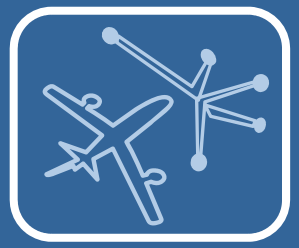




The Altitude Repair Module

Column Generation 2008

Daniel Villeneuve
Carl Mitchelson
Fabrice Lavier
Stéphane Bounkong

ADOPT
A Kronos Division

Overview

- The problem and its challenges
- A model and some algorithmic ideas
- The current approach
- Preliminary results and conclusions

Context

- Our client: international cargo airline
- Their customers: major freight forwarders
- Their business: airport-to-airport, heavy freight
- Impacts of business model on scheduling:
  - Average of 1 schedule change every 6 minutes
  - Every change requires manual fixing
  - Manual fixes do not take into account the global cost of the solution
  - Manual changes take time

ADOPT
A Kronos Division

Types of changes

- Mid-term schedule disturbances
  - Flight number changes
  - Equipment swaps
  - Delays and move ups
  - Cancellations
  - Crew illegalities

Glossary

- Base: pilot's “home” airport
- Leg: a flight segment, from airport X to airport Y
- Duty: sequence of legs, separated by “connections” (short waiting time)
- Trip (a.k.a. pairing): sequence of duties from base to base, separated by “layovers” (rest)
- Line: sequence of trips separated by “home base rest”, covering a month.

Scheduling process

- Plan for next month
  - Produce trips from flight legs (anonymous)
  - Produce lines from trips and bids (crew specific)
- React on day of operations (now)
  - Process changes in [now, now+2) with one team
  - Process changes in [now+2, now+10) with another team
- Consequence for pilots
  - Plan is used to identify work days, not much more

General problem statement

Given a coherent view up to time “now+2”:

- Produce a repaired and optimized solution for the 8-day interval [now+2, now+10)
- in a time frame that allows seamless integration of solutions into the client's real-time tracking system

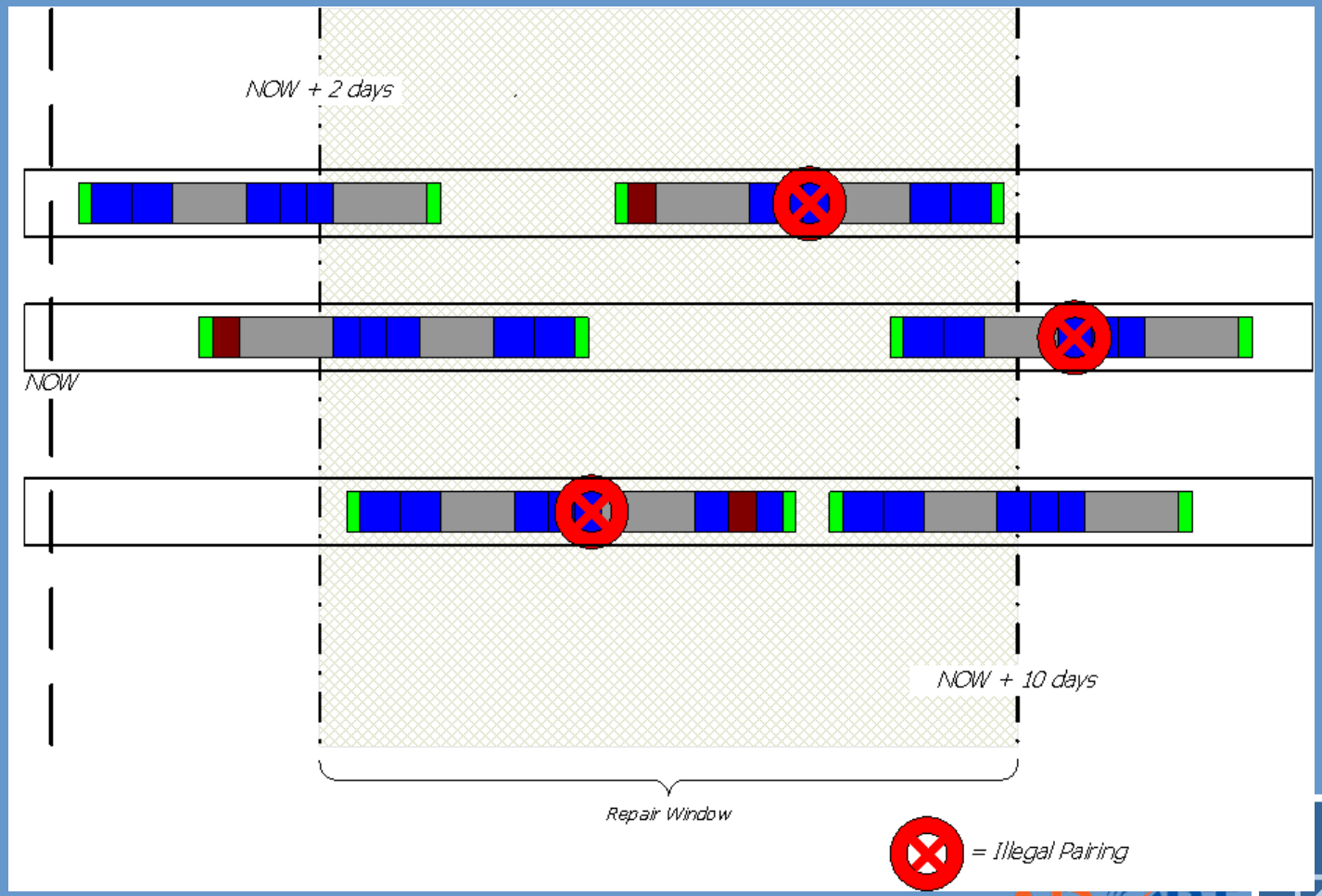
Client's main goal: reduce operational costs

No compromise to preserve current state

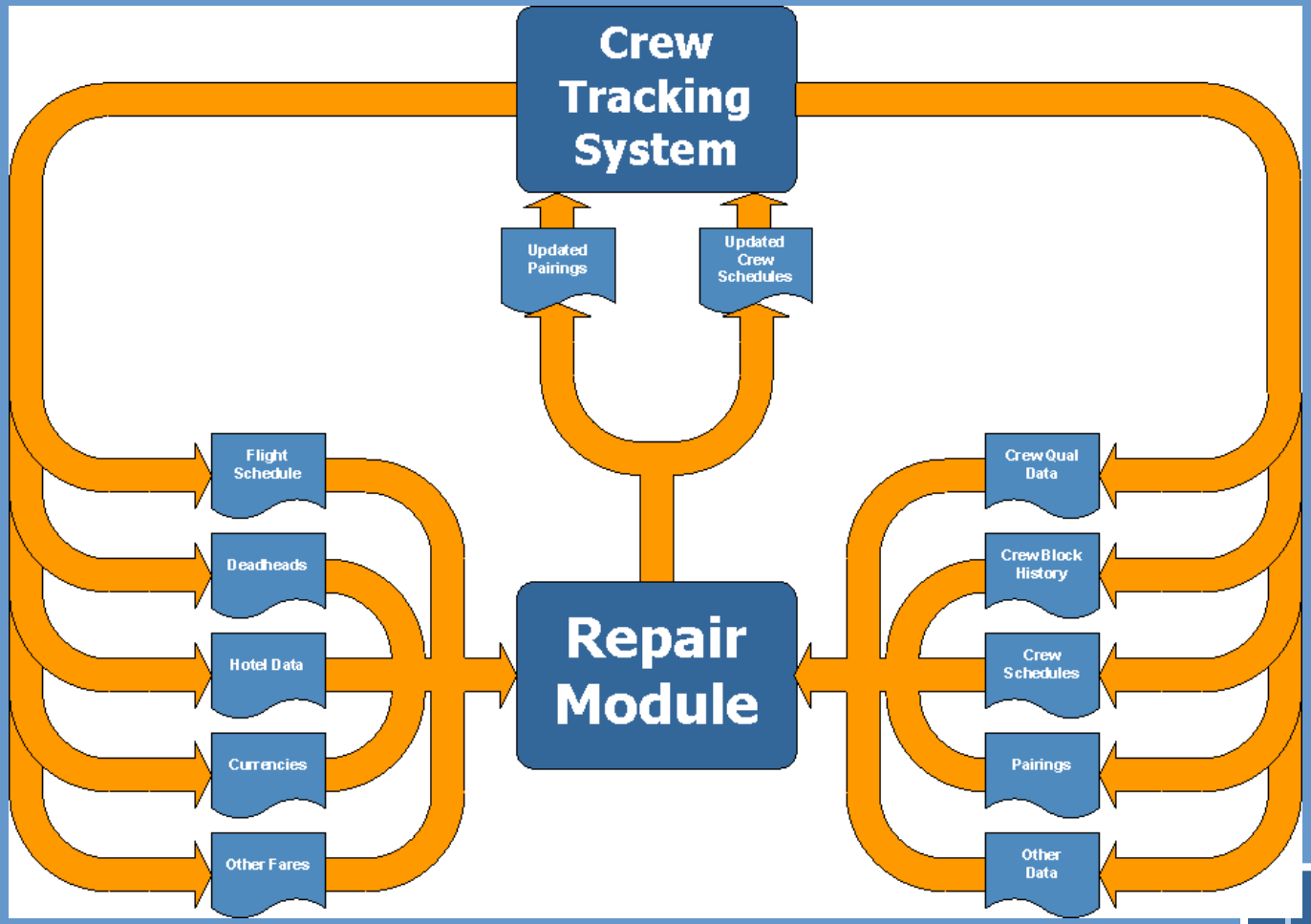
Note: coherent is not repaired

- Needed to prevent too much noise in data

Illustration – input data



Integration



ADOPT
A Kronos Division

Challenge – legs→lines

- Various boundary conditions for each pilot
- Target very few legs in open time
- Short horizon (in practice, between 4 and 8 days)

Producing trips separately from lines is risky

- Conflicts with carry-in trips and pre-assignments
- Likely to drop many (most?) legs in open time

⇨ Need new solver building lines from legs.

Challenge – runtime specs

~300 pilots, ~25 legs/day, 8 days ⇨ 30 minutes

Comparisons:

- Anonymous trip construction (planning)
  - Between 15 and 75 minutes
  - Relaxation: no line rules, no crew information
- Crew-specific lines from anonymous trips (planning)
  - About 40 minutes to get a legal solution (tabu search)
  - Relaxation: no trip rules

Challenge – “same-duty”

- Goal: minimize risk of missing connections associated to parts of crew not being available
- Rule: all pilots covering a leg must cover it using similar duties
- Similar: identical up to the last active leg, with same crew composition

⇨ Ripple effects associated to decisions on duties

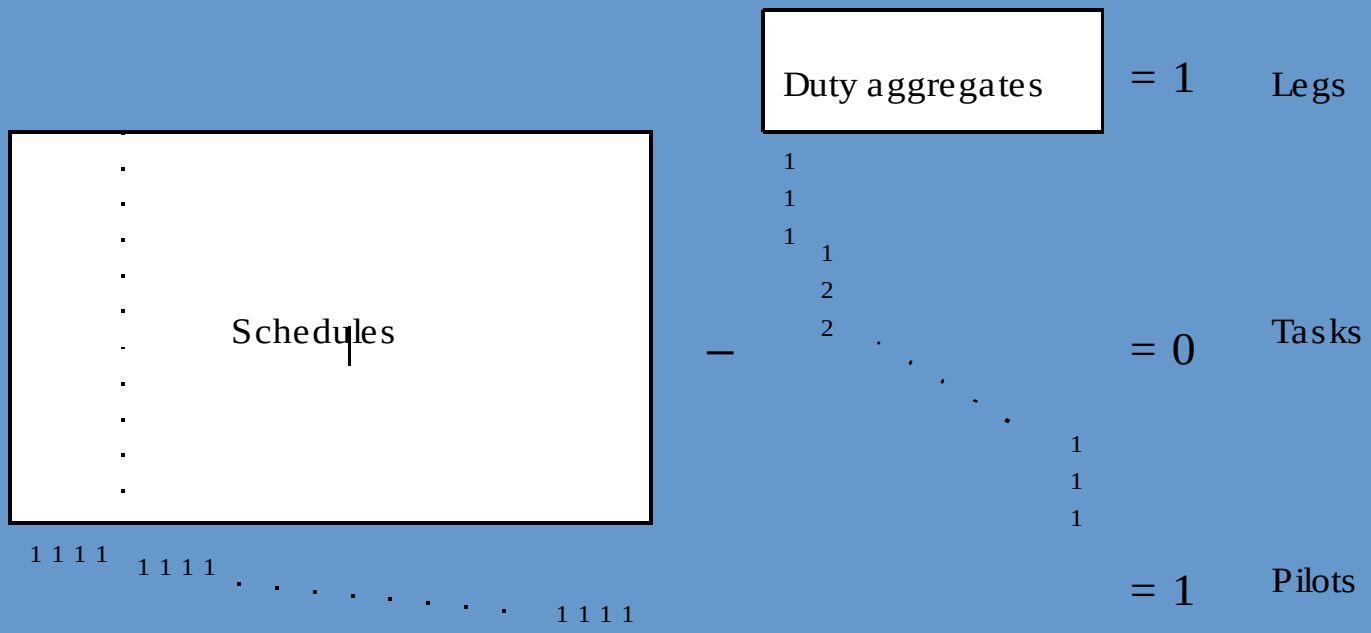
Solution approach – 2-level CRS

- Generate duties from legs using a duty generator
- Choose a set of duties covering legs
- Solve a duty-oriented CRS problem
- Provide feedback for choosing better duties
  - Favored feedback: using Benders decomposition
  - Other possibility: direct approach

Convergence is guaranteed (in theory)

Good performance reports in the literature

Model structure



Benders decomposition

- Linking variables are the duty aggregate binary variables
- Subproblem
  - CRS problem on tasks = (duty agg., crew position)
  - Solve linear relaxation while in Benders mode
  - Solve with integer constraints using best Benders (integer) solution found

Benders – LP then MIP

First solve Benders master problem as a LP

- Ref.: Mercier, Cordeau, Soumis (2005)

Pros:

- Faster to solve the master problem (marginal)
- Can use an interior-point LP algorithm
  - More central solutions, leading to fewer iterations

Cons:

- Cannot exploit the sparsity of CRS' rhs
- Numerical problems after a few tens of cuts

Benders – initial cuts

- Solution times still too long
- Too many iterations before finding a feasible CRS subproblem
- Idea: put a relaxed (and more tractable) copy of CRS in the master problem
  - network with side constraints

Pros: find feasible CRS solutions in 1 or 2 iter.

Cons: LP unstable after very few Benders cuts

Direct approach

- Benders decomposition experiments have shown
  - All variants work (solutions were produced)
  - Reducing solution times was proving difficult
- CRS subproblem is fully instantiated
- No clear advantages to use decomposition

Direct approach:

- Pros: quick feedback between schedule generation and choice of duty aggregates

Current approach

- Generate duties from legs using a duty generator
- Aggregate duties based on “same-duty” equivalence relation
- Build CRS networks mapping duties to agg.
- Use k-SPPRC (k > 1) to generate pilot lines
  - Some trip rules are only applied on complete paths
- Use “safe” branch-and-bound decisions to limit exploration of the tree

ADOPT
A Kronos Division

Current strategy

- Limit duty generation to about 25 000
- Use k=5 in k-SPPRC
- Branching rules, in decreasing priority:
  - Leg artificial variables: up first
  - Aggregate variables: closest integer
  - Task splitting: pilot with maximum participation first
  - Individual schedules: fixed to 1

Some results

- 4-day horizon
  - 249 pilots, 77 legs, 9932 duties, 533 aggregates
  - LR: 96s, SOL1: 677s, SOL2: 897s, total: 1082s
  - 78 b&b nodes, depth: 47, 5 legs in open time
- 5-day horizon
  - 247 pilots, 96 legs, 11 695 duties, 631 aggregates
  - LR: 293s, SOL1: 1532s, SOL2: 2164s, total: 2520s
  - 101 b&b nodes, depth: 60, 7 legs in open time

Current state

- Repair Module about to move into production

- Ideas untested yet:
  - Speed-up problem preparation
  - Use meta-heuristic on top of branch-and-bound
    - Take better advantage of incumbent solution
  - Relaxed (merged) pilot networks

Related problems

- What-if scenario analysis about conflicting business opportunities
- For airlines using a bidline approach to line construction
  - Pilots can bid for lines that conflict with their tasks
  - Could formulate the residual problem on dropped trips as a Repair problem
- When building new trips in planning mode, use up-to-date crew information to repair carry-in trips at the start of “next month”

The End

Questions, thoughts and comments are welcome

ADOPT
A Kronos Division