MAS-BPM: Multi-Agent System Bomber Problem Model

Zina Elguedria, Boutheina Jlifi, and Khaled Ghédira

L. SOIE, Stratégies d'Optimization des Informations et de la connaissancE
ISG, Institut Supérieur de Gestion, 2000 Le Bardo, Tunisie
zina.guedria@gmail.com, boutheina.jlifi@yahoo.fr,
khaled.ghedira@isg.rnu.tn

Abstract. Simulation has become unavoidable to help control societies with globalization and complexity. This term includes concepts as diverse as strategic decision support or staff training. Military interventions are characterized by using weapons systems increasingly sophisticated in highly diverse contexts. The use of force must control the risk of error to respond to military objectives. Multi-agent simulation is for a long time privileged for modeling and experimentation of complex systems. In this paper, we explore the challenge of simulating a system as complex as the Bomber problem with a MAS approach. Particularly, we demonstrate that MAS approach, models and simulates the Bomber Problem with efficiency. We illustrate our discussion with developed simulation results.

Keywords: Multi-agent System, Bomber Problem, Military domain, Simulation, Complex System, Interaction.

1 Introduction

1.1 Context and Motivation

An extensive literature is present in the military domain; however, the specificity of the bomber problem is that it has been specifically studied from mathematical component.

The problem of air defense emerged from the end of Worldwide War II as one of the most serious threats against the safety of a patrimony.

The first attempts to automate military defense systems were content to combine radar operations scattered with computers in order to achieve a defense system of the United States in Lincoln laboratory [1].

Since the simulation is a kind of representation of reality, then it is strongly related to the concept of model. Indeed, a model is an artificial reconstruction of a system, an entity, a phenomenon or a process. In addition, general military definition is to consider simulation as the real opposition: Anything that is not a real fight is a simulation. There are two main criteria to classify simulation techniques. Begin with the areas of application of simulation in the military domain which are as follows: Education and training; Prospect, studies and evaluations; Employment, planning,

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 61–72, 2013.

preparation and operation control, and decision support. The second classification criteria is composition or military level, as follows: defense system (strategic); the force system (operative); the fighting system (tactical); the weapon system (weapons system, information systems); subsystems of a weapon system [6]. The bomber problem can be classified as defense system and fighting system (strategic and tactical).

Multi-agent simulation is for a long time privileged for modeling and experimentation of complex systems as the bomber problem system.

1.2 Mathematical Formulation of the Bomber Problem

The Bomber problem BP [2-5] can be considered as a discrete time model in which a bomber must survive for t epochs before reaching the target where it will drop its bombs. At each epoch t with probability p, the bomber may encounter an enemy fighter plane.

The bomber has an integer number of Air-to-air anti-aircraft missiles in board, say n. We call these bombs in our approach, defense bombs. Indeed there are two objectives. The first is to accomplish the mission of the bomber successfully. The second objective is to optimize the number of defense bombs k in each encounter with an enemy fighter plane during an epoch t, say k (n, t).

The stochastic dynamic programming equation is clearly [5]:

$$V(0,x)=1 \tag{1}$$

$$V(n,t)=(1-p)\ V(n,t-1)+p\ (\mathrm{Max}(1-\alpha^{k})V(n-k,t-1) \tag{2}$$

Let k(n,t) be the optimal number of missiles to fire when the state is (n,t) and an enemy fighter is encountered. There are three obvious conjectures:

- (A) k(n,t) is nonincreasing in n;
- (B) k(n,t) is nondecreasing in n;
- (C) n-k(n,t) is nondecreasing in n.

Property (C) is easy to prove. Property (A) has been proved (although the proof is not easy. See [2, 3]. Property (B) has never been proved. But it has been tested extensively by numerical solution of the dynamic programming equation. This resolution of (B) remains an intriguing open problem that has been outstanding for 50 years.

The bomber problem is subject to a set of constraints related to the number of defense bombs as follows:

First, the encounter with enemy fighter plane constrains the use of defense bombs.

p is the probability that a meeting is true. Hence, we need to minimize the probability for each period t, thus we must minimize p (n, t).

If p is equal to 0 then there is no meeting and the number of defense bombs remains unchanged.

If the bomber shoots k of its n bomb, he has a survival probability $p = (1-a^{k})$. Therefore we need to maximize p while minimizing k. On the other hand, we must maximize the probability that the aircraft encountered destroy all his enemies and accomplish its mission. Thus, we have to minimize p ($\mathrm{Max}\ (1-\alpha^{k})$).

If the number of defense bombs is exhausted there will be two assumptions: Hypothesis 1: the number of bombs is static. The bomber tries to accomplish his mission by avoiding encounters with the enemy fighter plane.

Hypothesis 2: the number of bombs is renewable. The bomber must seek the nearest and the most secure source.

1.3 Bomber Problem as a Complex System

The Bomber problem is unsolved despite his appearance date since the 1960s [4]. It is classified in the heading of research problems unsolved by Richard Weber [4]. We propose to tackle it by bringing together different axes and research areas such as multi-agent systems, complex systems, and military simulation.

The Bomber problem can be considered as a complex system. A complex system can be defined as a system composed of subsystems, including design and operation involving different trades, and no one can apprehend as a whole. The design of a complex system therefore requires methods and tools to ensure compliance components of subsystems and system specifications eventually throughout their realization: quality of service and capacity of new subsystems. Such is the case of our complex system that involves information technology through multi-agent system simulation in a military application.

2 A Multi-Agent System for the Bomber Problem

2.1 Basic Concepts

Having fixed from the beginning that our goal is to design a distributed system to solve the Bomber Problem. Our distributed model of resolution and simulation has essentially the strength of multi-agent systems, which allow the resolution of problems in terms of cooperation, competition and negotiation in a society of agents. The basic idea of our MAS model is to design: a Base agent who will be the coordinator throughout the resolution process; a set of Craft agents who will be the bombers that are designed to perform one or more tasks assigned to them by the Base agent.

During the accomplishment of its mission, the Craft agent may encounter Enemy agents. This confrontation involves triggering a battle between them. The Craft agent can also carry out its tasks as a team, while forming coalitions. In the following, we detail our model.

2.2 Agent Structure

Each agent has a structure: acquaintances (the agents that it knows and it can communicate with), a local knowledge composed of its static and dynamic knowledge and a mailbox where it stores the received messages that will be later processed one by one.

Base Agent

Acquaintances. This agent communicates with all Craft agents.

Knowledge. Knowledge of a Base agent can be classified into two categories:
Static knowledge such as:

- T-Miss :the type of mission that can be either individual or collective,
- mDist : the matrix of distances between the different Craft agents and different targets,
- lMiss: the list of missions assigned to a Craft agent i.

Dynamic knowledge such as:

- (xi, yi): the position of a Craft agent,
- Nbrdead: the number of dead Craft agent during a mission,
- NbrDestroyed: the number of goals successfully destroyed by Craft agents,
- NbrMove: the number of move made by a Craft agent,
- baseRad: the basis of observations collected from different Craft agents the base agent.

Mailbox. The Base agent has a mailbox where it stores different messages it exchanged with other agents.

Craft Agent. This agent is a bomber. There are N instances of this type of agents in our system. It is a BDI[1] agent. It is defined by its:

Acquaintances. It communicates with all Craft agents and the Base agent.

Knowledge. Knowledge of Craft agent can be classified into two categories:
Static knowledge such as:

- its identificatior i,
- xi and yi : the coordinates of the position i in a map,
- Dmin (i, j): the minimum distance between two Craft agents to get into the same a team,
- QBdi : the capacity of defense bombs gauge of a Craft agent i,
- QBvi : the capacity of bombs gauge to destroy targets by a Craft agent i,
- QFi: the capacity of fuel gauge of a Craft agent i,
- tMiss : the type of mission that can be either individual or collective,
- lMiss : the list of missions assigned to a Craft agent i,
- lDepi : the list of dependent Craft agents to a Craft agent i,
- lSupi : list of supervisor Craft agents of a Craft agent i,
- mDist: the matrix of distances between the different Craft agents and different targets.

Dynamic knowledge such as:

- (xi, yi) : the position of a Craft agent,
- lEqui: the list of the equivalent Craft agents of the Craft agent i.
- lTeami: list coalitions or teams to which the Craft agent i can belong as a supervisor, and from whom it will propose coalitions with the other Craft agents.

[1] Belief, Desire, Intention.

- lHistMi: the history list of the missions carried out by the Craft agent i with mention of the result(success, failure).
- lHistFi: the history list of fights made by Craft agent i with the mention of outcome of the fight(victory, defeat, escape).
- jlifei: the level of gauge of life of a Craft agent i.
- jBvi: the level of bombs gauge to destroy targets by a Craft agent i.
- jBdi: the level of defense bombs gauge of a Craft agent i.
- jCi: the level of fuel gauge of a Craft agent i.
- aRadi: the global view of the Craft agent i, the observations that it transmits to the Base agent.
- lEtati: the list of the states of Craft agents met or with which a Craft agent i communicated. The state of a two Craft agents can be equal to "waiting" if they can or not belong to the same coalition (or equips), "enemy" if they are two enemies or "friend" if they belong to the same team.

Mailbox. The Craft agent has a mailbox where it stores different messages it exchanged with other agents.

We notice that Craft agent's behaviors and Base agent behaviors will later process one by one in global dynamic section.

Enemy Agent. An enemy agent is similar to a Craft agent with a difference that it is reactive and does not belong to the communication system of Craft agents and Base agent. The only interaction between this agent and the others is in a fight. An enemy agent can be described as follows:

- It is a reactive agent.
- Its knowledge is static.
- The enemy agent is destroyed due to a number of successful shots made by a Craft agent during a fight.

2.3 Global Dynamics

In this section, we will describe the global dynamics of the multi-agents system set up: First we start with Base agent behavior. Then we tackle the Craft agent behavior (a mission). In case something happens and requires modification of the Craft agent behavior, it must choose one of its exception scenarios. Principal scenarios of the process and We introduce also, the most important messages exchanged between the various agents to then incorporate them in their behaviors.

Base Agent Behavior. After its creation, the Base agent initiates the process. It is responsible of the creation and initialization (static and dynamic knowledge, acquaintances, and distance matrix) of a number of Craft agents sufficient to accomplish the available missions. It affects missions to various Craft agents. It collects observations from Craft agents and stores them in the radar database denoted by baseRad. The baseRad will be sent to the Craft agents involved.

Craft Agent Behavior. As in real world mission, the Craft agent behavior is a mission. The mission typically includes a sequence of events that must be completed to reach a successful conclusion of the mission. In the beginning, the Craft agent receives its list of missions. It seeks then, the optimal route to reach its target area. Once arrived, the Craft agent bombards the target. If it no longer has a mission in its list and if nothing happens during these phases, the mission is successfully completed

In the case where the list of missions contains more than just a single mission, the Craft agent had to complete all figuring mission. Once mission is completed, the Craft returns to the military base to renew its gauges and get new missions.

In real world of bomber, the aircraft may encounter enemy fighters, it may also have to ask for help from other bombers and accomplish mission in team. Whatever the case, if something happens and requires modification of planed mission; the Craft agent must choose one of its exception scenarios.

Scenario 1: Encounter with an Enemy and Fight. Encounter with an Enemy agent during accomplishment of the mission of the Craft agent may engender a fight. Both Enemy agent and Craft agent are characterized when fighting by:

- Life gauge jlifei: This gauge represents a number of successful shots needed to touch the adversary agent. The exhaustion of this gauge involves the complete destruction of the concerned agent.
- The defense bombs gauge jBdi: This gauge is the number of defense bombs, which an agent may use during Fight.

The life gauge and defense bombs gauge of an Enemy agent are determined randomly when meeting with a Craft agent. In fact, the Enemy agent is part of the external environment of our model, which implies that its behavior is unpredictable, as well as its knowledge and performance.

The Fight scenario between Enemy agent and Craft agent occurs in the following steps: First, the Craft agent checks its gauges level; if the gauges level are insufficient to accomplish a fight (jlifeior jBdi $\leq$ 1), the Craft agent sends a <ReinforcementMsg>to other Craft agents and the Base agent to ask for reinforcement, else with sufficient gauge level, the fight begin. Second, Craft agent pulls a defense bomb on its enemy fighter; The Craft agent defense bomb gauge is decremented by 1. If the shot successfully touches Enemy agent its jlifei is decremented by 1; else nothing happens. The previews step is repeated in turn between the Craft agent and the Enemy agent until one of them is destroyed. Finally, once the life gauge of an agent is zero, it is destroyed. The fight can be concluded with escape, victory or destruction of one of agent fighting.

Scenario 2: Reinforcement. A Craft agent having low gauges (jViei =1 oujBdi, jBvi, jBci $\leq$ 1), sends a<ReinforcementMsg> to the Base agent and other Craft agents for two situations:

- During a bombing mission;
- During a fight.

When receiving <ReinforcementMsg>from a Craft agent, the Base agent assigns the class "dependent" to it. It then sends a message to other Craft agents <askHelp>in the neighborhood of the Craft agent who needs reinforcement (distance ≤ dMin). The Craft Agents receiving a message <askHelp>have the possibility to reply with two sorts of message: It can refuse for three reasons, namely being in a bombing mission, a fight or the Craft agents already belongs to another team. It can also accept. Then, the Craft agent is available and confirms the possibility of forming a team. It enters a process of negotiation with other Craft agents to schedule the team members. After forming the team, Craft agents in "supervisors" class complete the fight interrupted and then the mission, while independent agents seek the path containing the least of danger to return to the base.

Scenario 3: Mission in Team. Craft agent responding with accepts to an <askHelp> message enters in a process of negotiation with other Craft agents to schedule the team members. The Craft agent who sent the request for reinforcement is marked as a dependent agent. The nondependent agents transmit to the other agents in their neighborhood, a message containing the levels of all their gauges. As answer, aCraft agent receives two types of messages:

- If state i is less than the state j, the answer is: I am dependent <UpdDep>.
- If state i is better than the state j, the answer is: I am supervisor <UpdSup>.

In case of receiving a <UpdDep>and when it is already a supervisor of other Craft agents, the Craft agent sends a message to his dependents <UpdSup>. They update the hierarchy of the team in their knowledge bases. The negotiation process ends when after an exchange of messages, the hierarchy does not change and only one Craft agent or set of equivalent supervisor Craft agents lead the team.
Supervisor Craft agent decides who will complete the mission or the fight. As for the final dependent Craft agents (in the bottom of the hierarchy) are trying go back to the military base with the least contact with the environment.

Scenario 4: Radar. The radar scenario is based on the mechanism of MANET[2]. Each Craft agent has radar. Once it detects an Enemy, it sends a message to the Base agent <aRad>which collects the observations of all Craft agents. Subsequently, the Base agent sends a message <BaseRad> to Craft agents located in neighborhoods of these detected enemies.

3 Experimental Design

The objective of this section is to demonstrate the efficacy of the deployment of our multi-agent approach in the context of BP simulation. In fact, our goal is to see through simulating scenarios that have been described, that the missions have been fulfilled with good satisfaction as soon as possible and that coordination led to good

[2] Mobile Ad Hoc Network.

results. To achieve this goal, in the same way that it is necessary to have a methodology when it is a question of conceiving complex applications, it is necessary to have tools and generic components making it possible to consider the reuse and facilitate the development of the applications containing multi-agents systems. To carry out a simulation MAS, several platforms exist. The implementation of our approach was realized using the multi-agent platform JADE based on JAVA language. In our implementation, the type of environment used is a grid whose construction is intuitive.

In our implementation, we note that a simulation run is a mission and each agent can move randomly and independently on a grid (from box to box), which is large enough to simulate a battlefield. Finally, we describe the results obtained from a set of simulations performed according to different configurations.

3.1 Communication and Interaction

Communications and interactions are the focus of our proposed model. We can identify two types of interactions: those that occur between Craft agents and which take part directly in the evolution of the model and those that occur between Craft agents and Base agent.

- Base-Craft interaction: interaction is initiated if a Craft agent wants to perform an action. This interaction is successful if the Craft agent performs desired action. If action type is a move, after a successful interaction, the agent who was in a position it teleports to a random position. In our simulation, a Craft agent can perform 12 actions. Include: moving, bombing, and shooting an enemy.
- Craft-Craft Interaction: An interaction is initiated if a Craft agent wants coordinate with another Craft agent. This interaction is successful if a Craft agent accepts a request for coordination and invites it to a meeting position to start the mission. In our implementation, a Craft agent may initiate team formation by sending a request to another Craft agent to accomplish a mission. Craft agent can: accept, confirm, refuse, or terminate coordination.

3.2 Implementation

A simulation configuration is: Craft agent number/Enemy agent Number/Target Number. The results of the implementation of MAS-BPM are shown in the table 1 according to various configurations.

Table 1 shows the average amount of time taken to complete six simulations. Time shown is in minutes. It also shows the number of coordination within six simulations. Then it shows the number of interaction within six simulations. The six tested simulations were with respectively 5, 10, 20, 25, 50 and 100 Craft agents. These configurations were in two types of grid; indeed we used the first grid with a size of 10x10 and the second one with size of 20x20.

Table 1. Simulation Results with grid size variation

Configuration	Duration		Coordination		Interaction	
	10x10	20x20	10x10	20x20	10x10	20x20
5/5/5	2	10	76	363	736	2872
10/8/8	2	14	135	178	944	5206
20/8/8	3	13	114	257	974	4678
25/10/10	3	17	138	321	1336	4266
50/80/50	17	43	244	554	1271	10568
100/80/80	21	86	396	1763	1724	8559

Time Effect

Simulation Results. Figure 1 plots the average amount of time taken to complete six simulations. The two curve graphs illustrate simulation duration in two types of grid. First curve graph starts in two minutes and ends in twenty one minutes (10x10 grid) beside, second curve graph starts in ten minutes and ends in eighty six minutes(20x20 grid).

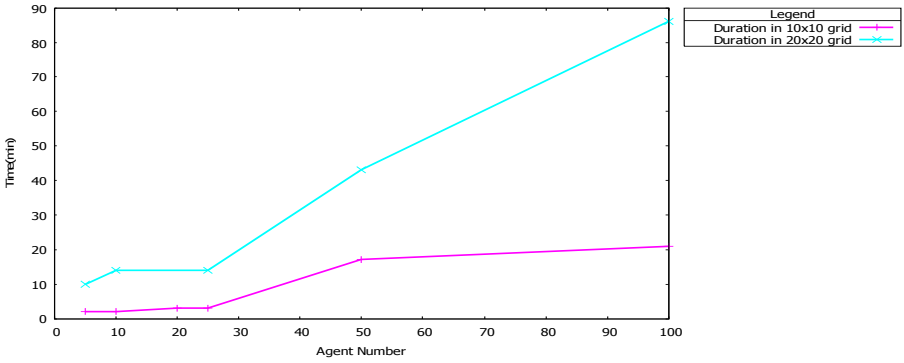

Fig. 1. Simulations duration with grid size variation

Discussion. Varying the grid size influences the variation of the CPU time during simulations. As proof, the first simulations with a grid of size 10x10, last a maximum of 21 minutes while simulations with a grid of size 20x20 are expected to exceed 20 minutes starting from 30 agents to achieve 86 minutes with 100 agents. It should be noticed that the simulation within a grid of size 60x60 lasts 300 minutes.

Coordination and Interaction Effect

Simulation Results. Figure 2 plots the number of coordination carried out during six simulations. The two curve graphs illustrate coordination number in two types of grid. Both curve graphs start zero coordination the first ends in 396 coordination (10x10 grid) beside, second curve graph ends in 1763 coordination (20x20 grid).

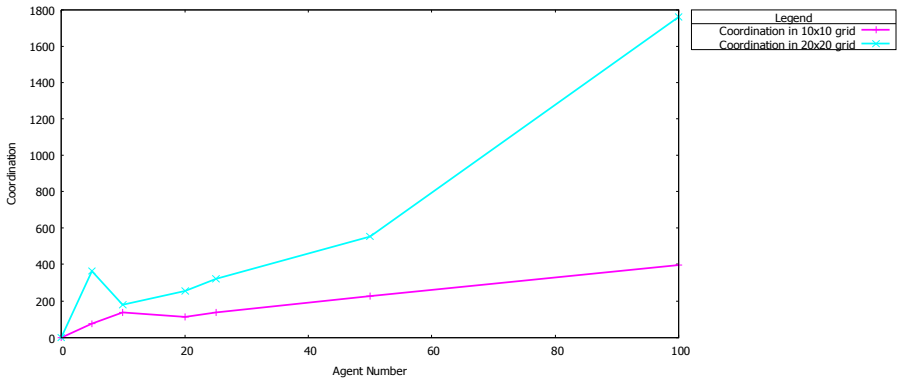

Fig. 2. Coordination number with grid size variation

Discussion. The simulation results demonstrate, as show the curves, in a grid of size 10x10, the effect of coordination is more visible when using a small number of Craft agent. In fact, coordination ensures the fulfillment of the mission and reduces the number of destroyed agents during the mission. Starting from 50 Craft agents, we notice an explosion of coordination number in all simulations tested. In this case, we opt the individual mission accomplishment. In against part, in a 20x20 grid size, the effect of coordination is more visible starting from 50 Craft agent or higher.

For a briefness issue; we have omitted the curve of the number of interaction. The simulation results demonstrate that in a grid of size 20x20, communication and interaction are more important than in a grid of size 10x10. In fact, communication and interaction ensure Craft agent position, enemy position and target position exchange between Craft agents. Indeed, in a small environment, there is no need for an important communication and as shows the figure 1 mission are briefly completed.

Success Ratio

Simulation Results. Figure 3 plots the number of destroyed Craft agent during six simulations in two types of grid. Figure 4 plots success ratio of eleven simulations according to two type of configuration the first one is 25/10/10 and the second one is 50/80/50.

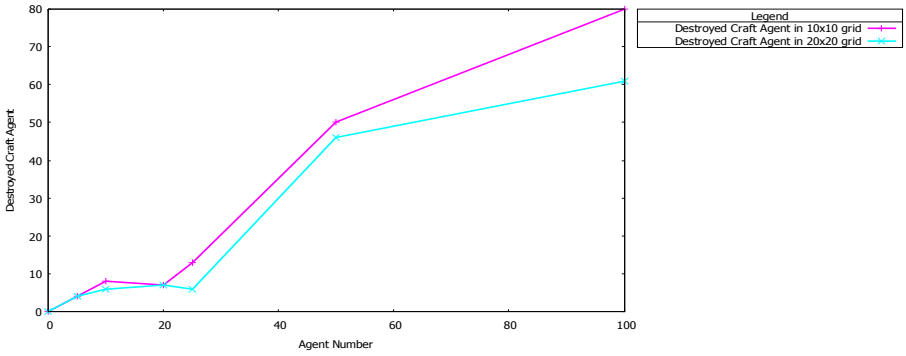

Fig. 3. Destroyed Craft agent with grid size variation

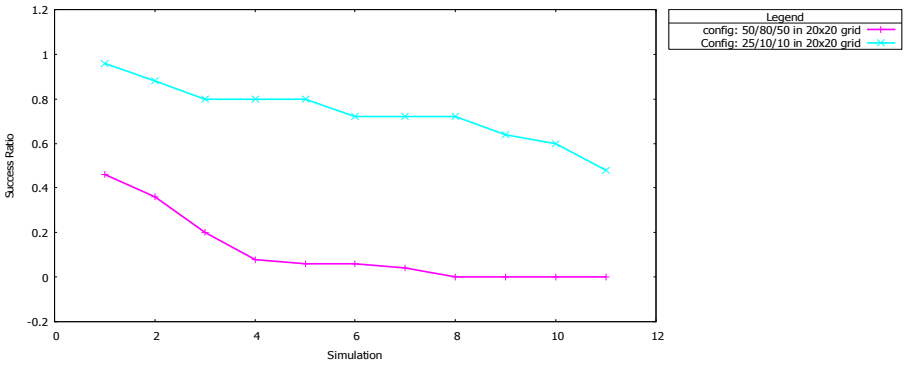

Fig. 4. Success Ratio

Discussion. The simulation results demonstrate, as show the curves in figure 3, in a grid of size 10x10, destroyed Craft agent is more important than in a grid of size20x20. In fact, the important communication and interaction shown in previous section in a 20x20 grid ensure more efficiency when carrying out a mission. Indeed, in a small environment, encounter with enemy is more randomly since there is no important communication between Craft agent to prevent each other when encountering an enemy.

We notice for all configurations, as show the curves in figure 4 the success ratio of a mission depends essentially on the choice of the number of Craft agents as well as good prediction of the number of Enemy agents that will encounter a Craft agent during its mission. As proof, the first simulation with 25 Craft agents against 10 Enemy agents gives success rates ranging from 48% to reach 96%. Beside, when choosing 50 Craft agents against 80 Enemy agents we had a less important success rate ranging from 4% to 46% and sometimes we notice a failure as a mission result, which justifies the importance of good preparation for the fight environment.

4 Conclusion and Perspectives

We have developed a society of agents, dynamically created and cooperating not only to provide a simulation of the Bomber problem but also to accomplish bombers missions and minimize the loss in terms of destroyed aircraft. An implementation and experiments were performed using JAVA language in the context of multi-agent systems. The experimental results show that our simulation deploys with efficacy the BP. In fact, in our case the MAS is replacing a method of optimization. That's why we obtained good results, namely, fewer destroyed Craft agent with more bombarded goals in a reduced amount of CPU time. Craft agents exploit their environment through communication; perform their mission based upon coordination and cooperation through a process of negotiation. Our simulation allowed us to have a good idea about the most effective configuration. It should be emphasized that the originality of our approach consists not only in the fact that the Bomber problem is not resolved but

also in the fact that it brings together different research areas such as multi-agent systems, complex systems and simulation in a military application.

As perspectives, we first propose the necessity to add an optimization method in Craft agents, when searching the target and optimizing the number of defense bombs to fire. Then, we implement our approach using JADEX or Jade4BDI, which are two platforms for the implementation of BDI agents. Finally, we will take into consideration all constraints such as time constraint.

Acknowledgement. We acknowledge and extend our heartfelt gratitude to Pr. Edward Tsang who introduced the Bomber Problem to us and gives a special care to help us to the accomplishment of this work.

References

1. Lemnios, W.Z., Grometstein, A.A.: Overview of the Lincoln Laboratory Ballistic Missile Defense Program. Lincoln Laboratory Journal 13(1), 9–32 (2002)
2. Klinger, A., Brown, T.A.: Allocating unreliable units to random demands. In: Karreman, H. (ed.) Stochastic Optimization and Control, pp. 173–209. Wiley, New York (1968)
3. Samuel, E.: On some problems in operations research. Journal of Applied Probability 7, 157–164 (1970)
4. Weber, R.R.: A problem of ammunition rationing. In: Radermacher, F.J., Ritter, G., Ross, S.M. (eds.) Conference Report: Stochastic Dynamic Optimization and Applications in Scheduling and Relatedfields, p. 148. Facultät für Mathematik und Informatik, University of Passau (1985)
5. Weber, R.R.: ABCs of the bomber problem and its relatives. Annals of Operations Research (2011), doi: 10.1007/s 10479-011-10152
6. ICHEAr 2009 « l'institue des hautes études de défense nationale de la France: Avenir de la simulation pour l.entraînement des forces: quels bénéfices pour le fonctionnement et quelles limites? »45e session nationale (2009), `http://www.ihedn.fr/`