StarCraft AI Competition: A Step Toward Human-Level AI for Real-Time Strategy Games

Sehar Shahzad Farooq, In-Suk Oh, Man-Jae Kim, Kyung Joong Kim

■ *This article reviews the two most recent IEEE Conference on Computational Intelligence and Games (CIG) StarCraft Artificial Intelligence (AI) Competitions organized by the authors; these were the fourth and fifth in a series of annual competitions initiated in 2011. StarCraft AI Competitions have been hosted in conjunction with three different events: the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE), CIG, and Student StarCraft AI Tournament (SSCAIT). The purpose of these competitions is to design bots that are able autonomously and successfully to play the StarCraft game by implementing real-time strategies. Recent results reveal the promising use of AI techniques in creating successful AI entries, but there is room for improvement with respect to the bots' ability to adapt and learn to defeat humans and scripted AI bots.*

StarCraft by Blizzard Entertainment is one of the most popular and famous real-time strategy (RTS) games. It provides a dynamic environment in which several agents interact to build military units with which to fight against an opponent. This game involves three distinct "races" (Protoss, Terran, and Zerg) who build different types of units and buildings and who have particular disadvantages and strengths. Players require a great deal of economic and military power to defeat their opponents while surviving and incurring minimal damage. RTS games differ from traditional board games in that they involve simultaneous movement in real time within partially observable and nondeterministic complex environments. Since the introduction of Brood-War API, StarCraft has been an important AI research platform for the development of game-playing bots using various approaches to handle a number of units and buildings by employing careful resource management and high-level tactics.

The First *StarCraft* AI Competition took place in 2010 and was organized by the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE); it involved four tournament categories (micromanagement, small-scale combat, tech-limited game, and complete game)

  ISSN 0738-4602

Competition Year	Number of Participants				Winner		
	Protoss	Terran	Zerg	Total	Race	Name	Win Ratio (%)
2015	7	4	3	14	Zerg	ZZZBot	81.03
2014	7	6	0	13	Terran	ICEBot	83.06
2013	4	4	0	8	Protoss	Skynet	91.1
2012	6	4	0	10	Protoss	Skynet	78.7
2011	6	2	2	10	Protoss	Skynet	86.6

Table 1. History of the IEEE CIG StarCraft AI Competition.

and attracted 26 participants. These categories were merged into a single one (complete game) in 2011 due to strong interest from participants. In 2011, AIIDE organizers developed tournament management software to host multiple clients and allow them to play the games against each other as participants. Along with the AIIDE, the Conference on Computational Intelligence and Games (CIG) and Student StarCraft AI Tournament (SSCAIT) also organized StarCraft AI competitions. Since that time, this competition has been held every year and hosted by the AIIDE and CIG. A competition report for the AIIDE StarCraft Competition was published by Buro and Churchill in *AI Magazine* 2012. Based on the 2010 and 2011 competitions, that report concluded that the new competition had been successfully introduced into the AI community. The current report reviews the two most recent IEEE CIG *StarCraft* AI Competitions,[1] whose results were announced during the CIG conference. Table 1 presents the history of the IEEE *StarCraft* AI Competitions organized by CIG.

CIG *StarCraft* AI Competition

Table 2 provides details regarding the participants in the two most recent years of competition, including the names of their bots and their authors, the races they personified in the competition, their win ratios, and their ranks. Half the participants took part in both the 2014 and the 2015 *StarCraft* AI Competitions. It is interesting to note that the winner of the 2014 competition (ICEBot) dropped to seventh place in the 2015 competition, with six entries outperforming the previous year's winner. Some participants had made a number of changes to their bots from the previous year. For example, in 2014 LetaBot, who represented the Terran race, used various strategies, including depth-first search and flood-fill algorithm, to calculate possible wall-in locations to thwart early attacks. (Wall-in locations are building positions in *StarCraft* that can block an opponent's attack or trap the opponent with low-range attacking units.) However in 2015, LetaBot replaced the flood-fill algorithm with A*, which decreases the CPU time required to calculate wall-in locations.

In 2014, ICEBot (referred to as ICEStarCraftBot 2014 in the IEEE CIG *StarCraft* AI Competition 2015) used a nondeterministic finite state machine to patrol enemy units. Furthermore, adaptive strategy rules, which were developed to predict enemy strategies, used potential flow algorithm to reduce the computational time required for pathfinding (Nguyen, Wang, and Thawonmas 2013). In 2015, only slight script changes were made to take into account the new maps fixed for the CIG *StarCraft* AI Competition 2015.

AIUR (Artificial Intelligence Using Randomness) used an epsilon-greedy algorithm in the mode manager to select a random mode from among a list of predefined modes, including Aggressive, Cheese, Rush, Fast-expanding, Defending, and Macro-game.[2] The defensive mode was selected at the start of the game regardless of the opponent but, during the tournament, AIUR could implement an offline learning feature in which the bot was trained against a given opponent.

OpprimoBot was based on a flexible, modular multiagent-based architecture that allowed easy modification of the bot's behavior. It used a finite state machine to adapt to the opponent's strategy (Ontanon et al. 2013).

The ZZZBot was the most effective bot in terms of solid defense (and was the winner of the IEEE CIG *StarCraft* AI Competition 2015). It used simple logic with limited micros for scouting, targeting, and resource gathering. The strategy was simple; the agent's objective was to act very quickly and attack the opponent without giving the opponent time to execute a defense. At a very early point in the game, the bot did nothing but collect minerals. As the mineral count reached 200, it switched to constructing a spawning pool to produce Zerglings. Such a maneuver can be easily overwhelmed if the opponent reacts

No.	Bot Name	Main Contributor	Race	Win Rate (%)	
				2015	2014
1	ZZZBot	Chris Coxe	Zerg	81	-
2	Tscmoo_Z	Vegard Mella	Zerg	74	-
3	Overkill	Sijia Xu	Zerg	62	-
4	LetaBot	Martin Rooijackers	Terran	62	68 (3rd Place)
5	Ximp	Tomas Vajda	Protoss	60	78 (2nd Place)
6	Tyr	Simon Prins	Terran	54	-
7	ICEStarCraft 2014	Nguyen Duc Tung et al.	Terran	54	83 (1st Place)
8	AIUR	Florian Richoux	Protoss	53	66
9	Tscmoo_T	Vegard Mella	Terran	53	-
10	UAlbertaBot2013	David Churchill	Protoss	48	60
11	WOPR	Sören Klett	Terran	46	57
12	NUSBot	Gu Zhan et al.	Protoss	22	22
13	OpprimoBot/BTHAI 2014	Johan Hagelback	Terran	21	32
14	NOVA	Alberto Uriarte	Terran	10	39
15	MassCraft	Dennis Soemers	Protoss	–	55
16	MooseBot	Adam Montgomerie	Protoss	–	38
17	TerranUAB	Filip Bober	Terran	–	34
18	CruzBot	Daniel Montalvo	Protoss	–	18

Table 2. IEEE CIG StarCraft AI Competition Participants (2014, 2015), Win Ratios, and Ranks.

quickly, but it allows little time for the opponent to decide to react, which is a key component in this strategy. The bot used a purely script-based program with no AI involved. Because current AI techniques do not yet allow bots to react to and appropriately handle unusual or unexpected situations, the script was highly successful in 2015.

Although human versus AI competition was not a part of the IEEE CIG *StarCraft* AI Competition, an offline competition was conducted involving an expert human player (In-Suk Oh, coorganizer of the competition and a former professional *StarCraft* II gamer). He played against most of the bots entered into the 2015 CIG *StarCraft* AI Competition. The test was motivated by the question of whether the bots were comparable to humans. However, the human player easily defended the very early Zergling attack by the first-ranked ZZZBot. Because the bot relies on a high-risk strategy, there is no way for it to recover if the strategy fails. Although it is effective against other AI bots, it fails to threaten human players. Tscmoo Bot, the second-ranked bot, maintained its initial strategy even though the human observed the opponent's important buildings, revealing the opponent's plan to produce aerial units. Hence, it was easily beaten by the human player, who applied an effective counterstrategy. OverKill Bot, the third-ranked bot, behaved similarly to Tscmoo Bot. This test revealed that, although bots are strong against other AI bots, they are not yet able to adapt to or to defeat humans.

Discussion

In the *StarCraft* AI competition, several factors affect the final outcome, including the bot's choice of a predefined strategy, the unexpectedness or novelty of its strategies, the ease of their implementation, and the choice of race. In the early days, Protoss was the most dominant race and repeatedly won the competition. This race is characterized by expensive but strong units, and it performs well with simple strategies. In contrast, the Terran race requires sophisticated micromanagement skills that reflect an understanding of the terrain to win games. In 2014, ICEBot won the competition, exhibiting highly specialized micromanagement skills boosted by a potential flow technique; this was the first time a winner had represented the Terran race. However, in 2015, the success of the Terran race was limited, and the Zerg race, armed with cheap and weak units, prevailed.

Surprisingly, the Zerg race was represented by three entries in 2015, each of which successfully ranked among the top three spots. The Zerg had not achieved this level of success throughout the entire history of the AI competition history, with no Zerg

entries at all in any of the previous three years. The success of the Zerg was based on the unexpectedness and novelty of its strategies, demonstrating that although AI techniques are important in the competition, unexpectedness or novelty has a strong impact on the results. Because the AI bots are not good at adapting their strategies during competition, a winning bot is likely to repeatedly defeat the same opponent. If the AI bot uses an unexpected strategy, it has a good chance of winning the competition, because AI bots have no way of creating a counter-strategy against an unknown attack on a site. However, among humans, it is unlikely that one would beat the same opponent using the same strategy more than once. Over the past few years of competitions, AI bots have been armed with a number of different strategies, some of which are no longer effective. However, unexplored novel strategies for AI bots remain, and the Zerg entries successfully exploited some of them to surprise the other agents this year.

Although the nature of the predefined strategy still has a large impact on outcomes, meaningful progress in AI techniques has been made. The bots are not as adaptable or intelligent as human players, but they exhibit the complex behaviors required for the implementation of real-time strategies against AI players. For example, a set of real-time search techniques has been adopted to enhance combat decision making and build-order planning (Cowling et al. 2013, Churchill and Buro 2011), reducing the burden on the AI programmer to design very complex rule-based systems using efficient tree search and goal-oriented planning. In addition, advances using influence maps and potential flows, which allow for the navigation of units while taking into consideration numerous factors, such as buildings, resources, and the opponent's units, have been made with respect to spatial representations of the gaming space. Although adaptation and learning are still in the early stages of development, some attempts have been made to change strategy based on previous experience with the same opponent.

In contrast with traditional AI methods, a data-driven approach for the *StarCraft* AI bots offers considerable potential. Because *StarCraft* is one of the most popular RTS games, numerous replays are available through gaming portals. For example, it is possible to download approximately 340,000 replays from bwreplays.com. This site contains many different games played by individuals of various skill levels, ranging from novice to professional players. Each replay includes all the actions taken by players during the game. Using replay analyzer software (BWChart or LMRB), it is possible to extract all the gaming events and use them as a resource for developing a statistical model of human player activity. For example, Oh, Cho, and Kim (2014) used replays to develop micromanagement skills based on the imitation of human patterns. By creating a customized replay analyzer using the BWAPI replay tool, it is possible to extract data that would not be available using more conventional analyzer software. In addition to the human replays, it is also possible to download replays between AI bots from the competition website. In contrast to tournaments involving human players, competition between AI bots can generate thousands of replays within days, providing a useful dataset for learning about the strengths and weaknesses of AI bots.

Due to the programming complexity of *StarCraft* AI, it is not easy to attract newcomers to this fascinating field; however, a recent open-source policy has made it easier to initiate AI entries. Much debate has arisen over the past few years regarding an open-source policy for submitted entries. In the early days of the competition, there was no explicit rule requiring participants to make their source code available. Because *StarCraft* AI programming requires many lines of codes to handle the various tasks involved in RTS games, the difficulty of starting from scratch can serve as a barrier for newcomers. To make progress, it is essential for the winner's source code to be open to allow improvements to be made to entries in the subsequent year. The creators of UAlbertaBot, the winner of the AIIDE 2013 competition, opened their source code to the public and encouraged newcomers to use their code to create a better bot.[3] In recent years, the AIIDE and CIG have required or encouraged entrants to make their source code available to the public. For CIG 2014, all entrants were requested to open their source code and, in 2015, although the open source policy remained optional, most entrants agreed to abide by it. The source code for all of the entries is currently available in the results section of the websites of the recent competitions. Using the source code of successful entries as a starting point for improvements is an excellent approach.

The sharing of core software modules and resources for *StarCraft* AI competitions is essential. Various resources currently exist to support the development of AI bots for *StarCraft*. For example, the Game AI competition portal[4] helps novice programmers initiate their first project in *StarCraft* AI (Kim and Cho 2013). Recently, SparCraft[5] has been developed to provide a simplified combat simulation. It can be combined with state-of-the-art search techniques (such as Monte Carlo Tree Search) proven to be effective in very complex games. Because the *StarCraft* AI competition is based on programming interfaces created through third-party hacking, many limitations remain with regard to access to the core functions of the game. For example, because there is no simulator for *StarCraft*, it is difficult to apply extensive tree search algorithms combined with simulation of future outcomes. D. Churchill made tournament management software available to the public.[6] Because AI competitions require thousands of games among entries, it is important to use tourna-

ment management software that automatically assigns AI bots to available machines, starts games, and collects results after each game. This allows multiple machines to run simultaneously and to complete thousands of games within days.

Future Competitions

The *StarCraft* AI competition has a relatively short history compared with other competitions. This means that there are interesting opportunities for improving the current versions of human-competitive AI bots. Current AI competition requires each AI bot to represent a single race. For example, if the AI is designed to play the Protoss race, it cannot change its race on a game-by-game basis. Alternatively, it is possible to randomly assign race for each game. Human players, on the other hand, may change race for tactical purposes in response to specific maps or opponents. Although flexible race options are not fully supported in the current tournament software and it is difficult for AI developers to prepare more than one race, having multirace AI players would be an interesting option.

Currently, the competitions assume that all matches will be played on a one-on-one basis. However, humans engage in different types of game-playing styles, such as two-on-two matches, in which two players play against two others. In such a game style, it is necessary for team members to communicate and cooperate with each other to win the game. High-level tactical strategies that involve the complementary skills of team members are needed to maximize a team's chances of winning. Given that we are not yet ready for human–AI competitions, even with one-on-one matches, creating a bot that can cooperate with humans or other bots is still a long way off. However, it is possible that new competitions, such as those involving two AI bots playing against two AI bots or joint teams of human and AI players, will be organized in the future.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP) (2013 R1A2A2A01016589).

Notes

1. The IEEE CIG *StarCraft* AI Competition website is located at cilab.sejong.ac.kr/sc_competition. Previous competition links may also be found on the website.

1. For information, see the AIUR website (aiur-group.github.io/AIUR).

2. See UAlbertaBot, code.google.com/p/ualbertabot.

3. code.google.com/p/sparcraft.

4. *StarCraft* AI Competition Tournament Management Software, is available at webdocs.cs.ualberta.ca/~cdavid/starcraftaicomp/tm.shtml.

6. The Game AI Competition portal is available at cilab.sejong.ac.kr/gc.

References

Buro, M., and Churchill, D. 2012. Real-Time Strategy Game Competitions, *AI Magazine* 33(3): 106–108.

Churchill D., and Buro, M. 2011. Build Order Optimizations in *StarCraft*. In *Proceedings of the 7th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment,* 14–19. Palo Alto, CA: AAAI Press.

Cowling, P. I.; Buro, B.; Bida, M.; Botea, A.; Bouzy, B.; Butz, M. V.; Hingston, P.; Munoz-Avila, H.; Nau, D.; Sipper, M. 2013. Search in Real-Time Video Games, Dagstuhl Follow-Ups. *Artificial and Computational Intelligence in Game*s 6: 1–19. dx.doi.org/10.4230/DFU.Vol6.12191.1

Kim, K.-J., and Cho, S.-B. Game 2013. AI Competitions: An Open Platform for Computational Intelligence Education, 2013. *IEEE Computational Intelligence Magazine* 8(3): 64–68. dx.doi.org/ 10.1109/MCI.2013.2264568

Nguyen, K. Q.; Wang, Z.; Thawonmas, R. 2013. Potential Flows for Controlling Scout Units in *StarCraft*. In *Proceedings of the 2013 IEEE Conference on Computational Intelligence in Games (CIG),* 1–7. Piscataway, NJ: Institute for Electrical and Electronics Engineers. dx.doi.org/10.1109/CIG.2013.6633662

Oh, I.-S.; Cho, H.-C.; Kim, K.-J. 2014. Imitation Learning for Combat System in RTS Games with Application to *StarCraft*. In *2014 IEEE Conference on Computational Intelligence and Games*. Piscataway, NJ: Institute for Electrical and Electronics Engineers. dx.doi.org/ 10.1109/CIG.2014.6932919.

Ontanon, S.; Synnaeve, G.; Uriarte, A.; Richoux, F.; Churchill, D.; and Preuss, M. 2013. A Survey of Real-Time Strategy Game AI Research and Competition in *StarCraft*. *IEEE Transactions on Computational Intelligence and AI in Games* 5(4): 293–311 dx.doi.org/10.1109/TCIAIG.2013.2286295

Sehar Shahzad Farooq is a Ph.D. student at Sejong University. He earned his BS in electrical engineering from the University of Faisalabad. His research interest includes computational intelligence, behavior interpretation, game player modeling, and physical robotics. He was coorganizer of the IEEE CIG *StarCraft* AI Competition 2015.

In-Seok Oh is a master's degree student at Sejong University, where he received a BS in engineering. His research interests include computational intelligence in games, case-based reasoning, reinforcement learning, and Monte Carlo tree search (MCTS). He was a coorganizer of the IEEE CIG StarCraft AI Competition in 2014 and 2015. He works currently as an intern for NCSoft Corporation.

Man-Je Kim is an undergraduate researcher at the Cognition and Intelligence Laboratory in Sejong University. His research interests include artificial intelligence, pattern recognizing, and deep learning.

Kyung-Joong Kim received a B.S., an M.S., and a Ph.D. in computer science from Yonsei University in 2000, 2002, and 2007, respectively. He worked as a postdoctoral researcher in the Department of Mechanical and Aerospace Engineering at Cornell University in 2007. He is currently an associate professor in the Department of Computer Science and Engineering at Sejong University. His research interests include artificial intelligence, game, and robotics.

The Twelfth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-16)

October 8–12, 2016

Embassy Suites by Hilton San Francisco Airport – Waterfront
Burlingame, California

Conference Chair
Nathan Sturtevant (University of Denver)

Colocated with the ACM SIGGRAPH Motion in Games (MIG) 2016 Conference

Workshops: October 8–9

www.aiide.org