Encore de la NP-complétude

François Schwarzentruber

20 janvier 2021

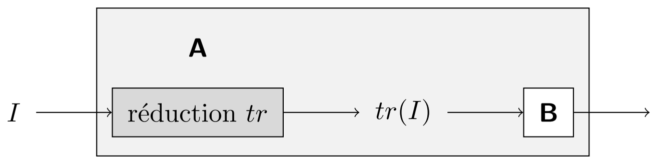
Bonne année !

1 Rappel

Définition 1 Un problème de décision est dans NP s'il existe un algorithme non-déterministe qui le décide en temps polynomial.

Définition 2 (Réduction polynomiale) Une *réduction polynomiale* d'un problème **A** à un problème **B** est une fonction tr, qui, à toute instance I de **A**, associe une instance $tr(I)$ de **B**, telle que

1. pour tout I de **A**, I est une instance positive de **A** ssi $tr(I)$ est une instance positive de **B** ;
2. $tr(I)$ calculable en temps $poly(|I|)$.



Définition 3 Un problème de décision est NP-dur si tout problème dans NP s'y réduit en temps polynomial.

Définition 4 Un problème est NP-complet s'il est dans NP et NP-dur.

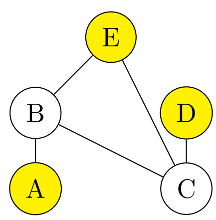
2 Mon premier catalogue de problèmes NP-complets

2.1 ENSEMBLE INDÉPENDANT

Application 5 Personnes à inviter. Relation d'incompatibilité. But : inviter le maximum de personnes.

Définition 6 (ensemble indépendant) Soit un graphe $G = (V, E)$ non orienté. Un ensemble indépendant est un ensemble de sommets $C \subseteq V$ tel que pour tout $(u, v) \in E$, $u \notin C$ ou $v \notin C$.

Exemple 7



Définition 8 (problème d'optimisation) **INDEPENDENT SET**

entrée : un graphe $G = (V, E)$ non orienté
sortie : un ensemble indépendant de cardinal maximal.

Définition 9 (problème de décision) **INDEPENDENT SET**

entrée : un graphe $G = (V, E)$ non orienté, k un entier
sortie : oui, s'il existe un ensemble indépendant de cardinal $\geq k$.

2.2 VERTEX COVER

Application 10 Placer le minimum de gardiens pour surveiller tous les couloirs (= arêtes).

Application 11 Infecter un nombre min de machines pour que rapidement toutes les machines le soient. [Fil09][p. 375]

Application 12 Bio-informatique [MDK18].

Définition 13 (couverture de sommets) Soit un graphe $G = (V, E)$. Une couverture de sommet est un ensemble $C \subseteq V$ tel que pour tout $(u, v) \in E$, $u \in C$ ou $v \in C$.

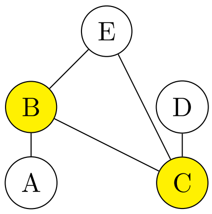
Définition 14 (problème d'optimisation) **VERTEX COVER**

entrée : un graphe $G = (V, E)$ non orienté
sortie : une couverture de sommets de cardinal minimal.

Définition 15 (problème de décision) **VERTEX COVER**

entrée : un graphe $G = (V, E)$ non orienté, k un entier
sortie : oui, s'il existe une couverture de sommets de cardinal $\leq k$.

Exemple 16



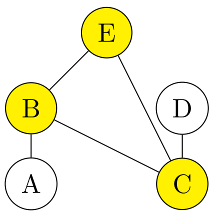
2.3 CLIQUE

Application 17 Trouver un groupe de personnes qui se connaissent toutes.

Application 18 En bioinformatique [MCC+12].

Définition 19 (clique) Soit un graphe $G = (V, E)$ non orienté. Une clique est un ensemble de sommets $C \subseteq V$ tel que $C^2 \subseteq E$.

Exemple 20



Définition 21 (problème d'optimisation) **CLIQUE**

entrée : un graphe $G = (V, E)$ non orienté
sortie : une clique maximale.

Définition 22 (problème de décision) **CLIQUE**

entrée : un graphe $G = (V, E)$ non orienté, k un entier
sortie : oui, s'il existe une clique de cardinal $\leq k$.

2.4 3D Matching

Définition 23 **3D Matching**

entrée : trois ensembles finis A, B, C de même cardinal n, disjoints deux à deux, et une relation $E \subseteq A \times B \times C$
sortie : oui s'il existe $R \subseteq E$ tel que tout élément de $A \cup B \cup C$ apparaît dans exactement un triplet de R, non sinon.

2.5 Équations zero-un

Application 24 Cas particulier de la programmation linéaire entière.

Définition 25 **Équations zero-un**

entrée : une matrice booléenne A de taille $m \times n$ avec des éléments de $\{0, 1\}$
sortie : oui, s'il existe un vecteur $x \in \{0, 1\}^n$ tel que $Ax = 1$, non sinon.

Exemple 26

$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} ? \\ ? \\ ? \\ ? \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

2.6 SUBSET SUM

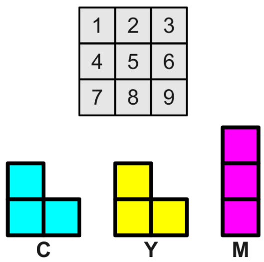
Définition 27 **SUBSET SUM**

entrée : un ensemble fini E d'entiers strictement positifs, un entier s ;
sortie : oui, s'il existe un sous-ensemble de E dont la somme vaut s, non sinon.

Exemple 28 $E = \{1, 3, 4, 9, 15, 32\}$ et $s = 30$.

2.7 EXACT COVER

Application 29 Ranger des objets.



{ C, 1, 4, 5 }
{ C, 1, 2, 4 }
{ C, 1, 2, 5 }
{ C, 2, 4, 5 }
{ M, 1, 4, 7 }
{ M, 2, 5, 8 }
{ M, 3, 6, 9 }

Définition 30 **EXACT COVER**

entrée : un ensemble U fini, une collection $S \subseteq 2^U$ de sous-ensembles de U
sortie : oui, s'il existe $I \subseteq S$ tel que tout élément de U appartient à exactement un sous-ensemble de I, non sinon.

Exemple 31 $U = \{1, 2, 3, 4, 5, 6\}$.
$S = \{\{1, 2, 5\}, \{2, 5\}, \{1, 3, 6\}, \{2, 3, 4\}, \{4\}, \{1, 5, 6\}\}$.

2.8 Sac à dos

Exemple 32 But : remplir un sac de poids en maximisant la valeur et sans dépasser un poids maximal $P = 10$.

Objet i	Poids p_i	Valeur v_i
1	6	30€
2	3	14€
3	4	16€
4	2	9€

Solution avec répétition : objets 1, 4, 4. Solution sans répétition : objets 1 et 3.

Définition 33 **Problème d'optimisation du sac à dos avec répétition (sans répétition)**

entrée : une collection d'objets $(p_i, v_i)_{i=1..n}$, un poids maximal P
sortie : la valeur maximale de $\sum_{i=1}^{n} n_i v_i$ avec $\sum_{i=1}^{n} n_i p_i \leq P$ et n_i dans $\mathbb{N}$ (dans $\{0, 1\}$)

Définition 34 **Problème de décision du sac à dos avec répétition (sans répétition)**

entrée : une collection d'objets $(p_i, v_i)_{i=1..n}$, un poids maximal P
sortie : oui s'il existe des n_i dans $\mathbb{N}$ (dans $\{0, 1\}$) tel que $\sum_{i=1}^{n} n_i v_i \geq V$ avec $\sum_{i=1}^{n} n_i p_i \leq P$

2.9 Cycle hamiltonien

Définition 35 (un cycle hamiltonien) Soit un graphe $G = (V, E)$ non orienté. Un cycle hamiltonien est un cycle dans G qui passent exactement une et une seule fois par chaque sommet de G.

Définition 36 **CYCLE HAMILTONIEN**

entrée : un graphe $G = (V, E)$ non orienté
sortie : oui, si G admet un cycle hamiltonien, non sinon.

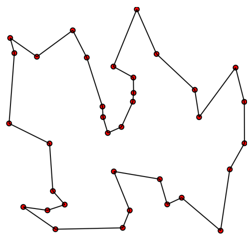
2.10 Voyageur de commerce

Application 37 Organisation d'un voyage.

Application 38 Minimiser le déplacement d'un forêt qui doit percer des trous.

Définition 39 (tour) Soit $G = (V, E)$ un graphe non orienté complet et $d : V \times V \to \mathbb{N}$ avec $d(i, j) = d(j, i)$. Un tour τ dans G est un cycle hamiltonien dans G. Son poids est $\sum_{e \text{ dans } \tau} d(e)$.

Exemple 40



Définition 41 (problème d'optimisation) **TSP**

entrée : un graphe non orienté complet $G = (V, E)$ et $d : V \times V \to \mathbb{N}$ avec $d(i, j) = d(j, i)$
sortie : un tour dans G de poids minimal

Définition 42 (problème de décision) **TSP**

entrée : un graphe non orienté complet $G = (V, E)$ et $d : V \times V \to \mathbb{N}$ avec $d(i, j) = d(j, i)$, k un entier
sortie : oui, s'il y a un tour dans G de poids $\leq k$; non sinon.

2.11 Multi-agent path finding

Application 43 Déplacement de robots dans un entrepôt. `http://mapf.info/`

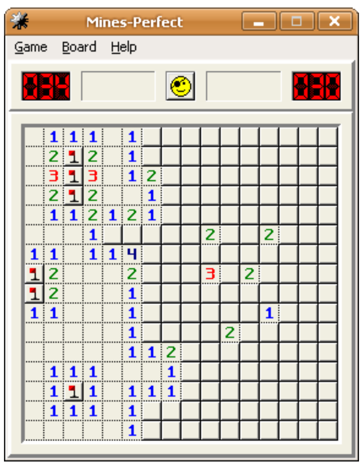
Définition 44 **MAPF**

entrée : un graphe non orienté $G = (V, E)$, des sommets initiaux $s_1, \ldots, s_n$, des sommets destination $t_1, \ldots, t_n$
sortie : des chemins $s_i \to^* t_i$ sans collision pour $i = 1..n$ tel que le maximum des longueurs des chemins soient minimal.

2.12 Démineur généralisé

Définition 45 **Démineur généralisé**

entrée : Une grille $n \times m$ chacune pouvant être blanche, ou contenir un chiffre entre 0 et 8
sortie : oui, s'il existe une disposition des mines dans les cases blanches qui corresponde aux informations disponibles, non sinon.



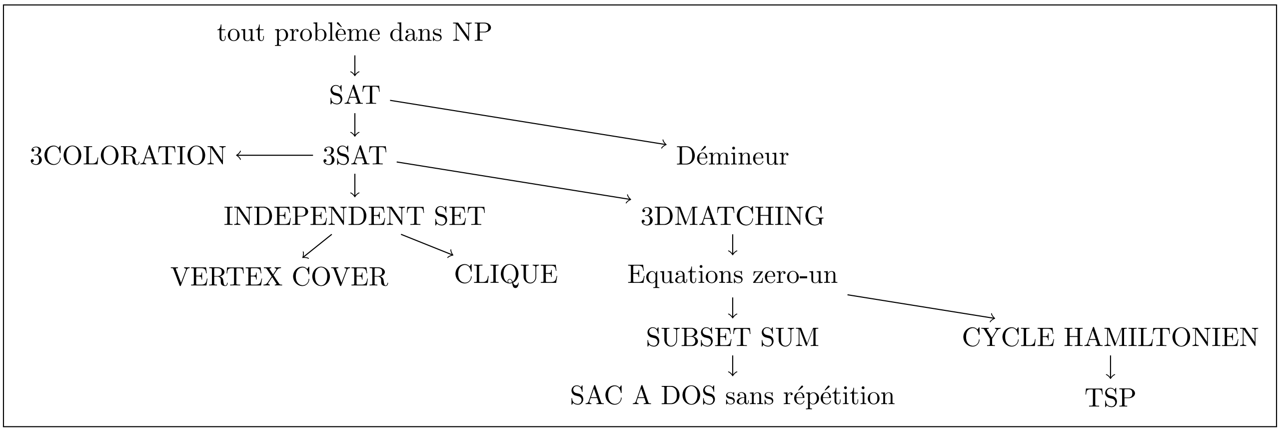
3 Frontières P/ NP

dans P	NP-complet
2SAT	**3SAT**
HORN-SAT	
ensembles indépendants sur les arbres	**ensembles indépendants**
couverture de sommets dans un graphe biparti	**couverture de sommets**
2-coloration	**3-coloration**
Arbre couvrant minimal	**Arbre de Steiner**
cycle eulérien	**cycle hamiltonien**
Sac à dos unaire	**Sac à dos**
Programmation linéaire réelle	**Programmation linéaire entière/mixte**
Flot max	

4 Que faire face à un problème NP-complet ?

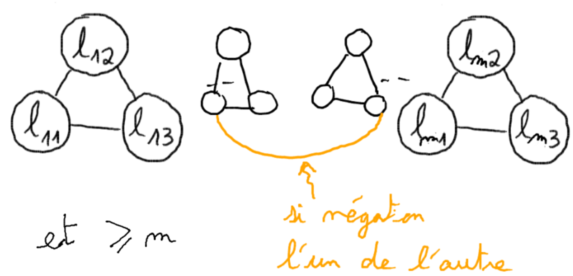
- Utiliser des solveurs SAT ou des solveurs de programmation linéaire mixte/entier
- Algorithme de recherche : backtracking, branch and bound, backjumping
- Recherche locale, recuit simulé
- Algorithmes d'approximation
- Algorithmes probabilistes
- Complexité paramétrée (exemple : tree-width)
- Algorithmique quantique

5 Exercices



Exercice 46 Montrer que **INDEPENDENT SET** est NP-complet.
Indication : La NP-dureté peut se démontrer par une réduction depuis **3SAT**.

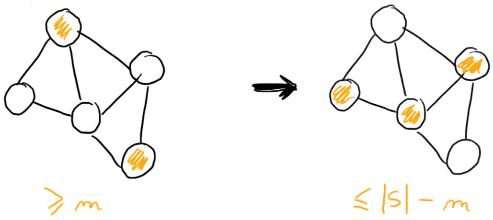
$$\bigwedge_{i=1}^{m} (l_{i1} \vee l_{i2} \vee l_{i3}) \Rightarrow$$



et $\geq m$

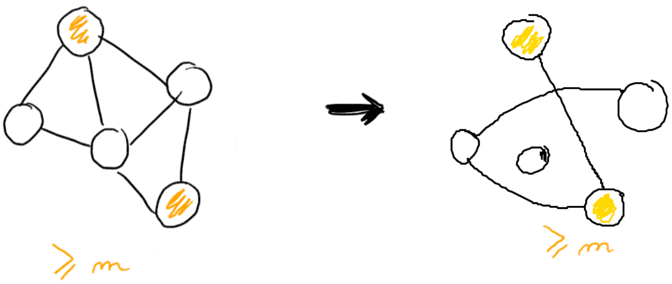
Exercice 47 Montrer que **VERTEX COVER** est NP-complet.

Indication : Réduction depuis **INDEPENDENT SET**.



Exercice 48 Montrer que **CLIQUE** est NP-complet.

Indication : Réduction depuis **INDEPENDENT SET**.



Exercice 49 **3D MATCHING** est NP-complet.

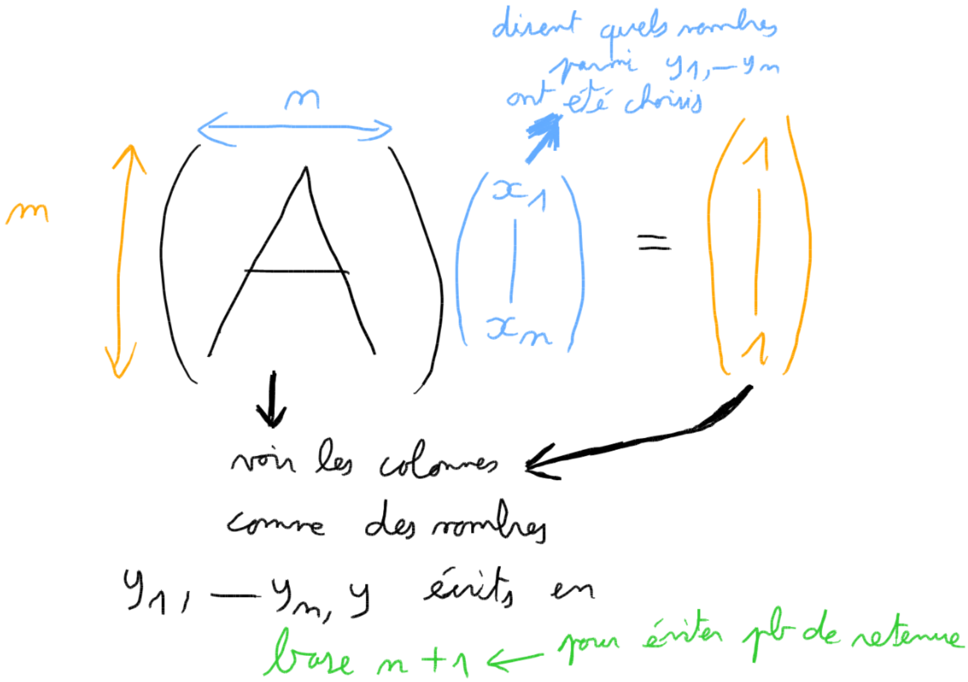
Indication : Réduction depuis 3SAT, cf. exam ALGO1

Exercice 50 Montrer que **Équations zero-un** est NP-complet.

Indication : Réduction depuis **3D MATCHING**. *Les variables sont $x_1, \ldots, x_n$ et x_i signifie "le triplet numéro i est sélectionné".*

Exercice 51 Montrer que **SUBSET SUM** est NP-complet.

Indication : Réduction depuis **Équations zero-un**.



Exercice 52 Montrer que **SAC A DOS sans répétition** est NP-complet.

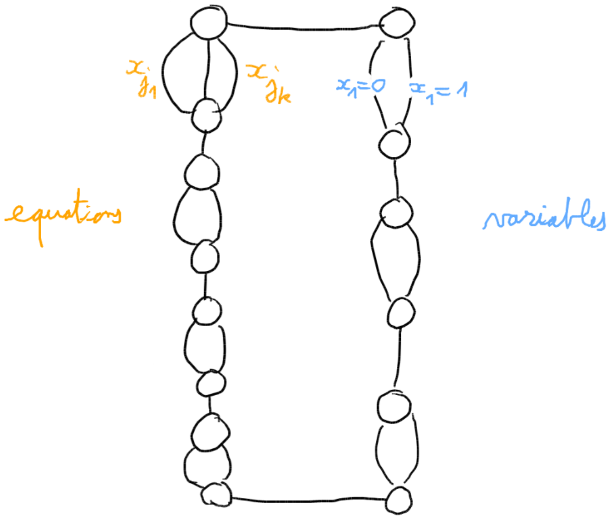
Indication : Réduction depuis **SUBSET SUM**.

Exercice 53 L'objectif de l'exercice est de démontrer que **CYCLE HAMILTONIEN** est NP-complet. On introduit d'abord le problème intermédiaire suivant :

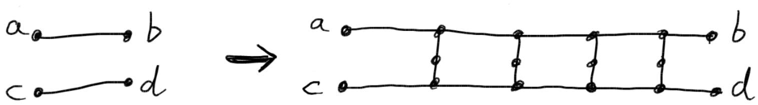
CYCLE HAMILTONIEN AVEC COUPLES D'ARETES

entrée : un graphe $G = (V, E)$ non orienté, un ensemble C de couples d'arêtes
sortie : oui, si G admet un cycle hamiltonien qui, pour chaque couple (e, e') de C emprunte e ou e', non sinon.

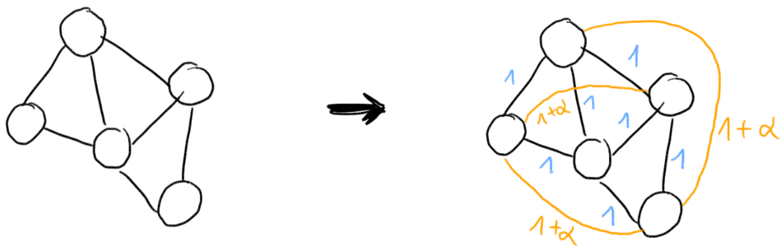
1. Montrer que **CYCLE HAMILTONIEN AVEC COUPLES D'ARETES** est NP-dur. *Indication : Par réduction depuis le problème* **ZOE**, *cf. dessin et en prenant* C = *les couples (occurrence de* x_i, *arête* $x_i := 0$*).*



2. Réduire de **CYCLE HAMILTONIEN AVEC COUPLES D'ARETES** dans **CYCLE HAMILTONIEN**. *Indication : Pour toute paire* $(\{a, b\}, \{c, d\})$ *de* C, *on ajoute le gadget suivant :*



Exercice 54 **TSP** est NP-complet. *Indication : Réduction depuis* **CYCLE HAMILTONIEN**.



Exercice 55 Nous allons esquisser la démonstration de la NP-complétude du **Démineur généralisé**.

1. Justifier que **Démineur généralisé** est dans NP.
2. On considère maintenant une instance de **SAT** vu comme un circuit. On simule un fil par le gadget suivant :

X ⟶

⋯	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	⋯
⋯	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	⋯
⋯	x	1	x'	x	1	x'	x	1	x'	x	1	x'	x	1	x'	x	⋯
⋯	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	⋯
⋯	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	⋯

On suppose que fil fait circuler la valeur 1 si les cases x contiennent des mines (et pas x'), ou la valeur 0 si les cases x' contiennent des mines. Répondez aux questions suivantes en lisant `http://www.minesweeper.info/articles/MinesweeperIsNPComplete.pdf`.

(a) Donner un gadget permettant de commencer/terminer un fil sans contraindre la valeur. Puis donner un gadget permettant de commencer/terminer un fil en forçant la valeur.

(b) Donner un gadget permettant de faire tourner un fil à angle droit, et un gadget permettant de diviser un fil en trois.
(c) Donner un gadget permettant de simuler une porte NON.
(d) En supposant que l'on a un gadget pour faire le ET, et de croiser des fils, conclure la démonstration.

6 Devoir maison

DM au choix :

1. 3SAT vers 3COLORATION,
2. 3SAT vers INDEPENDENT SET,
3. Equations zero-un vers CYCLE HAMILTONIEN,
4. SAT vers Démineur (difficile)
5. ou une autre ou plusieurs autres réductions que vous trouvez intéressante.s.

7 Notes bibliographiques

Ce cours est issu du livre [DPV08]. Il reprend l'essentiel du papier fondateur [Kar72]. Le livre [GJ79], assez vieux de Garey et Johnson, reste toujours d'actualité : une collection d'au moins 300 problèmes NP-complets. Vous trouverez une liste ici : `https://en.wikipedia.org/wiki/List_of_NP-complete_problems`

Références

[DPV08] Sanjoy Dasgupta, Christos H. Papadimitriou, and Umesh V. Vazirani. *Algorithms*. McGraw-Hill, 2008.

[Fil09] É. Filiol. *Les virus informatiques : théorie, pratique et applications*. Springer, 2009.

[GJ79] M. R. Garey and David S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[Kar72] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972.

[MCC+12] Tobias Marschall, Ivan G. Costa, Stefan Canzar, Markus Bauer, Gunnar W. Klau, Alexander Schliep, and Alexander Schönhuth. CLEVER : clique-enumerating variant finder. *Bioinformatics*, 28(22) :2875–2882, 10 2012.

[MDK18] Guillaume Marçais, Dan DeBlasio, and Carl Kingsford. Asymptotically optimal minimizers schemes. *Bioinformatics*, 34(13) :i13–i22, 06 2018.