
TER : Le Problème des fusiliers
Résolution par métaheuristiques
CAHIER DES CHARGES

Auteurs

FULCONIS ANGÉLIQUE
BENOUALI HAMINE
BOUHLEL OUALID
CASANOVA PIERRE

Encadrants

VÉREL SÉBASTIEN
CLERGUE MANUEL

Le 3 avril 2008

Table des matières

1	Présentation générale du problème	1
1.1	Problématique	1
1.2	Projet	1
1.3	Finalités	2
1.4	Contexte	2
1.4.1	État de l'art	2
1.4.2	Études menées sur des sujets voisins	3
1.4.3	Nature de prestations demandées	3
2	Description de l'étude	4
2.1	Approches envisagées	4
2.2	Méthode d'analyse	4
3	Contraintes non fonctionnelles	4
4	Gestion du projet	4
4.1	Priorités	4
4.2	Limites	5
4.3	Hypothèses, dépendances, contraintes	5
4.3.1	Hypothèses	5
4.3.2	dépendances et contraintes	5
4.4	Gestion du risque	5
4.5	Moyen de contrôle	5
4.6	Méthodes et outils employés	6
4.7	Planning	7
5	Annexe	9

1 Présentation générale du problème

1.1 Problématique

Depuis quelque temps, on constate l'essor grandissant de nouveaux types d'ordinateur : les machines dites parallèles qui possèdent plusieurs processeurs et une ou plusieurs mémoires. Il existe aujourd'hui une grande diversité de parallélismes pour lesquels beaucoup de travaux restent à faire. Ce projet concerne un type particulier de parallélisme : les automates cellulaires qu'avait inventé Von Neumann à la fin des années 40. Nous pensons que l'automate cellulaire est un modèle d'avenir, en effet, il présente des atouts non négligeable :

- Premièrement le nombre important de processeurs mis en parallèle.
- Deuxièmement, son fonctionnement synchrone nous permet de visualiser simplement la progression des calculs.

Le problème de la synchronisation d'une ligne de fusiliers est donc un prétexte pour aborder la programmation d'ordinateur massivement parallèle à fonctionnement synchrone. Ce TER permet en réalité de s'intéresser à certains aspects :

- trouver une méthode de synchronisation de machines communiquant de manière locale,
- la découverte de cette méthode se fait de manière automatique : possibilité de déterminer automatiquement des protocoles de communication locaux,
- étude de la faisabilité sur un modèle **simple et universel** de machines parallèles : l'automate cellulaire.

1.2 Projet

Il s'agit de résoudre un problème classique de synchronisation dans le domaine des automates cellulaires (AC), proposé par J.Mihill en 1957 : Comment synchroniser une ligne de fusiliers de façon à ce qu'ils se mettent à tirer ensemble, alors que l'ordre donné par le général depuis l'un des deux bords de l'escadron met un certain temps à se propager ?

Concrètement chaque fusilier est représenté par une cellule d'un automate cellulaire. Celles ci peuvent être dans différents états : repos (0), général (1), ou feu (4) ou dans l'un des états intermédiaires.

Quel que soit le nombre d'états, le problème consiste à trouver des règles de transition permettant la synchronisation des cellules dans le même état « feu ». Ces règles d'inférence, définissant l'état futur de la cellule, sont de tailles 3, i.e. le voisinage de chaque cellule contient elle-même ainsi que celles de droite et de gauche (qui peuvent correspondre aux extrémités). Le but du problème est de trouver les règles de transition qui vont permettre d'obtenir une configuration dans laquelle toutes les cellules se trouvent ensemble et pour la première fois dans l'état feu. Une solution est optimale quand le temps de synchronisation est minimal, c'est à dire $2n - 2$ pour un automate à n cellules.

1.3 Finalités

Ce problème qui jusqu'à présent a été étudié à la main, nous nous proposons de l'aborder de toute autre manière en développant des méthodes qui permettront de rechercher des solutions de façon automatique. Ce qui nous amènera aux finalités suivantes :

- l'analyse comparative des différentes méthodes existantes pour traiter ce genre de problèmes,
- le développement de nouveaux algorithmes de résolution du problème des fusiliers de façon générale et plus particulièrement pour cinq états (en temps minimal),
- la mise en place d'une page web permettant de visualiser les résultats obtenus,
- la création d'une bibliothèque d'algorithmes qui s'appliquent à la résolution des problèmes des fusiliers à p état,
- finalité pédagogique : apprendre les méthodes d'optimisation stochastique (métaheuristiques) et aborder un problème fondamental des systèmes dynamiques discrets,
- la résolution du problème pour six états si l'état d'avancement dans le planning nous le permet.

1.4 Contexte

Cette partie présente les études précédentes ainsi que les prestations demandées.

1.4.1 État de l'art

En 1957 J. Myhill inventa le problème dit « Firing squad » (Ligne de fusiliers) mais ce n'est qu'en 1964 que E.F. Moore publia le problème des fusiliers.

Un an plus tard Minsky et Mc Carthy résolvent ce problème avec la méthode « Diviser pour régner » avec une complexité en temps de $3n$.

La première solution non publiée en temps optimal fut découverte par E. Goto. L'automate comportait plusieurs milliers d'états.

Puis en 1967 Waksman et Balzer découvrirent parallèlement une solution optimale : l'une pour 16 états et l'autre pour 8 états.

Plus tard Balzer réussit à montrer que pour 4 états le problème n'avait pas de solution.

Puis en 1986, Mazoyer trouva une solution en temps minimal pour 6 états.

En 1993 Yunès a également effectué plusieurs travaux : en temps quelconque, il a démontré que pour 3 états il n'y a pas de solutions et qu'il y en a pour 7 et 8 états.

Récapitulatif :

Nombre d'états	Temps minimal
3 états	Pas de solution Balzer
4 états	Pas de solution Balzer
5 états	Ouvert
6 états	Une seule solution Mazoyer
7 états	Solution Mazoyer
8 états	Solution Balzer-Mazoyer

Nombre d'états	Temps quelconque
3 états	Pas de solution Yunes
4 états	Ouvert
5 états	Ouvert
6 états	Ouvert
7 états	Solution Yunes
8 états	Solution Yunes

Jusque là, les études étaient réalisées de façon manuelle.

Enfin en 2007, Mirela Frandes a également travaillé sur ce problème en utilisant une nouvelle approche, et obtenu des résultats la synchronisation des n fusiliers avec 5 états pour n entre 2 et 15 en temps non optimal. Pour 5 états, le problème reste ouvert : aucune solution n'a été trouvée en temps optimal.

1.4.2 Études menées sur des sujets voisins

Certaines études ont été réalisées en plaçant le général donnant l'ordre à une position non extrême. Il existe de très nombreuses variantes de ce problème, qui modifient par exemple la topologie des cellules ou la manière de calculer sur l'automate.

1.4.3 Nature de prestations demandées

Au cours du TER, nous nous emploierons à réaliser les tâches suivantes :

- analyse comparative des différentes stratégies pour traiter ce genre de problème. Celle-ci sont détaillées dans la section suivante. Pour cela, il est nécessaire d'implémenter les algorithmes et comparer les résultats obtenus,
- mise en place d'une page web permettant de visualiser les résultats obtenus,
- création d'une archive d'algorithmes qui s'appliquent aux problèmes des fusiliers à n états.

2 Description de l'étude

2.1 Approches envisagées

Dans le cadre de ce TER, quatre approches de résolution ont été envisagées :

- par métaheuristique : Hill Climbing, First Ascent, Recuit Simulé, Recherche Tabou, et Algorithme évolutionnaire, etc,
- par approche descendante et constructive : construction des règles locales par génération de signaux (interprétation des calculs au niveau des signaux, cf Mazoyer),
- par approche ascendante et analytique : à partir de la configuration finale, conception des règles locales de proche en proche et utilisation du backtracking en cas de conflit,
- approche mixte : entre deux des trois précédentes.

2.2 Méthode d'analyse

Le but est de développer et d'analyser les différentes approches. On devra comparer ces approches de façon statistique sur la base d'un échantillon significatif d'expérimentations, en fonction des résultats qu'elles auront obtenu, mais aussi en fonction de leur facilité de conception.

3 Contraintes non fonctionnelles

Il s'agira d'équilibrer le temps alloué au développement et à l'expérimentation de chaque approche de manière à ce que les échantillons statistiques dont nous disposerons permettent de les comparer efficacement statistiquement.

4 Gestion du projet

4.1 Priorités

Les différentes priorités que nous nous sommes fixé sont les suivantes :

- 1 analyse comparative des résultats.
- 2.a mise en oeuvre de l'approche 1.
- 2.b mise en oeuvre de l'approche 2.
- 2.c mise en oeuvre de l'approche 3.
- 2.d mise en oeuvre de l'approche 4.

- 3.a expérimentation de l'approche 1 sur 5 états.
- 3.b expérimentation de l'approche 2 sur 5 états.
- 3.c expérimentation de l'approche 3 sur 5 états.
- 3.d expérimentation de l'approche 4 sur 5 états.
- 4 Site Web.
- 5 Expérimentation sur 6 états.
- 6 Système d'archivage contenant tous les codes développés.

4.2 Limites

Il s'agira de ne pas perdre trop de temps à ne développer qu'une seule approche ou à n'aborder que le problème sur 5 états.

4.3 Hypothèses, dépendances, contraintes

4.3.1 Hypothèses

Nous avons déjà quelques intuitions concernant la faisabilité ou la plus ou moins grande efficacité des différentes approches. Si nous pensons que la première approche obtiendra des résultats, nous nous interrogeons sur leur qualité. La deuxième nous paraît de prime abord difficile à développer. Nous avons bon espoir que la troisième, quant à elle, nous fournisse des résultats de bonne qualité.

4.3.2 dépendances et contraintes

La méthode 4 requiert que deux au moins des trois méthodes précédentes soient implémentées.

4.4 Gestion du risque

Même si la méthode 2 peut être difficile à mettre en oeuvre, certains des concepts qui y sont associés pourront alimenter les autres approches. Ces dernières seront implémentées quoi qu'il arrive. Si l'approche 2 était abandonnée, la méthode 4 serait l'hybridation des méthodes 1 et 3.

4.5 Moyen de contrôle

Le temps nécessaire au développement de chacune des approches. En fonction de ce temps, une approche pourra être abandonnée.

4.6 Méthodes et outils employés

Nous utiliserons C++, Evolving Object, paradisEO, le logiciel R, et des outils dédiés pour les statistiques.

4.7 Planning

hamine benouali							avr. 2008 (Paris)	
	lundi	mardi	mercredi	jeudi	vendredi	samedi	dimanche	
	31	1	2	3	4	5	6	
14:30 - Finalisation du cahier des charges				11:30 - RDV Mr Verel :				
14:30 - Implementation des trois approches, et comparaison des premiers resultats @ Valrose								
10:30 - RDV Mr Verel :								
15:30 - MISE A JOUR SITE INTERNET @ Valrose								
Implementation des trois approches, et comparaison des premiers resul...								
14:30 - Experimentation des trois approches implementees @ Valrose								
MISE A JOUR SITE INTERNET @ Valrose								
Experimentation des trois approches implementees @ Valrose								
MISE A JOUR SITE INTERNET @ Valrose								
15:30 - Approfondissement des approches 1 et 3, et implementation de l'approche 4 @ Valrose								

lundi	mardi	mercredi	jeudi	vendredi	samedi	dimanche
28	29	30	1	2	3	4
MISE A JOUR SITE INTERNET @ Valrose						
15:30 - Approfondissement des approches 1 et 3, et implémentation de l'approche 4 @ Valrose						
5	6	7	8	9	10	11
MISE A JOUR SITE INTERNET @ Valrose						
15:30 - Finalisation et analyse des résultats @ Valrose						
12	13	14	15	16	17	18
MISE A JOUR SITE INTERNET @ Valrose						
15:30 - Rapport @ Valrose						
19	20	21	22	23	24	25
MISE A JOUR SITE INTERNET @ Valrose						
15:30 - Fin du rapport et préparation de la soutenance final @ Valrose						

5 Annexe

Définition

Les automates cellulaires sont à la fois un modèle de système dynamique discret et un modèle de calcul. Un automate cellulaire consiste en une grille régulière de cellules pouvant chacune prendre à un instant donné un état parmi un ensemble fini. Le temps est également discret et l'état d'une cellule au temps $t+1$ est fonction de l'état au temps t d'un nombre fini de cellules appelé son voisinage. On appelle *Automate Cellulaire Linéaire Fini* (ACLiF) tout doublet $\mathcal{A} = \langle \mathcal{Q}, \delta \rangle$, où :

- \mathcal{Q} est un ensemble fini d'états, dans lequel les plus importantes sont :
 q_l : dit état latent, quiescent ou de repos.
 q_i : dit état initial, départ ou général (car il donne l'ordre).
 q_f : dit état de feu ou de synchronisation.

Remarque : On utilisera par la suite un état appelé état de bord : $q_b \notin \mathcal{Q}$ (c'est le seul état du système que l'on considérera comme n'étant pas interne).

- δ est appelé fonction de transition locale et vérifie :

$$\delta : \mathcal{Q} \cup q_b \times \mathcal{Q} \times \mathcal{Q} \cup q_b \rightarrow \mathcal{Q}$$

- On impose que l'état q_l vérifie la condition :

$$\delta(q_b, q_l, q_l) = \delta(q_l, q_l, q_l) = \delta(q_l, q_l, q_b) = q_l$$

Définition

Une *ligne de n automates cellulaires* (ACFiL), notée \mathcal{A}_n , est constituée par n machines \mathcal{A} numérotées de 1 à n . Une ligne est appelée automate cellulaire fini de longueur n .

Définition Une *configuration* d'un ACLiF de longueur n est une fonction :

$$\mathcal{C}^n : [1, n] \rightarrow \mathcal{Q}$$

Définition On appelle *fonction de transition globale*, la fonction induit par la fonction de transition locale δ sur $\mathcal{Q}^* = \bigcup_{i=1}^{\infty} \mathcal{Q}^i$:

$$\Delta : \mathcal{Q}^* \rightarrow \mathcal{Q}^*$$

Définition Une *heuristique* est un algorithme de résolution basé sur l'expérience mais ne fournissant pas nécessairement une solution optimale. L'intérêt d'une heuristique est de fournir une solution réalisable pour des problèmes difficiles (NP) de complexité exponentielle. On parle de *métaheuristique* pour un ensemble d'heuristiques pouvant s'appliquer à

différents problèmes. Métaheuristiques de recherche Locale : Recuit simulé, recherche tabou, Hill Climbing Ces métaheuristiques sont des algorithmes à solution unique qui à partir d'une solution initiale cherchent une meilleure solution voisine jusqu'à un critère d'arrêt Enfin, *les algorithmes évolutionnaires* sont une famille d'algorithmes s'inspirant de la théorie de l'évolution pour résoudre des problèmes divers. Ils font ainsi évoluer un ensemble de solutions à un problème donné, dans l'optique de trouver les meilleurs résultats. Ce sont des algorithmes stochastiques, car ils utilisent itérativement des processus aléatoires.