State-of-the-Art and Open Challenges in RTS Game-AI and Starcraft

Khan Adil, Feng Jiang, Shaohui Liu*, Worku Jifara
School of Computer Science and Technology
Harbin Institute of Technology
NO. 92, Xidazhi Street, Harbin,
Heilongjiang, China

Zhihong Tian
The Cyberspace Institute of Advanced Technology
Guangzhou University
Guangzhou, Guang Dong, China

Yunsheng Fu
The Institute of Computer Applications China Academy of Engineering Physics
Mianyang 621900, China

***Abstract*—This paper presents a review of artificial intelligence for different approaches used in real-time strategy games. Real-time strategy (RTS) based games are quick combat games in which the objective is to dominate and destroy the opposing enemy such as Rome-total war, Starcraft, the age of empires, and command & conquer, etc. In such games, each player needs to utilize resources efficiently, which includes managing different types of soldiers, units, equipment's, economic status, positions and the uncertainty during the combat in real time. Now the best human players face difficulty in defeating the best RTS games due to the recent success and advancement of deep mind technologies. In this paper, we explain state-of-the-art and challenges in artificial intelligence (AI) for RTS games and Starcraft, describing problems and issues carried out by RTS based games with some solutions that are addressed to them. Finally, we conclude by emphasizing on game 'CIG & AIIDE' competitions along with open research problems and questions in the context of RTS Game-AI, where some of the problems and challenges are mostly considered improved and solved but yet some are open for further research.**

Keywords*—*Real Time Strategy (RTS); Game-AI; Starcraft; MMOG; AIIDE; CIG; MOBA

I. INTRODUCTION

The presence of a good artificial intelligence (AI) technology in the background of a game is one of the most key elements of the entertainment and restate ability of profitable computer games [1]. Although 'AI' is being applied successfully in a variety of games such as chess, backgammon or Checkers but real-time strategy decisions are expected to be made in games that do not seem too common and easy because predefined setups are usually used to simulate them that results in large search space and short real AI for learning [2]. So still some traditional planning approaches continue working in the real-time sense of games. For a decade or so, the game industry is improving and now has introduced some outstanding approaches like MMOG [1] (Massively Multiplayer Online Game) which is an online game-play capable of supporting a large number of players from hundreds to thousands concurrently in the same instance [3]. Similarly, another sub-genre is devised called MOBA (Multiplayer Online Battle Arena) which is also renown as an action real-time strategy (ARTS) in which a player controls a single character in one of two teams [4]. It involves defending a base alongside teammates on one side of a map while fighting the opposing side's units and structures. MOBA games are a mixture of action games, role-playing games, and real-time strategy games, in which players usually do not construct either buildings or units. Defense of the Ancients (DotA), a map-based 'Aeon of Strife' for Warcraft III: the reign of chaos[2], Starcraft- II[3] and the frozen throne was the first major titles of its genre and the first 'MOBA' for which sponsored tournaments were held. One of the latest 'MOBA' game is 'Middle-earth: the shadow of war' released in 2017 playable on 'PS4', windows and Xbox-one platforms.

II. REAL-TIME STRATEGY (RTS) GAMES

This paper emphasizes on real-time strategy (RTS) games. The term real-time strategy spawned from Dune II that multiplied and evolved to become a big cornerstone of the video game industry, especially when it comes to 'PC' gaming. RTS games take a lot of previous strategy game troops and mechanics, focusing on units, building and resource management, usually during times of warfare. For most games of the genre, one generally has to generate resources [2] and to use them for creating buildings and buildings that spawn additional units. The units usually consist of different builder or farmers and combat-oriented different types of soldiers [5]. Proper time management, intuitive use of forces and capabilities are compulsory, as one continuously tries upgrading and increasing bases and forces while putting pressure on the enemies, who are trying to do the same things. The real-time strategy genre has boomed with the inclusion of online multiplayer matchmaking games, like Warcraft-III and Starcraft that have spawned global tournaments.

RTS games are cornerstones of e-sports, as popular tournaments such as MLG[4] and GSL[5] cover them regularly, often times, having grand prizes going into the $100,000 + mark and into the millions based on the size, scope, and

[1] www.mmorpg.com

[2] Blizzard Entertainment: Warcraft III: blizzard.com/games/war3/, StarCraft II: blizzard.com/games/Sc2/

[3] Blizzard Entertainment: Warcraft III: blizzard.com/games/war3/, StarCraft II: blizzard.com/games/Sc2/

[4] http://www.majorleaguegaming.com/

[5] http://afreeca.tv/36840697

sponsors of the event. There are plenty of RTS games duking it out for a spot on the list in which some of them include 'The Lord of the Rings: the battle for middle-earth-II[6]', 'command & conquer: red alert III[7]', 'Age of empires-III[8]', 'Total war: shogun-II[9]', 'command & conquer: generals[10]', 'Empire earth-II[11]', 'Sins of a solar empire[12]', 'world in conflict[13]', 'Rise of nations[14]', 'Medieval II: total war[15], etc. Real-time strategy games have had a huge fan following since their inception. The 'USP' of such games is that there is no specific way to complete a mission. The players themselves have to devise strategies and plan out their approaches to ensure victory [6]. There are numerous RTS games out there that have received success, but Starcraft the most successful and popular is worthy to mention here.

III. Starcraft

Starcraft is probably the best competitive game of all times, surpassing every other game in conventional popularity. Its very first version was released in 1998 by Blizzard entertainment [2]. Starcraft is a science fiction based universe game introduces different combat teams like 'Protoss', 'Terran', and 'Zerg' [5]. A typical match GUI of an RTS game Starcraft is shown in Fig. 1.

A. Protoss

Protoss has access to powerful units, machinery and advanced technologies such as energy shields and localized warp capabilities, powered by their sonic traits. However, their forces have lengthy and expensive manufacturing processes, encouraging players to follow a quality strategy for their units over the quantity.

B. Terran

Terran lies between the two races, providing units that are versatile and flexible. They have access to a range of more ballistic military technologies and machinery, such as tanks and nuclear weapons.

C. Zerg

Zerg possess entirely organic units and structures, which can be produced quickly and at a far cheaper cost to resources, but are accordingly weaker, relying on sheer numbers and speed to overwhelm the enemies. A typical Starcraft 'GUI' on sweeping through the Zerg base is shown in Fig. 2. Although each race is unique in its composition, no race has an innate advantage over the other. Each species is balanced out so that they have different powers and abilities, but their overall strength is the same. The balance stays complete via infrequent patches provided by 'Blizzard'. Each race relies on two resources to sustain their game economies and to build their forces: minerals and vespene gas [2]. Minerals are needed for all units, structures and are obtained by using a worker unit to harvest the resource directly from mineral nodes scattered around the battlefield. Players require vespene gas to construct advanced units and buildings and acquire it by constructing a gas extraction, building on top of a geyser and using worker units to extract the gas from it. In addition, players need to regulate the supplies for their forces to ensure that they can construct the number of units they need.

Although the nature of the supply differs between the races, Protoss and Zerg building construction is limited to specific locations: Protoss buildings need to be linked to a power grid while almost every Zerg structure must be placed on a carpet of biomass, called 'creep', that is produced by certain structures [7]. Terrain buildings are far less limited, with certain primary base structures possessing the ability to take off and fly slowly to new locations. Multiplayer in Starcraft is powered by Blizzard-entertainment's battle.net internet service [8]. Through this, a maximum of eight players can compete in a variety of game modes, including simply destroying all other players on a level.

We studied other review papers in the same field that were either found outdated and specific to Starcraft competitions or to tactical and strategic techniques only, that is why we thought to write a review paper in detail with fresh, updated information and literature, so this paper aims to provide a perfect guide to the current and recent past research challenges and state of the arts in RTS games and Starcraft. It is organized as follows. Section IV explains state-of-the-art and challenges, details of recent research and inadequate methods, an RTS game strategy and an RTS game decision tree. Section V includes open research areas and problems. Section VI presents a brief introduction of open computational intelligence and game AI competitions. Finally, Section VII concludes the paper with discussions.

IV. State-of-the-Art and Challenges

For a decade, tremendous goals have been achieved and different contributions are made into computer games, particularly in real-time strategy games. But despite achieving goals and contributions though still, some challenges exist which are reviewed here along with some state-of-the-art methods and techniques.

Some prevailing efforts on AI for real-time strategy games are achieved in [5], particularly the efforts about the Starcraft which has appeared in the recent past as an incorporated test bed for such type of research area, and specifically AI problems faced by RTS games and the solutions that have been proposed to address them are overviewed.

Moreover, a summary of the results of some Starcraft 'AI' competitions is presented and the architectures used by the participants are examined. Besides, different challenges in the context of 'RTS game-AI' are also highlighted in which some of them are partially solved but yet some are open to further improvement and research.

[6] http://www2.ea.com/lotr-the-battle-for-middle-earth-2
[7] http://www2.ea.com/command-and-conquer-red-alert3
[8] https://www.ageofempires.com/games/aoeiii/
[9] https://www.totalwar.com/shogun2
[10] www.cncgeneralsworld.com
[11] www.ee2.eu
[12] https://www.sinsofasolarempire.com/
[13] http://worldinconflict.us.ubi.com/
[14] http://ron.heavengames.com/
[15] http://medieval2.heavengames.com/

Fig. 1. A typical match of Starcraft (workers are gathering resources for the remote building) [2].

Fig. 2. A typical Starcraft 'GUI' on sweeping through the Zerg base.

Real-time strategy video games have confirmed to be a real thought provoking area for studying artificial intelligence [9] but Current 'AI' results are inadequate by wide action and state spaces and real-time decisions. Most applications proficiently challenge different strategic or tactical sub-complications, however, there is no particular algorithm quick enough to be effectively functional to large challenges in RTS games. Hierarchical adversarial search structures are also considered which implement a different human perception at each level, from determining how to win the game at the top of the tree to different unit orders at the bottom most.

In addition, well thought out scenarios from the real-time strategy game Starcraft are considered in [10] as the new standard of reinforcement learning algorithms where micromanagement tasks are observed for the problems of low-level control of army members during the battle. From a reinforcement learning perspective, such scenarios are challenging because of the large state-action spaces, where there is no clear feature representation to evaluate state-action functions.

Further, The approaches to deal the micromanagement states with deep neural network controllers are provided by the game engines from raw state features [11].

Some heuristic reinforcement learning state-of-the-art algorithms that combine direct exploration in the policy space and backpropagation is also applied which allow collecting traces for learning using deterministic policies that seem more effective than $\epsilon -$greedy exploration. Experiments proved that non-trivial strategies are learned successfully with such algorithm for scenarios where both Q-learning and reinforcement learning struggle with armies of up to 15 agents that also suggest that well balanced and smart decision making at the tactical level is essential for intelligent agents to do fine in the field of real time strategy games [12].

Additionally, the Bayesian model is used for predicting the outcomes of isolated battles and to predict what units are needed to defeat a certain army.

Simulated battles are also used to train the model in order for reducing the dependency on player expertise. The model is applied to the game of Starcraft with the specific goal of using the predictor as a module for making high-level battle decisions and to assure that the model is capable of making precise predictions.

In ideal settings, still building robust AI systems are challenging due to massive action and state spaces and insufficient decent state evaluation functions with high-level action perceptions. So far, skillful human players are yet conveniently crushing the finest RTS game-AI systems, however, this may stop happening in the near future because of the latest accomplishment of deep convolutional neural networks in ‘Alpha-Go’, that proved how networks can be applied for evaluating complex game states precisely and to emphasis look ahead searches [13].

Convolutional neural networks for RTS game state evaluation that goes beyond commonly used material based evaluations by taking spatial relations between units into account that assess the CNN’s performance by matching it with several other assessment functions via tournaments played among various state-of-the-art search algorithms, however still despite its slow evaluation speed, the CNN based search performs is suggestively better related to simpler but quicker evaluations.

In short, over such encouraging preliminary outcomes together with the latest improvement in hierarchical search recommends that controlling human players in RTS games may not be far away.

Exploration is a vital part of play in recent video games [14]. It brings up to the discovery based events, in which players explore mechanisms, as well as specialties of the virtual world. Games with exploration maps are growing in gaming societies because spatial exploration is essential to play in real time strategy games and role playing games. To discover behavior patterns and understand gamer styles, the game-playing behavior of human players in exploration games needs to be investigated in order to help in designing and developing believable agents.

An experiment is conducted in [14] where 25 participants played three types of exploration games: in-game data, think-aloud data, questionnaire responses and post-game interview data. The data from all these exploration games were collected to achieve a deeper understanding of exploration preferences.

Further, thematic analysis was used to analyze data and map out four game exploration archetypes: “wanderers”, “seers”, “pathers”, and “targeters”. The behavioral traits of these four archetypes were also investigated by conducting an analysis of the four highlighted aspects: conception, strategic, hesitation, and reasoning.

Real-time strategy games are realistic with dynamic and time constraints game-playing by abandoning the turn-based rule of its ancestors [6]. Playing with and against computer-controlled players are a pervasive phenomenon in RTS games because of the convenience and the preference of groups of players. Therefore, better game playing agents are able to improve the game playing experience by acting as an intelligent adversary or traitors.

In terms of the economic expansion and tactical battlefield arrangement aspects, one way of enhancing game-playing performance of the agents is to understand the game environment. Such issues are addressed directly in accessing game maps and extracting strategic features of the traditional commercial RTS game-playing agents because human players are unable to access the same information which is a form of cheating ‘AI’, where it has been known to negatively affect player experiences. Thus, a scouting mechanism for RTS game-playing agents is developed in order to enable game units to explore game environments automatically in a realistic fashion [10]. Such research can be grounded in the prior robotic exploration work by which a hierarchical multi-criterion decision-making strategy could be presented to address the incomplete information problems in RTS settings.

Reinforcement learning algorithms with the generalized reward function are proposed in [15]. In the proposed method ‘Q-learning’ and ‘SARSA’ algorithms are used with generalized reward functions to train the reinforcement learning agent. The performance of the proposed algorithms is evaluated on a real-time strategy game called ‘Battle City’. There are two key benefits of having such an approach as compared to other works in an RTS.

1) The concept of the simulator could be ignored, which is often game specific and is usually hardcoded in any type of RTS games.

2) The proposed system can learn from interacting with any opponents and quickly change the strategy, according to the opponents and do not require any human intervention as used in prior practices or works.

The first deep learning model to effectively learn control policies directly from high-dimensional ‘sensory input’ [16] by means of reinforcement learning is suggested in [17]. The model is a convolutional neural network, trained with a variant of Q-learning whose input is raw pixels and whose output is a value function estimating future rewards. The proposed method is applied to seven ‘Atari’ 2600 games from the ‘Arcade’ learning environment, with no amendment of the architecture or learning algorithm. Where It outperformed all prior methodologies on six of the games and exceeded a human professional on three of them.

Fictitious play is a good renown imaginary model for learning in games, but, it has got minor considerations in applied applications to large problems. Two variants of fictitious play are implemented in behavioral strategies of an extensive-form game. The first variant is a full-width process that is realization equivalent to its normal-form counterpart and therefore receives its convergence assurances. But, its computational requirements are undeviating in time and space rather than exponentially.

The second variant, 'Fictitious Self Play', is a machine learning framework which implements fictitious play in a sample-based fashion.

The approaches in [10] compare experiments in imperfect information poker games, and show their convergence to estimate 'Nash equilibria'.

The problem of learning probabilistic models of high-level strategic behavior in the real-time strategy game Starcraft is studied in [18]. The models are automatically well-trained from collections of game records and intended to grab the mutual tactical states and decision points that come up in those games. Unlike most work in behavior, learning and predicting in RTS games, the data-centric approach in [18] is not biased by or limited to any set of predetermined strategic conceptions.

Moreover, since the behavior model is based on the well-developed and generic paradigm of 'Hidden Markov' model, it provisions a range of uses for the design of AI players and human assistant, for example the well-learned models can be used to decide probabilistic expectations of a player's future action based on observations to simulate possible future trajectories of a player, or to recognize aberrant or unique strategies in a game database.

Further, the well-learned qualitative organization of the models can be assessed by humans in order to classify mutual strategic elements. This approach is demonstrated by learning models from 331 professional level games which delivered both a quantitative and qualitative evaluation of the learned model's utilities.

As real-time strategy games are considered robust to typical adversarial tree search methods. So in recent times, a few methodologies to challenge the difficulty of such RTS games have appeared that use game state or move abstractions or both. Unfortunately, the primary experiments were limited to simpler RTS environments or lack testing against state-of-the-art game playing agents.

A new adversarial search framework based on scripts is proposed in [19] that can expose choice points to look-ahead search procedure. Where to choose a combination of a script and decisions for its choice points signifies a move to be performed next. Such moves can be performed in the actual game, thus letting the script play, or in an abstract representation of the game state that can be used by an adversarial tree search algorithm. Puppet search returns a prime variation of scripts and choices to be performed by the agent for the known time span. The algorithm is implemented in a complete Starcraft bot, where experiments showed that it matches or outperforms all of the individual scripts that it uses when playing against state-of-the-art bots from 2016 Starcraft 'AIIDE' competition.

A genetic algorithm to optimize the placement of the buildings in real-time strategy games is proposed in [20] where candidate solutions are evaluated by running base assaults simulations. The experimental results are presented in Sparcraft[16] using battle setups extracted from human and agent Starcraft games. The proposed system is able to turn base assaults that are losses for the defenders into wins, as well as decrease the number of surviving attackers. In short, performance is deeply dependent on the quality of the prediction of the attacking army composition used for training and its resemblance to the army used for assessment.

A. Algorithms, Techniques, and Methods Inadequate in RTS Game-AI

Almost all RTS games are based on 3D ground combat maps, with various terrain and obstacles adding interest to the battlefield. But they are different in some aspects from each other because of challenges in their frameworks and mechanism. Scalable search techniques based games are found to be fast in performing with short search space, but the optimal gameplay is hard to achieve in such games [9].

Similarly, with deep reinforcement learning, new architectures have been proposed in RTS games, but they suffer from deprived module integration and remain inadequate in performing as a unified architecture [21]. Different models are presented for managing micromanagement tasks like states and actions which could not show significant improvement and efficiency in units' movements [22].

In the same way, the different state-of-the-art policy learning algorithms are introduced with the passage of time, but none of them contributed significantly where there exists a space for improvement [23].

Deep 'CNN's' is used to evaluate RTS game states where games that focus on a single strategic component like 'combat' lacking spatial reasoning abilities, ignoring information such as unit position, terrain and other states [13].

Different game engines have developed that help in training models using the supervised learning and extracting metadata from games played during tournaments, but not found useful in reinforcement learning [24].

Behavioral models based on 'HMM' are presented where they are relatively found to be simple in behavior in terms of both observations and transitions [18], though some games are found to be the best test-beds for 'AI' experiments such as Starcraft that outperforms others in this regard [2].

It is vital to talk about different approaches or styles of players towards their game exploration which is a healthy open research area and where still space exists for more map.

Exploration styles to be introduced for the better and successful future of RTS Games, further, it would help in designing and developing efficient game agents as still most of the agents are struggling with the issue of partial observability and uncertainty [14].

Flocking and 'SOM' with experimental analysis are applied in RTS games that shown decidedly improvement in the previously experienced behavior of groups but not enough

[16] SparCraft: code.google.com/p/sparcraft/

because still unintelligent decision making is observed in RTS games in real time, which asks the demand for better intelligent grouping techniques to be applied in the future RTS games [25].

Overall, it can be concluded that in the field of RTS games several models, approaches, algorithms and techniques suffer from issues and unsatisfactory results and outputs, where the attention of the research community is necessarily required to keep the RTS games genre or era alive and active.

B. A Common RTS Combat Game AI Strategy

One of the important aspects of winning the battle is to have an effective battle strategy [26]. In Fig. 3, a common RTS game strategy is shown in which the barracks and bases are surrounded by a guarded perimeter that is always guarded and monitored by guard team. The guard team monitors and observe whether there is an enemy close to the base. During the design, ‘worker units’ are focused because of their certain role in the battle’s outcome.

The units are required to act according to the strategy in finding and taking the remote resource to the base, if there is not any base near then it starts building one [27]. Attacking units normally find the target and attack, but sometimes ‘workers’ defend if they need to, besides the bases and barracks, producing workers and attackers to combat is unavoidable.

A strong defense supporting the rush strategy by introducing the guard team that guards the bases and waits to target until someone comes close. A percentage of guard team is assigned to attacking units called ‘Assaults’ to attack the enemy when they are required or when they exceed a specific size or number. During ‘units’ production, the ratio of guards and assault is always kept in balance. With the passage of time, new units are assigned to assault team as the force grows larger. At least one of the three attack units i.e. Light, heavy and ranged are to be built at a specific time [28].

With a decision tree, it becomes easy and simple adding new nodes, decisions, and actions. The process of mapping out actions and trees is to record the prototype. So a decision tree prototype is notable in Fig. 4, being one of the simplest, easiest to implement, but most importantly quickly adaptable.

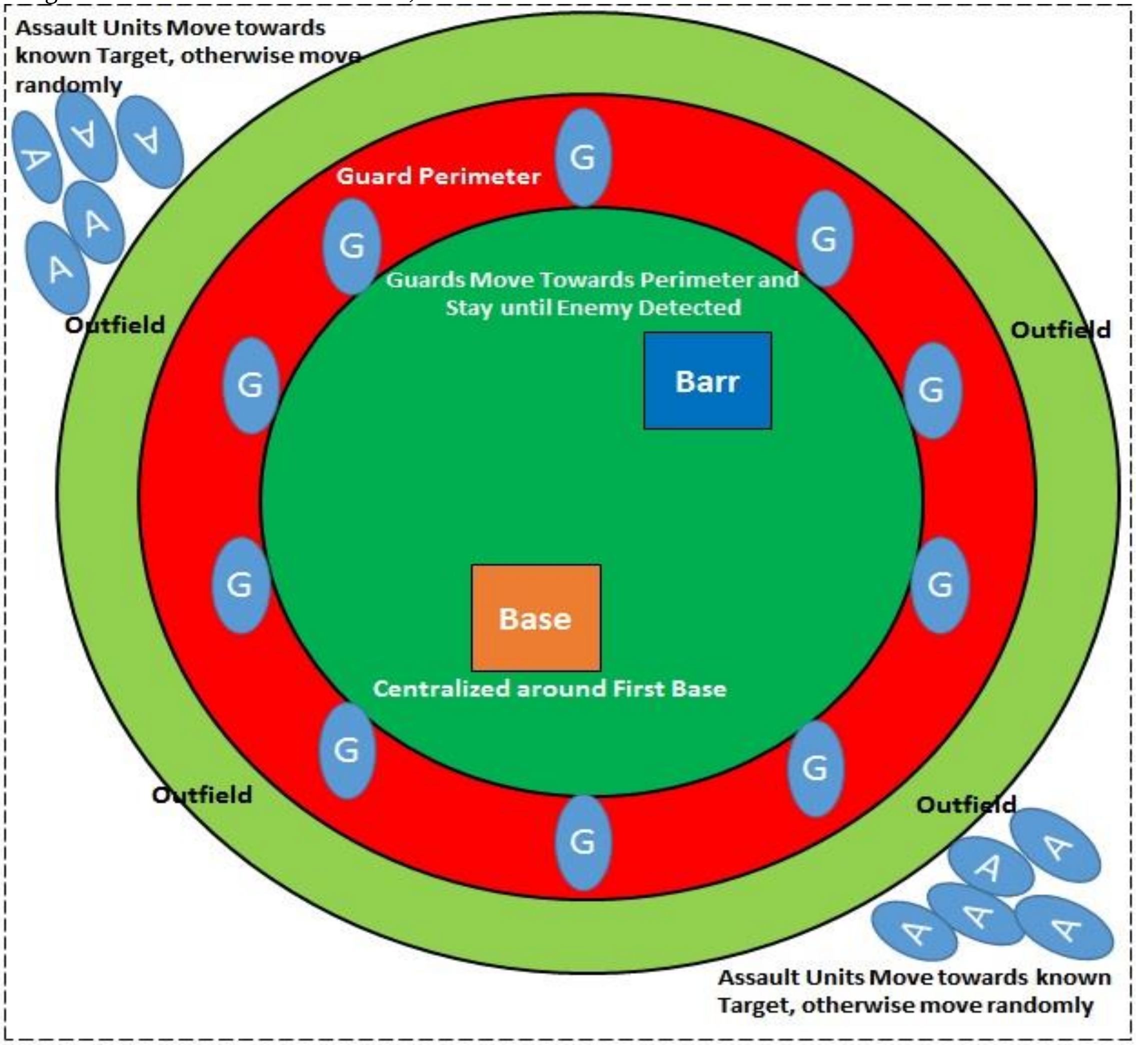

Assault Units Move towards known Target, otherwise move randomly
A
A
A
A
A
Guard Perimeter
G
Guards Move Towards Perimeter and Stay until Enemy Detected
Outfield
Outfield
Barr
Base
Centralized around First Base
Outfield
Outfield
A
A
A
A
A
Assault Units Move towards known Target, otherwise move randomly

Fig. 3. A common RTS combat game-AI strategy [28].

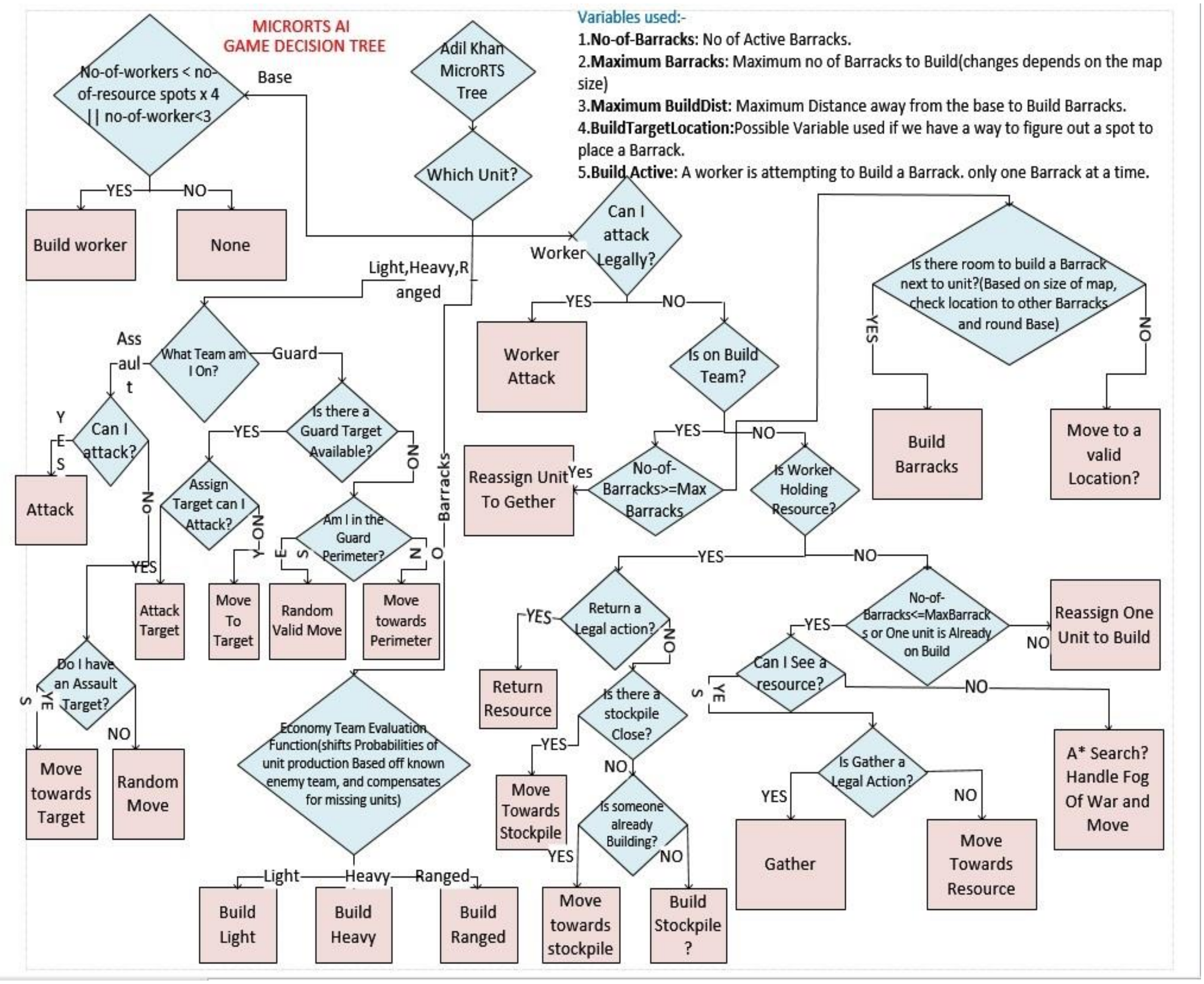
C. An RTS Combat Game Decision Tree

A common RTS combat game decision tree developed from the script of MicroRTS is given below in Fig. 4. The tree consists of roughly 4 mini trees: Barracks, base, attack unit, and worker. Each of these trees controls the specific type of unit. The base and barracks trees are roughly equivalent. The base simply checks if a worker cap has reached and to decide whether workers are to build [29].

The barrack is a little different, in so that the barrack first checks to see if the system will let build a particular unit and if so uses a random selection using weights. Each unit type has a weight associated with it in the knowledge base, and if the randomizer selects the range for that weight it will build that unit. For the attacking unit tree, it checks to see what team the unit is on- assault or guard. If it is an assault unit, it sees if it can attack an enemy, and if it can't it tries to move towards one if it has a target. If it has no target, it randomly moves. If it is a guard unit, it looks to see if there is a target and if there is, it moves and attack it. If a guard unit does not have a target, they move towards the guarded perimeter and randomly move inside the perimeter [30].

The worker has the most complex tree. It first checks to see if it needs to defend the base; if so, then it defends. Otherwise, it checks if it's in the build or gathers team. If the unit is on the build team, it checks to see if there are not enough barracks, and if not, find space to build one (using a BFS) and builds one. Once the barracks cap has been reached it is reassigned to the gathered team. The gather team first checks to see if the unit is holding a resource. If it is holding a resource, then it checks if a base is close enough to deposit it, and if it is, it will, otherwise it builds a new base next to the resources [31]. If it doesn't hold resources, it tries to find the closest resources (using a BFS) and move towards them.

Unfortunately, the breadth-first search on such resource finding does not provide an optimal macro level solution. It does not factor in the amount of the resource.


MICRORTS AI
GAME DECISION TREE
Adil Khan MicroRTS Tree
Which Unit?
Base
No-of-workers < no-of-resource spots x 4 || no-of-worker<3
YES
NO
Build worker
None
Worker
Can I attack Legally?
YES
NO
Worker Attack
Is on Build Team?
YES
NO
No-of-Barracks>=Max Barracks
Yes
Reassign Unit To Gether
Is there room to build a Barrack next to unit?(Based on size of map, check location to other Barracks and round Base)
YES
NO
Build Barracks
Move to a valid Location?
Is Worker Holding Resource?
YES
NO
Return a Legal action?
YES
NO
Return Resource
Is there a stockpile Close?
YES
NO
Move Towards Stockpile
Is someone already Building?
YES
NO
Move towards stockpile
Build Stockpile ?
No-of-Barracks<=MaxBarracks or One unit is Already on Build
YES
NO
Reassign One Unit to Build
Can I See a resource?
YES
NO
A* Search? Handle Fog Of War and Move
Is Gather a Legal Action?
YES
NO
Gather
Move Towards Resource
Light,Heavy,Ranged
What Team am I On?
Assault
Guard
Can I attack?
YES
No
Attack
YES
Do I have an Assault Target?
YES
NO
Move towards Target
Random Move
Is there a Guard Target Available?
YES
NO
Assign Target can I Attack?
YES
NO
Attack Target
Move To Target
Am I in the Guard Perimeter?
YES
NO
Random Valid Move
Move towards Perimeter
Barracks
Economy Team Evaluation Function(shifts Probabilities of unit production Based off known enemy team, and compensates for missing units)
Light
Heavy
Ranged
Build Light
Build Heavy
Build Ranged

Variables used:-

1.**No-of-Barracks:** No of Active Barracks.

2.**Maximum Barracks:** Maximum no of Barracks to Build(changes depends on the map size)

3.**Maximum BuildDist:** Maximum Distance away from the base to Build Barracks.

4.**BuildTargetLocation:**Possible Variable used if we have a way to figure out a spot to place a Barrack.

5.**Build Active:** A worker is attempting to Build a Barrack. only one Barrack at a time.

Fig. 4. A common RTS combat game decision tree developed from the script of MicroRTS [31].

V. Open Research Areas and Problems in RTS Game AI and Starcraft

A set of problems has been considered mostly solved, however, there are many other problems which are still open for research and need improvements. Some of them are as follows:

1) Strategic decision making in real-time domains is still an open problem [2].

2) A holistic approach, techniques that scale up to large RTS games as StarCraft [5].

3) How to achieve adaptive strategies that can recognize the opponent's intentions, and selects an adequate response [32].

4) Large-scale adversarial planning under real-time constraints is an open area for research [33].

5) Techniques developed for adversarial planning under the uncertainty of partially observable domains do not scale to RTS-games scenarios [33].

6) Integration of modules to make a unified architecture for RTS Games is an open problem for research [21].

7) Multi-agent reinforcement learning in RTS games, an active area for research and improvement [34].

8) Partial Observability [5].

9) Non- obvious quantification of state values [5].

*10)*The problem of featuring a dynamic and structured state [2].

*11)*How to exploit the massive amounts of existing domain knowledge (strategies, build-orders, replays and so on) [35].

*12)*Resource management [31].

*13)*Decision making under uncertainty [36].

*14)*Spatial and temporal reasoning [5].

*15)*Detecting the opponent's grouping behavior is a sub-problem of opponent modeling [32].

*16)*Collaboration (Between Multiple AIs) [2].

*17)*Opponent Modeling Learning [32].

*18)*Multi-scale AI and Cooperation [2], [9].

*19)*Fog-of-War Uncertainty [6].

VI. Open Computational Intelligence and Game AI Competitions

Computational intelligence and Game AI competitions are organized by the Game AI research communities and groups supported by AIIDE[17], CIG[18], and SSCAI[19] which are briefly described below.

A. *CIG (Computational Intelligence & Games)*

Games can be used as a challenging scenario for benchmarking methods from computational intelligence since they provide dynamic and competitive elements that are relevant to real-world problems [5]. The IEEE conference on 'Computational Intelligence and Games' is the premier annual event for researchers applying computational and artificial intelligence techniques to games [37]. The domain of the 'CIG' includes all sorts of 'CI/AI' applied to all sorts of games, including board games, video games, and mathematical games. The annual event series started in 2005 as a symposium, and as a conference since 2009.An overview over the present and all previous 'CIG' competitions and conferences can be reached at http://www.ieee-cig.org/ for further details.

B. *AIIDE (AI Interactive Digital Entertainment)*

During these events, programs play Starcraft brood war games against each other using 'BWAPI' [38], a software library that makes it possible to connect programs to the Starcraft: Brood-war game engine. The purpose of such competitions is to foster and evaluate the progress of AI research applied to real-time strategy (RTS) games [39]. RTS games pose a much greater challenge for AI research than chess because of hidden information, vast state, and action spaces, and the requirement to act quickly. The best human players still have the upper hand in RTS games, but in the years to come, this will likely to change [5].

An overview of the present and all previous AIIDE competitions and conferences can be reached at http://www.cs.mun.ca/~dchurcill/starcraftaicomp/ for further details.

VII. Discussion

This article covered an overview of 'Game AI' which concludes that real-time strategy games are exceptional platforms and sources for 'AI' practices, that carry an enormous list of open issues. Expert human players are still clearly superior to best computer programs. A unique objective of this paper is to make available a compact and integrated outlook of the research presented in the area of RTS Game-AI. We emphasized the present problems in RTS games and analyzed the latest improvements concerning these problems with a concentration on RTS based games, assumed that playing an RTS game is a right puzzling assignment, researchers plan to split such jobs into lesser assignments, that can be independently managed by 'AI' methods. Real-time strategy games involve several attention-grabbing sub-complications that are narrowly associated not only with other areas of AI research but to actual world issues as well. In short, despite continual improvement, each year, still healthy and positive research contributions are required in the field of RTS games and Starcraft to make them more efficient and productive.

Acknowledgment

The authors would like to thank the department of computer science, HIT Harbin, China, as this work is partially funded by the MOE–Microsoft Key Laboratory of Natural Language Processing and Speech, Harbin Institute of Technology, the Major State Basic Research Development Program of China (973 Program 2015CB351804) and the National Natural Science Foundation of China under Grant No. 61572155, 61672188 and 61272386.

Conflict of Interest

The authors declare that they have no conflicts of interest.

[17] AIIDE: www.aiide.org/, AIIDE StarCraft AI Competition: www.starcraftaicompetition.com

[18] CIG: http://www.cig2017.com/, CIG StarCraft AI Competition: http://ls11-www.cs.uni-dortmund.de/rts-competition

[19] http://sscaitournament.com

REFERENCES

[1] Kim, K.J. and S.B. Cho, "Game AI Competitions: An Open Platform for Computational Intelligence Education". Ieee Computational Intelligence Magazine, 2013. 8(3): p. 64-68.

[2] Robertson, G. and I. Watson, "A review of real-time strategy game AI". AI Magazine, 2014. 35(4): p. 75-104.

[3] Synnaeve, G. and P. Bessiere, "Bayesian modeling of a human mmorpg player". arXiv preprint arXiv:1011.5480, 2010.

[4] Waltham, M. and D. Moodley. "An Analysis of Artificial Intelligence Techniques in Multiplayer Online Battle Arena Game Environments". in Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists. 2016. ACM.

[5] Ontanón, Santiago, Gabriel Synnaeve, Alberto Uriarte, Florian Richoux, David Churchill, and Mike Preuss, "A survey of real-time strategy game AI research and competition in Starcraft". IEEE Transactions on Computational Intelligence and AI in games, 2013. 5(4): p. 293-311.

[6] Si, C., Y. Pisan, and C.T. Tan, "A Scouting Strategy for Real-Time Strategy Games". 2014: p. 1-8.

[7] Farooq, Sehar Shahzad, In-Suk Oh, Man-Jae Kim, and Kyung Joong Kim., "StarCraft AI competition: a step toward human-level AI for real-time strategy games". AI Magazine, 2016. 37(2): p. 102-107.

[8] Mattsson, Björn Persson, Tomáš Vajda, and Michal Čertický, "Automatic Observer Script for StarCraft Brood War Bot Games" (technical report). 2015.

[9] Stanescu, M., N.A. Barriga, and M. Buro. "Introducing Hierarchical Adversarial Search, a Scalable Search Procedure for Real-Time Strategy Games". in ECAI. 2014.

[10] Usunier, Nicolas, Gabriel Synnaeve, Zeming Lin, and Soumith Chintala (2016). "Episodic Exploration for Deep Deterministic Policies: An Application to StarCraft Micromanagement Tasks". arXiv preprint arXiv:1609.02993.

[11] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G. and Petersen, S., "Human-level control through deep reinforcement learning". Nature, 2015. 518(7540): p. 529-533.

[12] Stanescu, Marius, Sergio Poo Hernandez, Graham Erickson, Russel Greiner, and Michael Buro. "Predicting Army Combat Outcomes in StarCraft". in AIIDE. 2013. Citeseer.

[13] Stanescu, Marius, Nicolas A. Barriga, Andy Hess, and Michael Buro., "Evaluating Real-Time Strategy Game States Using Convolutional Neural Networks". Sept. 2016.

[14] Si, C., Y. Pisan, and C.T. Tan. "Understanding players' map exploration styles". in Proceedings of the Australasian Computer Science Week Multiconference. 2016. ACM.

[15] Ferber, J., "Multi-agent systems: an introduction to distributed artificial intelligence". Vol. 1. 1999: Addison-Wesley Reading.

[16] Sethy, H., A. Patel, and V. Padmanabhan, Real-Time Strategy Games: A Reinforcement Learning Approach. Procedia Computer Science, 2015. 54: p. 257-264.

[17] Adil Khan, F.J.,"Mobile Sink Random Mobility Model Impact in Wireless Sensor Nodes Energy Consumption Efficiency". International Review of Basic and Applied Sciences, 2016. 4(12): p. 317-325 (9).

[18] Heinrich, J., M. Lanctot, and D. Silver. "Fictitious self-play in extensive-form games". in Proceedings of the 32nd International Conference on Machine Learning, ICML. 2015.

[19] Dereszynski, Ethan W., Jesse Hostetler, Alan Fern, Thomas G. Dietterich, Thao-Trang Hoang, and Mark Udarbe, "Learning Probabilistic Behavior Models in Real-Time Strategy Games". in AIIDE. 2011.

[20] Barriga, N.A., M. Stanescu, and M. Buro. "Puppet Search: Enhancing Scripted Behavior by Look-Ahead Search with Applications to Real-Time Strategy Games". in Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference. 2015.

[21] Barriga, N.A., M. Stanescu, and M. Buro. "Building placement optimization in real-time strategy games". in Tenth Artificial Intelligence and Interactive Digital Entertainment Conference. 2014.

[22] Lample, G. and D.S. Chaplot, "Playing FPS games with deep reinforcement learning". arXiv preprint arXiv:1609.05521, 2016.

[23] Bhatti, S., et al., "Playing Doom with SLAM-Augmented Deep Reinforcement Learning". arXiv preprint arXiv:1612.00380, 2016.

[24] Mahendran, Aravindh, Hakan Bilen, João F. Henriques, and Andrea Vedaldi., "ResearchDoom and CocoDoom: Learning Computer Vision with Games". arXiv preprint arXiv:1610.02431, 2016.

[25] Preuss, Mike, Nicola Beume, Holger Danielsiek, Tobias Hein, Boris Naujoks, Nico Piatkowski, Andreas Thom, and Simon Wessing. , "Intelligent group movement and selection in real-time strategy games". 2008.

[26] Kuchem, M., M. Preuss, and G. Rudolph. "Multi-objective assessment of pre-optimized build orders exemplified for Starcraft 2". in Computational Intelligence in Games (CIG), 2013 IEEE Conference on. 2013. IEEE.

[27] Hsieh, J.-L. and C.-T. Sun. "Building a player strategy model by analyzing replays of real-time strategy games". in Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on. 2008. IEEE.

[28] Ontanón, S. "The combinatorial multi-armed bandit problem and its application to real-time strategy games". in Proceedings of the Ninth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. 2013. AAAI Press.

[29] Buro, M. and T. Furtak. RTS games as test-bed for real-time AI research. in Proceedings of the 7th Joint Conference on Information Science (JCIS 2003). 2003.

[30] Buro, M., M. Lanctot, and S. Orsten, The second annual real-time strategy game ai competition. Proceedings of GAMEON NA, Gainesville, Florida, 2007.

[31] Martin CERTICK´ Y, M.C.Y., Evolving Reactive Micromanagement Controller for Real-Time Strategy Games. 2015.

[32] Stanescu, M. and M. Čertický, Predicting Opponent's Production in Real-Time Strategy Games With Answer Set Programming. IEEE Transactions on Computational Intelligence and AI in Games, 2016. 8(1): p. 89-94.

[33] Stanescu, M., N.A. Barriga, and M. Buro. Hierarchical Adversarial Search Applied to Real-Time Strategy Games. in AIIDE. 2014.

[34] Ferber, J., Multi-agent systems: an introduction to distributed artificial intelligence. Vol. 1. 1999: Addison-Wesley Reading.

[35] Ponsen, Marc JV, Héctor Muñoz-Avila, Pieter Spronck, and David W. Aha. Automatically acquiring domain knowledge for adaptive game AI using evolutionary learning. in Proceedings Of The National Conference On Artificial Intelligence. 2005. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

[36] Frutos-Pascual, M. and B.G. Zapirain, Review of the Use of AI Techniques in Serious Games: Decision making and Machine Learning. 2015.

[37] Preuss, M. and G. Rudolph, Conference Report on IEEE CIG 2014 [Conference Reports]. IEEE Computational Intelligence Magazine, 2015. 10(1): p. 14-15.

[38] Churchill, David, Mike Preuss, Florian Richoux, Gabriel Synnaeve, Alberto Uriarte, Santiago Ontannón, and Michal Čertický., Starcraft bots and competitions. Encyclopedia of Computer Graphics and Games, 2016: p. 1-18.

[39] Sturtevant, N.R., Orkin, J., Zubek, R., Cook, M., Ware, S.G., Stith, C., Young, R.M., Wright, P., Squirrel (Brian) Eiserloh, Ramirez-Sanabria, A. and Bulitko, V. "Playable Experiences at AIIDE 2014". in AIIDE. 2014.

AUTHOR PROFILE

Adil Khan received his B.C.S Honors Degree in Computer Science from Edwards College Peshawar, M.S Degree in Computer Science from CUSIT, B.Ed. from University of Peshawar and C.T. from AIOU Islamabad Pakistan. Currently, he is a Ph.D. Scholar at the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001 PR China. Adil Khan is interested in Neural Networks, Machine Learning, Real Time Strategy Games and First-Person-Shooter Games (Game AI).