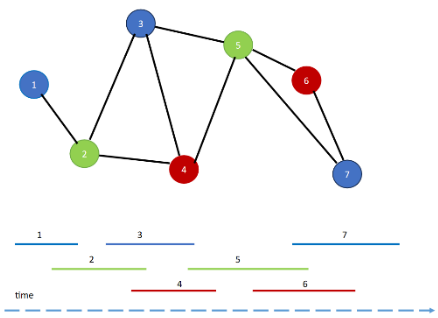
# Une génération de colonne assistée par machine quantique pour le problème de conversion de flotte de véhicules.

Eric Bourreau[1], Yagnik Chatterjee[2], Zaid Allybokus[2], Marko Rančiç[2]
[1] LIRMM, Université de Montpellier, CNRS, 161 rue Ada, 34392 Montpellier Cedex 5
[2] TotalEnergies, Tour Coupole - 2 place Jean Millier 92078, Paris la Défense cedex, France

## 1 Introduction

La situation climatique impose l'adaptation nécessaire des grandes institutions aux recommandations du GIEC pour lutter contre le changement climatique. L'introduction de véhicules électriques ou hybrides dans les flottes d'entreprise, favorisant le covoiturage, comme solution pour réduire les émissions de gaz à effet de serre et optimiser les déplacements s'appelle "la conversion de flotte de véhicules". Optimiser cela consiste à recueillir les besoins de déplacements des employés (origine, destination, horaire, volume) et planifier des tournées compatibles entre les différents trajets, en sélectionnant le meilleur véhicule avec la meilleure capacité tout en minimisant la taille de cette flotte.

Ce problème peut se modéliser comme un problème de coloration dans un graphe d'incompatibilité spatio-temporel. Comme illustré sur la figure ci-contre, chaque nœud représente un trajet et chaque arête une incompatibilité spatiale, temporelle ou capacitaire. Nous cherchons à minimiser le coût d'acquisition de cette flotte.




Pour résoudre ce problème nous proposons de réaliser une décomposition en génération de colonnes où le sous problème consiste à générer un tour, c'est-à-dire trouver un stable de poids minimum dans ce graphe d'incompatibilité.

Depuis quelques années des machines quantiques sont disponibles. Nous proposons de résoudre les sous-problèmes via un algorithme variationnel quantique pour découvrir les nouvelles colonnes en s'hybridant avec un solveur linéaire sur machine classique pour le problème maître.

## 2 Description formelle du problème de conversion de flotte

Soit *G*=(*K*,*E*) un graphe où les nœuds représentent des trajets et où les arêtes *(k,k')* représentent l'incompatibilité entre les trajets *k* et *k'*. Notre problème de coloration minimale peut se décrire par le modèle suivant. Soit $x_c^k$ le booléen représentant le fait d'affecter la couleur *c* au trajet *k* et $y_c$ le booleen d'activation du véhicule *c*, nous avons le modèle suivant où $\gamma_c$ est le coût d'acquisition d'un véhicule, et $\Gamma_c^k$ le coût d'utilisation d'un véhicule *c* par l'utilisateur *k* durant son trajet. Soit $A^k$ la liste des véhicules autorisés pour un trajet *k* et *v(c)* le véhicule de couleur *c*.

$$
\begin{aligned}
\min \; & \sum_{c \in C} \gamma_c y_c + \sum_{k \in K, c \in C} \Gamma_c^k x_c^k & \\
\text{s.t.} \; & \sum_c x_c^k \geq 1 & (\forall k \in K) \\
& x_c^k \leq y_c & (\forall c \in C)(\forall k \in K) \\
& x_c^k + x_c^{k'} \leq 1 & (\forall \{k,k'\} \in \mathcal{I})(\forall c \in C) \\
& x_c^k = 0 & (\forall k \in K)(\forall c \in C)(v(c) \notin A^k) \\
& y_c, x_c^k \in \{0,1\}. &
\end{aligned}
$$

Il est possible de décomposer ce problème de coloration comme une couverture des nœuds par des stables. Un stable *I* est un sous ensemble *I* ⊆ *E* tel que pour tout couple k,k' ∈ *I*, *(k,k')* ∉ *E*. Soit $a_I^k$ la composition du stable *I* sur le graphe *G* et $\Gamma_I$ le coût de ce stable.

Nous avons un problème maître restreint ainsi que son dual (après reformulation) :

$$\min \sum_{I\in\Lambda} \Gamma_I x_I$$
$$\text{s.t.} \sum_{I\in\Lambda} a_I^k x_I \geq 1 \qquad (\forall k \in K)$$

$$\max \sigma = \sum_k y_k(\mu'_k - \Gamma_c^k)$$
$$\text{s.t. } y_k + y_j \leq 1 \qquad (\forall (k,j) \in E)$$
$$y_k \in \{0,1\} \qquad (\forall k \in K)$$

Attention, notez que $y_c$ défini sur le modèle initial n'est pas la même variable que $y_k$ dans le dual.

## 3 Génération de colonnes assistée par machine quantique

Le problème de pricing peut se réécrire sous forme quadratique comme un QUBO. Soit $w_k = {\mu'}_k - \Gamma_c^k$ la fonction objective *z* s'écrit simplement en ajoutant des contraintes d'incompatibilités pénalisées, $z = \sum_k y_k w_k - P\sum_{(i,j)\in E}(y_i y_j)$ qui s'exprime donc sous forme quadratique $z = yQy^T$ car $y_k^2 = y_k$.

Ce QUBO peut être représenté sous la forme d'un hamiltonien et son coût sera mesuré par $H(\theta) = -\langle\psi(\theta)|\,Q|\,\psi(\theta)\rangle$ où $\psi(\theta)$ est un ansatz représentant le vecteur $y_k$.

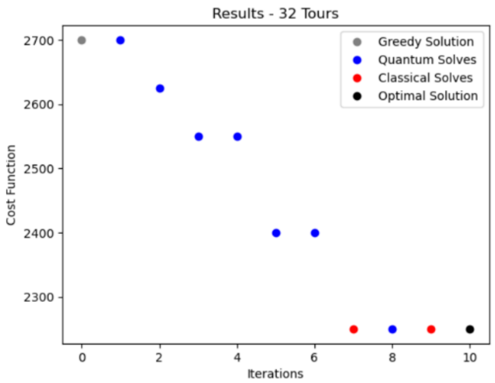
Soit $|\psi(\theta)\rangle = U(\theta)Had^{\otimes n}|0\rangle^{\otimes n}$avec

$$U(\theta) = \begin{bmatrix} e^{i\pi R(\theta_1)} & 0 & 0 & \dots \\ 0 & e^{i\pi R(\theta_2)} & 0 & \dots \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & e^{i\pi R(\theta_n)} \end{bmatrix} \quad \text{où} \quad R(\theta_k) = \begin{cases} 0 & \text{if } 0 \leq \theta_k < \pi \\ 1 & \text{if } \pi \leq \theta_k < 2\pi \end{cases}$$

La partie intéressante de cette représentation de $Q$ (une matrice *n* x *n*) et $\psi(\theta)$ (un vecteur de taille *n*) réside dans le fait que nous pouvons n'utiliser que *log(n)* qbits pour les coder sur une machine quantique (algorithme[1] présenté lors de la conférence ROADEF 2023). Nous avons utilisé un algorithme génétique pour optimiser les paramètres $\theta$ ($\theta^*$) afin de minimiser $H^*(\theta^*) = min\, H(\theta)$.

## 4 Résultats numériques

Des instances de taille 32, 64 et 128 nœuds, nécessitant seulement 5, 6 ou 7 qbits, ont été générées. Nous commençons par un glouton puis appelons l'algorithme quantique pour générer les colonnes améliorantes, alimentées par le problème maître résolu avec un solveur PLNE. Comme l'algorithme variationnel fourni des solutions approchées, nous avons ajouté un modèle de pricing « classique » avec un solveur linéaire pour la preuve d'optimalité.




Au final, cette résolution du problème de conversion de flottes par une génération de colonnes associée à un algorithme quantique peut résoudre des instances de taille réaliste (plus de 100 nœuds).

---

[1] Y. Chatterjee, E. Bourreau, and M. J. Rančić, Solving various np-hard problems using exponentially fewer qubits on a quantum computer, arXiv preprint arXiv:2301.06978 (2023)