

Titre: Résolution de problèmes d'optimisation combinatoire par des
Title: approches d'inspiration quantique

Auteur: Ali M'Hammedi Alaoui
Author:

Date: 2023

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: M'Hammedi Alaoui, A. (2023). Résolution de problèmes d'optimisation
Citation: combinatoire par des approches d'inspiration quantique [Mémoire de maîtrise,
Polytechnique Montréal]. PolyPublie. <https://publications.polymtl.ca/57039/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/57039/>
PolyPublie URL:

**Directeurs de
recherche:** Tarek Ould-Bachir
Advisors:

Programme: Génie informatique
Program:

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

**Résolution de problèmes d'optimisation combinatoire
par des approches d'inspiration quantique**

ALI M'HAMMEDI ALAOUI

Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie informatique

Décembre 2023

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

**Résolution de problèmes d'optimisation combinatoire
par des approches d'inspiration quantique**

présenté par **Ali M'HAMMEDI ALAOUI**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Nora BOULAHIA CUPPENS, présidente

Tarek OULD-BACHIR, membre et directeur de recherche

Quentin CAPPART, membre

DÉDICACE

À mes parents

Quoi que je dise, je ne saurais point vous remercier comme il se doit. Votre affection me couvre, votre bienveillance me guide et votre présence à mes côtés a toujours été ma source de force pour affronter tous les obstacles.

À ma chère sœur

En signe d'amour, et de gratitude pour tout le soutien, les encouragements et les conseils que tu m'as prodigués tout au long de ma vie. Je te souhaite un avenir radieux et plein de bonheur.

À toute ma famille et tous mes amis

Avec qui j'ai partagé des moments magnifiques de ma vie. Merci de m'écouter et de m'épauler quand j'en ai eu besoin.

A tous ceux que j'aime et qui m'aiment

REMERCIEMENTS

Au terme de ce projet de fin de mémoire de maîtrise recherche, je remercie toute personne ayant contribué de loin ou de près à la réalisation de ce projet dans de bonnes conditions.

Je tiens à remercier le professeur Tarek Ould-Bachir pour son encadrement et assistance pédagogique et sa participation à l'acheminement de ce rapport. Sa supervision et son expertise ont été d'une importance capitale pour mener à bien ce travail de recherche. Je tiens également à le remercier chaleureusement pour sa contribution essentielle à l'acheminement de ce mémoire de maîtrise, qui a pu être réalisé grâce à son soutien constant.

Je souhaite également exprimer ma reconnaissance envers les membres de l'équipe InfinityQ. Leur soutien inestimable et leur collaboration ont été d'une valeur inestimable tout au long de ce projet. En particulier, je tiens à remercier Nakita Rao et Aurélie Hélouis pour leur implication, leur expertise et leur disponibilité. Leur contribution significative a permis d'atteindre les objectifs fixés et a été une source d'inspiration pour moi tout au long de ce parcours.

Je tiens également à exprimer ma gratitude envers MITACS et InfinityQ pour leur soutien financier. Leur investissement dans ce projet a joué un rôle crucial dans sa réalisation, en offrant les ressources nécessaires pour mener à bien les différentes étapes de recherche. Leur confiance en mes capacités a été un moteur supplémentaire pour persévérer et atteindre les résultats obtenus.

Enfin, je souhaite remercier tous mes proches, amis et collègues qui m'ont soutenu tout au long de cette aventure. Leurs encouragements, conseils et soutiens moral ont été essentiels pour me maintenir motivé et concentré sur mes objectifs. Leur présence bienveillante a été un véritable soutien tout au long de ce parcours exigeant.

RÉSUMÉ

Ce mémoire de thèse propose d'évaluer la résolution de problèmes d'optimisation combinatoire NP-difficile par des approches d'inspiration quantique. Au cœur de cette recherche se trouve l'examen du modèle de Ising, une représentation mathématique fondamentale dans la mécanique statistique et la physique quantique. L'objectif est d'explorer comment les principes de ce modèle de la physique statistique peuvent être exploités pour résoudre efficacement des problèmes complexes, dont la complexité croissante dépasse les capacités des ordinateurs classiques. Ce mémoire contribue ainsi à l'élargissement des horizons de la résolution de problèmes d'optimisation, ouvrant la voie à des applications potentielles dans divers domaines, de la logistique à la conception de médicaments.

Les problèmes d'optimisation combinatoires NP-difficile sont nombreux. Dans le cadre de ce mémoire, le point se fera sur les deux problèmes à savoir le TSP et le JSSP utilisés beaucoup en industrie. Le mémoire se concentrera ainsi sur la résolution de ces deux problèmes par des approches d'inspiration quantique. Le recuit simulé, la machine de D-Wave et la machine à bifurcation de Toshiba en font partie. Le mémoire compare aussi les résultats avec d'autres approches dites classiques comme Google-OR et Concorde.

Les résultats obtenus par le biais des outils cités précédemment sont de mauvaise qualité comparé aux outils classiques de résolution. Le mémoire vise donc à proposer des méthodes pour améliorer les performances en termes de temps et de qualité de solutions. On a montré qu'avec ces méthodes et approches, les résultats sont allés de mauvais à résultats convaincant. On guise de conclusion, le modèle d'Ising est une approche révolutionnaire pour contrer les problèmes difficiles, mais celui-ci requiert des méthodes de résolution efficaces.

ABSTRACT

This thesis aims to assess the resolution of NP-hard combinatorial optimization problems through quantum-inspired approaches. At the core of this research lies the examination of the Ising model, a fundamental mathematical representation in statistical mechanics and quantum physics. The objective is to explore how the principles of this statistical physics model can be leveraged to efficiently solve complex problems, whose increasing complexity surpasses the capabilities of classical computers. This thesis thus contributes to expanding the horizons of optimization problem-solving, paving the way for potential applications in various fields, from logistics to drug design.

Combinatorial optimization problems that fall under the NP-hard category are numerous. In the context of this thesis, the focus will be on two specific problems, namely the Traveling Salesman Problem (TSP) and the Job Shop Scheduling Problem (JSSP), both widely used in the industry. The thesis will primarily explore the resolution of these two problems using quantum-inspired approaches, including simulated annealing, D-Wave quantum machines, and Toshiba's bifurcation machine. Additionally, the thesis will compare the results obtained from these quantum-inspired methods with those achieved using classical approaches such as Google-OR and Concorde.

The results obtained through the aforementioned tools have demonstrated lower quality compared to classical solving methods. Therefore, the thesis aims to propose methods to enhance performance in terms of both solution quality and execution time. It has been shown that, with these novel methods and approaches, results have improved from poor to convincing. In conclusion, the Ising model represents a revolutionary approach to tackle challenging problems; however, it requires efficient solving methods to fully harness its potential.

TABLE DES MATIÈRES

DÉDICACE	iii
REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vi
TABLE DES MATIÈRES	vii
LISTE DES TABLEAUX	x
LISTE DES FIGURES	xi
LISTE DES SIGLES ET ABRÉVIATIONS	xii
CHAPITRE 1 INTRODUCTION	1
1.1 Au-delà de la loi de Moore	1
1.2 Machines d’inspiration quantique	2
1.3 Objectifs de recherche	3
1.4 Organisation du mémoire	3
CHAPITRE 2 REVUE DE LITTÉRATURE	5
2.1 Introduction	5
2.2 Approches d’inspiration quantique	5
2.2.1 Le modèle de Ising	6
2.2.2 Modèles apparentés au modèle de Ising	7
2.2.3 Comparaison des modèles	9
2.3 Méthodes de recuit simulé	11
2.3.1 Formulation générale	11
2.3.2 Application du recuit simulé au modèle de Ising	11
2.3.3 Recuit quantique	12
2.4 Machines de Ising	12
2.4.1 Recuit quantique matériel	13
2.4.2 Recuit numérique matériel	14
2.4.3 Bifurcation adiabatique	15

2.4.4	Machine de Ising cohérente	15
2.4.5	Machine de Ising à base d'oscillateurs analogiques	17
2.4.6	Machine de Ising creuse	17
2.4.7	Bits probabilistes	18
2.5	Application à des problèmes d'optimisation combinatoires	18
2.5.1	Problèmes considérés dans ce travail	18
2.5.2	Évaluations de performance des machines de Ising	19
2.5.3	Résolution du TSP et du JSSP par modèle de Ising	19
2.5.4	Outils et méthodes de résolution pour TSP	20
2.5.5	Outils de résolution pour JSSP	24
CHAPITRE 3	PROBLÈME DU VOYAGEUR DE COMMERCE	25
3.1	Définition du problème	25
3.2	Un problème factoriel	25
3.3	Formulation du problème	26
3.3.1	Formulation générale	27
3.3.2	La formulation de MTZ	28
3.3.3	La formulation de DFJ	28
3.3.4	Formulation QUBO	29
3.4	Implémentations avec des approches d'inspiration quantique	31
3.4.1	Implémentation du SA	31
3.4.2	Implémentation sur la plateforme de D-Wave	32
3.4.3	Implémentation sur la plateforme de Toshiba	34
3.5	Expériences, résultats et discussions	34
3.5.1	Résultats sur D-Wave et Toshiba	34
3.5.2	Recuit simulé du modèle de Ising	36
3.5.3	Impact du choix du voisinage pour SA	37
3.5.4	Comparaison de SA avec Concorde	37
3.5.5	Discussion générale	40
CHAPITRE 4	PROBLÈME DE SÉQUENÇAGE DES TÂCHES	42
4.1	Définition du problème	42
4.2	Formulation du problème	43
4.2.1	Formulation générale	43
4.2.2	Formulation QUBO	44
4.3	Implémentation avec des approches d'inspiration quantique	47
4.3.1	Jeu de données	47

4.3.2	Exécution sur D-Wave et Toshiba	48
4.4	Recuit simulé du modèle de Ising	49
4.5	Discussion générale	53
CHAPITRE 5	CONCLUSION	54
5.1	Retour synthétique sur les travaux de maîtrise	54
5.2	Limitations de la recherche	54
5.3	Améliorations futures	54
RÉFÉRENCES		56

LISTE DES TABLEAUX

Tableau 2.1	Récapitulatif des modèles	10
Tableau 3.1	Temps de calcul du TSP	26
Tableau 4.1	Exemple de problème de JSSP	42
Tableau 4.2	Jeu de données du JSSP	48

LISTE DES FIGURES

Figure 2.1	Graphe de chimère 3×3	13
Figure 2.2	Étapes de l'algorithme de Christofides	23
Figure 2.3	Exemple d'un k-opt, $k=2$	23
Figure 2.4	Exemple d'un k-opt, $k=3$	23
Figure 3.1	Exemple de solution pour le TSP	25
Figure 3.2	Évolution du temps d'exécution avec Gurobi	26
Figure 3.3	Évolution du temps d'exécution avec Concorde	27
Figure 3.4	Exemple d'élimination de sous-tournées	29
Figure 3.5	Évolution de la probabilité de succès de la solution de D-Wave Hybride	35
Figure 3.6	Distribution des solutions en fonctions du nombre de villes	35
Figure 3.7	Comparaison entre les différents solveurs	36
Figure 3.8	Évolution de l'énergie en fonction du temps pour l'instance à 50 villes et la fonction bit_flip	37
Figure 3.9	Évolution de l'énergie en fonction du temps pour les deux options sug- gérées	38
Figure 3.10	Évolution de l'énergie en fonction du temps pour l'instance à 100 villes et la fonction 2-opt	38
Figure 3.11	Comparaison de la solution entre le SA et l'optimal de Concorde en num_simulations = 1 et timeout = 10 s	39
Figure 3.12	Comparaison de la solution entre le SA et l'optimal de Concorde en num_simulations = 1 et timeout = 30 s	40
Figure 3.13	Comparaison de la solution entre le SA et l'optimal de Concorde en num_simulations = 1 et timeout = 60 s	41
Figure 4.1	Exemple de solution du problème JSSP	42
Figure 4.2	Comparaison entre les différents solveurs D-Wave, SA et Toshiba . . .	48
Figure 4.3	Évolution de l'énergie par rapport au temps pour jssp_5 avec un bit flip	49
Figure 4.4	Évolution de l'énergie par rapport au temps pour jssp_5 avec élimina- tion de variables	50
Figure 4.5	Solution avec flip aléatoire	51
Figure 4.6	Solution avec élimination de variables	51
Figure 4.7	Évolution de l'énergie par rapport au temps pour jssp_6 avec bit flip	52
Figure 4.8	Évolution de l'énergie par rapport au temps jssp_6 avec élimination de variables	52

LISTE DES SIGLES ET ABRÉVIATIONS

API	Application Programming Interface
ASIC	Application-specific Integrated Circuit
BQM	Binary quadratic model
aSB	Adiabatic Simulated Bifurcation
CP	Constraint Programming
CIM	Coherent Ising Machine
CPU	Central Processing Unit
DA	Digital Annealer
DFJ	Dantzig-Fulkerson-Johnson
DOPO	Degenerate Optical Parametric Oscillator
FPGA	Field-Programmable Gate Arrays
GPU	Graphical Processing Unit
IP	Integer Programming
JSSP	Job-Shop Scheduling Problem
LP	Linear Programming
MCMC	Markov Chain Monte-Carlo
MQMC	Modèle Quantique de Markov Caché
MTZ	Miller-Tucker-Zemlin
OIM	Oscillator Ising machine
OEPO	Optoelectronic parametric oscillator
PC	Puissance de Calcul
QA	Quantum Annealing
QUBO	Quadratic Unconstrained Binary Optimization
SA	Simulated Annealing
SB	Simulated Bifurcation
SHIL	Subharmonic Injection Locking
sIM	sparse Ising Machine
SNM	Stochastic Nanomagnet
SQBM	Simulated Quantum Bifurcation Machine
TPU	Tensor Processing Unit
TSP	Traveling Salesman Problem
VRP	Vehicle Routing Problem

CHAPITRE 1 INTRODUCTION

1.1 Au-delà de la loi de Moore

L'avènement de la société de l'information à la fin du XX^e siècle a été un bouleversement majeur. En rendant l'information et la communication plus facilement accessibles, les ordinateurs ont révolutionné de nombreux secteurs tels que la communication, le divertissement, l'éducation ou la santé. Notamment, les ordinateurs ont augmenté la vitesse et la précision de nombreuses tâches, permettant d'accroître la productivité et la croissance économique. Ces changements majeurs ont été rendus possibles grâce à la loi de Moore. Proposée en 1965 par Gordon Moore, co-fondateur d'Intel, cette loi stipule que le nombre de transistors sur une puce informatique double tous les 18 à 24 mois, entraînant une augmentation correspondante de la puissance de calcul (PC) et des performances des ordinateurs [1]. Cette prédiction s'est avérée exacte et a eu un impact significatif sur nos sociétés.

Les besoins en puissance de calcul ne semblent pas reculer dans un avenir proche, bien au contraire, et ce notamment du fait de facteurs tels que le développement de nombreuses nouvelles technologies, le nombre croissant d'appareils connectés et la demande grandissante de traitement et d'analyse des données. Par exemple, l'utilisation de l'intelligence artificielle et des technologies d'apprentissage automatique augmentant, les besoins en ressources informatiques puissantes pour prendre en charge ces applications augmenteront également. Cependant, un ralentissement notable de la loi de Moore a été remarqué durant la récente décennie, avec une croissance de la PC de seulement 2% par an durant les plus récentes années, par opposition à des taux supérieurs à 50% durant les années fastes [1]. La fin complète de la loi de Moore entraînerait un ralentissement majeur de l'innovation, avec des répercussions économiques et sociétales considérables (réduction de l'activité économique, pertes d'emplois, crises sociales, etc.).

Dans un article resté célèbre, Hennessey et Patterson ont suggéré que le ralentissement de la croissance de la PC des ordinateurs pourrait être contrecarré par l'exploitation d'architectures dédiées [2]. Les architectures dédiées sont une nouvelle façon de construire des ordinateurs permettant d'optimiser les performances de calcul pour des tâches spécifiques, et ce en utilisant des circuits intégrés spécialisés et des systèmes hybrides. Dans les dernières décennies, plusieurs de ces architectures ont été développées et mises en place dans la plupart des industries les aidant à mener à bien leurs projets de manière efficace et consistante. Un exemple célèbre est l'unité de traitement de tenseur (*tensor processing unit*: TPU) développé par Google Inc. et dédié à l'accélération de l'inférence dans les réseaux de neurones d'un

facteur de 10 [2]. D'autres avenues de recherche suggèrent que la solution pourrait provenir des méthodes de calcul non conventionnelles. Les méthodes de calcul non conventionnelles désignent des approches alternatives pour résoudre les problèmes mathématiques ou informatique. On peut citer par exemple des méthodes chimiques comme le calcul moléculaire, des méthodes biochimiques comme le calcul par DNA, des méthodes probabilistes comme le calcul stochastique, ou encore des méthodes physiques telles que le calcul optique et le calcul quantique [3].

Les approches quantiques suscitent actuellement beaucoup d'intérêt, tant dans le milieu académique qu'industriel. Cependant, il existe aujourd'hui beaucoup de limitations aux ordinateurs quantiques (OQ). D'une part, la réalisation physique des OQ est un défi technique important, limitant le nombre de qubits pouvant être offerts. Par exemple, les qubits doivent être maintenus à des températures extrêmement basses et il faut les protéger des perturbations extérieures, ce qui rend leur réalisation complexe et coûteuse [4]. D'autre part, il existe un manque dans le développement des algorithmes quantiques. En effet, peu d'algorithmes quantiques ont été développés. Les auteurs citent souvent l'algorithme de Grover et celui de Shor [5], mais ces derniers se limitent à certains types de problèmes, et un manque de diversité reste patent. Des méthodes dites d'inspiration quantique pourraient en revanche combler un besoin au niveau des problèmes d'optimisation combinatoire, comme en témoignent les offres commerciales de D-Wave, Toshiba ou Fujitsu [6]. Ce mémoire propose justement d'étudier le potentiel de ces technologies, d'en déterminer les limitations et de proposer des solutions pour en améliorer les performances dans la résolution de certains problèmes classiques tels que le problème du voyageur de commerce et le problème du séquençage des tâches.

1.2 Machines d'inspiration quantique

Les méthodes d'inspiration quantique sont des approches qui ne tiennent pas du quantique à proprement parler, mais qui s'en inspirent en cela qu'elles mettent en œuvre des mécanismes qui s'approchent de concepts quantiques tels que la superposition, l'intrication ou l'effet tunnel. Par exemple, les machines de Ising [6] s'inspirent d'un modèle issu de la physique statistique, tel que proposé en 1925 par le physicien Ernst Ising [7]. Le modèle de Ising explique la transition de phase ferromagnétique dans les matériaux cristallins, et repose sur le fait que chaque site d'un réseau cristallin est occupé par un spin de valeur $+1$ ou -1 . Nous discuterons ce modèle plus en détail dans ce mémoire. Notons à ce stade que de nombreuses compagnies ont développées des machines basées sur le modèle de Ising, parmi lesquelles D-Wave Inc. (recuit quantique (*quantum annealing*: QA)) [8], Toshiba (bifurcation simulée (*Simulated Bifurcation*: SB)) [9] et Fujitsu (recuit digital (*Digital annealer*: DA)) [10].

Ces machines sont principalement utilisées dans la résolution de problèmes d’optimisation combinatoire.

Les problèmes d’optimisation combinatoires sont très fréquents en industrie, et apparaissent dans des domaines tels que la planification de la production, la logistique, la gestion de la chaîne d’approvisionnement, la gestion des ressources humaines, la finance etc. Certains problèmes combinatoires sont dits NP-difficiles. Les problèmes d’optimisation combinatoire NP-difficiles sont considérés parmi les problèmes les plus difficiles en informatique. On citera à titre d’exemples le problème du voyageur de commerce, le problème de sac à dos, le problème de coloration de graphe, etc.

1.3 Objectifs de recherche

L’objectif principal de ce mémoire est d’explorer l’utilisation des approches d’inspiration quantique, en particulier le modèle d’Ising, dans la résolution des problèmes d’optimisation combinatoire NP-difficiles. Pour ce faire, deux problèmes typiques seront étudiés et testés, à savoir le problème du voyageur de commerce (*traveling salesman problem*: TSP) [11–13] et le problème d’ordonnancement de tâches (*job-shop scheduling problem*: JSSP) [14]. Nous considérerons différentes plateformes commerciales dans notre travail, comme D-Wave [8] et Toshiba [9]. À titre comparatif, nous considérerons également des méthodes plus classiques telles que le recuit simulé. Ces expérimentations vont aider à évaluer les performances de cette approche sur différentes instances de problèmes ainsi que sa mise à l’échelle.

En somme, ce mémoire vise à contribuer à la compréhension de l’utilisation des approches d’inspiration quantique pour résoudre des problèmes d’optimisation combinatoire, d’en évaluer les performances et de proposer, à la lumière de nos résultats, des améliorations pouvant contribuer au domaine. Ainsi, ce travail vise les objectifs spécifiques suivants :

- Évaluer les performances des technologies d’inspiration quantique dans la résolution des problèmes NP-difficiles ;
- Identifier les limites de ces implémentations ; et
- Proposer des solutions efficaces pour en améliorer la performance et les valider expérimentalement.

1.4 Organisation du mémoire

L’articulation des chapitres du mémoire va se faire de la manière suivante. Dans le deuxième chapitre, une revue de littérature va être présentée dans le but de faire accrocher le lecteur

dans le contexte et lui faire comprendre les principes de base pour la bonne compréhension du reste du mémoire. Ce chapitre contiendra deux premières sous-sections où vont être présentées les problèmes d'optimisation combinatoires, les technologies quantiques ainsi que les approches d'inspiration quantique. Ce chapitre présentera également, dans une troisième section, les travaux réalisés en lien avec ces technologies, les problématiques qui ont émergé des résultats, ainsi que les éventuelles améliorations envisagées. Dans le troisième et quatrième chapitre, le mémoire introduira deux problèmes d'optimisation combinatoires complexes, à savoir le problème du voyageur du commerce et le problème d'ordonnancement de tâches. Ces deux chapitres visent à modéliser ces deux problèmes de manière formelle et mathématique et aussi dans leurs formulations en modèle Ising, présenter leurs mises en oeuvre et les comparer aux implantations déjà existantes ainsi qu'avec les approches d'inspiration quantique en termes de performance et de qualité de la solution obtenue. Et finalement, ces deux chapitres feront une étude quantitative en comparant les différentes performances de chacune des technologies appliquée à chacun des deux problèmes. Finalement, le mémoire se clôturera en donnant une synthèse des travaux faits ainsi que les limitations des différentes approches données pour résoudre les deux problèmes combinatoires cités précédemment en proposant les possibles améliorations à rapporter aux travaux de ce mémoire.

CHAPITRE 2 REVUE DE LITTÉRATURE

2.1 Introduction

Dans ce chapitre de revue de la littérature, nous explorerons les concepts fondamentaux des approches inspirées de la mécanique quantique, en mettant l'accent sur la machine d'Ising et le recuit simulé, ainsi que sur les différentes applications et les résultats obtenus dans ces domaines. Nous examinerons également les avantages et les limites de ces approches, ainsi que les tendances actuelles et futures de la recherche dans ce domaine. Nous discuterons des implications de ces approches pour positionner les travaux présentés dans ce mémoire.

2.2 Approches d'inspiration quantique

Les approches d'inspiration quantique sont des méthodes qui se basent sur les principes de la mécanique quantique pour résoudre des problèmes complexes comme l'optimisation. Sans prétendre à l'exhaustivité, voici une liste d'exemples :

1. Le recuit quantique [15, 16] dans laquelle un système mis à haute température est refroidi afin de se stabiliser à son niveau d'énergie le plus bas. C'est précisément le modèle utilisé par la compagnie D-Wave avec son produit *quantum annealer*.
2. Le modèle quantique de Markov caché [17] (MQMC), qui est une version améliorée du modèle de Markov caché [18]. Dans un MQMC, les observations sont toujours des états classiques, mais l'état du système est représenté par un état quantique.
3. La marche quantique [19] basée sur la marche aléatoire classique [20]. Elle consiste en une généralisation des marches aléatoires classiques, où les particules (en l'occurrence *les marcheurs*) sont décrits par des états quantiques.
4. Les réseaux de neurones d'inspiration quantique visant à implémenter un réseau de neurones basé sur les propriétés de la mécanique quantique. Les implémentations faites jusqu'à maintenant se limitent à un réseau de neurone à trois couches [21].

Bien que prometteuses en terme de performance, ces approches sont toujours en cours de développement et au stade de la recherche. Dans ce qui suit, nous allons présenter le modèle de Ising qui fera l'objet de l'étude proposée ici.

2.2.1 Le modèle de Ising

Les approches d'inspiration quantique se basent sur un modèle mathématique inspiré de la mécanique statistique. Le modèle de Ising en est un du genre. Il s'agit d'un modèle simple qui décrit l'interaction entre des particules magnétiques constitué d'un réseau de n spins σ_i , $i \in \{1, \dots, n\}$, où chaque spin peut prendre deux valeurs, $+1$ (spin "up") ou -1 (spin "down"). Le vecteur colonne $\boldsymbol{\sigma}$ décrit ainsi l'état du système : $\boldsymbol{\sigma} = (\sigma_i)_{i \in \{1, \dots, n\}}$. L'hamiltonienne du système (une mesure de l'énergie du système) est donnée par :

$$H(\boldsymbol{\sigma}) = - \sum_{\langle i, j \rangle} J_{i, j} \sigma_i \sigma_j - \sum_i h_i \sigma_i \quad (2.1)$$

où $\langle i, j \rangle$ indique une somme sur les paires de spins adjacents, $J_{i, j}$ est l'interaction entre les spins σ_i et σ_j , et h_i est le champ magnétique externe appliqué sur le spin σ_i . L'interaction entre deux spins est dite ferromagnétique si $J_{i, j} > 0$, et antiferromagnétique si $J_{i, j} < 0$.

L'hamiltonienne du modèle de Ising peut être exprimée de façon matricielle :

$$H(\boldsymbol{\sigma}) = -\boldsymbol{\sigma}^T \mathbf{J} \boldsymbol{\sigma} - \mathbf{h} \boldsymbol{\sigma} \quad (2.2)$$

où $\mathbf{J}$ est une matrice composées des termes $J_{i, j}$, et $\mathbf{h}$ est un vecteur ligne composé des champs magnétiques externes : $\mathbf{h} = (h_i)_{i \in \{1, \dots, n\}}$.

Si on définit $\boldsymbol{\sigma}_e$, un vecteur colonne composé de $\sigma_0 = +1$ et $\boldsymbol{\sigma}$, il est possible de définir $\mathbf{J}_e$:

$$\mathbf{J}_e = \begin{pmatrix} 0 & \frac{1}{2} \mathbf{h} \\ \frac{1}{2} \mathbf{h}^T & \mathbf{J} \end{pmatrix} \quad (2.3)$$

On obtient alors l'hamiltonienne comme suit :

$$H(\boldsymbol{\sigma}) = -\boldsymbol{\sigma}_e^T \mathbf{J}_e \boldsymbol{\sigma}_e \quad (2.4)$$

2.2.2 Modèles apparentés au modèle de Ising

Les modèles que nous allons examiner ici sont des variantes du modèle de Ising, conçus pour étudier des phénomènes spécifiques ou pour résoudre des problèmes particuliers. Parmi ces modèles, on peut citer le modèle de Potts, le modèle de Heisenberg, le modèle XY et le modèle Curie–Weiss. Nous explorerons dans ce qui suit les similitudes et les différences entre ces modèles et le modèle de Ising.

Le modèle de Potts

Le modèle de Potts [22] est un outil de modélisation mathématique utilisé pour étudier le comportement des structures pouvant être modélisées par des graphes complexes. Il s'agit d'une généralisation du modèle d'Ising qui permet à chaque spin d'adopter l'un des q états différents, contrairement au modèle d'Ising qui n'autorise que deux états possibles. Ainsi, le nombre total de configurations possibles dans ce modèle est de q^n , où n représente le nombre de spins. Ce modèle permet d'examiner les principaux composants d'un complexe et de prévoir comment ils interagiront pour déterminer le comportement général du complexe.

L'expression de l'hamiltonienne pour calculer l'énergie du système est définie comme suit :

$$H = - \sum_{\langle i,j \rangle} J_{i,j} \delta_{\sigma_i, \sigma_j} + \sum_i h_i \sigma_i \quad (2.5)$$

où $\langle i, j \rangle$ indique une somme sur les paires de spins adjacents, $J_{i,j}$ est l'interaction entre les spins σ_i et σ_j . Le symbole $\delta_{\sigma_i, \sigma_j}$ représente l'indice de Kronecker :

$$\delta_{\sigma_i, \sigma_j} = \begin{cases} 1 & \text{si } \sigma_i = \sigma_j \\ 0 & \text{sinon} \end{cases} \quad (2.6)$$

Ce modèle s'est révélé d'une grande utilité dans divers domaines scientifiques, soulignant son adaptabilité et sa polyvalence. Tout d'abord, en médecine, il est appliqué à la modélisation de la croissance tumorale, contribuant ainsi à l'amélioration des stratégies thérapeutiques. En mathématiques, il est exploité pour résoudre des problèmes tels que la coloration de graphes et l'ordonnancement [22].

Le modèle de Heisenberg

Le modèle de Heisenberg [23] s'agit d'un modèle de la mécanique quantique basé sur l'interaction entre les moments magnétiques (ou spins) des particules, en particulier les électrons, au sein d'un matériau. Dans ce modèle, les spins sont traités comme des opérateurs quantiques, ce qui signifie que leurs composantes sont quantifiées et qu'ils obéissent aux règles de la mécanique quantique. Les spins peuvent s'aligner parallèlement ou antiparallèlement à leurs voisins, et le modèle détermine l'état d'énergie le plus bas du système en fonction des interactions entre les spins. L'expression de l'hamiltonienne est donnée comme suit :

$$H = - \sum_{\langle i,j \rangle} J_{i,j} \vec{S}_i \cdot \vec{S}_j \quad (2.7)$$

où $J_{i,j}$ est l'interaction entre les spins i et j , $\vec{S}_i$ et $\vec{S}_j$ sont les opérateurs de spin pour les particules i et j respectivement. Si $J_{i,j}$ est positif, l'interaction entre les spins i et j est ferromagnétique, ce qui signifie que les spins ont tendance à s'aligner parallèlement pour minimiser l'énergie du système. Si $J_{i,j}$ est négatif, l'interaction est antiferromagnétique, et les spins ont tendance à s'aligner antiparallèlement.

Modèle XY

Le modèle XY, introduit en 1961 dans le papier fondateur par Lieb, Schultz et Mattis [24], consiste en une rangée de spins $\frac{1}{2}$. Dans ce modèle, seules les interactions entre les spins voisins sont prises en compte, excluant ainsi les interactions à longue distance. Son hamiltonienne est exprimée en fonction des composantes x et y , d'où son nom, comme suit :

$$H_\gamma = \sum_i [(1 + \gamma) S_i^x S_{i+1}^x + (1 - \gamma) S_i^y S_{i+1}^y] \quad (2.8)$$

où γ représente le degré d'anisotropie (degré de dépendance de la direction) dans le plan x - y . Les variables S_i^x et S_i^y représentent les matrices de Pauli de la mécanique quantique, définies comme suit :

$$S_i^x = \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad S_i^y = \frac{1}{2} \begin{pmatrix} 0 & -j \\ j & 0 \end{pmatrix} \quad (2.9)$$

où $j = \sqrt{-1}$ représente l'unité imaginaire des nombres complexes.

Dans le modèle XY, la configuration des spins peut varier en fonction de la manière dont la chaîne est définie à ses extrémités. La fin de la chaîne peut être soit "libre", ce qui signifie que les spins situés aux extrémités de la chaîne ne sont pas liés les uns aux autres. Dans cette

configuration, ces spins aux extrémités peuvent interagir avec d'autres spins plus éloignés dans la chaîne, ce qui permet d'étudier les interactions à longue distance et des propriétés spécifiques du système. En revanche, la chaîne peut également être "fermée en boucle". Dans ce cas, les spins aux extrémités sont reliés les uns aux autres, formant une boucle continue, de sorte que $S_{N+1}^x \equiv S_1^x$ et $S_{N+1}^y \equiv S_1^y$. Lorsque la chaîne est fermée en boucle, les spins à l'extrémité de la chaîne sont identiques à ceux du début, ce qui crée une structure périodique. Cette configuration est utile pour l'étude de systèmes périodiques où les spins interagissent selon une topologie circulaire. Ainsi, la façon dont la chaîne est définie à ses extrémités joue un rôle dans la manière dont les spins interagissent et influence les propriétés observées du modèle XY [25].

Modèle Curie–Weiss

Le modèle Curie-Weiss [26] est un modèle où N spins interagissent avec tous les autres spins avec la même force d'interaction. Afin d'éviter la possibilité que l'hamiltonienne prenne des valeurs supérieures à $\mathcal{O}(N)$, la force de cette interaction doit être proportionnelle à $\frac{1}{N}$, N étant le nombre de spins du modèle. L'hamiltonienne du modèle de Curie-Weiss est exprimée de la manière suivante :

$$H = -\frac{1}{N} \sum_{\langle i,j \rangle} s_i \cdot s_j - h \sum_i s_i \quad (2.10)$$

Ce modèle a un comportement paramagnétique (sans préférence) à haute température et un comportement ferromagnétique à basse température. La température à laquelle le modèle change de sa nature ferromagnétique vers sa nature paramagnétique s'appelle la température de Curie. Dans [27], Opoku *et al.* présentent une application de ce modèle pour aborder une problématique de prise de décision liée aux interactions sociales au sein d'une communauté.

2.2.3 Comparaison des modèles

Le tableau 2.1 compare de manière brève les principales différences entre les modèles de Ising, Potts, Heisenberg, XY et Curie-Weiss.

TABLEAU 2.1 Récapitulatif des modèles

Modèle	Type de système	Spin	Interaction	Transition de phase	Dimension
Ising	Système magnétique simple avec des spins discrets	Les spins sont des variables discrètes prenant les valeurs +1 ou -1	Les spins voisins interagissent, favorisant l'alignement des spins	Présente une transition de phase du paramagnétisme au ferromagnétisme	Peut être étudié en deux dimensions ou plus
Potts	Système généralisé basé sur le modèle d'Ising	Les spins peuvent prendre plusieurs valeurs discrètes (plus de deux).	Les spins voisins peuvent interagir de différentes manières, favorisant la formation de domaines	Présente des transitions de phase similaires à l'Ising, mais avec une plus grande variété de comportements possibles.	Étudié en deux dimensions ou plus
Heisenberg	Système magnétique où les spins sont des vecteurs tridimensionnels	Les spins sont des vecteurs de longueur constante sur une sphère	Les spins voisins interagissent en tendant à s'aligner, mais ils peuvent pointer dans n'importe quelle direction de l'espace	Présente des transitions de phase plus complexes que l'Ising, avec une transition de phase magnétique continue en trois dimensions	Étudié principalement en trois dimensions
XY	Système magnétique où les spins sont des vecteurs bidimensionnels	Les spins sont des vecteurs bidimensionnels de longueur constante sur un plan	Les spins voisins tendent à s'aligner sur un plan, mais ils peuvent avoir une orientation continue	Présente une transition de phase appelée transition Kosterlitz-Thouless, caractérisée par la création de paires de vortex-antivortex.	Étudié en deux dimensions
Curie-Weiss	Modèle de champ moyen pour un système magnétique	Les spins sont traités de manière continue et non discrète	Les spins sont influencés par un champ magnétique externe et par des interactions entre les spins	Il présente une transition de phase paramagnétique-ferromagnétique dans un champ magnétique externe	Est un modèle de champ moyen, donc il n'a pas de dimensionnalité spécifique

2.3 Méthodes de recuit simulé

La méthode de recuit simulé (*simulated annealing*: SA) est une métaheuristique de recherche locale permettant de trouver l'optimum d'une fonction à plusieurs variables. L'algorithme est conçu pour trouver aussi une approximation dans un large espace de recherche. Son utilisation est très populaire dans la résolution de problèmes d'optimisation combinatoires difficiles.

2.3.1 Formulation générale

La principale différence entre SA et d'autres algorithmes de recherche locale, tels que la descente de gradient, réside dans sa capacité à échapper aux optima locaux, avec l'espoir de trouver l'optimum global. Cette opération est réalisée grâce à l'utilisation d'une probabilité d'acceptation, qui contrôle la transition entre une configuration meilleure et une configuration moins favorable. Cette approche est inspirée de l'algorithme de Metropolis-Hastings [28, 29], qui introduit le facteur de Boltzmann dans le calcul de la probabilité de transition de s à s' :

$$p(s \longrightarrow s') = e^{-\frac{\Delta E}{kT}} \quad (2.11)$$

où $\Delta E = \epsilon_{s'} - \epsilon_s$ représente la différence d'énergie entre deux configurations du système s et s' respectivement, k est la constante de Boltzmann, et T est la température du système.

Pour accepter ou rejeter un nouvel état est comparée à un nombre aléatoire tiré d'une distribution uniforme $\mathcal{U}(0, 1)$. Si la probabilité est supérieure au nombre généré aléatoirement, l'état s' est accepté. Dans le cas contraire, l'état est rejeté.

2.3.2 Application du recuit simulé au modèle de Ising

Dans le contexte de la résolution du modèle d'Ising, le recuit simulé fonctionne de manière similaire. Au départ, une configuration initiale aléatoire des spins est générée. La température initiale est définie à un niveau élevé, ce qui signifie que le système a une grande probabilité d'accepter des configurations de spins de haute énergie. Cela permet d'explorer largement l'espace des solutions, y compris des configurations de spins sous-optimales.

La clé du recuit simulé réside dans le processus de refroidissement progressif. À chaque itération, le système subit une perturbation, généralement sous la forme d'un changement aléatoire de quelques spins. L'énergie de la nouvelle configuration est calculée, et la comparaison est faite avec l'énergie de la configuration précédente. Si la nouvelle configuration a une énergie plus basse, elle est acceptée comme la nouvelle solution. Cependant, si la nouvelle configuration a une énergie plus élevée, elle peut toujours être acceptée avec une certaine

probabilité, contrôlée par la différence d'énergie et la température actuelle. À mesure que la température diminue au fil des itérations, la probabilité d'accepter des configurations de haute énergie diminue également. Cela permet au recuit simulé de progressivement converger vers une solution optimale ou sous-optimale, en évitant les optimums locaux. Ce processus se poursuit jusqu'à ce que la température atteigne un seuil bas, à partir duquel le recuit simulé se stabilise et retourne la meilleure configuration trouvée comme résultat de l'optimisation.

2.3.3 Recuit quantique

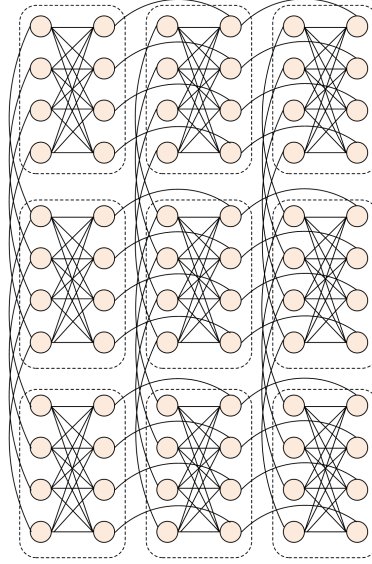
Le QA réfère à un ensemble d'heuristiques visant à trouver le minimum global d'une fonction objectif tout en tirant parti des principes de la mécanique quantique [30, 31]. Cette fonction objectif est souvent exprimée sous la forme de l'hamiltonienne du modèle Ising, qui correspond à un problème d'optimisation combinatoire.

Au début du processus, le système est placé dans un état de superposition de tous les états possibles à l'aide d'un champ transversal. Ensuite, le système évolue selon l'équation de Schrödinger tout en incorporant l'hamiltonienne et en réduisant progressivement le champ transversal. C'est ce champ qui engendre l'effet tunnel [30], permettant au système de franchir les barrières énergétiques pour atteindre des niveaux d'énergie plus bas. L'approche du recuit quantique diffère de celle de l'algorithme du recuit simulé (SA) en ce sens que ce dernier se base sur des changements de température pour sortir d'un minimum local.

2.4 Machines de Ising

Une machine d'Ising est un dispositif conçu pour rechercher ou approximer le minimum de la fonction qui décrit l'énergie du modèle d'Ising [6]. Ce modèle est associé à des problèmes NP-difficiles, qui appartiennent à la classe de problèmes impliquant un grand nombre de variables, et dont l'objectif est de trouver une configuration minimisant la fonction de coût. Ce qui distingue la machine d'Ising des méthodes conventionnelles, c'est sa capacité à aborder ces problèmes de manière plus efficace, en contournant les limitations temporelles et de puissance de calcul associées aux approches classiques. Cette efficacité devient particulièrement cruciale à mesure que les problèmes à résoudre deviennent plus vastes, car les méthodes traditionnelles peuvent devenir impraticables en raison de la disparition progressive de la loi de Moore.

Les machines d'Ising offrent une variété d'approches pour leur mise en œuvre, qu'elles relèvent du domaine classique ou quantique. Lorsqu'elles sont mises en œuvre à l'aide de techniques classiques telles que le recuit simulé ou le recuit quantique, ces machines présentent des caractéristiques quantiques, notamment l'effet tunnel. Une autre approche importante consiste

FIGURE 2.1 Graphe de chimère 3×3 

à utiliser des systèmes dynamiques, éliminant ainsi la nécessité de considérer la température pour conduire le système vers son état d'énergie minimale. Un exemple illustrant cette approche est la machine d'Ising cohérente, qui repose sur un réseau d'oscillateurs optiques couplés [32]. Dans la suite de cette section, nous explorerons différentes méthodes d'implémentation existantes pour les machines d'Ising.

2.4.1 Recuit quantique matériel

La méthode du recuit quantique englobe une famille de machines d'Ising commercialisées par D-Wave Inc. [8]. Ces machines quantiques d'Ising utilisent des boucles supraconductrices en tant que spins, interconnectées au moyen de jonctions Josephson [33]. Leur fonctionnement requiert des températures extrêmement basses, inférieures à 80 mK (-273.07°C), et nécessite une surface au sol considérable pour accueillir le système de refroidissement requis.

Les partisans de cette approche estiment qu'en exploitant l'effet tunnel quantique, ces machines peuvent offrir des accélérations significatives pour résoudre certains paysages énergétiques complexes et accéder à des niveaux d'énergie autrement inaccessibles [34]. Cependant, une étude récente menée par Vallejo, Benítez et Cano en 2021 [35] a révélé que les performances de la machine D-Wave présentent des limites, en particulier dans le cas de problèmes de grande envergure.

Cet état de faits est dû au fait que les machines D-Wave utilisent une topologie de réseau formant des grilles bidimensionnelles de qubits interconnectés. Ainsi, le modèle D-Wave 2000Q

utilise une topologie de graphe de chimère [36], telle que celle illustrée à la Figure 2.1. La génération qui a suivi, Advantage systems, utilise une topologie plus dense de graphe de pégase [36]. Les qubits Pegasus ont une longueur nominale de 12 (chaque qubit est connecté à 12 qubits orthogonaux via des coupleurs internes) et un degré de 15 (chaque qubit est couplé à 15 qubits différents).

Ainsi, dans les machines D-Wave, chaque qubit est directement connecté à un nombre restreint d'autres qubits situés à proximité immédiate, ce qui limite la connectivité globale du système. Cette limitation peut rendre complexe la modélisation de certains types de problèmes qui nécessitent des interactions entre un grand nombre de qubits éloignés les uns des autres. Ces contraintes topologiques ont un impact significatif sur la manière dont certains problèmes doivent être abordés et résolus au sein de ces machines.

Selon une étude expérimentale menée par [32], il a été démontré que pour mapper un problème impliquant N spins sur l'architecture Pegasus de D-Wave, il est nécessaire d'utiliser environ $N^2/8$ qubits. Cette évaluation est essentielle pour comprendre les limitations matérielles et les exigences en termes de ressources quantiques lors de la résolution de problèmes complexes sur les machines D-Wave.

2.4.2 Recuit numérique matériel

Le recuit numérique est une méthode qui implémente le model d'Ising sur un circuit numérique [37] et a abouti à la mise en marché par Fujitsu d'une machine de Ising basée sur une technologie CMOS conventionnelle appelée *Digital Annealer* (DA) [10]. Le DA comprends 8 000 nœuds entièrement connectés.

Le principe de l'implémentation de Fujitsu repose sur la méthode de Monte-Carlo par chaînes de Markov (*Markov Chain Monte-Carlo*: MCMC) [38]. Cette méthode combine les concepts des chaînes de Markov et de la méthode Monte-Carlo pour explorer efficacement l'espace des échantillons et générer des échantillons qui suivent la distribution de probabilité cible. Rappelons qu'une chaîne de Markov est un processus stochastique dans lequel l'avenir est prédit uniquement en fonction de l'état présent, sans tenir compte de la manière dont cet état a été atteint [39]. En utilisant la méthode de Monte-Carlo, il est possible de simuler des échantillons de la distribution de probabilité que l'on souhaite échantillonner. Plusieurs méthodes ont été proposées pour atteindre un optimum global.

L'ajout d'une compensation énergétique a été proposé pour réduire le temps passé dans un minimum local [38]. Dans cette optique, l'implémentation matérielle de Fujitsu vise à maximiser le parallélisme [37]. Pour y parvenir, l'inversion de plusieurs bits est considérée si-

multanément. En suivant le critère de Metropolis-Hasting [40], l'un de ces candidats est choisi pour devenir la nouvelle solution. De plus, une méthode d'échange de répliques (ou "replica exchange" en anglais) est utilisée. Au lieu d'initialiser plusieurs répliques du recuit avec la même température, chaque réplique est initialisée avec une température préalablement choisie. Ce mécanisme permet d'accélérer davantage la convergence vers un équilibre global. Une évaluation des performances a été réalisée dans l'article [37] pour comparer le recuit numérique à d'autres approches classiques et heuristiques sur différents problèmes d'optimisation, tels que le problème de la coupe maximale (max cut). Les résultats démontrent que le DA performe de manière optimale par rapport aux autres approches dans 80% des cas.

2.4.3 Bifurcation adiabatique

La bifurcation adiabatique est une méthode permettant d'implémenter une machine de Ising en se basant sur des oscillateurs non-linéaires, s'inspirant des travaux précédents tels que ceux présentés dans les références [41, 42]. La bifurcation adiabatique est un concept qui trouve son origine en physique, en particulier en mécanique quantique. Le terme "adiabatique" fait référence à un processus lent et sans perte d'énergie. Une machine basée sur les principes de la bifurcation adiabatique est actuellement commercialisée par Toshiba sous le nom de bifurcation simulée (*Simulated Bifurcation* — SB) [9]. Contrairement aux ordinateurs quantiques, la machine de SB fonctionne dans un environnement numérique classique, ce qui la rend plus accessible et pratique pour une utilisation industrielle et commerciale courante. Deux implémentations matérielles sont actuellement disponibles : une sur technologique circuit logique programmable (*field-programmable gate arrays*: FPGA) et une autre employant des implémentation processeur graphique (*graphical processing unit*: GPU) [9]. Selon [9], la SB permet de connecter totalement l'ensemble des nœuds si le nombre de nœud est raisonnablement bas. Toshiba affirme pouvoir simuler une machine avec un million de nœuds, mais cette implémentation repose sur une réseau peu dense.

2.4.4 Machine de Ising cohérente

La machine cohérente de Ising (*Coherent Ising machine*: CIM) est un dispositif physique qui tire parti de la dynamique des systèmes optiques pour simuler le comportement des spins dans le modèle d'Ising. En utilisant des oscillateurs optiques ou des lasers couplés, la CIM encode les problèmes d'optimisation en un réseau de spins dont les interactions sont représentées par les liaisons optiques. Les états des spins évoluent vers un état d'énergie minimale, qui correspond à la solution optimale du problème.

Une implémentation matérielle de la CIM est basé sur un réseau d'oscillateurs paramétriques

optiques déphasés [43]. Un oscillateurs paramétriques optiques déphasé (*Degenerate Optical Parametric Oscillator*: DOPO) est un type d'oscillateur optique qui génère deux photons de la même phase et de la moitié de la fréquence du photon original. Ces photons sont décrits comme "dégénérés" parce qu'ils partagent la même état quantique. Dans une CIM, chaque DOPO représente un spin dans le modèle d'Ising, pouvant être dans l'un des deux états de phase, correspondant aux états $+1$ ou -1 du spin. Les DOPO dans une CIM sont couplés les uns aux autres à travers un réseau optique. Ce couplage représente les interactions entre les spins dans le modèle d'Ising. La force et la nature de l'interaction entre deux DOPO reflètent la force et le signe de l'interaction entre deux spins dans le modèle. En ajustant les paramètres des DOPOs et leurs interactions, la CIM cherche à amener l'ensemble du système vers un état d'énergie minimale.

Dans leur étude, Kenta *et al.* décrivent une CIM composée de 16 nœuds, où chaque nœuds est connecté à trois autres [43]. Cette configuration a démontré une probabilité de succès de 99.6% pour trouver le minimum global du problème de graphe cubique. Par ailleurs, dans [44] Peter *et al.* ont proposé une CIM de 100 nœuds, où chaque nœuds connecté à tous les autres. Cette étude a concentré ses expériences sur l'échelle de Möbius. Les auteurs ont ainsi démontré que le temps total de calcul nécessaire pour obtenir les états fondamentaux (avec une précision de 100%) augmente rapidement avec la taille du problème, bien qu'il reste relativement faible en termes absolus pour $N = 100$ (0.2 s). L'augmentation du temps total de calcul est beaucoup moins marquée lorsque la précision requise est moindre : pour une précision de 95%, le temps nécessaire passe d'environ 1 ms pour $N = 10$ à environ 4 ms pour $N = 100$. En 2021, la compagnie Nippon Telegraph and Telephone (NTT) a indiqué avoir réalisé une CIM comportant 100 000 nœuds basée sur DOPO [45].

Oscillateur paramétrique optoélectronique

En utilisant des DOPO, les spins artificiels sont représentés par les phases de courtes impulsions optiques et stockés dans une longue cavité à fibre. Cependant, la stabilité de ce système est faible car la phase optique est très sensible aux variations du délai de la cavité [46]. Une autre manière d'implémenter les CIM afin de limiter cet inconvénient repose l'oscillateur paramétrique optoélectronique (*optoelectronic parametric oscillator*: OEPO). Dans [46], Qizhuang *et al.* présentent une implémentation d'une machine de Ising comportant 25 600 nœuds et basée sur OEPO. Dans cette implémentation, les spins sont représentés par les phases des impulsions micro-ondes courtes générées dans une cavité optoelectronique. Pour résoudre le problème d'Ising, les interactions entre les spins sont d'abord programmées dans le réseau d'oscillation, soit dans le domaine analogique, soit dans le domaine numérique. Ensuite, le

gain de la cavité est progressivement augmenté pour rechercher l'état de perte minimale. Parallèlement, les amplitudes des spins augmentent et leurs phases sont ajustées vers la configuration la plus basse. Lorsque le réseau d'oscillation fonctionne de manière stable, il est possible de mesurer la configuration de phase correspondante, à partir de laquelle la meilleure réponse aux problèmes donnés peut être résolue.

2.4.5 Machine de Ising à base d'oscillateurs analogiques

Une machine de Ising à base d'oscillateurs (*oscillator-based ising machine*: OIM) est une autre forme de machine de Ising utilisant des composants discrets RLC [47]. Le fonctionnement de l'OIM se fonde sur le principe du verrouillage de l'injection sous-harmonique. Sous cette injection, le système peut générer des états en déphasage représentant les états des spins. Dans leur article [48], Tianshi *et al.* proposent une OIM composée de 8 jusqu'à 240 nœuds. Pour la OIM composée de 64 et 240 nœuds, les auteurs ont proposé une architecture de type toroïdal [49] composée de 20 circuits imprimés. Les auteurs ont pu achever l'optimum global pour la OIM64 et des énergies proches de l'optimum global pour OIM240.

Dans leur étude, Tianshi *et al.* [47] comparent leur implémentation de l'OIM à d'autres méthodes de résolution telles que *scatter search*, CirCut [50], *variable neighborhood search with pathy ranking* [51] et SA. Cette comparaison a été réalisée en utilisant le problème de coupe maximale (Max-Cut) comme cas d'étude. Les résultats mettent en évidence que l'OIM atteint des valeurs maximales et surpasse parfois les autres approches. De plus, il est notable que le temps nécessaire pour atteindre ces valeurs est considérablement plus court par rapport aux autres méthodes.

2.4.6 Machine de Ising creuse

Dans [52], Aadit *et al.* présentent une nouvelle architecture exploitant le potentiel du parallélisme pour accélérer la simulation de l'algorithme de MCMC. Cependant, cette architecture est efficace uniquement lorsque la matrice d'interconnexions $\mathbf{J}$ est creuse. Le document propose également des approches pour augmenter davantage la densité de cette matrice au maximum. Le principe de cette parallélisation repose sur l'utilisation de multiples horloges déphasées pour contrôler l'activation des bits probabilistes. Le papier présente des résultats de comparaison entre la machine de Ising sparse (*sparse ising machine*: sIM) et d'autres alternatives telles que la machine de D-Wave, le Tensor Processing Unit (TPU) et le Graphical Processing Unit (GPU). Dans tous les problèmes combinatoires choisis, la sIM surpasse de loin les performances des autres alternatives.

2.4.7 Bits probabilistes

Une autre approche pour résoudre le modèle d’Ising est basée sur la magnétisation d’un réseau de nano-aimant stochastique (*stochastic nanomagnet*: SNM) [53]. Dans cette approche, chaque nano-aimant représente un spin du modèle d’Ising, et son état est déterminé par l’aimantation, soit $|\uparrow\rangle$ ou $|\downarrow\rangle$. Le couplage entre les nano-aimants se fait par le biais d’un courant électrique. Ce courant électrique, qui se transforme en courant de spin grâce à l’effet de Hall au niveau de chaque spin, induit une orientation spécifique aux électrons.

Ce modèle permet de favoriser les interactions à longue portée autant que les interactions locales, ce qui permet au système de s’étendre facilement. Pour résoudre le modèle d’Ising et atteindre l’énergie minimale, le système évolue en changeant de manière aléatoire l’état de chaque SNM. La simulation s’arrête lorsque le système atteint un état d’équilibre d’énergie minimale ou lorsqu’il se bloque dans un minimum local. Le papier décrit également une méthode permettant de sortir d’un tel minimum local, à savoir l’application d’un bruit par le biais d’une variation de l’amplitude du courant de spin.

2.5 Application à des problèmes d’optimisation combinatoires

Le modèle d’Ising a trouvé des applications importantes dans la résolution de divers problèmes d’optimisation combinatoires grâce à sa capacité à formuler ces problèmes en termes d’énergie minimale. Plus précisément, les variables de décision du problème sont associées aux spins et les contraintes et objectifs du problème sont traduits en termes d’énergie. L’objectif est de trouver la configuration de spins qui minimise l’énergie totale, ce qui équivaut à trouver la meilleure solution pour le problème d’optimisation. Des exemples de ces problèmes font partie des 21 problèmes NP-complets de Karp [54]. On peut citer le problème de la coupe maximale [55], le problème de coloration de graphes [56], le problème de verres de spin (SK Sherrington-Kirkpatrick) [57] et autres problèmes.

2.5.1 Problèmes considérés dans ce travail

Nous avons choisi d’évaluer les performances des machines de Ising dans la résolution de deux problèmes spécifiques, à savoir le problème du voyageur de commerce (TSP) et le problème de séquençage de tâches (JSSP). Ce choix découle du contexte dans lequel s’est déroulée notre recherche, et des intérêts de notre partenaire industriel. Rappelons que le problème du TSP est un défi classique de l’optimisation combinatoire, qui trouve des applications directes dans le secteur de la logistique. Le problème du JSSP est quant à lui une généralisation des problèmes d’ordonnancement rencontrés dans les industries manufacturières.

2.5.2 Évaluations de performance des machines de Ising

L'évaluation des performances des machines de Ising dans la résolution de problèmes d'optimisation combinatoire est un sujet de recherche actif et en évolution constante. Avec l'avancement de la technologie quantique, notamment les machines de Ising quantiques et les simulateurs quantiques, de nouvelles opportunités et défis se présentent pour résoudre des problèmes complexes. Les chercheurs s'efforcent de comprendre les limites et les avantages de ces machines par rapport aux approches classiques, ainsi que de développer des méthodes efficaces pour tirer parti de leur potentiel. L'optimisation de paramètres, la conception d'algorithmes adaptés et la comparaison rigoureuse avec d'autres méthodes d'optimisation sont au cœur de ces recherches. Cette quête vise à exploiter pleinement le pouvoir des machines de Ising pour résoudre des problèmes du monde réel de manière plus efficace et précise.

Dans l'article [58], Hamerly *et al.* démontrent que la CIM offre de bien meilleures performances que les machines de D-Wave (2X et 2000Q) sur les problèmes de coupe maximale et de verres de spin. Ils ont utilisé une CIM de 100 nœuds de Stanford ainsi qu'une CIM de 2048 nœuds de NTT pour leurs expériences. De plus, les performances du SB ont été évaluées dans l'article de Goto et Kanao [59], où ils ont montré que le SB produisait des résultats de haute qualité avec des temps d'exécution rapides par rapport à la CIM ou SA. Cette rapidité d'exécution est due à la méthode utilisée pour résoudre les équations, à savoir la méthode d'Euler semi-implicite, qui est exacte et dépend uniquement du taux d'incrémentations [60]. De plus, dans [61], Goto *et al.* ont comparé plusieurs approches, notamment le SA sur CPU et GPU, le SB et le DA. Leurs résultats montrent que, pour un problème impliquant un million de spins, le SB s'est révélé efficace et a produit des résultats proches de l'optimal avec un temps de résolution de l'ordre de quelques centaines de secondes, contrairement aux autres approches.

2.5.3 Résolution du TSP et du JSSP par modèle de Ising

Contrairement aux problèmes plus classiques de coupe maximale et de spins de verre, qui sont caractérisés par la densité de la matrice $\mathbf{J_e}$, les problèmes de TSP et de JSSP possèdent généralement des solutions plus conventionnelles qui correspondent à un optimum global dans des scénarios pratiques. En d'autres termes, il est possible d'obtenir des solutions de haute qualité pour le TSP et le JSSP sans nécessiter des approches quantiques ou des machines spécifiques, contrairement à des problèmes comme la coupe maximale et les spins de verre qui tirent pleinement parti des capacités des machines de Ising.

Le TSP peut être mappé sur un modèle d'Ising en traitant les villes comme des spins et en

configurant les interactions entre les spins de manière à représenter les distances ou les coûts de déplacement entre les villes. L'objectif est de minimiser l'énergie totale du système, ce qui correspond à trouver l'itinéraire le plus court dans le TSP. Plusieurs travaux ont exploré cette approche, en démontrant comment le TSP peut être résolu par des algorithmes basés sur le modèle d'Ising pour un nombre de villes inférieur à 10 [62, 63]. Cependant, pour les grandes instances du problème, elle présente des défis en termes de mise en œuvre pratique, de précision des solutions et de temps de calcul. Dans [64], Tao *et al.* ont démontré qu'une approche clusterée permet d'améliorer les performances du modèle de Ising.

En plus d'une exécution sur les machines de Toshiba, la littérature rapporte des exécutions sur D-Wave. Warren analyse quatre programmes logiciels pour résoudre le TSP symétrique sur Dwave Chimera [65]. Les résultats montrent comment une application peut fonctionner à la fois sur des plates-formes informatiques classiques et quantiques, en tirant parti de leurs forces respectives. Jain se penche également sur le TSP en le reformulant en tant que problème optimisation binaire quadratique sans contrainte (*quadratic unconstrained binary optimization*: QUBO) adapté au modèle de Ising [66]. Des expériences pratiques sont menées avec l'ordinateur quantique D-Wave Advantage de 5 000 qubits, et les performances sont comparées à un solveur classique. Les résultats révèlent que l'ordinateur quantique ne peut traiter que des instances de problème comportant 8 villes ou moins, avec des performances inférieures à celles du solveur classique, tant en termes de temps de résolution que de précision. Ces travaux illustrent les défis et les limites actuelles de la résolution du TSP sur D-Wave.

Venturelli *et al.* ont proposé la première formulation en modèle de Ising du JSSP [67]. Dans cette même publication, Venturelli *et al.* ont évalué le JSSP sur une machine D-Wave 2, mais le petit nombre de qubits disponibles, soit 512 qubits, ne permettait qu'une analyse limitée. Une évaluation du JSSP sur D-Wave 2000Q, qui dispose de 2048 qubits a été menée dans [68], mais elle nécessitait une partitionnement du problème. Enfin, plusieurs problèmes ont été évalués avec succès sur un recuit numérique classique capable de gérer jusqu'à 8192 variables, en utilisant la même formulation du JSSP développée pour les recuits quantiques [69]. Cela a montré une amélioration de la qualité des solutions sur certaines instances. Finalement, Carugno *et al.* ont démontré l'efficacité de ces dispositifs sur des problèmes très contraints, mais indiquent que des recherches supplémentaires sont nécessaires pour exploiter pleinement le potentiel du recuit quantique dans ce contexte [70].

2.5.4 Outils et méthodes de résolution pour TSP

Le problème du voyageur de commerce est l'un des problèmes les plus célèbres en optimisation combinatoire. Il pose la question suivante : "Un voyageur doit visiter un ensemble

de villes exactement une fois, puis revenir à sa ville de départ. Quel est le chemin le plus court qu'il peut emprunter pour accomplir cette tâche?" En raison de sa complexité combinatoire, le TSP présente un défi mathématique et algorithmique, et il a attiré l'attention de chercheurs du monde entier pour développer des méthodes de résolution efficaces et des approches d'optimisation pour résoudre ce problème fondamental.

Il existe plusieurs approches algorithmiques permettant de résoudre le TSP. Certaines offrant une solution exacte, d'autre des solutions approximées. Les approches offrant une solution exacte se basent sur certaines hypothèses simplificatrices, telle que l'inégalité triangulaire. D'autres approches existent fournissant une solution approximative comme l'heuristique de Christofides [71] et Lin-Kernighan [72] basée sur la méthode *k-opt* [73]. À noter que les approches heuristiques cités sont efficaces pour résoudre le TSP.

L'outil Concorde

Le logiciel Concorde [74] est un logiciel qui permet de résoudre le problème du TSP en donnant une solution déterministe en se basant sur plusieurs méthodes graphiques et heuristiques. Le logiciel utilise la méthode *branch-and-cut* [75] qui combine les deux méthodes *branch-and-bound* [76] et la méthode graphique des plans sécants (en anglais *cutting-plane method*) [77].

La méthode branch-and-bound

La méthode *branch-and-bound* est une méthode couramment utilisée pour résoudre de manière optimale le TSP. Son principe consiste à diviser le problème initial en plusieurs sous-problèmes plus petits et à éliminer ceux qui ne peuvent pas contenir la solution optimale en utilisant une fonction de coût. Cette approche repose sur la construction d'un arbre de recherche, dont la racine représente la ville de départ de la tournée. Chaque nœud enfant représente une ville non encore visitée dans le parcours, et chaque arête indique la distance entre les deux villes.

Le processus commence par l'estimation d'une borne inférieure de la distance optimale, pouvant être obtenue de manière arbitraire à l'aide d'une heuristique, telle que l'heuristique du plus proche voisin ou en utilisant l'arbre couvrant de poids minimum (à l'aide de l'algorithme de *Prim* ou de *Kruskal*). Ensuite, la méthode calcule le coût de chaque nœud enfant, sélectionne celui ayant le coût le plus bas, puis élimine ce nœud. Ce processus se répète jusqu'à ce que toutes les villes aient été visitées, et la tournée se termine par le retour à la ville de départ.

La méthode des plans sécants

La méthode des plans sécants [77] est une technique d'optimisation permettant de trouver une solution entière à un problème linéaire avec des contraintes entières. Son principe consiste à itérer plusieurs fois jusqu'à trouver une solution entière. À chaque itération des plans sécants sont ajoutés pour relaxer le problème linéaire. L'avantage de cette méthode se manifeste dans le fait qu'à chaque itération l'espace de recherche se réduit de manière significative. Ceci a pour effet de rendre la recherche vers l'optimal efficace.

L'heuristique de Christofides

L'heuristique de Christofides [71] proposée en 1976 donne une solution au TSP dans un graphe complet $G = (V, E)$ non orienté. La solution représente une solution approximative bornée à 50% de la solution optimale. Son principe consiste à trouver une tournée de poids minimum. Il se base sur deux autres problèmes à savoir le problème de l'arbre de coût minimum et le problème du cycle eulérien, considérés comme des problèmes faciles.

Algorithm 1 L'algorithme de Christofides

Étape 1 : Déterminer un arbre $T = (V, E')$ de G couvrant de poids minimum.

Étape 2 : Déterminer un couplage M de coût minimum dans le sous-graphe de G induit par les sommets de T ayant un degré impair.

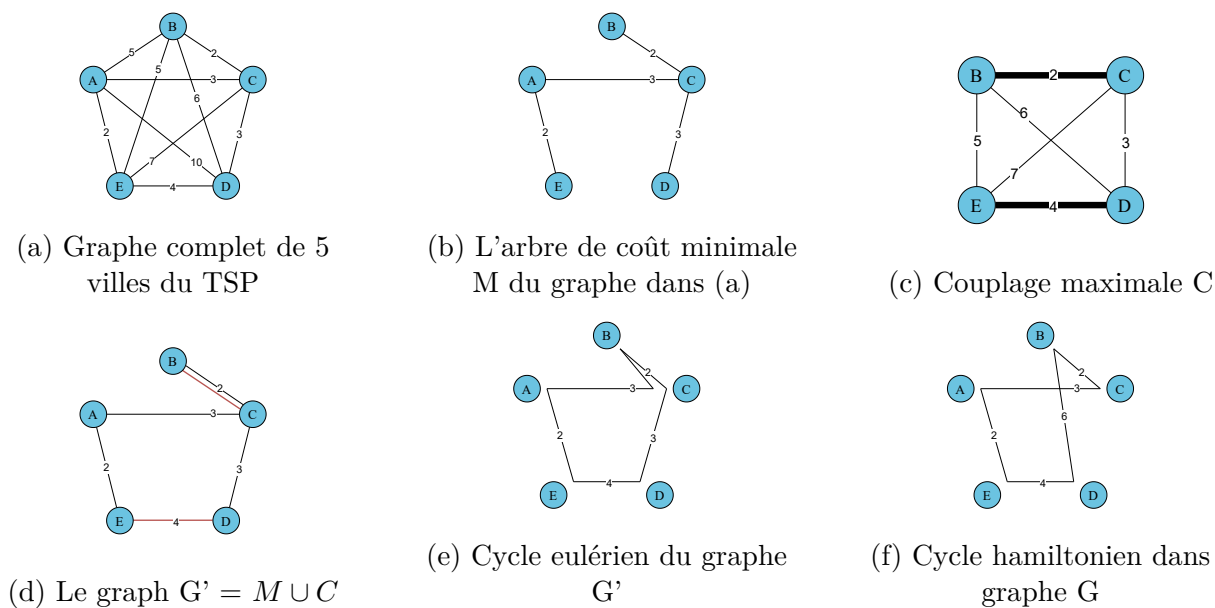
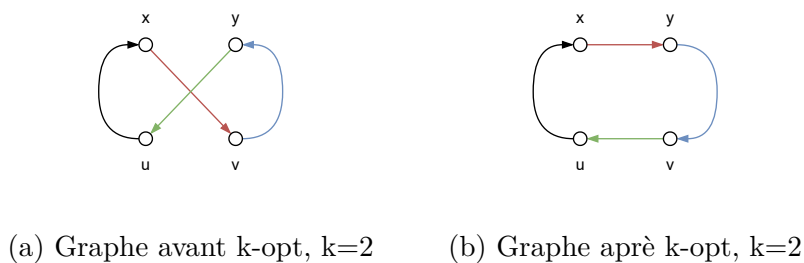
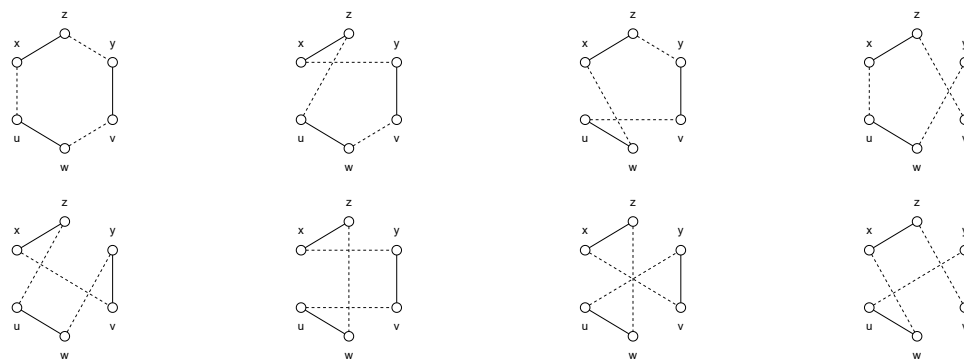
Étape 3 : Le graphe $G' = (V, E' \cup M)$ a tous ses sommets de degré pair, et on peut donc construire un cycle eulérien qui passe au moins une fois par chaque sommet de G . Ce cycle peut éventuellement être raccourci s'il passe plusieurs fois par un même sommet.

Une illustration de cet algorithme a été représentée dans la Figure 2.2 sur un graphe de 5 villes décrivant en détails les étapes de construction du cycle hamiltonien. La Figure 2.2f donne une solution au problème du TSP. Cette solution peut être améliorée grâce des méthodes comme k-opt.

La méthode k-opt

La méthode k-opt est utilisée principalement pour améliorer la qualité d'une solution. Son principe consiste à faire le choix de k arêtes du graphe et les échanger avec d'autres k arêtes et espérer trouver un cycle raccourci. La Figure 2.3 illustre cette méthode de manière générale pour k=2. Dans cet exemple, il existe une manière de relier la chaîne $[u, x]$ et la chaîne $[v, y]$, comme représenté dans la Figure 2.3b. La Figure 2.4 illustre la méthode k-opt avec $k = 3$. Dans ce cas, sept (7) manières d'arranger les arêtes existent.

FIGURE 2.2 Étapes de l'algorithme de Christofides

FIGURE 2.3 Exemple d'un k-opt, $k=2$ FIGURE 2.4 Exemple d'un k-opt, $k=3$ 

En termes de complexité, cette méthode croît rapidement en $\mathcal{O}(n^k)$ ce qui limite l'utilisation de larges valeurs pour k . En effet, pour un k donné, il existe $\binom{n}{k} \approx \mathcal{O}(n^k)$ façons de choisir k arêtes parmi n arêtes. Cette méthode s'avère efficace à une certaine valeur de k . Dans la littérature, la valeur maximale atteinte est de 5. Au delà de cette valeur, le temps d'exécution est très grand. Dans la littérature, le k -opt est combiné avec autres mécanismes d'optimisations. Comme par exemple la méthode LKH, qui se base sur k -opt et ajoute une couche permettant de choisir avec efficacité les arêtes menant à la solution optimale.

2.5.5 Outils de résolution pour JSSP

De même pour le problème JSSP, plusieurs approches algorithmiques existent offrant aussi une solution exacte et approximées. À savoir par programmation par contraintes utilisée par l'outil Google OR-Tools. La méthode de programmation linéaire mixte-entière avec l'outil CPLEX permet de résoudre le problème avec optimalité ou l'approxime au facteur prêt.

Google OR-Tools est une suite de logiciels open-source développée par Google qui propose une collection d'outils et de bibliothèques d'optimisation pour divers problèmes d'optimisation combinatoire. Ces outils sont conçus pour résoudre des problèmes d'optimisation complexes dans différents domaines, notamment la recherche opérationnelle, la logistique, la planification, et plus encore.

Tous les problèmes d'optimisation cités précédemment ont un nombre astronomique de solutions et que l'ordinateur classique ne pourra les parcourir tous. OR-Tools propose différents modules adaptés afin de combler cette limitation. La spécialisation de Google OR-Tools est dans la résolution des problèmes d'optimisation spécifiques, tels que : programmation par contraintes (*constraint programming*: CP), programmation linéaire (*linear programming*: LP), programmation linéaire entière (*mixed-integer programming*: MIP), problème de tournées de véhicules (*vehicle routing problem*: VRP) et problèmes de graphes.

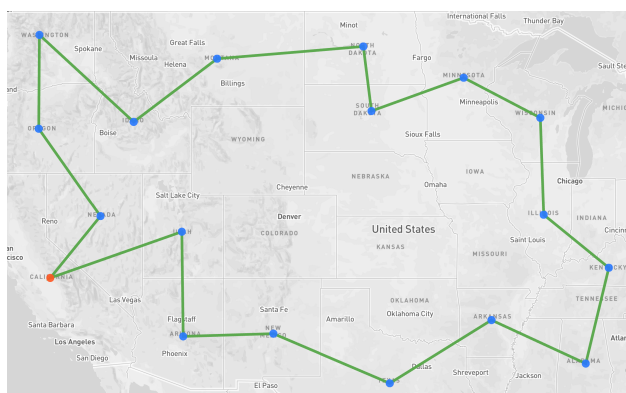
Dans le contexte de ce mémoire, cet outil jouera un rôle essentiel en générant des résultats supplémentaires afin de les comparer avec les données obtenues pour le problème JSSP dans sa configuration standard. Cette approche renforcera notre analyse en fournissant une base de référence complète pour évaluer les performances dans la résolution du JSSP.

CHAPITRE 3 PROBLÈME DU VOYAGEUR DE COMMERCE

3.1 Définition du problème

Le problème du Voyageur de Commerce (TSP) est un défi d'optimisation combinatoire visant à déterminer le parcours le plus court permettant de passer une fois par chaque ville tout en minimisant la distance totale du trajet. Il est classé parmi les problèmes NP-difficiles, ce qui signifie qu'il est réputé difficile à résoudre en temps polynomial. Les applications de ce problème sont nombreuses, notamment dans l'industrie des circuits imprimés, où il vise à minimiser les chaînes d'inspections, ainsi que dans les réseaux de fibres optiques, où il contribue à optimiser le câblage alimentant les équipements.

FIGURE 3.1 Exemple de solution pour le TSP, générée par www.tspvis.com



3.2 Un problème factoriel

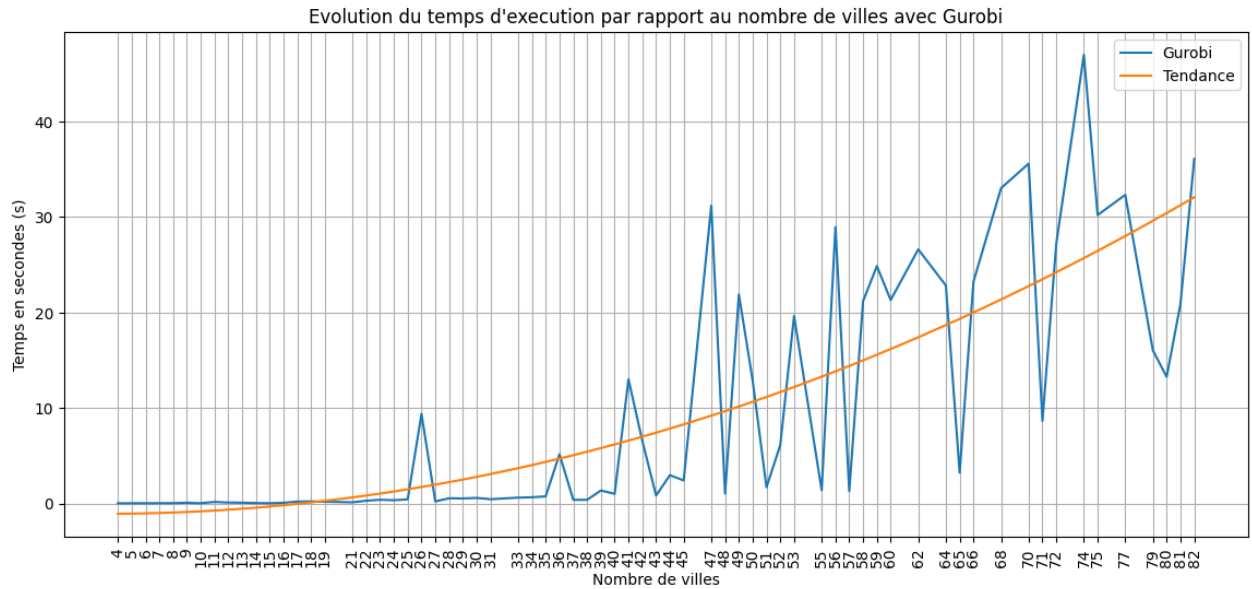
Le problème du voyageur de commerce (TSP) est caractérisé par une croissance exponentielle du nombre de combinaisons possibles à mesure que le nombre de villes N augmente. En effet, le nombre total de combinaisons possibles pour relier N villes distinctes est de l'ordre de $N!$. Ainsi, le temps d'exécution d'un algorithme de recherche exhaustif pour évaluer toutes ces combinaisons est de l'ordre de $(N-1)!/2$, car les trajets circulaires présentent des duplications et on peut fixer la ville de départ. Le tableau 3.1 illustre cette croissance exponentielle en fournissant le nombre total de combinaisons possibles et le temps nécessaire pour les évaluer toutes, ce qui montre à quel point le problème devient rapidement intraitable à mesure que N augmente. Les résultats ont été obtenus avec un ordinateur dotée d'un processeur Intel Core i7 de 11^e génération de fréquence 2,3 Ghz.

TABLEAU 3.1 Temps de calcul du TSP

Nombre de villes	Nombre de combinaisons	Temps de calcul (s)
4	3	0.000145
5	12	0.000294
6	60	0.002341
7	360	0.014121
8	2 520	0.152290
9	20 160	1.443108
10	181 440	15.009096
11	1 814 400	177.865574

La Figure 3.2 présente l'évolution du temps d'exécution lors de l'utilisation de Gurobi [78] pour résoudre le TSP, tandis que la Figure 3.3 illustre le temps d'exécution correspondant lors de l'utilisation de l'outil Concorde [74]. Il est clair que Concorde se distingue par son efficacité en matière de résolution du TSP, ce qui souligne l'importance d'avoir des solveurs spécialisés pour des problèmes spécifiques comme le TSP.

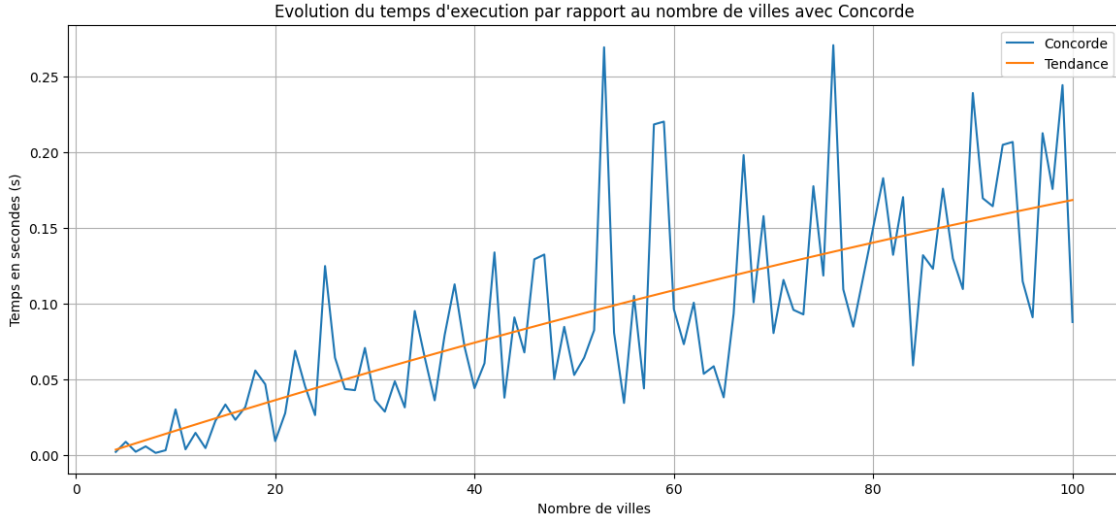
FIGURE 3.2 Évolution du temps d'exécution par rapport au nombre de villes avec Gurobi



3.3 Formulation du problème

Le problème peut être formulé sous la forme d'une fonction objectif et de contraintes, en tant que problème d'optimisation linéaire en nombres entiers [79]. La fonction objectif vise à

FIGURE 3.3 Évolution du temps d'exécution par rapport au nombre de villes avec Concorde



minimiser la distance totale du trajet, tandis que les contraintes garantissent le passage par toutes les villes une et une seule fois. Dans la littérature, deux formulations sont couramment utilisées : la formulation de Miller-Tucker-Zemlin (MTZ) [80] et la formulation de Dantzig-Fulkerson-Johnson (DFJ) [81]. Ces deux formulations expriment le problème de manière différente.

3.3.1 Formulation générale

Dans le problème du TSP de N villes, les villes sont numérotées de 1 à N .

Soit $x_{i,j}$ une variable binaire :

$$x_{i,j} = \begin{cases} 1 & \text{la ville } j \text{ suit la ville } i \\ 0 & \text{sinon} \end{cases} \quad (3.1)$$

et soit $c_{i,j}$ la distance entre la ville i et la ville j . L'objectif du est de minimiser la distance totale du chemin exprimée en fonction des variables décrites auparavant de la manière suivante :

$$\sum_{\substack{(i,j) \in \{1,\dots,N\}^2 \\ i \neq j}} c_{i,j} x_{i,j} \quad (3.2)$$

Pour avoir un chemin valide consistant à visiter une seule fois chaque ville, le système doit faire en sorte de rentrer et sortir une seule fois par chaque ville. Ceci peut être exprimé de la manière suivante :

$$\forall i \in \{1..N\}, \sum_{j=1}^N x_{i,j} = 1 \quad (3.3)$$

$$\forall j \in \{1..N\}, \sum_{i=1}^N x_{i,j} = 1 \quad (3.4)$$

3.3.2 La formulation de MTZ

Dans [80], Miller, Tucker et Zemlin (MTZ) ont présenté le problème de manière très générale en prenant en compte le nombre de fois où il faut revenir au nœud de départ et également la longueur du chemin en termes du nombre d'arêtes à parcourir. Cette formulation introduit la variable u_i qui décrit l'ordre de visite de la ville i , ainsi $u_i < u_j$ veut dire que la ville i est visitée avant la ville j . Autrement dit, la variable u_i est égale au nombre d'arêtes parcourues avant d'arriver à la ville i . On a alors :

$$u_j \geq u_i + 1 \quad \text{si} \quad x_{i,j} = 1 \quad (3.5)$$

L'expression est généralisée de la manière suivant :

$$u_j + (N - 2) \geq u_i + (N - 1)x_{i,j}, \quad \forall i, j \in \{2, \dots, N\} \quad \text{et} \quad i \neq j \quad (3.6)$$

Cette nouvelle contrainte permet de forcer le chemin de passer par tous les chemins en une tournée et d'éliminer les sous-tournées disjointes visitant toutes les villes.

3.3.3 La formulation de DFJ

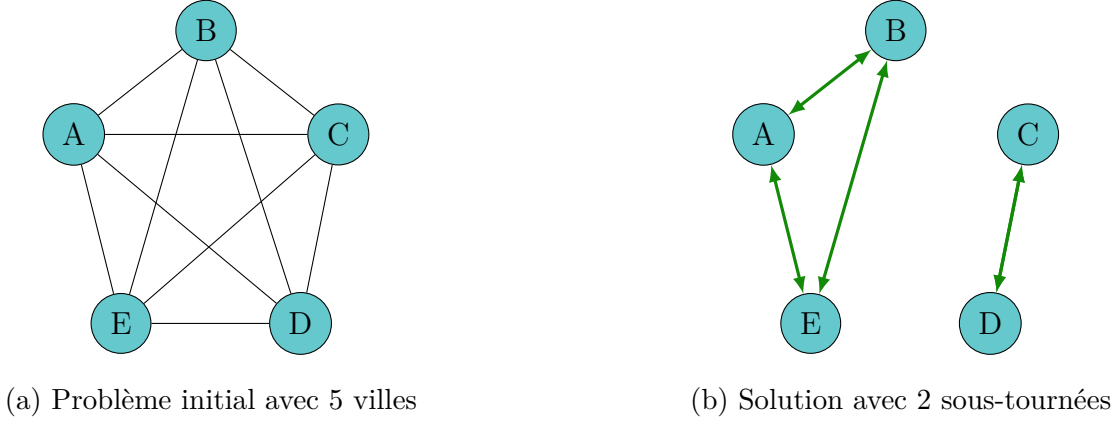
Martin et al. ont introduit dans [81] une nouvelle contrainte nommée *Contrainte d'élimination de sous-tournées*. Cette contrainte permet d'éliminer les cas où le chemin est une union disjointe de sous-tournées. Elle est exprimée de la manière suivante :

$$\sum_{i \in Q} \sum_{\substack{i \neq j \\ j \in Q}} x_{i,j} \leq |Q| - 1, \quad \forall Q \subsetneq \{1, \dots, N\}, \quad |Q| \geq 2 \quad (3.7)$$

où Q est un sous-ensemble strict des villes de cardinalité 2. Cette expression stipule que sur un ensemble de m villes, il faut avoir au plus $m - 1$ arêtes pour connecter les m villes. Ceci va éliminer les sous-tournées par exemple les tournées A-B-C et C-D comme indiqué dans la

figure suivante :

FIGURE 3.4 Exemple d'élimination de sous-tournées



3.3.4 Formulation QUBO

Le QUBO est un problème d'optimisation combinatoire faisant partie de la classe des problèmes NP-difficiles. Ce modèle introduit des termes quadratiques et linéaires et sa forme générale est décrite par l'équation 3.8.

$$H(\mathbf{x}) = \sum_{\langle i,j \rangle} Q_{i,j} x_i x_j + \sum_i Q_{i,i} x_i \quad (3.8)$$

Afin de formuler le problème sous la forme QUBO en fonction des variables du problèmes, on considère les spins dans une matrice de dimension carrée de taille $N \times N$. Elle est représentée de la manière suivante pour un problème de 4 villes A,B,C et D :

	A	B	C	D
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

(3.9)

Dans cette représentation matricielle, chaque colonne représente une ville et chaque rangée représente l'ordre de la visite. Pour chaque ville donnée u et chaque ordre de visite i , on assigne une variable binaire qu'on note $x_{u,i}$. Ainsi la traduction de cette matrice est le chemin $A \rightarrow B \rightarrow C \rightarrow D$.

La formulation QUBO est donc décrite comme suit :

$$H = A \sum_{(uv) \in E} W_{uv} \sum_{j=1}^N x_{u,j} x_{v,j+1} + B \sum_{v=1}^N (1 - \sum_{j=1}^N x_{v,j})^2 + C \sum_{j=1}^N (1 - \sum_{v=1}^N x_{v,j})^2 \quad (3.10)$$

Le premier terme décrit la longueur du chemin. En effet, si les deux termes $x_{u,j}$ et $x_{v,j+1}$ à un moment donné j sont tous les deux égaux à 1, donc le chemin connecte la ville u à la ville v . Pour un problème composé de 4 villes, si on suppose que le chemin est $B \rightarrow C \rightarrow A \rightarrow D$, la valeur de l'expression 3.10 après élimination des termes nuls est donnée par :

$$W_{BC}x_{B,0}x_{C,1} + W_{CA}x_{C,1}x_{A,2} + W_{AD}x_{A,2}x_{D,3} + W_{DB}x_{D,3}x_{B,0} = W_{BC} + W_{CA} + W_{AD} + W_{DB} \quad (3.11)$$

Le deuxième terme décrit la contrainte qui stipule que chaque ville devra être visitée une seule fois dans la tournée. Cette contrainte élimine ainsi les sous-tournées comme décrit dans la formulation DFJ. Si la tournée contient une telle sous-tournée alors l'hamiltonienne ne sera pas minimale. Le troisième terme décrit la contrainte qui stipule qu'à chaque moment une seule ville peut être visitée à la fois.

Les hyper-paramètres A , B et C aident à calibrer l'intensité de la fonction objectif et les contraintes. Le choix des valeurs de B et C doit être fait judicieusement pour ne pas éliminer les solutions pouvant représenter la solution optimale au problème. La valeur de A peut être mise à 1.

La formulation QUBO peut ensuite être utilisée pour formuler le modèle de Ising en utilisant le changement de variable $\sigma_i = 2x_i - 1$:

$$\begin{aligned} H_{tsp} = & \frac{A}{4} \sum_{k \neq l} \sum_i W_{kl} \sigma_{ik} \sigma_{(i+1)l} + \frac{A}{2} \sum_{k \neq l} \sum_i W_{kl} \sigma_{ik} \\ & + \frac{B}{4} \sum_i \sum_k \sum_l \sigma_{ik} \sigma_{il} + \frac{(n-2)B}{2} \sum_i \sum_k \sigma_{ik} \\ & + \frac{C}{4} \sum_i \sum_k \sum_j \sigma_{ik} \sigma_{jk} + \frac{(n-2)C}{2} \sum_i \sum_k \sigma_{ik}, \end{aligned} \quad (3.12)$$

3.4 Implémentations avec des approches d’inspiration quantique

Cette partie visera à présenter les différentes implémentations avec les différentes approches et méthodes à savoir le SA, le recuit quantique de D-Wave et la machine à bifurcation de Toshiba. L’accès aux machines de D-Wave est donné gratuitement pour une durée d’un mois pour leur dernière génération de processeur quantique. Par contre l’accès à la machine à bifurcation de Toshiba a été fourni par le partenaire industriel du projet.

3.4.1 Implémentation du SA

L’algorithme 2 repose sur SA pour résoudre le problème du TSP en utilisant le modèle de Ising. La température T diminue au fil des itérations pour permettre à l’ensemble d’états du système de converger vers un optimum global ou local. L’idée clé réside dans la probabilité d’accepter une modification d’état, même si elle entraîne une augmentation de l’énergie du système. Cette probabilité dépend de la différence d’énergie ΔE entre l’état actuel et l’état proposé, ainsi que de la température courante T . Plus la température diminue, moins il est probable d’accepter des états de haute énergie, ce qui permet de converger vers la configuration de plus basse énergie, correspondant à la solution recherchée.

Dans notre étude, nous avons exploré différentes implémentations pour déterminer les voisins dans le cadre du TSP. La première méthode consiste à inverser la valeur d’un seul spin dans le modèle d’Ising, choisi aléatoirement. La deuxième implémentation implique l’inversion de deux rangées dans la représentation matricielle du TSP, également choisies aléatoirement. Cela se traduit par l’échange de l’ordre de visite de deux villes dans la tournée. Enfin, la troisième méthode s’appuie sur l’algorithme 2-opt, présenté au Chapitre 2, qui consiste à inverser une paire de connexions entre deux paires de villes adjacentes dans la tournée. Si la configuration de spins initiale est valide, ainsi le seront les suivantes.

Ces différentes implémentations ont été évaluées pour déterminer leur efficacité dans la résolution du TSP en utilisant l’algorithme de recuit simulé du modèle d’Ising. Nous avons analysé leur impact sur les performances de l’algorithme en termes de qualité de solution, de temps d’exécution et de capacité à échapper aux optimaux locaux. Cette étude nous a permis de mieux comprendre comment l’algorithme de recuit simulé peut être adapté pour résoudre le TSP, en exploitant différentes manières de déterminer les voisins dans le modèle d’Ising, et d’identifier les avantages et les limites de chaque approche.

Algorithm 2 Recuit simulé du modèle d'Ising

Entrées :

Température initiale T_{init} ,

Température finale T_{final} ,

Nombre d'itérations maximal N_{max} ,

Configuration initiale σ ,

Fonction d'énergie H ,

Taux de refroidissement α .

Sortie :

Configuration σ avec énergie minimale.

$T \leftarrow T_{\text{init}}$ // Initialisation de la température

$iter \leftarrow 0$ // Initialisation du compteur d'itération

$\sigma_{\text{min}} \leftarrow \sigma$ // Configuration minimale

$E_{\text{min}} \leftarrow H(\sigma)$ // Énergie minimale

while $T > T_{\text{final}}$ **and** $iter < N_{\text{max}}$ **do**

 Choisir un voisin σ_{voisin} de σ .

 Calculer $\Delta E \leftarrow H(\sigma_{\text{voisin}}) - H(\sigma)$.

if $\Delta E < 0$ **ou** $\exp\left(-\frac{\Delta E}{T}\right) > \text{aléatoire}(0, 1)$ **then**

$\sigma \leftarrow \sigma_{\text{voisin}}$

if $H(\sigma) < E_{\text{min}}$ **then**

$E_{\text{min}} \leftarrow H(\sigma)$

$\sigma_{\text{min}} \leftarrow \sigma$

end if

end if

$T \leftarrow T \times \alpha$ // Refroidissement

$iter \leftarrow iter + 1$ // incrémentation du compteur d'itérations

end while

Renvoyer σ_{min} // Configuration avec énergie minimale

3.4.2 Implémentation sur la plateforme de D-Wave

La plateforme de D-Wave offre un environnement de développement pour ses machines quantiques grâce au kit de développement (*software development kit*: SDK) Ocean, ainsi que différents solveurs. D-Wave propose plusieurs méthodes pour modéliser le problème. La première consiste à utiliser des fonctions utilitaires pour coder les contraintes et la fonction objectif. Les autres méthodes consistent à soumettre des problèmes en envoyant la matrice Q de l'équation 3.8 ou la matrice $\mathbf{J}$ et le vecteur $\mathbf{h}$ de l'équation 2.1. Chaque méthode de modélisation est conçue pour un solveur spécifique. La première méthode est destinée au solveur hybride, tandis que les deux autres méthodes utilisent la machine quantique qui applique le

Modélisation 2

Cette modélisation consiste à créer la matrice Q de l'équation 3.8 et la transformer au modèle d'Ising vers la matrice $\mathbf{J}_e$. Cette modélisation ne permet pas de résoudre le problème du TSP avec plus de 14 villes contrairement à la modélisation 1. En effet, l'architecture actuelle *Advantage* de D-Wave basée sur la topologie *Pegasus* contient 5 640 et un problème de 14 villes nécessite $\frac{(14^2)^2}{8} = 4\,802$ qubits physiques.

3.4.3 Implémentation sur la plateforme de Toshiba

L'implémentation sur la plateforme de Toshiba ne diffère pas de celle de D-Wave. En effet, tout comme l'interface de programmation (*application programming interface*: API) de D-Wave, il est possible de coder les termes linéaires et quadratiques à partir des valeurs dans la matrice Q . L'API se charge de transformer cette matrice en $\mathbf{J}_e$ au moment où le problème est envoyé au solveur.

3.5 Expériences, résultats et discussions

Dans cette section, nous présenterons une série d'expériences entreprises pour aborder la résolution du TSP en utilisant des machines de type Ising. Nous évaluerons notamment de nouvelle approche pour explorer l'espace des solutions en combinant l'intervertissement des lignes et k-opt basée et le recuit simulé pour résoudre le problème. Les résultats mettent en évidence les avantages et les limites des approches proposées dans le contexte du TSP. Enfin, une discussion générale sera proposée pour récapituler les résultats obtenus, mettre en évidence les défis persistants et discuter des perspectives futures.

3.5.1 Résultats sur D-Wave et Toshiba

La Figure 3.5 illustre la probabilité de succès dans la résolution du problème du voyageur de commerce (TSP) en utilisant l'ordinateur quantique D-Wave. Pour chaque instance de problème, 10 exécutions distinctes ont été réalisées. Les résultats révèlent que le solveur D-Wave éprouve des difficultés à résoudre le TSP de manière optimale, avec une probabilité de succès inférieure à 40% à partir de neuf villes.

La Figure 3.6 offre un aperçu de la distribution des solutions obtenues à partir de dix essais pour chaque instance du problème. Une observation importante est que, même lorsque la probabilité de succès est nulle pour certaines de ces instances, les solutions obtenues ne sont pas très éloignées de l'optimum. Cela suggère que, bien que D-Wave puisse avoir des

difficultés à trouver la solution optimale dans certains cas, les solutions qu'il fournit tendent à être relativement proches de la meilleure solution possible. Cela est de moins en moins vrai avec la croissance de la taille du problème.

FIGURE 3.5 Évolution de la probabilité de succès de la solution de D-Wave Hybride en fonction du nombre de villes

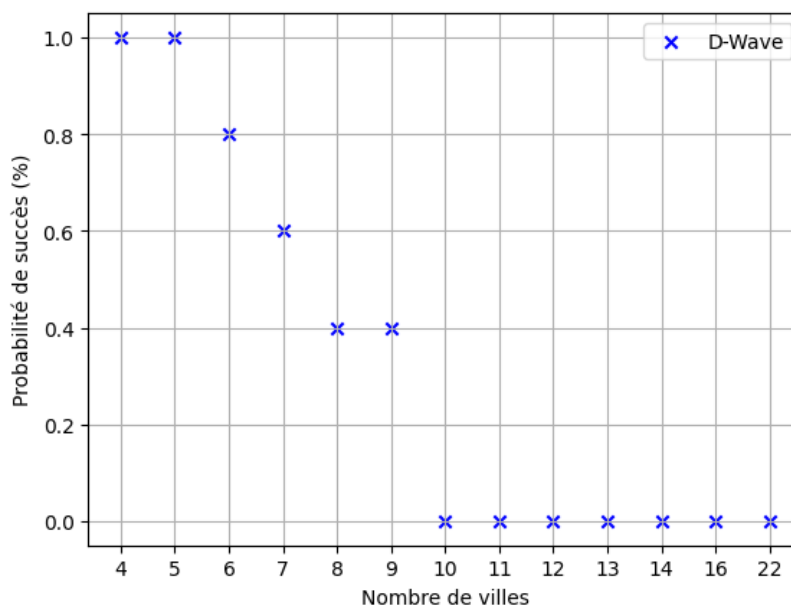
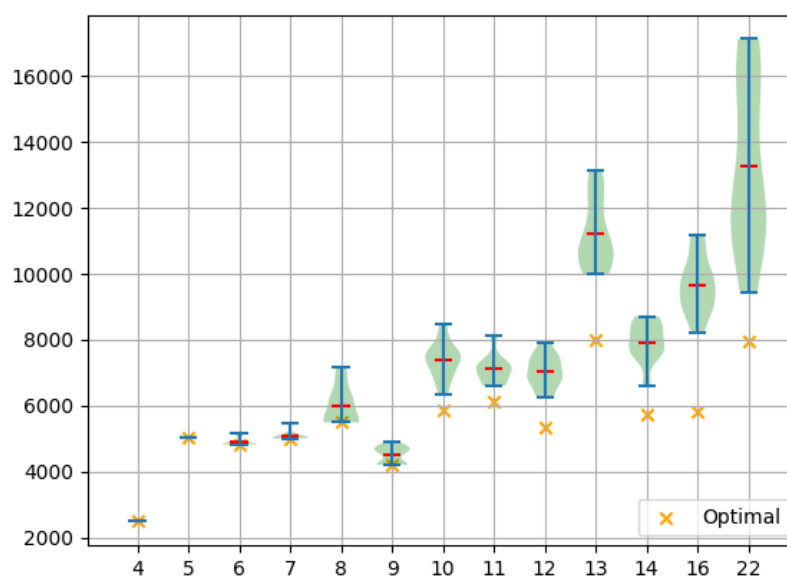


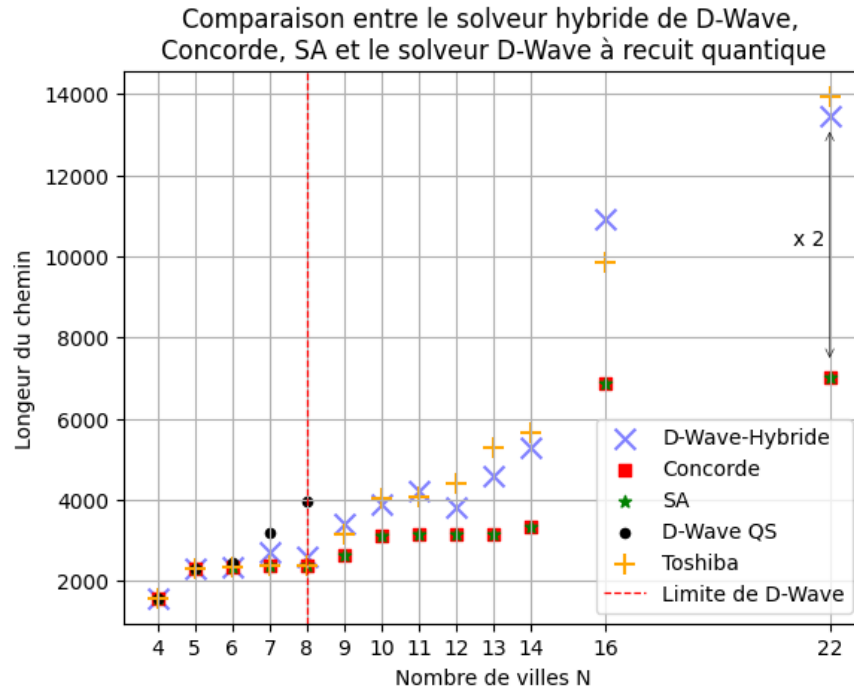
FIGURE 3.6 Distribution des solutions en fonctions du nombre de villes



Toshiba se démarque par sa capacité à résoudre des problèmes de plus grande envergure, principalement grâce à sa machine dotée d'un nombre supérieur de spins entièrement connectés.

D'un autre côté, D-Wave est également capable de repousser les limites précédentes en termes de taille de problème, mais cela nécessite généralement l'utilisation d'une approche hybride, comme discutée précédemment. La Figure 3.7 synthétise l'ensemble des résultats obtenus pour un jeu de données de petite taille. Il est à noter que la solution optimale est générée par Concorde [74]. En outre, la Figure 3.7 inclut une solution obtenue par recuit simulé, que nous avons implémentée et dont les détails sont abordés dans la section suivante. Cela démontre la diversité des approches disponibles pour la résolution de problèmes complexes.

FIGURE 3.7 Comparaison entre les différents solveurs



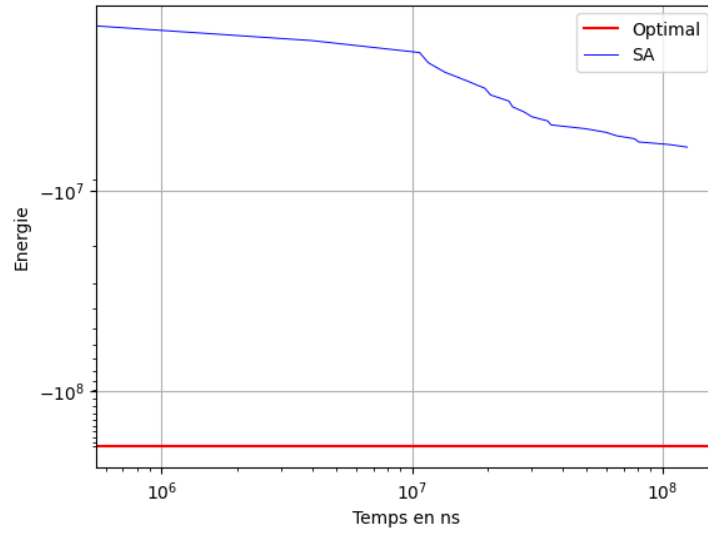
3.5.2 Recuit simulé du modèle de Ising

La résolution du modèle de Ising s'effectue efficacement à l'aide de l'algorithme du recuit simulé, tel que présenté dans la section 2.3.2. Dans cet algorithme, la manipulation des spins se fait en inversant un seul spin à la fois de manière aléatoire, comme illustré dans l'algorithme 2. Cependant, il convient de noter que cette approche s'avère inadéquate pour le TSP, en raison de la taille extrêmement vaste de l'espace des solutions. Par conséquent, pour aborder le TSP, nous nous appuyons sur deux autres méthodes alternatives, à savoir la permutation de lignes (*row_swap*) et la technique du *2-opt*. Ces approches alternatives s'avèrent essentielles pour explorer efficacement les solutions possibles dans le contexte du TSP.

3.5.3 Impact du choix du voisinage pour SA

Les figures 3.8 et 3.9, montrent respectivement l'évolution de l'énergie du système avec l'énergie optimale en fonction du temps pour l'instance avec 50 villes avec les trois versions de la fonction de *random_flip* à savoir *bit_flip*, *row_swap* et *2-opt* appliquées au SA. On constate d'après les trois figures que l'énergie se rapproche plus rapidement vers l'optimal en utilisant les deux fonctions *row_swap* et *2-opt* contrairement à la fonction *random_flip* en atteignant l'optimal pour la fonction *2-opt*.

FIGURE 3.8 Évolution de l'énergie en fonction du temps pour l'instance à 50 villes et la fonction *bit_flip*



Les fonctions *row_swap* et *2-opt* permettent de réduire l'espace de recherche de 2^{n^2} à n^2 . De plus, comme le montre la Figure 3.9, on observe que l'option *2-opt* converge plus rapidement vers l'optimum. Selon nos expériences, cette observation demeure valable jusqu'à $n = 100$. La Figure 3.10 présente l'évolution de l'énergie du système pour le cas de $n = 100$ villes. L'absence d'atteinte de l'optimal dans cette expérience s'explique par le fait que lorsque le système se rapproche de l'énergie optimale, le recuit simulé (SA) a du mal à proposer de nouvelles solutions avec de meilleures énergies en raison de la taille du problème.

3.5.4 Comparaison de SA avec Concorde

Les Figures 3.11, 3.12 et 3.13 comparent les performances de l'algorithme de recuit simulé (SA) avec l'option *2-opt* par rapport à l'algorithme Concorde. Ces comparaisons ont été effectuées pour une seule simulation avec des limites de temps de 10 secondes, 30 secondes, et 60 secondes, respectivement, pour chaque instance du problème.

FIGURE 3.9 Évolution de l'énergie en fonction du temps pour les deux options suggérées

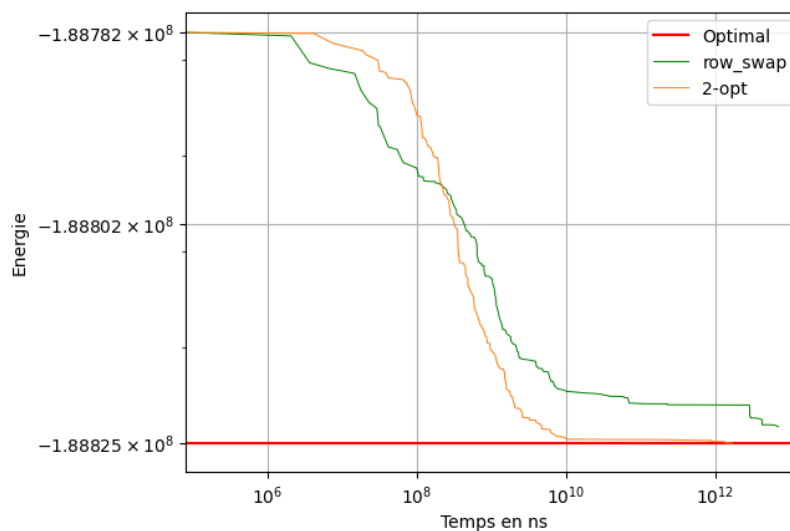
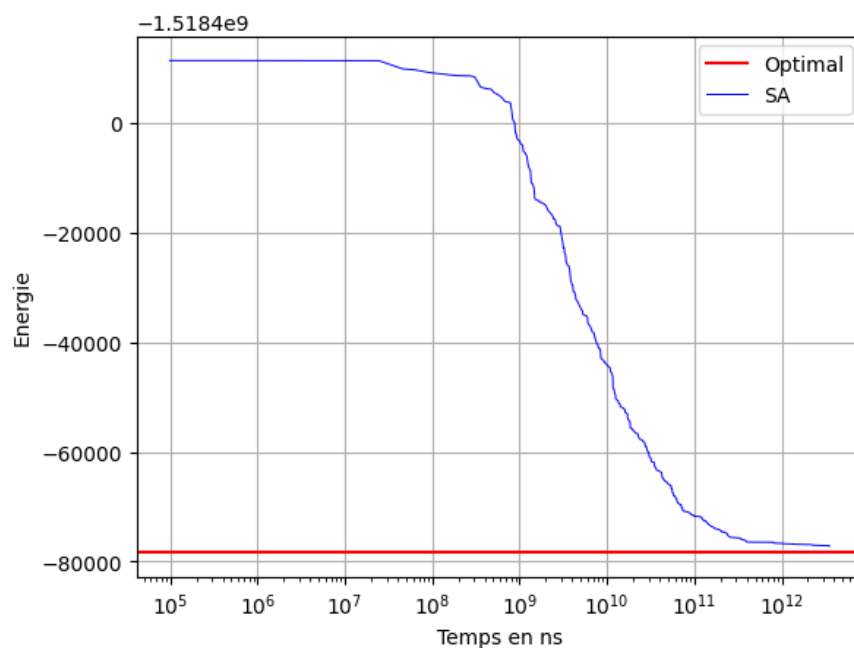
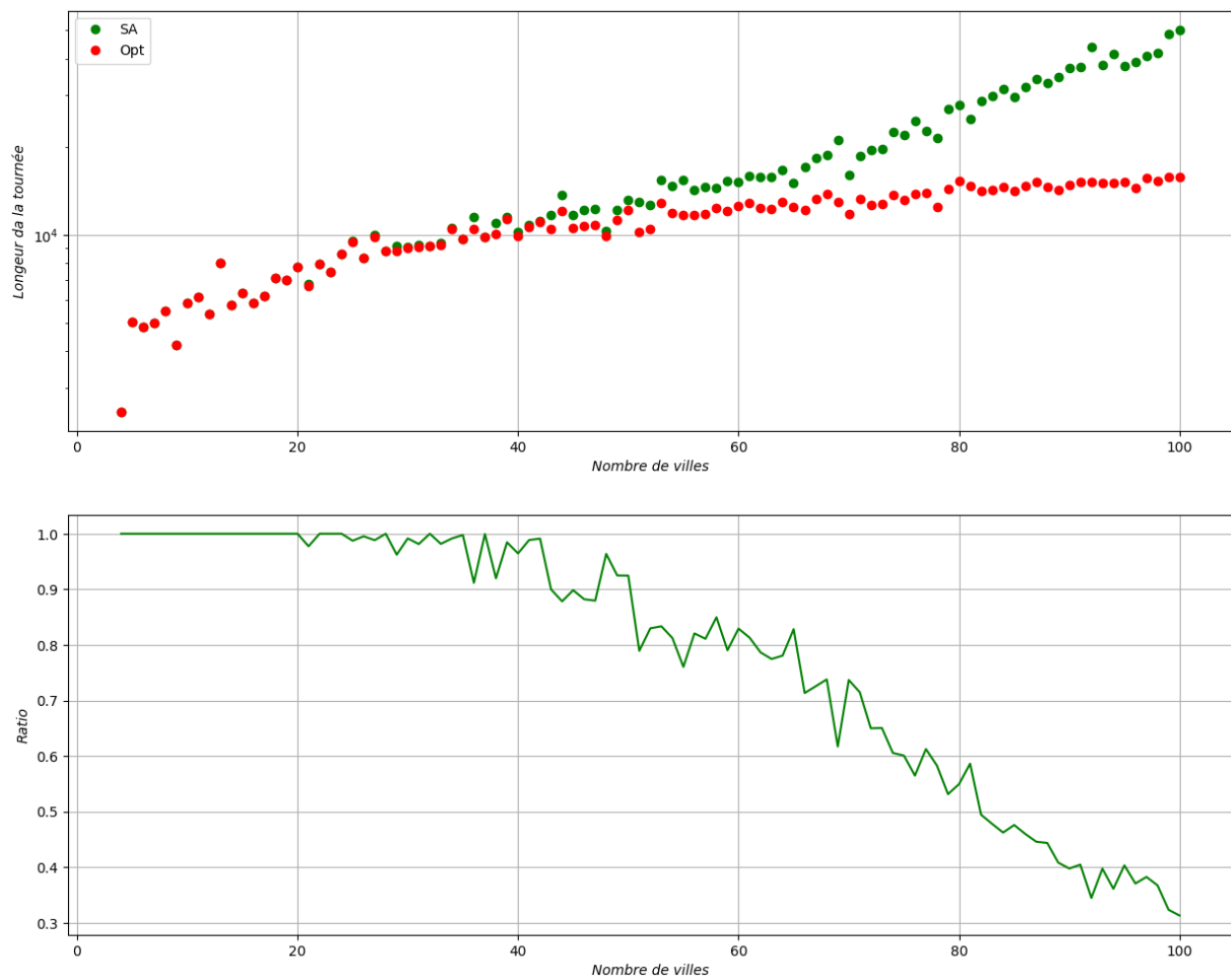


FIGURE 3.10 Évolution de l'énergie en fonction du temps pour l'instance à 100 villes et la fonction 2-opt



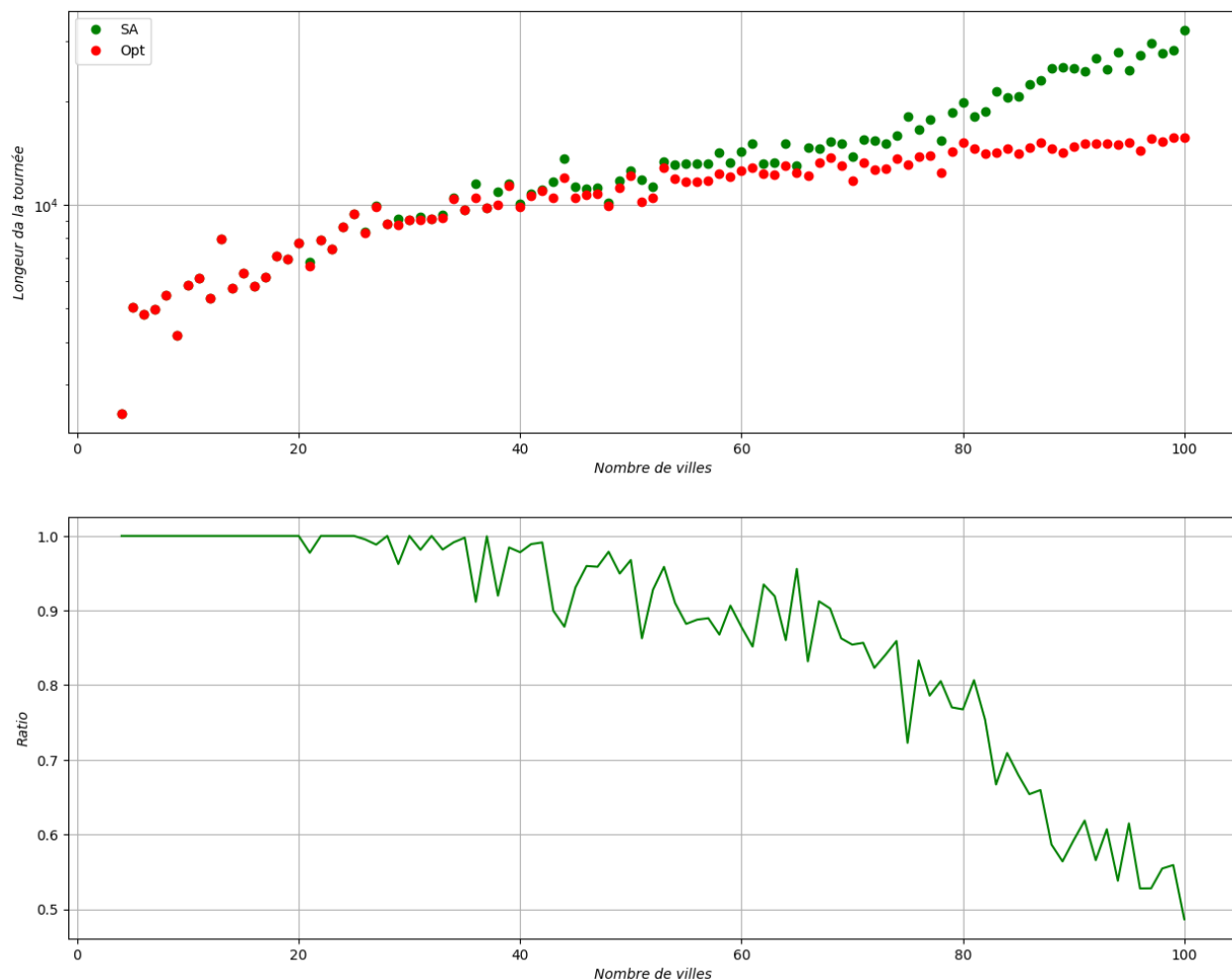
Comme le montre la Figure 3.11, la solution optimale est atteinte ou approchée à environ 10% près pour les problèmes comportant 50 villes ou moins. Au-delà de cette limite, la qualité des solutions diminue, atteignant seulement 30% de l'optimum pour $n = 100$ villes.

FIGURE 3.11 Comparaison de la solution entre le SA et l'optimal de Concorde en $\text{num_simulations} = 1$ et $\text{timeout} = 10$ s



Cependant, les performances s'améliorent considérablement à la Figure 3.12 lorsque la limite de temps est triplée, atteignant 50% de l'optimum pour $n = 100$ villes.

FIGURE 3.12 Comparaison de la solution entre le SA et l'optimal de Concorde en $\text{num_simulations} = 1$ et $\text{timeout} = 30$ s

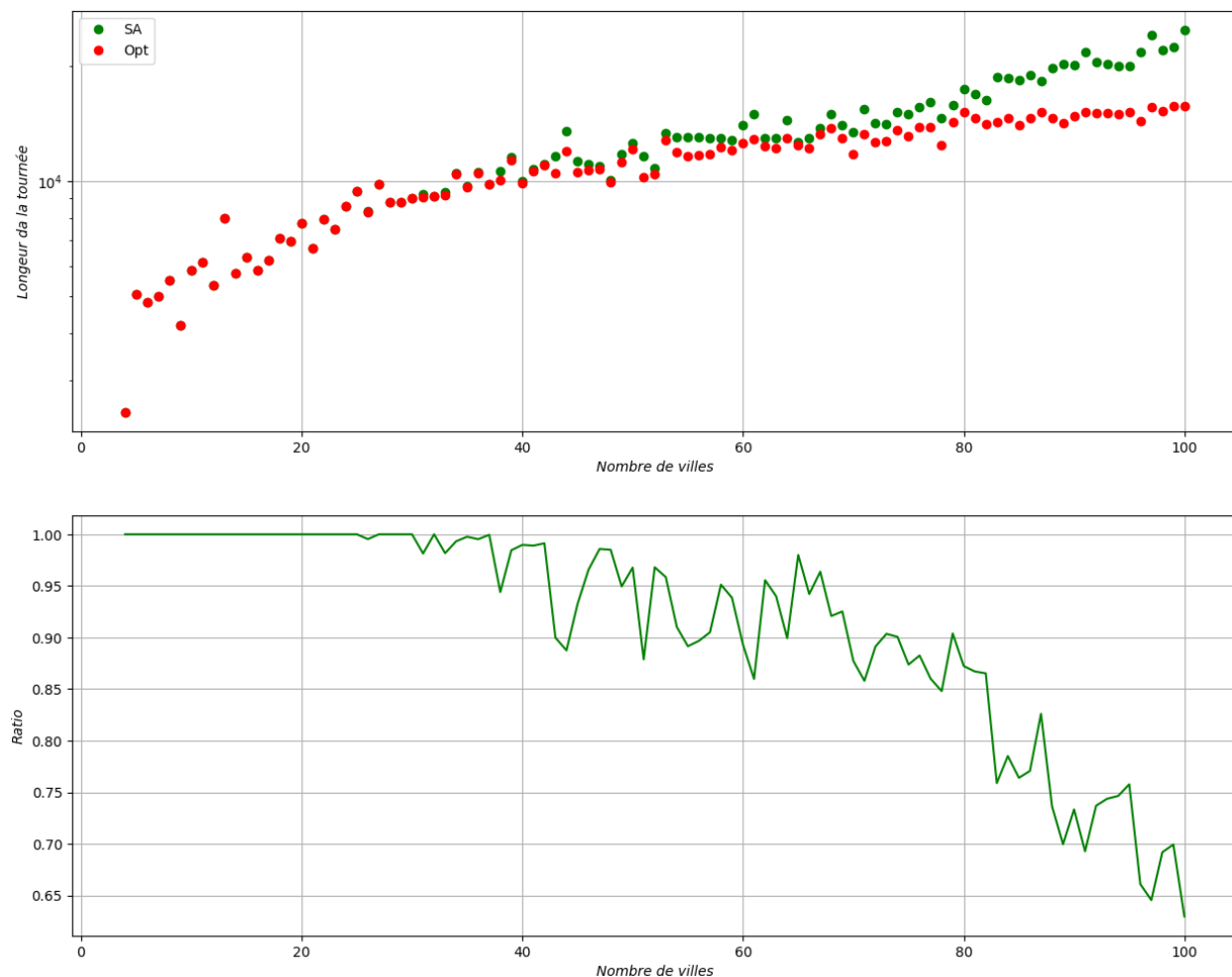


Enfin, la Figure 3.13 présente les résultats pour une durée de temps six fois plus longue que l'originale, où le système atteint 65% de l'optimum pour $n = 100$ villes.

3.5.5 Discussion générale

Ce chapitre a confirmé les constatations antérieures de la littérature, à savoir que les machines de Ising commerciales ont des limites en ce qui concerne la taille des problèmes de TSP qu'elles peuvent résoudre de manière optimale, que ce soit en raison de la topologie du réseau de la machine ou de ses capacités intrinsèques. Cependant, nous avons pu démontrer que des améliorations significatives peuvent être apportées en adoptant une stratégie de voisinage plus efficace, permettant de réduire l'espace des solutions tout en maintenant des performances acceptables.

FIGURE 3.13 Comparaison de la solution entre le SA et l'optimal de Concorde en $\text{num_simulations} = 1$ et $\text{timeout} = 60$ s



Néanmoins, notre recherche a également souligné l'importance cruciale du facteur temps dans cette approche. Cette constatation suggère un potentiel prometteur pour une accélération matérielle de la méthode proposée. En résumé, cette section met en évidence la nécessité d'explorer davantage les opportunités d'amélioration des performances des machines de Ising, tout en soulignant l'importance de combiner des approches algorithmiques optimisées avec une évolution matérielle pour relever les défis posés par les problèmes de grande taille.

CHAPITRE 4 PROBLÈME DE SÉQUENÇAGE DES TÂCHES

4.1 Définition du problème

Le JSSP est un problème d'optimisation combinatoire qui s'utilise beaucoup dans le domaine de la planification de la production. Son principe consiste à déterminer l'ordre d'exécution optimal des tâches (*job*) sur un ensemble de machines de manière à minimiser certains indicateurs tels que le temps total de traitement des toutes les tâches, le temps d'attente ou d'autres coûts liés à la production.

TABLEAU 4.1 Exemple de problème de JSSP avec 2 jobs et 2 machines. Le contenu de la table correspond à la durée des tâches. Les tâches peuvent être exécutées dans le désordre

Job/Machine	1	2
1	2	1
2	1	1

FIGURE 4.1 Exemple de solution du problème du tableau 4.1



Dans ce problème, on définit plusieurs *jobs*. Chaque *job* contient une séquence d'opérations. Chaque opération s'effectue sur une machine spécifique pendant une durée déterminée. L'objectif est de trouver la séquence optimale des opérations tout en respectant les contraintes de disponibilités de ressources et les contraintes de précédence dans chaque *job* ainsi que minimisant le coût de la solution.

Ce problème est complexe en raison de la nécessité de prendre en compte toutes les contraintes du problème comme la précédence et la disponibilité. Il existe de nombreuses variantes du JSSP, mais dans ce mémoire, seule la variante standard du JSSP sera considérée.

4.2 Formulation du problème

4.2.1 Formulation générale

Plusieurs modèles du JSSP ont été développés dans la littérature : [82] décrit deux types de modèles, à savoir : le *modèle disjonctif*, où les variables de décision prennent comme valeur le temps de début de chaque opération ; et le *modèle indexé par le temps*, où les variables de décision sont binaires. On se limitera au *modèle indexé par le temps* dans ce mémoire, en raison de son expressivité par le modèle Ising.

Dans ce modèle on définit la variable de décision $x_{i,j,t}$ telle que :

$$x_{i,j,t} = \begin{cases} 1 & \text{si job } j \text{ commence au temps } t \text{ à la machine } i \\ 0 & \text{sinon} \end{cases} \quad (4.1)$$

Le problème du JSSP peut s'exprimer ainsi :

$$\min \quad C_{max} \quad (4.2)$$

$$tq \quad \sum_{t \in T} x_{i,j,t} = 1 \quad j \in J, i \in M \quad (4.3)$$

$$\sum_{t \in T} (t + p_{ij}) x_{i,j,t} \leq C_{max} \quad j \in J, i \in M \quad (4.4)$$

$$\sum_{j \in J} \sum_{t' \in T_{i,j,t}} x_{i,j,t'} \leq 1 \quad i \in M, t \in T, T_{i,j,t} = \{t - p_{ij} + 1, \dots, t\} \quad (4.5)$$

$$\sum_{t \in T} (t + p_{\sigma_{h-1}^j}) x_{\sigma_{h-1}^j, j, t} \leq \sum_{t \in T} t \cdot x_{\sigma_h^j, j, t} \quad j \in J, h = 2, \dots, m \quad (4.6)$$

$$x_{i,j,t} \in \{0, 1\} \quad j \in J, i \in M, t \in T \quad (4.7)$$

où J l'ensemble des jobs, M l'ensemble des machines, T l'ensemble des valeurs possibles de temps de début. Ce dernier est un ensemble borné supérieurement par le temps maximale que peuvent prendre toutes les opérations exécutées une après l'autre. Le modèle définit la variable σ_h^j par la h -ième opération du job j .

L'équation 4.2 représente la fonction objectif visant à minimiser le temps total d'exécution de toutes les opérations. La contrainte 4.3 stipule qu'un job j doit s'exécuter une seule fois sur chaque machine i . La contrainte 4.4 limite le temps total d'exécution à ne pas dépasser C_{max} . La contrainte 4.5 garantit qu'une machine ne peut exécuter qu'une seule opération à

la fois à tout moment. Enfin, la contrainte 4.6 est la contrainte de précédence, qui exige que les opérations au sein de chaque job j soient exécutées dans l'ordre spécifié.

4.2.2 Formulation QUBO

La formulation repose sur une représentation simple indexée sur le temps. On définit l'ensemble $\{O_1, O_2, \dots, O_n\}$ de toutes les opérations de toutes les *jobs*. Pour chaque opération, on attribue une collection de variables binaires qui correspondent aux nombreux temps de démarrage discrets potentiels que le processus peut avoir.

$$x_{i,t} = \begin{cases} 1, & \text{opération } O_i \text{ commence au temps } t \\ 0, & \text{sinon} \end{cases} \quad (4.8)$$

Ici le temps t est bornée supérieurement par la somme de toutes les durées de toutes les tâches.

La contrainte pour chaque opération de s'exécuter une et une seule fois est exprimée par :

$$\sum_t x_{i,t} = 1, \quad \forall i \implies \sum_i \left(\sum_t x_{i,t} - 1 \right)^2 \quad (4.9)$$

La contrainte de disponibilité pour chaque machine, soit qu'une machine peut exécuter une seule opération à la fois, est exprimée par :

$$\sum_{(i,t,k,t') \in R_m} x_{i,t} x_{k,t'} = 0 \quad \forall m \implies \sum_m \left(\sum_{(i,t,k,t') \in R_m} x_{i,t} x_{k,t'} \right), \quad (4.10)$$

où $R_m = A_m \cup B_m$ et

$$A_m = \{(i, t, k, t') : (i, k) \in \mathcal{I}_m \times \mathcal{I}_m, \quad i \neq k, t \geq 0, T \geq t', 0 < t' - t < p_i\} \quad (4.11)$$

$$B_m = \{(i, t, k, t') : (i, k) \in \mathcal{I}_m \times \mathcal{I}_m, \quad i < k, t = t', p_i > 0, p_j > 0\} \quad (4.12)$$

$$\mathcal{I}_m = \{i : \text{la tâche } i \text{ s'exécute sur la machine } m\} \quad (4.13)$$

A_m est défini de manière à ce que la contrainte interdise à l'opération O_j de commencer à l'instant t' si une autre opération O_i est encore en cours, ce qui se produit si O_i a commencé à l'instant t et si $t' - t$ est inférieur à p_i .

B_m est défini de manière à ce que deux travaux ne puissent pas démarrer en même temps, à moins qu'au moins l'un d'entre eux ait un temps d'exécution égal à zéro.

La contrainte de précédence dans chaque *job*.

$$\sum_{k_{n-1} < i < k_n, t+p_i > t'} x_{i,t} x_{i+1,t'}, \quad \forall n \implies \sum_n \left(\sum_{k_{n-1} < i < k_n, t+p_i > t'} x_{i,t} x_{i+1,t'} \right) \quad (4.14)$$

Finalement la formulation en QUBO du JSSP s'écrit

$$\mathcal{H} = A \sum_i \left(\sum_t x_{i,t} - 1 \right)^2 \quad (4.15)$$

$$+ B \sum_m \left(\sum_{(i,t,k,t') \in R_m} x_{i,t} x_{k,t'} \right) \quad (4.16)$$

$$+ C \sum_n \left(\sum_{k_{n-1} < i < k_n, t+p_i > t'} x_{i,t} x_{i+1,t'} \right) \quad (4.17)$$

où A , B et C sont des hyperparamètres strictement positifs.

L'équation du Hamiltonien peut être simplifiée en développant le carré de l'équation 4.9 en utilisant la formule

$$\left(\sum_i^n x_i \right)^2 = \left(\sum_i^n x_i \right) \left(\sum_i^n x_i \right) = \sum_i^n \sum_j^n x_i x_j \quad (4.18)$$

L'équation 4.17 devient alors

$$\mathcal{H} = (1) \quad - 2A \sum_i \left(\sum_t x_{i,t} \right) + A \sum_i \left(\sum_t \sum_{t'} x_{i,t} x_{i,t'} \right) \quad (4.19)$$

$$(2) \quad + B \sum_m \left(\sum_{(i,t,k,t') \in R_m} x_{i,t} x_{k,t'} \right) \quad (4.20)$$

$$(3) \quad + C \sum_n \left(\sum_{k_{n-1} < i < k_n, t+p_i > t'} x_{i,t} x_{i+1,t'} \right) \quad (4.21)$$

La fonction objectif pour minimiser le coût total d'exécution est composée de deux termes.

Le premier terme consiste à pénaliser le système s'il y a du temps qui n'est pas utilisé au début de chaque job :

$$\sum_{i \in F} \sum_t tx_{i,t} \quad (4.22)$$

où

$$F = \{i \text{ tq } i \text{ représente la première opération de chaque job}\} \quad (4.23)$$

Le deuxième terme quant à lui permet de d'exécuter toutes les dernières opérations pour chaque job le plus tôt possible :

$$\sum_{i \in L} \sum_t tx_{i,t} \quad (4.24)$$

où

$$L = \{i \text{ tq } i \text{ représente la dernière opération de chaque job}\} \quad (4.25)$$

L'hamiltonienne 4.17 devient alors

$$\mathcal{H} = (1) \quad - 2A \sum_i \left(\sum_t x_{i,t} \right) + A \sum_i \left(\sum_t \sum_{t'} x_{i,t} x_{i,t'} \right) \quad (4.26)$$

$$(2) \quad + B \sum_m \left(\sum_{(i,t,k,t') \in R_m} x_{i,t} x_{k,t'} \right) \quad (4.27)$$

$$(3) \quad + C \sum_n \left(\sum_{k_{n-1} < i < k_n, t+p_i > t'} x_{i,t} x_{i+1,t'} \right) \quad (4.28)$$

$$(4) \quad + D \sum_{i \in F} \sum_t tx_{i,t} \quad (4.29)$$

$$(5) \quad + E \sum_{i \in L} \sum_t tx_{i,t} \quad (4.30)$$

où A , B , C , D et E sont des hyperparamètres strictement positifs.

4.3 Implémentation avec des approches d’inspiration quantique

Comme pour le TSP, nous avons exécuté le problème du JSSP sur D-Wave, Toshiba, et en utilisant le recuit simulé. D-Wave fournit une implémentation pour résoudre le JSSP mais de manière hybride. L’implémentation commence par résoudre le problème avec un algorithme classique de type glouton et ensuite essaye d’améliorer la solution avec des méthodes quantiques. Pour l’implémentation avec SA et Toshiba, on se limite à des problèmes de petite taille car la taille du modèle d’Ising explose rapidement. En effet, pour un problème de JSSP de n jobs et m machines, le modèle contient $max_time \times m \times n$, ce qui explique la limitation de cette implémentation.

Pour l’approche par recuit simulé, nous avons considéré différentes façons d’explorer l’espace des solutions. La première est l’approche naïve du *random_flip* qui consiste à inverser la valeur d’un spin dans le modèle d’Ising. La deuxième implémentation consiste quant à elle à échanger les valeurs de deux spins pour un même job choisi au hasard. Cette dernière a été combinée avec une méthode d’élimination de variable (*variable pruning*) inspirée du papier [67]. Celle-ci permet de réduire l’espace de recherche pour le SA. Elle consiste à éliminer les cas où les opérations dans le même job ne soient pas exécutées avant que les opérations antérieures ne soient entièrement complétées.

4.3.1 Jeu de données

Étant donné la relative complexité du problème du JSSP, nous avons opté pour la création manuelle d’un jeu de données spécifiquement conçu pour nos expérimentations. Le Tableau 4.2 répertorie les six instances qui ont été prises en considération lors des tests empiriques. Pour chaque instance, ce tableau fournit le nombre de jobs et de machines impliquées. Cette approche de génération manuelle des données nous a permis de contrôler précisément les caractéristiques des instances, garantissant ainsi une bonne base pour l’évaluation des performances de notre algorithme d’amélioration du recuit simulé. La taille des problèmes va croissante.

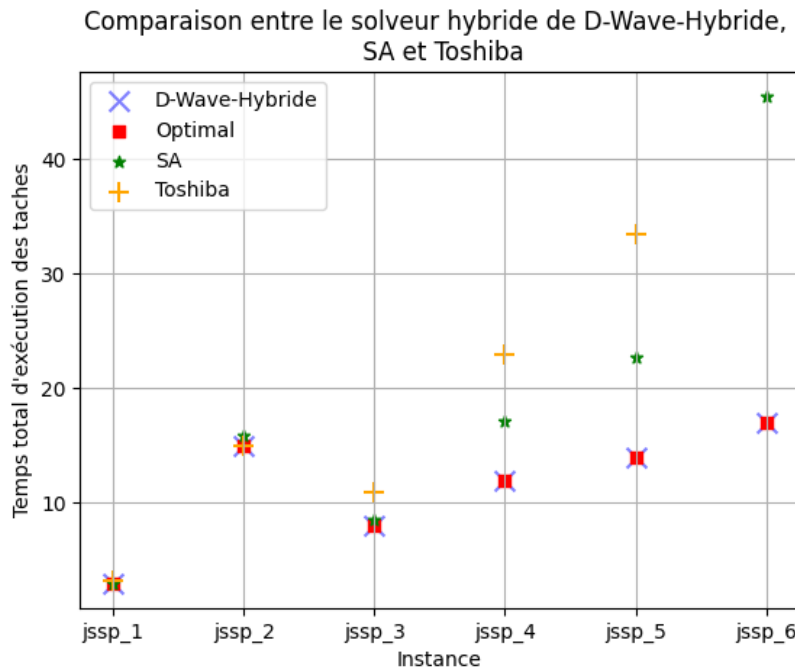
TABLEAU 4.2 Description des instances du jeu de données utilisé pour évaluer le JSSP

Machine/Job	Nb de jobs	Nb de machines	Nb de spins
jssp_1	2	2	20
jssp_2	2	3	144
jssp_3	3	3	162
jssp_4	3	4	276
jssp_5	4	4	528
jssp_6	5	5	1275

4.3.2 Exécution sur D-Wave et Toshiba

La Figure 4.2 présente les résultats de la résolution des six instances du JSSP en utilisant les plateformes D-Wave et Toshiba, ainsi que notre implémentation de SA du modèle de Ising. Pour chaque instance, 10 simulations sont exécutées. L'optimum global est obtenu grâce à Google OR.

FIGURE 4.2 Comparaison entre les différents solveurs D-Wave, SA et Toshiba



Concernant D-Wave, il est important de noter qu'il s'agit d'une solution hybride exploitant un code fourni par la compagnie. Cette approche hybride de D-Wave a démontré une constance remarquable en produisant systématiquement le résultat optimal pour toutes les instances. En revanche, la méthode de Toshiba, reposant entièrement sur le modèle Ising, montre une

légère déviation progressive par rapport à l'optimum global et présente même un résultat aberrant pour l'instance `jssp_6`.

Quant à notre implémentation du recuit simulé (SA), elle produit des résultats intermédiaires, montrant une déviation croissante de l'optimum global à mesure que la complexité des instances augmente. Ces résultats mettent en évidence les avantages et les limites des différentes approches pour la résolution du JSSP, soulignant le potentiel des solutions hybrides basées sur D-Wave pour maintenir la qualité de la solution sur des instances complexes. Cependant, ils mettent également en lumière la nécessité d'explorer davantage les stratégies d'amélioration pour les méthodes basées sur le modèle Ising.

4.4 Recuit simulé du modèle de Ising

Nous avons effectué une comparaison approfondie entre notre implémentation de l'algorithme de recuit simulé (SA) et deux approches spécifiques, à savoir le *random_flip* et l'"élimination de variable". Les résultats de cette comparaison sont illustrés dans les Figures 4.3 et 4.4, qui représentent l'évolution de l'énergie pour l'instance de test `jssp_5`. L'analyse de ces graphiques révèle clairement que l'approche d'élimination de variable offre des avantages significatifs en termes de temps de convergence et d'optimisation des résultats.

FIGURE 4.3 Évolution de l'énergie par rapport au temps pour `jssp_5` avec un bit flip

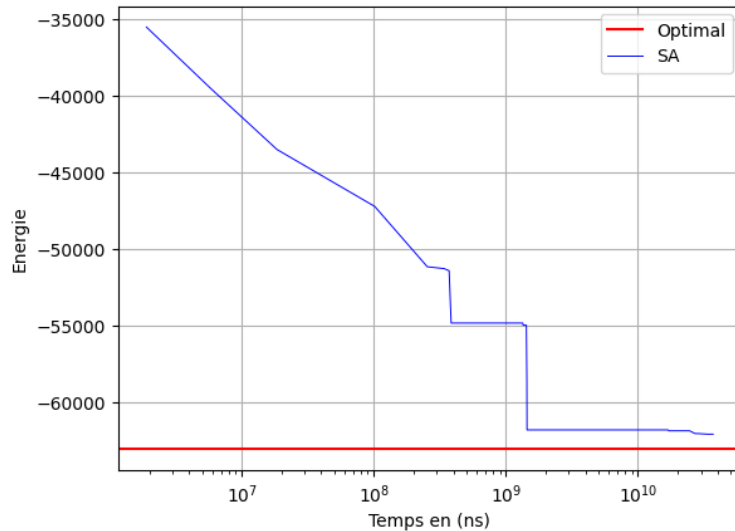
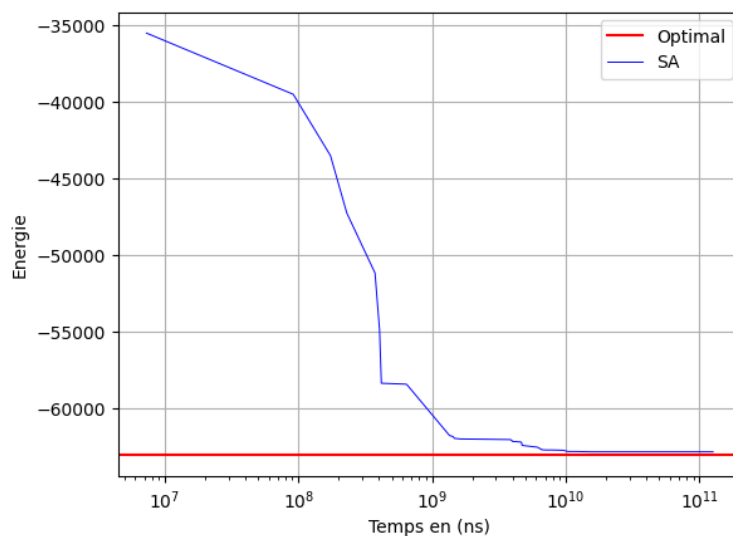


FIGURE 4.4 Évolution de l'énergie par rapport au temps pour jssp_5 avec élimination de variables



Lorsque nous avons recours à l'élimination de variable, nous observons une amélioration marquée du processus de convergence, avec des itérations se rapprochant plus rapidement de la solution optimale. Cette amélioration se traduit également par une meilleure qualité de la solution obtenue à la fin du processus d'optimisation. Ces résultats démontrent l'efficacité de l'élimination de variable en tant que technique d'amélioration pour l'algorithme de recuit simulé dans le contexte du JSSP, ouvrant ainsi la voie à des améliorations potentielles dans la résolution de problèmes complexes de planification de la production.

Les figures 4.5 et 4.6 montrent la différence entre deux solutions trouvées par les deux versions de fonctions *random_flip* pour jssp_5.

FIGURE 4.5 Solution avec flip aléatoire

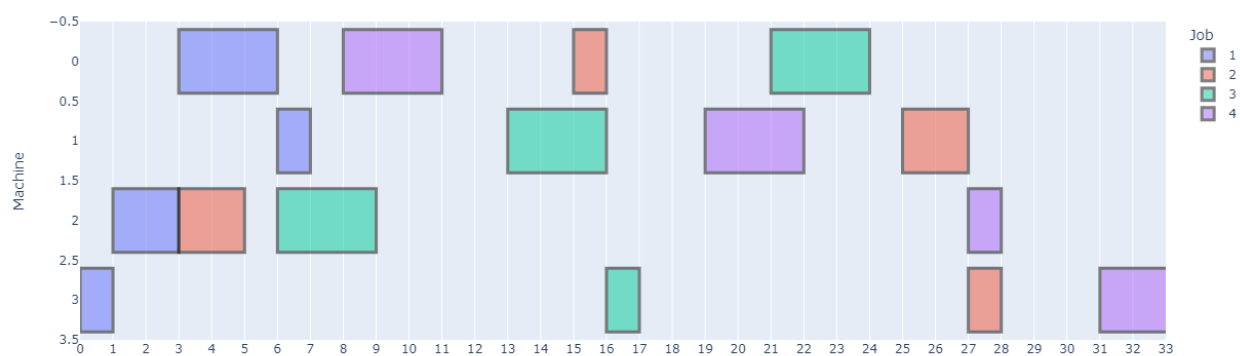
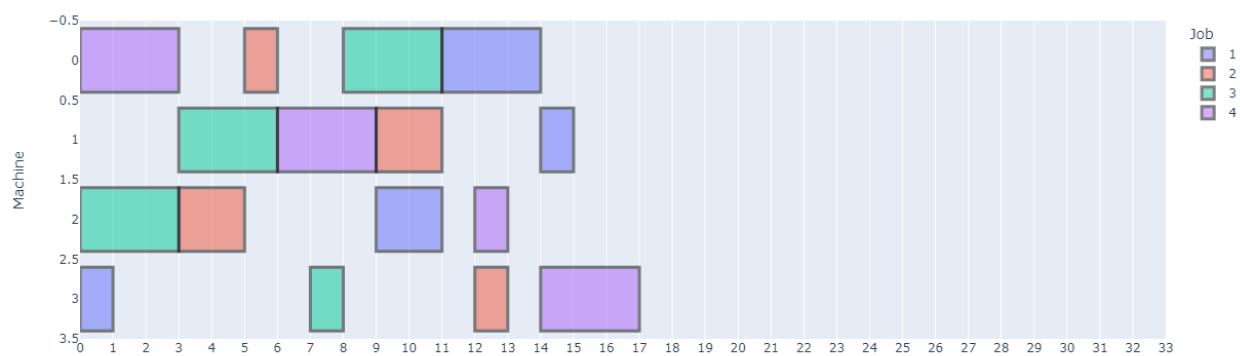


FIGURE 4.6 Solution avec élimination de variables



Les figures 4.7 et 4.8 confirment ce résultat pour l'instance jssp_6. On remarque dans la figure 4.8 que la solution optimale n'est pas atteinte malgré l'élimination de variables. Ceci est expliqué par la taille du problème qui induit un grand nombre de variables dans le modèle d'Ising.

FIGURE 4.7 Évolution de l'énergie par rapport au temps pour jssp_6 avec bit flip

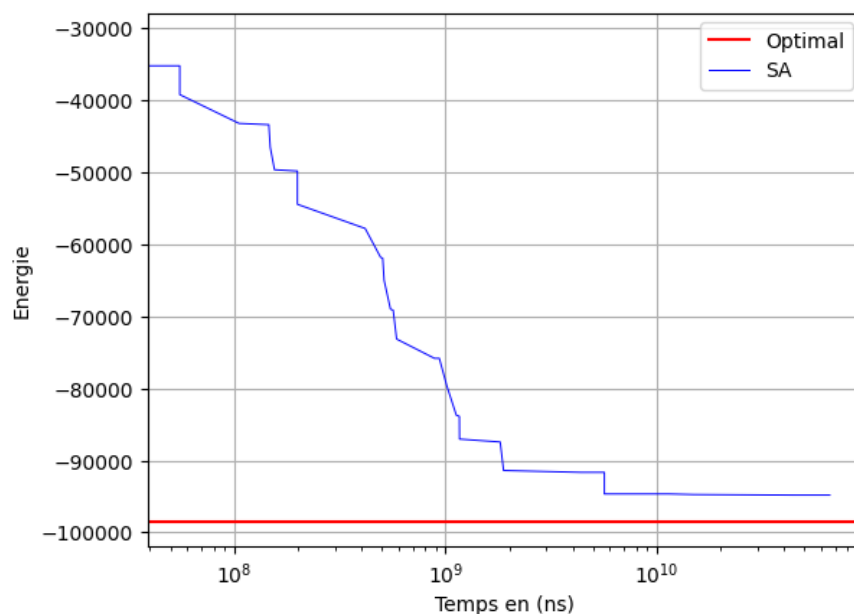
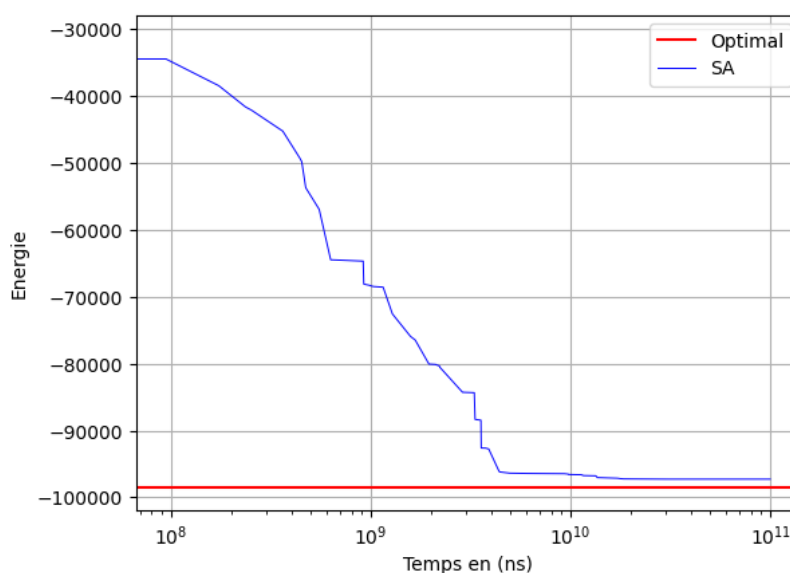


FIGURE 4.8 Évolution de l'énergie par rapport au temps jssp_6 avec élimination de variables



4.5 Discussion générale

Ce chapitre a permis de considérer le problème de JSSP et sa solution par modèle de Ising. JSSP, bien qu'étant un problème d'optimisation combinatoire connu, s'avère encore plus complexe que le TSP. Actuellement, il semble que le JSSP soit en dehors de la portée des machines de Ising commerciales disponibles. Cela suggère la nécessité de repousser les limites de cette technologie pour résoudre des problèmes d'une telle envergure.

Ce chapitre a également offert des contributions en proposant des améliorations à l'algorithme de recuit simulé du modèle de Ising pour le JSSP. Ces améliorations ont montré des signes prometteurs d'amélioration des performances par rapport aux méthodes traditionnelles. Néanmoins, il convient de noter que des travaux complémentaires seront nécessaires pour adapter l'approche de Ising à des instances réalistes du JSSP. Les défis persistants liés à la résolution de ce problème complexe devraient encourager la poursuite de la recherche dans ce domaine.

CHAPITRE 5 CONCLUSION

5.1 Retour synthétique sur les travaux de maîtrise

Ce travail de mémoire a consisté à évaluer l'utilisation du modèle d'Ising pour la résolution des problèmes d'optimisation combinatoire, notamment au travers de machines de Ising commerciales. Deux exemples de problèmes ont été choisis, à savoir le TSP et le JSSP. Pour chacun de ces problèmes, une définition et une formulation générale ont été expliquées, suivies d'une modélisation dans le modèle d'Ising. Différentes approches pour la résolution de ces deux problèmes ont été explorées, parmi lesquelles le SA. Des tests de performances de l'algorithme ont été réalisés, ainsi qu'une comparaison de la qualité des solutions obtenues par rapport à d'autres approches. Des méthodes d'accélération du SA ont également été présentées et comparées.

5.2 Limitations de la recherche

Les tests effectués au cours de cette étude étaient relativement limités, en particulier en ce qui concerne JSSP. Il serait intéressant d'étendre ces tests à un éventail plus large d'instances de JSSP, y compris des cas plus complexes, afin d'obtenir une vision plus complète des performances de l'algorithme de recuit simulé (SA).

De plus, il est important de noter que ce travail a été principalement un travail de prototype, et le langage Python a été utilisé pour les implémentations. Cette décision a entraîné des limitations en termes de temps d'exécution et d'étendue des tests réalisés. Cependant, des optimisations au niveau du programme pourraient permettre d'élargir considérablement l'étendue des expériences, notamment en utilisant des langages de programmation plus performants. Cette perspective ouvrirait la porte à des tests sur des instances de problèmes encore plus grandes et complexes, renforçant ainsi la robustesse des résultats obtenus.

5.3 Améliorations futures

Parmi les améliorations futures à apporter au travail présenté dans ce mémoire, on peut envisager l'adoption de méthodes parallèles. Tel est le cas dans la littérature où l'on parle de la méthode *parallel tempering*. Cette approche utilise le SA comme méthode de base et permet de lancer en parallèle plusieurs instances de SA tout en utilisant une méthode d'échange de températures. La revue de littérature montre théoriquement que cette méthode converge plus

rapidement vers une solution de haute qualité.

Une autre amélioration possible consiste à réécrire les implémentations dans un autre langage de programmation, tel que le *C++* ou le *Rust*, par exemple. En effet, l'utilisation de tels langages permet d'accélérer le calcul et le traitement en raison de leur gestion fine des ressources et de leur proximité avec le matériel. Cependant, il convient de noter que cela pourrait également introduire une complexité accrue en termes de développement.

Une autre piste d'amélioration réside dans l'exploration des accélérations matérielles au travers de dispositifs tels que les FPGA ou les GPU. L'utilisation de ces accélérations matérielles pourrait considérablement réduire les temps de calcul, ouvrant ainsi la possibilité de traiter des problèmes de plus grande envergure. De plus, cela permettrait d'élargir la portée des expérimentations et de renforcer la robustesse des résultats obtenus.

RÉFÉRENCES

- [1] J. L. Hennessy, D. A. Patterson et K. Asanović, *Computer architecture : a quantitative approach*, 5^e éd., ser. Morgan Kaufmann series in computer architecture and design. Waltham, MA : Morgan Kaufmann, 2012.
- [2] J. L. Hennessy et D. A. Patterson, “A new golden age for computer architecture,” *Commun. ACM*, vol. 62, n^o. 2, p. 48–60, jan 2019.
- [3] K. Svozil, “First international conference on unconventional models of computation umc’98. an unconventional review,” Jan 1998, arXiv :quant-ph/9801060. [En ligne]. Disponible : <http://arxiv.org/abs/quant-ph/9801060>
- [4] M. Fellous-Asiani *et al.*, “Limitations in quantum computing from resource constraints,” *PRX Quantum*, vol. 2, n^o. 4, p. 040335, Nov 2021.
- [5] M. A. . Nielsen et I. L. . Chuang, *Quantum computation and quantum information*, 10^e éd. Cambridge : Cambridge University Press, 2010.
- [6] N. Mohseni, P. L. McMahon et T. Byrnes, “Ising machines as hardware solvers of combinatorial optimization problems,” *Nature Reviews Physics*, vol. 4, n^o. 6, p. 363–379, may 2022.
- [7] E. Ising, “Contribution to the theory of ferromagnetism,” *Journal for Physics*, vol. 31, n^o. 1, p. 253–258, Feb 1925.
- [8] N. Dattani, S. Szalay et N. Chancellor, “Pegasus : The second connectivity graph for large-scale quantum annealing hardware,” Jan 2019, arXiv :1901.07636 [quant-ph]. [En ligne]. Disponible : <http://arxiv.org/abs/1901.07636>
- [9] H. Goto, K. Tatsumura et A. R. Dixon, “Combinatorial optimization by simulating adiabatic bifurcations in nonlinear hamiltonian systems,” *Science Advances*, vol. 5, n^o. 4, p. eaav2372, Apr 2019.
- [10] M. Ayodele, “Comparing the digital annealer with classical evolutionary algorithm,” Jun 2022, arXiv :2205.13586 [cs]. [En ligne]. Disponible : <http://arxiv.org/abs/2205.13586>
- [11] A. Dan *et al.*, “Clustering approach for solving traveling salesman problems via ising model based solver,” dans *2020 57th ACM/IEEE Design Automation Conference (DAC)*, 2020, p. 1–6.
- [12] A. Lucas, “Ising formulations of many np problems,” *Frontiers in Physics*, vol. 2, 2014.
- [13] T. Zhang et J. Han, “Efficient traveling salesman problem solvers using the ising model with simulated bifurcation,” dans *Proceedings of the 2022 Conference & Exhibition on*

- Design, Automation and Test in Europe*, ser. DATE '22. Leuven, BEL : European Design and Automation Association, Mar 2022, p. 548–551.
- [14] K. Kurowski *et al.*, “Hybrid quantum annealing heuristic method for solving job shop scheduling problem,” dans *Computational Science – ICCS 2020*, ser. Lecture Notes in Computer Science, V. V. Krzhizhanovskaya *et al.*, édit. Cham : Springer International Publishing, 2020, p. 502–515.
 - [15] B. Apolloni, N. Cesa-Bianchi et D. De Falco, *A Numerical Implementation of “quantum Annealing”*, ser. Dipartimento di Scienze dell’ Informazione : Rapporto interno. Univ., Forschungszentrum BiBoS Bielefeld-Bochum Stochastik, 1988. [En ligne]. Disponible : <https://cesa-bianchi.di.unimi.it/Pubblicazioni/quantumAnnealing.pdf>
 - [16] B. Heim, T. F. Rønnow, S. V. Isakov et M. Troyer, “Quantum versus classical annealing of ising spin glasses,” *Science*, vol. 348, n°. 6231, p. 215–217, 2015.
 - [17] S. Srinivasan, G. Gordon et B. Boots, “Learning hidden quantum markov models,” 2017.
 - [18] M. Franzese et A. Iuliano, *Hidden Markov Models*. Oxford : Academic Press, Jan 2019, p. 753–762.
 - [19] J. Kempe, “Quantum random walks : An introductory overview,” *Contemporary Physics*, vol. 44, n°. 4, p. 307–327, Jul 2003.
 - [20] K. Kitaura, R. Matsuo et H. Ohsaki, “Random walk on a graph with vicinity avoidance,” dans *International Conference on Information Networking (ICOIN)*, Jan 2022, p. 232–237.
 - [21] S. Shi *et al.*, “Quantum-inspired complex convolutional neural networks,” Oct 2021, arXiv :2111.00392 [quant-ph]. [En ligne]. Disponible : <http://arxiv.org/abs/2111.00392>
 - [22] L. Beaudin, *A Review of the Potts Model : Its connection to the Tutte Polynomial and its Application to Complex Experiments*. Rose-Hulman Undergraduate Mathematics Journal, 2007, vol. 8, n°. 1.
 - [23] A. S. T. Pires, “The Heisenberg model,” dans *Theoretical Tools for Spin Models in Magnetic Systems*, ser. 2053-2563. IOP Publishing, 2021, p. 1–16.
 - [24] E. Lieb, T. Schultz et D. Mattis, “Two soluble models of an antiferromagnetic chain,” *Annals of Physics*, vol. 16, n°. 3, p. 407–466, 1961.
 - [25] K. Harada et N. Kawashima, “Kosterlitz-thouless transition of quantum XY model in two dimensions,” *Journal of the Physical Society of Japan*, vol. 67, n°. 8, p. 2768–2776, 1998.
 - [26] A. Bovier *et al.*, *Statistical Mechanics of Disordered Systems : A Mathematical Perspective*. Cambridge, United Kingdom : Cambridge University Press, 2006.

- [27] A. A. Opoku, K. O. Edusei et R. K. Ansah, “A conditional curie–weiss model for stylized multi-group binary choice with social interaction,” *Journal of Statistical Physics*, vol. 171, n^o. 1, p. 106–126, Apr 2018.
- [28] N. Metropolis *et al.*, “Equation of State Calculations by Fast Computing Machines,” *The Journal of Chemical Physics*, vol. 21, n^o. 6, p. 1087–1092, 12 2004.
- [29] V. Cevher et J. McClellan, “Fast initialization of particle filters using a modified metropolis-hastings algorithm : mode-hungry approach,” dans *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 2004, p. ii–129.
- [30] Y. Dong et Z. Huang, “An improved noise quantum annealing method for tsp,” *International Journal of Theoretical Physics*, vol. 59, n^o. 12, p. 3737–3755, Dec 2020.
- [31] Z. I. Tabi *et al.*, “Evaluation of quantum annealer performance via the massive mimo problem,” *IEEE Access*, vol. 9, p. 131658–131671, 2021.
- [32] R. Hamerly *et al.*, “Experimental investigation of performance differences between coherent ising machines and a quantum annealer,” *Science Advances*, vol. 5, n^o. 5, p. eaau0823, 2019.
- [33] R. Harris *et al.*, “Experimental demonstration of a robust and scalable flux qubit,” *Phys. Rev. B*, vol. 81, p. 134510, Apr 2010.
- [34] S. Yarkoni, E. Raponi, T. Bäck et S. Schmitt, “Quantum annealing for industry applications : introduction and review,” *Reports on Progress in Physics*, vol. 85, n^o. 10, p. 104001, Sep 2022.
- [35] J. Vallejo Benítez Cano, “D-wave systems quantum computing : State-of-the-art and performance comparison with classical computing,” p. 56, 2021.
- [36] D.-W. Systems. (2023) D-wave topology. Accédé : 2023/12/20. [En ligne]. Disponible : <https://docs.ocean.dwavesys.com/en/stable/concepts/topology.html>
- [37] S. Matsubara *et al.*, “Digital annealer for high-speed solving of combinatorial optimization problems and its applications,” dans *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan 2020, p. 667–672.
- [38] W. Gilks, S. Richardson et D. Spiegelhalter, édit., *Markov Chain Monte Carlo in Practice*. New York : Chapman and Hall/CRC, Nov 1995.
- [39] S. P. Meyn et R. L. Tweedie, *Markov Chains and Stochastic Stability*, 2^e éd. Cambridge University Press, April 2009.
- [40] J. Frausto-Solis, “Tuned simulated annealing based on boltzmann and bose-einstein distribution applied to maxsat problem,” *Journal of Asian Scientific Research*, vol. 4, n^o. 11, p. 14–26, Jan 2014.

- [41] S. E. Nigg, N. Lörch et R. P. Tiwari, “Robust quantum optimizer with full connectivity,” *Science Advances*, vol. 3, n°. 4, p. e1602273, Apr 2017.
- [42] H. Goto, “Bifurcation-based adiabatic quantum computation with a nonlinear oscillator network,” *Scientific Reports*, vol. 6, n°. 11, p. 21686, Feb 2016.
- [43] K. Takata *et al.*, “A 16-bit coherent ising machine for one-dimensional ring and cubic graph problems,” *Scientific Reports*, vol. 6, n°. 1, p. 34089, Sep 2016.
- [44] P. L. McMahon *et al.*, “A fully programmable 100-spin coherent ising machine with all-to-all connections,” *Science*, vol. 354, n°. 6312, p. 614–617, 2016.
- [45] NTT Research, “100,000-spin coherent ising machine,” <https://group.ntt/en/newsrelease/2021/09/30/210930a.html>, September 2021, accessed : 2023/11/26.
- [46] Q. Cen *et al.*, “Large-scale coherent ising machine based on optoelectronic parametric oscillator,” *Light : Science & Applications*, vol. 11, n°. 1, p. 333, Nov 2022.
- [47] T. Wang, L. Wu, P. Nobel et J. Roychowdhury, “Solving combinatorial optimisation problems using oscillator based ising machines,” *Natural Computing*, vol. 20, n°. 2, p. 287–306, Jun 2021.
- [48] T. Wang, L. Wu et J. Roychowdhury, “New computational results and hardware prototypes for oscillator-based ising machines,” dans *Proceedings of the 56th Annual Design Automation Conference 2019*, ser. DAC ’19. New York, NY, USA : Association for Computing Machinery, 2019.
- [49] P. Zhang, Gary Chartrand, *Chromatic Graph Theory*. New York : Chapman and Hall/-CRC, Sep 2008.
- [50] S. Burer, R. D. C. Monteiro et Y. Zhang, “Rank-two relaxation heuristics for max-cut and other binary quadratic programs,” *SIAM Journal on Optimization*, vol. 12, n°. 2, p. 503–521, Jan 2002.
- [51] V. F. Yu et M. F. Maghfiroh, “A variable neighborhood search with path-relinking for the capacitated location routing problem,” *Journal of Industrial and Production Engineering*, vol. 31, n°. 3, p. 163–176, Apr 2014.
- [52] N. A. Aadit *et al.*, “Massively parallel probabilistic computing with sparse ising machines,” *Nature Electronics*, vol. 5, n°. 7, p. 460–468, Jul 2022.
- [53] B. Sutton, K. Y. Camsari, B. Behin-Aein et S. Datta, “Intrinsic optimization using stochastic nanomagnets,” *Scientific Reports*, vol. 7, n°. 1, p. 44370, Mar 2017.
- [54] R. M. Karp, “Reducibility among combinatorial problems,” dans *50 Years of Integer Programming*, 1972.

- [55] C. W. Commander, *Maximum cut problem, MAX-CUT* *Maximum Cut Problem, MAX-CUT*. Boston, MA : Springer US, 2009, p. 1991–1999. [En ligne]. Disponible : https://doi.org/10.1007/978-0-387-74759-0_358
- [56] T. R. Jensen et B. Toft, *Graph Coloring Problems*, ser. Wiley-Interscience series in discrete mathematics and optimization. John Wiley & Sons, 1995.
- [57] K. H. Fischer et J. A. Hertz, *Spin Glasses*, ser. Cambridge Studies in Magnetism. Cambridge University Press, 1991.
- [58] R. Hamerly *et al.*, “Experimental investigation of performance differences between coherent ising machines and a quantum annealer,” *Science Advances*, vol. 5, n°. 5, 2019.
- [59] T. Kanao et H. Goto, “Simulated bifurcation for higher-order cost functions,” *Applied Physics Express*, vol. 16, n°. 1, p. 014501, Dec 2022.
- [60] T. Zhang, Q. Tao, B. Liu et J. H. 0001, “A review of simulation algorithms of classical ising machines for combinatorial optimization,” dans *IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2022, p. 1877–1881.
- [61] H. Goto *et al.*, “High-performance combinatorial optimization based on classical mechanics,” *Science Advances*, vol. 7, n°. 6, p. eabe7953, Feb 2021.
- [62] T. Zhang, Q. Tao et J. Han, “Solving traveling salesman problems using ising models with simulated bifurcation,” dans *International SoC Design Conference (ISOCC)*, 2021, p. 288–289.
- [63] T. Zhang et J. Han, “Efficient traveling salesman problem solvers using the ising model with simulated bifurcation,” dans *Design, Automation and Test in Europe Conference and Exhibition*, 2022, p. 548–551.
- [64] Q. Tao et J. Han, “Solving traveling salesman problems via a parallel fully connected ising machine,” dans *Proceedings of the 59th ACM/IEEE Design Automation Conference*, ser. DAC ’22. New York, NY, USA : Association for Computing Machinery, 2022, p. 1123–1128.
- [65] R. H. Warren, “Solving the traveling salesman problem on a quantum annealer,” *SN Applied Sciences*, vol. 2, n°. 1, déc. 2019.
- [66] S. Jain, “Solving the traveling salesman problem on the d-wave quantum computer,” *Frontiers in Physics*, vol. 9, nov. 2021.
- [67] D. Venturelli, D. J. Marchand et G. Rojo, “Quantum annealing implementation of job-shop scheduling,” *arXiv*, vol. quant-ph, 2015. [En ligne]. Disponible : <https://arxiv.org/abs/1506.08479>

- [68] K. Kurowski *et al.*, “Hybrid quantum annealing heuristic method for solving job shop scheduling problem,” dans *Computational Science (ICCS)*. Cham : Springer International Publishing, 2020, p. 502–515.
- [69] B. Denkena, F. Schinkel, J. Pirnay et S. Wilmsmeier, “Quantum algorithms for process parallel flexible job shop scheduling,” *CIRP Journal of Manufacturing Science and Technology*, vol. 33, p. 100–114, 2021.
- [70] C. Carugno, M. Ferrari Dacrema et P. Cremonesi, “Evaluating the job shop scheduling problem on a d-wave quantum annealer,” *Scientific Reports*, vol. 12, n^o. 1, p. 6539, 2022.
- [71] M. T. Goodrich et R. Tamassia, *Algorithm Design and Applications*, B. Golub, édit. Wiley, 10 2014.
- [72] S. Lin et B. W. Kernighan, “An effective heuristic algorithm for the traveling-salesman problem,” *Operations Research*, vol. 21, n^o. 2, p. 498–516, Apr 1973.
- [73] K. Helsgaun, “General k-opt submoves for the lin–kernighan tsp heuristic,” *Mathematical Programming Computation*, vol. 1, n^o. 2, p. 119–163, Oct 2009.
- [74] “Concorde tsp solver,” <https://www.math.uwaterloo.ca/tsp/concorde.html>.
- [75] B. Fleischmann, “A new class of cutting planes for the symmetric travelling salesman problem,” *Mathematical Programming*, vol. 40, n^o. 1, p. 225–246, Jan 1988.
- [76] E. Balas et P. Toth, “Branch and bound methods for the traveling salesman problem,” Carnegie-Mellon University, Management Sciences Research Group, Technical Report ADA126957, 03 1983.
- [77] H. Marchand, A. Martin, R. Weismantel et L. Wolsey, “Cutting planes in integer and mixed integer programming,” *Discrete Applied Mathematics*, vol. 123, n^o. 1, p. 397–446, 2002.
- [78] Gurobi Optimization, LLC, “Gurobi optimizer reference manual,” 2023. [En ligne]. Disponible : <https://www.gurobi.com>
- [79] D. S. Johnson, “Combinatorial optimization : Algorithms and complexity. by christos h. papadimitriou and kenneth steiglitz,” *The American Mathematical Monthly*, vol. 91, n^o. 3, p. 209–211, Mar 1984.
- [80] C. E. Miller, A. W. Tucker et R. A. Zemlin, “Integer programming formulation of traveling salesman problems,” *Journal of the ACM*, vol. 7, n^o. 4, p. 326–329, Oct 1960.
- [81] M. Grötschel et G. L. Nemhauser, “George dantzig’s contributions to integer programming,” *Discrete Optimization*, vol. 5, n^o. 2, p. 168–173, May 2008.
- [82] W.-Y. Ku et J. C. Beck, “Mixed integer programming models for job shop scheduling : A computational analysis,” *Computers & Operations Research*, vol. 73, p. 165–173, 2016.