

Ordonnancement cumulatif avec calendriers et heures supplémentaires

Samuel Cloutier
Claude-Guy Quimper

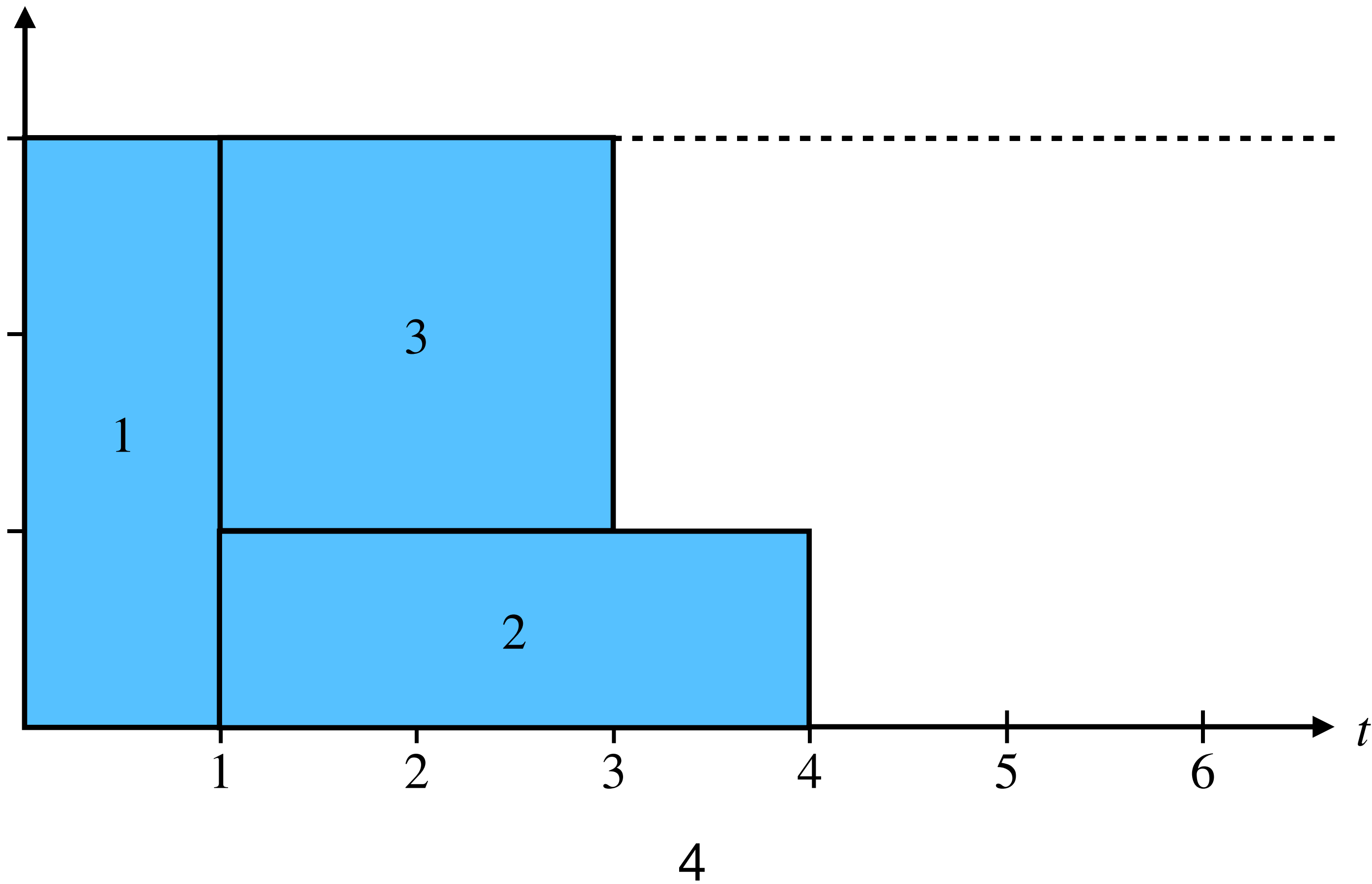
Introduction

- Dans quel ordre accomplir des tâches ne pouvant pas toutes être faites en même temps?
- Problème d'ordonnancement
- Utile pour : Réfection de navires, planification de production manufacturière, planification des tâches d'un robot à deux bras...

Plan

- Le problème d'ordonnancement cumulatif
- Programmation par contraintes
- Les calendriers et les heures supplémentaires
- Contrainte **CALENDAR**
- Contrainte **CUMULATIVEOVERTIME**
- Résultats

Ordonnancement cumulatif



La programmation par contraintes

- Type de programmation déclarative
- On dicte des variables, des contraintes, une fonction objectif
- Un solveur général résout le problème pour nous

```
1
2 var 1..20: x;
3 var 2..4: y;
4 var 0..5: z;
5
6 constraint z ^ y < x;
7 solve maximize z;
8
```

Exécution

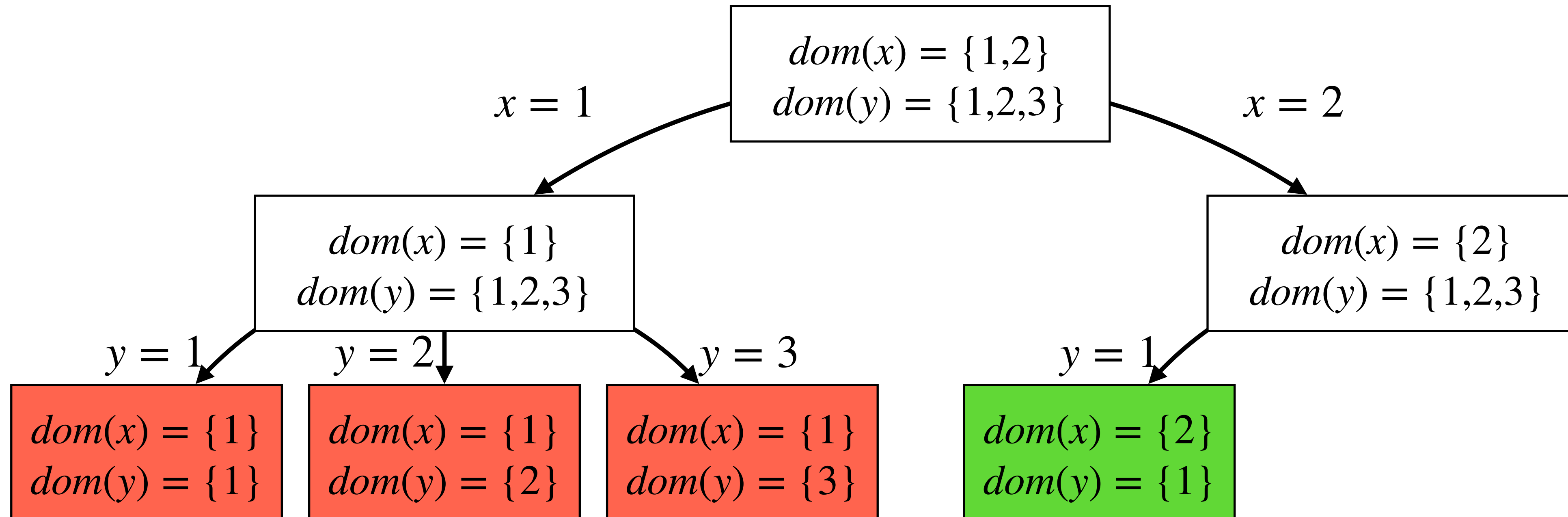


```
Running Test_simple.mzn
x = 17;
y = 2;
z = 4;
-----
=====
Finished in 170msec.
```

Fonctionnement d'un solveur

La recherche

Contrainte: $y < x$



Fonctionnement d'un solveur

La recherche - le filtrage

- Il est possible de couper des branches sans solutions
- Exemple simpliste:

$$\begin{array}{ll} x \neq y & x \in \{1,3\} \\ x \neq z & y \in \{1,2\} \\ y \neq z & z \in \{1,2\} \end{array}$$

- Après avoir branché sur $x = 1$, on peut détecter directement un conflit
- Une contrainte globale *AllDifferent* aurait filtré la valeur 1 sans branchement
- Un domaine filtré est dit cohérent (avec divers niveaux de cohérences)

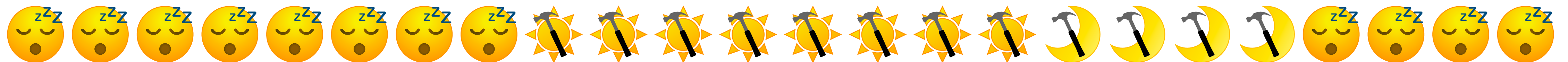
Fonctionnement d'un solveur

La recherche - les heuristiques de recherche

- L'ordre dans lequel l'arbre est exploré a un impact important
- Si l'on fait toujours les bons choix, le filtrage est presque obsolète
- Dans un contexte d'optimisation, une heuristique trouvant rapidement des solutions de qualité aidera le Branch-and-Bound
- Les heuristiques dépendent grandement du problème
- Il n'y a pas de garantie
- Le filtrage évite de visiter des branches sans intérêt, les (bons) heuristiques tentent de nous faire explorer des branches prometteuses.

Calendriers et temps supplémentaire

- Chaque temps ne permet pas toujours de faire progresser une tâche (par exemple, les fins de semaine)
- Le temps supplémentaire est possible pour terminer des tâches plus tôt
- Chaque tâche possèdera un calendrier
- 🛠️ dénote une heure de travail régulière, 😴 une heure de fermeture et 🌙 une heure de travail supplémentaire
- Exemple de journée 8-16 avec plage de 4 heures supplémentaires:



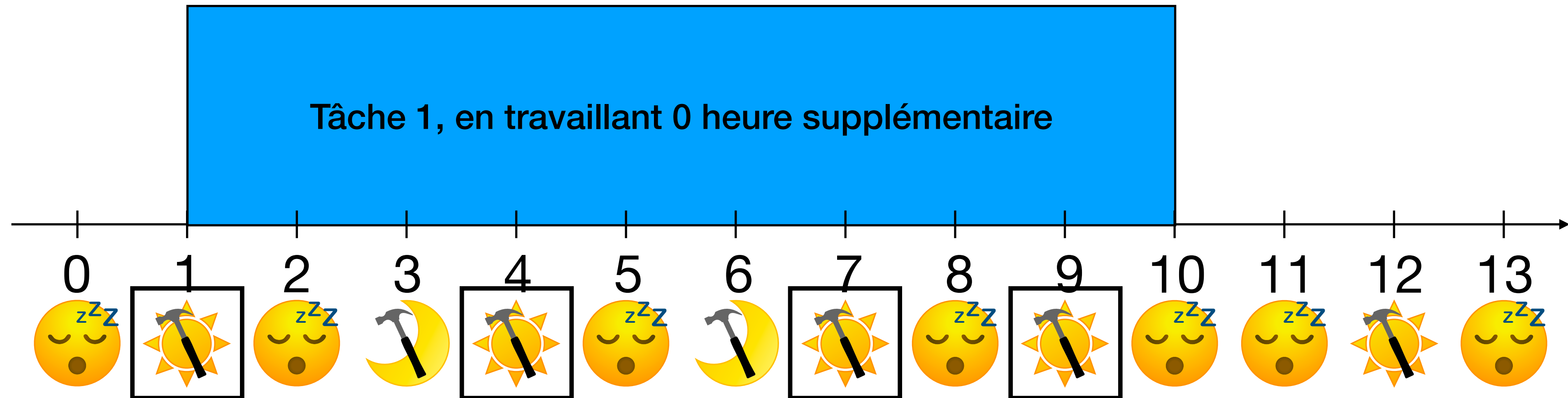
Calendriers et temps supplémentaire

- Les durées (nouvelle variable) ne sont pas fixes et dépendent du contexte
- L'utilisation de temps supplémentaire (nouvelle variable) a un coût additionnel
- Règles à respecter:
 - Une tâche ne peut commencer ou se terminer à une heure non travaillée
 - Le temps travaillé dans la période d'exécution d'une tâche doit être sa durée de travail
 - On ne peut travailler plus d'heures supplémentaires qu'il y en a dans la période d'exécution d'une tâche

Calendriers et temps supplémentaire

Exemple d'effet

- Tâche de durée de travail 4
- Valeurs possibles (T, D, S) :
 $(1, 9, 0)$



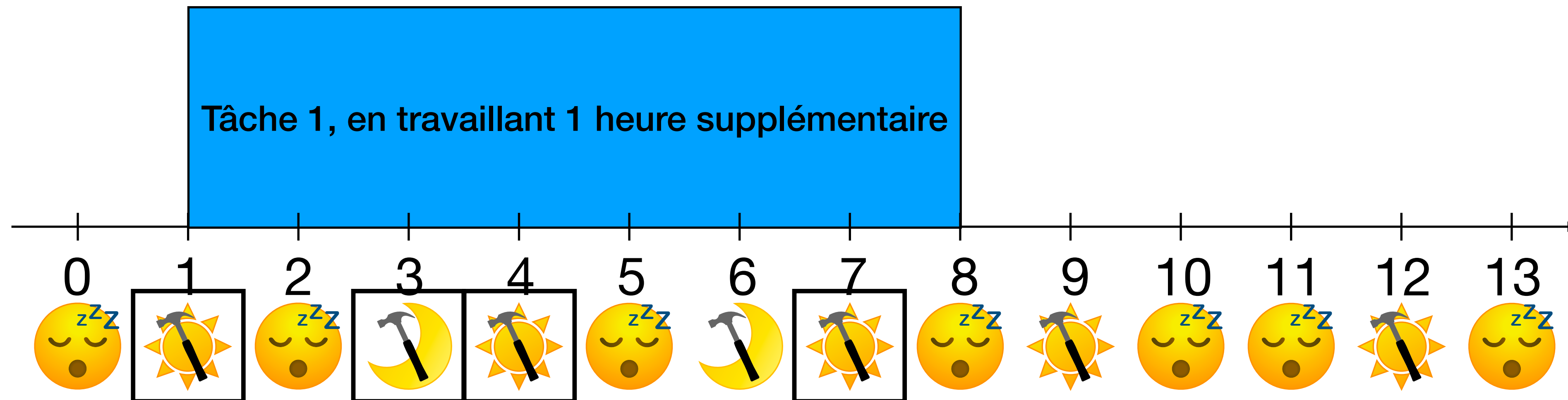
Calendriers et temps supplémentaire

Exemple d'effet

- Tâche de durée de travail 4

Valeurs possibles (T, D, S) :

$(1, 9, 0)$ $(1, 7, 1)$



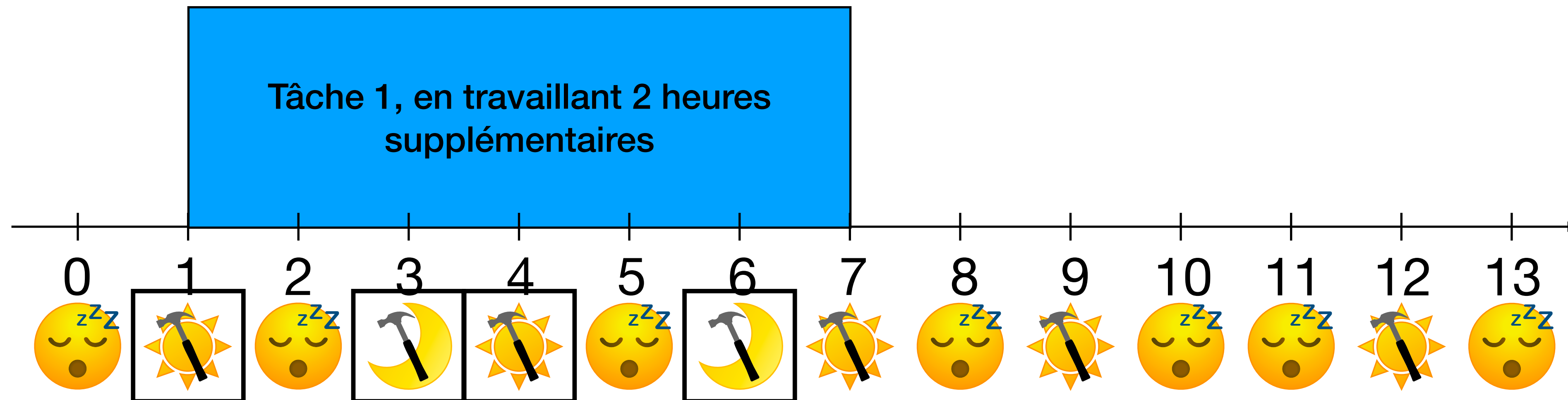
Calendriers et temps supplémentaire

Exemple d'effet

- Tâche de durée de travail 4

Valeurs possibles (T, D, S) :

$(1,9,0)$ $(1,7,1)$ $(1,6,2)$



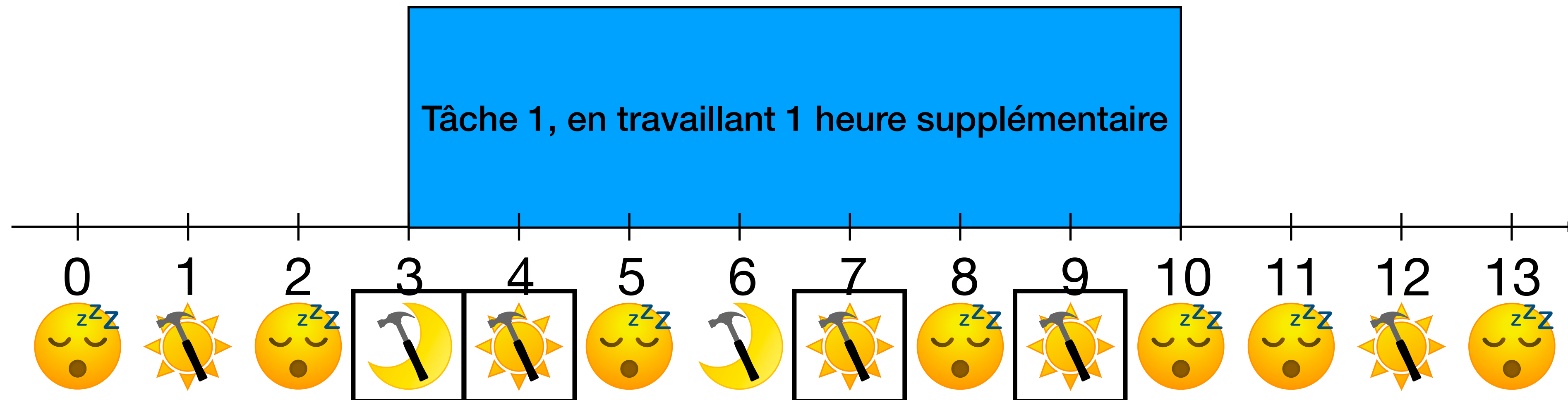
Calendriers et temps supplémentaire

Exemple d'effet

- Tâche de durée de travail 4

Valeurs possibles (T, D, S) :

$(1,9,0)$ $(1,7,1)$ $(1,6,2)$
 $(3,7,1)$



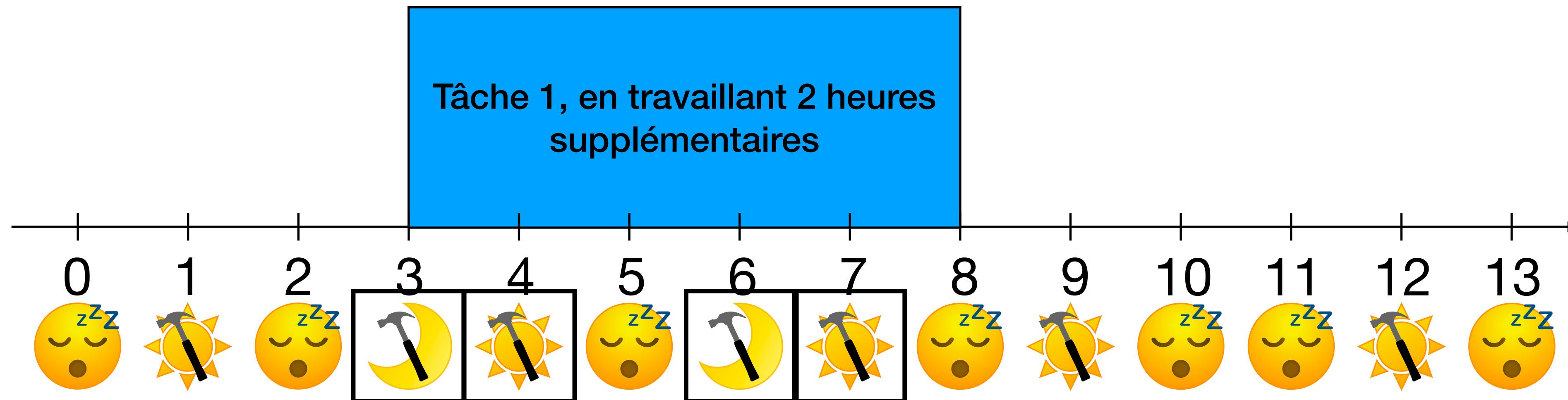
Calendriers et temps supplémentaire

Exemple d'effet

- Tâche de durée de travail 4

Valeurs possibles (T, D, S) :

$(1,9,0)$	$(1,7,1)$	$(1,6,2)$
$(3,7,1)$	$(3,5,2)$	



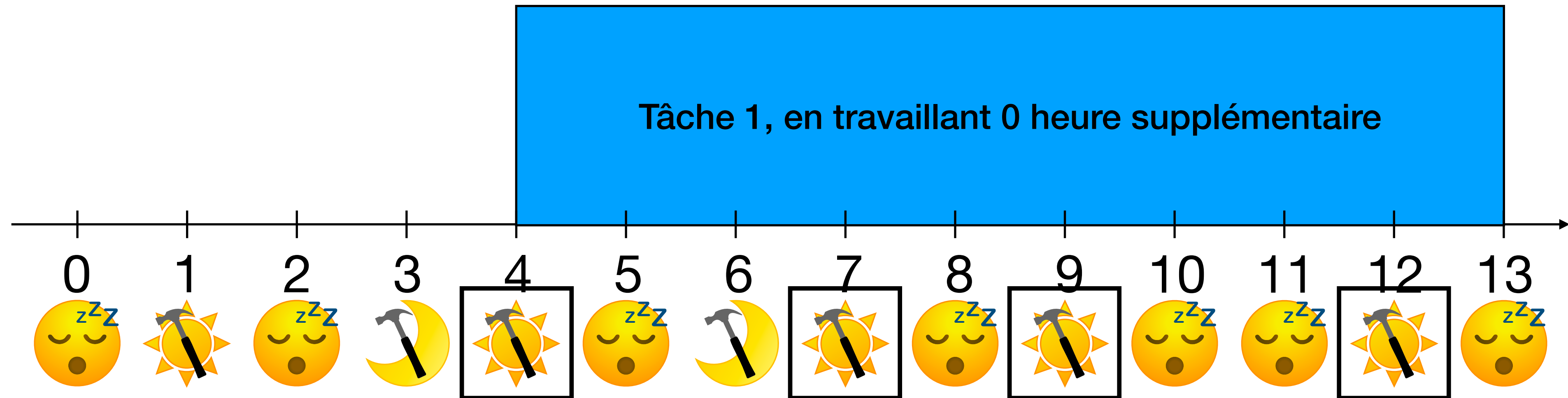
Calendriers et temps supplémentaire

Exemple d'effet

- Tâche de durée de travail 4

Valeurs possibles (T, D, S):

(1,9,0)	(1,7,1)	(1,6,2)
(3,7,1)	(3,5,2)	(4,9,0)



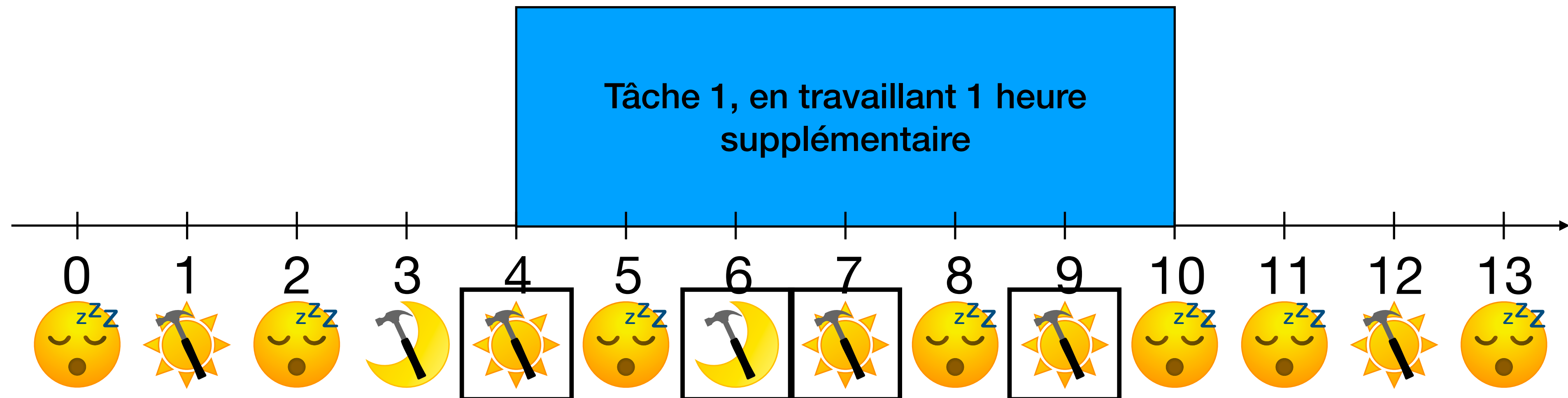
Calendriers et temps supplémentaire

Exemple d'effet

- Tâche de durée de travail 4

Valeurs possibles (T, D, S) :

$(1, 9, 0)$	$(1, 7, 1)$	$(1, 6, 2)$
$(3, 7, 1)$	$(3, 5, 2)$	$(4, 9, 0)$
$(4, 6, 1)$		



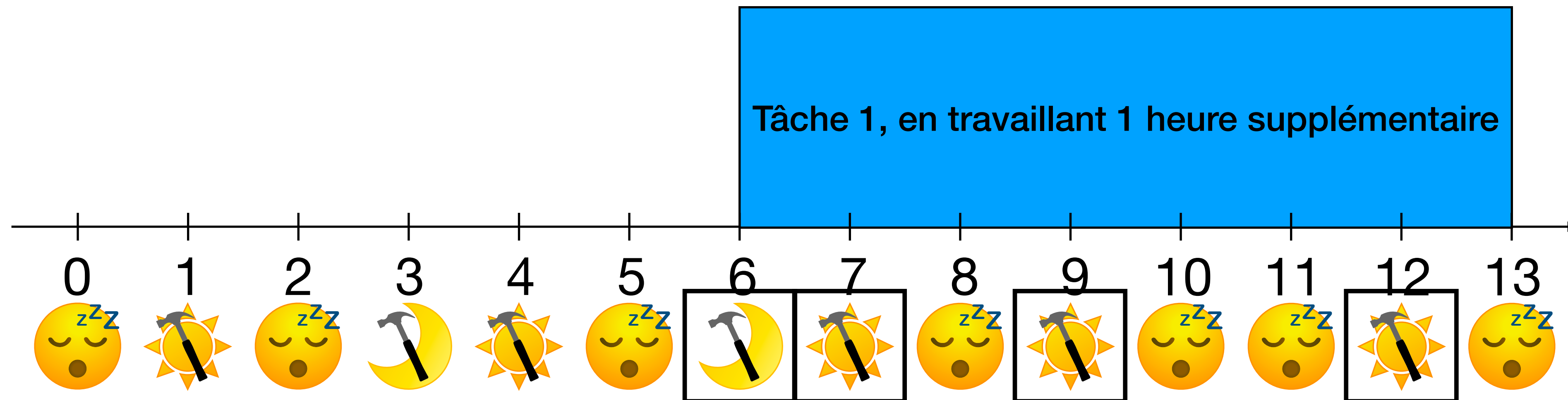
Calendriers et temps supplémentaire

Exemple d'effet

- Tâche de durée de travail 4

Valeurs possibles (T, D, S) :

$(1,9,0)$	$(1,7,1)$	$(1,6,2)$
$(3,7,1)$	$(3,5,2)$	$(4,9,0)$
$(4,6,1)$	$(6,7,1)$	



Nos contributions

Contrainte **CALENDAR**

- Permet de filtrer jusqu'à cohérence de bornes les variables de temps de début, durées et surtemps selon les calendriers.

Valeurs possibles (T, D, S) :

$(1, 9, 0)$ $(1, 7, 1)$ $(1, 6, 2)$
 $(3, 7, 1)$ $(3, 5, 2)$ $(4, 9, 0)$
 $(4, 6, 1)$ $(6, 7, 1)$

T : Temps de début

D : Durée réelle

S : Surtemps travaillé

- Avec la tâche (durée de travail 4) et le calendrier précédent, $dom(T) = \{0..8\}$, $dom(D) = \{4..8\}$, $dom(S) = \{0..4\}$ sont filtrés à:

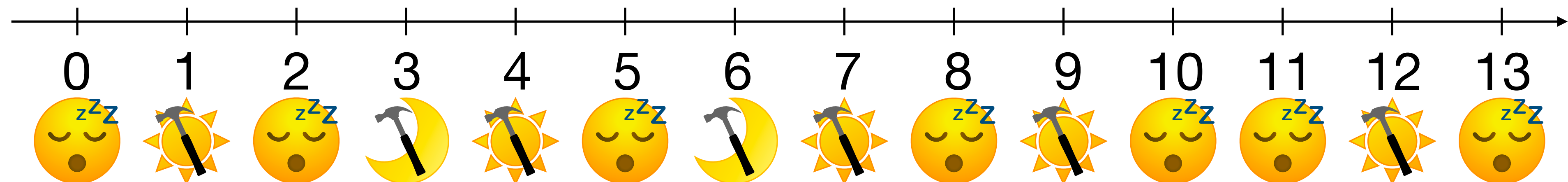
$dom(T) = \{1..6\}$ $dom(D) = \{5..7\}$ $dom(S) = \{1..2\}$

Structures clefs pour le filtrage

- On veut faire le filtrage rapidement
- On précalcule quatre vecteurs qui permettront plusieurs calculs en temps constant

A

-1	0	0	0	1	1	1	2	2	3	3	3	4	4
----	---	---	---	---	---	---	---	---	---	---	---	---	---

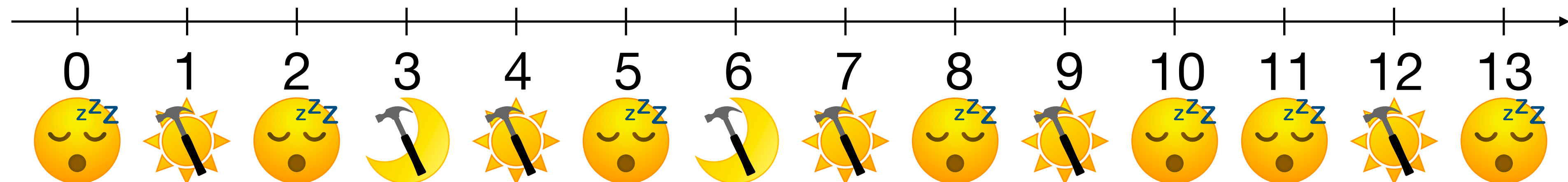


Structures clefs pour le filtrage

- On veut faire le filtrage rapidement
- On précalcule quatre vecteurs qui permettront plusieurs calculs en temps constant

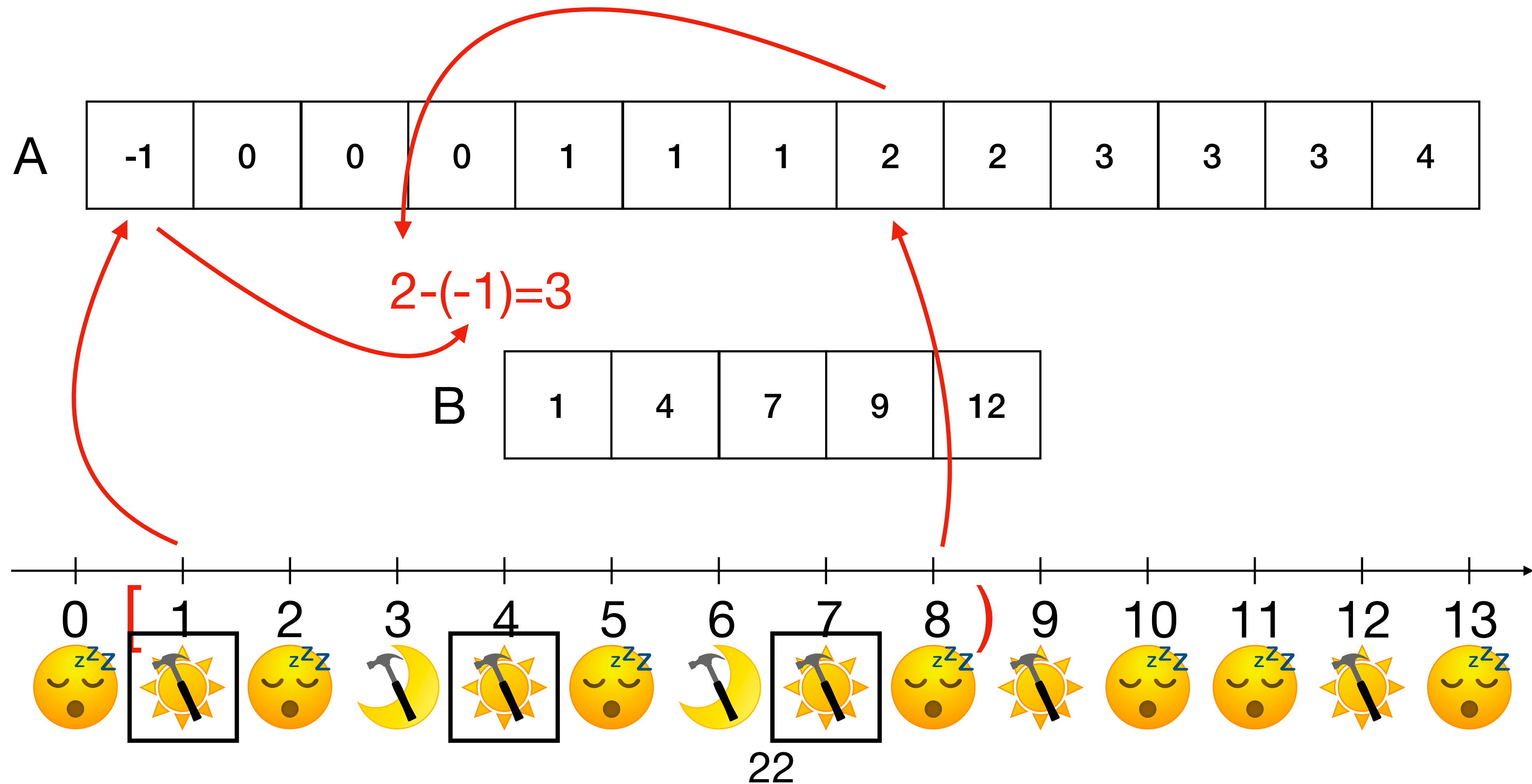
B

1	4	7	9	12
---	---	---	---	----



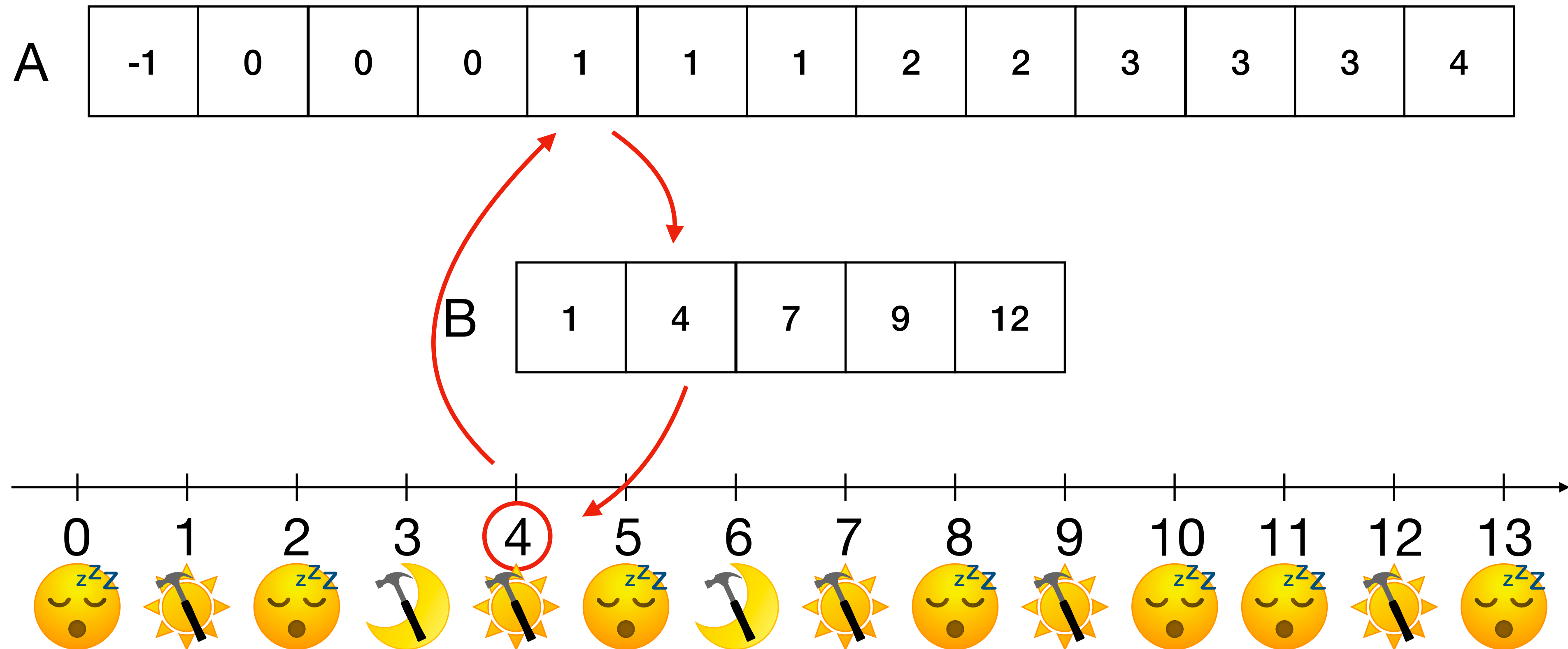
Fonctions utilitaires

Compter le temps régulier dans un intervalle



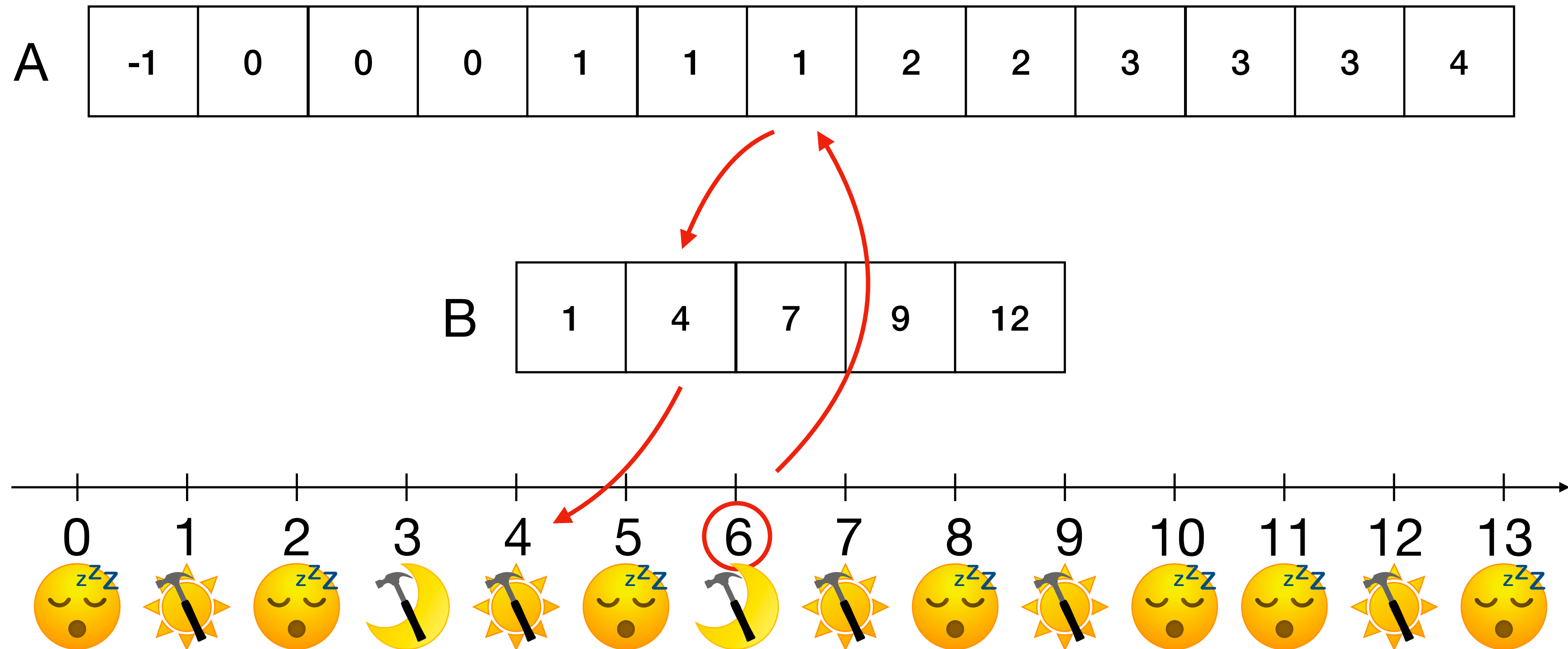
Fonctions utilitaires

Plus grand temps régulier qui n'est pas après un temps donné



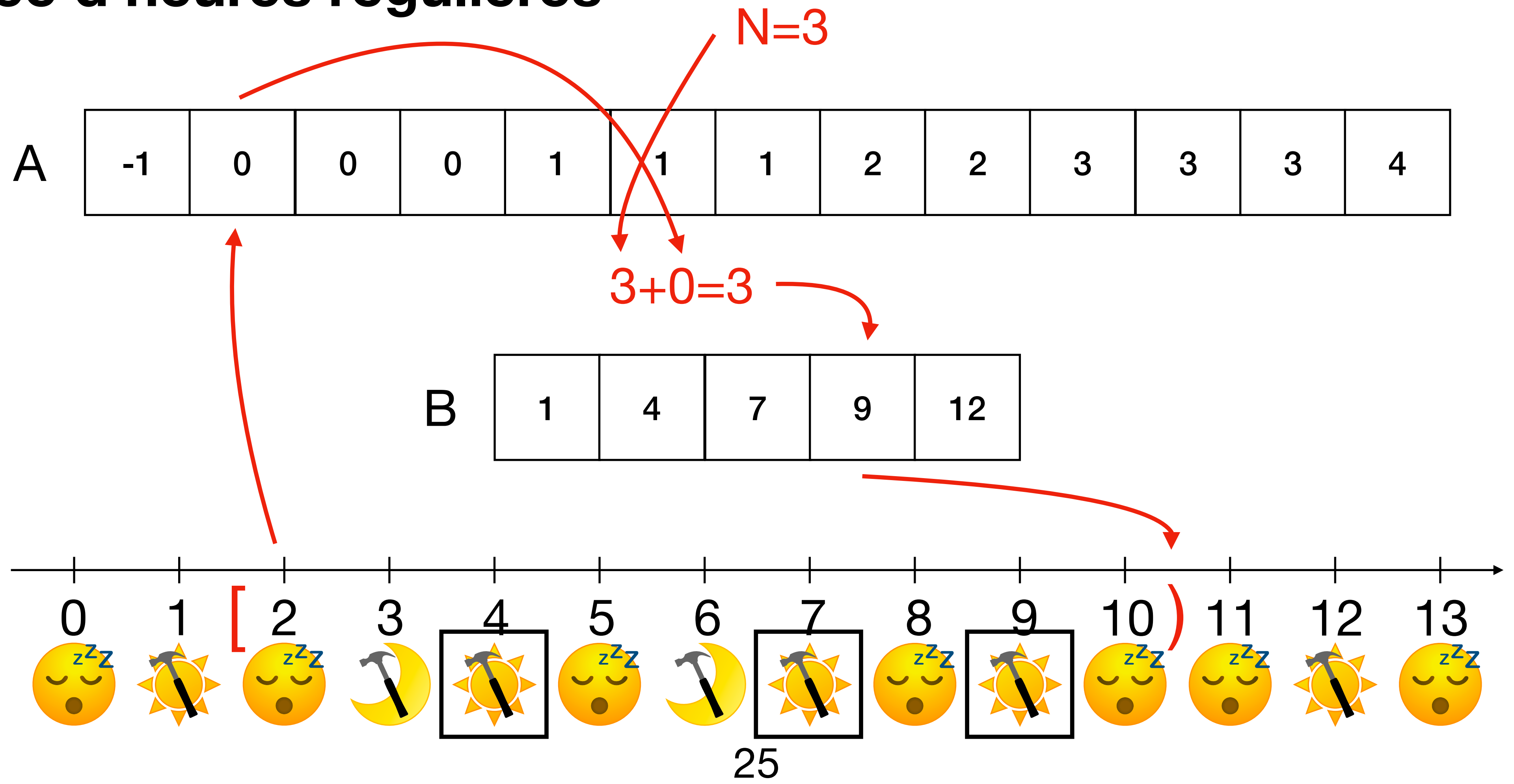
Fonctions utilitaires

Plus grand temps régulier qui n'est pas après un temps donné



Fonctions utilitaires

Trouver la plus petite fin d'intervalle possédant une quantité donnée d'heures régulières



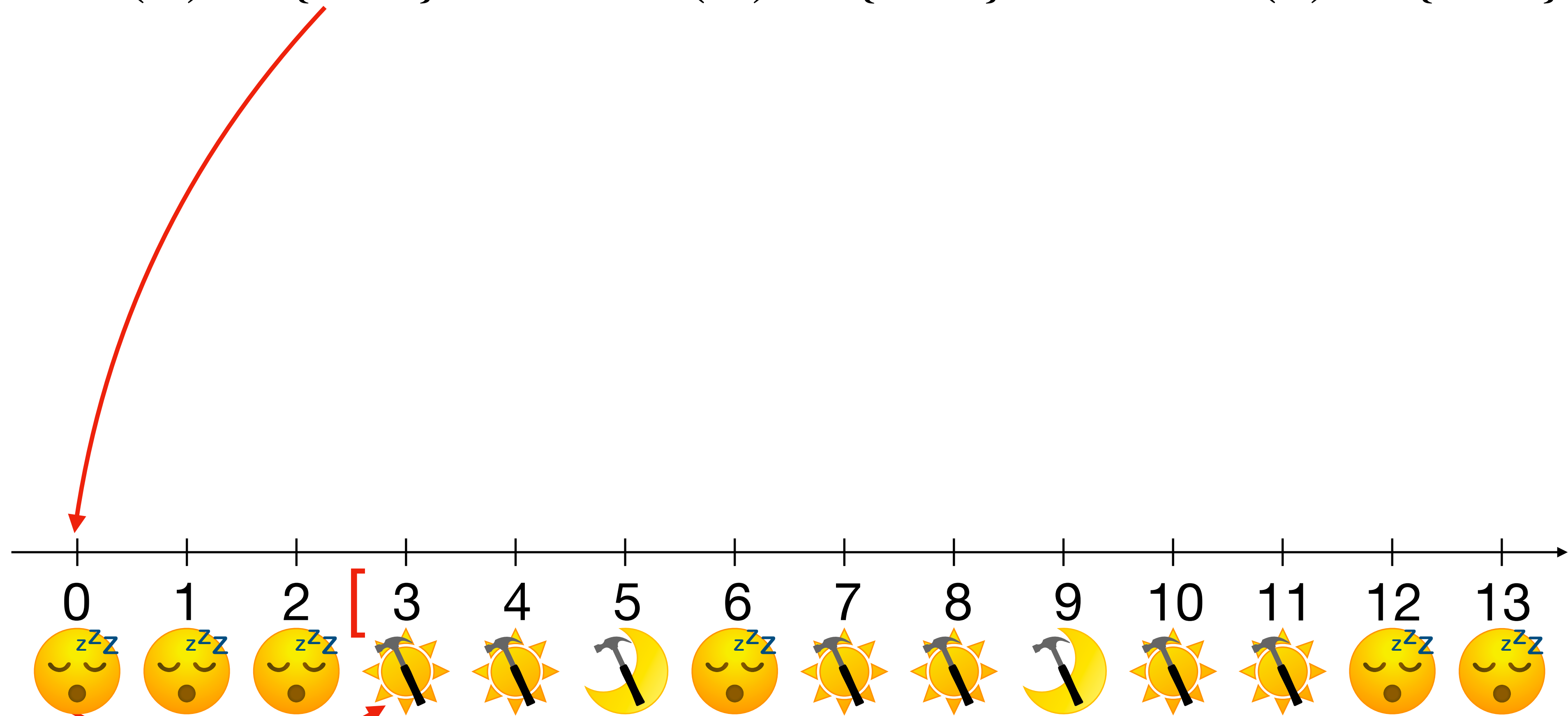
Exemple de filtrage

Borne inférieure du temps de début, tâche de durée de travail 2

$$\text{dom}(T) = \{0..8\}$$

$$\text{dom}(D) = \{3..6\}$$

$$\text{dom}(S) = \{0..1\}$$



Exemple de filtrage

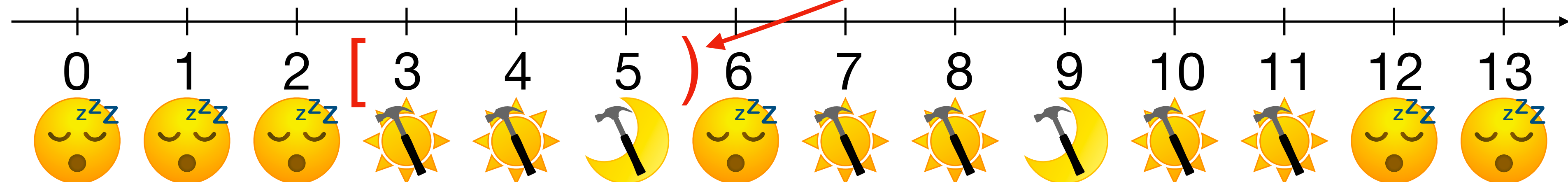
Borne inférieure du temps de début, tâche de durée de travail 2

$$dom(T) = \{0..8\}$$

$$dom(D) = \{3..6\}$$

$$dom(S) = \{0..1\}$$

1. Ajouter la durée de travail



Exemple de filtrage

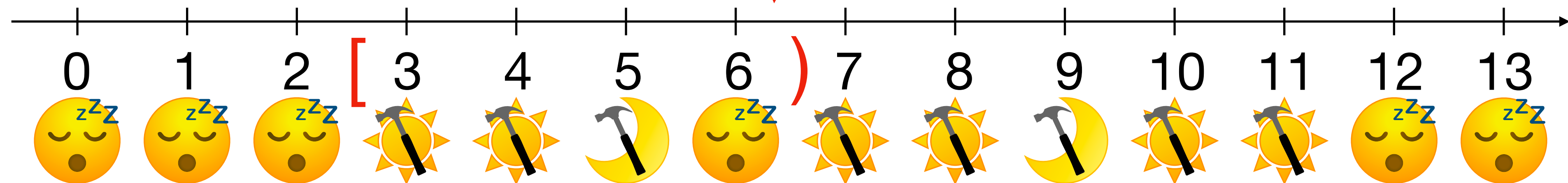
Borne inférieure du temps de début, tâche de durée de travail 2

$$\text{dom}(T) = \{0..8\}$$

$$\text{dom}(D) = \{3..6\}$$

$$\text{dom}(S) = \{0..1\}$$

2. Ajuster à la durée minimale



Exemple de filtrage

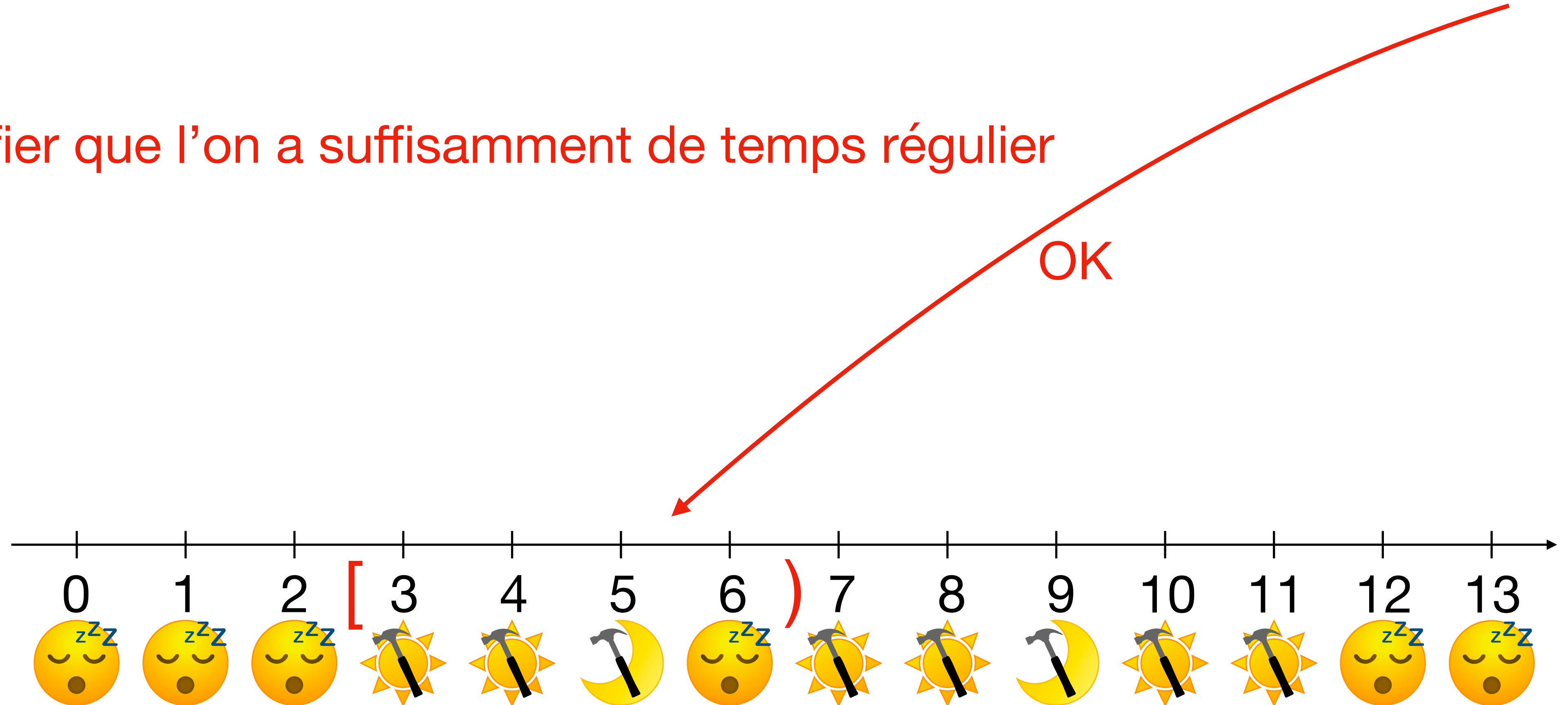
Borne inférieure du temps de début, tâche de durée de travail 2

$$dom(T) = \{0..8\}$$

$$dom(D) = \{3..6\}$$

$$dom(S) = \{0..1\}$$

3. Vérifier que l'on a suffisamment de temps régulier



Exemple de filtrage

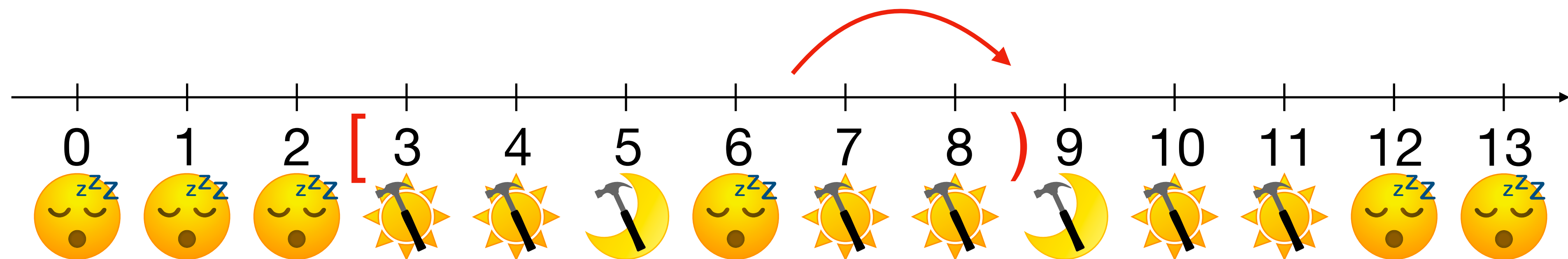
Borne inférieure du temps de début, tâche de durée de travail 2

$$dom(T) = \{0..8\}$$

$$dom(D) = \{3..6\}$$

$$dom(S) = \{0..1\}$$

4. Corriger la queue de surtemps non travaillée



Exemple de filtrage

Borne inférieure du temps de début, tâche de durée de travail 2

$$dom(T) = \{0..8\}$$

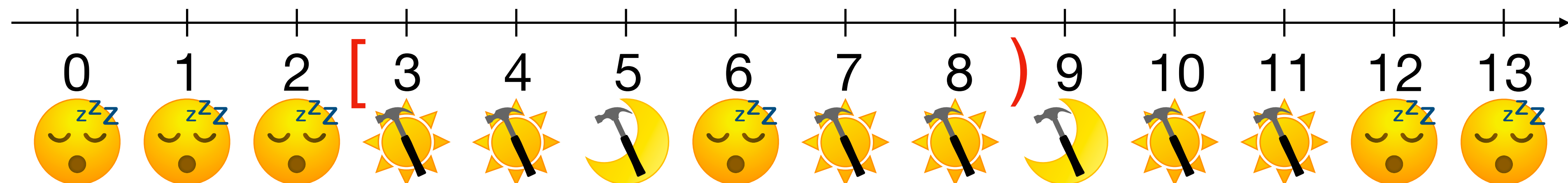
$$dom(D) = \{3..6\}$$

$$dom(S) = \{0..1\}$$

5. Vérifier le maximum de D , le minimum de S

et les têtes/queues non travaillées

Oops



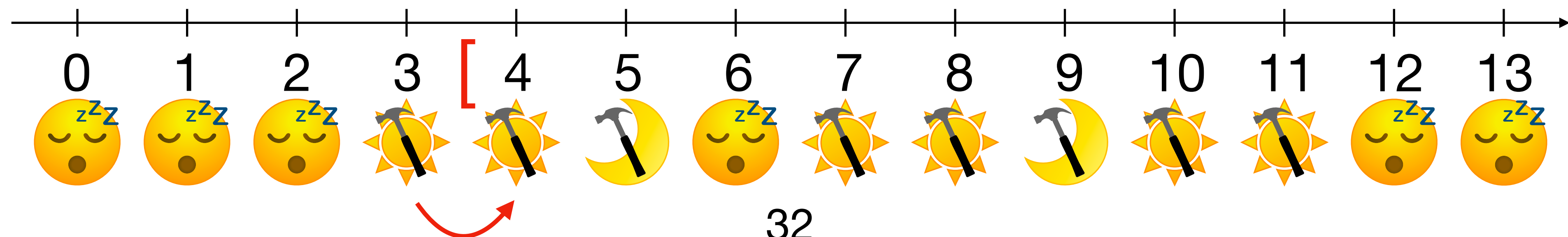
Exemple de filtrage

Borne inférieure du temps de début, tâche de durée de travail 2

$$dom(T) = \{0..8\}$$

$$dom(D) = \{3..6\}$$

$$dom(S) = \{0..1\}$$



Exemple de filtrage

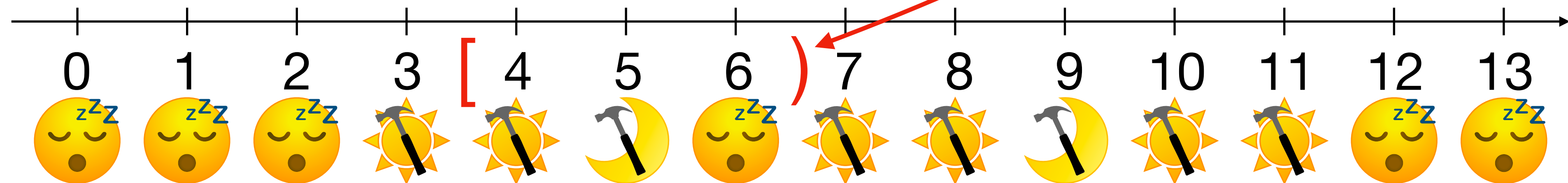
Borne inférieure du temps de début, tâche de durée de travail 2

$$\text{dom}(T) = \{0..8\}$$

$$\text{dom}(D) = \{3..6\}$$

$$\text{dom}(S) = \{0..1\}$$

1. Ajouter la durée de travail



Exemple de filtrage

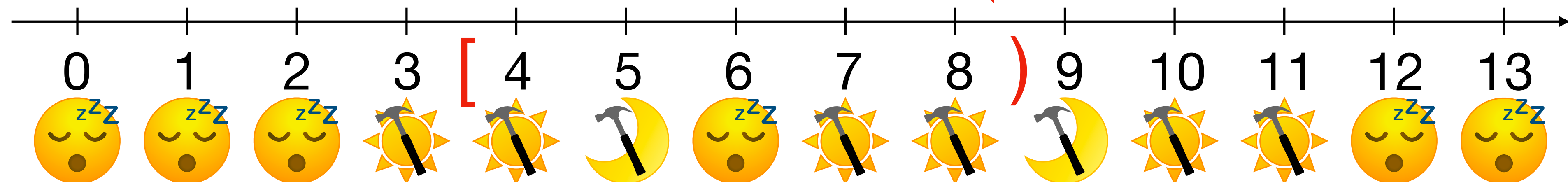
Borne inférieure du temps de début, tâche de durée de travail 2

$$dom(T) = \{0..8\}$$

$$dom(D) = \{3..6\}$$

$$dom(S) = \{0..1\}$$

2. Ajuster à la durée minimale



Exemple de filtrage

Borne inférieure du temps de début, tâche de durée de travail 2

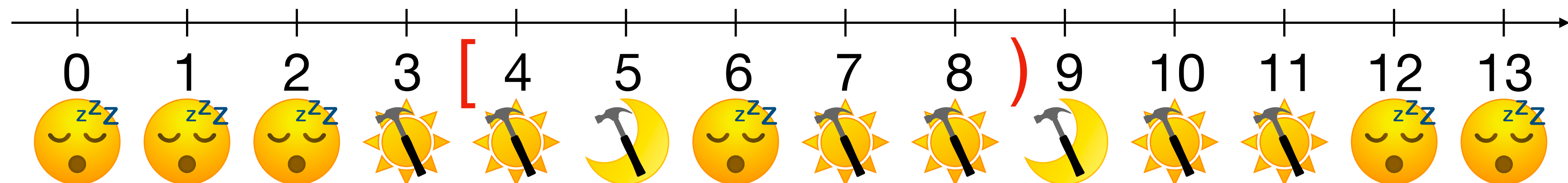
$$dom(T) = \{0..8\}$$

$$dom(D) = \{3..6\}$$

$$dom(S) = \{0..1\}$$

3. Vérifier que l'on a suffisamment de temps régulier

OK



Exemple de filtrage

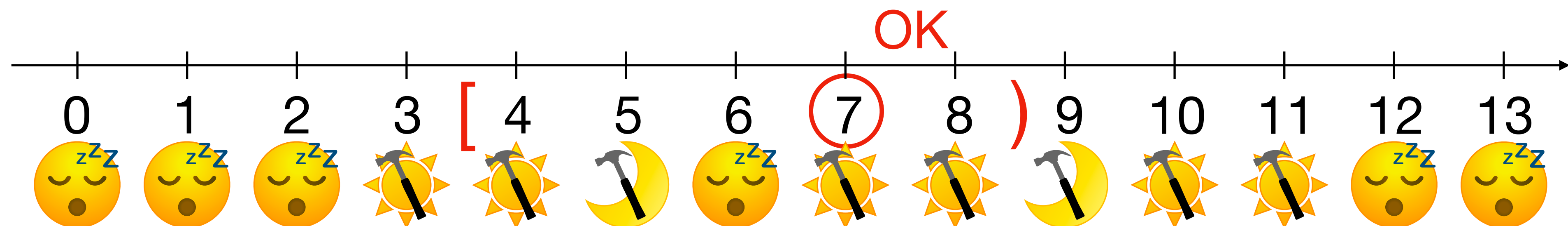
Borne inférieure du temps de début, tâche de durée de travail 2

$$dom(T) = \{0..8\}$$

$$dom(D) = \{3..6\}$$

$$dom(S) = \{0..1\}$$

4. Corriger la queue de surtemps non travaillée



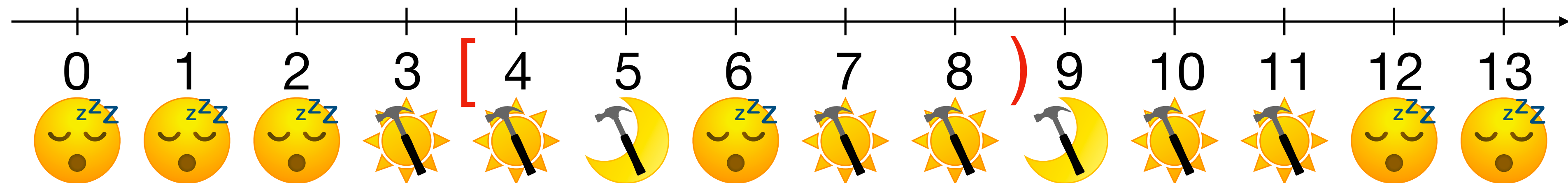
Exemple de filtrage

Borne inférieure du temps de début, tâche de durée de travail 2

$$dom(T) = \{4..8\}$$

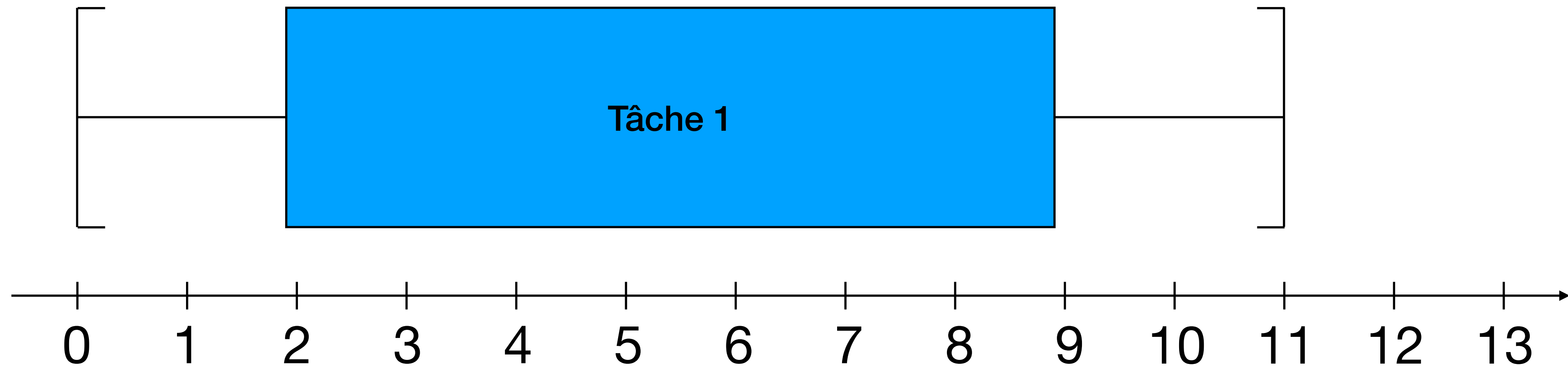
$$dom(D) = \{3..6\}$$

$$dom(S) = \{0..1\}$$

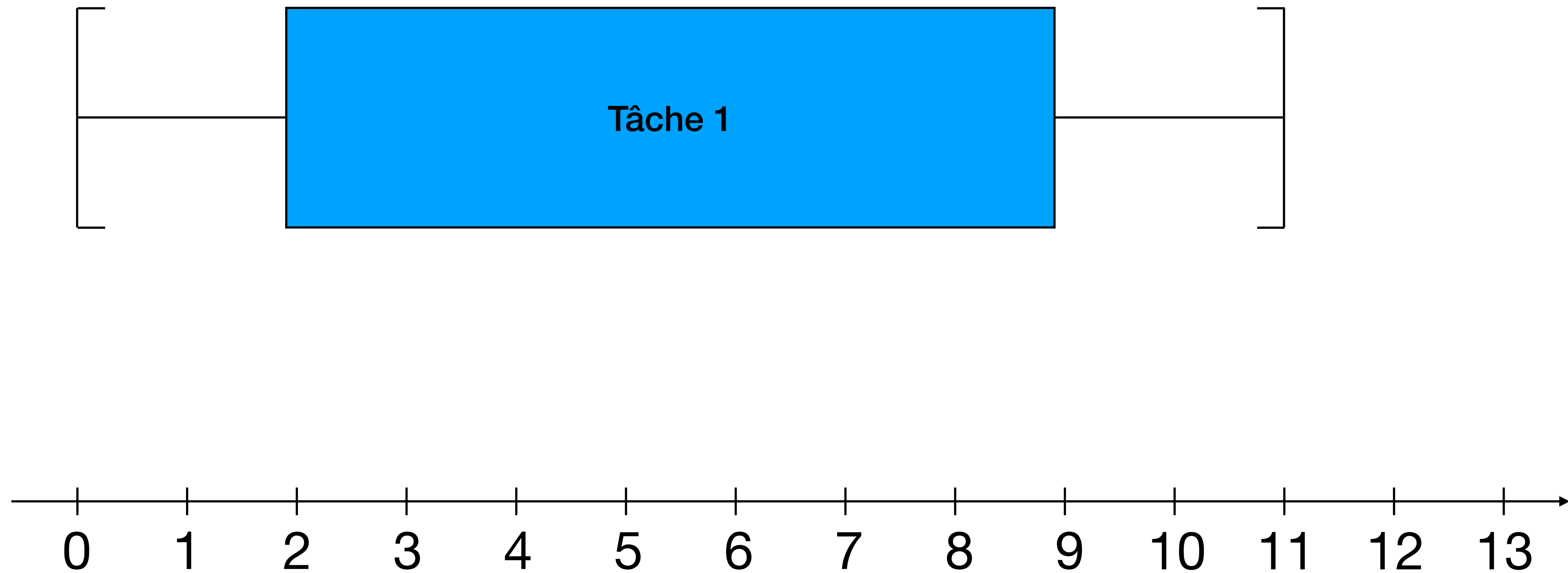


Filtrage de la CUMULATIVE par Time-Tabling

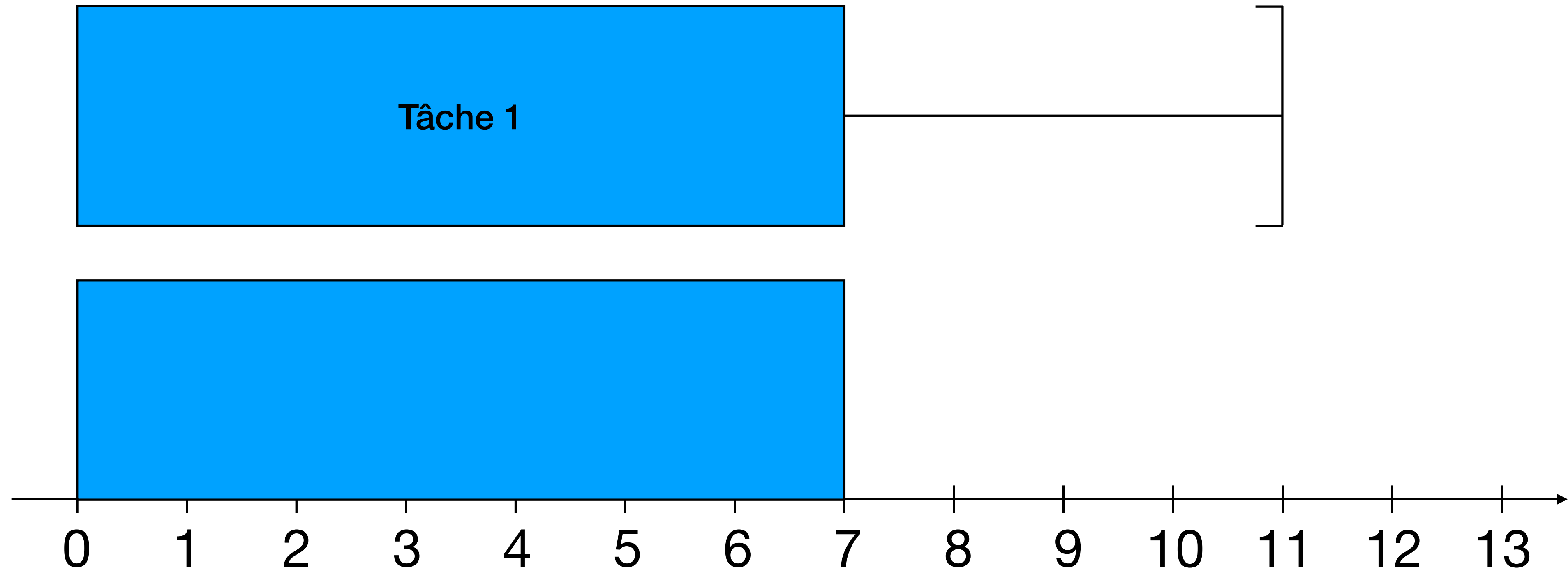
- Pour certaines contraintes, filtrer autant que possible est trop lent.
- On applique donc une cohérence plus faible



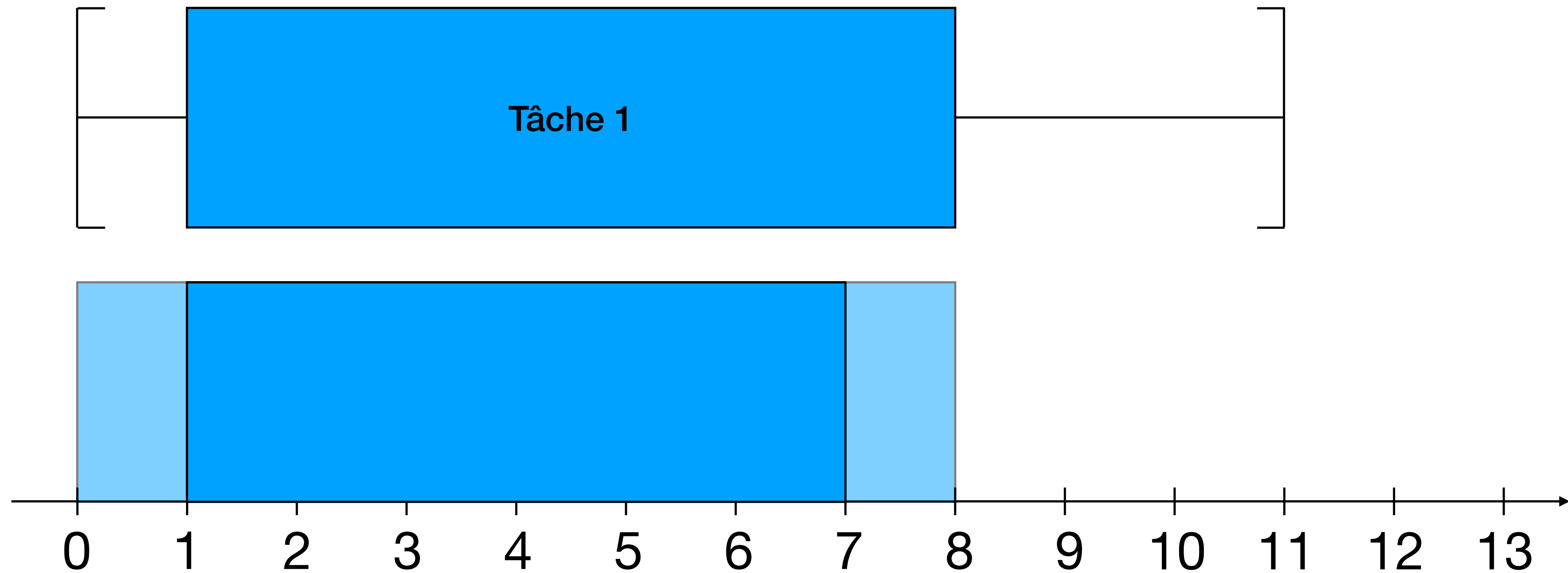
Profil de Time-Tabling



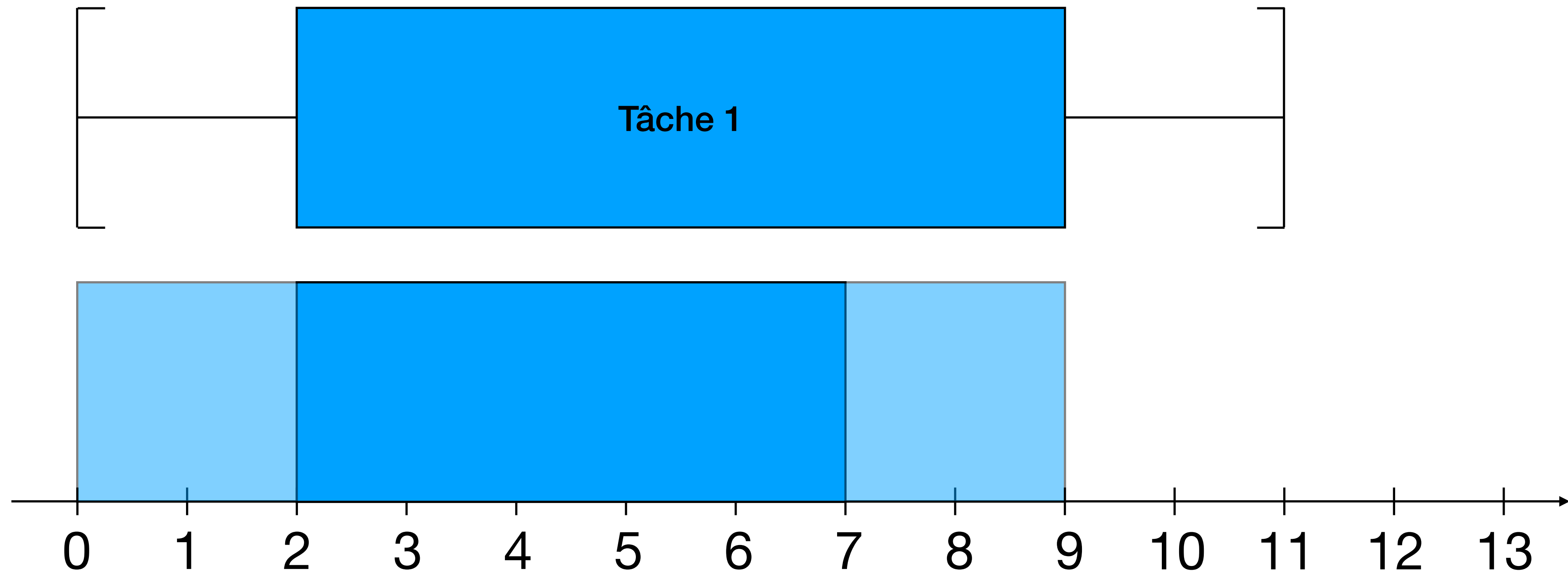
Profil de Time-Tabling



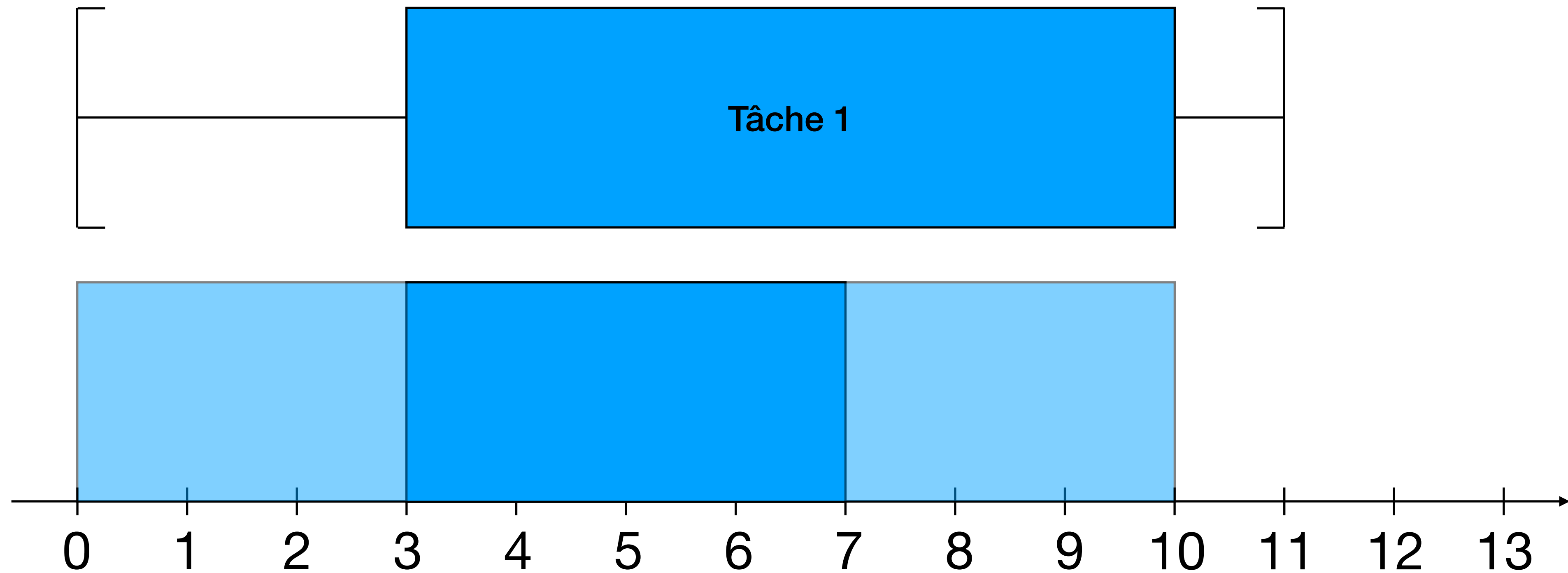
Profil de Time-Tabling



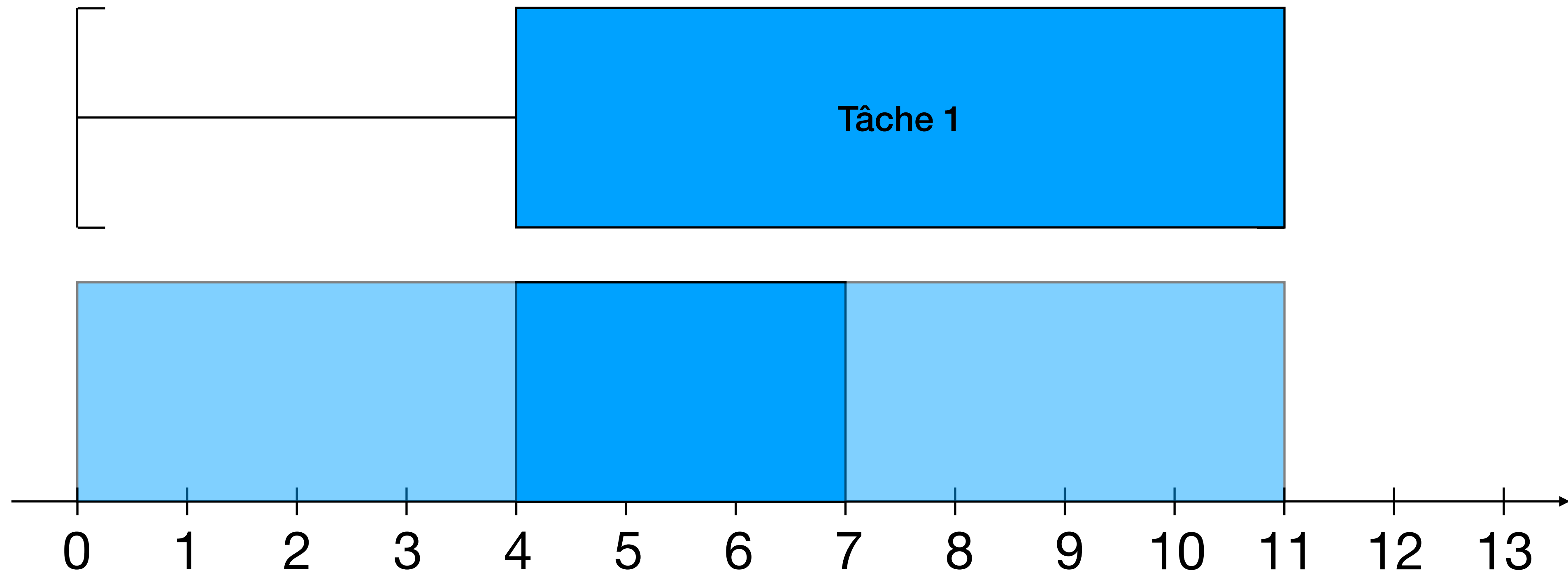
Profil de Time-Tabling



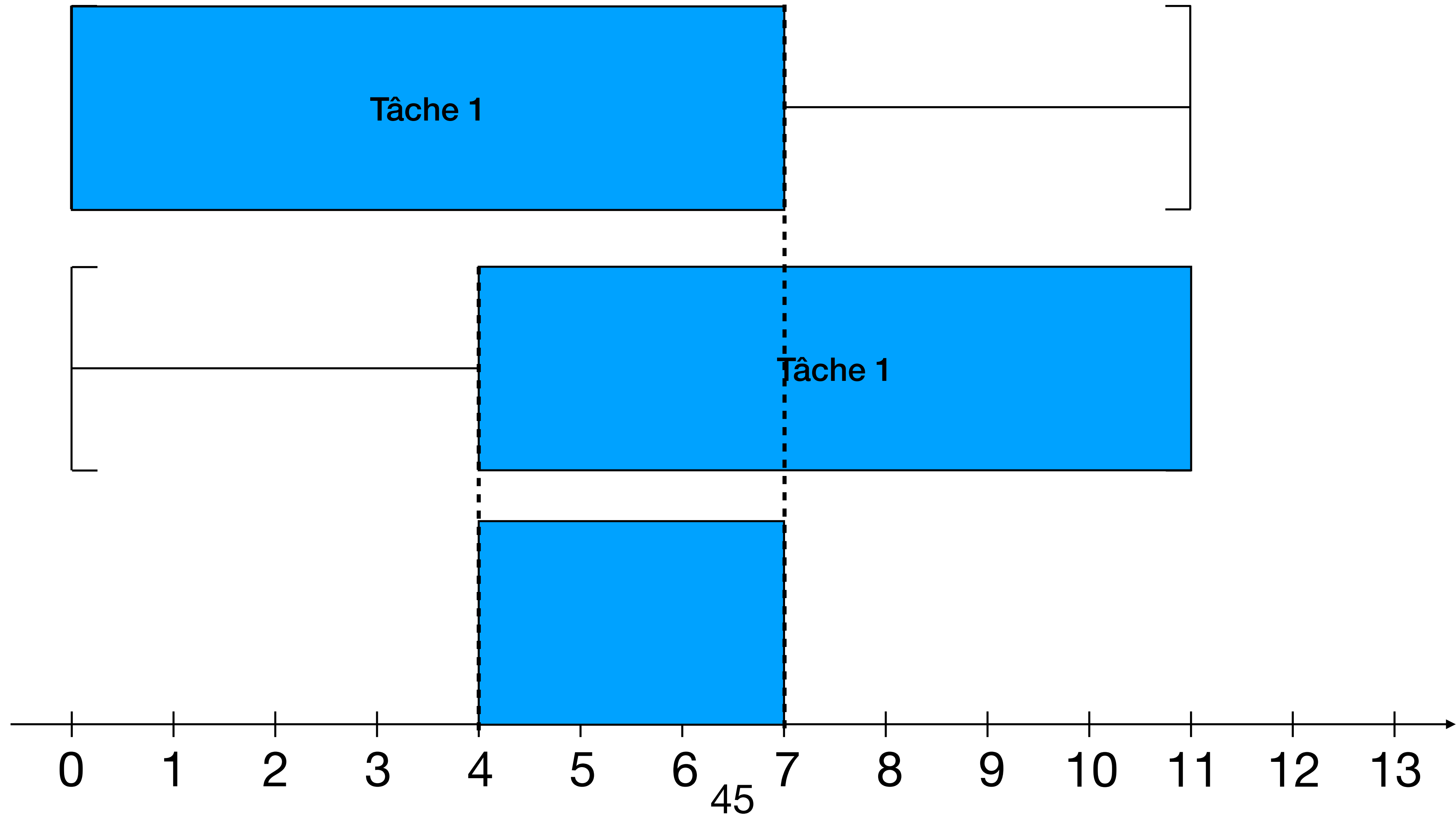
Profil de Time-Tabling



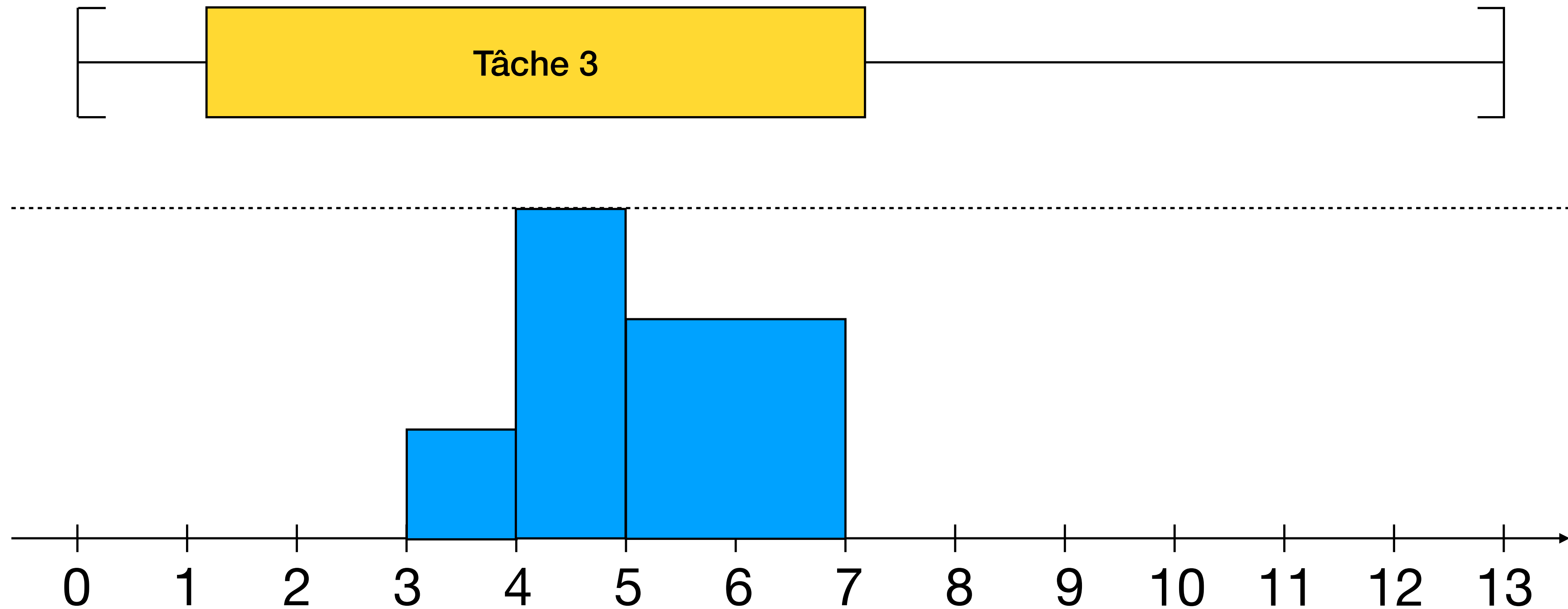
Profil de Time-Tabling



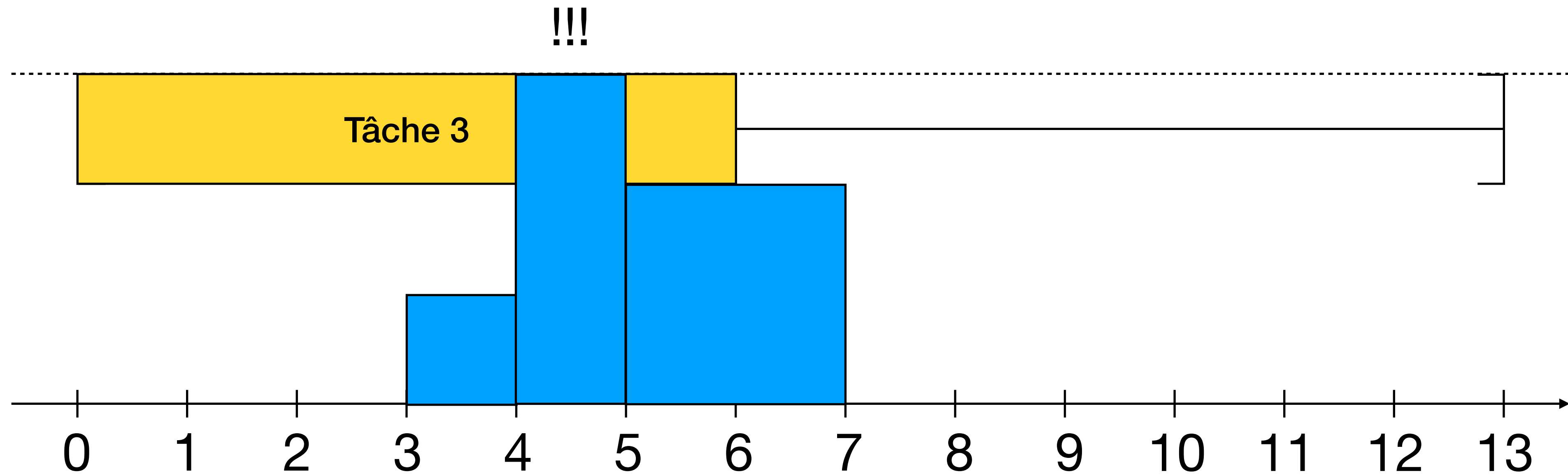
Profil de Time-Tabling



Filtrage avec Time-Tabling

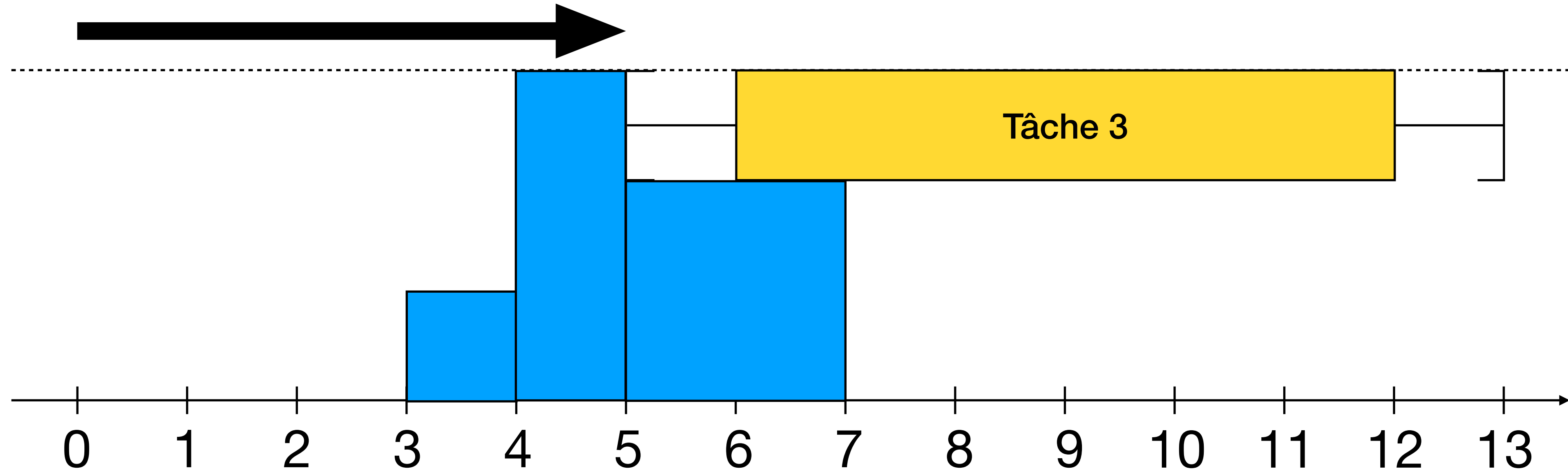


Filtrage avec Time-Tabling



Filtrage avec Time-Tabling

Filtrage du plus petit temps de début

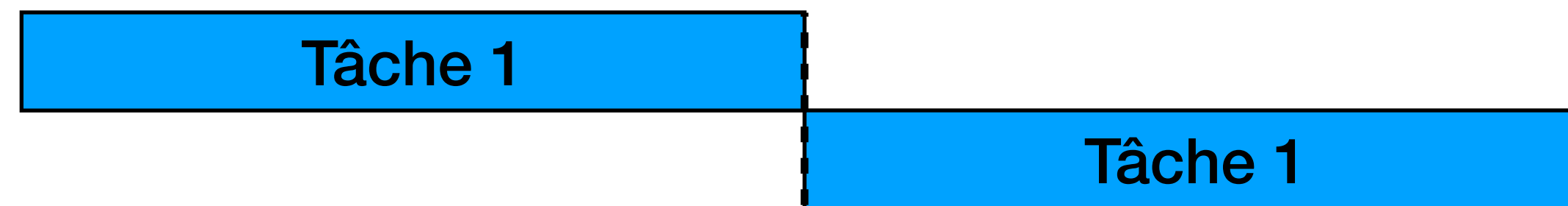


Nos contributions

Contrainte CUMULATIVEOVERTIME

- Introduit directement l'impact des calendriers dans le filtrage Time-Tabling

Sans
calendrier



Avec
calendrier

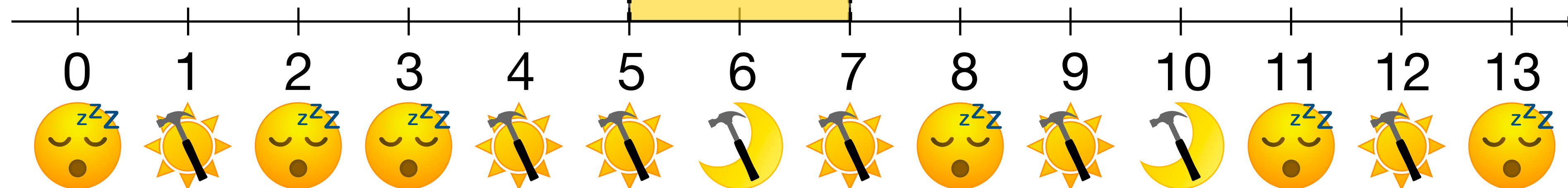


Durée de travail de 4

$$dom(T) = \{1..5\}$$

$$dom(D) = \{4..7\}$$

$$dom(S) = \{0..1\}$$



Nos contributions

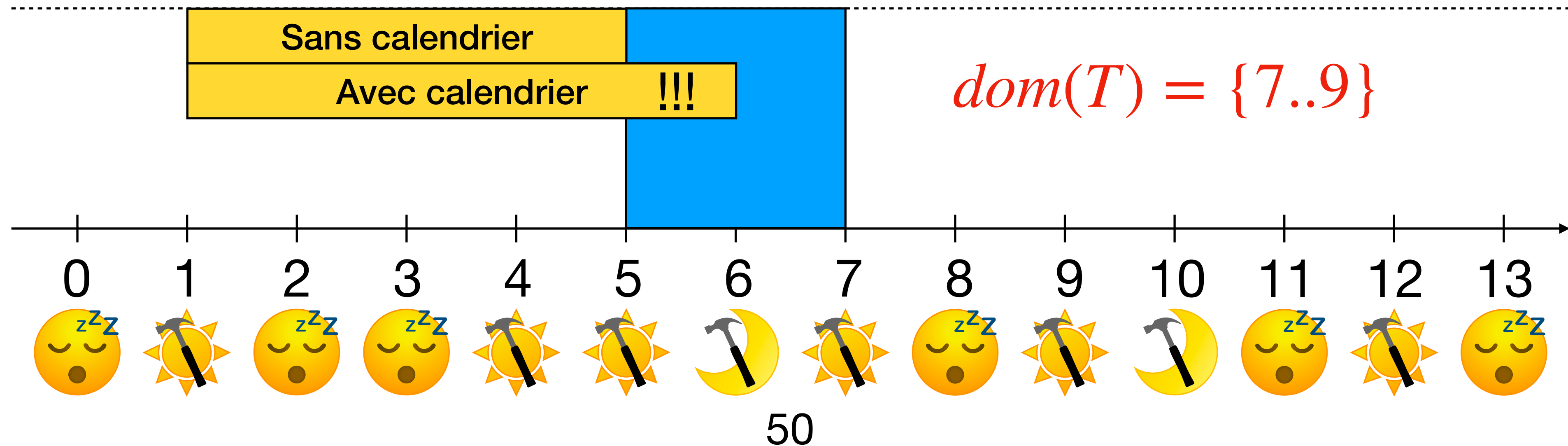
Contrainte CUMULATIVE OVERTIME

Durée de travail de 3

$$dom(T) = \{1..9\}$$

$$dom(D) = \{4..5\}$$

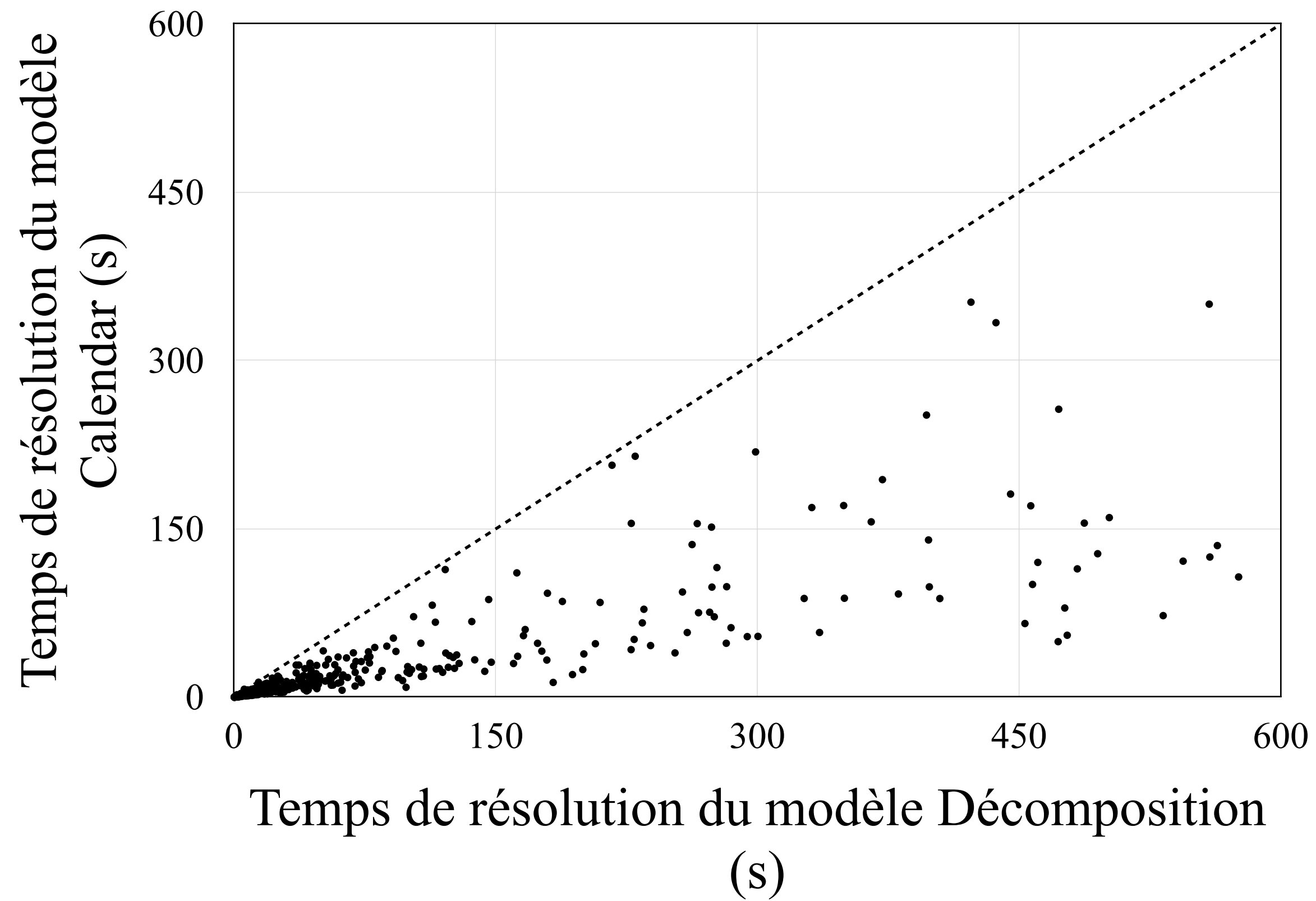
$$dom(S) = \{0..1\}$$



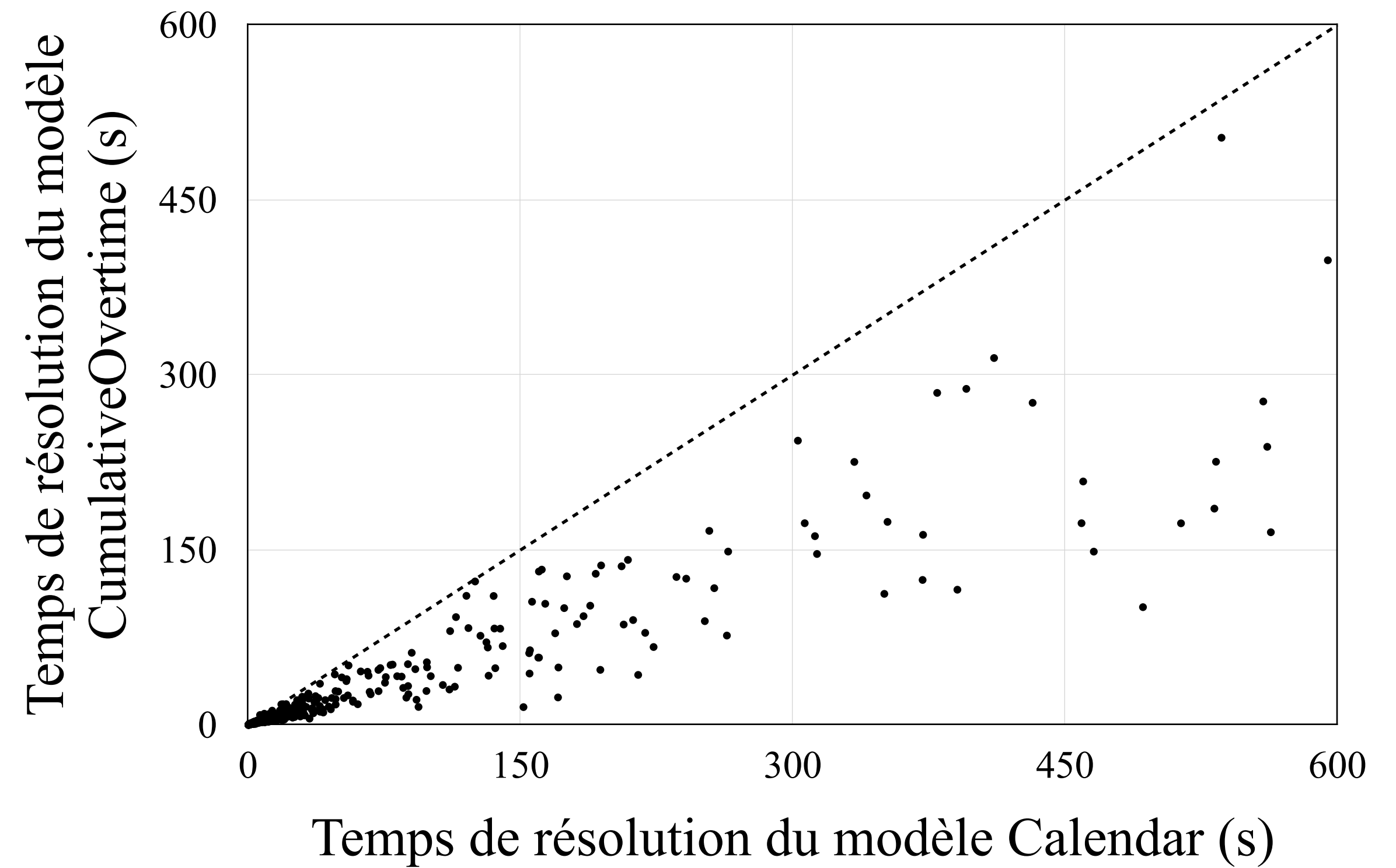
Résultats

Amélioration des temps de résolution

Minimisation des coûts de surtemps



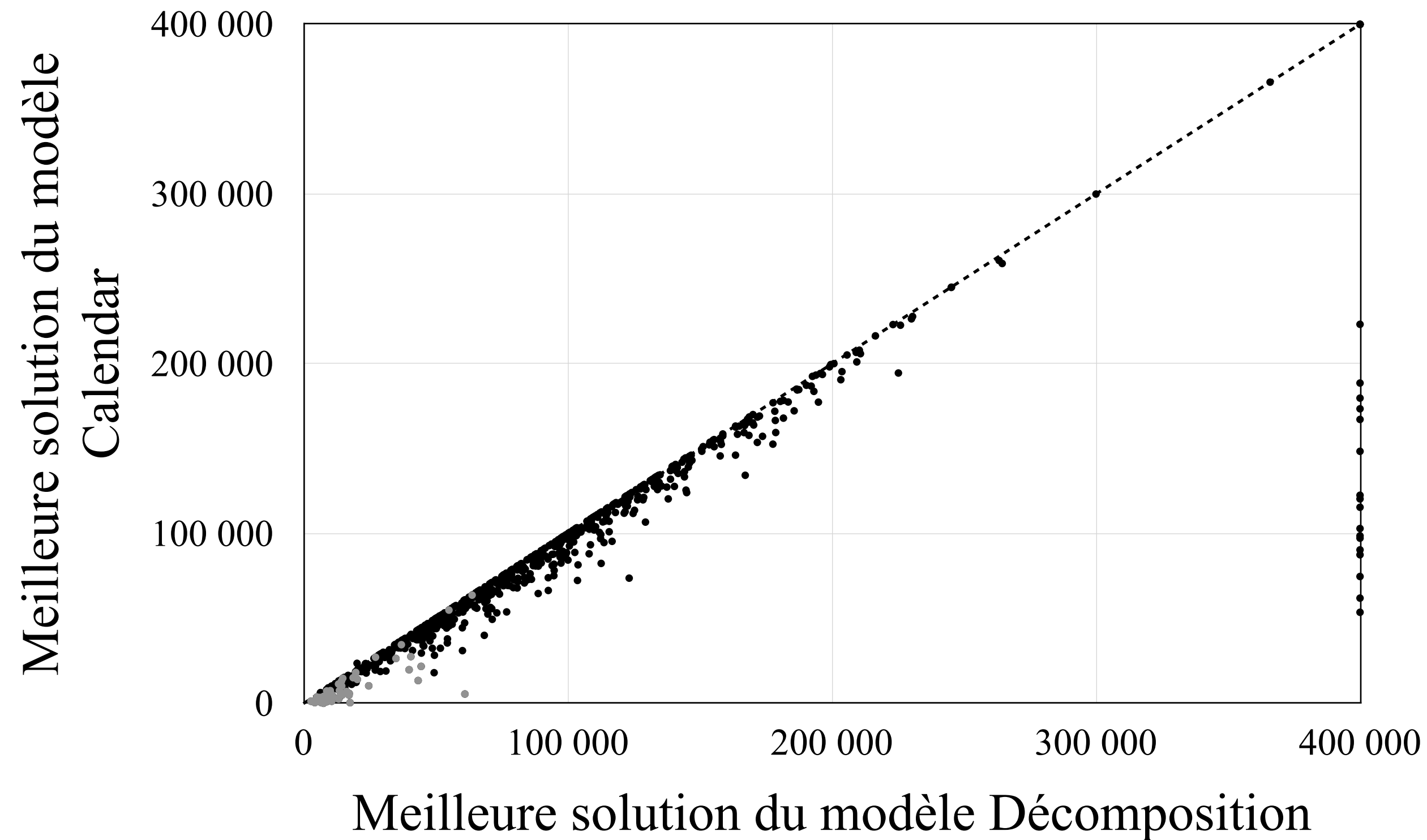
Minimisation des coûts de surtemps



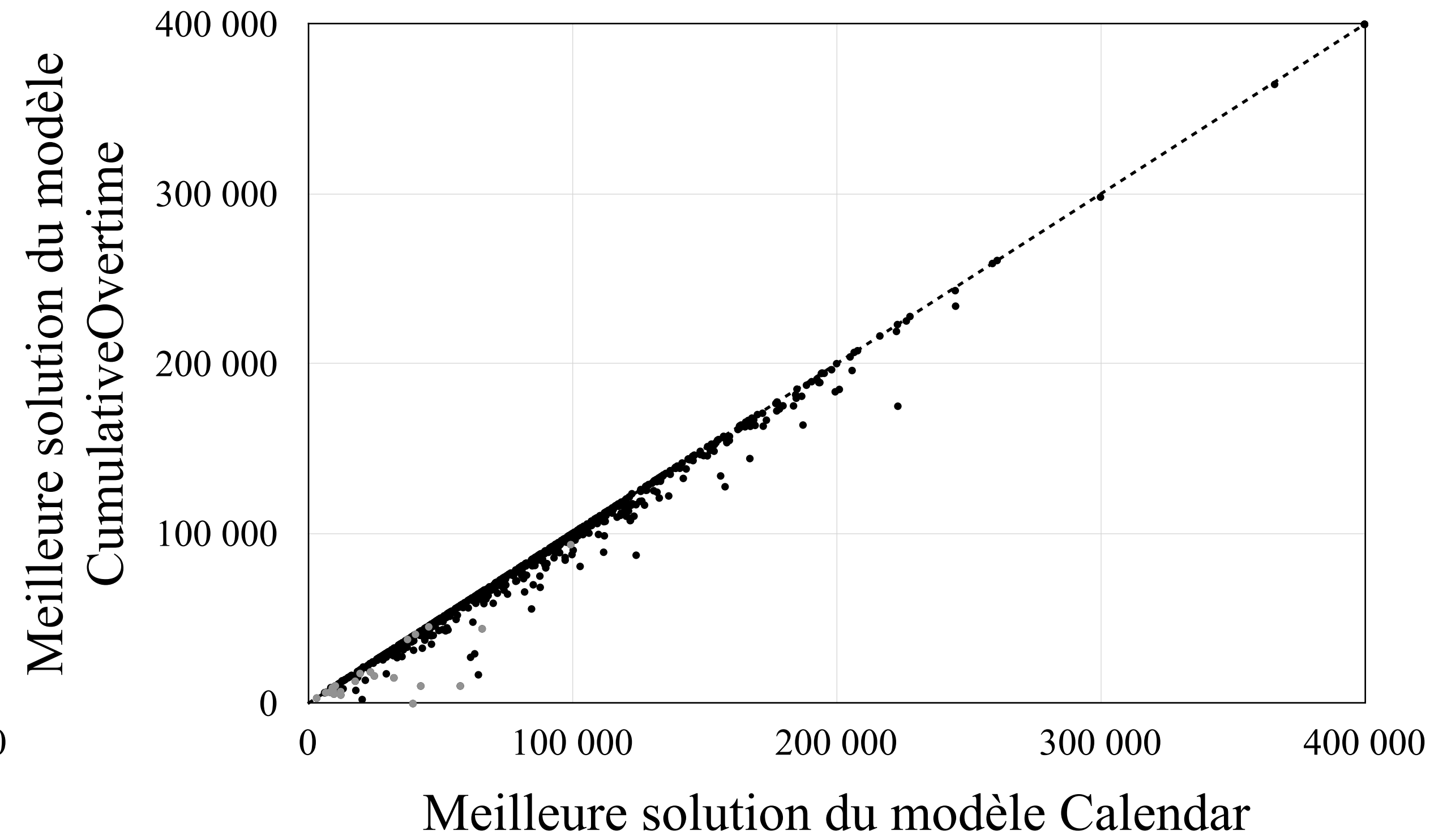
Résultats

Amélioration des solutions

Minimisation des coûts de surtemps



Minimisation des coûts de surtemps



Conclusion

- Implémentation de contraintes gérant les calendriers et le temps supplémentaire dans les problèmes d'ordonnancement
- Amélioration des performances des modèles associés