



Le cuisinier pressé

par **Adrien Krähenbühl**, Julien Narboux
et Basile Sauvage

Webinaire "Saveurs Savantes"
Le lundi 22 juin 2026

Public cible & compétences visées

- ▶ Public cible : **fin collège et lycée**
- ▶ Séance de **45-60 minutes**
- ▶ Organisation : idéalement en binôme
- ▶ **Compétences** travaillées :
 - Modéliser un problème
 - Exécuter un algorithme



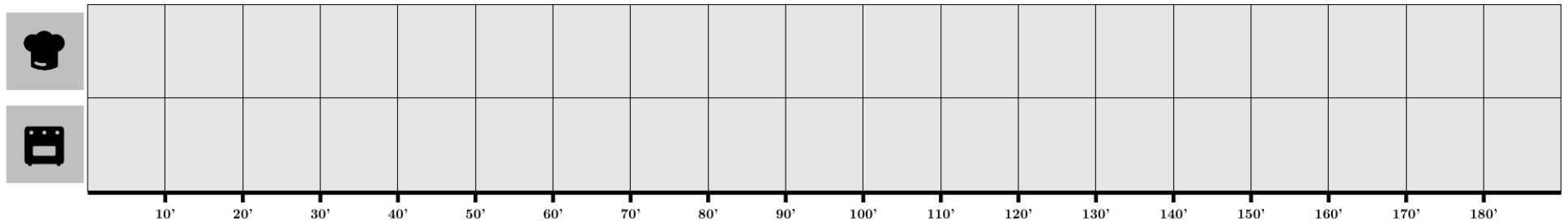
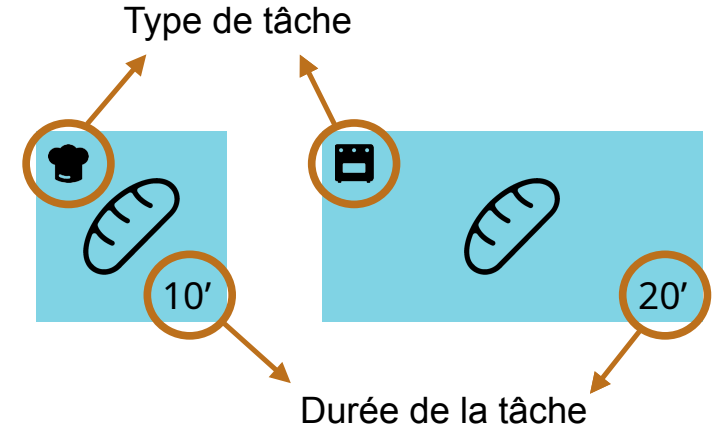
Le matériel

Notre cuisinier a **un ensemble de plat à préparer**.

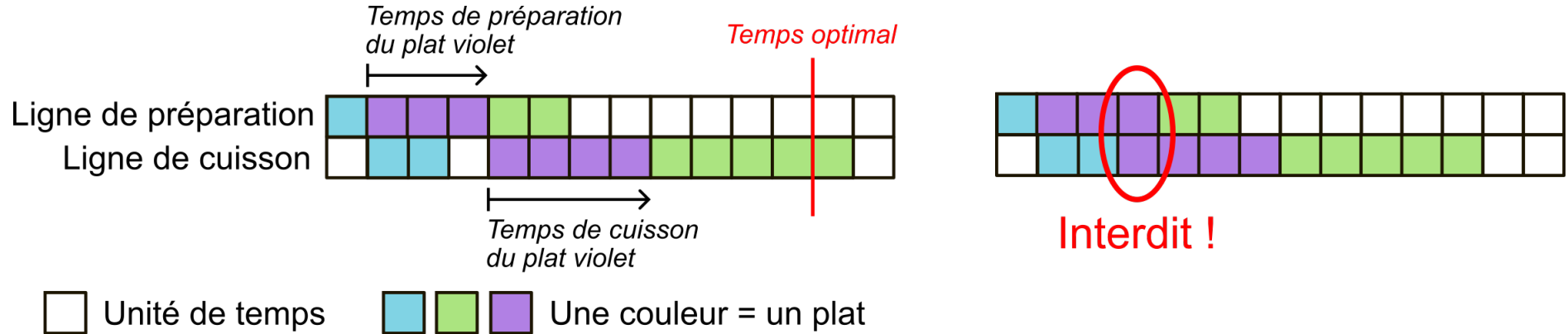
Pour chaque plat, il y a **deux tâches** à réaliser :

 La **préparation** en un certain temps **P**,

 La **cuisson** en un certain temps **C**.



Contraintes et objectif



L'objectif est de **trouver l'ordre des tâches qui permet de terminer l'ensemble des plats en temps minimum.**



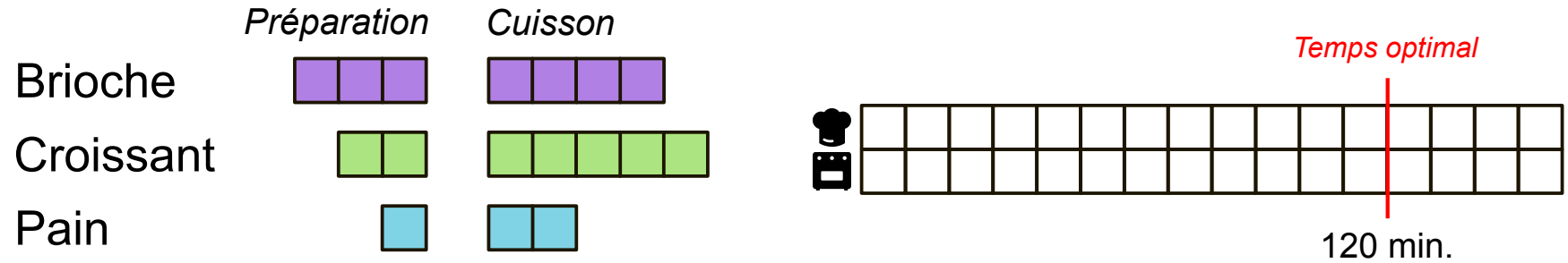
Trois **contraintes** :

- Le cuisinier ne peut préparer qu'un seul plat à la fois,
- Le four ne peut cuire qu'un seul plat à la fois,
- Un plat ne peut cuire qu'une fois sa préparation terminée.



Déroulé de l'activité

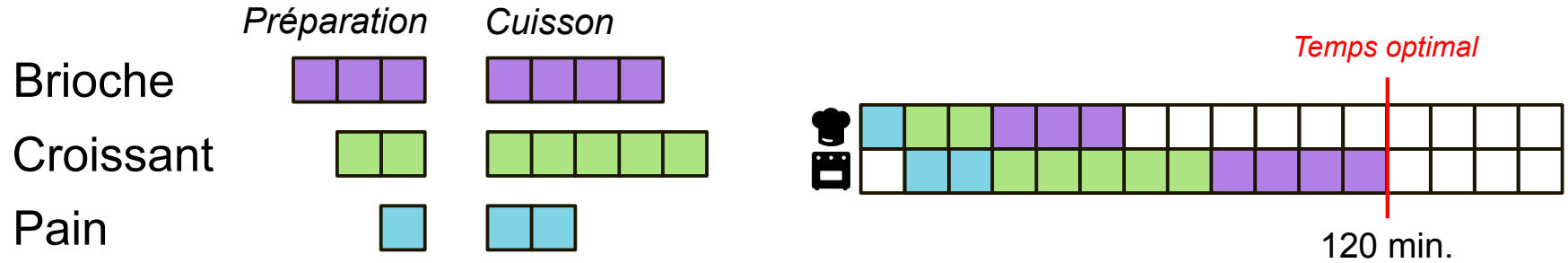
1^{er} défi – La boulangerie



Objectif : préparer les **3 plats** en **120 minutes**.



1^{er} défi – La boulangerie



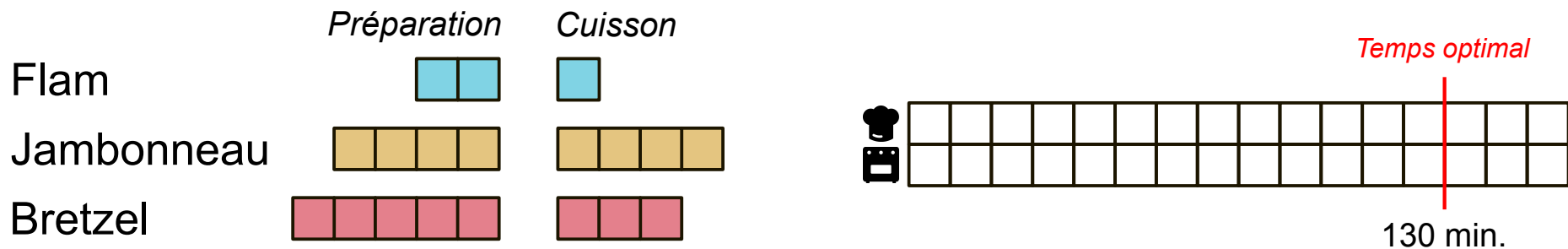
Objectif : préparer les 3 plats en 120 minutes.



Spécificité de l'ordre qui minimise le temps :
Par temps de préparation croissant.



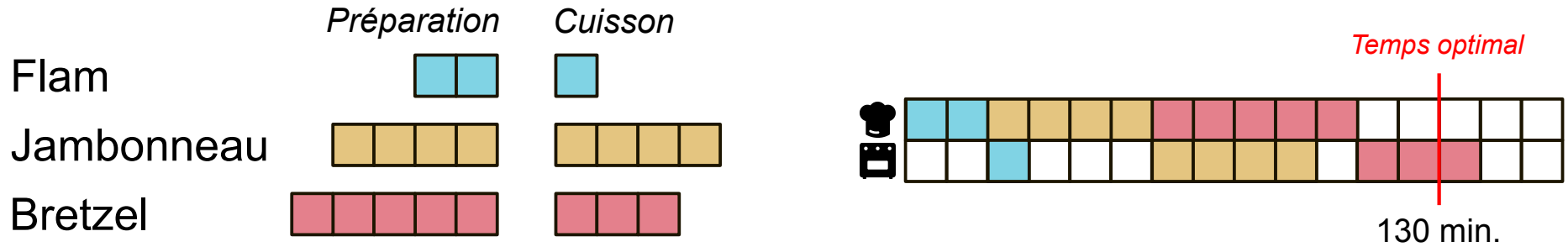
2^{ème} défi – Le restaurant alsacien



Objectif : préparer les **3 plats** en **130 minutes**.



2^{ème} défi – Le restaurant alsacien



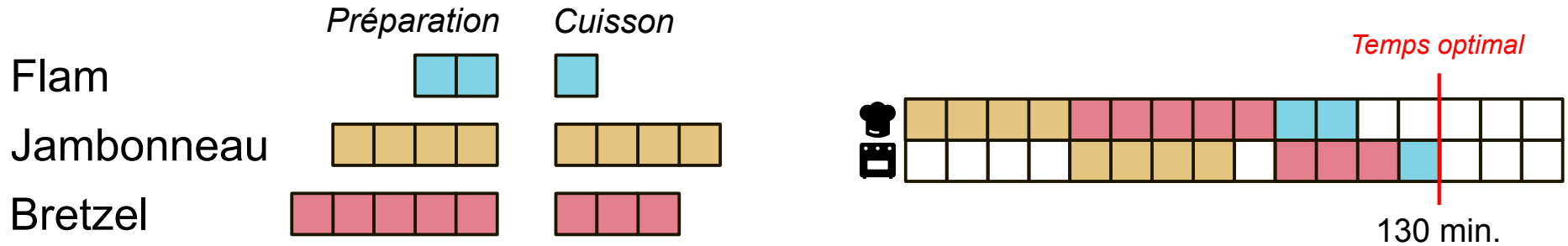
Objectif : préparer les 3 plats en 130 minutes.



La solution du 1er défi ne fonctionne pas.



2^{ème} défi – Le restaurant alsacien



Objectif : préparer les 3 plats en 130 minutes.



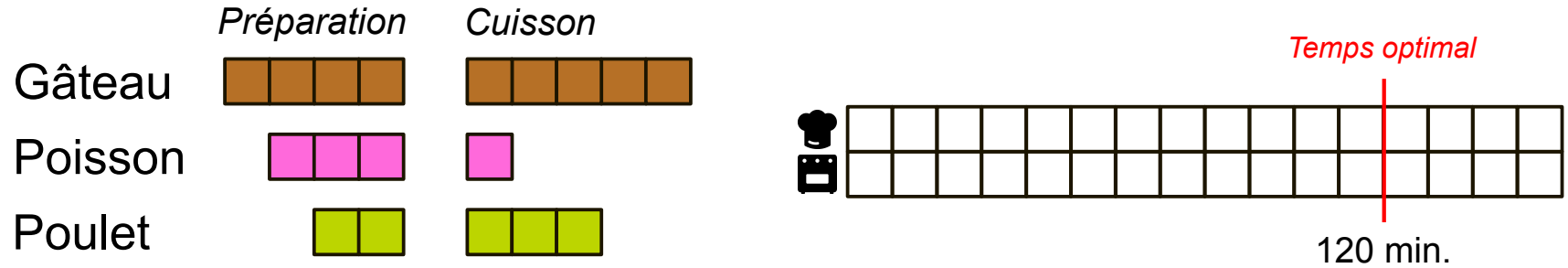
La solution du 1er défi ne fonctionne pas.



Spécificité de l'ordre qui minimise le temps :
Par **temps de cuisson décroissant**.



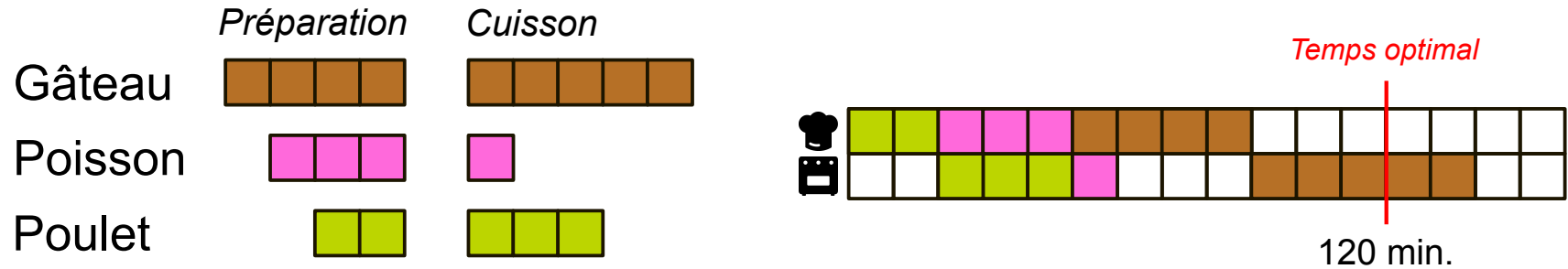
3^{ème} défi – La cantine scolaire



Objectif : préparer les **3 plats** en **120 minutes**.



3^{ème} défi – La cantine scolaire



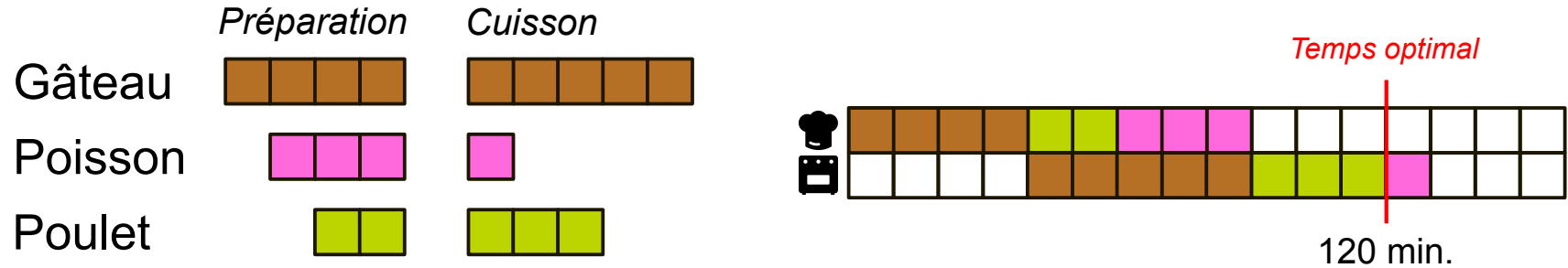
Objectif : préparer les 3 plats en 120 minutes.



La solution du 1er défi ne fonctionne pas.



3^{ème} défi – La cantine scolaire



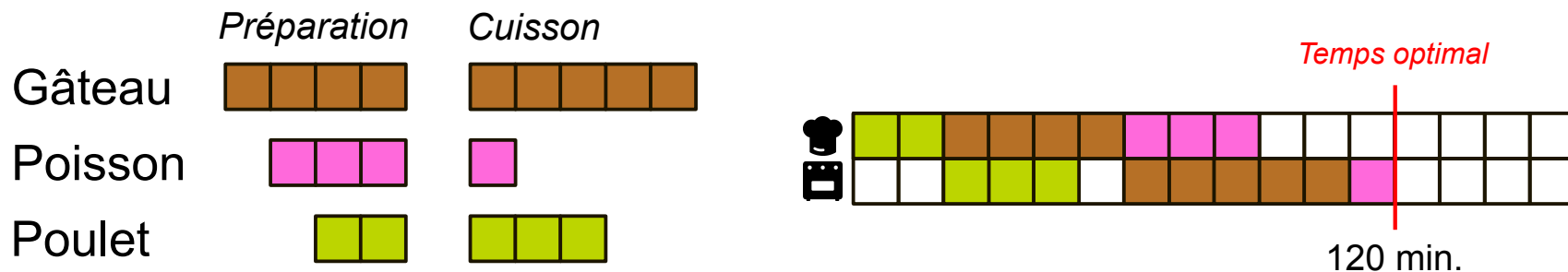
Objectif : préparer les 3 plats en 120 minutes.



La solution du 1er défi ne fonctionne pas... ni celle du 2ème.



3^{ème} défi – La cantine scolaire



Objectif : préparer les 3 plats en 120 minutes.



La solution du 1er défi ne fonctionne pas... ni celle du 2ème.



Spécificité de l'ordre qui minimise le temps :
Rien de notable...



4^{ème} défi – Au four et au moulin

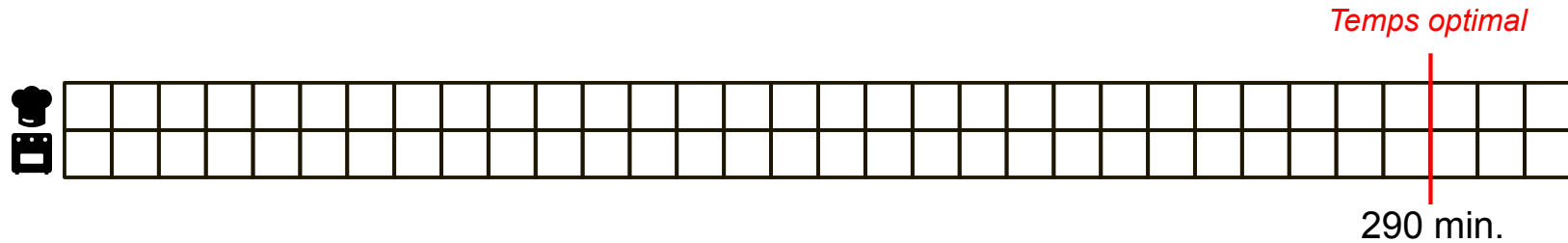
	<i>Préparation</i>	<i>Cuisson</i>
Gâteau		
Jambonneau		
Brioche		
Croissant		
Pain		
Flam		
Poisson		
Poulet		
Bretzel		



Objectif : préparer les **9 plats** en **290 minutes**.



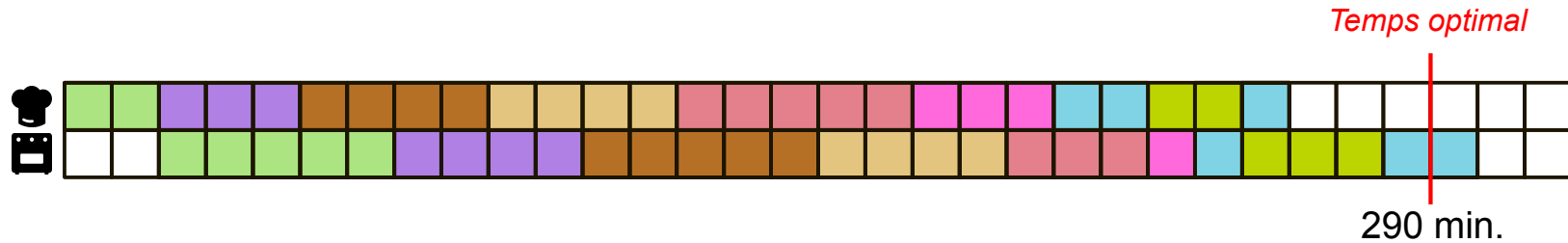
4^{ème} défi – Au four et au moulin



Objectif : préparer les 9 plats en 290 minutes.



4^{ème} défi – Au four et au moulin



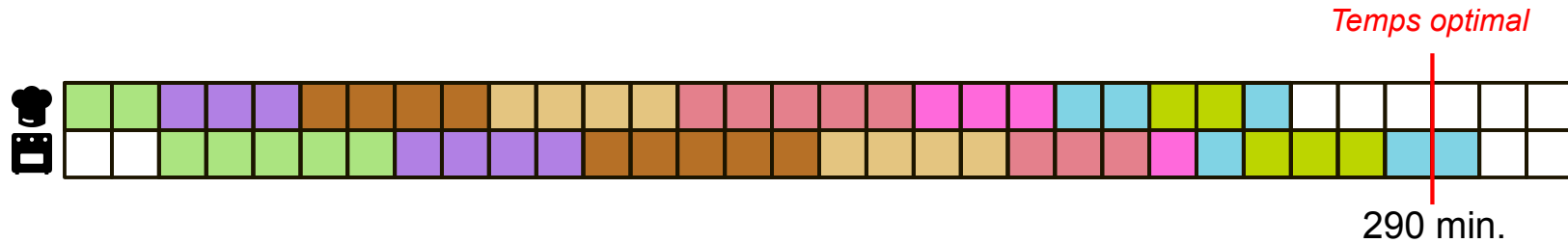
Objectif : préparer les 9 plats en 290 minutes.



Recherche fastidieuse, pas de stratégie évidente.



4^{ème} défi – Au four et au moulin



Objectif : préparer les 9 plats en 290 minutes.



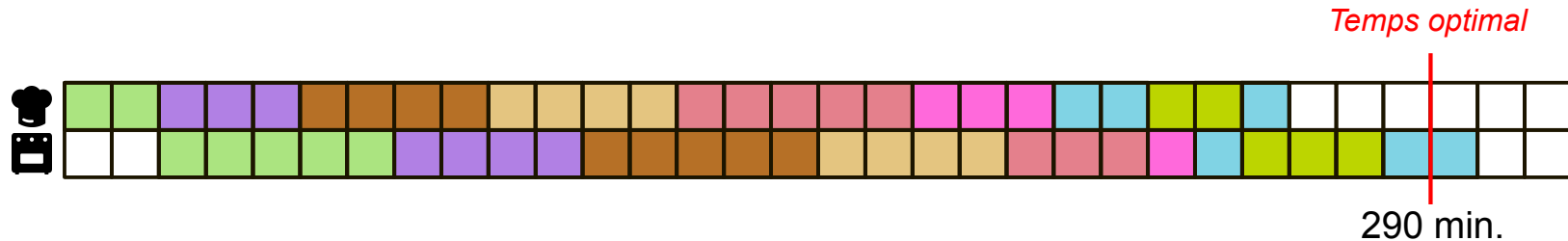
Recherche fastidieuse, pas de stratégie évidente.



Nombre de cas à tester : **$9! = 362\,880$**
(si même ordre pour préparation et cuisson)



4^{ème} défi – Au four et au moulin



Objectif : préparer les 9 plats en 290 minutes.



Recherche fastidieuse, pas de stratégie évidente.



Nombre de cas à tester : **9! = 362 880**
(si même ordre pour préparation et cuisson)



Existe-t-il un algorithme qui trouve une solution dans tous les cas ?



L'algorithme de Johnson

Deux façons d'exprimer l'algorithme :

Version A

Trier **toutes les tâches par ordre croissant** :

1. Si c'est une **préparation**, je place le plat associé (préparation+cuisson) **au début**,
2. Sinon, c'est une **cuisson**, je place le plat associé (préparation+cuisson) **à la fin**.



L'algorithme de Johnson

Deux façons d'exprimer l'algorithme :

Version A

Trier **toutes les tâches par ordre croissant** :

1. Si c'est une **préparation**, je place le plat associé (préparation+cuisson) **au début**,
 2. Sinon, c'est une **cuisson**, je place le plat associé (préparation+cuisson) **à la fin**.
-

Version B

Séparer **les plat en deux groupes** :

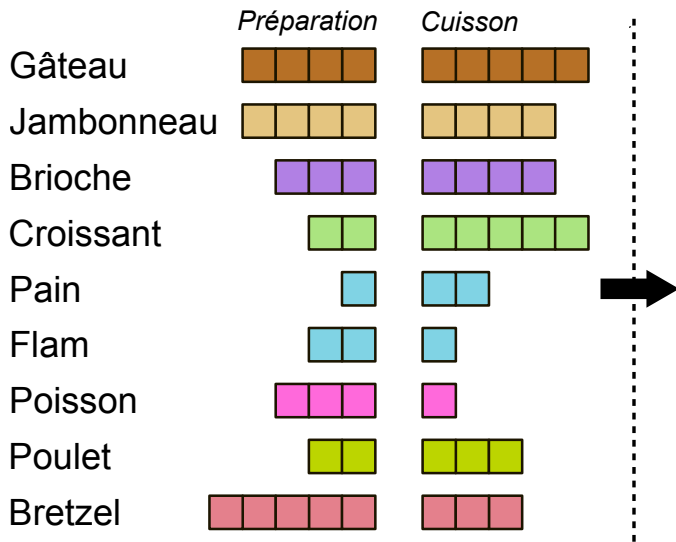
1. D'un côté les plats qui ont un **temps de préparation < au temps de cuisson**
 - ▶ Les trier par ordre croissant de temps de préparation.
2. De l'autre côté les plats qui ont un **temps de cuisson < au temps préparation**
 - ▶ Les trier par ordre décroissant de cuisson.

puis concaténer le groupe de plats **2** au groupe de plats **1**.



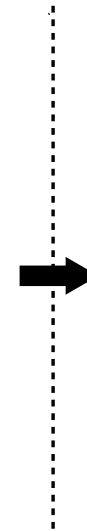
Déroulé de l'algorithme A de Johnson

Tâches à trier



Tâches triées

Préparation Cuisson



Ordre de Johnson

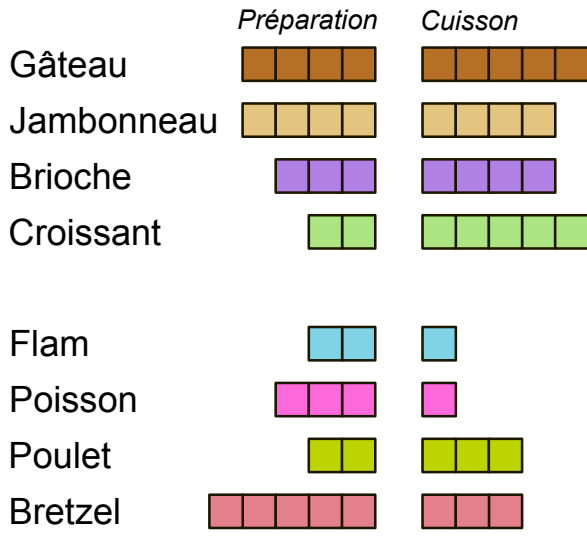
Préparation Cuisson

Trie des plats par ordre croissant des tâches.

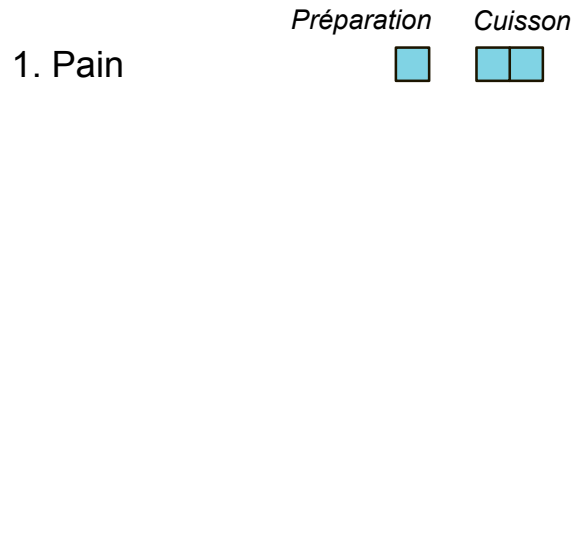


Déroulé de l'algorithme A de Johnson

Tâches à trier



Tâches triées



Ordre de Johnson

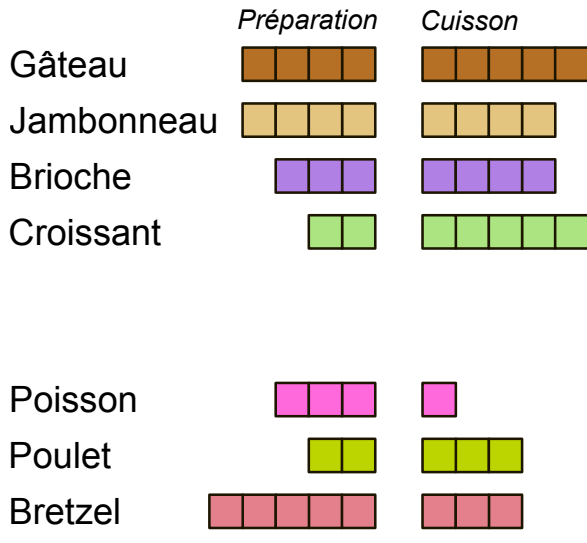


Possibilité de commencer par le **pain**, la **flam** ou le **poisson**.
(Tâches minimales de 10 minutes)

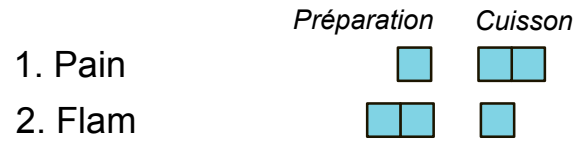


Déroulé de l'algorithme A de Johnson

Tâches à trier



Tâches triées



Ordre de Johnson

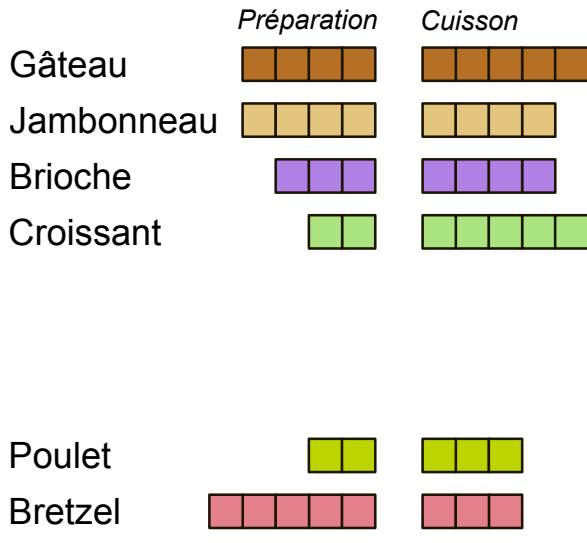
	Préparation	Cuisson
--	-------------	---------

Possibilité de commencer par le **pain**, la **flam** ou le **poisson**.
(Tâches minimales de 10 minutes)

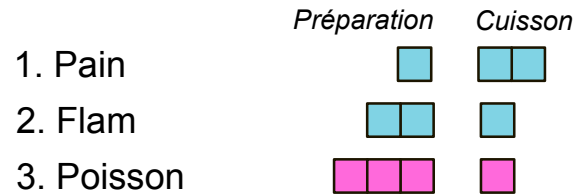


Déroulé de l'algorithme A de Johnson

Tâches à trier



Tâches triées



Ordre de Johnson

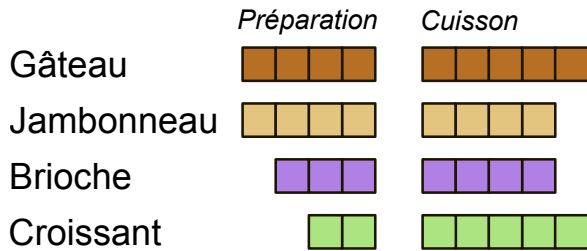
	Préparation	Cuisson
--	-------------	---------

Possibilité de commencer par le **pain**, la **flam** ou le **poisson**.
(Tâches minimales de 10 minutes)

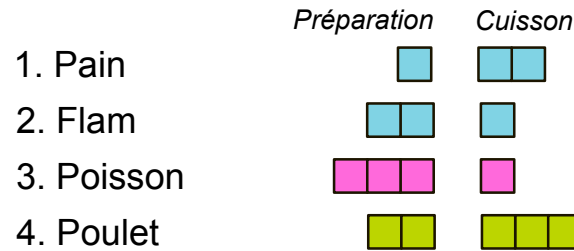


Déroulé de l'algorithme A de Johnson

Tâches à trier



Tâches triées



Ordre de Johnson

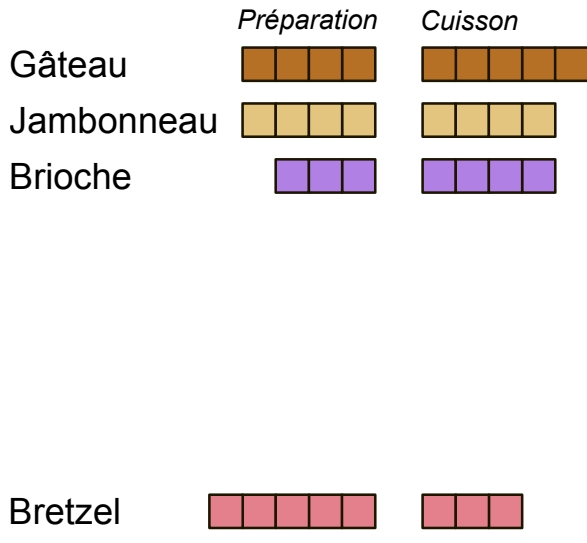
Préparation Cuisson

Possibilité de continuer par le **croissant** ou le **poulet**.
(Tâches minimales de 20 minutes)

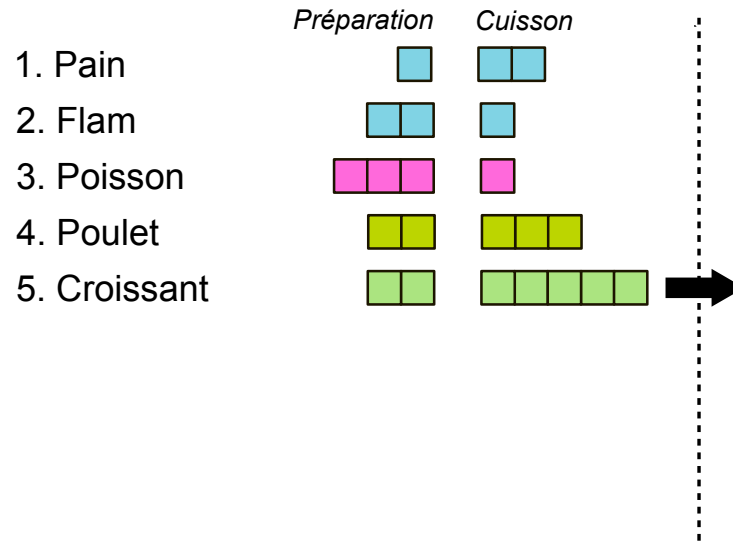


Déroulé de l'algorithme A de Johnson

Tâches à trier



Tâches triées



Ordre de Johnson

Préparation Cuisson

Possibilité de continuer par le **croissant** ou le **poulet**.
(Tâches minimales de 20 minutes)



Déroulé de l'algorithme A de Johnson

Tâches à trier

	Préparation	Cuisson
Gâteau		
Jamonneau		
Brioche		

Tâches triées

	Préparation	Cuisson
1. Pain		
2. Flam		
3. Poisson		
4. Poulet		
5. Croissant		
6. Bretzel		

Ordre de Johnson

Préparation	Cuisson
-------------	---------

Possibilité de continuer par la **brioche** ou la **bretzel**.
(Tâches minimales de 30 minutes)

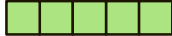


Déroulé de l'algorithme A de Johnson

Tâches à trier

	Préparation	Cuisson
Gâteau		
Jambonneau		

Tâches triées

	Préparation	Cuisson
1. Pain		
2. Flam		
3. Poisson		
4. Poulet		
5. Croissant		
6. Bretzel		
7. Brioche		

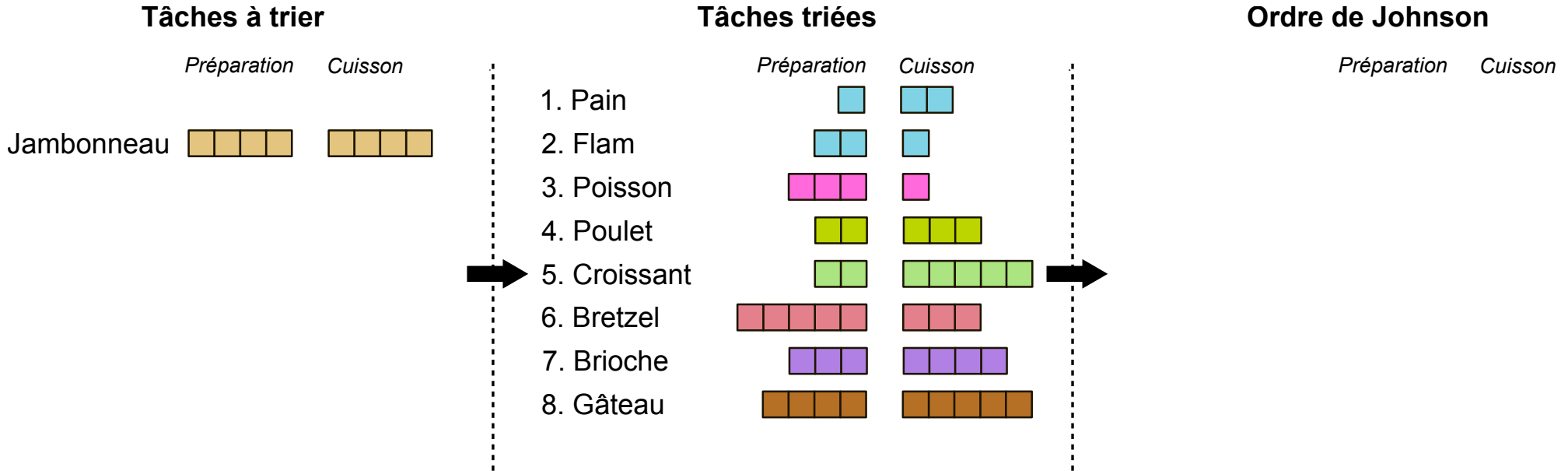
Ordre de Johnson

Préparation	Cuisson
-------------	---------

Possibilité de continuer par la **brioche** ou la **bretzel**.
(Tâches minimales de 30 minutes)



Déroulé de l'algorithme A de Johnson



On termine par le **jambonneau** ou le **gâteau**.
(Tâches minimales de 40 minutes)



Déroulé de l'algorithme A de Johnson

Tâches à trier

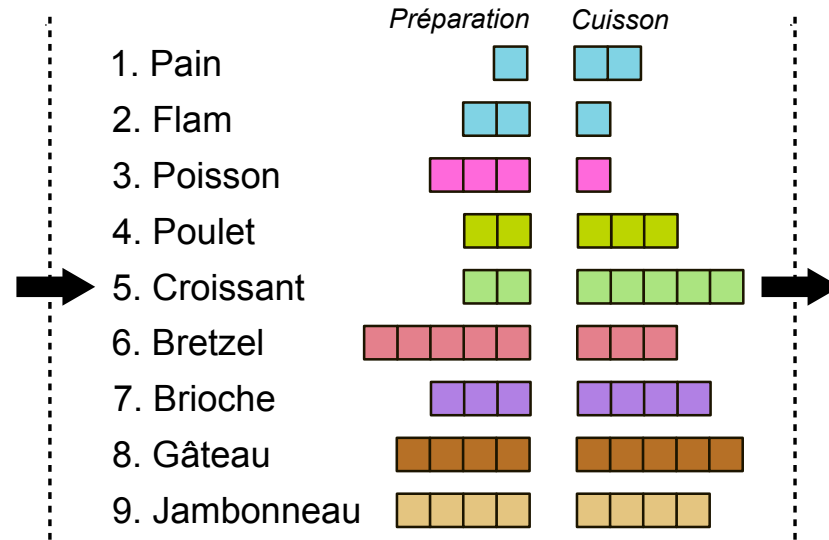
Préparation Cuisson

Tâches triées

Préparation Cuisson

Ordre de Johnson

Préparation Cuisson



On termine par le **jambonneau** ou le **gâteau**.
(Tâches minimales de 40 minutes)



Déroulé de l'algorithme A de Johnson

Tâches à trier

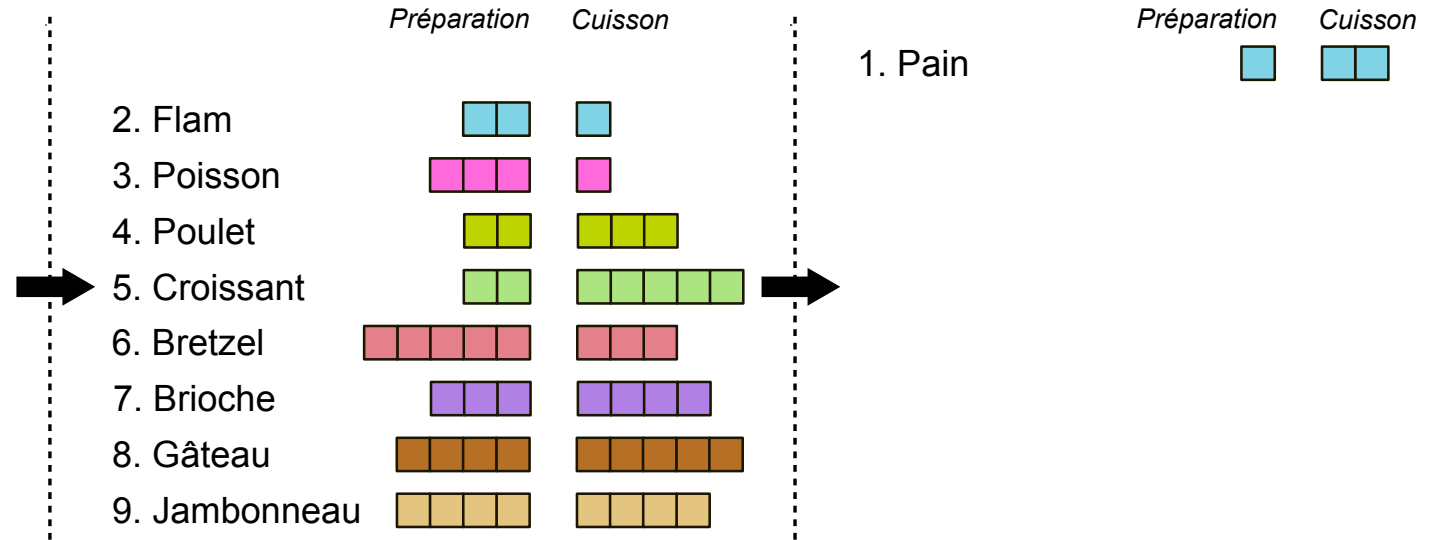
Préparation Cuisson

Tâches triées

Préparation Cuisson

Ordre de Johnson

Préparation Cuisson



Ordonnancement de Johnson

Le **pain** a un temps de **Préparation** < temps de **Cuisson** => **au début**.



Déroulé de l'algorithme A de Johnson

Tâches à trier

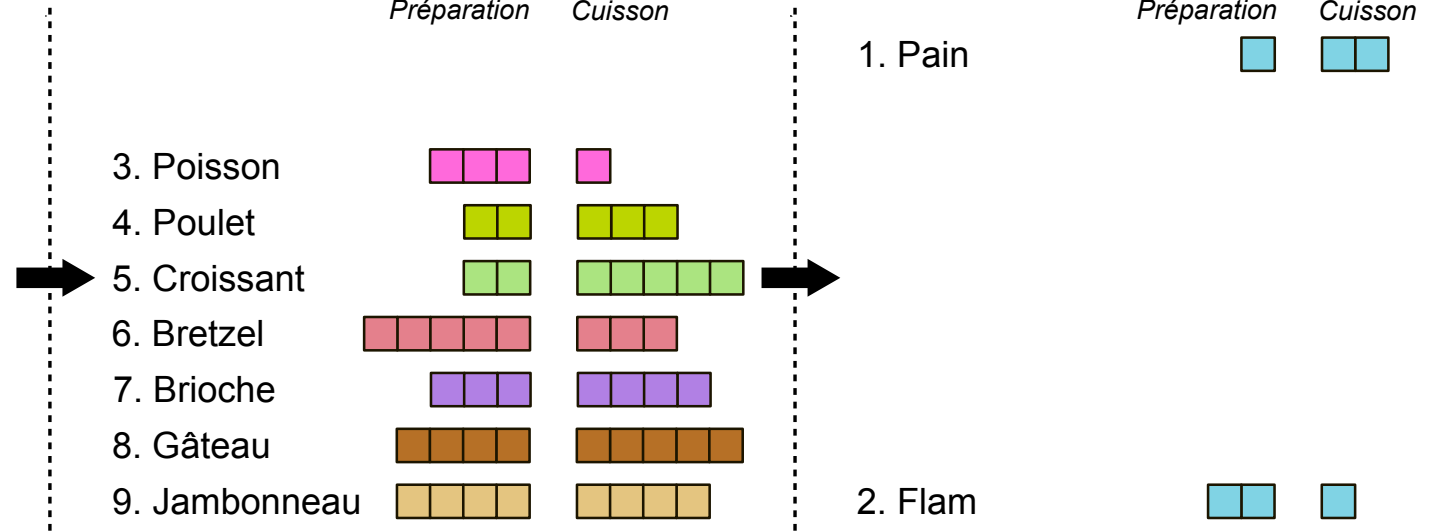
Préparation Cuisson

Tâches triées

Préparation Cuisson

Ordre de Johnson

Préparation Cuisson



Ordonnancement de Johnson

La **flam** a un temps de **Cuisson** < temps de **Préparation** => à la fin.



Déroulé de l'algorithme A de Johnson

Tâches à trier

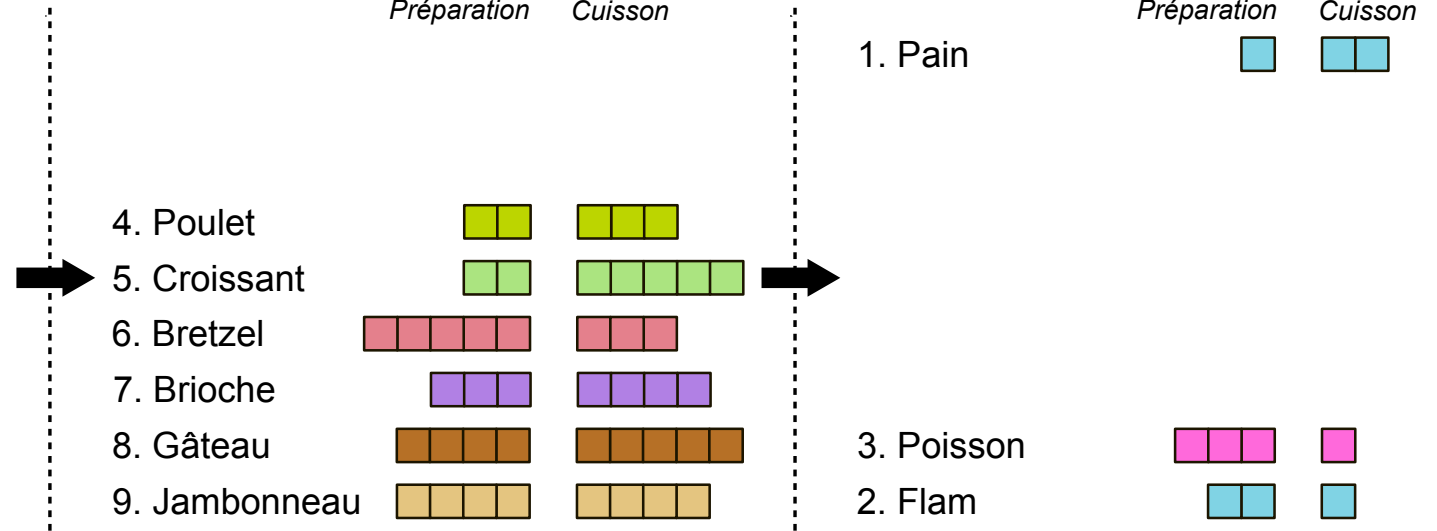
Préparation Cuisson

Tâches triées

Préparation Cuisson

Ordre de Johnson

Préparation Cuisson



Ordonnancement de Johnson

Le **poisson** a un temps de **Cuisson** < temps de **Préparation** => **à la fin**.



Déroulé de l'algorithme A de Johnson

Tâches à trier

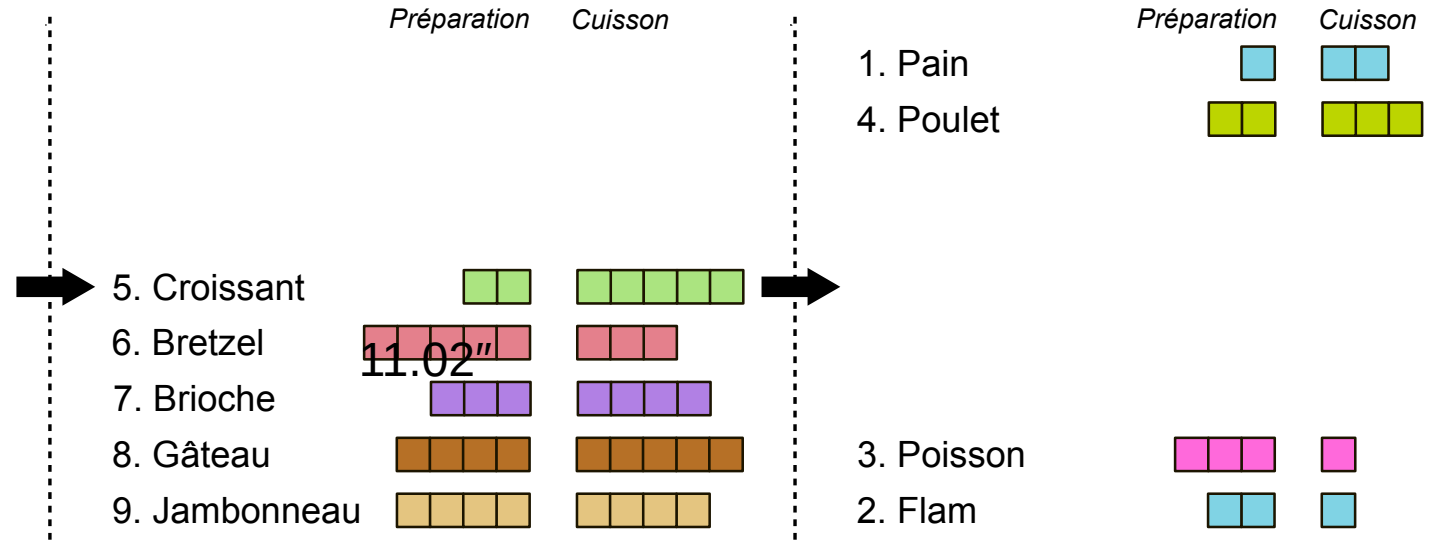
Préparation Cuisson

Tâches triées

Préparation Cuisson

Ordre de Johnson

Préparation Cuisson



Ordonnancement de Johnson

Le **poulet** a un temps de **Préparation** < temps de **Cuisson** => **au début**.



Déroulé de l'algorithme A de Johnson

Tâches à trier

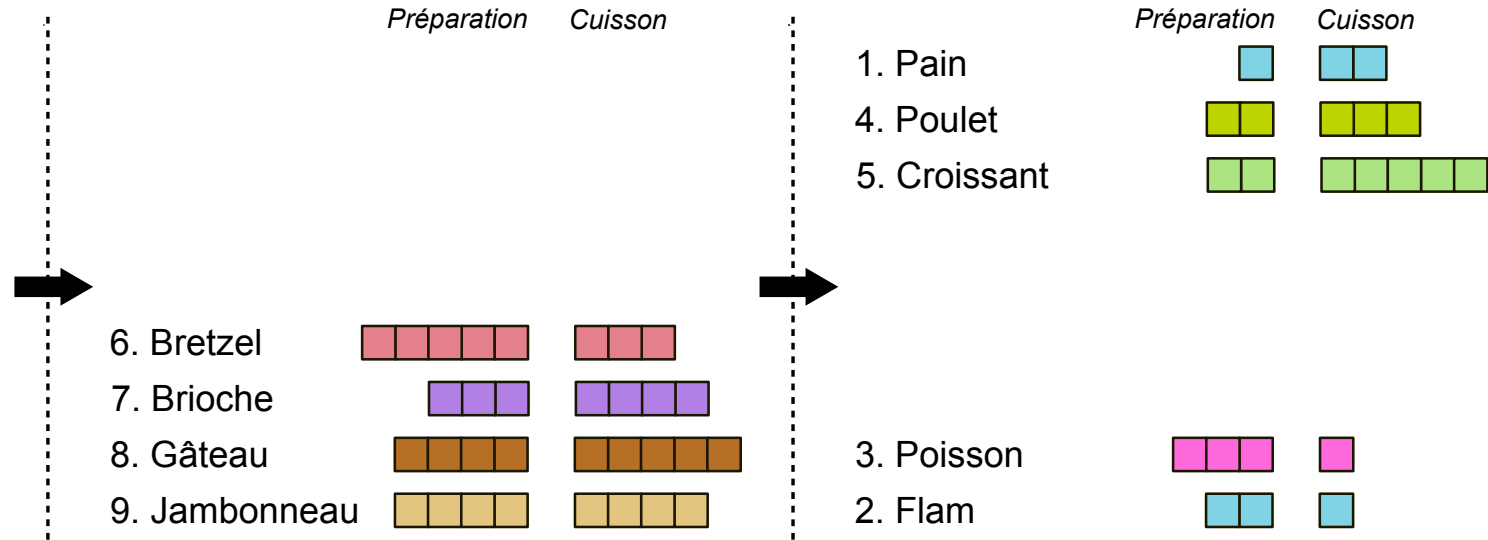
Préparation Cuisson

Tâches triées

Préparation Cuisson

Ordre de Johnson

Préparation Cuisson



Ordonnancement de Johnson

Le **croissant** a un temps de **Préparation** < temps de **Cuisson** => **au début**.



Déroulé de l'algorithme A de Johnson



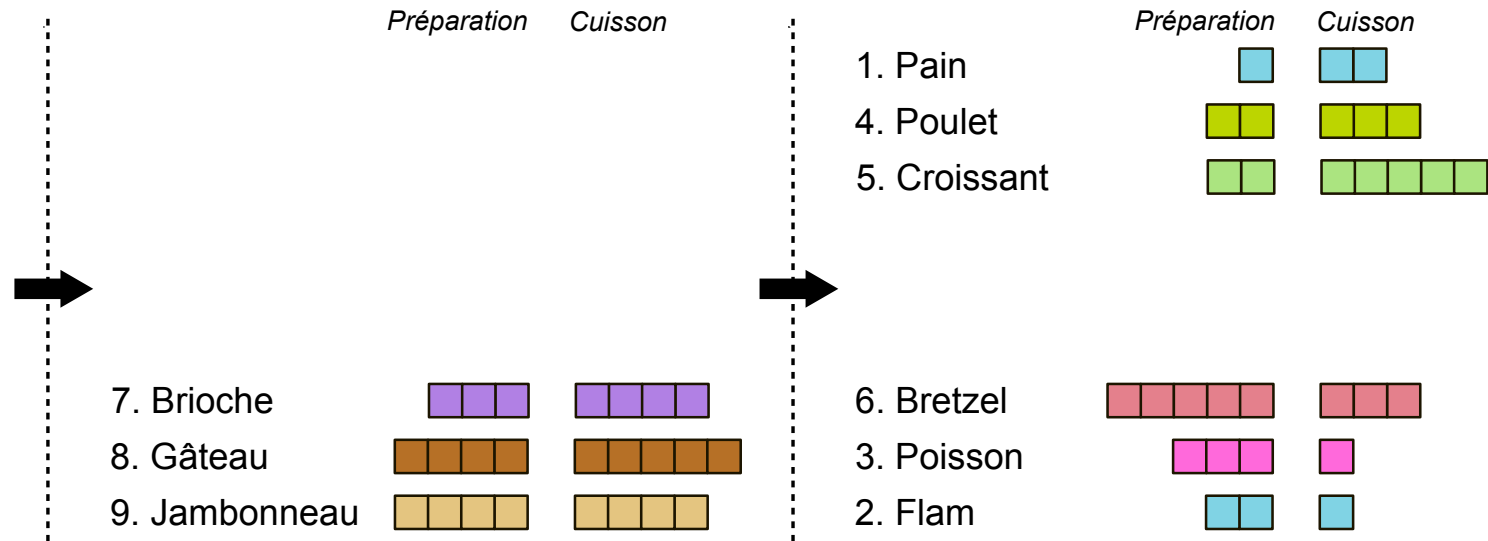
Préparation Cuisson

Tâches triées

Préparation Cuisson

Ordre de Johnson

Préparation Cuisson



Ordonnancement de Johnson

La **bretzel** a un temps de **Cuisson** < temps de **Préparation** => à la fin.



Déroulé de l'algorithme A de Johnson

Tâches à trier

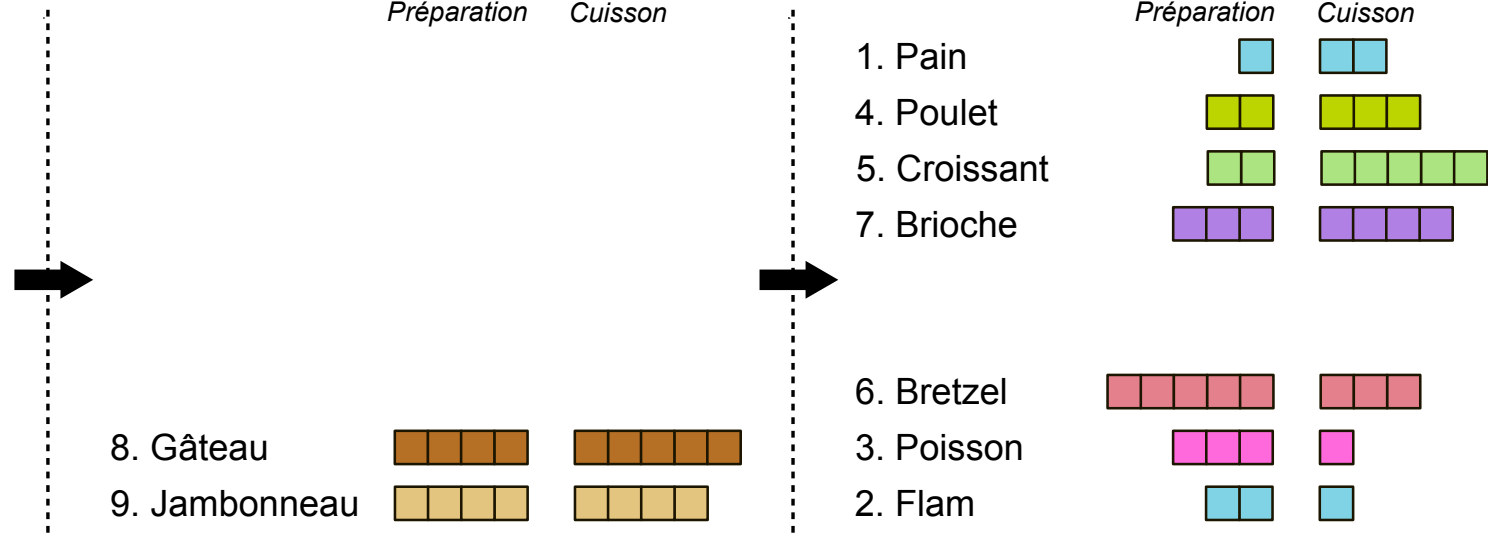
Préparation Cuisson

Tâches triées

Préparation Cuisson

Ordre de Johnson

Préparation Cuisson



Ordonnancement de Johnson

La **brioche** a un temps de **Préparation** < temps de **Cuisson** => **au début**.



Déroulé de l'algorithme A de Johnson

Tâches à trier

Préparation Cuisson



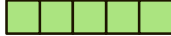
Tâches triées

Préparation Cuisson

9. Jambonneau  

Ordre de Johnson

Préparation Cuisson

1. Pain  
4. Poulet  
5. Croissant  
7. Brioche  
8. Gâteau  

6. Bretzel  
3. Poisson  
2. Flam  

Ordonnancement de Johnson

Le **gâteau** a un temps de **Préparation** < temps de **Cuisson** => **au début**.



Déroulé de l'algorithme A de Johnson

Tâches à trier

Préparation Cuisson



Tâches triées

Préparation Cuisson



Ordre de Johnson

	Préparation	Cuisson
1. Pain		
4. Poulet		
5. Croissant		
7. Brioche		
8. Gâteau		
9. Jambonneau		
6. Bretzel		
3. Poisson		
2. Flam		

Ordonnancement de Johnson
Le **jambonneau** se retrouve au “**centre**”.



Déroulé de l'algorithme A de Johnson

Tâches à trier

Préparation Cuisson



Tâches triées

Préparation Cuisson



Ordre de Johnson

	Préparation	Cuisson
1. Pain		
4. Poulet		
5. Croissant		
7. Brioche		
8. Gâteau		
9. Jambonneau		
6. Bretzel		
3. Poisson		
2. Flam		



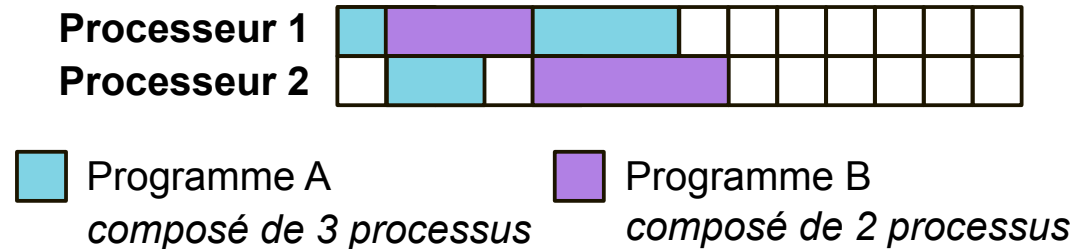
Solution de Johnson.

290 min.



Analyse

Lien avec l'informatique



Ordonnancement de tâches

- ▶ Un **processeur** peut exécuter autant de tâches en parallèle qu'il a de cœurs.
- ▶ Dans l'exemple du cuisinier :
 - Cuisine = processeur
 - Cuisinier et four = **2 cœurs** du processeur
 - Plats = **programmes** découpés en plusieurs tâches
- ▶ Ordre des tâches : exécution séquentielle d'un programme

Dans un ordinateur, c'est le **système d'exploitation** qui choisit l'ordonnancement.



Questions

- ▶ Peut-on être sûr que l'algorithme de Johnson donne toujours la solution optimale ?
- ▶ L'algorithme fonctionne-t-il encore si nous avons plus de deux tâches ?
- ▶ Le temps optimal était donné, comment faire si on ne le connaît pas ?
- ▶ Cet algorithme est-il utilisé dans les ordinateurs ?



Complexité





- ▶ Si on ajoute une **troisième tâche**, par exemple « laver les ustensiles », alors l'algorithme de Johnson ne fonctionne plus et **il n'existe pas d'algorithme général efficace pour résoudre le problème**.
- ▶ En pratique, les problèmes d'ordonnancement sont difficiles. Par exemple, le défi du cuisinier pressé avec **15 plats** nécessitant chacun **15 tâches différentes** est un problème qui n'a **pas de solution optimale connue** (réf. Julien N. ?).
- ▶ Nous pouvons démontrer que, si nous savions résoudre ce problème de manière efficace, alors nous saurions résoudre toute une collection d'autres problèmes. Tous les problèmes de cette collection sont dit **"NP-difficiles"**.



En pratique

- ▶ Dans l'activité, **les temps optimaux sont connus**. Dans un ordinateur, ce n'est pas le cas : l'OS ne sait pas d'avance à quel moment l'utilisateur va exécuter les programmes.
- ▶ Nous avons un algorithme pour des cas **statiques** et **déterministes**, alors que les systèmes d'exploitation gèrent des cas **dynamiques** et **non déterministes**.
- ▶ Les systèmes d'exploitation mettent en place des **stratégies** pour optimiser le temps d'utilisation des processeurs. Exemples de stratégie :
 - ✓ Premier arrivé, premier servi
 - ✓ Plus court en premier



Il n'y a pas de preuve d'une stratégie meilleure que les autres.



Ressources

- ▶ Activité publiée dans le numéro 67/68 de **Tangente éducation** :
 - <https://www.librairie-infinimath.com/collections/new-tangente-education-1/products/numero-67-68-tangente-education-linformatique-de-branche-tome-2>
- ▶ Matériel à imprimer sur le dépôt **Github** “Médiation Info Strasbourg”
 - <https://github.com/jnarboux/MediationInfoStrasbourg/wiki/Le-cuisinier-press%C3%A9>
- ▶ **Article** de l’algorithme de Selmer Martin Johnson

S. M. Johnson, “Optimal two- and three-stage production schedules with setup times included”, Naval Research Logistics Quarterly, 1(1):61–68, 1954. <https://doi.org/10.1002/nav.3800010110>

