
GACS : une approche ascendante pour la coordination spatiale

Fabien Flacher^{*,} — Olivier Sigaud^{*}**

^{*} AnimatLab-LIP6,
8 rue du capitaine Scott
75015 Paris
fab_hid@yahoo.com

^{**} Dassault Aviation, DGT/DPR/ESA
78 Quai Marcel Dassault
92552 St-Cloud Cedex
Olivier.Sigaud@lip6.fr

RÉSUMÉ. La mise au point de mécanismes de coordination spatiale pour des agents évoluant dans des univers continus et dynamiques est un problème difficile. Alors que la démarche descendante ne parvient pas à appliquer sa méthode de décomposition de façon satisfaisante sur cette classe de problèmes, l'approche ascendante obtient des résultats plus convaincants, mais elle implique souvent de fastidieux réglages manuels qui posent des problèmes de passage à l'échelle. Notre démarche pour traiter cette difficulté consiste à adjoindre à un formalisme de coordination spatiale ascendante un algorithme évolutionniste multicritère dédié à ce type de problèmes. Nous montrons sur un problème de coordination spatiale traité précédemment par Balch et Hybinette que les solutions obtenues avec notre plate-forme, GACS, sont comparables à celles obtenues par ces auteurs, malgré un investissement moindre de la part du concepteur. De plus, les solutions compétitives obtenues avec GACS sont plus simples que celles proposées par Balch et Hybinette, ce qui nous permet de conclure à la supériorité de notre approche.

ABSTRACT. The design of spatial coordination mechanisms for dynamical and continuous multi-agent setting is a difficult challenge. While the top-down decomposition approach is inefficient on such problems, the bottom-up approach is more promising, but requires a tedious manual parameter tuning which raises scaling-up issues. Our own approach consists in replacing the manual tuning by a specially designed multicriteria evolutionary algorithm devoted to the tuning of our spatial coordination formalism. In this paper, through a quantitative comparison on a complex spatial coordination problem treated previously by Balch and Hybinette, we show that our system, GACS, finds a population of solutions as efficient as this predecessor though our approach requires less involvement from the designers and can find simpler solutions.

MOTS-CLÉS : coordination spatiale, algorithmes évolutionnistes multicritères, adaptation.

KEYWORDS: spatial coordination, multicriteria evolutionary algorithm, adaptation.

1. Introduction

La conception de simulations réalistes de grande envergure suscite un intérêt croissant dans l'industrie mais reste un défi pour la recherche en informatique. Les nombreuses difficultés techniques et scientifiques proviennent en partie du fait que ces « mondes virtuels » sont souvent peuplés par des groupes plus ou moins autonomes d'agents ou entités, créatures artificielles ou véhicules simulés, dont le comportement doit être convaincant. Ainsi, indépendamment des applications, qu'il s'agisse d'animation dans les films, de jeux vidéo, de simulations militaires ou de robotique collective, la qualité de ces simulations dépend notamment de celle des comportements de groupe exhibés par les agents.

Le problème minimal auquel le groupe d'agents est confronté est celui de la coordination spatiale. En effet, pour exhiber un comportement de groupe et satisfaire des objectifs collectifs, chaque agent, à chaque instant, doit au moins être capable de se positionner et s'orienter correctement en fonction de la position et du déplacement de tous les autres agents. Cela suppose qu'il prenne en compte ses buts, ses contraintes et les buts et contraintes de tous les autres membres du groupe. Il en résulte une forte interdépendance entre les déplacements individuels. Ces contraintes participent largement à la complexité de la réalisation des mécanismes de coordination spatiale. La multiplicité des agents ne permet pas d'utiliser les algorithmes de planification classique (Latombe, 1991) qui exigeraient une connaissance exhaustive de la situation de chacun des agents. La difficulté provient de ce que l'interdépendance des déplacements rend laborieuse et délicate la prédiction de toutes les configurations par lesquelles peut passer un groupe d'agents. De plus, la complexité de cette interdépendance augmente avec le nombre d'agents impliqués. Ce phénomène pose également un problème de robustesse au changement d'échelle à la majorité de ces techniques, étant donné la multitude des cas pour lesquels il faut concevoir une solution.

Face à ces défis, des chercheurs ont tenté d'appliquer des approches inspirées de l'intelligence artificielle située (Drogoul *et al.*, 1999) et de l'approche animat (Meyer, 1998). Celles-ci reposent sur une démarche ascendante composée de deux mécanismes : la génération d'une variété de solutions potentielles et l'auto-organisation ou la sélection automatique d'une solution conforme aux spécifications du problème.

Des résultats importants dans ce domaine proviennent des travaux de Reynolds (Reynolds, 1987). Il définit pour chaque agent une combinaison de vecteurs de déplacement construits à partir de perceptions locales. Ces vecteurs permettent par exemple d'éviter la collision entre agents ou de s'aligner sur la vitesse moyenne et le cap moyen de ses voisins. Avec les « *boids* »¹, il a démontré la faisabilité de la génération de comportements coordonnés à partir de l'auto-organisation d'un petit ensemble de modules comportementaux. Mais l'ensemble des déplacements possibles est pré-défini manuellement, ce qui réduit considérablement la variété des comportements collectifs potentiellement réalisables. La robotique collective a également produit des

1. Contraction de *bird-oids*.

résultats comparables, notamment ceux de Matarić (Matarić, 1994a; Matarić, 1995) qui a montré l'existence d'un ensemble de comportements élémentaires (exploration prudente, agrégation, dispersion et retour au nid) qui, correctement combinés, suffisent à résoudre des problèmes de coordination spatiale robotique complexes comme l'encerclement ou le déplacement en troupeau. Cependant, comme dans le cas de Reynolds, ces comportements élémentaires sont prédéfinis manuellement. Arkin et son équipe ont aussi examiné la conception des mécanismes de coordination spatiale (Arkin, 1989; Arkin, 1992). Ils ont proposé une théorie qui décompose un comportement global en *schémas*, sortes d'atomes comportementaux, exprimés sous formes de champs vectoriels attractifs ou répulsifs. Les schémas sont combinés en sommant des champs attractifs pour les objectifs et répulsifs pour les obstacles. Balch (Balch *et al.*, 1998; Balch *et al.*, 2000) a montré que la génération et le maintien de formation peuvent être synthétisés en adaptant cette théorie. Mais, malgré ses innovations, son approche souffre des mêmes inconvénients que ceux de Reynolds ou Matarić, à savoir la nécessité de prédéfinir à la main un ensemble de schémas élémentaires.

Ces techniques peuvent être étendues. De fait, elles reposent toutes sur une méthode générale décomposable en deux parties : la génération d'une variété de contrôleurs potentiels de coordination spatiale et leur ajustement vers un contrôleur solution, correspondant aux spécifications du problème. Plutôt que de réaliser la seconde partie manuellement, on peut les étendre en utilisant des procédés de mise au point automatique tels que les algorithmes évolutionnistes. Des travaux comme (Clark *et al.*, 1992; Reynolds, 1992; Pearce *et al.*, 1992; Matarić, 1994b; Ram *et al.*, 1994; Reynolds, 1994) se sont déjà intéressés à la mise au point automatique du déplacement d'un agent unique par des méthodes évolutionnistes, mais ils ne traitent pas des problèmes de coordination spatiale.

À notre connaissance, (Reynolds, 1992) est la seule référence dans laquelle un algorithme évolutionniste est utilisé pour résoudre un problème de coordination spatiale, mais le mécanisme de coordination lui-même n'est pas discuté.

Dans cet article, nous présentons un mécanisme de coordination spatiale fondé sur un formalisme de représentation et un algorithme évolutionniste multicritère spécialement conçu pour résoudre le problème de la coordination spatiale. Le point de départ de ce travail est le modèle de coordination spatiale proposé par Balch et Arkin, généralisé pour obtenir un ensemble de contrôleurs potentiels plus vaste, tout en maîtrisant la durée d'exploration de cet espace de recherche nettement plus important.

Dans (Flacher *et al.*, 2004), nous avons déjà comparé un contrôleur publié par Balch et Hybinette (Balch *et al.*, 2000) avec deux contrôleurs obtenus par évolution : l'un reprenait le formalisme de Balch et Hybinette mais optimisait ses paramètres par évolution génétique tandis que l'autre était engendré par notre plate-forme, GACS². Nous avons montré que le contrôleur engendré par GACS était sensiblement plus efficace et plus simple que son concurrent. Ici, nous nous livrons à une autre comparaison, plus globale, entre une population engendrée automatiquement avec le formalisme de

2. Pour « Génération Ascendante de la Coordination Spatiale ».

Balch et une population engendrée avec le nôtre. Cette comparaison permet de mettre en évidence les atouts intrinsèques de notre formalisme et des opérateurs génétiques spécifiques qui lui sont appliqués. Nous montrons que la recherche d'une solution s'effectue dans un espace beaucoup plus vaste mais, grâce à une efficacité supérieure, elle débouche sur des solutions dont la performance est comparable, alors que leur expression peut être nettement plus simple.

Cet article est organisé comme suit : dans la section 2 nous décrivons notre modèle généralisé de coordination spatiale. La section 3 est consacrée à la présentation de l'expérience et du contrôleur original de Balch. La section 4 présente les résultats obtenus. Ces résultats sont discutés dans la section 5. Niveau de performance égale. Nous terminons l'article par une conclusion qui met en valeur l'intérêt de notre approche.

2. Génération d'agents coordonnés spatialement

2.1. Présentation générale

Pour traiter des problèmes de coordination spatiale dans des univers continus et dynamiques, nous avons choisi une approche ascendante fondée sur la combinaison de forces d'attraction et de répulsion entre tous les agents et les objets pertinents dans leur environnement. Comme dans toute approche ascendante, ces forces sont définies à un niveau élémentaire puis modifiées localement jusqu'à atteindre le comportement global recherché. Néanmoins, au lieu de nécessiter une fastidieuse modification manuelle des forces par un processus d'essais et erreurs, le système GACS fait appel à un algorithme génétique pour trouver automatiquement des forces qui permettent aux agents d'atteindre un objectif de coordination qui s'exprime comme l'optimisation d'un ensemble de critères numériques de performance. L'originalité de ce travail se manifeste dans trois domaines complémentaires.

1) Le formalisme employé pour exprimer numériquement les forces d'attraction et de répulsion est proche de celui employé dans le cadre de la théorie des schémas de Arkin et Balch (Balch *et al.*, 1998), mais il est plus général. Offrant davantage de degrés de liberté à un algorithme d'optimisation, il permet d'obtenir des solutions plus variées. Nous allons présenter ce formalisme dans la section suivante.

2) Pour sélectionner un contrôleur de coordination spatiale adapté à un problème donné au sein de l'espace des contrôleurs possibles, l'approche repose sur un algorithme génétique, ce qui suppose d'exprimer numériquement le fait qu'une solution satisfait mieux le problème qu'une autre. Pour exprimer numériquement des objectifs de coordination qui peuvent être complexes, nous faisons appel à une approche multicritère, ce qui nous permet de sélectionner des contrôleurs optimaux vis-à-vis d'un ensemble de critères donnés. Nous aborderons ce point à la section 2.3.

3) Le choix d'exprimer nos contrôleurs sous la forme d'une combinaison linéaire de forces élémentaires d'attraction et de répulsion induit la définition d'une structure spécifique des génomes. Les opérateurs de recombinaison génétique tirent parti de

cette structure pour réaliser une exploration plus efficace de l'espace de recherche engendré par les contrôleurs, ce que nous exposerons à la section 2.3.2.

2.2. Un formalisme pour la coordination spatiale

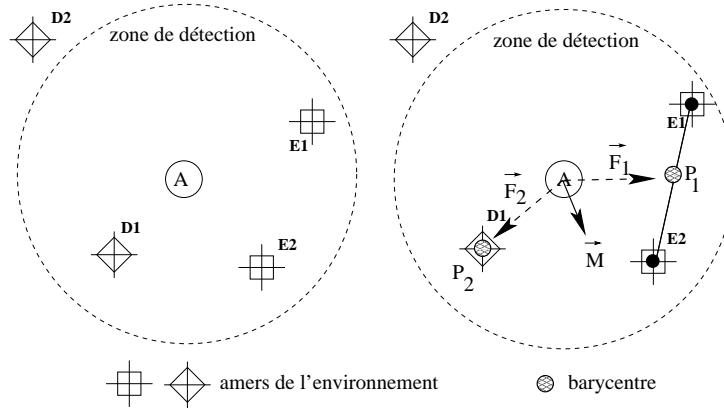


Figure 1. *Prise en compte des objets pertinents. L'agent détecte trois objets (E_1 , E_2 et D_1). D_2 n'est pas perçu car il se situe en dehors de la zone de détection. Les points d'intérêt P_1 et P_2 sont construits comme des barycentres des objets, respectivement entre E_1 et E_2 , de type « carré », pour P_1 et sur D_1 seul, de type « losange », pour P_2*

Comme indiqué en introduction, la théorie des schémas d'Arkin (Arkin, 1989) permet de spécifier un comportement sous la forme d'une combinaison de schémas élémentaires dont les contributions s'ajoutent pour définir à tout instant le déplacement de l'agent. Chaque schéma élémentaire définit une fonction d'attraction ou de répulsion associée directement à une entité pertinente de l'environnement. Cette fonction, calculée en fonction de la distance qui sépare l'agent de l'entité considérée, est définie a priori et linéaire par morceaux.

Notre propre formalisme reprend chacun de ces éléments, mais lui ajoute d'une part la construction de « points d'intérêt » qui résultent de l'association de plusieurs entités pertinentes et, d'autre part, la possibilité de régler l'ensemble des paramètres qui interviennent dans le calcul du déplacement à l'aide d'algorithmes évolutionnistes.

Formellement, le déplacement \vec{D}_i d'un agent A_i est donc modélisé par une somme de δ forces \vec{F}_k . Chacune de ces forces est un vecteur prenant l'agent A_i pour origine et construit à partir d'une direction et d'une intensité comme indiqué sur la figure 2.

La somme vectorielle de ces forces, modulées par un gain, permet de décrire le déplacement que l'agent doit s'efforcer de réaliser. Plutôt que de s'exercer directement

à partir des entités pertinentes, elles s'exercent à partir de barycentres d'un ensemble d'entités pertinentes qui se situent dans la zone de détection de l'agent (voir figure 1). Nous appelons « points d'intérêt » d'un agent les barycentres auxquels sont associées des forces dans le contrôleur de cet agent.

Cet élément de notre formalisme permet d'exprimer simplement le fait que le déplacement d'un agent doit parfois être relatif à un ensemble d'objets considérés simultanément plutôt que dirigé vers chacun des objets considérés séparément. Par exemple, un joueur de football qui cherche à marquer un but doit se diriger vers un point situé derrière le ballon par rapport à la cage adverse. Ce point, « derrière le ballon par rapport à la cage », s'exprime très simplement comme un barycentre associé à la cage et au ballon auxquels sont affectés des coefficients bien choisis.

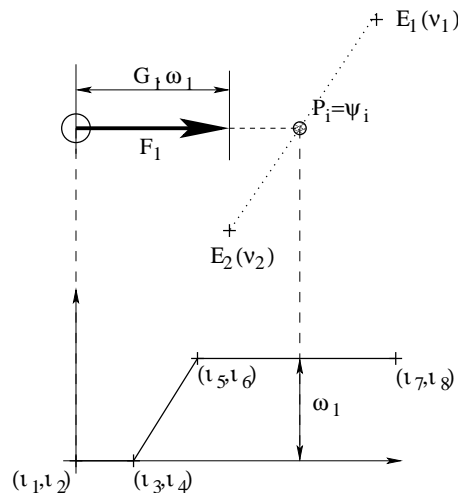


Figure 2. Schéma explicatif de la construction de la force F_1 de la figure 1. La partie haute de la figure représente l'agent en relation avec le point d'intérêt P_1 de la figure 1. La partie basse représente la fonction d'attraction/répulsion associée à ce point d'intérêt. La fonction ω_1 , paramétrée par $t_{1,1}, \dots, t_{8,1}$ renvoie l'intensité partielle de F_1 , qui est ensuite modulée par le gain G_1

Jusqu'ici, nous avons expliqué comment les divers objets et agents pertinents qui entourent un agent sont pris en compte dans le formalisme pour construire un contrôleur sensible à la position relative de ces objets et agents. Il nous reste à voir comment le formalisme exprime les forces d'attraction et de répulsion associées aux divers points d'intérêts et comment il gère la présence de η agents.

On commence par déterminer parmi tous les objets et agents pertinents au sein de l'environnement de l'agent ceux qui sont perçus par l'agent. On les ordonne par type et par ordre de proximité décroissante à l'agent. On définit ensuite pour chaque force

\vec{F}_k une fonction ψ_k , paramétrée par α réels, qui prend en entrée les positions relatives de α objets ou agents pertinents, notés O_j , et qui rend le point d'intérêt P_k :

$$\psi_k^{\nu_{1k}, \dots, \nu_{\alpha k}}(O_1, \dots, O_\alpha) = \vec{OP}_k = \sum_{j=1}^{\alpha} \nu_{jk} \cdot \vec{OO}_{jk}, \text{ avec } \forall k, \sum_{j=0}^{\alpha} \nu_{jk} = 1 \quad [1]$$

La fonction ψ_k est donc la fonction qui construit le point d'intérêt P_k en fonction des objets et agents pertinents perçus par l'agent.

L'intensité d'une force individuelle \vec{F}_k est obtenue en multipliant une fonction normalisée ω_k par un gain G_k . La fonction ω_k représente l'attraction (si elle est positive) ou la répulsion (si elle est négative) que le point d'intérêt P_k fait subir à l'agent. Les gains constants G_k permettent d'exprimer le fait que certains points d'intérêt ont une influence prépondérante sur le comportement par rapport à d'autres.

En pratique, la fonction ω_k est définie comme une fonction linéaire par morceaux, paramétrée par β réels, prenant en entrée la distance $\| \vec{AP}_k \|$ qui sépare l'agent du point d'intérêt P_k . Définir ω_k comme une fonction linéaire par morceaux revient à définir linéairement la valeur de ω_k sur β intervalles successifs :

$$\omega_k^{\iota_{1,k}, \dots, \iota_{\beta,k}} = \begin{cases} \frac{\iota_{4,k} - \iota_{2,k}}{\iota_{3,k} - \iota_{1,k}}(x - \iota_{1,k}) + \iota_{2,k}, & x \in [\iota_{1,k}, \iota_{3,k}] \\ \vdots \\ \frac{\iota_{i+3,k} - \iota_{i+1,k}}{\iota_{i+2,k} - \iota_{i,k}}(x - \iota_{i,k}) + \iota_{i+1,k}, & x \in [\iota_{i,k}, \iota_{i+2,k}] \\ \vdots \\ \frac{\iota_{\beta,k} - \iota_{\beta-2,k}}{\iota_{\beta-1,k} - \iota_{\beta-3,k}}(x - \iota_{\beta-3,k}) + \iota_{\beta-2,k}, & x \in [\iota_{\beta-3,k}, \iota_{\beta-1,k}] \end{cases} \quad [2]$$

Le déplacement \vec{D}_i de l'agent A_i résulte d'une combinaison linéaire de forces. Il est donc donné par l'équation suivante :

$$\vec{D}_i = \sum_{k=1}^{\delta} G_k \times \omega_k(\| \vec{A}_i P_k \|) \times \left(\sum_{j=1}^{\alpha} \nu_{jk} \cdot \vec{A}_i O_{jk} \right) \quad [3]$$

Finalement, selon notre formalisme, un mécanisme de coordination spatiale est construit globalement pour η agents à partir de η fonctions de déplacement \vec{D}_i , $i \in [1, \eta]$, où η est le nombre d'agents. Nous ne traitons que le cas où tous les agents sont identiques et partagent le même contrôleur afin de limiter le nombre de paramètres à faire évoluer. Il construit donc une unique fonction \vec{D}_1 qui vaut pour tous les agents.

Dans ce que nous avons présenté jusqu'ici, la principale originalité de notre formalisme vis-à-vis de la théorie des schémas est l'introduction des barycentres qui permet

de relâcher le couplage étroit entre un schéma et une entité pertinente. Dans la mesure où l'on peut toujours associer un unique point d'intérêt à chaque entité pertinente, notre formalisme apparaît plus général que la théorie des schémas.

Nous allons voir à présent que, ayant fait apparaître explicitement l'ensemble des paramètres qui interviennent dans le calcul du déplacement, nous pouvons régler simultanément tous ces paramètres à l'aide de méthodes évolutionnistes, ce qui nous permet d'obtenir des solutions beaucoup plus variées que dans le cadre de la théorie des schémas, où l'on se contente de faire varier quelques paramètres bien choisis.

Nous verrons enfin que l'adjonction des points d'intérêt est un facteur crucial du succès du recours massif à des algorithmes évolutionnistes, car elle permet aux algorithmes de découvrir des « bloc de construction ³ » (Goldberg, 1989) qui jouent un rôle déterminant dans son efficacité.

2.3. Sélection automatique d'un contrôleur performant

Les objectifs associés à différents problèmes de coordination spatiale peuvent être plus ou moins complexes et faire appel à des contraintes de différents types. Ayant fait le choix de régler les paramètres de nos contrôleurs à l'aide d'un algorithme évolutionniste, nous sommes contraints de mesurer la satisfaction d'un problème de coordination spatiale donné sous la forme de critères numériques. Il nous est rapidement apparu que représenter différents objectifs complexes à l'aide d'un critère unique n'était pas satisfaisant sur le plan méthodologique, car cela imposait un travail de réglage de la fonction d'évaluation correspondante au moins aussi difficile que de régler directement des contrôleurs à la main. Nous avons donc choisi une approche faisant appel à plusieurs critères à la fois, ce qui nous a naturellement conduit à nous tourner vers le domaine des algorithmes évolutionnistes multicritères, qui sont en plein essor depuis quelques années (Coello, 2003).

2.3.1. Méthodes évolutionnistes multicritères

Alors que dans un problème d'optimisation simple, toutes les solutions peuvent être ordonnées en fonction de la valeur qu'elles atteignent pour le critère qu'elles optimisent, le trait caractéristique des problèmes multicritères est que l'espace des solutions n'est que partiellement ordonné, c'est-à-dire que plusieurs solutions peuvent être équivalentes au regard des critères qu'elles optimisent. Par exemple, si l'on cherche à maximiser deux critères c_1 et c_2 , une première solution peut être maximale sur c_1 et nulle sur c_2 . Inversement, une autre solution peut être maximale sur c_2 et nulle sur c_1 . Aucune des deux solutions n'est meilleure que l'autre.

Il arrive cependant qu'une solution soit moins bonne qu'une autre sur chacun des critères qui caractérisent le problème. On dit alors que la première solution est dominée par la seconde. On appelle « solutions non dominées » l'ensemble des solutions

3. *Building block.*

qui ne sont dominées par aucune autre solution. Pour un problème d'optimisation multicritère donné, toutes les solutions non dominées forment un ensemble de compromis optimaux, nommé « front de Pareto » (Pareto, 1896).

Les algorithmes évolutionnistes multicritères partent d'une population de solutions quelconques et cherchent, en adaptant les mécanismes classiques des méthodes évolutionnistes, à atteindre une population d'individus qui appartiennent tous au front de Pareto du problème posé. La plupart des techniques cherchent en outre à faire en sorte que la population finale soit répartie le plus uniformément possible sur le front de Pareto afin d'en donner la meilleure approximation possible. Cette seconde contrainte n'est pas toujours pertinente pour nos applications complexes.

On distingue classiquement deux générations d'algorithmes, que nous n'allons pas passer en revue ici ⁴. La seconde génération d'algorithmes se distingue de la première par des algorithmes sensiblement plus performants, mais aussi plus complexes et plus préoccupés d'obtenir une meilleure couverture du front de Pareto sur un certain nombre de problèmes jouets choisis pour les difficultés spécifiques qu'ils posent.

Pour notre part, nous utilisons un algorithme génétique librement inspiré de l'algorithme NPGA de J. Horn (Horn *et al.*, 1994), qui appartient à la première génération. Ce choix est justifié par l'efficacité satisfaisante de cet algorithme, par sa capacité à produire des solutions variées et surtout par sa simplicité de mise en œuvre (Deb, 2001).

Comme l'algorithme NPGA, notre algorithme de sélection repose sur la détermination de strates de solutions non dominées. La première de ces strates est l'ensemble des solutions non dominées (le front de Pareto). La seconde strate est l'ensemble des solutions non dominées lorsqu'on a retiré la première strate, etc.

Dans les algorithmes évolutionnistes les plus standards, l'évolution de la population se fait génération après génération en conservant un certain pourcentage des meilleurs individus de la génération précédente dans la génération suivante, puis en recombinaison entre eux ces individus pour créer de nouveaux individus qui viennent compléter la nouvelle génération. Dans notre algorithme, de la même façon, on attribue à chaque individu un pourcentage de chance d'être conservé d'une génération à la suivante en fonction de la strate à laquelle il appartient – ce pourcentage étant d'autant plus élevé qu'il appartient à une strate plus élevée de la population – puis on remplace les individus qui n'ont pas été conservés par des descendants de ceux qui l'ont été.

Un tel mécanisme n'inclut aucune pression vers une diversité maximale entre les individus. Toutefois, nous allons voir qu'il donne des résultats satisfaisants. Avant cela, il nous reste à présenter les spécificités du codage du génome des individus et des opérateurs de recombinaison que nous leur appliquons.

4. Nous renvoyons le lecteur à (Coello, 2003) et (Deb, 2004) pour approfondir cette question.

2.3.2. *Codage*

Comme nous l’avons indiqué précédemment, le nombre d’agents impliqués dans les problèmes que nous abordons pouvant être grand, plutôt que de rassembler le codage d’une fonction de déplacement pour chaque agent dans un même génome, nous faisons le choix de ne coder qu’une fonction de déplacement unique dans le génome et d’attribuer cette fonction identiquement à tous les agents. Une alternative pourrait consister à faire évoluer des fonctions de déplacements distincts pour les différents agents, mais se poseraient alors des problèmes de co-évolution réputés difficiles.

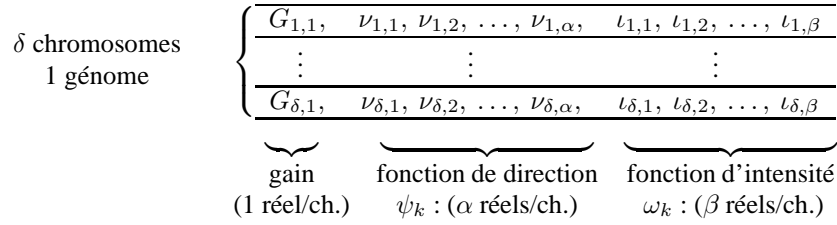


Figure 3. Description de la structure d’un génome généré par GACS

Le génome code une fonction de déplacement pour un agent. Comme nous l’avons vu précédemment, cette fonction de déplacement se décompose sous la forme d’un ensemble de δ forces. Le génome est donc constitué de δ chromosomes indépendants. Chaque chromosome représente avec $(1 + \alpha + \beta)$ valeurs réelles le gain, la fonction de direction et la fonction d’intensité permettant de déterminer une force, comme il apparaît sur la figure 3.

Tous les réels qui constituent un génome appartiennent à l’ensemble $[0, 1]$. Toutes les manipulations de ces réels par les opérateurs génétiques se font vers ce même ensemble afin de garantir la généricité des opérateurs de recombinaison. Lors de la transformation du génome en un contrôleur, ces paramètres génétiques sont traduits vers l’intervalle adéquat, qui est généralement propre à une simulation particulière.

2.3.3. *Opérateurs génétiques*

GACS fait appel à deux types d’opérateurs génétiques : la mutation et le croisement. La mutation, en opérant par légères modifications aléatoires, permet d’effectuer une exploration locale dans le voisinage d’une solution. Le croisement échange le matériel génétique entre deux solutions et permet d’explorer l’espace de recherche intermédiaire entre différentes solutions. Ces deux opérateurs sont appliqués avec des probabilités p_M et p_C qui diffèrent selon les expériences.

La mutation ajoute à chaque réel d’un génome une valeur aléatoire issue d’une loi normale. Cet ajout est effectué pour chaque valeur avec la probabilité p_M qui dépend

de la taille du génome, de sorte que la probabilité de mutation globale d'au moins un réel dans tout le génome soit égale à 1.

Le nombre de chromosomes est variable d'un individu à l'autre. Si les parents possèdent respectivement (n_{C1} et n_{C2}) chromosomes, le croisement, appliqué avec une probabilité p_C , consiste à choisir aléatoirement n_{C3} chromosomes parmi les ($n_{C1} + n_{C2}$) chromosomes des parents. Le résultat de ce mécanisme est que la complexité des contrôleurs en termes de nombre de forces impliquées peut s'adapter par dérive génétique à la complexité du problème posé. Par ailleurs, le choix d'échanger des chromosomes entiers entre individus est justifié par le fait que chaque chromosome représente intégralement l'influence d'un point d'intérêt, donc d'une combinaison de points pertinents, sur l'agent. Par conséquent, il s'agit d'un bloc de construction pertinent pour notre mécanisme d'évolution. Or l'identification de blocs de construction est au cœur des mécanismes qui font l'efficacité des algorithmes évolutionnistes. Faciliter la détection de ces blocs de construction est donc crucial pour le succès de notre approche.

3. Expériences

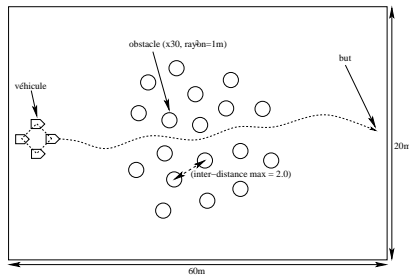


Figure 4. *Expérience de maintien en formation ; un groupe d'agents doit atteindre un but tout en restant en formation et en évitant les obstacles*

3.1. Simulation

L'expérience présentée ici est inspirée de celle développée dans les travaux de Balch et Hybinette (Balch *et al.*, 2000) sur le maintien de formation. Dans cette étude, Balch s'appuie sur la théorie des schémas pour traduire son expertise sous la forme d'un paramétrage particulier d'un ensemble de schémas prédéfini. Nous avons choisi de reproduire cette expérience car GACS s'inspire de cette théorie et s'efforce de la généraliser pour automatiser la mise au point des mécanismes de coordination spatiale.

Originellement, cette expérience a été développée sur la plate-forme *Team Bots*⁵ dans le but de prototyper des architectures de contrôle pour des véhicules militaires terrestres sans conducteurs. Notre simulation, reprenant les mêmes spécifications, implique quatre véhicules holonomes traversant une zone encombrée d'obstacles,

5. Disponible à l'adresse : <http://www.cs.cmu.edu/coral/minnow>.

comme indiqué sur la figure 4, selon une formation qui contraint les véhicules à se déplacer en losange en suivant le véhicule de tête. La zone de manœuvre est un rectangle de $20m \times 60m$. Trente obstacles de $1m^2$ de surface et distants de $2m$ les uns des autres sont répartis autour du centre de la zone de façon à empêcher que le groupe de véhicules puisse traverser la zone encombrée sans rompre la formation.

3.2. Contrôleurs de Balch

Afin d'effectuer une étude comparative de notre modèle avec celui de Balch, nous avons reproduit le contrôleur qui a été développé pour cette expérience dans le cadre de la théorie des schémas. Ce contrôleur est composé d'un total de quatorze schémas instanciés à partir d'un répertoire de six schémas abstraits : **éviter-les-obstacles**, **éviter-les-robots**, **aller-vers-le-but**, **maintenir-la-formation**, **aller-au-centre** et **bruiter**. À chaque pas de temps, chaque instance de schéma génère un vecteur représentant un composant du futur mouvement d'un véhicule. Chaque vecteur est multiplié par un gain indiquant son importance relative vis-à-vis des autres vecteurs. Les vecteurs résultants sont sommés pour obtenir la direction et l'intensité du mouvement.

Les vecteurs de sortie des schémas sont déterminés à partir de deux fonctions d'intensité (ou de magnitude) qui renvoient l'intensité du vecteur selon la distance de l'agent au point visé par le schéma. Ces fonctions sont présentées dans la figure 6. La première fonction (a) est décroissante et paramétrée par les réels M et S . Associée à un gain négatif, cette fonction permet de représenter une répulsion de plus en plus forte si l'agent s'éloigne de l'objet concerné, un obstacle ou un autre agent selon le cas. La seconde fonction (b) est croissante et paramétrée par les réels C et D . Associée à un gain positif, cette fonction permet de représenter une attraction de plus en plus forte si l'agent s'éloigne de l'objet concerné, le but, le centre de la formation ou la position idéale dans la formation selon le cas.

La position idéale d'un agent au sein de la formation est représentée par un *site d'attachement* relatif à l'agent le plus proche. Les *sites d'attachements* introduits par Balch sont construits à partir de la position des autres robots, comme illustré sur la figure 5. Le vecteur de sortie du schéma restant, **bruiter**, est construit à partir d'une direction aléatoire, et d'une intensité égale à 1. Il est paramétré par P qui est le nombre d'itérations entre deux changements de direction. Dans toutes nos expériences, nous fixons $P = 5$.

Les différents paramètres et les gains définis par Balch et réutilisés ici sont présentés dans le tableau 1.

Nous avons étudié les performances d'un contrôleur engendré par ce formalisme dans (Flacher *et al.*, 2004) et montré que nous parvenions à reproduire qualitativement les résultats publiés dans (Balch *et al.*, 2000).

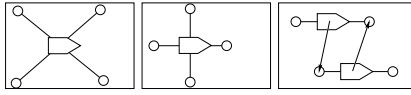


Figure 5. Du point de vue d'un agent, un certain nombre de sites d'attachement (les cercles) sont associés à tous les autres agents. Ces sites sont les points visés par le schéma *maintenir-la-formation*

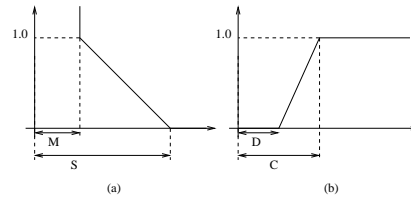


Figure 6. Fonctions d'intensité utilisées par les schémas : (a) *éviter-les-obstacles* et *éviter-les-robots*, (b) *aller-vers-le-but*, *aller-au-centre* et *maintenir-la-formation*

Schémas	Gains
éviter-les-obstacles $M = 0,664$ $S = 2,564$	-1,1
éviter-les-robots $M = 0,4$ $S = 2,3$	-1,1
aller-vers-le-but $D = 0,0$ $C = 0,0$	0,7
aller-au-centre $D = 2,0$ $C = 3,0$	0,6
maintenir-la-formation $D = 0,0$ $C = 1,0$	1,3
bruiter $P = 5,0s$	0,1

Tableau 1. Paramètres des six schémas du contrôleur original de Balch

3.3. Contrôleurs de GACS

Le travail de spécification pour utiliser GACS dans le même cadre expérimental est beaucoup plus simple. Il suffit de donner la liste des amers pertinents présents dans l'environnement, à savoir les obstacles, les autres aéronefs de la formation et le but. Tout le reste, de la construction des barycentres au réglage des fonctions d'intensité, est entièrement du ressort de l'algorithme évolutionniste. On voit donc que, à ce stade, l'investissement du concepteur humain est bien moindre avec notre approche.

Dans cet article, afin de réaliser une comparaison globale des formalismes de représentation, nous allons comparer les performances de GACS à celles obtenues avec le formalisme de Balch en optimisant les paramètres des deux représentations avec un algorithme génétique sélectionnant les individus sur la base des mêmes critères. Nous

donnons la spécification de ces critères et de l'algorithme génétique employé dans les sections suivantes.

3.4. Critères de performance

Pour évaluer le comportement collectif de maintien de formation dans un champ d'obstacles, nous avons recours à 3 critères : la qualité du maintien de formation, de l'évitement d'obstacles et de l'atteinte d'une zone dans laquelle se situe le but.

1) Une récompense est déterminée à chaque instant à partir de l'erreur de positionnement du groupe de véhicules vis-à-vis de la formation idéale. La moyenne de cette récompense sur la durée nécessaire à l'atteinte de la zone cible donne l'évaluation du maintien de formation. Plus formellement, si t_M est le temps alloué pour la résolution de la tâche globale et si r_t est la fonction de récompense qui, à un groupe d'agent donné, associe une valeur positive ou nulle, on a :

$$f_1(a) = \frac{1}{t_M} \sum_{t=1}^{t_M} r[e_f(a)], \quad a \in \mathcal{A} \quad [4]$$

où, si P_1^t, P_2^t, P_3^t et P_4^t sont les positions idéales dans la formation à l'instant t :

$$e_f(a) = \sum_{k=1}^4 |\overrightarrow{A_k P_k}|, \quad r(x) = \begin{cases} 1.0 & \text{if } x < 0.05 \\ \frac{0.5-x}{0.45} & \text{if } x \in [0.05, 0.5] \\ 0.0 & \text{if } x > 0.5 \end{cases} \quad [5]$$

2) Le critère d'évitement d'obstacles f_2 est défini de façon à mesurer la capacité des agents à contourner les obstacles tout en passant à proximité de ceux-ci, afin d'éviter que les agents puissent optimiser le critère en se contentant de ne pas traverser des zones encombrées d'obstacles :

$$f_2(a) = \frac{P_o}{P_o + e^q} \quad [6]$$

où P_o est la somme cumulée sur chaque pas d'une évolution du nombre d'agents qui se situent à l'intérieur de zones définies par les obstacles dont on aurait doublé le rayon, et q est le nombre de collisions cumulé sur tous les agents au cours d'une évolution. Ce nombre est pris en compte exponentiellement afin d'amplifier son impact dans le calcul du critère.

3) A la fin du temps alloué pour la résolution de la tâche, on détermine une récompense à partir de la distance moyenne de tous les véhicules à la zone cible. Cette récompense est l'évaluation de l'atteinte de la zone cible. Plus précisément, si t_M est le temps alloué pour la résolution, si C est la cible, e_g la fonction qui associe, à un

groupe donné d'agents, l'erreur de positionnement cumulée vis-à-vis de la cible et r la fonction de récompense définie dans l'équation (4), on pose :

$$f_3(a) = r[e_g(a)], e_g(a) = \sum_{k=1}^4 |\overrightarrow{A_k C}|, a \in \mathcal{A} \quad [7]$$

3.5. Méthode d'évaluation des contrôleurs

Pour engendrer les contrôleurs de Balch aussi bien que les nôtres, nous utilisons l'algorithme génétique multicritère décrit dans la section 2.3.1 avec les critères de maintien de formation, d'évitement d'obstacles et d'atteinte de but décrits à la section 3.4.

Dans le cas des contrôleurs de Balch, l'algorithme génétique doit régler les paramètres de gain et des fonctions d'intensité des 14 schémas. L'un deux, *bruiter*, n'a qu'un paramètre de gain, donc il y a au total $3 \times 13 + 1 = 40$ paramètres à régler.

Ces paramètres sont représentés par des nombres réels dans $[0, 1]$ qui sont convertis dans $[-100, 100]$ pour les gains et dans $[0, 100]$ pour les paramètres M, S, C et D. L'algorithme génétique n'utilise qu'un opérateur de mutation, dont le taux μ décroît de 1.0 vers 0.1 avec le nombre de générations, selon l'équation de récurrence :

$$\mu = 0,1 \times (1,0 - \tau) \times \mu_{init} + \mu \times \tau \quad [8]$$

On pose $\mu(0) = 1,0$, $\mu_{init} = 1,0$ et la température $\tau \in [0, 9, 1[$ varie d'une expérience à l'autre.

Dans le cas de GACS, l'algorithme doit régler $\sum_{k=1}^{\delta} (1 + \alpha_k + \beta_k)$ paramètres. Dans la population initiale, $\delta = 6$, $\alpha_k = 6$ and $\beta_k = 6$, donc chaque génome contient 78 paramètres, eux aussi représentés par des nombres réels dans $[0, 1]$.

Le nombre de chromosomes peut évoluer grâce à l'opérateur de croisement, donc ce nombre n'est pas constant au cours des évolutions. Nous utilisons une probabilité de croisement $p_C = 40\%$ et une probabilité de mutation $p_M = 5\%$.

Dans les deux cas, nous faisons tourner 10 évolutions séparées sur 10 ordinateurs différents pendant 335 générations⁶. Chaque génération est constituée de $n_P = 100$ individus, le nombre de pas de temps maximum alloué à une simulation est $t_M = 500$ et l'évaluation de chaque contrôleur est donnée par la moyenne de chacun des critères obtenues à partir de $n_E = 25$ positions initiales.

Chaque évolution engendre un front de Pareto. On fusionne alors les 10 fronts de Pareto engendrés par chaque approche en ne retenant que les individus non dominés au sein de la population globale engendrée par les 10 évolutions, donc parmi une population globale de 1 000 individus.

6. GACS est un code IBM Java 1.4.1.02. Ce code est exécuté ici sur une machine Pentium IV à 2,4 GHz sous linux. Dans ce cadre, une évolution standard dure environ 5 jours.

Enfin, on fusionne de la même façon le front de Pareto engendré avec GACS et celui engendré avec l'approche de Balch. Le résultat final est décrit dans la section suivante.

4. Résultats

Dans le cas de l'approche de Balch, le front de Pareto final résultant de la fusion de 10 évolutions contient 332 individus (min = 10 individus par machine, max = 60). Dans le cas de GACS, il contient 240 individus (min = 6 individus par machine, max = 44).

Dans les deux cas, toutes les évolutions contribuent de façon significative à la constitution du front de Pareto. Cela indique que les différentes évolutions convergent régulièrement vers des individus de performance comparable.

À présent, si l'on fusionne les 572 individus engendrés par les deux approches, le front de Pareto résultant contient 356 individus, dont 159 (44,66 %) engendrés par l'approche de Balch et 197 (55,34 %) engendrés par GACS.

Dans cette population, certains individus ont un excellent score en maintien de formation et/ou évitement d'obstacles, mais ils n'atteignent pas le but. En fait, ils obtiennent une excellente performance sur les deux premiers critères en évitant de traverser le champ d'obstacles. Pour éliminer ces solutions insatisfaisantes engendrées par la définition de nos critères de performance, nous supprimons tous les individus qui obtiennent moins de 90% sur le critère d'atteinte de but.

La population obtenue après ce filtrage contient 118 individus, parmi lesquels 62 (52,54 %) sont engendrés par l'approche de Balch et 56 (47,46 %) sont engendrés par GACS. On note donc que si la proportion d'individus issus de GACS était supérieure avant le filtrage, elle est à présent inférieure, ce qui indique qu'une plus grande proportion d'individus non dominés issus de GACS sacrifiaient l'atteinte du but au profit des deux autres critères.

La projection de cette population finale dans un espace défini par les critères de maintien de formation et d'évitement d'obstacles apparaît sur la figure 7. Le fait de ne pas représenter la troisième dimension, définie par le critère d'atteinte du but, explique la présence sur la figure d'individus qui semblent dominés selon les deux dimensions affichées.

La figure 8 représente de façon compacte la performance obtenue à l'issue de chacune des dix évolutions avec GACS et avec l'approche de Balch.

Tous ces résultats sont analysés et discutés dans la section suivante à la lumière de plusieurs critères pertinents d'un point de vue industriel : la performance globale, l'effort de conception, l'originalité et la simplicité des solutions.

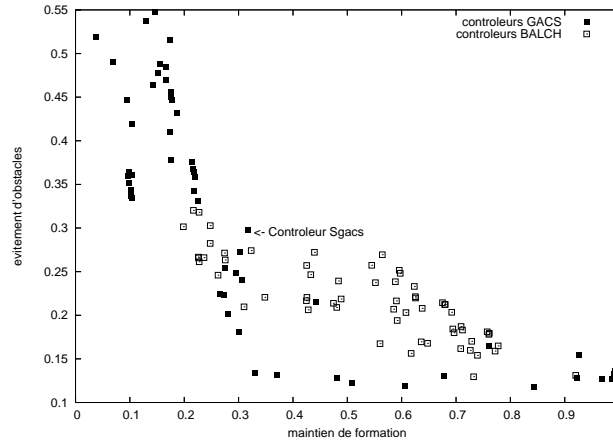


Figure 7. Projection bidimensionnelle du front de Pareto final après filtrage des individus qui obtiennent moins de 90 % sur le critère d'atteinte de but. Le contrôleur S_{GACS} est décrit à la section 5.3

5. Discussion

5.1. Performance globale

Comme nous l'avons montré dans la section précédente, les populations obtenues avec GACS et avec l'approche de Balch ne se dominent pas l'une l'autre. Plusieurs tentatives sous des conditions expérimentales légèrement différentes ont montré que le front de Pareto final est toujours une combinaison équilibrée d'individus engendrés par les deux approches, ce qui indique que les performances engendrées par les deux approches sont comparables.

La figure 8 permet de comprendre les raisons de la faible variabilité des résultats d'une expérience à l'autre. En effet, il apparaît que chaque évolution produit des individus dont la distance aux points zénith et idéal varie dans un intervalle différent à chaque fois, mais les valeurs moyennes de ces distances sont proches d'une évolution à l'autre. Ainsi, l'allure du front de Pareto est différent d'une expérience à l'autre, mais le motif général est toujours similaire, impliquant à peu près autant d'individus engendrés par l'une et l'autre approches.

Cependant, bien que les positions relatives des solutions non dominées dans l'espace des critères change à chaque fois, il semble que, comme c'est le cas sur la figure 7, les meilleurs contrôleurs engendrés avec l'approche de Balch soient souvent plus proches du point zénith et plus proche du centre du front de Pareto que ceux engendrés par GACS.

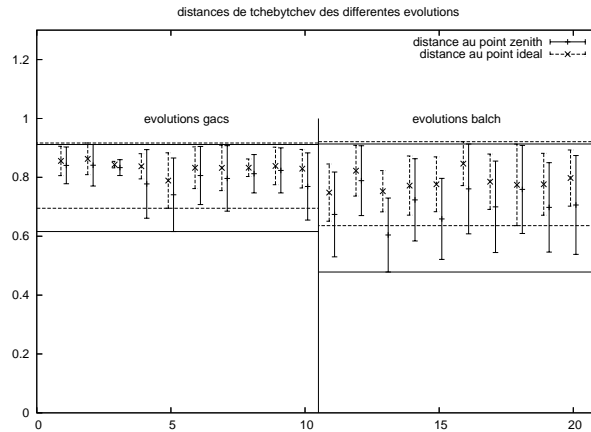


Figure 8. Variabilité des distances de Tchebicheff des individus non dominés au point idéal et au point zénith d'une évolution à l'autre. On appelle point idéal le point défini par la performance maximale théorique sur chacun des critères (ici, $(1,1,1)$). On appelle point zénith le point défini par la performance maximale atteinte par au moins un individu sur chacun des critères. Enfin, la distance de Tchebicheff utilisée ici est la distance donnée par la norme infinie, c'est-à-dire $D_T(A, P) = \max\{|x_P - x_A|, |y_P - y_A|, |z_P - z_A|\}$

Nous devons reconnaître que ce résultat plaide en faveur de l'approche de Balch car les solutions qui se trouvent autour du centre du front de Pareto représentent un meilleur compromis entre les différents critères. En effet, les contrôleurs situés aux extrémités du front de Pareto sacrifient l'un des critères pour optimiser leur performance sur l'autre. Dans notre exemple, on a ainsi affaire à des agents qui ignorent complètement les obstacles pour rester parfaitement en formation ou, au contraire, qui évitent soigneusement les obstacles en ne tenant pas compte de la nécessité de rester en formation. Toutefois, il subsiste toujours des contrôleurs engendrés par GACS dans la région centrale, ce qui se traduit par l'existence de solutions satisfaisantes non dominées par celles de Balch.

Pour obtenir avec GACS davantage de solutions au centre du front de Pareto, on pourrait être tenté d'orienter le processus de sélection de l'algorithme génétique de façon à éliminer au fur et à mesure les solutions qui ne se trouvent pas dans la zone intéressante.

Il s'avère qu'une telle démarche ne fonctionne pas. L'étude des ancêtres des individus qui sont au centre du front de Pareto dans la dernière génération montre que ces ancêtres sont répartis sur l'ensemble de la région atteignable par l'algorithme génétique. Par conséquent, nous ne pourrions pas obtenir ces individus satisfaisants

sans commencer par conserver lors des générations précédentes des individus qui se situent aux extrémités du front de Pareto.

Ce phénomène s'explique aisément. Les solutions non dominées aux extrémités de l'espace des critères sont des solutions spécialisées qui optimisent un critère particulier. Elles contiennent des chromosomes dédiés à la satisfaction de ce critère et optimisent les chromosomes en question.

Par suite, une bonne solution de compromis est susceptible d'être obtenue par un croisement qui regroupe deux solutions spécialisées de ce type, à condition que le gain des forces correspondantes soit fixé à la bonne valeur. On voit que la structure de nos génomes joue un rôle déterminant dans la dynamique d'évolution de nos populations de solutions.

5.2. *Effort de conception*

Notre version paramétrique des contrôleurs de Balch fait appel à sensiblement moins de paramètres que le modèle de GACS, mais cette parcimonie provient d'un effort de conception réalisé par un expert pour fixer a priori une partie des paramètres potentiels. C'est aussi le cas dans les travaux de (Reynolds, 1994) et (Ram *et al.*, 1994). De ce point de vue, notre approche avec GACS doit être comprise comme un moyen de faire l'économie de l'implication d'un expert en confiant l'ajustement de la totalité des paramètres possibles à un algorithme évolutionniste. On a vu que cela se traduit par un effort de spécification bien moindre lorsqu'il s'agit d'appliquer GACS à un cadre expérimental particulier.

Cependant, attendre d'un algorithme évolutionniste qu'il règle un plus grand nombre de paramètres peut avoir un prix, correspondant à un dilemme bien connu. Soit l'algorithme effectue une recherche dans un espace des paramètres très vaste et il risque de voir la performance réduite par une trop grande lenteur de convergence, soit on restreint l'espace des paramètres en fixant certains d'entre eux à la main, et la qualité de la solution trouvée par l'algorithme est fortement dépendante de ce réglage manuel préalable. Nos résultats indiquent que notre représentation spécifique et les opérateurs dédiés qui optimisent les paramètres sont suffisamment efficaces pour rivaliser avec des solutions partiellement réglées à la main.

En effet, là où Ram (Ram *et al.*, 1994) et Reynolds (Reynolds, 1994) se contentent de faire évoluer un ensemble de paramètres avec un algorithme génétique standard sans prendre en compte la sémantique des paramètres, notre plate-forme a été définie de façon à ce que les opérateurs de mutation et de croisement maximisent les chances de découvrir des solutions efficaces. Ainsi, il y a deux « blocs de construction » importants dans notre formalisme :

- les points d'intérêt correspondant aux barycentres des points pertinents pour l'agent ;
- la définition des forces liant les agents à ces points d'intérêt.

Notre opérateur de croisement a été défini de façon à ce que les nouvelles solutions puissent incorporer directement ces forces utilisées dans les solutions précédentes. En conséquence, les chromosomes définissant de telles forces sont conservés et se répandent naturellement dans la population.

En outre, lorsque la connaissance est disponible dans le cadre de problèmes industriels plus complexes, rien ne nous empêche de fixer certains paramètres et de faire évoluer automatiquement les autres avec notre approche. De ce point de vue, notre approche peut clairement être considérée comme plus générale que celle de Balch. Nous avons testé cette possibilité de fixer certains paramètres à la main dans le cadre d'un projet industriel complexe que nous ne décrivons pas dans le cadre de cette publication.

Pour terminer sur la question de l'effort requis de la part du concepteur, il faut reconnaître que la contrepartie du recours massif à un algorithme évolutionniste est que le mode de fonctionnement de certains contrôleurs peut être difficile à interpréter par le simple examen des forces qui le constituent. Mais, d'une part, une telle interprétation n'est pas forcément nécessaire lorsqu'on se contente de rechercher une solution qui marche sans vouloir absolument la comprendre et, d'autre part, il existe en général parmi les solutions proposées des contrôleurs simples dont l'interprétation s'avère aisée, comme nous allons le voir à la section suivante.

5.3. Simplicité des solutions

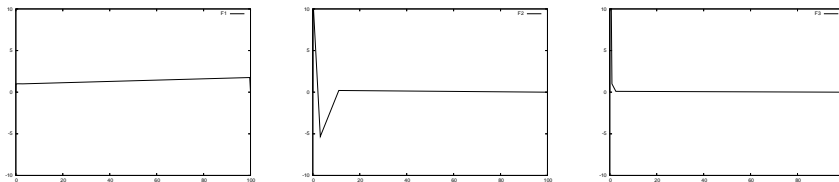


Figure 9. Fonctions d'intensité ω_1 à ω_3 des forces F_1 à F_3 dans le contrôleur S_{GACS} . Les gains correspondants sont 0,69, -1,09 et -1,30 respectivement

Dans le contexte de l'approche de Balch, tous les contrôleurs utilisent systématiquement 14 schémas dont la combinaison est fixée une fois pour toutes. Une conséquence importante de notre représentation plus flexible des génomes est qu'elle peut trouver des solutions plus originales ou plus simples que celles que l'on peut obtenir en se contentant de régler quelques paramètres.

Par exemple, au sein de la population apparaissant sur la figure 7, le contrôleur le plus simple engendré par GACS et désigné par une flèche, S_{GACS} , est constitué uniquement des trois fonctions d'intensité présentées sur la figure 9 :

- la force F_1 est liée au but et indique simplement que, plus l'agent est loin du but, plus celui-ci l'attire (ω_1 est légèrement croissante) ;

– la force F_2 est liée au véhicule le plus proche. Elle indique que si deux véhicules sont très proches (pour des distances comprises entre 0 et 3,20m), ils se repoussent mutuellement, s'ils sont entre 3,20 et 11,1m, ils s'attirent mutuellement et sinon, ils s'ignorent ;

– la force F_3 est liée à l'obstacle le plus proche et se contente de repousser l'agent s'il s'en approche trop.

Ce contrôleur trouvé par GACS est à la fois simple, intuitif et performant. Plusieurs autres contrôleurs de la même population sont constitués de 7 fonctions d'intensité et sont plus complexes à analyser. Comprendre ces solutions simples et originales peut être une source d'inspiration fructueuse pour un expert.

Cette simplicité plus importante tient à deux raisons. D'une part, nous n'imposons par le nombre de fonctions d'intensité alors que, dans l'approche de Balch, ce nombre est fixé. On voit que trois forces suffisent là où Balch en utilise quatorze. D'autre part, notre approche bénéficie d'une plus grande flexibilité dans la façon dont sont construits les points sur lesquels s'exercent ces forces. En effet, dans l'approche de Balch, les forces sont associées soit directement à des objets pertinents, soit à des sites d'attachement dont la construction a été imposée par le concepteur. Dans notre approche, au contraire, les forces peuvent être associées à tout point de l'espace grâce au mécanisme de construction des points d'intérêt, qui sont des barycentres quelconques de points pertinents. Dans le cas du contrôleur très simple que nous venons de décrire, il s'avère que le recours à des barycentres n'est pas nécessaire, puisque chaque force n'est reliée qu'à un seul point pertinent. Toutefois, ce mécanisme a montré toute son utilité dans le cadre d'autres applications que nous avons traitées avec notre approche (Flacher *et al.*, 2002; Flacher *et al.*, 2003).

5.4. Conclusion de la comparaison

Même si la performance obtenue avec la représentation utilisée dans GACS n'est pas meilleure que celle de l'approche de Balch, nous pouvons conclure de cette discussion que notre représentation est dotée de plusieurs avantages vis-à-vis de celle de Balch. Grâce à sa plus grande flexibilité, elle permet en effet de trouver, à performance équivalente, des solutions plus originales ou plus simples, tout en impliquant moins d'efforts de la part du concepteur. D'un point de vue industriel, ce résultat suffit à justifier le recours à notre formalisme de représentation et aux opérateurs génétiques qui s'y appliquent plutôt qu'à une simple optimisation paramétrique classique telle que celle que nous avons appliquée au formalisme de Balch.

6. Conclusion

Nous avons présenté une méthode de mise au point de contrôleurs de coordination spatiale qui suit deux lignes directrices. D'une part, nous avons généralisé le modèle de coordination proposé par Arkin et Balch dans le cadre de la théorie des

schémas. D'autre part, nous avons représenté ce modèle de façon à faciliter l'exploration de l'espace de recherche par un algorithme évolutionniste multicritère dédié. À notre connaissance, notre travail est le seul dans lequel un algorithme évolutionniste multicritère est utilisé pour résoudre des problèmes de coordination spatiale.

Les expériences décrites dans cet article ont mis en évidence les avantages que procurent notre représentation structurée et les opérateurs génétiques spécifiques que nous lui appliquons vis-à-vis d'approches évolutionnistes plus classiques. En effet, notre approche réduit l'implication du concepteur tout en permettant d'obtenir des solutions simples et originales, dont les performances sont comparables à celles de contrôleurs mis au point manuellement par des experts du domaine et optimisés par un algorithme évolutionniste.

Bien que dédié à la résolution de problèmes de coordination spatiale, notre plateforme n'est spécialisée vers aucune sous-classe de problèmes de ce type, et elle a déjà été appliquée avec succès dans plusieurs domaines tels que le regroupement de moutons simulés par des agents bergers (Flacher *et al.*, 2002; Flacher *et al.*, 2003) ou des simulations militaires de plus grande envergure que celles présentées ici ⁷.

Remerciements

Les auteurs remercient Dassault Aviation pour le soutien financier qui a permis la réalisation de ce travail.

7. Bibliographie

- Arkin R. C., « Motor Schema-Based Mobile Robot Navigation », *The International Journal of Robotics Research*, vol. 8, n° 4, p. 92-112, 1989.
- Arkin R. C., « Behavior-Based Robot Navigation for Extended Domains », *Adaptive Behavior*, vol. 2, n° 1, p. 201-225, 1992.
- Balch T., Arkin R. C., « Behavior-based Formation Control for Multi-robot Teams », *IEEE Transactions on Robotics and Automation*, vol. 14, n° 6, p. 1-15, december, 1998.
- Balch T., Hybinette M., « Social Potentials for Scalable Multi-Robots Formations », *IEEE International Conference on Robotics and Automation (ICRA-2000)*, p. 73-80, 2000.
- Clark R., Arkin R. C., Ram A., « Learning Momentum : On-line Performance Enhancement for Reactive Systems », *International Conference on Robotics and Automation*, p. 112-116, 1992.
- Coello C. A., « Multi-Objective Optimization with Evolutionary Computation », in J. Foster (ed.), *GECCO-03 Tutorial Program*, p. 359-383, 2003.
- Deb K., *Multi-Objective Optimization using Evolutionary Algorithms*, Wiley, 2001.
- Deb K., « Evolutionary Multi-Objective Optimization », in R. Poli (ed.), *GECCO-04 Tutorial Program*, p. 585-609, 2004.
- Drogoul A., Meyer J.-A., *Intelligence Artificielle Située*, Hermès, Paris, France, 1999.

7. Quelques animations représentant ces différentes expériences sont accessibles sur <http://animatlab.lip6.fr/pages/RechercheProjetXBotBASC>.

- Flacher F., Sigaud O., « Spatial Coordination through Social Potential Fields and Genetic Algorithms », *From Animals to Animats 7 : Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior*, MIT Press, Cambridge, MA, p. 389-390, 2002.
- Flacher F., Sigaud O., « Coordination spatiale émergente par champs de potentiel », *TSI*, vol. 22, n° 2, p. 171-195, 2003.
- Flacher F., Sigaud O., « BASC, a Bottom-up Approach to automated design of Spatial Coordination », in S. Schaal, A. Ijspeert, A. Billard, S. Vijayakumar, J. Hallam, J.-A. Meyer (eds), *From Animals to Animats 8 : Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior*, MIT Press, Cambridge, MA, p. 435-444, 2004.
- Goldberg D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, Reading MA, 1989.
- Horn J., Nafpliotis N., Goldberg D. E., « A Niche Pareto Genetic Algorithm for Multiobjective Optimization », *Proceedings of the First IEEE Conference on Evolutionary Computation*, vol. 1, p. 82-87, 1994.
- Latombe J., *Robot Motion Planning*, Kluwer Academic Publisher, 1991.
- Matarić M. J., *Interaction and Intelligent Behavior*, PhD thesis, MIT AI Mobot Lab, 1994a.
- Matarić M. J., « Rewards Functions for Accelerated Learning », in W. W. Cohen, H. Hirsch (eds), *Proceedings of the Eleventh International Conference on Machine Learning*, Morgan Kaufmann Publishers, San Francisco, CA, p. 181-189, 1994b.
- Matarić M. J., « Designing and Understanding Adaptive Group Behavior », *Adaptive Behavior*, vol. 4, n° 1, p. 51-80, 1995.
- Meyer J.-A., « The Animat Approach : Simulation of adaptive behavior in Animals and Robots », in F. Alexandre, J.-D. Kant (eds), *Actes de la conférence Neurosciences Pour l'Ingénieur*, Publications Loria, Nancy, 1998.
- Pareto V., *Cours d'Economie Politique*, Rouge, Lausanne, 1896.
- Pearce M., Arkin R., Ram A., « The Learning of Reactive Control Parameters through Genetic Algorithms », *Proceedings 1992 International Conference on Intelligent Robotics and Systems (IROS)*, p. 130-137, 1992.
- Ram A., Arkin R., Boone G., Pearce M., « Using Genetic Algorithms to Learn Reactive Control Parameters for Autonomous Robotic Navigation », *Adaptive Behavior*, vol. 2, n° 3, p. 277-304, 1994.
- Reynolds C., « Flocks, Herds and Schools : a Distributed Behavioral Model », *Computer Graphics*, vol. 21, p. 25-34, 1987.
- Reynolds C., « An Evolved, Vision-Based Behavioral Model of Coordinated Group Motion », *From Animals to Animats 2 : Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, p. 384-392, 1992.
- Reynolds C., « Evolution of Corridor Following Behavior in a Noisy World », *From Animals to Animats 3 : Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, vol. 3, p. 402-410, 1994.