

Where Eagles Dare : Etude des nombres faibles de Schur  
Travail inter-atelier 2  
Mathématiques et informatique théoriques

Richard Abou Chaaya, Thomas Bianco, Benoît Ferté, Gabriel Merlin, Mario Michelessa

Encadrants : Joanna Tomasik, Arpad Rimmel

17 Avril 2019

# Contents

1	Contexte . . . . .	2
2	Description mathématique . . . . .	2
3	Pistes et ressources . . . . .	2
4	Planning . . . . .	3
5	Avancement . . . . .	3
	5.1 Articles lus . . . . .	3
	5.2 Codes réalisés . . . . .	4
6	Annexes . . . . .	4

# 1 Contexte

En 1912, Issai Schur publie un article portant sur la question du "dernier théorème de Fermat" dans sa version modulo  $n$  : la question est de savoir si

$$\forall k, \exists x, y, z \text{ tels que } x^k + y^k \equiv z^k \pmod n$$

Pour montrer ce résultat, Schur démontre un théorème qui portera son nom, et qui s'énonce ainsi : si l'on colore avec un nombre fini de couleurs les entiers de 1 à  $n$ , il existera  $x, y, z$  de la même couleur tels que  $x + y = z$ .

La question qui se pose alors est : pour un nombre fixé  $k$  de couleurs, combien d'entiers successifs peut-on colorer sans qu'il existe 3 entiers de même couleur satisfaisants cette propriété ? On dira alors de l'ensemble formé par les nombres d'une même couleur qu'il est sans somme.

Cette question s'inscrit dans le cadre plus large de la théorie de Ramsey, une branche de la combinatoire qui tente de répondre à des questions de la forme "à partir de quelle taille minimale une structure donnée vérifie-t-elle nécessairement une propriété choisie ?"

Le nombre maximum d'entiers pouvant être colorés par  $k$  couleurs sans somme sera appelé nombre de Schur de  $k$ , ou  $S(k)$ . On va définir plus rigoureusement  $S(k)$ .

## 2 Description mathématique

On appelle nombre de Schur, noté  $S(k)$ , le plus grand entier  $N$  tel que :

$$\begin{aligned} &\exists A_1 \dots A_k \text{ tels que } \bigcup_{i=1}^k (A_i) = \{1, N\} \\ &\text{et} \\ &\forall i \in \{1, k\} \nexists a, b, c \in A_i \text{ tels que } a + b = c \end{aligned}$$

Dans le cas où  $a \neq b$ , on appellera  $N$  nombre de Schur faible, noté  $WS(k)$ .

On connaît actuellement les valeurs exactes des nombres de Schur jusqu'à  $k = 5$ , et les nombres de Schur faibles jusqu'à  $k = 4$ . Au-delà, on ne connaît que des bornes, supérieures et inférieures. L'enjeu de ce projet est de parvenir à améliorer ces bornes. Il est très difficile d'obtenir les valeurs exactes des nombres de Schur. En effet, pour prouver une borne inférieure, il suffit d'exhiber une partition correspondante. Mais pour prouver une valeur exacte de  $S(k)$ , il faut montrer que toutes les partitions des  $S(k) + 1$  premiers entiers en  $k$  ensembles comportent un ensemble avec somme : la recherche exhaustive devient très vite impossible en termes de temps de calcul.

k	1	2	3	4	5	6	7	8	9	10	11	12
$S(k)$	1	4	13	44	160 [6]							
borne inf						536 [2]	1680 [2]	5041 [5]	15124 [5]	51120 [7]	172216 [7]	575664 [7]
borne sup						1927	13490	107921	971290	8956901	98525912	1182310945
$WS(k)$	2	8	23	66								
borne inf					196 [1]	582 [3]	1740 [4]	5201 [4]	15596 [4]	51520	172216	575664
borne sup					1630	11742	95900	876808	8877690	98641010	1193556233	15624736140
Valeur max atteinte	2	8	23	66	196	582	1740	5201	15596	44455 [9]	127575 [9]	372389 [9]

Les bornes supérieures viennent de [9] et [10]

## 3 Pistes et ressources

On peut aborder le problème selon deux angles : plutôt informatique, en cherchant des heuristiques qui amélioreraient les bornes inférieures de l'état de l'art, ou plutôt mathématique, en cherchant des formules générales sur les nombres de Schur ou en trouvant des constructions systématiques de partitions.

Pour le premier aspect, nous avons choisi de nous intéresser d'abord à l'article de B. Bouzy qui décrit une heuristique utilisant l'algorithme de Nested Monte-Carlo pour créer des partitions sans-somme à partir de partitions plus petites. Un premier objectif est d'implémenter son algorithme, puis de le modifier grâce à des idées trouvées dans le reste de la bibliographie pour concevoir une heuristique plus performante.

Pour le deuxième aspect, nous nous sommes penchés sur l'article d'Abbot-Hanson [7] qui prouve une relation récursive sur les nombres de Schur. Dans un premier temps, nous avons compris les démonstrations de l'article, le nouvel objectif est de l'utiliser pour construire réellement les partitions correspondant aux bornes inférieures obtenues grâce aux formules.

## 4 Planning

Divers articles seront étudiés tout au long du projet.

En parallèle, des codes seront implémentés pour tenter de trouver des bornes les plus hautes possibles.

Les prochaines étapes sont l'étude des résultats obtenus avec l'algorithme de Nested Monte-Carl et de la faisabilité d'un algorithme basé sur la preuve du théorème de Abbott-Hanson

## 5 Avancement

### 5.1 Articles lus

#### An abstract procedure to compute weak Schur numbers, B.Bouzy

Une bonne solution de  $WS(n+1)$  semble contenir une bonne solution de  $WS(n)$ . De plus, on remarque que les bonnes solutions contiennent des suites de nombres presque consécutifs. On exécute donc un algorithme récursif de la façon suivante:

- On trouve une bonne partition de  $WS(n-1)$
- On ajoute une suite de nombres presque consécutifs dans k-ième colonne (environ  $2WS(n-1)$  nombres)
- On remplit les k-1 premières colonnes avec une bnne solution

On peut optimiser cet algorithme en recherchant des trous spécifiques dans les suites de nombres, qui permettent de mettre plus de nombres par la suite, et en calculant une solution exacte pour  $k=3$  avant de lancer la récurrence.

On peut également adapter cet algorithme pour pouvoir utiliser l'algorithme de Monte-Carlo.

#### Lower bounds for $S(n)$ , Abbott et Hanson

On généralise le problème de Schur, le problème devient :

Soient  $a_1, a_2, \dots, a_N \in \mathbb{Z}$ . Trouver  $f(m)$ , le plus grand entier tel qu'il existe une m-partition de  $\{1, f(m)\}$  pour laquelle, dans chaque sous-espace, il n'existe pas  $x_1, x_2, \dots, x_N$  vérifiant :

$$\sum a_i x_i = 0$$

Schur correspond au cas où  $a_1 = a_2 = 1$  et  $a_3 = -1$

Notations :

- $A = \sum_{i=1}^t a_i > B = \sum i = t + 1^N a_i$
- M, N entiers tels que:  
 $(A-1)f(m) \leq M < N$   
 $1 \leq N \leq \{\frac{A}{A-1}M\}$
- $\mu = -B + 1, -B + 2 \dots A - 1$  si  $N < \frac{A}{A-1}M$   
 $\mu = -B + 1, -B + 2 \dots A - 2$  sinon
- $H(M, N)$  est le plus petit nombre permaetant de partitionner  $[1, N]$  en évitant :  
 $\sum_{i=1}^t a_i x_i = \sum i = t + 1^N a_i x_i + \mu N$
- $h(M) = \min H(M, N) = H(M_1, N_1)$  avec M, N vérifiant les hypothèses précédentes

Alors on a :

$$f(n + h(m)) \geq N_1 f(n) + f(m)$$

De plus, on a également :

$$f(m + n) \geq (2f(m) + 1)f(n) + f(m)$$

ce qui, appliqué à Schur, donne:

$$S(m + n) \geq (2S(m) + 1)S(n) + S(m)$$

De plus, la preuve de ces théorème est constructive, et fournit une piste intéressante pour développer de nouveaux algorithmes.

## 5.2 Codes réalisés

Une représentation intéressante des éléments d'une partition est de les considérer comme un tableau de booléens, le  $i$ -ème nombre valant 1 si  $i$  est présent dans la partition. Cela permet de vérifier très rapidement si un nombre peut être ajouté dans une partition en respectant la propriété sum-free.

### algorithme exhaustif

Dès  $n=4$ , l'algorithme prend trop de temps pour s'exécuter.

### Nested Monte-Carlo

Le code a été mis sous forme d'un exécutable avec quatre paramètres :

- le nombre d'ensembles de la partition
- le niveau
- le nombre d'iterations au niveau 0
- le nombre de simulation de haut niveau

L'algorithme renvoie la meilleure partition sans-somme trouvée ainsi que l'écart-type et la moyenne des partitions optimales générées au plus haut niveau.

La moyenne augmente globalement lorsque le niveau ou le nombre d'iterations augmente, tandis que l'écart type diminue. Les performances augmentent plus avec le niveau qu'avec le nombre d'iterations.

### Algorithme de Abbott-Hanson

L'idée est de reprendre la démonstration du théorème de Abbott et Hanson [7], afin de créer des partitions à  $n + m$  ensembles à partir de partitions à  $n$  et  $m$  ensembles. Il faut pour cela d'abord trouver une bonne partition à  $n$  et à  $m$  ensembles, mais non optimale, sinon l'algorithme se bloque, puis, à partir de la nouvelle partition créée, on lance l'algorithme de Monte-Carlo pour améliorer le résultat.

## 6 Annexes

Des slides, supports pour une présentation orale, ont été réalisées à chaque séance.

Les différents codes se trouvent sur un git.

# Bibliography

- [1] S. Eliahou, J. M. Marin, M. P. Revuelta and M. I. Sanz. Weak Schur numbers and the search for G. W. Walker's lost partitions, in *Computers Mathematics with Applications*, Vol. 63, 2012, 175–182.
- [2] H. Fredricksen and M. M. Sweet. Symmetric sum-free partitions and lower bounds for Schur numbers, in *Electronic Journal of Combinatorics*, Vol. 7, R32, 2000.
- [3] S. Eliahou, C. Fonlupt, J. Fromentin, V. Marion-Poty, D. Robilliard and F. Teytaud. Investigating Monte-Carlo methods on the weak Schur problem, in *Evolutionary Computation in Combinatorial Optimization*, Vol. 7832 of *Lecture Notes in Computer Science*, Springer, Berlin, 2013, 191–201.
- [4] F. Rafilipojoana. LOWER BOUNDS ON THE WEAK SCHUR NUMBERS UP TO 9 COLORS.
- [5] I. Schur. Über die Kongruenz  $xm + ym = zm \pmod{p}$ , in *Jahresbericht der Deutschen Mathematiker-Vereinigung*, Vol. 25, 1917, 114–116
- [6] M. Heule. Schur Number Five, in the Thirty-Second AAAI Conference on Artificial Intelligence
- [7] Abbott and Hanson, A problem of Schur and its generalizations, *Acta Arithmetica* XX
- [8] A lower bound for weak Schur numbers with a deterministic algorithm, Ghada Ben Hassine, Pierre Bergé, Arpad Rimmel, Joanna Tomasik
- [9] Irving, 1973, An extension of Schur's theorem on sum-free partition
- [10] P. Bornshtein. On an extension of a theorem of Schur. *Acta Arith.*, 101:395–399, 2002