GACS, an Evolutionary Approach to the Spatial Coordination of Agents

Fabien Flacher
Dassault Aviation, DGT/DPR/ESA
78, Quai Marcel Dassault
92552 St-Cloud Cedex, FRANCE
fabien.flacher@lip6.fr

Olivier Sigaud AnimatLab-LIP6
8, rue du capitaine Scott
75015 PARIS
olivier.sigaud@lip6.fr

ABSTRACT

Our approach to the spatial coordination problem relies on parametrized force fields. Through a quantitative comparison on a complex spatial coordination problem treated with a similar approach by Balch and Hybinette, we show that our system, GACS, finds a population of solutions as efficient as the ones found through an evolutionary optimisation of their handcrafted solution. Moreover GACS generates simpler solutions and requires much less involvement from the designer.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Coherence and coordination

General Terms

Experimentation

Keywords

Spatial Coordination, Multicriteria Evolutionary Algorithm

1. INTRODUCTION

Designing large scale realistic simulations is a more and more common industrial necessity, *e.g.* for movies, video games or military simulations. But it is still a challenging research problem in computer science. In particular, agents in these worlds should at least be correctly positioned and oriented with respect to each other and to their environment, which is the spatial coordination problem.

Some researchers take advantage of techniques stemming from Situated Artificial Intelligence or from the Animat approach. These techniques rely on a two-step bottom-up method: the generation of a variety of potential spatial coordination controllers, and their auto-organization towards a solution controller, corresponding to the specifications of the problem. But the efficiency of these techniques depend on two critical factors. First, it is necessary to make sure that the set of potential controllers is large enough to contain all the relevant solutions. Second, searching a good solution in this set must be easy enough to guarantee that a correct solution can be found in a reasonable delay.

In [3], we already presented our platform, GACS, as an extension of Arkin and Balch's approach with a more flexible parameter set and a dedicated evolutionary mechanism. We compared three controllers, the handcrafted controller published by Balch, a tuned version of it obtained thanks to our multicriteria GA, and a controller generated by GACS under the same experimental conditions. This time, we provide a more global comparison of the populations obtained with the same GA but slightly different evaluation criteria, thanks to a methodological innovation consisting in merging Pareto fronts. We show that, though GACS searches a much bigger parameter space, it finds competitive solutions with respect to these criteria. Then we highlight that, our formalism being more flexible, GACS finds some solutions that are simpler than the ones generated with Balch's representation.

2. GACS: GENERATING AGENT COORDINATION IN SIMULATION

GACS coordinates agents spatially thanks to a combination of forces. Each force is encoded as a chromosome in the genome of the agent. The originality of our approach relies into three features:

- The formalism used to define our forces is similar to the one used in the schema theory from Arkin and Balch, but it is more general. Indeed, rather than defining *ad hoc* attachment sites as Balch does, we bind the forces to barycenters of relevant points, the parameters of these barycenters being adjusted thanks to the evolutionary algorithm.
- In order to tune parameters for a given problem, we use a multicriteria evolutionary algorithm, which results in an easier definition of the performance of the system and in the automatic generation of a population of non dominated solutions.
- The genome of the agents is defined so that the different forces expressed in a controller are treated as *building blocks* by special purpose operators in our genetic algorithm. This results in improved performances.

3. EXPERIMENTAL SET-UP

Our experiments reproduce those of Balch and Hybinette on a formation maintenance problem [1], which 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. Because of the conflict between maintaining formation and avoiding obstacles, we need to use explicitly different criteria to evaluate the collective behavior of our agents.

We define three separate criteria: *formation maintenance*, *obstacle avoidance* and *goal reach*.

• The fitness f_1 for the *formation maintenance* is based on the estimation of the position error with respect to the ideal formation configuration of each agent. The mathematical formulation is given in [3].

• The obstacle avoidance criterion f_2 measures the capacity of agents to avoid obstacles while staying close to them:

$$f_2(a) = \frac{P_o}{P_o + e^q} \tag{1}$$

where P_o is the sum over each steps of one run of the number of agents inside an area defined by obstacles of doubled size, and q is the number of collisions obtained by counting for each step the number of agent colliding with each obstacle.

• The *goal reach* criterion f_3 is given by the average distance of all agents to the goal at the end of a run, as in [3].

In order to make a comparative study between our model and Balch's, we use his schema-based controller described in [1, 3], improved with the possibility to tune the parameters with our GA.

We run our GA with the above criteria on Balch's approach as well as on ours. In both cases, we run 10 separate evolutions during 335 generations. Each run is launched with a population of 100 controllers, each evaluated from 25 different random initial configurations. Each run generates a Pareto front. Then we merge the 10 Pareto fronts by selecting the population of non dominated individuals over the 10 runs. Finally, we merge the Pareto front obtained with Balch's approach with the one obtained with GACS, and compare the contribution of each population.

4. RESULTS

With Balch's representation, the final Pareto front after the merge of 10 runs contains 332 controllers (min = 10 controllers from one run, max = 60). With GACS, it contains 240 controllers (min = 6 controllers, max = 44). In both cases, all runs contribute significantly to the final Pareto front. They consistently converge towards controllers of comparable performance.

After discarding all controllers scoring less than 90% on the *goal reach* criterion, the filtered population contains 118 controllers, among which 62 (52.54%) come from Balch's representation and 56 (47.46%) are generated by GACS.

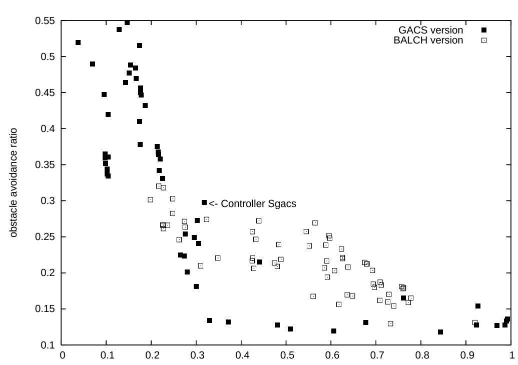

Figure 1: Two-dimension projection of the final Pareto front after having filtered controllers scoring less than 90% on the *goal reach* criterion.

The projection of this final population in a space defined by the *formation maintenance* and *obstacle avoidance* criteria is shown on figure 1.

5. DISCUSSION

The population obtained with GACS and with Balch's approach do not completely dominate each other. Though the relative positions of the non dominated solutions in the criteria space is different each time, the non dominated controllers generated by Balch's approach are more often around the center of the front while GACS controllers are by the sides. This result speaks in favor of Balch's approach since solutions around the center of the Pareto front represent a better compromise between the criteria. But there are always some non dominated solutions generated by GACS in the central area, too.

Our evolved version of Balch's solution requires to tune significantly less parameters than ours, but this results from the careful design of the handcrafted "schemata" by an expert who used his intuition to specify all other parameters in advance. Our approach spares this expert involvement. Furthermore we can fix some parameters and run our GA on the other ones. In that sense, our formalism is more general than Balch's with respect to evolutionary optimization.

And finally, our system can find much simpler solutions than the one generated by Balch's approach. The controller S_{gacs} only uses 3 straightforward forces and can be a fruitful source of inspiration for an expert.

6. CONCLUSION

The automatic exploration and selection properties provided by our platform bring methodological advantages over other bottom-up approaches relying either on manual design or on weaker evolutionary methods: they find simpler solutions and significantly reduce the implication of the designer without being detrimental to the performance. From an industrial point of view, this result justifies the use of GACS rather than Balch's representation.

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 a multi-agent instance of the flock-herding problem [2] or some unpublished industrial military simulations.

7. ACKNOWLEDGMENTS

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

8. REFERENCES

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

[2] F. Flacher and O. Sigaud. 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, 2002. MIT Press.

[3] F. Flacher and O. Sigaud. BASC, a Bottom-up Approach to automated design of Spatial Coordination. In S. Schaal, A. Ijspeert, A. Billard, S. Vijayakumar, J. Hallam, and J.-A. Meyer, editors, *From Animals to Animats 8: Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior*, pages 435–444, Cambridge, MA, 2004. MIT Press.