Alternative Pricing in Column Generation for Airline Crew Rostering

Emily Curry

May 29, 2018

Department of Mathematical Sciences
Chalmers University of Technology

Aim

Investigate and implement alternative pricing methods in the column generation framework for the airline crew rostering problem at Jeppesen

Outline

- Introduction to the airline crew rostering problem
- Mathematical formulation and the column generation framework
- The pricing problem at Jeppesen
- The alternative pricing methods
- Results
- Conclusions and future work

Airline Crew Rostering

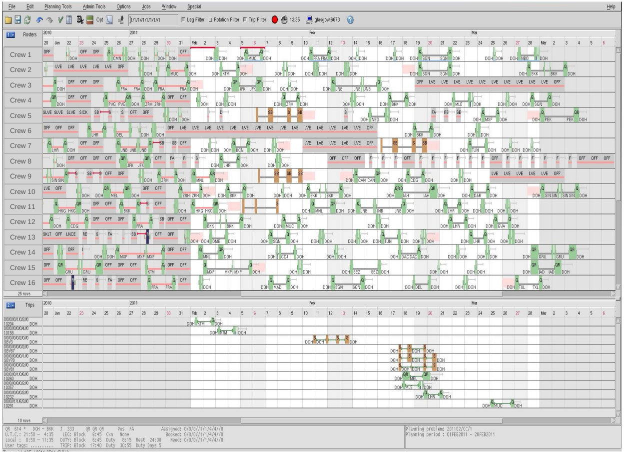
Airline Crew Rostering

Create monthly personalized schedules (rosters) for crew members, e.g. pilots and flight attendants, such that all flights are staffed

Objectives:

- reduce crew costs
- create fair schedules
- create robust solution

Airline crew rostering - problem description



Legal and complete rosters

Airline crew rostering

Difficulties:

- Rules and regulations
- Large scale

Mathematical Formulation and Column Generation

Mathematical formulation

Given:

$\mathcal{T}$ - set of tasks

$\mathcal{C}$ - set of crew members

Each roster can be modeled as a binary column vector $\boldsymbol{a}_j$, where

$$\boldsymbol{a}_j = \begin{bmatrix} \boldsymbol{e}_k \\ \boldsymbol{p}_j \end{bmatrix}, \ j \in \mathcal{J}_k, \ k \in \mathcal{C} \tag{1}$$

where $\boldsymbol{e}_k$ unit vector and $\boldsymbol{p}_j \in \{0, 1\}^{|\mathcal{T}|}$

Mathematical formulation

$$A = \begin{array}{c} \begin{matrix} \overbrace{\hphantom{1\ 1\ 1}}^{\text{Crew 1}} & \overbrace{\hphantom{1\ 1\ 1}}^{\text{Crew 2}} & \cdots & \overbrace{\hphantom{1\ 1\ 1}}^{\text{Crew } |\mathcal{C}|} \end{matrix} \\ \begin{bmatrix} 1 & 1 & 1 & & & & \cdots & & & \\ & & & 1 & 1 & 1 & \cdots & & & \\ & & & & & & \ddots & & & \\ & & & & & & \cdots & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & \cdots & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & \cdots & 1 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 \end{bmatrix} \end{array} \begin{array}{l} \left.\vphantom{\begin{matrix}1\\1\\1\\1\end{matrix}}\right\} |\mathcal{C}| \text{ assignment constraints} \\ \left.\vphantom{\begin{matrix}1\\1\\1\\1\end{matrix}}\right\} |\mathcal{T}| \text{ activity constraints} \end{array} .$$

Mathematical formulation

$$\begin{aligned} \min \ & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t. } & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \in \{0,1\}^n, \end{aligned} \tag{2}$$

where $x_j = 1$ if $\boldsymbol{a}_j$ should be assigned, else $x_j = 0$

Difficulties:

Large number of variables/columns

$\implies$ Solve using column generation

Column generation

Master problem (MP)

$$\begin{aligned} \min \ & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t. } & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \mathbf{0} \end{aligned} \tag{3}$$

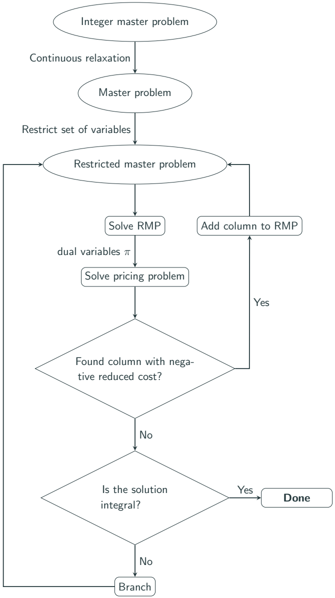
Column generation

Restricted master problem (RMP)

$$\begin{aligned} \min \quad & \sum_{j \in \mathcal{J}'} c_j x_j \\ \text{s.t.} \quad & \sum_{j \in \mathcal{J}'} \boldsymbol{a}_j x_j = \boldsymbol{b} \\ & x_j \geq 0, \ \forall j \in \mathcal{J}', \end{aligned} \tag{4}$$

where $\mathcal{J}' \subseteq \mathcal{J}$

Column generation



The Pricing Problem

The pricing problem

Aim:

Generate *legal* columns with negative reduced cost for each crew member $k \in \mathcal{C}$, i.e. solve the reduced cost-function

$$\min_{\boldsymbol{a}_j} \; c(\boldsymbol{a}_j) - \boldsymbol{\pi}^\top \boldsymbol{a}_j, \; \forall j \in \mathcal{J}_k \tag{5}$$

Challenges:

- Complex rules and regulations
- Nonadditivity
- Large scale

Rules are separated from the core algorithm using the proprietary business rule engine Rave

$\implies$ solution methods independent of the rules set

The pricing problem

Current methods:

- Shortest path with resource constraints
- Local search

Alternative methods:

- Binary particle swarm optimization (BPSO)
- Surrogate modeling with linearization and shortest path
- Surrogate modeling without linearization

The Alternative Pricing Methods

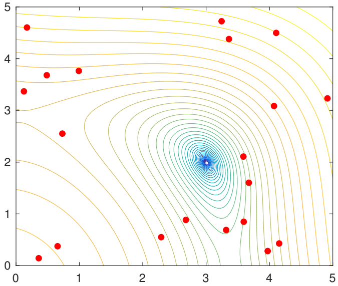
Particle swarm optimization (PSO)

Stochastic method inspired by swarming behavior found in nature for solving continuous problems found in complex engineering systems

Particle swarm optimization (PSO)

Idea:

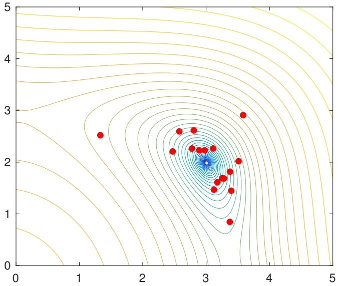
"Particles" associated with a **position** and **velocity** move in the search space influenced by the best known local position as well as the best global position



Particle swarm optimization (PSO)

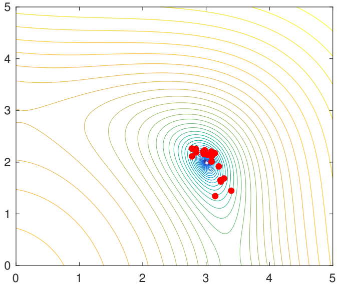
Idea:

"Particles" associated with a **position** and **velocity** move in the search space influenced by the best known local position as well as the best global position



Particle swarm optimization (PSO)

Idea:
"Particles" associated with a **position** and **velocity** move in the search space influenced by the best known local position as well as the best global position



Binary particle swarm optimization (BPSO)

Velocities passed through a *transfer function* used as a probability of the position in the next iteration

$\Longrightarrow$ tasks belonging to "good" columns will be more likely assigned

Binary particle swarm optimization (BPSO)

1. Initialize positions, using **entire** set of tasks, and velocities for all particles
2. Calculate reduced cost for each particle using the position
3. Update the best position found by each particle as well as the best position found by the entire swarm
4. Update the velocity of each particle
5. Calculate the value of the transfer function for each particle
6. Update the position for each particle
7. Go to Step 2 until a stopping criterion has been reached

Surrogate modeling

Idea:

Create a surrogate function $s(\boldsymbol{p})$ using a set of data points S from the original function that mimics the behavior of the underlying model

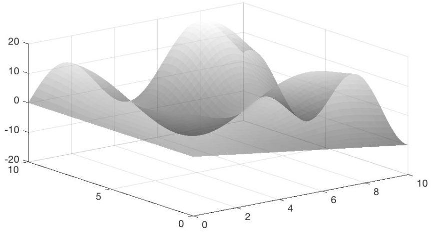
Surrogate modeling with radial basis functions

Surrogate modeling with *radial basis function* $\phi(r) = r^2 \log(r)$,

$$s(\boldsymbol{p}) = \sum_{l=1}^{n} \lambda_l \phi(||\boldsymbol{p} - \boldsymbol{p}_l||_2) + \boldsymbol{b}^\top \boldsymbol{p} + a \tag{6}$$

$\implies$ interpolation equations

$$s(\boldsymbol{p}_l) = f(\boldsymbol{p}_l), \quad l = 1, 2, \ldots, |S| \tag{7}$$



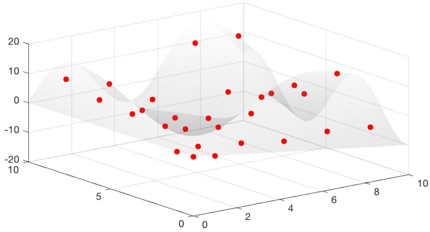
Surrogate modeling with radial basis functions

Surrogate modeling with *radial basis function* $\phi(r) = r^2 \log(r)$,

$$s(\boldsymbol{p}) = \sum_{l=1}^{n} \lambda_l \phi(||\boldsymbol{p} - \boldsymbol{p}_l||_2) + \boldsymbol{b}^\top \boldsymbol{p} + a \quad (6)$$

$\implies$ interpolation equations

$$s(\boldsymbol{p}_l) = f(\boldsymbol{p}_l), \quad l = 1, 2, \ldots, |S| \quad (7)$$



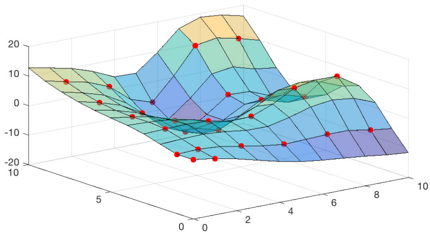
Surrogate modeling with radial basis functions

Surrogate modeling with *radial basis function* $\phi(r) = r^2 \log(r)$,

$$s(\boldsymbol{p}) = \sum_{l=1}^{n} \lambda_l \phi(||\boldsymbol{p} - \boldsymbol{p}_l||_2) + \boldsymbol{b}^\top \boldsymbol{p} + a \tag{6}$$

$\implies$ interpolation equations

$$s(\boldsymbol{p}_l) = f(\boldsymbol{p}_l), \quad l = 1, 2, \ldots, |S| \tag{7}$$



Surrogate modeling with radial basis functions

1. Create initial set of samples
2. Fit surrogate model using the set of samples from **reduced** set of tasks
3. Use surrogate model to search for candidate points
4. Go to 2 until stopping criterion has been reached

Surrogate modeling: Find candidate point using linearization

Linear approximation

$$\bar{s}(\boldsymbol{p}) = \sum_{l=1}^{n} \lambda_l ||\boldsymbol{p} - \boldsymbol{p}_l||_2^2 + \boldsymbol{\beta}^\top \boldsymbol{p} + \alpha. \tag{8}$$

$\implies$ Form linear edge costs in network from linearization

$\implies$ Find shortest path as candidate point

Surrogate modeling: Find candidate point without linearization

Find solution to nonlinear surrogate function using BPSO as a comparison to the linearization

BPSO:
Evaluate the particles' positions using the surrogate function

Results

Results - test cases

Test case	Number of crew	Number of tasks	Median pricing size
1	600	4 000	2 400
2	1 000	3 000	1 700
3	2 300	3 500	400
4	1 700	3 500	1 700
5	600	3 000	1 400

Results

Performance measures related to

- Negative reduced cost
  - Hit rate
  - Mean of best negative reduced cost
  - Minimum of negative reduced cost
- Improvement in objective for RMP

Results - negative reduced cost

	BPSO		Linearized surrogate		Nonlinear surrogate	
	Early phase	Later phase	Early phase	Later phase	Early phase	Later phase
Hit rate (%)						
1	59.25	29.20	4.41	1.50	88.57	79.27
2	57.22	36.50	1.85	0.38	52.75	61.82
3	80.96	60.26	25.04	13.50	52.70	40.13
4	48.88	26.12	10.62	5.37	47.78	41.43
5	98.70	76.44	60.78	16.52	94.81	47.30

Results - negative reduced cost

	BPSO		Linearized surrogate		Nonlinear surrogate	
	Early phase	Later phase	Early phase	Later phase	Early phase	Later phase
	Hit rate (%)					
1	59.25	29.20	4.41	1.50	88.57	79.27
2	57.22	36.50	1.85	0.38	52.75	61.82
3	80.96	60.26	25.04	13.50	52.70	40.13
4	48.88	26.12	10.62	5.37	47.78	41.43
5	98.70	76.44	60.78	16.52	94.81	47.30

Results - negative reduced cost

	BPSO		Linearized surrogate		Nonlinear surrogate	
	Early phase	Later phase	Early phase	Later phase	Early phase	Later phase
Normalized mean of successfully solved pricing problems						
1	−1.00	−0.11	−0.053	−0.0076	−0.082	−0.028
2	−1.00	−0.98	−0.36	−0.012	−0.65	−0.33
3	−1.00	−0.67	−0.42	−0.36	−0.59	−0.34
4	−1.00	−0.47	−0.16	−0.083	−0.26	−0.15
5	−1.00	−0.70	−0.27	−0.054	−0.40	−0.092
Normalized min of reduced cost						
1	−1.00	−0.066	−0.058	−0.0062	−0.059	−0.032
2	−0.83	−1.00	−0.62	−0.054	−0.62	−0.42
3	−1.00	−0.76	−0.60	−0.41	−0.62	−0.39
4	−1.00	−0.94	−0.14	−0.11	−0.24	−0.53
5	−1.00	−0.87	−0.35	−0.17	−0.46	−0.20

Results - negative reduced cost

	BPSO		Linearized surrogate		Nonlinear surrogate	
	Early phase	Later phase	Early phase	Later phase	Early phase	Later phase
Normalized mean of successfully solved pricing problems						
1	−1.00	−0.11	−0.053	−0.0076	−0.082	−0.028
2	−1.00	−0.98	−0.36	−0.012	−0.65	−0.33
3	−1.00	−0.67	−0.42	−0.36	−0.59	−0.34
4	−1.00	−0.47	−0.16	−0.083	−0.26	−0.15
5	−1.00	−0.70	−0.27	−0.054	−0.40	−0.092
Normalized min of reduced cost						
1	−1.00	−0.066	−0.058	−0.0062	−0.059	−0.032
2	−0.83	−1.00	−0.62	−0.054	−0.62	−0.42
3	−1.00	−0.76	−0.60	−0.41	−0.62	−0.39
4	−1.00	−0.94	−0.14	−0.11	−0.24	−0.53
5	−1.00	−0.87	−0.35	−0.17	−0.46	−0.20

Results - Improvement in RMP

	BPSO		Linearized surrogate		Nonlinear surrogate	
	Early phase	Later phase	Early phase	Later phase	Early phase	Later phase
1	0.94	0.50	0.18	0.021	1.00	1.00
2	1.00	1.00	0.082	0.0085	0.46	0.36
3	1.00	1.00	0.64	0.86	0.62	0.97
4	1.00	0.91	0.12	0.071	0.91	1.00
5	1.00	1.00	0.13	0.020	0.56	0.25

Results - Improvement in RMP

	BPSO		Linearized surrogate		Nonlinear surrogate	
	Early phase	Later phase	Early phase	Later phase	Early phase	Later phase
1	0.94	0.50	0.18	0.021	1.00	1.00
2	1.00	1.00	0.082	0.0085	0.46	0.36
3	1.00	1.00	0.64	0.86	0.62	0.97
4	1.00	0.91	0.12	0.071	0.91	1.00
5	1.00	1.00	0.13	0.020	0.56	0.25

Conclusions and Future Work

Conclusions

- BPSO pricer had overall best performance
- Nonlinearized surrogate pricer more robust at the later stage of the column generation process
- Linearized surrogate pricer worst performance $\implies$ linearization does not preserve rank of columns

Future work

- Impact of task selection for the surrogate modeling methods
- Implement MINLP solver that uses explicit surrogate function expression
- Reduce pricing run times - how much without affecting performance?
- In depth analysis of the performance compared to existing methods

Questions?