Artificial Intelligence and Its Applications

Ashish Kumar[1], Ansh Jhawar[2], Deepak Bisht[3], Pius Alex[4]

[1,2,3,4]*B. tech Student, Department of Computer Science & Engineering, Dronacharya Group of Institution, Greater Noida, India*

Abstract - **We are living in a time where Artificial Intelligence has already made a strong hold in our day-to-day life. Its applications include manufacturing done in industries, weather forecasting, gameplays, robots and almost all the machines that we are using.**

Future without Artificial Intelligence will make our lives difficult, and we cannot think of a bright future without it. Some people argue of the fact that "How Artificial Intelligence can be dangerous for our future?". Computer Engineers are working hard in this field as they think that with Artificial Intelligence can be made easy and it can the greatest invention of Human Mankind if it is a Success. So, this paper discusses some of the most interesting challenges to which the games research community members may face in the area of the application of artificial or computational intelligence techniques to the design and creation of video games.

Index Terms - **Artificial Intelligence, Friendly UI, Non-player characters, Procedural Content Generation, Affective Computing, Human-like behavior.**

INTRODUCTION

First we will give you brief introduction on "What is Artificial Intelligence?"

It is the study and development of Intelligent machines which have the power and capabilities to take decisions for themselves according to what has been programed in them in different situations. It is the study and development of machines which are intelligent and programming software for them so that they can learn, reason collect knowledge, communicate, manipulate and perceive the objects. Studies made on computation makes it possible to perceive and act.

What makes Artificial Intelligence machines smarter because its emphasis on perception, reasoning and action. It makes use of artificial neurons and scientific theorems to make its actions. Artificial Intelligence is more permanent, less expensive, consistent, has more ease of duplication and dissemination can perform some tasks much faster and better than humans, all these points give artificial intelligence an upper hand on natural Intelligence.

The various techniques that are used in Artificial Intelligence are Evolutionary computing, Neural Network, Fuzzy Logic and Hybrid Artificial Intelligence. It has a wide range of fields where it is applied like Speech Understanding (Google assistant, Siri, Cortana), scene recognition, Intelligent Computer Aided Instruction, Neural Computing, Robotic and Sensory Types, Computer Vision and Intelligent Computer Aided Instructions, Games AI.

AREAS OF ARTIFICIAL INTELLIGENCE

A. Friendly AI

It would be a very good thing if humanity knew how to choose into existence a powerful optimization process with a particular target. Or in more colloquial terms, it would be nice if we knew how to build a nice AI.

To describe the field of knowledge needed to address that challenge, I have proposed the term "Friendly AI." In addition to referring to a body of technique, "Friendly AI" might also refer to the product of technique—an AI created with specified motivations. When I use the term Friendly in either sense, I capitalize it to avoid confusion with the intuitive sense of "friendly."

One common reaction I encounter is for people to immediately declare that Friendly AI is an impossibility, because any sufficiently powerful AI will be able to modify its own source code to break any constraints placed upon it.

The first flaw you should notice is a Giant Cheesecake Fallacy. Any AI with free access to its own source would, in principle, possess the ability to modify its own source code in a way that changed the AI's optimization target. This does not imply the AI has the motive to change its own motives. I would not knowingly swallow a pill that made me enjoy committing murder, because currently I prefer that my fellow humans not die. But what if I try to modify myself, and make a mistake? When computer engineers prove a chip valid—a good idea if the chip has 155 million transistors and you can't issue a patch afterward—the engineers use human-guided, machine-verified formal proof. The glorious thing about formal mathematical proof, is that a proof of ten billion steps is just as reliable as a proof of ten steps. But human beings are not trustworthy to peer over a purported proof of ten billion steps; we have too high a chance of missing an error. And present-day theorem-proving techniques are not smart enough to design and prove an entire computer chip on their own—current algorithms undergo an exponential explosion in the search space. Human mathematicians can prove theorems far more complex than modern theorem-proves can handle, without being defeated by exponential explosion. But human mathematics is informal and unreliable; occasionally someone discovers a flaw in a previously accepted informal proof. The upshot is that human engineers guide a theorem-prove through the intermediate steps of a proof.

The human chooses the next lemma, and a complex theorem-prove generates a formal proof, and a simple verifier checks the steps. That's how modern engineers build reliable machinery with 155 million interdependent parts.

Proving a computer chip correct requires a synergy of human intelligence and computer algorithms, as currently neither suffices on its own. Perhaps a true AI could use a similar combination of abilities when modifying its own code—would have both the capability to invent large designs without being defeated by exponential explosion, and also the ability to verify its steps with extreme reliability. That is one way a true AI might remain know ably stable in its goals, even after carrying out a large number of self-modifications.

This paper will not explore the above idea in detail. (Though see Schmidhuber (2007) for a related notion.) But one ought to think about a challenge, and study it in the best available technical detail, before declaring it impossible—especially if great stakes depend upon the answer. It is disrespectful to human ingenuity to declare a challenge unsolvable without taking a close look and exercising creativity. It is an enormously strong statement to say that you cannot do a thing—that you cannot build a heavier-than-air flying machine, that you cannot get useful energy from nuclear reactions, that you cannot fly to the Moon. Such statements are universal generalizations, quantified over every single approach that anyone ever has or ever will think up for solving the problem. It only takes a single counterexample to falsify a universal quantifier. The statement that Friendly (or friendly) AI is theoretically impossible, dares to quantify over every possible mind design and every possible optimization process—including human beings, who are also minds, some of whom are nice and wish they were nicer. At this point there are any number of vaguely plausible reasons why Friendly AI might be humanly impossible, and it is still more likely that the problem is solvable but no one will get around to solving it in time.

But one should not so quickly write off the challenge, especially considering the stakes.

B. Learning and adaptive systems: The ability to adapt behavior bagged on previous experience, and to develop general rules concerning the world based on such experience.

B.1 Concept Formation

B.2 Cybernetics

C. Problem solving: Ability to formulate a problem in a suitable representation, to plan for its solution and to know when new information is needed and how to obtain it.

C.1 Automatic Program Writing

C.2 Heuristic Search

C.3 Inference (Resolution-Based Theorem Proving, Plausible Inference and Inductive Inference)

C.4 Interactive Problem Solving

D. Robots: A combination of most or all of the above abilities with the ability to move over terrain and manipulate objects.

D.1 Industrial Automation (e.g., Process Control, Assembly Tasks, Executive Tasks)

D.2 Other (Agriculture, Fishing, Mining, Sanitation, Construction, etc.)

D.3 Exploration

D.4 Household

D.5 Military

D.6 Transportation/Navigation

D.7 Security

E. Games: The ability to accept a formal set of rules for games and to translate these rules into a representation or structure which allows problem-solving and learning abilities to be used in reaching an adequate level of performance.

Research in artificial intelligence may take advantage of the wide variety of problems that videogames offer, such as adversarial planning, real-time reactive behaviors and planning, and decision making under uncertainty. For instance, real-time strategy games, which are a portion of the whole videogames, are being used as test beds and frameworks for brand new artificial intelligence techniques, as stated by our previous study on real-time strategy and artificial intelligence.

This paper aims at some interesting trends that seems to guide the future of videogames, and the challenges that they offer to academia, focusing on the application of artificial intelligence and, more precisely, computational intelligence, i.e. bio-inspired optimization techniques and meta-heuristics. We want to clarify that the universe of uses of optimization techniques on the development and game design is extremely broad and we do not pretend to make an exhaustive tour on it in this paper; in fact we recommend the interested reader a reading of other papers that have been published in the literature and serve as a basis to learn about the state of the art. We focused, instead, on certain research areas that will influence significantly in the creation of commercial games over the next decade, we refer to the procedural content generation, affective computing, which has an impact in player satisfaction and the creation of behaviors or strategies of decision making for non-playable characters (NPC).

APPLICATIONS OF ARTIFICIAL INTELLIGENCE

A. Procedural Content Generation:

Procedural content generation refers to the creation of videogame content through algorithmic means, such as levels, textures, characters, rules and missions (the content may be essential to game mechanics or optional). The fact of procedurally generate the content may reduce the expenses of hiring many designers in order to create the content manually and even being a source of inspiration for them, suggesting novel designs. Moreover, it is possible to establish some criteria the generated content must meet, such as adjusting the created level to the player's game style in order to offer a continuous challenge to her. If the generation process is made in real time and the content is diverse enough then it may be possible to create real infinite games, which offers the player a brand new gaming experience every time she starts a new one. These benefits are well-known by the industry, as exposed by the use of these kind of techniques in successful commercial games such as No man Sky, Temple Run, Skyrim And most popular Minecraft.

Many distinctions may be drawn when dealing with procedural content generation and its procedures. Regarding when the content is generated, it might be during the execution of the game (online generation) or during the development (offline generation). If we speak about the main objective of the generated content, this could be necessary for the game progression, hence it is mandatory to ensure that the content is valid, or it should be optional, as the decoration of levels.

Another question is the nature of the generation algorithm, that is, if we have a purely stochastic algorithm, in which content is created from random seed,or, conversely, a deterministic algorithm, where the content is generated by a parameter vector. The third possibility is the hybridization of both perspectives, designing an algorithm with a stochastic and a deterministic component working together. Looking at the objectives to be met, the creation process can be done in a constructive manner, ensuring the validity of the content throughout the process.

The other option is to follow a scheme of generating and testing, where a lot of content, which goes through a validation phase and subsequent disposal of that which does not comply with the restrictions, is generated. The latter scheme is the most currently employed by the research community, and it is based on the search of the content in the space of possible solutions. The validation is done by assigning values to content so that its level of quality is quantified according to the objectives. Apart from maps and levels, there are other examples of content that may be generated procedurally such as music, stories for role-playing games, game rules and quests.

These techniques are commonly used to generate maps and levels, as evidenced by the large number of papers devoted to this issue. For instance, authors of approach the problem of matchmaking in multiplayer videogames evolving maps for a first-person shooter in order to improve the game balancing for certain combinations of player skills and strategies. With a similar content type, authors of presented a genetic algorithm for the generation of levels for the Angry Birds game whose objective is to minimize the elements' movement during a period of time, obtaining stable structures during the process.

B. Behaviors of NPC (Non-player characters):

Traditionally the Artificial Intelligence (AI) of a game has been coded manually using predefined sets of rules leading to behaviors often encompassed within the so-called artificial stupidity, which results in a set of known problems such as the feeling of unreality, the occurrence of abnormal behaviors in unexpected situations, or the existence of predictable behaviors, just to name a few. Advanced techniques are currently used to solve these problems and achieve NPCs with rational behavior that takes logical decisions in the same way as a human player. The main advantage is that these techniques perform automatically the search and optimization process to find these smart strategies. Bio-inspired algorithms are the basis of many of these advanced methods, as they are a suitable approach in this regard, because they are able to produce solutions of great complexity as an emerging result of the optimization process, and its adaptive capacity allows them to incorporate information provided by the user. Due to this, there are several successful proposals that follows this approach.

For instance, co-evolution is one of the heuristic techniques inspired by the natural evolution principles that has been widely used in videogame AI programming. In the author presents a research that were capable of evolving the morphology and behavior of virtual creatures through competitive coevolution that interact in a predator/prey environment. Other interesting papers used co-evolution to obtain game strategies for artificial players of a war game called Tempo. Machine learning is used as well when modeling the behavior of artificial players. Authors of have used self-organizing maps in order to improve the maneuvering of platoons in a real-time strategy game. By analyzing data obtained from the sensors, the authors of have developed an algorithm for an artificial pilot so it is able to learn the racetrack and drive through it autonomously. The authors obtained several features from the maps of a real-time strategy game and use them to determine a NPC's behavior.

NPC Movement Using Path-Finding:

Artificial intelligence computer game must provide a way for a non-playing character to move throughout the game world. For example, When then player is on one side of the building and the monster is on the other, through which path through the building the monster will reach the player? This is the NPC movement problem.AI Search Methods are used to find the path in computer games. A* algorithm is the most widely used for path negotiation because of its flexibility and also because it determine the shortest path between two points. Typical A* algorithms have three main attributes, fitness, goal, and heuristic or f, g, and h respectively. g is the cost to travel from the start node to some node between the goal. h is the heuristic or estimated cost to get from this node to the goal. f is the sum of g and h, or the total estimated cost of a path going through this node. The A* algorithm also maintains an Open list of the nodes that have not been explored yet and a Closed list of nodes that have been explored. The following is pseudo code for the A* algorithm.

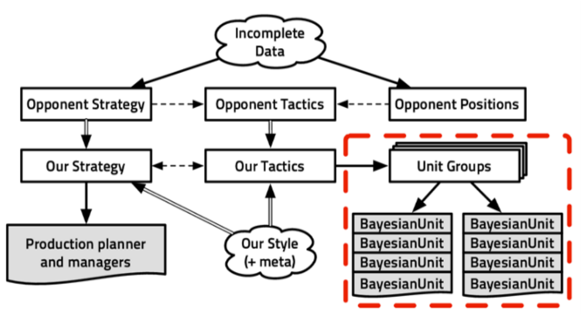
```
1. Let P = the starting point.
2. Assign f, g, and h values to P.
3. Add P to the Open list. At this point P is the only node on the Open list.
4. Let B = the best node from the Open list (best node has the lowest f-value).
   a. If B is the goal node, then quit. A path has been found.
   b. If the Open list is empty, then quit. A path has been found.
5. Let C = a valid node connected to B.
   a. Assign f, g, and h values to C.
   b. Check whether C is on the Open and Closed list.
      I. If so, check whether the new path is more efficient (lower f-value).
         1. If so, update path. ii. Else, add C to open list.
   c. Repeat step 5 for all valid children of B.
6. Move B from the Open list to the closed list and repeat.
From step 4.
```

NPC Decision Making Using Bayesian Networks:

In the previous example of the monster negotiating a path to the player, a different problem must be solved first before negotiating the path. The problem is does the monster even knows the player is present in the building? If the game designers give the full information of the game world to the non-playing character then there would be no fun in playing the game. This is an example of NPC Decision making. In this AI is needed to make the non-playing character to act in a human like way. When the player enters the building from the other side, the monster will be unaware of the presence of the player because of the wall between them. If the player enters causing a noise disturbance, then the monster will sense the player and will start negotiating the shortest path as discussed in the NPC movement using path finding. One AI technique that is used to implement this is a Bayesian Network. It helps NPC to perform complex reasoning in a human like fashion. In this the computer calculates the probability of the monster sensing the player if the player has entered the building. This expression can be written as;

P (B|A) = P (B|A) P (A) / P (B)

Where P (B|A) is the probability that the monster would sense the player if the player had actually tripped. and P (A) is the probability of the monster sensing the player. and P (B) is the probability of the player tripping.



CONCLUSION

The challenges in the research lines that we have mentioned throughout this paper are huge and certainly affect other areas beyond the field of video games.

For instance, the generation of quasi-human behavior is something that is being already investigated and traditionally have their seed in the famous Turing test. The possibilities opened up by applying science to videogames are vast: the integration of feelings in artificial players and the option to build a direct channel between them and the sentimental perception of the player through the so-called Affective Computing.

Regarding procedural content generation, it has been shown that is a hot field in academia, with a large number of papers related to it. Moreover, the videogame industry is successfully using many of the advances obtained by academia, although there are many non-tackled challenges in this sense.

We end this paper mentioning that there are many areas related to the use of Computational/Artificial Intelligence that have not been specifically described here, where research might find additional challenges such as player modeling, computational narrative and AI-assisted game design among others. We are dealing with stimulating challenges not only for the near future, but the present.

While scientists have not yet realized the full potential and ability of artificial intelligence. This technology and its applications will likely have far-reaching effects on human life in the years to come.

REFERENCES

[1] Charles Weddle, Graduate Student, Florida State University “Artificial Intelligence and Computer Games”, unpublished.

[2] Association, E.S.: Essential facts about the computer and video game industry (2015).

[3] F. D. Laramee, Genetic Algorithms: Evolving the Perfect Troll, AI Game Programming Wisdom, Charles River Media, Inc., Hingham, MA, 2002.

[4] J. Matthews, Basic A* Path finding Made Simple, AI Game Programming Wisdom, Charles River Media, Inc., Hingham, MA, 2002.

[5] Buro, M.: ORTS: A hack-free RTS game environment. In: Schae_er, J., etal. (eds.) Computers and Games. Lecture Notes in Computer Science, vol. 2883, pp. 280_291.

[6] 6. Avery, P., et al.: Coevolving a computer player for resource allocation games: using the game of Tempo as a test space. Ph.D. thesis, School of Computer Science University of Adelaide (2008).

[7] Freeman, D.: Creating emotion in games: The craft and art of emotioneering& trade. Compute. Entertain. 2(3), 15_15 (Jul 2004).

[8] Ferreira, L., Toledo, C.: A search-based approach for generating angry birds’ levels. In:

Computational Intelligence and Games (CIG), 2014 IEEE Conference on. pp. 1_8. IEEE (2014).

[9] Fernández-Ares, A., García-Sánchez, P., Mora, A.M., Merelo, J.: Adaptive bots for real-time strategy games via map characterization. In: Computational Intelligence and Games (CIG), 2012 IEEE Conference on. pp. 417_721. IEEE (2012).17

[10] Collins, K.: An introduction to procedural music in video games. Contemporary Music Review 28(1), 5_15 (2009).

[11] Avery, P.M., Michalewicz, Z.: Static experts and dynamic enemies in coevolutionary games. In: IEEE Congress on Evolutionary Computation. pp. 4035_4042 (2007).

[12] Browne, C., Togelius, J., Sturtevant, N.R.: Guest editorial: General games. IEEE Trans. Comput. Intellig. and AI in Games 6(4), 317_319 (2014).

[13] Font, J., Mahlmann, T., Manrique, D., Togelius, J.: A card game description language. In: Esparcia-Alcázar, A. (ed.) Applications of Evolutionary Computation, Lecture Notes in Computer Science, vol. 7835, pp. 254_263. Springer Berlin Heidelberg (2013).

[14] Yannakakis, G.N., Hallam, J.: Real-time game adaptation for optimizing player satisfaction. Computational Intelligence and AI in Games, IEEE Transactions on 1(2), 121_133 (2009)