

B. SCHNETZLER

**Des opérateurs d'échange et une méthode
de relaxation pour le problème du voyageur
de commerce**

RAIRO. Recherche opérationnelle, tome 26, n° 1 (1992), p. 57-81

http://www.numdam.org/item?id=RO_1992__26_1_57_0

© AFCET, 1992, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
<http://www.numdam.org/>

DES OPÉRATEURS D'ÉCHANGE ET UNE MÉTHODE DE RELAXATION POUR LE PROBLÈME DU VOYAGEUR DE COMMERCE (*)

par B. SCHNETZLER ⁽¹⁾

Résumé. — *Dans cet article, on présente un algorithme pour la résolution approchée du problème du voyageur de commerce dans un espace métrique. Cet algorithme s'appuie sur des opérateurs d'échange d'origine topologique et s'inspire de concepts issus de la physique. Ses performances sont meilleures que celles de l'algorithme de Lin et Kernighan. Par exemple, pour un circuit de 532 sommets, la précision moyenne est de 1 % et le meilleur résultat est l'optimum prouvé. La complexité empirique de l'algorithme est en $O(N^{1.5})$.*

Mots clés : Voyageur de commerce; opérateurs d'échange; relaxation; recuit simulé.

Abstract. — *We give a new approach to find acceptable low-cost minima for the Travelling Salesman Problem, in a metric space. We introduce a new relaxation method, and above all we use topological criteria to define the exchange operators. This approach leads to a new algorithm better than Lin et Kernighan algorithm. For a 532 cities path, the average accuracy is 1% and the best result is the true optimum. Running times grow empirically as $O(N^{1.5})$.*

Keywords : Travelling Salesman problem; exchange operators; relaxation; simulated annealing.

1. INTRODUCTION

Durant ces dernières années, les méthodes connexionistes [1 à 5], méthodes issues d'analogies avec la physique (physique statistique et physique du solide) et la biologie, ont introduit un regain d'intérêt pour le problème classique du *voyageur de commerce*. On rappelle que ce problème consiste à rechercher le chemin fermé (appelé *tour*) le plus court reliant les N sommets d'un ensemble E . Le cas des distributions aléatoires des sommets ou des distances a été abondamment étudié par les physiciens, car il présente des analogies théoriques avec les *verres de spins* (matériaux magnétiques) [6]. Les performances des algorithmes introduits ne sont néanmoins pas acceptables : les temps d'exécution et les écarts vis-à-vis de l'optimum sont trop élevés [7].

(*) Reçu octobre 1990.

(¹) Institut National de Recherche sur les Transports et leur Sécurité, 2, avenue du Général Malleret-Joinville, 94114 Arcueil Cedex, France.

L'objectif du travail entrepris a donc été le développement d'un nouvel algorithme heuristique, plus efficace relativement aux deux critères précédents.

L'algorithme présenté repose sur le principe classique de transformation d'un tour T en un tour T' plus court, par un échange de k arcs du tour T avec k arcs de l'ensemble $S - T$ (S est l'ensemble de tous les arcs possibles). Son efficacité est due à une définition des opérateurs d'échange fondée sur des critères topologiques : les longueurs des arcs sont des distances. On introduit aussi une technique d'ordonnement des opérations d'échange, appelée *relaxation isothermique*, qui repose en partie sur les mêmes principes physiques que le *recuit simulé* [1, 6].

L'organisation de l'article est le suivant. Les principes de l'algorithme sont présentés dans le paragraphe 2, la réalisation est décrite dans le paragraphe 3, et les résultats sont rapportés dans le paragraphe 4. Enfin dans le paragraphe 5, nous étudions les propriétés des tours construits avec des distributions uniformes des sommets.

2. PRINCIPES DE L'ALGORITHME

Soit S l'ensemble de tous les arcs possibles entre les sommets d'un ensemble E . Un tour T est un sous-ensemble de S définissant un chemin fermé passant une fois et une seule par tous les sommets de E . Le problème du voyageur de commerce consiste à rechercher le tour le plus court. La technique employée consiste à partir d'un tour initial quelconque que l'on modifie par une série de transformations : chaque transformation doit diminuer la longueur du tour. La transformation d'un tour T en un tour T' est réalisée par un échange de k arcs du tour T avec k arcs de l'ensemble $S - T$. L'opération d'échange doit être telle que la connexité est conservée.

La méthode d'obtention du chemin initial est décrite dans les paragraphes 2.2 et 2.3. Les opérateurs d'échange sont présentés dans les paragraphes 2.1, 2.4 et 2.5. La méthode d'ordonnement des opérations d'échange est décrite dans le paragraphe 2.6.

2.1. Séparations des opérations locales et globales

Si on considère le tour optimal, la probabilité qu'un sommet d'être lié avec un sommet éloigné est négligeable. On appelle plus proches voisins d'un sommet, ceux qui ont une probabilité non négligeable d'être liés avec lui dans le tour optimal. Pour un sommet donné, on appelle interaction à courte portée, une opération (*opération locale*) qui n'implique que les plus proches

voisins de ce sommet; on appelle interaction à longue portée, une opération (*opération globale*) qui implique des sommets ne faisant pas partie des plus proches voisins. Compte tenu du caractère fortement non local [6] de la définition d'un tour, dû à la contrainte de connexité, les interactions à longue portée sont d'une grande importance. Ces opérations, locales ou globales, correspondent à des configurations topologiques particulières, et peuvent être réparties dans un certain nombre de *classes* prédéfinies (voir 2.4). A chacune de ces classes on associe un opérateur.

En théorie, la complexité algorithmique d'une opération dont la portée est R est en $O(n!)$, où n désigne le nombre de sommets présents dans ce rayon R . En pratique, pour les opérations locales on fixe le nombre de plus proches voisins (le coût unitaire d'une opération est borné), et on peut considérer que la complexité de ces opérations sera linéaire en fonction de n . La complexité des opérations globales, difficile à estimer théoriquement, dépend de la classe de l'opération et se révèle très inférieure à $O(n!)$.

L'algorithme comprend trois étapes. La première étape est une phase de recherche des plus proches voisins de chaque sommet. La seconde étape correspond à la construction du plus court chemin que l'on puisse obtenir à partir du seul calcul d'optima locaux (complexité linéaire). La troisième étape est une transformation du chemin obtenu, à l'aide d'opérations locales et d'opérations globales. On recherche simultanément des optima locaux et un optimum global. La recherche indépendante des optima locaux est une application du principe *diviser pour régner*.

2.2. Détermination des plus proches voisins

Soient E l'ensemble des sommets, $V_i(R)$ l'ensemble des voisins d'un sommet i dans un rayon R , et d_{ij} la distance entre les sommets i et j

$$V_i(R) = \{j \in E : d_{ij} < R\}$$

En pratique il est plus commode de définir l'ensemble des plus proches voisins en fonction du nombre de ses éléments : on définit alors R comme une fonction de n . On appelle $V_i(R_n)$ l'ensemble des n plus proches voisins du sommet i .

$$V_i(R_n) = \{j \in E : d_{ij} < R_n\} \quad \text{avec cardinal } (V_i(R_n)) = n$$

Dans le cas de la géométrie euclidienne à deux dimensions (*fig. 1*), la détermination des N plus proches voisins peut être réalisée très simplement. On définit un maillage rectangulaire quelconque de l'espace (en pratique, il

suffit que le nombre de mailles soit supérieur au nombre de sommets). Un sommet appartient nécessairement à une maille, donc à une cellule d'une matrice; plusieurs sommets peuvent appartenir à la même maille. La recherche des plus proches voisins d'un sommet est effectuée en considérant les cellules voisines dans la matrice. On augmente le rayon de la recherche jusqu'au moment où on trouve le nombre souhaité de plus proches voisins.

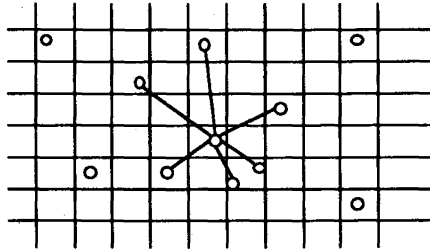


Figure 1. — Détermination des plus proches voisins d'un nœud.

Cette démarche peut être généralisée sans difficulté dans le cas d'un espace à p dimensions. En revanche la méthode est difficilement applicable si la distance est non euclidienne ($d_{ij} \neq \sqrt{\sum_p (x_{ip} - x_{jp})^2}$): les sommets présents dans les cellules voisines ne sont pas nécessairement des sommets proches au sens de la distance. Dans ce dernier cas on peut pour chaque sommet i calculer toutes les distances d_{ij} , et opérer un tri pour rechercher les plus proches voisins. Dans la suite de l'algorithme, la méthode est indépendante de la nature de la distance et de la dimension de l'espace.

2.3. Fabrication d'un chemin initial

La construction d'un chemin initial est réalisée à partir d'un algorithme récursif (la procédure *faire*). A chaque étape, pour tous les sommets introduits dans le chemin à l'étape précédente, on introduit dans le chemin tous leurs voisins ne faisant pas déjà partie de celui-ci. On part d'un chemin contenant un seul sommet. Le critère d'arrêt est l'absence de sommets hors du chemin.

```

PROCEDURE faire (
  IN OUT chemin          -- chemin courant
  IN  vague,             -- nœuds entrés dans le chemin à l'étape précédente
  IN  dernier,          -- nombre de nœuds entrés à l'étape précédente
)
  nouvelle_vague        -- tableau des nœuds entrés à l'étape courante
  nouveau_dernier       -- nombre de nœuds entrés à l'étape courante
BEGIN
  nouveau_dernier := 0 ;

```

```

FOR v IN 1 .. dernier LOOP
  LOOP
    voisin := voisin_suivant_et_hors_du_chemin (vague(v))
    EXIT WHEN voisin = NULL
    insertion(voisin, chemin)
    nouveau_dernier := nouveau_dernier + 1
    nouvelle_vague (nouveau_dernier) := voisin
  END LOOP
END LOOP
IF nouveau_dernier /= 0 THEN faire (chemin, nouvelle_vague, nouveau_dernier)
END faire
-- Construction du chemin : au départ le chemin ne contient qu'un seul sommet
dernier := 1
vague (dernier) := sélection_aléatoire_d_un_sommet
chemin (1) := vague (dernier)
faire (chemin , vague, dernier)

```

Pour les besoins de l'expérience, en sélectionnant aléatoirement le premier sommet du chemin il est possible de produire (au plus) autant de chemins initiaux différents qu'il existe de sommets. Il est possible d'augmenter le nombre de chemins différents produits en utilisant différentes procédures d'insertion d'un sommet dans le chemin : suivant qu'on recherche ou non un optimum au moment de l'insertion, on peut ainsi faire varier sensiblement la longueur du chemin initial.

L'opération permettant de construire le chemin est l'insertion dans le chemin de tous les voisins d'un sommet. Le nombre de plus proches voisins étant fixe, le coût de l'opération est borné. L'opération est réalisée une fois pour chaque sommet. La complexité de l'algorithme est donc linéaire.

2.4. Opérateurs d'échange

Traditionnellement les opérateurs sont définis comme des opérations d'échange sur les éléments d'un ensemble [8], permettant de passer d'une permutation à une autre. Une opération est valide si elle diminue la longueur du tour associé à la permutation. Notre approche consiste à définir des classes d'opérateurs à partir de critères topologiques.

Opération de déplacement (fig. 2.1 a)

Le déplacement d'un sommet consiste à rompre ses liaisons (2 arcs) ainsi qu'une liaison reliant deux de ses voisins, puis à relier (1 arc) les deux voisins initiaux et à insérer (2 arcs) le sommet entre ses deux nouveaux voisins. L'opération est valide si elle diminue la longueur du tour. C'est une opération locale. Sa complexité dépend linéairement du nombre de plus proche voisins.

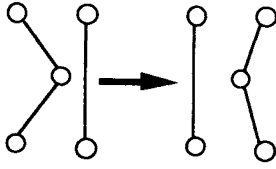


Figure 2.1a

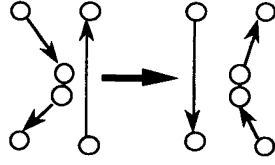


Figure 2.1b

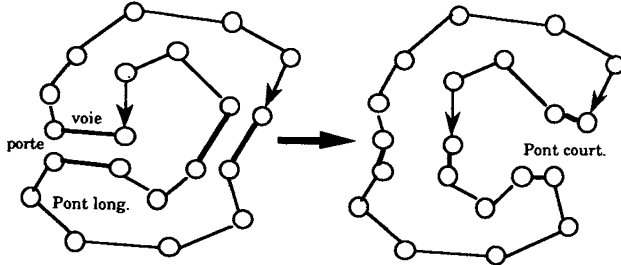


Figure 2.2a

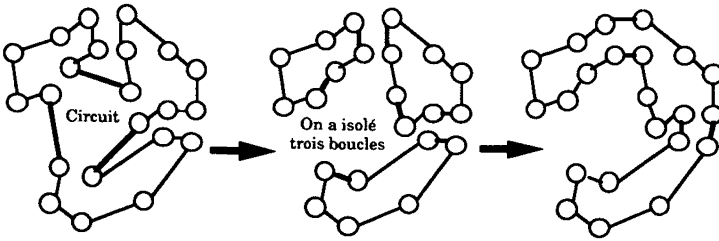


Figure 2.2b

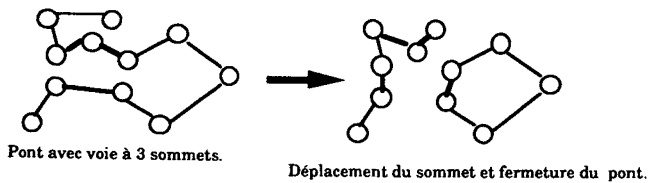


Figure 2.3

Opération de transfert de pont (fig. 2.2 a)

On appelle *porte* un couple de deux sommets proches mais non reliés. On appelle *pont* deux portes reliées entre elles par deux arcs, appelés *voies*, de sens opposés. Fermer un pont consiste à détruire les deux voies et à relier entre eux les sommets des deux portes. La fermeture d'un pont découpe alors le chemin en deux boucles. Pour rétablir la connectivité, il est nécessaire de les

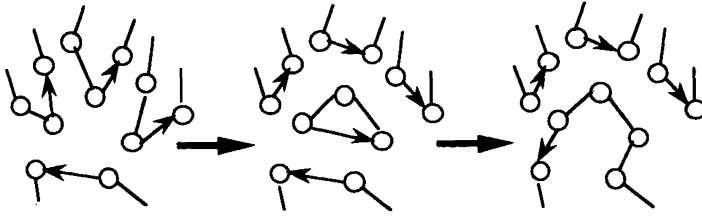


Figure 2.4a

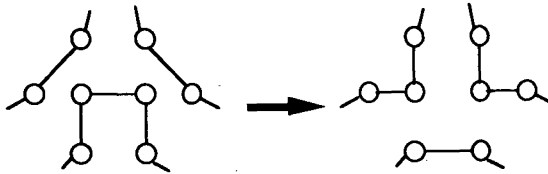


Figure 2.4b

Figure 2 (ci-dessus et ci-contre). — Opérateurs d'échange.

rattacher en un autre endroit en ouvrant un pont. Si la longueur des voies d'un pont est supérieure à la longueur des arcs qui joindraient deux à deux leurs extrémités, la fermeture du pont diminue la longueur du chemin. Par convention on appelle *longueur du pont* le gain de longueur provoqué par la fermeture du pont. Le transfert de pont est donc une optimisation globale qui consiste à rompre l'attachement d'une boucle en un endroit du chemin (*pont long*), pour la rattacher par un pont de moindre longueur en un autre lieu (*pont court*).

Au moment du rattachement, il est éventuellement nécessaire de changer le sens du chemin sur une des boucles (à cause des croisements du chemin initial).

D'après les résultats expérimentaux, la complexité de l'opération est en $O(N^{1.5})$. La recherche sur le chemin de la présence des ponts de longueur positive est en $O(N)$, et leur fréquence (empirique) est en $O(N)$. Une fois trouvé un premier pont permettant de créer deux boucles, on recherche un second pont permettant de rattacher ces deux boucles (si ce pont n'existe pas aucune opération n'est réalisée). Le coût de cette opération est donc en $O(N^{0.5})$.

Généralisation du déplacement (fig. 2.1 b)

Au lieu de déplacer un sommet, on déplace un couple de sommets, et on inverse le sens du parcours entre les deux sommets. Cette opération se justifie

quand le déplacement du couple ne peut être décomposé en deux opérations de déplacement individuel. On peut définir des opérations de déplacement d'ordre plus élevé, en considérant des séquences de plusieurs sommets. La généralisation du déplacement est d'un intérêt théorique limité, car elle n'est qu'un cas particulier du transfert de pont, où le pont long et le pont court se rattachent aux mêmes sommets. Elle présente néanmoins un intérêt pratique pour les ordres 2 et 3 : ces situations sont relativement fréquentes, et l'opération est moins complexe que l'opération de transfert de ponts. C'est une opération locale.

Généralisation du transfert de pont (fig. 2.2b)

On généralise le mécanisme de transfert de pont au cas où on a un *circuit* (pont à plusieurs entrées) formé de trois portes reliées entre elles. La fermeture des portes du circuit découpe le chemin en trois boucles. On rattache, par l'intermédiaire d'un pont, la boucle la plus courte à une des deux autres. On rattache ensuite la boucle restante à la boucle nouvellement formée. On définit des opérations de transfert d'ordre plus élevé en considérant des circuits formés d'un plus grand nombre de portes. Ces opérations ne présentent un intérêt que dans le cas où elles ne peuvent être décomposées en opérations de transfert d'ordre moins élevé.

Extension de la notion de pont, opérateurs mixtes (fig. 2.3)

Il est possible d'étendre la notion de *pont* en considérant des *voies* de plus de deux sommets : les portes sont reliées par une succession d'arcs. Dans ce cas la fermeture des *portes* aux extrémités du pont isole les sommets intermédiaires, qu'il devient nécessaire de déplacer pour les rattacher au chemin. Dans le cas où il n'est pas possible de décomposer la série d'opérations (les déplacements puis le transfert de ponts), il est nécessaire d'utiliser un opérateur réalisant toutes les opérations simultanément. L'extension de la notion de pont conduit donc à un opérateur dont la mise en œuvre suppose la définition de deux paramètres : le *paramètre d'ordre* indique le nombre de portes d'un circuit, et le *paramètre de longueur* le nombre maximal de sommets d'une voie. La complexité d'un tel opérateur tient à l'impossibilité de connaître *a priori* la longueur des voies impliquées dans la définition d'un pont.

A titre expérimental, dans le cas d'un circuit à 2 portes on a autorisé le paramétrage de la longueur de la seconde voie d'un pont : celle-ci peut avoir 2 ou 3 sommets. Sauf indication contraire cet opérateur n'est pas employé.

Extensions de la notion de déplacement (fig. 2.4)

L'extension de la notion de déplacement conduit à deux nouveaux opérateurs locaux. L'expérience a montré que ces opérateurs ont un effet négatif sur les performances de l'algorithme. Ils ne sont donc pas utilisés.

Le premier opérateur (*fig. 2.4a*) rompt les liaisons de n sommets non consécutifs, et forme une boucle avec ces n sommets. La boucle est ensuite rattachée au chemin.

Le second opérateur (*fig. 2.4b*) rompt toutes les liaisons de n sommets consécutifs sur le chemin, et déplace individuellement chacun de ces sommets. Chaque sommet a une destination différente.

Pour comparaison on donne les opérateurs d'échange utilisés par la plupart des algorithmes : opérateurs 2-OPT (*fig. 3a*) et 3-OPT (*fig. 3b*). L'algorithme de Lin et Kernighan [8] généralise l'opération d'échange pour un nombre quelconque d'arcs, mais il ne définit pas *a priori* des classes d'opérateurs. L'objet des heuristiques de cet algorithme est, à chaque itération, de rechercher une opération k -OPT (k n'étant pas fixé) permettant de passer d'un tour T à un tour T' plus court. On rappelle qu'il nous a été possible de définir *a priori* des classes d'opérateurs (cas particuliers de k -OPT), parce que nous avons considéré que les longueurs des arcs étaient des distances. Une application d est une distance si elle vérifie les axiomes suivants :

- $d_{ij} = d_{ji}$
- $d_{ij} = 0 \Leftrightarrow i = j$
- $d_{ij} \leq d_{ik} + d_{kj}$

Si une de ces propriétés cesse d'être vérifiée les considérations qui ont présidé à la conception des opérateurs ne sont plus justifiées.

On remarquera que, dans le cas symétrique, les interactions à longue portée sont des opérations impliquant nécessairement au moins 4 arcs. Dans le cas non symétrique ($d_{ij} \neq d_{ji}$), une opération 2-OPT correspond déjà à une interaction à longue portée : la variation de longueur du tour dépend de l'inversion du sens sur une portion du chemin (*fig. 3a*).

$$\Delta L = -d_{i, i+1} - d_{j-1, j} + d_{i, j-1} + d_{i+1, j} + \sum_{k=j-1}^{i+2} (d_{k, k+1} - d_{k+1, k})$$

Le caractère non local du problème est alors fortement accru, et il devient nécessaire de repenser la conception des opérateurs.

Si d_{ij} est symétrique mais ne vérifie pas la condition $d_{ij} \leq d_{ik} + d_{kj}$ (on dit alors que d_{ij} est un indice de dissimilarité) il peut être intéressant de remplacer

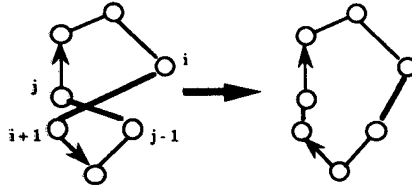


Figure 3.a

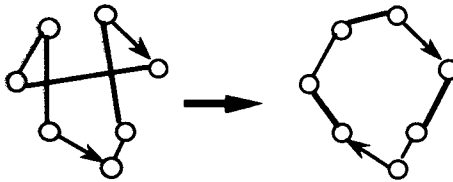


Figure 3.b

Figure 3. – Opérateurs d'échange 2-OPT et 3-OPT.

un seul arc par deux arcs. Si l'indice de dissimilarité n'est pas trop éloigné d'une distance, il peut être envisageable d'adapter l'algorithme en repensant la conception des opérateurs.

2.5. Optimalité d'un tour

Sur un ensemble E de N éléments, on définit une permutation $P=(1, 2, \dots, N)$ et le tour $T=(1, 2, \dots, N, 1)$ associé, de longueur

$$L = d_{N,1} + \sum_{i=1}^{N-1} d_{i,i+1}$$

où $d_{i,j}$ est la distance entre les sommets i et j . Un tour est dit λ -OPT (λ -optimal) si on ne peut obtenir un tour plus court en remplaçant λ arcs du tour par un autre ensemble de λ arcs. Un tour λ -optimal est donc λ' -optimal si $\lambda' \leq \lambda$. Un tour de N sommets est optimal si et seulement si il est N -OPT. L'optimalité d'un algorithme déterministe dépend donc de l'ordre (λ) de ses opérateurs d'échange; néanmoins en pratique on atteint une valeur proche de l'optimum avec des opérateurs d'un ordre peu élevé.

Indépendamment de l'optimalité, les opérateurs d'échange d'un ordre élevé présentent un autre intérêt : si $\lambda' \leq \lambda$, une seule opération λ -OPT peut, dans certains cas, se substituer à plusieurs opérations λ' -OPT. Des opérateurs λ -OPT permettent d'atteindre plus rapidement un λ' -optimum ($\lambda' \leq \lambda$). On peut ainsi, dans l'exemple ci-dessous, passer du tour $(1, 2, 3, 4, 5, 6, 1)$ au

tour (1, 3, 4, 5, 2, 6, 1) par une seule opération 3-OPT (déplacement du sommet 2), ou par la succession de 2 opérations 2-OPT (fig. 4).

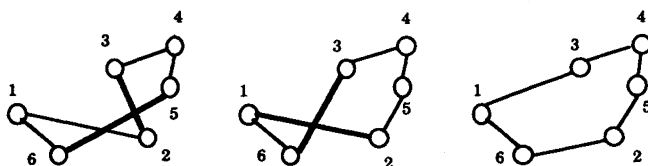


Figure 4. — Succession de deux opérations 2-OPT.

2.6. La conduite de la relaxation isothermique

On emploie le terme de *relaxation* par analogie avec la physique : la relaxation est le retour à l'état d'équilibre (l'optimum) d'un système précédemment dans un état excité.

L'ordonnancement des opérations de relaxation se fait en essayant de maintenir un équilibre entre les diverses parties du chemin. On commence par examiner les possibilités de suppression des arcs les plus longs : les possibilités de suppression d'un arc sont définies par les opérateurs d'échange (déplacement, et transfert de pont généralisé jusqu'au 4^e ordre). On parcourt donc le chemin en essayant d'appliquer un opérateur pour tous les sommets dont la longueur de l'arc de départ dépasse un seuil fixé à l'avance (fig. 5).

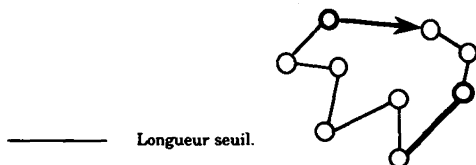


Figure 5. — Sommets (arcs) justifiant une tentative d'application d'un opérateur.

La stratégie suivie consiste à effectuer un *parcours* du chemin pour chaque type d'opérateur. Un *parcours* consiste en la visite de N sommets du tour (un sommet peut être visité 0 ou plusieurs fois).

```
-- parcours du chemin pour l'opérateur de déplacement
PROCEDURE relaxation_deplacement (
  IN      seuil,      -- seuil définissant les arcs à considérer
  IN OUT longueur    -- longueur du chemin
)
iterations := 0
noeud      := un_des_sommets_du_tour
LOOP
```

```

    IF distance (noeud, noeud.suivant) > seuil THEN
        tentative_de_deplacement (noeud, longueur)
    END IF
    noeud := noeud.suivant
    iterations := iterations + 1
    EXIT WHEN iterations = nombre_de_sommets
END LOOP
END relaxation_deplacement

```

On appelle *cycle* la succession des parcours du chemin réalisés pour chacun des opérateurs.

```

-- cycle de relaxation (4 parcours du chemin pour les 4 opérateurs de la relaxation)
relaxation_deplacement (seuil, longueur_chemin)
relaxation_pont_2_portes (seuil, longueur_chemin)
relaxation_pont_3_portes (seuil, longueur_chemin)
relaxation_pont_4_portes (seuil, longueur_chemin)

```

Après chaque parcours on recalcule la valeur du seuil, en fonction de la longueur moyenne des arcs du chemin et du *taux courant*. Un taux courant de 110 % signifie que le seuil est une longueur égale à 110 % de la longueur moyenne des arcs du chemin. Au cycle suivant la valeur du taux est abaissée (exemple : on passe de 110 à 109 %). Une *phase de relaxation* est arrêtée quand, entre deux cycles, la variation relative de longueur du chemin est inférieure à une certaine précision.

```

PROCEDURE ordonne (
    IN      taux_initial,      -- seuil
    IN      reduction_taux,    -- vitesse d'abaissement du seuil
    IN OUT  longueur,         -- longueur du chemin
    IN      precision         -- critère d'arrêt
)
    taux_courant := taux_initial
    derniere_longueur := longueur
    LOOP
        seuil := longueur * taux_courant / nombre_de_sommets
        relaxation_deplacement (seuil, longueur_chemin)
        seuil := longueur * taux_courant / nombre_de_sommets
        relaxation_pont_2_portes (seuil, longueur_chemin)
        seuil := longueur * taux_courant / nombre_de_sommets
        relaxation_pont_3_portes (seuil, longueur_chemin)
        seuil := longueur * taux_courant / nombre_de_sommets
        relaxation_pont_4_portes (seuil, longueur_chemin)

        taux_courant := taux_courant - reduction_taux

        EXIT WHEN (derniere_longueur - longueur) / longueur < precision
    END LOOP
END ordonne

```

L'algorithme comprend plusieurs phases de relaxation avec un pas d'abaissement du seuil de plus en plus grand. L'abaissement rapide du seuil permet d'augmenter le nombre de nouveaux sommets à examiner à chaque cycle. En fin de relaxation, compte tenu du petit nombre de sommets pouvant être impliqués dans des opérations et du faible gain relatif apporté par chaque opération, cette augmentation est indispensable. En effet la probabilité qu'un arc déjà court puisse être remplacé par un arc plus court est faible. Si cette opération est néanmoins possible, le gain à en attendre sera de toute manière peu élevé (le nouvel arc ne pourra pas être beaucoup plus court). Pour les mêmes raisons le critère d'arrêt à une valeur plus élevée en début de relaxation (moins grande précision) qu'en fin de relaxation.

```

PROCEDURE relaxation (
    IN      taux_initial,
    IN      reduction_initiale,
    IN OUT  longueur_chemin
)
    taux_courant      := seuil_initial
    reduction_cycle   := reduction_initiale
    WHILE taux_courant > 0.0 LOOP
        ordonne (taux_courant, reduction_taux, longueur_chemin, précision)
        reduction_taux      := reduction_taux * acceleration
        taux_courant        := taux_courant - reduction_cycle
    END LOOP
    -- phase de relaxation finale
    ordonne (0.0, 0.0, précision_finale)
END relaxation

```

En faisant abstraction des unités physiques, on assimile la longueur des arcs à une température ou à une énergie. La relaxation du chemin est alors équivalente à un processus de refroidissement. On donne le nom de *relaxation isothermique* à cette technique, car elle consiste à sélectionner les arcs à supprimer parmi ceux dont la longueur dépasse un certain seuil (l'isotherme). L'objectif est d'abaisser régulièrement et uniformément la température dans tout l'espace. Les principes physiques et les conditions de mise en œuvre sont donc ceux des algorithmes de recuit simulé, aussi peut-on consulter les références [1, 6]. L'idée essentielle est que, pour éviter la production d'un optimum local, le processus de relaxation doit être suffisamment lent et régulier pour avoir le temps d'établir un équilibre :

- l'énergie de l'état initial doit être élevée;
- la vitesse de refroidissement ne doit pas être trop élevée.

La technique de relaxation isothermique, étant un processus déterministe (à l'opposé de la technique de recuit simulé), autorise une énergie initiale élevée. Pour des chemins initiaux dont la longueur est comprise entre 130 et

300 % de la longueur optimale, les temps de calcul ne diffèrent que de quelques pour cent, et la précision que de quelques millièmes.

De plus les opérateurs de transfert, établissant les échanges à longue portée, permettent d'échapper plus facilement à un optimum local. L'algorithme est donc peu sensible à la géométrie du tour initial.

3. RÉALISATION ET PARAMÉTRAGE DE LA RELAXATION

3.1. Paramétrage et optimisation de l'algorithme

Les valeurs des paramètres sont le résultat de compromis issus de résultats expérimentaux. Il est bien évident que ces valeurs ne sont pas indépendantes entre elles, et qu'elles sont aussi fonction de la distribution des sommets, du chemin initial, etc. Dans l'ensemble, seuls les ordres de grandeur ont une importance significative sur la précision des résultats obtenus :

- le taux initial définit le nombre d'arcs à prendre en compte lors du premier cycle de la relaxation;
- le paramètre de réduction initiale du seuil définit le nombre de sommets introduits à chaque cycle;
- la précision est un critère d'arrêt pour un cycle de relaxation;
- l'accélération définit le nouveau pas d'abaissement du seuil en fin de relaxation. C'est une valeur sans grande importance, et qui peut être très élevée, car la quasi-totalité de la relaxation est réalisée dans la première phase. Pour une distribution uniforme des sommets, la première phase correspond à des taux compris entre 110 et 100 %.

TABLEAU I
Paramètres de l'algorithme de relaxation.

	Distribution uniforme	Distribution non uniforme
Longueur tour initial.	160 % tour optimal	160 % tour optimal
Voisins.	16	20
Taux-initial.	1,1 (110 %)	1,1 (110 %)
Réduction-initiale.	0,01	0,01
Précision.	0,005	0,005
Précision-finale.	0,0005	0,0005
Accélération.	10,0	10,0

Dans certains cas l'opération de transfert de pont effectue des optimisations qui ont un caractère local. Compte tenu du coût de cet opérateur, il est préférable d'employer des opérateurs locaux spécialisés réalisant les mêmes

optimisations. Ce sont les opérateurs de déplacement généralisé, ainsi que l'opérateur 2-OPT. Cette dernière opération est en effet un cas particulier du transfert de pont : le *pont long* et le *pont court* prennent appui sur les mêmes sommets, et on inverse le sens du chemin sur une des boucles.

relaxation_deplacement_1_sommet	(seuil, longueur)
relaxation_2_OPT	(seuil, longueur)
relaxation_deplacement_2_sommets	(seuil, longueur)
relaxation_deplacement_3_sommets	(seuil, longueur)

3.2. Le logiciel

Le logiciel est écrit en langage ADA et compte environ 1000 instructions ⁽¹⁾. Pour comparer les temps d'exécution avec les algorithmes concurrents [7, 9] écrits en FORTRAN, la génération du code exécutable a été réalisée en omettant les opérations de contrôle (exemple : débordement de tableau, pointeur non nul, etc.) prévues par le langage ADA, mais absentes des autres langages.

Pour une exécution donnée, le paramètre d'ordre indique les opérateurs employés. La relaxation d'ordre 1 utilise uniquement des opérateurs locaux. Pour les ordres supérieurs on utilise les opérateurs locaux, plus les opérateurs de transfert généralisé jusqu'à l'ordre spécifié (exemple : à l'ordre 3, on considère les circuits de 2 et de 3 portes).

4. RÉSULTATS

Le paragraphe 4.1 est une comparaison des performances de l'algorithme à celles des algorithmes de référence. Pour une revue détaillée des algorithmes voir D. Johnson [9]. Le paragraphe 4.2 est une description et une critique d'un algorithme fondé sur la technique de recuit simulé. La comparaison des résultats avec la relaxation isothermique est intéressante, car ces deux algorithmes ont de nombreux points en commun.

4.1. Comparaison avec les méthodes classiques

L'algorithme de relaxation est comparé avec celui de Lin et Kernighan, qui semble être le « meilleur algorithme connu » [7, 9] pour une résolution approchée. Le résultat le plus important est que, pour une distribution non

⁽¹⁾ Le logiciel est disponible sur simple demande auprès de l'auteur. Il est un peu plus complexe que sa représentation dans cet article.

uniforme de 532 sommets, le meilleur résultat de la relaxation isothermique est l'optimum. Les résultats pour 10 000 sommets montrent aussi que la précision des résultats est insensible au nombre de sommets.

La complexité empirique de l'algorithme a été mesurée avec des distributions uniformes de 1 000 à 10 000 sommets. Elle est en $O(N^{1.5})$ pour la relaxation d'ordre 2, en $O(N^{1.5})$ pour l'ordre 3, et en $O(N^{1.6})$ pour l'ordre 4. La comparaison avec d'autres algorithmes est extrêmement difficile, car la complexité dépend de la nature de la distribution des longueurs des arcs (e.g. : distance ou indice de dissimilarité, distribution uniforme ou non uniforme des sommets, etc.). Miller *et al.* [10] donnent des exemples très significatifs. Ceci-dit, la complexité (pour des distances) de l'algorithme de Lin et Kerningham [8] est en $O(N^{2.2})$. Il existe cependant des versions plus récentes et plus performantes de cet algorithme. La complexité (pour des

TABLEAU II

Résultats avec une distribution non uniforme de 532 sommets [11], et une distribution uniforme de 10 000 sommets dans un carré de côté unité [7]. Les temps de calcul donnés par [7] et [11] ont été « normalisés » en fonction de la puissance (7 MIPS) de l'ordinateur (Apollo DN4500) que nous avons utilisé. On indique entre parenthèses les valeurs données par les auteurs : temps de calculs moyens et ordinateur employé.

	532 sommets (50 essais)	10 000 sommets
Optimum (Padberg et Rinaldi).	$L = 27,686$ $T = 600$ mn (3 600 mm-VAX 11/780)	$L \approx 75,0$ [évaluation théorique voir 5.1]
Algorithme de Lin et Kerningham.	$L = 28,150$ à $29,143$? ?	? $T = 40$ mn (> 60 mn-VAX 8550)
Algorithme de relaxation d'ordre 1.	$L = 28,277$ à $29,542$ $\langle L \rangle = 28,714$ $\langle T \rangle = 0,25$ mn	$L = 78,4$ $T = 5$ mn
Algorithme de relaxation d'ordre 2.	$L = 27,793$ à $28,671$ $\langle L \rangle = 28,270$ $\langle T \rangle = 0,30$ mn	$L = 75,20$ $T = 14$ mn
Algorithme de relaxation d'ordre 3.	$L = 27,726$ à $28,549$ $\langle L \rangle = 28,116$ $\langle T \rangle = 0,40$ mn	$L = 74,6$ $T = 25$ mn
Algorithme de relaxation d'ordre 4.	$L = 27,685$ à $28,575$ $\langle L \rangle = 28,058$ $\langle T \rangle = 0,50$ mn	$L = 74,6$ $T = 53$ mn
Algorithme de relaxation.	$L = 27,717$ à $28,297$ $\langle L \rangle = 27,976$ $\langle T \rangle = 0,75$ mn	$L = 74,6$ $T = 52$ mn
- ordre 4.		
- voie à 3 sommets.		
Recuit simulé 2-OPT.	$L = 28,258$ à $29,564$ $\langle L \rangle = 28,833$ $\langle T \rangle = 2,30$ mn	$L = 97,6$ $T = 56$ mn
Recuit simulé 2-OPT et 3-OPT.	$L = 27,765$ à $28,866$ $\langle L \rangle = 28,321$ $\langle T \rangle = 4,0$ mn	$L = 82,3$ $T = 108$ mn

indices de dissimilarité) d'une de ces versions a été estimée en $O(N^{1.4})$ à partir des résultats donnés par D. Johnson [9].

L'algorithme de Padberg et Rinaldi [11] est conçu pour donner une solution exacte au problème du voyageur de commerce : le résultat est l'optimum. Une particularité importante de cet algorithme est qu'il nécessite comme données initiales un ensemble de tours partiellement optimisés (50 tours obtenus par l'algorithme de Lin et Kerningham). A partir de ces données initiales, pour 532 sommets l'algorithme obtient le tour optimal en 600 minutes. Ce temps doit être comparé aux 45 secondes nécessaires à la relaxation isothermique pour obtenir en moyenne un résultat proche de 1 % de l'optimum, ou aux 37 minutes (50 essais) pour obtenir un résultat proche de 0,1% de l'optimum.

4.2. Comparaison avec les méthodes connexionistes

Les méthodes connexionistes regroupent des travaux d'origines très diverses [12, 13], qui semblent n'avoir pour point commun que la prétention à être adaptés à des architectures parallèles. Les objectifs de ces algorithmes dépassent généralement la résolution du seul problème du voyageur de commerce; ce qui rend évidemment la comparaison difficile. Néanmoins si on s'intéresse uniquement à ce problème on constate que seuls les algorithmes spécifiques bâtis autour de la technique du recuit simulé peuvent prétendre à des résultats significatifs.

L'algorithme de Kirkpatrick [1] est un algorithme de recuit simulé. Il procède par relaxation, d'un chemin initial en utilisant l'opérateur 2-OPT, ou les opérateurs 2-OPT et 3-OPT (déplacement de sommets) dans notre version.

Schéma de l'algorithme de recuit simulé :

- la seule opération d'échange est le déplacement d'un sommet
- la conduite de la relaxation est effectuée de la manière suivante :
 - on sélectionne aléatoirement une configuration correspondant à un déplacement.
 - soit ΔL la variation de longueur du chemin produite par le déplacement, et T et K deux paramètres de la relaxation. Le paramètre T est une variable qui a la dimension d'une longueur, et K est une constante sans dimension.
 - règle de transition :
 - $\Delta L < 0$: le déplacement est effectué

$\Delta L > 0$: le déplacement est réalisé avec la probabilité $p = K \exp(-\Delta L/T)$. En autorisant une certaine fluctuation de la longueur du chemin, il est possible de donner une certaine « énergie » à un sommet lui permettant de franchir une barrière et d'échapper ainsi à un optimum local.

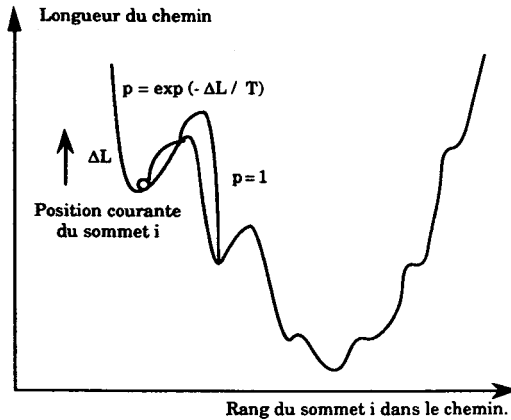


Figure 6. — A un instant t donné, la courbe indique la longueur du chemin en fonction de la position d'un sommet i dans le chemin (position courante ou position issue du déplacement envisagé).

- au cours du temps, on abaisse lentement la valeur de T : la hauteur des barrières que peut franchir un sommet diminue.

L'algorithme de Bonomi et Lutton [2], optimisation de celui de Kirkpatrick, est le meilleur des algorithmes connus de l'auteur [1, 3, 4, 5]. De même que l'algorithme de relaxation isothermique, il ne considère que des échanges de couples (2-OPT) d'arcs voisins. Les excellentes performances annoncées [2], dans le cas particulier d'une distribution uniforme des sommets, sont cependant trompeuses. Elles proviennent de l'utilisation d'un tour initial très proche du tour final : l'espace est décomposé en pavés (4 096 pavés pour 10 000 sommets) entre lesquels on impose certaines liaisons horizontales et verticales (fig. 7). L'efficacité de l'algorithme est donc uniquement due à la rupture de la *symétrie de rotation* [6] (distribution uniforme des sommets), et non à la technique de recuit simulé ou au principe *diviser pour régner* (restriction aux couples d'arcs voisins). L'algorithme de recuit simulé, utilisant le même chemin initial que celui employé par la relaxation isothermique, ainsi que la même technique de calcul des plus proches voisins, donne des résultats très inférieurs (tableau II).

Pour 532 sommets, on remarquera quand même le bon résultat obtenu par la technique de recuit simulé (meilleur résultat à 0,28 % de l'optimum, à

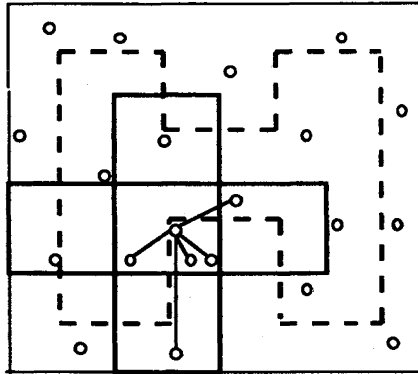


Figure 7. – Aire de recherche des plus proches voisins et géométrie du tour initial (trait pointillé) pour un pavage de 16 rectangles.

comparer au 1,7 % de Lin et Kerninghan) avec les opérateurs 2-OPT et 3-OPT. En revanche dès que le nombre de sommets est élevé, l'efficacité de l'algorithme de recuit simulé est médiocre.

Cette faible efficacité peut être attribuée aux problèmes combinatoires apparaissant avec les interactions à longue portée : statiquement on doit considérer un plus grand nombre de combinaisons (opérateurs k -OPT, $k > 3$), ou dynamiquement on doit réaliser un plus grand nombre d'opérations. Si la méthode de recuit simulé permet de franchir des barrières élevées, elle ne permet pas de franchir des barrières de grande largeur (fig. 8). Si ΔE est l'énergie nécessaire pour atteindre un plateau (en une seule opération) et si I est le nombre d'itérations (opérations) nécessaires au franchissement du pla-

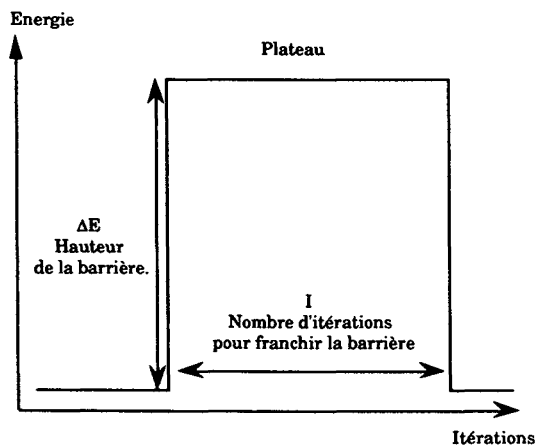


Figure 8. – Barrière de potentiel.

teau, la probabilité réelle de franchissement est

$$p = (1/2)^I \exp(-\Delta E/T).$$

Pour une valeur de I élevée ($I \geq 10$), la probabilité de franchissement est donc insignifiante. Donc, même si en théorie une opération λ -OPT peut être réalisée par la succession de I opérations λ' -OPT, la probabilité que la série d'opérations λ' -OPT soit réalisée est assez faible.

Bien évidemment, on peut combiner la technique du recuit simulé avec l'emploi d'opérateurs de transfert de pont. Une telle solution est techniquement envisageable, mais elle supprime l'un des intérêts de la technique de recuit simulé : sa simplicité.

5. RÉSULTATS POUR UNE DISTRIBUTION UNIFORME

Dans le cas d'une distribution uniforme des sommets, on connaît la limite de la loi suivie par la longueur du plus court chemin. La connaissance de cette loi nous a permis d'évaluer les performances de l'algorithme (voir 4.1). Dans le paragraphe 5.2, on étudie plus précisément le comportement de la longueur du plus court chemin pour des distributions uniformes des sommets.

5.1. Loi suivie par le chemin

On considère une distribution de sommets dans un espace S de dimension p et de forme quelconque (la convexité de l'espace n'est pas requise). On note $\mu(s)$ la densité de probabilité de la distribution des sommets. La limite de la loi $L(N)$ suivie par la longueur du plus court chemin hamiltonien [14], est alors

$$\lim_{N \rightarrow \infty} L(N) N^{-(p-1)/p} = \beta_p p^{1/2} \int_S \mu(s)^{(p-1)/p} ds$$

où β_p est une constante pour laquelle il n'existe pas de définition analytique. Dans le cas particulier d'un espace de dimension 2 et d'une distribution uniforme, la loi s'écrit

$$\lim_{N \rightarrow \infty} L(N) N^{-1/2} = \beta_2 2^{1/2} S^{1/2}.$$

Bien que l'expression de β_2 ne soit pas connue, sa borne supérieure est définie théoriquement ($\beta_2 \leq 0,6508$). Pour des distributions aléatoires et uniformes

des sommets sa valeur a été estimée par des simulations ($\beta_2 \approx 0,53$). D'après les résultats du paragraphe 4. 1, cette dernière valeur est certainement surestimée.

La loi pour une distribution uniforme est une limite supérieure pour les lois relatives à des distributions non uniformes; autrement dit pour un nombre de sommets fixé le chemin sera de longueur maximale si la distribution des sommets est uniforme.

5.2. Résultats en fonction de la répartition des sommets

On obtient des distributions de sommets de natures différentes en faisant varier la période du générateur de nombre aléatoire ⁽²⁾. Pour des valeurs

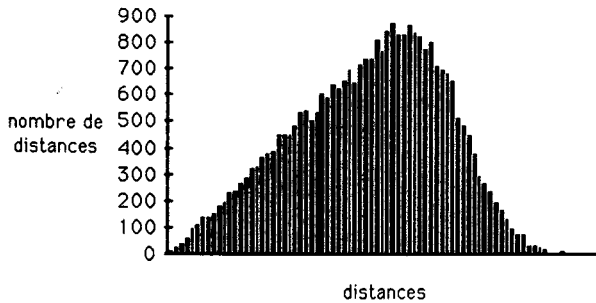


Figure 9 a. — Distribution pour un générateur de période $m=999_983$.

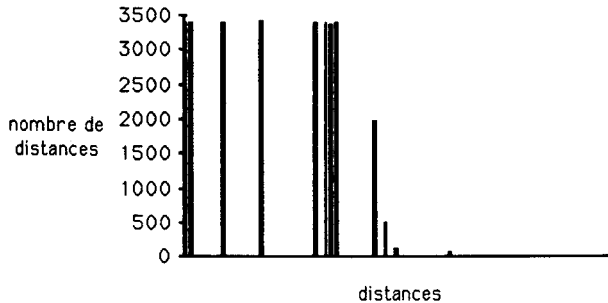


Figure 9 b. — Distribution pour un générateur de période $m=32_749$.

Les distances de valeurs élevées sont vraisemblablement la conséquence d'effets de bord.

Figure 9. — Distributions des distances aux 16 plus proches voisins. Les distributions ont été obtenues sur un échantillon d'environ 2 000 sommets pour une génération de 30_000 sommets dans un carré de côté unité. Les distances sont réparties en 100 classes (80 classes représentées) de largeur $(l_{\max}-l_{\min})/100$.

⁽²⁾ On a utilisé la méthode de Lehmer : $a_{i+1} = ((a_i \times b) + j) \text{ MODULO } m$, avec (entre autres) les paramètres suivants ($b=409, j=2067, m=999_983$) et ($b=417, j=2071, m=32_749$). La périodicité du générateur est égale à m .

de N très inférieures à la périodicité du générateur les sommets sont distribués aléatoirement. On observe des distributions, des distances entre sommets, continues et ayant une forme vaguement triangulaire (fig. 9 a). Pour des valeurs de N proches de la période du générateur les corrélations entre sommets deviennent importantes et des structures régulières apparaissent. Les distributions des distances deviennent discrètes et uniformes (fig. 9 b).

La recherche du plus court chemin avec des distributions discrètes donne des longueurs de chemin supérieures à la prévision relative à une valeur de 0,53 pour β_2 (fig. 10). Ni la modification du chemin initial, ni celle des paramètres de la relaxation, ni l'emploi d'opérateurs d'ordre plus élevé, ni l'emploi d'autres opérateurs (extensions des opérateurs de déplacement), ne modifient sensiblement la précision des résultats.

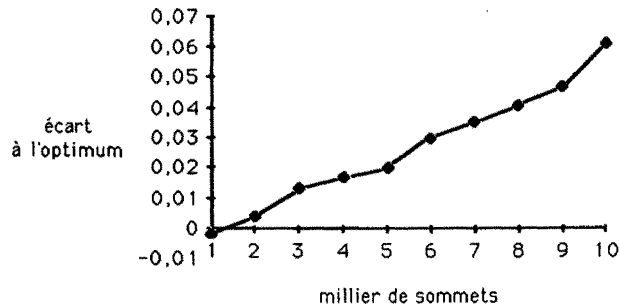


Figure 10. — Résultats avec un générateur de période $m = 32_749$. Représentation de l'erreur relative $\varepsilon = (l_m - l_t) / l_t$ entre la mesure et l'optimum théorique.

On vérifie aussi que le tour obtenu a bien une allure régulière. Les différentes parties du tour sont de longueurs statistiquement équivalentes, et de même forme géométrique. *A priori* le tour produit est normal. On constate néanmoins que les distributions des longueurs des arcs du tour final diffèrent en fonction des distributions initiales des distances. Une répartition aléatoire

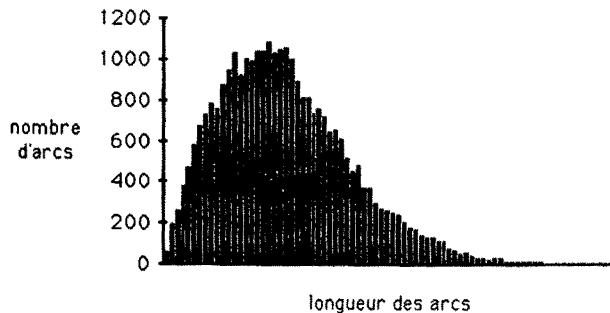


Figure 11 a. — Distribution pour un générateur de période $m = 999_983$.

des sommets donne une distribution de Poisson (*fig. 11 a*). Une répartition régulière des sommets donne trois classes de longueurs d'arcs (*fig. 11 b*).

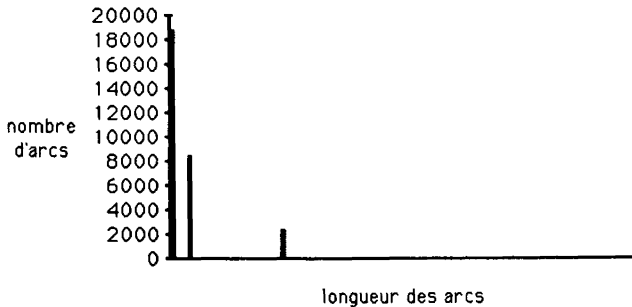


Figure 11 b. — Distribution pour un générateur de période $m=32_749$.

Figure 11. — Distributions des longueurs des arcs du tour final. La distribution a été obtenue pour une génération de 30_000 sommets dans un carré de côté unité. Les distances sont réparties en 100 classes (80 classes représentées).

Cette dégradation des performances pourrait être expliquée par l'inadaptation du principe de l'abaissement continu du seuil au cas des distributions discrètes des longueurs des arcs. L'abaissement régulier du seuil n'entraîne pas la prise en compte progressive d'un plus grand nombre de sommets (on introduit zéro ou un grand nombre de sommets), et le processus de relaxation cesse d'être régulier. D'après des essais de variation irrégulière du seuil effectués avec d'autres distributions on constate une dégradation de quelques pour cents. Cette explication ne suffit donc pas à expliquer un écart qui atteint près de 30 % pour 30 000 sommets.

L'explication la plus vraisemblable est que dans tous les cas le tour obtenu est bien proche de l'optimum, et donc que la loi suivie par la longueur du tour diffère suivant que la distribution des sommets est aléatoire ou régulière. Une distribution aléatoire n'étant pas localement uniforme il est logique que le tour soit plus long (*voir 5.1*) que dans le cas d'une distribution régulière, qui est entièrement uniforme. Dans le cadre de cette hypothèse on constate que pour une distribution régulière de 30 000 sommets la longueur du tour final ($L=169,0$) est assez proche de la limite théorique ($L=159,4$) obtenue avec la valeur de $\beta_2=0,6508$, et très éloignée de la longueur obtenue ($L=129,8$) avec la valeur de $\beta_2=0,53$. Cette explication peut être aisément corroborée en réalisant les mêmes mesures avec d'autres algorithmes.

7. CONCLUSIONS

Les performances de l'algorithme de relaxation se comparent avantageusement à celles des algorithmes classiques. Par ailleurs il partage avec les algorithmes connexionistes la possibilité d'une implémentation massivement parallèle (1 processeur ou 1 processus par sommet). Les opérations non locales imposant une synchronisation complexe et coûteuse, on doit néanmoins être très prudent quant au gain apporté par une telle implémentation.

On doit surtout remarquer l'intérêt que présente une réflexion sur la conception des opérateurs, alors que jusqu'ici les travaux ont essentiellement porté sur les méthodes d'ordonnement des opérations.

La technique de relaxation présentée est susceptible d'améliorations. La définition de la conduite de la relaxation à partir de critères statistiques, devrait permettre d'éliminer le caractère empirique du paramétrage. En particulier il paraît indispensable de tenir compte des propriétés de la distribution des distances entre sommets.

Il ne semble pas qu'il soit intéressant d'adapter l'algorithme pour le cas asymétrique, pour lequel il semble exister un excellent algorithme [10]. En revanche il peut être intéressant d'étudier le cas où d n'est pas une distance mais un indice de dissimilarité.

REMERCIEMENTS

Jean Abadie, Simon Cohen et Laurent Hivert ont lu et corrigé les différentes versions de cet article, dont la rédaction fut laborieuse. Jean Vannimenus a bien voulu m'écouter, et m'a ainsi évité une erreur.

BIBLIOGRAPHIE

1. S. KIRPATRICK, C. D. GELATT et M. P. VECCHI, Optimization by Simulated Annealing, *Science*, 1983, 220, p. 671-680.
2. E. BONOMI, J. L. LUTTON, The N -city Travelling Salesman Problem: Statistical Mechanics and Metropolis algorithm, *S.I.A.M. Rev.*, 1984, 26, 4, p. 551-568.
3. J. J. HOPFIELD et D. W. TANK, Neural Computation of Decisions in Optimization Problems, *Biol. Cybernet.* 1985, 52, p. 141-152.
4. R. DURBIN et D. WILLSHAW, An Analogue Approach to the Travelling Salesman Problem Using an Elastic Net Method, *Nature*, 1987, 326, p. 689-691.
5. B. ANGÉNIOL, G. DE LA CROIX VAUBOIS et J. Y. LE TEXIER, Self Organizing Feature Maps and the Traveling Salesman Problem, *Neural Networks*, 1988, 1, p. 289-293.

6. M. MÉZARD, G. PARISI et M. A. VIRASORO, Spin Glass Theory and Beyond, *World Sci. Lecture Notes Phys.*, 1987, 9.
7. D. S. JOHNSON, More Approaches to the Travelling Salesman Guide, *Nature*, 1987, 330, p. 525.
8. S. LIN et B. W. KERNINGHAN, An Effective Heuristic Algorithm for the Traveling-Salesman Problem, *Oper. Res.*, 1973, 21, p. 498-516.
9. D. S. JOHNSON, Local Optimization and the Traveling Salesman Problem, *Proceedings of the 17th Colloquium on Automata Languages and Programming*, Springer-Verlag, New York, 1990, p. 446-461.
10. D. L. MILLER et J. F. PEKNY, Exact Solution of Large Asymmetric Traveling Salesman Problems, *Science*, 1991, 251, p. 754-761.
11. (a) M. PADBERG et G. RINALDI, Optimization of a 532 City Symmetric Travelling Salesman Problem, 1986, *Actes des « Journées du 20^e anniversaire du groupe combinatoire de l'A.F.C.E.T. »*, I.N.R.I.A.-Rocquencourt, 1986, p. 387-403; (b) M. PADBERG et G. RINALDI, Optimization of a 532 City Symmetric Travelling Salesman Problem by Branch and Cut, *Oper. Res. Lett.*, 1987, 6, p. 1-7.
12. D. G. BOUNDS, New Optimization Methods from Physics and Biology, *Nature*, 1987, 329, p. 215-219.
13. F. FOGELMAN SOULIE et P. GALLINARI, Méthodes connexionistes d'apprentissage, École Internationale d'Informatique, A.F.C.E.T., Brest, 1989.
14. J. BEARDWOOD, J. H. HALTON et J. M. HAMMERSLEY, The Shortest Path Through Many Points, *Proceedings of the Cambridge Philosophical Society*, 1959, 55, p. 299-327.