Call for AI Research in RTS Games

Michael Buro
Department of Computing Science, University of Alberta, Edmonton, AB, T6J 2E8, Canada
email: mburo@cs.ualberta.ca

Abstract

This position paper discusses AI challenges in the area of real–time strategy games and presents a research agenda aimed at improving AI performance in these popular multi–player computer games.

RTS Games and AI Research

Real–time strategy (RTS) games such as Blizzard Entertainment's Starcraft(tm) and Warcraft(tm) series form a large and growing part of the multi–billion dollar computer games industry. In these games several players fight over resources, which are scattered over a terrain, by first setting up economies, building armies, and ultimately trying to eliminate all enemy units and buildings. The current AI performance in commercial RTS games is poor. The main reasons why the AI performance in RTS games is lagging behind developments in related areas such as classic board games are the following:

- RTS games feature hundreds or even thousands of interacting objects, imperfect information, and fast–paced micro–actions. By contrast, World–class game AI systems mostly exist for turn–based perfect information games in which the majority of moves have global consequences and human planning abilities therefore can be outsmarted by mere enumeration.
- Video games companies create titles under severe time constraints and do not have the resources and incentive (yet) to engage in AI research.
- Multi–player games often do not require World–class AI performance in order to be commercially successful as long as there are enough human players interested in playing the game on–line.
- RTS games are complex which means that it is not easy to set up an RTS game infrastructure for conducting AI experiments. Closed commercial RTS game software without AI interfaces does not help, either. The result is a lack of AI competition in this area which in the classic games sector is one of the most important driving forces of AI research.

To get a feeling for the vast complexity of RTS games, imagine to play chess on a 512×512 board with hundreds of slow simultaneously moving pieces, player views restricted to small areas around their own pieces, and the ability to gather resources and create new material.

While human players sometimes struggle with micro–managing all their objects, it is the incremental nature of the actions that allows them to outperform any existing RTS game AI. The difference to classic abstract games like chess and Othello in this respect is striking: many moves in these games have immediate global effects. This makes it hard for human players to consider deep variations with all their consequences. On the other hand, computers programs conducting full–width searches with selective extensions excel in complex combinatorial situations. A notable exception is the game of go in which — like in RTS games — moves often have only incremental effects and today's best computer programs are still easily defeated by amateurs (Müller 2002). It is in these domains where the human abilities to abstract, generalize, reason, learn, and plan shine and the current commercial RTS AI systems — which do not reason nor adapt — fail.

Other arguments in favor of AI research in RTS games are:

- (RTS) games constitute well–defined environments to conduct experiments in and offer straight–forward objective ways of measuring performance,
- RTS games can be tailored to focus on specific aspects such as how to win local fights, how to scout effectively, how to build, attack, and defend a town, etc.,
- Strong game AI will likely make a difference in future commercial games because graphics improvements are beginning to saturate. Furthermore, smarter robot enemies and allies definitely add to the game experience as they are available 24 hours a day and do not get tired.
- The current state of RTS game AI is so bad that there are a lot of low–hanging fruits waiting to be picked. Examples include research on smart game interfaces that alleviate human players from tedious tasks such as manually concentrating fire in combat. Game AI can also help in the development of RTS games — for instance by providing tools for unit balancing.

- Finally, progress in RTS game AI is also of interest for the military which uses battle simulations in training programs (Herz & Macedonia 2002) and also pursues research into autonomous weapon systems.

Research Agenda

The main goal behind the AI research being proposed here is *not* to increase the entertainment value of RTS games, but rather to create the strongest RTS game AI possible. The former goal is pursued by the commercial games industry, whereas the latter tries to push the cognitive abilities of machines to new levels. Note, however, that increased playing strength can be converted into higher entertainment value by adapting to the player's performance level to keep games challenging. Also, we acknowledge that commercial RTS game AI often cheats to compensate for its lack of sophistication. Tricks of the trade include map revealing and faster resource gathering. The resulting AI systems may outperform human players and may even create challenging encounters, but they do not advance our understanding of how to create intelligent entities.

In order to repeat the classic game AI success in RTS games, progress in the following key areas is required:

- Adversarial planning under uncertainty.

  Because of the huge number of possible actions at any given time and their microscopic effects in RTS games, agents cannot afford to think at the game action level. Instead, abstractions of the world state have to be found that allow programs to conduct forward searches in abstract spaces and to translate found solutions back into the original state space. Human problem solvers conduct such planning all the time, but programs are not that advanced yet. Consequently, current RTS game AI systems can be confused by simple maneuvers like flanking attacks and diversions. All high–level decisions in RTS games such as when to build what, where to scout, and where to extend and attack are based on search in abstract state spaces augmented by beliefs about the world. Because the environment is hostile and dynamic, adversarial real–time planning approaches need to be investigated. Without this planning ability there is not much hope for RTS game AI to defeat human players at the commander level.

- Learning and opponent modeling.

  One of the biggest shortcomings of current (RTS) game AI systems is their inability to learn quickly. Human players only need a couple of games to form accurate hypotheses about their opponents' playing styles and weaknesses and to exploit them in future games. New efficient machine learning techniques have to be developed to tackle these important problems. Applied to RTS games, the resulting AI would be able to react to discovered tendencies such as a bias towards air–force or turtling. One way to incorporate gathered information into the decision process is to adjust beliefs formed by the planning module. Dynamic scripting which has been applied to a role–playing game in (Spronck, Sprinkhuizen-Kuyper, & Postma 2003) looks like a promising starting point.

- Spatial and temporal reasoning.

  Understanding the importance of static terrain features like choke points and dynamic spatial properties such as visibility and enemy influence is crucial for generating successful plans in RTS games. Another important aspect is the temporal relationship among actions. To illustrate the necessity of spatial and temporal reasoning for guiding decisions in RTS games consider the following scenario in which a base is attacked and the defender's tank force is on its way to the attacker's base. The decision on whether to return and defend the home base or to proceed and destroy the attacker's base depends partly on how long it takes for the tanks to reach either location and whether the tank force and local defenses are capable of thwarting the attack. Combining terrain knowledge with simple heuristics such as the air–distance lower bound is often sufficient to pick the best course of action. Current RTS game AI largely ignores these issues and falls victim to simple common–sense reasoning (Forbus, Mahoney, & Dill 2002).

A continuing exponential hardware speedup will certainly help to improve AI performance in RTS games. However, without clever abstractions and algorithms — comparable to the impact of the alpha–beta improvement over minimax search — fast progress is unlikely because of enormous branching factors. Although eventually all of the problems mentioned above have to be tackled, gradual improvements can have tremendous effects. The idea is simple: because current AI systems do not reach human planning, learning, and reasoning levels, machines can at least aid them playing RTS games. To some extent they already do (pathfinding and default unit behavior). However, there are numerous other ways of improving game performance which can be easily integrated in RTS game interfaces. For example, when attacking a group of units, current RTS game often require players to micro–manage their attack force in order to manually concentrate fire or to intercept fleeing units. It should be possible to create AI modules to handle those local combat situations much more efficiently than humans who have to point and click on units to give them orders. What makes this hybrid AI approach attractive is that now human players can choose their favorite AI plugins. Moreover, players then can concentrate on high–level decisions without being forced to compete with the World's fastest mouse virtuosos in terms of speed. This in turn could lead to an AI arms race provided the RTS game is popular. The hope is that the focus of AI modules will then shift to higher–level tasks such as orchestrating large–scale attacks on enemy fortifications or securing resource locations. As more and more high–level tasks can be delegated to AI modules, it is conceivable that eventually the human RTS game player – even when aided by AI – will be defeated.

Before this vision can become reality, the necessary infrastructure has to be developed. RTS game companies are reluctant to add interfaces to their products which would allow researchers to play RTS games remotely and to gauge the playing strength of RTS AI systems by means of tournaments. We therefore began in 2001 to create a free soft-

Figure 1: A screen shot of the ORTS OpenGL 3d graphics interface. On the left–hand side a playfield sector is displayed showing terrain features and game objects. The right–hand side shows action buttons and the mini–map.

ware RTS game engine called ORTS [(Buro 2002) (Buro 2003), Fig. 1] with the following design goals:

- Server/client operation. The entire game state is only known to the server which sends out individual views to players. Thus, unlike any other RTS game currently available,[1] ORTS is not prone to map–revealing hacks which have become a serious problem for on–line tournaments.
- Flexibility. ORTS is an RTS game engine which allows users to define all game properties using a script language. This means that AI researchers can tailor ORTS games towards particular research needs.
- Open protocol. In contrast to commercial RTS games, the communication protocol between the ORTS server and clients is open. Users can connect whatever client software they wish. This includes highly customizable GUIs, AIs running on super computers, or hybrid systems.
- Total game control. Units are entirely controlled by the client software. There is no built–in default behavior which in other RTS games is often contra–productive (e.g. automatically fired tank shells hurting friendly units) or non–existent (meaning that players have to baby–sit units). In the ORTS game model, clients can generate a single action for each unit under their control in each simulation cycle.
- Free software. Licensed under the GNU Public License the ORTS source code is available and everybody is invited to study and improve it.

The ORTS software and articles that describe our AI research in RTS games and related work in more detail are available at `www.cs.ualberta.ca/~mburo/orts`. We are currently working hard on a software release featuring a multi–race standard ORTS game, a game portal similar to Battle.net(tm), and a highly customizable GUI complete with client scripting support and animated 3d models.

[1] Including the free software RTS engine Stratagus (www.stratagus.org)

Conclusion

In this paper we have motivated AI research in RTS games and outlined a research agenda whose goal it is to produce AI systems that reason, learn, and plan in this popular and challenging domain.

We invite AI researchers to participate in creating and improving RTS game AI systems, to contribute to the ORTS project, and to compete on-line. Competition is a powerful driving force for AI innovation which has been witnessed in the areas of classic board games, planning, and auctions. We hope that it will also lead to strong RTS game AI and elevate our understanding of real–time decision making under uncertainty.

Acknowledgments

Thanks go to Tim Furtak and Keith Yerex for their important contributions to the ORTS project. This research is supported in part by grants from NSERC, Canada.

References

Buro, M. 2002. ORTS: A hack-free RTS game environment. In *Proceedings of the Third International Conference on Computers and Games*, 156–161.

Buro, M. 2003. Real-time strategy games: A new AI research challenge. In *Proceedings of the International Joint Conference on AI*, 485–486.

Forbus, K.; Mahoney, J.; and Dill, K. 2002. How qualitative spatial reasoning can improve strategy game AIs. *IEEE Intelligent Systems* 17(4):25–30.

Herz, J., and Macedonia, M. 2002. Computer games and the military: Two views. *Defense Horizons, Center for Technology and National Security Policy, National Defense University* 11.

Müller, M. 2002. Computer go. *Artificial Intelligence Journal* 134(1–2):145–179.

Spronck, P.; Sprinkhuizen-Kuyper, I.; and Postma, E. 2003. Online adaptation of game opponent AI in simulation and in practice. In Mehdi, Q., and Gough, N., eds., *Proceedings of the 4th International Conference on Intelligent Games and Simulation*, 93–100.