**Introduction**

Nous sommes souvent confronté à des problèmes d’optimisation dans une multitude de domaine industrielle ou académique tel que la minimisation d’un coût de production, l’optimisation du parcours d’un véhicule, l’amélioration des performances d’un circuit électronique…etc. la plus part de ces problème sont NP-difficiles, l’utilisation des méthodes exacte garantissant la résolution optimale du problème est beaucoup trop couteuse en terme de temps et de calcul et parfois inutile. Dans ce cas l’utilisation des méthodes approchées permet d’obtenir des solutions acceptables, de bonne qualité dans un temps raisonnable dont font partit les méta-heuristiques.

Le domaine des Méta-heuristiques est encor jeune, plusieurs chercheurs ont adapté des idées de différent domaines dans le but de développer des procédures plus puissantes ; le Recuit Simulée est fondé sur des processus physique en métallurgie, tandis que les méthodes génétiques essayent d’imiter les phénomènes biologiques d’évolution naturelle. De même la méthode Tabou peut s’apparenter à une technique fondée sur des concepts d’intelligence Artificielle [Rego, Rou].

La recherche tabou a obtenu des solutions optimales et prés optimales pour une grande variété de problèmes classiques et pratiques, dans des applications allant de la planification aux télécommunications et de la reconnaissance de caractères aux réseaux de neurones [Glo, 90].Le travaille que nous proposons dans cet exposé rentre dans ce contexte, on va présenter la méthode tabous.

Après avoir donné un petit historique sur le développement des heuristiques, et la définition de la recherche tabou tout en expliquant le mot tabou et l’origine de la méthode, nous donnerons quelques définitions de bases pour pouvoir entamer dans un troisième temps le principe général de la méthode dans lequel nous nous rentrons dans les détailles de la méthode en analysant les différentes stratégies qui existe dans cette méthode.

ensuite nous donnerons quelques problèmes d’optimisation combinatoire utilisant la méthode tabou, nous verrons à cette occasion toute la richesse et la variété de l’application de la méthode. Puis nous citons les avantages et les inconvénients de la recherche tabou, en plus nous donnerons une comparaison de la méthode avec d’autres techniques de l’optimisation.

Enfin, nous conclurons notre exposé en donnant quelques perspectives concernant la méthode étudiée.

**1. Historique Scientifique**

**1.1. Les méta-heuristiques**

Les méta-heuristiques [Hao, 99] contiennent souvent une technique ou une astuce permettant d’éviter de se retrouver piégé dans des minima locaux, en explorant tout l’espace des solutions, de façon à augmenter la probabilité de rencontrer le minimum global.

***Développement des heuristiques***

• Tendance dans les années 70 : techniques d’amélioration des solutions par recherche locale

 ⇒ Procédure de recherche itérative qui améliore une solution de départ en lui appliquant une

 série de modifications locales (mouvements)

 ⇒ Arrêt lorsqu’un optimum local est trouvé

• 1983 : Une nouvelle heuristique apparaît, le *Recuit Simulé*

 ⇒ Permet une exploration aléatoire contrôlée de l’espace des solutions.

• 1986 : Bien que son origine remonte à 1977, la *Recherche Tabou* n’est proposée qu’au milieu des années 80 par Fred Glover

 ⇒ Méthode développée pour résoudre des problèmes combinatoires (la plupart NP-durs)

 ⇒ Révolution de cette méthode par rapport aux autres: permet de surmonter le problème des optima locaux par l’utilisation de listes taboues (Principe de mémoire)

• Par la suite : algorithmes génétiques, colonies de fourmis, …, [Via, Aya, 04].

**1.2 Recherche tabou**

 *Tabou (en français):* un sujet qu’il est préférable de ne pas aborder si l’on veut respecter les codes de la société.

 *La recherche tabou dans le domaine de la recherche opérationnelle*

La recherche tabou est une méthode d’optimisation mathématique de la famille des techniques de recherche locale présentée pour la première fois par Fred Glover en 1986 [Glo, Tai, Wer, 92], et elle est devenue très classique en optimisation combinatoire.

Elle se distingue des méthodes de recherche locale simples par l’introduction de la notion d’historique dans la politique d’exploration des solutions afin de diriger au mieux la recherche dans l’espace. Cette méthode s’est révélée particulièrement efficace et a été appliquée avec succès à de nombreux problèmes difficiles [Glo, Lag, 98].

**2. Définition de base**

**2.1. Définition des Variables**

• i : la solution actuelle

F(i)

• i’ : la prochaine solution atteinte (solution voisine)

F (i’)

• V(i): l’espace de solutions voisines à i (l’ensemble des i’)

Dans le cadre de l’optimisation combinatoire, en pratique, on aura tout intérêt à définir le voisinage en considérant l’ensemble des modifications élémentaires que l’on peut appliquer à une solution *i* donnée.

 V(i)

• m : mouvement de i à i’

• i globale : la solution optimale *globale* qui minimise la fonction objectif F ( ).

• i\* : la solution optimale actuelle

F (i\*)

**2.2. Définition des termes**

***Mouvement non améliorateur***: un mouvement qui nous sortirait d’un minimum local i\* en nous amenant à une solution voisine i’ *pire* que l’actuelle.

***Mouvement tabou***: un mouvement non souhaitable, comme si on redescendait à un minimum local d’où on vient juste de s’échapper.

Et donc on a jusqu’a présent :

 m

 F (i’)

 F (i) F (i) global

 V (i)

***T* :** liste des mouvements tabous. Il peut exister plusieurs listes simultanément. Les éléments de la liste sont t(i,m).

Une liste T avec trop d’éléments peut devenir très restrictive. Il a été observé que trop de contraintes (tabous) forcent le programme à visiter des solutions voisines peu alléchantes à la prochaine itération.

Une liste T contenant trop peu d’éléments peu s’avérer inutile et mener à des mouvements cycliques.

 ***a (i,m)* :** *critères d’aspiration*, détermine quand il est avantageux d’entreprendre m, malgré son statut tabou.

**3. Principe de la méthode**

**3.1. Idée de départ**

 Se déplacer de solution en solution (en visitant éventuellement des solutions moins bonnes) en ***s’interdisant de revenir à une solution déjà rencontrée* [**Run, Esc**]**.

* A chaque itération, on examine V(i) et on va sur la meilleure solution i’ même si le coup remonte (F (i’) >F(i)).

2 donc La recherche Tabou ne s'arrête pas au premier optimum trouvé.

* Le danger serait alors de revenir à *i*  immédiatement, puisque *i* est meilleure que *i’*. Pour éviter de tourner ainsi en rond, on crée une liste *T* qui mémorise les dernières solutions visitées et qui interdit tout déplacement vers une solution de cette liste. Cette liste *T* est appelée liste Tabou.
* On conserve en cours de route la meilleure solution trouvée i\*.
* On stoppe dés que le critère de fin est vérifié.

**3.2 Algorithme tabou**

(1) *Générer* une solution initiale S de manière aléatoire

(2) S\*🡨 S ; C\*🡨 F(S) / S\* est la meilleure solution rencontrée, C\* est sont cout et F la

 fonction objectif

(3) *Ajouter* S a la liste tabou ; K🡨 0

(4) **Répéter tant qu’**un critère de fin n’est pas vérifié

(5) *Choisir* parmi le voisinage de SK ,V(SK), le mouvement qui minimise F et qui

 n’appartient pas à la liste tabou, meilleur(SK)

(6) SK+1 🡨 meilleur(SK)

(7) **Si** la liste tabou est pleine **alors**

(8) *Remplacer* le dernier élément de la liste tabou par SK+1

(9) **Si non**

(10) *Ajouter* SK+1 a la liste tabou

(11) **Fin si**

(12) **Si** (C (SK+1) < C\*) **alors**

(13) S\*🡨 SK+1, C\*🡨 C (SK+1)

(14) **Fin si**

(15) **Fin** d’algorithme

Lorsque la mémoire est pleine, elle est gérée comme une liste circulaire en FIFO (*First In First Out*) : on élimine le plus vieux point tabou et on insère la nouvelle solution. La taille de la mémoire permet de ne pas saturer rapidement les ressources disponibles pour la recherche et permet de surcroît d’adapter facilement la méthode à un espace de recherche dynamique.

***Le critère d’arrêt :***

Le critère d’arrêt sert à déterminer le moment où l’on considère que la solution trouvée est d’assez bonne qualité pour être recevable. On peut par exemple :

- fixer un nombre maximum d’itérations

- après un nombre fixe d’étapes n’ayant pas amélioré la solution *s\**.

- fixer un temps limite après lequel la recherche doit s’arrêter.

A partir de cet algorithme initial, certaines adaptations ont été élaborées. Ces améliorations ont été introduites afin de pallier à des problèmes constatés dans l’analyse de l’exploration de l’espace de recherche. [Glo, Lag, 98] recense toutes ces techniques.

**3.3. Diverses améliorations**

***- la stratégie d’intensification :***

Il s'agit de repérer les éléments faisant partie des meilleures solutions trouvées, qui seront utilisées pour générer de nouvelles solutions, devant être proches de l'optimum. Par exemple, [Cha, Kap, 93] utilise cette technique en repartant de la meilleure solution avec une liste tabou vide. Cet examen approfondi peut permettre de dégager quelques propriétés communes définissant les régions intéressantes de l’espace de recherche. Il est alors aisé d’orienter la recherche vers ces zones « prometteuses » en rendant tabou tous les points menant à sortir de ces régions, ou bien on peut également ajouter une pénalité dans la fonction objectif pour les solutions appartenant à d’autre région. Elle est dite aussi : mémoire a moyen terme (quand une région semble contenir de bonne solution, une procédure intelligente est d’intensifier la recherche dans cette région).

 F(i)

***- la stratégie de diversification :***

D’une manière symétrique, lorsque le processus de recherche parcourt une branche sur une longue période, il est possible de le stopper et de diversifier la recherche sur une autre zone de l’espace. L’algorithme reprend alors généralement sur une autre solution générée aléatoirement. Mais il est possible d’utiliser une stratégie plus fine en mémorisant les solutions les plus fréquemment visitées et en imposant un système de pénalités, afin de favoriser les mouvements les moins souvent utilisés en forçant ainsi l'exploration de nouvelles régions [Fai, Ker, 92], [Roc, Tai, 95]. On la trouve aussi sous le nom : mémoire a long terme (car il n y’a pas la notion d’apprentissage du passé).

 F(i)

Donc la diversification permet de bien couvrir l’espace des solutions, et de déterminer les zones « prometteuses », tant dit que l’intensification permet d’approfondir la recherche, à l’intérieur de chacune des zones prometteuses localisées. [Che, 2000]

La mémoire a court terme se présente sous la forme de la liste tabou ou sont stockés les derniers solutions visités.

***- la stratégie d’aspiration :***

L’utilisation de points tabous peut empêcher, dans certains cas, la méthode tabou d’atteindre une solution intéressante, le critère d’aspiration a été introduit par Glover dans

[Glover et Laguna, 1993] à cet effet et consiste à enlever le statut tabou associé à une transformation si celle-ci permet d’aboutir à une solution meilleure que toutes les solutions trouvées jusqu'à présent. Mais ce critère ne se limite pas à ce cas particulier et il est également possible d’utiliser une fonction d’aspiration dont le but est de toujours aller d’une solution à une solution meilleure. [Pic, 04].

***- la stratégie de détermination de la taille de la liste tabou :***

La taille de la liste tabou est à déterminer empiriquement, ces stratégies peuvent être statiques (détermination de cette taille en fonction de la nature du problème) ou dynamique (Cette méthode adapte la taille de la liste tabou lorsque des cycles ont été détectés dans le parcours de l’espace de recherche par l’algorithme tabou [Bat, Tec, 94]).

En effet la taille de la liste tabou est une donnée primordiale car, une liste trop petite peut conduire à un cycle, alors qu’une liste trop grande peut interdire des transformations intéressantes.

***- Stratégie de sélection du meilleur voisin :***

Elle peut se faire avec la politique *Best-Fit*; au cours de laquelle on choisit le meilleur voisin non tabou, ou bien de la politique *First-Fit* qui consiste a choisir le premier voisin qui satisfait les contraintes tabous. Cette dernière méthode est plus rapide, et elle est souvent réservée à des problèmes dans lesquels la taille du voisinage ne permet pas d’effectuer une évaluation complète (voisinage de grande taille).

***3.4. Parallélisations***

Le développement des réseaux et des machines parallèles a amené plusieurs auteurs à s’intéresser à la parallélisassions de la méthode tabou. L’intérêt de cette évolution réside principalement dans le gain en temps de calcul, mais également dans la définition de nouveaux algorithmes. De nombreux modèles ont été proposés dans la littérature depuis le début des recherches sur la méthode tabou mais on recense principalement deux types de parallélisassions.

***Parallélisme bas-niveau :***

Dans ce type de parallélisations, on considère une machine maître qui exécute un algorithme tabou séquentiel. Les mouvements possibles dans le voisinage du point courant sont évalués à chaque itération en parallèle sur des machines esclaves. Le processus maître reçoit ensuite toutes les évaluations effectuées par les processus esclaves qui lui permettent de prendre une décision quant au chemin à suivre pour la prochaine itération de l’algorithme. Cette technique a notamment été employée dans [Cra, Gen, 97], [Tai, 93] et [Cha, Kap, 93].

Cette méthode de parallélisation est très efficace et l’augmentation du nombre de ressources mises à disposition diminue de manière quasi linéaire le temps de calcul nécessaire à l’obtention de l’optimum, [Cha, Kap, 93].

***Parallélisation par décomposition de l’espace de recherche :***

Ce type de parallélisation est très utilisé dans la recherche tabou. Elle prend en considération plusieurs algorithmes tabou parcourant l’espace de recherche à partir de points initiaux différents ou en utilisant des stratégies différentes. On peut décomposer l’ensemble de ces algorithmes en trois catégories :

1. les algorithmes utilisant un même point de départ mais des stratégies de recherche différentes [Bat, Tec, 1992].

2. utilisant plusieurs points de départ avec la même stratégie d’exploration de l’espace de recherche [Reg, Rou, 92].

3. utilisant à la fois des points initiaux et des stratégies différents [Cra, Gen, 97].

Les algorithmes tabou en concurrence peuvent également agir de manière totalement indépendante (la solution globale étant déterminée à la fin) ou au contraire s’échanger des informations grâce à des communications synchrones.

**4. Différentes applications de la méthode**

La méthode tabou est utilisée en optimisation afin de converger rapidement vers les extrema locaux d’un espace de recherche. C’est une méthode d’exploration locale qui a montré de bonnes aptitudes à la recherche rapide de bonnes solutions à un problème d’optimisation.

La recherche tabou a été adapté a plusieurs problèmes dont en citant :

**La recherche d’information sur Internet** [Pic, 04**] :**

Le problème de recherche d’information sur Internet consiste à parcourir le Web de pages en pages à la recherche de documents pertinents en suivant les liens hypertextes les reliant.

Ce problème peut être modélisé comme un problème d’optimisation : Internet est un graphe où les nœuds représentent les pages Web et où les arcs orientés correspondent aux hyperliens entre les documents.

La fonction d’évaluation nécessaire à la recherche pour déterminer la valeur de la solution trouvée est en réalité une évaluation de la pertinence de la page par rapport à la requête de l’utilisateur.

L’initialisation d’un point de l’espace de recherche se fait par l’intermédiaire de l’opérateur Ocreat qui interroge les moteurs de recherche classiques et évalue un lien donné en résultat. La recherche d’un voisin d’un point S dans l’espace de recherche est réalisée au moyen de l’opérateur Oexplo. L’exploration locale est effectuée en prenant en compte les liens directs du point exploré, c’est-à-dire les documents à une distance de 1 lien.

L’arrêt de la recherche s’effectue de manière automatique lorsqu’un certain nombre de pages Web ont étés traitées. Cependant, il existe d’autres causes menant à la fin de la recherche ; par exemple s’il n’y a plus de page Web issue des moteurs de recherche à explorer.

La recherche de voisin en voisin s’arrête lorsqu’on se trouve face à une page stérile (inefficace), c’est-à-dire sans liens sortants non tabous. Dans ce cas, la recherche reprend typiquement à partir d’une page issue d’un moteur de recherche par l’intermédiaire de l’opérateur Ocreat..

La liste tabou contient toutes les pages déjà fournies par l’opérateur Ocreat.

Et voici l’algorithme adapté à la problématique :

**1)** Définir la fonction d’évaluation f en fonction de la requête de l’utilisateur,

**2)** StopAlgo ← faux

**3)** P ← Ocreat

**4)** Ajouter P à la liste tabou

**5)** tant que (StopAlgo=faux) et la condition d’arrêt n’est pas vérifiée faire

**a)** répéter P’ ← Oexplo(P,1)

**b)** jusqu’à (P’ ∉ liste tabou) ou (il n’existe pas de voisin non tabou de P)

**c)** si il n’existe pas de voisin non tabou de P alors

 **i)** si il n’existe pas de pages issues de moteurs de recherche alors StopAlgo ← vrai

 **ii)** sinon P ← Ocreat

**d)** fin si

**e)** si la liste tabou est pleine alors

 Remplacer le dernier élément de la liste tabou par la solution P’

**f)** sinon Ajouter P’ à la liste tabou

**g)** P ← P’

**6)** fin tant que

**Application pour le Sudoku :**

Le jeu Sudoku a récemment atteint une popularité internationale, le succès de ce jeu vient probablement de la simplicité de ses règles : placer les chiffres de 1 à 9 dans une grille de 9 par 9 de telle sorte que chaque chiffre n’apparaît qu’une seule fois par ligne, par colonne et par région de 3 par 3. Ce problème peut évidemment être considéré comme un problème de satisfaction de contraintes et ainsi résolu par différentes méthodes de recherche locale dont la recherche tabou. Le Sudoku, généralisé à des grilles de n2 x n2 à remplir avec les nombres de 1 à n2, est NP-complet, [Yat, Set, 02].



Un chemin de recherche locale p = (s1, s2, .., sn) avec si une affectation (instanciation de toutes les cellules de la grille), i.e, une grille remplie entièrement. Chaque affectation appartient à l’espace de recherche S, l’ensemble des grilles possibles.

V le voisinage du dernier élément de p tel que V ⊆ N (sn).

le couple (p, V) est la configuration de recherche locale avec p = (s1, s2, .., sn) et V ⊆ N(sn). Les fonctions de voisinage correspondent à des fonctions d’une configuration à une autre (p, V) → (p, V ∪ V’).

La méthode tabou interdit les mouvements vers des affectations déjà visitées lors des l derniers pas de recherche.

Nous utilisons une liste l de taille 10 et les fonctions VoisinageTabou et MouvementMeilleur, VoisinageTabou : p = (s1, . . . , sn) et V’= {s ∈ N(sn)| ∄k, n − l ≤ k ≤ n, sK = s}.

MouvementMeilleur : sn+1 = s’ t.q eval(s’) = mins’’∈V eval(s’’).

De même la recherche tabou est utilisé pour résoudre les problèmes de transport tel que le problème du voyageur de commerce, le design de réseaux, dans des problèmes d’affectation et d’allocation, dans l’optimisation de graphe (coloration de graphes), dans la télécommunication (conception de réseaux, routage d’appels,…), dans la logique et l’intelligence artificielle (reconnaissance et classification de formes, réseaux de neurones).

**5. Avantages et inconvénients**

**5.1. Avantages**

🡪 La recherche tabou a l’avantage d’être facilement paramétrable. Il existe en effet au maximum deux paramètres qui peuvent réellement influencer la recherche. Le premier consiste à bien choisir la taille de la liste tabou, Elle est généralement déterminée empiriquement et varie avec les problèmes, mais c’est une donnée primordiale : une taille trop petite peut amener l’algorithme à boucler sur une zone locale de l’espace de recherche tandis qu’une trop grande taille saturera rapidement les ressources disponibles. Le deuxième est le choix du critère d’arrêt. Ce dernier peut s’avérer difficile à déterminer pour éviter de ne prolonger trop longtemps la recherche ou de rater l’optimum global recherché. [Pic, 04].

🡪 L’efficacité de la méthode Tabou fait qu’elle est largement employée dans les problèmes d’optimisation combinatoire : elle a été testée avec succès sur les grands problèmes classiques (voyageur de commerce, ordonnancement d'ateliers) et elle est fréquemment appliquée sur les problèmes de constitution de planning, de routage, d’exploration géologique, etc. [Bap, 06].

🡪 Possibilité de compromis entre le temps de calcul et la qualité de solution.

🡪 Possibilité d'intégrer des connaissances spécifiques du problème.

**5.2. Inconvénients**

🡪 Toutes les méthodes dites méta-heuristiques [Hao 99], ont une caractéristique commune : l’absence de garantie de résultat. En effet dans la majorité des cas, elles ne disposent en retour aucune information sur la qualité des solutions obtenues.

🡪 Difficulté de prévoir la performance (qualité et temps).

🡪 La méthode Tabou exige une gestion de la mémoire de plus en plus lourde à mesure que l’on voudra raffiner le procédé en mettant en place des stratégies de mémorisation complexe.

**6. Comparaison avec d’autres techniques**

Si on raisonne par rapport à l’usage que font les métaheuristiques de la ***fonction objectif***, on trouve que certaines la laissent « telle quelle » d’un bout à l’autre du processus de calcul, tandis que d’autres la modifient en fonction des informations collectées au cours de l’exploration – l’idée étant toujours de « s’échapper » d’un minimum local, pour avoir davantage de chance de trouver l’optimal. La *recherche tabou*  est un exemple de métaheuristique qui modifie la fonction objectif.

Il y a des métaheuristiques qui ont la faculté de **mémoriser** des informations à mesure que leur recherche avance, et celles qui fonctionnent sans mémoire, en aveugle, et qui peuvent revenir sur des solutions qu’elles ont déjà examinées. Le meilleur représentant des métaheuristiques avec mémoire reste la recherche Tabou.

A l'inverse du recuit simulé qui génère de manière aléatoire une seule solution voisine  à chaque itération, Tabou examine un échantillonnage de solutions de *V(s)* et retient la meilleure *s’* même si *f (s’)>f(s)*. La recherche Tabou ne s'arrête donc pas au premier optimum trouvé.

 La méthode d’acceptation d’un candidat dans la méthode du Recuit simulé est probabiliste tant dit que dans la recherche tabou elle est déterministe.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A. du Recuit Simulé | Méthode Tabou | A. génétique | A. à colonies de fourmis hybridé |
| Facilité d’adaptation |  | - | - | - |
| Connaissance |  | + | + | + |
| Qualité | + | ++ | + | +++ |
| Rapidité | - | - | -- | -- |

Comparaison générale des principales métaheuristiques

Ce tableau compare les principales métaheuristiques que nous avons vues dans les séances précédentes.

Les critères de comparaisons retenus sont les suivants :

- facilité d'adaptation au problème

- possibilité d'intégrer des connaissances spécifiques au problème,

- qualité des meilleures solutions trouvées,

- rapidité, c'est-à-dire temps de calcul nécessaire pour trouver une telle solution

**Conclusion**

La qualité des solutions trouvées par la recherche tabou dépend de l’équilibre à trouver entre un balayage de tout l’espace des solutions (diversification) et une exploration locale poussée (l’intensification). Le choix d’une bonne représentation, d’un bon voisinage, sont également, des facteurs influençant grandement l’efficacité de la méthode.

Les différentes métaheuristiques que nous avons vus (dans d’autres exposés), recuit simulé, algorithmes génétiques, colonie de fourmis et Tabou sont maintenant perçues comme des points de départ au développement de nouveaux algorithmes d’optimisation.

Les différentes caractéristiques de ces métaheuristiques sont recomposées entre elles pour former de nouvelles méthodes, dont ANTabou [Rou, Fon, Rob, 99] qui est une hybridation entre un algorithme d’optimisation par colonie de fourmis et la recherche tabou, qui a obtenu de très bons résultats.

Enfin, la méthode tabou est toujours sous étude en faisant l’objet d’améliorations continuelles.

**Bibliographies**

[Bap, 06] Baptiste Autin, les métaheuristiques en optimization combinatoire, mémoire présente en vue d’obtenir l’examen probatoire en informatique, *Conservatoire National Des Arts Et Metiers,Paris*, 2006.

[Bat, Tec, 92] R. Battitti, G. Tecchiolli, Parallel based search for combinatorial optimization - genetic algorithms and tabu, *Microprocessors and MicroSystems*, 16(7) :351–367, 1992.

[Bat, Tec, 94] R. Battiti, G. Tecchiolli, The reactive tabu search, *ORSA journal on computing*, 6(2), 1994.

[Cha, Kap, 93]  Jaishankar Chakrapani et Jadranka Skorin-Kapov, Massively parallel tabu search for the quadratic assignment problem, Ann. Oper. Res, 41(1-4):327–341, 1993.

[Che, 2000] R.Chelouah,Adaptation aux problèmes à variables continues de plusieurs métaheuristiques d'optimisation combinatoire : la méthode tabou, les algorithmes génétiques et les méthodes hybrides. Application en contrôle non destructif*.* Thèse de Doctorat, *l’Université de Cergy-Pontoise*, décembre 2000.

[Cra, Gen, 97] Teodor Gabriel Crainic, Michel Toulouse, et Michel Gendreau. Toward a taxonomy of parallel tabu search heuristics, *INFORMS Journal on Computing*, 9(1):61–72, 1997.

[Fai, Ker, 92] U. Faigle and W. Kern. Some convergence results for probabilistic tabu search.

ORSA *Journal on Computing*, pages 32-39, 1992.

[Glo, 90] Fred Glover, Tabu Search: A Tutorial, 1990.

[Glo, Lag, 98] FredGlover, Manuel Laguna, Tabu Search, *Kluwer Academic Pub*, 1998.

[Glo, Tai, Wer, 92] Glover, Taillard, Werra, guide de Tabu Search, rapport technique N°ORWP, 92101, Département de mathématique, Ecole Polytechnique, Séderal de LAUSAN, SWIDZERLANT, 1992.

[Hao, 99] J. Hao, P. Galinier & M. Habib. Métaheuristiques pour l’optimisation combinatoire et l’affectation sous contraintes. Revue d’Intelligence Artificielle, vol. 13, no. 2, pages 283-324, 1999.

[Pic, 04] Fabien Picarougne, Recherche d’information sur Internet par algorithmes évolutionnaires, thèse de doctorat, Ecole doctorale : santé, science et technologie, *Université François Rabelais Tours*, 2004.

[Reg, Rou, 92]  C. Rego, C. Roucairol, A parallel tabu search algorithm using ejection chains for the vehicle routing problem, In Meta-Heuristics: Theory and Applications, pages 253–295, Dordrecht, Norwell, MA, 1992. *Kluwer Academic Publishers*, J.P. Kelly L.H. *Osman, editor*.

[Rego,Rou] Sésar Rego, Cathrine Roucail, Rapport de recherche.

[Roc, Tai, 95] Y.Rochat and E.D. Taillard. Probabilistic diversification and intensification in local search for vehicule routing. *Journal of heuristics*, pages 147-167, 1995.

 [Rou, Fon, Rob, 99] Olivier Roux, Cyril Fonlupt, Denis Robilliard, Co-operative improvement for a combinatorial optimization algorithm, In [79], pages: 231,241. 1999.

[Run, Esc] Runo, Escolceif, Métaheuristique et Optimisation Combinatoire.

[Tai, 93] Eric Denis Taillard, Recherches iteratives dirigees paralleles, PhD thesis, *Ecole Polytechnique Fédérale de Lausanne*, 1993.

[Via, Aya, 04] M.A.Viau, J.Ayas, La Recherche Tabou, Novembre 2004.

[Yat, Set, 02] T. Yato, T. Seta, Complexity and Completeness of Finding Another Solution and its Application to Puzzles, In Proc, of *the National Meeting of the Information Processing Society of Japan* IPSJ SIG Notes 2002-AL-87-2, 2002.