Building and Optimizing Timetables for Airport Employees

Philippe Morignot, Laurent Somek, Christophe Miller, Bruno Gaudinat

PACTE NOVATION
PACTE NOVATION

- Start up company
- Created on April 1994
- 70 persons
- Assets: 512 kEuros
- Business strategy:
  - technical expertise.
  - Know-how accumulation.
Main Problem (1)


• Connects to existing ADP databases.

• Interactive Gantt chart.

⇒ We focus on the optimization part.
Main problem (2)

- Build a timetable for each group of airport on-ground employees.
- Optimize this timetable regarding various quality criteria.
- Insert incoming tasks on the fly into a timetable.
- Perform the main operation (assignment) in less than 1 minute.
• The MAXIME system may be used in three modes:
  – **Planning** (MAX-P): tasks are assigned to shifts in a row 6 months in advance during one night through a computer background task.
  – **Day-to-day** (MAX-J): tasks may be changed and inserted into an existing timetable, due to unexpected flight events.
End user

- A **supervisor** of a group of on-ground agents/employees.
- No specific knowledge in computer science.
- S/he defines in real-time and/or in advance who will work on what.
- So far, **one day** to prepare a single timetable.
Tasks

• Each task is fixed in time (*regular task*) (e.g., check in).

• Some tasks are attached to a shift, but are movable in time within a range (*movable tasks*) (e.g., medical visit).

• Some movable tasks are optional (*optional movable tasks*) (e.g., lunch/dinner breaks).

• Task must **not** overlap + inter-task time.
5 modules

Tasks
- Task reduction and split
- Assignment of tasks to agents
- Optional movable tasks increase
- Global social equity increase
- Task insertion

Shifts
- Deterministic alg.
- SOLVER
- “Tabu” search alg.
- “Tabu” search alg.
- A* search alg.

Timetable
Task reduction and split

- The workload curve reveals main peaks at 11am and 3pm.
- **Counter-productive**, since the supervisor must hire agents the whole shift for a few minutes of actual work.
- **Goal**: reduce peaks by task reduction and split.

⇒ Deterministic sweepline algorithm.
Assignment of tasks to agents (1)

• **Goal:** assign tasks to shifts while minimizing the number of unassigned tasks and, if possible, maximizing the number of empty shifts.

⇒ Use constraint programming and heuristics for the min/maximization criteria (see “Discussion” part).
Assignment of tasks to agents (2)

- **Primary model**: An integer variable is attached to each task: the number of the shift the task is assigned to.
- **Dual model**: an integer set variable is attached to each shift, each element is a task number.
Assignment of tasks to agents (3)

• The constraints are:
  – Integrity;
  – Adequation;
  – Non overlapping;
  – Movable tasks always present;
  – Minimum number of optional movable tasks.

• Solutions are enumerated, so as to heuristically minimize the number of unassigned tasks.
Assignment of tasks to agents (4)

- Heuristics for choosing a shift for a task are:
  - Leftmost (earliest in the shift);
  - Most filled (shift);
  - Least filled;
  - Leftmost + most filled
  - Regret (smallest number of tasks which could have taken its place);
  - Priority (comparison workload and shift load).
Movable tasks increase (1)

- The previous module has a fixed number of optional movable tasks (e.g., lunch, dinner breaks).
- **Goal:** increase the number of optional movable tasks.
- A first solution already exists
  ⇒ **Tabu algorithm** with an escape mechanism (to avoid local minima).
Movable tasks increase (2)

- Tasks are moved or switched among shifts, to increase the number of optional movable tasks.
Global social equity increase

• **Goal**: no agent should complain because s/he has a clearly worse shift than the others regarding complexity, diversity, mobility, productivity.

• A first solution already exists (i.e., generated by the previous module).

⇒ As before, **tabu algorithm** with an escape mechanism.
Task Insertion (1)

- New flights may come unexpectedly. They must be serviced as regular ones.
- New incoming tasks must be inserted into an already built timetable, while minimizing the number of changes into this existing timetable.

→ A* algorithm
Task insertion (2)

- An A* algorithm is a best-first search in the space of paths.
- **State:** a whole timetable + the task to insert.
- **Successors:** move or switch of tasks.
- **Cost function from the initial state to the current state:** the number of states.
- **Cost function to the expected solution:** zero.
Task insertion (3)

- Several tasks may be inserted at once: completely solve the first one and then the others.

- **Drawback**: Computationally explosive.

- Limited in time: At least, it protects the supervisor from obvious changes (distant of less than 7 states in practice in 15 sec.).
Implementation

• The 5 models have been developed on PC machines with C++ and ILOG SOLVER.
• Ported on Bull’s RS6000 machine for performance increase.
• Task assignment takes 20 sec. for 600 tasks and 150 shifts. 15 iterations in 15 sec. for each tabu algorithm (17,000 scanned states per performed state).
Discussion

- A **second** model for task assignment: mixed integer programming (ILOG CPLEX).
- But 2.5 hours for the relaxed solution!!!
- A **third** model for task assignment: scheduling (ILOG SCHEDULER).
- But 5 movable tasks per shift only and less flexibility regarding dynamic heuristics.
Conclusion

• 5 modules for tasks assignment and quality improvement of timetables.
• Good quality of the built timetables, say expert supervisors => ADP expresses satisfaction.
• Operational for ADP supervisors at Roissy and Orly on a daily basis since June 2000.