



**Modelling V: scheduling jobs
on multi-capacity resources**

Job Shop Scheduling

Last lecture

A job is a set of tasks, and each job has one task per resource.

The tasks in a job are totally ordered, and one must finish before the next one starts.

Only one task can be active on a machine at any one time.

Given a solution, the finish time of the last task to finish is the *makespan*.

The aim is to minimise the makespan.

Job Shop Scheduling

Last lecture

A job is a set of tasks, and **each job has one task per resource.**

The tasks in a job are totally ordered, and one must finish before the next one starts.

Only one task can be active on a machine at any one time.

Given a solution, the finish time of the last task to finish is the *makespan*.

The aim is to minimise the makespan.

Scheduling processes in limited memory

Suppose we want a number of jobs to do in various software tools, but we have limited memory

- each tool or package occupies at least a minimum amount of available memory

Can we open all tools at the one time?

If not, how do we sequence them to get the workflow completed by a deadline?

Resource: available memory

Task: computational process, which occupies memory

Scheduling vehicle repairs

A maximum number of vehicles can be in the repair shop at any one time, and the staff have a limited number of vehicles they can handle.

Can we schedule all repair requests in a given time?

Resource: space, repair workers

Task: vehicle repair, which occupies space and staff

Managing Customer Power Demand

Electricity companies want to limit the total power demand in a district, since if they receive higher demand, a new generator has to be brought onstream.

They impose a power threshold on consumers – if you stay below the threshold, you get a low price per unit, but if you exceed it, you are charged a higher price all day.

Each appliance use demands certain power. Can we schedule all appliance uses in a time window and stay below the threshold?

Resource: power

Task: appliance use, which demands power

Finite Capacity Resource Scheduling

Each resource or machine will have a maximum capacity at any time point.

Each task has a duration and a resource consumption.

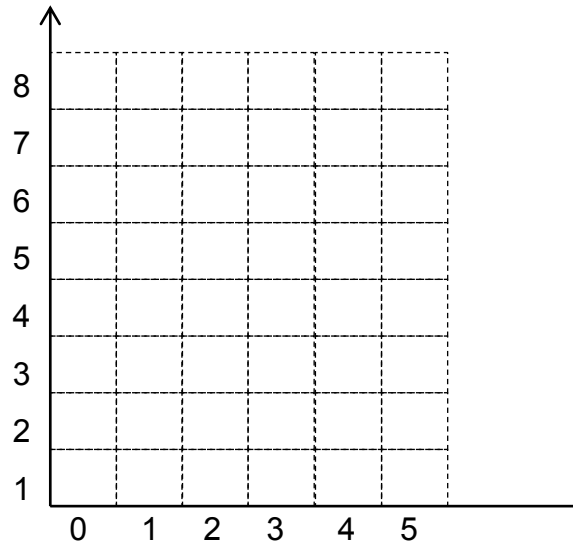
Can we schedule all tasks within a fixed period, but without exceeding the resource capacity?

Sample Problem

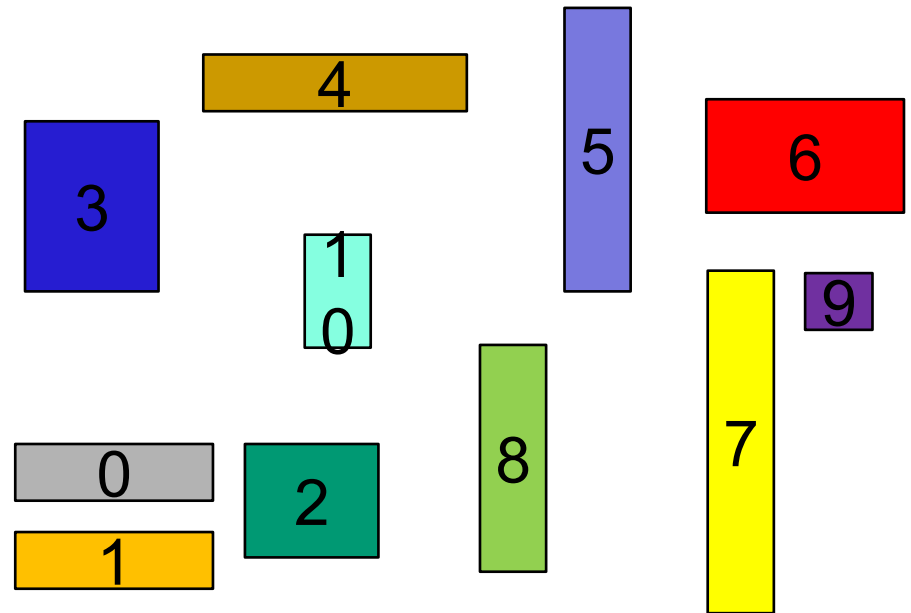
Can we fit all of these tasks into the resource profile?

Note it is not a rectangle packing problem – task 4 does not have to be scheduled at a constant height...

consumption



time

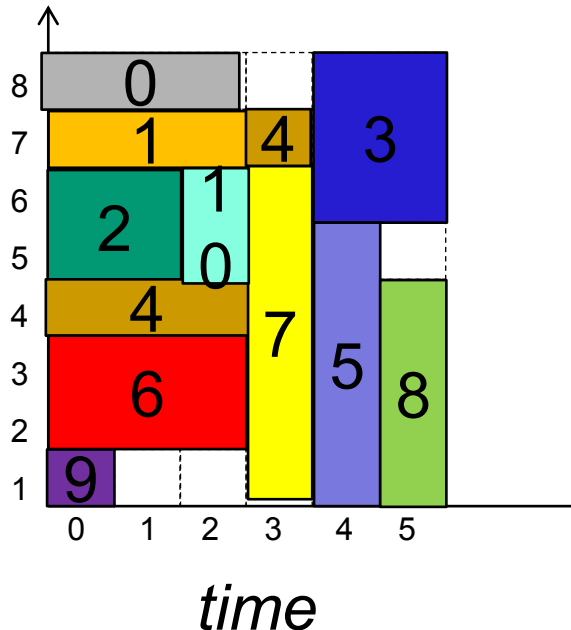


Sample Problem

Can we fit all of these tasks into the resource profile?

Note it is not a rectangle packing problem – e.g. task 4 does not have to be scheduled at a constant height...

consumption



How do we solve it as a constraint problem?

Constraint Model

Each task has a start time, a duration, an end time, and a consumption rate. The duration and consumption are fixed.

$\text{start} + \text{duration} = \text{end}$

Define a task to be active in time t if $t \in [\text{start}, \text{end}-1]$

Assign start times to the tasks so that for each time point, the sum of the consumption rates for each active task is less than or equal to the resource capacity.

Global constraint: cumulative

Last lecture

•cumulative

```
public static Constraint cumulative(Task[] TASKS, IntVar[] HEIGHTS,  
IntVar CAPACITY, boolean INCREMENTAL)
```

Cumulative constraint: Enforces that at each point in time, the cumulated height of the set of tasks that overlap that point does not exceed a given limit.

Parameters:

TASKS - TASK objects containing start, duration and end variables

HEIGHTS - integer variables representing the resource consumption of each task

CAPACITY - integer variable representing the resource capacity

INCREMENTAL - specifies if an incremental propagation should be applied

Returns:

a cumulative constraint

Using the Constraint Model

Note that this model describes the basic problem of fitting a set of tasks into one resource profile.

The model can be extended by

- adding precedence constraints between tasks
- adding non-overlap constraints between tasks
- adding time windows for tasks
- adding multiple resources with their own tasks and constraints between the tasks on different resources
- ...
- adding any variables and constraints we like ...

Changing Capacity Resource Scheduling

We don't need to stick with a fixed resource capacity.

E.g. staff numbers may drop during lunch breaks, or during vacations

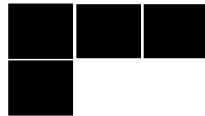
The electricity company may allow a higher threshold when it knows extra generators will be available.

```
public static Constraint cumulative(Task[] TASKS, IntVar[] HEIGHTS,  
IntVar CAPACITY, boolean INCREMENTAL)
```

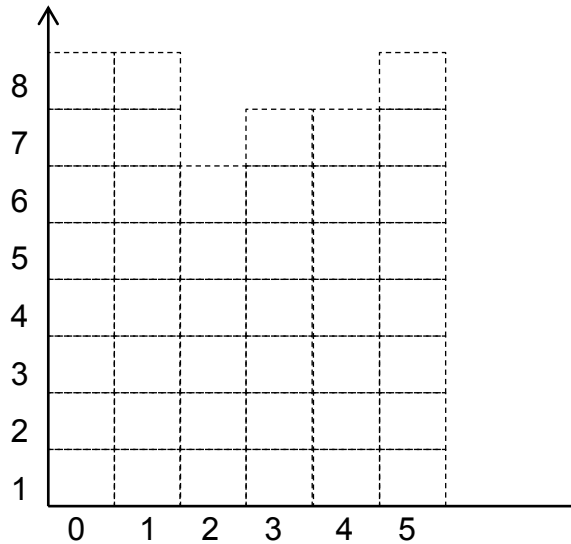
... but capacity is a single value for the resource?

Changing Resource Capacity

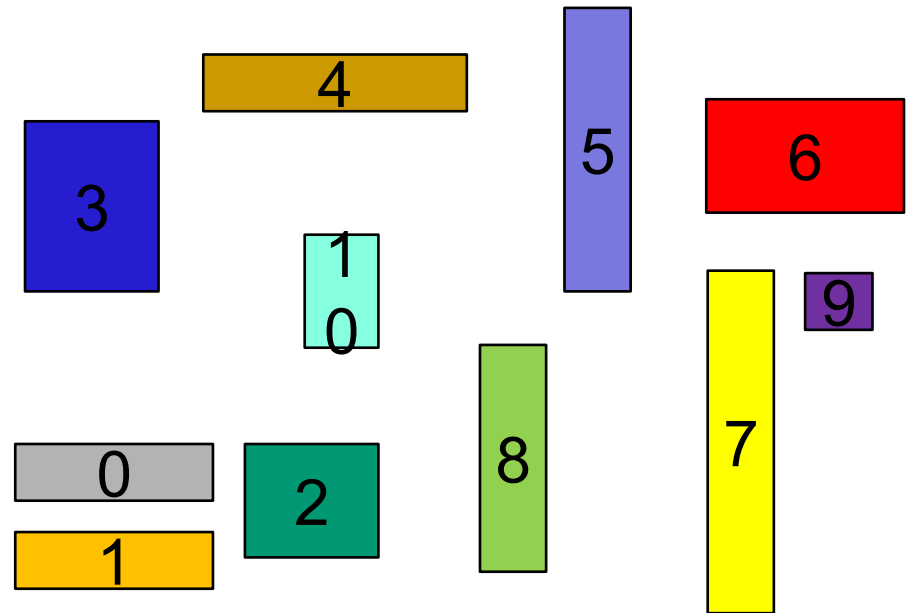
- set the capacity to the maximum required
- add dummy tasks into the problem with fixed start times, durations and heights, to represent the drop in capacity



consumption



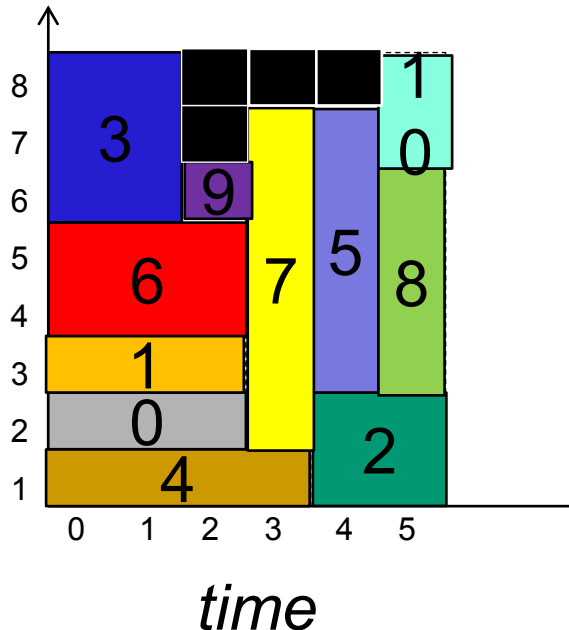
time



Changing Resource Capacity

- set the capacity to the maximum required
- add dummy tasks into the problem with fixed start times, durations and heights, to represent the drop in capacity

consumption



Allowing optional tasks

When the problem has no solution, we might still want to find a schedule which allows start times for as many of the tasks as possible.

```
public static Constraint cumulative(Task[] TASKS, IntVar[] HEIGHTS,  
IntVar CAPACITY, boolean INCREMENTAL)
```

Instead of fixing the heights (or consumption rates), we can take advantage of the IntVar, and allow a choice of values

- the original height
- 0, to represent omitting the task from the solution

Maximising the assigned task

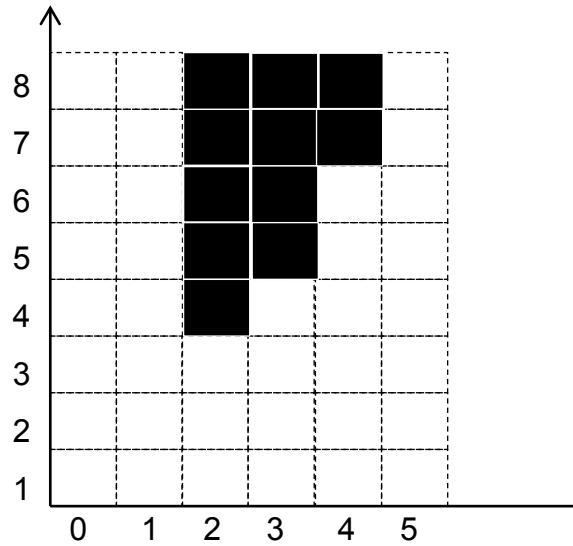
For each task, create a boolean variable which we set to true if and only if the task is assigned its original height.

Then search for the solution that maximise the sum of those booleans.

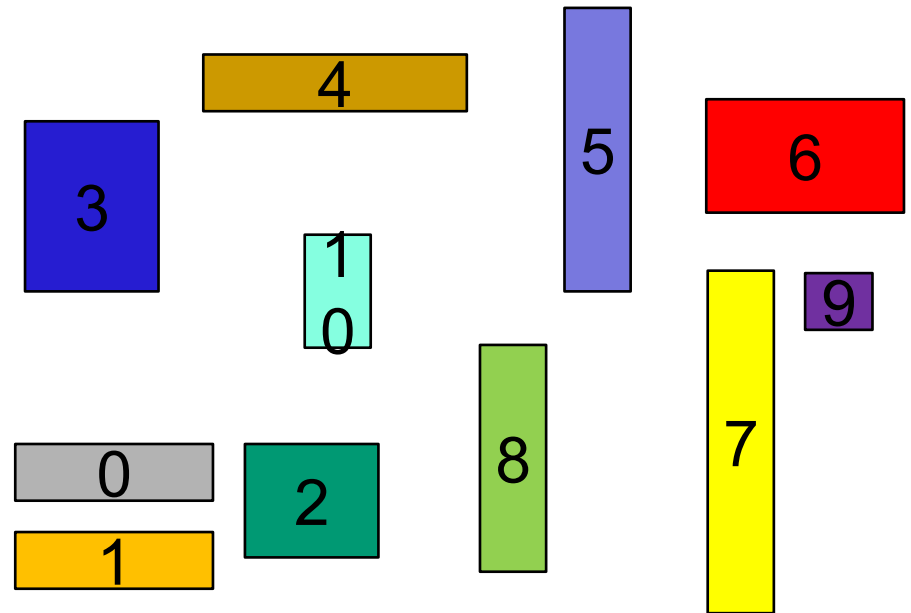
$$\text{assigned}[\text{task}] == 1 \iff \text{height}[\text{task}] == \text{heightdata}[\text{task}]$$

Maximising assigned tasks

consumption

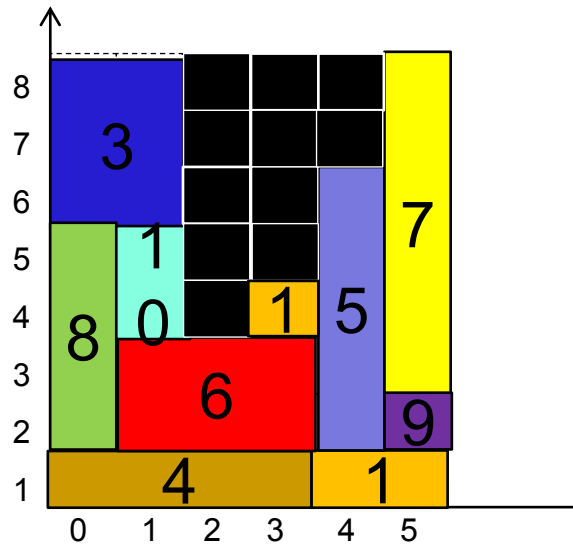


time



Maximising assigned tasks

consumption



0

2

time

Note

Allowing height to be a $\{0, h\}$ variable works if there are no other constraints on that task for this resource.

It is still scheduled, but since its height can be set to 0, it can overlap with any task without consuming resource.

If there were e.g. precedence constraints or time windows, there might not be a solution even if height = 0.

But we could use the height variable to allow tasks whose resource consumption changes depending on what time it is scheduled ...

Next lecture ...

more modelling

Acknowledgments: example taken from Choco 2 Tutorials