Cours de Master Recherche Spécialité CODE : Résolution de problèmes combinatoires

Christine Solnon

LIRIS, UMR 5205 CNRS / Université Lyon 1

2007

Plan du cours

- 1 - Introduction
  - Qu'est-ce qu'un problème complexe ?
  - Exemples de problèmes complexes
- 2 - Approches complètes
  - Structuration de l'espace de recherche en Arbre
    - Application à la planification et la résolution de CSPs
  - Structuration de l'espace de recherche en Treillis
    - 6/11/2007 et 8/11/2007 : **Fabien De Marchi**
- 3 - Approches incomplètes
  - Algorithmes gloutons
  - Recherche locale
  - Algorithmes génétiques
    - 30/10/2007 : **Guillaume Beslon**
- 4 - Optimisation par colonies de fourmis
  - Principes généraux de la méta-heuristique ACO
  - Applications
  - Diversification vs intensification de la recherche

Plan de la première partie : introduction

- Notions (Rappels ?) de complexité(s)
  - Complexité systémique
  - Complexité d'un algorithme
  - Complexité d'un problème
- Quelques exemples de problèmes « complexes »
  - SAT = Satisfiabilité de formules booléennes
  - Problèmes de satisfaction de contraintes
  - Problèmes de planification
  - Problèmes dans les graphes
  - Recherche d'ensembles fréquents
- Notion de transition de phases

  ...ou comment prédire la difficulté d'un problème...

Plan de la première partie : introduction

Complexité systémique

Qu'est-ce qu'un système complexe ?

- Ensemble de composants en interaction
- Exemples : cerveau, colonie de fourmi, société, ...

Caractéristiques des systèmes complexes

- Auto-organisation des composants / stimuli extérieurs
- Comportement observé dépendant de l'échelle
  $\leadsto$ le « tout » ne se réduit pas à la somme de ses composants
  $\leadsto$ émergence de comportements « intelligents »
- Comportement global non prévisible de façon analytique
  $\leadsto$ dynamique des interactions non linéaires (rétroaction)
  $\leadsto$ comportement « chaotique » (effet papillon, krach boursier)

Plus d'infos sur **http://www.calresco.org**

Complexité d'un algorithme

Qu'est-ce que la complexité d'un algorithme ?

- Estimation des ressources nécessaires à l'exécution
  - Complexité en temps = estimation du nombre d'instructions
  - Complexité en espace = estimation de l'espace mémoire

  $\rightsquigarrow$ comparer 2 algorithmes / la taille n des données en entrées
- Ordre de grandeur $\mathcal{O}$
  $\mathcal{O}(f(n)) \rightsquigarrow \exists c, n_0$ tq $\forall n > n_0$, nb instructions $< c.f(n)$
  - $\mathcal{O}(log(n))$: logarithmique
  - $\mathcal{O}(n)$: linéaire
  - $\mathcal{O}(n^k)$: polynomial
  - $\mathcal{O}(k^n)$: exponentiel

Pour en savoir plus

Introduction à l'algorithmique de Cormen, Leiserson et Rivest
http://www2.toki.or.id/book/AlgDesignManual/BOOK/BOOK3/NODE126.HTM

Complexité de quelques algorithmes (1)

Rechercher un élément dans un tableau de n éléments

- Algorithme de recherche séquentielle : $\mathcal{O}(n)$
- Algorithme de recherche dichotomique / tableau trié : $\mathcal{O}(log(n))$

Trier un tableau de n éléments

- Algorithme de tri par sélection : $\mathcal{O}(n^2)$
- Algorithme de tri par insertion :
  $\rightsquigarrow$ dans le meilleur des cas (tableau déjà trié) : $\mathcal{O}(n)$
  $\rightsquigarrow$ dans le pire des cas (tableau trié à l'envers) : $\mathcal{O}(n^2)$
- Algorithme de tri rapide (quicksort) :
  $\rightsquigarrow$ dans le meilleur des cas (pivot = médiane) : $\mathcal{O}(n.log(n))$
  $\rightsquigarrow$ dans le pire des cas (pivot = elt max ou min) : $\mathcal{O}(n^2)$
- Algorithme de tri par tas : $\mathcal{O}(n.log(n))$

Complexité de quelques algorithmes (1)

Rechercher un élément dans un tableau de n éléments

- Algorithme de recherche séquentielle : $\mathcal{O}(n)$
- Algorithme de recherche dichotomique / tableau trié : $\mathcal{O}(log(n))$

Trier un tableau de n éléments

- Algorithme de tri par sélection : $\mathcal{O}(n^2)$
- Algorithme de tri par insertion :
  $\leadsto$ dans le meilleur des cas (tableau déjà trié) : $\mathcal{O}(n)$
  $\leadsto$ dans le pire des cas (tableau trié à l'envers) : $\mathcal{O}(n^2)$
- Algorithme de tri rapide (quicksort) :
  $\leadsto$ dans le meilleur des cas (pivot = médiane) : $\mathcal{O}(n.log(n))$
  $\leadsto$ dans le pire des cas (pivot = elt max ou min) : $\mathcal{O}(n^2)$
- Algorithme de tri par tas : $\mathcal{O}(n.log(n))$

Complexité de quelques algorithmes (2)

Maximiser une fonction linéaire sous des contraintes linéaires

- Algorithme du simplexe : $\mathcal{O}(2^n)$ dans le pire des cas... faiblement polynomial en pratique !
- Algorithme du point intérieur : polynomial

Chercher un circuit hamiltonien dans un graphe

- Algorithme énumérant toutes les permutations des sommets : $\mathcal{O}(n!)$

Complexité de quelques algorithmes (2)

Maximiser une fonction linéaire sous des contraintes linéaires

- Algorithme du simplexe : $\mathcal{O}(2^n)$ dans le pire des cas... faiblement polynomial en pratique !
- Algorithme du point intérieur : polynomial

Chercher un circuit hamiltonien dans un graphe

- Algorithme énumérant toutes les permutations des sommets : $\mathcal{O}(n!)$

Complexité d'un problème

Complexité d'un problème X

- Complexité du meilleur algorithme résolvant X...
- ...pour l'instance la plus difficile de X
  Une instance de X = une valuation des données en entrée de X

Classes de complexité

- Définies par rapport à la machine de Turing
- Pour les problèmes de décision seulement
  $\rightsquigarrow$ réponse = *oui* ou *non*
  Exemple : 4 appartient-il au tableau $[1, 2, 4, 6, 12]$?
- Principales classes : P, NP, NP-complet, NP-difficile, indécidable

Complexité d'un problème

Complexité d'un problème X

- Complexité du meilleur algorithme résolvant X...
- ...pour l'instance la plus difficile de X
  Une instance de X = une valuation des données en entrée de X

Classes de complexité

- Définies par rapport à la machine de Turing
- Pour les problèmes de décision seulement
  $\rightsquigarrow$ réponse = *oui* ou *non*
  Exemple : 4 appartient-il au tableau $[1, 2, 4, 6, 12]$?
- Principales classes : P, NP, NP-complet, NP-difficile, indécidable

Classes de complexité (1)

La classe P

Problèmes pour lesquels il existe un algorithme **Polynomial**

- Complexité de $X \leq \mathcal{O}(n^k)$ avec
  - n = taille des données de X en entrée
  - k = constante indépendante de n

Exemples de problèmes de la classe P

- Rechercher un élément dans un tableau
- Trier un tableau, un fichier, une liste, ...
- Rechercher un plus court chemin entre 2 sommets d'un graphe
- ...

Classes de complexité (2)

La classe NP

- Problèmes pour lesquels il existe un algo **Polynomial**...
  ...pour une machine **Non déterministe** !
- Machine de Turing non déterministe
  - Exécute un nombre fini d'alternatives en parallèle
  - $X \in \text{NP} \Rightarrow \text{vérif}(X) \in \text{P}$
    où $\text{vérif}(X)$ = décider si une donnée est solution de X

Relation entre P et NP

- $\text{P} \subseteq \text{NP}$
- Conjecture : $\text{P} \neq \text{NP}$
  ...1 million de dollars à gagner !
  (cf http://www.claymath.org/millennium/)

Classes de complexité (2)

La classe NP

- Problèmes pour lesquels il existe un algo **Polynomial**...
  ...pour une machine **Non déterministe** !
- Machine de Turing non déterministe
  - Exécute un nombre fini d'alternatives en parallèle
  - $X \in \mathrm{NP} \Rightarrow \text{vérif}(X) \in \mathrm{P}$
    où vérif(X) = décider si une donnée est solution de X

Relation entre P et NP

- $\mathrm{P} \subseteq \mathrm{NP}$
- Conjecture : $\mathrm{P} \neq \mathrm{NP}$
  ...1 million de dollars à gagner !
  (cf http://www.claymath.org/millennium/)

Classes de complexité (3)

Problèmes NP-complets

- Les problèmes les plus difficiles de la classe NP
- X est NP-complet $\leadsto$ $X \in$ NP et $X \in P \Rightarrow$ P=NP
- Théorème de [Cook 1971] : SAT est NP-complet
- Depuis 1971, nombreux problèmes montrés NP-complets
  ... cf la suite du cours !

Démonstration de NP-complétude

- Montrer que le problème X appartient à NP
- Trouver une procédure polynomiale pour transformer un problème NP-complet en X

Classes de complexité (3)

Problèmes NP-complets

- Les problèmes les plus difficiles de la classe NP
- X est NP-complet $\leadsto$ $X \in$ NP et $X \in P \Rightarrow$ P=NP
- Théorème de [Cook 1971] : SAT est NP-complet
- Depuis 1971, nombreux problèmes montrés NP-complets

  ... cf la suite du cours !

Démonstration de NP-complétude

- Montrer que le problème X appartient à NP
- Trouver une procédure polynomiale pour transformer un problème NP-complet en X

Classes de complexité (4)

Problèmes NP-difficiles

Les problèmes au moins aussi difficiles que ceux de NP

- X est NP-difficile $\leadsto X \in \mathrm{P} \Rightarrow \mathrm{P} = \mathrm{NP}$
  (mais X n'est pas forcément dans NP)
- NP-complet $\subset$ NP-difficile

Problèmes indécidables

Problèmes pour lesquels il n'existe pas d'algorithme

- Exemples : problème de l'arrêt, équations diophantiennes

Pour en savoir plus

« Computational complexity » de Papadimitriou
et le zoo des complexités `http://www.complexityzoo.com/`

Classes de complexité (4)

Problèmes NP-difficiles

Les problèmes au moins aussi difficiles que ceux de NP

- X est NP-difficile $\leadsto$ $X \in \mathrm{P} \Rightarrow \mathrm{P} = \mathrm{NP}$
  (mais X n'est pas forcément dans NP)
- NP-complet $\subset$ NP-difficile

Problèmes indécidables

Problèmes pour lesquels il n'existe pas d'algorithme

- Exemples : problème de l'arrêt, équations diophantiennes

Pour en savoir plus

« Computational complexity » de Papadimitriou
et le zoo des complexités `http://www.complexityzoo.com/`

Classes de complexité (4)

Problèmes NP-difficiles

Les problèmes au moins aussi difficiles que ceux de NP

- X est NP-difficile $\rightsquigarrow$ $X \in$ P $\Rightarrow$ P = NP
  (mais X n'est pas forcément dans NP)
- NP-complet $\subset$ NP-difficile

Problèmes indécidables

Problèmes pour lesquels il n'existe pas d'algorithme

- Exemples : problème de l'arrêt, équations diophantiennes

Pour en savoir plus

« Computational complexity » de Papadimitriou
et le zoo des complexités `http://www.complexityzoo.com/`

Complexité des problèmes d'optimisation

- Classes de complexité définies pour les problèmes de décision
  - Réponse = *oui* ou *non*
  - Exemple : Peut-on colorier ce graphe avec 4 couleurs ?
- Problèmes d'optimisation
  - Réponse = valeur qui optimise une fonction objectif donnée
  - Exemple : Plus petit nombre de couleurs permettant de colorier ce graphe ?
- Problème de décision associé à un problème d'optimisation
  - Existe t'il une meilleure solution ?
  - Exemple : Peut-on colorier ce graphe avec moins de 5 couleurs ?
- Complexité du problème d'optimisation / problème de décision
  - si le problème de décision est NP-complet, alors le problème d'optimisation est dit NP-difficile

Complexité des problèmes d'optimisation

- Classes de complexité définies pour les problèmes de décision
  - Réponse = *oui* ou *non*
  - Exemple : Peut-on colorier ce graphe avec 4 couleurs ?
- Problèmes d'optimisation
  - Réponse = valeur qui optimise une fonction objectif donnée
  - Exemple : Plus petit nombre de couleurs permettant de colorier ce graphe ?
- Problème de décision associé à un problème d'optimisation
  - Existe t'il une meilleure solution ?
  - Exemple : Peut-on colorier ce graphe avec moins de 5 couleurs ?
- Complexité du problème d'optimisation / problème de décision
  - si le problème de décision est NP-complet,
    alors le problème d'optimisation est dit NP-difficile

Complexité des problèmes d'optimisation

- Classes de complexité définies pour les problèmes de décision
  - Réponse = *oui* ou *non*
  - Exemple : Peut-on colorier ce graphe avec 4 couleurs ?
- Problèmes d'optimisation
  - Réponse = valeur qui optimise une fonction objectif donnée
  - Exemple : Plus petit nombre de couleurs permettant de colorier ce graphe ?
- Problème de décision associé à un problème d'optimisation
  - Existe t'il une meilleure solution ?
  - Exemple : Peut-on colorier ce graphe avec moins de 5 couleurs ?
- Complexité du problème d'optimisation / problème de décision
  - si le problème de décision est NP-complet,
    alors le problème d'optimisation est dit NP-difficile

Complexité des problèmes d'optimisation

- Classes de complexité définies pour les problèmes de décision
  - Réponse = *oui* ou *non*
  - Exemple : Peut-on colorier ce graphe avec 4 couleurs ?
- Problèmes d'optimisation
  - Réponse = valeur qui optimise une fonction objectif donnée
  - Exemple : Plus petit nombre de couleurs permettant de colorier ce graphe ?
- Problème de décision associé à un problème d'optimisation
  - Existe t'il une meilleure solution ?
  - Exemple : Peut-on colorier ce graphe avec moins de 5 couleurs ?
- Complexité du problème d'optimisation / problème de décision
  - si le problème de décision est NP-complet,
    alors le problème d'optimisation est dit NP-difficile

Difficulté des problèmes NP-complets/difficiles

- Croissance exponentielle ?!

n	2^n	Temps (si 10^9 instr/s)
30	$\approx 10^9$	$\approx$ 1 seconde
40	$\approx 10^{12}$	$\approx$ 16 minutes
50	$\approx 10^{15}$	$\approx$ 11 jours
60	$\approx 10^{18}$	$\approx$ 32 ans
70	$\approx 10^{21}$	$\approx$ 317 siècles

- En théorie, on ne pourra traiter des données de taille > 50
- En pratique, on peut souvent faire bien mieux...
- De très nombreux problèmes réels sont NP-difficiles
  - Mesurer la similarité d'objets
  - Extraire des connaissances à partir de données
  - Trouver un plan d'action pour atteindre un objectif
  - Construire un emploi du temps
  - ...

Plan de la première partie : introduction

Exemple 1 : le problème SAT

Instance du problème SAT

- Formule booléenne sous forme clausale
- Exemple :
  $(a \vee \neg b) \wedge (\neg a \vee c \vee \neg d) \wedge (\neg b \vee \neg c \vee \neg e) \wedge (a \vee b \vee d \vee e)$

Problème de décision

- Peut-on satisfaire toutes les clauses ?
- NP-complet dans le cas général
  ... mais polynomial si toutes les clauses ont au plus 2 littéraux

Problème d'optimisation

- Plus grand nombre de clauses pouvant être satisfaites ?
- NP-difficile dans le cas général

Espace de recherche

Ensemble des valuations possibles $\leadsto \mathcal{O}(2^n)$ si n variables

Exemple 1 : le problème SAT

Instance du problème SAT

- Formule booléenne sous forme clausale
- Exemple :
  $(a \lor \neg b) \land (\neg a \lor c \lor \neg d) \land (\neg b \lor \neg c \lor \neg e) \land (a \lor b \lor d \lor e)$

Problème de décision

- Peut-on satisfaire toutes les clauses ?
- NP-complet dans le cas général
  ... mais polynomial si toutes les clauses ont au plus 2 littéraux

Problème d'optimisation

- Plus grand nombre de clauses pouvant être satisfaites ?
- NP-difficile dans le cas général

Espace de recherche

Ensemble des valuations possibles $\rightsquigarrow \mathcal{O}(2^n)$ si n variables

Exemple 1 : le problème SAT

Instance du problème SAT

- Formule booléenne sous forme clausale
- Exemple :
  $(a \vee \neg b) \wedge (\neg a \vee c \vee \neg d) \wedge (\neg b \vee \neg c \vee \neg e) \wedge (a \vee b \vee d \vee e)$

Problème de décision

- Peut-on satisfaire toutes les clauses ?
- NP-complet dans le cas général
  ... mais polynomial si toutes les clauses ont au plus 2 littéraux

Problème d'optimisation

- Plus grand nombre de clauses pouvant être satisfaites ?
- NP-difficile dans le cas général

Espace de recherche

Ensemble des valuations possibles $\rightsquigarrow \mathcal{O}(2^n)$ si n variables

Exemple 1 : le problème SAT

Instance du problème SAT

- Formule booléenne sous forme clausale
- Exemple :
  $(a \vee \neg b) \wedge (\neg a \vee c \vee \neg d) \wedge (\neg b \vee \neg c \vee \neg e) \wedge (a \vee b \vee d \vee e)$

Problème de décision

- Peut-on satisfaire toutes les clauses ?
- NP-complet dans le cas général
  ... mais polynomial si toutes les clauses ont au plus 2 littéraux

Problème d'optimisation

- Plus grand nombre de clauses pouvant être satisfaites ?
- NP-difficile dans le cas général

Espace de recherche

Ensemble des valuations possibles $\rightsquigarrow \mathcal{O}(2^n)$ si n variables

Exemple 2 : les CSPs (1)

Définition

Problème de Satisfaction de Contraintes (CSP) défini par (X, D, C)

- $X = \{X_1, X_2, \ldots, X_n\}$ est l'ensemble des variables
- $D(X_i)$ est le domaine de X_i
- C est l'ensemble des contraintes du problème

Exemple

- $X = \{X_1, X_2, X_3, X_4\}$
- $D(X_i) = \{0, 1\}$, pour toute variable $X_i \in X$
- $C = \{X_1 \neq X_2, \quad X_3 \neq X_4, \quad X_1 + X_3 < X_2\}$

Pour en savoir plus...

...voir les sessions 1 et 2 du cours en ligne
`http://www710.univ-lyon1.fr/~csolnon/PPC.html`

Exemple 2 : les CSPs (2)

Problème de décision

- Peut-on satisfaire toutes les contraintes ?
  $\rightsquigarrow$ trouver une affectation satisfaisant les contraintes
- NP-complet dans le cas général
  ... polynomial dans certains cas (ex: contraintes linéaires / réels)
  ... parfois indécidable (ex: contraintes non linéaires / réels)

Problèmes d'optimisation

- Plus grand nb de contraintes pouvant être satisfaites ?
- CSP basés sur les semi-anneaux
  $\rightsquigarrow$ Expression de préférences entre contraintes
  $\rightsquigarrow$ Chercher à optimiser ces préférences
- NP-difficile dans le cas général

Espace de recherche (cas général)

Ensemble des affectations possibles

- k^n valuations si $|X| = n$ et $|D(X_i)| = k, \forall X_i \in X$

Exemple 2 : les CSPs (2)

Problème de décision

- Peut-on satisfaire toutes les contraintes ?
  $\rightsquigarrow$ trouver une affectation satisfaisant les contraintes
- NP-complet dans le cas général
  ... polynomial dans certains cas (ex: contraintes linéaires / réels)
  ... parfois indécidable (ex: contraintes non linéaires / réels)

Problèmes d'optimisation

- Plus grand nb de contraintes pouvant être satisfaites ?
- CSP basés sur les semi-anneaux
  $\rightsquigarrow$ Expression de préférences entre contraintes
  $\rightsquigarrow$ Chercher à optimiser ces préférences
- NP-difficile dans le cas général

Espace de recherche (cas général)

Ensemble des affectations possibles

- k^n valuations si $|X| = n$ et $|D(X_i)| = k, \forall X_i \in X$

Exemple 2 : les CSPs (2)

Problème de décision

- Peut-on satisfaire toutes les contraintes ?
  $\rightsquigarrow$ trouver une affectation satisfaisant les contraintes
- NP-complet dans le cas général
  ... polynomial dans certains cas (ex: contraintes linéaires / réels)
  ... parfois indécidable (ex: contraintes non linéaires / réels)

Problèmes d'optimisation

- Plus grand nb de contraintes pouvant être satisfaites ?
- CSP basés sur les semi-anneaux
  $\rightsquigarrow$ Expression de préférences entre contraintes
  $\rightsquigarrow$ Chercher à optimiser ces préférences
- NP-difficile dans le cas général

Espace de recherche (cas général)

Ensemble des affectations possibles

- k^n valuations si $|X| = n$ et $|D(X_i)| = k, \forall X_i \in X$

Exemple 3 : Problèmes de planification (1)

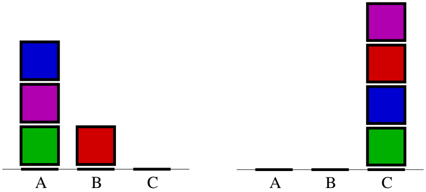
Instance d'un problème de planification

- Définie par
  - Un ensemble (éventuellement infini) d'états E
  - Un état initial $E_0 \in E$
  - Un ensemble d'états finaux $F \subseteq E$
  - Un ensemble d'op. O permettant de passer d'états en états $E_i \xrightarrow{O_k} E_j$ si O_k permet de passer de E_i à E_j
- Un plan est une suite d'opérations permettant de passer de l'état initial E_0 à un état final de F

$$\text{Plan} = E_0 \xrightarrow{O_1} E_1 \xrightarrow{O_2} E_2 \ldots E_{n-1} \xrightarrow{O_n} E_n \text{ avec } E_n \in F$$

Exemple 3 : Problèmes de planification (2)

Exemple : le monde des blocs

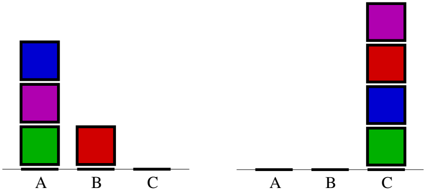


Etat initial   Etat final

Opérations = prendre un cube au sommet d'une pile et le poser au sommet d'une autre pile

Exemple 3 : Problèmes de planification (2)

Exemple : le monde des blocs



Etat initial Etat final

Opérations = prendre un cube au sommet d'une pile et le poser au sommet d'une autre pile

Exemple de plan pour ce problème :

- déplacer le cube bleu de A vers B,
- déplacer le cube rose de A vers B,
- déplacer le cube vert de A vers C,
- ...

Exemple 3 : Problèmes de planification (3)

Problème de décision

- Existe-t'il un plan ?
- Algorithme polynomial / graphe $G = (E, O)$

  ...mais le nombre d'états est généralement exponentiel par rapport à leur taille !

Problèmes d'optimisation

- Trouver le plus petit plan / nombre d'opérations
- Trouver le meilleur plan / objectif donné

Exemple 3 : Problèmes de planification (3)

Problème de décision

- Existe-t'il un plan ?
- Algorithme polynomial / graphe $G = (E, O)$

  ...mais le nombre d'états est généralement exponentiel par rapport à leur taille !

Problèmes d'optimisation

- Trouver le plus petit plan / nombre d'opérations
- Trouver le meilleur plan / objectif donné

Exemple 4 : Problèmes dans les graphes (1)

Définition d'un graphe

- $G = (S, A)$ tel que
  - S = ensemble de sommets
  - $A \subseteq S \times S$ = ensemble d'arêtes (ou arcs si orientés)
- Les arêtes/arcs et sommets peuvent être étiquetés

Modélisation d'objets par des graphes

- Sommets $\rightsquigarrow$ composants de l'objet
  Etiquettes associées aux sommets $\rightsquigarrow$ couleur, taille, forme, ...
- Arêtes $\rightsquigarrow$ relations binaires entre les composants
  Etiquettes associées aux arêtes $\rightsquigarrow$ position relative, distance, ...

Exemple 4 : Problèmes dans les graphes (1)

Définition d'un graphe

- $G = (S, A)$ tel que
  - S = ensemble de sommets
  - $A \subseteq S \times S$ = ensemble d'arêtes (ou arcs si orientés)
- Les arêtes/arcs et sommets peuvent être étiquetés

Modélisation d'objets par des graphes

- Sommets $\leadsto$ composants de l'objet
  Etiquettes associées aux sommets $\leadsto$ couleur, taille, forme, ...
- Arêtes $\leadsto$ relations binaires entre les composants
  Etiquettes associées aux arêtes $\leadsto$ position relative, distance, ...

Exemple 4 : Problèmes dans les graphes (2)

Exemples de problèmes polynomiaux sur les graphes

- Déterminer les sous-ensembles de sommets connectés
- Trouver le plus court chemin entre 2 sommets
- Trouver un chemin passant 1 fois par chaque arc (Eulérien)
- Connecter un ensemble de sommets au plus faible coût
- Maximiser un flot de véhicules dans un réseau de transport

Exemples de problèmes NP-complets/difficiles

- Trouver un plus court cycle hamiltonien
  $\rightsquigarrow$ Tournée d'un voyageur de commerce
- Trouver k sommets connectés 2 à 2 par des arêtes
  $\rightsquigarrow$ Problème de la clique
- Comparer des graphes
  $\rightsquigarrow$ Problèmes d'appariements de graphes
- Colorier les sommets
  $\rightsquigarrow$ Allouer des ressources/contr. d'excl.
- Partitionner les sommets
  $\rightsquigarrow$ Classification, conception VLSI, ...

Exemple 4 : Problèmes dans les graphes (2)

Exemples de problèmes polynomiaux sur les graphes

- Déterminer les sous-ensembles de sommets connectés
- Trouver le plus court chemin entre 2 sommets
- Trouver un chemin passant 1 fois par chaque arc (Eulérien)
- Connecter un ensemble de sommets au plus faible coût
- Maximiser un flot de véhicules dans un réseau de transport

Exemples de problèmes NP-complets/difficiles

- Trouver un plus court cycle hamiltonien
  $\rightsquigarrow$ Tournée d'un voyageur de commerce
- Trouver *k* sommets connectés 2 à 2 par des arêtes
  $\rightsquigarrow$ Problème de la clique
- Comparer des graphes
  $\rightsquigarrow$ Problèmes d'appariements de graphes
- Colorier les sommets
  $\rightsquigarrow$ Allouer des ressources/contr. d'excl.
- Partitionner les sommets
  $\rightsquigarrow$ Classification, conception VLSI, ...

Exemple 4.1 : Le voyageur de commerce

- Soit un ensemble de villes reliées par des routes
  $\rightsquigarrow$ trouver la + courte tournée passant par chaque ville

Exemple 4.1 : Le voyageur de commerce

- Modélisation par un graphe étiqueté $G = (S, A)$ tel que
  - S = ensemble des villes
  - A = ensemble des routes reliant 2 villes
  - Chaque arête est étiquetée par la longueur de la route
- Problème de décision/satisfaction $\rightsquigarrow$ NP-complet
  - *Trouver un cycle hamiltonien de longueur totale $\leq k$.*
- Problème d'optimisation $\rightsquigarrow$ NP-difficile
  - *Trouver le cycle hamiltonien de plus petite longueur.*
- Espace de recherche = ensemble des permutations de S
  $\rightsquigarrow$ $n!$ permutations possibles si $|S| = n$

Exemple 4.1 : Le voyageur de commerce

- Modélisation par un graphe étiqueté $G = (S, A)$ tel que
  - S = ensemble des villes
  - A = ensemble des routes reliant 2 villes
  - Chaque arête est étiquetée par la longueur de la route
- Problème de décision/satisfaction $\rightsquigarrow$ NP-complet
  - *Trouver un cycle hamiltonien de longueur totale $\leq k$.*
- Problème d'optimisation $\rightsquigarrow$ NP-difficile
  - *Trouver le cycle hamiltonien de plus petite longueur.*
- Espace de recherche = ensemble des permutations de S
  $\rightsquigarrow$ $n!$ permutations possibles si $|S| = n$

Exemple 4.1 : Le voyageur de commerce

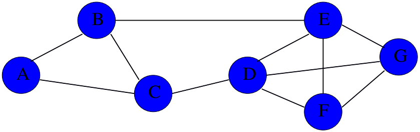
- Modélisation par un graphe étiqueté $G = (S, A)$ tel que
  - S = ensemble des villes
  - A = ensemble des routes reliant 2 villes
  - Chaque arête est étiquetée par la longueur de la route
- Problème de décision/satisfaction $\rightsquigarrow$ NP-complet
  - *Trouver un cycle hamiltonien de longueur totale $\leq k$.*
- Problème d'optimisation $\rightsquigarrow$ NP-difficile
  - *Trouver le cycle hamiltonien de plus petite longueur.*
- Espace de recherche = ensemble des permutations de S
  $\rightsquigarrow$ $n!$ permutations possibles si $|S| = n$

Exemple 4.1 : Le voyageur de commerce

- Modélisation par un graphe étiqueté $G = (S, A)$ tel que
  - S = ensemble des villes
  - A = ensemble des routes reliant 2 villes
  - Chaque arête est étiquetée par la longueur de la route
- Problème de décision/satisfaction $\rightsquigarrow$ NP-complet
  - *Trouver un cycle hamiltonien de longueur totale* $\leq k$.
- Problème d'optimisation $\rightsquigarrow$ NP-difficile
  - *Trouver le cycle hamiltonien de plus petite longueur.*
- Espace de recherche = ensemble des permutations de S
  $\rightsquigarrow$ $n!$ permutations possibles si $\mid S \mid = n$

Exemple 4.2 : Le problème de la clique

- Clique = ensemble de sommets tous connectés par des arêtes

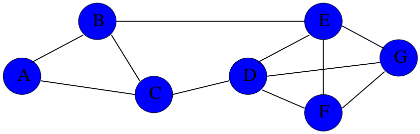


$\{A, B\}$, $\{A, B, C\}$, $\{D, E, F\}$, $\{D, E, F, G\}$, ... sont des cliques

- Problème de décision/satisfaction $\rightsquigarrow$ NP-complet
  - *Trouver une clique comportant k sommets.*
- Problème d'optimisation $\rightsquigarrow$ NP-difficile
  - *Trouver la plus grande clique (clique maximum).*
- Espace de recherche = ensemble des sous-ensembles de S
  $\rightsquigarrow 2^n$ sous-ensembles possibles si $\mid S \mid = n$
- Problème dual = problème du stable

Exemple 4.2 : Le problème de la clique

- Clique = ensemble de sommets tous connectés par des arêtes

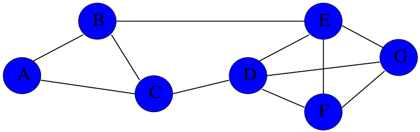


$\{A, B\}$, $\{A, B, C\}$, $\{D, E, F\}$, $\{D, E, F, G\}$, ... sont des cliques

- Problème de décision/satisfaction $\rightsquigarrow$ NP-complet
  - *Trouver une clique comportant k sommets.*
- Problème d'optimisation $\rightsquigarrow$ NP-difficile
  - *Trouver la plus grande clique (clique maximum).*
- Espace de recherche = ensemble des sous-ensembles de S
  $\rightsquigarrow 2^n$ sous-ensembles possibles si $\mid S \mid = n$
- Problème dual = problème du stable

Exemple 4.2 : Le problème de la clique

- Clique = ensemble de sommets tous connectés par des arêtes

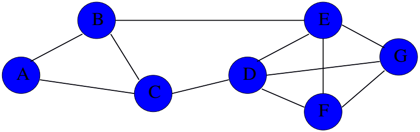


$\{A, B\}$, $\{A, B, C\}$, $\{D, E, F\}$, $\{D, E, F, G\}$, ... sont des cliques

- Problème de décision/satisfaction $\rightsquigarrow$ NP-complet
  - *Trouver une clique comportant k sommets.*
- Problème d'optimisation $\rightsquigarrow$ NP-difficile
  - *Trouver la plus grande clique (clique maximum).*
- Espace de recherche = ensemble des sous-ensembles de S
  $\rightsquigarrow 2^n$ sous-ensembles possibles si $\mid S \mid = n$
- Problème dual = problème du stable

Exemple 4.2 : Le problème de la clique

- Clique = ensemble de sommets tous connectés par des arêtes

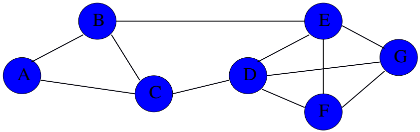


$\{A, B\}$, $\{A, B, C\}$, $\{D, E, F\}$, $\{D, E, F, G\}$, ... sont des cliques

- Problème de décision/satisfaction $\rightsquigarrow$ NP-complet
  - *Trouver une clique comportant k sommets.*
- Problème d'optimisation $\rightsquigarrow$ NP-difficile
  - *Trouver la plus grande clique (clique maximum).*
- Espace de recherche = ensemble des sous-ensembles de S
  $\rightsquigarrow$ 2^n sous-ensembles possibles si $\mid S \mid = n$
- Problème dual = problème du stable

Exemple 4.2 : Le problème de la clique

- Clique = ensemble de sommets tous connectés par des arêtes



$\{A, B\}$, $\{A, B, C\}$, $\{D, E, F\}$, $\{D, E, F, G\}$, ... sont des cliques

- Problème de décision/satisfaction $\leadsto$ NP-complet
  - *Trouver une clique comportant k sommets.*
- Problème d'optimisation $\leadsto$ NP-difficile
  - *Trouver la plus grande clique (clique maximum).*
- Espace de recherche = ensemble des sous-ensembles de S
  $\leadsto$ 2^n sous-ensembles possibles si $\mid S \mid = n$
- Problème dual = problème du stable

Exemple 4.3 : Similarité de graphes

Pourquoi mesurer la similarité de graphes ?

- Recherche d'info : trouver le document le + similaire à la requête
- Classification : regrouper les objets similaires
- RàPC : trouver un pb (déjà résolu) similaire au pb à résoudre
- ...

Comment mesurer la similarité de graphes ?

- Apparier les sommets des 2 graphes (mise-en-encorrespondance)
- Plusieurs appariements possibles $\leadsto$ trouver le meilleur
  $\leadsto$ celui qui permet d'apparier le plus d'arêtes
  $\leadsto$ celui qui permet d'apparier le plus d'étiquettes
- Similarité fonction du nombre d'arêtes/étiquettes appariées

Exemple 4.3 : Similarité de graphes

Pourquoi mesurer la similarité de graphes ?

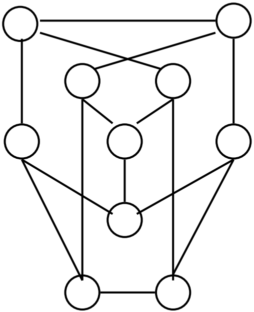
- Recherche d'info : trouver le document le + similaire à la requête
- Classification : regrouper les objets similaires
- RàPC : trouver un pb (déjà résolu) similaire au pb à résoudre
- ...

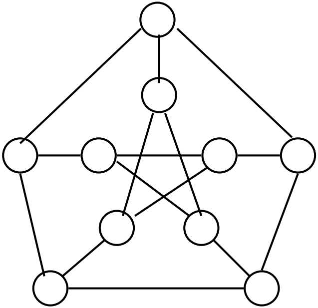
Comment mesurer la similarité de graphes ?

- Apparier les sommets des 2 graphes (mise-en-encorrespondance)
- Plusieurs appariements possibles $\rightsquigarrow$ trouver le meilleur
  $\rightsquigarrow$ celui qui permet d'apparier le plus d'arêtes
  $\rightsquigarrow$ celui qui permet d'apparier le plus d'étiquettes
- Similarité fonction du nombre d'arêtes/étiquettes appariées

Exemple 4.3 : Similarité de graphes

- Isomorphisme de graphes ⇒ tester l'équivalence
- Isomorphisme de sous-graphe ⇒ tester l'inclusion
- Plus grand sous-graphe commun ⇒ quantifier l'intersection
- Meilleur appariement multivoque ⇒ quantifier la similarité

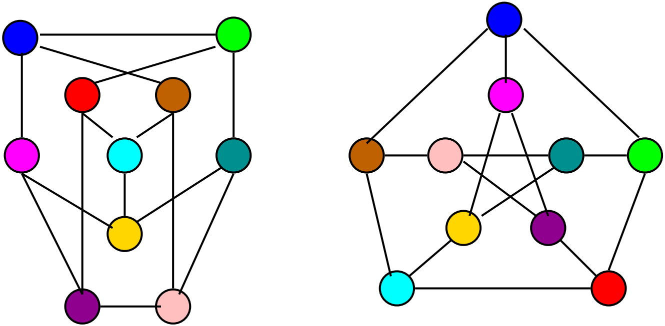




Graphes identiques ?

Exemple 4.3 : Similarité de graphes

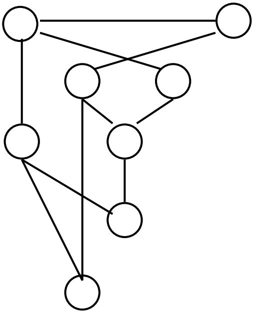
- Isomorphisme de graphes ⇒ tester l'équivalence
- Isomorphisme de sous-graphe ⇒ tester l'inclusion
- Plus grand sous-graphe commun ⇒ quantifier l'intersection
- Meilleur appariement multivoque ⇒ quantifier la similarité

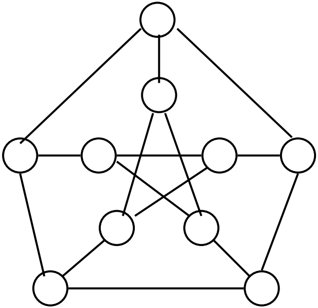


Complexité ouverte : Problème isomorphique-complet !!!

Exemple 4.3 : Similarité de graphes

- Isomorphisme de graphes $\Rightarrow$ tester l'équivalence
- Isomorphisme de sous-graphe $\Rightarrow$ tester l'inclusion
- Plus grand sous-graphe commun $\Rightarrow$ quantifier l'intersection
- Meilleur appariement multivoque $\Rightarrow$ quantifier la similarité

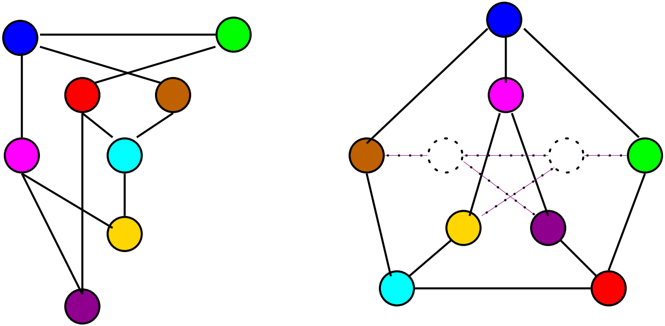




Inclusion ?

Exemple 4.3 : Similarité de graphes

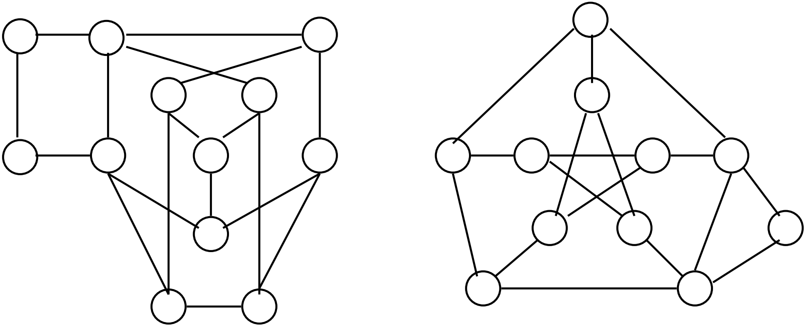
- Isomorphisme de graphes ⇒ tester l'équivalence
- Isomorphisme de sous-graphe ⇒ tester l'inclusion
- Plus grand sous-graphe commun ⇒ quantifier l'intersection
- Meilleur appariement multivoque ⇒ quantifier la similarité



Problème NP-complet

Exemple 4.3 : Similarité de graphes

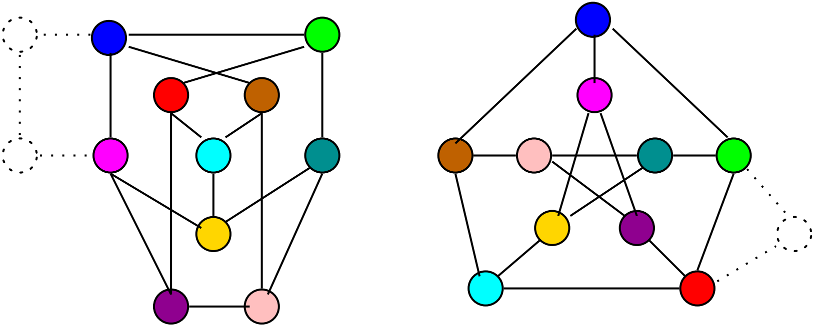
- Isomorphisme de graphes ⇒ tester l'équivalence
- Isomorphisme de sous-graphe ⇒ tester l'inclusion
- Plus grand sous-graphe commun ⇒ quantifier l'intersection
- Meilleur appariement multivoque ⇒ quantifier la similarité



Nombre de sommets communs ?

Exemple 4.3 : Similarité de graphes

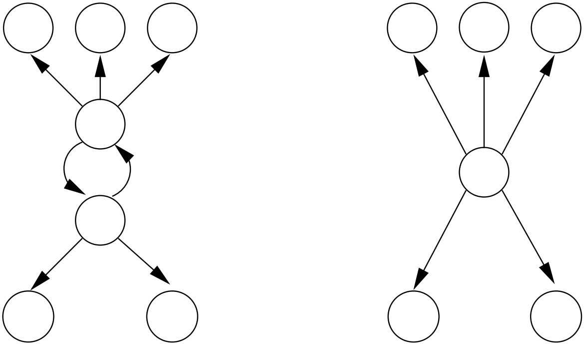
- Isomorphisme de graphes ⇒ tester l'équivalence
- Isomorphisme de sous-graphe ⇒ tester l'inclusion
- Plus grand sous-graphe commun ⇒ quantifier l'intersection
- Meilleur appariement multivoque ⇒ quantifier la similarité



Problème NP-difficile

Exemple 4.3 : Similarité de graphes

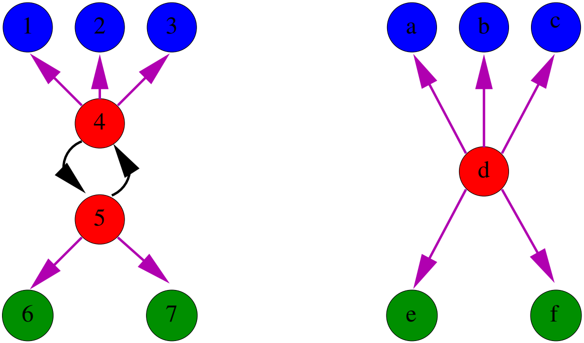
- Isomorphisme de graphes $\Rightarrow$ tester l'équivalence
- Isomorphisme de sous-graphe $\Rightarrow$ tester l'inclusion
- Plus grand sous-graphe commun $\Rightarrow$ quantifier l'intersection
- Meilleur appariement multivoque $\Rightarrow$ quantifier la similarité



Fusions / éclatements de sommets possibles

Exemple 4.3 : Similarité de graphes

- Isomorphisme de graphes ⇒ tester l'équivalence
- Isomorphisme de sous-graphe ⇒ tester l'inclusion
- Plus grand sous-graphe commun ⇒ quantifier l'intersection
- Meilleur appariement multivoque ⇒ quantifier la similarité



Problème NP-difficile

Exemple 5 : Recherche d'ensembles fréquents

- Contexte : Extraction de Connaissances à partir de Données
- Exemple de données :
  - $I = \{i_1, i_2, \ldots, i_m\}$ un ensemble d'attributs (items)
  - $D = \{T_1, T_2, \ldots, T_k\}$ un ensemble de transactions tq $T_j \subseteq I$

attributs :	i_1	i_2	i_3	i_4
T_1 :	×		×	
T_2 :	×	×	×	
T_3 :		×		×
T_4 :		×	×	×
T_5 :	×	×	×	

- Exemples de « connaissances » :
  - Ensembles fréquents : $\{i_1, i_3\}$ a une fréquence $\geq 3/5$
  - Règles d'association : $\{i_1, i_3\} \Rightarrow \{i_2\}[3/5, 2/3]$
- Espace de recherche = ensemble des sous-ensembles de I

$\leadsto$ 2^n sous-ensembles possibles si $|I| = n$

$\leadsto$ ... le nombre k de transactions dans D est souvent très grand

Exemple 5 : Recherche d'ensembles fréquents

- Contexte : Extraction de Connaissances à partir de Données
- Exemple de données :
  - $I = \{i_1, i_2, \ldots, i_m\}$ un ensemble d'attributs (items)
  - $D = \{T_1, T_2, \ldots, T_k\}$ un ensemble de transactions tq $T_j \subseteq I$

attributs :	i_1	i_2	i_3	i_4
T_1 :	$\times$		$\times$	
T_2 :	$\times$	$\times$	$\times$	
T_3 :		$\times$		$\times$
T_4 :		$\times$	$\times$	$\times$
T_5 :	$\times$	$\times$	$\times$	

- Exemples de « connaissances » :
  - Ensembles fréquents : $\{i_1, i_3\}$ a une fréquence $\geq 3/5$
  - Règles d'association : $\{i_1, i_3\} \Rightarrow \{i_2\}[3/5, 2/3]$
- Espace de recherche = ensemble des sous-ensembles de I

$\leadsto$ 2^n sous-ensembles possibles si $|I| = n$

$\leadsto$ ... le nombre k de transactions dans D est souvent très grand

Exemple 5 : Recherche d'ensembles fréquents

- Contexte : Extraction de Connaissances à partir de Données
- Exemple de données :
  - $I = \{i_1, i_2, \ldots, i_m\}$ un ensemble d'attributs (items)
  - $D = \{T_1, T_2, \ldots, T_k\}$ un ensemble de transactions tq $T_j \subseteq I$

attributs :	i_1	i_2	i_3	i_4
T_1 :	×		×	
T_2 :	×	×	×	
T_3 :		×		×
T_4 :		×	×	×
T_5 :	×	×	×	

- Exemples de « connaissances » :
  - Ensembles fréquents : $\{i_1, i_3\}$ a une fréquence $\geq 3/5$
  - Règles d'association : $\{i_1, i_3\} \Rightarrow \{i_2\}[3/5, 2/3]$
- Espace de recherche = ensemble des sous-ensembles de I

$\leadsto$ 2^n sous-ensembles possibles si $|I| = n$

$\leadsto$ ... le nombre k de transactions dans D est souvent très grand

Exemple 5 : Recherche d'ensembles fréquents

- Contexte : Extraction de Connaissances à partir de Données
- Exemple de données :
  - $I = \{i_1, i_2, \ldots, i_m\}$ un ensemble d'attributs (items)
  - $D = \{T_1, T_2, \ldots, T_k\}$ un ensemble de transactions tq $T_j \subseteq I$

attributs :	i_1	i_2	i_3	i_4
T_1 :	$\times$		$\times$	
T_2 :	$\times$	$\times$	$\times$	
T_3 :		$\times$		$\times$
T_4 :		$\times$	$\times$	$\times$
T_5 :	$\times$	$\times$	$\times$	

- Exemples de « connaissances » :
  - Ensembles fréquents : $\{i_1, i_3\}$ a une fréquence $\geq 3/5$
  - Règles d'association : $\{i_1, i_3\} \Rightarrow \{i_2\}[3/5, 2/3]$
- Espace de recherche = ensemble des sous-ensembles de I

$\rightsquigarrow$ 2^n sous-ensembles possibles si $|I| = n$

$\rightsquigarrow$ ... le nombre k de transactions dans D est souvent très grand

Plan de la première partie : introduction

Transition de phases

- Toutes les instances d'un problème ne sont pas aussi difficiles
- Exemple : Décider de la satisfiabilité d'une instance SAT donnée
  - La difficulté dépend du nombre n de variables...
    $\rightsquigarrow$ taille de l'espace de recherche = 2^n
  - ... mais aussi du nombre p de clauses
    - Peu de clauses $\Rightarrow$ Instance sous-contrainte
      $\rightsquigarrow$ Solution trouvée facilement
    - Beaucoup de clauses $\Rightarrow$ Instance sur-contrainte
      $\rightsquigarrow$ Inconsistance facilement montrée... ou solution facilement trouvée
    - entre les 2 $\Rightarrow$ ça se complique !!!
- Etude sur des instances de 3-SAT (3 litteraux par clause)
  - Génération aléatoire d'instances, en fonction
    - du nombre n de variables ($n \in \{10, 20, 30, 40, 50\}$),
    - du nombre p de clauses ($n \leq p \leq 8n$)
  - Etude du taux d'instances satisfiables en fct du ratio p/n
  - Etude du temps de résolution en fct du ratio p/n

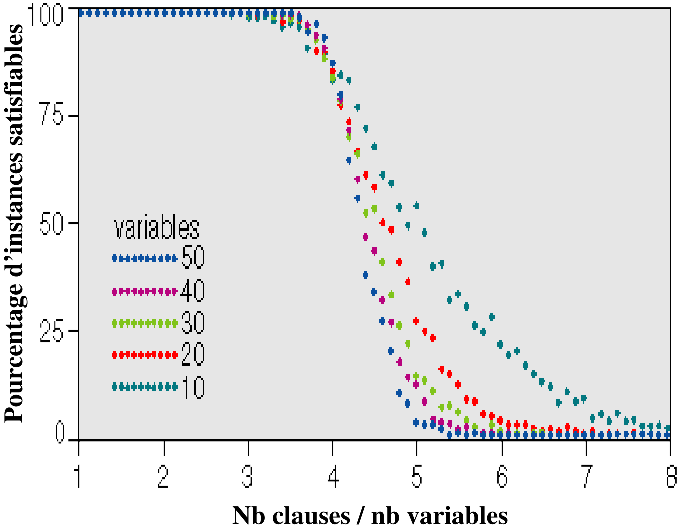
Transition de phases

- Toutes les instances d'un problème ne sont pas aussi difficiles
- Exemple : Décider de la satisfiabilité d'une instance SAT donnée
  - La difficulté dépend du nombre n de variables...
    $\leadsto$ taille de l'espace de recherche = 2^n
  - ... mais aussi du nombre p de clauses
    - Peu de clauses $\Rightarrow$ Instance sous-contrainte
      $\leadsto$ Solution trouvée facilement
    - Beaucoup de clauses $\Rightarrow$ Instance sur-contrainte
      $\leadsto$ Inconsistance facilement montrée... ou solution facilement trouvée
    - entre les 2 $\Rightarrow$ ça se complique !!!
- Etude sur des instances de 3-SAT (3 litteraux par clause)
  - Génération aléatoire d'instances, en fonction
    - du nombre n de variables ($n \in \{10, 20, 30, 40, 50\}$),
    - du nombre p de clauses ($n \leq p \leq 8n$)
  - Etude du taux d'instances satisfiables en fct du ratio p/n
  - Etude du temps de résolution en fct du ratio p/n

Transition de phases

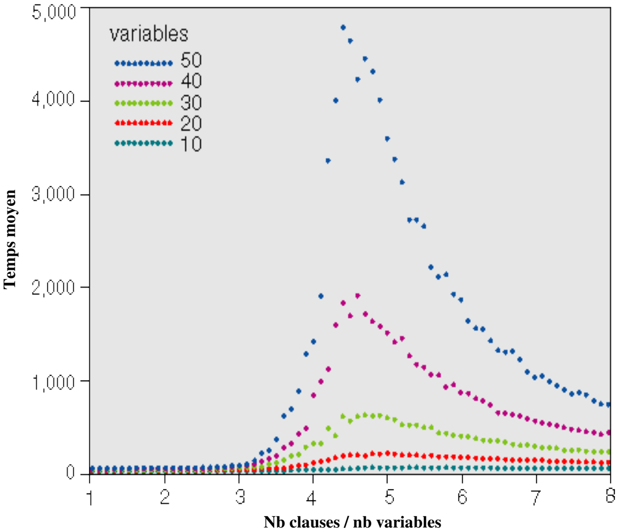
- Toutes les instances d'un problème ne sont pas aussi difficiles
- Exemple : Décider de la satisfiabilité d'une instance SAT donnée
  - La difficulté dépend du nombre n de variables...
    $\rightsquigarrow$ taille de l'espace de recherche = 2^n
  - ... mais aussi du nombre p de clauses
    - Peu de clauses $\Rightarrow$ Instance sous-contrainte
      $\rightsquigarrow$ Solution trouvée facilement
    - Beaucoup de clauses $\Rightarrow$ Instance sur-contrainte
      $\rightsquigarrow$ Inconsistance facilement montrée... ou solution facilement trouvée
    - entre les 2 $\Rightarrow$ ça se complique !!!
- Etude sur des instances de 3-SAT (3 litteraux par clause)
  - Génération aléatoire d'instances, en fonction
    - du nombre n de variables ($n \in \{10, 20, 30, 40, 50\}$),
    - du nombre p de clauses ($n \leq p \leq 8n$)
  - Etude du taux d'instances satisfiables en fct du ratio p/n
  - Etude du temps de résolution en fct du ratio p/n

Transition de phases



(image empruntée à des transparents de Toby Walsh)

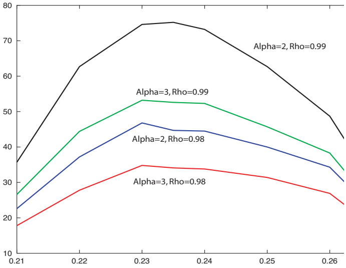
Transition de phases



(image empruntée à des transparents de Toby Walsh)

Transition de phases

- Pour 3-SAT : changement abrupte entre satisfiable et non satisfiable quand $\frac{\text{nb clauses}}{\text{nb variables}} = 4.3$
- Ce phénomène est appelé transition de phases
  - Correspond à un pic de difficulté pour la résolution
  - Apparait aussi pour de très nombreux autres problèmes
  - Indépendant de l'algorithme utilisé
- Exemple : résolution de CSPs binaires aléatoires par ACO



On fixe :
- le nb de variables = 100
- la taille du domaine des variables = 8
- le nombre de contraintes = 14%
On fait varier le nombre de couples de valeurs enlevés par contrainte de 21% à 27%

$\rightsquigarrow$ **Pic autour de 23%**

Transition de phases

Où se trouve la transition de phases ?

- Là où la probabilité d'être satisfiable est de 50%
- Estimer le taux de constrainedness κ d'une instance :
  - soit 2^n la taille de l'espace de recherche de l'instance
  - soit p le nombre estimé de solutions de l'instance
  - $\kappa = 1 - \frac{log_2(p)}{n}$
- La transition de phases se trouve là où $\kappa = 1$:
  - Si $\kappa \approx 0$ alors l'instance est sous-contrainte
  - Si $\kappa \approx 1$ alors l'instance est critique
  - Si $\kappa \approx \infty$ alors l'instance est sur-contrainte

Transition de phases

A quoi ça sert ?

- A tester son nouvel algorithme sur les instances vraiment difficiles
- A concevoir de meilleurs algorithmes
- ... et puis c'est plutôt fascinant, non ?

Pour en savoir plus

- Where the really hard problems are ?

  P. Cheeseman, B. Kanefsky et W.M. Taylor, [IJCAI'1991]
- http://www.dcs.st-and.ac.uk/∼apes/kappa/obj1.html