2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology

Optimization-based Collision Avoidance for Cooperating Airplanes

David Šišlák[†], Přemysl Volf[†], Michal Pěchouček[†],
Niranjan Suri[‡], David Nicholson[§] and David Woodhouse[§]

[†]Agent Technology Center, FEE, Czech Technical University in Prague, Czech Republic
[‡]Florida Institute for Human & Machine Cognition, Florida, USA
[§]Advanced Technology Centre, BAE Systems, United Kingdom
contact author: `sislakd@fel.cvut.cz`

Abstract

Application of autonomous intelligent systems into airspace domain is very important nowadays. The paper presents decentralized collision avoidance algorithm utilizing a solution of the defined optimization problem where efficiency criteria, collision penalties and airplanes' missions are integrated in an objective function. Two different implementation approaches used for stochastic Probability Collectives optimizer are presented and evaluated – a complex distributed multi-agent deployment among participating airplanes and the Process Integrated Mechanism inspired architecture. Both approaches have been validated and evaluated on the multi-agent framework AGENTFLY providing precise simulation for airspace operations.

1. Introduction

In the paper, an optimization based approach to the collision avoidance problem is presented – airplanes search such a series of control actions that would allow them to avoid collision effectively. The *Probability Collectives* (PC) framework [1], [2] is used as an optimization solver. The PC is a stochastic optimizer using probabilistic operators optimizing over a variable space. Efficiency criterion, collision penalty and airplanes' goals are incorporated into a shared objective function. The optimal control is a set of actions which minimize the objective function. The major benefit of the PC optimizer is that the whole optimization process can be distributed among several agents controlling airplanes – several parts can be performed simultaneously.

The work related to the application of Probability Collectives into airplane collision avoidance domain formed part of the Argus DARP (Defence and Aerospace Research Partnership) project. This was a collaborative project involving BAE Systems, QinetiQ, Rolls-Royce, the University of Oxford and the University of Southampton, and was funded by the industrial partners together with the UK EPSRC, MoD, and TSB.

The Process Integrated Mechanism concept is supported in part by the U.S. Air Force office of Scientific Research under grant number FA9550-08-1-0218 and in the past by the Office of Naval Research Coordinated Operations program and the U.S. Army Research Laboratory's TEAM Performance program. Further support is by the Italian MiUR in the frame of the PRIN project "MEnSA - Agent oriented methodologies: engineering of interactions and relations with the infrastructures".

The AGENTFLY is supported by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant number FA8655-06-1-3073 and by Czech Ministry of Education grant number 6840770038.

The paper compares two implementation approaches to the PC deployment for the collision avoidance problem: (i) *parallelized optimization* and (ii) *centralized optimization*. In the first approach, the PC optimization process is done cooperatively by a group of agents. Each optimized variable from PC is mapped to one agent controlling one airplane. This approach can fully profit from a parallelized execution of the PC optimization, but on the other hand it requires a complex negotiation protocol. The second approach requires collection of optimization inputs, selection of a host where the optimization will be performed and distribution of the solution to all involved airplanes. To implement these tasks in the most straightforward way the Process Integrated Mechanism (PIM) [3] has been adopted. The programmer doesn't need to take care about any synchronization and communication issues. All is done automatically utilizing migration of the mobile agent among involved airplanes. On the other hand, such an approach cannot utilize the parallelization potential of the stochastic optimization.

2. Collision Avoidance Optimization Task

The problem of detection and removal of collisions in a group of airplanes is included in the *optimal control* problem. The optimal control problem is formulated as finding the control inputs (heading changes) which minimize a common objective function that penalizes deviations from target oriented controls (fly to their desired way-points) and collision occurrence (based on the required separation distance). The optimization is subject to constraints which ensure that the control inputs are within the flight envelope limitations (e.g. the maximum angular velocity of heading changes). It is supposed that all airplanes can communicate together and always participate in the running PC optimization which is invoked regularly after a given interval Δt.

For simplicity of description, it is supposed that there are N airplanes $\mathcal{A} = \langle A_1, A_2, \ldots A_N \rangle$ which are flying at the same flight level and this level is not changed during the flight and the control action x_i for every A_i is defined as a tuple $\langle \omega_i, g_i \rangle$, where ω_i is the heading angular velocity and $g_i \in \{0, 1\}$

978-0-7695-3801-3/09 $26.00 © 2009 IEEE
DOI 10.1109/WI-IAT.2009.180

specifies whether an action is taken considering the entire future trajectory ($g_i = 0$) or just for the next interval Δt and then proceeding to the airplane's target ($g_i = 1$). Cruise speeds v_i of all A_i are fixed. For each A_i, the next target way-point $\overline{w}_i$ is specified. The function $\mathbf{pos}(A_i)$ returns the current position $\overline{p}_{A_i}$. Whenever the Euclidean distance $|\overline{p}_{A_i}, \overline{w}_i|$ is below the specified tolerance, $\overline{w}_i$ is accomplished by the airplane A_i. In this case, $\overline{w}_i$ is set to the next mission way-point of A_i if there is any. If its final mission way-point has been already reached, the A_i is removed from the set of existing airplanes $\mathcal{A}$. The separation distance R_i specifies that there should not be any other airplane closer than R_i to A_i.

For each PC optimization, the definition set $\mathcal{X}_i$ for x_i contains $2m_i+1$ control actions. There are m_i actions with ω_i values evenly selected from an interval $\langle -\omega_i^{max}, \omega_i^{max} \rangle$ with $g_i = 0$ (repeated application of ω_i), m_i actions with ω_i values selected the same way but with $g_i = 1$ (a single application of ω_i and then proceeding to $\overline{w}_i$) and the single action x_i^{opt}. The ω_i^{max} bounds the maximum available angular velocity, m_i is an odd integer greater than 2 (this ensures that the straight flight manoeuvre is included) and x_i^{opt} is an optimal control action with respect to the specified Δt navigating A_i directly to $\overline{w}_i$. Actions in the set $\mathcal{X}_i$ are ordered by ω_i values (it does not matter whether ascending or descending).

The $\Delta\tau$ is a general parameter for all $\mathcal{A}$ defining the future trajectory sampling interval. The function $\mathbf{f}_i(x_i, k)$ returns the future position of A_i after k intervals $\Delta\tau$ when A_i applies the action x_i considering its current position $\mathbf{pos}(A_i)$ and A_i constraints. The common objective function $\mathrm{G}(\overline{x})$ used for evaluation of joint samples in the PC optimization is

$$\mathrm{G}(\overline{x}) = \mathrm{G}^{col}(\overline{x}) + \alpha \mathrm{G}^{dev}(\overline{x}) \ . \tag{1}$$

It consists of two parts which are summed together using a balancing factor α. The G^{col} penalizes separation violation among all $\mathcal{A}$ using planned future positions and requires separation distances. It is computed as

$$\begin{aligned} \mathrm{G}^{col}(\overline{x}) &= \sum_{A_i \in \mathcal{A}} \sum_{\substack{A_j \in \mathcal{A}, \\ A_j \neq A_i}} \mathrm{G}_i^{col}(x_i, x_j) \ , \\ \mathrm{G}_i^{col}(x_i, x_j) &= \\ \sum_{k=1}^{k_{max}} \beta^{(k-1)} &[\max(R_i - |\mathbf{f}_i(x_i, k), \mathbf{f}_j(x_j, k)|, 0)]^2 \ , \end{aligned} \tag{2}$$

where the factor $\beta \in (0, 1)$ is used for balancing the penalty between earlier and later collisions (smaller β penalizes more earlier separation violations) and the k_{max} defines the look-ahead horizon which is reflected in the objective function. The second part of the objective function G^{dev} penalizes deviation from airplanes' optimal trajectories to their next way-points,

$$\begin{aligned} \mathrm{G}^{dev}(\overline{x}) &= \sum_{A_i \in \mathcal{A}} \mathrm{G}_i^{dev}(x_i) \ , \\ \mathrm{G}_i^{dev}(x_i) &= [\mathbf{f}_i(x_i, k_{max}) - \mathbf{f}_i(x_i^{opt}, k_{max})]^2 \ . \end{aligned} \tag{3}$$

The deviation is expressed as the Euclidean distance between the position at the end of the look-ahead horizon k_{max} after applying the evaluated action and the optimal control action in the case when other airplanes are not considered.

3. Parallelized Optimization

Collectives in the PC algorithm can be taken as groups of self-interested, learning agents that act together to minimize the objective function (Equation 1). Each variable is maintained by one agent. Thus, each agent searches for the optimal action for one airplane (Section 2). $A_i \in \mathcal{A}$ denotes the agent providing control to airplane A_i. Each A_i keeps the current probability distribution q_i for its action variable x_i. Computation of the expected utility value (value of the common objective function $\mathrm{G}(\overline{x})$, Equation 1, and the convergence test requires cooperation of all agents. Sampling and updating of all variables in the iterative procedure of the PC algorithm are performed independently.

The A_i is configured using several airplane-oriented parameters: the maximum available angular velocity ω_i^{max}, the number of discrete steers m_i and the separation distance R_i. Moreover, the A_i manages its given mission and defines the next way-point w_i. All A_i use following common configuration parameters: the size of the sample block in each iteration N_{SB}, the look-ahead horizon k_{max}, the look-ahead step interval $\Delta\tau$, the balancing factor α, the collision penalty time factor β and the annealing schedule parameters.

Distributed implementation of the PC optimization procedure executed by each agent is presented in Algorithm 1. The same algorithm is invoked regularly with Δt period based on the current time for all airplanes. It is required that all airplanes have their time synchronized with an error much smaller than Δt. First, each agent performs an initial setup of the optimal action x_i^{opt}, definition set $\mathcal{X}_i$, probability distribution q_i as uniform discrete distribution over $\mathcal{X}_i$ and temperature T (lines 1–4).

The iterative optimization loop of PC algorithm lowering $\mathrm{E}_q(\mathrm{G}(\overline{x}))$ is implemented at lines 5–26. Agents prepare sample blocks s_i (N_{SB} actions selected from $\mathcal{X}_i$ using Monte-Carlo sampling) and prediction points $pred_i$ (for each action in s_i agents apply function $\mathbf{f}_i(x_i, k)$ where $k = 1..k_{max}$), lines 6 and 7. Then agents exchange their $pred_i$, lines 8 and 9. Computation of the common objective function $\mathrm{G}(\overline{x})$ (equations 1, 2 and 3) for each joint action $\overline{x}$ in sample block s_i is distributed among all agents. Each agent A_i computes G_i of the rewritten objective function

$$\begin{aligned} \mathrm{G}(\overline{x}) &= \sum_{A_i \in \mathcal{A}} \mathrm{G}_i(\overline{x}) \ , \\ \mathrm{G}_i(\overline{x}) &= \alpha \mathrm{G}_i^{dev}(x_i) + \sum_{\substack{A_j \in \mathcal{A} \\ A_j \neq A_i}} \mathrm{G}_i^{col}(x_i, x_j) \ . \end{aligned} \tag{4}$$

The deviation part of the objective function $\mathrm{G}_i^{dev}(x_i)$ is prepared at line 10. Each agent waits for other sample block predictions from $\mathcal{A} \backslash A_i$ agents and adds the collision part of the objective function for each processed $pred_j$, lines 11–15. After processing all predictions each agent has the value $\mathrm{G}_i(\overline{x})$ for each sample block action and it sends these values to all others agents, lines 16 and 17. Then, the agent waits for all other parts and sums the values into G_i, lines 18–22. At this

Input: $\mathcal{A}$
Output: ω_i

{1} $x_i^{opt} \leftarrow$ `Optimal action(`$\overline{w}_i$`)`;
{2} $\mathcal{X}_i \leftarrow$ `Get` x_i `definition set(`ω_i^{max}, m_i, x_i^{opt}`)`;
{3} $q_i \leftarrow$ `Get initial distribution(`$\mathcal{X}_i$`)`;
{4} $T \leftarrow$ `Initialize temperature`;
{5} **while** $true$ **do**
{6} $\quad s_i[N_{SB}] \leftarrow$ `Sampling(`$\mathcal{X}_i$, q_i, N_{SB}`)`;
{7} $\quad pred_i[N_{SB} \times k_{max}] \leftarrow$ `Predictions(`s_i, k_{max}`)`;
{8} $\quad wait\_set \leftarrow \mathcal{A} \backslash A_i$;
{9} $\quad$ `Send(`$pred_i$, $wait\_set$`)`;
{10} $\quad \mathrm{G}_i[N_{SB}] \leftarrow$ `Own dev cost(`$pred_i$, x_i^{opt}, α`)`;
{11} $\quad$ **while** $wait\_set \neq \emptyset$ **do**
{12} $\qquad A_j, pred_j \leftarrow$ `Fetch other predictions`;
{13} $\qquad \mathrm{G}_i \leftarrow$ `Add col cost(`G_i, R_i, β, $pred_i$, $pred_j$`)`;
{14} $\qquad wait\_set \leftarrow wait\_set \backslash A_j$;
{15} $\quad$ **end**
{16} $\quad wait\_set \leftarrow \mathcal{A} \backslash A_i$;
{17} $\quad$ `Send(`G_i, $wait\_set$`)`;
{18} $\quad$ **while** $wait\_set \neq \emptyset$ **do**
{19} $\qquad A_j, \mathrm{G}_j \leftarrow$ `Fetch other costs`;
{20} $\qquad \mathrm{G}_i \leftarrow$ `Add costs(`G_i, G_j`)`;
{21} $\qquad wait\_set \leftarrow wait\_set \backslash A_j$;
{22} $\quad$ **end**
{23} $\quad q_i \leftarrow$ `Update distribution(`$\mathcal{X}_i$, q_i, s_i, G_i, T`)`;
{24} $\quad T \leftarrow$ `Update temperature(`T`)`;
{25} $\quad$ **if** `Converged(`G_i`)` **then break**;
{26} **end**
{27} **return** `Sample final control(`$\mathcal{X}_i$, q_i`)`;

Algorithm 1: Agent PC optimization pseudo-code

point, all agents have the same values of the objective function evaluation in their G_i.

Update of the agent probability distribution minimizing Lagrangian $\mathfrak{L}_i(q_i, T)$ with the current T is done at line 23. Then the temperature T is decreased according to the common annealing schedule, line 24. The convergence test of the iterative optimization procedure is done simultaneously by all agents, line 25. It is not necessary to communicate during this phase as all agents have the same $\mathrm{G}(\overline{x})$ values. Finally, the agent selects the final control according to its stabilized probability distribution q_i, line 27.

4. Centralized Optimization

Centralized PC optimization requires collection of optimization inputs, selection of a host where the optimization will be performed and finally distribution of the solution to all involved airplanes. The concept of the *Process Integrated Mechanism* (PIM) [3] has been adopted to implement the centralized approach. The PIM model consists of a single *Coordinating Process* (CP) and a set of *Components* each capable of running the CP. The CP cycles among the components as required in order to execute the CP algorithm. The underlaying PIM run-time manages the actual movement of the CP across the components, and presents the programmer with a virtual machine in which there is a single coordinating process operating with a unified global view where, in fact, data remains distributed across the components. The programmer needs not be concerned with the details of the process moving among the processors.

{1} $\langle \mathcal{A}, c_{PC}, id \rangle \leftarrow$ `Get initial configuration()`;
{2} **while** $true$ **do**
{3} $\quad \mathcal{S} \leftarrow$ `Get airplanes' states(`id, $\mathcal{A}$`)`;
{4} $\quad \Omega \leftarrow$ `PC optimization(`c_{PC}, $\mathcal{S}$`)`;
{5} $\quad$ `Apply new control(`id, Ω`)`;
{6} $\quad id \leftarrow id + 1$;
{7} **end**

Algorithm 2: PIM coordination process pseudo-code implementing centralized PC optimization for CA

The pseudo-code of CP implementing the described collision avoidance is in Algorithm 2. Initially, the CP reads the initial configuration specifying a set of involved airplanes $\mathcal{A}$, PC configuration parameters c_{PC} and an optimization ID id, line 1. The c_{PC} includes parameters for PC optimization like N_{SB}, k_{max}, α, β and annealing schedule parameters. Then, the CP repeats lines 3–6 until the CP is present. The CP provides periodical control of involved airplanes with period Δt. Each control input is uniquely identified by the optimization ID id. The invocation of the method `Get airplanes' states` blocks until the new control is required for a specified id and also cause migration of the CP among all airplanes $\mathcal{A}$ to collect information about their current state and parameters collected in $\mathcal{S}$, line 3. Once the CP has the necessary information about all airplanes it performs centralized `PC optimization` (Section 2), line 4. Finally, the CP calls the method `Apply new control` which causes migration among all airplanes $\mathcal{A}$ and sets a new control action ω_i which is included for each airplane in the set Ω, line 5. After that, the CP increments id, line 6.

5. Evaluation

Both presented implementation approaches of the same optimization algorithm have been evaluated in multi-agent airspace evaluator AGENTFLY [4]. The algorithm has been tested in a super-conflict configuration [4] with a varying number of airplanes (from 2 to 10). Due to the fact that the algorithm is based on the stochastic PC optimizer, each configuration has been measured in 50 repetitive runs and the presented values are averages from the same configurations. The parameters were set as follows: $N_{SB} = 180$, $\alpha = 1$, $\beta = 0.995$, $\Delta t = 10\ s$, $\Delta\tau = 1\ s$, $k_{max} = 125$, $v_i = 35\ m/s$, $\omega_i^{max} = 8\ ^\circ/s$, $m_i = 5$ and $R_i = 500\ m$. Airplane parameters v_i, ω_i^{max}, m_i and R_i are the same for all airplanes in the configuration. The diameter of the circle where the start and target positions are located is 13 km. Airplanes' positions are well-proportioned on the circle.

During experiments we validated that both implementations provides the same quality of the solution measured as the average trajectory lengthening of the shortest possible trajectories regardless collisions. Results from the experiments are shown in Figure 1. The top chart presents the overall communication flow which was exchanged among all airplanes during the whole flight. The flow amount is linear in the number of airplanes in the collision configuration. The parallelized approach requires more than 8 times higher communication flow

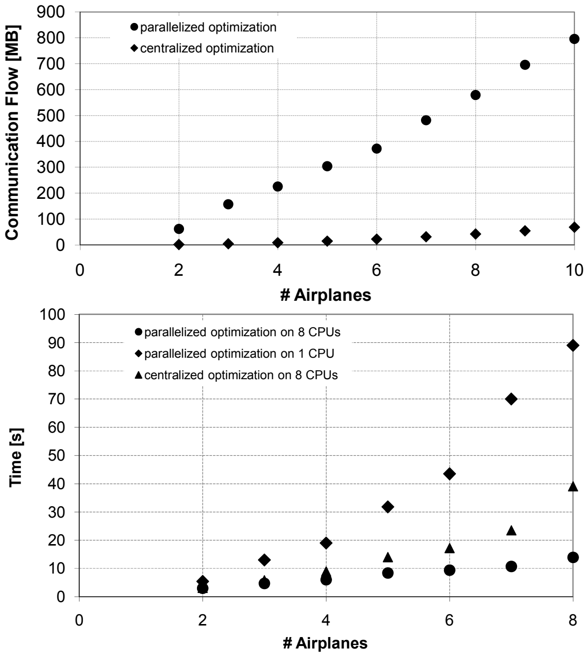

Fig. 1. Experiment results: total communication flow among all airplanes (top), overall time spent for optimization (bottom).

to provide control for the same CA problem. In the parallelized approach, each airplane sends its predictions, partial cost values of the objective function to others. For this communication each airplane utilizes multi-recipient messages which are dispatched in multi-cast manner. The number of optimization iterations is limited due to the used annealing schedule which speeds up the convergence of finding the extreme of the objective function. Thus, there is linear dependence on the number of messages. However, in the case when multi-cast messaging is not usable, the dependence will be quadratic. On the other hand, the centralized approach doesn't require such multi-recipient messages and the communication flow is given only by the migration of the CP among airplanes. In the centralized approach, the communication flow is generated only by the CP migration among airplanes. The CP is required to visit each airplane twice during each optimization: (i) once the CP collects the current airplane state and (ii) when the CP applies the control result to all airplanes. Thus the dependence is always linear and no multi-cast messages are required.

The bottom chart presents the overall time spent by all optimizations (regularly invoked each Δt) in the given configuration. The parallelized approach has been measured in two computer configurations: (i) using 8 computers (connected together by 1Gbps network) each running at most one agent responsible for airplane control and (ii) using only 1 computer where each agent runs in its own process (still requires IP-based communication among processes) but they all share the same CPU. The slowest solution is provided by the parallelized version running on 1 CPU due to communication requirements and also to execution of several code parts by each agent (which is used to remove additional communication). The centralized approach requires less than one half of the time required by the previous version. There was no sense to measure the centralized version in both computer configurations because the computation is executed just at one of the computers. The parallelized version with each agent running on its own computer consumes almost 3 times less time than the centralized version.

6. Conclusion

The parallelized implementation approach based on multi-agent system is much more complex than the centralized one utilizing PIM model. The multi-agent implementation requires transformation of the main PC optimization algorithm that is executed by several agents in parallel. Synchronization parts have to be carefully inserted in the implementation and the common objective function has to be optimally split among all agents so that minimum parts are computed by multiple agents redundantly and only limited information is exchanged among agents. On the other hand, the implementation of the coordination process in PIM model is clearly straight-forward. The PC optimization is implemented in a centralized way and underlying PIM components automatically take care of its transparent migration within the airplane group. Such implementation is very fast and does not require any modification of the algorithm.

During the experimental evaluation of both approaches, it was identified that the price for the complex parallel multi-agent solution is compensated by the performance of the algorithm. The parallelization of the algorithm accelerates the computation up to three times in comparison to the centralized version. On the other hand, the centralized version consumes communication bandwidth only for a limited number of coordination process migrations among airplanes and thus requires about 8 times less flow to solve the same problem. Thus, if there are limits in the communication flow, the centralized approach is better. If fast optimization is required, it is better to use the parallelized approach.

References

[1] C. F. Lee and D. H. Wolpert, "Product distribution theory for control of multi-agent systems," in *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 522–529.

[2] D. H. Wolpert, "Information theory – the bridge connecting bounded rational game theory and statistical physics," in *Complex Engineered Systems*, D. Braha, A. A. Minai, and Y. Bar-Yam, Eds. Berlin: Springer, 2006, pp. 262–290.

[3] K. M. Ford, N. Suri, K. Kosnar, P. Jisl, P. Benda, M. Pechoucek, and L. Preucil, "A game-based approach to comparing different coordination mechanisms," in *IEEE International Conference on Distributed Human-Machine Systems (DHMS)*. IEEE, 2008.

[4] M. Pěchouček and D. Šišlák, "Agent-based approach to free-flight planning, control, and simulation," *IEEE Intelligent Systems*, vol. 24, no. 1, 2009.