Audition pour le poste de Maître de Conférences

Université de Montpellier · LIRMM

Mathieu Mari

12 mai 2023

Parcours

Enseignement

Recherche

Parcours

2023 2022 2021 — Varsovie

2020 2019 2018 — ENS Ulm

2017 — CNAM

2016 2015 2014 — ENS Rennes

2013 — Nice

Postdoc

IDEAS-NCBR: *algorithmes en ligne, intelligence artificielle*

Université de Varsovie: *Géométrie algorithmique, algos FPT*

Doctorat

Thèse: *Algorithmique* (C. Mathieu, C.-C. Huang)

Enseignement: : UPMC et Paris-Diderot (204h)

Visite (Santiago, Chili)
Algos FPT
Algo en ligne

Licence-Master

Master 2 : *Recherche Opérationnelle*

Master 2 : MEEF

Agrégation : Mathématiques,
option Informatique (rang : 34)

Master 1 : Mathématiques

Licences : Mathématique + Informatique

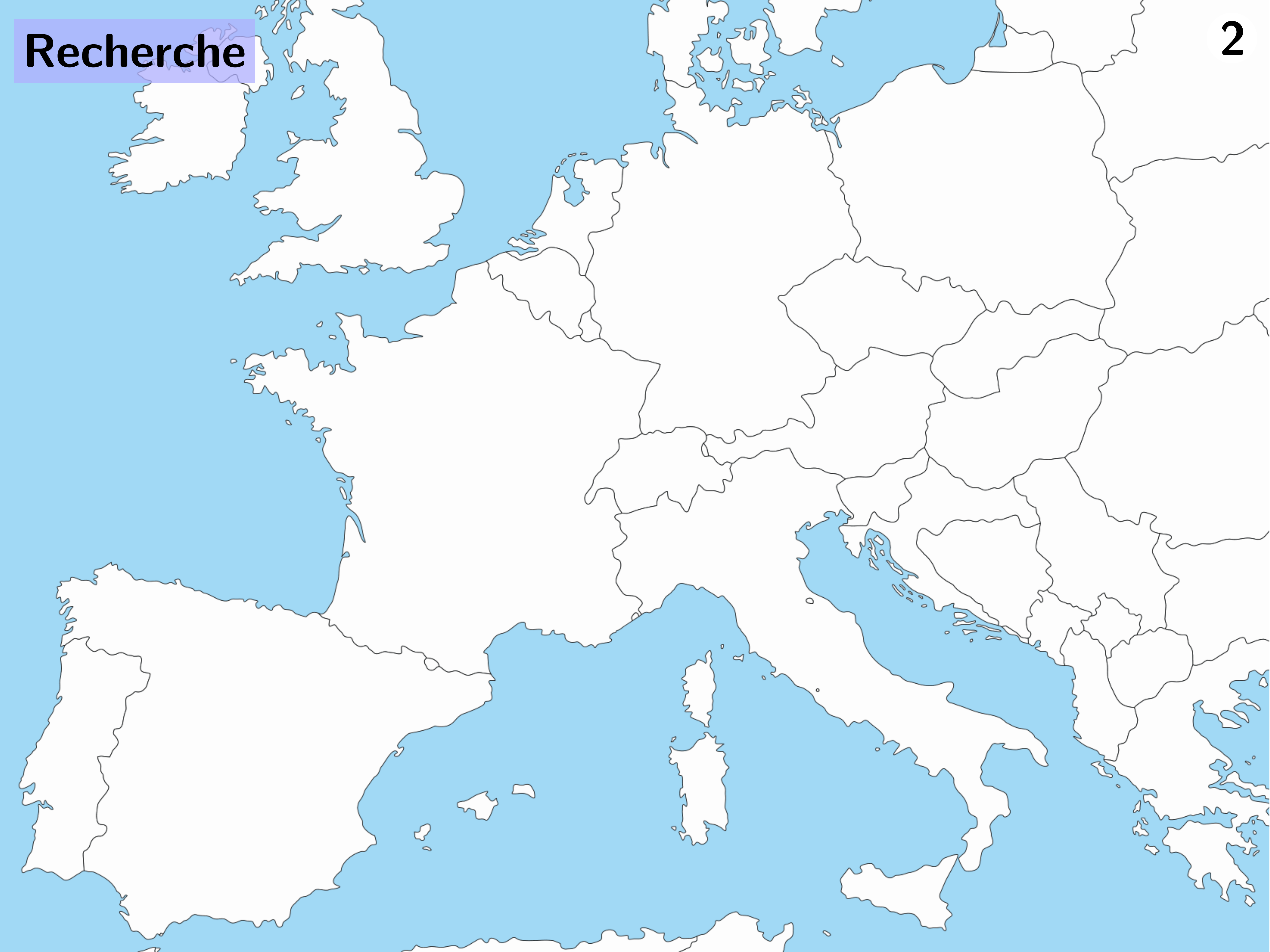
MPSI - MP*

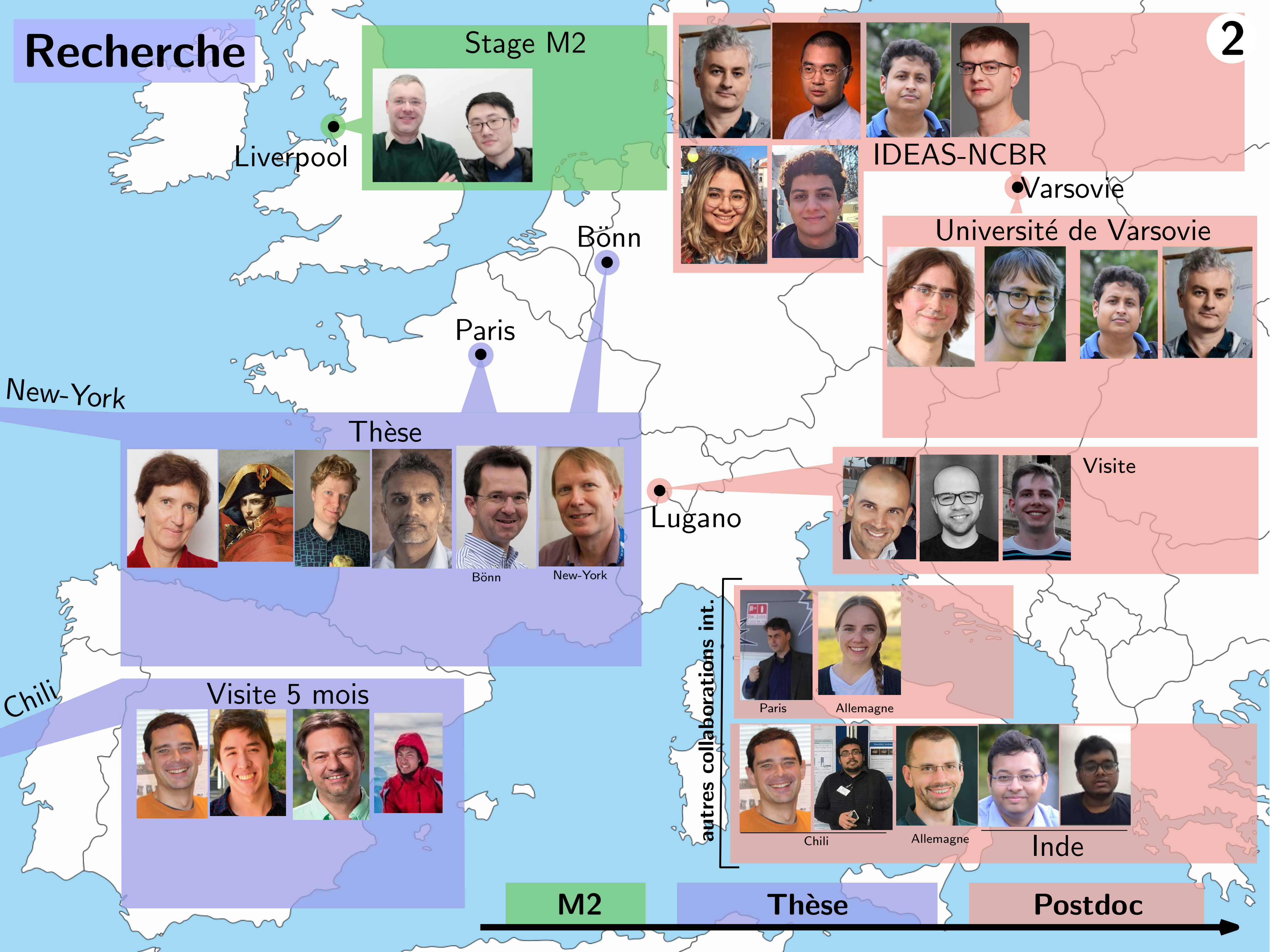
Stage M2 (Liverpool)
Algos d'approximation

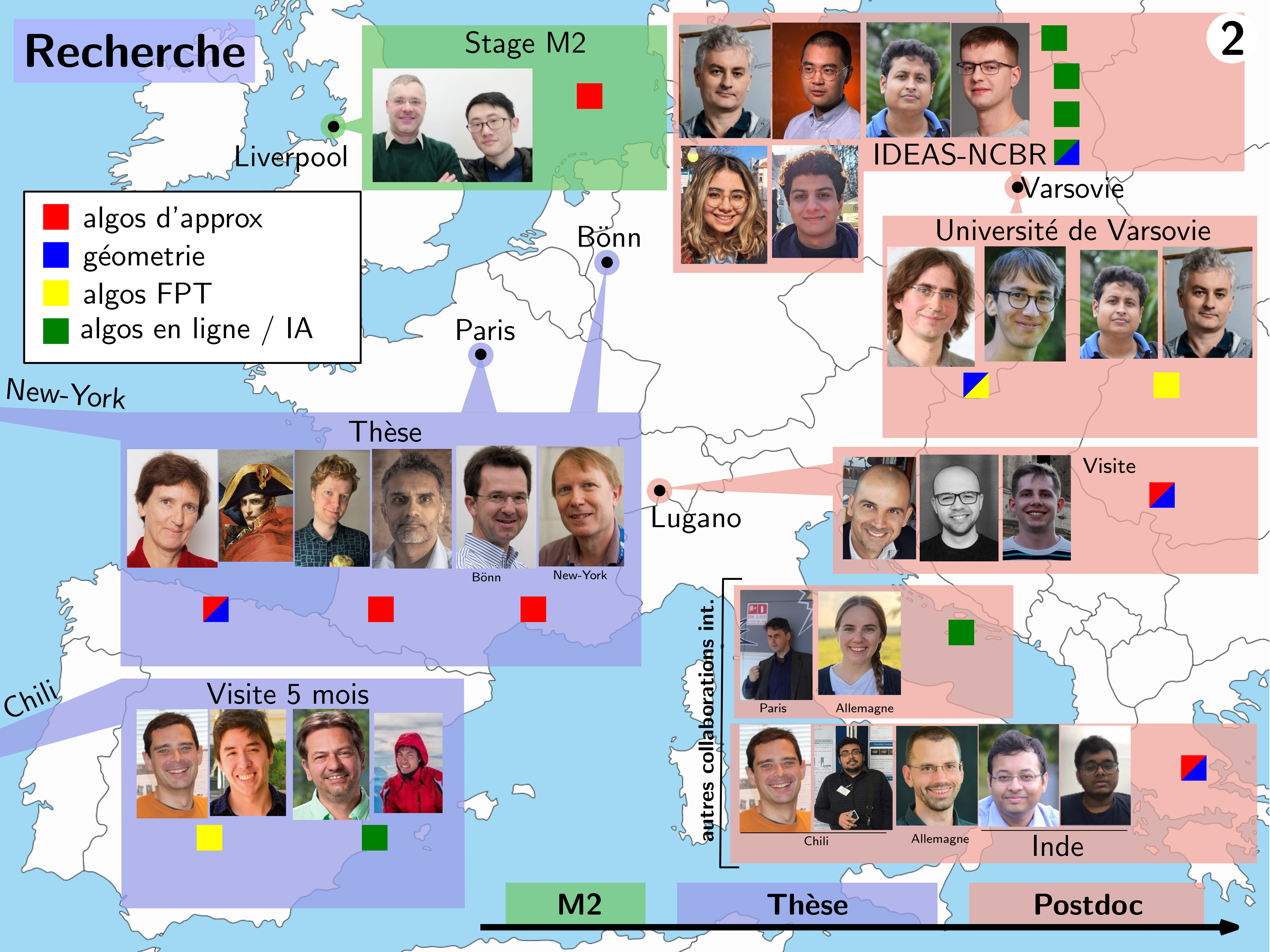
Stage M1 (Le cap)
coloration, graphes

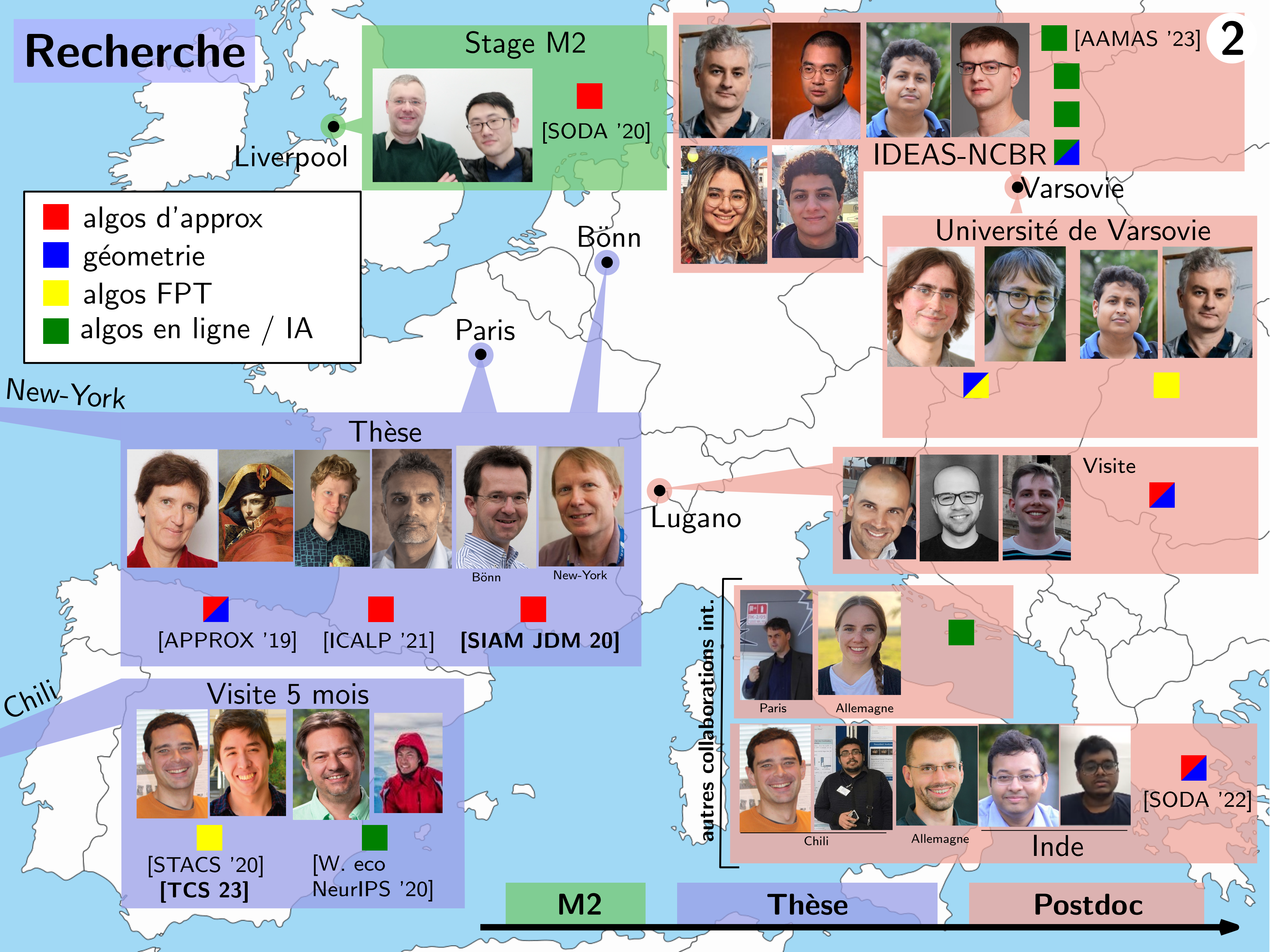
Stage L3 (Lille)
Topologie algébrique

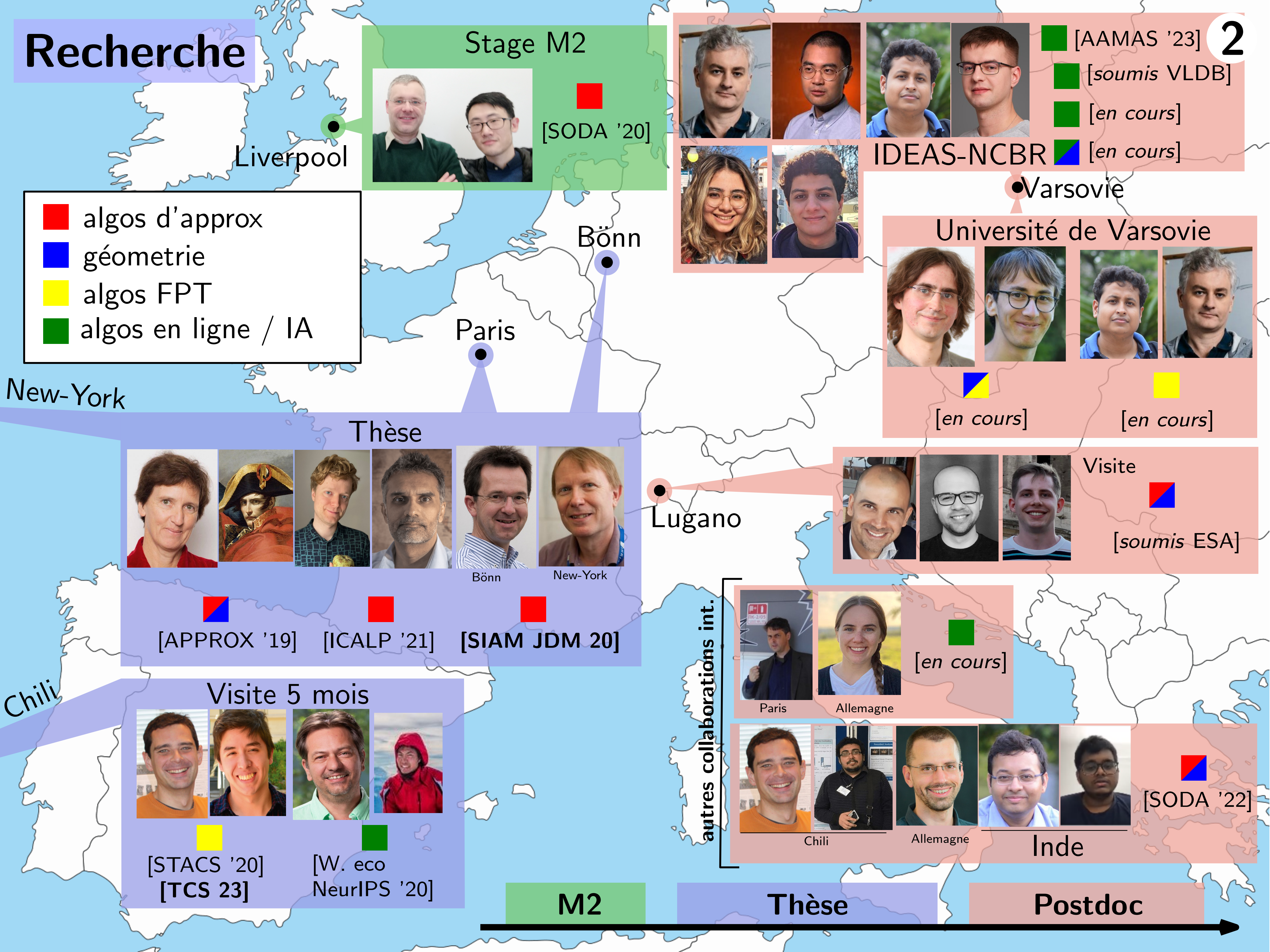
Recherche

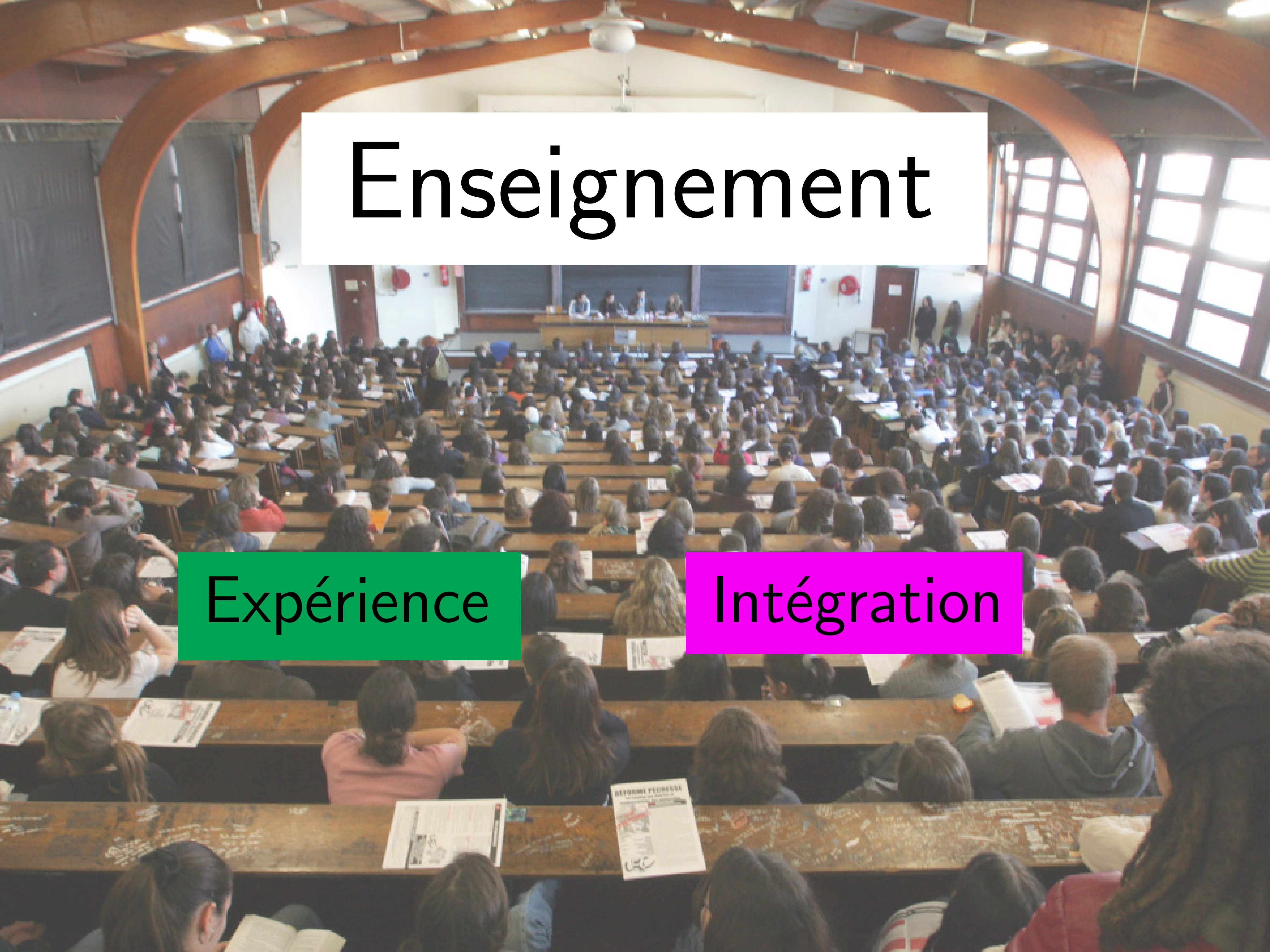



Recherche
2
Stage M2
Liverpool
IDEAS-NCBR
Varsovie
Université de Varsovie
Bönn
Paris
New-York
Thèse
Bönn
New-York
Lugano
Visite
Chili
Visite 5 mois
autres collaborations int.
Paris
Allemagne
Chili
Allemagne
Inde
M2
Thèse
Postdoc


Recherche
2
algos d'approx
géometrie
algos FPT
algos en ligne / IA
Stage M2
Liverpool
IDEAS-NCBR
Varsovie
Université de Varsovie
Bönn
Paris
New-York
Thèse
Bönn
New-York
Lugano
Visite
Chili
Visite 5 mois
autres collaborations int.
Paris
Allemagne
Chili
Allemagne
Inde
M2
Thèse
Postdoc


Recherche
2
Stage M2
Liverpool
[SODA '20]
[AAMAS '23]
IDEAS-NCBR
Varsovie
Université de Varsovie
algos d'approx
géometrie
algos FPT
algos en ligne / IA
Bönn
Paris
New-York
Thèse
Bönn
New-York
[APPROX '19]
[ICALP '21]
[SIAM JDM 20]
Lugano
Visite
Chili
Visite 5 mois
[STACS '20]
[TCS 23]
[W. eco
NeurIPS '20]
autres collaborations int.
Paris
Allemagne
Chili
Allemagne
Inde
[SODA '22]
M2
Thèse
Postdoc


Recherche
2
Stage M2
Liverpool
[SODA '20]
IDEAS-NCBR
[AAMAS '23]
[soumis VLDB]
[en cours]
[en cours]
Varsovie
algos d'approx
géometrie
algos FPT
algos en ligne / IA
Bönn
Paris
New-York
Université de Varsovie
[en cours]
[en cours]
Thèse
Bönn
New-York
Lugano
Visite
[soumis ESA]
[APPROX '19]
[ICALP '21]
[SIAM JDM 20]
autres collaborations int.
Paris
Allemagne
[en cours]
Chili
Visite 5 mois
[STACS '20]
[TCS 23]
[W. eco
NeurIPS '20]
Chili
Allemagne
Inde
[SODA '22]
M2
Thèse
Postdoc


Enseignement
Expérience
Intégration

Enseignements effectués

Algorithmique avancée	30h TD	L3	(info)	U. Varsovie
Algorithmique	12h TD	L3/M1	(Bio-Info)	U. Paris-Diderot
Graphes et Combinatoire	32h TD	L2	(Maths)	UPMC
Python pour les mathématiques	64h TP	L2/L3	(Maths)	
Eléments d'arithmétique, codes correcteurs, cryptographie	32h TD	L2	(Maths)	
Suites, Intégrales et Algèbre linéaire	64h TD	L1	(Maths)	

Encadrement

2 mois

Olivier Idir
(co: O. Serre, P. Ohlmann)

6 mois

Timothé Picavet
(co: Mi. Pilipczuk)

6 mois

Nima Khodaveisi

6 mois

Shanli Alefkhani

Enseignements effectués

Algorithmique avancée	30h TD
Algorithmique	12h TD
Graphes et Combinatoire	32h TD
Python pour les mathématiques	64h TP
Eléments d'arithmétique, codes correcteurs, cryptographie	32h TD
Suites, Intégrales et Algèbre linéaire	64h TD

Besoins

Algorithmique
Complexité, calculabilité, décidabilité

Calcul formel

Cryptographie

Systèmes / Programmation web

Enseignements effectués

Algorithmique avancée	30h TD
Algorithmique	12h TD
Graphes et Combinatoire	32h TD
Python pour les mathématiques	64h TP
Eléments d'arithmétique, codes correcteurs, cryptographie	32h TD
Suites, Intégrales et Algèbre linéaire	64h TD

Besoins

Algorithmique
Complexité, calculabilité, décidabilité

Calcul formel

Cryptographie

Systèmes / Programmation web

Enseignements effectués

Algorithmique avancée	30h TD
Algorithmique	12h TD
Graphes et Combinatoire	32h TD
Python pour les mathématiques	64h TP
Eléments d'arithmétique, codes correcteurs, cryptographie	32h TD
Suites, Intégrales et Algèbre linéaire	64h TD

Besoins

Algorithmique
Complexité, calculabilité, décidabilité

Calcul formel

Cryptographie

Systèmes / Programmation web

Enseignements effectués

Algorithmique avancée	30h TD
Algorithmique	12h TD
Graphes et Combinatoire	32h TD
Python pour les mathématiques	64h TP
Eléments d'arithmétique, codes correcteurs, cryptographie	32h TD
Suites, Intégrales et Algèbre linéaire	64h TD

Besoins

Algorithmique
Complexité, calculabilité, décidabilité

Calcul formel

Cryptographie

Systèmes / Programmation web

Enseignements effectués

Algorithmique avancée	30h TD
Algorithmique	12h TD
Graphes et Combinatoire	32h TD
Python pour les mathématiques	64h TP
Eléments d'arithmétique, codes correcteurs, cryptographie	32h TD
Suites, Intégrales et Algèbre linéaire	64h TD

Besoins

Algorithmique
Complexité, calculabilité, décidabilité

Calcul formel

Cryptographie

Systèmes / Programmation web

projet personnel serveur

Enseignements effectués

Algorithmique avancée	30h TD
Algorithmique	12h TD
Graphes et Combinatoire	32h TD
Python pour les mathématiques	64h TP
Eléments d'arithmétique, codes correcteurs, cryptographie	32h TD
Suites, Intégrales et Algèbre linéaire	64h TD

Autres expertises d'enseignement

Programmation

Recherche Opérationnelle

Capes NSI

UE d'algorithmique avancée (géométrie, algos en ligne)

Besoins

Algorithmique
Complexité, calculabilité, décidabilité

Calcul formel

Cryptographie

Systèmes / Programmation web

Enseignements effectués

Algorithmique avancée	30h TD
Algorithmique	12h TD
Graphes et Combinatoire	32h TD
Python pour les mathématiques	64h TP
Eléments d'arithmétique, codes correcteurs, cryptographie	32h TD
Suites, Intégrales et Algèbre linéaire	64h TD

Autres expertises d'enseignement

- Programmation
- Recherche Opérationnelle
- Capes NSI
- UE d'algorithmique avancée (géométrie, algos en ligne)

Besoins

Algorithmique
Complexité, calculabilité, décidabilité

Calcul formel

Cryptographie

Systèmes / Programmation web

Enseignements effectués

Algorithmique avancée	30h TD
Algorithmique	12h TD
Graphes et Combinatoire	32h TD
Python pour les mathématiques	64h TP
Eléments d'arithmétique, codes correcteurs, cryptographie	32h TD
Suites, Intégrales et Algèbre linéaire	64h TD

Autres expertises d'enseignement

- Programmation
- Recherche Opérationnelle

Capes NSI

UE d'algorithmique avancée (géométrie, algos en ligne)

Besoins

Algorithmique
Complexité, calculabilité, décidabilité

Calcul formel

Cryptographie

Systèmes / Programmation web

Enseignements effectués

Algorithmique avancée	30h TD
Algorithmique	12h TD
Graphes et Combinatoire	32h TD
Python pour les mathématiques	64h TP
Eléments d'arithmétique, codes correcteurs, cryptographie	32h TD
Suites, Intégrales et Algèbre linéaire	64h TD

Besoins

Algorithmique
Complexité, calculabilité, décidabilité

Calcul formel

Cryptographie

Systèmes / Programmation web

Autres expertises d'enseignement

- Programmation
- Recherche Opérationnelle
- Capes NSI

UE d'algorithmique avancée (géométrie, algos en ligne)

Enseignements effectués

Algorithmique avancée	30h TD
Algorithmique	12h TD
Graphes et Combinatoire	32h TD
Python pour les mathématiques	64h TP
Eléments d'arithmétique, codes correcteurs, cryptographie	32h TD
Suites, Intégrales et Algèbre linéaire	64h TD

Autres expertises d'enseignement

- Programmation
- Recherche Opérationnelle
- Capes NSI
- UE d'algorithmique avancée (géométrie, algos en ligne)

Besoins

Algorithmique
Complexité, calculabilité, décidabilité

Calcul formel

Cryptographie

Systèmes / Programmation web

Recherche

Résultat

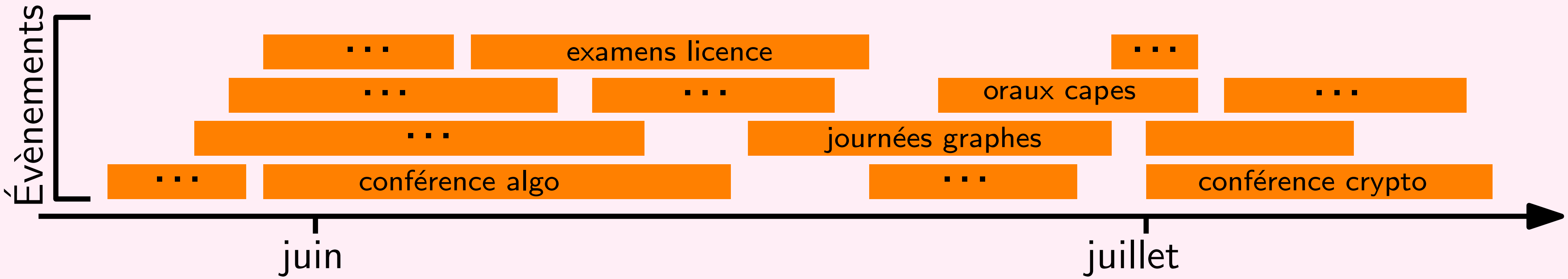
Axe 1

Axe 2

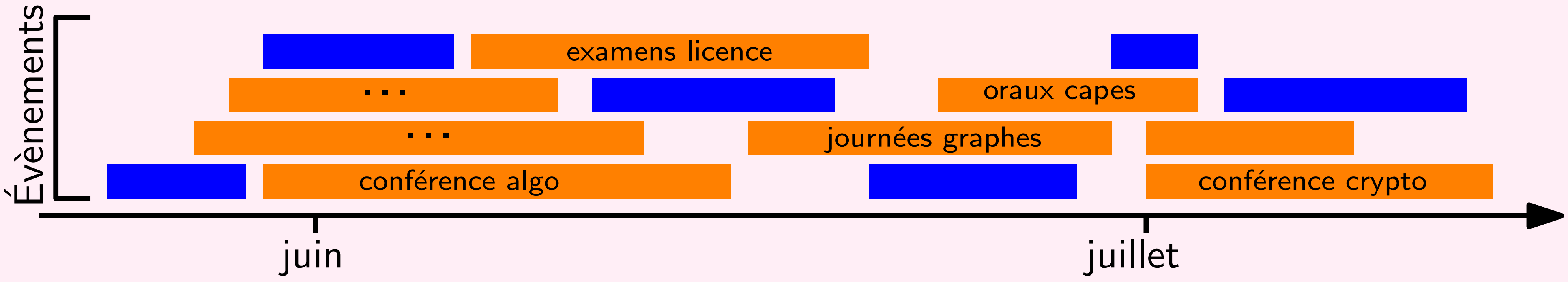
Intégration

projet

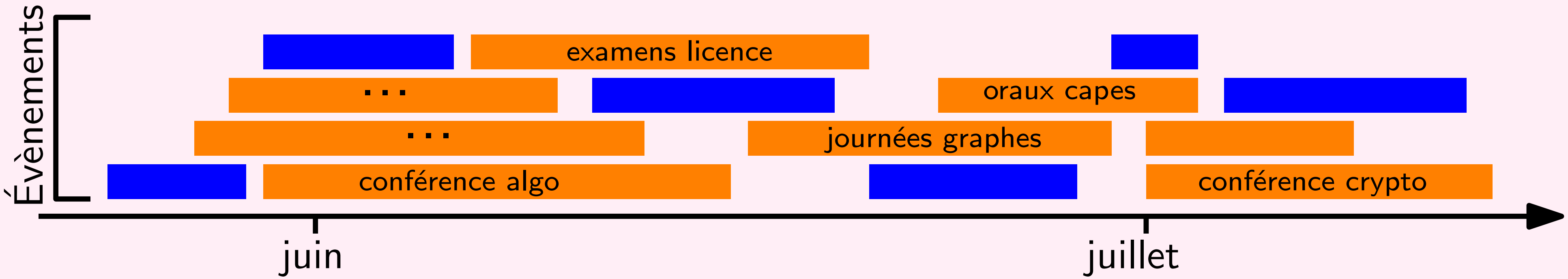
Résultat


Évènements
· · ·
examens licence
· · ·
· · ·
· · ·
oraux capes
· · ·
· · ·
journées graphes
· · ·
conférence algo
· · ·
conférence crypto
juin
juillet

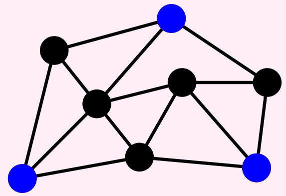
Résultat


Évènements
examens licence
...
oraux capes
...
journées graphes
conférence algo
conférence crypto
juin
juillet

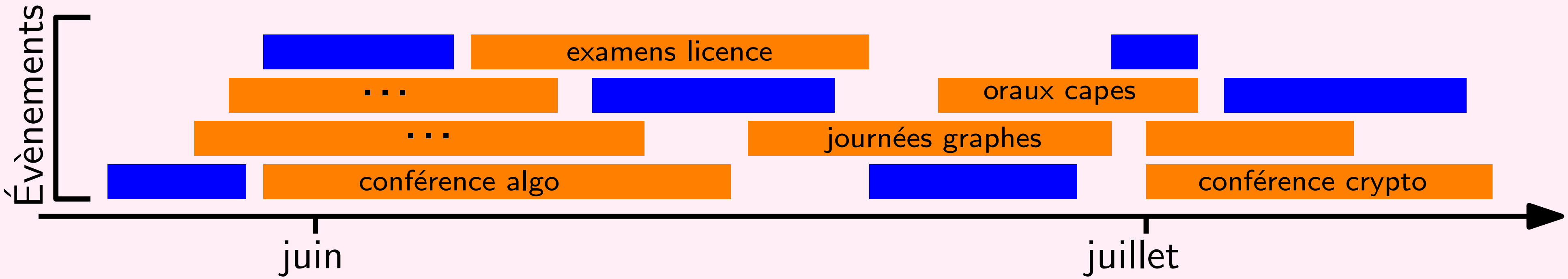
Résultat



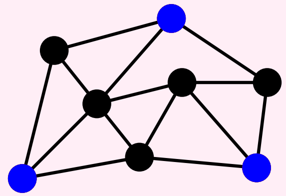
Problème du **stable maximum** sur un **graphe d'intervalles**



Résultat

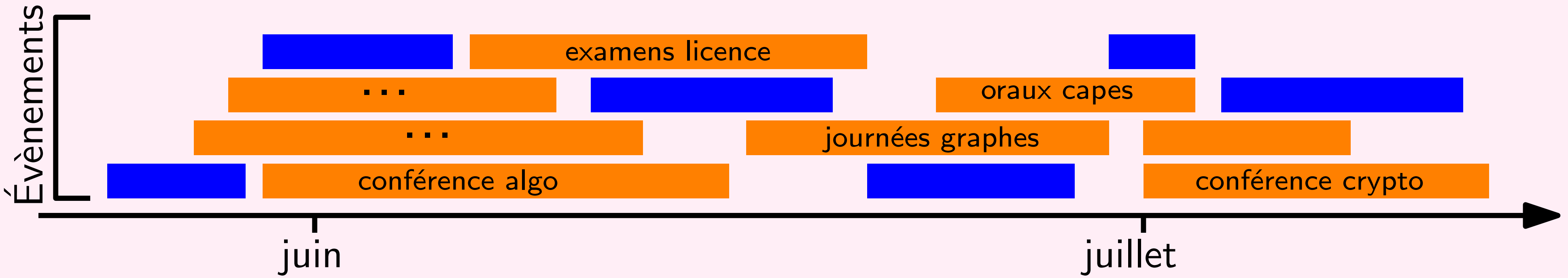


Problème du **stable maximum** sur un **graphe d'intervalles**



- Problème fondamental
- Nombreuses applications

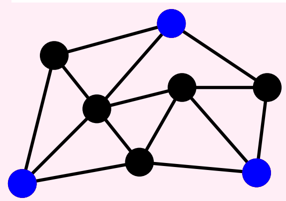
Résultat



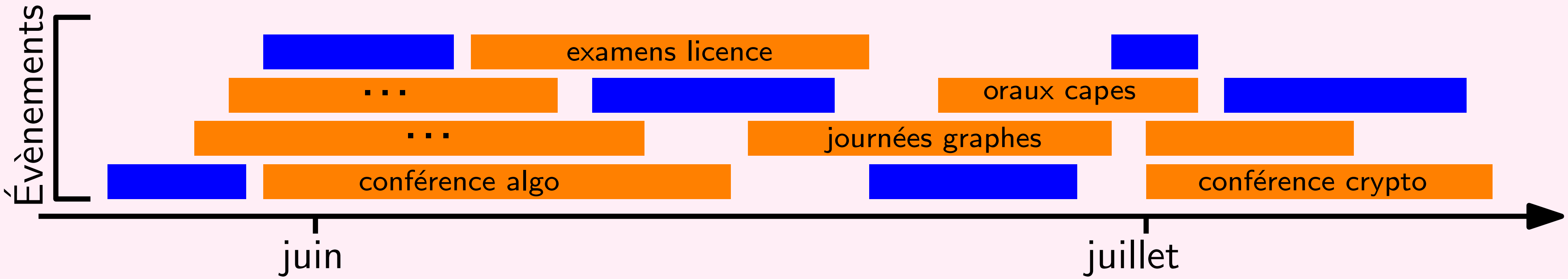
Problème du **stable maximum** sur un **graphe d'intervalles**

- Problème fondamental
- Nombreuses applications

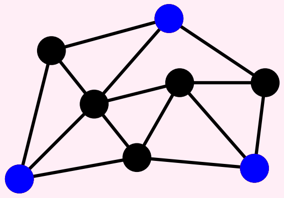
→ Complexité ?
NP-complet



Résultat



Problème du **stable maximum** sur un **graphe d'intervalles**

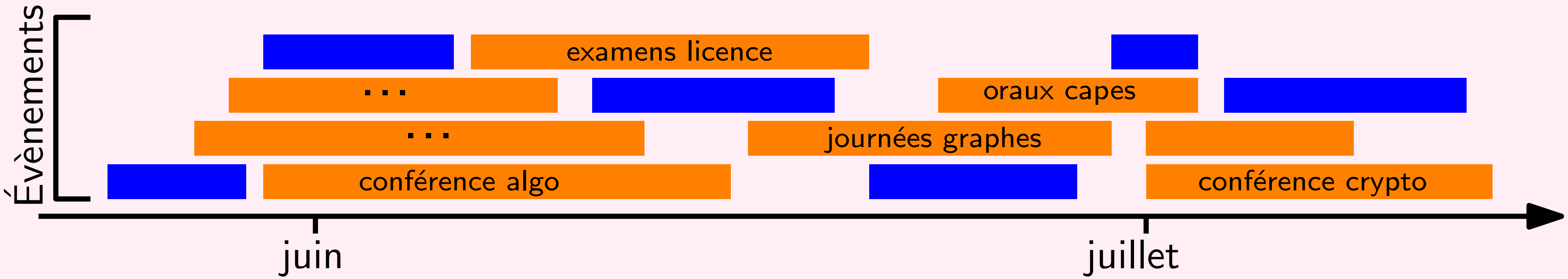


- Problème fondamental
- Nombreuses applications

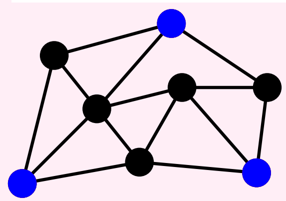
→ Complexité ? Approximabilité ?

NP-complet

Résultat



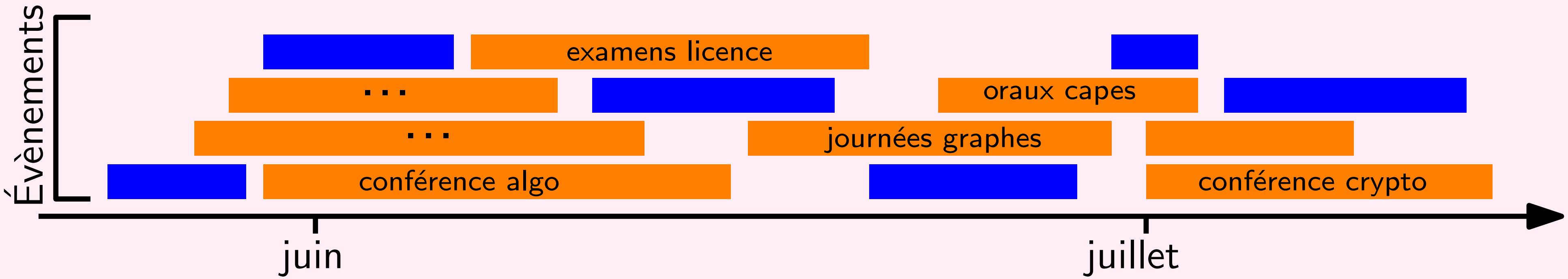
Problème du **stable maximum** sur un **graphe d'intervalles**



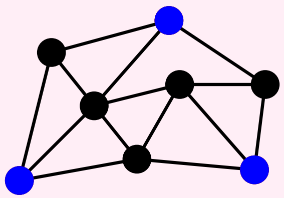
- Problème fondamental
- Nombreuses applications

→ Complexité ? NP-complet Approximabilité ? "impossible" !!

Résultat



Problème du **stable maximum** sur un **graphe d'intervalles**

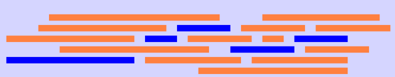


- Problème fondamental
- Nombreuses applications

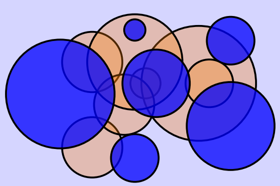
→ Complexité ? NP-complet — Approximabilité ? “impossible” !!

Objets **géométriques** disjoints

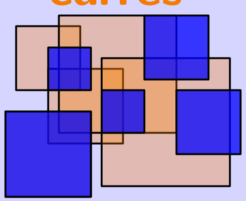
intervalles



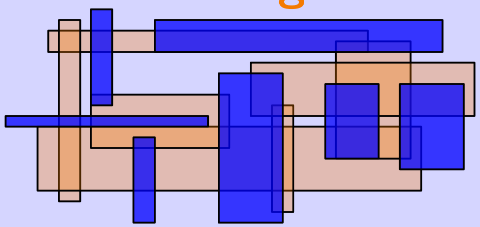
disques



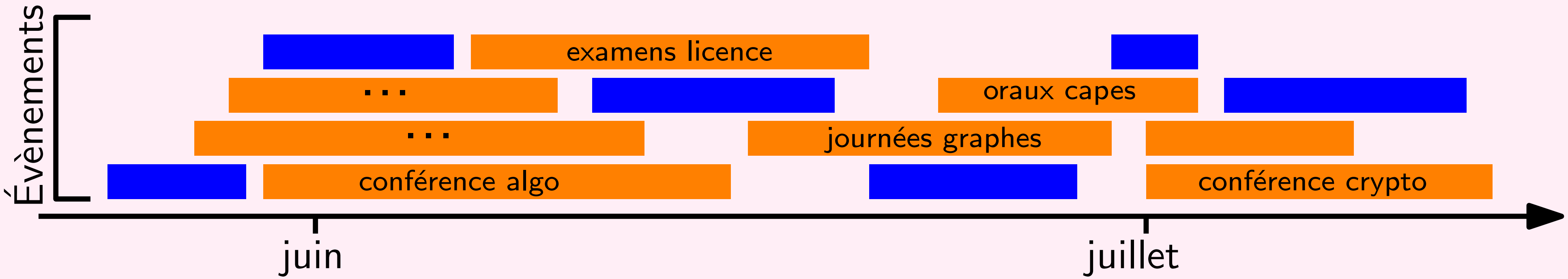
carrés



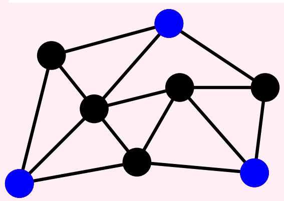
rectangles



Résultat



Problème du **stable maximum** sur un **graphe d'intervalles**

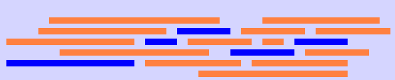


- Problème fondamental
- Nombreuses applications

→ Complexité ? NP-complet — Approximabilité ? "impossible" !!

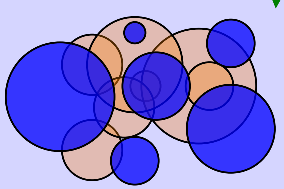
Objets **géométriques** disjoints

intervalles ✓



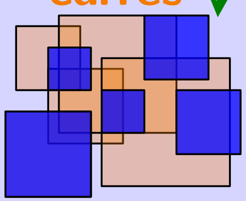
Polynomial

disques ✓



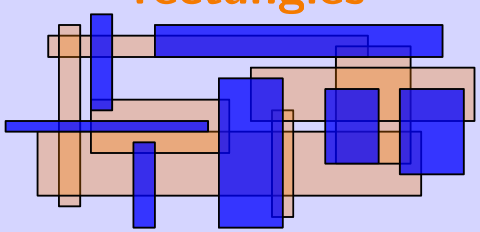
NP-complet
$(1+\epsilon)$-approx

carrés ✓



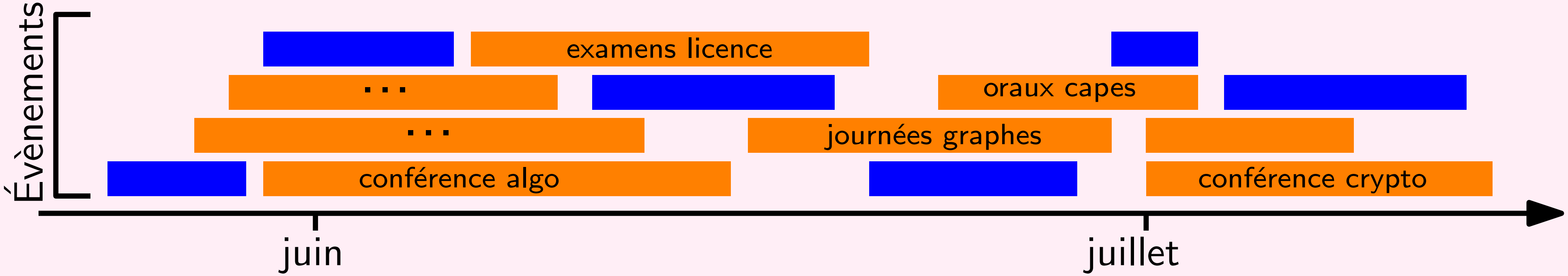
NP-complet
$(1+\epsilon)$-approx

rectangles

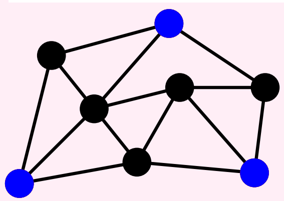


NP-complet

Résultat



Problème du **stable maximum** sur un **graphe d'intervalles**

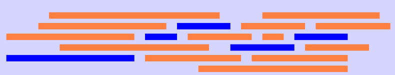


- Problème fondamental
- Nombreuses applications

→ Complexité ? NP-complet — Approximabilité ? "impossible" !!

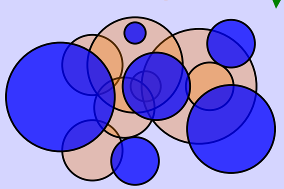
Objets **géométriques** disjoints

intervalles ✓



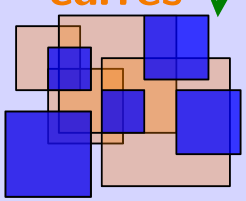
Polynomial

disques ✓



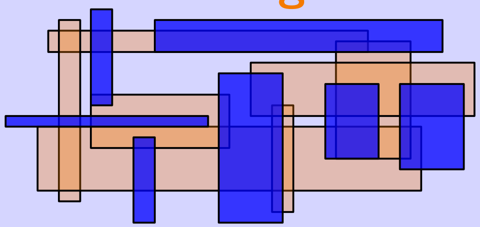
NP-complet
$(1+\epsilon)$-approx

carrés ✓



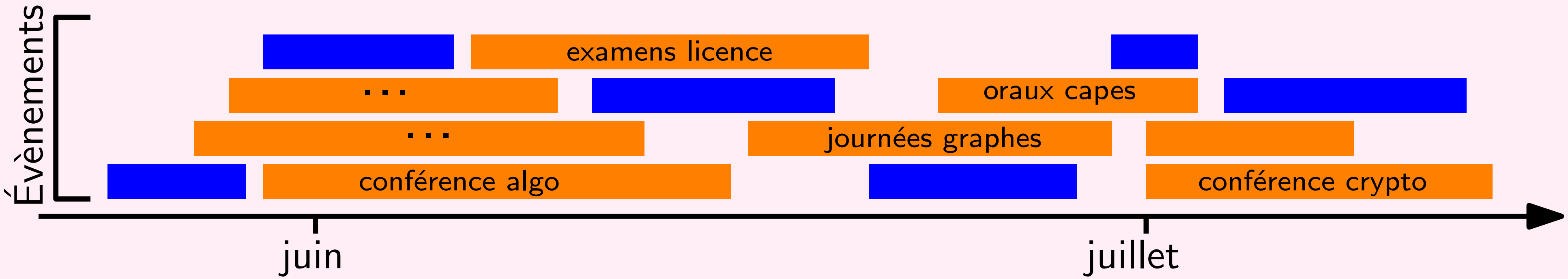
NP-complet
$(1+\epsilon)$-approx

rectangles

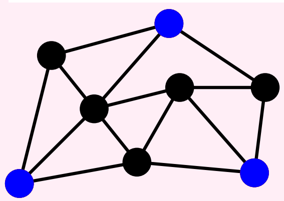


NP-complet
Approx constante ???

Résultat



Problème du **stable maximum** sur un **graphe d'intervalles**

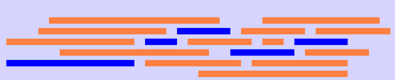


- Problème fondamental
- Nombreuses applications

→ Complexité ? NP-complet — Approximabilité ? "impossible" !!

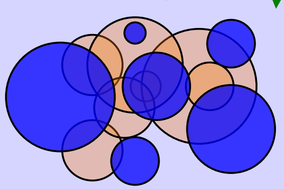
Objets **géométriques** disjoints

intervalles ✓



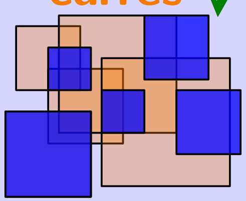
Polynomial

disques ✓



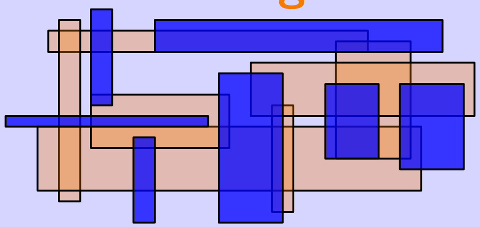
NP-complet
$(1+\epsilon)$-approx

carrés ✓



NP-complet
$(1+\epsilon)$-approx

rectangles



NP-complet
Approx constante ???

Meilleure approx pour les **rectangles** :

$O(\log n)$ — [KMP, **SODA'98**]

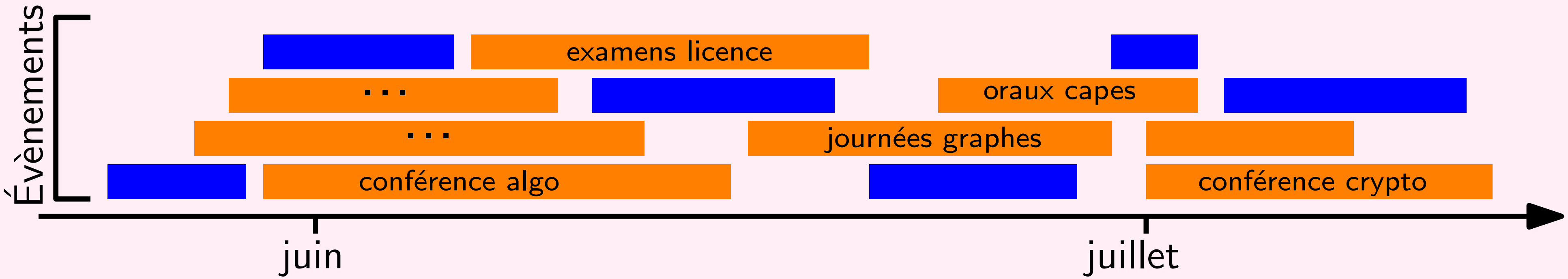
$O(\log \log n)$ — [CC, **SODA'09**]

10 — [M, **FOCS'21**]

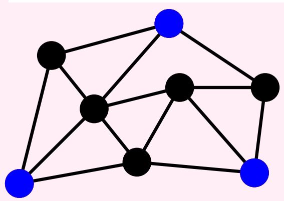
→ temps

(n : nombre de rectangles en entrée)

Résultat



Problème du **stable maximum** sur un **graphe d'intervalles**

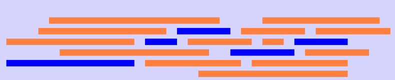


- Problème fondamental
- Nombreuses applications

→ Complexité ? NP-complet — Approximabilité ? "impossible" !!

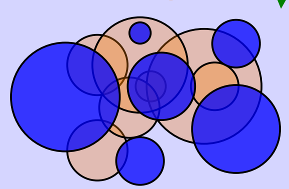
Objets **géométriques** disjoints

intervalles ✓



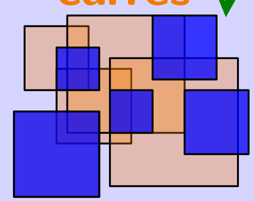
Polynomial

disques ✓



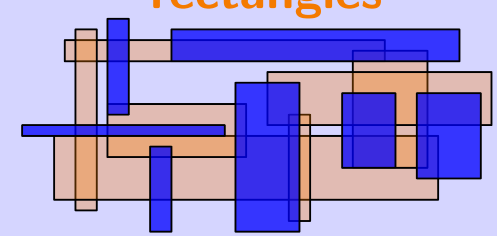
NP-complet
$(1+\epsilon)$-approx

carrés ✓



NP-complet
$(1+\epsilon)$-approx

rectangles



NP-complet
Approx constante ???

Meilleure approx pour les **rectangles** :

$O(\log n)$ — [KMP, **SODA'98**]

$O(\log \log n)$ — [CC, **SODA'09**]

10 — [M, **FOCS'21**]

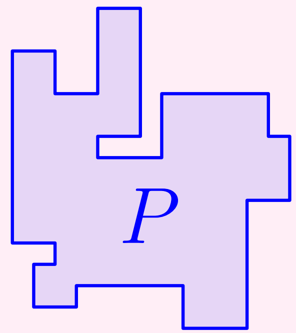
$\mathbf{2+\epsilon}$ — [GK**M**MRW, **SODA '22**]

< 2 ?

temps

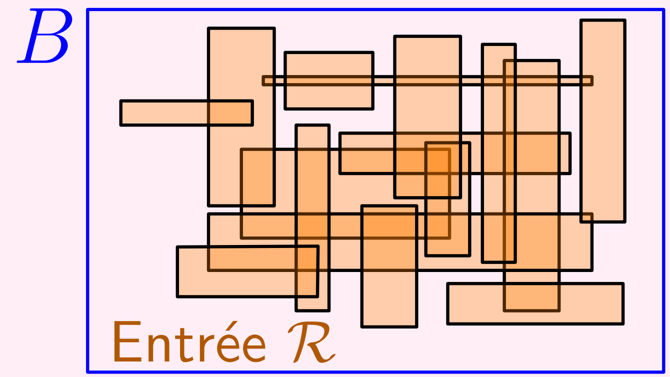
(n : nombre de rectangles en entrée)

L'algorithme. $\longrightarrow$ Programmation dynamique

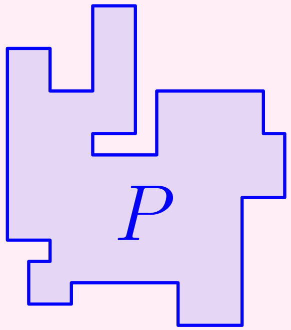
clés $=$ $\Big\{$ 

$\Big|$ P a au plus $1/\epsilon$ côtés

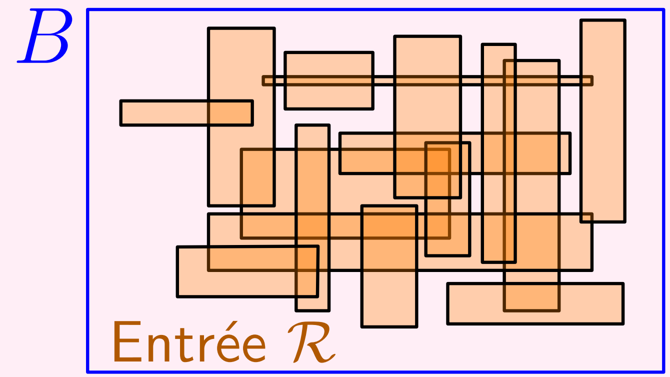
$P \subseteq B$ $\Big\}$

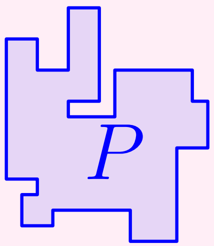


L'algorithme. → Programmation dynamique

clés = { 

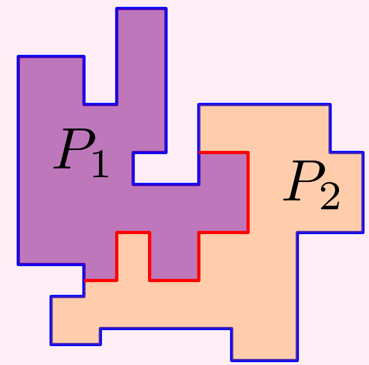
| P a au plus $1/\epsilon$ côtés, $P \subseteq B$ }



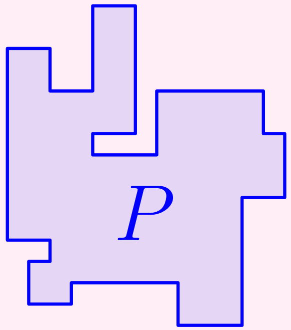
DP(

) =

- **0** si P ne contient aucun rectangle,
- **1** si P contient un seul rectangle $R \in \mathcal{R}$,
- $\max_{P_1, P_2} \{DP(P_1) + DP(P_2)\}$, sinon. →

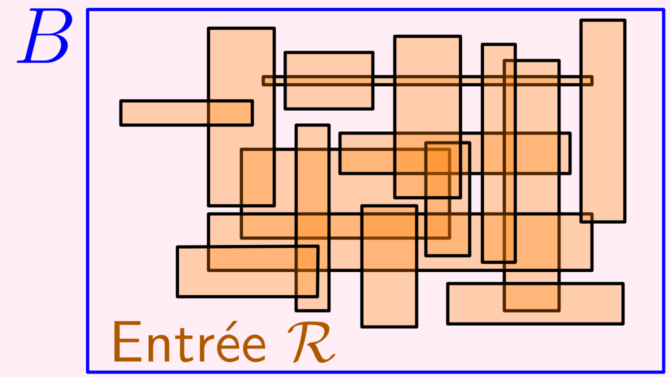


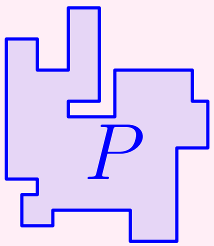
L'algorithme. → Programmation dynamique

clés = $\Bigg\{$ 

$\Bigg|$ P a au plus $1/\epsilon$ côtés

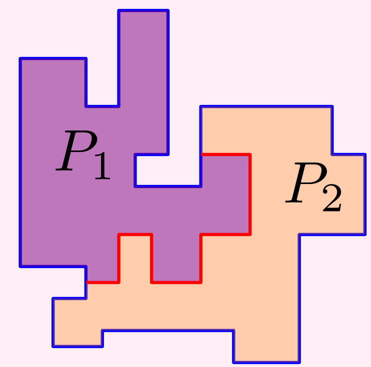
$P \subseteq B$ $\Bigg\}$



DP(

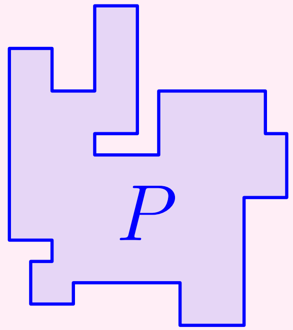
) =

- **0** si P ne contient aucun rectangle,
- **1** si P contient un seul rectangle $R \in \mathcal{R}$,
- $\max_{P_1, P_2}\{DP(P_1) + DP(P_2)\}$, sinon. →

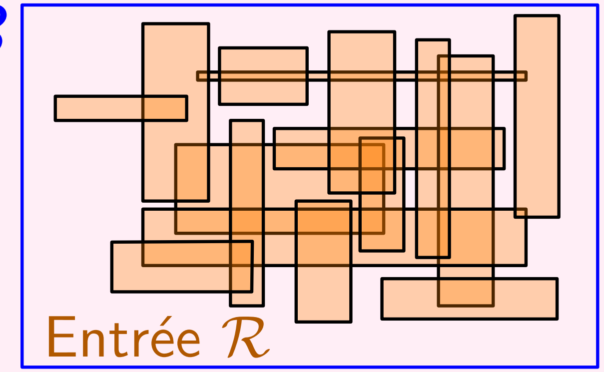


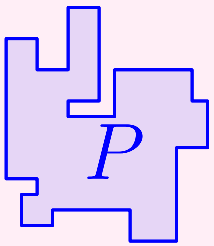
sortie : $DP(B)$

L'algorithme. $\longrightarrow$ Programmation dynamique

clés $=$ $\Bigg\{$ 

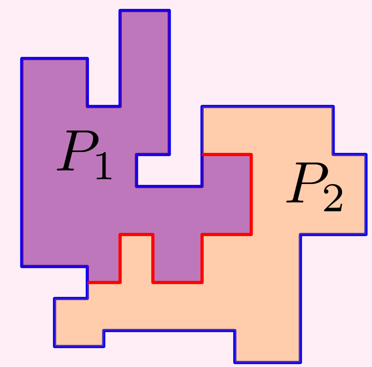
$\Bigg|$ P a au plus $1/\epsilon$ côtés, $P \subseteq B$ $\Bigg\}$

B 

DP(

) $=$

- **0** si P ne contient aucun rectangle,
- **1** si P contient un seul rectangle $R \in \mathcal{R}$,
- $\max_{P_1, P_2} \{DP(P_1) + DP(P_2)\}$, sinon. $\longrightarrow$



sortie : $DP(B)$

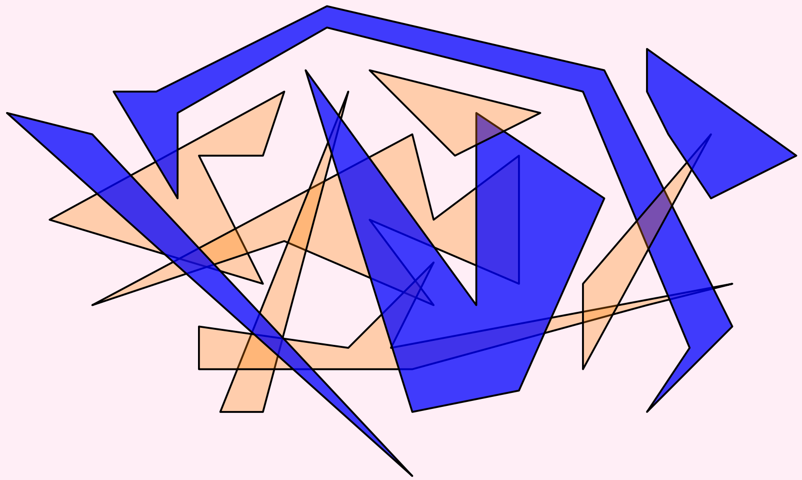
Analyse. But: Prouver que $DP(B) \geq OPT/(2+\epsilon)$

Lemme structurel. Pour tout ensemble indépendant $\mathcal{I} \subseteq \mathcal{R}$, il existe une façon de “découper récursivement” B qui “préserve” une fraction au moins $\frac{1}{2+\epsilon}$ de $\mathcal{I}$.

preuve (**25 pages**) :
- schéma de chargement,
- observations géométriques,
- énumération de cas, etc.

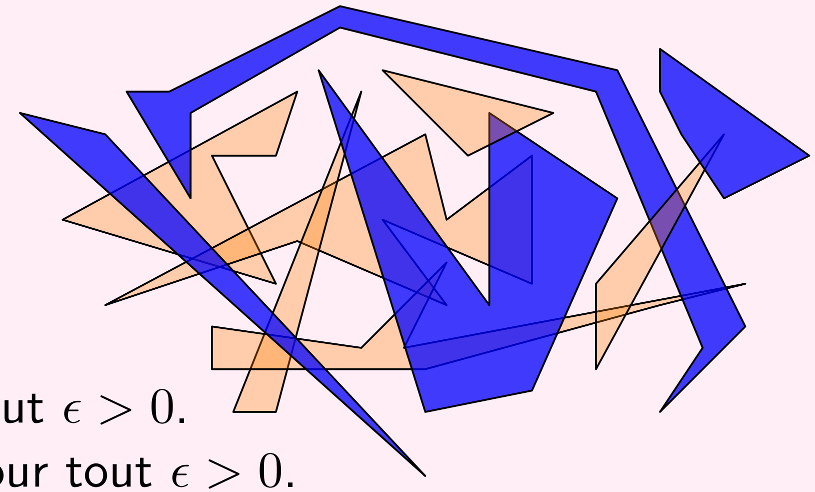
Axe 1 : Objets géométriques disjoints

Objectif. Comprendre l'approximabilité du problème du stable maximum dans les graphes d'intersection de **polygones**.



Axe 1 : Objets géométriques disjoints

Objectif. Comprendre l'approximabilité du problème du stable maximum dans les graphes d'intersection de **polygones**.

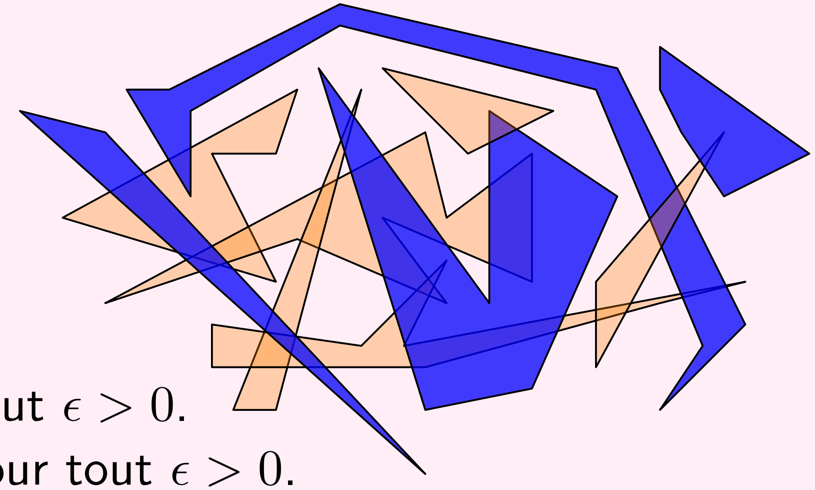


État de l'art :

- Approx en temps polynomial : $O(n^{\epsilon})$, pour tout $\epsilon > 0$.
- Approx en temps $\underbrace{\text{\textit{quasi}-polynomial}}_{2^{poly \log n}}$: $1 + \epsilon$, pour tout $\epsilon > 0$.

Axe 1 : Objets géométriques disjoints

Objectif. Comprendre l'approximabilité du problème du stable maximum dans les graphes d'intersection de **polygones**.



État de l'art :

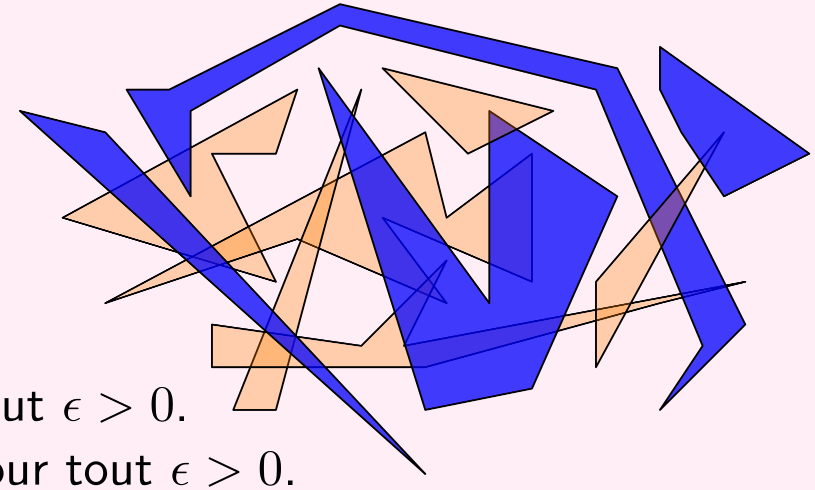
- Approx en temps polynomial : $O(n^\epsilon)$, pour tout $\epsilon > 0$.

- Approx en temps $\underbrace{\text{\textit{quasi}-polynomial}}_{2^{poly \log n}}$: $1+\epsilon$, pour tout $\epsilon > 0$.

Il "doit" aussi exister une $(1+\epsilon)$-approx en temps **polynomial**.

Axe 1 : Objets géométriques disjoints

Objectif. Comprendre l'approximabilité du problème du stable maximum dans les graphes d'intersection de **polygones**.



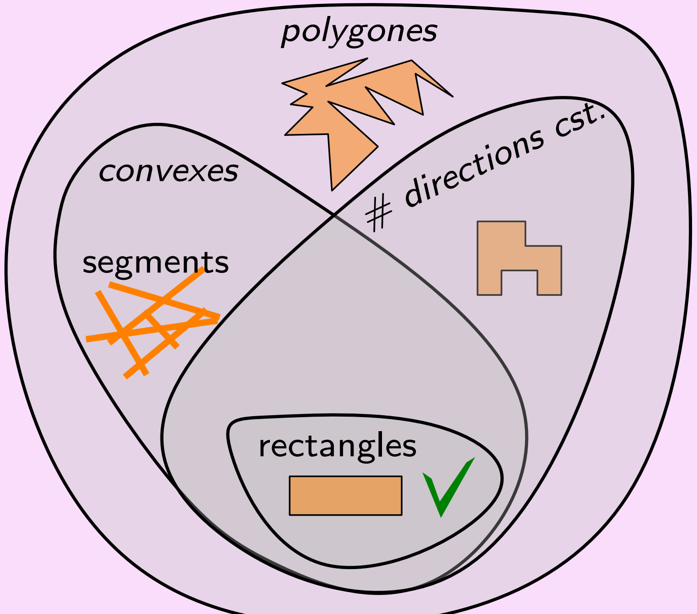
État de l'art :

- Approx en temps polynomial : $O(n^\epsilon)$, pour tout $\epsilon > 0$.
- Approx en temps $\underbrace{\textit{quasi}\text{-polynomial}}_{2^{poly \log n}}$: $1+\epsilon$, pour tout $\epsilon > 0$.

Il "doit" aussi exister une $(1+\epsilon)$-approx en temps **polynomial**.

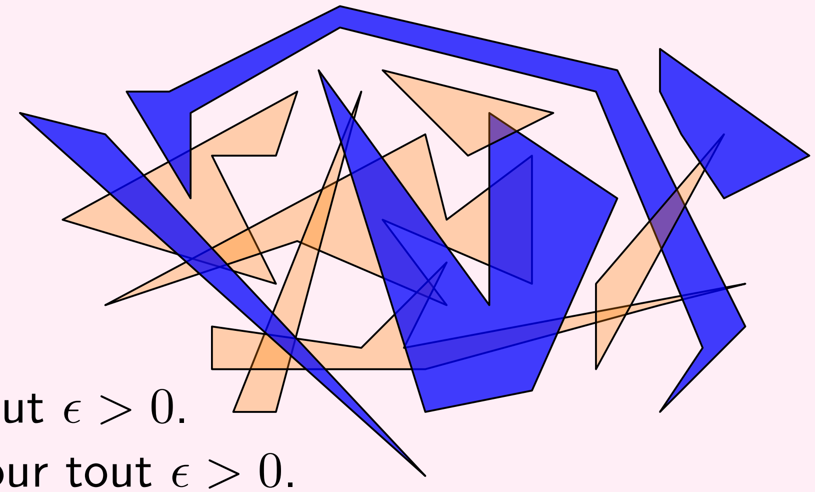
Directions.

- Classes particulières :



Axe 1 : Objets géométriques disjoints

Objectif. Comprendre l'approximabilité du problème du stable maximum dans les graphes d'intersection de **polygones**.



État de l'art :

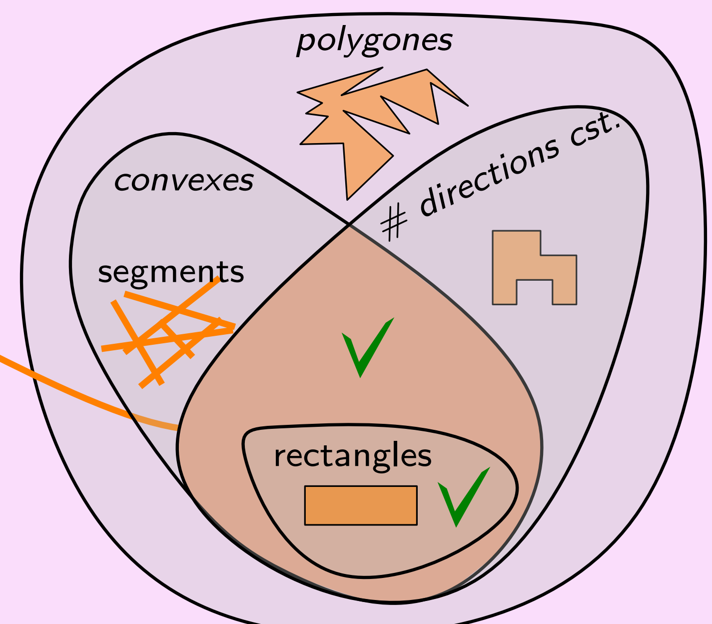
- Approx en temps polynomial : $O(n^\epsilon)$, pour tout $\epsilon > 0$.
- Approx en temps $\underbrace{\textit{quasi}\text{-polynomial}}_{2^{poly \log n}}$: $1+\epsilon$, pour tout $\epsilon > 0$.

Il "doit" aussi exister une $(1+\epsilon)$-approx en temps **polynomial**.

Directions.

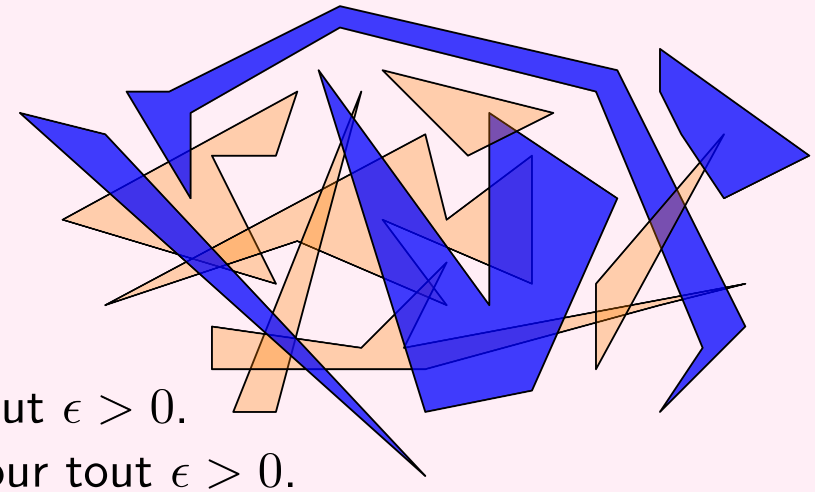
- Classes particulières :

convexe et d directions $\Rightarrow O(d)$-approx
[GMUT, soumis ESA]



Axe 1 : Objets géométriques disjoints

Objectif. Comprendre l'approximabilité du problème du stable maximum dans les graphes d'intersection de **polygones**.



État de l'art :

- Approx en temps polynomial : $O(n^{\epsilon})$, pour tout $\epsilon > 0$.
- Approx en temps $\underbrace{\textit{quasi}\text{-polynomial}}_{2^{poly \log n}}$: $1 + \epsilon$, pour tout $\epsilon > 0$.

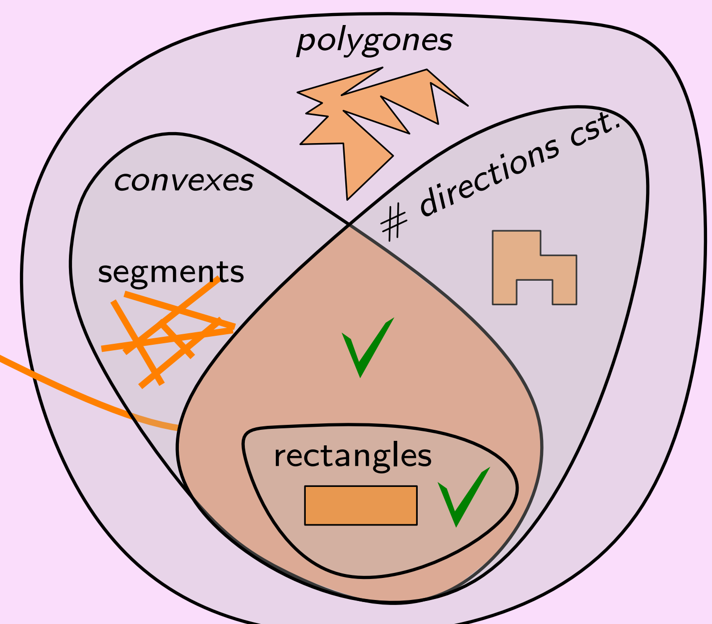
Il “doit” aussi exister une $(1 + \epsilon)$-approx en temps **polynomial**.

Directions.

- Classes particulières :

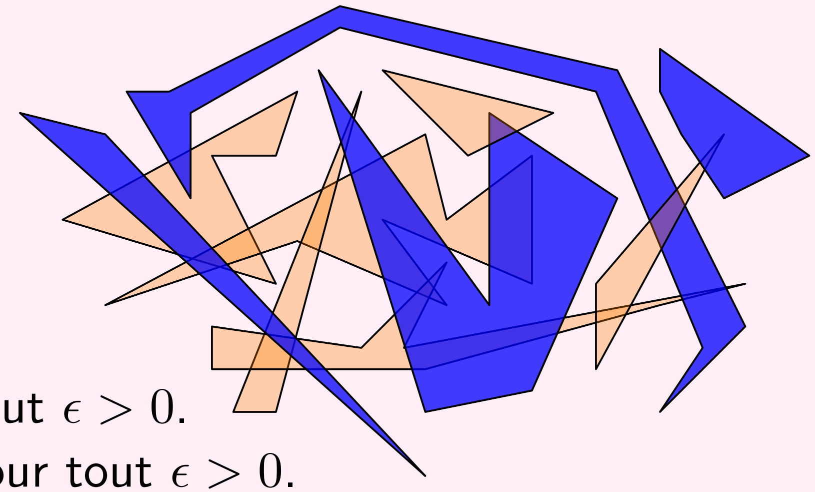
  convexe et d directions $\Rightarrow O(d)$-approx
  [GMUT, soumis ESA]

- Cas pondéré et questions combinatoires



Axe 1 : Objets géométriques disjoints

Objectif. Comprendre l'approximabilité du problème du stable maximum dans les graphes d'intersection de **polygones**.



État de l'art :

- Approx en temps polynomial : $O(n^{\epsilon})$, pour tout $\epsilon > 0$.
- Approx en temps *quasi*-polynomial : $1 + \epsilon$, pour tout $\epsilon > 0$.

$2^{poly \log n}$

Il "doit" aussi exister une $(1 + \epsilon)$-approx en temps **polynomial**.

Directions.

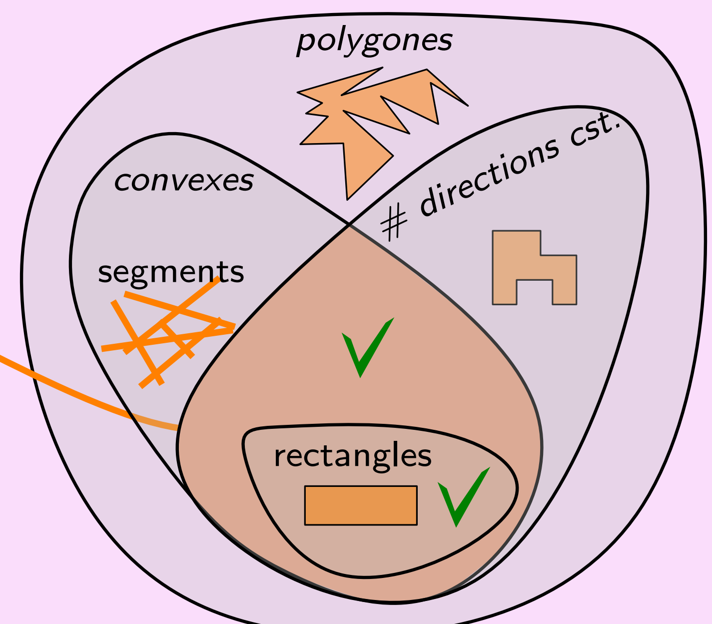
- Classes particulières :

  convexe et d directions $\Rightarrow O(d)$-approx
  [GMUT, soumis ESA]

- Cas pondéré et questions combinatoires

- Problèmes connexes : *Minimum hitting set*, version *packing*

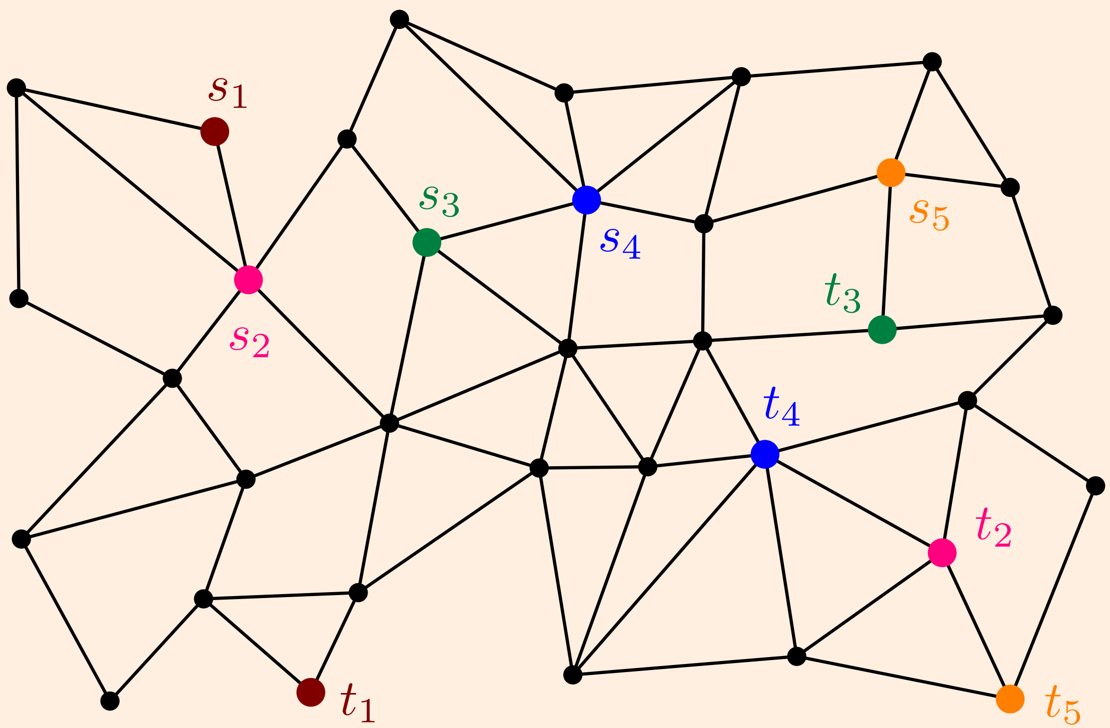
  Approx FPT [MPP, en cours]



Axe 2 : Chemins disjoints

Objectif. Mieux comprendre la complexité et l'approximabilité de deux versions optimisation du problème des chemins disjoints, dans les graphes planaires et de genre borné.

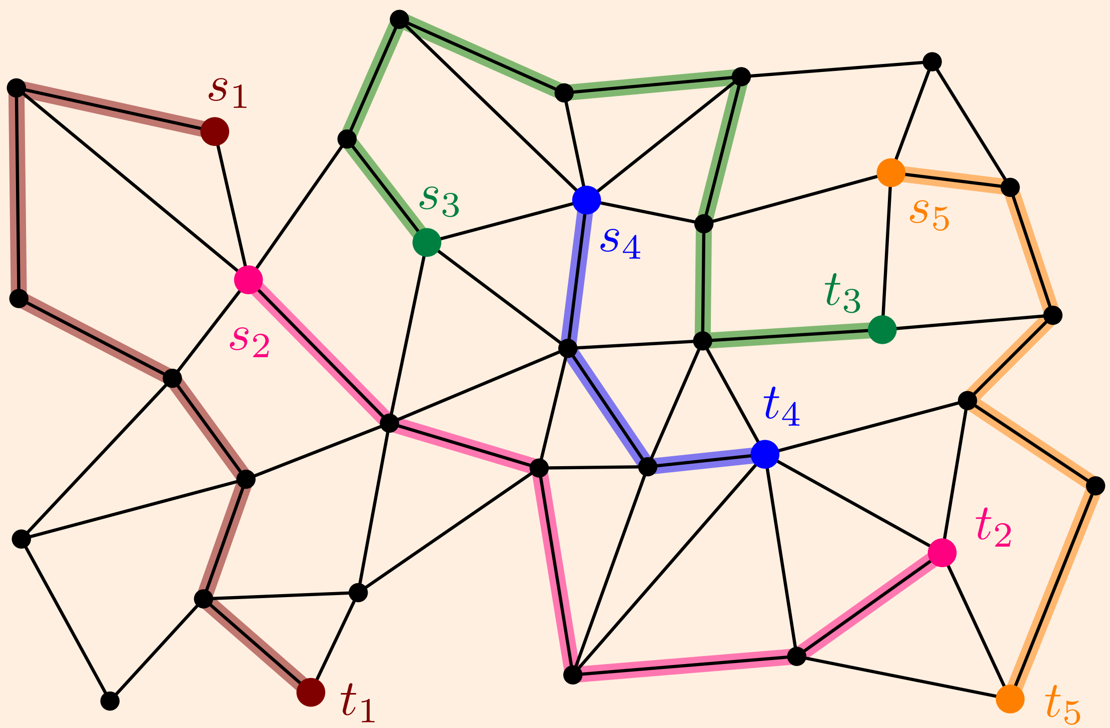
Entrée : Un graphe G, et un ensemble de paires $H = \{(s_1, t_1), (s_2, t_2), \ldots\}$



Axe 2 : Chemins disjoints

Objectif. Mieux comprendre la complexité et l'approximabilité de deux versions optimisation du problème des chemins disjoints, dans les graphes planaires et de genre borné.

Entrée : Un graphe G, et un ensemble de paires $H = \{(s_1, t_1), (s_2, t_2), \ldots\}$
Sortie : Un ensemble de chemins disjoints connectant chacune des paires.



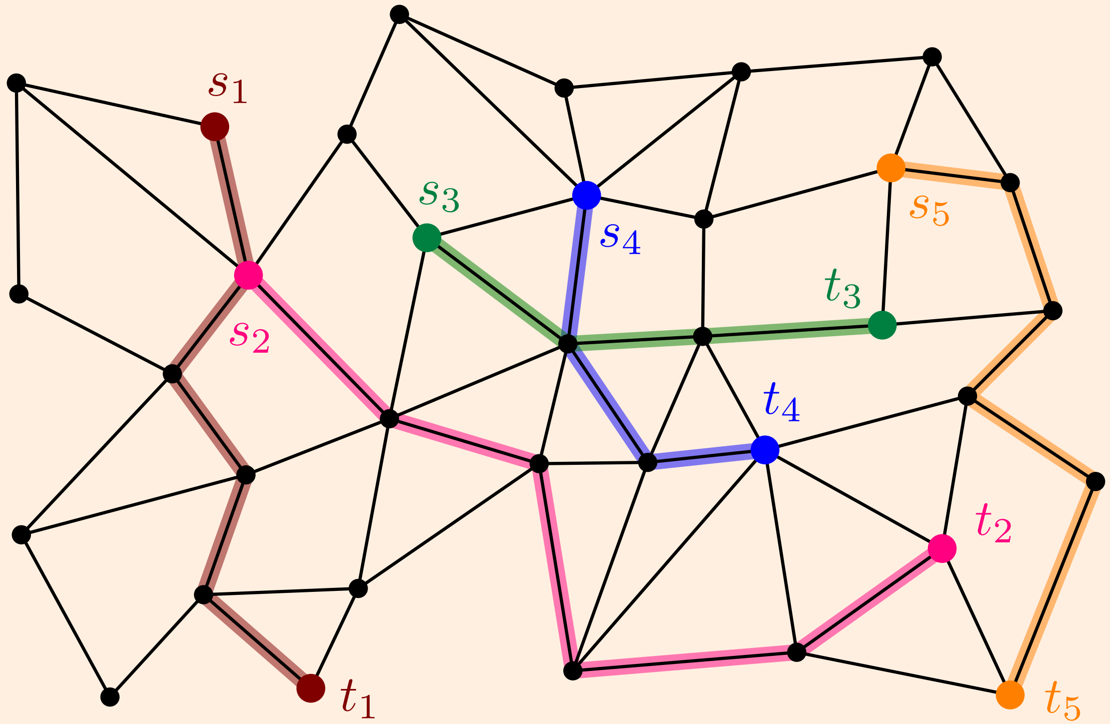
Axe 2 : Chemins disjoints

Objectif. Mieux comprendre la complexité et l'approximabilité de deux versions optimisation du problème des chemins disjoints, dans les graphes planaires et de genre borné.

(Version *arêtes*-disjointes)

Entrée : Un graphe G, et un ensemble de paires $H = \{(s_1, t_1), (s_2, t_2), \ldots\}$

Sortie : Un ensemble de chemins **arêtes-disjointes** connectant chacune des paires.



Axe 2 : Chemins disjoints

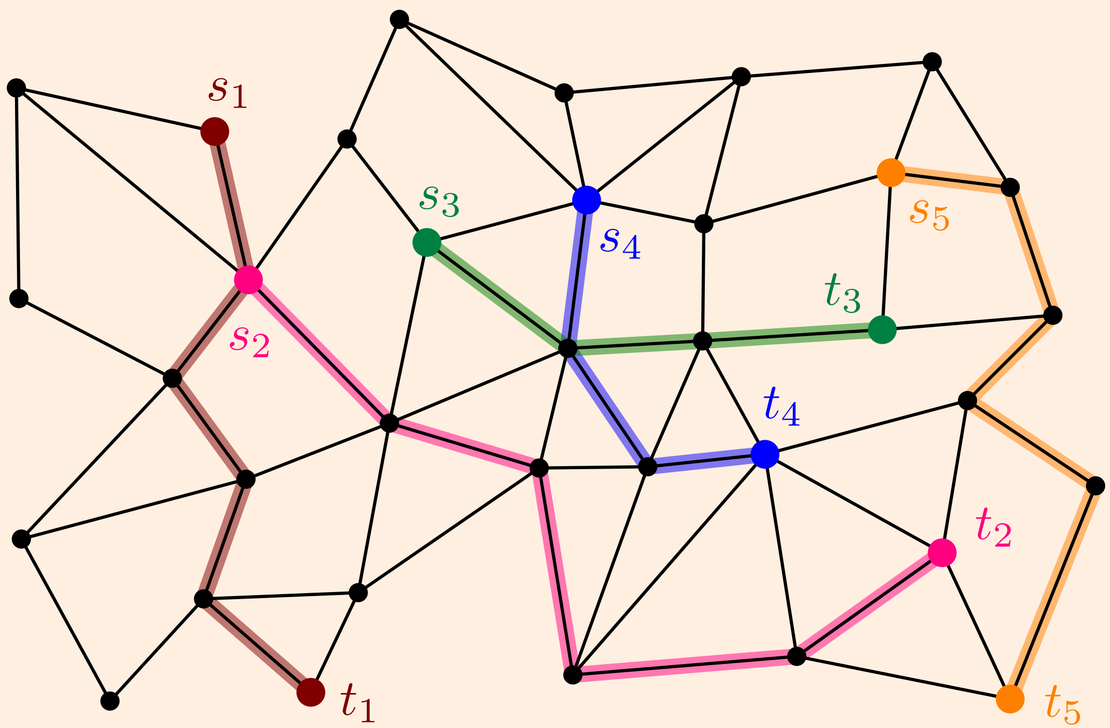
Objectif. Mieux comprendre la complexité et l'approximabilité de deux versions optimisation du problème des chemins disjoints, dans les graphes planaires et de genre borné.

Maximiser le nombre de paires connectées

(Version *arêtes*-disjointes)

Entrée : Un graphe G, et un ensemble de paires $H = \{(s_1, t_1), (s_2, t_2), \ldots\}$

Sortie : Un ensemble de chemins **arêtes-disjointes** ~~connectant chacune des paires~~.
un nombre maximum de paires.



Axe 2 : Chemins disjoints

Objectif. Mieux comprendre la complexité et l'approximabilité de deux versions optimisation du problème des chemins disjoints, dans les graphes planaires et de genre borné.

Maximiser le nombre de paires connectées

(Version *arêtes*-disjointes)

Entrée : Un graphe G, et un ensemble de paires $H = \{(s_1, t_1), (s_2, t_2), \ldots\}$

Sortie : Un ensemble de chemins **arêtes-disjointes** ~~connectant chacune des paires~~.
un nombre maximum de paires.

Tous les problèmes "naturels" sur les graphes planaires admettent de "bonnes" approximations ?

Axe 2 : Chemins disjoints

Objectif. Mieux comprendre la complexité et l'approximabilité de deux versions optimisation du problème des chemins disjoints, dans les graphes planaires et de genre borné.

Maximiser le nombre de paires connectées

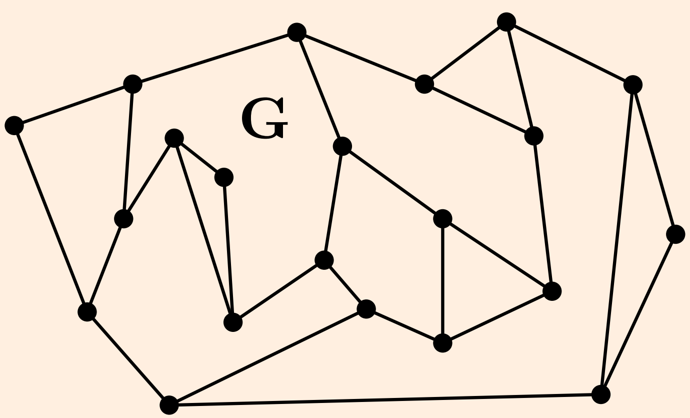
(Version *arêtes*-disjointes)

Entrée : Un graphe G, et un ensemble de paires $H = \{(s_1, t_1), (s_2, t_2), \ldots\}$

Sortie : Un ensemble de chemins **arêtes-disjointes** ~~connectant chacune des paires~~.
un nombre maximum de paires.

Tous les problèmes "naturels" sur les graphes planaires admettent de "bonnes" approximations ?

Non [CKN'17] : Sous certaines hypothèses, ce problème d'admet **pas d'approximation constante, même quand $\mathbf{G}$ est planaire** et sous-cubique.



Axe 2 : Chemins disjoints

Objectif. Mieux comprendre la complexité et l'approximabilité de deux versions optimisation du problème des chemins disjoints, dans les graphes planaires et de genre borné.

Maximiser le nombre de paires connectées

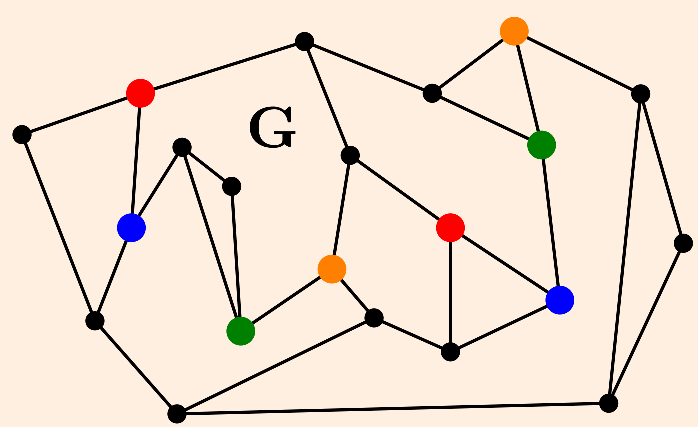
(Version *arêtes*-disjointes)

Entrée : Un graphe G, et un ensemble de paires $H = \{(s_1, t_1), (s_2, t_2), \ldots\}$

Sortie : Un ensemble de chemins **arêtes-disjointes** ~~connectant chacune des paires~~.
un nombre maximum de paires.

Tous les problèmes "naturels" sur les graphes planaires admettent de "bonnes" approximations ?

Non [CKN'17] : Sous certaines hypothèses, ce problème d'admet **pas d'approximation constante, même quand $\mathbf{G}$ est planaire** et sous-cubique.



Axe 2 : Chemins disjoints

Objectif. Mieux comprendre la complexité et l'approximabilité de deux versions optimisation du problème des chemins disjoints, dans les graphes planaires et de genre borné.

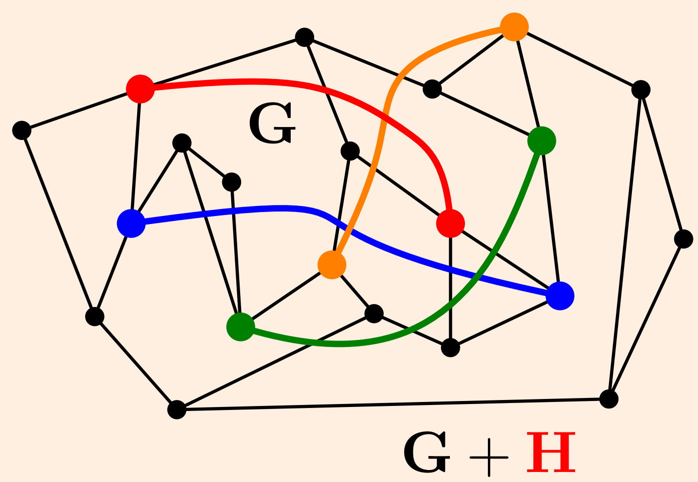
Maximiser le nombre de paires connectées

(Version *arêtes*-disjointes)

Entrée : Un graphe G, et un ensemble de paires $H = \{(s_1, t_1), (s_2, t_2), \ldots\}$
Sortie : Un ensemble de chemins **arêtes-disjointes** ~~connectant chacune des paires~~.
un nombre maximum de paires.

Tous les problèmes "naturels" sur les graphes planaires admettent de "bonnes" approximations ?

Non [CKN'17] : Sous certaines hypothèses, ce problème d'admet **pas d'approximation constante, même quand $\mathbf{G}$ est planaire** et sous-cubique.



Axe 2 : Chemins disjoints

Objectif. Mieux comprendre la complexité et l'approximabilité de deux versions optimisation du problème des chemins disjoints, dans les graphes planaires et de genre borné.

Maximiser le nombre de paires connectées

(Version *arêtes*-disjointes)

Entrée : Un graphe G, et un ensemble de paires $H = \{(s_1, t_1), (s_2, t_2), \ldots\}$

Sortie : Un ensemble de chemins **arêtes-disjointes** ~~connectant chacune des paires~~.
un nombre maximum de paires.

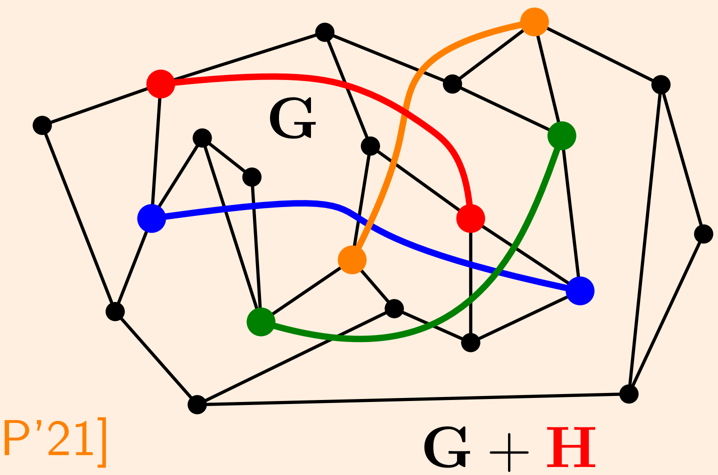
Tous les problèmes "naturels" sur les graphes planaires admettent de "bonnes" approximations ?

Non [CKN'17] : Sous certaines hypothèses, ce problème d'admet **pas d'approximation constante, même quand $\mathbf{G}$ est planaire** et sous-cubique.

Objectif: Bonnes approximations quand $\mathbf{G} + \mathbf{H}$ est plongé dans une surface.

motivation : définir une vraie notion de planarité.

outils : programme linéaire, topologie, coloration, ...



✓ planaire : $O(1)$-approx [HMMSV, SIAM JDM 21]

✓ orientable, genre g : $O(g^2)$-approx [HMMV, ICALP'21]

Axe 2 : Chemins disjoints

Objectif. Mieux comprendre la complexité et l'approximabilité de deux versions optimisation du problème des chemins disjoints, dans les graphes planaires et de genre borné.

Maximiser le nombre de paires connectées

Minimiser la longueur totale

Entrée : Un graphe G, et un ensemble de paires $H = \{(s_1, t_1), (s_2, t_2), \ldots\}$
Sortie : Un ensemble de chemins disjoints connectant chacune des paires, minimisant la somme des longueurs des chemins.

complexité paramétrée ?

Objectif: Un algorithme exact en temps $n^{f(|H|)}$ quand G est planaire.

motivation : problème fondamental, complexité mal comprise.

outils : décompositions de graphes, sommets non-pertinents, flots, ...

→ Grille ✓ [**M**MS, en cours]

Objectif: Bonnes approximations quand $\mathbf{G} + \mathbf{H}$ est plongé dans une surface.

motivation : définir une vraie notion de planarité.

outils : programme linéaire, topologie, coloration, ...

✓ planaire : $O(1)$-approx [H**M**MSV, SIAM JDM 21]

✓ orientable, genre g : $O(g^2)$-approx [H**M**MV, ICALP'21]

Intégration : Equipe AlGCo

Algorithmes, Graphes et Combinatoire

mots-clés

théorie des graphes, graphes planaires et plongés sur des surfaces, graphes d'intersection d'objets géométriques, complexité paramétrée, ...

Collaborations.

Ce que je peux apporter à l'équipe / au laboratoire.

- Renforcer la composante *algorithmes d'approximation*
- nouvelle thématique : algorithmes en ligne
- collaborations internationales : Varsovie, Lugano, Chili, ...

Ce que l'équipe peut m'apporter.

- un environnement idéal
- nouvelles directions : structure des graphes, complexité paramétrée, ...

Résumé

Recherche

Diversité thématique

Algos d'approx
Géométrie
Complexité paramétrée
Algorithmes en ligne

Publications et communication

6 conférences internationales (+3 soumis)
2 journaux (+3 revisions)
4 projets en cours
19 exposés et séminaires

Collaborations

Paris, Varsovie, Lugano, Chili,
Bönn, New-York, . . .

Enseignement

Parcours

Agrégation
M2 Recherche Opérationnelle
Master MEEF

Experience (234h TD/TP)

Algorithmique / graphes **74h TD**
Python / calcul math. **64h TP**
Cryptographie **32h TD**

Encadrement

Trois stages de 6 mois
Un stage de 2 mois