

Métaheuristique basée sur le recuit quantique

Belaube, Bergé, Cavarec, de Méric de Bellefon, Doutre

12/05/2015

Table des matières

1	Traveling Salesman Problem	4
1.1	Présentation du problème	4
1.2	Cartographie des minima locaux	4
2	Recuit Simulé	6
2.1	Principes physiques	6
2.2	Implémentation	7
3	Recuit Quantique	13
3.1	Fonctionnement	13
3.2	Réglages et paramètres	14
4	Résultats	16

Introduction

Dans cet article, nous cherchons à expliquer au lecteur le principe et l'intérêt de deux métaheuristiques : le recuit simulé et son amélioration quantique. Les métaheuristiques permettent d'établir un compromis entre la résolution d'un problème difficile (un problème difficile est un problème dont la solution exacte ne peut pas être obtenue en temps polynomial) et le temps de calcul nécessaire pour le résoudre. Si ces algorithmes n'apportent pas une solution exacte au problème, ils l'apportent néanmoins rapidement et souvent avec d'excellents résultats. Nous nous sommes plus particulièrement attachés à l'étude du problème du voyageur de commerce (Traveling Salesman Problem ou TSP) mais nous souhaiterions souligner le fait que ce type d'algorithmes peut être adapté à une immense variété de problèmes difficiles.

L'idée générale consiste à explorer efficacement l'espace des possibles afin d'obtenir une bonne solution. Ces algorithmes s'inspirent en particulier du chauffage suivi du refroidissement des métaux, aussi appelé recuit. Dans ce contexte, l'objectif des physiciens est de trouver l'état pour lequel l'énergie potentielle du métal est minimale en évitant les minima locaux présentés par la courbe énergétique. Le recuit simulé et le recuit quantique sont donc des algorithmes à utiliser lorsque la surface de coût n'est pas de nature convexe.

Nous définirons dans un premier temps le problème TSP. Nous verrons alors le principe du recuit simulé, les notions de physique sous-jacentes et son implémentation pratique. Nous traiterons dans un second temps des nombreux apports du recuit quantique, jusqu'alors peu étudié. Enfin, nous illustrerons ces méthodes avec les résultats obtenus sur le problème TSP.

1 Traveling Salesman Problem

1.1 Présentation du problème

Le problème du voyageur de commerce est défini par un ensemble de N points (des villes) séparées entre elles par un poids ou une distance $d_{i,j}$. L'objectif est de permettre au voyageur de commerce de trouver la route la plus courte lui permettant de visiter toutes les villes une fois tout en revenant à son point de départ. TSP trouve de nombreuses applications dans le domaine de la logistique mais aussi dans des domaines plus exotiques tels que la conception de bras robotisés, la fabrication de puces électroniques et même le séquençage de l'ADN. En 1972, Richard M. Karp a montré que le problème TSP était NP-complet. Il est facile de montrer que pour un problème avec N villes, il existe $\frac{(N-1)!}{2}$ routes possibles, soit environ le nombre estimé de particules fondamentales dans l'univers. Supprimer la contrainte d'une unique visite ne réduit pas la complexité du problème.

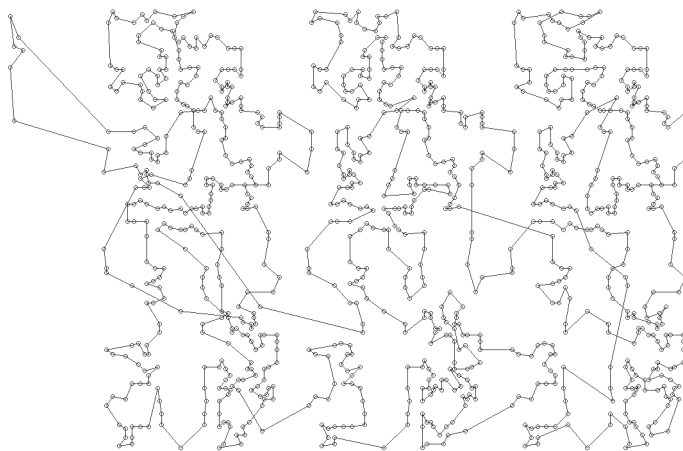


FIGURE 1 – Un exemple de route solution pour le benchmark pr1002

Remarquons que $d_{i,j}$ est une notion abstraite. Le graphe ne doit pas nécessairement respecter l'inégalité triangulaire ni même comporter des arêtes affectées de poids symétriques. Ces variantes ne manquent souvent pas d'accroître la nature non convexe du problème en créant de nouveaux minima locaux.

1.2 Cartographie des minima locaux

L'idée est de tenter d'établir une carte des minima locaux pour affiner le paramétrage. Cela fournirait également un outil supplémentaire pour comparer deux recuits différents entre eux, en comparant leur localisation dans le voisinage

proche du minimum global du graphe. Le problème est a priori complexe car il n'est pas possible de représenter en deux dimensions une carte où l'on pourrait voir une courbe continue dessinant la surface des énergies en fonction des routages. En effet, deux routages peuvent avoir un rapprochement très grand et pour autant ne pas avoir des énergies voisines, puisque l'énergie est liée au poids des arêtes et non à l'existence ou non de ces arêtes. Ainsi, il convient d'essayer d'estimer la densité des minima locaux, en regard à une norme donnée. Une nouvelle difficulté apparaît : positionner les répliques les unes par rapport aux autres en peu de temps de calcul car cette cartographie est a priori en temps exponentiel.

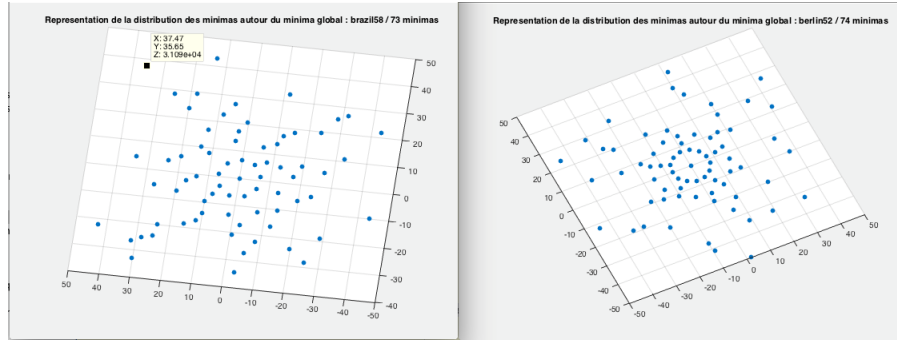


FIGURE 2 – Cartographie des minima locaux sur le benchmark brazil58

Le lieu où la densité des minima nous intéresse est centré sur le minimum global. On décide alors de comparer la distance des répliques correspondant à des minima locaux au minimum global seul pour diminuer le temps de calcul. L'idée est donc d'estimer une distance radiale d'un minimum local au minimum global, sans tenir compte a priori de la distance des minima locaux entre eux. Ainsi, chaque minimum global trouvé peut être placé une carte, on place aléatoirement sur un cercle centré sur le minimum global, de rayon son rapprochement au minimum global. L'idée de placer chaque minimum local aléatoirement sur un cercle permet une visualisation en trois dimensions de la densité de minima, et souligne la possibilité d'intégrer une information supplémentaire dans le modèle en jouant sur l'angle aléatoire - par exemple, pour deux minima différents qui ont une distance identique au minimum global, prendre en compte la distance entre eux et en rendre compte dans leur éloignement sur leur cercle.

2 Recuit Simulé

2.1 Principes physiques

C'est au début des années 80 que Kirkpatrick a généralisé la notion de recuit simulé. En métallurgie, le recuit consiste à chauffer un métal jusqu'à sa température de fusion puis à le refroidir très lentement jusqu'à ce que les atomes se réarrangent pour regagner l'état solide. L'état fondamental du solide sera atteint seulement si la température initiale est assez élevée et si le refroidissement est suffisamment lent. Si ces conditions ne sont pas respectées, le métal se retrouve dans un minimum local d'énergie potentielle, c'est à dire un état méta-stable.

Il convient donc de jouer avec la variation de température pour permettre au système de sortir des minima locaux. En 1953, Metropolis avait déjà développé un algorithme recréant l'évolution d'un métal au cours du recuit. En partant d'un état dont l'énergie est égale à E_i , on applique une légère variation de température. À l'équilibre thermique, le système peut se retrouver dans un état j dont l'énergie est E_j .

La probabilité qu'un solide soit dans un état i avec une énergie E_i à la température T est donnée par : $\mathbb{P}(X=i) = e^{\frac{-E_i}{k_B T}} / \sum_j e^{\frac{-E_j}{k_B T}}$. Plus l'énergie de l'état i est faible par rapport à celle des états j , plus le système a des chances de se retrouver dans cet état. C'est pourquoi Metropolis choisit de fixer la probabilité d'acceptation d'un passage A de l'état i à j à la température T par $\mathbb{P}(A) = e^{\frac{(E_i - E_j)}{k_B T}}$.

En faisant l'analogie entre une solution du problème et un état du système et entre l'énergie d'un état et le coût de la solution, on peut alors construire l'algorithme du recuit simulé :

Algorithm 1 Recuit Simulé

```
sol  $\leftarrow$  RealisationAleatoire
for it = 0; it < N; it ++ do
  sol  $\leftarrow$  mutation(sol)
   $\Delta E \leftarrow E_j - E_i$ 
  if  $\Delta E < 0$  then
    Accepter cette mutation
  else
    Ne l'accepter qu'avec une probabilité égale à  $\mathbb{P}(A)$ 
  end if
  Choisir  $T_{k+1}$  plus petit que  $T_k$ 
end for
return sol
```

Ainsi, aux hautes températures, le système change d'état quasiment aléatoirement. Il explore la surface d'énergie de manière globale. Lorsque la température décroît, le système a de moins en moins de chance d'accepter de passer dans un état qui augmente son énergie. La recherche d'une solution devient alors locale car l'état courant ne peut plus franchir une barrière d'énergie importante. Plus le refroidissement est lent, plus le système a de chances d'atteindre un état dont l'énergie est proche de l'énergie fondamentale. Le lecteur attentif aura compris que les éléments importants du recuit simulé sont : la température initiale, le taux de décroissance de la température à chaque itération et enfin la nature de la mutation élémentaire. K_B est arbitrairement fixé égal à 1.

Dans une section suivante, nous montrerons comment nous sommes parvenus à fixer les paramètres importants du recuit.

2.2 Implémentation

Pour utiliser cet algorithme sur le problème du voyageur de commerce, il faut commencer par définir les différents éléments indispensables au recuit : l'état, l'énergie et la mutation. Un état est défini comme une solution possible au problème donné. Par exemple, pour le problème SAT, l'état est l'assignation, à chaque variable, d'un booléen. Pour le problème TSP, l'état est une route c'est-à-dire une suite d'arêtes où chaque noeud n'est traversé qu'une seule et unique fois. D'un point de vue informatique, une route sera représentée par une liste de noeuds. Le graphe est représenté par une matrice carrée des distances $d_{i,j}$. La taille de cette matrice définit le nombre de noeuds dans la route. A chaque état est associée une énergie. Cette énergie correspondra ici à la longueur totale de la route : c'est la somme des longueurs de chacune des arêtes (i, j) de la route, chaque longueur étant donné par la matrice du graphe.

0	2	3	4	5
2	0	6	8	1
3	6	0	3	5
4	8	3	0	2
5	1	5	2	0

TABLE 1 – Exemple de graphe à 5 noeuds

Pour le graphe décrit sur le tableau 2.2, l'énergie correspondant à la route définie par les noeuds $0 \rightarrow 2 \rightarrow 4 \rightarrow 1 \rightarrow 3$ vaut $3+5+1+8+4$ donc 21. On se réfère ici à une indexation Java, commençant à 0. De plus, comme c'est le cas pour ce graphe, tous les problèmes traités dans ce document seront symétriques. Pour ce qui est de la mutation, nous utiliserons 2-opt car c'est la mutation sur TSP qui modifie le moins d'arêtes, à savoir 2, entre l'état courant et l'état muté. Nous démontrons en figure 4 que c'est la mutation la plus efficace face à des mutations qui modifient davantage d'arêtes. La mutation 2-opt consiste à

choisir deux noeuds i et j différents de la route, à les échanger et à échanger les noeuds 'intermédiaires', autrement dit d'échanger le noeud suivant i et le noeud précédent j , etc... L'implémentation de cette mutation est expliquée sur la figure 3.

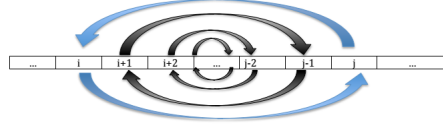


FIGURE 3 – Fonctionnement de la mutation 2-opt

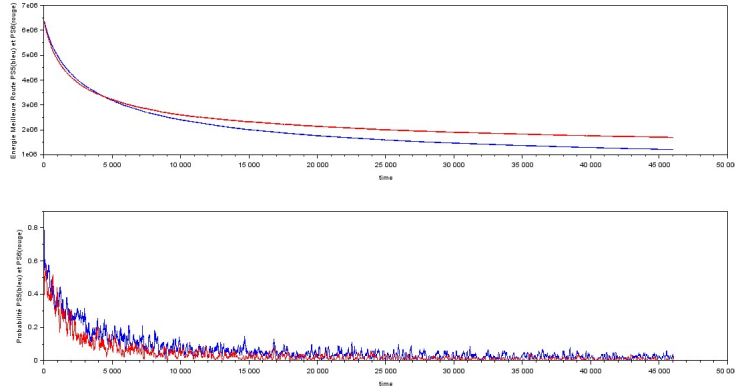


FIGURE 4 – Comparaison entre 2-opt(en bleu) et 3-opt(en rouge)

L'étude de l'influence des paramètres est primordiale si l'on souhaite exécuter un recuit efficace. Pour un recuit simulé, étant donné que K est fixé, le seul paramètre à prendre en compte est la température. Mais ce paramètre n'est pas fixe, il décroît et il y a donc trois données à définir : la température de début T_{deb} , la température de fin T_{fin} et le mode de décroissance. On s'est très vite aperçu, et cela est parfaitement cohérent, que fixer une fonction température T_n ne donnait pas des résultats satisfaisants sur tous les benchmarks. Cette expérience est décrite dans le tableau 5 où on pose $T_{deb} = 1000$, décroissance exponentielle avec un facteur de décroissance de 0.99. La température de fin est très proche de 0. On a $T_n = 1000 \times 0.99^n$ avec $n = 0..100 \times 58 \times 58$

En réalité, il n'existe pas de fonction température universelle, compatible avec tous les graphes. Pour utiliser correctement le recuit simulé, il faut adapter la température à chaque graphe.

NOM	DIMENSION	TPS EXEC	SOL OPT	% ECART MIN	% ECART MOY	SOL MOY	SOL MIN
gr17	259	2085	0	8.753477218		2100.71	2085
a280	968	2579	28.84605129	37.36717676		3542.699489	3322.939663
ali535	1533	202339	1.002772575	2.861430569		208128.79	204368
bayg29	221	1610	0	5.075465839		1691.715	1610
bayg29	234	2020	0.207029703	3.785148515		2096.46	2026
berlin52	300	7545	0.031136998	4.688853369		7896.387871	7544.365902
bier127	451	118282	0.036886041	1.773162568		120379.3321	118325.393
brazil58	326	25395	0	0.85770821		25612.815	25395
brg180	439	1950	37.43589744	56.90512821		3056.65	2680
burma14	215	3323	0.210953024	5.941378223		3340.99	3330
ch130	402	6110	3.795537011	9.017224284		6660.952404	6341.907311
ch150	456	6528	4.837087003	11.15650347		7256.296546	6830.70904
d198	493	15780	2.745952606	5.092914961		16583.66198	16213.31132
d493	1202	35002	4.708471004	7.530598177		37637.85997	36681.56082
d857	2597	48912	6.103438896	6.587684555		53112.40827	51897.31393

FIGURE 5 – Solution en énergie du recuit simulé pour une température arbitraire

On va, pour cela, essayer de trouver un algorithme qui détermine correctement la température du recuit en fonction du graphe traité : un paramètre universel. Rappelons rapidement quel est le rôle de la température. Elle ajuste la valeur de la probabilité d'acceptation lorsque la mutation nous dirige vers un état plus haut en énergie : $p = e^{-\frac{\Delta E}{KT}}$. Il faut paramétrer de telle sorte que les probabilités d'acceptation soient quasi-idéales pour une bonne utilisation du recuit. Sur une évaluation, p dépend donc de ΔE et de T . Pour concevoir notre paramètre, on va émettre deux hypothèses importantes. Tout d'abord, on émet l'hypothèse que l'efficacité du recuit simulé dans son ensemble dépend de T_n et de la distribution globale de toutes les différences d'énergie ΔE engendrées par des 2-opt sur le graphe. On construira donc un échantillon trié de 1000 ΔE . Dans un second temps, on suppose qu'il existe un intervalle $\Delta E_i \leq T_n \leq \Delta E_j$ où le recuit est efficace pour tout graphe. Autrement dit, on veut trouver i et j universels. L'objectif est donc de déterminer ces deux indices désormais. Sur plusieurs benchmarks, on lance des recuits pour une décroissance linéaire, $T_{fin} = 0$ avec différentes températures de départ $T_{deb} = 2.\Delta E_k$. On notera les différentes valeurs de k qui donnent les meilleurs recuits pour se faire une idée de l'intervalle $[i, j]$.

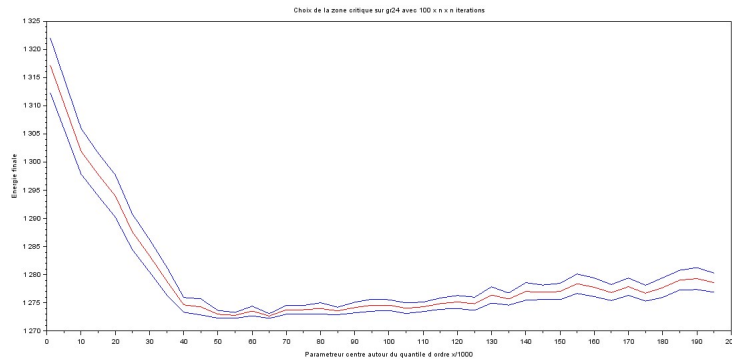


FIGURE 6 – Performance du recuit pour $T_{deb} = 2.\Delta E_k$ sur gr24

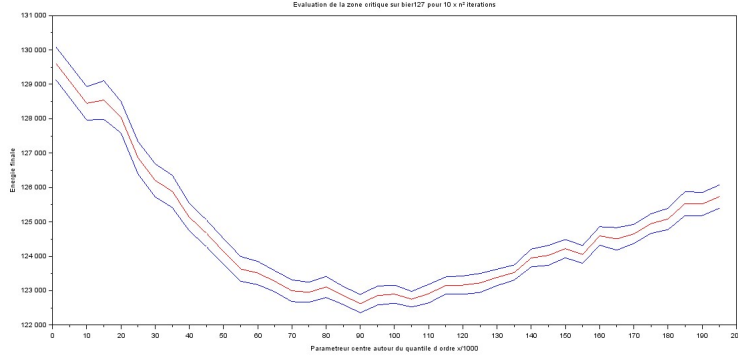


FIGURE 7 – Performance du recuit pour $T_{deb} = 2.\Delta E_k$ sur bier127

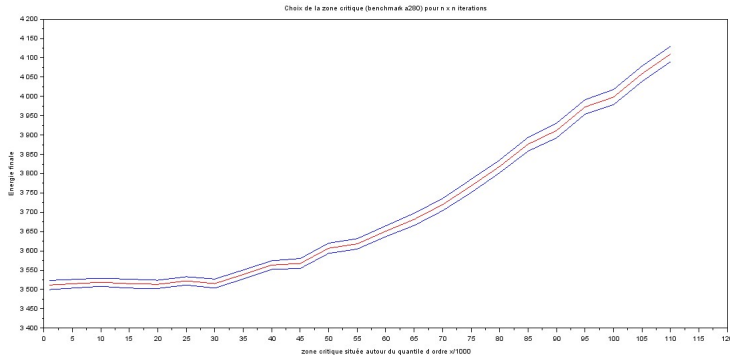


FIGURE 8 – Performance du recuit pour $T_{deb} = 2.\Delta E_k$ sur a280

Les trois figures précédentes présentent les résultats de cette expérience sur plusieurs exemples. On constate que sur chacune d'entre elles, la zone $2.\Delta E_{50} \leq T_{deb} \leq 2.\Delta E_{100}$ fournit les meilleurs résultats. Il faut donc concentrer la température autour du cinquième centile et du premier décile de la distribution des ΔE . Pour une décroissance linéaire, on posera $T_{deb} = \Delta E_{100}$ et $T_{fin} = \Delta E_{50}$. Pour une décroissance exponentielle, on posera $T_{deb} = 10.\Delta E_{100}$ et $T_{fin} = \Delta E_{50}$.

Les figures 9 et 10 nous fournissent les performances de notre paramétreur sur un lot d'une quinzaine de benchmarks. Si l'on compare avec la figure 5, le pourcentage moyen est clairement amélioré. Seul le graphe brg180 résiste à notre paramétreur : en regardant la matrice des distances de ce benchmark, on s'aperçoit qu'elle est assez particulière : les énergies des arêtes sont très similaires, certaines sont infinies,... Cela crée des discontinuités importantes dans

NOM	DIMENSION	TPS EXEC	SOL OPT	% ECART MIN	% ECART MOY	SOL MOY	SOL MIN	:
gr17	259	2085	0	0.18701438		2088,275	2085	
a280	968	2579	1,337252375	5,059399027	2709,481901	2613,487739		
ali535	1533	202339	1,152027044	8,155580486	218840,92	204670		
bayg29	221	1610	0	1,83447205	1639,535	1610		
bayg29	234	2020	0	0,946287129	2039,115	2020		
berlin52	300	7542	0,031369688	7,972347917	8143,27448	7544,365902		
bielr127	451	118282	0,896942926	5,591072146	124895,232	119342,922		
brazil58	326	25395	0	2,172376452	25046,675	25395		
brg180	439	1950	142,5641026	148,3410256	4842,65	4730		
burma14	215	3323	0,210953024	1,385958919	3367,88	3330		
ch130	402	6110	0,569560736	4,574159634	6389,481154	6144,800161		
ch150	456	6528	0,634036117	4,99506633	6854,07793	6569,389881		
d198	493	15780	0,861403733	3,135271252	16274,7458	15915,92951		
d493	1202	35002	1,717663834	4,65172305	36630,1961	35603,2167		
d657	2597	48912	1,025264867	2,990266755	50374,59928	49413,47755		

FIGURE 9 – Performance du recuit avec le paramètreur en décroissance linéaire

NOM	DIMENSION	TPS EXEC	SOL OPT	% ECART MIN	% ECART MOY	SOL MOY	SOL MIN	:
gr17	259	2085	0	0.186981181		2089,14	2085	
a280	968	2579	2,422036536	7,168410285	2763,673302	2641,464322		
ali535	1533	202339	2,371268021	4,235233630	210908,53	207137		
bayg29	221	1610	0	1,642236025	1636,44	1610		
bayg29	234	2020	0	1,141584158	2043,06	2020		
berlin52	300	7542	0,031369688	2,052040354	7696,764884	7544,365902		
bielr127	451	118282	2,114090913	2,609480071	121368,9452	118714,3473		
brazil58	326	25395	0	0,799182754	25597,965	25395		
brg180	439	1950	138,974359	123,6666667	4361,5	3920		
burma14	215	3323	0,210953024	0,591333131	3342,65	3330		
ch130	402	6110	1,336241833	4,018627794	6355,538158	6126,688672		
ch150	456	6528	3,406061941	4,528243677	6823,473187	6614,206341		
d198	493	15780	2,317030601	2,310321641	16144,56875	15936,74217		
d493	1202	35002	5,060838262	4,877795483	36709,32598	36067,47327		
d657	2597	48912	5,307659938	8,195847117	52920,75274	51508,08263		

FIGURE 10 – Performance du recuit avec le paramètreur en décroissance exponentielle

la distribution des ΔE et le paramètreur prend souvent une valeur trop élevée pour T sur tout le recuit. A noter que non seulement brg180 se résoud très bien avec $T = 0$ mais aussi que de telles discontinuités ne peuvent apparaître avec un graphe euclidien.

Algorithm 2 Paramètreur linéaire

```

l ← listeVide
for k = 0; k < 1000; k ++ do
    r ← newRouteAleatoire
    r ← mutation(r)
     $\Delta E \leftarrow E_j - E_i$ 
    if  $\Delta E > 0$  then
        l.ajoute( $\Delta E$ )
    end if
end for
l.trie()
 $T_{deb} = l.get(100)$ 
 $T_{fin} = l.get(50)$ 
coefficientDeRefroidissement =  $\frac{T_{fin} - T_{deb}}{N}$ 

```

L'application du recuit simulé sur TSP pose le problème des paramètres. La recherche de la bonne température pour un seul graphe nécessite déjà de nombreux tests. Nous avons proposé à l'utilisateur une méthode permettant de

calibrer automatiquement la température. Cette méthode ne correspond bien évidemment pas au paramétrage idéal. Cependant, si le temps nous est compté, elle évite des tests encombrants pour trouver T et donne une bonne approximation du minimum global. Le temps d'exécution du paramétreur est négligeable comparé à celui du recuit simulé qui ne dépasse jamais quelques secondes pour des graphes de moins de 1000 noeuds traités en plusieurs millions d'évaluations.

3 Recuit Quantique

3.1 Fonctionnement

Le recuit quantique est un méta-heuristique qui approfondit le modèle du recuit simulé. Il nous faut d'abord évoquer ce qu'on pourrait appeler un recuit simulé 'en parallèle' ou SA parallèle qui, au lieu de travailler sur un seul état, traiterait plusieurs états en même temps. On aurait alors le choix entre appliquer, sur un problème, un recuit simulé pour $10.n^2$ itérations ou un recuit parallèle de 10 répliques, chacune en n^2 itérations. On appelle particule l'ensemble de ces 10 répliques. L'avantage du SA parallèle est de dompter, plus que ce que fait le SA classique, le côté combinatoire de ces heuristiques. Par exemple, s'il existe un graphe avec 10 minimums locaux profonds en énergie, le SA classique va se bloquer dans l'un de ces minima alors que le SA parallèle va en explorer plusieurs. Par contre, la décroissance du SA parallèle en énergie est plus lente que pour un recuit simulé classique. Le recuit quantique ou QA est un SA en parallèle dont l'hamiltonien a deux composantes : potentielle et cinétique. La composante cinétique est calculé à partir des matrices de spin des états de la particule. Pour TSP, la matrice de spin notée S d'une route est une matrice triangulaire supérieure. Pour $i < j$, on ajoute 1 aux indices (i,j) si l'arête correspondante appartient à la route, -1 sinon. La diagonale est constituée de zéros.

0	1	-1	-1	1
0	0	-1	1	-1
0	0	0	1	1
0	0	0	0	-1
0	0	0	0	0

TABLE 2 – Matrice de spin de la route $0 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 4$

Les répliques sont agencées sous forme de chaînage ou de liste en début de recuit. Ainsi, une réplique a deux répliques voisines fixes tout au long de l'algorithme. L'énergie cinétique est proportionnelle aux produits scalaires des matrices de spin entre routes voisines. On a alors $H_{cin} = J_{\Gamma} \cdot (\sum_{k=0}^P \sum_{i < j} S_{(i,j),k} \cdot S_{(i,j),k+1} + S_{(i,j),P} \cdot S_{(i,j),0})$ où P est le nombre de répliques. Γ est un paramètre semblable à T dans la mesure où il faudra savoir définir Γ_{deb} , Γ_{fin} et le mode de décroissance. On utilisera l'expression suivante pour déterminer J_{Γ} : $J_{\Gamma} = -0.5T \cdot \ln(\tanh(-\frac{\Gamma}{PT}))$. L'objectif du recuit quantique est donc de minimiser l'hamiltonien $H = H_{pot} + H_{cin} = \frac{1}{P} \cdot \sum_{k=0}^P H_{pot,k} + H_{cin}$.

Algorithm 3 Recuit Quantique

```
p ← creeParticule
for it = 0; it < N; it ++ do
  for k = 0; k < P; k ++ do
    sol ← p.getReplique(k)
    sol ← mutation(sol)
     $\Delta E \leftarrow E_j - E_i + \Delta E_{cin}$ 
    if  $\Delta E < 0$  then
      Accepter cette mutation
    else
      Ne l'accepter qu'avec une probabilité égale à  $\mathbb{P}(A)$ 
    end if
    Choisir  $T_{k+1}$  plus petit que  $T_k$ 
    Choisir  $\Gamma_{k+1}$  plus petit que  $\Gamma_k$ 
    Modifier l'ordre de parcours des répliques sans modifier le chainage
  end for
end for
bestSol ← p.getMeilleureReplique
return bestSol
```

3.2 Réglages et paramètres

Le réglage de la température se fait avec le paramètre présenté dans le chapitre 'Recuit Simulé'. On pose $T = \Delta E_{50}$ (cinquième centile de la distribution des ΔE) Un facteur important à régler est le nombre de répliques à utiliser. Le nombre de répliques idéal dépend du nombre de noeuds du graphe que l'on traite et du nombre d'itérations. Traiter un graphe de 100 noeuds en $10.n^2$ ne nécessite pas d'employer le même nombre de répliques qu'avec $100.n^2$. On effectue cette expérience sur un graphe de 52 noeuds : on trace le résultat en fonction du nombre de répliques pour différentes valeurs du nombre d'itérations sur la figure 11. Pour la courbe bleue, le nombre de répliques idéal est 1. En revanche, lorsque le nombre d'évaluations augmente (courbe noire), le nombre de répliques idéal est proche de 10.

Le dernier paramètre auquel il faut s'intéresser est Γ . Nous avons décidé de paramétrer universellement Γ comme nous l'avons fait avec T . La méthode est assez similaire : on va déterminer sa valeur en fonction de l'échantillon des ΔE . On suppose qu'il existe un k universel idéal tel que $\Delta E_{cin} = \frac{\Delta E_k}{P}$ en fin de recuit. La variation élémentaire de ΔE_{cin} vaut $4J_\Gamma$: le terme 4 provient du fait que, lors de la modification d'une arête, chaque route a deux voisins et les valeurs de spin passent de +1 à -1. On obtient donc la relation : $J_\Gamma = \frac{\Delta E_k}{4P}$.

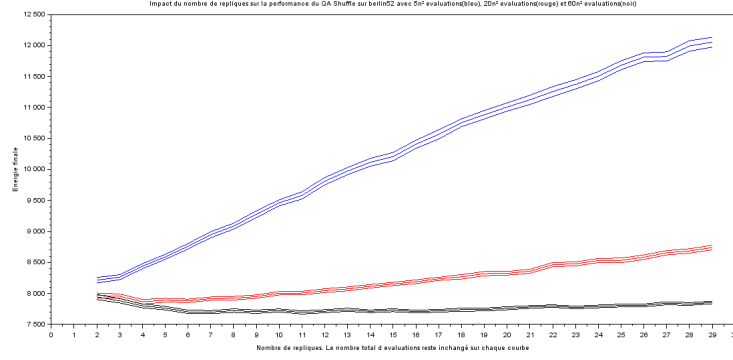


FIGURE 11 – Performance du recuit quantique en fonction de P en $5n^2$ (bleu) , $20n^2$ (rouge) et $60n^2$ (noir)

Cette relation est valable pour la dernière itération du QA. On en déduira donc Γ_{fin} . On pose arbitrairement $\Gamma_{fin} = 100\Gamma_{fin}$. On détermine la valeur de k à l'aide d'expériences sur différents benchmarks. Comme l'illustre la figure 12, la valeur $k = 200$ convient parfaitement. A noter que le paramétrage de Γ est bien moins important que celui de T : ici, l'énergie varie seulement de quelques unités pour les différents paramétrages.

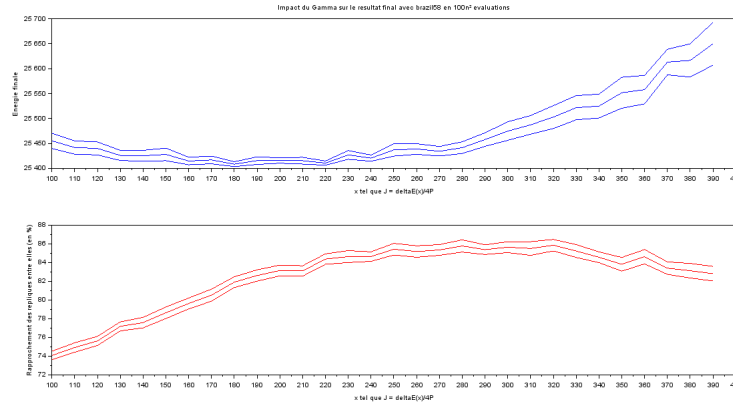


FIGURE 12 – Performance du recuit quantique en fonction de J_{fin}

4 Résultats

Il s'agira, dans cette partie, de comparer les performances des deux algorithmes présentés précédemment. Dire lequel des deux algorithmes est le meilleur n'est pas tâche aisée : pour certains paramètres, sur un certain graphe, le recuit simulé donne de meilleurs résultats que le recuit quantique sur un même nombre d'évaluations global. Par exemple, lorsque le nombre de répliques est extravagant sur le recuit quantique, on descend très lentement en énergie, contrairement au recuit simulé qui se rapproche rapidement du minimum global. Inversement, le recuit quantique peut être préférable dans d'autres dispositions. Il faudra donc faire attention à comparer "le meilleur recuit simulé" au "meilleur recuit quantique" pour un même nombre d'évaluations.

On traitera d'abord le comportement global des deux recuits. Un premier constat est celui concernant la valeur de l'énergie potentielle tout au long du recuit. A travers la figure suivante, on remarque que les deux algorithmes adoptent des attitudes globalement différentes.

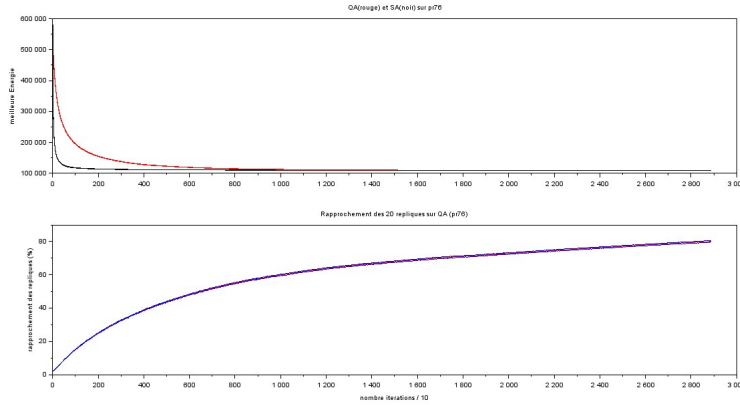


FIGURE 13 – Evolution des énergies sur pr76 et rapprochement des répliques pour QA

Comme on peut le voir, la décroissance du recuit quantique en énergie est plus lente. D'ailleurs, plus le nombre de répliques est important, plus cette décroissance sera lente. En fin de recuit, le recuit simulé stagne car ça prend du temps de trouver un état de meilleure énergie. Le recuit quantique le rattrape en rapprochant les répliques les unes aux autres : cela permet d'explorer davantage d'états potentiellement intéressants. Ce rapprochement se mesure en déterminant le nombre d'arêtes communes entre les différentes routes. Si toutes

les routes sont identiques, le rapprochement vaut 100. On a tracé les rapprochements pour un recuit de 20 répliques sans énergie cinétique (recuits simulés en parallèle) et avec énergie cinétique (recuit quantique) :

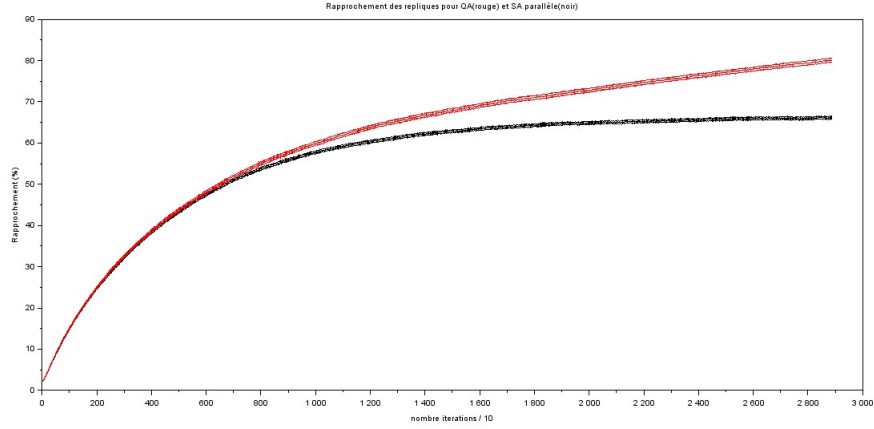


FIGURE 14 – Rapprochement des répliques pour QA et SA parallèle sur pr76

Le rapprochement sur les recuits simulés en parallèle est uniquement dû à la convergence des répliques vers le minimum global. L'écart entre les deux courbes illustre le rôle du terme cinétique dans l'hamiltonien. C'est en quelque sorte le pari lancé par le recuit quantique : rapprocher les répliques entre elles pour explorer l'espace convexe constitué par ces routes.

La comparaison des deux recuits se fera pour des probabilités d'acceptations identiques afin de ne pas influencer le résultat final de l'expérience. Pour cela, on prend $T_{SA} = T_{QA} \times P$ où P est le nombre de répliques. La température est réglée par notre paramètreur, de même que Γ . Le nombre de répliques idéal est utilisé. La courbe obtenue sur l'évolution des énergies est celle de la figure 13. Il faut zoomer sur les dernières évaluations du recuit.

Le recuit quantique obtient une énergie finale inférieure à celle du recuit simulé. De plus, on voit sur la figure 15 que les intervalles de confiance à 95% ne se chevauchent pas, ce qui donne un avantage considérable au recuit quantique.

Une autre comparaison a été effectuée avec un autre benchmark à 58 noeuds (figure 16).

Il est intéressant de voir que tout l'intérêt du recuit quantique repose dans l'ajout du terme cinétique : en effet, SA parallèle obtient une moyenne inférieure à SA alors que le recuit quantique est avantageux comparé à ces deux algorithmes.

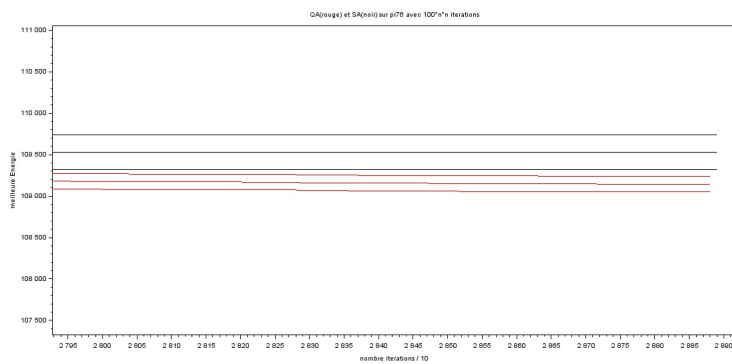


FIGURE 15 – Meilleures Energies des dernières itérations des expériences SA et QA

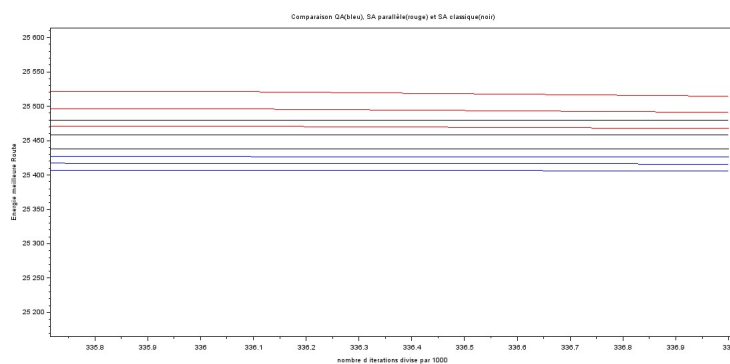


FIGURE 16 – Evolution de l'énergie sur SA, SA parallèle et QA en fin de recuit

Conclusion

Table des figures

1	Un exemple de route solution pour le benchmark pr1002	4
2	Cartographie des minima locaux sur le benchmark brazil58	5
3	Fonctionnement de la mutation 2-opt	8
4	Comparaison entre 2-opt(en bleu) et 3-opt(en rouge)	8
5	Solution en énergie du recuit simulé pour une température arbitraire	9
6	Performance du recuit pour $T_{deb} = 2.\Delta E_k$ sur gr24	9
7	Performance du recuit pour $T_{deb} = 2.\Delta E_k$ sur bier127	10
8	Performance du recuit pour $T_{deb} = 2.\Delta E_k$ sur a280	10
9	Performance du recuit avec le paramètre en décroissance linéaire	11
10	Performance du recuit avec le paramètre en décroissance expo- nentielle	11
11	Performance du recuit quantique en fonction de P en $5n^2$ (bleu) , $20n^2$ (rouge) et $60n^2$ (noir)	15
12	Performance du recuit quantique en fonction de J_{fin}	15
13	Evolution des énergies sur pr76 et rapprochement des répliques pour QA	16
14	Rapprochement des répliques pour QA et SA parallèle sur pr76 .	17
15	Meilleures Energies des dernières itérations des expériences SA et QA	18
16	Evolution de l'énergie sur SA, SA parallèle et QA en fin de recuit	18

List of Algorithms

1	Recuit Simulé	6
2	Paramètre linéaire	11
3	Recuit Quantique	14