



Etude de l'algorithme GRASP pour le problème d'ordonnancement FJSSP-nfa

OS10 - A18

Adrien WARTELLE

Quoc Nhat Han TRAN

Imad BOUCETTA



Plan de présentation

1. Définition de Flexible Job Shop Scheduling Problem & FJSSP-nfa
2. L'algorithme GRASP & Application
3. Finalisation de solutions & Résultats



Flexible Job Shop Scheduling Problem

- Nombre de tâches et de machines sont spécifiés.
- Chaque tâche consiste d'une séquence d'opérations fixe.
- Chaque opération peut être traitée sur un ensemble de machines avec des différentes durées.
- Une machine ne peut exécuter qu'une opération à la fois.



FJSSP-nfa : non fixed availability

- Les tâches de maintenance préventive doivent être exécutées dans une fenêtre temporelle.
- Une opération interrompue par la maintenance, doit être redémarré quand la machine disponible.
- Une machine ne peut traiter qu'une tâche ou une opération de maintenance à la fois.



Objectif

- Minimisation de la date de fin de l'ordonnancement C_{\max} :

$$\min C_{\max} = \max_{1 \leq i \leq n} \{c_{i,n_i}\}$$

- Minimisation de la charge maximale d'occupation des machines W_{\max} :

$$\min W_{\max} = \max_{1 \leq j \leq m} \left\{ \sum_{i=1}^n \sum_{k=1}^{n_i} t_{ikj} x_{ikj} + \sum_{l=1}^{L_j} p_{jl} \right\}$$

- Minimisation de la charge totale d'occupation des machines W_{tot} :

$$\min W_{\text{tot}} = \sum_{j=1}^m \sum_{i=1}^n \sum_{k=1}^{n_i} t_{ikj} x_{ikj} + \sum_{j=1}^m \sum_{l=1}^{L_j} p_{jl}$$



Objectif

- Minimisation d'une combinaison linéaire :

$$w_1 * W_{tot} + w_2 * W_{max} + w_3 * C_{max}$$

- w_1, w_2 et w_3 sont des paramètres donnés en fonction des préférences d'objectif avec

$$w_1 + w_2 + w_3 = 1$$



Complexité

- Le problème est NP-difficile.
- Le problème doit traiter l'affectation des opérations aux machines ainsi que leur séquençement.
- Les méthodes exactes par graphes disjonctifs sont limitées à 20 tâches et 10 machines
- Les métaheuristiques sont les plus adaptées pour traiter ce problème (i.e. hGA, GRASP ...)



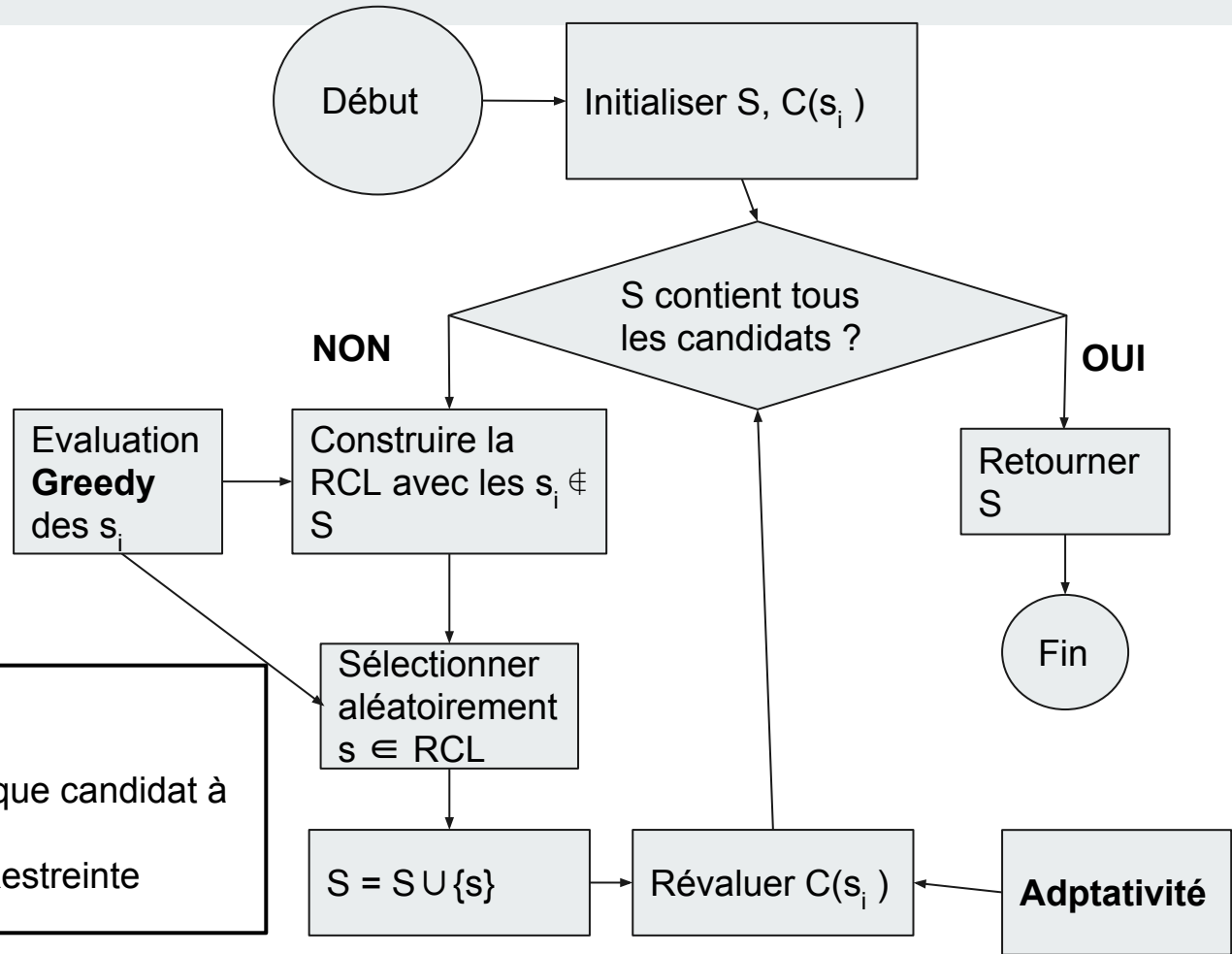
L'algorithme GRASP

- GRASP : Greedy Randomized Adaptive Search Procedure
- Metaheuristic avec **2 phases** : Construction itérative, Recherche locale.
- **Greedy** : utilisation d'une fonction d'évaluation pour l'ajout de candidats
- **Randomized** : choix randomisé des candidats
- **Adaptive** : mise à jour des coût d'ajouts à chaque itération
- **Search** : Recherche locale (randomisé) de solutions dans le voisinage de la solution d'une heuristique de base

Phase de construction

- itérative
- greedy
- randomisé

S : Solution
 s_i : Candidats
 $C(s_i)$: Coût d'ajout de chaque candidat à la solution S
RCL : Liste de Candidats Restreinte



Construction de la RCL

- OFV : W_{tot} (affectation),
 C_{max} (ordonnancement)
- Candidats : opérations

OFV($S \cup \{s_i\}$): Valeur de la fonction objective avec l'ajout de s_i
 α : "largeur" relative de la RCL entre 0 et 1
P(s_i) : probabilité de sélectionner le candidat i

Calcul des
 $C(s_i) = \text{OFV}(S \cup \{s_i\})$

Calcul de **Range** = $\text{OFV}_{\text{max}} - \text{OFV}_{\text{min}}$

Calcul de **Width** =
 $\text{Range} * \alpha$

$\text{RCL} = \{s_i, \text{OFV}(S \cup \{s_i\}) \in [\text{OFV}_{\text{min}}, \text{OFV}_{\text{min}} + \text{Width}]\}$

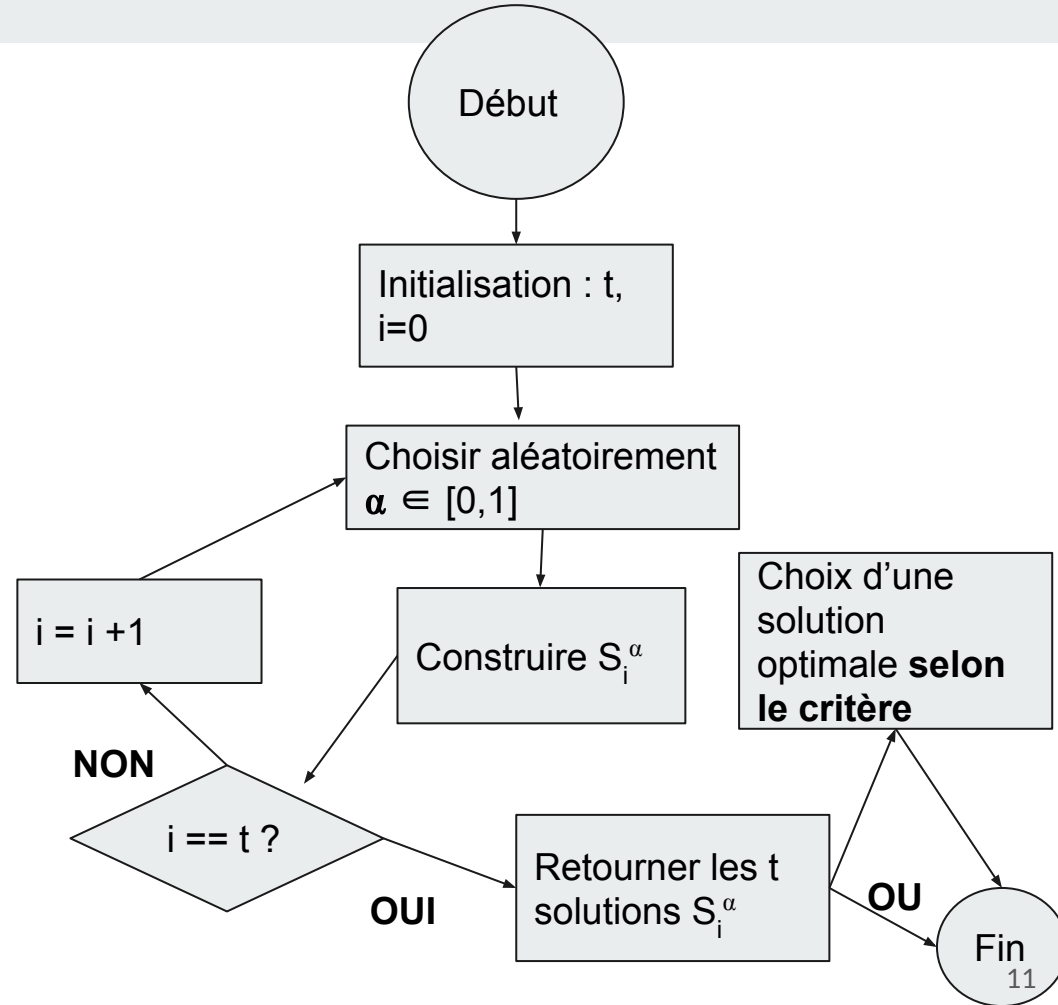
Rank (sort) candidates in RCL

$P(s_i) = (1/r) / \sum_{\text{RCL}} (1/r)$

Phase de recherche locale

-Choix de α

t : nombre d'itérations (de recherches)
 S_i^α : Solutions



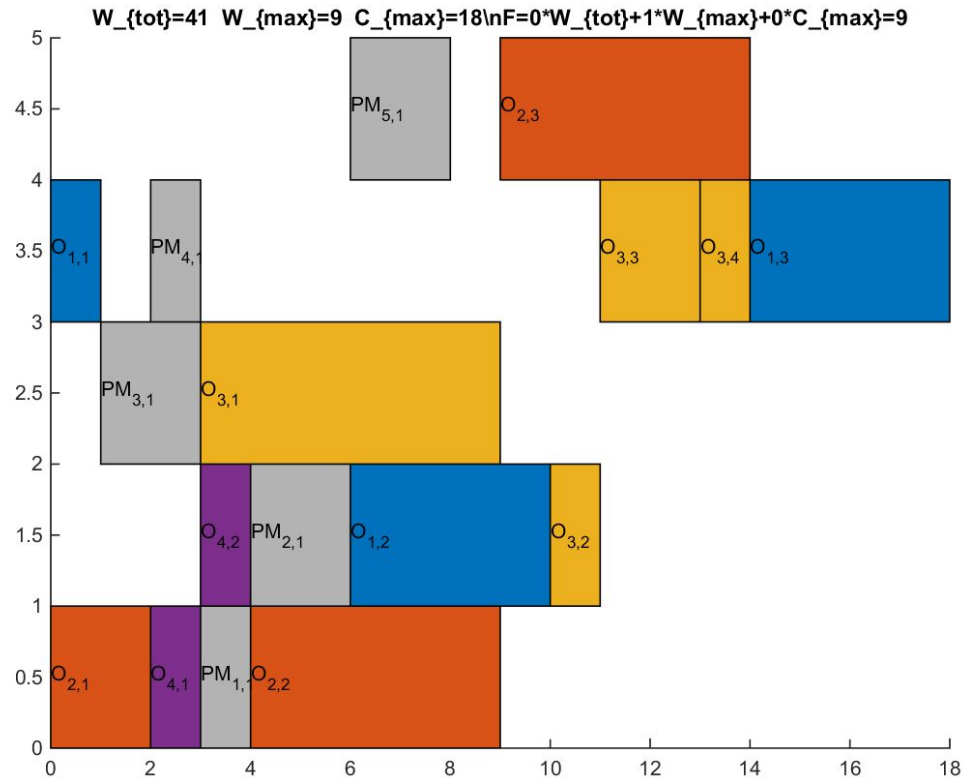


Finalisation de solution

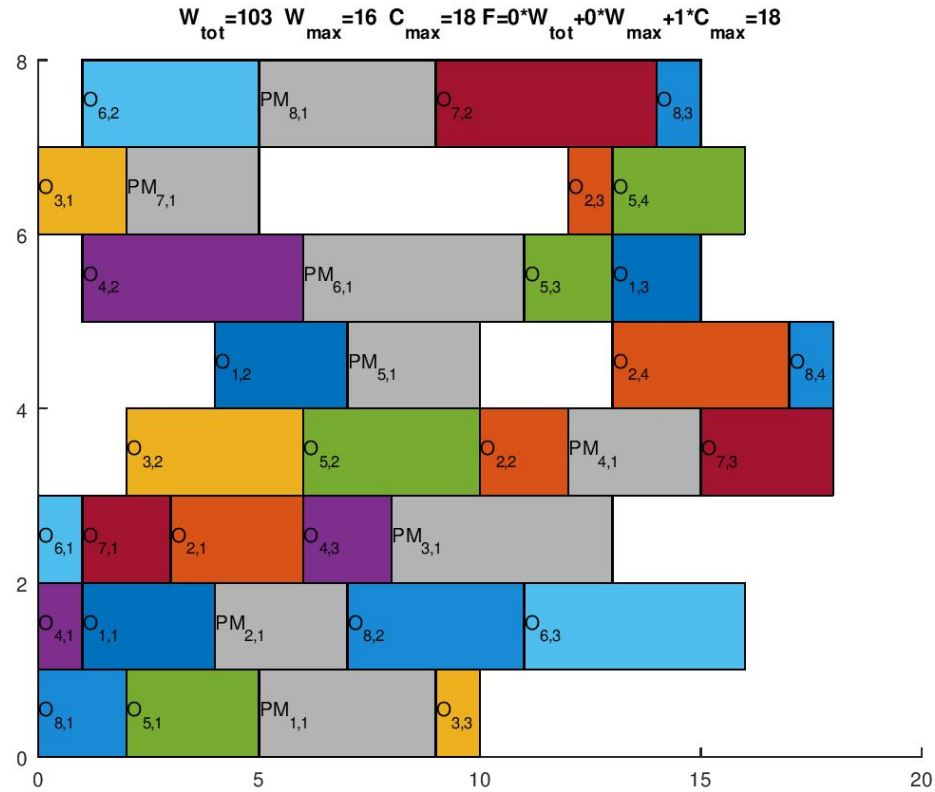
- A partir de l'ensemble des solutions construites, déterminer celle qui minimise $F(x,y,z)$ la somme linéaire de C_{\max} , W_{\max} , W_t .

- Critères possibles :

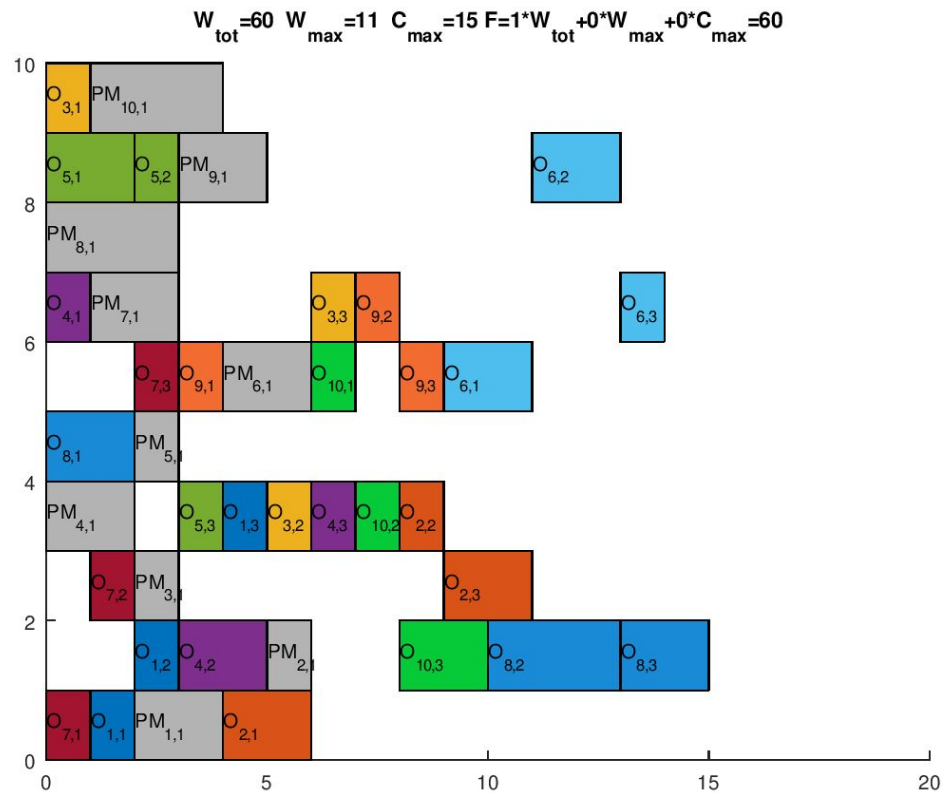
- + Min C_{\max} ($F(1,0,0)$)
- + Min W_{\max} ($F(0,1,0)$)
- + Min W_t ($F(0,0,1)$)
- + Mélange. Ex : $F(0.5,0.3,0.2)$



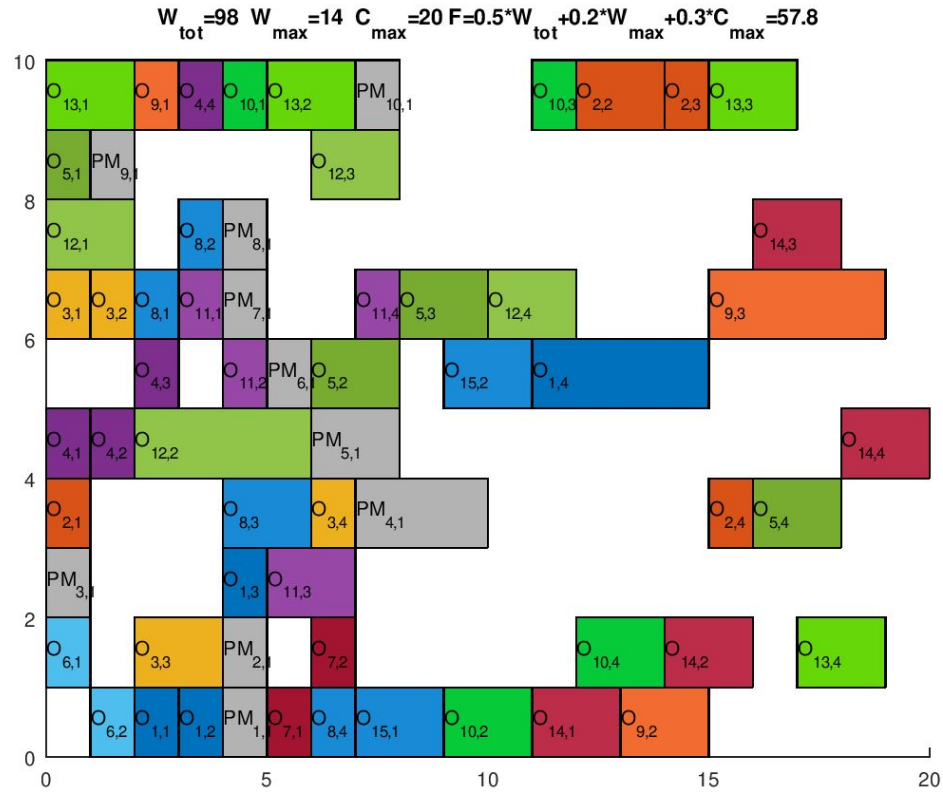
Un ordonnancement optimum local pour 4x5. Minimiser W_{max}



Un ordonnancement optimum local pour 8x8. Minimiser C_{max}



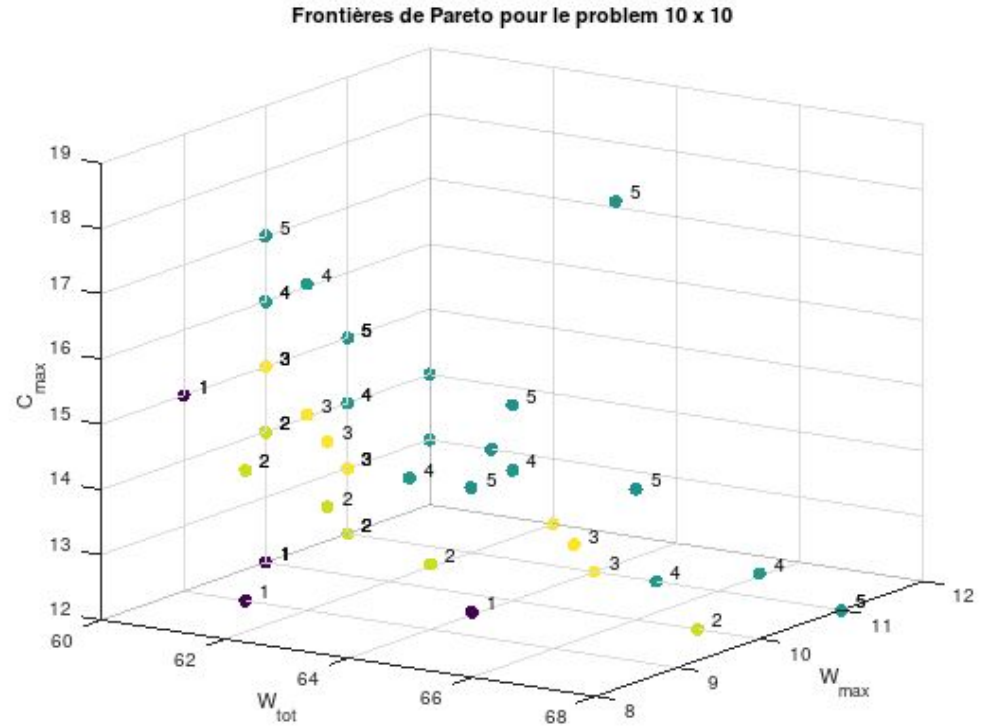
Un ordonnancement optimum local pour 10x10. Minimiser W_t



Un ordonnancement optimum local pour 15x10. Minimiser un mélange $F(0.5, 0.2, 0.3)$

Analyse du front Pareto

- Proches à la frontière
- Comparer les solutions
- Ressembler à un cône inversé





Nos résultats et comparaisons avec l'article

- 1000 itérations.
- Nos résultats s'approchent ceux de l'article \Rightarrow Reproductivité
- Temps d'exécution faible \Rightarrow Faisabilité & Efficacité

Critères	Valeurs de l'article	Nos valeurs
W_t	107	98
W_{max}	13	14
C_{max}	16	20
$F(0.5; 0.3; 0.2)$	60.3	57.2
$F(0.5; 0.2; 0.3)$	60.9	57.8

TABLE 3 – Résultat pour problème 15×10

Problème	Temps d'exécution (ms)
4×5	28.6767
8×8	150.71
10×10	427.403
15×10	1661.43

TABLE 5 – Temps d'exécution de chaque problème exemplaire