Projet Informatique Fondamentale

Année 2025-2026

1 Problème des poules

On considère N poules, poursuivies par un renard, qui veulent traverser une rivière le plus vite possible à l'aide d'une barque. Elles ont à leur disposition une barque qui ne peut contenir qu'au plus C poules à la fois. Seulement, certaines poules (mouillées) ont peur du courant présent dans la rivière et réduisent en conséquence leur vitesse durant la traversée. La durée prise par C' poules (avec $C' \leq C$) pour traverser la rivière en une fois est celle de la plus peureuse (la durée la plus longue).

Par exemple, la poule Bravette n'est pas très peureuse et pourrait traverser seule en 4 minutes. La poule Tremblotte est très peureuse et si elle traversait seule, elle mettrait 20 minutes. Si Bravette et Tremblotte traversent ensemble, elles mettront 20 minutes car Bravette s'adapte à Tremblotte. Par contre, si Bravette revient seule sur la berge de départ, elle ne mettra que 4 minutes. La difficulté du problème est qu'il n'y a qu'une seule barque, il faut donc potentiellement faire plusieurs allers et retours pour que toutes les poules puissent traverser et, bien sûr, il faut au moins une poule, par aller et par retour, pour ramer.

On note $T_i \in \mathbb{N}$ le temps de traversée de la poule $i \in \{1, \ldots, N\}$. La question est de savoir quel est le temps minimum pour que toutes les poules traversent.

2 Exemple

Prenons l'instance $N = 4, T_1 = 1, T_2 = 3, T_3 = 6, T_4 = 8$ et $C = 2$. Toutes les poules peuvent traverser en 18 minutes, mais comment ? Avant de regarder la solution, essayez de la trouver par vous-même.

Type de voyage	Poules dans barque	Temps	Poules berge A	Poules berge B
Aller	$\{1, 2\}$	3	$\{3, 4\}$	$\{1, 2\}$
Retour	$\{1\}$	1	$\{1, 3, 4\}$	$\{2\}$
Aller	$\{3, 4\}$	8	$\{1\}$	$\{2, 3, 4\}$
Retour	$\{2\}$	3	$\{1, 2\}$	$\{3, 4\}$
Aller	$\{1, 2\}$	3	$\varnothing$	$\{1, 2, 3, 4\}$

3 Questions

Le but du projet est de résoudre, en fonction de N, C et les temps T_i, le problème énoncé avant, à l'aide d'un solveur SAT. Puisqu'il s'agit d'un problème de minimisation, les questions qu'on posera au solveur SAT seront du type "est-il possible de faire traverser toutes les poules en une durée au plus T ?", où T sera donné en entrée. On appellera ce dernier problème le "problème des poules *contraint*", qui est un problème de décision. Il suffira ensuite de faire une recherche du T minimal pour résoudre le problème des poules. Voici les questions :

1. Donner une réduction du problème des poules contraint vers le problème SAT, qui peut être calculée par un algorithme de complexité polynomiale en temps, en fonction de la taille des entrées (on supposera que T est donné en base unaire). Expliquer votre réduction.
2. Implémenter cette réduction en Python avec la librairie PySAT. Plus précisément, remplissez le code de la fonction `gen_solution` dans le fichier `project.py`. Vous pouvez utiliser la méthode `CardEnc.atmost` qui encode directement la contrainte "au plus k littéraux sont vraies, étant donné un ensemble de littéraux". Vous pouvez également utiliser le solveur `Minicard` qui gère les contraintes de cardinalité de manière native (voir la documentation de la classe `CNFPlus` de Pysat).
3. En vous basant sur l'implémentation précédente, implémenter une procédure pour résoudre le problème des poules (trouver le T minimal). Plus précisément, remplissez le code de la fonction `find_duration` dans le fichier `project.py`.

La fonction `gen_solution` doit renvoyer `None` s'il n'existe pas de solution et une liste de paires `(t, chicken)` représentant le fait qu'à l'instant `t`, toutes les poules de la liste `chicken` partent sur la barque. Conformément à l'exemple donné dans la section 2, l'appel `gen_solution([1, 3, 6, 8], 2, 18)` doit renvoyer :

```
[(0, [1, 2]), (3, [1]), (4, [3, 4]), (12, [2]), (15, [1, 2])]
```

La fonction `find_duration` doit quand à elle simplement renvoyer la valeur de T (un entier).

4 Consignes de remise

Le projet est à réaliser **obligatoirement** en trinômes. Les groupes doivent être formés sur l'UV pour le vendredi 21 novembre à 16h. Après cela, les groupes incomplets seront fusionnés de manière automatique par nos soins. Il ne sera pas possible de s'inscrire après cette date !

Dans le devoir associé sur l'UV, vous devez rende un unique fichier ZIP appelé `<M1>_<M2>_<M3>.zip` où les `Mi` sont les matricules de membres du groupe (donc par exemple `366726_408282_425410.zip`) contenant :

— le rapport pour la question 1 au format PDF ;

— le(s) fichier(s) source du rapport (LaTeXou Typst) ;
— le fichier `project.py` complété pour les questions 2 et 3.

Attention : le projet ne doit être remis qu'une seule fois par trinôme. De plus, il ne vous est pas demandé de rendre les fichier `test.py` et `utils.py` car nous remettrons ceux fournis sur l'UV dans *chacun* des projets remis. De plus, nous vous demandons de mettre tout votre code dans le fichier `project.py` et de ne pas en remettre plusieurs.

La date de remise est fixée au dimanche 21 décembre à 23h59.

5 Suggestions

Cette section contient des suggestions pour la résolution du projet. Nous vous invitons à y réfléchir une première fois entre vous avant de regarder la suite. Notez que la modélisation proposée n'est en rien obligatoire et n'est qu'une suggestion, comme son nom l'indique. Vous pouvez tout à fait avoir le maximum des points avec une modélisation radicalement différente.

5.1 Suggestion de modélisation

Nous vous donnons ici des suggestions de variables à utiliser et leur sémantique, pour résoudre le projet. **Ce ne sont que des suggestions, que vous pouvez très bien décider de pas suivre si vous êtes plus à l'aise avec une autre approche du problème !**

Initialement, toutes les poules sont sur la berge A et elles veulent atteindre la berge B. On nommera *aller* tout mouvement de A vers B, et *retour*, tout mouvement de B vers A. Vous pourriez utiliser les variables suivantes :

— $dep_{t,p,s}$ vraie si il y a un départ de la barque à l'instant $t \in \{0, \dots, T\}$, qui contient la poule $p \in \{1, \dots, N\}$ et de type $s \in \{Aller, Ret\}$;
— $A_{p,t}$ (resp. $B_{p,t}$) vraie si la poule p est sur la berge A (resp. B) à l'instant t. Pour simplifier, on supposera qu'une poule qui voyage reste sur sa berge de départ jusqu'au moment où la barque arrive de l'autre côté, et seulement à ce moment-là la poule change de berge. Une autre solution consisterait à avoir des variables qui représentent la présence de chaque poule dans la barque à chaque instant ;
— $dur_{t,d}$ vraie s'il y a un départ à l'instant t et que la traversée a une durée $d \in \{0, \dots, \max_i\{T_i\}\}$ (d doit donc être la durée maximale de toutes les poules qui voyagent à l'instant t)
— $side_t$ vraie si la barque est du côté A, fausse si elle est du côté B (comme précédemment, on suppose que le changement de côté ne s'effectue qu'aux arrivées) ;
— il sera également pratique d'utiliser les variables DEP_t, vraie s'il existe un départ à l'instant t, et ARR_t vraie s'il existe une arrivée à l'instant t ;
— enfin, on utilisera les variables ALL_t, vraie si toutes les poules sont sur la berge B à l'instant t.

Notez que les variables essentielles sont les variables $dep_{t,p,s}$ qui indiquent quelles sont les poules qui partent à quel instant et vers quel berge : avoir la valeur de vérité de ces variables permet de reconstruire une solution au problème, s'il y en a une. Les autres variables sont utiles pour exprimer les contraintes du problème. N'hésitez pas à en ajouter d'autres si cela est utile pour réduire le problème vers SAT.

5.2 Suggestions question 1

Si vous utilisez les variables précédentes, voici quelques suggestions pour la question 1 :

1. commencez par exprimer des contraintes qui définissent les variables DEP_t, ARR_t, ALL_t (en fonction des autres variables) ;
2. exprimez ensuite des contraintes qui vont assurer que si $dur_{t,d} = 1$, alors il existe un voyage qui commence au temps t, et d'une durée d. Pour rappel, d doit être le temps maximum des durées de toutes les poules qui partent à l'instant t. Il faudra donc exprimer (1) qu'il n'y a pas de poules qui part au temps t et dont la durée est strictement supérieure à d et, (2) il existe une poule qui part dont la durée de voyage est d exactement. Ceci peut s'exprimer en fonction des variables $dep_{t,p,s}$;
3. vous devez ensuite donner des contraintes sur l'évolution de la population des poules entre les berges : par exemple, si la barque arrive en B, alors la berge B contient maintenant les poules qui ont voyagé. Utilisez les variables $A_{p,t}$, $B_{p,t}$, $dep_{t,p,s}$ et $dur_{t,d}$ pour cela ;
4. il faudra également exprimer des contraintes qui définissent la dynamique des mouvements : une stricte alternance entre les allers et les retours, au plus C poules dans la barque, voyage d'un côté à l'autre que si la barque est présente, etc.

5.3 Suggestions question 2

Pour la question 2, vous pouvez mettre le nom des variables comme chaîne de caractère dans l'appel à `vpool`, cela rend le code plus lisible et permet de faire la différence entre variables qui ont les mêmes indices (par exemple, ARR_1 et DEP_1). Par exemple, un appel à `vpool.id(("dep", 0,1,"Aller"))` va créer une variable interne au solveur qui correspond à $DEP_{0,1,Aller}$.

Important : vous pouvez utiliser la méthode `CardEnc.atmost` de PySAT (voir la documentation) pour créer automatiquement des clauses pour la contrainte "au plus C poules sont sur la barque"). Comme dit précédemment, vous pouvez également utiliser le solveur `Minicard` qui gère les contraintes de cardinalité de manière native (voir la documentation de la classe `CNFPlus` de PySAT).