VERNA, D.. CLOS efficiency : Instantiation. In *Proceedings of the International Lisp Conference*, pages 76–90. Association of Lisp Users

Cet article présente les résultats d'une recherche exprérimentale en cours sur le comportement et les performances de CLOS, la couche orientée objet de Common Lisp. Notre but est d'évaluer le comportement et les performances de trois des caractéristiques principales de tout système objet dynamique : l'instantiation, l'accès aux membres et la sélection dynamique de méthodes. Cet article décrit les résultats de la partie instantiation. Nous évaluons l'efficacité du processus d'instantiation en Lisp et en C++, selon un ensemble de paramètres tels les types des membres ou encore les hiérarchies de classes. Nous montrons que dans une configuration où la sûreté est privilégiée sur la rapidité, le comportement de C++ et Lisp peut être très différent, y compris quand on compare des compilateurs Lisp entre eux. D'un autre côté, nous montrons que lorsque la compilation est optimisée pour l'efficacité du code produit, l'instantiation peut devenir plus rapide en Lisp qu'en C++.

Conférences

Open World Forum Deux membres du LRDE, Akim Demaille et Olivier Ricou, étaient présents au premier Forum Mondial du Libre (Open World Forum)[17] qui s'est tenu à Paris les 1er et 2 décembre 2008. Les deux enseignants-chercheurs se sont exprimés sur l'importance de la formation au logiciel libre dans le cursus des ingénieurs en informatique et son rôle dans l'éducation supérieure.

ACCU'09 Didier Verna, conférencier invité.

Revisiting the Visitor : the "Just Do It" Pattern

Un patron de conception logicielle est une règle tripartite exprimant une relation entre un contexte, un problème et une solution. Le célèbre « GoF Book » décrit 23 de ces patrons de conception. L'influence de ce livre est majeure sur la communauté du génie logiciel. Cependant, Peter Norvig note que 16 d'entre eux sont soit invisibles soit plus simples en Dylan ou Lisp (Design Patterns in Dynamic Programming, Object World, 1996).

Nous prétendons que ceci n'est pas une conséquence de la notion de patron elle-même, mais plutôt de la manière dont ceux-ci sont généralement décrits ; le GoF étant un cas typique sur ce sujet. Tandis que les patrons de conception sont supposés être généraux et abstraits, le GoF est en fait très orienté vers les langages à objets traditionnels tels que C++. Il en résulte que la majorité de ces patrons de conception sont en réalité plus des motifs de programmation, ou des « idiomes » si l'on choisit d'adopter la terminologie du « POSA Book » (Pattern-Oriented Software Architecture).

Dans cette session, nous examinons les patrons de conception du point de vue des langages dynamiques, et plus spécifiquement sous l'angle de CLOS, le système orienté objet de Common Lisp. En prenant le patron VISITEUR comme illustration, nous montrons comment un patron utile du point de vue général peut se retrouver incorporé au langage lui-même, parfois jusqu'au point de disparition totale.

La leçon à retenir est que les patrons de conception doivent être employés avec précaution, et qu'en particulier, ils ne remplaceront jamais une connaissance approfondie de votre langage de prédilection (dans notre cas, la maîtrise des fonctions de première classe et génériques, les fermetures lexicales et le protocole méta-objet). En prenant des patrons de conception au pied de la lettre, on risque de rater la solution évidente et le plus souvent la plus simple : le patron « Faites-le et c'est tout ».

ITiCSE'09 Le LRDE et l'EPITA participent à l'organisation de la 14e conférence annuelle ACM[18]-SIGCSE[19] sur l'innovation et les technologies dans l'enseignement de l'informatique (14th ACM-SIGCSE Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE 2009[20]). Celle-ci aura lieu à Paris du 3 au 8 juillet 2009, et sera hébergée par l'Université Pierre et Marie Curie (Paris 6, France)[21] et organisée par Patrick Brézillon, du LIP6[22]. Roland Levillain, enseignant-chercheur au LRDE, est membre du comité d'organisation et administrateur du site Web de la conférence (`http://iticse09.lrde.org`), hébergé au LRDE.

[17] Forum Mondial du Libre (Open World Forum), `http://www.openworldforum.org/`.
[18] ACM, `http://www.acm/org`.
[19] SIGCSE, `http://www.sigcse/org`.
[20] ITiCSE 2009, `http://iticse09.lrde.org`.
[21] Université Pierre et Marie Curie (Paris 6, France), `http://www.upmc.fr/`.
[22] LIP6, `http://www.lip6.fr/`.

L'air de rien N° 16

Épitéen et Docteur

L'aléastriel du **L**aboratoire de **R**echerche et de **D**éveloppement de l'**E**PITA[1] Numéro 16, Avril 2009

Édito — Dix ans de gestation

par *Akim Demaille*

Au nombre des missions assignées au LRDE dès sa naissance compte la création d'un pont vers le monde universitaire. Comme toute école d'ingénieurs, il est en effet une fraction des étudiants qui veulent embrasser la carrière de la recherche, doubler leurs compétences d'ingénieur par celles de chercheur. Et pour cela, il faut commencer par suivre à la faculté un Master Recherche, autrefois dénommé DEA.

Les plus anciens s'en souviennent, il fut un temps où les épitéens étaient mal reçus dans le monde universitaire : *Que viennent-ils faire ici, ces codeurs ?* Pour que le saut vers l'université réussisse, deux choses étaient nécessaires.

Tout d'abord il fallut monter un cursus de formation par la recherche au sein de l'EPITA afin que nos étudiants ingénieurs ne se retrouvent pas démunis face à leurs pairs de l'université, ayant reçu une formation plus orientée sur les aspects théoriques et formels. Cette formation, c'est CSI, la septième option de spécialisation de l'EPITA, qui repose d'une part, sur une formation scolaire très empreinte de théorie, à la façon de SCIA (Sciences Cognitives et Informatique Avancées), et d'autre part, sur un investissement de deux ans sur de véritables projets de recherche, ceux du LRDE.

Puis il a fallu le courage, le travail et le talent des pionniers, les premiers CSI, qui par leurs exceptionnels résultats dans leurs différentes filières ont montré que non seulement ils n'avaient pas peur de la théorie, mais qu'en plus, épitéens à part entière, ils n'avaient vraiment pas peur du code non plus. C'est à ces Pitou, Couder, sebc, Yann et autres Pollux que l'on doit le renom des CSI dans les Masters Recherche les plus courus des grandes universités parisiennes.

Pourquoi ne parler d'eux qu'aujourd'hui, alors que le LRDE fête ses dix ans ? La période de gestation d'un chercheur est longue ! Pour former les premiers CSI, il a fallu trois années. Compter ensuite un à deux ans d'éducation du monde extérieur pour qu'il reconnaisse en nos étudiants des candidats aptes à réussir le difficile concours des Masters Recherche. Puis une année de Master. Enfin, trois à quatre années de doctorat (pour aboutir à la thèse). Et nous y voilà, presque dix sont passés.

L'air de rien s'est déjà fait l'écho des soutenances de thèse de Jérôme Darbon et d'Alexis Angelidis (N°1), et d'Alexandre Duret-Lutz (N°11). Bien d'autres ont soutenu depuis, dont nous ne parlerons pas dans ce numéro, faute de place : Sylvain Berlemont, Clément Faure, Guillaume Pitel, Pierre-Yves Strub… Mais il était temps que l'air de rien rende hommage à nos docteurs.

Alors voici un numéro spécial, consacré aux thèses soutenues récemment par d'anciens étudiants de CSI. À l'avenir nous tâcherons de les mentionner plus régulièrement, mais ce numéro, condensant ainsi ces présentations de thèses, offre un avantage : il permet de constater que CSI mène à tout ! Derrière l'acronyme trompeur, Calcul Scientifique et Image, se cache avant tout « Formation par la Recherche ». En réalité le sigle initial était « SCI », comme dans Sciences !

[1] L'air de rien, `http://publis.lrde.epita.fr/LrdeBulletin`.

Des programmes corrects par construction — Yann RÉGIS-GIANAS[2], CSI 2003, Docteur 2007

Comment vérifier l'absence d'erreurs dans un programme ? Les méthodes traditionnelles de l'ingénieur font appel à des tests : il s'agit de s'assurer que le programme se comporte correctement quand on le soumet à un ensemble fini de couples (entrée, résultat attendu). Seulement, même si le nombre de tests est important, rien ne prouve qu'il n'existe pas un couple non testé pour lequel la machine n'a pas le comportement attendu.

Souvenons-nous donc qu'un programme n'est pas une machine ordinaire, c'est une machine logique et à ce titre, sujet à l'expertise du mathématicien. Pour montrer l'absence d'erreurs du programme quand on l'utilise correctement, on cherchera à prouver le théorème suivant : « Pour toute entrée I vérifiant la propriété P alors la machine calcule une sortie O vérifiant la propriété Q ». Ces propriétés P et Q, qui forment la *spécification formelle* du programme, sont infiniment plus riches que tout jeu de tests. En effet, une fois prouvée la conformité d'un programme à sa spécification, on a la garantie que toutes ses exécutions futures seront correctes !

Donner un sens aux programmes informatiques — donner une sémantique aux langages de programmation — est la condition *sine qua non* pour pouvoir montrer des théorèmes de correction de programmes. Une sémantique formelle d'un langage définit le sens de ses opérations calculatoires a une échelle si élémentaire qu'elle rappelle celles des règles de la logique. Cette axiomatisation fournit un système de preuves pour les programmes. Avec une armée de mathématiciens et suffisamment de temps, on doit donc pouvoir prouver n'importe quel programme !

Malheureusement, le nombre d'opérations atomiques des programmes qui nous entourent atteint facilement le million. Or, prouver plusieurs millions de théorèmes n'est envisageable pour personne même avec beaucoup de bonne volonté ! Une première étape consiste alors, comme en mathématiques, à abstraire le plus possible les opérations atomiques en fonctions réutilisables. C'est la raison d'être des langages de haut-niveau, comme les langages fonctionnels, qui sont au cœur de ma thèse. On factorise alors le travail de preuve.

Néanmoins, il n'est pas rare qu'une centaine de preuves soit encore nécessaires pour montrer la correction d'un simple programme fonctionnel de quelques dizaines de lignes. C'est la puissance de calcul de l'ordinateur qui arrive alors à la rescousse : un théorème ou une preuve sont des objets symboliques manipulables informatiquement. Pourquoi ne pas utiliser la machine pour prouver nos théorèmes ? Le domaine de l'*analyse statique*, auquel je contribue dans ma thèse avec l'étude des types algébriques généralisés, a pour but d'exhiber des classes de propriétés des programmes qu'on sait prouver automatiquement dans les limites exhibées par Alan Turing (certaines propriétés, comme décider si un programme termine, sont indécidables).

Lorsque l'on atteint ces limites, le programme de vérification a besoin d'aide. On doit alors mettre en place une interface d'interaction entre le programmeur, qui connaît *a priori* les propriétés générales assurant le fonctionnement de son programme — les invariants — et le prouveur automatique, qui s'occupe de les vérifier à chaque étape du calcul. Cette interface d'interaction a été inventée en 1969 par C.A.R Hoare : le programmeur annote d'abord son programme avec des assertions logiques, les fameux invariants, et un algorithme produit un ensemble d'obligations de preuve traitées automatiquement si possible. Ma thèse a contribué à l'adaptation de cette dernière méthode aux langages fonctionnels.

En novembre 2007, j'ai soutenu la thèse que j'ai effectuée au sein du projet Gallium de l'INRIA Rocquencourt[3]. Cette équipe s'intéresse à la conception des langages de programmation sûrs s'appuyant sur des systèmes de types statiques, des systèmes de preuve et de la compilation certifiée. On doit par exemple à cette équipe l'implémentation du langage Objective Caml. Depuis septembre 2008, je suis maître de conférence de l'Université Paris 7 et membre du laboratoire « Preuves, Programmes et Systèmes » [4].

[2] Yann RÉGIS-GIANAS, `http://www.pps.jussieu.fr/~yrg`.
[3] Projet Gallium de l'INRIA Rocquencourt, `http://gallium.inria.fr`.
[4] Laboratoire « Preuves, Programmes et Systèmes », `http://www.pps.jussieu.fr`.

parties manquantes grâce à l'apprentissage automatique. Par exemple, dans une tâche d'apprentissage avec des données arborescentes comme l'analyse grammaticale, on peut écrire un CR-algorithme contenant des briques de parseur puis utiliser des techniques d'apprentissage pour automatiquement lier ces briques les unes aux autres. Dans ce cas, l'apprentissage suppose la connaissance d'un ensemble d'exemples d'apprentissage : des paires contenant une phrase de langage naturel et l'arbre d'analyse grammaticale correspondant.

Dans ma thèse, j'ai montré comment formaliser des CR-algorithmes comme des processus de décision Markoviens. L'utilisation de ce cadre formel permet d'utiliser de nombreux algorithmes d'apprentissage existants pour apprendre des CR-algorithmes. J'ai notamment exploré l'utilisation d'algorithmes d'apprentissage par renforcement et montré que ces algorithmes permettaient de résoudre une large variété de problèmes de prédiction structurée d'une manière simple et très efficace. En particulier, j'ai développé des méthodes capables d'apprendre des transformations complexes entres des arbres HTML et XML, simplement à partir d'exemples de la transformation.

Bien que les CR-algorithmes et les algorithmes d'apprentissage correspondants aient été initialement motivés par la prédiction structurée, leur domaine d'application est plus large. En particulier, j'ai exploré l'utilisation de cette approche pour résoudre des problèmes de « apprendre à chercher ». Pour valider ce nouveau domaine d'application, je me suis basé sur le jeu « le compte est bon ». La taille moyenne de l'espace de recherche dans ce jeu est de 10^6 solutions candidates. Grâce à l'apprentissage d'un CR-algorithme, je suis arrivé à une solution qui résout la majorité des parties en n'explorant que 5 à 15 solutions candidates. Une démo de ce système est disponible en ligne[15].

L'ensemble du code source développé au cours de ma thèse (plus de 80 000 lignes de code) est documenté et publié en code ouvert[16]. **Je suis actuellement en train de développer un compilateur de CR-algorithmes** permettant d'automatiser le travail que j'ai fait manuellement durant ma thèse. J'espère ainsi démocratiser cette approche qui consiste à créer des nouveaux paradigmes mêlant programmation classique avec apprentissage automatique.

Cette thèse a été réalisée au LIP6 au sein de l'équipe MALIRE (Machine Learning and Information Retrieval). Je prévois de la soutenir avant le mois de juin prochain.

Publications

Disponibles sur publis.lrde.epita.fr.

DEHAK, R., DEHAK, N., AND KENNY, P.. The LRDE systems for the 2008 NIST speaker recognition evaluation. In *NIST-SRE 2008*, Montréal, Canada

> Cet article décrit le système de vérification du locuteur présenté par le LRDE en collaboration avec le CRIM à la campagne d'évaluation des systèmes de vérification du locuteur organisée par NIST tous les deux ans. Le système présenté est le résultat de la fusion de quatre systèmes de base. Le premier est basé sur les paramètres acoustiques de la parole et les trois autres systèmes sont basés sur les paramètres prosodiques qui permettent de représenter le style de la voix (manière de prononcer). L'utilisation des systèmes prosodiques a permis d'avoir une amélioration des performances des systèmes basés sur les paramètres acoustiques.

DEHAK, N., KENNY, P., DEHAK, R., GLEMBER, O., DUMOUCHEL, P., BURGET, L., HUBEIKA, V., AND CASTALDO, F.. Support vector machines and joint factor analysis for speaker verification. In *IEEE-ICASSP*, Taipei - Taiwan

> Cet article présente les résultats obtenus durant le workshop CLSP qui a eu lieu à l'université John Hopkins University l'été dernier. Il présente différentes approches pour combiner les méthodes Support Vector Machines (SVM) avec la méthode Joint Factor Analysis (JFA) pour la vérification du locuteur. La méthode JFA est l'une des plus performantes méthodes utilisées pour la vérification du locuteur, elle permet de mieux représenter la variabilité du canal et du locuteur. L'exploitation des SVMs en combinaison avec cette méthode permet d'avoir une meilleure discrimination dans les espaces définis par la JFA. Nous avons obtenu de meilleures performances en utilisant ces deux méthodes.

[15] Démo de ce système est disponible en ligne, `http://nieme.lip6.fr/Nieme/LCeB`.
[16] Publié en code ouvert, `http://nieme.lip6.fr`.

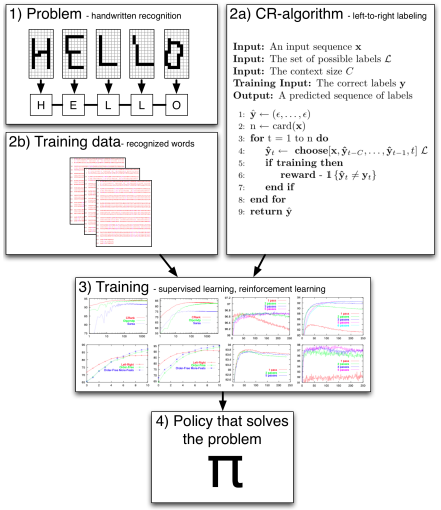
Choose-Reward Algorithms— Learning the inference procedure — Francis MAES[14], CSI 2004, Docteur 2009

Le domaine de l'apprentissage automatique aborde la question de l'approximation d'une fonction à partir d'exemples de couples d'entrée et de sortie correspondantes. Historiquement, ce domaine s'est d'abord intéressé à deux cas : la classification et la régression. La classification consiste à apprendre une fonction dont l'entrée est un vecteur de nombres réels, et la sortie est une classe discrète. Un exemple d'une telle tâche est la classification du courrier électronique, décrit sous forme de vecteur, en *spam* et *non-spam*. Dans le cas de la régression, on travaille sur le même type d'entrées et les sorties sont des scalaires (par exemple prédire le prix d'une voiture, en fonction d'un vecteur de caractéristiques décrivant cette voiture). Néanmoins, de nombreuses applications se décrivent naturellement sur des structures de données plus complexes que des classes, des scalaires ou des vecteurs réels. Citons parmi celles-ci, la prédiction de la structure secondaire des protéines, la traduction automatique, la restauration d'images, l'analyse de l'écrit, la reconnaissance de la parole ou encore l'analyse syntaxique du langage naturel. La recherche s'est donc intéressée au problème de la prédiction de sorties structurées (ou *prédiction structurée*) telles que des séquences, des arbres ou des graphes.

La prédiction de sorties structurées peut être vue comme une généralisation de la classification où chaque sortie possible est une classe. Le problème vient du nombre de classes possibles, qui pour la majorité des données structurées grandit exponentiellement avec la taille des données considérées. Prenons par exemple, une tâche de reconnaissance de mots écrits, où toutes les combinaisons de lettres de l'alphabet seraient autorisées. Le nombre de mots possibles à n lettres est dans ce cas 26^n. Cette augmentation exponentielle du nombre de sorties possibles rend les méthodes de classification usuelles inutilisables.

La majorité des approches existantes pour la prédiction structurée repose sur l'idée d'apprendre une fonction de compatibilité entre entrées et sorties. Muni d'une telle fonction, la prédiction se fait en cherchant la sortie qui est la plus compatible avec l'entrée donnée. Cela correspond à un problème de recherche combinatoire dans l'espace des sorties possibles : pour identifier la bonne sortie, il faut calculer la compatibilité de l'entrée avec toutes les sorties possibles. Dans le cas de structures de données simples, il existe des algorithmes de programmation dynamique qui sont efficaces pour résoudre ce problème. Néanmoins leur application implique des restrictions fortes sur le type de problèmes qui sont abordables.

Au lieu d'apprendre la compatibilité entre entrées et sorties, puis de chercher la meilleure sortie possible pour une entrée donnée, il existe une approche beaucoup plus directe : apprendre à construire une bonne sortie immédiatement. Dans cette approche, il ne s'agit plus d'apprendre une fonction de compatibilité, mais d'« apprendre un programme » capable de construire des sorties.



Ma thèse a porté sur l'**apprentissage de programmes capables de résoudre des problèmes de prédiction structurée**. Ces travaux m'ont amené à développer un nouveau cadre de programmation mélangeant programmation traditionnelle et apprentissage automatique : les CR-algorithmes (Choose/Reward Algorithms). Ce cadre permet d'écrire des squelettes de programmes et de remplir les

[14]Francis MAES, `http://nieme.lip6.fr/Research/`.

Détection d'évènements visuels significatifs — Nicolas BURRUS[5], CSI 2004, Docteur 2008

En vision par ordinateur, la détection consiste à rechercher automatiquement dans une image si des entités particulières sont présentes. Les applications sont nombreuses et variées, par exemple dans un contexte de vidéo-surveillance pour détecter automatiquement si des personnes sont présentes dans le champ de la caméra, dans un contexte robotique pour faire interagir un robot avec les objets de son environnement, dans un contexte militaire pour détecter des tanks ou des cibles à partir d'images aériennes, ou encore dans le domaine grand public avec la détection de visages qui est maintenant intégrée à la plupart des appareils photos numériques.

Au cœur du problème de détection se cache un problème de décision : étant donnée une zone de l'image candidate (un groupe de pixels), il s'agit de décider si elle est associée ou non à un objet à détecter. L'apparence des objets dans l'image est généralement très variable, pour plusieurs raisons. Tout d'abord, des conditions d'acquisition comme l'illumination ou le point de vue peuvent changer. Les objets peuvent également avoir une variabilité intrinsèque. Par exemple, un visage change d'apparence d'une personne à l'autre et certaines personnes peuvent avoir des lunettes. Enfin, une image n'est que la projection d'une scène 3D sur un support 2D, ce qui entraîne une perte d'information, qui sera de plus perturbée par diverses sources de bruit.

Toutes ces variations rendent la prise de décision complexe et fondamentalement incertaine. Cela pousse à une modélisation statistique de l'apparence des objets à détecter dans l'image. On cherche alors à calculer la probabilité qu'un objet soit présent dans la zone candidate. L'objet sera détecté si cette probabilité est suffisamment forte.

Le calcul de cette probabilité reste cependant délicat et nécessite souvent d'injecter beaucoup d'*a priori*. Ce dernier peut se matérialiser par des paramètres numériques de l'algorithme que l'utilisateur doit ajuster à la main. De façon plus élégante, le calcul de cette probabilité peut être appris automatiquement via les techniques classiques d'intelligence artificielle (réseaux de neurones, etc.) à partir d'exemples de groupes de pixels pour lesquels la vérité est connue.

Toujours en raison de la grande variabilité des entrées, il reste difficile dans les deux cas d'obtenir des algorithmes robustes et applicables à une grande classe d'images. Dans le cas empirique l'utilisateur ne pourra en effet tester que quelques images ; pour l'apprentissage automatique il est compliqué de collecter une base d'exemples suffisamment grande et représentative.

Une autre approche a été proposée par le CMLA (Centre de Mathématique et de Leurs Applications) de l'ENS Cachan. Au lieu de modéliser directement l'apparence des objets à détecter, l'idée est de raisonner *a contrario* en recherchant des zones de l'image dont certaines propriétés ne peuvent *pas* être — statistiquement — le résultat du « hasard ». Les propriétés sont choisies judicieusement pour que les zones détectées correspondent aux objets recherchés. Seul un modèle statistique de ce que le hasard peut générer est alors requis. Il est souvent beaucoup plus simple à définir, et des algorithmes relativement universels et sans paramètres peuvent être obtenus.

Dans mon travail de thèse, j'ai cherché à étendre cette méthodologie à des applications qui sortaient du cadre analytique initialement proposé, en ayant recours à de l'apprentissage. En particulier, un algorithme de détection d'objets à partir d'une base de photos a été proposé. Il apprend à partir de quelques images naturelles ne contenant pas les objets de la base (faciles à collecter !) quelles similarités peuvent apparaître accidentellement avec les photos de la base. Les objets sont alors détectés *a contrario* lorsque les similarités sont trop fortes pour être accidentelles.

La thèse, financée par la Délégation Générale pour l'Armement (DGA)[6], a été effectuée au sein du Laboratoire Electronique et Informatique (UEI)[7] de l'ENSTA[8] et du Laboratoire d'InfoRmatique en Image et Systèmes d'information (LIRIS)[9] de l'INSA Lyon[10]. Elle a été soutenue le 11 Décembre 2008 et je suis maintenant en Post-doc à l'Université de Liège.

[5]Nicolas BURRUS, `http://nicolas.burrus.name/research`.
[6]Délégation Générale pour l'Armement (DGA), `http://www.defense.gouv.fr/dga`.
[7]Laboratoire Electronique et Informatique (UEI), `http://uei.ensta.fr`.
[8]ENSTA, `http://www.ensta.fr`.
[9]Laboratoire d'InfoRmatique en Image et Systèmes d'information (LIRIS), `http://liris.cnrs.fr`.
[10]INSA Lyon, `http://www.insa-lyon.fr`.

Pressions sélectives multiples pour l'évolution de réseaux de neurones destinés à la robotique — Jean-Baptiste MOURET[11], CSI 2004, Docteur 2008

La famille des *algorithmes évolutionnistes* regroupe les méta-heuristiques d'optimisation globales inspirées par la théorie de l'évolution darwinienne. L'impressionnante capacité de la sélection naturelle à adapter des organismes à leur environnement a inspiré de nombreux travaux visant à concevoir automatiquement des structures (graphes, réseaux de neurones, plans de robots, programmes…) répondant à un objectif décrit sous la forme d'une fonction de performance, appelée *fonction de fitness*. Néanmoins, ces méthodes peinent à obtenir des résultats dès que l'on aborde des problèmes complexes, notamment lorsque la réussite d'une étape intermédiaire est requise mais n'apporte pas de gain de fitness.

En paléontologie, l'explication de l'apparition de nombreuses caractéristiques complexes des organismes vivants s'appuie sur l'*exaptation*, c'est-à-dire l'utilisation pour une fonction nouvelle d'un trait adapté initialement à une autre fonction. Pour observer une exaptation, il est nécessaire qu'il existe plusieurs fonctions différentes à réaliser, chacune avec une valeur adaptative. De manière abstraite, on peut considérer que l'évolution s'est ainsi servie d'un gradient de sélection pour une sous-partie de l'organisme avant d'utiliser cette sous-partie pour réaliser une fonction plus complexe. Afin de répondre aux difficultés actuelles des algorithmes évolutionnistes, notamment pour la synthèse de réseaux de neurones pour piloter des robots, nous avons proposé d'importer le concept d'exaptation et de le faciliter ajoutant un ou des objectifs, même si seul l'un d'eux nous intéresse. Pour cela, nous avons envisagé l'utilisation d'algorithmes évolutionnistes multiobjectifs basés sur la dominance de Pareto, des méthodes d'optimisation qui permettent de trouver un ensemble de compromis Pareto-optimaux entre objectifs et non pas une unique solution « optimale ».

Il est souvent possible d'émettre des hypothèses sur des étapes intermédiaires pouvant aider à la résolution de problèmes trop complexes pour être abordés avec les méthodes actuelles. Nous avons montré qu'on peut alors les exploiter en définissant un objectif pour chaque étape puis en utilisant un algorithme multiobjectif. Ainsi, les solutions pour les objectifs les plus simples sont « exaptées » pour réaliser les fonctions plus complexes. On obtient alors **un processus évolutionniste guidé par des étapes intermédiaires mais capable d'ignorer les étapes inutiles** et de découvrir automatiquement leur niveau de difficulté. Cette approche a été testée avec succès sur un problème de robotique évolutionniste simulé mettant en jeu de nombreuses sous-tâches et en laissant le processus déterminer leurs dépendances mutuelles.

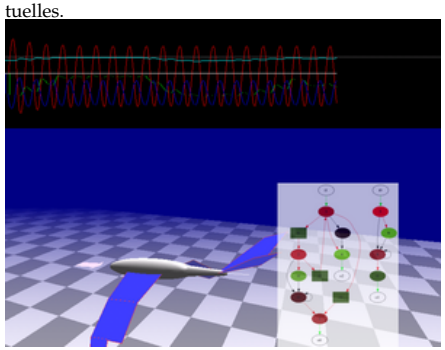


FIG. 1 – Extrait d'une des vidéos sont disponibles en ligne montrant l'apprentissage du vol par des oiseaux artificiels.

En l'absence de pression de sélection, une autre possibilité est d'ajouter un objectif poussant à une exploration efficace du voisinage des solutions déjà obtenues. La comparaison de différentes méthodes pour appliquer ce concept à l'évolution de réseaux de neurones montre l'intérêt de maintenir la diversité parmi les comportements des solutions, et non leurs génotypes. De bons résultats ont notamment été obtenus pour créer des réseaux approchant une fonction logique avec une fitness trompeuse et pour résoudre le problème de robotique évolutionniste abordé précédemment

La transposition du concept d'exaptation de sous-structures à l'évolution de réseaux de neurones revient à appliquer une pression de sélection sur des modules du réseau, éventuellement reliés à des modules génotypiques. Après avoir synthétisé avec cette méthode des réseaux de neurones approchant une fonction logique, nous avons analysé les performances de différentes variantes de cette idée avec la valeur de Shapley (une approche issue de la théorie des jeux pour allouer équitablement des gains entre les participants à un jeu coalitionnel). Les résultats obtenus montrent que les éléments clefs des performances sont ceux établissant le lien entre les modules du génotype, du phénotype et les objectifs.

Ces résultats encourageants[12] peuvent être appliqués à un large spectre de problèmes de robotique évolutionniste, du pilotage de robots à l'obtention de contrôleurs rythmiques.

[11] Jean-Baptiste MOURET, `http://animatlab.lip6.fr/MouretAccueilEn`.

Algorithmes distribués pour la sécurité et la qualité de service dans les réseaux *ad hoc* mobiles — Ignacy GAWEDZKI[13], CSI 2003, Docteur 2008

La question du routage dans les réseaux *ad hoc* mobiles est un domaine de recherche très actif ces dernières années. Les protocoles arrivant aujourd'hui à maturité ont été mis au point dans l'hypothèse qu'aucun des dispositifs prenant part au fonctionnement du réseau n'est malveillant. Pourtant il existe de nombreuses applications où une telle hypothèse n'est pas vérifiée et de nombreux moyens existent pour saper le bon fonctionnement du réseau.

Dans cette thèse je me suis intéressé à la mise en place d'une solution ayant pour but de rendre les protocoles de routage proactifs résistants à l'action d'entités malveillantes. Il s'agit d'enrichir le protocole de routage afin de permettre aux nœuds d'effectuer de façon distribuée une surveillance du réseau et de fournir une mesure de la menace représentée par chacun des autres nœuds. De ces mesures est ensuite extraite une métrique de qualité de service qui est prise en compte par le protocole de routage. L'effet recherché par l'incorporation de cette métrique est de pouvoir contourner les nœuds qui sont les plus suspects d'après les différentes méthodes de détection utilisées.

Nous proposons en particulier de détecter la perte, intentionnelle ou non, de paquets de données, qui demeure une façon très simple de nuire au bon fonctionnement du réseau, qu'elle soit motivée par l'égoïsme ou non. La détection est réalisée par une vérification distribuée du principe de conservation de flot, basée sur l'échange de compteurs de paquets entre les nœuds voisins. Nous proposons également une méthode de diffusion de ces valeurs qui permet un meilleur passage à l'échelle. Les résultats de cette détection ne pouvant pas servir directement pour incriminer un nœud, ils ne sont utilisés que pour maintenir un degré de suspicion local envers les autres. Les degrés de suspicion locaux sont ensuite diffusés dans le réseau et combinés entre eux pour former un degré de suspicion global envers chaque nœud, qui sert de métrique de qualité de service.

L'application de la solution au protocole OLSR est décrite et ses performances sont évaluées par simulation. Nous observons notamment que la solution permet de diminuer l'impact des nœuds malveillants de façon notable et que malgré le surcoût de contrôle considérable par rapport à l'utilisation d'OLSR classique, la part supplémentaire de capacité du médium utilisée reste faible.

Nous présentons enfin la plateforme de mise en œuvre expérimentale du routage OLSR avec qualité de service et sécurité. Notre objectif est de pouvoir à la fois faire fonctionner nos solutions dans des mises en place réelles et de pouvoir rapidement déceler les problèmes associés à l'utilisation de matériel courant disponible pour le grand public.

Où trouver les manuscrits de thèse de nos anciens ?

guillaume_pitel.2004.phd.pdf	clement_faure.2007.phd.pdf	yann_regis-gianas.2007.phd.pdf
ignacy_gawedzki.2008.phd.pdf	jean-baptiste_mouret.2008.phd.pdf	nicolas_burrus.2008.phd.pdf
pierre-yves_strub.2008.phd.pdf	francis_maes.2009.phd.pdf	...

[12] Résultats encourageants, `http://animatlab.lip6.fr/MouretVideosEn`.

[13] Ignacy GAWEDZKI, `http://www.lri.fr/~ig/`.