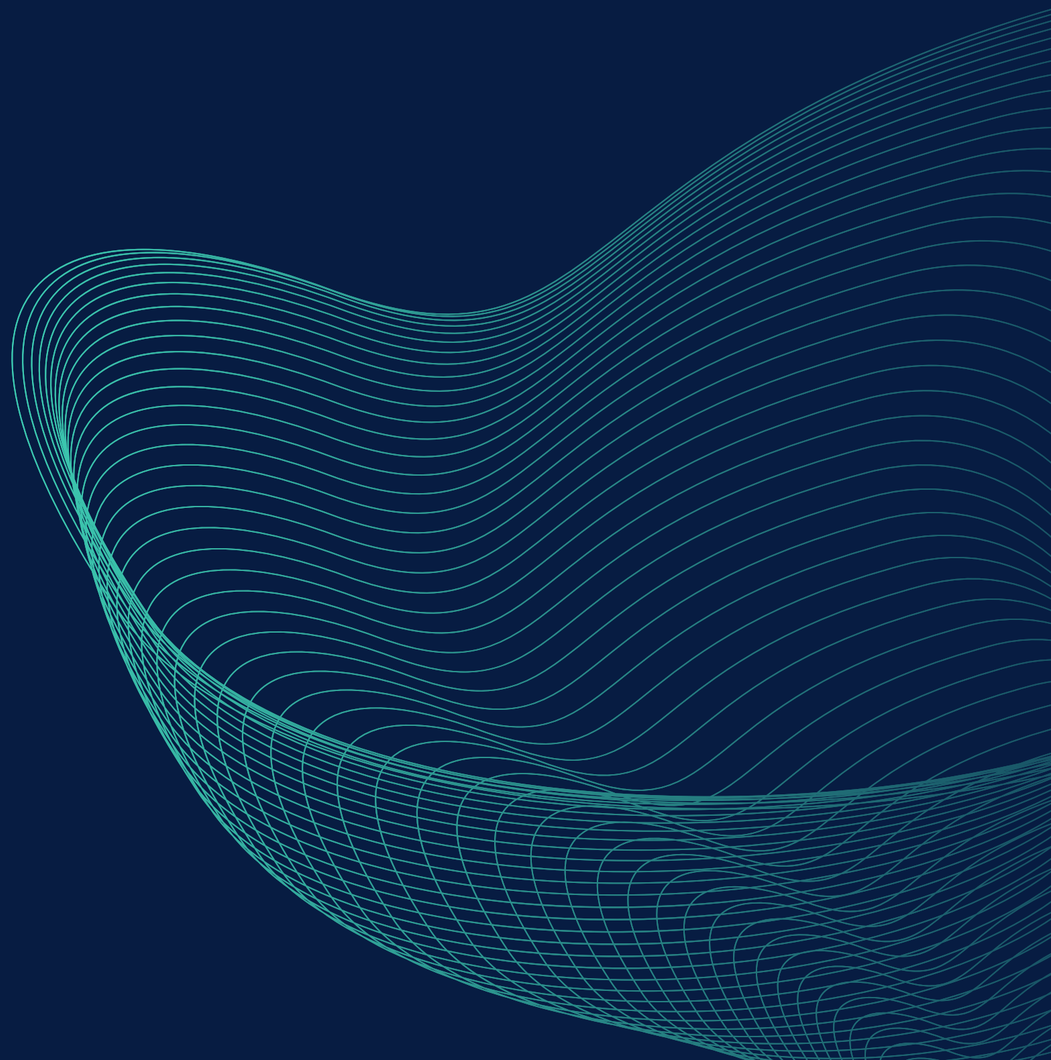
Projet SDP – Systèmes de Décision



Introduction

Contraintes :

- Qualification du personnel
- Un seul projet en même temps
- Congés du personnel à respecter
- Date de livraison du projet
- Nombre de jours de travail

Objectifs :

- Maximiser le gain financier
- Minimiser le nombre maximum d'affectations par employé
- Minimiser la durée maximale d'un projet

Personnel	
nom	compétences
alice	A,B
bob	C
charlie	A
david	B
eve	A,B,C

	Projets		
	A	B	C
I	4	3	2
II	3	2	7
III	4	1	5
IV	3	3	3
V	1	1	4

→ Comment modéliser le problème ?

→ Quelles sont les solutions intéressantes ?

→ Comment choisir parmi ces solutions ?

Plan de la présentation

1) Modélisation du problème de planning

2) Calcul de la surface des solutions non-dominées

3) Modèle de préférence

1) Modélisation du problème de planning

Paramètres d'une instance

Paramètres relatifs aux indices :

- $\mathcal{W}orkers = \{0, \ldots, W-1\}$ est l'ensemble du personnel ($W \in \mathbb{N}\backslash\{0\}$).
- $\mathcal{J}obs = \{0, \ldots, J-1\}$ est l'ensemble des projets ($J \in \mathbb{N}\backslash\{0\}$).
- $\mathcal{S}kills = \{0, \ldots, S-1\}$ est l'ensemble des compétences ($S \in \mathbb{N}\backslash\{0\}$).
- $\mathcal{D}ays = \{0, \ldots, D-1\}$ est l'ensemble des jours de travail (horizon $D \in \mathbb{N}\backslash\{0\}$).

Paramètres relatifs aux projets :

- $\forall j \in \mathcal{J}obs, \quad Gains[j] \in \mathbb{N}$ est le gain obtenu si le projet j est réalisé.
- $\forall j \in \mathcal{J}obs, \quad Penalities[j] \in \mathbb{N}$ est la pénalité par jour de retard du projet j.
- $\forall j \in \mathcal{J}obs, \quad Due\_Dates[j] \in \mathcal{D}ays$ est la date de livraison du projet j négociée avec le client.
- $\forall (j, s) \in \mathcal{J}obs \times \mathcal{S}kills, \quad Work\_Days[j, s] \in \mathbb{N}$ est pour une compétence donnée s, le nombre de jours de travail par personne nécessaires pour que le projet j soit réalisé.

Paramètres relatifs au personnel :

- $\forall (w, s) \in \mathcal{W}orkers \times \mathcal{S}kills, \quad Qualifications[w, s] \in \{0, 1\}$ vaut 1 si le travailleur w possède la compétence s, 0 sinon.
- $\forall (w, d) \in \mathcal{W}orkers \times \mathcal{D}ays, \quad Vacations[w, d] \in \{0, 1\}$ vaut 1 si le travailleur w est en congé le jour d, 0 sinon.

Variables de décision pour le planning

Variable modélisant un planning complet :

- $\forall (w, j, s, d) \in \mathcal{W}orkers \times \mathcal{J}obs \times \mathcal{S}kills \times \mathcal{D}ays, \quad Works[w, j, s, d] \in \{0, 1\}$ vaut 1 si le collaborateur w travaille sur le projet j, le jour d, avec la compétence s, 0 sinon.

Variable pour calculer le gain total :

- $\forall j \in \mathcal{J}obs, \quad Is\_Realized[j] \in \{0, 1\}$ vaut 1 si le projet j est réalisé, 0 sinon.

Contraintes liées au planning

- Contrainte de qualification du personnel : $\forall (w, j, s, d) \in \mathcal{W}orkers \times \mathcal{J}obs \times \mathcal{S}kills \times \mathcal{D}ays$,

$$\mathcal{W}orks[w, j, s, d] \leq \mathcal{Q}ualifications[w, s]$$

- Contrainte d'unicité de l'affectation quotidienne du personnel : *réalisée par la contrainte de congé*
- Contrainte de congé : $\forall (w, d) \in \mathcal{W}orkers \times \mathcal{D}ays$,

$$\sum_{(j,s) \in \mathcal{J}obs \times \mathcal{S}kills} \mathcal{W}orks[w, j, s, d] \leq 1 - \mathcal{V}acations[w, d]$$

- Contrainte de couverture des qualifications du projet : $\forall (j, s) \in \mathcal{J}obs \times \mathcal{S}kills$,

$$\sum_{(w,d) \in \mathcal{W}orkers \times \mathcal{D}ays} \mathcal{W}orks[w, j, s, d] = \mathcal{I}s\_Realized[j] \times \mathcal{W}ork\_Days[j, s]$$

- Contrainte d'unicité de la réalisation d'un projet : *réalisée par définition de* $\mathcal{I}s\_Realized[j] \in \{0, 1\}$

Autres variables de décision

- Pour le calcul de la durée et des pénalités en cas de retard :

- $\forall (j,d) \in \mathcal{J}obs \times \mathcal{D}ays$, $Started\_After[j,d] \in \{0,1\}$ vaut 1 à partir du jour où le travail est commencé pour un projet donné j, 0 sinon.
- $\forall (j,d) \in \mathcal{J}obs \times \mathcal{D}ays$, $Finished\_Before[j,d] \in \{0,1\}$ vaut 1 à partir du jour où le travail est fini pour un projet donné j, 0 sinon.

Indice jour	0	1	2	3	4	5	6	7	8
Jour travaillé	0	0	1	1	0	1	0	0	0
$Started\_After$	0	0	1	1	1	1	1	1	1
$Finished\_Before$	0	0	0	0	0	0	1	1	1

Contraintes détaillées en annexe

Autres variables de décision

- Pour le calcul de la durée maximale :

$Max\_Duration \in \mathbb{N}$ représente la durée maximale tout projet confondu.

- Pour le calcul du nombre maximal d'affectations :

- $\forall (w, j) \in Workers \times Jobs, \quad Is\_Assigned[w, j] \in \{0, 1\}$ vaut 1 si le collaborateur w est affecté au projet j, 0 sinon.
- $Max\_Assigned \in \mathbb{N}$ représente le nombre maximal de projets auxquels un quelconque collaborateur est affecté.

Contraintes détaillées en annexe

Objectifs

Objectif principal : Maximisation du gain financier de l'entreprise

$$\max f_1 = \sum_{j \in \mathcal{J}obs} \left(Gains[j] \times Is\_Realized[j] - Penalities[j] \times \sum_{d \in \{Due\_Dates[j]+1,\ldots,D-1\}} (1 - Finished\_Before) \right)$$

Objectifs secondaires :

- Minimisation du nombre maximum d'affectations par employé

$$\min f_2 = Max\_Assigned$$

- Minimisation de la durée maximale d'un projet

$$\min f_3 = Max\_Duration$$

Résultats sur l'instance toy

	A	B	C	due date	gain	penalty
Job1	1	1	1	3	20	3
Job2	1	2	0	3	15	3
Job3	1	0	2	4	15	3
Job4	0	2	1	3	20	3
Job5	0	0	2	5	10	3

	qualifications	vacations
Olivia	[A, B, C]	[]
Liam	[A, B]	[0]
Emma	[C]	[1]

	0	1	2	3	4
Olivia	B	C	B	C	C
Liam	None	A	B	A	None
Emma	C	None	C	C	None

Planning obtenu sur *l'instance toy* en optimisant séquentiellement les objectifs.

- Gain financier = 65
- Nombre maximal d'affectations = 2
- Durée maximale = 3

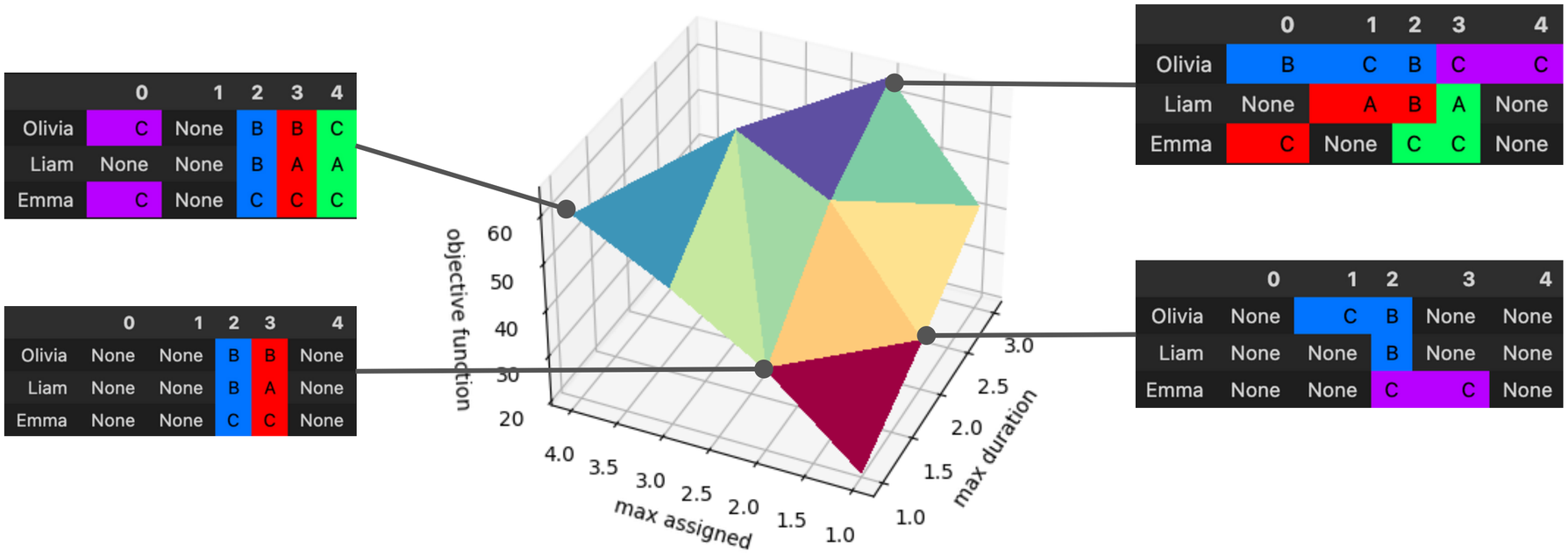
2) Calcul de la surface des solutions non-dominées

Calcul de la surface des solutions non-dominées

$$f = f_1 - \epsilon_2 f_2 - \epsilon_3 f_3$$

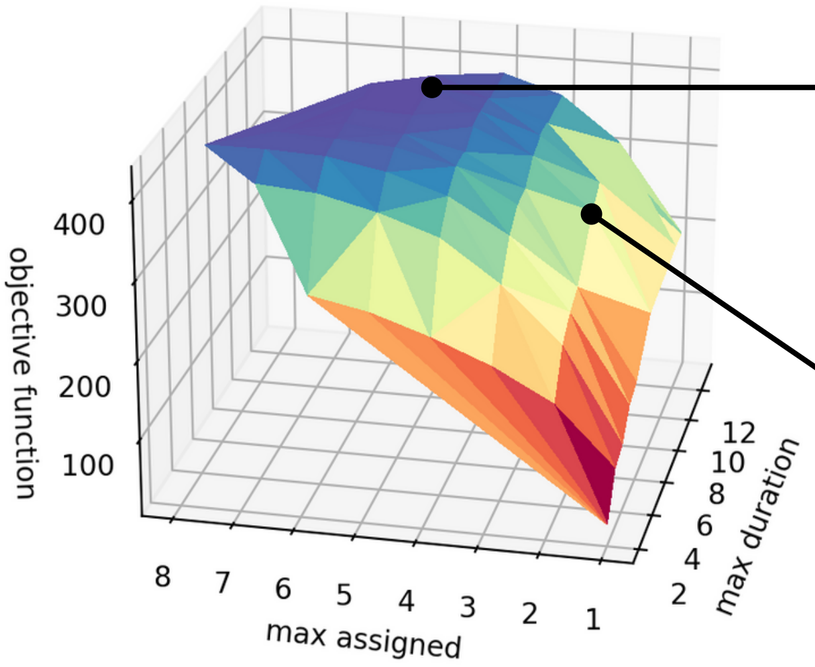
```
ε_duration = horizon
ε_assigned = len(jobs)
while ε_duration ≥ 0 :
    add constraint(max duration <= ε_duration)
    next_ε_duration = 0
    while ε_assigned ≥ 0 :
        add constraint(max assigned <= ε_assigned)
        solution ← model.optimize()
        save(solution)
        ε_assigned = solution's assigned - 1
        next_ε_duration = max(next_ε_duration, solution's assigned)
        remove previous constraint on assigned
    ε_duration = next_ε_duration - 1
    remove previous constraint on duration
```

Surfaces des instances



Surface de l'instance jouet

Surfaces des instances



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Olivia	None	None	C	C	A	C	A	C	A	A	C	A	A	A	A	A	A	A	A	A	A	None
Liam	None	None	D	D	D	D	D	D	D	D	D	D	E	E	D	D	E	E	E	E	E	D
Emma	B	B	B	H	H	H	H	None	None	H	H	H	H	H	H	H	B	B	H	B	B	None
Noah	I	J	J	J	J	J	J	I	I	I	J	J	G	G	G	G	G	G	D	D	D	D
Amelia	J	G	G	J	J	G	J	J	J	J	G	E	E	E	None	None	E	G	E	F	E	E

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Olivia	None	None	B	C	A	C	C	C	A	C	A	A	None	None	None	None	None	None	None	None	None	None
Liam	None	None	D	D	D	D	D	D	D	D	None	None	A	A	A	A	A	A	None	None	None	None
Emma	None	None	B	B	H	H	H	None	None	H	H	H	None	None	None	None	None	None	None	None	None	None
Noah	J	G	G	J	J	J	G	G	H	H	H	H	G	G	G	G	None	None	None	None	None	None
Amelia	None	None	None	None	J	J	F	F	F	F	F	J	F	None	None	None	None	None	None	None	None	None

Surface de l'instance medium

3) Modèle de préférence

Modèle de préférence

Objectif : Classer les solutions non-dominées et les séparer en 3 catégories

inacceptable < correct < satisfaisant

Score d'un planning = Somme pondérée des 3 critères normalisés

Labélisation d'un planning = En fonction du score et de 2 seuils

Modèle de préférence

Résolution d'un problème d'optimisation linéaire
Exemples de plannings labélisés pour calculer les poids et seuils

3) Contraintes:

- Normalisation :

$$\omega_1 + \omega_2 + \omega_3 = 1$$

- Catégorie inacceptable : $\forall (f_1, f_2, f_3) \in \mathcal{I}$,

$$f_1\omega_1 + f_2\omega_2 + f_3\omega_3 \leq \theta_1 - \epsilon$$

- Catégorie correcte :
  - $\forall (f_1, f_2, f_3) \in \mathcal{C}$,

$$f_1\omega_1 + f_2\omega_2 + f_3\omega_3 \leq \theta_2 - \epsilon$$

  - $\forall (f_1, f_2, f_3) \in \mathcal{C}$,

$$f_1\omega_1 + f_2\omega_2 + f_3\omega_3 \geq \theta_1 + \epsilon$$

- Catégorie satisfaisante : $\forall (f_1, f_2, f_3) \in \mathcal{S}$,

$$f_1\omega_1 + f_2\omega_2 + f_3\omega_3 \geq \theta_2 + \epsilon$$

- $(\omega_1, \omega_2, \omega_3) \in [0, 1]^3$ sont les poids des trois critères.
- $(\theta_1, \theta_2) \in [0, 1]^2$ sont les seuils des catégories.
- $\epsilon \in [0, 1]$ représente la marge de séparation des catégories

4) Objectifs: $\max \epsilon$

Afin de séparer au maximum les 3 catégories

Modèle de préférence

Labélisation de 6 solutions non-dominées de *toy_instance*
inacceptable : résultat financier trop faible ou trop grand nb maximum d'affectations
correct/satisfaisant : gain prioritaire

	0	1	2	3	4
Olivia	C	None	B	B	C
Liam	None	None	B	A	A
Emma	C	None	C	C	C

	0	1	2	3	4
Olivia	B	C	B	None	None
Liam	None	A	B	B	None
Emma	C	None	C	None	None

	0	1	2	3	4
Olivia	B	C	B	C	C
Liam	None	A	B	A	None
Emma	C	None	C	C	None

inacceptables < **corrects** < **satisfaisants**

	0	1	2	3	4
Olivia	None	None	A	None	None
Liam	None	None	B	None	None
Emma	None	None	C	None	None

	0	1	2	3	4
Olivia	None	None	B	B	C
Liam	None	None	B	A	A
Emma	None	None	C	C	C

	0	1	2	3	4
Olivia	B	B	C	C	None
Liam	None	B	A	A	None
Emma	C	None	C	None	None

Résultats avec le modèle de préférence

Poids obtenus : $\omega_1 = 0.50$, $\omega_2 = 0.41$ et $\omega_3 = 0.09$.

Cohérents avec les préférences données en entrée du modèle de priorisation des critères.

gain > max_assignations >> duree_max

Meilleur planning obtenu sur le *toy_instance* :

- Meilleur gain parmi les solutions non-dominées
- Remplit les conditions sur le nombre maximum d'assignations et la durée maximum de projet

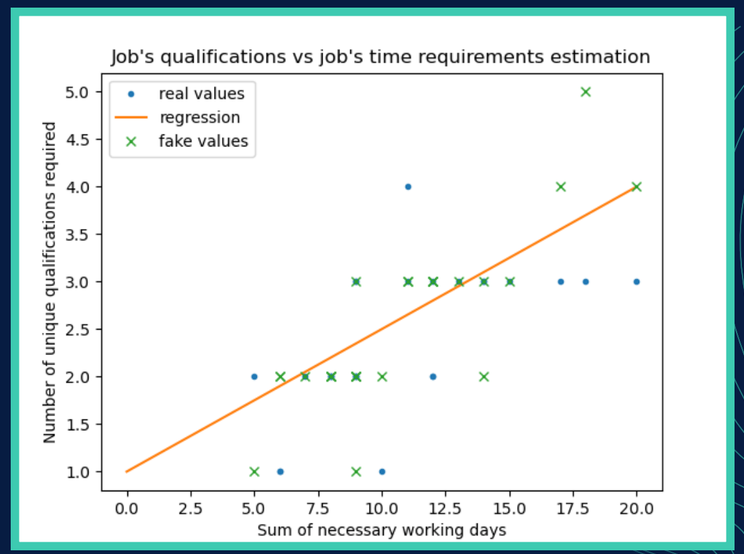
Annexe : Création d'instances aléatoires

- **Horizon** → loi exponentielle privilégiant les horizons courts (~5 jours)
- Nombre de **travailleurs**, **jobs** et **compétences** → randint(1, 10)

- **Membre du personnel :**
  - **Qualifications** → Nombre et choix tiré uniformément parmi les compétences existantes
  - **Congés** → Tirage privilégiant les petits nombres, choix des jours uniforme jusqu'à l'horizon

- **Projets à réaliser :**
  - **Gain** → randint(10, 80)
  - **Date de rendu** (aléatoire)

  - Nombre de **jours de travail**
  - **Qualifications requises**



Annexe : Autres contraintes

- $\forall (w, j, s, d) \in \mathcal{W}orkers \times \mathcal{J}obs \times \mathcal{S}ills \times \mathcal{D}ays,$

$$Works[w, j, s, d] \leq Started\_After[j, d]$$

- $\forall (j, d) \in \mathcal{J}obs \times \mathcal{D}ays \backslash \{D - 1\},$

$$Started\_After[j, d] \leq Started\_After[j, d + 1]$$

- $\forall (j, d) \in \mathcal{J}obs \times \mathcal{D}ays,$

$$1 - Started\_After[j, d] \leq Is\_Realized[j]$$

(pour des raisons de praticité, on fixe $Started\_After$ à 1 lorsque le projet n'est pas réalisé)

- $\forall (w, j, s, d) \in \mathcal{W}orkers \times \mathcal{J}obs \times \mathcal{S}ills \times \mathcal{D}ays,$

$$Works[w, j, s, d] \leq 1 - Finished\_Before[j, d]$$

- $\forall (j, d) \in \mathcal{J}obs \times \mathcal{D}ays \backslash \{D - 1\},$

$$Finished\_Before[j, d] \leq Finished\_Before[j, d + 1]$$

- $\forall (j, d) \in \mathcal{J}obs \times \mathcal{D}ays,$

$$1 - Finished\_Before[j, d] \leq Is\_Realized[j]$$

(pour des raisons de praticité, on fixe $Finished\_Before$ à 1 lorsque le projet n'est pas réalisé)

Annexe : Autres contraintes

- $Max\_Duration : \forall j \in \mathcal{J}obs,$

$$\sum_{d \in \mathcal{D}ays} Started\_After[j,d] - Finished\_Before[j,d] \leq Max\_Duration$$

- $Is\_Assigned :$
  - $\forall (w,j,s,d) \in \mathcal{W}orkers \times \mathcal{J}obs \times \mathcal{S}ills \times \mathcal{D}ays,$

$$Works[w,j,s,d] \leq Is\_Assigned[w,j]$$

  - $\forall (w,j) \in \mathcal{W}orkers \times \mathcal{J}obs,$

$$Is\_Assigned[w,j] \leq \sum_{(s,d) \in \mathcal{S}ills \times \mathcal{D}ays} Works[w,j,s,d]]$$

- $Max\_Assigned : \forall w \in \mathcal{W}orkers,$

$$\sum_{j \in \mathcal{J}obs} Is\_Assigned[w,j] \leq Max\_Assigned$$