Curruculum Vitæ Détaillé

FLORIAN LETOMBE

Centre de Recherche en Informatique de Lens (CRIL)
Université d'Artois
rue de l'Université, S.P. 16
F-62307 Lens Cedex
Mél : `letombe@cril.univ-artois.fr`

Sommaire

1 Curriculum vitæ . . . **2**
- 1.1 Notice Individuelle . . . 2
- 1.2 Cursus Universitaire . . . 3
- 1.3 Expérience Professionnelle . . . 4

2 Recherche . . . **6**
- 2.1 Activités de Recherche . . . 6
- 2.2 Perspectives de Recherche . . . 7
- 2.3 Publications et Séminaires . . . 8

3 Activités d'enseignement . . . **10**
- 3.1 Détail des enseignements . . . 10
- 3.2 Contenu des enseignements . . . 11
- 3.3 Nota bene . . . 12

4 Responsabilités collectives . . . **14**

---

1 Curriculum vitæ

1.1 Notice Individuelle

Nationalité	Française
Âge	27 ans (né le 16 octobre 1979)
Lieu de naissance	Arras (62)
Situation familiale	Pacsé
Obligations militaires	Dégagé
Adresse Professionnelle	CRIL–Université d'Artois
	Faculté J.Perrin, SP 18
	62 307 Lens CEDEX
Téléphone	03 21 79 17 95 (bureau Faculté)
	03 21 79 19 51 (bureau IUT-SRC)
Mél.	letombe@cril.univ-artois.fr
Site internet	`http://www.cril.univ-artois.fr/~letombe`
Fonction	ATER à l'IUT de Lens de l'Université d'Artois
Laboratoire	Centre de Recherche en Informatique de Lens
	(CRIL – CNRS FRE 2499)
Établissements d'Enseignement	Faculté des Sciences et IUT de Lens
	Université d'Artois

1.2 Cursus Universitaire

2002-2005	**Doctorat d'Université : Spécialité informatique** Université d'Artois Laboratoire d'accueil : Centre de Recherche en Informatique de Lens Responsables : Sylvie COSTE-MARQUIS, Daniel LE BERRE et Pierre MARQUIS (directeur de thèse) Laboratoire d'accueil : Centre de Recherche en Informatique de Lens Titre : « De la validité des formules booléennes quantifiées : étude de complexité et exploitation de classes traitables au sein d'un prouveur QBF » Thèse soutenue le 13 décembre 2005, devant le jury composé de : Rapporteurs : M^r Éric Monfroy – PR, Université de Nantes M^r Pascal Nicolas – MCF (HdR), Université d'Angers Examinateurs : M^r Pierre Marquis – PR, Université d'Artois M^{me} Sylvie Coste-Marquis – MCF, Université d'Artois M^r Éric Grégoire – PR, Université d'Artois M^r Daniel Le Berre – MCF, Université d'Artois Invités : M^r Salem Benferhat – PR, Université d'Artois M^r Lakhdar Saïs – PR, Université d'Artois Mention : Très honorable
2002	**DEA d'Informatique** Université des Sciences et Technologies de Lille Laboratoire d'accueil : Centre de Recherche en Informatique de Lens Mémoire : « Représentations implicites en logique propositionnelle » Responsables : Sylvie COSTE-MARQUIS, Daniel LE BERRE et Pierre MARQUIS Mention : Assez Bien
2001	**Maîtrise d'Informatique** Université d'Artois Laboratoire d'accueil : Centre de Recherche en Informatique de Lens Mémoire : « Implantation d'algorithmes fondés sur les formes normales négatives décomposables de formules propositionnelles » Responsables : Sylvie COSTE-MARQUIS et Pierre MARQUIS Mention : Assez Bien
2000	**Licence d'Informatique** Université d'Artois Mention : Assez Bien
1999	**DEUG MIAS (Mathématiques, Informatique et Applications aux Sciences)** Université d'Artois
1997	**Baccalauréat Scientifique (série S)** Lycée Baudimont à Arras (62)

1.3 Expérience Professionnelle

2005-2007 Attaché temporaire d'enseignement et de recherche.

2002-2005 Moniteur de l'enseignement supérieur.
Allocataire ministériel de recherche pour la préparation de la thèse : « De la validité des formules booléennes quantifiées : étude de complexité et exploitation de classes traitables au sein d'un prouveur QBF »

Résumé : Cette thèse est centrée sur le problème QBF de décision de la validité des formules booléennes quantifiées : étant donnée une formule de la forme $\Sigma = \forall y_1 \, \exists x_1 \ldots \forall y_n \, \exists x_n.\phi$ où ϕ est une formule propositionnelle construite sur $\{x_1, y_1, \ldots, x_n, y_n\}$ (la matrice de Σ), il s'agit de déterminer si pour toute affectation d'une valeur de vérité à y_1 dans ϕ, il existe une affectation d'une valeur de vérité à x_1 dans ϕ, ..., pour toute affectation d'une valeur de vérité à y_n dans ϕ, il existe une affectation d'une valeur de vérité à x_n dans ϕ qui rend ϕ valide. Le problème QBF est calculatoirement difficile (**PSPACE**-complet). Il importe donc de mettre en évidence des cas particuliers où sa résolution pratique est envisageable. Dans cette thèse, nous avons considéré des restrictions de QBF portant sur la matrice des instances. Notre objectif principal était double : (1) identifier la complexité de QBF pour des restrictions non considérées jusqu'ici et (2) explorer dans quelle mesure les classes polynomiales pour QBF peuvent être exploitées au sein d'un prouveur QBF général afin d'améliorer son efficacité.
Concernant le premier point, nous avons montré que QBF restreint aux fragments cibles pour la compilation de « connaissances » étudiés dans [Darwiche & Marquis 2002], qui sont traitables pour SAT, reste **PSPACE**-complet et donc intraitable en pratique. Nous avons également mis en évidence le lien étroit qui existe entre notre étude et le problème de compilabilité de QBF.
Concernant le second point, nous avons décrit une nouvelle heuristique de branchement Δ visant à promouvoir la génération de formules Horn renommables quantifiées durant le parcours de l'arbre de recherche effectué par une procédure de type Q-DPLL pour QBF. Les résultats expérimentaux présentés montrent qu'en pratique, hormis notre prouveur **Qbfl**, les prouveurs QBF actuels ne sont pas capables de résoudre facilement les instances Horn quantifiées ou Horn renommables quantifiées ; ceci suffit à justifier l'intérêt de l'approche suivie. Les résultats obtenus montrent également que, muni de l'heuristique Δ, **Qbfl** améliore significativement ses performances sur certains types d'instances, même s'il obtient des résultats moyens en général, comparé aux autres prouveurs QBF actuels.

Mots clés : formules booléennes quantifiées, logique propositionnelle, compilation de « connaissances », restrictions traitables, complexité, heuristiques de branchement.

2002 Stage de DEA de 5 mois au sein du CRIL.

Sujet : « Représentations implicites en logique propositionnelle »

Résumé : Notre travail porte sur la représentation d'ensembles d'impliquants/impliqués de formules propositionnelles. Il a pour objectif de définir et d'évaluer des méthodes permettant de représenter implicitement de tels ensembles.
Notre contribution tient en trois points :

1. Nous proposons plusieurs extensions de la méthode des Méta-Produits proposée par J.-C. Madre et O. Coudert pour caractériser de façon récursive l'ensemble des impliquants premiers d'une formule. Plusieurs ensembles ont été visés (impliquants/impliqués fondamentaux, de taille bornée ou non) et les caractérisations récursives obtenues constituent le socle d'algorithmes - tous programmés - pour calculer de tels ensembles.
2. Nous évaluons le langage de représentation implicite d'ensembles d'impliqués fondamentaux obtenu en déterminant s'il supporte ou non en temps polynomial plusieurs requêtes et transformations basiques.
3. Nous évaluons enfin empiriquement les algorithmes proposés en mesurant en particulier la taille des représentations implicites obtenues sur un ensemble important de jeux d'essai.

Mots clés : logique propositionnelle, représentation implicite, impliquants, impliqués, Diagrammes de Décision Binaires (BDDs), Méta-Produits.

2001 Stage de Maîtrise de 4 mois au sein du CRIL.

Sujet : « Implantation d'algorithmes fondés sur les formes normales négatives décomposables (DNNFs) de formules propositionnelles »

Résumé : Nous implantons des algorithmes introduits par A. Darwiche permettant de représenter, dans un premier temps, des formules propositionnelles sous la forme d'arbres de décomposition. Nous utilisons ensuite ces derniers afin de construire les DNNFs associées. En outre, des opérations de simplification, de conditionnement, d'oubli, de comptage des modèles et de satisfiabilité à partir de DNNFs sont implémentées.

Mots clés : logique propositionnelle, Formes Normales Négatives Décomposables (DNNFs).

2 Recherche

2.1 Activités de Recherche

Le test de validité de formules booléennes quantifiées présente depuis quelques années un engouement croissant en France et à l'étranger, notamment en Autriche et en Italie où la communauté est particulièrement active. Ce phénomène est dû au fait que l'on peut réduire de nombreux problèmes d'intelligence artificielle tels que l'existence de plans d'action ou divers raisonnements non monotones (abduction, logique par défaut, circonscription, etc ...) au test de validité des formules booléennes quantifiées, tel que l'illustre la figure 1.

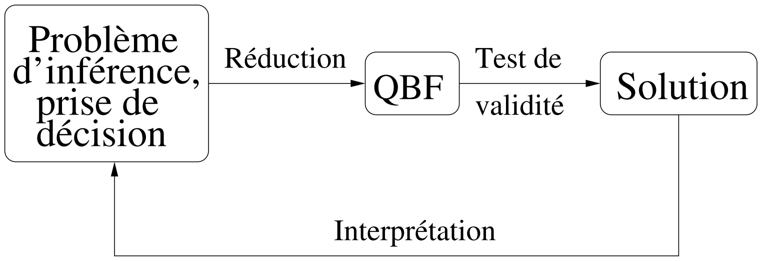

FIG. 1 – QBF : un langage pivot.

L'utilisation des formules booléennes quantifiées pour exprimer des tâches de raisonnement avancées a été évoquée par M. Cadoli, A. Giovanardi et M. Schaerf en 1998. Elle est une généralisation naturelle d'une méthode similaire appliquée pour les problèmes NP-complets. De tels problèmes sont souvent résolus par une réduction à SAT, le problème de satisfiabilité de la logique propositionnelle classique, qui est NP-complet.

Pour la suite, nous désignons les formules booléennes quantifiées par QBF et le test de validité de telles formules par QBF. Nous savons déjà que QBF est un problème PSPACE-complet[1] [Stockmeyer & Meyer 1973] et que certaines restrictions sont décidables en temps polynomial.

Parmi les prouveurs QBF existants, nous pouvons citer *QKN*, *Evaluate*, *Decide*, QSOLVE, *QuBE*, *Semprop*, ou encore plus récemment *Quaffle*, *sKizzo*, ou *Qbfl* (mon prouveur), qui depuis, pour la plupart, sont sujets à de constantes améliorations.

Parmi les orientations possibles pour la résolution de QBFs, nous nous sommes dirigés vers des méthodes basées sur la notion de restriction. L'idée clé est de reconnaître les instances pour lesquelles des algorithmes spécifiques peuvent fournir une solution beaucoup plus efficacement que les prouveurs QBF généraux. Plusieurs restrictions traitables de QBF ont déjà été identifiées. En effet, un algorithme en temps linéaire existe pour les QBFs à matrice CNF contenant des clauses d'au plus deux littéraux [Aspvall, Plass & Tarjan 1979]. Il existe également un algorithme cubique capable de résoudre les formules de Horn quantifiées [Kleine-Büning, Karpinski & Flögel 1995].

Notre travail se divise en deux parties distinctes : une partie théorique et une partie pratique.

La partie théorique a consisté à identifier la classe de complexité de QBF pour certaines restrictions prouvées polynomiales pour SAT. Nous avons montré dans quelle mesure il est possible d'étendre les méthodes polynomiales de résolution de formules propositionnelles « classiques » aux QBFs.

La partie pratique se décompose elle-même en deux objectifs. Le premier est d'écrire un prouveur QBF efficace capable de résoudre un maximum d'instances lors des évaluations comparatives de prouveurs QBF. Ces évaluations existent depuis 2003 (`http://qbflib.org/qbfeval`) et mon prouveur *Qbfl* y participe depuis la première édition. Pour cela, nous incluons les algorithmes po-

[1]PSPACE est la classe des problèmes décidables en espace polynomial. Elle contient tous les problèmes de la Hiérarchie Polynomiale.

lynomiaux que nous créons lors de la phase théorique et nous ajoutons également de nouvelles heuristiques orientées spécialement vers ces classes polynomiales spécifiques. Nous avons, pour notre second objectif, proposé des instances à l'évaluation de prouveurs QBF. Ces dernières sont le fruit de générateurs de formules Horn quantifiées et de formules Horn renommables quantifiées implantés par nos soins. Les instances proposées sont facilement résolues par *Qbfl* et les résultats de la dernière évaluation permettent de montrer que les autres prouveurs ne sont pas efficaces sur ce type d'instances.

2.2 Perspectives de Recherche

Ces travaux appellent un certain nombre de perspectives. L'une d'entre elles consiste à étendre davantage le panorama de complexité de QBF. En particulier, il serait intéressant de déterminer en quelles mesures la notion de modèles K-booléens de [Kleine-Buning *et al.* 2003], dans le cas où K est une classe de fonctions booléennes qui peuvent être encodées en espace polynomial, pourrait être exploitée pour donner naissance à de nouvelles restrictions de QBF qui sont calculatoirement plus faciles (selon les hypothèses usuelles de la théorie de la complexité).

Dans le même ordre d'idées, une perspective est d'élargir notre investigation à l'étude de restrictions de QBF basées sur les quantifications. À l'instar de ce que nous avons fait pour impliquants et les impliqués premiers, il s'agit d'étudier des restrictions portant conjointement sur la nature du préfixe et de la matrice des instances. En effet, d'un point de vue théorique, ne considérer que des restrictions sur le préfixe (i.e., sans aucune exigence sur la matrice) ne semble pas aussi intéressant. Alors que le fait de limiter le nombre d'alternances de quantificateurs dans le préfixe provoque immédiatement une diminution de la complexité de QBF de **PSPACE** à un niveau donné dans la hiérarchie polynomiale, cela ne conduit pas à la traitabilité (sous les hypothèses habituelles de la théorie de la complexité) ; ainsi, dans le cas limite où aucune alternance de quantificateurs n'apparaît dans le préfixe, QBF se réduit à SAT ou à UNSAT (dépendant de la nature des quantifications des variables), et aucun de ces problèmes ne peut vraisemblablement appartenir à **P** si aucune supposition sur la matrice n'est faite. Bien entendu, cela ne signifie pas que la façon dont les quantifications sont traitées n'a pas d'impact sur l'efficacité des prouveurs QBF. Par exemple, une QBF Σ de la forme $\forall\{x_1, \ldots, x_{n-1}\}\exists\{x_n\}.\phi$ où ϕ est une formule CNF peut être résolue en temps quadratique (construire une représentation CNF de $\exists\{x_n\}.\phi$ en utilisant la résolution pour oublier x_n dans ϕ, réduire ensuite toutes les clauses résultantes en y supprimant tout littéral construit à partir de $\{x_1, \ldots, x_{n-1}\}$; Σ est valide si et seulement si la formule CNF résultante ne contient pas la clause vide). Pourtant, les prouveurs QBF de type Q-DPLL qui traitent les quantificateurs du plus externe au plus imbriqué peuvent requérir un temps exponentiel pour résoudre des formules de ce type. Une étude plus approfondie de l'interaction entre les restrictions basées sur le préfixe et celles basées sur la matrice constitue donc une perspective de recherche.

Pour poursuivre le volet théorique, nous envisageons aussi d'étudier la possibilité d'étendre les méthodes de résolution polynomiales pour SAT à des fragments incomplets, tels que celui des formules ordonnées par exemple, au cas QBF. Ce dernier point ouvre des perspectives très intéressantes. En effet, les formules ordonnées sont plus expressives que les formules Horn. De plus, il existe une extension des formules ordonnées, les formules ordonnées renommables qui sont également plus expressives que les formules Horn renommables.

Si de telles classes se révèlent également polynomiales pour QBF, il serait intéressant d'étudier comment les exploiter au sein d'un prouveur de type Q-DPLL (soit de façon passive, par un test de détection systématique ou non aux points de choix, soit de façon plus active – comme nous l'avons fait pour les formules Horn renommables CNF) en mettant au point des heuristiques visant à leur promotion.

D'un point de vue plus pratique, nous souhaitons aussi améliorer l'efficacité de notre prouveur en y incorporant les techniques utilisées dans les prouveurs les plus efficaces (en particulier le retour arrière intelligent).

Tous ces travaux pourraient faire l'objet d'une extension aux QCSPs. Il serait, en outre, intéressant de regarder si certaines des méthodes de résolution polynomiales peuvent s'étendre à ce problème de satisfaction de contraintes quantifiées.

2.3 Publications et Séminaires

Revues d'audience internationale avec comité de rédaction :

[1] S. Coste-Marquis, D. Le Berre, F. Letombe, P. Marquis.
« Complexity Results for Quantified Boolean Formulae Based on Complete Propositional Languages ».
Journal on Satisfiability, Boolean Modeling and Computation (JSAT), vol. 1, p. 61–88, 2006.

Manifestations d'audience internationale avec comité de lecture :

[2] S. Coste-Marquis, D. Le Berre, F. Letombe, P. Marquis.
« Propositional Fragments for Knowledge Compilation and Quantified Boolean Formulae ».
In Proceedings of The 20^{th} National Conference on Artificial Intelligence (AAAI'05), p. 288–293, Pittsburgh, Pensylvania, USA, 2005.

[3] S. Coste-Marquis, D. Le Berre et F. Letombe.
« A Branching Heuristic Directed to Renamable Quantified Horn Formulas ».
In Proceedings of The 8^{th} International Conference on Theory and Applications of Satisfiability Testing (SAT'05), Lecture Notes in Computer Science 3569, p. 393–399, St. Andrews, UK, 2005 (Papier court).

Manifestations d'audience nationale :

[4] F. Letombe.
« Deux fragments polynomiaux complets pour le problème de la validité des formules booléennes quantifiées ».
Dans les Actes des $2^{èmes}$ Journées Francophones de Programmation par Contraintes (JFPC'06), p. 387–396, 2006.

[5] S. Coste-Marquis, D. Le Berre, F. Letombe, P. Marquis.
« Fragments propositionnels pour la compilation de connaissances et formules booléennes quantifiées ».
Dans les Actes du $15^{ème}$ Congrès Francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle (RFIA'06), 2006.

[6] F. Letombe.
« Une heuristique de branchement dirigée vers les formules Horn renommables quantifiées ».
Dans les Actes des $7^{èmes}$ Rencontres Jeunes Chercheurs en Intelligence Artificielle (RJCIA'05), p. 183–196, 2005.

Thèse pour le grade de docteur en Informatique (Doctorat) :

[7] F. Letombe.
« De la validité des formules booléennes quantifiées : étude de complexité et exploitation de classes traitables au sein d'un prouveur QBF ».
Centre de Recherche en Informatique de Lens, CNRS FRE 2499, soutenue le 13 décembre 2005.

Mémoire de DEA :

[8] F. Letombe.
« Représentations implicites en logique propositionnelle ».
Centre de Recherche en Informatique de Lens, CNRS FRE 2499, 2002.

Rapports techniques :

[9] F. Letombe.
« Algorithmique pour les Formules Booléennes Quantifiées ».
Centre de Recherche en Informatique de Lens, CNRS FRE 2499, 2003.

[10] F. Letombe.
« Classes polynomiales pour les Formules Booléennes Quantifiées ».
Centre de Recherche en Informatique de Lens, CNRS FRE 2499, 2004.

Conférences :

J'ai été amené à présenter mes travaux lors des conférences nationales et internationales suivantes :
- conférences internationales : AAAI'05 à Pittsburgh (Pensylvania, USA),
- conférences nationales : RJCIA'05 à Nice, RFIA'06 à Tours, JFPC'06 à Nîmes.

Séminaires :

- « Algorithmique pour les Formules Booléennes Quantifiées », Centre de Recherche en Informatique de Lens, CNRS FRE 2499, Paris, 2003. Dans le cadre de l'Action Spécifique n°83 du département STIC du CNRS au cours de l'année 2002-2003.
- « Classes polynomiales pour les Formules Booléennes Quantifiées », Centre de Recherche en Informatique de Lens, CNRS FRE 2499, Lens, 2004.

3 Activités d'enseignement

Pour la seconde année consécutive, j'effectue mes enseignements en qualité d'ATER au sein de l'IUT de Lens (Université d'Artois) au département Services et Réseaux de Communication (SRC). Durant mes trois années de thèse, j'ai effectué mes enseignements au sein de la Faculté des Sciences Jean Perrin de Lens (Université d'Artois) en qualité de moniteur de l'enseignement supérieur. Enfin, j'ai réalisé des vacations durant mon stage de DEA. Le tableau suivant synthétise mes activités d'enseignement.

3.1 Détail des enseignements

ATER Année 2006–2007			
Formation	**Matière**	**Type**	**Volume horaire (équivalent TD)**
SRC $1^{ère}$ Année	Algorithmique	CM	18h
SRC $1^{ère}$ Année	Algorithmique	TD	48h
SRC $1^{ère}$ Année	Algorithmique	TP	32h
SRC $1^{ère}$ Année	Systèmes d'exploitation Architecture des ordinateurs	CM	9h
SRC $1^{ère}$ Année	Systèmes d'exploitation Architecture des ordinateurs	TD	12h
SRC $1^{ère}$ Année	Systèmes d'exploitation Architecture des ordinateurs	TP	16h
SRC $2^{ème}$ Année	Programmation Orientée Objets	CM	18h
SRC $2^{ème}$ Année	Programmation Orientée Objets	TD	12h
SRC $2^{ème}$ Année	Programmation Orientée Objets	TP	28h
		Total :	193h

ATER Année 2005–2006			
Formation	**Matière**	**Type**	**Volume horaire (équivalent TD)**
SRC $1^{ère}$ Année	Algorithmique	CM	18h
SRC $1^{ère}$ Année	Algorithmique	TD	48h
SRC $1^{ère}$ Année	Algorithmique	TP	32h
		Total :	98h

Moniteur Année 2004–2005			
Formation	**Matière**	**Type**	**Volume horaire (équivalent TD)**
LMI $3^{ème}$ Année	Algorithmique	TP	16h
LMI $1^{ère}$ Année	Programmation en Scheme	TD	24h
LMI $1^{ère}$ Année	Programmation en Scheme	TP	16h
LMI $1^{ère}$ Année	Programmation en Pascal	TP	1h
		Total :	57h

Moniteur Année 2003–2004			
Formation	**Matière**	**Type**	**Volume horaire (équivalent TD)**
DEUG MIAS $2^{ère}$ Année	Programmation en Pascal	TP	10h
DEUG MIAS $1^{ère}$ Année	Programmation en Pascal	TD	21h
DEUG MIAS $1^{ère}$ Année	Programmation en Pascal	TP	18h
DEUG SV $1^{ère}$ Année	Programmation en Maple	TD	10h
DEUG SV $1^{ère}$ Année	Programmation en Maple	TP	6h40
		Total :	65h40

Moniteur Année 2002–2003			
Formation	**Matière**	**Type**	**Volume horaire (équivalent TD)**
DEUG MIAS $2^{ère}$ Année	Programmation en Pascal	TP	10h
DEUG MIAS $1^{ère}$ Année	Programmation en Pascal	TP	24h
DEUG MIAS $1^{ère}$ Année	Programmation en Scheme	TP	12h
DEUG SV $1^{ère}$ Année	Programmation en Maple	TD	10h
DEUG SV $1^{ère}$ Année	Programmation en Maple	TP	13h20
		Total :	69h20

Vacataire Année 2001–2002			
Formation	**Matière**	**Type**	**Volume horaire (équivalent TD)**
DEUG MIAS $1^{ère}$ Année	Programmation en Pascal	TP	40h
DEUG SV $1^{ème}$ Année	Bureautique	TP	8h
		Total :	48h

3.2 Contenu des enseignements

Enseignements en DEUG (ou Licence $1^{ère}$ et $2^{ème}$ année)

- Langage Scheme (TP) : l'apprentissage du langage Scheme est proposé aux étudiants de $1^{ère}$ année de DEUG MIAS ou Licence Mathématiques Informatique (LMI) actuellement. Le but est d'aborder à un niveau élémentaire les notions de base telles que expressions, variables, fonctions, structures de contrôle, structures de données, itérateurs ; mais surtout à familiariser les étudiants à la récursivité.
  J'ai participé, les deux années pendant lesquelles je l'ai enseigné, à l'élaboration de sujets de TP (3 par an) ainsi qu'à l'écriture des sujets d'examen. La correction des 2 examens par an est également à notre charge.
- Langage Scheme (TD) : le public est le même que pour les TP. J'ai préparé trois sujets de TD.
- Langage Maple (TP) : le public concerné est composé des étudiants en $1^{ère}$ année de DEUG SV (LSV1). Le but est de faire découvrir les différentes fonctionnalités offertes par ce logiciel. Cela constitue également une très bonne approche, pour ces étudiants qui découvrent la programmation, des langages interprétés.
  J'ai participé à l'élaboration de 2 sujets de TP par an et par groupe (soit 6 sujets), aux énoncés des examens ainsi que la correction des contrôles de mes étudiants.
- Langage Maple (TD) : même public que précédemment. J'ai participé à l'écriture des différents sujets et à la correction du contrôle continu et des examens.
- Langage Pascal (TP) : cet enseignement est dispensé aux étudiants de $1^{ère}$ et $2^{ème}$ année de DEUG MIAS. Le but est d'initier les étudiants de $1^{ère}$ année, d'une part, à la pratique de l'informatique (découverte de windows) et, d'autre part, à l'apprentissage les rudiments de la programmation. Nous avons préparé en binôme les différents TP ainsi que l'élaboration d'un mini-projet afin d'évaluer leurs capacités. Nous avions la charge de corriger ce projet ainsi que de proposer comme contrôle, une modification sur ce projet.
  Il en est de même pour les étudiants de $2^{ème}$ année, à la différence près que, forts des connaissances acquises en $1^{ère}$ année, ils sont sujets à une exigence accrue de notre part. Les prémices de l'algorithmique sont abordés.
- Langage Pascal (TD) : cet enseignement est dispensé aux étudiants de $1^{ère}$ année de DEUG MIAS. L'objectif est d'initier les étudiants à la programmation (itération, choix, procédures et fonctions, tableaux, types énumérés). J'avais la charge de corriger 1/3 des copies lors des différentes évaluations (DS+Examens).

- Bureautique (TP) : cet enseignement est dispensé à des étudiants en $1^{ère}$ année de DEUG SV. Il constitue une initiation pratique à l'informatique (utilisation de windows) et à l'apprentissage du traitement de texte (Word) ainsi que du tableur (Excel). J'ai participé à l'élaboration des différents sujets de TP, ainsi qu'à l'écriture et la correction des examens pour mon groupe.

Enseignement en Licence d'informatique (ou Licence $3^{ème}$ année)

- Algorithmique, structure de données et programmation (TP) : dans un premier temps, il était nécessaire de procéder à quelques rappels sur le langage PASCAL et les commandes de base Linux. En effet, le caractère hétérogène du public oblige à quelques remises à niveau. Ensuite nous avons abordé des structures de données de base (piles, files). Nous avons fait des rappels sur la récursivité et la manipulation des pointeurs. Enfin nous avons proposé un projet qui consistait à programmer un annuaire réversible dont la structure de donnée était un arbre binaire de recherche à double entrée (une pour les noms, une autre pour les numéros de téléphone).
  Tous les sujets de TP et l'énoncé du projet ont été réalisés en binôme. Nous avions la charge de corriger les projets de nos étudiants ainsi que le contrôle TP.

Enseignement en IUT SRC ($1^{ère}$ et $2^{ème}$ année)

- Algorithmique (CM) : l'investissement est total pour ce cours puisqu'il a fallu l'écrire dans son intégralité. J'ai choisi le langage C++ pour pseudo-langage afin de préparer les étudiants, dans un premier temps, à leurs cours de C++ suivis en seconde année, mais également pour leur avenir en entreprise car ce langage est très utilisé et performant. Ce cours contient les bases de l'algorithmique (historique, types de base, notions de programmation structurée, analyse et décomposition de problèmes, etc). De plus, j'ai écrit un sujet d'examen en contrôle continu (DS) et un sujet d'examen final pour le premier semestre.
- Algorithmique (TD) : tous les sujets sont à ma charge et je suis le seul à les assurer.
- Algorithmique (TP) : de même, j'écris tous les sujets pour moi et un collègue. Deux examens de TP sont proposés aux étudiants pour chaque semestre.
- Systèmes d'exploitation et architecture des ordinateurs (CM) : de même, ce cours est entièrement écrit par mes soins. Il a pour but de présenter aux étudiants le mode de fonctionnement des ordinateurs ainsi que de leur faire découvrir Linux.
- Systèmes d'exploitation et architecture des ordinateurs (TD) : tous les sujets sont à ma charge et je suis le seul à les assurer.
- Systèmes d'exploitation et architecture des ordinateurs (TP) : de même, j'écris tous les sujets pour moi et un collègue.
- Programmation Orientée Objets (CM) : là encore, je conçois entièrement ce cours qu'il convient d'adapter à un public de communicants.
- Programmation Orientée Objets(TD) : tous les sujets sont à ma charge et je suis le seul à les assurer.
- Programmation Orientée Objets (TP) : de même, j'écris tous les sujets pour moi et un collègue.

3.3 Nota bene

J'ai également suivi des stages de formation avec le Centre d'Initiation à l'Enseignement Supérieur (`http://ustl.univ-lille1.fr/cies`) du Nord Pas-de-Calais Picardie dont le détail est donné sur mon site.

Soulignons enfin que mes année de thèse m'ont conduit à programmer en C, en C++ et en BASH. De par mon sujet de thèse, je suis très à même de présenter des méthodes de résolution de problèmes par contraintes (SAT, CSP, QBF, QCSP). Je me sens capable d'assurer des enseignements dans chacun de ces domaines (système, programmation par contrainte, etc) mais reste ouvert à toute autre forme d'enseignement.

4 Responsabilités collectives

J'ai participé aux différents tremplins d'automne dont le principe était de montrer aux étudiants de terminale les différents débouchés en informatique. J'ai aussi participé une journée à Science en Fête, en expliquant de manière ludique aux élèves de primaire et de collège quelques applications (colonies de fourmis, problème du voyageur de commerce,..).

Mes enseignements m'ont également conduit à participer à la surveillance de DS et d'examens et à participer à des jurys divers (grands jurys, jurys de soutenance de projets tutorés, etc).

J'ai enfin fait partie du comité d'organisation de JFPC'05 (Premières Journées Francophones de Programmation par Contraintes) qui s'est déroulé à Lens en juin 2005.