Contribution of computational intelligence tools in the improvement of strategy games

Mohamed Amine Chikh Touami
exact science Departement
University Mustapha Stembouli
Mascara, Algeria
Email: ma_chikhtouhami@hotmail.com

Homer Simpson
Twentieth Century Fox
Springfield, USA
Email: homer@thesimpsons.com

James Kirk
and Montgomery Scott
Starfleet Academy
San Francisco, California 96678–2391
Telephone: (800) 555–1212
Fax: (888) 555–1212

***Abstract*—To evolve a decision engine for an autonomous agent, this paper will represent an Evolutionary Algorithm to define a set of rules that the bot based on to achieve some goals in the game. Planet war is the game used in this study, which Google AI community has been choosing it as a game for their Google AI challenge in 2010, requires that the bot reach some adaptabilities in the aim of beat different enemies in different kind of maps. The Evolutionary Algorithm used to tune multiple parameters to define the bot's behavior.**

I. Introduction

the large entertainment industry sector is the video games. the commercial Video games budget is rising every year, it reaches 25.1 billion dollars in 2010[8]. Many designers and developers are interested in the community of video games around the world. The main objective of these video games community is to provide amusement to their players.The question is how to do the this; there has been no detailed investigation of the full mechanism to put the work in the correct way.[1] In the past, a number of researchers have sought to determine more realistic games.the developers overlooked their games artificial intelligence and focusing their works to high-quality graphics games.In the last year, these features are not sufficient to guarantee a good profit to the games companies. Moreover, the human players are able to defeat these games without much effort and lose the interest. In the recent years, there has been an increasing interest about more features in video games community such as art, psychology narrative to name but a few.[1]

In light of recent events in video games, it is becoming extremely difficult to ignore the existence of one paradigm that is fast becoming a key instrument in most video games is the Artificial Intelligence. The AI is an important aspect of the conception of this kind of games.Moreover, it has seen the rapid of development in many fields such as human player imitation, procedural content generation(PCG), automating game testing, opponent modeling, and computational narrative, among others.[1] Recent development in the AI such as computational intelligence and their applications heightened the need for new challenges to the video games community. Whilst, the debate continues about the best strategies for the management of all these features and technics.[1] Nonetheless, in the past, developers focusing their works on non-players-characters (NPCs) games to control and design the behavior of the players by the integration of the AI for the opponent modeling to design an intelligent player behavior, in order to increase the player interest on these games.[1]

The NPCs developers are conscious that stupid behavior of a virtual player will make this kind of games less interesting and easy to beat. As a solution the video games designers new playing rules such as play around levels.In addition, the difficulties are increased from the bottom levels to the high levels.However, players lose their interesting about these games when they are able to win and defeat all the opponents in the end of the game. Whilst, to make an opponent's behavior evolve with player's abilities is an interesting challenge for developers to make this kind of games more attractive. A considerable amount propositions of literature has been published about behavior's self-adapt to players abilities.[1]

Recently, researchers have shown an increased interest in developing of AI for RTS games. RTS games are a sub-genre of strategy video games, which all the actions made in real-time. The contenders have the capability to control (make a decision) a set of distributed units and structures during the game with the aim of (a) destroy the opponent assets (b)create additional structures to reached some goals in the game or to secure area.[1], [5], [8], [3], [4] The RTS games can be considered as a resource gathering games. Typically, the participants have the possibility to create more units but it's depending on the number of resource gathering with specific units in a specific zone on the map, in order to achieve multiple objectives during the game.[1], [8] At the most of RTS games they employ two levels of AI[5], while can classify these two levels by (a)strategic when it will make a decision over the whole set of units (b) tactical when they decide the behavior of each these small units.In addition, by the real time aspect challenge these difficulties are increased.One of the greatest challenges inherently bounded to the RTS games is the real-time aspect[3]. It associated to make the decision without waiting for the others to move.

Moreover, the actions are made simultaneously during the course of the game. Some researchers have been carried out on these challenges on the computational intelligence in the game conference (IEEE CIG 2010) while less than 10% of them deal with it.[5], [3] One of the international AI challenges is Google AI challenge. Which, the participants with the aim to design an AI programs(bots) to play a real-time strategy game.

One of the hardest tasks in RTS games is the bots conception. For simulating the other contenders, designers use the bot concept.Bots are intelligent agents interact with human players in order to compete them within any computer-based framework[4].This challenge is made by experts human to design bots behavior from their life experiences and experimentation in order to increase the game's challenges.[3] The intelligence bots have a pivotal role[4] in the conception of an RTS games. In addition, a set of parameters are required previous the running of the bot, moreover a good parameters values are make a better behavior from the bot. So, for tuning the parameters previously, is wide research area like an optimization problem itself.[5] This technic is used to determine which parameters are more important for the bot to tune.[5]In a typical RTS games, the contenders(two players or more) have to make a decision under the uncertainty[4](the map is covering by a fog of war, the player will just know their units and structure in the game). There are more difficulties need to understand[4] such as spatial reasoning, strategy planning, and opponent's strategy prediction. Therefore, there are hundreds of units simultaneously acting, many possible positions on the map, units control, among others make the conception of the game more challenging and open a wide space of researches. However, current methods of computational intelligence have proven many types of research proposed potential solutions to solve different tasks such as player experience modeling (PEM), procedural content generation (PCG) and data mining[8].

RTS games are essential for the wide range of technologies[1] in order to design AI, e.g., planning in the uncertainty(lack of information), opponent modeling, temporal reasoning, just to name a few. Like a classic problem, is to provide a non-cheating and humain-like as an important features in the conception of the virtuel players(bots).

A. Planet War

As we mentioned previously one of the international AI competition is organized by google community. In 2010 google choose like a game planet wars for it AI challenge competition. In this pager, we work with a version of Glacon game in order to design a fighter bots. A planet war match[5] runs on a map with several distributed planets in different positions, such as each of them has a number describes the quantity of starships its owns. In each turn, the number of starship will change. Each planet is owned by a player, opponent or neutral planet(owned by nobody),

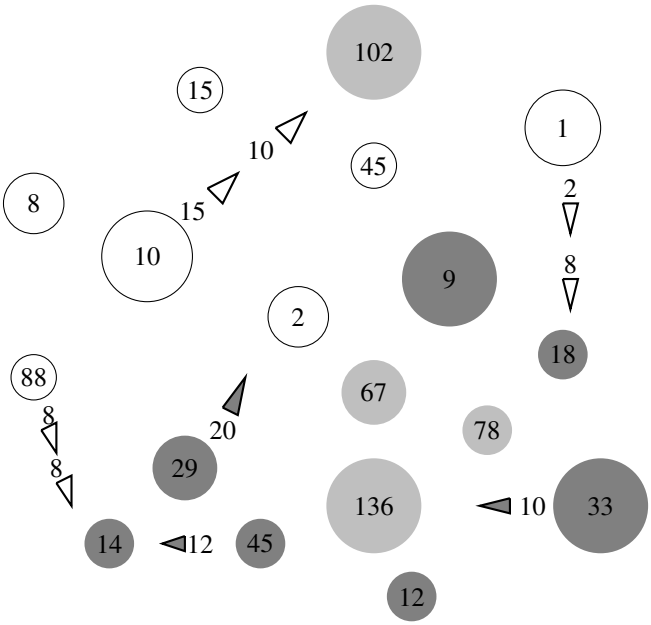

Fig. 1. Simulated of an early stage of a run in Planet Wars. White planets belongs to the player (blue color in the game), dark grey belong to the opponent (red in the game), and light grey planets belong to no player. The triangles are fleets, and the numbers (in planets and triangles) represent the starships.

moreover, each planet belongs to the player (no neutral one) will increase their number of starships after each step according to their growth rate, this rate explain the additional starship's number will generate each planet in each step. The main purpose of the planet war game[5] is to challenge all players and try to defeat all of them, that mean to destroy all enemy's starships. Planet war is an RTS game by its nature. to achieve a specific objective each player has a limit number of turns to do it and try to win (beat opponent).

Moreover, the bot's design faces two major problems: the first one [5]is that the bot can't store any information about its previous turns like actions it did, actions of its opponent or even the map's state. Since in each turn, the bot meets with an unknown map as it just begins the game for the first time. The second one[3] is that the time requires to move(make an action) is just one second. So these difficulties make the bot's design like a great challenge. In this kind of game, everything has a sum of properties[5]. Such as, planets have the x and y coordination for their locations in the map, ownersID, number of starship and growth rate. To conquest other planets, the player needs to send fleets for this aim.Each fleet has a playerID parameter, number of starships, source planetID, destination planetID, turn's number need to reach the target. In each simulated turn[5], the player has the ability to send fleets in a battle with other planet's starships with the source planetID and target planetID, while, the player has the ability[5] to make just one action per turn. After each move (action) the planets belong to the players[1] increase their ship's number according to their growth rate. To make

fleets previously sent reach enemies planets its take time of many turns according to the distance between the source planet and the target planet, moreover, if the source and target planet owned by the same player then it considered like a reinforcment[5], [1] by adding the ship's number on the fleet to the number hosted on the target planet. when the fleets reach the target its will make battles with local target ships, otherwise, if the target is a neutral planet, the player must to sends at least one additional ship to own the planet. While, if the target is owned by the enemy the two players will fight until one with the highest starship number owns the planet. When the player sent fleets to reach some goals[1], he can't change their directions. In addition, the player can send more fleet in the next turns even that the first ones do not yet arrive at their destinations. In the end of the match[1], the player with the highest number of starships wins. If a player loses all their starships then the game will end also faster. whilst, the players have the same starship's number at the end of the game there is a draw[1].

B. immune system overview

One of the most inspirational things that computational based to solve their problems[2] is the biology and its various concepts. Among the natural concepts, the immune system has emerged as one of the important aspects to design optimisation algorithms. An optimization problem[6] is to look for the optimum element from a set of potential solutions. There are many optimization problems in all of the scientific areas. For this purpose, many methods and algorithms have been developed. The most useful algorithms[6] are heuristics.They include divers algorithms such as artificial bee colony, firefly swarm optimisation, ant colony optimization, genetic algorithms, just to mention a few. While the Artificial immune system (AIS) is a common algorithm devised by Decastro[6] that uses population-based heuristic concept. The mechanisms used in the biological immune system against various types of pathogens made this system more interesting[7] for computer researchers to exploit its technics. This kind of algorithms is used to solve many different optimization problems in science and technology areas. However, the immune system is more complex and its application in technologies areas is a challenge, likewise, the biological immune system[2] still under wide active research, whilst, the AIS is adopted just for few mechanisms inspired from the human immune system including the clonal selection.

C. natural immune system

The biological immune system including human system contains cells, molecules, and organs in its structure to defend the body against the diseases. In the aim to make an immune response[6], the system has to distinguish between the body cells as a self-cells and the foreign cells as nonself-cells(antigens). The immune system response is activated by a suitable mechanism[6] to defeat the nonself-cells, this mechanism is defined by the type of the antigens, specific antigen implies a specific response and a specific process. In order to make a fast response in case of the similar antigen, the immune system develops memory cells for this purpose.

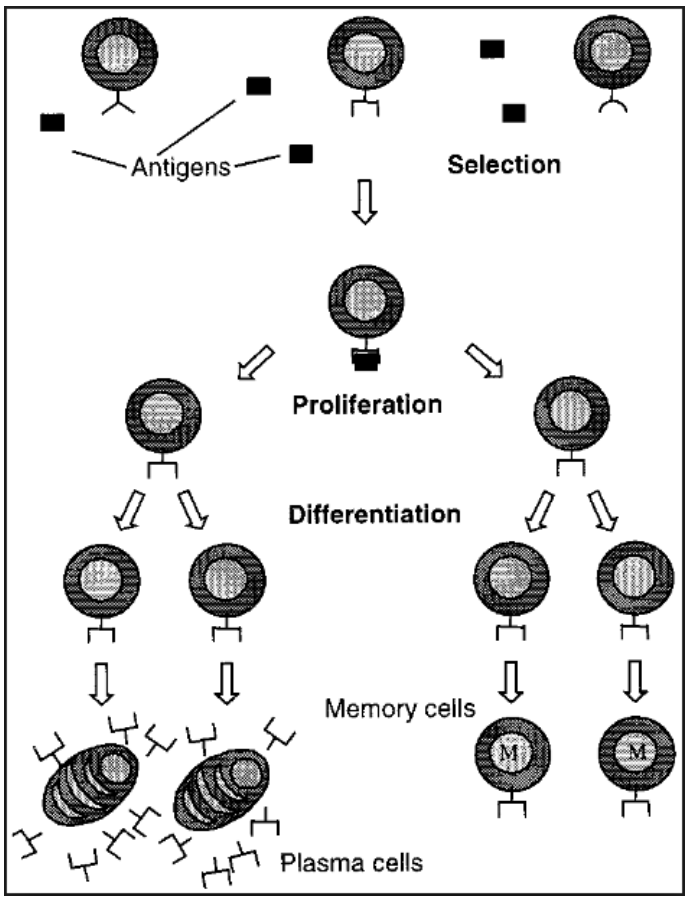

Fig. 2. Clonal selection mechanism

Clonal selection: One of the important mechanism used to defend against the nonself-cells[6] is the clonal selection. This immune response describes the process how the immune system will stimulus against a specific antigen[2] by proliferating a specific type of cells that only those can recognize the antigen.

When the antigen is inside the body, the immune response[6] by giving the B-cells the ability to secrete antibodies. After that, the T-cells give a signal to the B-cells to proliferate and mature to terminal antibodies secreting cells, it's the plasma cells. The proliferation of the B-cells is according to the affinity level, higher affinity implies more clone will be generated, this whole process named affinity maturation.

The clonal selection process will pass with various steps[2]:

a) : The cloned cells undergo to a mutation process.

b) : the self-reactive receptor will be eliminated.

c) : proliferate the mature cells those can detect the antigen.

D. Artificial Immune System

Inspiring from the biological immune system an algorithm has been developed called Artificial Immune System (AIS)[6]. The search technic is similar to the natural immune system by

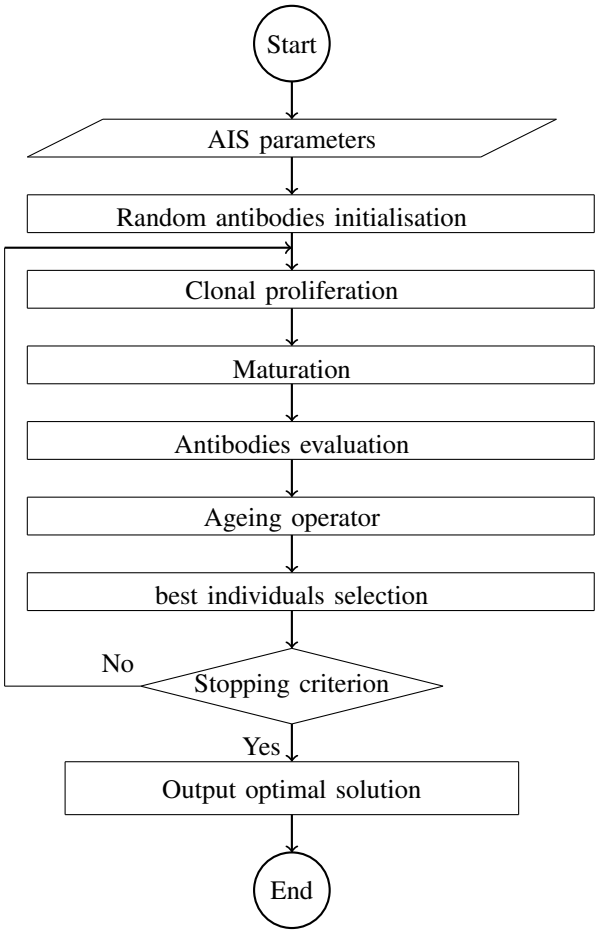

Fig. 3. AIS diagram

the implication between the fitness function and the affinity maturation in the natural system.

1) Initialisation: A random population N is generated in the search space, as a process that used in the other heuristic algorithms. This population considered like antibodies.

2) Clonal proliferation: In this step, the antibodies will clone (proliferate) according to their fitness(affinity).

3) Maturation: The maturation technic is similar to the mutation one with a mutation probability P. this mutation is applied to equation (1) :

$$x_{id} = x_{id} + k(x_{d_{max}} - x_{d_{min}}).N(0,1) \qquad (1)$$

where x_{id} represent the dimension d of the antibody i, x_{max} and x_{min} represent the min and the max bounds of the variable i, N(0,1) is the standard distribution and k is the scale factor.

4) Evaluation: In the evaluation step, the fitness function of every antibody is calculated by computing the affinity values.

5) Ageing operator: This factor is used to eliminate the individuals lost more. Indeed, the Ageing operator leads to upgrading the initial population.

E. AIS pseudocode

REFERENCES

[1] M. N. Collazo, C. Cotta, and A. J. Fernández-Leiva. Virtual player design using self-learning via competitive coevolutionary algorithms. *Natural Computing*, 13(2):131–144, 2014.

[2] D. Dasgupta, Z. Ji, and F. Gonzalez. Artificial immune system (ais) research in the last five years. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 1, pages 123–130. IEEE, 2003.

[3] A. Fernández-Ares, P. García-Sánchez, A. M. Mora, P. A. Castillo, J. J. M. Guervós, D. Camacho, M. Gómez-Martín, and P. González-Calero. Designing competitive bots for a real time strategy game using genetic programming. In *CoSECivi*, pages 159–172, 2014.

[4] A. Fernández-Ares, P. García-Sánchez, A. M. Mora, and J. Merelo. Adaptive bots for real-time strategy games via map characterization. In *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*, pages 417–721. IEEE, 2012.

[5] A. Fernández-Ares, A. M. Mora, J. J. Merelo, P. García-Sánchez, and C. Fernandes. Optimizing player behavior in a real-time strategy game using evolutionary algorithms. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 2017–2024. IEEE, 2011.

[6] A. R. Jordehi. A chaotic artificial immune system optimisation algorithm for solving global continuous optimisation problems. *Neural Computing and Applications*, 26(4):827–833, 2015.

[7] J. Kim and P. J. Bentley. An evaluation of negative selection in an artificial immune system for network intrusion detection. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, pages 1330–1337. Morgan Kaufmann Publishers Inc., 2001.

[8] R. Lara-Cabrera, C. Cotta, and A. J. Fernández-Leiva. A review of computational intelligence in rts games. In *Foundations of Computational Intelligence (FOCI), 2013 IEEE Symposium on*, pages 114–121. IEEE, 2013.