What I know

Flexible job-shop scheduling problem (FJSP) 🡪 JxM problem of N jobs and M machines.

Requirements :

1. Each job has nj operations, {Oj,1,Oj,2,…,Oj,nj}, operation Oj,i+1 must be started after Oj,i is completed.
2. Each operation has to be processed by one of machines
3. Pjik is the processing time of Oj,i on machine k
4. Cj,i is the completion time of Oj,i
5. Ok is the set of operations that machine k must process
6. Jobs are independent
7. One machine can only process one operation at a time
8. Operations cannot be interrupted

Major objective criteria :

* Makespan :
* Total workload :
* Maximum workload :

We could use other type of rewards such as :

1. Date of completion
   1. greatest delay
   2. sum of delays and advances (tasks completed ahead of schedule)
   3. total number of tardy jobs.
2. Travel time
   1. (task) travel time
   2. average travel time
   3. maximum travel time
3. System usage
   1. machine usage
   2. system usage, i.e., average utilization of each machine.

Those criteria can change, we can add the leadtime as well

Implementation exemple : [Notebook (jckantor.github.io)](https://jckantor.github.io/ND-Pyomo-Cookbook/04.03-Job-Shop-Scheduling.html) , [Modeling Examples — Python-MIP documentation](https://python-mip.readthedocs.io/en/latest/examples.html)

Simplification pour commencer :

1. Il y a une seule machine possible par job, pas 2
2. Les ressources humaines sont illimitées
3. Les jobs durent soit 0.5 ou 1 jour
4. Les jobs ne peuvent que réussir.
5. La création du tween n’est pas comprise dans le planning
6. On commence avec les lots 3.75 et 11.25

# Task decomposition

En partant d’un lot X, décomposer le lot en nj operations :

Une image contenant table

Description générée automatiquement

Convert into JSON :

Une image contenant texte, personne, capture d’écran

Description générée automatiquement

# Problème : comment mettre en place un State qui soit consistant

Idée : [Microsoft Word - 2020 04 10 - Proceedings - Gesamtdokument - Hd - v12 (uni-hannover.de)](https://www.repo.uni-hannover.de/bitstream/handle/123456789/9732/Rinciog%202020%20-%20Sheet-Metal%20Production%20Scheduling%20Using%20AlphaGo%20-%20CPSL2020.pdf?sequence=1&isAllowed=y)

[Hybrid Deep Neural Network Scheduler for Job-Shop Problem Based on Convolution Two-Dimensional Transformation (nih.gov)](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6652087/)

[Learning to Dispatch for Job Shop Scheduling via Deep Reinforcement Learning | Papers With Code](https://paperswithcode.com/paper/learning-to-dispatch-for-job-shop-scheduling)

[2104.03760.pdf (arxiv.org)](https://arxiv.org/pdf/2104.03760.pdf) with code [prosysscience/Job-Shop-Scheduling (github.com)](https://github.com/prosysscience/Job-Shop-Scheduling)

[IEEE Xplore Full-Text PDF:](https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9529186)

[Microsoft Word - BQF7FQROFS (iop.org)](https://iopscience.iop.org/article/10.1088/1742-6596/1848/1/012029/pdf) 🡪 single agent DJSSP

uestions Bruno :

* Quelles sont les règles de la chaine ?
  + Tamisage uG, attente libération QC
  + Tamisage polymère 🡪 30 jours
  + Entre mélange et extrusion 🡪 24h max
  + Millieu 🡪 7 jours pour l’utiliser
  + Sortie lyo 🡪 capsulage 🡪 5 jours max
  + 7 semaines entre bouchonnage lyo à irradation
* Mélangeur et combinaison fractions, même machine ?
  + Combinaison fractions 🡪 sans machine, simplement mélanger et utiliser l’homogénisatur
  + Mélanger 🡪 mélangeur à bille
* Le temps des étapes, soit une demi journée soit une journée ? vraiment ?
  + 02-009474-75-76
  + Réserver la journée ou la demi-journée
* Tamisage Ug 🡪 6600 peut être fait en 4h, si 20000 journée entière
* Si une étape échoue
  + Broyage poly 🡪 on refait si problème
  + Broyage Ug 🡪 on récupère au max, si problème en finit l’étape (le lendemain par exemple)
  + Les autres étapes 🡪 ca dépend
* Broyage poly et uG 🡪 2 personnes
* Tamisage 🡪 2 personnes
* Mélanges 🡪 3 personnes
* Combinaison 🡪 2 personnes
* Millieu 🡪 2 perosnnes
* Extrusion 🡪 2 personnes
* Lyo(perry, mise en supspension, débounchonnage et mise en lyo) 🡪 8 personnes
* Déchargement lyo 🡪 1 personne si petit échel et 2 pour une grande
* Capsulage 🡪 2 personnes
* IV 🡪 il faut compter 5 personnes pendant 2 jours pour une grande échelle. Si petit échel 4 personnes pendant une journée
* Envoi et retour IR, actuellement souvent le mercredi mais c’est flexible 🡪 regarder avec patricia
* On a une force de travail assez constante

# Reinforcement learning basic

Une image contenant table

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

[Reinforcement Learning Explained Visually (Part 5): Deep Q Networks, step-by-step | by Ketan Doshi | Towards Data Science](https://towardsdatascience.com/reinforcement-learning-explained-visually-part-5-deep-q-networks-step-by-step-5a5317197f4b)

[Tensorforce: a TensorFlow library for applied reinforcement learning — Tensorforce 0.6.5 documentation](https://tensorforce.readthedocs.io/en/latest/index.html)

# Digital tweens

A lire pour le digital tweens : [Production Planning and Resource Management of Manufacturing Systems in Python | by Will Keefe | Oct, 2021 | Towards Data Science](https://towardsdatascience.com/production-planning-and-resource-management-of-manufacturing-systems-in-python-5458e9c04183)

# Etat du projet

## 04.11.2021

Le digital tween est capable d’ordonnacer des lots aléatoirement. Il est capable de refaire une étape si le temps est échu. Globalement ça foncitionne.

Point à faire :

* Créer des classes pour plateau\_operation, plateau\_machine, plateau\_personnel
* Créer la classe plateau\_personnel
* Coder encore plus orienté object, pour l’instant, trop de logique dans le fichier production\_line. C’est plus très lisible.
* Certaines opérations ne peuvent pas vraiment être recommencée. Ajouter une option pour recommencer totalement la production. (pas urgent)
* Ajouter une fonction pour supprimer les jobs du plateaux de jeu tout en gardant le planning. Les jobs contenu dans la liste de job de la classe production\_line doivent être garder (historique des lots)
* Créer un uml du bordel
* Implémenter les kpis
* Implémenter une fonction de récompense simple. Récompense négative à essayer 🡪 -1 ou 0 pour l’instant
* Ajouter les fonctions avec les employés, qui feront aussi parti du plateau de jeu
* Ajouter les jours fériés dans le planning, pour les jours férié, ajouté une variable du style (next\_jour\_ferié\_in … jours)

Actuellement, généré 1000 données avec 14jobs prend 7 minutes 10. Il faut donc optimiser l’algorithme

[Guide to Reinforcement Learning with Python and TensorFlow (rubikscode.net)](https://rubikscode.net/2021/07/13/deep-q-learning-with-python-and-tensorflow-2-0/)