ML-based heuristic branch-and-price for the aircrew pairing problem

Pierre Pereira
Emeric Courtade
Daniel Aloise
Frédéric Quesnel
François Soumis
Yassine Yaakoubi

ESG UQÀM POLYTECHNIQUE MONTRÉAL

Introduction

The crew pairing problem

Description

- **Pairing :** Sequence of flights forming one or several days of work.
- **Goal :** Minimize cost, optimize for other KPIs.
- **Constraints :** Cover all flights, comply with regulations, etc.
- Pairings assigned to crew members in a second step.
- Large instances : several tens of thousands of flights

Solving the CPP in practice

Branch-and-price

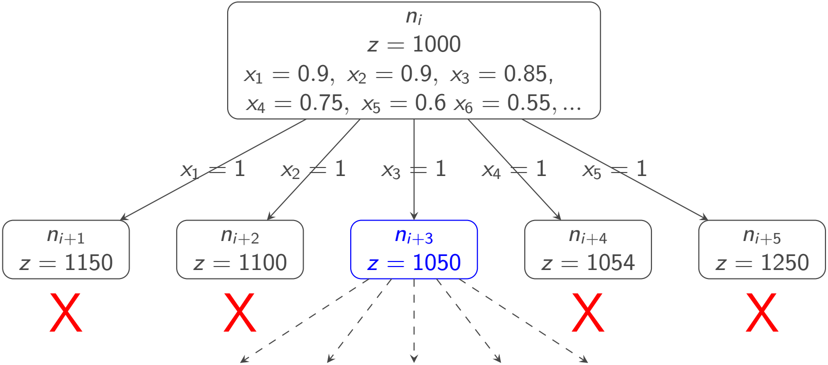
- Columns = pairings
- RMP : Set-partitioning problem.
- Subproblems : Constrained shortest path problems.

Heuristics *everywhere*

- Heuristic branching.
  That's the topic of the talk !
- Early stopping for relaxation.
  This will be relevant later.
- Exact pricing in our case
  Although heuristic pricing is common...

Heuristic branching schemes

Heuristic strong branching (HSB)



- Candidates : highest fractional value
- Slow but better results.

Digression - strong branching in practice

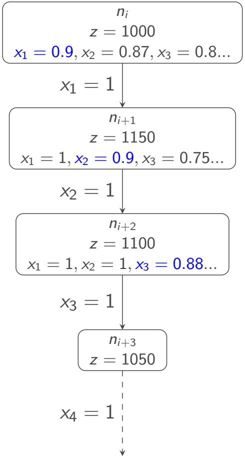
In our software...

- Generated columns are kept between nodes...
- So if you use HSB...
- And solve in decreasing order of candidate value...
- And solve heuristically each relaxation...
- The last (worse) candidate has more columns...

Therefore the worse candidate has an advantage !

Heuristic branching schemes

Diving heuristic (DH)



- Candidates : highest fractional value.
- Possible to fix several columns at once.
- Fastest but risky.

Our goal

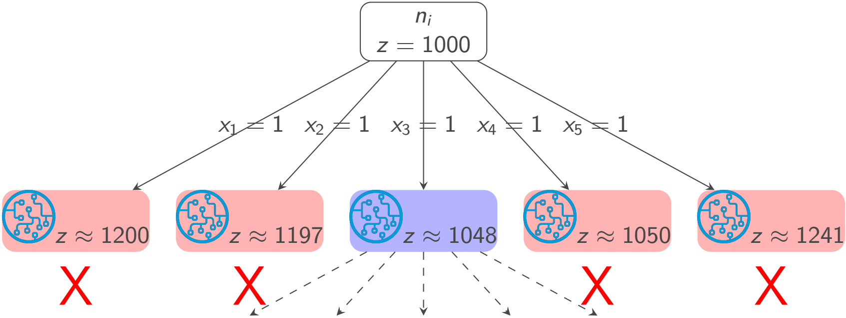
General objective

Develop a branching heuristic as good as heuristic strong branching, but as fast as heuristic diving.

Specific objectives

Develop a machine learning based heuristic branching scheme that imitates heuristic strong branching.

Our goal



- As good as HSB (if predictions are good enough).
- As fast as DH
- Considers several factors (not just fractional values).

Imitating strong branching

But what about Alvarez et al. (2017) ?

Alvarez et al. (2017)

- Small MIPS solved using B&B
  $\rightarrow$ Fixed probelem size.
- Exact method
  $\rightarrow$ We want to increases the lower bound the *most*.

Our research

- Large MIPS solved using B&P
  $\rightarrow$ Variable problem size.
- Heuristic method
  $\rightarrow$ We want to increases the lower bound the *least*.

Learning framework

Basic idea

- Solve several CPPs with HSB.
- Record data on branching decisions.
- Train a ML model to imitate HSB.
- Replace HSB with the ML model.

Data

Pairing data

- CPP instances from Kasirzadeh (2017)
- 7 base instances between 1000 and 7800 flights.
- $\rightarrow$ 63 instances by slightly perturbating the base instances.
  - Removing a few flights.
  - Perturbating side constraints.

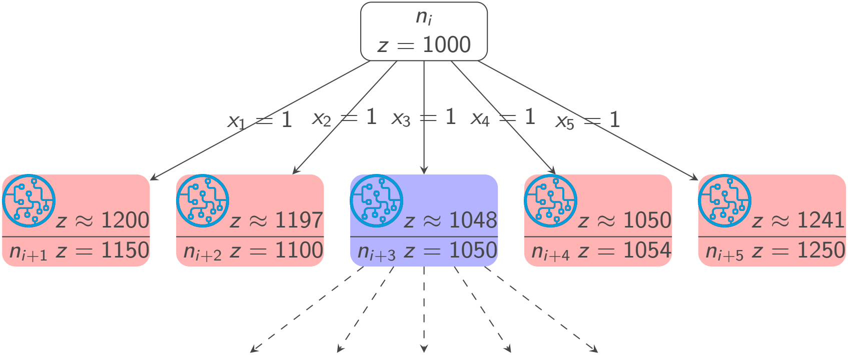
ML data

- 5 candidate columns per node
- 1 entry per candidate column
- Raw score : RMP optimal value after fixing the candidate column
- Learning features

Learning features

- Candidate column information
  - Value in RMP
  - Cost
  - Nb. tasks
- Parent node information
  - Nb. columns in RMP
  - Dual
  - % columns conflicting with candidate
  - % columns conflicting with positive value
  - ...
- Solving process information
  - Node depth
  - % of tasks fixed

Regression or classification ?



Normalization ?

Necessary to normalize the raw scores

- Tempered softmax

$$s'_i = \frac{e^{-s_i/t}}{\sum_{k \in K} e^{-s_k/t_k}}$$

  K = set of candidates.
- Improved normalization proposed
  - Similar to tempered softmax
  - Takes into account average and standard deviation among candidates.

ML Models

Linear regression

- Simple model, fast to train.
- DH heuristic included in the solution space.

Deep neural network

- Small-ish (3 layers, 150 neurons/layer)
- Should be able to derive more complex rules

Transformer encoder

- Consider all 5 candidates at once.
- Very small.
- It's what the cool kids do.

Training procedure

K-fold cross validation

- 1 base instance (9 instances total) left for testing
- Training performed on the remaining 6 base instances (42 instances total)
- Rotate the testing instances

Training

- 20% of training data put aside for validation.
- Trained with ADAM

Experimental Results

Prediction accuracy

Neural network accuracy

- $\approx 90\%$ *top-1* accuracy
- $\approx 95\%$ *close-to-best* accuracy

Diving heuristic accuracy

- $\approx 60\%$ *close-to-best* accuracy

Results for the CPP

CPU

Time (s)	I1	I2	I3	I4	I5	I6	I7
strong branching	178	227	889	40 481	44 639	51 478	109 365
diving heuristic	**49**	**55**	218	**9 037**	**8 690**	**11 450**	**23 999**
linear (our norm)	52	73	246	9 271	9 880	11 878	26 560
linear (softmax)	57	69	236	9 441	9 833	12 247	24 777
MLP (our norm)	52	71	248	9 743	10 450	12 406	25 084
MLP (softmax)	57	73	246	10 000	10 013	13 092	26 191
transformer (our norm)	55	62	**217**	10 713	10 052	12 027	26 763
transformer (softmax)	54	68	244	9 899	10 739	12 012	25 142

Results for the CPP

Solution value

Solution w.r.t. HSB (%)	I1	I2	I3	I4	I5	I6	I7	mean
diving heuristic	1.87	0.27	1.14	0.05	0.12	0.42	0.05	0.56
linear (our norm)	1.11	0.51	0.72	0.05	0.10	0.24	0.67	0.49
linear (softmax)	0.88	0.06	0.56	0.21	0.17	0.37	-0.04	0.31
MLP (our norm)	1.45	0.50	0.55	**-0.42**	0.21	0.47	0.15	0.42
MLP (softmax)	**0.68**	0.42	0.58	0.13	0.13	0.35	0.18	0.35
transformer (our norm)	0.92	0.42	**0.35**	-0.11	**0.07**	0.40	**-0.06**	**0.28**
transformer (softmax)	1.73	**-0.08**	0.89	0.57	0.27	**0.10**	0.12	0.51

Results for the CPP

Solution value

Solution w.r.t. HSB (%)	I1	I2	I3	I4	I5	I6	I7	mean
diving heuristic	1.87	0.27	1.14	0.05	0.12	0.42	0.05	0.56
linear (our norm)	1.11	0.51	0.72	0.05	0.10	0.24	0.67	0.49
linear (softmax)	0.88	0.06	0.56	0.21	0.17	0.37	-0.04	0.31
MLP (our norm)	1.45	0.50	0.55	**-0.42**	0.21	0.47	0.15	0.42
MLP (softmax)	**0.68**	0.42	0.58	0.13	0.13	0.35	0.18	0.35
transformer (our norm)	0.92	0.42	**0.35**	-0.11	**0.07**	0.40	**-0.06**	**0.28**
transformer (softmax)	1.73	**-0.08**	0.89	0.57	0.27	**0.10**	0.12	0.51

The bad news :

- Not much to gain in the first place...

Future work

Future work

Find a better-suited application

- Large-scale problem.
- Large gap between HSB and DH.

Improved learning

- Reinforcement learning.
- Features related to the application.
- How much training data is needed ?

Improved branching

- Fixing several candidates at once.
- Consider a larger pool of candidates.

Questions ?

Crew pairings, a costly chore,
Heuristics used, to solve once more,
Strong branching best, but slow to soar,
New strategy learned, solutions galore.

– ChatGPT