BASC, a bottom-up approach to automated design of spatial coordination

Fabien Flacher[*,**] **Olivier Sigaud**[**]

[*]Dassault Aviation, DGT/DPR/ESA
78, Quai Marcel Dassault, 92552 St-Cloud Cedex

[**]AnimatLab-LIP6, 8, rue du capitaine Scott, 75015 PARIS
fabien.flacher@lip6.fr olivier.sigaud@lip6.fr

Abstract

The design of spatial coordination mechanisms for a group of agents is a challenging problem to which the top-down approach of traditional Artificial Intelligence often fails to find convincing solutions. Bottom-up approaches have given more promising results, but they often rely on a tedious manual hand-crafting that raises scalability issues. In this paper, we show that combining a bottom-up spatial coordination mechanisms with specially designed evolutionary methods to search the space of solutions is an efficient approach to such problems. More precisely, we show the benefits of our platform, BASC, through a quantitative comparison with previous work published by Balch and Hybinette and we conclude on the methodological issues raised by our work.

1. Introduction

Imagine you are a gnu in the wilderness. To avoid feeding one of the local predators, you need to move adequately. For example, you must not go too far away from your mates, staying in the pack of gnus, the shape of which will globally reduce the number of individuals exposed to danger. You can also complicate the task of your predators by making difficult an attack by surprise. For that purpose, you can move so as to avoid obstructing the line of sight of your mates, so that each one can concentrate his attention on a piece of savannah, while the others cover the rest. In order to do this, at each instant, you must observe your neighbors, locate and orient yourself according to their movements, in order to satisfy the collective objective of the group: this is an example of spatial coordination problem.

The scope of this problem is very large. It concerns as well animation in movies, educational, entertainment, insect colony and military simulations (Van Panurak et al., 2000, Collins et al., 2000) and collective robotics (Luke and Spector, 1996, Nolfi and Floreano, 1998). Thus a lot of researchers in different fields try to understand or to reproduce natural mechanisms in order to realize spatial coordination.

This problem raises specific difficulties: the agents should coordinate themselves in an environment often partially observable and should consider their goals, constraints and the goals and constraints of all other agents in order to be correctly located and oriented. This generates a strong interdependence between the movements of each agent. These properties participate in the complexity of the design of the spatial coordination mechanisms. The partial observability problem prevents the use of classical motion planning algorithms which generally need a complete knowledge of the motion problem and the strong interdependence between movements makes ineffective most of the top-down approaches originating from traditional Artificial Intelligence. Such approaches consist in predicting, from the problem specifications, all the possible configurations and associating *a priori* a solution to each relevant configuration. But, predicting all possible configurations of a spatial coordination problem is all the more tedious that the complexity of the interdependence between movements increases with the number of agents.

From these observations, some researchers have tried to apply techniques drawn from Situated Artificial Intelligence or the Animat approach. They rely on a bottom-up method decomposed into two parts: the generation of a variety of potential solutions and the auto-organization or the automatic selection of solutions corresponding to the specifications of the problem.

Important results in this domain stem from Reynolds' *steering behaviors* (Reynolds, 1987). He defined for every agent a combination of speed vectors built from local information and making it possible to avoid inter-agent collision, to match velocity or to gather flock. He demonstrated the generation of successful coordinated group behavior (the so-called *boids*) from the auto-organization of a small set of local behavioral modules. But an important limitation of his approach results from the manual definition and tuning of all speed vectors.

Comparable results were obtained by collective robotics researchers such as Matarić (Matarić, 1994, Matarić, 1995). She exhibited a set of basic behaviors[1] which, once carefully tuned, suffice to solve complex spa-

[1]exploration, aggregation, dispersion and homing

tial coordination problems involving robots, such as encircling or flocking. However, as Reynolds, she had to pre-define these basic behaviors manually.

Arkin and colleagues also investigated spatial coordination mechanisms (Arkin, 1989, Arkin, 1992). They built a theory which decomposes a behavior into *schemas*, sorts of behavioral atoms, represented as attractive or repulsive vector fields. Schemas are combined by summing attractive fields for goals and repulsive fields for obstacles. The collective behavior emerges from the combination of schemas. In some extensions of this theory (Balch and Arkin, 1998, Balch and Hybinette, 2000), Balch did show that the generation and maintenance of formations can be synthesized by combining schemas. But, despite of Balch's innovations, their approach suffers from the same drawbacks as those of Reynolds or Matarić, concerning the manual pre-definition of a set of basic schemas.

Globally, most bottom-up approaches handling the spatial coordination problem were developed according to a similar principle and share analogous limitations. They manually define a set of basic solutions which is consistently restricted and strongly depending on the specific expertise of the designers. From the combination and auto-organization of these solutions, a spatial coordination mechanism emerges, which is again manually selected by designers. Thus, if the overall mechanism does not fit, designers must exploit their expertise to modify or to tune some parameters in the solution set or in their combination. It is clear that such a process is often tedious and would collapse in front of large scale problems. Papers from Arkin and colleagues regularly exposing new schemas to extend the scope of their theory underline the difficulty to find and to manually tune an ideal set of basic schemas[2].

Arisen from this observation, our approach, BASC, is an extension of these bottom-up approaches to the spatial coordination problem, in particular that of Arkin and Balch, with evolutionary mechanisms. Our goal is to get collective behaviors without any manual tuning. For that purpose, we define in section 2. a formal model which reproduces the properties of Arkin and Balch's theory and is completely customizable by a search algorithm. Thus, we have no pre-tuned schemas, but rather a generic definition of any schema. An exploration and automatic selection algorithm manipulates this definition and extracts a specific solution to a given problem.

Since we extend Arkin and Balch's framework, we reproduce in section 3. one of their experiments, consisting in maintaining a group of vehicles in formation while they move to a specified goal and avoid obstacles. In section 4., we verify Balch's results and then we show that we automatically obtain a solution comparable to that of Balch with much less implication from the designer. We then compare our approach to other frameworks coupling the evolutionary tuning of parameters to bottom-up spatial coordination mechanisms, and we show that BASC both looks for a solution in a larger search space and finds them more efficiently in that space. We conclude on the methodological issues raised by this work.

[2]this ideal set is also sought by Matarić

2. A Bottom-up Approach to Spatial Coordination

2.1 About BASC

Our bottom-up approach for the generation of spatial coordination properties consist of into two modules.

A first module allows to synthesize spatial coordination mechanisms resuming and extending principles from the bottom-up approaches mentioned in the introduction. As in the schema theory, this module represents the control of the movement of agents as force fields. But we extend this framework by giving a more generic and parametrized definition of these forces.

Most spatial coordination problems being fundamentally multi-objective, a second module is devoted to the exploration and the automatic selection of controllers by approximating the Pareto Front[3] of the problem. This module is implemented as a Genetic Algorithm (GA) specialized in this type of optimization.

2.2 Spatial coordination synthesis

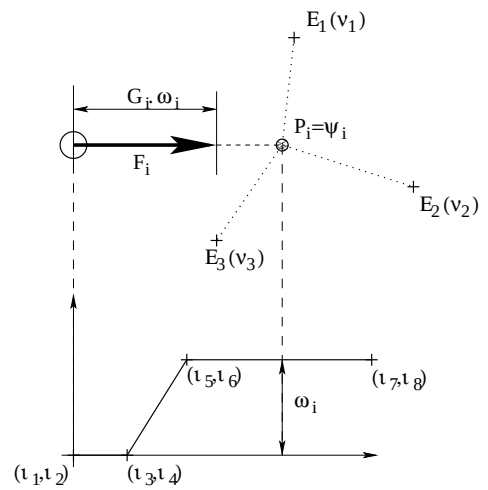

Figure 1: *Example of the construction of one force F_i. The point of interest P_i, targeted by F_i, is built as the barycenter of the perceived elements E_1, E_2, E_3 respectively weighted by ν_1, ν_2 and ν_3. The function ω_i, parametrized by $\iota_1, \cdots, \iota_8$ returns the partial intensity of F_i, finally modulated by the gain G_i.*

[3]the set of *non-dominated* solutions on all the criteria, see section 2.3.

A spatial coordination mechanism needs to correctly position and orient the considered group of η mobile agents. In order to compute this overall displacement, each agent moves by following a movement law built from a combination of forces. Thus, the future movement $\overrightarrow{M_i}$ of an agent A_i is modelled by a set of δ forces $\overrightarrow{F_k}$. Each of these forces is a vector originating from the agent and built from a direction and an intensity.

$$\overrightarrow{M_i} = \sum_{k=1}^{\delta} \overrightarrow{F_k} \tag{1}$$

The direction of a force $\overrightarrow{F_k}$ is obtained by connecting the agent A to a particular point of the environment, named *point of interest*. We define the *points of interest* P_k as barycenters of a some potentially relevant elements of the environment. To determine this barycenter at each time step, the agent A_i establishes a list of elements E_j perceived in its immediate surrounding. The barycenter P_k is then built by associating a weight to every object E_j perceived and referenced. Thus, we define for each force a linear combination ψ_k, parametrized by α reals, taking α elements E_i of the environment $\mathcal{E}$ as entry and producing the point P_k :

$$\psi_k^{\nu_1,\cdots,\nu_\alpha}(E_1,\cdots,E_\alpha) = P_k = \sum_{j=1}^{\alpha} \nu_j.E_j, \quad \sum_{j=0}^{\alpha} \nu_j = 1 \tag{2}$$

The intensity of a force $\overrightarrow{F_k}$ is obtained via the product between a gain G_k and a normalized function ω_k according to the distance of the agent A to the corresponding point of interest P_k. In practice, we define this function of intensity ω_k as a simple piece wise linear function, parametrized by β reals, taking the distance $\| \overrightarrow{AP_k} \|$ as entry and producing the intensity of the force $\overrightarrow{F_k}$:

$$\omega_k^{\iota_1,\cdots,\iota_\beta} = \begin{cases} \frac{\iota_4-\iota_2}{\iota_3-\iota_1}(x-\iota_1)+\iota_2, & x \in [\iota_1,\iota_3] \\ \vdots & \\ \frac{\iota_{i+3}-\iota_{i+1}}{\iota_{i+2}-\iota_i}(x-\iota_i)+\iota_{i+1}, & x \in [\iota_i,\iota_{i+2}] \\ \vdots & \\ \frac{\iota_\beta-\iota_{\beta-2}}{\iota_{\beta-1}-\iota_{\beta-3}}(x-\iota_{\beta-3})+\iota_{\beta-2}, & x \in [\iota_{\beta-3},\iota_{\beta-1}] \end{cases} \tag{3}$$

The gain G_k represents the relative importance of $\overrightarrow{F_k}$ w.r.t. the other forces exercized on the agent.

The movement function $\overrightarrow{M_i}$ of the agent A_i is given by equation (4).

$$\overrightarrow{M_i} = \sum_{k=1}^{\delta} G_k \times \omega_k(\| \overrightarrow{A_iP_k} \|) \times (\sum_{j=1}^{\alpha} \nu_j.\overrightarrow{A_iE_j}) \tag{4}$$

Finally, with our formalism, we are able to synthesize a spatial coordination mechanism from η movement functions $\overrightarrow{M_i}$, $i \in [1,\ \eta]$, one for each agent A_i. Thanks to the parametric aspect of our model, we can represent this mechanism by a set of ρ reals (cf. equation (5)).

$$\rho = \eta.\rho_h, \quad \rho_h = \delta.(1+\alpha_M+\beta) \tag{5}$$

2.3 Automatic exploration and selection

Given a multi-objective optimization problem, a solution must be simultaneously optimized on several criteria. Such a solution is an optimum in the partially ordered space of criteria, and several optimal solutions can be equivalent w.r.t. the criteria they optimize[4]. Such solutions are said *non dominated*. For a given multi-objective problem, the set of *non dominated* solutions is called the Pareto Front (Pareto, 1896).

Since spatial coordination problems can often be formalized as multi-objective optimization problems, we use a GA freely inspired from the one Goldberg applied to the approximation of Pareto Fronts (Goldberg, 1989). This algorithm is efficient enough, flexible and simple to implement, even if several more powerful algorithms have been published recently (Deb, 2001). As any such algorithm, it can be divided into three sub-parts: the genetic encoding representing the parameter setting of a solution, the genetic operators allowing the random modification of solutions and the selection methods allowing the convergence toward the final solution set. These three sub-parts are described in the following sections.

2.3.1 Genetic encoding

In our framework, since the number of agents can increase, we choose to look for an homogeneous spatial coordination mechanism. This means that the same movement function $\overrightarrow{M_i}$ is bred and applied to all agents. Thus, as indicated in equation (5), the number ρ_h of parameters coding this function is independent of the number of agents η.

Moreover, in our GA, one genome codes a movement function for one agent. As previously shown, this function is decomposed into δ attraction-repulsion forces. As a consequence, the genome consists of δ independent chromosomes. Every chromosome represents the gain, the direction function and the intensity function determining one force with $(1+\alpha+\beta)$ real numbers.

The real numbers encoding a genome are chosen in [0, 1]. Any manipulation of these values realized by the genetic operators keep them in [0, 1]. During the transformation of the genome into the movement function, these genetic parameters are mapped to the adequate intervals which depend on the specific application.

[4]For instance, if we try to maximize two criteria c_1 and c_2, a solution can be maximal on c_1 and not on c_2. On the contrary, another solution can be maximal on c_2 and not on c_1. In fact, none of these solutions is globally optimal, they do not dominate each other.

2.3.2 Genetic operators

We use two genetic operators: mutation and cross-over. The mutation operates by random modifications and results in a local exploration in the neighborhood of a solution. The cross-over exchanges genetic materials between two solutions and investigates the search space between different solutions.

The mutation operator is applied by adding a random value depending on a normal law to each parameter of the genome. This addition is made for every value with a probability p_M. The cross-over is applied with a probability p_C in two steps. First, the chromosomes of two solutions are gathered into a set of $(n_{C1} + n_{C2})$ chromosomes. Then, n_{C3} chromosomes are randomly chosen in this set and copied into a new individual.

2.3.3 Selection algorithm

The selection algorithm is designed to retain the most promising genomes and to apply the genetic operators to them so as to generate a new population of solutions. The Pareto dominance principle is applied to perform the selection process. It consists in using the dominance relation to rank each genome from its performances and to derive its selection probability. We used a ranking method based on the *non dominance strata*. The first stratum is the Pareto Front and every genome in this stratum is assigned the rank 1. The second stratum consists of the *non dominated* solutions remaining when the first stratum is removed. The corresponding genomes are assigned the rank 2, etc. The selection probability is proportionally distributed according to the rank of the genomes. Then, as many genomes as needed to create a new population are randomly selected from a roulette wheel algorithm (Goldberg, 1989) on this probability mapping.

3. Experimental set-up

3.1 Simulation environment

The experiments presented here are inspired from the work of Balch and Hybinette on a formation maintenance problem (Balch and Hybinette, 2000). On this problem, Balch applied the schema theory as a spatial coordination model. He mapped his expertise of this problem into the parameter setting of a set of schemas. We chose to reproduce this experiment because of the similarity between BASC and this schema theory. Indeed, our formalism includes the one defined by Balch and Arkin and extends it toward a completely automated design process. Our concept of point of interest is equivalent to that of point aimed by the schema, our intensity function is equivalent to Balch and Arkin's function of magnitude and we both modulate it by a gain.

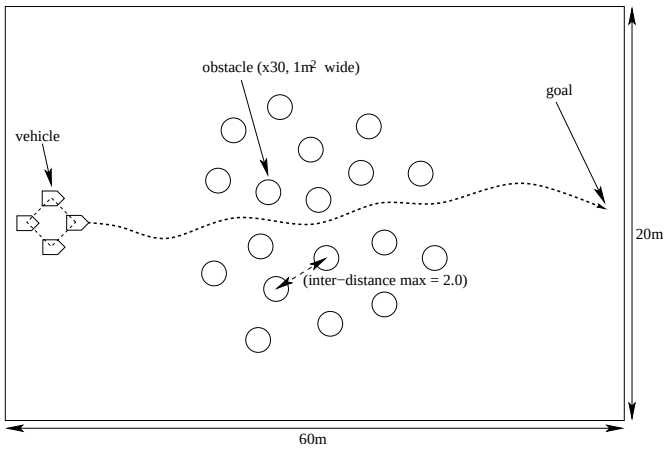

Figure 2: *Formation maintenance experiment: a group of vehicles must reach a goal while staying in formation and avoiding obstacles.*

At the origin, this experiment was developed on the *Team Bots platform*[5] by Balch to prototype control architectures for unmanned ground military vehicles. Our simulation resumes their specifications. As indicated in figure 2, it implies four holonome vehicles moving across a field as quickly as possible while maintaining a geometric formation and avoiding collisions with obstacles and other vehicles. The field is 20m x 60m wide. 30 obstacles, each $1m^2$ wide, are distributed around the center of the field in a way that prevents the group of vehicles from crossing the field of obstacles without deforming its formation[6]. We consider the geometric formation indicated on the same figure, the four vehicles moving in diamond inscribed into a circle of radius $r_d = 1.5$ meter and following a leader vehicle.

3.2 Performance criteria

Because of the conflict between maintaining formation and avoiding the obstacles, we need to use explicitly different criteria to evaluate the collective behavior of our agents. Thus, we defined three separate criteria (*formation maintenance*, *obstacle avoidance* and *goal reach*), as follows :

- A punishment is determined at each time step from the position error to the ideal formation configuration. The final reward on the run duration t_M is the average over the run. Let P_1^t, P_2^t, P_3^t and P_4^t be the correct position in the formation at instant t. We have the following fitness f_1 for the *formation maintenance* criterion:

[5] avaible online at `http://www.cs.cmu.edu/coral/minnow`.

[6] A maximum inter-distance of 2.0 between each obstacles is imposed.

$$f_1(a) = \frac{1}{t_M} \sum_{t=1}^{t_M} r[e_f(a)],\ a \in \mathcal{A} \tag{6}$$

where e_f is a function giving the formation error of a given group of agents, r a reward function and $\mathcal{A}$ the group of agents:

$$e_f(a) = \sum_{k=1}^{4} |\overrightarrow{A_k P_k}|, r(x) = \begin{cases} 1.0 & \text{if } x < 0.05 \\ \frac{0.5-x}{0.45} & \text{if } x \in [0.05, 0.5] \\ 0.0 & \text{if } x > 0.5 \end{cases} \tag{7}$$

- The number of collisions between vehicles and obstacles during the run is memorized. The difference between the number of collisions q and a maximum number of authorized collisions C_M give the fitness f_2 for the *obstacle avoidance* criterion. Formally, we have:

$$f_2(a) = \frac{c_M - q(a)}{c_M},\ a \in \mathcal{A} \tag{8}$$

- When the assigned time for the task resolution is over, a reward is determined from the average distance of all vehicles to the goal. This reward is the fitness f_3 for the *goal reach* criterion. Formally, if G is the goal and e_g the function giving the cumulated position error of a group of agents w.r.t. the goal at the assigned time for the resolution, we have:

$$f_3(a) = r[e_g(a)],\ e_g(a) = \sum_{k=1}^{4} |\overrightarrow{A_k G}|, a \in \mathcal{A} \tag{9}$$

3.3 Balch's schema-based controller

In order to make a comparative study between our model and Balch's, we first reproduced a schema-based controller according to (Balch and Hybinette, 2000). This controller consists of six schemas: `avoid-static-obstacles`, `avoid-robots`, `move-to-goal`, `move-to-unit-center`, `maintain-formation` and `noise`. At each time step, each schema generates a vector representing a component of the movement of each vehicle. Each vector is multiplied by a gain indicating the relative importance of its associated schema. The six resulting vectors are added to obtain the global direction and intensity of the movement.

The six output vectors of schemas are determined from two functions which return the magnitude of the vector according to the distance to the point aimed by the schema. These functions are detailed in figure 3. The first function is decreasing and is parametrized by

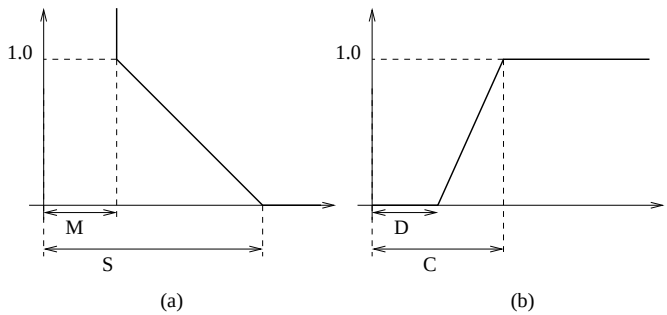

Figure 3: *Intensity functions used by* `avoid-static-obstacles` *and* `avoid-robots` *schemas (a), and* `move-to-goal`*,* `move-to-unit-center` *and* `maintain-formation` *schemas (b)*

the real numbers M and S. Associated to a negative gain, this function realizes an repulsion which increases if the agent get closer to the target, an obstacle for the `avoid-static-obstacles` schema or a robot for the `avoid-robots` schema. The second function is increasing and is parametrized by the real numbers C and D. Associated to a positive gain, this function realizes an attraction which increases if the agent goes away from the target, a goal for the `move-to-goal` schema, the barycenter of the group for the `move-to-unit-center` schema or the ideal position in the formation for the `maintain-formation` schema. This ideal position is the closest attachment site of the vehicle. Attachment sites are built from the position of the others robots, as indicated in figure 4. The output vector of the last schema `noise` is built from a random direction in $[O, 2\pi]$. It is parametrized by P, the number of iterations between any change of direction.

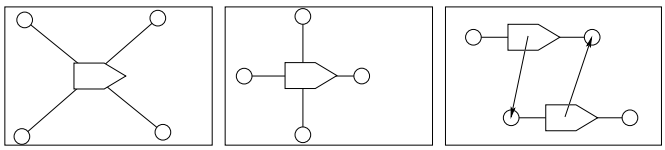

Figure 4: *From the point of view of each vehicle, every other vehicle has several attachement sites, shown with circles, which are the target of the* `maintain-formation` *schema.*

The functions parameters and gains of the various schemas developed by Balch and rebuilt here are summarized in table 1. The corresponding solution will be referred to as S_{balch} in the following.

3.4 Controller generated by BASC

We submitted the same task to BASC in order to obtain a controller for this specific spatial coordination problem. The evolution tuned the gains, directions and

Schemas	Gains
`avoid-static-obstacles` $M = 0.664$ $S = 2.564$	-1.1
`avoid-robots` $M = 0.4$ $S = 2.3$	-1.1
`move-to-goal` $D = 0.0$ $C = 0.0$	0.7
`move-to-unit-center` $D = 2.0$ $C = 3.0$	0.6
`maintain-formation` $D = 0.0$ $C = 1.0$	1.3
`noise` $P = 5.0s$	0.1

Table 1: *Balch's schema-based controller parameters.*

$\delta = 6$ chromosomes / genome		
	parameters per	
type	chromosome	genome
gain G_k	1	6
direction function $\nu_{k,1}, \ldots, \nu_{k,\alpha}$	$\alpha_M = 6$	36
intensity function $\iota_{k,1}, \ldots, \iota_{k,\beta}$	$\beta = 10$	60

Table 2: *Genome encoding of a controller.*

intensity functions parameters (cf. section 2.3.1) during 5 days[7]. 102 parameters on the whole describe the global movement function of each agent as indicated in table 2. These parameters are optimized via 5 evolutions, launched with a population of $n_P = 100$ individuals and a probability of cross-over $p_C = 40\%$, and mutation $p_M = 5\%$. The maximum time assigned to a simulation is $t_M = 500$, the maximum number of authorized collisions is $C_M = 30$ and every evaluation of a controller is realized from $n_E = 25$ initial positions. The average value of the criteria is calculated on these various trials to evaluate the controllers. We present in the following section a comparison of the solution S_{balch} and the best solution obtained with BASC, S_{basc}.

4. Results

4.1 Performance comparison

The results obtained with S_{balch} correspond to those presented in (Balch and Hybinette, 2000). However, this equivalence is essentially qualitative because Balch described only a single quantitative evaluation of his solution: the robustness of task duration when the number of agents increases. As we can notice in figure 5, S_{balch} is scalable on this criterion, since the performance remains close to the average value 623.36 (with a standard deviation of 7.2% of the average duration). Balch concludes that his solution his scalable because the reported average duration standard deviation represents only 10% of this average, which seems to indicate that this performance is stable in spite of the increasing number of agents.

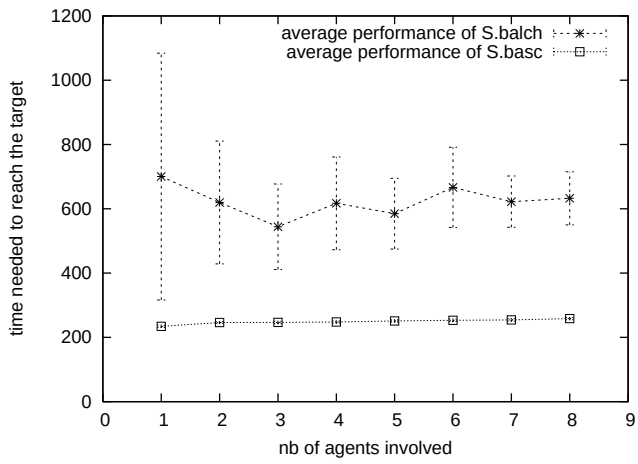

Figure 5: *Task duration scalability. Curves show the average duration over 100 runs, starting with random initial obstacles and vehicle positions. As in Balch's results, S_{Balch} is scalable though we additionally show here with error-bar that is not very robust. Moreover, S_{basc} is also scalable, perfectly robust (the error-bars are shown but reduced to points), and globally faster.*

Table 3.a reports the average performances of S_{balch} and figure 6.1 illustrates the average progress of these performances on 100 runs. From these data, we can observe that S_{balch} generates a correct spatial coordination mechanism which is only disrupted during the crossing of the obstacle field. As shown in figure 6.1, as soon as the group enters the obstacle zone (stage (1.b)), it slows down and forgets the formation until it leaves the zone (stage (1.c)). Then the group resumes a satisfactory speed and a correct formation. This restoring capability of the nominal behavior of the group illustrates the stability and robustness of the generated spatial coordination mechanism. However, the performance of 89.3% to the *goal reach* criterion does not only result from the time lags. Indeed, in 10% of cases, the group fails to exit the cluttered zone. This gap justifies the important standard deviation (14.9%) and error-bars (see figure 5) measured on this performance. This phenomenon results from the fact that Balch's solution adds to each agent an `avoid-static-obstacles` schema for every detected obstacle. When the configuration of the obstacles field is such that the cumulated sum of all `avoid-static-obstacles` schemas begin to be superior

[7]BASC is a Java 1.4.1 code source executed on a 2.4 GHz Pentium IV operating under linux.

solutions	obstacles avoidance	formation maintenance	goal reach
a: solutions tested during 500 iterations			
S_{balch}	90.2 (13.9)	53.1 (10.9)	89.3 (14.9)
S_{basc}	98.5 (3.1)	29.1 (4.8)	100.0 (0.0)
b: solutions tested during 190 iterations			
S_{basc}	98.5 (3.1)	67.6 (8.0)	89.8 (0.8)
$S_{basc} - \overrightarrow{F_4}$	99.8 (1.2)	62.4 (6.9)	88.3 (1.2)
c: confidence* of difference between $S_{basc(190)}$ and			
S_{balch}	$> 99.9\%$	$> 99.9\%$	$> 95.0\%$
$S_{basc} - \overrightarrow{F_4}$	$> 95.0\%$	$> 95.0\%$	$> 95.0\%$

* *using standard distribution-free Mann & Whitney U test*

Table 3: *Performances in % of the different solutions on the 3 criteria. Results are averaged over 100 runs (standard deviation indicated into brackets). Part c of the table gives the statistical significance of difference between the solutions* $S_{basc(190)}$, S_{balch} *and* $S_{basc} - F_4$.

to the cumulated sum from all others schemas, the group stops moving toward the goal, keeping stuck in a local minimum. The more dense the obstacle field is, the more likely it is to encounter such configurations.

Quite as for S_{balch}, the task realization duration from S_{basc} is scalable w.r.t. the number of agents (cf. figure 5). Furthermore, S_{basc} is globally faster than S_{balch} since it needs only 248.91 steps on average (with a standard deviation of 2.7%) to cross the field. More exactly, we notice in table 3.a that the solution S_{basc} perfectly manages to reach the goal (100 % with null standard deviation) and is thus more robust than S_{balch} on this criterion. Nevertheless, S_{balch} is clearly better concerning the *formation maintenance* while both *obstacle avoidance* performances are equivalent. These solutions do not dominate each other in Pareto dominance terms because they are each specialized on one of the aspects of the problem. In fact, a detailed study of the average performances of S_{basc} illustrated in figure 6 shows that the *formation maintenance* performance of S_{basc} is continuously decreasing though the vehicles always succeed to reach the goal (stage (2.c)). Actually, once the goal is reached, the vehicles cannot correctly maintain the formation because they do not stop moving and oscillate around the goal. Since the *formation maintenance* evaluation is a cumulated reward (see equation (6)), S_{basc} is finally weaker than S_{balch} on that criterion. But as shown on table 3.b, when we test S_{basc} during 190 iterations[8] only, the *goal reach* performance of the two solutions become equivalent while S_{basc} starts performing better than S_{balch} on *formation maintenance*. In fact, in that case, a Mann & Whitney statistical test on the average difference significance (see table 3.c), shows that S_{basc} is significantly better than S_{balch} on all criteria. Thus, at a given time w.r.t. the realization of the task, S_{basc} dominates S_{balch} in Pareto dominance terms.

[8] 190 iterations is the average duration needed by S_{basc} to obtain a similar performance on *goal reach* criterion than that obtained by S_{balch} in 500 iterations

force	gain	barycenter P_k expression	function
$\overrightarrow{F_1}$	-38.95	$\overrightarrow{AP_1} = \overrightarrow{AT}$	ω_1
$\overrightarrow{F_2}$	-16.50	$\overrightarrow{AP_2} = \overrightarrow{AV_1}$	ω_2
$\overrightarrow{F_3}$	-90.10	$\overrightarrow{AP_3} = \overrightarrow{AO_1}$	ω_3
$\overrightarrow{F_4}$	-6.11	$\overrightarrow{AP_4} = 0.86.\overrightarrow{AT} + 0.14.\overrightarrow{AO_2}$	ω_4

Table 4: *The four forces used by* S_{basc}. *G is the goal,* V_1 *is the closest vehicle and* O_i *is the* i^{th} *closest detected obstacle.*

4.2 S_{basc} analysis

Solutions obtained via BASC can be difficult to analyze. The large number of parameters in our model gives rise to a large variety of possible solutions. Even so, the solution selected by the GA in this vast set is often complex since the generated behavior results from the interaction of several dynamical interferences. However, with some simplifications, we made this analysis and partially understood how S_{basc} operates. Its genome codes for 4 forces summarized in table 4 and figure 7.

The role of the first three forces is rather obvious:

- $\overrightarrow{F_1}$ attracts every vehicle towards the goal,
- $\overrightarrow{F_2}$ tends to place vehicles on a circle of radius r_0. Indeed, from the point of view of each vehicle, this force is directed towards the closest other vehicle. Moreover, by looking at the intensity function ω_2, we see that it crosses the x axis at $x_0 = 2.26$. As a consequence, if the vehicle influenced by $\overrightarrow{F_2}$ is less than x_0 meters away from the closest vehicle, ω_2 returns a negative value, generating a repulsive force: the concerned vehicle is then repelled by the closest vehicle. Otherwise, it is attracted. Thus $\overrightarrow{F_2}$ generates a stabilization at distance x_0 to the closest vehicle. As all the vehicles move in the same way, they place themselves on a circle of radius $r_0 = \frac{x_0}{\sqrt{2}} \approx 1.60$ and as far from each other as possible.
- $\overrightarrow{F_3}$ repulses every vehicle from the closest obstacle.

Each of these three forces handles one of the three aspects of the problem. Nevertheless as we can notice in table 3.c, the *formation maintenance* is significantly less efficient if we remove the force $\overrightarrow{F_4}$ as confirmed by a Mann & Whitney statistical test. $\overrightarrow{F_4}$ is oriented towards a point which is always very close to the goal and seems

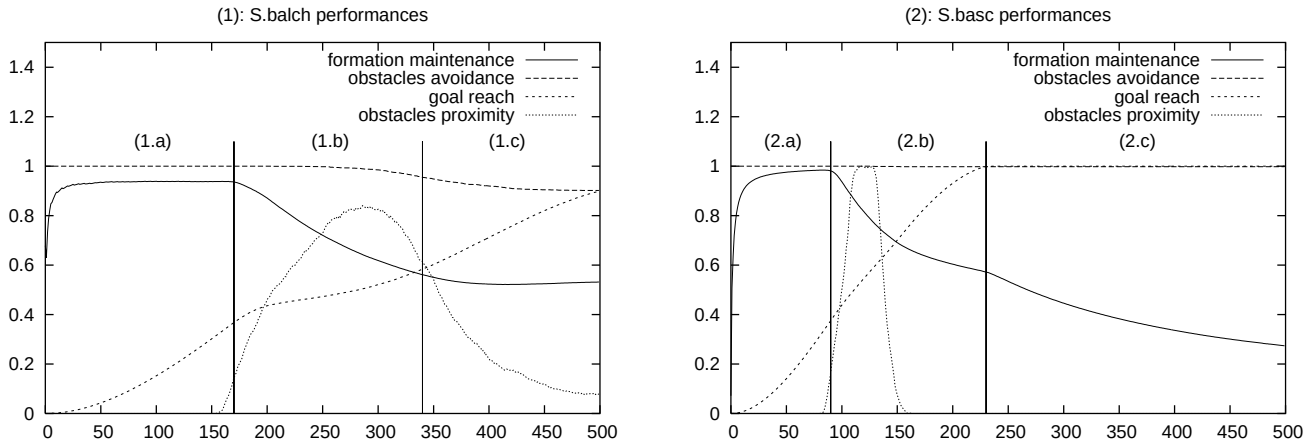

Figure 6: *Performances of S_{balch} (1) and S_{basc} (2). Each curve shows the average progress overs 100 runs on each criteria. The obstacles proximity curves show an evaluation of the virtual collisions number between vehicles and obstacles (the amplitude of this curve is divided by 20 for the sake of clarity). The average behavior of S_{balch} is decomposed into three stages: a stage (1.a) during which vehicles successfully maintain formation and get closer to the goal; a stage (1.b) during which vehicles slow down (constant part of the goal reach curve) and deform their formation (drop of the formation maintenance curve); and a last stage (1.c) during which vehicles resume their previous coordination by getting closer to the goal and maintaining a correct formation. The average behavior of S_{basc} is decomposed into the same stages, but at the last stage vehicles do not stop moving and oscillate around the goal.*

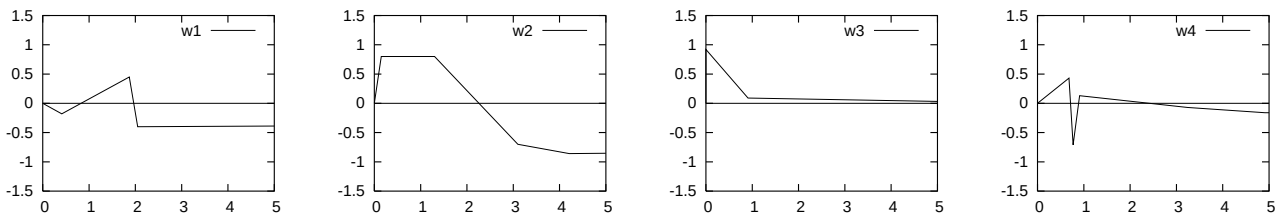

Figure 7: *Evoldved intensity functions ω_1, ω_2, ω_3, ω_4 modulating the intensity of the forces F_1 to F_4 in S_{basc}.*

to play a role in the orientation of the formation in presence of obstacles. But a more detailed geometrical analysis is still necessary before we completely understand its mode of operation.

5. Discussion

Our experiments have demonstrated the possibility to automatically generate solutions to a difficult spatial coordination problem. In particular, the automatic exploration and selection properties provided by our bottom-up approach bring methodological advantages: they significantly reduce the implication of the designer and can even find solutions that they would not succeed to design.

Indeed, it is clear that Balch's schema-based solution that we manually rebuilt tackles the obstacles field crossing problem and is also scalable w.r.t. the number of agents. Admittedly, the simplicity of the schema theory makes it possible to exploit human expertise to develop new schemas. But we have shown that, in front of the variability of this complex problem, the schema-based solution supplied by Balch does not always realize a robust spatial coordination mechanism: it fails in front of too dense fields of obstacles. An expert could possibly correct the lack of robustness, by finding the flaws and manually tuning schemas to generate a new solution. But, as soon as the scale of the problem increases, new interactions problems between agents will arise, increasing the number of possible configurations. As the difficulty to prevent other flaws in the new built solutions increases, experts must re-iterate this endless manual trial-and-error design process, making it more and more tedious. The alternative bottom-up approach developed in BASC is exempt from this drawback. The trial-and-error design process is automated by using a GA. As a result, the effort necessary to define each basic schema is not required anymore and the system will find solutions adapted to the particular problem at hand.

Relying on evolutionary generate-and-test methods,

the system can find original solutions, eventually exploiting counterintuitive relations between agents and their environment, as it was the case in the experiment presented in section 4.2. Indeed, where an expert would probably propose forces such as $\overrightarrow{F_1}$ to $\overrightarrow{F_3}$, BASC has added a force $\overrightarrow{F_4}$ which results in improved performance and requires careful analysis to be understood. Eventually, original solutions proposed by BASC can be reused by the expert in the context of more complex problems.

Furthermore, we have shown that the solution obtained via our bottom-up approach builds a spatial coordination mechanism which dominates the mechanism manually built by Balch. This fact speaks in favor of BASC, but must be considered carefully. Indeed, GAs have a tendency to over-specialize the solutions they generate, *i.e.* to optimize them for the particular conditions under which they are tested. Here, for example, the solutions are tested over 25 different runs during the evolution and on 100 different runs in the final evaluation. The solution discovered via BASC is more successful on these 100 tests and, though each one starts with different initial positions, these tests share a certain number of properties, specific to this experiment (strong density of obstacles concentrated in the center of the zone of operation, for instance). These shared properties can be exploited by the GA to specialize the solutions for these particular cases. They will be more successful in one type of experiment (the one tested), but globally less successful under very different conditions.

From this remarks, we must admit that a comparison with S_{balch} would have been more fair if the solution had been specifically tuned with a GA under the same conditions. Actually, evolutionary algorithms have been applied as optimization processes or extensions of several bottom-up approaches to spatial coordination problems. In particular, works extending on the initial ideas of Reynolds proposed a reduced version of the *steering behaviors* automatically tuned by genetic programming algorithm (Reynolds, 1992, Reynolds, 1994). In this extended version, the basic behaviors can represent in theory any navigation mechanism. Unfortunately, Reynolds has only applied these ideas to mono-agent navigation or predator-prey problems and did not investigate the capacity of these approaches to generate spatial coordination mechanisms. Similarly, Arkin and colleagues did extend the theory of schemas with evolutionary approaches. For example, results exposed in (Ram et al., 1994) show that some model parameters such as the range of influence of some schemas (equivalent to the parameters C D of `maintain-formation` or `move-to-goal` schemas) or the gains representing their relative importance in the final combination can be optimized by a GA. In our framework, this would correspond to setting all the parameters δ_i, α_i to an *ad hoc* value and letting the others ($G_{k,i}$, $\beta_{k,i}$) evolve thanks to a basic GA. But, as it is the case for Reynolds, these works were only applied to the automated design of schemas in a mono-agent navigation case.

The GA dilemma is well-known: either the algorithm searches through the complete parameter space, and the algorithm may be too slow to converge, or the search space is restricted by setting some parameters manually, and the quality of the solutions found by the GA heavily relies on this manual definition. (Reynolds, 1994) and (Ram et al., 1994) have chosen the second solution, but they did so in an implicit way without considering that the parameters they have set manually could have been evolved too. Our approach rather consists in making all the parameters explicitly customizable by a GA, and eventually setting some of them by hand if the search space needs to be reduced. We have tested the possibility of setting some parameters manually in the context of a complex industrial project not described here.

Furthermore, since the search space is potentially very large, our formalism and the corresponding evolutionary algorithm have been specialized to efficiently find good solutions to spatial coordination problems. Indeed, whereas Ram (Ram et al., 1994) and Reynolds (Reynolds, 1994) just evolve a set of parameters with a basic GA without taking into account the semantics of the parameters, our framework has been defined so that the mutation and cross-over operators maximize the chance to discover better solutions out of promising ones. For instance, there are two important *building blocks* (Goldberg, 1989) in our formalism: the point of interest corresponding to a barycenter of relevant objects in the environment, and the definition of a force binding the agent to such a point of interest. Our cross-over operator has been defined so that new solutions can incorporate as such some forces used by the previous solutions. As a consequence, the chromosomes defining useful forces are kept and spread naturally into the population. We have shown in section 4.2 that this feature results in the efficient identification of a set of meaningful forces adequately solving the spatial coordination problem.

Moreover, our system explicitly takes into account the fact that a lot of spatial coordination problems are multi-objective. As a consequence, we use a dedicated Pareto optimization algorithm while all the works discussed so far rely on a single-objective standard GA.

However some authors also take into account the multi-objective aspect of spatial coordination problems. (Pirjanian and Mataric, 2000) define a set of behaviors as objective functions mapping the decisions variables towards $[0, 1]$. Thanks to their model, the agent can foresee the consequences of its actions and choose Pareto-optimal actions according to multicriteria decision theory. On the same line, (Panatier et al., 2000) use potential fields to automatically build for every agent a model

of the behavior of the others. Thus, each agent can foresee the short term future situation and select the best action. Nevertheless, both models impose strong *a priori* constraints on the definition of the basic behaviors, as in the case of Reynolds or Mataric.

Finally, though being dedicated to spatial coordination problems, our framework is not specialized towards any particular subclass of such problems, and has already been applied successfully to several different applications, from a multi-agent instance of the flock-herding problem (Flacher and Sigaud, 2002) to industrial military simulations. Being focused exclusively on simulation applications, we have not tried to address the issue of applying our framework to robotics, but Arkin and Balch did so in the case where the robots accurately know each other's positions, which shows the feasibility of such a transposition.

6. Conclusion and Future Work

As a general conclusion, we have shown that using a bottom-up approach combining a parametric model of spatial coordination with a special purpose GA is a powerful methodology which requires a small design effort and is more efficient in terms of performance than other bottom-up approaches relying either on manual design or on weaker evolutionary methods.

Adopting our approach implies several methodological issues. First, the problem consisting in designing directly a spatial coordination mechanism has been changed into finding the evaluation criteria that match the requirements of the applications, which is not a trivial task. Second, the result of the evolutionary process may eventually be hard to analyze and rely on counterintuitive mechanisms, but this drawback can also be an asset if the analysis reveals an innovative solution to the problem. Third, depending on the complexity of the application, the designer must decide which parameters should be set manually and which should be submitted to the evolutionary process. Here, there is a tradeoff between the designer and the CPU available. In order to make this tradeoff less critical to the usability of our approach, our agenda of future research consist in incorporating in our framework efficient methods coming from the Learning Classifier Systems paradigm, which should hopefully result in a much faster convergence.

Acknowledgments

The authors thanks Dassault Aviation for the financial support that made this work possible.

References

Arkin, R. C. (1989). Motor Schema-Based Mobile Robot Navigation. *The International Journal of Robotics Research*, 8(4):92–112.

Arkin, R. C. (1992). Behavior-based robot navigation for extended domains. *Adaptive Behavior*, 2(1):201–225.

Balch, T. and Arkin, R. C. (1998). Behavior-based Formation Control for Multi-robot Teams. *IEEE Transactions on Robotics and Automation*, 14(6):1–15.

Balch, T. and Hybinette, M. (2000). Social Potentials for Scalable Multi-Robots Formations. In *IEEE International Conference on Robotics and Automation (ICRA-2000)*, pages 73–80.

Collins, D. J., Agah, A., Wu, A. S., and Schultz, A. C. (2000). The effects of team size on the evolution of distributed micro air vehicles. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pages 949–956, Las Vegas, Nevada, USA. Morgan Kaufmann.

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley.

Flacher, F. and Sigaud, O. (2002). Spatial Coordination through Social Potential Fields and Genetic Algorithms. In *From Animals to Animats 7: Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior*, pages 389–390, Cambridge, MA. MIT Press.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, Reading MA.

Luke, S. and Spector, L. (1996). Evolving teamwork and coordination with genetic programming. In *Genetic Programming 1996: Proceedings of the First Annual Conference*, pages 150–156, Stanford University, CA, USA. MIT Press.

Matarić, M. J. (1994). *Interaction and Intelligent Behavior*. PhD thesis, MIT AI Mobot Lab.

Matarić, M. J. (1995). Designing and Understanding Adaptive Group Behavior. *Adaptive Behavior*, 4(1):51–80.

Nolfi, S. and Floreano, D. (1998). Co-evolving predator and prey robots: Do arm races arise in artificial evolution? *Artificial Life*, 4:311–335.

Panatier, C., Sanza, C., and Duthen, Y. (2000). Adaptive Entity thanks to Behavioral Prediction. In *From Animals to Animats 6: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior (supplement book)*, pages 295–303, Cambridge, MA. MIT Press.

Pareto, V. (1896). *Cours d'Economie Politique*. Rouge, Lausanne.

Pirjanian, P. and Mataric, M. (2000). Multi-robot target acquisition using multiple objective behavior coordination. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*.

Ram, A.and Arkin, R., Boone, G., and Pearce, M. (1994). Using genetic algorithms to learn reactive control parameters for autonomous robotic navigation. *Adpative Behavior*, 2(3):277–304.

Reynolds, C. (1987). Flocks, Herds and Schools: a Distributed Behavioral Model. In *Computer Graphics*, volume 21, pages 25–34.

Reynolds, C. (1992). An evolved, vision-based behavioral model of coordinated group motion. In *From Animals to Animats 2: Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, pages 384–392.

Reynolds, C. (1994). Evolution of corridor following behavior in a noisy world. In *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, volume 3, pages 402–410.

Van Panurak, H., Brueckner, S., Sauter, J., and Matthews, R. (2000). Distinguishing Environmental and Agent Dynamics: A Case Study in Abstraction and Alternate Modeling Technologies. In *ESAW Workshop at ECAI'00*.