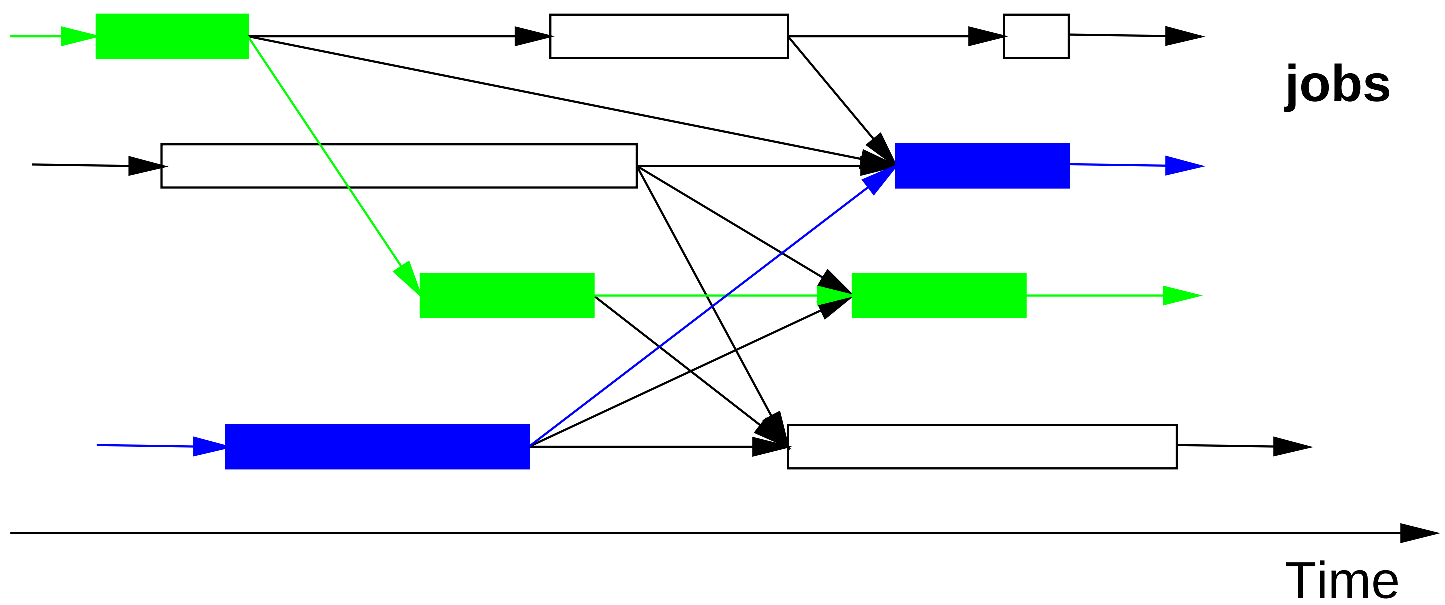
$$\underbrace{x_{n_a} + \cdots + x_{n_{a'}}}_{\text{person } a} \underbrace{- x_{n_b} \cdots - x_{n_{b'}}}_{\text{person } b} = 0$$

Constraint Shortest Path

Crew Rostering

- Assign rosters to named individuals
- respect many rules and regulations

} shortest path (= lower bound for feasible roster)
(edge costs ≈ dual values)

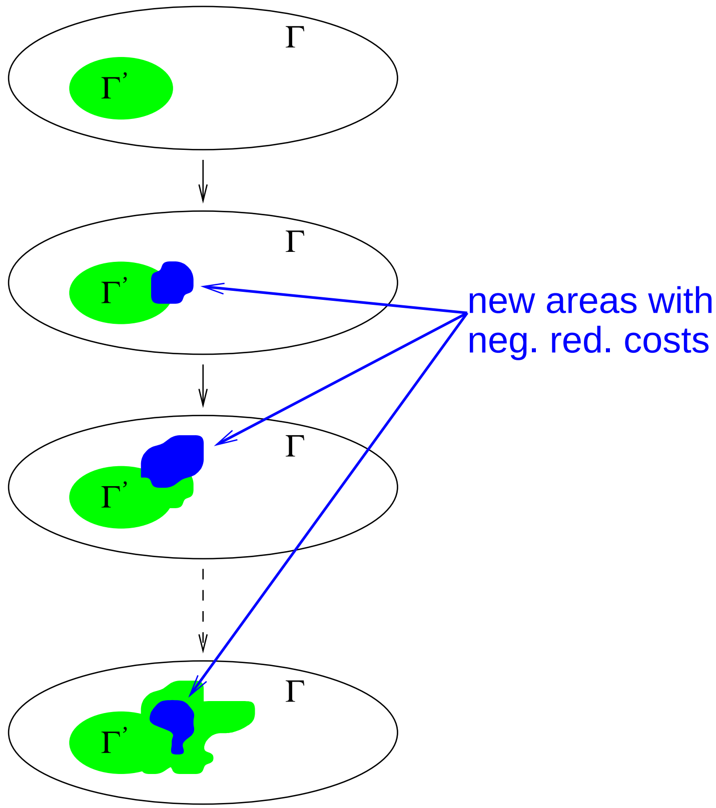


Outline of Column Generation

Subproblem:

- Generate further columns with negative reduced costs
- Use dual values of LP solution to determine reduced costs

Γ' :=`getInitialColumns()`
repeat
  λ := `solveLP(`Γ'`)` {get new duals}
  $\{x_{j_1}, \ldots, x_{j_k}\}$:= `solveSubproblem(`λ`)`
  $\Gamma' = \Gamma' \cup \{x_{j_1}, \ldots, x_{j_k}\}$
until $(\{x_{j_1}, \ldots, x_{j_k}\} = \emptyset)$



The Subproblem: Generate a single 'good' path

Modify Network: Add source s, sink t, edges (s,i), (i,t) to DAG $\rightarrow \mathcal{N}$

Subproblem: Find an (s,t)-path with negative reduced costs

$y_i \in \{0,1\}$	$y_i = 1 \iff$ node i is on that path
z	cost of that path

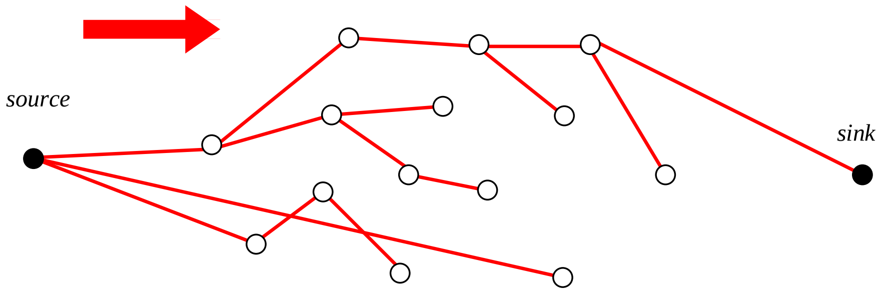
For the simple problem: $y_s = 1$, $y_t = 1$, $z = 1$

negative reduced costs constraint (NRC)

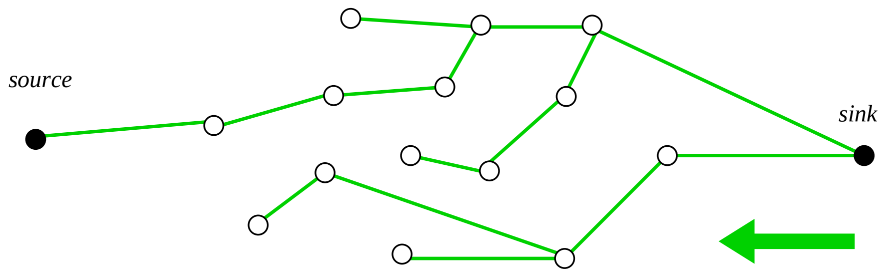
$$\boxed{z - \sum_{i \in V} \lambda_i y_i \leq 0}$$

(where λ_i is the dual value of the constraint corresponding to node i in the master problem)

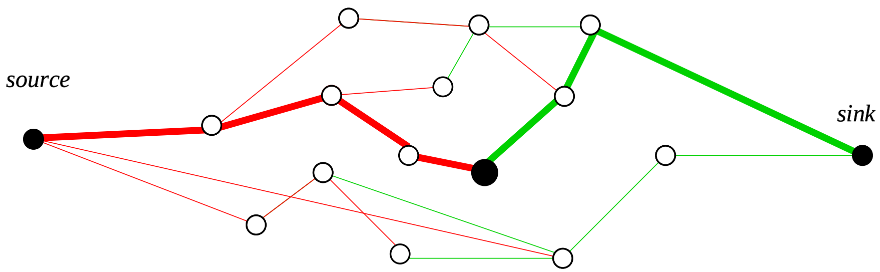
Example



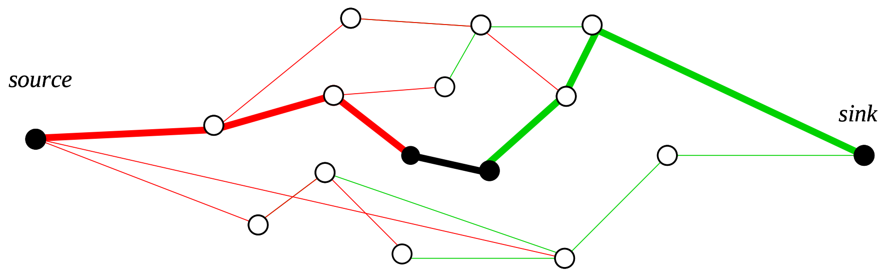
shortest paths from source to all nodes



shortest paths from sink to all nodes



$z_{s,i} + z_{i,t} > \max(z)$
$\Rightarrow$ pos(Y) := pos$(Y) \setminus \{i\}$



$z_{s,i} + c_{i,j} + z_{j,t} > \max(z)$
$\Rightarrow$ suppress edge (i, j)

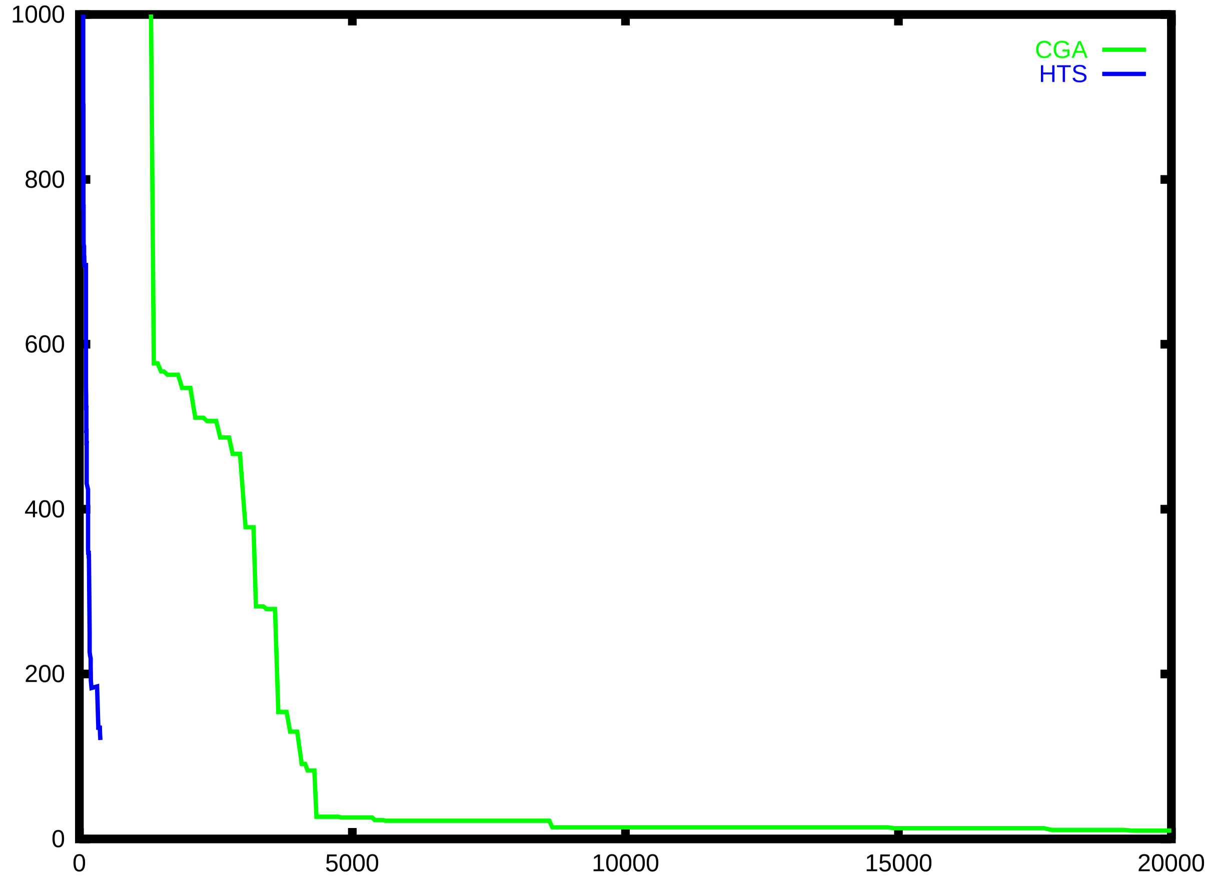
Interference with other Airline Rules

- Each time an edge/node is required it is added to the required set of the path
- This may trigger propagation of other constraints (work time limits, exclude overlapping jobs, . . .).
- This may trigger the shortest path constraint again.

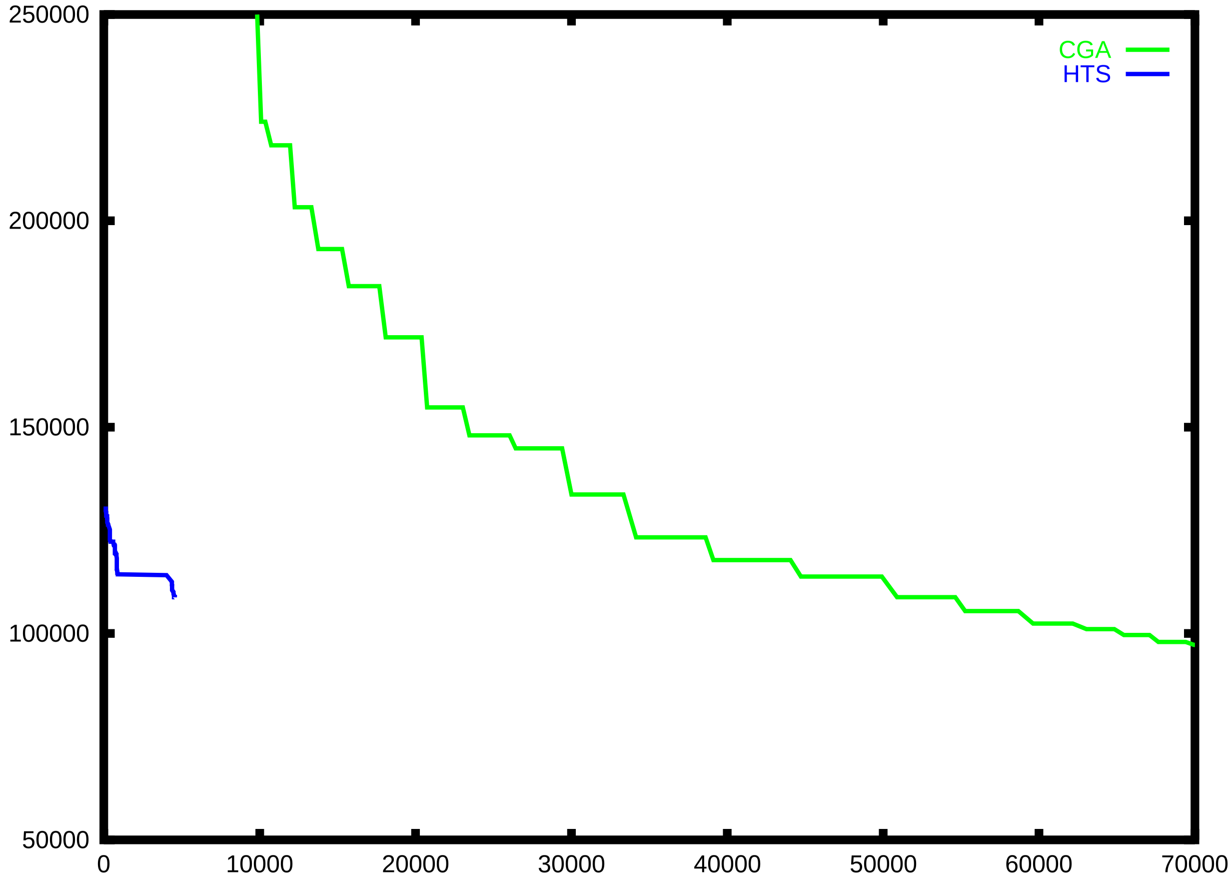
$\Rightarrow$ all constraints communicate via *domains*.

$\Rightarrow$ an incremental shortest path algorithm may save a lot of computational effort.

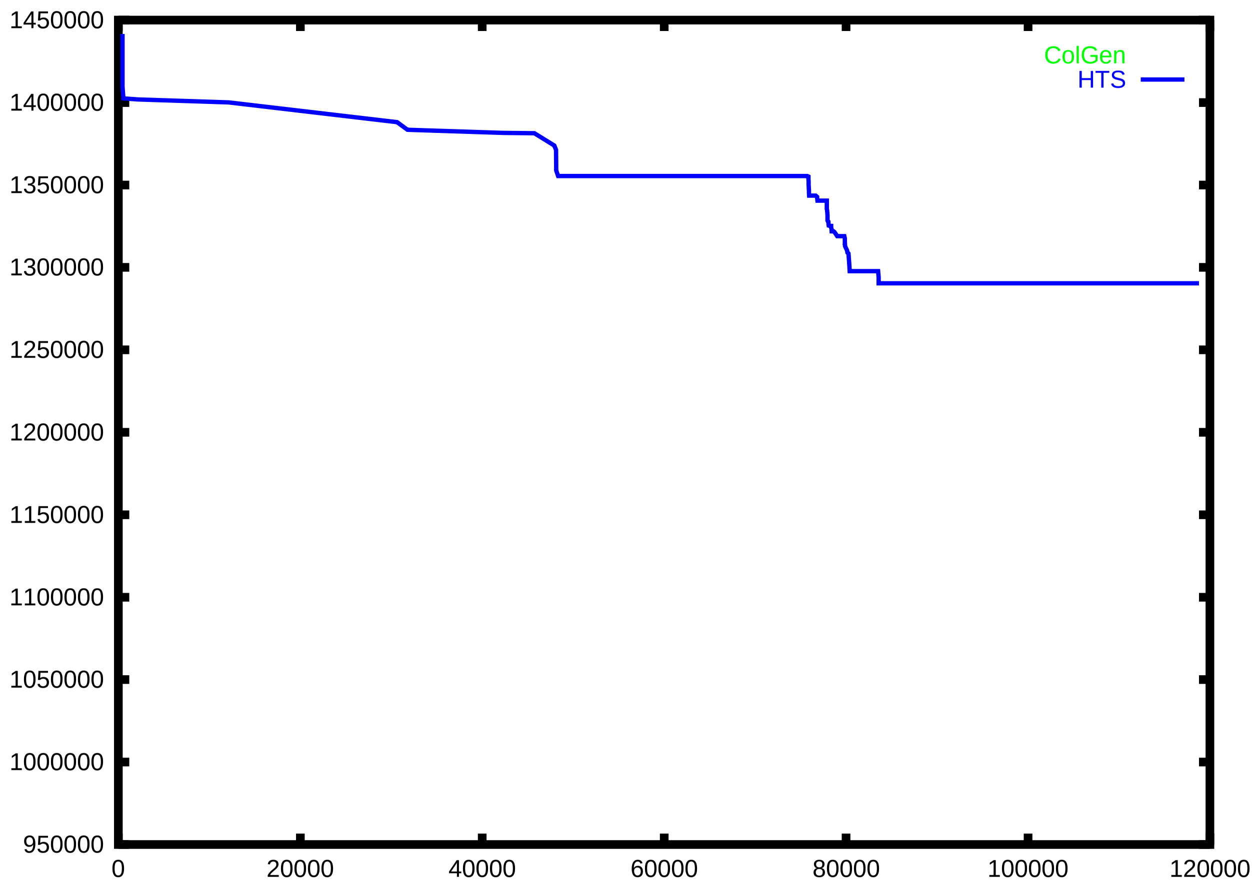
Results: 7 Crew Member, 129 Pairings



Results: 30 Crew Member, 279 Pairings



Results: 65 Crew Member, 959 Pairings



No solution found within 120 000 sec. using GCA

Observation: Column Generation

- First Solution needs long time
- Initial solution is of bad quality
- Converges nicely towards optimal solution
- Problems when combining roster (next slide. . .)

Conflicts when Combining Rosters

1	1	1	1	1															1
					1	1	1	1	1	1	1								1
												1	1	1	1	1	1	1	1
1		1	1		1		1	1			1		1		1	1			2
	1		1			1		1	1		1	1		1		1	1		1
		1		1			1			1					1				1
1				1	1						1		1						1
	1	1				1	1		1	1				1	1		1		1
	1			1	1				1	1		1	1				1		1
1			1			1		1						1		1			1

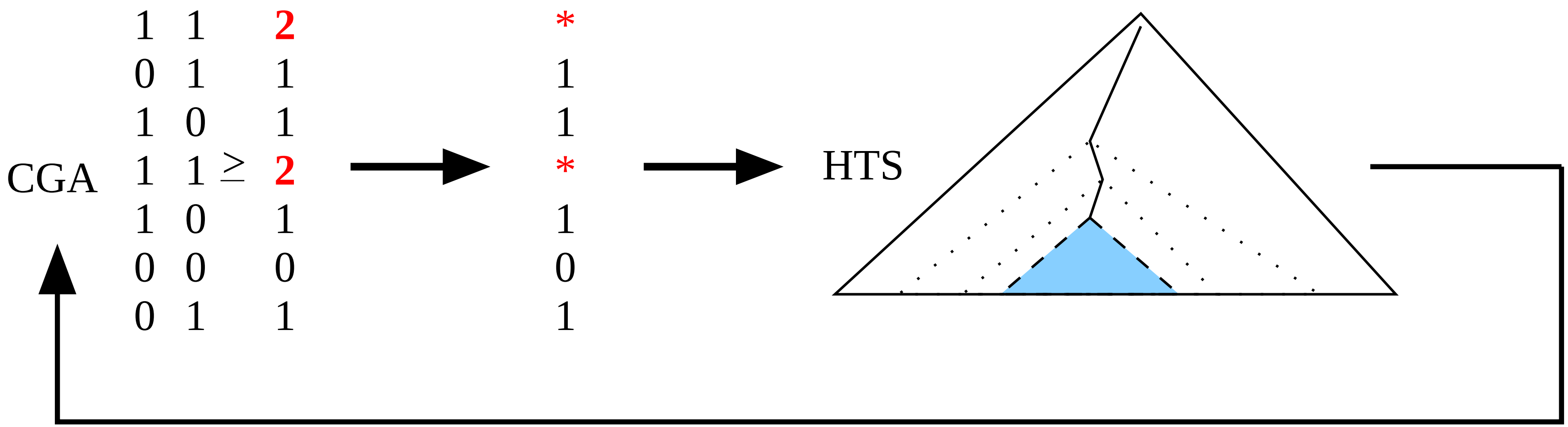
Integrated Approach: 1+1 $>$ 2

Column Generation (CGA)	Heuristic Tree Search (HTS)
First Solution needs long time	Feasible solutions can be found quickly
Bad initial solution	Heuristic information and search methods guide the search towards good initial solutions
converges nicely towards optimal solution	Search gets stuck in local minima Convergence “stops” after initial bunch of solutions
problems when combining roster	(there is no “combining” in HTS)

First Integration: Set Covering + Repair

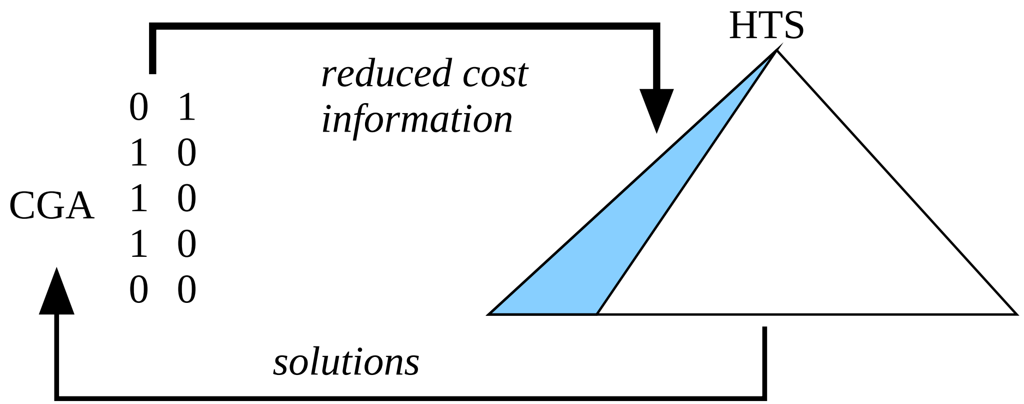
- Increase startup-time of column generation by feeding it with solutions of heuristic tree search
- Increase convergence by relaxing *set partitioning* structure of IP to *set covering*
- Now over covered pairings/crew member are infeasible!
- Deassign over covered pairings and perform a HTS on the partial solution.

First Integration: Set Covering + Repair



Second Integration: Dual Values guiding HTS

- Increase startup-time of column generation by feeding it with solutions of heuristic tree search
- Increase convergence of CGA by adding solutions that are found by HTS based on *reduced cost* optimization



Second Integration: Dual Values guiding HTS

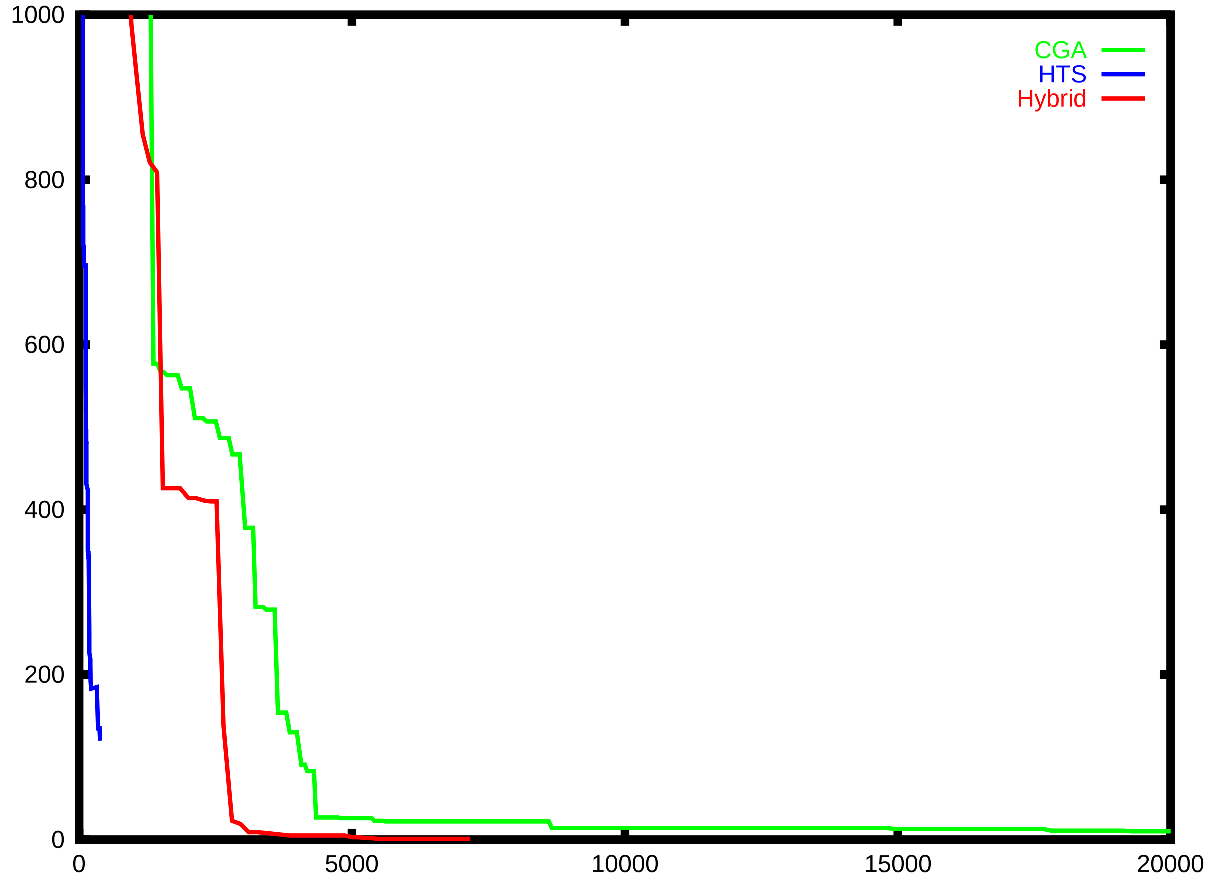
- In the CGA approach first solutions are poor because dual values of initial dummy rosters lead to poor regions of the solution space

- HTS heuristics offer good initial solutions

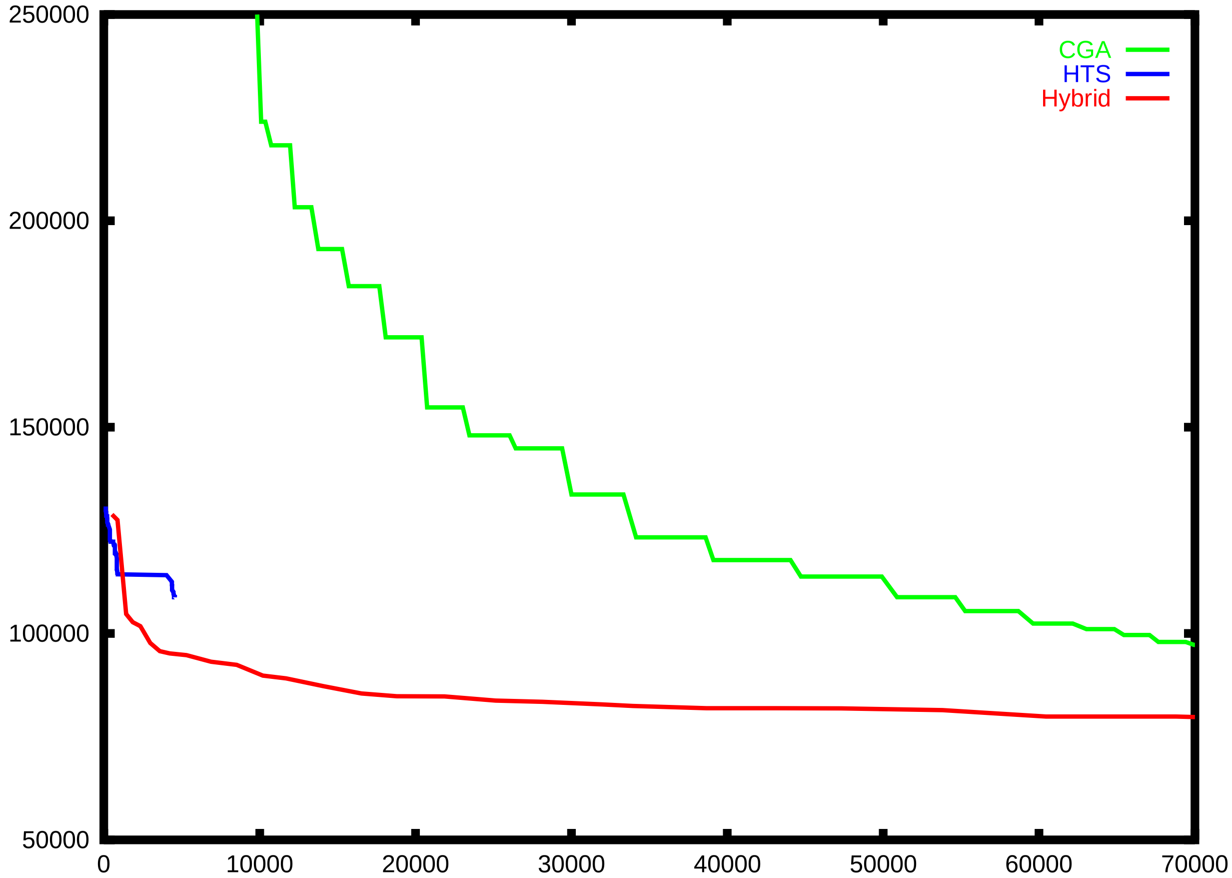
- by feeding HTS with reduced cost information HTS can quickly search for *combine-able solutions* that are good with resp. to dual values

- hence convergence is improved

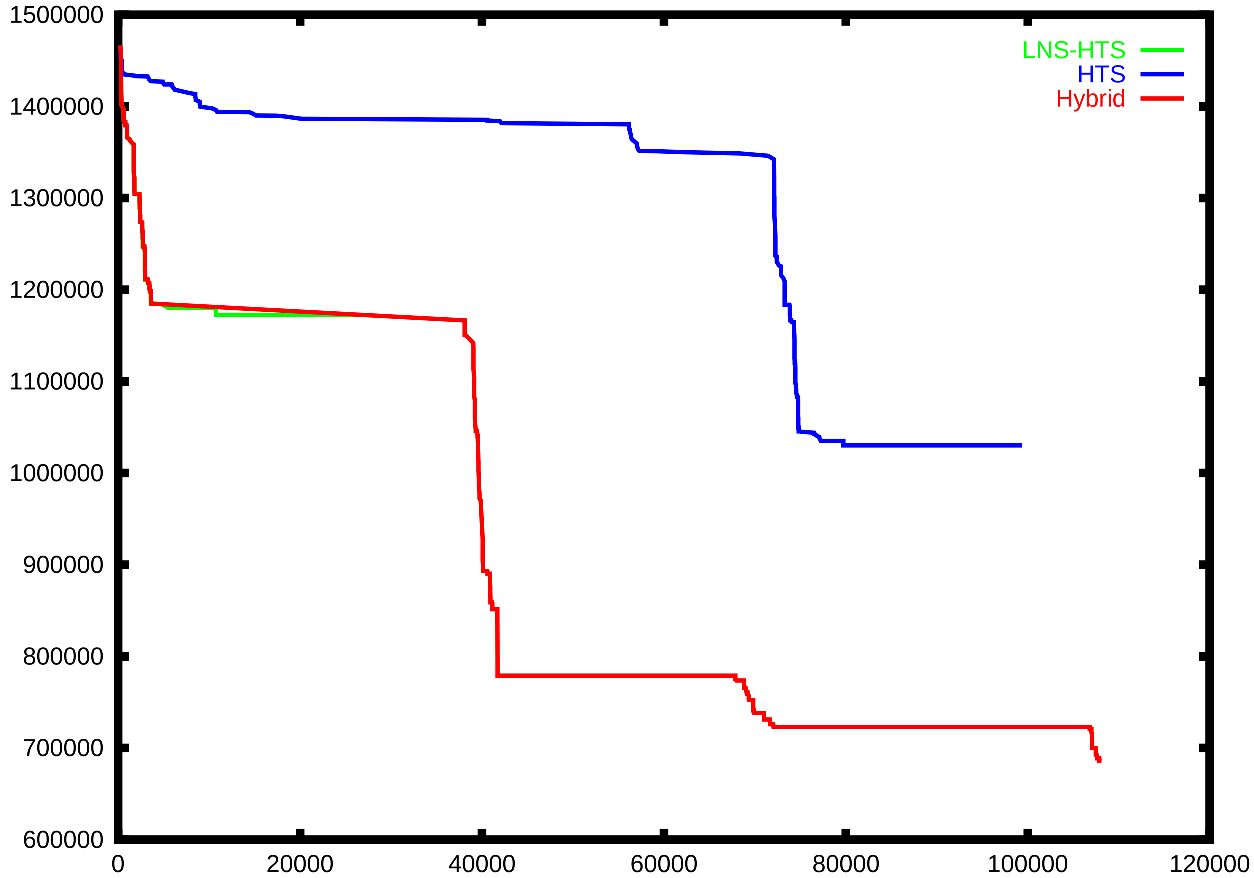
Results: 7 Crew Member, 129 Pairings



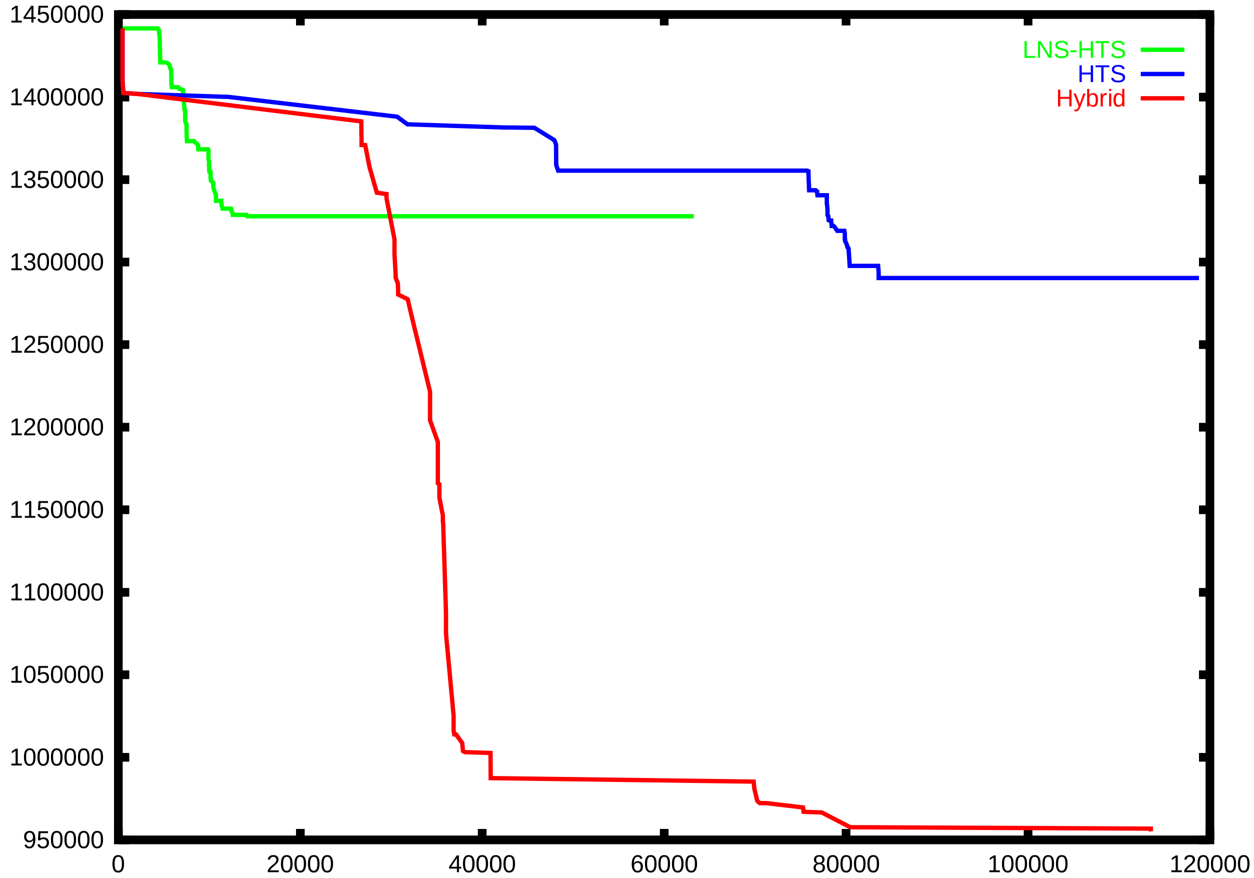
Results: 30 Crew Member, 279 Pairings



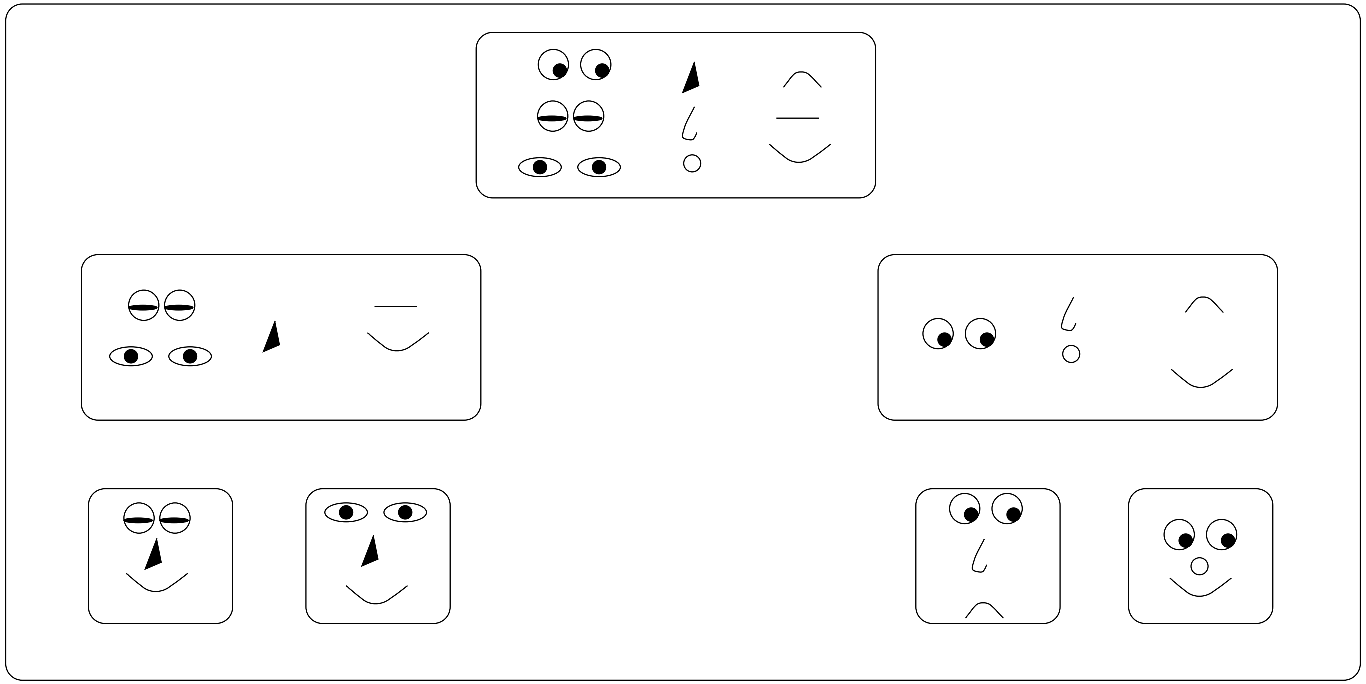
Results: 50 Crew Member, 766 Pairings



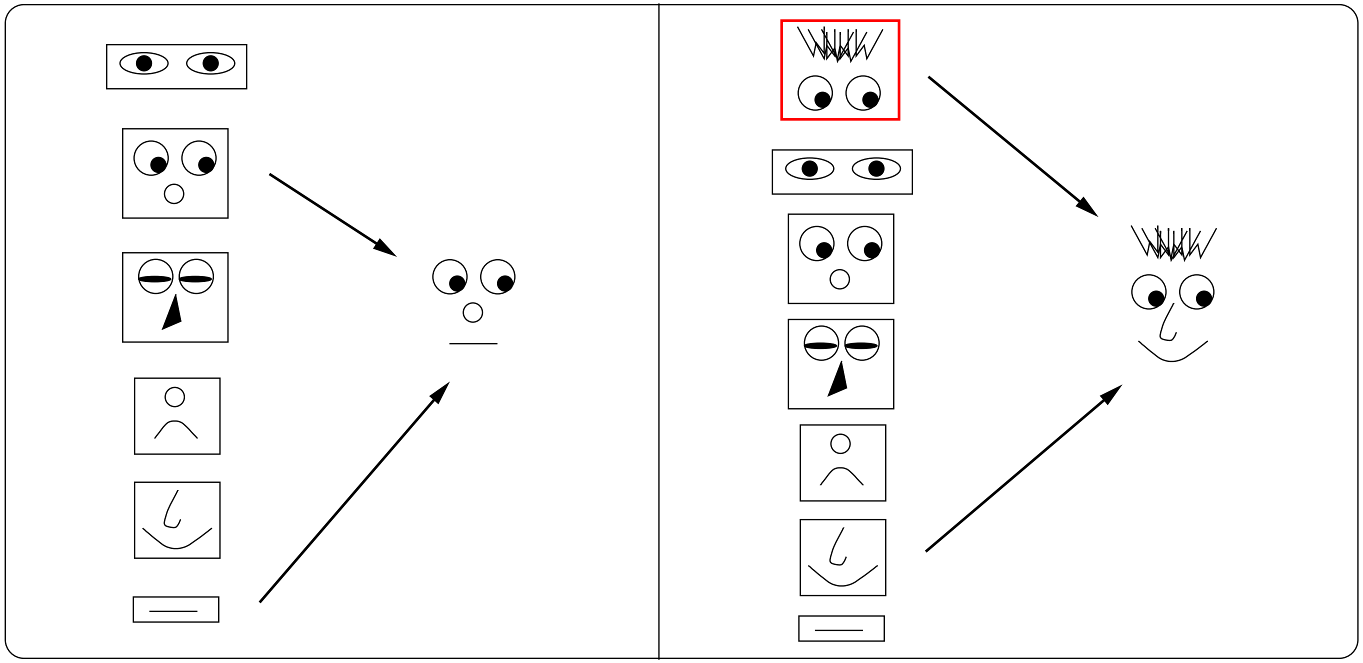
Results: 65 Crew Member, 959 Pairings



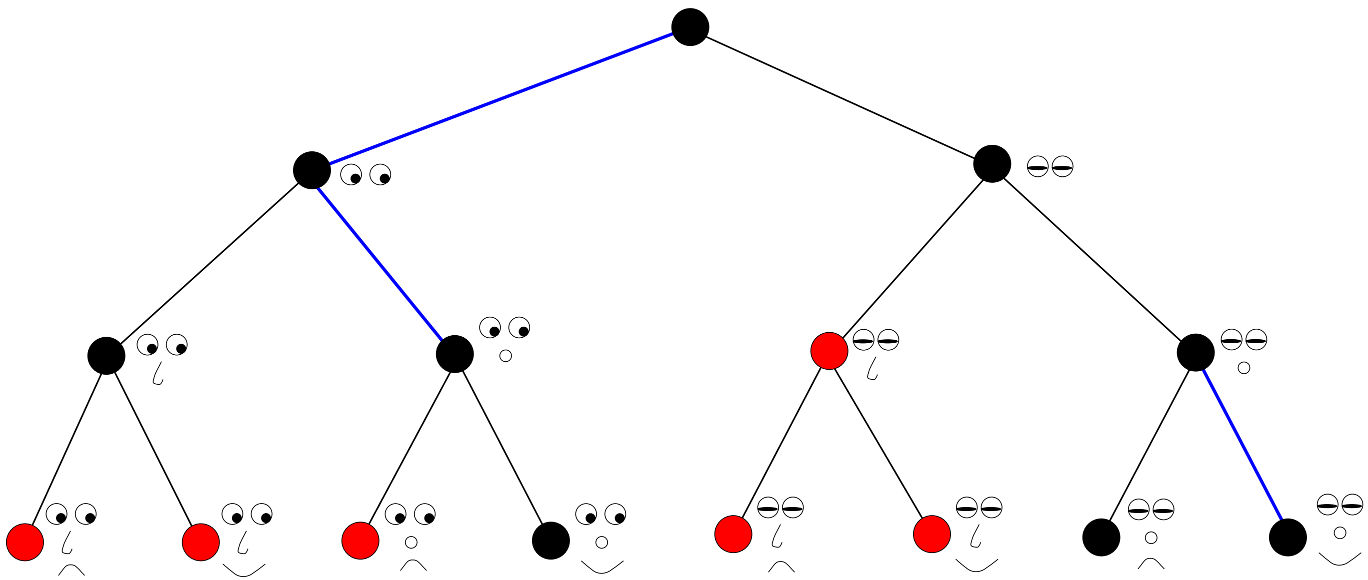
Constraint Programming



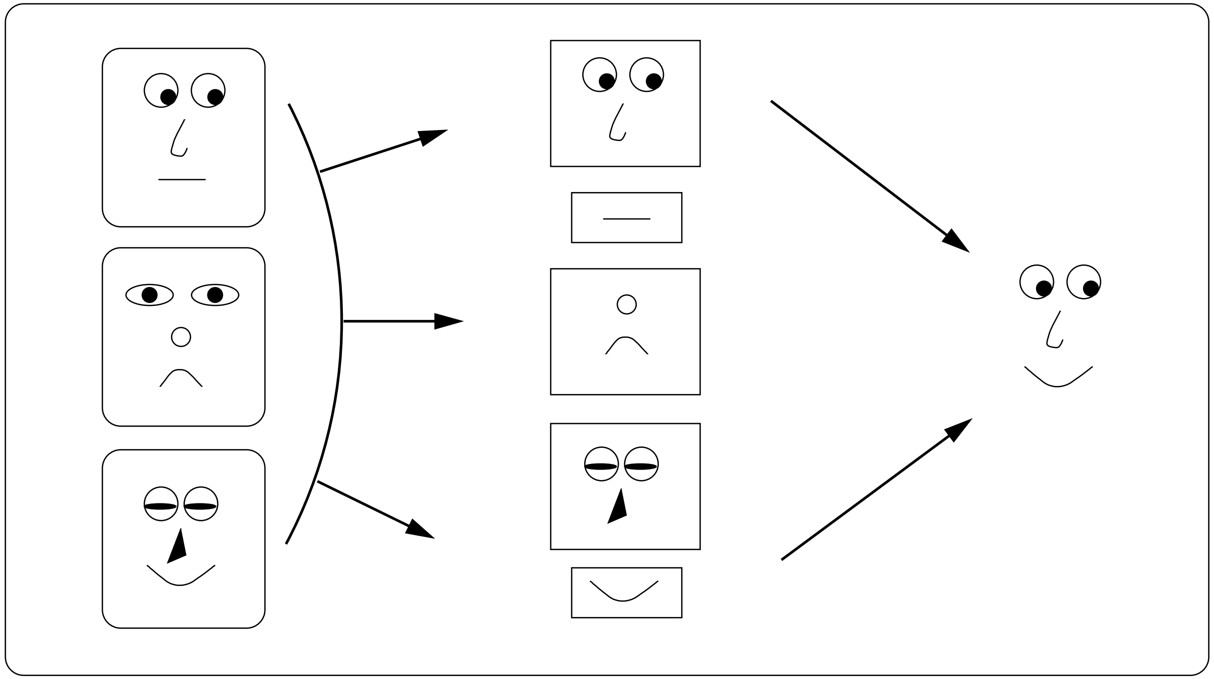
Column Generation



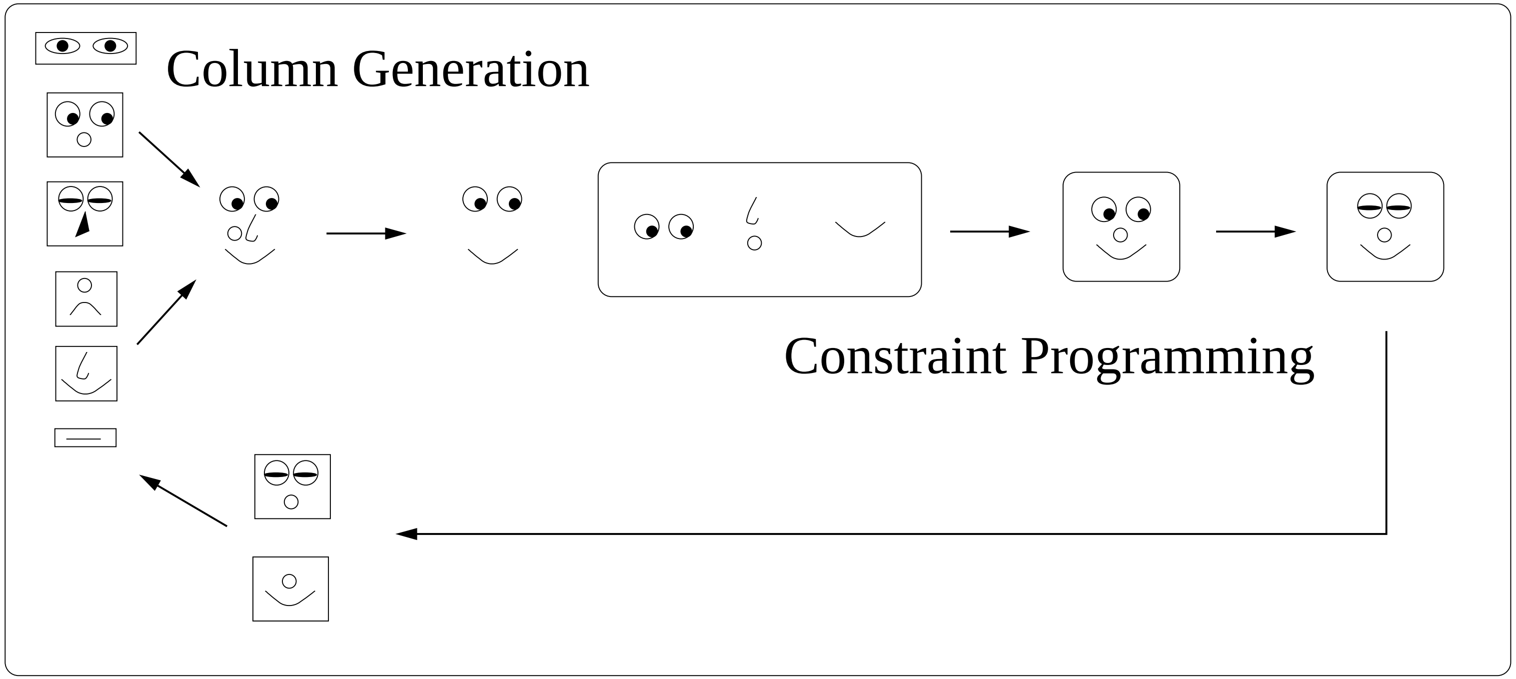
The Decision Tree



Initial Columns



An Integrated Approach



Conclusions

Isolated Approach

Constraint Programming

+ generic potential, resolves local conflicts
− non-local constraints, gets stuck in local optima

Column Generation

+ global view, bounds
− initial columns, suffers from local conflicts

Integration

Constraint Programming

+ provides good first solutions quickly
+ resolves overcoverings in the master problem

Column Generation

+ guides to promising regions of the search space
+ provides bounds for pruning

Combination of Operations Research and Constraint Programming offers *flexible* and still *efficient* modeling and solving of practical Airline Crew Assignment Problems.