# APPLICATION D'UNE METHODE DE RECUIT SIMULE A L'IMPLANTATION DE TRAITEMENTS SONAR

Marc REVOL (*)
François GRIZARD (**)

**(*) Thomson Sintra ASM**
1, avenue Aristide Briand, 94117 ARCUEIL Cedex
**(**) Institut National des Télécommunications**
Les Epinettes 9 rue Charles Fourier, 91011 EVRY Cedex

## RESUME

Cet article présente différentes méthodes orientées vers l'aide à la conception de logiciels de traitements du signal, permettant plus particulièrement de définir des implantations statiques de chaînes sonar sur des automates de calcul . L'objectif est d'inscrire ces méthodes dans la réalisation d'un outil d'aide à la conception afin de proposer rapidement plusieurs solutions adaptées aux contraintes et de rechercher des optimisations pour les solutions jugées les plus pertinentes .

## SUMMARY

The purpose of this paper is to present some methods which enable to assist in conceiving signal processing softwares, and more specially in defining static processing organisation to be installed in processing automatons . The major issue is to realize a tool for aided design, so that to propose quickly several suitable solutions, and to optimize the most interesting ones .

### 1 Introduction

*L'ensemble des fonctions retenues pour définir un système de traitements sonar peut donner lieu à de multiples organisations de logiciels, suivant les différents ordonnancements de tâches retenus et selon les différentes architectures envisageables du matériel. Un outil d'aide à la conception a été développé dans le double objectif de fournir rapidement des solutions acceptables à court terme et de rechercher des optimisations suivant plusieurs stratégies pour les organisations définitives des logiciels.*

*On décrit ici les principales fonctions réalisées, en insistant davantage sur la description et les performances de deux algorithmes d'optimisation qui paraissent bien adaptés à ce type de problème .*

*Une phase dite d'évaluation rapide permet d'estimer la faisabilité d'une implantation. Pour cela, partant des spécifications fonctionnelles du sonar, on effectue préalablement le chiffrage du coût des traitements contenus dans les chaînes, en termes de charges CPU et de débits d'informations. Puis on découpe les traitements en fonction des parallélismes possibles; enfin on regroupe les atomes de traitement ainsi obtenus en modules logiciels individuellement implantables sur les automates de calcul Cette étape est réalisée en s'appuyant uniquement sur des heuristiques de placement propres à la description du graphe fonctionnel décomposé en atomes logiciels.*

*Dans un deuxième temps, on recherche une amélioration de l'allocation en utilisant des algorithmes d'optimisation plus performants et des contraintes plus réalistes liées à l'architecture du matériel envisagé. Cette optimisation est faite, soit en reconsidérant totalement le problème d'ordonnancement des atomes, soit en prenant comme base de départ l'organisation résultant de la phase précédente.*

### 2 Evaluation

#### 2.1 Chiffrage des coûts de traitement

Chaque fonction du sonar est décrite sous forme d'une arborescence de traitements élémentaires correspondant à des enchaînements d'algorithmes de traitement de signal et de l'information, ininterruptibles, et de charge moyenne constante dans le temps (figure 2).

Les traitements élémentaires appartiennent à une bibliothèque d'algorithmes connus dont on sait évaluer les coûts en termes de charges CPU, de volume mémoire, de débits d'informations échangées; ces coûts représentent les contraintes moyennes que devront respecter les allocations.

#### 2.2 Organisation des symétries de traitement

Les traitements élémentaires peuvent être parallélisés selon différentes symétries (fréquentielle, géométrique, temporelle) correspondant à des dimensions invariantes de la transformation opérée.

La recherche de la meilleure organisation des parallélismes des traitements successifs permet de réduire le nombre de transpositions de données qui se traduisent par une augmentation des charges de calcul et du volume mémoire. Une transposition consiste à réorganiser en entrée le tableau des données transmises par le traitement précédent de façon à ce qu'elles puissent être traitées dans la symétrie choisie.

La démarche consiste à établir préalablement toutes les symétries possibles de chaque traitement élémentaire et à rechercher pour l'ensemble des combinaisons possibles toutes les transpositions nécessaires. On ne retient alors que les organisations qui minimisent le surcoût en charges CPU et en volume mémoire.

#### 2.3 Découpage des logiciels

Cette étape est rendue nécessaire par l'existence de traitements élémentaires volumineux dont certains peuvent excéder la capacité de calcul d'une machine. On constitue alors des atomes élémentaires de logiciels, de taille homogène, qui soient suffisamment petits vis à vis des capacités de traitement des automates pour autoriser des adaptations souples par ordonnancement des atomes. La normalisation de la taille des atomes permet de simplifier le problème de l'ordonnancement qui porte alors sur des briques de tailles équivalentes quel que soit

le traitement réalisé.

Le découpage est effectué en fonction du type de symétrie de traitement retenu précédemment et du type de machine (SIMD ou MIMD). Cette décomposition en tâches parallèles induit une répartition adaptée pour les flux de données d'entrée et de sortie .

## 3 Association par heuristiques

Cette association a pour but de composer rapidement les atomes précédents sous forme de modules logiciels de taille adaptée aux capacités de calcul des automates de traitement.

Les modules sont constitués en s'appuyant sur des contraintes globales relativement indépendantes de l'architecture du matériel, prenant en compte la charges CPU maximale, la taille mémoire maximale et les débits d'entrée-sortie maximaux tolérés par module logiciel.

Cette étape est réalisée en s'appuyant uniquement sur des heuristiques de placement relatives au graphe fonctionnel décomposé en atomes.

Les atomes issus du découpage sont considérés dans un ordre qui, en moyenne, va de l'entrée vers la sortie du graphe, ce qui permet de prendre en compte prioritairement les traitements généralement les plus contraignants.

Les charges moyennes de communication sont minimisées en rapprochant au plus près les atomes qui s'échangent des flots importants.

Les atomes qui utilisent les mêmes données sont rapprochés, ce qui permet de réduire le nombre de chemins différents nécessaires au routage des données.

Enfin, un critère de distance permet de choisir entre plusieurs atomes concurrents à l'association avec un module en formation et de considérer comme plus prioritaires les atomes qui consomment le plus de données et ceux qui donnent naissance à des sous-graphes réclamant une plus faible charges CPU par information produite :

(1) $D_{ij} = D_{ji} = F_{ij} * \Sigma_k C_k / \Sigma_k F_k * Coef$

i est père de j

k décrit les modules du sous-graphe de j (j compris)

$D_{ij}$ distance entre les atomes i et j

$F_{ij}$ est le flot produit par i et consommé par j

$F_k$ est le débit de sortie du module k

$C_k$ est le besoin CPU du module k

Coef est un coefficient de normalisation

## 4 Optimisation

### 4.1 Objectif

Cette approche optimale permet de rechercher, pour une structure fonctionnelle qui donne a priori satisfaction dans la phase précédente, une optimisation de l'organisation des atomes en tenant compte des contraintes de l'architecture matérielle.

Parmi les méthodes possibles, on n'a pas retenu les approches par exploration combinatoire qui conduisent à des temps de réponse prohibitifs pour les problèmes NP-complets de ce type (réf (5),(6)). On a préféré l'utilisation de méthodes qui semblent prometteuses pour ce type de problème, de type méthodes d'auto-organisation ou méthodes inspirées du fonctionnement de la machine de Boltzman ( Recuit Simulé, réf (1),(4) ). Les solutions attendues, bien que ne l'atteignant pas toujours, se rapprochent sensiblement de l'optimum, tout en réclamant des temps de calcul acceptables.

### 4.2 Modèle d'architecture

L'optimisation est toujours définie au titre d'une architecture précise qui introduit des contraintes spécifiques sur les moyens de communication, d'échange et de traitement.

Dans le cas des essais réalisés, on a retenu une architecture composée de machines de traitements SIMD, à NPE processeurs parallèles, regroupées par "*paquets*" de NH machines communiquant par un lien unidirectionnel en anneau (figure 1). Les paquets de machines communiquent entre eux par les liens unidirectionnels reliant les machines de même rang.

Les données d'entrée issues des antennes et les données de sortie sont fournies par des unités d'échange spécifiques.

### 4.3 Fonction de coût

Toutes les méthodes proposées utilisent une fonction de coût multi-variables qui permet d'évaluer l'efficacité des répartitions proposées.

Dans le cas de la méthode du Recuit Simulé, la fonction de coût est le critère principal qui guide la recherche d'une solution; on recherche l'organisation des atomes qui minimise la fonction de coût.

Dans le cas des méthodes d'auto organisation, le coût est une conséquence issue de la démarche retenue pour effectuer la répartition ; il n'influe pas sur la recherche d'une solution, mais intervient comme un élément d'évaluation a posteriori des solutions proposées.

La fonction de coût reflète les contraintes que l'on veut voir respectées, c.a.d. :

- pas de dépassement des capacités limites, tant sur les débits imposés aux liens entre machines que sur les charges de calcul des machines,
- répartition homogène des charges de calcul et des débits,
- débits les plus faibles possibles sur les liens.

Les tendances exprimées par ces contraintes sont traduites sous la forme d'une fonction croissante par rapport aux dépassements et aux inhomogénéités de répartition .

On a retenu pour expression générale, une combinaison de fonctions $f_i$, polynomiales ou exponentielles, de la forme :

(2)

$$Coût = f_1(NBR) * f_2(DCC) + f_3(NOAr) * f_4(DTAr) + f_5(NOS) * f_6(DRS)$$

où,

NBR; nombre de dépassements de capacité CPU machine,

DCC; distortion de répartition de la charge de calcul,

NOAr; nombre de dépassements de débit sur les liens,

DTAr; débit total sur les liens ,

NOS; nombre de dépassements sur les débits des unités d'échange en sortie,

DRS; distortion de répartition des débits de sortie des unités d'échange en sortie.

### 4.4 Méthode du *Recuit Simulé*

Cette méthode (réf (1),(4) ) est utilisée pour rechercher le minimum d'une fonction à variables multiples en s'inspirant directement de la physique des cristaux. On sait que l'état final d'une matière en fusion dépend de la manière dont est effectué le refroidissement; le cristal correspond au minimum global d'énergie potentielle alors que les états amorphes correspondent à des minima locaux. Le but du recuit est d'obtenir un état cristallin, c'est à dire de minimiser l'énergie potentielle.

Dans le cas du placement des atomes de traitement, on recherche une répartition qui ***minimise la fonction de coût*** ci-dessus ; celle-ci joue le rôle de l'énergie potentielle du recuit physique.

Par similitude, on introduit une grandeur appelée

***température*** dont la valeur initiale est en principe d'autant plus grande que la fonction de coût est complexe ( du point de vue du nombre de ses minima locaux).

La méthode consiste à déformer progressivement l'organisation de départ des atomes de façon à diminuer la fonction de coût. On tolère des remises en cause occasionnelles en acceptant des solutions qui entrainent une augmentation du coût, avec une probabilité, fonction de la température, de type (3) **A * exp(-K * Coût/T)** (par analogie avec la probabilité de présence d'un système dans un état d'énergie *Coût* à la température *T*), qui permet de palier les convergences locales.

Après avoir défini un état initial de la répartition en plaçant aléatoirement les atomes sur les machines, on initialise la température à une valeur élevée et on réitère les actions suivantes:

- déplacement d'un certain nombre d'atomes vers une autre machine,

. le nombre d'atomes déplacés est fonction de la température,

. les atomes déplacés sont tirés aléatoirement avec une loi équirépartie,

. les machines destinatrices sont elles aussi choisies aléatoirement,

- choix des routages des flots parmi les routages les plus simples,

- évaluation du coût C' de la nouvelle organisation ainsi produite.

- si le coût est plus faible que celui de l'implantation initiale, on accepte les déplacements, par contre si le coût est plus élevé, on évalue par tirage aléatoire la possibilité de retenir cet état. Pour cela, on calcule la probabilité de passage P du système de l'état initial de coût C, à l'état final de coût C' par la formule (4) **P=exp-(C-C')/(K*T)**. Cette probabilité est alors comparée avec la valeur obtenue par tirage d'un nombre aléatoire S équiréparti entre 0 et 1 qui simule l'occurence d'un tel état.

- l'état final est retenu si **P<=S** , sinon il est rejeté.

**Mise en oeuvre :**

La difficulté porte essentiellement sur le réglage des paramètres de la fonction de coût, sur le choix de la température initiale et de la loi de décroissance de la température.

Typiquement, à température forte par rapport aux valeurs de coût, on va chercher dans toutes les directions mais sans jamais pouvoir approfondir . En effet, à peine tombé dans un minimum, on aura toutes les chances d'en ressortir en acceptant une des multiples propositions d'organisation à coût plus élevé qui seront fort probablement soumises. A température faible, on ne fera qu'approfondir la recherche du minimum près duquel on est déjà placé. C'est donc dans la zone intermédiaire que tout ce joue. La température y est suffisamment faible pour ne pas toujours remettre tout en cause et suffisamment élevée pour permettre d'éviter des *pièges* locaux. C'est le choix de cette zone qui conditionne le résultat final.

On a ainsi constaté que :

- si l'on veut approfondir une recherche, on a davantage intérêt à ralentir le refroidissement qu'à augmenter la température initiale,

- une phase préalable de normalisation autour d'une valeur choisie (typiquement 0.1) du rapport coût sur température sur quelques essais, permet de réduire les oscillations et le temps de calcul ,

- la fonction de diminution du nombre d'atomes déplacés à chaque itération doit être une fonction croissante de la température pour éviter les oscillations et pour fournir un critère d'arrêt à l'algorithme ,

## 4.5 Méthode par Approches Successives

Cette méthode (réf. (2)) , plus directive que la précédente, fait partie, avec la méthode dite *Elastic Net* (réf. (3)), des méthodes auto adaptatives qu'on a étudiées. Elle permet de tenir compte des affinités entre atomes et des contraintes de l'architecture pour proposer des solutions.

Tout d'abord, on définit une surface susceptible de représenter au mieux l'architecture . Dans notre cas, la surface la mieux adaptée est celle d'un tore; les machines sont disposées de façon équirépartie sur le tore (dans ce cas le placement est continu et un atome est généralement situé à côté d'une machine ).

Ensuite, on calcule la proximité entre les atomes du graphe orienté. La distance retenue est identique à celle de la méthode heuristique ( expression (1)).

Après avoir placer aléatoirement les atomes sur la surface qui supporte les machines, on réitère les actions suivantes :

- Pour chaque machine, on recherche l'atome le plus proche et on lui fait subir une attraction (Da ) vers la machine de la forme :

$$(5)\quad Da = k * d * \exp( -(a*d)^2 / 2*T^2 )$$

où,

d est la distance de l'atome à la machine,

a est un paramètre d'ajustement,

K compris entre 0 et 1, diminue quand la charge de la machine augmente.

- On attire de proche en proche les atomes reliés par un chemin dans le graphe à l'atome sélectionné lors de l'étape précédente. On commence par attirer les atomes les plus proches, puis on explore tous les autres atomes jusqu'à ce qu'on atteigne les limites du sous-graphe. Un atome de niveau N est attiré par ses parents de niveau N-1, d'autant plus qu'il en est *proche*, au sens de la distance entre atomes dans le graphe.

- On abaisse la température et on réitère les opérations précédentes.

Par rapport à l'algorithme initial (rèf. (2)), on a apporté les adaptations suivantes:

- suppression du mécanisme de création-destruction des noeuds et remplacement par un mécanisme permettant à chaque machine d'attirer les atomes à concurrence d'une certaine somme de charges CPU ( typiquement, 1.3 fois la capacité CPU restant disponible dans la machine),

- remplacement du circuit des étapes par un graphe (ouvert) d'atomes de traitement et introduction d'un critère de distance entre atomes,

- prise en compte d'une notion d'ordre entre étapes permettant de prendre en compte des liens de communications unidirectionnels

**Mise en oeuvre:**

Outre le choix de la surface sur laquelle on dispose les machines et de la distance dans le graphe, la difficulté est d'accorder le coefficient d'échelle entre la température initiale et les distances moyennes dans l'*espace des machines*. L'expression du coefficient d'attraction **($d*\exp(-d^2/T^2)$)** fait en effet apparaître une zone dans laquelle l'attraction peut être une fonction croissante de la distance. Il est nécessaire que le rapport d/T soit suffisamment grand pour éviter les oscillations, sans que T soit trop petit pour que l'attraction soit suffisante. On a constaté que pour un bon fonctionnement il est préférable que le maximum d'attraction se fasse pour une distance inférieure à la demi-distance entre deux machines voisines.

## 5 Performances

Les différentes fonctions décrites dans cet article ont été mises en place dans un logiciel d'aide à la conception de traitements sonar.

Des tests comparatifs ont été réalisés sur des graphes de traitements et des architectures simples, sur SUN 3/60 (voir tableaux 1 et 2).

A partir des essais réalisés, on a constaté que la complexité des méthodes Elastic Net et Approche Successive croît moins vite que celle du Recuit Simulé, eu égard à leur plus grand dirigisme.

Les méthodes Elastic Net et Approche Successive sont comparables, mais la seconde est plus facilement réglable.

Sur des graphes simples, la méthode heuristique semble aussi performante que les méthodes d'optimisation qui sont beaucoup plus coûteuses en temps de calcul et plus difficiles à régler. La méthode heuristique a donné de bons résultats dans le cas de graphes complexes; un tel travail reste encore à faire dans le cas des méthodes optimales.

Enfin, toutes les méthodes autorisent de pouvoir introduire des stratégies différentes par simples modifications des paramètres .

## Bibliographie


(1) Optimization by simulated annealing
S.Kirkpatrick, C.D.Gelat,Jr. and M.P.Vecchi
Science 220:671-680, 1983

(2) Self-organizing feature maps and the travelling salesman problem
B.Angéniol, G. de La Croix Vaubois, J.Y. Le texier
Neuro Networks Vol. 1, pp 289-293, 1988

(3) An analogue approach to the travelling salesman problem
R.Durbin , D. Willshaw
Nature Vol. 326, 16 April 1987

(4) Affectation dynamique de ressources d'un satellite en opération par une machine de Boltzmann à sensibilisation
P. Bourret, C. Gaspin, M. Samuelides
Neuro Nimes 88

(5) Machines scheduling problems
Classification, complexity and computations
A.H.G. Rinnooy Kan
Martinus Nijhoff, The Hague 1976

(6) Annals of operations research
Scheduling under ressource constraints - deterministic models
J. Blazewicz, W. Cellary, R. Slowinski, J. Weglarz
Scientific publishing company, Basel Switzerland
Vol.7 1986


TABLEAU 1

| Recuit Simulé | | | |
|---|---|---|---|
| Graphe 0 | | graphe 11 | |
| T=500<br>cr=0.02<br>R0 | optimum<br>890 tours | T=200<br>cr=0.05<br>R2 | mauvais<br>928 tours |
| T=200<br>cr=0.02<br>R0 | optimum<br>487 tours | T=200<br>cr=0.05<br>R3 | mauvais<br>2113 tours |
| T=100<br>cr=0.02<br>R1 | optimum<br>374 tours | T=200<br>cr=0.05<br>R4 | bon<br>414 tours |
| | | T=1000<br>cr=0.01<br>R5 | très bon<br>2994 tours |

TABLEAU 2

| Approches Successives | | | |
|---|---|---|---|
| Graphe 0 | | graphe 1 | |
| T=20<br>cr=0.1<br>R0' | optimum<br>39 tours | T=200<br>cr=0.05<br>R2' | moyen<br>90 tours |
| | | T=500<br>cr=0.01<br>R2' | moyen<br>435 tours |
| | | T=200<br>cr=0.01<br>R4' | moyen<br>343 tours |
| | | T=200<br>cr=0.01<br>R5' | optimum<br>404 tours |

**case 1 : Conditions initiales**

**T: température initiale**
**cr: coefficient de refroidissement**
**R réglages de l'algorithme**

**case 2 : Résultats**

**Qualité de la solution**
**Nombre de tours explorés**

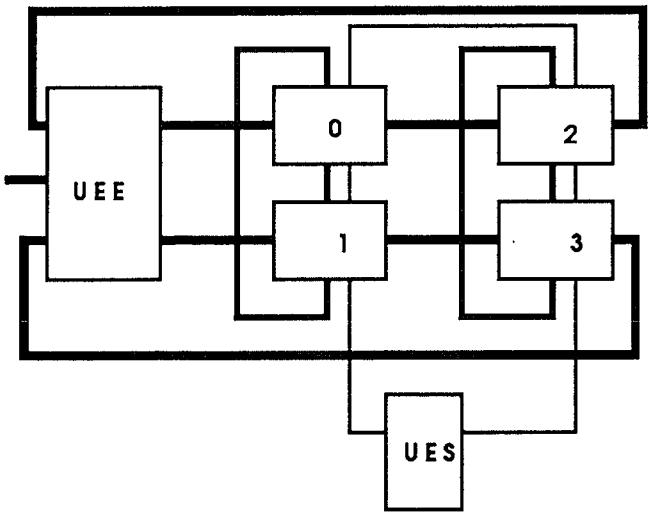
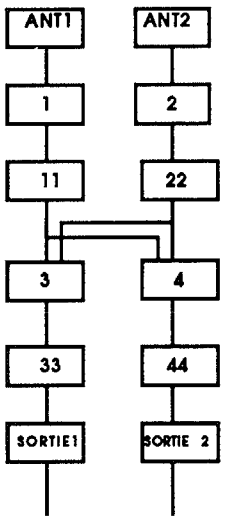



Figure 1 : Exemple d'architecture

GRAPHE 1




GRAPHE 11

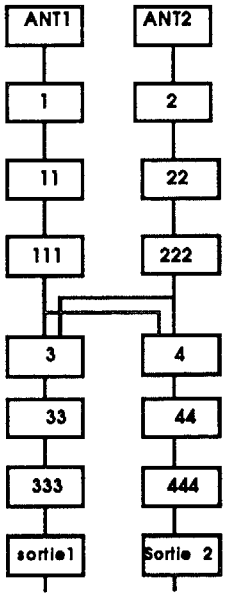



Figure 2 : Exemples de graphes fonctionnels