Automated Sequential Resource Planning

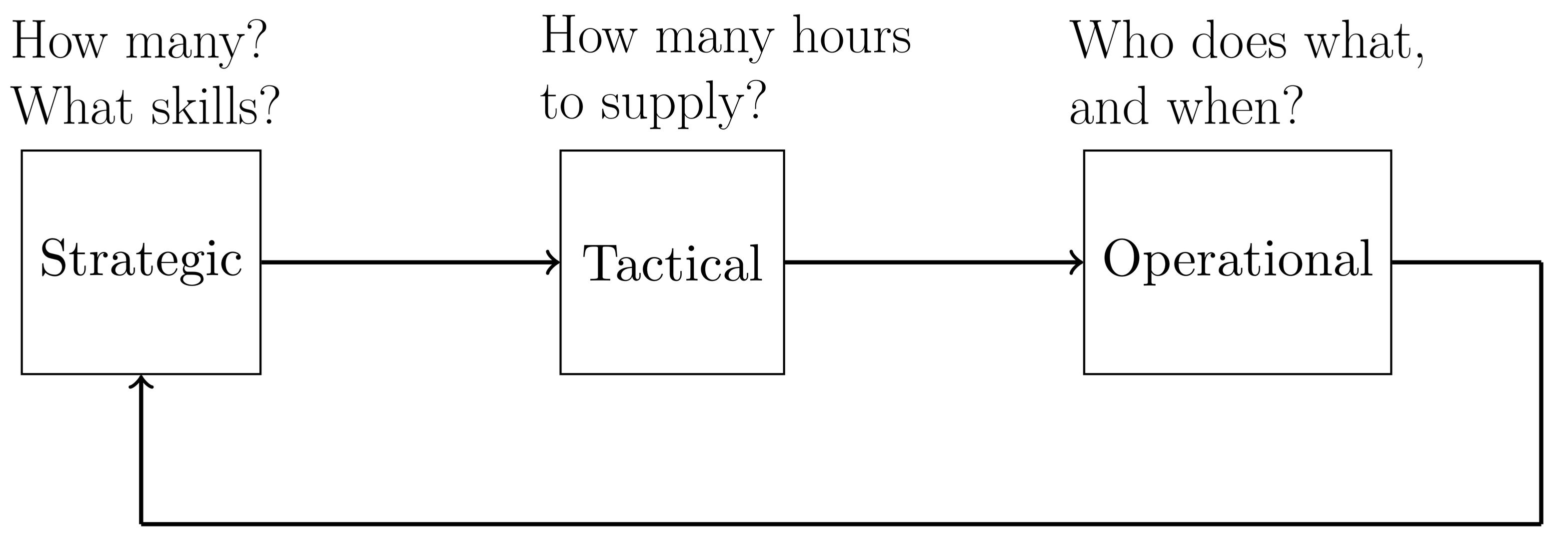
Ben Black[1], Chris Kirkbride[2], Vikram Dokka[2], Russell Ainslie[3]

[1]STOR-i CDT, Lancaster University, [2]Management Science, Lancaster University, [3]BT

BT's Multi-skilled Workforce

- Multiple types of jobs each requiring **different skills**.
- Over **30,000 engineers** working in the field.
- Jobs and supply hours both **aggregated by skill**.

The Three Stages of Planning



Tactical Planning: matching **supply** (engineers' hours) with **demand** (job-hours) across all days and skills.

BT's Tactical Planning Problem

On each day $t = 1, \dots, T$:

1. A plan should give the number of engineer-hours to dedicate to each skill $j \in \mathcal{J} := [J]$ on each day $\tau \in \mathcal{T}(t) := \{t+1, \dots, t+L\}$.
2. Jobs are either **appointed** (AP) or **non-appointed** (NA):
   a. AP: done **on** their due date.
   b. NA: done **by** their due date.
3. The forecasts for numbers of NA and AP jobs due on each τ for each j are updated.

Task: create a new plan **automatically**, given the old plan and new demand, to maximise job completions in the future.

A Sequential Planning Model

Re-solving a planning model **every day** can be **inefficient**. Instead, we **update the old plan**. At the start of day $t = 1, \dots, T$:

- Yesterday's plan is P^t, it covers all $j \in \mathcal{J}$ and $\tau \in \mathcal{T}(t-1)$.
- The newest job-forecasts are $u^t_{j,\tau}$, $v^t_{j,\tau}$ for each (j, τ).
- There are c^t_τ hours of supply still unallocated on each day $\tau \in \mathcal{T}(t)$.
- The **state of the system** is defined as $S^t = (P^t, u^t, v^t, c^t)$.

Given S^t, we want to adjust P^t to create P^{t+1}. For each (j, τ):

1. $N^t_{j,\tau}$ is the number of hours of supply to **add** into the plan.
2. $R^t_{j,\tau}$ is the number of hours of supply to **remove** from the plan.
3. $M^t_{j,\tau,\tau-k}$ is the number of NA jobs to **move** from day τ to $\tau - k$.

Using our **action** at time t, $A^t = (N^t, M^t, R^t)$, the new plan and capacities are:

$$P^{t+1} = P^t + N^t - R^t,$$
$$c^{t+1}_\tau = c^t_\tau - \sum_{j \in \mathcal{J}} N^t_{j,\tau} + \sum_{j \in \mathcal{J}} R^t_{j,\tau}.$$

A feasible action $A^t \in \mathcal{A}(S^t)$ satisfies a number of **constraints**, e.g. not adding more supply than is available, not moving jobs to days with no free supply, etc.

The **cost** of A^t given S^t, where $q(P^{t+1}, u^t, v^t)$ is a penalisation for missed jobs, is:

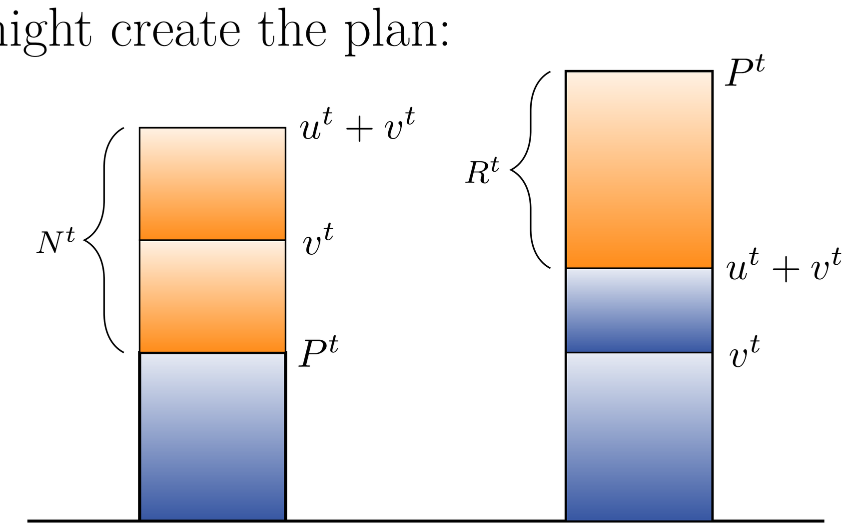
$$C(A^t, S^t) = \sum_{j \in \mathcal{J}} \sum_{\tau \in \mathcal{T}(t)} \left\{ \sum_{k=1}^{k_{\max}} a_{j,k} M^t_{j,\tau,\tau-k} - p_j N^t_{j,\tau} + w_j R^t_{j,\tau} \right\} + q(P^{t+1}, u^t, v^t),$$

The **objective** is $\min_{A \in \mathcal{A}(S)} \left\{ \sum_{t=1}^{T} C(A^t, S^t) \right\}$.

Two Myopic Algorithms

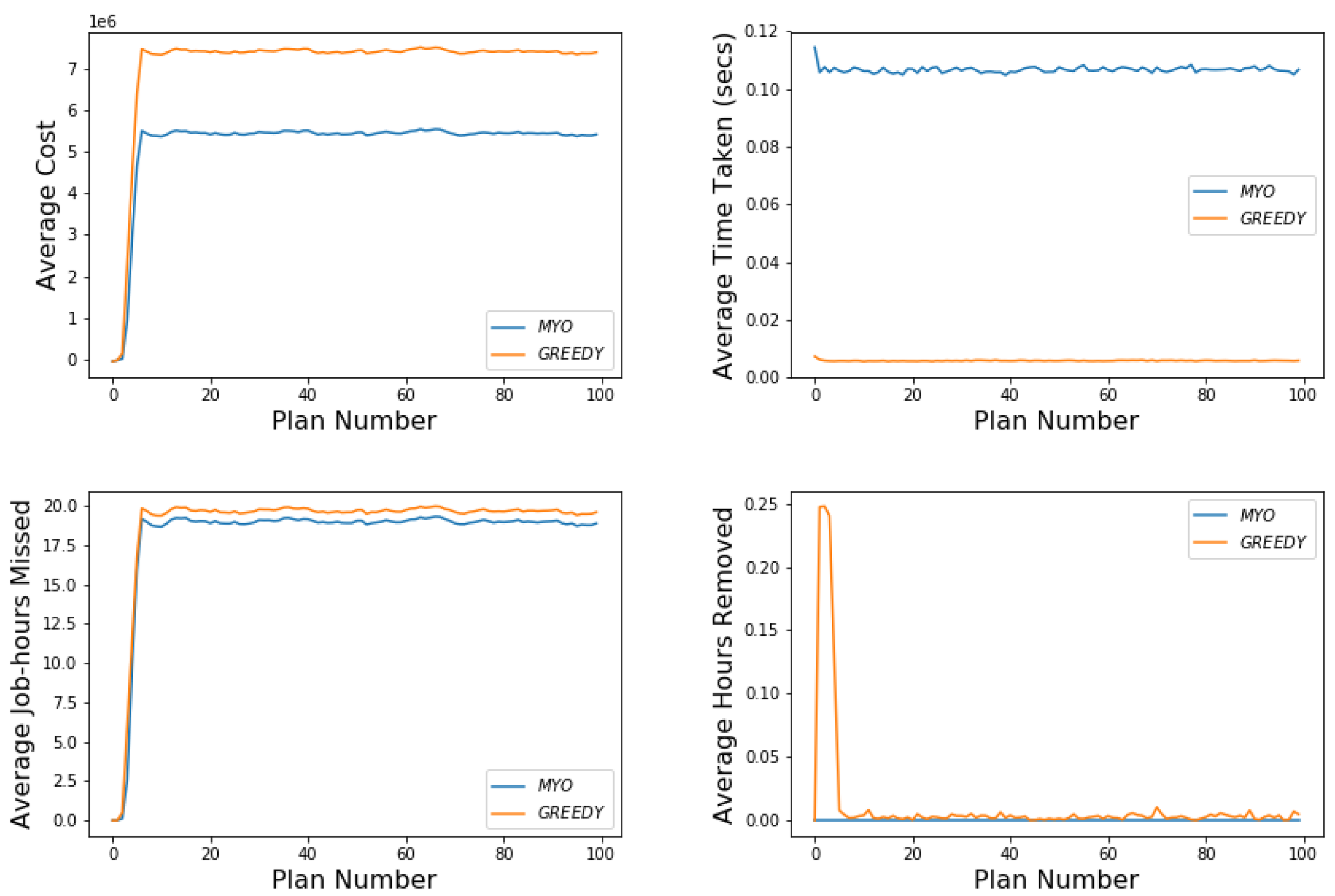
Currently we have only studied algorithms that are **myopic** (short-sighted):

1. ***MYO***: solve the problem at each t as a linear program, $\min_{A^t \in \mathcal{A}(S^t)} C(A^t, S^t)$.
2. ***GREEDY***: based on how a human might create the plan:
   a. Add supply to cover as many jobs as possible, prioritising AP.
   b. Move incomplete NA jobs to days with spare supply.
   c. Remove any extra supply that cannot be used.



Experiments and Results

To study how the algorithms compare, we tested them each on the same 100 sets of $T = 100$ days of demand, with $J = L = 7$.



Results Summary: *GREEDY* is much faster, but has higher cost. However, the output plans are almost identical, so there isn't really any reason not to use *GREEDY*.

Future Work and Project Aims

1. **Incorporate planning levers**, i.e. ways to control capacity to complete more jobs (e.g. overtime, contractors).
2. **Predict due dates**. BT only estimate the number of jobs that **will exist**, not that **will be due**, on each day.
3. **Use reinforcement learning** algorithms to produce plans that won't need to be changed in the future.

References

R. T. Ainslie, S. Shakya, J. McCall, and G. Owusu. Optimising skill matching in the service industry for large multi-skilled workforces. In *Research and Development in Intelligent Systems XXXII*. Springer, 2015.

Russell Ainslie, John A. W. McCall, Sid Shakya, and Gilbert Owusu. Tactical plan optimisation for large multi-skilled workforces using a bi-level model. *2018 IEEE Congress on Evolutionary Computation (CEC)*, 2018.

http://www.lancs.ac.uk/~blackb

b.black1@lancaster.ac.uk