Fournées

Année 2014 – 2015

Abigail DAH, Tiffany SERIN, Manon ESPALIVET, Emma ANGOT, élèves de 1ère S
Aurore TEULIER, Céline et Sandrine PENEAUD, Anaïs JEKER, Vincent GIACOMELLO, élèves de 3ème
Encadrés par Mme VERNHET, Mme BOUSQUET, M. PASQUET, M.BEDUER

Établissements : Lycée Raymond Savignac, Collège de Cajarc

Chercheur : Julien MONCEL, Laboratoire d'analyse et d'architecture des systèmes (LAAS-CNRS), Université de Toulouse 1

Introduction. « Les fournées » est un sujet de recherche mathématique qui nous a été proposé à la rentrée 2014 par Julien Moncel, chercheur et professeur à l'IUT de Rodez. Tout au long de nos recherches, nous avons travaillé conjointement entre élèves du lycée Raymond Savignac et élèves du collège de Cajarc. Nous avons également collaboré avec des étudiants de l'IUT de Rodez, qui ont mis au point un logiciel modélisant notre sujet. Nous avons organisé plusieurs séminaires entre collégiens, lycéens, et étudiants au cours de l'année, dans le but de mettre en commun nos recherches et nos avancées. Nous sommes finalement allés présenter notre travail lors du congrès Maths-en-Jeans à Angers les 27 et 28 mars 2015.

Présentation du sujet

Dans le sujet « Fournées », nous nous intéressons à la cuisson d'un certain nombre d'objets, regroupés par fournées.

Chaque objet présente plusieurs caractéristiques :

- Il possède une taille qui lui est propre, notée t_j.
- Il est caractérisé par une durée de cuisson minimale, notée D_{min}, et une durée de cuisson maximale, notée D_{max}. Son temps de cuisson devra donc être compris dans l'intervalle $[D_{min} ; D_{max}]$.

Pour planifier une fournée, nous devons donc tenir compte de deux facteurs.

- Il faut tout d'abord veiller à ce que la somme des tailles de tous les objets de la fournée soit inférieure ou égale à la capacité du four que nous noterons k par la suite.
- Chaque fournée doit ensuite être cuite selon un temps précis de cuisson. Celui-ci doit respecter les durées minimales et maximales de cuisson de chaque objet. En d'autres termes, le temps de la cuisson de la fournée doit être compris entre la durée minimale et maximale de cuisson de chaque objet.

L'objectif est de cuire tous les objets en un temps d'utilisation du four minimal.

Voici donc un exemple :

Numéro Objet	Durée de cuisson minimale : D_{min}	Durée de cuisson maximale : D_{max}	Taille : t_j
1	10	15	5
2	6	10	5
3	12	20	4
4	3	6	2
5	5	7	2
6	5	15	1

Dans cet exemple, nous avons six objets à faire cuire. On suppose que le four a une capacité k=10.

Nous pouvons par exemple faire une fournée avec les objets n°1 et n°2, puisque la somme de leur taille ne dépasse pas la capacité du four (5+5=10=k), et si la durée de la fournée est 10min, alors les durées minimales et maximales de cuisson de chaque objet sont respectées.

– Cas particuliers

On a d'abord travaillé sur deux objets seulement en considérant que les deux objets peuvent être cuits en même temps.

On note :

D_{min1} : durée minimale de l'objet 1.

D_{max1} : durée maximale de l'objet 1.

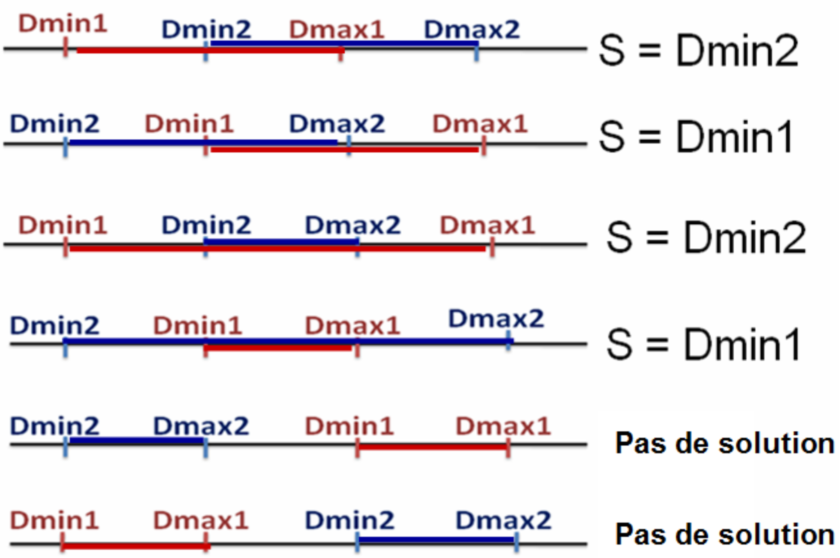
D_{min2} : durée minimale de l'objet 2.

D_{max2} : durée maximale de l'objet 2.

En comparant ces nombres, nous en avons déduit six cas possibles.

Nous avons étudié chacun d'eux pour déterminer s'il existe une intersection entre les intervalles de cuisson $[D_{min1} ; D_{max1}]$ et $[D_{min2} ; D_{max2}]$.

On note S le temps commun minimal pour faire cuire les deux objets ensemble : S correspond donc à la borne inférieure de l'intersection des deux intervalles



(1)

Une fois les six cas possibles définis, nous avons créé un algorithme nous permettant de résoudre notre problème lorsque nous choisissions deux objets (c'est à dire trouver leur temps minimal de cuisson) après avoir vérifié qu'ils peuvent être cuits ensemble dans le four.

Pour cela, il suffit de rentrer les tailles et les durées de cuisson minimales et maximales des deux objets ainsi que la capacité du four.

Dans un premier temps, l'algorithme va additionner les tailles des deux objets pour comparer le résultat avec la taille du four. Si la somme obtenue est supérieure à la taille du four alors ces deux objets seront cuits dans deux fournées différentes, sinon l'algorithme poursuit sa démarche de résolution.

Ensuite, Il va comparer les intervalles de cuisson des deux objets. Deux cas vont alors s'offrir à lui :

- soit les deux intervalles de cuisson ne présentent aucune intersection, les deux objets seront donc cuits dans deux fournées différentes ;
- soit l'algorithme trouve une intersection et par conséquent choisit la borne inférieure de celle-ci, c'est à dire le temps minimal commun aux deux objets.
-

- Cas général

Cette démarche consiste à trouver toutes les combinaisons possibles d'objets que l'on peut faire cuire ensemble selon leur temps de cuisson et leur taille, afin de choisir la plus adaptée. Cette méthode, nommée « intersection » doit nous permettre de résoudre n'importe quel problème, quel que soit le nombre d'objets, leurs durées de cuissons minimales et maximales, leurs tailles, et la capacité du four.

Tout au long de l'explication de cette méthode, nous nous appuierons sur l'exemple du I).

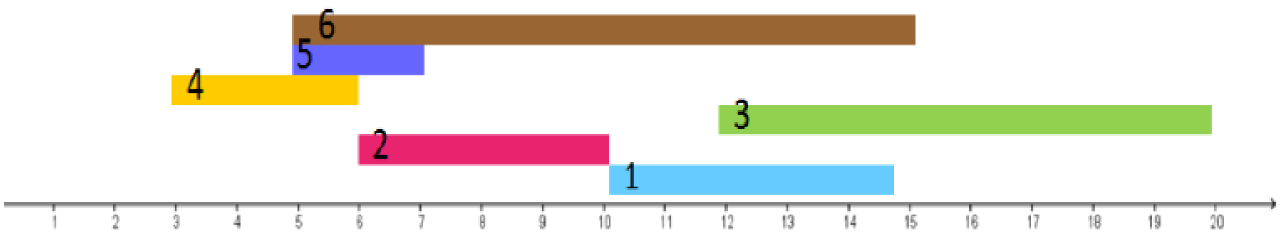
A – Combinaisons par temps de cuisson

Tout d'abord, il faut répertorier tous les objets compatibles par temps de cuisson, c'est-à-dire quels objets peuvent cuire ensemble. Pour cela, nous avons défini un intervalle de cuisson pour chaque objet, dont D_{min} représente la borne inférieure, et D_{max} représente la borne supérieure.

Objet	Dmin	Dmax	Intervalle de cuisson
1	10	15	[10 ;15]
2	6	10	[6 ;10]
3	12	20	[12 ;20]
4	3	6	[3 ;6]
5	5	7	[5 ;7]
6	5	15	[5 ;15]

Tableau associant à chaque objet son intervalle de cuisson

On représente chaque intervalle sur un graphique. Pour savoir si deux objets peuvent être cuits ensemble, il faut regarder si leurs intervalles de cuisson ont une intersection.



Graphique des intervalles de cuisson de chaque objet

On relève ensuite toutes les combinaisons possibles entre objets. Dans ce cas, les combinaisons obtenues sont :

Pour deux objets :	Pour trois objets :	Pour quatre objets :

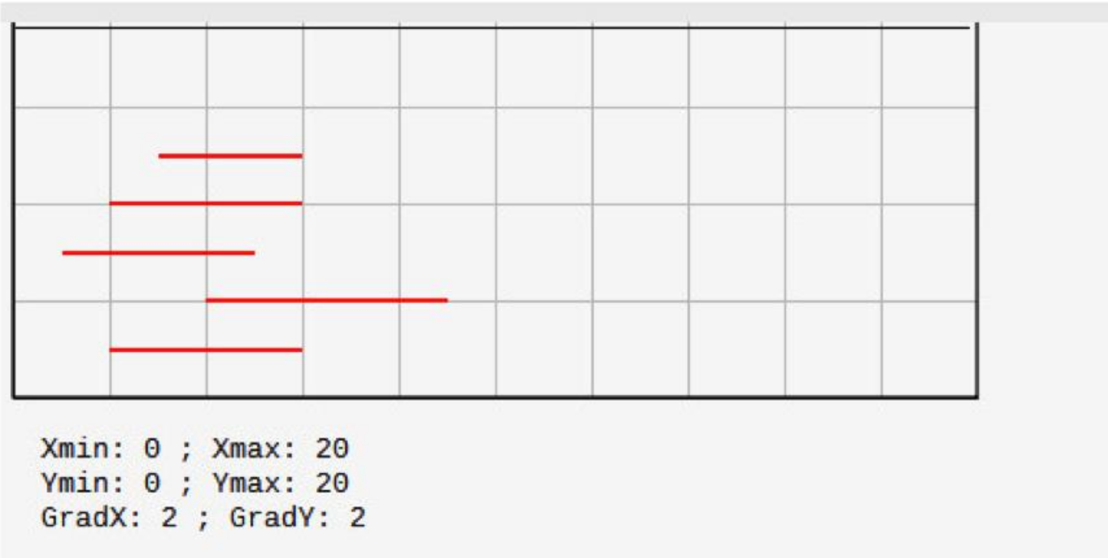
4-6	4-5-6	2-4-5-6
4-5	5-2-6	
6-5	3-1-6	
6-2	4-5-2	
6-1	1-2-6	
6-3		
5-2		
2-4		
1-3		
2-1		

Dans la continuité de notre démarche pour les cas généraux, nous avons voulu trouver un moyen de rendre plus rapide la réalisation de ce graphique. Pour cela, nous avons créé un algorithme capable de tracer un graphique identique à celui que nous tracions manuellement.

Pour cela, il suffit de rentrer au préalable le nombre d'objet ainsi que les durées minimales et maximales de cuisson de chaque objet.

```
VARIABLES
  n EST_DU_TYPE NOMBRE
  i EST_DU_TYPE NOMBRE
  Dmin EST_DU_TYPE LISTE
  Dmax EST_DU_TYPE LISTE
DEBUT_ALGORITHME
  i PREND_LA_VALEUR 1
  LIRE n
  POUR i ALLANT_DE 1 A n
    DEBUT_POUR
    LIRE Dmin[i]
    LIRE Dmax[i]
    TRACER_SEGMENT (Dmin[i],i)->(Dmax[i],i)
    FIN_POUR
  i PREND_LA_VALEUR i+1
FIN_ALGORITHME
```

L'algorithme trace ensuite un graphique composé de segments. Chaque objet est ainsi représenté par un segment et placé dans le graphique tel que les deux extrémités du segment représentent chacune la durée minimale et maximale de l'objet, et tel que l'ordonnée du segment corresponde au numéro de l'objet.



Cet algorithme nous a permis d'accélérer notre démarche en supprimant le traçage manuel de ce graphique, cependant on doit répertorier les combinaisons d'objets par temps de cuisson toujours de manière manuelle.

B- Combinaisons par tailles (2)

Deux objets sont compatibles par taille si et seulement si la somme de la taille de chaque objet n'excède pas la capacité du four.

Pour obtenir toutes les combinaisons possibles, on prend chaque objet et on en ajoute un deuxième, et on regarde la somme de leurs tailles respectives.

On distingue alors trois cas de figures :

Cas 1 :

Si cette somme est supérieure à la capacité du four, alors les deux objets ne sont pas compatibles.

Cas 2 :

Si cette somme est égale à la capacité du four, alors les deux objets sont compatibles et le four est complet, on ne pourra pas ajouter un troisième objet.

Cas 3 :

Si cette somme est inférieure à la capacité du four, alors les deux objets sont compatibles et il reste de la place dans le four égale à la différence entre la capacité du four et la somme de la taille des deux objets. Dans ce dernier cas, on peut ajouter un troisième objet. Pour savoir si ce troisième objet est compatible, on réitère la même méthode. On continue ainsi de suite avec toutes les objets pour connaître toutes les combinaisons possibles.

On note les résultats obtenus dans des tableaux à double entrée.

Dans les tableaux suivants :

- La colonne de gauche représente les objets que l'on considère déjà présents dans le four
- La ligne du dessus représente les objets que l'on s'apprête à ajouter dans le four
- Les cases grisées empêchent une combinaison d'être prise en compte plusieurs fois.
- On reporte dans chaque case la somme de la taille des objets de la ligne et de la colonne correspondante.
- Lorsque le nombre reporté est égal à la capacité du four, la capacité maximale du four est atteinte et l'on ne pourra pas ajouter un objet, cela correspond aux cases orange dans le tableau suivant.

	Objet 1	Objet 2	Objet 3	Objet 4	Objet 5	Objet 6
Objet 1						
Objet 2	10					
Objet 3	9	9				
Objet 4	7	7	6			
Objet 5	7	7	6	4		
Objet 6	6	6	5	3	3	

Tableau des compatibilités entre deux objets

	Objet 1	Objet 2	Objet 3	Objet 4	Objet 5	Objet 6
Objets 1+3						
Objets 1+4						
Objets 1+5						
Objets 1+6						
Objets 2+3						
Objets 2+4						
Objets 2+5						
Objets 2+6						
Objets 3+4	11	11				
Objets 3+5	11	11				
Objets 3+6	10	10				
Objets 4+5	9	9	8			
Objets 4+6	8	8	7			
Objets 5+6	8	8	7	5		

Tableau des compatibilités entre trois objets

	Objet 1	Objet 2	Objet 3	Objet 4	Objet 5	Objet 6
Objets 1+4+5						
Objets 1+4+6						
Objets 1+5+6						
Objets 2+4+5						
Objets 2+4+6						
Objets 2+5+6						
Objets 3+4+5						
Objets 3+4+6						
Objets 3+5+6						
Objets 4+5+6	10	10	9			

Tableau des compatibilités entre quatre objets.

D'après ces tableaux, on reporte ensuite toutes les combinaisons trouvées :

Pour deux objets :	Pour trois objets :	Pour quatre objets :
1-2	1-3-6	1-4-5-6
1-3	1-4-5	2-4-5-6
1-4	1-4-6	3-4-5-6
1-5	1-5-6	
1-6	2-3-6	
2-3	2-4-5	
2-4	2-4-6	
2-5	2-5-6	
2-6	3-4-5	
3-4	3-4-6	
3-5	3-5-6	
3-6	4-5-6	
4-5		
4-6		
5-6		

Dans le but de faciliter notre démarche et de même que pour les combinaisons par temps de cuisson, nous nous sommes interrogés sur la possibilité de créer un algorithme nous permettant de classer les objets par taille.

Cependant à cause de la difficulté de la tache et d'un manque de compétences en matière d'algorithme notre tentative s'est soldée par un échec.

C- Intersection

Après avoir répertorié toutes les combinaisons possibles par taille et temps de cuisson, il faut comparer les deux et ne retenir que les combinaisons qui sont communes aux deux critères. Les combinaisons retenues représentent donc les objets pouvant être cuits ensemble, car ils sont compatibles à la fois par temps de cuisson et par taille.

Voici donc les combinaisons obtenues :

Pour deux objets :	Pour trois objets :	Pour quatre objets :
1-2	1-3-6	2-4-5-6
1-3	2-4-5	
1-6	2-4-6	
2-4	2-5-6	

2-5		
2-6		
3-6		
4-5		
4-6		
5-6		

Chaque combinaison des tableaux ci-dessus représente une fournée. Nous devons donc associer à chaque fournée son temps de cuisson. Ce temps est compris entre la durée minimale et maximale de cuisson de chaque objet, et il correspond au plus petit temps commun à tous les objets.

Les tableaux ci-dessous associent à chaque fournée leur temps de cuisson.

Fournées avec 1 objet	1	2	3	4	5	6
Temps de cuisson de la fournée	10	6	12	3	5	5

Fournées avec 2 objets	1-2	1-3	1-6	2-4	2-5	2-6	3-6	4-5	4-6	5-6
Temps de cuisson de la fournée	10	12	10	6	6	6	12	5	5	5

Fournées avec 3 objets	1-3-6	2-4-5	2-4-6	2-5-6	4-5-6
Temps de cuisson de la fournée	12	6	6	6	5

Fournée avec 4 objets	2-4-5-6
Temps de cuisson de la fournée	6

Le but est ensuite de minimiser les fournées, c'est-à-dire d'en faire le moins possible. Ici, nous pouvons faire deux fournées au minimum. Il faut ensuite choisir la combinaison de fournées dont la somme des temps de cuisson sera la plus petite possible.

Deux solutions s'offrent alors à nous dans cet exemple :

On peut choisir de faire une fournée avec les objets 2-4-5-6 puis une deuxième fournée avec les objets 1-3, la somme des temps de cuisson de ces deux fournées est alors 18min.

On peut également choisir de faire une fournée avec les objets 1-3-6 puis une deuxième fournée avec les objets 2-4-5, la somme des temps de cuisson de cette fournée est alors 18min.

Dans cet exemple, les deux alternatives proposées mettront l'une et l'autre 18min, on a donc deux solutions. Si les temps avaient été inégaux, il aurait fallu choisir le plus petit.

Conclusion

Après une année de recherche, nous avons réussi à répondre pleinement au problème lorsqu'il n'y a que deux objets, grâce à l'étude de toutes les situations possibles, ce qui nous a permis de créer un algorithme efficace dans cette situation. En ce qui concerne les cas avec plus d'objets, nous avons bien trouvé une méthode nous permettant de faire cuire tous les objets dans un temps minimal d'utilisation du four, mais cette méthode reste lente et fastidieuse, les erreurs de calculs peuvent alors être nombreuses. C'est pourquoi nous avons essayé de la mettre sous forme d'algorithme, mais nous n'avons réussi à créer un algorithme que pour la première étape, c'est-à-dire répertorier les combinaisons d'objets selon leurs temps de cuisson. Il serait alors intéressant de chercher à simplifier cette méthode, même manuellement, pour pouvoir ensuite la mettre sous forme d'algorithme plus facilement. Il serait alors aussi intéressa nt d'étudier le sujet en le simplifiant comme par exemple en choisissant des objets de la même taille ou avec des durées de cuisson identiques.

Annexe 1 : Algorithme pour deux objets

```
VARIABLES
  taille_de_lobjet_1 EST_DU_TYPE NOMBRE
  durée_minimale_de_lobjet_1 EST_DU_TYPE NOMBRE
  durée_maximale_de_lobjet_1 EST_DU_TYPE NOMBRE
  taille_de_lobjet_2 EST_DU_TYPE NOMBRE
  durée_minimale_de_lobjet_2 EST_DU_TYPE NOMBRE
  durée_maximale_de_lobjet_2 EST_DU_TYPE NOMBRE
  capacité_du_four EST_DU_TYPE NOMBRE
  Temps_minimal_commun_aux_deux_objets EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
  LIRE taille_de_lobjet_1
  LIRE durée_minimale_de_lobjet_1
  LIRE durée_maximale_de_lobjet_1
  LIRE taille_de_lobjet_2
  LIRE durée_minimale_de_lobjet_2
  LIRE durée_maximale_de_lobjet_2
  LIRE capacité_du_four
  SI (taille_de_lobjet_1+taille_de_lobjet_2<=capacité_du_four) ALORS
    DEBUT_SI
    SI (durée_minimale_de_lobjet_1<=durée_minimale_de_lobjet_2 ET durée_minimale_de_lobjet_2<=durée_maximale_de_lobjet_1) ALORS
      DEBUT_SI
      Temps_minimal_commun_aux_deux_objets PREND_LA_VALEUR durée_minimale_de_lobjet_2
      AFFICHER Temps_minimal_commun_aux_deux_objets
      FIN_SI
    SI (durée_minimale_de_lobjet_2<=durée_minimale_de_lobjet_1 ET durée_minimale_de_lobjet_1<=durée_maximale_de_lobjet_2) ALORS
      DEBUT_SI
      Temps_minimal_commun_aux_deux_objets PREND_LA_VALEUR durée_minimale_de_lobjet_1
      AFFICHER Temps_minimal_commun_aux_deux_objets
      FIN_SI
    SI (durée_maximale_de_lobjet_1<durée_minimale_de_lobjet_2 OU durée_maximale_de_lobjet_2<durée_minimale_de_lobjet_1) ALORS
      DEBUT_SI
      AFFICHER "Il faut faire deux fournées"
      FIN_SI
    FIN_SI
    SINON
      DEBUT_SINON
      AFFICHER "Il faut faire deux fournées"
      FIN_SINON
FIN_ALGORITHME
```

(3)

Annexe 2 : Logiciel des étudiants

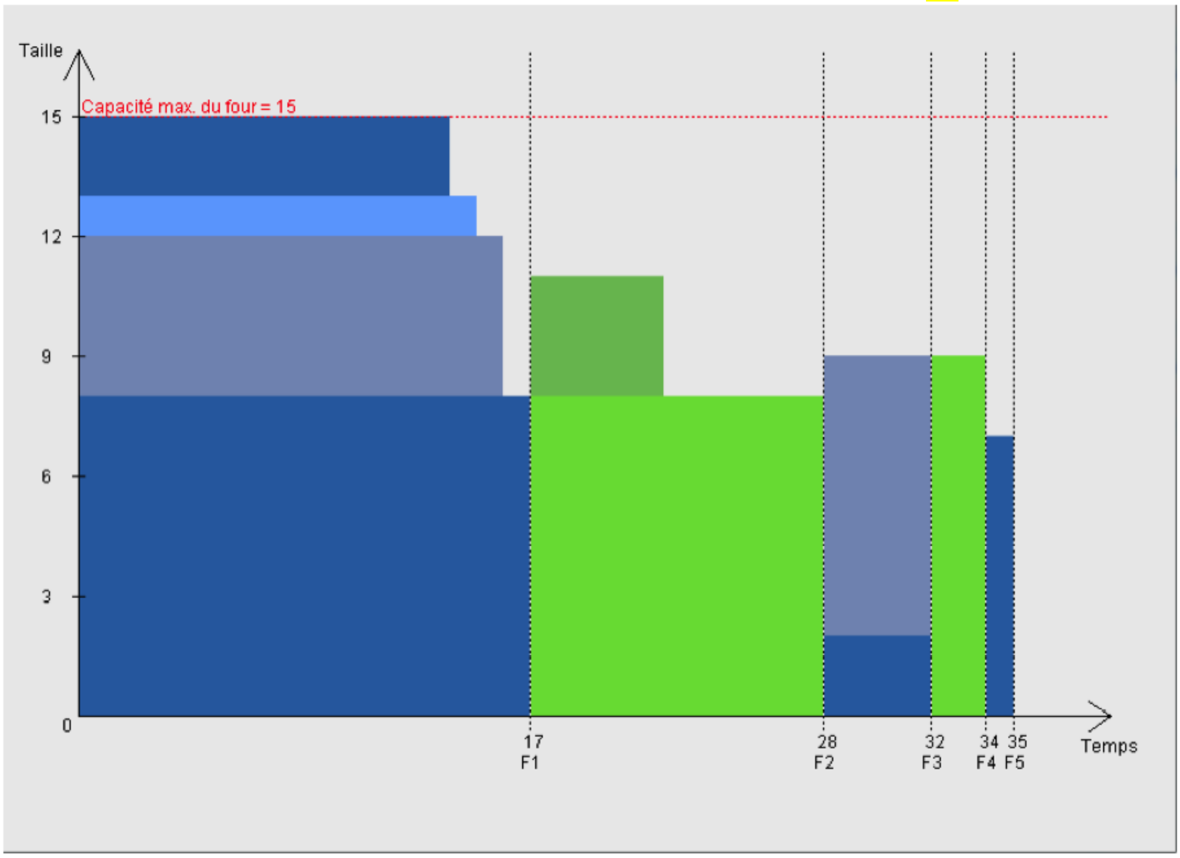
Des étudiants de l'IUT de Rodez nous ont créé un logiciel qui nous a permis de schématiser notre problème. Dans ce logiciel, nous pouvions choisir entre trois modes :

-le mode Manuel : Nous rentrons manuellement nos objets dans chaque fournée et le logiciel se bloque automatiquement lorsque nous rentrons deux objets non compatibles au niveau de la taille et la durée de cuisson.

-le mode Algorithme : nous pouvions choisir entre trois modes d'algorithme (par taille, par durée minimale de cuisson ou par durée maximale de cuisson).

-le mode Editeur de Fichier : qui nous permet de créer notre propre problème avec un four et des objets de notre choix.

Voici un exemple de représentation graphique après traitement par le logiciel (4)



Chaque rectangle représente un objet. En abscisse, sont représentées les durées minimales de chaque objet. En ordonnée, sont représentées les tailles de chaque objet. La ligne rouge représente la capacité du four, à ne pas dépasser. Chaque colonne, séparées entre elles par une ligne en pointillés représente une fournée.

Notes d'édition

(1) Dans les deux derniers cas, il faut deux cuissons et le temps de cuisson total est Dmin1 + Dmin2.

(2) Dans cet exemple les objets sont ordonnés par taille. Il conviendra dans le cas général de toujours ordonner les objets par taille pour trouver plus facilement toutes ces combinaisons.

(3) On peut remplacer le dernier « SI » (10 lignes avant la fin de l'algorithme) par un simple « SINON ».

(4) Il manque dans cette représentation la durée maximale de la cuisson de chaque objet. Pour complètement schématiser le problème, il faut ajouter cette information au graphique.