Scheduling

AIBT-April 2023 Guillaume Povéda, Florent Teichteil

Institut Supérieur de l'Aéronautique et de l'Espace

SUPAERO

Program of today

-Before 9h, Coffee.

-9-10h30 : This presentation.

- Some examples of where scheduling occur in real life
- Overview of methods to solve scheduling problem

-10h45-12h : First practical session, automatic basic solution to schedule your project (RCPSP)

-13h30-14h30 : Constraint programming introduction + first CP model

-15h-17h : Hands-on : coding a complex manufacturing scheduling problem model in CP

Intro : What is scheduling ? By ChatGPT

A scheduling problem refers to a class of problems in which a set of tasks needs to be completed within a certain timeframe, subject to various constraints and objectives. The goal of scheduling is to determine the order and timing of tasks in a way that optimizes some performance criteria.

Fields of application

1. Manufacturing: Scheduling is used to optimize production processes, minimize production time and costs, and ensure timely delivery of products.
2. Logistics: Scheduling is used to plan and manage transportation routes, optimize the use of resources such as vehicles and warehouses, and ensure timely delivery of goods.
3. Healthcare: Scheduling is used to manage patient appointments, schedule surgeries and procedures, and allocate staff and resources to ensure that patients receive high-quality care.
4. Education: Scheduling is used to manage class schedules, allocate teaching resources, and ensure that students have access to the courses they need to complete their degree requirements.
5. Project management: Scheduling is used to plan and manage the execution of projects, allocate resources, and ensure that project deadlines are met.
6. Sports: Scheduling is used to plan and manage sports events, allocate resources such as stadiums and equipment, and ensure that games and matches are played on schedule.
7. Information technology/high performance computing: Scheduling is used to optimize computer processes, allocate computing resources, and ensure that systems run smoothly and efficiently.
8. Fleet management/allocation: Scheduling is used to manage staff schedules, allocate resources such as vehicles and equipment, and ensure that customer needs are met in a timely and efficient manner.

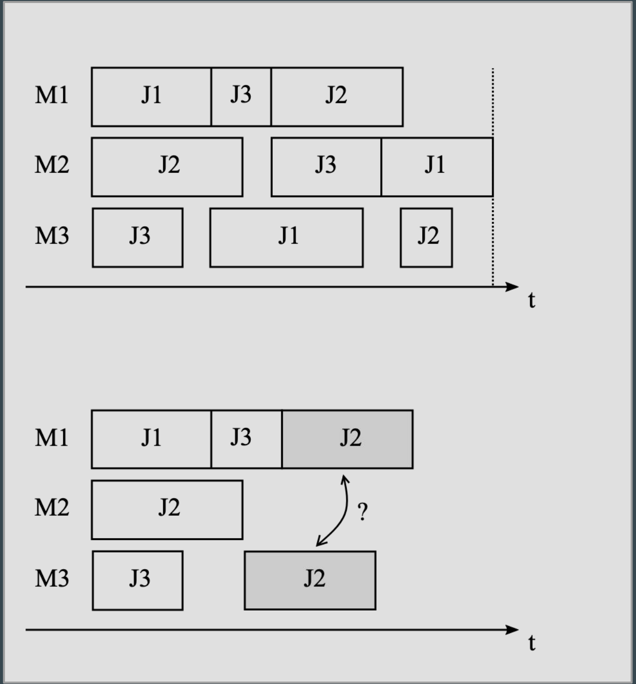
1. Manufacturing

Borreguerro, Portoleau et al. 2021]

Scheduling solvers can be a powerful tool for improving manufacturing processes by optimizing the allocation of resources, <u>reducing lead times</u>, increasing <u>productivity</u>, improving <u>quality</u>, enhancing <u>flexibility</u>, and minimizing <u>costs</u>.

In Airbus for example,
-Final assembly line task scheduling
-Optimisation of Sequencing of aircraft
-Optimisation of resource allocation to tasks
...

1. Manufacturing II Examples

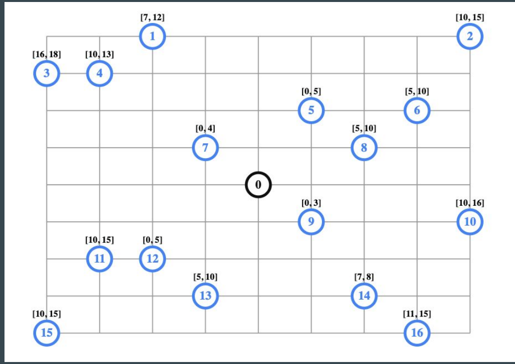


Source: localsolver tutos

<u>Job shop scheduling problem</u>
-Jobs are composed of several <u>tasks</u> that should be done in order.
-Each task are to be done in <u>**one**</u> given machine and has a processing time.
-Each jobs or task can have deadline or release dates.

<u>Flexible shop scheduling problem</u>
-Jobs are composed of several <u>tasks</u> that should be done in order.
-Each task are to be done in possibly <u>**seveval**</u> given machine and has a processing time.
-Each jobs or task can have deadline or release dates.

2. Logistics



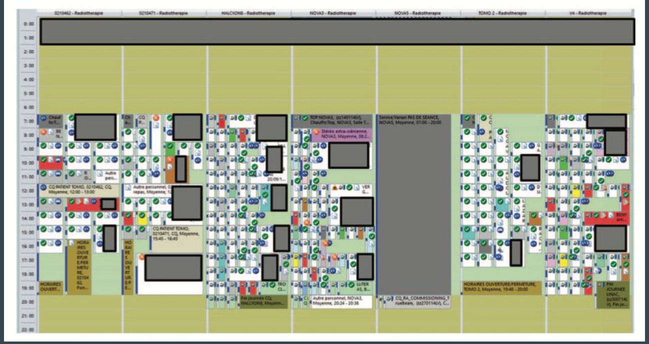
Vehicle routing problem appear in a lot of application : pickup and delivery problem, taxi scheduling, manufacturing robot planning, logistics in general.

Examples of very famous routing problems :

Traveling salesman problem (TSP) : compute the shortest path for 1 vehicle visiting all the clients
Vehicle routing problem (VRP) : considering a fleet of N vehicle, dispatch them to visit all client, optimize for example the longest path among the vehicles
Capacitated Vehicle routing problem (CVRP) : Same as VRP, considering capacity for each vehicle, constraining the number of trips it can do. (example : fuel capacity)
(C)VRPTW : Same as CVRP, including time window constraint in visiting the clients.

Possible “aparté”
https://developers.google.com/optimization/routing

3. Health



	monday	tuesday	wednesday	thursday	friday
slot 1	Patient 1	Patient 6	Patient 1	Patient 6	Patient 1
slot 2	Patient 2	Patient 2	Patient 2	Patient 2	Patient 2
slot 3					
...	...	...	...	...	...
slot s	Patient 3	Patient 1	Patient 3		Patient 3
slot $s+1$				Patient 1	
...	...	...	...	...	...
slot $\lvert\mathcal{S}\rvert-2$	Patient 4	Patient 4	Patient 4	Patient 5	Patient 5
slot $\lvert\mathcal{S}\rvert-1$				Patient 4	Patient 4
slot $\lvert\mathcal{S}\rvert$	Patient 5	Patient 5	Patient 5		

<u>Scheduling in health domain :</u>

Scheduling a treatment, typically a cancer is a highly constrained scheduling problem where the objective function to maximize is the efficiency of the resulting treatment.

<u>**Examples :**</u>

<u>Nurse scheduling problem :</u>

One of the classic scheduling problem is named nurse scheduling problem, it consists in assigning nurses to shifts of works in the week, following some workload constraints.

<u>Radiotherapy planning</u> :

Considering a queue of patient, each having a required treatment (type of machine, # of sessions, agenda preference) : allocate patient to agenda slots/machines.

<u>(Paper :</u>

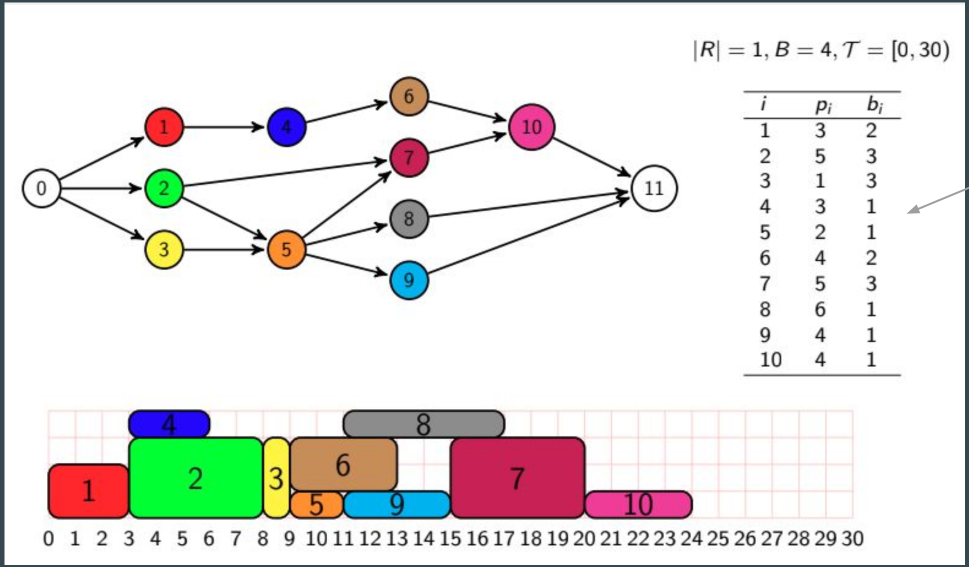
<u>https://link.springer.com/article/10.1007/s10729-020-09510-8)</u>

4. Education

2023				2024								
Septembre	Octobre	Novembre	Décembre	Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre
1	1	1	1	1 1	1	1 AIBT111	1	1	1	1 27	1	1
2	2 40	2	2	2	2 AIBT111	2	2 14	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3 23	3	3	3
4 36	4	4	4 49	4	4	4 10	4	4	4	4	4	4
5	5	5	5	5 AIBT106	5 6	5	5	5	5	5	5 32	5
6	6	6 45	6	6	6	6	6	6 19	6	6	6	6 Soutenances
7	7	7	7	7	7	7	7	7	7	7	7	7
8	8	8	8	8 2	8	8	8 15	8	8	8 28	8	8
9	9 41	9	9	9 AIBT106	9	9	9	9	9	9	9	9
10	10 AIBT200	10	10	10 ISAE	10	10	10	10	10 24	10	10	10
11 37	11 + python	11	11 50	11	11	11 11	11	11	11	11	11	11
12	12 ISAE	12	12 AIBT105	12 AIBT111	12 7	12 AIBT200	12	12	12	12	12 33	12
13	13	13 46	13 ISAE	13	13	13 + entretiens	13	13 20	13	13	13	13
14	14	14	14	14	14	14 ISAE	14	14	14	14	14	14
15	15	15	15 AIBT111	15 3	15	15	15 16	15	15	15 29	15	15
16	16 42	16	16	16	16	16	16	16	16	16	16	16
17	17	17 AIBT102	17	17	17	17	17	17	17 25	17	17	17
18 38	18	18	18 51	18	18	18 12	18	18	18	18	18	18
19	19	19	19	19	19 8	19	19	19	19	19	19 34	19
20	20	20 47	20	20	20	20	20	20	20 Hackathon	20	20	20
21	21	21 AIBT102	21	21	21	21	21	21 21	21 Hackathon	21	21	21
22	22	22 ISAE	22	22 4	22	22	22 17	22	22	22 30	22	22
23	23 43	23	23	23	23 AIBT103	23	23 AIBT108	23	23	23	23	23
24	24 AIBT101	24 AIBT111	24	24	24	24	24 ISAE	24	24 26	24	24	24
25 39	25 ISAE	25	25 52	25	25	25 13	25	25	25 AIBT110	25	25	25
26 AIBT200	26	26	26	26	26 9	26 AIBT107	26 AIBT111	26	26 ISAE	26	26 35	26
27 + python	27 AIBT111	27 48	27	27	27 AIBT103	27 TBS	27	27 22	27	27	27	27
28 ISAE	28	28	28	28	28 ISAE	28	28	28 AIBT109	28 AIBT111	28	28	28
29	29	29	29	29 5	29	29 AIBT111	29 18	29 ISAE	29	29 31	29	29
30	30 44	30	30	30 AIBT104		30	30	30	30	30	30	30
	31		31	31 TBS		31		31 AIBT111		31	31	

Scheduling an whole year of course agenda, taking into account duration of courses, precedence constraints, clashes, bank holiday etc, is a very common example of scheduling that we all met in our student life.

5. Project management



i	p_i	b_i
1	3	2
2	5	3
3	1	3
4	3	1
5	2	1
6	4	2
7	5	3
8	6	1
9	4	1
10	4	1

Projects are organized in subtasks with precedence, intermediary milestones, deadlines etc.

Example :

RCPSP : Resource constraint project scheduling problem

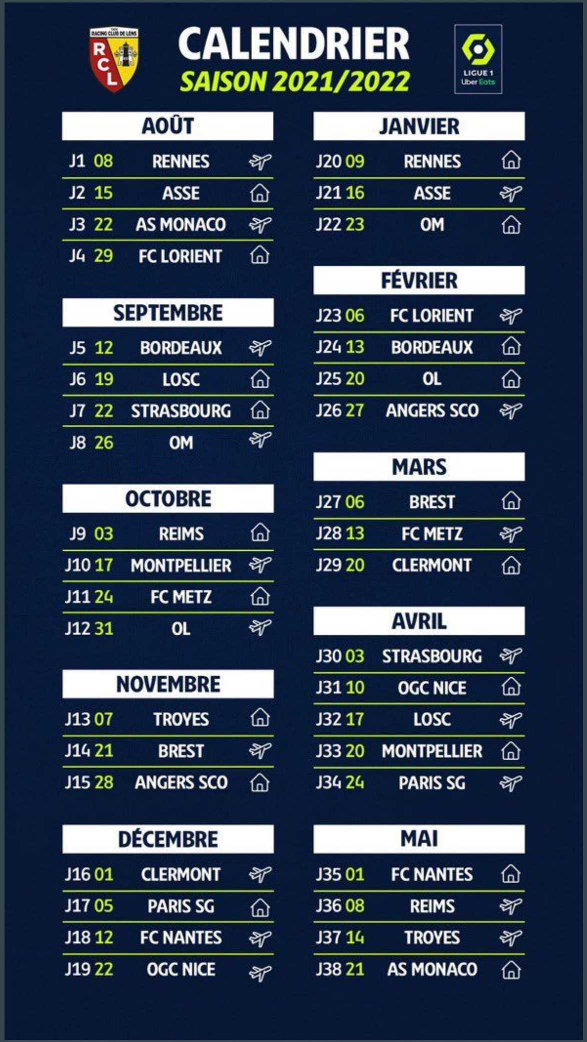
Schedule task fulfilling precedence constraints, cumulative resource consumption constraints.

Sport scheduling

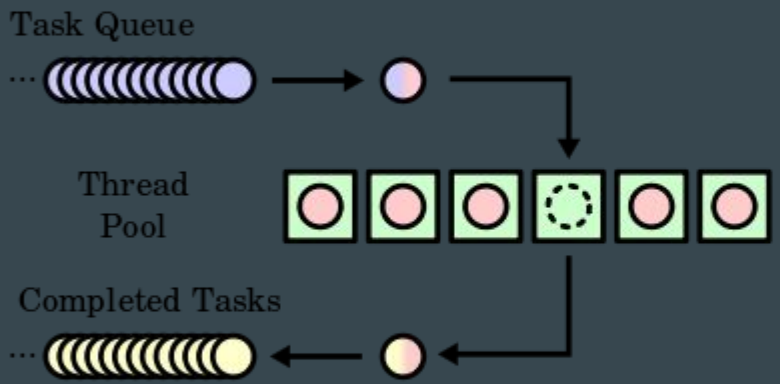
Organize tournaments is a highly constrained scheduling problem :
Constraint can be for eg :

- no more than 2 games in a row at home
- Meet all the team in the first half of the season, then re-meet them in the second half but in the other field.

etc.



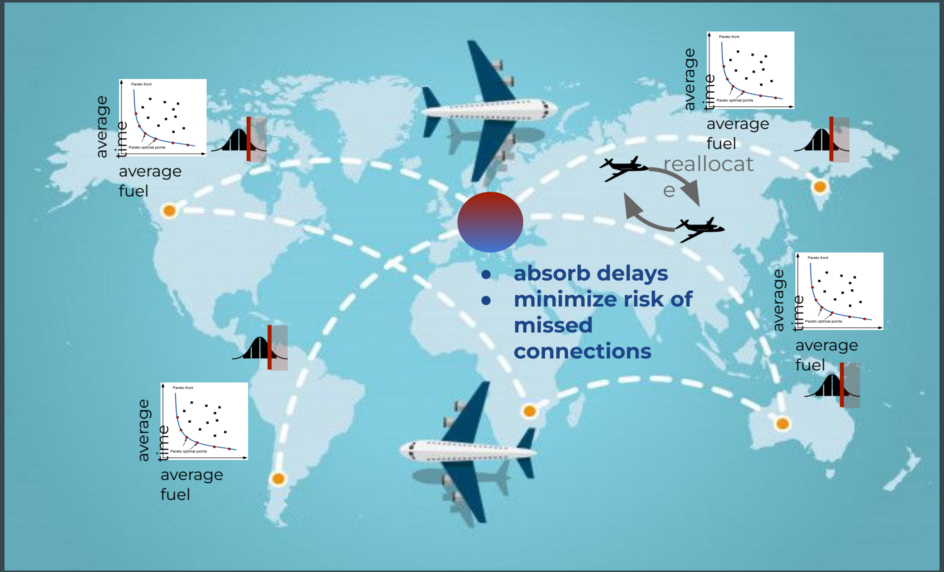
Computer scheduling



Scheduler are present in every OS to orchestrate the order of execution of different computation/task.
It is based on priority rules or more advanced concept

In the HPC (high performance computing) world, with many workers/process/thread, schedulers can play an important part in the efficiency of the overall computation.

Fleet management/allocation

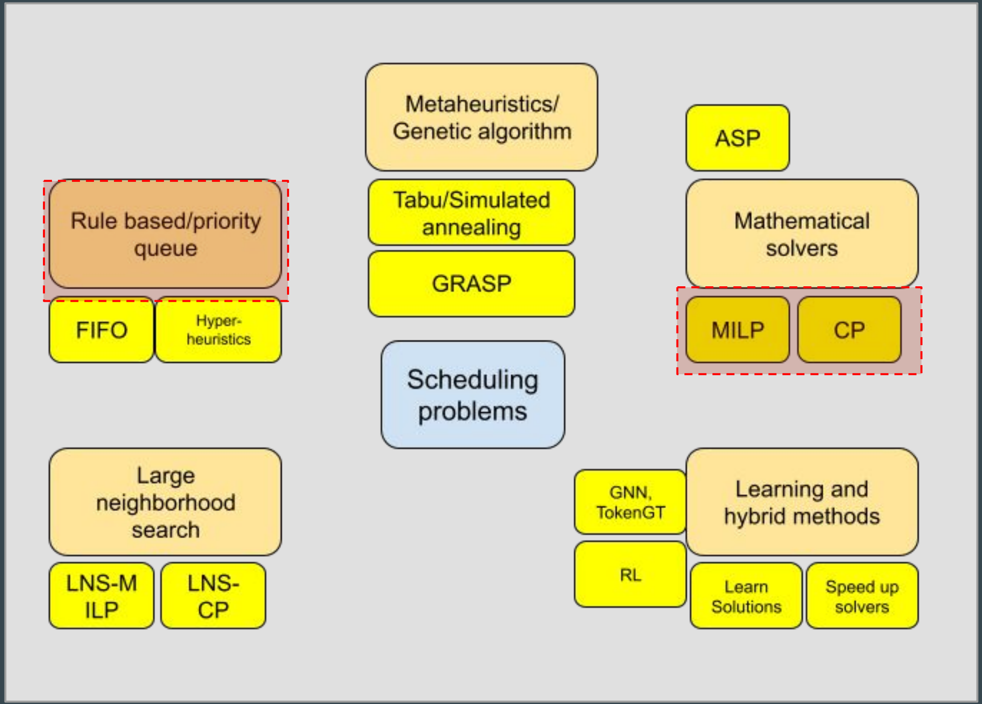


Quite overlapping with what we described in slide Slide 7: 2. Logistics, fleet scheduling is very important for airlines.

Possible subproblems are :
- Build a time table of flights in the world maximising profit + satisfying demand
- allocating individual MSN (=aircraft id) to each of the flights.
- fleet sizing : choose the right amount of aircraft and their type

Solving Scheduling problems

Solving methods overview



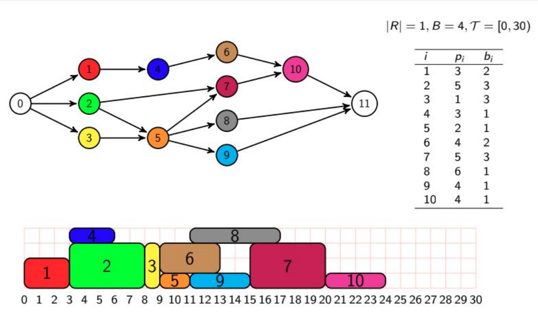
To solve scheduling problems, a lot of methods have been studied in the OR (Operational Research) community, aiming at improving the decision making of various organisation, as listed in the previous slides !

In this AIBT course, we will give an introduction to mathematical solvers and also code some basic rule based solvers.

Mathematical formulation of scheduling problem: RCPSP example

RCPSP problem in english :

- Scheduling problem with standard “finish-start” precedence constraints and resources of limited availabilities.
- Goal : find the start time of tasks while satisfying precedence and resource constraints.
- Objective function : Minimize the makespan (total project duration). Other ones can exist of course.

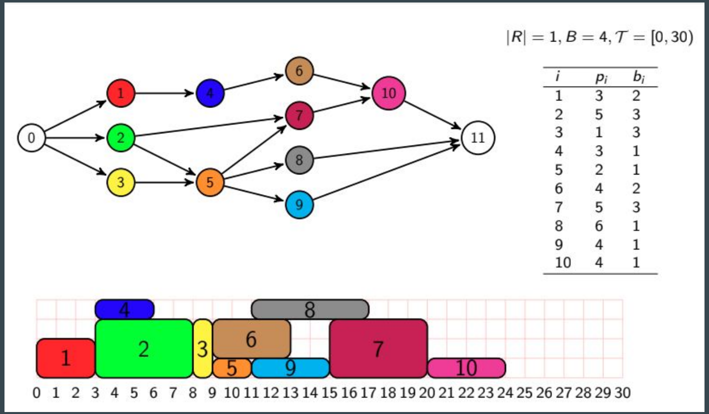


i	p_i	b_i
1	3	2
2	5	3
3	1	3
4	3	1
5	2	1
6	4	2
7	5	3
8	6	1
9	4	1
10	4	1

Mathematical formulation of scheduling problem: RCPSP example

Input data of RCPSP's :

- R set of resources, limited constant availability $B_k \geq 0$,
- A set of activities, duration $p_i \geq 0$, resource requirement $b_{ik} \geq 0$ on each resource k,
- E set of precedence constraints (i, j), i, j $\in$ A, i < j
- T time interval (scheduling horizon)



RCPSP : Variables and constraints

<u>Variables</u>

- $S_i >= 0$, starting time of activity i
- C_{max} the makespan of the project

<u>Constraints</u>

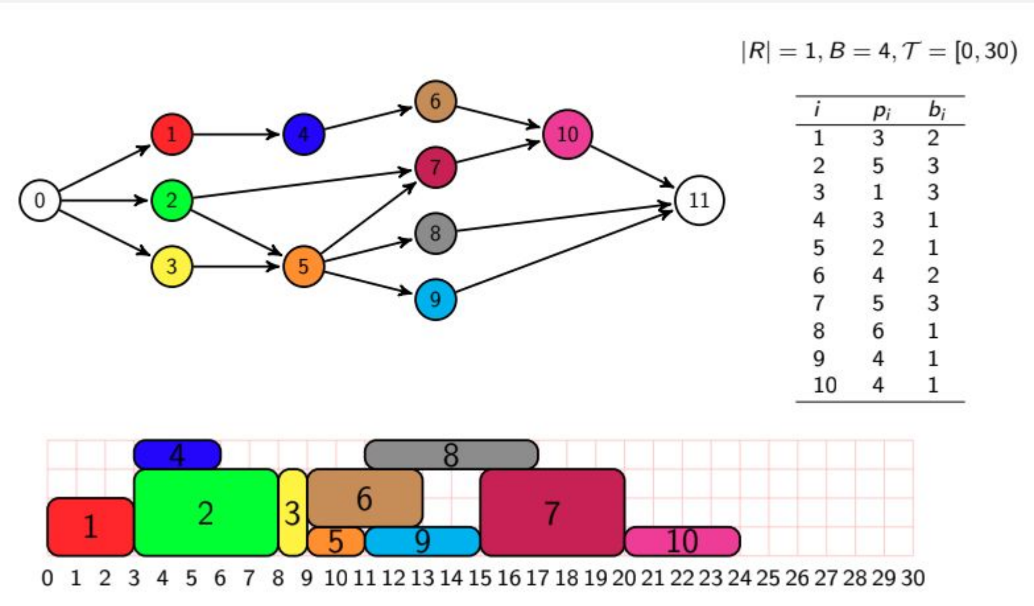
Precedence constraints :

$\forall (i,j) \in E,\ S_j \geq S_i + p_i$

<u>Resource constraints</u>

$\forall t \in [0,T],\ \sum_{i \in A(t)} b_{ik} \leq B_k$

Where $A(t) = \{j \in A | t \in [Sj, Sj + pj)\}$ is the set of activity that are active at time “t”



Greedy procedure to compute a schedule : SGS

<u>Procedure for scheduling :</u> List of Task -> Feasible schedule.

One such procedure for RCPSP is called SGS for serial generation scheme.
What it does is to schedule the task as soon as possible following the order of priority.

For some problems :

Optimal permutation found using scheduling procedure $\Longleftrightarrow$ Optimal schedule

Example SGS for a jobshop (Try it by yourself)

Example instance :
Job 1 = [(J11, M1, 2), (J12, M3, 3), (J13, M2, 2)]
Job 2 = [(J21, M2, 3), (J22, M1, 2), (J23, M3, 1)]
Job 3 = [(J31, M3, 3), (J32, M1, 3), (J33, M2, 2)]

TASK PRIORITY :
[J11, J31, J12, J32, J21, J13, J22, J23, J33]

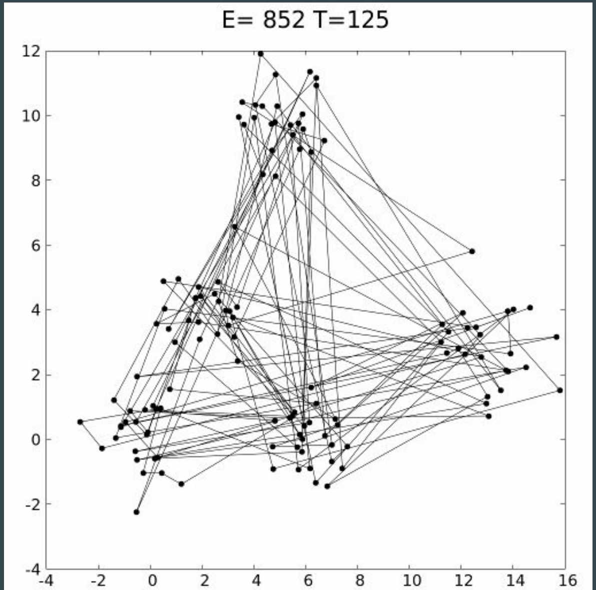
Example SGS for a jobshop

Example instance :
Job 1 = [(J11, M1, 2), (J12, M3, 3), (J13, M2, 2)]
Job 2 = [(J21, M2, 3), (J22, M1, 2), (J23, M3, 1)]
Job 3 = [(J31, M3, 3), (J32, M1, 3), (J33, M2, 2)]

TASK PRIORITY :
[J11, J12, J31, J32, J21, J13, J22, J23, J33]

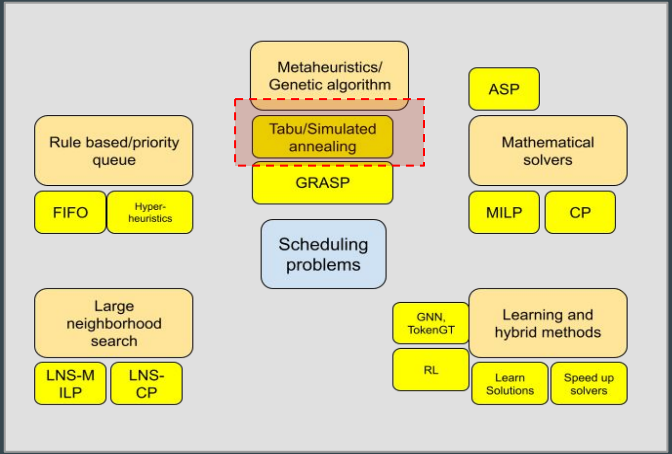
1) Metaheuristics for scheduling

Optimisation problem that can be used for permutation can therefore be used for scheduling !



<u>GreedyLocalSearch()</u>

```
current_solution, fitness_value
For i in 1:N:
    solution = neighbor(current_solution)
    Fit = fitness(solution)
    If Fit>fitness_value:
        current_solution =Fit
        fitness_value = Fit
EndFor
Return current_solution
```



2) Mixed Integer linear programming

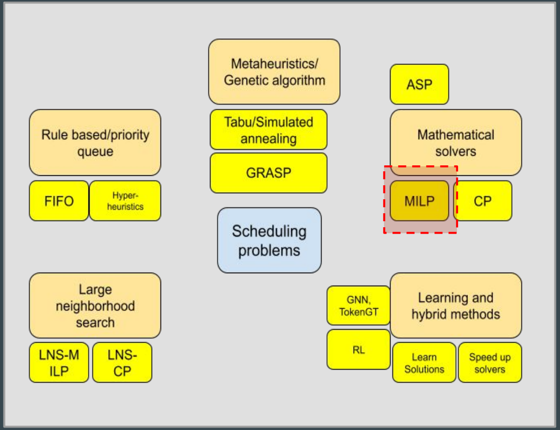
$$min_x/max_x z = c^T x \qquad A \in \mathbb{R}^{(p,n)}$$

$$\text{s.t} \quad Ax \leq b \qquad c \in \mathbb{R}^n$$

$$x \in \mathbb{R}^n_+ \qquad b \in \mathbb{R}^p$$

Implementing MILP models for scheduling problem is possible.

It is usually done using **time-indexed** variables representing the starting time of the task



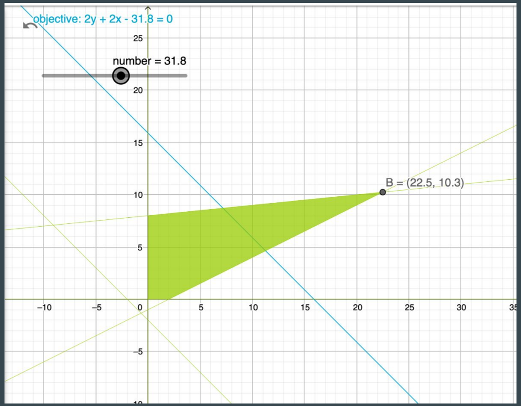
Example and geometric interpretation : Linear programming

$$max_{x,y} 2x + 2y$$

$$\frac{x}{2} - y \leq 1$$

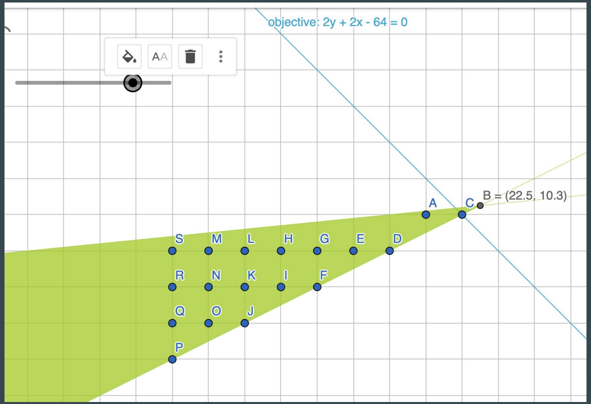
$$-\frac{x}{10} + y \leq 8$$

$$-x - y \leq 2$$



The problem without integer variable can be solved easily by the simplex or interior-point method.

But solving on integer values can be exponentially harder :
Algorithms :
-Branch and bound
-Cut generation



Mixed Integer linear programming for scheduling

Variables :

$x_{it} \in \{0,1\}$ indicating that the task ‘i’ starts at time ‘t’

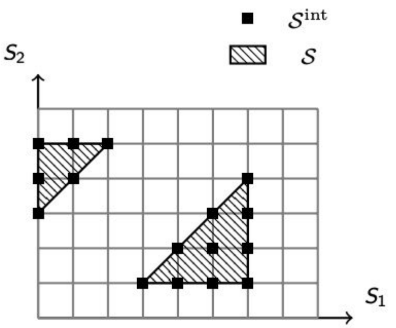
[Optional] $S_i \in [0, \text{horizon}]$, the starting time of the task ‘i’

Constraints :

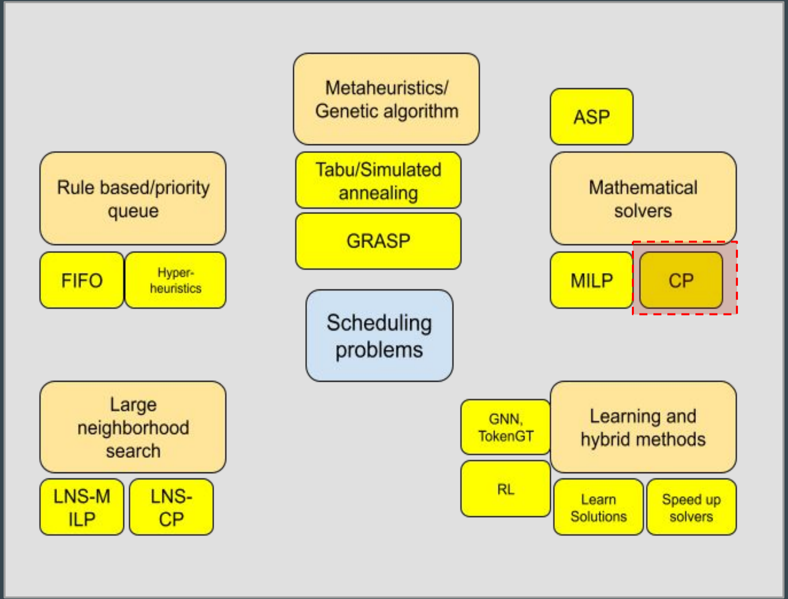
- Convention constraints for the time-indexed :
  - $\forall i \in \text{tasks}, \sum_t x_{it} = 1$
  - $\forall i \in \text{tasks}, \sum_t x_{it} . t = S_i$
- Precedence constraints :

  $\forall (i,j) \in E, S_j \geq S_i + p_i$
- Resource constraints :
  - $\sum_{i \in \text{tasks}} b_{ik} . \sum_{t' \in [t-pi+1,t]} x_{it'}$

$$
\begin{aligned}
S_1 &= x_{1,1} + 2x_{1,2} + 3x_{1,3} + 4x_{1,4} + 5x_{1,5} + 6x_{1,6} \\
S_2 &= x_{2,1} + 2x_{2,2} + 3x_{2,3} + 4x_{2,4} + 5x_{2,5} \\
x_{1,0} + x_{1,1} + x_{1,2} + x_{1,3} + x_{1,4} + x_{1,5} + x_{1,6} &= 1 \\
x_{2,1} + x_{2,2} + x_{2,3} + x_{2,4} + x_{2,5} &= 1 \\
x_{1,0} + x_{1,1} + x_{2,1} &\leq 1 \\
x_{2,1} + x_{2,2} + x_{1,0} + x_{1,1} + x_{1,2} &\leq 1 \\
x_{2,2} + x_{2,3} + x_{1,1} + x_{1,2} + x_{1,3} &\leq 1 \\
x_{2,3} + x_{2,4} + x_{1,2} + x_{1,3} + x_{1,4} &\leq 1 \\
x_{2,4} + x_{2,5} + x_{1,3} + x_{1,4} + x_{1,5} &\leq 1 \\
x_{2,5} + x_{1,4} + x_{1,5} + x_{1,6} &\leq 1 \\
x_{1,t} \in \{0,1\} &\quad t \in \{0,\ldots,6\} \\
x_{2,t} \in \{0,1\} &\quad t \in \{1,\ldots,5\}
\end{aligned}
$$



Constraint Programming



From your course of Optimisation (link).

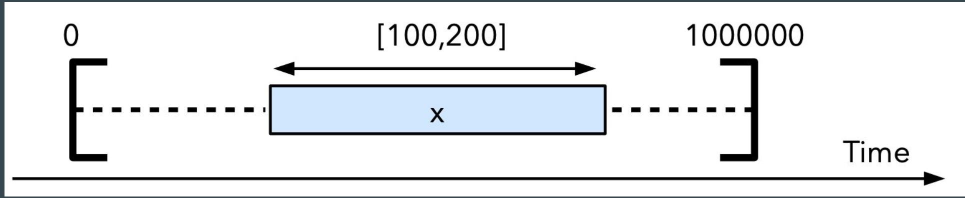
A constraint satisfaction problem consists of:

-a finite set of variables x1,...xn
-a corresponding set of finite domains d1,...dn
-a finite set of constraints over subsets of variables, i.e. relations expressing which assignations of variables to values are acceptable.

Constraint Programming for scheduling : Interval variables

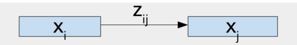
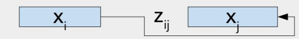
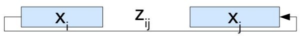
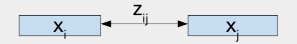
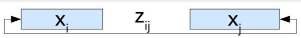
Scheduling is about **Time.**
Start of the art solvers are using adapted abstraction to model scheduling problem. This is the case of Ortools-CPSAT solver and CP-Optimizer.

dvar interval x in 0..1000000 size 100..200;

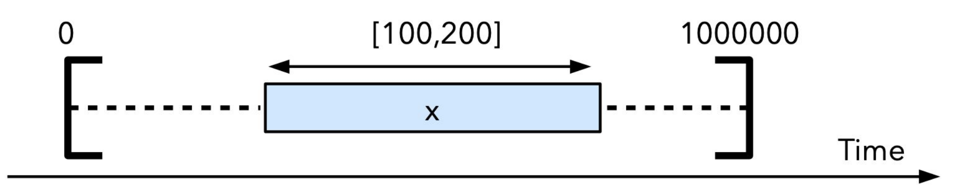


https://icaps17.icaps-conference.org/tutorials/T3-Introduction-to-CP-Optimizer-for-Scheduling.pdf
P. Laborie, J. Rogerie. Reasoning with Conditional TimeIntervals. Proc. FLAIRS-2008, p555-560.

Constraint Programming for scheduling : Constraints on interval (#1)

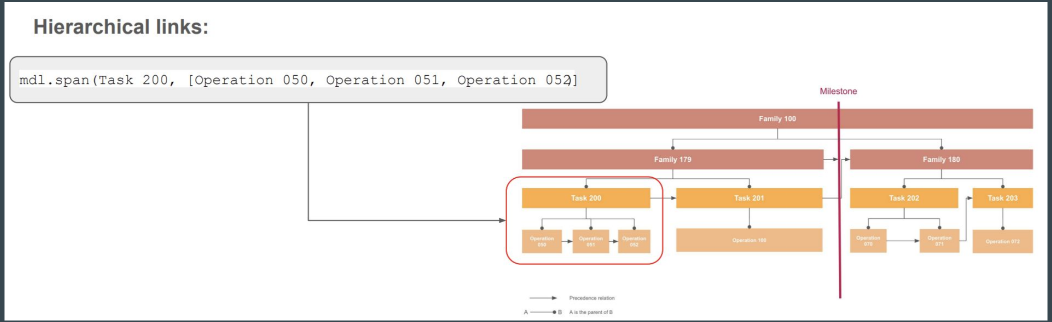
Constraint name	Semantics $(x_i \neq \bot) \wedge (x_j \neq \bot) \Rightarrow$	Pictogram
endBeforeStart	$e(x_i) + z_{ij} \leq s(x_j)$	
startBeforeStart	$s(x_i) + z_{ij} \leq s(x_j)$	
endBeforeEnd	$e(x_i) + z_{ij} \leq e(x_j)$	
startBeforeEnd	$s(x_i) + z_{ij} \leq e(x_j)$	
endAtStart	$e(x_i) + z_{ij} = s(x_j)$	
startAtStart	$s(x_i) + z_{ij} = s(x_j)$	
endAtEnd	$e(x_i) + z_{ij} = e(x_j)$	
startAtEnd	$s(x_i) + z_{ij} = e(x_j)$	

Modern CP solver focused on scheduling capture the essence of scheduling problems, notably to specify precedence constraints.

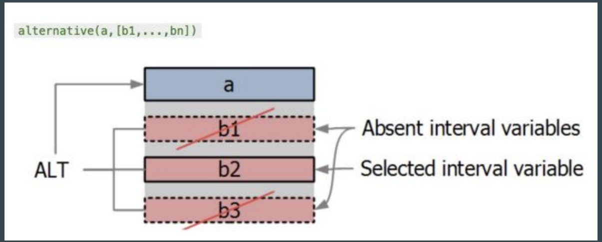


Constraint Programming for scheduling : Constraints on interval (#2)

Other important powerness of expressivity :



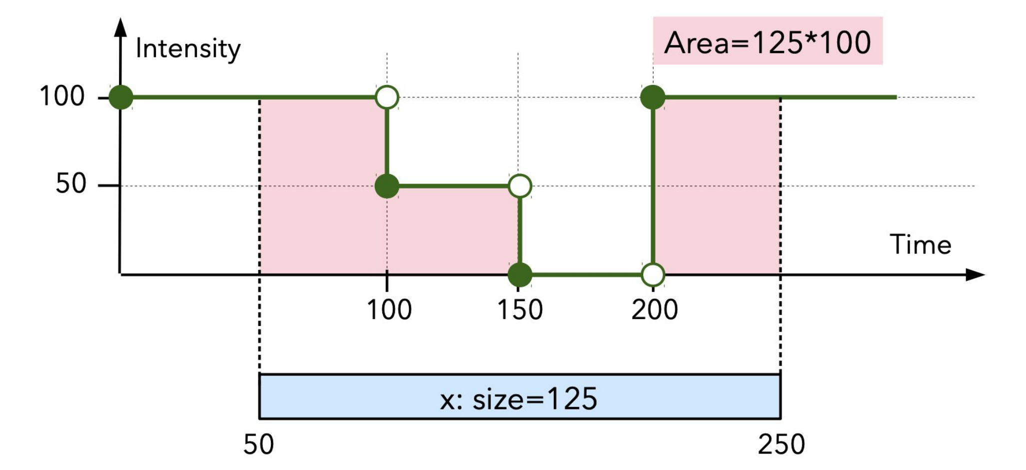
<u>Hierarchical project scheduling =</u>
concept of meta tasks



<u>Optional intervals =</u>
Used to allow different ways of executing a task, for example.

Constraint Programming for scheduling : Constraints on interval (#2)

Other important powerness of expressivity :



Intensity function :
If a task need some quantity of resource, but this resource is not available at each time (i.e calendar, breaks). You can define a task with an intensity function (a step function between 0 and 100).

The length of the task can therefore be longer than its duration

Learning to schedule :

- Reinforcement learning to schedule job-shop
  https://github.com/jolibrain/wheatley
  https://github.com/instadeepai/jumanji
- Learn how to solve combinatorial optimisation problem with GNN :

BONUS SLIDES

BONUS : Graph Neural Network

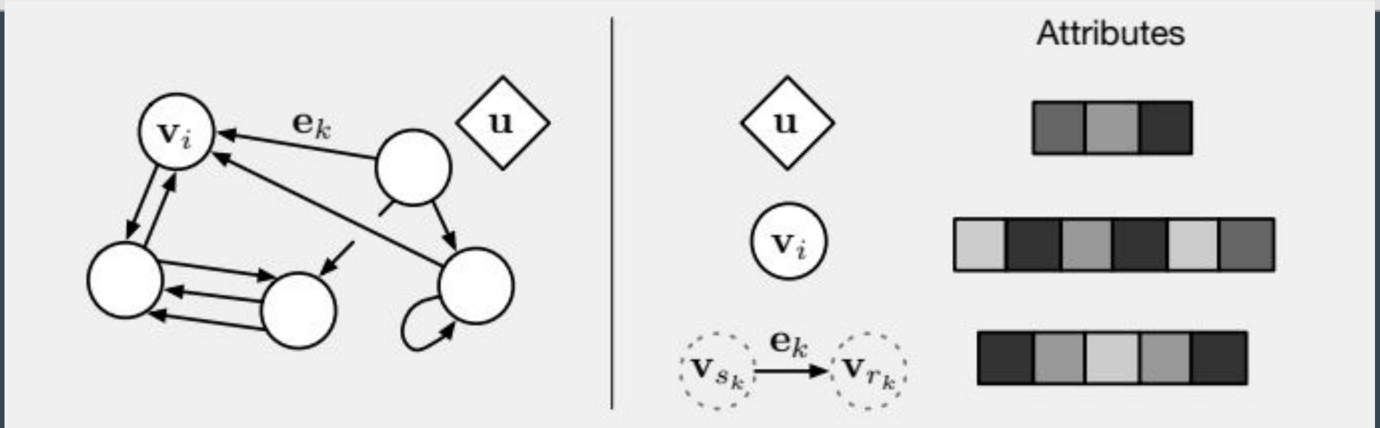
Definition of a graph

$$G = (u, V, E)$$
$$E = \{(e_k, r_k, s_k)\}_{k=1:N^e} \quad with \quad N^e = |E|$$
$$V = \{v_i\}_{i=1:N^v} \quad with \quad N^v = |V|$$

$"u"$ can be seen as a global parameter e.g. the type of a problem or the gravitational field.

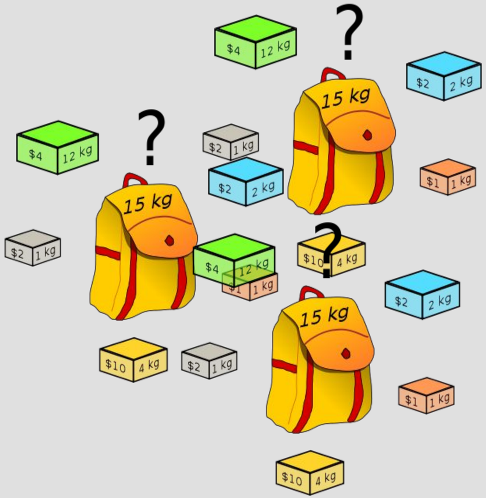
$"r_k"$ Receiver only integers

$"s_k"$ Sender “”

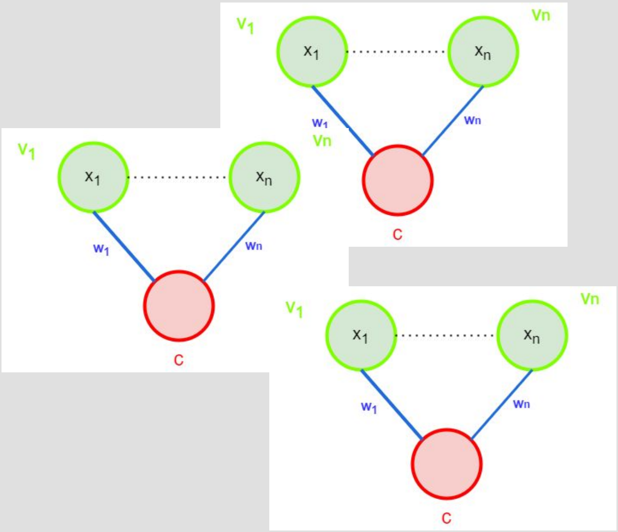


BONUS : Supervised approach to solve the knapsack problem with GNN

Supervised approach : The model is trained to solve MILP problems using knapsack problems as training data.



Knapsack problems



Training X (graphs).

$$Y_{knapsack1} = \begin{bmatrix} 1 \\ \vdots \\ 0 \end{bmatrix}$$

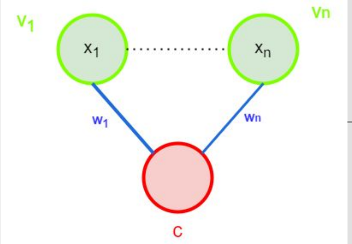
$$Y_{knapsack2} = \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix}$$

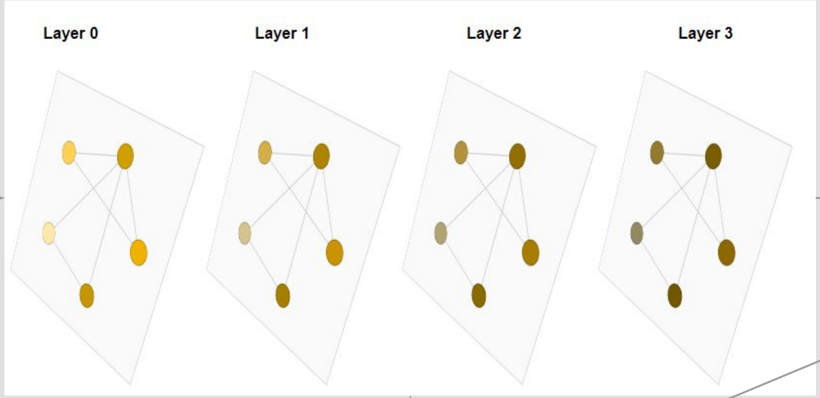
$$Y_{knapsack3} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

Training Y (solutions of different knapsack problems).

Supervised approach to solve the knapsack problem with GNN

GNN





Output

$$Y'_{knapsack_{train}} = \begin{bmatrix} 0.8 \\ \vdots \\ 0.1 \end{bmatrix}$$

Knapsack test

$$Y_{knapsack_{train}} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

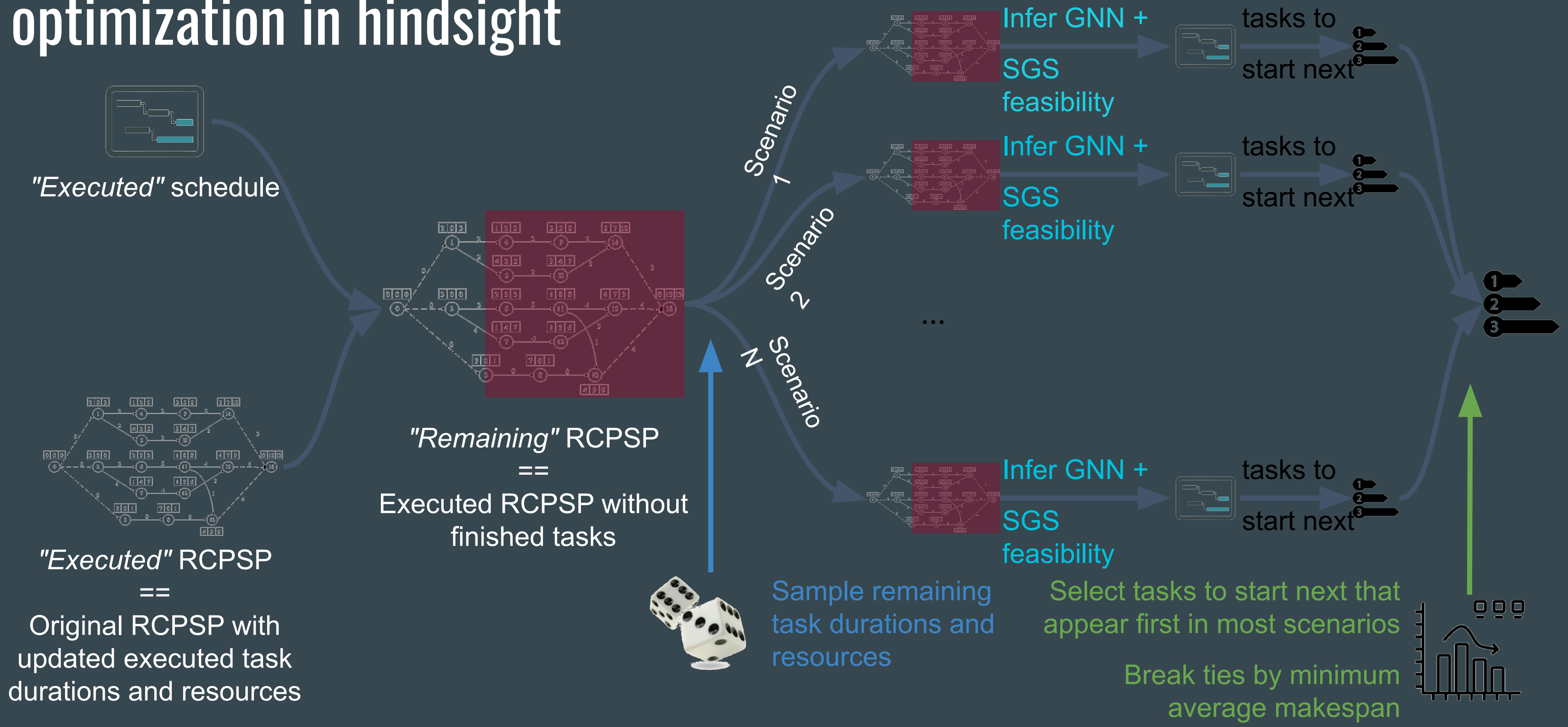
Solution given by a solver

Backpropagation : loss result is used for updating the weights.

$$L(Y_{knapsack_{train}}, Y'_{knapsack_{train}}) = -\sum_{i=1}^{n} Y_{knapsack_{train}} \log(Y'_{knapsack_{train}})$$

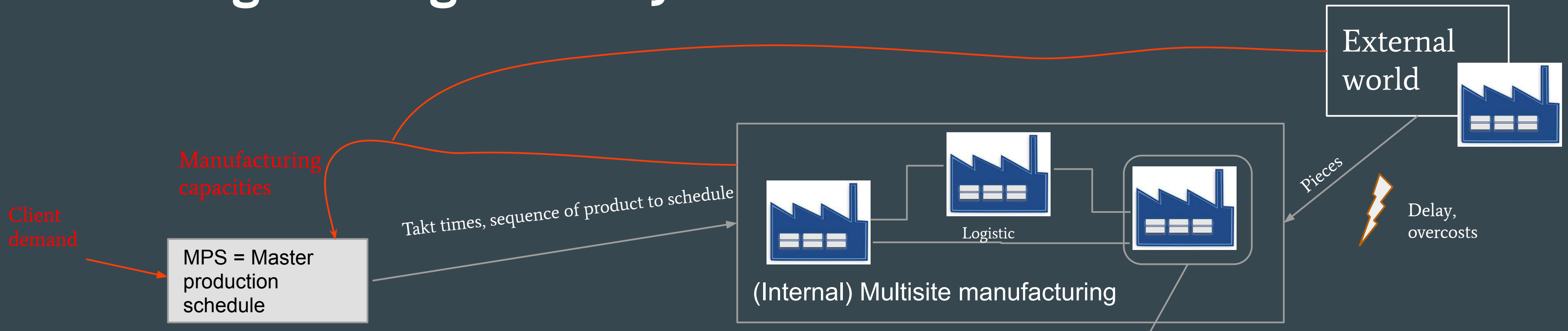
Cross-entropy (how two distributions are similar) used for Node classification.

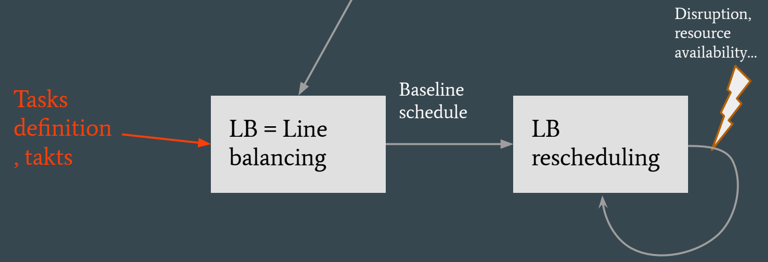
BONUS : Next tasks online selection using GNN inference and optimization in hindsight



Wrapping up

Scheduling in a big industry :





A master production schedule (MPS) chooses what products a manufacturer will produce, when and in what quantities. An MPS links sales demand with manufacturing capacity. The purpose of master production scheduling is to create a realistic plan that minimizes overstock while maximizing on-time delivery.

The line balancing : schedule a set of tasks to accomplish the production of a piece, or assembly of different pieces.

Remaining of the day :

3 notebooks :

- Morning : Scheduling Notebook #1 :
    - Goal : discover RCPSP
    - Exercise : Implement the SGS procedure for RCPSP problem
- Afternoon : Scheduling Notebook #2 Constraint programming
    - Goal : discover Ortool-Cpsat library, be able to state variable and constraint of a simple scheduling problem
    - Exercise : Implement CP model for Job shop scheduling problem
- Handson (2h):
    - Exercise : CP model for Resource constraint project scheduling problem.
    - More advanced models to code if time allows.

Discrete optimization lib

Problem definition

- Decision variable
- Evaluation function
- Constraints

Solve()

Solvers

- ***Greedy solvers*** (Rule based, priority queue)
- ***Metaheuristics*** (local search, genetic)
- ***Mixed Integer Linear Programming***
- ***Constraint Programming***
- ***Hybrid method*** for eg. Large neighborhood search +CP

Thanks