

Projet : Optimisation de problèmes pseudo-booléens quadratiques (QUBO)



Master Informatique - 1ère année

année 2024-2025

1 Contexte

Les problèmes quadratiques à variables binaires (Unconstrained Binary Quadratic Problem - UBQP) sont des problèmes combinatoires génériques NP-difficiles qui modélisent des nombreuses applications [1]. Ces problèmes sont aussi les modèles d'entrée de nombreux algorithmes exécutés sur les nouveaux ordinateurs quantiques. Ainsi, de nombreuses recherches tentent de trouver la meilleure modélisation sous forme QUBO, ou tentent de développer des algorithmes d'optimisation sur machines classiques, sur machines quantiques ou de manière hybride (quantique/classique).

Le but du projet est de développer des algorithmes d'optimisation performants sur les problèmes QUBO en utilisant des techniques d'optimisation combinatoire et numérique.

1.1 Définition

Une instance du problème QUBO de dimension n est définie à partir d'un vecteur $H = (h_i)_{i \in \{1, \dots, n\}}$ et d'une (demi) matrice de nombres entiers (positifs ou négatifs) $Q = (q_{ij})_{i, j \in \{1, \dots, n\}}$. Seuls les nombres q_{ij} tels que $i < j$ sont non nuls. La fonction objective f à **minimiser** est définie de la manière suivante lorsque les variables binaires x_i appartiennent à $\{0, 1\}$:

$$\forall x \in \{0, 1\}^n, f(x) = \sum_{i=1}^n h_i (-1)^{x_i} + \sum_{i=1}^{n-1} \sum_{j=i+1}^n q_{ij} (-1)^{x_i} (-1)^{x_j}$$

À noter que lorsque $x_i = 0$, le terme $(-1)^{x_i} = 1$, et lorsque $x_i = 1$, le terme $(-1)^{x_i} = -1$.

1.2 Interprétation

Une interprétation possible de ce problème est d'imaginer que le but est de minimiser l'énergie totale d'un système "magnétique" où les variables binaires représentent le spin magnétique (Nord/Sud) de chaque élément du système. Un spin d'un élément peut être soit positif égale à $+1$, soit négatif égale à -1 . Par convention, lorsque $x_i = 0$ le spin est $+1$, et lorsque $x_i = 1$ le spin est -1 . L'énergie totale du système est la somme des énergies individuelles de chaque élément, et des énergies d'interaction entre chaque paire d'éléments. L'énergie individuelle de l'élément i , paramétrées par le nombre q_{ii} (positif ou négatif), est définie par $q_{ii}(-1)^{x_i}$. Si le spin est positif, l'énergie est ajoutée. Si le spin est négatif, l'énergie est soustraite. De la même façon, l'énergie d'interaction entre les éléments i et j est définie par $q_{ij}(-1)^{x_i}(-1)^{x_j}$. L'énergie q_{ij} est ajoutée lorsque les spins sont de même signe.

La figure 1 donne un exemple pour la dimension $n = 4$ d'un problème QUBO. Le vecteur H et la matrice Q correspondante sont égale à :

$$H = [2, -2, -2, 3] \quad Q = \begin{bmatrix} 0 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Les interactions entre les éléments 2 et 3 ($q_{23} = 0$) et les éléments 2 et 4 ($q_{24} = 0$) sont nulles. La solution représentée à gauche est la solution $x_1 = 1010$ ayant respectivement pour spins $-1, 1, 1, -1$. La fitness de cette solution est donc : $f(x_1) = -h_1 + h_2 - h_3 + h_4 - q_{12} + q_{13} - q_{14} - q_{34} = -2 - 2 + 2 + 3 + 1 + 1 + 1 - 2 = 2$. La solution représentée à droite est la solution $x_2 = 1001$ ayant respectivement pour spins $-1, 1, 1, -1$.

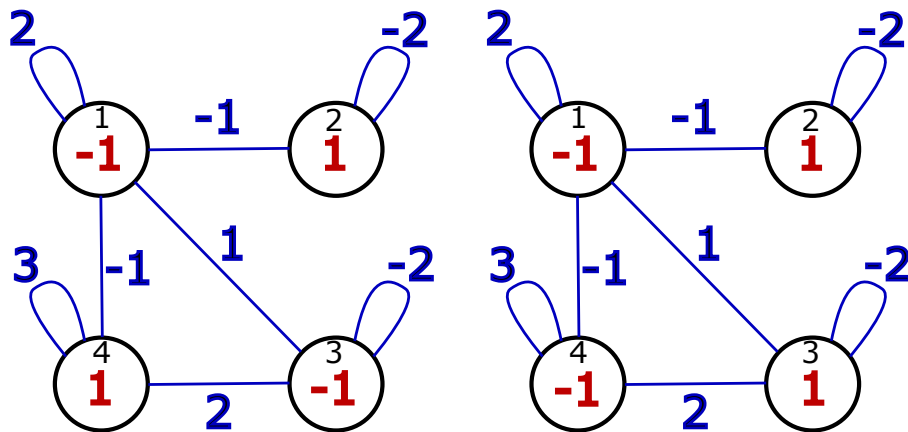


Figure 1: Exemple d'instance de problème QUBO de dimension $n = 4$. Les identifiants des éléments sont écrits en noir à l'intérieur des nœuds, les valeurs des énergies h_i et q_{ij} sont représentées en bleu sur les arêtes et la valeur des spins des éléments en rouge. À gauche, la solution $x_1 = 1010$ a pour fitness $f(x_1) = 2$. À droite le minimum global du problème avec la solution $x_2 = 1001$ ayant pour fitness $f(x_2) = -12$.

La fitness de cette solution est donc : $f(x_2) = -h_1 + h_2 + h_3 - h_4 - q_{12} - q_{13} + q_{14} - q_{34} = -2 - 2 - 2 - 3 + 1 - 1 - 1 - 2 = -12$. En énumérant tout l'espace de recherche, on peut montrer que la solution x_2 est l'unique minimum global de cette instance de problème QUBO.

2 Méthodologie

Les algorithmes seront testés et comparés sur les instances PUBO_i qui tiennent compte de l'importance des variables [3]. Le dépôt avec 1000 instances du problème et le code python et c++ de la fonction d'évaluation est disponible ici : <https://gitlab.com/verel/pubo-importance-benchmark/>.

Vous pouvez entraîner vos algorithmes à partir des 990 premières instances (de l'identifiant 0 à 989), et vous devez présenter les résultats (performances des algorithmes, etc.) sur les 10 dernières instances (de l'identifiant 990 à 999). Les algorithmes devront tous avoir le même temps de calcul, *i.e.* 30 secondes de calcul sur la machine "étalon".

Pour connaître l'équivalent du temps sur votre machine de la machine étalon, compiler le code fournis (voir gitlab : https://gitlab.dpt-info.univ-littoral.fr/m1-wedsci/projet_2025_m1_optimisation) et exécuter le. Le temps d'exécution sur la machine étalon est de 1 seconde. Vous pouvez alors calculer le ratio du temps d'exécution sur votre propre machine. Indiquer dans votre rapport, le temps d'exécution du code étalon sur votre machine. Les algorithmes doivent s'exécuter sur un temps égale à 30 secondes sur la machine étalon.

Les algorithmes qui par nature sont stochastiques, devront être testés au moins sur 30 exécutions.

3 Techniques d'optimisation combinatoire

Dans cette section, nous vous proposons de mettre au point des algorithmes d'optimisation combinatoire pour résoudre le problème d'optimisation.

Questions :

2.1 Initialisation :

- 2.1.a Coder une recherche aléatoire.
- 2.1.b Tester et commenter les résultats de votre recherche aléatoire.

2.2 Calcul incrémental :

- 2.2.a En utilisant la définition du problème QUBO, nous pouvons exprimer sous forme algébrique la variation de fitness : $\Delta_u f(x) := f(x \oplus u) - f(x)$ où $x \oplus u$ est la solution x dans laquelle la variable x_u est modifiée.

Montrez que :

$$\Delta_u f(x) = -2(-1)^{x_u} \left(\sum_{i=1}^{u-1} q_{iu}(-1)^{x_i} + h_u + \sum_{j=u+1}^n q_{uj}(-1)^{x_j} \right)$$

2.2.b Comparer la complexité de calculer directement $f(x \oplus u)$ et $\Delta_u f(x)$. Conclure.

2.2.c Nous pouvons également calculer sous forme algébrique la variation de la variation : $\Delta_{u,v}^{(2)} f(x) := \Delta_u f(x \oplus v) - \Delta_u f(x)$.

Montrez que :

$$\Delta_{u,v}^{(2)} f(x) = \begin{cases} 4q_{uv}(-1)^{x_u}(-1)^{x_v} & \text{si } u < v \\ 2\Delta_u f(x) & \text{si } u = v \\ 4q_{vu}(-1)^{x_u}(-1)^{x_v} & \text{si } v < u \end{cases}$$

2.2.d En utilisant la formule de 2.2.c, calculer la complexité du calcul de la mise à jour de la variation de fitness pour tous les voisins de x (à une distance de Hamming de 1) lorsqu'une variable x_j est modifiée. Comparer cette complexité à la recherche du meilleur voisin.

2.3 Recherches locales :

2.3.a Coder un algorithme de type Iterated Local Search (ILS).

2.3.b Coder un algorithme de type Tabu Search (TS).

2.3.c Comparer les performances de vos algorithmes ILS et TS.

2.4 Pour aller plus loin (optionnel) :

2.4.a Proposer et coder votre propre algorithme (algorithme évolutionnaire, opérateurs différents, etc.)

2.4.b Comparer les performances avec les algorithmes ILS et TS.

4 Techniques d'optimisation numérique

Dans cette section, nous vous proposons de mettre au point un algorithme d'optimisation numérique. En effet, il est possible de considérer le même problème avec des variables réelles (flottants) entre $[-1, 1]$ au lieu des variables binaires $\{0, 1\}$. Cette technique est récemment mise en œuvre pour le problème SAT [2]. Ici, $z_i = -1$ correspond à la valeur binaire $x_i = 1$, et $z_i = 1$ correspond à la valeur $x_i = 0$. La fonction dépendant des variables réelles z_i est définie par:

$$\forall z \in [-1, 1]^n, f(z) = \sum_{i=1}^n h_i z_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n q_{ij} z_i z_j$$

Ainsi, les valeurs de la fonction QUBO sont identiques pour les deux encodages lorsque $z_i = (-1)^{x_i}$.

Le principe algorithmique utilise une descente de gradient pour minimiser f avec les variables réelles de $[-1, 1]^n$. Si après un pas de gradient, une variable z_i est en dehors de $[-1, 1]$, alors z_i est projeté de nouveau sur $[-1, 1]$ par $z_i = \text{sign}(z_i) \in \{-1, 1\}$. On pourrait imaginer d'autres techniques pour conserver les variables dans l'intervalle $[-1, 1]$ (paramétrisation, pénalité, etc.).

À la fin de l'algorithme de gradient, les variables sont transformées en variable binaire par $x_i = 0$ si $z_i \geq 0$, et $x_i = 1$ si $z_i < 0$, donnant une solution au problème combinatoire QUBO original.

Le gradient de la fonction f est donné par les dérivées partielles suivantes:

$$\forall u \in \{1, \dots, n\}, \frac{\partial f}{\partial z_u}(z) = \sum_{i=1}^{u-1} q_{iu} z_i + h_u + \sum_{j=u+1}^n q_{uj} z_j$$

Questions :

3.1 Descente de gradient :

3.1.a Coder une descente de gradient pour minimiser f qui projette la solution finale sur $\{0, 1\}^n$.

3.2.b Comparer les performances de la descente de gradient avec les algorithmes combinatoires.

3.2 Algorithme hybride :

3.2.a Imaginer un algorithme qui combine un algorithme combinatoire avec une descente de gradient.

3.2.b Analyser et comparer les performances de votre proposition.

A Meilleure valeurs connues

Les valeurs de la fonction objective des meilleures solutions connues pour chaque instances sont :

id	best known
990	-70459
991	-102182
992	-112475
993	-96038
994	-41790
995	-72707
996	-115145
997	-187096
998	-87290
999	-101785

References

- [1] Gary Kochenberger, Jin-Kao Hao, Fred Glover, Mark Lewis, Zhipeng Lü, Haibo Wang, and Yang Wang. The unconstrained binary quadratic programming problem: a survey. *Journal of combinatorial optimization*, 28:58–81, 2014.
- [2] Anastasios Kyrillidis, Anshumali Shrivastava, Moshe Vardi, and Zhiwei Zhang. Fouriersat: A fourier expansion-based algebraic framework for solving hybrid boolean constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1552–1560, 2020.
- [3] Sara Tari, Sébastien Verel, and Mahmoud Omidvar. Puboi: A tunable benchmark with variable importance. In *European Conference on Evolutionary Computation in Combinatorial Optimization (Part of EvoStar)*, pages 175–190. Springer, 2022.