

*Action COPRIN**Contraintes, OPTimisation, Résolution par INtervalles**Sophia Antipolis*

THÈME 2B



*R*apport
d'Activité

2001

Table des matières

1	Composition de l'équipe	3
2	Présentation et objectifs généraux	3
3	Fondements scientifiques	4
4	Domaines d'applications	6
4.1	Panorama	6
5	Logiciels	6
5.1	Bibliothèque ALIAS	6
5.2	Décomposition et ALIAS	7
6	Résultats nouveaux	7
6.1	Robotique et Théorie des mécanismes	7
6.1.1	Conception optimale et singularités des robots parallèles	8
6.1.2	Micro-robot	9
6.1.3	Suspension automobile	9
6.1.4	Edition et analyse de modèle	10
6.2	Généralisations correctes pour l'induction mathématique	10
6.3	Algorithmes génétiques pour le coloriage de graphes	11
6.4	Portabilité d'un système à base de connaissances en Java	11
6.5	Résolution de système de contraintes géométriques par rigidité, consistance et analyse par intervalles	12
6.6	Décomposition des CSP en domaines finis	13
6.7	Contraintes globales pour l'optimisation	13
6.8	Génération de jeux de test logiciel	15
6.9	Extension du modèle des tableurs pour les systèmes interactifs	16
7	Contrats industriels (nationaux, européens et internationaux)	17
7.1	Institut Laue Langevin	17
7.2	Constructions Mécaniques des Vosges	18
7.3	Alcatel Space Industries	18
8	Actions régionales, nationales et internationales	18
8.1	Actions nationales	18
8.1.1	Projet RNTL "INKA"	18
8.2	Actions européennes	18
8.3	Actions internationales	19
8.3.1	Relations bilatérales internationales	19

9	Diffusion de résultats	19
9.1	Animation de la Communauté scientifique	19
9.2	Participation à des colloques	19
9.3	Enseignement	20
9.3.1	Cours dans le domaine de recherche et cours de DEA	20
9.3.2	Jurys de thèse et d'HDR	20
9.3.3	Responsabilités d'enseignements	20
9.3.4	Autres cours	20
9.4	Thèses	21
10	Bibliographie	21

1 Composition de l'équipe

Responsable scientifique

Jean-Pierre Merlet [DR2 Inria]

Responsable permanent

Michel Rueher [professeur Unsa]

Assistante de projet

Corinne Zuzia [AJT Inria, à mi-temps dans le projet]

Personnel Inria

Yves Papegay [CR2 Inria]

Personnel Cermics

Bertrand Neveu [ingénieur en chef des Ponts et Chaussées]

Personnel Unsa

Gilles Trombettoni [maître de conférence Unsa]

Chercheurs doctorants

Christophe Jermann [boursier BDI CNRS]

Blaise Madeline [boursier MENESR jusqu'au 30 septembre 2001 puis DEMI-ATER]

Luc Rolland [boursier INRIA, partagé avec le projet Spaces]

Pascal Urso [boursier MENESR jusqu'au 30 septembre 2001 puis DEMI-ATER]

Ingénieur de Recherche

Claude Michel [ingénieur de recherche, CDD RNTL]

Chercheurs invités

Arnold Neumaier [professeur à l'institut de mathématiques de l'université de Vienne, Autriche, un séjour d'une semaine en septembre 2001]

Collaborateurs extérieurs

Hélène Collavizza [maître de conférence UNSA]

Yahia Lebbah [maître assistant Université d'Oran Es-Senia, Algérie]

Jean-Pierre Regourd [maître de conférence UNSA]

Stagiaires

Alexis Anglada [stagiaire DEA Informatique UNSA, du 5 mars au 30 septembre 2001]

Damien Hasse [stagiaire ESSI, jusqu'au 15 avril 2001]

Cédric Hyppolite [stagiaire ESSI, jusqu'au 15 avril 2001]

Alexandre Maria [stagiaire 1^{re} année ENPC du 9 avril au 6 juillet]

Sylvain Miermont [stagiaire ESSI, du 19 mars au 19 juin 2001]

Patrick Petitmengin [stagiaire ESSI, du 19 mars au 19 juin 2001]

Delphine Pouille [stagiaire maîtrise UNSA, du 1^{er} juillet au 31 août 2001]

Thomas Rouch [stagiaire maîtrise ESIL, du 11 juin au 10 septembre 2001]

Raphaël Scateni [stagiaire 1^{re} année ENPC du 9 avril au 6 juillet]

Guillaume Wenger [stagiaire 1^{re} année ENPC, du 9 juillet au 9 septembre 2001]

2 Présentation et objectifs généraux

Mots clés : Contraintes, optimisation, analyse par intervalles, consistances, analyse numérique, lien formel-numérique, CAO, géométrie, génération de jeux de test, théorie des

mécanismes, robotique.

Le projet COPRIN, commun UNSA/CNRS/ENPC/INRIA, a pour motivation scientifique la résolution de systèmes de *contraintes*. Dans le cadre de ce projet, une contrainte se définit à partir d'un ensemble de relations f impliquant n inconnues X et pouvant utiliser l'ensemble des opérateurs et fonctions mathématiques usuels (ainsi la fonction $\sin(x+y) + \log(\cos(e^x) + y^2)$ est pour nous admissible).

Les problèmes qui nous intéressent sont d'une part, la résolution de systèmes de contraintes ($(f(X) = 0, f(X) \leq 0)$), d'autre part la recherche d'optimalité ou d'existence de propriété (il existe deux valeurs X_1, X_2 telles que $f(X_1) > 0$ et $f(X_2) < 0$). Par ailleurs, les domaines où nous recherchons les solutions sont bornés : purement continus (intervalle) ou mixtes (mêlant intervalles et ensembles finis de réels).

Chacun des trois partenaires du projet a déjà proposé des méthodes de résolution (approche dite "par contraintes" pour l'IS/CERMICS et "analyse par intervalles" pour l'INRIA) qui ont en commun d'utiliser l'arithmétique d'intervalles. Partager les mêmes structures de données permet de travailler indifféremment avec des méthodes des deux types et c'est en jouant sur la complémentarité des approches que nous estimons pouvoir produire un algorithme efficace. De plus, l'utilisation de l'arithmétique d'intervalles, qui permet de gérer les erreurs d'arrondis, nous permet :

- de fournir des solutions qui sont exactes (dans le sens où l'on peut calculer les solutions avec une précision arbitraire),
- de traiter des problèmes pour lesquels les coefficients sont incertains.

Le principe général des algorithmes que nous développons est le suivant :

- traitement du domaine courant par des opérateurs d'exclusion qui garantissent l'absence de solution dans le domaine,
- si l'étape précédente échoue, on applique des opérateurs de filtrage qui permettent éventuellement de réduire la taille du domaine,
- on utilise sur le domaine réduit des opérateurs d'existence qui permettent éventuellement de détecter la présence d'un sous-domaine contenant une solution unique. Ces opérateurs sont associés à un schéma numérique permettant de calculer cette solution,
- éventuellement si la taille du domaine résiduel est petite, on peut estimer qu'il est une solution du problème,
- sinon, on crée, à partir du domaine résiduel, deux nouveaux domaines en choisissant une des variables et en coupant son domaine propre en deux parties. Ces deux domaines sont ajoutés à la liste des domaines à traiter,
- le processus est répété successivement sur chacun des domaines de la liste.

L'algorithme s'arrête lorsque l'ensemble des domaines de la liste a été traité.

Notre travail consiste donc à développer des nouveaux opérateurs soit d'application générale, soit pour des systèmes avec des structures spécifiques, en particulier ceux issus de nos domaines d'application privilégiés comme la théorie des mécanismes où le génie logiciel.

3 Fondements scientifiques

Mots clés : contraintes, analyse par intervalles, lien symbolique-numérique, robustesse,

implantation, applications, robotique, théorie des mécanismes, chimie moléculaire, santé.

La motivation scientifique à l'origine de COPRIN est le désir de développer un algorithme combinant méthodes de consistance et d'analyse par intervalles pour la résolution de systèmes de *contraintes*, avec comme outil de base *l'arithmétique par intervalles*. Dans le cadre de ce projet, une contrainte se définit à partir d'un ensemble de relations $f(X)$ pouvant utiliser l'ensemble des opérateurs et fonctions mathématiques usuels (ainsi la fonction $\sin(x + y) + \log(\cos(e^x) + y^2)$ est pour nous admissible).

Les problèmes qui nous intéressent sont, d'une part la satisfaction de systèmes de contraintes ($(f(X) = 0, f(X) \leq 0)$), d'autre part la recherche d'optimalité ou d'existence de propriété (il existe une valeur de X telle que $f(X) = 0$, ou deux valeurs X_1, X_2 telles que $f(X_1) > 0$ et $f(X_2) < 0$).

Nous recherchons les solutions du problème dans un domaine borné (appelé une *boîte*), soit continu, soit mixte, c'est à dire où certaines variables sont dans un domaine continu alors que les autres ne peuvent avoir qu'un nombre discret de valeur.

Nos méthode consiste à développer différents opérateurs qui sont appliqués en séquence sur une boîte :

1. *opérateurs d'exclusion* : ils permettent d'affirmer qu'il n'existe pas de solution du problème dans la boîte.
2. *opérateurs rétractants* : ils tentent de réduire la largeur des intervalles de la boîte.
3. *opérateurs d'existence et d'unicité* : ils détectent dans la boîte courante une sous-boîte qui contient une et une seule solution du problème et ils sont en général associés à une méthode numérique qui permet de la calculer de manière sûre.

Après application de ces opérateurs si la boîte n'a pas été rejetée en phase 1 et qu'elle n'est pas modifiée par les phases suivantes, nous procédons à une bisection d'une des variables pour créer deux nouvelles boîtes qui seront ultérieurement soumises au même traitement. Le choix de la variable de bisection permettant d'obtenir le plus rapidement une satisfaction des tests 1 ou 3 fait évidemment partie de nos recherches. Les recherches sur ces opérateurs se font, soit pour le cas le plus général, soit pour des systèmes spécifiques dont la structure particulière est prise en compte pour améliorer l'efficacité des opérateurs.

Par ailleurs, nous développons des méthodes de calcul formel pour :

- rendre commode l'utilisation des algorithmes développés (pas ou peu de nécessité d'écrire du code C++, pour résoudre un système, les méthodes se chargeant de cette génération et de l'exécution du code),
- procéder à des analyses sémantiques des expressions à traiter pour générer automatiquement des opérateurs rétractants, pour obtenir une évaluation des expressions plus fines (l'arithmétique d'intervalles est très sensible à la mise en forme des équations),
- permettre le calcul de solutions avec une précision arbitraire : après isolation des solutions dans des intervalles par un programme C++ la détermination effective des solutions peut se faire dans une procédure de calcul formel avec une précision arbitraire.

4 Domaines d'applications

4.1 Panorama

Mots clés : chimie moléculaire, robotique, santé, théorie des mécanismes.

Clairement, le champ d'application de nos méthodes est extrêmement large, mais il est tout aussi clair qu'au sein d'un projet de la taille de COPRIN, il ne saurait être question d'aborder l'ensemble des applications. Nous avons donc décidé de nous concentrer sur trois sujets : la théorie des mécanismes (pour lequel existe une expertise dans le projet), la chimie moléculaire et le génie logiciel.

- Théorie des mécanismes : nos activités portent sur la conception optimale des mécanismes, sur la modélisation et l'analyse géométrique, en particulier pour la machine-outil, les suspensions automobiles et la robotique médicale,
- Chimie moléculaire : nous nous intéressons aux particularités géométriques de molécules organiques. Certains problèmes de ce domaine ont des analogues en théorie des mécanismes, ce qui a facilité nos recherches dans cette branche,
- Génie logiciel : nous nous concentrons sur le problème de la génération automatique de jeux de tests, c'est à dire la production de données d'entrée pour lesquelles un point sélectionné dans une procédure sera exécuté.

5 Logiciels

5.1 Bibliothèque ALIAS

Participants : Jean-Pierre Merlet [correspondant], Yves Papegay, Thomas Rouch, Delphine Pouille.

Mots clés : analyse par intervalles, géométrie algébrique effective, lien symbolique-numérique, robustesse.

La bibliothèque ALIAS (*Algorithms Library of Interval Analysis for Systems*) a pour objet d'appliquer l'analyse par intervalles aux problèmes d'analyse et de résolution des systèmes. L'écriture de cette bibliothèque a commencé en 1998.

Notre effort, cette année, s'est porté sur quatre axes principaux :

- amélioration de l'efficacité des algorithmes en intégrant en particulier des opérateurs de consistance,
- développement de solveurs spécifiques : nous avons implanté une version préliminaire d'un solveur d'équations de distance, sujet sur lequel le projet porte un effort théorique important,
- amélioration de l'interface **Maple**,
- création d'une interface pour **Mathematica** ,
- parallélisation des algorithmes : par nature, l'analyse par intervalles se prête bien à des calculs distribués. Nous avons amélioré cette année le système de distribution de charge et avons effectué nos premiers essais sur le cluster de PC de l'UR.

Ces efforts ont été motivés par les résultats expérimentaux que nous avons observés sur deux systèmes test :

- la détermination de trajectoires de suspension automobile,
- le problème de la synthèse de robot 3R : on spécifie un certain nombre de points de passage pour l'organe terminal (au maximum 5, le système devenant sur-contraint au delà) et l'on doit déterminer les géométries du robot qui permettent d'atteindre ces points de passage. C'est un problème réputé très difficile :
 - pour 3 points de passage, le seul résultat connu a été obtenu dans un temps variant de 2 heures à 70 heures sur une machine parallèle avec 64 processeurs à 400 MHz,
 - pour 5 points de passage, qui amène à résoudre un système de 30 équations, aucune méthode n'a pu fournir, ne serait ce qu'une solution.

Nous avons tout d'abord testé nos méthodes pour trois points de passage : sur le cluster et en utilisant 22 processeurs, les solutions sont trouvées dans un temps moyen de 4 heures. Nous nous sommes alors attaqués au problème avec 5 points de passage. Les tests préliminaires ont montré que l'on pouvait obtenir des solutions en un temps raisonnable (moins de 2 jours) mais nous n'avons encore pas déterminé si le programme arriverait à sa fin.

La version courante de la bibliothèque ALIAS est disponible sous <http://www-sop.inria.fr/coprin/logiciels/ALIAS>.

5.2 Décomposition et ALIAS

Participants : Alexandre Maria, Raphaël Scateni, Bertrand Neveu.

Mots clés : décomposition, ALIAS.

Deux stagiaires de l'ENPC ont étudié comment utiliser la bibliothèque logicielle du projet, ALIAS, pour résoudre des systèmes d'équations non linéaires décomposables en sous-systèmes pouvant être résolus en séquence. Cette approche, qui avait auparavant été étudiée [1] avec `IlogSolver` a été implantée en ALIAS et testée sur un problème académique formé d'équations de distance entre points : elle a donné de premiers résultats prometteurs.

6 Résultats nouveaux

6.1 Robotique et Théorie des mécanismes

Participants : Jean-Pierre Merlet, Yves Papegay, Luc Rolland, Thomas Rouch.

Mots clés : robotique, theorie des mecanismes, robots paralleles.

Nous poursuivons nos travaux dans ce domaine, actuellement en pleine explosion en raison des applications potentielles pour la machine-outil, l'automobile et le médical. Dans ce contexte, nous nous sommes intéressés à :

- l'amélioration des méthodes de conception optimale pour les robots parallèles,
- la détection des singularités pour les robots parallèles,

- l'analyse du modèle géométrique des mécanismes de suspension automobile : il s'agit ici de déterminer de manière exacte les trajectoires possibles d'un corps particulier du mécanisme.

Dans le domaine médical, nous poursuivons notre développement d'un micro-robot pour l'endoscopie.

6.1.1 Conception optimale et singularités des robots parallèles

Participants : Jean-Pierre Merlet, Luc Rolland.

Les performances des robots parallèles présentent une sensibilité très forte aux paramètres qui définissent les dimensions du mécanisme. Il est donc important de procéder à une étude soignée du dimensionnement afin d'obtenir une machine optimale. Nous travaillons depuis plusieurs années sur une méthodologie originale, l'approche *espace des paramètres*. L'espace des paramètres est l'espace pour lequel chaque dimension est associée à un des paramètres de conception : un point de cet espace correspond donc à une géométrie unique du robot. L'approche consiste alors à considérer successivement chacun des critères de performance intervenant dans le cahier des charges et de déterminer quelle est la région de l'espace des paramètres (ou une approximation de cette région) qui contient l'ensemble des robots satisfaisant le critère (ou une version moins forte de ce critère). Après avoir calculé ces régions pour différents critères, nous calculons l'intersection des régions obtenues, qui est la seule qui contient les robots satisfaisant l'ensemble des critères. On procède ensuite à une discrétisation de cette région (donc on ne considère qu'un nombre limité de robots) et nous vérifions pour chacun des robots s'il satisfait l'ensemble des critères du cahier des charges pour obtenir ainsi un ou plusieurs robots qui sont optimaux.

Cette approche présente deux difficultés :

1. le calcul des régions dans l'espace des paramètres,
2. la vérification des performances d'un robot donné (déterminer ces performances revient la plupart du temps à résoudre un problème d'optimisation contraint).

Pour le premier point, nous avons proposé jusqu'à maintenant des algorithmes spécifiques pour un type de performances (par exemple l'espace de travail du robot) : ces algorithmes sont très efficaces mais nécessitent pour chaque critère une analyse théorique lourde. Nous désirons donc développer une autre approche qui consiste à séparer le problème en un pré-traitement du critère pour le mettre dans une forme qui permettrait son traitement par un module, qui sera lui indépendant du critère. L'analyse par intervalle est un outil de choix dans ce type d'approche, puisque le pré-traitement consiste alors simplement à développer une procédure qui permet de dire si le critère est satisfait pour l'ensemble des valeurs des paramètres de conception défini dans des intervalles (l'ensemble de ces intervalles définit une boîte pour les paramètres de conception) et que le module de traitement a alors une structure très proche des algorithmes de résolution de système qui sont développés dans le projet. En particulier, on peut procéder à une implantation distribuée du module de traitement puisqu'il traite les boîtes d'une liste et que le traitement d'une boîte est indépendant de celui des autres boîtes.

Nous avons entamé nos travaux sur cette approche général en développant un algorithme qui permet de traiter le critère espace de travail du robot (un certain nombre de positions

spécifiques doivent faire partie de l'espace de travail du robot) avec un nombre étendu de paramètres par rapport à l'algorithme du même type que nous avons proposé auparavant [12]. Les premiers résultats sont très encourageants puisque l'implantation distribuée a permis de résoudre le problème en un temps de calcul inférieur à celui de notre algorithme précédent, alors que le nombre de paramètres est passé de 2 à 5.

Pour l'analyse de performances, nous désirons utiliser la même approche que pour la conception optimale : mise en forme du critère puis utilisation d'un module de traitement unique. Cette approche a été testée sur l'analyse des singularités qui consiste à déterminer si le déterminant de la matrice inverse jacobienne du robot peut s'annuler pour une position du robot dans un espace de travail spécifié [11]. Le pré-traitement consiste à calculer avec un outil symbolique (**Maple**) une forme analytique du déterminant qui est écrit littéralement dans un fichier. Le module de traitement peut alors évaluer des bornes sur ce déterminant pour des boîtes définissant différentes positions du robot en utilisant la formule analytique du fichier qui est soumis au parser de la bibliothèque **ALIAS** [7]. Avec cette approche, on peut alors traiter l'analyse des singularités de n'importe quel type de robot en écrivant simplement le programme **Maple** qui permet de calculer le déterminant de sa matrice inverse jacobienne. Là aussi, l'utilisation de l'implantation distribuée pour le module de traitement s'impose et nous a permis des gains de temps de calcul considérables puisque nous sommes maintenant capables de détecter la présence de singularité pour un robot de type Gough en quelques minutes.

6.1.2 Micro-robot

Participant : Jean-Pierre Merlet.

Nous continuons notre développement d'un micro-robot parallèle à 3 degrés de liberté pour l'endoscopie dans le cadre d'une collaboration avec le laboratoire LMARC de l'Université de Besançon, le Technion d'Haifa et la société DG Créations, avec le soutien financier initial de l'AFIRST. Deux prototypes ont été développés, l'un d'un diamètre de 7mm pour une longueur de 28mm, l'autre d'un diamètre de 8.6mm et de même longueur mais pouvant exercer des forces plus importantes. L'intégration des capteurs et des moteurs dans ces prototypes va se faire d'ici la fin de l'année 2001 avec un développement en parallèle des algorithmes de commande.

Par ailleurs, nous pourrions commencer en 2002 les premiers essais cliniques en collaboration avec le Dr Dumon de l'hôpital Ste Marguerite de Marseille dans le cadre d'une ACI "Technologie de la Santé".

6.1.3 Suspension automobile

Participants : Yves Papegay, Jean-Pierre Merlet, Thomas Rouch.

Nous nous intéressons de près depuis deux ans à l'étude des mécanismes de suspensions automobiles et de leurs modèles géométriques qui fournissent des systèmes d'équations de faible degré (1 ou 2) et dont le nombre de variable est petit (environ 10) et sur lesquels pourtant les algorithmes classiques de résolution numériques itératifs (de type Newton-Raphson avec estimée initiale) fournissent des résultats parfois incorrects, et toujours incomplets.

Une étude approfondie des mécanismes à 5 barres a mis en évidence les limites des méthodes exactes formelles ou algébriques que nous avons utilisées sur des mécanismes plus simples et nous avons poursuivi nos travaux sur l'analyse géométrique de ce type de mécanismes en utilisant des méthodes d'analyse par intervalles.

En utilisant les implémentations de la bibliothèque **ALIAS** d'algorithmes spécifiques au traitement des systèmes d'équations de dimension 1, nous avons ainsi pu résoudre l'analyse géométrique de la suspension de type 4SK-CS, que nous n'avions pas pu traiter par une approche de type multi-résultant.

L'étude précise de la complexité des calculs en fonction de la taille des mécanismes, l'analyse fine des propriétés des objets mathématiques qui apparaissent génériquement dans ces calculs, l'amélioration de la modélisation et l'adaptation des méthodes effectives employées continuent à nous préoccuper pour obtenir une simulation robuste et efficace.

6.1.4 Edition et analyse de modèle

Participant : Yves Papegay.

En réponse à une demande d'une équipe de modélisation aérodynamique d'Airbus, nous avons repris une activité concernant l'édition et l'analyse de modèle que nous avons menés en collaboration avec Aerospatiale jusqu'en 1999.

Dans différents milieux industriels, le processus de modélisation/simulation s'appuie sur des articles scientifiques qui décrivent des comportements physiques et des méthodes de calculs à l'aide de formules ; il consiste à réaliser, à partir de ces articles, des logiciels de simulations numériques qui implémentent ces méthodes et ces calculs.

Le grand nombre de paramètres et d'équations qui entrent en jeu dans ces modèles, la diversité de leur origine, les contraintes de validité qui leurs sont attachées complexifient ce travail de production logicielle et notamment sa phase de test.

En associant à ses paramètres et à ses équations un hypergraphe qui représente leur interdépendance, nous pouvons ensuite analyser la cohérence dimensionnelle d'un modèle, son domaine de validité sur les paramètres. Nous pouvons aussi en extraire des programmes de calculs générés à partir de la simple donnée des entrées et des sorties, et produire automatiquement les codes d'évaluation numériques correspondants.

La réalisation d'une maquette en **Mathematica** est en cours.

6.2 Généralisations correctes pour l'induction mathématique

Participant : Pascal Urso.

Mots clés : induction mathématique, preuve de théorème.

Le sujet de ma thèse, encadré par le Pr. Emmanuel Kounalis, est l'induction en mathématiques. Cette année nous nous sommes attaqués au problème, réputé difficile, de trouver une méthode correcte de généralisation pour l'induction mathématique. Nous avons étudié différentes heuristiques pour finalement nous tourner vers une approche complètement nouvelle. En effet, nous avons cherché à comprendre comment sont formés les termes d'une formule mathématique dont les fonctions sont définies inductivement. Nous avons montré qu'une analyse

syntaxique de la définition des fonctions pouvait nous permettre de dégager dans une formule les sous-termes à généraliser de manière correcte. Cette analyse nous a permis d'élaborer un nouveau système de preuve qui a été implanté dans le logiciel *Nice* écrit en Java et muni d'une interface graphique.

Des articles sur les deux sujets (généralisation et système) seront proposés à des conférences spécialisées dans le domaine, notamment IJCAR. Le travail de recherche théorique de cette thèse étant maintenant terminé, la thèse sera présentée en mars 2002.

6.3 Algorithmes génétiques pour le coloriage de graphes

Participants : Blaise Madeline, Bertrand Neveu.

Mots clés : algorithmes génétiques, coloriage de graphes.

Dans la littérature, on mentionne dans la plupart des cas, des Algorithmes Génétiques très spécialisés, efficaces pour un seul type de problème, voire pour une seule instance de problème et il est très rarement question de généralisation. De plus, ces algorithmes spécialisés étant aussi régis par des paramètres dont le réglage est long, nécessitent de multiples tests. Les méthodes adaptatives semblent plus prometteuses et peuvent être utilisées pour un grand nombre de problèmes sans réglage particulier.

Le problème, lorsque l'on veut rendre certains opérateurs auto-adaptatifs, est que la plupart des méthodes d'adaptativité endogènes rendent la convergence de l'algorithme bien moins efficace. En effet, le fait d'intégrer au code génétique de chaque individu un réglage possible des paramètres, demande à l'algorithme de converger à la fois vers les paramètres optimaux et vers la solution du problème lui-même. On arrive souvent à une situation où l'algorithme oscille entre plusieurs solutions qui ne sont pas optimales. Ces techniques sont donc mieux adaptées à de l'optimisation en environnement dynamique, plutôt qu'à la satisfaction de contraintes, et au coloriage de graphes.

Une voie plus intéressante est celle d'opérateurs adaptatifs dont les critères d'adaptation sont exogènes. On peut, par exemple, adapter les taux de mutation et de croisement, en fonction du degré de convergence de l'algorithme, qui lui est calculable. En effet, dans le cadre des problèmes de satisfaction de contraintes, la fonction d'évaluation d'un individu est associée au nombre de contraintes satisfaites. Donc, contrairement aux problèmes d'optimisation, on connaît la valeur théorique du meilleur individu. Nous avons plus particulièrement étudié les coloriages de graphes sur-contraints, et nous avons montré que les algorithmes génétiques se révélaient très efficaces sur ce type de problème par rapport aux méthodes classiques [8][9]. Nous avons notamment étudié les applications possibles dans le domaine des réseaux tout optique (technologie WDM) à partir d'un exemple fourni par le projet Mascotte.

6.4 Portabilité d'un système à base de connaissances en Java

Participants : Guillaume Wenger, Bertrand Neveu.

Mots clés : système à base de connaissances, langage Java, règles de production.

Nous avons accueilli pendant 2 mois un stagiaire de l'ENPC, Guillaume Wenger, qui a

réalisé une étude de faisabilité pour le portage du logiciel **Erasmus** en Java. **Erasmus** est un système expert d'entretien routier, développé par la société TWS et écrit en 1991 en **Smeci** et **Lelisp**. Ce stage a été co-encadré par Bertrand Neveu et Frédéric Allez (TWS). Il a, en particulier, été étudié comment utiliser le logiciel JRules d'Ilog.

6.5 Résolution de système de contraintes géométriques par rigidité, consistance et analyse par intervalles

Participants : Christophe Jermann, Bertrand Neveu, Michel Rueher, Gilles Trombettoni.

Mots clés : contraintes géométriques, rigidité, décomposition, intervalles.

Un *système de contraintes géométriques* est constitué par un ensemble d'*objets géométriques* (points, droites, ellipses ...) et de *contraintes géométriques* (distances, angles, incidences, ...) plongés dans un espace euclidien muni d'un repère cartésien global. Les positions, orientations et dimensions des objets géométriques sont définies par les valeurs de leurs *paramètres géométriques*, des inconnues réelles. La résolution d'un tel système consiste à fournir des valeurs pour ces inconnues de telle sorte que toutes les contraintes du système soient satisfaites. Selon le domaine d'application, il peut être intéressant de fournir toutes les solutions d'un tel système (comme en conformation de molécules par exemple) ou bien de ne fournir qu'une seule solution correspondant à un critère donné (optimisation, en CAO par exemple).

Nous étudions plus particulièrement les méthodes dites de *rigidification récursive* qui tirent parti de la propriété de *rigidité* de certains systèmes de contraintes géométriques. Cette propriété est essentiellement étudiée par les communautés *CAO*, *théorie des mécanismes et topologie structurelle*. Les systèmes présentant souvent cette propriété se trouvent dans les domaines d'applications liés à ces trois communautés : reconstruction d'image, CAO, conception de mécanisme, architecture, conformation de molécules, ...

Le schéma général de la méthode de rigidification récursive est séparé en 2 phases :

- Planification. Cette phase décompose le système en sous-systèmes rigides et fournit un ordre partiel pour l'assemblage de ces sous-systèmes.
- Assemblage. Cette phase résout d'abord chaque sous-système en respectant l'ordre fourni par la phase précédente, puis assemble les solutions des sous-systèmes en solutions du système entier.

Ce schéma permet d'obtenir toutes les solutions par un processus de retour-arrière lors de la résolution des sous-systèmes, et permet aussi d'obtenir la solution optimale pour un critère donné.

Nous avons principalement étudié les algorithmes de rigidification récursive proposés par la communauté CAO, ceux-ci offrant des domaines d'applications plus importants que ceux des deux autres communautés. Notre étude a porté sur les 2 phases de la méthode :

- Planification. Les algorithmes proposés par la communauté CAO sont intéressants car, basés sur des algorithmes de flots, ils savent traiter tout système de contraintes géométriques en espace euclidien de dimension quelconque. Cependant, ils utilisent une notion heuristique de rigidité basée sur une analyse des degrés de liberté. Nous avons tâché d'identifier les différences entre la notion de rigidité et cette heuristique afin de détermi-

ner les limites de cette méthode de rigidification récursive. Nous espérons ainsi produire un nouvel algorithme basé sur une heuristique plus conforme à la notion de rigidité avec des champs d'applications identifiables où l'heuristique correspondra à la notion de rigidité. Ces champs devraient comprendre un certain nombre d'applications comme la conformation de molécules.

- Assemblage. Etant donné que l'ensemble des solutions peut nous intéresser, nous avons proposé d'utiliser *la propagation d'intervalles* pour ce faire. Cette méthode a l'avantage de permettre d'encadrer toutes les solutions avec la précision voulue. La complémentarité de l'équipe COPRIN va jouer à plein dans ce domaine puisque les travaux sur les solveurs généraux (consistances + analyse par intervalles + analyse numérique + calcul formel) comme ceux sur les solveurs spécifiques (par exemple pour les contraintes de distance) pourront être intégrés dans cette phase.

6.6 Décomposition des CSP en domaines finis

Participants : Gilles Trombettoni, Bertrand Neveu, Alexis Anglada.

Mots clés : décomposition, contraintes, structure, algorithme, solutions partielles.

Une voie de recherche pour la résolution des problèmes de satisfaction de contraintes en domaines finis concerne l'analyse de la structure du graphe de contraintes (graphe de dépendances entre variables et contraintes) ou de la micro-structure (graphes de dépendances entre les valeurs).

Dans les années 90, plusieurs algorithmes ont été conçus qui utilisaient des algorithmes de graphe ou des techniques existant dans d'autres communautés, comme les bases de données, les décompositions de matrices creuses ou la programmation dynamique.

Mais jusqu'à présent, le principal obstacle à l'utilisation de ces méthodes sur des problèmes réels est leur complexité en espace. Elles conduisent en effet généralement à stocker des solutions partielles, qui peuvent être en grand nombre. Récemment, des approches hybrides mêlant une recherche arborescente classique et des résolutions complètes de certains sous-problèmes ont été proposées.

Nous avons dans le cadre du stage de DEA d'Alexis Anglada, plus particulièrement étudié un nouvel algorithme, qui utilise la notion d'arbre de synthèse. Cet algorithme cherche à assembler des solutions partielles de sous-problèmes en stockant un nombre limité. La décomposition en sous-problèmes est elle-même déterminée par la structure du problème. Une première implantation de cet algorithme a été réalisée en Java.

6.7 Contraintes globales pour l'optimisation

Participants : Michel Rueher, Jean-Charles Régis.

Mots clés : contraintes globale, optimisation, ordonnancement.

Pour résoudre des problèmes d'optimisation, dans le domaine discret, l'approche qui a eu le plus de succès dans la communauté "contraintes" repose sur la définition de nouvelles contraintes globales qui englobent la fonction objectif et un sous-ensemble homogène de contraintes

du problème. Dans ce cas, les algorithmes polynomiaux associés aux contraintes globales évitent de refaire plusieurs fois un travail quasi-identique pour chaque problème de décision et permettent de propager les effets des réductions de bornes.

Nous avons aussi proposé une contrainte globale pour des problèmes dont la fonction objectif est définie par $y = \sum x_i$ et où les variables x_i sont sujettes à des contraintes du type : $x_j - x_i \leq c$. Un algorithme polynomial (dérivé de l'algorithme de calcul des plus courts chemins de Dijkstra) a été proposé pour éliminer des valeurs des domaines des variables [6].

Cette contrainte pourra s'appliquer dans des problèmes d'ordonnement.

Un effort général du projet porte sur le traitement des *équations de distance*. On considère des points X dans un espace de dimension n qui sont soumis à des contraintes de distance :

$$\sum_{j=1}^{j=n} (X_i^j - X_k^j)^2 = L_{ik}^2$$

où L_{ik} représente la distance connue entre les points X_i, X_k . Parmi les points X peut exister des *points virtuels* dont les coordonnées s'expriment comme combinaison linéaire des coordonnées d'autres points (par exemple si $l > 3$ points d'un même solide dans l'espace sont impliqués dans les équations alors il y aura $l - 3$ points virtuels).

Le problème à résoudre est de déterminer les coordonnées inconnues des points X . Nous avons entamé le développement d'un solveur spécifique pour ce type de système en utilisant certaines de ces propriétés :

- la structure des équations permet de déterminer un pavé contenant l'ensemble des solutions du système,
- l'évaluation par intervalle des équations de distance est exacte,
- la structure des équations permet de mettre en œuvre une consistance 2B efficace
- on peut spécialiser le théorème de Kantorovitch en utilisant la structure des équations, ce qui conduit à augmenter la taille de la boule dans laquelle converge la méthode de Newton par un facteur m/n où m est le nombre d'inconnues,
- par un changement de variable linéaire, on peut mettre chaque équation sous une forme qui comporte une partie non-linéaire et une partie linéaire dans les inconnues : on peut alors utiliser la méthode de Yamamura pour construire un opérateur rétractant prenant en compte la totalité des équations.

Ce solveur spécifique a été utilisé pour résoudre deux problèmes : le modèle géométrique direct d'un robot parallèle et la détermination de la conformation d'une molécule à 100 atomes. Dans le premier cas nous obtenons des temps de calcul actuellement légèrement supérieur à la méthode de calcul la plus connue à ce jour (FGb et RS par Faugère, Rouillier, <http://calfor.lip6.fr/~jcf/Software/FGb/index.html>). Dans le second cas, nous traitons un système de grande taille (environ 400 équations) dans un temps d'environ 15 minutes. Nous estimons que la structure des équations devrait permettre le développement d'opérateurs rétractant globaux (prenant en compte l'ensemble des équations) et qu'il doit être possible d'affiner la taille du pavé dans lequel on recherche les solutions.

6.8 Génération de jeux de test logiciel

Participants : Claude Michel, Yahia Lebbah.

Mots clés : test structurel, génération automatique de jeux de tests, programmation par contraintes, consistances partielles, traitement des nombres flottants.

Ce travail s'effectue dans le cadre d'une collaboration avec THALES Airborne Systems dans le cadre du Projet RNTL INKA.

Le test reste une étape essentielle dans le cycle de vie du logiciel : le respect des exigences imposées par les organismes normatifs (ISO, Afnor, DoD,...) concernant le test, représente un coût non négligeable dans le développement des logiciels lorsque les exigences de sécurité sont importantes. Une des difficultés majeures pour l'automatisation du processus de test structurel réside dans la production automatique des cas de test, c'est-à-dire la détermination d'un ensemble de valeurs d'entrée pour lesquelles un point choisi du programme sera exécuté.

Nous avons proposé une nouvelle méthode [3, 19] où ce problème est transformé en problème de résolution de contraintes.

La traduction du programme initial en un système de contraintes est effectuée en utilisant la forme SSA et les dépendances de contrôle. Des opérateurs spécifiques ont été introduits pour autoriser le traitement de contraintes propres à cette application. Un point clé de la résolution du système de contraintes réside dans l'utilisation de contraintes gardées. INKA, le système prototype qui a été développé, permet de traiter des programmes utilisant un sous-ensemble significatif des constructions du langage `C` (e.g., avec des tableaux, instructions itératives `while`, certains pointeurs). Les premiers résultats expérimentaux sur des exemples académiques montrent que INKA est concurrentiel avec les méthodes traditionnelles.

Dans le cadre du Projet RNTL¹ INKA nous sommes spécialement intéressés depuis un an au traitement des nombres à virgule flottante. Les difficultés proviennent essentiellement de l'arithmétique des nombres à virgule flottante. La pauvreté des propriétés mathématiques de ces arithmétiques se traduit par une plus grande sensibilité aux problèmes sous-jacents de l'ATDG. En particulier, la correspondance entre le code source testé, qui sert de base à la génération des contraintes, et le code objet exécuté est délicate à établir. Par exemple, chaque bibliothèque mathématique a ses propres spécificités dont INKA doit tenir compte. Plus généralement, le résultat de l'évaluation d'une expression $e(x_1, \dots, x_n)$ pour un n -uplet de nombres à virgule flottante dépend de divers paramètres (e.g., le mode d'arrondi, la bibliothèque mathématique, l'unité de calcul en virgule flottante).

Les contraintes sur les flottants étant définies par des expressions arithmétiques, l'utilisation directe des solveurs sur les intervalles² pourrait être envisagée pour résoudre des contraintes sur les flottants. L'utilisation directe de tels solveurs se heurte toutefois à deux problèmes majeurs :

- *Les solveurs sur les intervalles ne sont pas conservatifs sur les flottants* i.e., ils peuvent supprimer des solutions sur les flottants qui ne sont pas des solutions sur les réels (e.g.,

¹Réseau National de recherche et d'innovation en Technologies Logicielles

²Les contraintes sur les flottants sont certes des contraintes sur les domaines finis mais la taille des domaines exclut des techniques classiques des CSP (il y a plus de 10^{18} nombres à virgule flottante dans l'intervalle $[-1, 1]$ lorsque les flottants sont représentés par des `double`.)

l'équation $16.0 = 16.0 + x$ est considérée par un solveur basé sur un filtrage par 2B-consistance (e.g., PROLOG IV) comme étant équivalente à $x = 16.0 - 16.0$ qui produit comme résultat $x = 0$ alors que, sur les flottants, cette équation admet pour solution n'importe quel flottant dans l'intervalle $[-8.88178419700125232e-16, 1.77635683940025046e-15]$.

Ces problèmes sont évidemment amplifiés par les transformations symboliques effectuées par certains solveurs pour mieux réduire les intervalles.

- Les solutions fournies par les solveurs sur les intervalles sont de petits intervalles qui peuvent contenir une solution sur les réels alors que les solutions recherchées sur les flottants sont des n -uplets de nombres à virgule flottante. Ainsi, PROLOG IV nous indique que des solutions de l'équation $x^2 = 2$ peuvent être contenues dans l'intervalle $]1.4142135, 1.4142136[$.

Nous avons, de ce fait, proposé un nouveau solveur pour les contraintes sur les flottants[17], reposant sur un algorithme de filtrage conservatif, i.e., un algorithme qui ne supprime aucune solution sur les flottants et capable d'énumérer sur les flottants.

Schématiquement, celui-ci repose sur les principes suivants :

- Une définition d'un véritable système de contraintes sur les flottants avec notamment un mode d'évaluation des contraintes qui tient compte du mode d'arrondi et qui correspond à celui des expressions du langage C respectant le standard ANSI C.
- Un algorithme de filtrage conservatif qui permet de réduire le domaine des variables sans perdre aucune solution sur les flottants ; cet algorithme utilise en particulier des techniques de consistance partielle telles que la 2B-consistance ou la Box-consistance comme heuristique de choix.
- Un algorithme d'énumération qui permet de générer des n -uplets de flottants.

6.9 Extension du modèle des tableurs pour les systèmes interactifs

Participants : Gilles Trombettoni, Bertrand Neveu.

Mots clés : contraintes, systèmes interactifs, NP-complétude, tableurs.

Le modèle des contraintes fonctionnelles est utilisé dans des applications graphiques interactives comme les tableurs ou les interfaces graphiques. Dans ce modèle, chaque contrainte est associée à un ensemble de fonctions, appelée r-méthodes, qui permettent de la satisfaire quand elle est violée : une r-méthode calcule une valeur pour certaines variables, dites variables de sortie, en fonction de la valeur courante des autres variables de la contrainte, dites variables d'entrée. A l'introduction d'une nouvelle contrainte, le rétablissement de la cohérence du système s'opère en deux phases : une phase de planification sélectionne une r-méthode pour chaque contrainte et construit ainsi un graphe orienté de r-méthodes ; une phase de résolution exécute les r-méthodes une à une en respectant l'ordre partiel calculé précédemment.

Quand une seule r-méthode est autorisée pour chaque contrainte, ce modèle est celui des contraintes uni-directionnelles, appelé aussi modèle des tableurs. Ce modèle évite la phase de planification. Il est utilisé dans les tableurs. Son succès s'explique par l'explication simple des recalculs que l'on peut fournir à l'utilisateur et à la *prédictibilité de la prochaine solution* : chaque perturbation entraîne le calcul d'un ordre partiel unique entre les r-méthodes, ce qui

conduit à une solution unique du système.

Quand plusieurs r-méthodes possibles sont autorisées pour chaque contrainte, ce modèle est celui des contraintes multi-directionnelles. Ce modèle est plus puissant et permet de manière simple l'hybridation avec des solveurs capables de résoudre plusieurs contraintes simultanément, notamment des solveurs reposant sur l'arithmétique d'intervalles. Il perd malheureusement la bonne propriété de prédictibilité de la prochaine solution.

Nous avons proposé l'extension du modèle des contraintes multi-directionnelles à l'aide de *liens sémantiques* afin de bénéficier du meilleur des deux formalismes. Pour chaque contrainte du système et pour chaque variable de la contrainte, un lien sémantique spécifie le sous-ensemble des r-méthodes qui sont sélectionnables quand la variable est modifiée. Les résultats obtenus sont les suivants :

- Nous avons prouvé que le problème de planification des contraintes devient NP-complet, pour la recherche de plans avec ou sans circuits.
- Nous avons trouvé une sous-classe polynomiale intéressante où les liens n'autorisent qu'une seule r-méthode. Nous avons conçu un algorithme de planification quadratique appelé **AzureLink**. Cet algorithme est linéaire à l'ajout d'une nouvelle contrainte.
- **AzureLink** rend le système de contraintes prédictible quand les perturbations sont ajoutées incrémentalement au système.

Ces travaux théoriques sont décrits dans un article qui sera soumis dans une revue prochainement et qui a été présenté dans un workshop de CP'2001 [18]. Dans le futur, nous chercherons à hybrider **AzureLink** avec un solveur par intervalles, comme **ALIAS**, dans un cadre applicatif bien déterminé, comme un logiciel de dessin ou un générateur d'interfaces graphiques "intelligent".

7 Contrats industriels (nationaux, européens et internationaux)

7.1 Institut Laue Langevin

Participant : Jean-Pierre Merlet.

L'Institut Laue-Langevin utilise un réacteur nucléaire pour produire un flux neutronique servant à différentes expériences dans les domaines de la physique des particules, de la bioscience et de la science des matériaux. La diffusion des neutrons donne des informations intéressantes sur la structure et la dynamique de la matière. Une expérience, en cours dans cet Institut, consiste à utiliser le flux neutronique pour déterminer l'état de contraintes résiduelles dans des pièces mécaniques. Pour cela, il est nécessaire de déplacer la pièce, souvent de dimensions importantes, par rapport au flux, ceci avec une grande précision. L'Institut désire utiliser pour cela une structure parallèle : une version préliminaire nous a été soumise et nous avons procédé à l'analyse de ces performances. Celle-ci s'étant avérée insuffisante, l'Institut nous a chargé de procéder à une étude de conception optimale. De plus, il nous a été demandé de présenter un exposé sur nos techniques de résolution de système, qui pourrait avoir une application pour certains problèmes traités par cet Institut.

7.2 Constructions Mécaniques des Vosges

Participants : David Daney, Luc Rolland, Jean-Pierre Merlet.

Nous continuons notre partenariat avec cette PME, en collaboration avec le projet SPACES. David Daney a effectué un post-doctorat industriel dans cette société l'année dernière avant de rejoindre SPACES pour un post-doctorat, tout en poursuivant la collaboration avec CMW. Luc Rolland, quant à lui, continue son travail sur l'étude des imprécisions de suivi de trajectoire en utilisant les procédures rapides de calcul de modèle géométrique direct développés par le projet SPACES.

7.3 Alcatel Space Industries

Participant : Jean-Pierre Merlet.

Nous avons initié l'année dernière une collaboration avec cet industriel sur la conception d'un robot parallèle pour le test d'antenne (qui a d'ailleurs été une réussite puisque Alcatel Space Industries a décidé d'en commander quatre exemplaires supplémentaires). Cette collaboration se poursuivra pour la conception de structure flexible avec une thèse commençant à la fin de l'année 2001.

8 Actions régionales, nationales et internationales

8.1 Actions nationales

8.1.1 Projet RNTL "INKA"

Participants : Claude Michel, Michel Rueher, Yahia Lebbah.

Nous avons la responsabilité de la coordination des recherches du projet RNTL INKA. Il s'agit d'un projet pré-compétitif dont l'objectif est la réalisation d'un outil pour la génération automatique de jeux de test structurel à partir du source de programmes C ou C++. Nos partenaires sont :

- Thales "Systèmes Aéroportés"
- Axlog Ingénierie
- le LSR (IMAG)
- le LIFC (Besançon)
- Thomson-CSF Detexis

Notre contribution scientifique dans ce projet porte sur la recherche de techniques de résolution des contraintes pour les systèmes générées à partir d'instructions manipulant des nombres flottants. La durée du projet est de deux ans.

8.2 Actions européennes

Nous sommes impliqués dans le RTN "Network on constraints" (M. Rueher).

8.3 Actions internationales

8.3.1 Relations bilatérales internationales

Nous avons établi une coopération avec l'université de Louvain-la-Neuve (Belgique) qui se traduit par le co-encadrement d'une thèse. Nous avons aussi entamé une collaboration avec l'université de Vienne, en Autriche, qui s'est traduite par une visite d'une semaine du Pr. Neumaier dans le projet.

9 Diffusion de résultats

9.1 Animation de la Communauté scientifique

- J-P. Merlet est membre du Conseil Scientifique du laboratoire LMARC de Besançon, membre suppléant de la commission de spécialistes (61ème section) de l'université de Nice.
- J-P. Merlet est éditeur du journal électronique « Electronic Journal of Computational Kinematics ». Ce journal est soutenu par la commission technique « Computational Kinematics » de l'IFTToMM (International Federation on the Theory of Machines and Mechanisms), dont il a été réélu président pour un mandat de quatre ans.
- J-P. Merlet a été membre du comité de programme de la conférence IEEE Int. Conf. on Robotics and Automation et « General Chairman » du Comité Scientifique de la conférence "Computational Kinematics" qui s'est tenue en Corée et a été relecteur pour les journaux : IEEE Transactions on Robotics and Automation, ASME J. of Mechanical Design, J. of Intelligent and Robotic Systems, Mechanism and Machine Theory, Robotics and Autonomous System, Robotica, European J. of Mechanics, Int. J. for Numerical Methods in Engineering.
- B. Neveu a été président du comité de programme des journées nationales sur la résolution pratique des problèmes NP-complets (JNPC 2001), a été membre du comité de programme du workshop ATAI 2001, (Advances and trends in AI for problem solving) et a été relecteur du congrès RFIA 2002 et du congrès CP 2001.
- M. Rueher est président du comité des projets du laboratoire I3S/UMR 6070 du CNRS, responsable du projet "Contraintes" du laboratoire I3S, vice-président de la Commission de Spécialistes, section 27 de l'Unsa et responsable de la filière LOG de l'Essi (Ecoles Supérieure en Sciences Informatiques de l'université de Nice - Sophia Antipolis). Il est relecteur d'articles pour les journaux dans "Reliable Computing", "Constraints", ToCL, "journal Annals of Mathematics and AI".
- G. Trombettoni a été membre du comité de programme de JNPC 2001 et est membre du comité de programme de AMAI'2002 (conference on artificial intelligence and mathematics).

9.2 Participation à des colloques

- J-P. Merlet est intervenu dans un séminaire de prospective du Sénat sur la robotique [15].

- M. Rueher a participé à JFPC 2001 : 10^{es} journées francophones de programmation en logique et programmation par contraintes à Paris (24-27 avril).
- B. Neveu, Gilles Trombettoni, Blaise Madeline et Christophe Jermann ont participé à JNPC 2001 7^{es} journées nationales sur la résolution pratique des problèmes NP-complets à Toulouse (28-30 juin).
- B. Madeline a participé à MIC 2001 : 4th Metaheuristics International Conference à Porto, Portugal (16-20 juillet).
- C. Michel, Bertrand Neveu, Michel Rueher et Gilles Trombettoni ont participé à CP 2001 Seventh International Conference on Principles and Practice of Constraint Programming à Paphos, Chypre (26 novembre-1 décembre).

9.3 Enseignement

Il faut souligner que l'équipe a une activité d'enseignement importante, compte-tenu de la présence en son sein d'un professeur et de deux maîtres de conférences.

9.3.1 Cours dans le domaine de recherche et cours de DEA

J-P. Merlet a donné des cours de *robotique* à l'Ensta, à l'ISIA et à l'Ecole des Mines de Paris (15 heures).

C. Michel a assuré un cours de DEA sur la *réflexivité dans les langages fonctionnels* au DEA d'informatique (3 heures).

B. Neveu a participé au cours d'*intelligence artificielle* de l'Entpe de Lyon (6 heures).

M. Rueher, B. Neveu et G. Trombettoni ont assuré des cours et des TP de *programmation par contraintes* en DEA d'informatique à l'Unsa et à l'Essi (10 heures).

M. Rueher a donné un cours d'*introduction à la programmation par contraintes* en maîtrise d'informatique (Unsa).

9.3.2 Jurys de thèse et d'HDR

J-P. Merlet a été membre de jury pour une thèse et pour une HDR.

M. Rueher a rapporté 2 thèses, a été président de jury de thèse à deux reprises et examinateur d'une thèse.

9.3.3 Responsabilités d'enseignements

H. Collavizza est *responsable d'études* de la 1^{re} année de l'Essi.

G. Trombettoni est responsable des enseignements en informatique à l'Iut Gtr. Il est également responsable de l'*année spéciale* en GTR (formation en un an pour des étudiants ayant déjà un niveau bac+2).

9.3.4 Autres cours

H. Collavizza a assuré ses enseignements à l'Essi. Elle a donné un cours et des TP d'*architecture*, ainsi que des TP d'*algorithmique* et de *Java* en première année. Elle a donné des cours et des TD de *logique et de programmation en logique* en 2^{me} année.

B. Madeline a été *moniteur* à la Faculté des Sciences de l'Unsa où il a enseigné en 2^{me} année du DEUG MIAS (Math-Informatique) (39 heures), il a donné des TP de *Système Informatique* (10 heures), des TD sur les *Outils Formels pour l'Informatique* et des TP sur les *Structures de Données et Algorithmique* en Java (42 heures).

C. Michel a donné des TP d'*Unix* (commandes, shell scripts) en 1^{re} année d'IUT GTR (60 heures).

M. Rueher a assuré des TD de *bases de données* en Essi 2^{me} année. Il a également donné des cours et des TD de *logique et de programmation en logique* en ESSI 2^{me} année. Il a encadré un *projet d'étude* de 3^{me} année de l'ESSI.

G. Trombettoni a assuré des cours, TD et TP dans divers enseignements en Iut Gtr 1^{re} et 2^e année : *Unix* (commandes, shell scripts), *programmation système* (processus, mémoire, sockets), *algorithmique, Java, C*.

P. Urso a été *moniteur* à la Faculté des Sciences de l'UNSA. Il a enseigné *l'informatique et programmation* (en Java) à deux groupes de DEUG MIAS 1^{re} année (26 heures par groupe). Il a également enseigné la *conception objet en Java* aux étudiants de la maîtrise MASS (26 heures).

9.4 Thèses

Thèses en cours :

1. B. Madeline, *Algorithmes évolutionnaires, adaptatifs, spécialisés pour la résolution des problèmes de satisfaction de contraintes*, université de Nice-Sophia Antipolis.
2. C. Jermann, *Résolution de contraintes géométriques par propagation d'intervalles et propagation locale*, université de Nice-Sophia Antipolis.
3. L. Rolland, *Algorithmes algébriques pour la commande de robots parallèles de haute précision*, université de Nancy.
4. P. Urso, *Preuves par induction dans le calcul de situation*, université de Nice-Sophia Antipolis.

10 Bibliographie

Ouvrages et articles de référence de l'équipe

- [1] C. BLIEK, B. NEVEU, G. TROMBETTONI, « Using Graph Decomposition for Solving Continuous CSPs », in : *Principles and Practice of Constraint Programming, CP'98, LNCS, 1520*, Springer, p. 102–116, 1998.
- [2] B. N. C. JERMANN, G. TROMBETTONI, M. RUEHER, « A Constraint Programming Approach for Solving Rigid Geometric Systems », in : *Proceedings of CP'2000, Sixth International Conference on Principles and Practice of Constraint Programming, 1894*, LIM, p. 119–134, 2000.
- [3] A. GOTLIEB, B. BOTELLA, M. RUEHER, « A CLP Framework for Computing Structural Test Data », in : *Proceedings of CL2000 (First International Conference on Computational Logic, Constraints Stream), London, 1861*, LNCS, Springer Verlag, p. 399–413, 2000.
- [4] J.-P. MERLET, *Parallel robots*, Kluwer, Dordrecht, Pays-Bas, 2000.

- [5] Y. PAPEGAY, « From Modeling to Simulation with Symbolic Computation : An Application to Design and Performance Analysis of Complex Optical Devices », *in : Proceedings of the Second Workshop on Computer Algebra in Scientific Computing*, Springer Verlag, Munich, June 1999.
- [6] J. RÉGIN, M. RUEHER, « A global constraint combining a sum constraint and difference constraints », *in : CP'2000, Sixth International Conference on Principles and Practice of Constraint Programming, 1894*, LNCS, Springer Verlag, p. 384–395, 2000.

Articles et chapitres de livre

- [7] J.-P. MERLET, « A parser for the interval evaluation of analytical functions and its applications to engineering problems », *J. Symbolic Computation* 31, 2001, p. 475–486.

Communications à des congrès, colloques, etc.

- [8] B. MADELINE, B. NEVEU, « Méthodes de recherche arborescente comparées aux algorithmes génétiques sur des problèmes de coloriage de graphes sur-contraints », *in : Proc. of JNPC'01*, p. 185–195, Toulouse, France, juin 2001.
- [9] B. MADELINE, B. NEVEU, « Tree Search Methods versus Genetic Algorithms for Over-Constrained Graph Coloring Problems », *in : Proc. of MIC'01, II*, p. 731–736, Porto, Portugal, juillet 2001.
- [10] J.-P. MERLET, M. DAHAN, « Le micro-robot parallèle MIPS », *in : Quatrièmes Journées du Pôle Micro-robotique*, Lyon, France, juillet 2001.
- [11] J.-P. MERLET, D. DANNEY, « A formal-numerical approach to determine the presence of singularity within the workspace of a parallel robot », *in : Computational Kinematics*, F. Park, C. Iurascu (éditeurs), p. 167–176, Séoul, Corée, mai 2001.
- [12] J.-P. MERLET, « An improved design algorithm based on interval analysis for parallel manipulator with specified workspace », *in : IEEE Int. Conf. on Robotics and Automation*, Séoul, Corée, mai 2001.
- [13] J.-P. MERLET, « Méthodes de résolutions et d'analyse d'équations : Applications en robotique », *in : Journées Nationales de la Recherche en Robotique*, Giens, France, octobre 2001.
- [14] J.-P. MERLET, « Micro parallel robot MIPS for medical applications », *in : IEEE Int. Conf. on Emerging Technologies and Factory Automation*, Antibes, France, octobre 2001.
- [15] J.-P. MERLET, « Perspectives court et moyen terme pour la robotique », *in : Rencontre internationale de prospective du Sénat*, Paris, France, juin 2001.
- [16] J.-P. MERLET, « System-Solving and Parallel Robots », *in : Workshop on Robot Mechanics*, Paris, France, juillet 2001.
- [17] C. MICHEL, M. RUEHER, Y. LEBBAH, « Solving constraints over floating-point numbers », *in : CP'2001, Seventh International Conference on Principles and Practice of Constraint Programming*, Paphos, Chypre, décembre 2001.
- [18] G. TROMBETTONI, B. NEVEU, « Links for Boosting Predictable Interactive Constraint Systems », *in : UICS'01, International Workshop on User Interaction in Constraint Satisfaction, in the CP'2001 conference*, Paphos, Chypre, décembre 2001.

Rapports de recherche et publications internes

- [19] B. BOTELLA, A. GOTLIEB, C. MICHEL, M. RUEHER, P. TAILLIBERT, « Génération automatique de cas de test structurels avec les techniques de programmation par contraintes », *rapport de recherche n° 2001-12*, I3S/CNRS, Université de Nice–Sophia Antipolis, décembre 2001.