Plan

259 / 369

Encore des maths

La base de l'informatique quantique analogique est la théorie de la complexité

Dans les grandes lignes :

- certains problèmes "compliqués" peuvent se traduire sous la forme d'une expérience de physique quantique
- on modifie le problème que l'on veut résoudre pour le ramener à cette expérience
- on fait une "expérience instrumentée" dans un QPU analogique
- on en déduit la solution du problème initial

Parmi les implémentations analogiques, on trouve dans le paysage actuel :

- les QPU D-Wave basés sur des boucles supraconductrices
- les QPU Pasqal basés sur des atomes froids manipulés par des lasers
- les machines quantiques photoniques comme ce que propose Qandela

P et NP

En théorie de la complexité informatique, on identifie différentes classes, parmi elles

- les problèmes **de classe P** peuvent être résolu dans un temps qui varie de manière polynomiale avec la taille du problème. Les problèmes P peuvent être résolus avec du HPC
- les problèmes **de classe NP** sont *non-deterministic polynomial*, le temps de résolution est "pire" que polynomiale, mais si on me propose une solution, vérifier qu'elle résoud effectivement le problème prend un temps polynomial.

ATTENTION : NP ne **signifie pas** non-polynomial (c'est un très mauvais acronyme).

Exemples de P et de NP

Les problèmes polynomiales sont ceux du HPC classique, par exemple

- trouver le PGCD ou le PPCM de deux nombres très grands,
- inverser une matrice,
- savoir si un nombre est premier (algorithme AKS)

Exemple de problème NP : factorisation d'un produit de deux nombres premiers

- Soit le nombre 62062883, quelles sont (de tête) ses facteurs ?
- C'est compliqué à trouver...

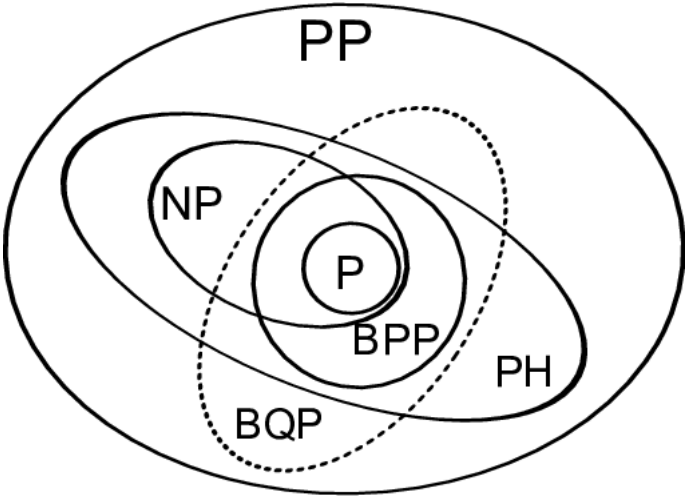
J'affirme que $62062883 = 7877 \times 7879$,

- On m'a exposé une solution potentielle
- vérifier qu'elle est correcte est simple... on fait la multiplication (c'est polynomial)

La factorisation est typiquement un problème NP.

262 / 369

N et NP ne sont pas les seules classes de complexité



Les problèmes NP-Complets - théorème de Cook

Certains problèmes sont dit **NP-difficiles** si tous les problèmes NP peuvent se ramener à résoudre ce problème en particulier.

Un problème NP qui est également NP-difficile est dit **NP-complet**.

Le théorème de Cook démontre l'existence de problèmes NP-Complets.

Résoudre un problème NP-Complets permet d'y ramener tous les autres problèmes NP au prix d'un algorithme polynomiale.
Il existe de nombreux problèmes NP-complets (QUBO, Voyageur de commerce, SAT, MIS, ...) on en recensait plus de 21 en 1972.

Plan

Les problèmes de Karp 1/2

En 1972, dans la foulée du théorème de Cook en 1971, le mathématicien Karp recensce 21 problèmes qui sont tous de nature **NPC**, on peut passer de l'un à l'autre au prix d'une transformation polynomiale.

- SATISFIABILITY : le problème SAT pour les formules en forme normale conjonctive
- CLIQUE : le problème de détection de clique dans un graphe
- SET PACKING : Set packing (empaquetage d'ensemble)
- VERTEX COVER : le problème de couverture par sommets, le problème MIS qui lui ets dual
- SET COVERING : le problème de couverture par ensembles
- FEEDBACK ARC SET : feedback arc set
- FEEDBACK NODE SET : feedback vertex set
- DIRECTED HAMILTONIAN CIRCUIT
- UNDIRECTED HAMILTONIAN CIRCUIT
- INTEGER PROGRAMMING : voir optimisation linéaire en nombres entiers
- 3-SAT : problème SAT dont les clauses ont 3 arguments au plus
- CHROMATIC NUMBER : coloration de graphe

266 / 369

Les problèmes de Karp 2/2

- CLIQUE COVER : partition en cliques
- EXACT COVER : couverture exacte
- MATCHING à 3 dimensions : appariement à 3 dimensions
- STEINER TREE : arbre de Steiner
- HITTING SET : ensemble intersectant
- KNAPSACK : problème du sac à dos
- JOB SEQUENCING : séquençage de tâches
- PARTITION : problème de partition
- MAX-CUT : problème de la coupe maximum

Plan

L'équation de Schrödinger : les racines du mal...

Pour comprendre l'informatique analogique quantique, il faut garder en tête la notion d'hamiltonien qui dérive directement de l'équation de Schrödinger

$$i\hbar\frac{\partial}{\partial t}\Psi(r,t) = -\frac{\hbar^2}{2m}\nabla^2\Psi(r,t) + V(r,t)\Psi(r,t).$$

La signification des termes est la suivante :

- $\hbar = \frac{h}{2\pi} = 1.05457.10^{-34}$, ou h est la constante de Planck dont la valeur est $6,62607015.10^{-34} m^2 kg/s$
- ∇^2 est le laplacien, aussi défini par $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$
- m est la masse de la particule ;
- $V(r,t)$ est l'énergie potentielle de la particule à la position r et à l'instant t ;

Hamiltoniens et équation stationnaire

Si l'on s'intéresse aux solutions *stationnaire* de l'équation de Schrödinger, donc indépendante du temps, on peut réduire celle-ci à la forme suivante

$$-\frac{\hbar^2}{2m}\nabla^2 + V(r)]\Psi(r) = E\Psi(r)$$

dans laquelle E est l'énergie de la particule. On désigne l'opérateur $-\frac{\hbar^2}{2m}\nabla^2 + V(r)$, sous le nom d'hamiltonien, noté H.
L'équation de Schrödinger stationnaire se ramène alors à la forme assez simple suivante :

$$H\Psi = E\Psi$$

D'un point de vue formel, résoudre cette équation revient mathématiquement à identifier les valeurs propres et les vecteurs propres de l'hamiltonien H.

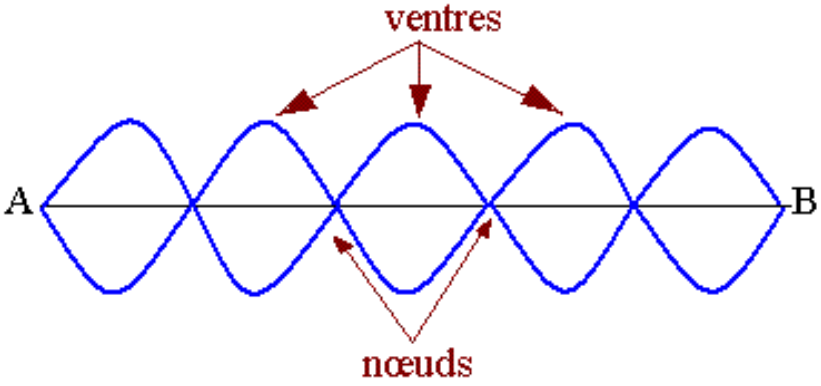
Ondes Stationnaires 1/2

Une onde stationnaire est une onde dont la forme ne dépend pas de l'espace, juste du temps. Par exemple, les harmoniques d'une corde de guitare, et dont les "nœuds" et les "ventres" sont fixes, forme une onde stationnaire. Du point de vue mathématique, on peut écrire que l'onde est le produit de deux facteurs, un qui dépend du temps et un autre qui dépend de l'espace. La fonction d'onde devient donc

$$\Phi(r,t) = \Psi(t)u(r)$$

La fonction $u(r)$ donne la forme en fonction de la position, soit l'amplitude de l'onde au point r, elle est en particulier nulle sur les nœuds de l'onde et maximale sur les ventres. C'est une fonction à valeurs dans $\mathbb{R}$ tandis que $\Phi(t)$ est à valeurs dans $\mathbb{C}$.

Ondes Stationnaires 2/2



Injection dans l'équation de Schrödinger

Si j'injecte le formalisme de l'onde stationnaire $\left|\Phi(r,t)\right\rangle = \left|\Psi(t)\right\rangle u(r)$ cette équation devient

$$i\hbar\frac{\partial}{\partial t}\left|\Psi(t)\right\rangle u(r) = -\frac{\hbar^2}{2m}\nabla^2\left|\Psi(t)\right\rangle u(r) + V(x,t)\left|\Psi(t)\right\rangle u(r).$$

Si j'introduis l'opérateur hamiltonien et que je simplifie les termes qui dépendent de l'espace pour ne garder que ceux qui dépendent du temps, on obtient

$$i\hbar\frac{\partial}{\partial t}\left|\Psi(t)\right\rangle = H\left|\Psi(t)\right\rangle.$$

273/369

Solutions stationnaires

Cette équation est à valeur dans $\mathbb{C}^2$, elle est toutefois analogue aux équations dans $\mathbb{R}$. En effet, si l'on considère l'équation suivante

$$\frac{df}{dt} = \alpha f(t)$$

Il est connu que la solution est de la forme

$$f(t) = C.e^{\alpha T} + D$$

L'équation de Schrödinger a la forme suivante, compte tenu de $\hbar = h/2\pi$,

$$\frac{d\left|\Psi(t)\right\rangle}{dt} = -i\frac{2\pi}{h}H\left|\Psi(t)\right\rangle$$

Dont la solution est de la forme

$$\left|\Psi(t)\right\rangle = e^{-\frac{2\pi}{h}H.t}\left|\Psi(0)\right\rangle$$

Opérateurs hermitiens et unitaires

L'opérateur H étant hermitien, iH est antihermitien et l'opérateur $U(t) = e^{-\frac{2\pi}{h}H.t}$ est unitaire. Ce point explique en particulier pourquoi l'informatique quantique ne s'intéresse qu'aux opérateurs unitaires.
Il est intéressant de s'intéresser aux valeurs propres et aux vecteurs propres de l'hamiltonien, soit les vecteurs $|\phi_n\rangle$ et les coefficients réels, homogènes à une énergie, E_n tels que

$$H|\phi_n\rangle = E_n|\phi_n\rangle$$

. En effet, Schrödinger est linéaire, donc les combinaisons linéaires de solutions sont des solutions. Si l'on s'intéresse aux valeurs propres et aux vecteurs propres, l'équation prend la forme

$$\frac{d\left|\phi_n(t)\right\rangle}{dt} = -i\frac{2\pi}{h}E\left|\phi_n(t)\right\rangle$$

Dont la solution est de la forme

$$\left|\phi_n(t)\right\rangle = e^{-\frac{2\pi}{h}E_n.t}\left|\phi_n(0)\right\rangle$$

Connaissant les vecteurs propres de H , les solutions sont

$$\left|\psi(t)\right\rangle = \sum_{i=1}^{n} e^{-\frac{2\pi}{h}E_i.t}\left|\phi_i(0)\right\rangle$$

Théorème adiabatique

En 1928, les physiciens Max Born et Vladimir Fock, énonce le théorème adiabatique :

> *Un système physique est maintenu dans son état propre instantané si une perturbation donnée agit sur lui suffisamment lentement et s'il y a un intervalle significatif entre la valeur propre et le reste du spectre de l'hamiltonien.*

En d'autres termes, si l'on sait placer un système quantique dans un état d'énergie de base connu, et qu'on le fait évoluer assez lentement et sans lui apporter d'énergie (c'est une évolution adiabatique) alors le système final, qui sera décrit par un nouvel hamiltonien, sera, lui aussi, dans un état d'énergie de base.

Calcul quantique adiabatique

Toute l'informatique quantique analogique s'appuie sur ce principe :

- on encode le problème à résoudre sous la forme d'un hamiltonien ;
- on part d'un système quantique dans un état de base connu avec un hamiltonien de référence connu ;
- on fait évoluer cet hamiltonien depuis celui de référence vers celui qui encode le problème, en respectant les contraintes du théorème adiabatique ;
- on dispose à la fin de l'état de base de l'hamiltonien de destination, celui qui encode le problème

Cette approche est très efficace, elle permet de trouver rapidement et de façon purement analogique, des minima de fonctions très complexes, dont la recherche avec des ordinateurs classiques relèvent de la nature *NP hard*.

Simulated Annealing

Le QUBO peut être résolu, par des méthodes HPC classiques, à l'aide de l'algorithme de *Simulated Annealing* ou "recuit simulé" créé en 1983.

Cet algorithme se base sur une idée empruntée à la métallurgie. Dans ce domaine, il est utile d'amener le métal à son niveau d'énergie le plus bas, où il devient ductile et est plus facile à travailler. Les forgerons réalisent cela en alternant des cycles comprenant des refroidissements lents et des étapes de réchauffage, ou *recuits*[1].

L'algorithme de recuit simulé suit une approche dite "métaheuriistique" qui dérive de l'algorithme de Metropolis-Hastings , issu de la modélisation des phénomènes thermodynamiques. De nombreuses implémentations existent, en particulier dans l'outil Mathematica. Il a souvent été utilisé pour résoudre des problèmes de graphes, en particulier ceux relatifs à la topologie de grands1réseaux informatiques.

1. par opposition, un refroidissement rapide, ou *trempe* va laisser le métat dans un état d'énergie élevé où il est dur mais peut casser, pour forger des couteaux par exemple

Quantum Annealing

Le *Quantum Anneling* est une implémentation matérielle, via des phénomènes quantiques, du QUBO.
La forme quadratique à résoudre est construite sous la forme d'un hamiltonien que l'on peut physiquement construire. On va chercher l'état de base de cet hamiltonien qui correspondra à la valeur minimale, au sens de QUBO, de la forme quadratique.

Dans cette approche adiabatique, on dispose d'un hamiltonien "simple" dont on connaît bien l'état de base, noté H_B et d'un hamiltonien H_P qui représente notre problème (un hamiltonien dont on cherche la plus petite valeur propre). On fait évoluer l'hamiltonien sur le temps T de la manière suivante

$$H(t) = (1 - s(t))H_B + s(t)H_P, s(t = 0) = 0, s(t = T) = 1$$

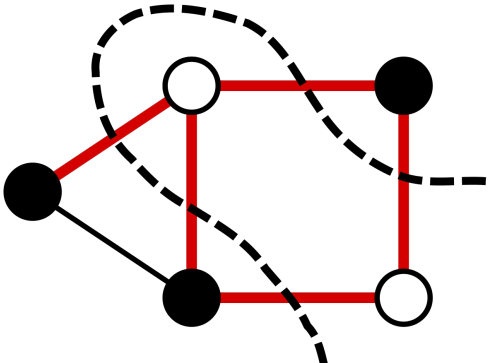
La fonction s est continue, monotone croissante.

279/369

Plan

Le problème Max-Cut 1/2

Étant donné un graphe, formé de sommets reliés par des arêtes, il s'agit de réaliser une coupe, donc de séparer le graphe en deux sous-ensembles complémentaires, tels que cette coupe au moins autant d'arêtes que n'importe quelle autre coupe possible.
Le problème Max-Cut est souvent associé à une notion de pondération des arêtes entre les sommets. On peut ainsi définir un "poids de la coupe" qui correspondra à la somme des poids des arêtes coupées. On dira qu'une arête est coupé quand les deux sommets, à chacune de ses extrémités, ne sont pas dans le même ensemble réalisé par la coupe.



Le problème Max-Cut 2/2

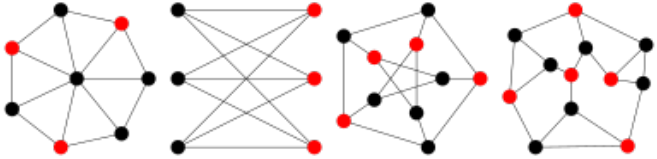
Remarque : d'una manière analogue, on peut définir un Min-Cut, une coupe mininum telle que le poids soit la plus faible valeur possible.
On peut associer deux types de problème à une coupe maximum :

- problème de *décision* : étant donné un graphe G et un entier k, existe-t'il une coupe de G dont le poids est au moins égal à k ;
- problème d'*optimisation* : étant donné un graphe G, quelle est la coupe maximum, celle qui maximise le poids ;

On peut démontrer que le problème Max-Cut se résout en un temps polynomial quand le graphe est planaire, il se ramène alors à l'identification des arêtes du graphe qui n'ont pas de sommets en commun. Un graphe est planaire s'il admet une représentation sagittale dans un plan sans que les arêtes se croisent.
Quand les graphes deviennent plus complexes, le problème de décision est NP-complet mais le problème d'optimisation est NP-dur.

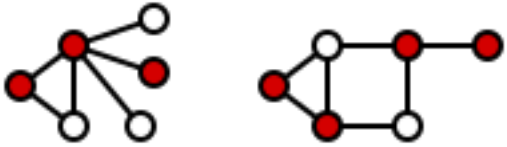
Le problème MIS

Étant donné un graphe G on peut définir un *ensemble indépendant* (ou *independant set*) par un sous-ensemble S de sommets de G tel qu'il n'existe aucune arête qui relit deux éléments de S.
Il existe souvent plusieurs solutions à ce problème.
Les MIS sont des *sous-ensembles dominants*. On dit qu'un sous-ensemble D d'un graphe G est dominant si chaque sommet de G est soit un élément de D, soit dispose d'un voisin dans D (donc il existe une arête qui le relie à un élément de D) s'il n'est pas dans D. Les MIS sont les plus grands sous-ensembles dominants.



Le problème MVC

Le problème MVC, acronyme de *Minimum Vertex Cover*, ou *problème de couverture par sommets*, est un problème dual du problème MIS.
Une couverture par sommets, aussi appelée *transversal* d'un graphe G est un ensemble C de sommets tel que chaque arête de $G = (V, E)$ est incidente à au moins un sommet de C, C'est à dire un sous-ensemble de sommets $S \subseteq V$ tel que pour chaque arête (u, v) de G on a $u \in S$ ou $v \in S$. On dit que l'ensemble C couvre les arêtes de G.



Le problème SAT 1/2

Le problème SAT ou le *problème de satisfaisabilité booléenne* est un problème de décision. Étant donné des variables booléennes (qui prennent les valeurs *vrai* ou *faux*) et une proposition, c'est-à-dire une formule qui combine ces variables avec des opérateurs booléens, on cherche à savoir s'il existe une combinaison de valeurs des variables qui rende vraie cette proposition .
Par exemple, la proposition $(p \wedge q) \vee \neg p$ est vraie pour toutes valeurs de q si p a la valeur *faux*, de même la proposition $(p \wedge \neg p)$ ne peut être satisfaite par aucune valeur de p.
Le problème SAT est l'archétype même des problèmes NP-complets. On peut identifier différentes formes simplifiées de SAT. vant d'aller plus loin, on doit définir ce qu'est une *Forme Normale Conjonctive*, ou CNF [2]. Une conjonction est une opération "AND", une forme normale conjonctive est une équation booléenne qui est une évaluation d'une succession de clauses qui ne contiennent pas de AND.

2. en anglais l'acronyme devient *Conjunctive Normal Form*

Le problème SAT 2/2

Une CNF est donc de la forme $(clause1) \wedge (clause2) \wedge \cdots \wedge (clauneN)$. Chaque clause est elle-même de la forme $a \vee \cdots b$ avec éventuellement des négations $\neg v$. Une CNF s'écrit donc

$$\bigwedge_{i=1}^{p}(\bigvee_{j=1}^{q} l_{ij}) \text{ avec } l_{ij} = a_{ij} \text{ ou } l_{ij} = \neg a_{ij}$$

Les simplifications les plus courantes du problème SAT sont :

- le problème **CNF-SAT** correspond au cas où la proposition est une CNF ;
- le problème **3-SAT** est une restriction de CNF-SAT où chaque clause comporte au plus 3 variables ;
- le problème **2-SAT** est une restriction de CNF-SAT où chaque clause comporte au plus 2 variables.

Le problème 2-SAT est de complexité P, mais 3-SAT est de difficulté NP.
On peut montrer que le problème SAT se ramène toujours au problème 3-SAT.
Le problème SAT est la base de la démonstration du théorème de Cook.

Le problème QUBO 1/2

L'acronyme *QUBO* signifie *Quadratic Unconstrained Binary Optimisation*. Il permet de résoudre des problèmes d'optimisation qui se ramènenent, dans les grandes lignes, à la recherche d'optima d'une forme quadratique.
Etant donnée un entier $n \in \mathbb{N}$, étant donné $\mathbb{B}^n = \{0; 1\}n$, l'ensemble des vecteurs de taille n formés de 0 et de 1, étant donné une forme quadratique f_Q représentée par une matrice $Q \in n\mathbb{R}^{n \times n}$, quel est la valeur $x^* \in \mathbb{B}^n$ qui minimise $f_Q(x) = x^T.Q.x$
Le problème QUBO possède différentes propriétés :

- multiplier Q par un facteur α ne change pas l'optimum ;
- inverse le signe de Q (mettre un moins devant) revient à chercher un maximum plutôt qu'un minimum ;
- si Q est une matrice diagonale, le problème est trivial également, le bit de rang i sera 0 si Q_{ii} est positif et 1 sinon ;

Le problème QUBO 2/2

D'une manière générale, on ajoute parfois au terme quadratique un terme linéaire, l'énoncé devient alors

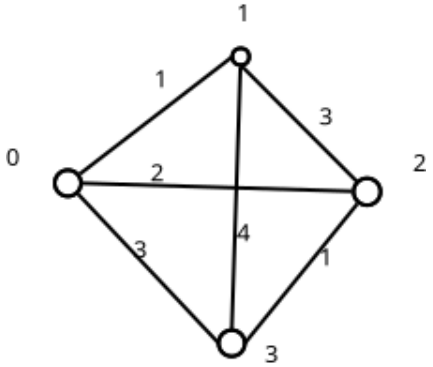
$$Q \in \mathbb{R}^{n \times n}, c \in \mathbb{R}^n, \text{ trouver la valeur } x^* \text{ qui minime } f(x) = x^T.Q.x + c^T.x$$

Le problème QUBO trouve des applications dans de nombreux domaines, car il est bien adapté à la recherche d'un optimum. Il intéresse ainsi les domaines de la finance, de l'économie, mais aussi la logistique etl'intelligence artificielle.
Le problème QUBO est de nature NPC

Plan

Résoudre MaxCut avec QUBO 1/3

Considérons le graphe suivant, il dispose de 4 noeuds, numérotés de 0 à 4, les arêtes portent différents poids allant de 1 à 4.
Ainsi,l'arête entre les noeuds 1 et 4 porte le poids 4 et celle entre 2 et 3 le poids 1.



Résoudre MaxCut avec QUBO 2/3

On peut traduire ce graphe par une "matrice de connectivité" W dont les coefficients w_{ij} sont

- 0 si $i = j$
- le poids de l'arête entre i et j si $i \neq j$

Dans notre exemple, on va construire la matrice suivante

$$W = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 4 \\ 2 & 3 & 0 & 1 \\ 3 & 4 & 1 & 0 \end{pmatrix}$$

Dans le cadre du problème MaxCut, on cherche une coupe maximale, donc un sous-ensemble E des nœuds du graphe qui maximise la coupe.
Connaissant E, on peut associer au noeud de rang i la valeur 1 s'il est dans E et 0 sinon. Cela nous permet de construire un vecteur x de valeur binaire. Inversement, un vecteur binaire $x \in \mathbb{B}^4$ décrit parfaitement une coupe.

Résoudre MaxCut avec QUBO 3/3

Considérons à présent la fonction de coût suivante qui représenté par un vecteur binaire x

$$x \in \mathbb{B}^4, C(x) = \sum_{i,j} W_{ij} x_i (1 - x_j)$$

- x_i est non nul si le noeud i est dans E
- $1 - x_j$ est non nul si le noeud i **n'est pas** dans E
- le terme $x_i(1 - x_j)$, qui est associé à l'arête entre i et j, sera non nulle si cette arête est dans la coupe, donc si elle relie deux points dans E et hors de F
- on fait donc la somme des poids W_{ij} des arêtes dans la coupe, les autres termes sont nuls.

$$c \in \mathbb{R}^4, c_i = \sum_j W_{ij} \text{ et } Q = -W, \forall i, j Q_{ij} = -W_{ij}$$

$$C(x) = \sum_{i,j} W_{ij} x_i (1 - x_j) = C(x) = -\sum_{i,j} W_{ij} x_i x_j) + \sum_i c_i . x_i = x^T . Q . x + c^T . x$$

ce qui est la formulation classique du QUBO

MIS et MVC se ramènent au QUBO 1/2

Le problème MVC, donc le problème MIS, peuvent s'exprimer sous la forme d'un QUBO.
Considérons MVC, pour définir une couverture par sommet, on va associer une valeur binaire x_i au sommet de rang i selon que le nœud est dans la couverture ou non.
Formellent, cela revient à minimiser la somme des x_i en garantissant que pour chaque arête de E, entre les noeuds i et j, au moins l'un des deux est dans la couvertures donc
$x_i + x_j \geq 1, \forall (i,j) \in E$
La méthode classiquement utilisée pour exprimer la contrainte consiste à ajouter un terme de "pénalité" qui va augmenter la fonction de coût dans les cas où la contrainte n'est pas réalisée.
Considérons le terme suivant

$$x \in \{0;1\}, y \in \{0;1\}, penalite(x,y) = 1 - x - y + xy$$

On peut identifier les cas suivants selon les valeurs de x et y

MIS et MVC se ramènent au QUBO 2/2

x	y	x+y	xy	1-x-y+xy	contrainte réalisée ?
0	0	0	0	1	NON
0	1	1	0	0	OUI
1	0	1	0	0	OUI
1	1	2	1	0	OUI

L'expression $1 - x - y + xy$ traduit donc la réalisation ou non de la contrainte $x + y \geq 1$. On va donc construire la fonction de coût suivante à minimiser

$$C(x) = \sum_i x_i + \sum_{i,j} (1 - x_i - x_j + x_i x_j)$$

Il est classique d'ajouter le paramètre P, un réel positif, pour accélérer la convergence

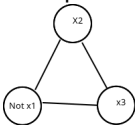
$$C_P(x) = \sum_i x_i + \sum_{i,j} P.(1 - x_i - x_j + x_i x_j) = (1 - 2P) \sum_i x_i + P \sum_{i,j} x_i.x_j$$

Le problème 3-SAT est un problème de type QUBO 1/2

Le problème 3-SAT, peut être résolu à l'aide d'un QUBO. La procédure consiste à traduire la formulation de 3-SAT en un roblème de graphe de type MIS qui peut être lui-même transformé en QUBO.
Considérons une CNF qui comprend n variables et m clauses.

- pour chaque clause, on construit un petit graphe à 3 sommets, chaque sommet étant une variable ou la négation d'une variable ;
- chacun de ses "sous-graphes"[3], est connecté en raccordant les nœuds qui représentent une variable et sa négation.

Par exemple, la clause $\neg x_1 \vee x_2 \vee x_3$ sera représentée par le graphe de la figure suivante

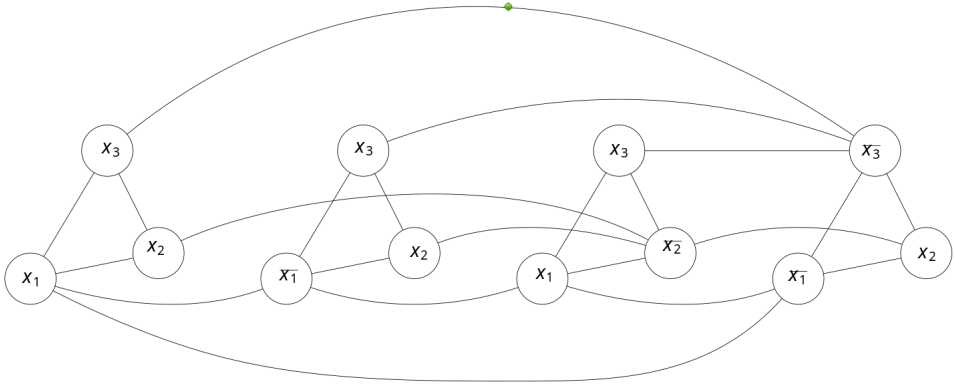


3. un tel sous-graphe est désigné sous le nom de *clique*

Le problème 3-SAT est un problème de type QUBO 2/2

On interconnecte ensuite les sous-graphes en reliant les variables et leurs négations.
Considérons la CNF suivante

$$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3)$$



Problème 3-SAT via QUBO/MIS

On peut démontrer que la résolution du problème MIS sur ce graphe permet de trouver une solution maximale de la clause correspondante. Si la taille de ce MIS est égale au nombre de clauses dans la CNF, alors celle-ci peut être satisfaite.

On rappelle que SAT veut savoir si une clause peut oui ou non être satisfait, mais ne cherche pas forcément une combinaison de variables booléennes qui satisfasse. SAT cherche juste à savoir si une telle solution existe.

Si la taille du MIS trouvée est inférieure au nombre de clauses, alors la CNF ne peut pas être satisfaite.

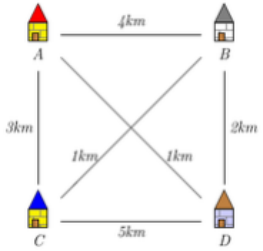
Le problème du voyageur de commerce est un QUBO 1/2

Le *TSP* ou *Traveling Salesperson Problem*, ou *problème du voyageur de commerce* est un problème très classique en recherche opérationnelle. Il s'agit d'optimiser la tournée d'un véhicule qui effectue sa tournée entre son dépôt et différents clients situés dans un réseau routier.

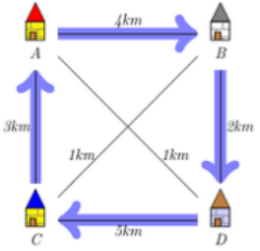
Etant donné n villes et les distances entre toutes les paires de villes, trouver un chemin de longueur totale minimale qui passe exactement une fois par chaque ville et revienne à la ville de départ.

On modélisera TSP comme un problème sur un graphe non orienté pondéré où les villes sont définies comme les sommets du graphe et où les chemins entre les villes, parcourues par le voyageur de commerce, sont les arêtes. Le coût d'une arête entre deux sommets est la distance entre les deux villes correspondantes.

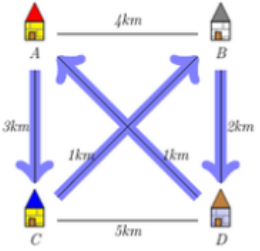
Le problème du voyageur de commerce est un QUBO 2/2



(a) Instance du problème



(b) Solution triviale et mauvaise



(c) Solution optimale

Fonction de coût de TSP/QUBO 1/3

Il est possible de formuler le TSP sous la forme d'un QUBO en considérant un ensemble de villes numérotées de 1 à N, on définira le coefficient x_{ij} tel que $x_{ij} = 1$ s'il existe une route entre la ville de rang i et la ville de rang j, sinon $x_{ij} = 0$.

Les coefficients x_{ij} permettent de définir un graphe dans lesquels les villes sont les sommets, on empruntera la route entre i et j si $x_{ij} = 1$. On associe un poids à l'arête ij qui correspond au temps de parcours entre les deux villes, cela permet la définition d'un coefficient de coût C_{ij}

Il existe différentes manières de formaliser TSP, voici l'une d'entre elles où résoudre TSP revient à minimiser la fonction binaire sur le vecteur x

$$C^A(x) = \sum_{i}^{N} \sum_{j=1, j\neq i}^{N} C_{ij} x_{ij} = 1$$

Fonction de coût de TSP/QUBO 2/3

On note qu'on évite les cas $i = j$, puisqu'un voyage entre deux villes implique forcément des villes différentes. On doit également ajouter les contraintes suivantes

$$\forall i \in \{1, 2, \cdots, N\} \sum_{j=1, j\neq i}^{N} x_{ij} = 1$$

qui traduit le fait qu'il n'existe, dans la tournée du voyageur de commerce, qu'un seul passage par la ville de rang i, on ici chercher à minimiser la fonction de coût suivante

$$\forall i \in \{1, 2, \cdots, N\}, C_i^B(x) = \sum_{j=1, j\neq i}^{N} (1 - x_{ij})^2$$

Fonction de coût de TSP/QUBO 3/3

Par ailleurs, on doit traduire le fait que pour tout ensemble de villes, il existe toujours au moins une route qui part de l'une d'entre elles pour aller dans une ville qui n'appartient pas à cet ensemble, soit

$$\forall S \subseteq \{1, 2, \cdots, N\}, S \neq \emptyset, \sum_{i \in S} \sum_{j \in \{1,2,\cdots,N\} \setminus S} x_{ij} \geq 1$$

Ce qui revient à minimiser la fonction

$$\forall S \subseteq \{1, 2, \cdots, N\}, S \neq \emptyset, C_S^C(x) = \sum_{i \in S} \sum_{j \in \{1,2,\cdots,N\} \setminus S} (x_{ij} - 1)$$

La fonction de coût de TSP est un QUBO

La fonction de coût va se construire la fonction de coût total en effectuant deux actions :

- réunir les x_{ij} sur une seule dimension, en mettant les "lignes" de la forme $x_{i1}x_{i2}\cdots x_{iN}$ les unes en dessous des autres et en effectuant le changement de variables correspondant dans les équations ci-dessus,
- en construisant une fonction de coût global comme la somme des fonctions de coût précédentes.

La fonction qui en résulte est de la forme

$$C^T(x) = \sum_{i,j=1}^{N} Q_{ij}x_i x_j + \sum_{i=1}^{N} C_i x_i = x^t.Q.x + C^t.x$$

qui correspond à la forme générique d'un problème QUBO.

QUBO et ses amis : conclusion partielle

Le problème QUBO semble au centre des algorithmiques présentées. Le fait qu'il soit "presque résoluble" par du HPCn depuis 1983 n'y est pas étranger.

D'autres problèmes NPC sont également intéressants, tels que MIS ou MWIS.

Passer d'un problème NP à un problème NPC est toujours de complexité P (théorème de Cook), mais trouver cette procédure est **très** compliqué.

QUBO est implémenté matériellement par les *Quantum Annealers* du constructeur **D-Wave**

On verra que le problème MIS est au coeur des ordinateurs analoique du constructeur **Pasqal**

Plan