SCHEDULING AIRLINE GROUND STAFF WORKERS UNDER FLIGHT DELAY UNCERTAINTY

15.094 Class Project

Julie Poullet
Samuel Gilmour

MOTIVATION AND AIMS

- Ground staff agents employed by airlines carry out essential jobs at airports, include checking-in passengers and cleaning planes.
- Failure to schedule teams to complete these jobs forces airlines to hire expensive backup agents in their place.
- Airlines therefore need to develop ground staff schedules robust to the large uncertainty which affects the starting times of these jobs.
- This project aimed to:
  - Develop a solution method for the robust formulation that can schedule workers for job sets of practical size.
  - Investigate how adding adjustability to schedules in a two-stage forumulation improves performance.

PROBLEM DESCRIPTION

INPUTS

- W Number of workers available to the airline.
- J Number of jobs. Each $j \in [J]$ has start time t_j^b and end time t_j^e.
- F Number of flights, with a function $\text{flight}(j) \in [F]$ for each $j \in [J]$.
- T Time defining boundary between first and second-stage jobs.

UNCERTAINTY

Uncertainty set of flight delay, so each $j \in [J]$ starts at time $t_j^b + u_{\text{flight}(j)}$ and ends at time $t_j^e + u_{\text{flight}(j)}$. Based on Air France data, we selected:

$$\mathcal{U} = \{\mathbf{u} \in [-30 \text{ mins}, 60 \text{ mins}]^F : \|\mathbf{u}\|_1 \leq F \cdot 15 \text{ mins}\}$$

DECISIONS

- $x_{ij}^1, x_{ij}^2(\mathbf{u}) \in \{0,1\}$ First and second-stage decisions for completing job j with worker i.
- $y_j^1, y_j^2(\mathbf{u}) \in \{0,1\}$ First and second-stage decisions for completing job j with a backup agent.

OBJECTIVE

Minimise the worst case number of jobs completed by backup agents such that all jobs are completed by either a worker or backup agent.

ASSUMPTIONS

- Workers and backup agents complete the entire job if they are assigned to it.
- Job durations are deterministic.
- Workers and backup agents can only work on a single job at any one time.

EFFECT OF ADAPTABILITY

- **Figure 1** shows an example where adaptable $x_{ij}(\mathbf{u})$ variables cover all jobs with strictly fewer workers than static x_{ij} variables.
- In this example, assume the uncertainty set is given by:

$$\mathcal{U} = \{\mathbf{u} \in [-1, 1]^3 : \|\mathbf{u}\|_1 \leq 2\}$$

- Pairs of jobs can clash separately, but not all three at once. Static variables prohibit a single worker from completing more than one job, but adaptable variables complete all tasks with 2 workers.

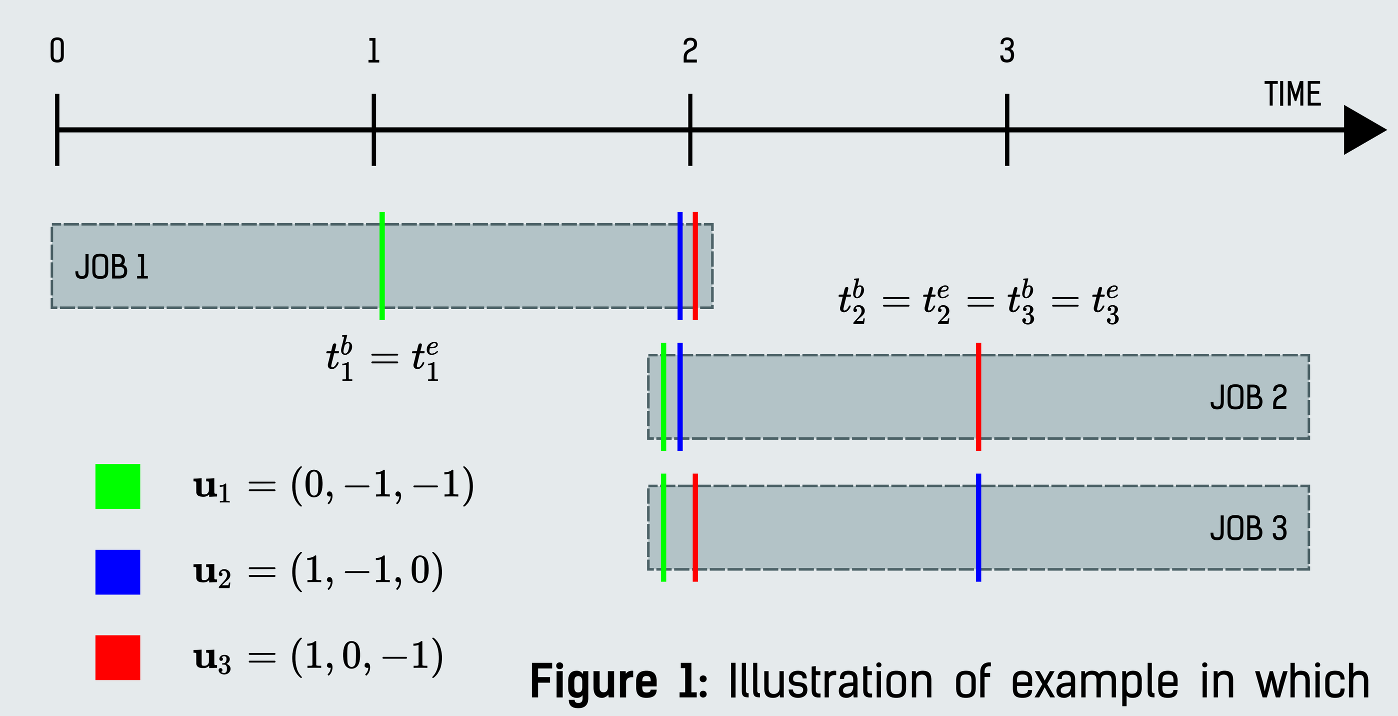


Figure 1: Illustration of example in which adaptability strictly improves objective.

NAIVE MIP FORMULATION

- The problem was first written as a binary program with $z_{ij}(\mathbf{u}) \in \{0,1\}$ equal to 1 if job j finishes after k starts when $\mathbf{u}$ is realised:

$$-Mz_{jk}(\mathbf{u}) \leq (t_k^b + u_{\text{flight}(k)}) - (t_j^e + u_{\text{flight}(j)}) \quad \forall \mathbf{u} \in \mathcal{U}$$

- As a robust problem with binary adaptable variables, the adaptive partitioning scheme in Bertsimas & Dunning (2016) was implemented to solve this problem.
- At each iteration of the algorithm, the uncertainty set was partitioned and used to solve a static problem with copies of second-stage variables in each partition.
- The Voronoi partitioning scheme in this paper resulted in 168 partitions and an intractable problem at one iteration (for 50 jobs):

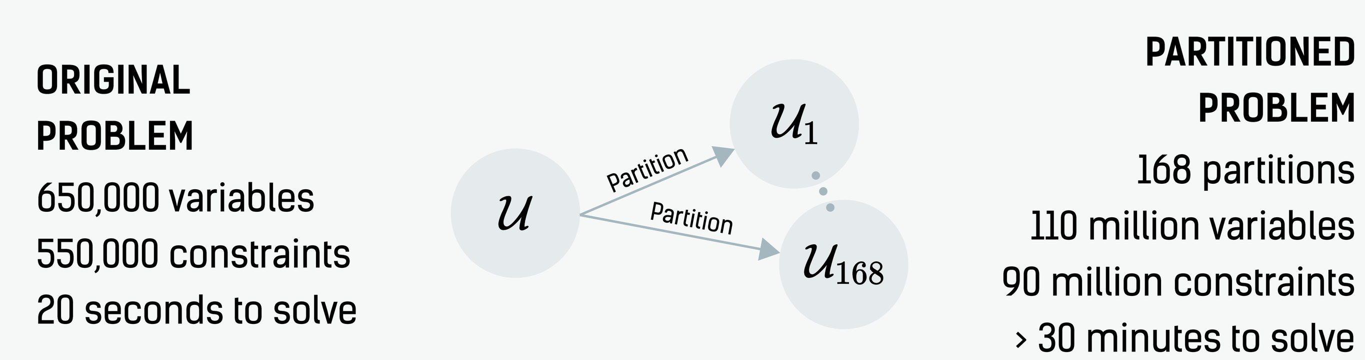


Figure 2: Growth in problem size after single iteration of partitioning.

GRAPH FORMULATION & RESULTS

- The problem was efficiently reformulated as a modified network flow as in **Figure 3**, with one graph for each uncertainty partition:

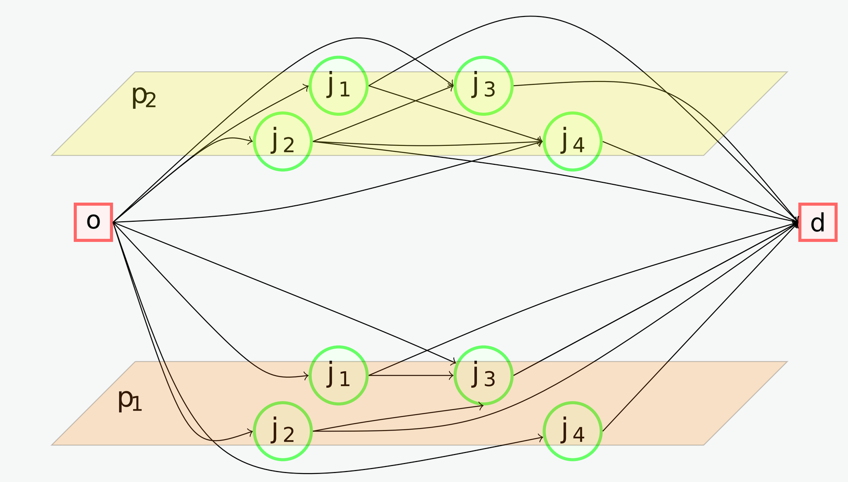
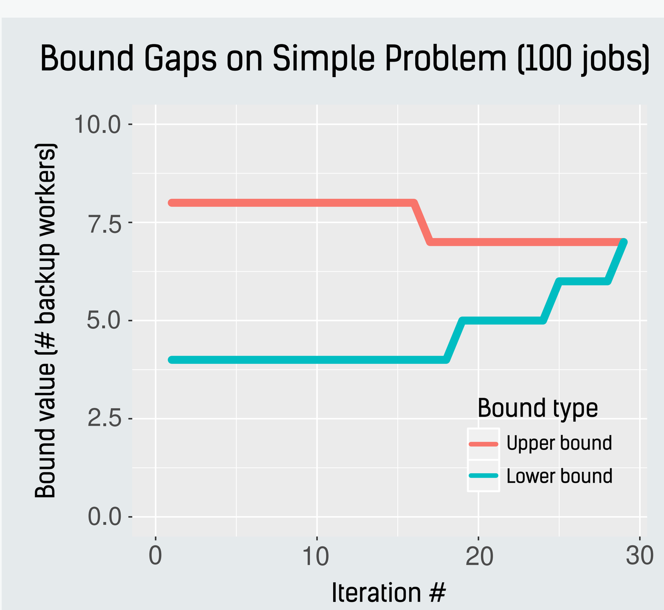


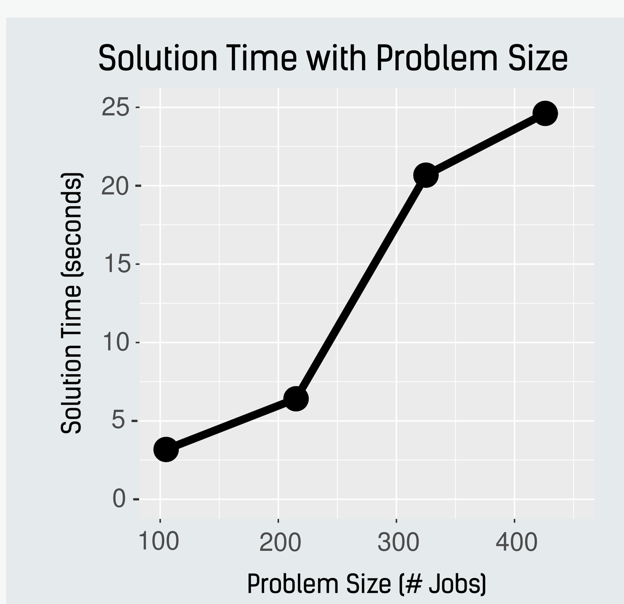
Figure 3: Structure of graph reformulation.

- Nodes correspond to jobs, and arcs correspond to job successions that are feasible in the given partition of uncertainty.
- The graph reformulation contained fewer variables than the MIP, since it is known a priori that a link between jobs j and k in partition p is feasible if:

$$\min_{\mathbf{u} \in \mathcal{U}_p} (t_k^b + u_{\text{flight}(k)}) - (t_j^e + u_{\text{flight}(j)}) \geq 0$$

- Combining the graph formulation with a partitioning heuristic that exploited problem structure made problems of up to 400 jobs solvable, but the two-stage formulation provided minimal gain:





CONCLUSIONS

- A sensible graph model minimising the number of variables allows the two-stage problem to solve for instances of up to 400 jobs.
- Adding adaptability to worker scheduling decisions results in only small improvements to the objective when tested on actual data.

REFERENCES

Bertsimas, D., & Dunning, I. (2016). Multistage robust mixed-integer optimization with adaptive partitions. Operations Research, 64(4), 980-998.