Requirements for resource management game AI

Steven de Jong, Pieter Spronck, Nico Roos
Department of Computer Science, Universiteit Maastricht, Netherlands
Email: {steven.dejong, p.spronck, roos}@cs.unimaas.nl

Abstract

This paper examines the principles that define resource management games, popular and challenging constructive computer games such as SIMCITY and VIRTUAL U. From these principles, it is possible to derive requirements for intelligent programs designed to play such games, as a replacement of a human player.

A first step for research in the domain of intelligent programs playing resource management games is presented, in the form of a hybrid AI approach that combines abductive planning with evolutionary learning. The results obtained by this approach are promising.

1 Artificial intelligence in interactive computer games

One of the main goals of AI research always has been the development of artificial intelligence that is as versatile as human intelligence. For many years, the game of CHESS has been the *drosophila melanogaster* of artificial intelligence research [McCarthy, 1990], but in the last decades, it is argued that games such as CHESS are not able to address all abilities of intelligence sufficiently, because they are abstract and completely deterministic [Pfeiffer and Scheier, 1999; Laird and van Lent, 2001]. Computers are good at calculation and therefore inherently good at dealing with abstract and deterministic problems, but being good at calculation clearly is not the same as being able to make intelligent decisions [Pfeiffer and Scheier, 1999].

In recent years, the computer games industry has received increasing attention from the artificial intelligence community, because interactive computer games, while being a closed world, can require a computer to perform tasks that currently only humans are able to perform sufficiently well. Interactive computer games are therefore seen as an ideal testbed for alleged computer intelligence [Laird and van Lent, 2001]. Potential tasks for AI in computer games can vary from entertaining the human player to actually being able to play an entire game.

Since designers of interactive computer games aim at a commercial rather than a scientific goal, computer controlled players for such games are rarely, if ever, designed to explore the game autonomously to arrive at high-quality decision making capabilities (i.e., high-quality game AI). Therefore, if game AI is applied in practise, it is often static and only encompasses the designers' insight on what the game constitutes. We observe three problems here.

First of all, from a scientific perspective, we should aim at developing AI methods that do not need designers' insight in order to work well. A method that works well because it reproduces knowledge programmed ahead of time, may be able to play an intelligent game, but cannot be considered to be intelligent itself. Moreover, human designers may make mistakes, or fail to take into account relevant information, which makes the game AI of inferior quality.

Second, methods using knowledge programmed ahead of time will not be able to adapt when a game's parameters change, or when opponents are encountered that follow strategies not taken into account. In commercial computer games, the specifications are often changed to resolve balance issues. Adaptive AI methods offer an advantage here: if changes are needed, the methods can adapt easily and do not need to be rebuilt.

Third, even if our only goal is to develop computer-controlled players for commercial games, we must realize that most human players expect that a computer plays fairly (i.e., it does not use knowledge that human players do not have) [Scott, 2002]. For example, Garry Kasparov was furious with Deep Blue when the computer had beaten him in CHESS, because the grandmaster was convinced that it had been using information provided during the game by a team of CHESS players [Jayanti, 2003].

In this paper, we examine the principles that define resource management games, which are interactive computer games with a constructive nature. These principles lead to a list of requirements for intelligent programs designed to play such games, as a replacement of a human player. Exploratory research in the domain of intelligent programs playing resource management games shows that hybrid AI approaches, combining abductive planning with evolutionary learning, are a possible way of dealing with these requirements.

In section 2 of this paper, we address the relevance of research in the domain of resource management games. In section 3, we discuss the principles underlying these games and derive the required capabilities of resource management game players (including computer-controlled players). Section 4 presents a possible solution method that is able to deal with these capabilities, and section 5 continues with a brief overview of experiments that compare the performance of this method with more conventional methods. In section 6, we conclude and look at future work.

2 Relevance of research into resource management games

We define resource management games as interactive computer games where the main problem for the player[1] is to use limited resources to construct and maintain a complex virtual environment. These games have proved to be challenging and entertaining, because of their many layers of complexity, arising from a rather small and understandable set of actions. They usually require a player to construct buildings and move units in a large grid world. While playing a resource management game, the player tries to reach a certain goal by carefully distributing limited resources. A famous example is the game SIMCITY, in which the player builds and manages a city.

Resource management games share many ideas with strategy games such as WARCRAFT [Laird and van Lent, 2001; Buro, 2004; Ponsen and Spronck, 2004], but their nature and game play differ significantly from these games, as explained below.

The main difference in nature is that in strategy games, the player constructs buildings and controls units of a virtual army in order to defeat opponent armies, whereas in resource management games, the focus is on the construction and long-term maintenance of buildings, transport networks, et cetera. Due to this difference in nature, strategy games are in use by many military organizations as a source of inspiration [Laird and van Lent, 2001], whereas resource management games receive more attention from economists and managers [Sawyer, 2002]. One might summarize that strategy games have a destructive nature, whereas resource management games have a constructive nature.

One of the two main differences in game play between the genres is the fact that strategy games are always finite games – once the enemy has been defeated, the game ends – whereas resource management games do not need to be finite, with goals such as 'build and maintain a large city' or 'transport many passengers'. A second difference in game play is that most strategy games progress in near-continuous time, hence the name real-time strategy (RTS) games [Ponsen and Spronck, 2004], whereas most resource management games progress in clearly observable discrete time. Near-continuous games, such as real-time strategy games, enable all players to move at the same time, which entails that players must possess both the capacity to respond quickly to urgent matters and the capacity to think about strategic problems. Discrete or turn-based games, such as resource management games, are similar in pace to games such as CHESS: each player takes his turn to perform a limited number of actions, and is allowed to think about which actions to perform. Because of this, the challenges posed by discrete games often require more structured thinking and less intuitive response than those posed by near-continuous games – for example, a player has to construct a sound planning towards some end goal.

There are three reasons why it is relevant to perform research in the area of AI for computer-controlled players for resource management games. First, because of the many challenges involved in playing interactive computer games, a computer-controlled player for such games must be able to deal with many aspects of human-level intelligence [Laird and van Lent, 2001]. Resource management games are able to address aspects such as high-level planning, which are not often found in other genres of interactive computer games, but rather in classical board games such as CHESS. Therefore, resource management games can be said to bridge the gap between classical games and interactive computer games. Second, problems found in many other domains (such as other genres of computer games, management and economics) closely resemble problems typically found in resource management games. Third, resource management games are being developed not only as pure entertainment, but also as educative tools. For example, Harvard's Business School is known to use resource management games in its student training program [Sawyer, 2002]. AI techniques that are able to deal with educative resource management games such as VIRTUAL U can be valuable sources of strategic information for scientists and students; in other words, the solutions AI techniques come up with, can be analysed and used as inspiration for people playing these games.

3 Game principles and players' capacities

A typical resource management game is played in a simulated world, usually a grid that the player looks at in bird's eye view. As time passes, a player and/or the game itself can place various kinds of structures on the grid, for example roads and buildings. The content of the grid defines the state of the game world, which is internally represented by a set of parameters. The game's dynamics are defined by a set of actions that determine the transition from the current state of the game into a new state. Both the game itself and the player can execute these actions. When playing the game, the player has to cope with many temporary objectives that eventually en-

[1] Henceforth, we will use the term 'player' to indicate both a human player and a computer-controlled player, unless indicated otherwise.

able him to reach the end goals of the game. Playing a resource management game requires thorough planning: the player must find out which actions to perform in order to eventually reach the end goals of the game, given the current state of the game world. To be able to do this, he must take into account the fact that resources and dynamical aspects play an important role[2].

Resource management games use a set of principles that lead to challenging and surprising game play for the player. We will illustrate these principles with a basic resource management game called FACTORY, in which the player can build houses and factories in order to control unemployment rates in a city. Figure 1 illustrates a possible state and interface for this game. The goal of the game is to fill the entire grid and reach a stable unemployment rate of 0%. A player can place houses and factories on the grid using the buttons BUILD HOUSE and BUILD FACTORY. The game's response to placing objects on the grid is a new value for the player's money and the current unemployment rate in the city.

The game's actual progress is determined by one or more of the principles outlined below. If we are aiming at developing a computer-controlled player for resource management games, we must keep in mind that such a player must be able to address all of these principles.

1. *Dependence on resources* – Every resource management game contains this principle, which entails that resources are limited and that there are dependencies between some of them. If for example we build one factory, for which we require 100,000 euros, at most 100 unemployed people will find a job and will start paying taxes.
2. *Dependence on location* – This principle implies that the location on the grid where we build objects influences the effect on the game world. For example, if we build a factory too close to a residential area, people will start complaining about pollution and decide to stop paying taxes, whereas if we build the factory too far from the residential area, people will refuse to go to work.
3. *Dependence on time* – This principle implies that in some cases, the effect of a certain action is delayed. For example, if we build a factory now, it will be finished in 12 months (game time).
4. *Need for planning* – A player cannot execute all possible actions immediately from the beginning of the game. For example, the player must gather at least 100,000 euros before a factory can be built. Furthermore, the game rules might stipulate that a factory can only be built if there are sufficient people available. These people must have housing, which means we can only build a factory after we have built at least a few houses for the workers to live in. Thus, if the player wants to build a factory, he must come up with a sequence of actions that makes the game progress from the current state to a state in which building a factory is possible.
5. *Competition* – In many resource management games, the player has to compete with computer-controlled opponents or co-operate with allies, sharing limited resources effectively.
6. *Multiple objectives* – Resource management games often require the player to pursue more than one objective at once in order to reach the end goals of the game. For example, in the FACTORY game, the player will be trying to build as many factories as possible, but also to keep his financial situation healthy and to outperform any competition, and he does all this to reach the game's goal: creating a city that has an unemployment rate of 0%.
7. *Non-determinism* – Many games include elements of non-determinism. Practical implementations vary from adding odd-behaving competitors or natural disasters to a game to introducing noise on the effects of rules that specify game dynamics. For example, if we build a factory, we expect 100 people to find a job, but the actual number of people that do find a job varies.

From these seven principles, we derive the following required capabilities for a player of resource management games. He must be able (1) to cope with resource dependencies, (2) to allocate space effectively, (3) to predict future game states, (4) to create a sound planning, (5) to handle competition , (6) to perform multi-objective reasoning, and (7) to deal with non-determinism. All of these capabilities are actually used for only one task: selecting actions that should be executed. In deterministic games, the sequence of actions selected by each of the players defines the outcome completely. In non-deterministic games, the sequence of actions also plays an important role in the outcome – after all, if the outcome is determined more by randomness than by the actions executed, the game is more a game of chance than a real resource management game. Thus, action selection is the core task of any resource management player, including computer-controlled players.

4 Methods

There are many ways of developing a computer-controlled player that can perform action selection in resource management games, using the required capacities outlined in the previous section. A possible approach is to use *hybrid AI*, which we define as AI in which elements of symbolic and behavior-based artificial intelligence are combined. We give three reasons here to support our statement that a hybrid AI approach should be suitable for the domain of resource management games.

[2] In this respect, the planning problems encountered in resource management games differ substantially from classical planning problems, that are successfully addressed by various symbolic AI techniques. In these classical planning problems, the focus is on state transitions, usually in a deterministic or even static environment without competitors.

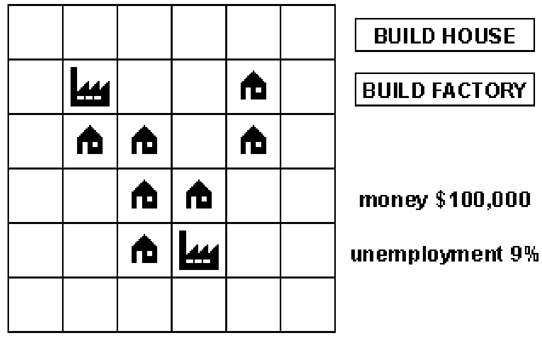

Figure 1: FACTORY: a simple resource management game.

First, in resource management games, we see that some of the required capabilities for computer-controlled players are typically provided by symbolic AI approaches, such as the ability to plan, whereas other skills are typically provided by behavior-based approaches, such as the ability to deal with non-determinism.

Second, in most resource management games, the capacity to plan is especially important for a player to be successful, as has been explained above. Intuitively, we would say that symbolic planning methods would therefore be a suitable way of dealing with resource management games [Fasciano, 1996]. However, planning in a dynamical environment with many parameters, such as a resource management game, leads to search spaces that are usually too large to handle with regular planning algorithms. Moreover, many games do not provide their players with perfect information – for example, in SIMCITY, a variety of random events takes place. Finally, if a game includes some form of competition, such as the game DUNGEON KEEPER, it becomes highly non-deterministic and requires the player to perform adversarial planning, which is harder than regular planning.

Third, research supports the power of a hybrid approach in many genres of interactive computer games. For example, a hybrid AI approach leads to satisfying performance in role playing games [Spronck *et al.*, 2003] and strategy games [Ponsen and Spronck, 2004].

In [de Jong, 2004], research is presented that compares the performance of a hybrid AI approach to that of a purely symbolic and a purely behavior-based approach, to determine whether hybrid AI is indeed a good approach of choice for resource management games. For this comparison, three solution methods have been devised, viz.

1. *a behavior-based approach using a neural network*. The input of the fully connected network consists of the values of all parameters in the game world. The output consists of a preference value for each action. Thus, in each round, the network is used to derive preferences for all actions, based on the current state of the game. The action that is most preferred is then executed, if possible. The neural network is trained by weight optimization with a genetic algorithm, by means of offline learning.
2. *an abductive planner*. This method builds a search tree, consisting of nodes representing actions and arcs representing a post-/precondition relationship between these actions (similar to figure 2). It then starts at the root of this tree and finds an action that (1) is currently executable and (2) contributes the most to the goal of the game. The latter should be determined using domain knowledge, namely a heuristic that provides the required information.
3. *evolutionary planning heuristics*. This hybrid method is a combination of the aforementioned two methods. Using a neural network as described under 1, we derive preferences for all actions, given the current state of the game. Then, these preferences are used as a heuristic for tree traversal in an abductive planner.

The approach presented under 3 follows quite intuitively from the other two approaches, since both approaches have a problem. First, the purely behavior-based approach cannot be expected to perform well in all games, because planning is a task that is too complex for a neural network. Second, as explained under 2, an abductive planner must possess domain knowledge in order to determine which action contributes most to the goal of the game. Without such domain knowledge, the planner would have to build and traverse the entire search tree, which is not a feasible task due to the size of the search space and the fact that there are non-deterministic factors. In the case of rather small single-player games, such as the games which were experimented upon, we can easily provide the planner with domain knowledge; developing a heuristic for efficient tree traversal is not difficult here. In larger games however, we would like the computer-controlled player to function without requiring additional information (such as a heuristic) programmed in ahead of time by developers. As has been mentioned earlier, many commercial games are changed frequently to resolve balance issues, and in practise, this often means that the game AI needs to be changed as well. An adaptive system has an obvious advantage here.

The evolutionary planning heuristics approach deals

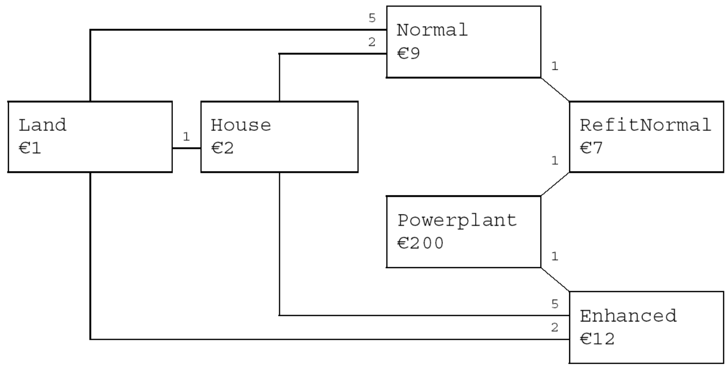

Figure 2: The schedule of actions in the experimental game PLANNING II.

with these two problems by providing a heuristic that is not developed ahead of time, but learned by repeatedly playing the game. This leads to both a less complex task for the neural network part of the method, and an adaptive heuristic for the abductive planner part.

5 Experiments and results

This section presents the results obtained by [de Jong, 2004] using the methods described in the previous section. The power of these methods has been assessed in small resource management games that are quite complicated to solve and representative for many games in the genre. All experiments were performed in the resource management game builder TAKING CONTROL [de Jong, 2004], which enables users to define and play their own resource management games, based on a rigid but feature-rich formalism of parameters and actions.

We will now provide a more detailed analysis of the game developed for one particular experiment, namely the game labelled PLANNING II in [de Jong, 2004]. The game is reminiscent of the FACTORY game discussed earlier. The player must build housing and factories. The location of these buildings does not effect the game's progress – only the quantity is a factor of influence. There are two types of factories, viz. coal-operated factories and electricity-operated factories – the latter are more enhanced and produce more output (i.e., more money). In order to be able to build enhanced factories, the player must first build a power plant, which is an expensive building. In addition to building new enhanced factories, it is also possible to refit existing normal, coal-operated factories, which then become enhanced ones. A schedule of actions and preconditions in this game is represented in figure 2. For example, the figure shows that in order to build a normal (factory), a player must acquire 5 pieces of land and 2 houses, and must invest 9 euros. The goal of the game is to make as much money as possible within a given number of turns, for example 300, with a given amount of money in the first round, for example 50 euros.

We will look at three interesting aspects of the game to gain insight into its complexity. First, we can observe that the game is clearly divided into three phases. In the first phase, a player must strive for as many normal factories as possible. This phase ends when the player has earned enough money to buy a power plant with. A short second phase of the game then starts, ending with the power plant being built. Then, there is a third phase in which the player must try to obtain as many enhanced factories as possible, either by refitting normal factories or by building completely new enhanced factories. Thus, in each of the phases of the game, the player must pursue different objectives, and in the last phase, there are alternatives for reaching the objective.

Second, we can observe that it is possible to determine the search space size of this game. If it is played for only 300 rounds, the search tree resulting from it contains $7^{300} \approx 3.4 * 10^{253}$ possible action sequences, since in each round, the player can choose one of the six actions or pass [de Jong, 2004]. Clearly, an undirected search process in this tree will not finish within feasible time.

Third, it is possible to derive a good solution for this game by implementing a rule-based approach with a set of intuitive rules such as: $powerplants = 1 \wedge money > 6 \wedge normalfactories > 0 \rightarrow RefitNormalFactory$. The solution obtained by this method in case of a game starting with 50 euros and running for 300 rounds, turns out to be 15,780 euros [de Jong, 2004].

In the experiment, the score of the three methods presented earlier has been compared to that of the rule-based approach. A first notable result is that the purely behavior-based approach does not find a satisfactory solution even after hundreds of generations; its best performance is a game in which nothing at all is built, resulting in a score equal to the initial amount of money, i.e., 50 euros. Depending on the heuristic used, the abductive planner is able find the same solution as the rule-based approach (15,780 euros), but as has been mentioned before, it then uses knowledge that has been added ahead of time; without such knowledge, the search space cannot be traversed efficiently by the planner. The evolutionary planning heuristics approach is also able to find this solution of 15.780 euros, but it does not require the inclusion of additional knowledge. It does require time to learn (approximately 100 evolutionary runs of 50 individuals each), but for a problem with a search space size of $3.4 * 10^{253}$ action sequences, this can be considered to be an excellent achievement.

Four additional experiments using hybrid AI methods are described in [de Jong, 2004]. The evolutionary planning heuristics approach has been used in one of these additional experiments, with comparable results. At the time of writing, large quantities of complex resource management problems are being addressed by the three methods presented, with preliminary results indicating that evolutionary planning heuristics outperform both other methods on average, due to the fact that the heuristics found are actually better suited to the problems than the heuristics developed by hand by the developers of these problems.

All experiments performed lead to the same conclusion, namely that hybrid AI approaches are able to address many of the skills required for the direction of a computer-controlled player for resource management games, without requiring developers to program significant parts of the behavior ahead of time. Instead, the hybrid AI approach uses trial and error to find satisfactory strategies. This makes hybrid AI approaches both quite straightforward to develop and highly adaptive when it comes to changes in the game at hand. If the definition of a game changes (for example, a power plant now costs 100 euros), all we need to do is perform a new learning process which, possibly starting from the current solution method, will derive a better solution method.

6 Conclusions and future work

Developing computer-controlled players for resource management games is a challenging task, because these games have many layers of complexity, arising from a small and understandable set of actions. Playing a resource management game requires many skills that currently only human intelligence possesses sufficiently. Furthermore, resource management games are relevant for research because their degree of realism makes them valuable tools for educative purposes.

Exploratory research in the domain of computer-controlled players for resource management games shows that hybrid AI approaches, in which various techniques from symbolic and behavior-based AI are combined, are able to learn how to play our simple resource management game, without requiring the programming of significant elements of the player's behavior ahead of time. This indicates hybrid AI approaches might be fair choice for the development of computer-controlled players for resource management games, especially since they are straightforward to develop and robust to changes in game definitions.

For future research, it would be important to determine whether hybrid approaches can be used in more complex games than the examples presented here, such as product chain management games and games in which the location of objects built plays an important role. Moreover, the principle of competition and its relationship to adversarial planning should be addressed.

References

[Buro, 2004] Michael Buro. Call for AI research in RTS games. In *Proceedings of the AAAI-04 workshop on AI in games, San Jose 2004,* pages 139–142, 2004.

[de Jong, 2004] Steven de Jong. Hybrid AI approaches for playing resource management games. Master's thesis, Universiteit Maastricht, the Netherlands, 2004.

[Fasciano, 1996] Mark Fasciano. Real-time case-based reasoning in a complex world. Technical report, The University of Chicago, USA, 1996.

[Jayanti, 2003] Vikram Jayanti. Game over: Kasparov and the machine. Documentary, 2003. http://www.imdb.com/title/tt0379296.

[Laird and van Lent, 2001] John Laird and Michael van Lent. Human-level AI's killer application: Interactive computer games. *AI Magazine,* Summer 2001:15–25, 2001.

[McCarthy, 1990] John McCarthy. Chess as the drosophilia of AI. In T. A. Marsland and J. Schaeffer, editors, *Computers, Chess, and Cognition*, pages 227–237. Springer-Verlag, 1990.

[Pfeiffer and Scheier, 1999] Rolf Pfeiffer and Christian Scheier. *Understanding Intelligence*. MIT Press, 1999.

[Ponsen and Spronck, 2004] M. Ponsen and P.H.M. Spronck. Improving adaptive game AI with evolutionary learning. In Q. Mehdi, N.E. Gough, S. Natkin, and D. Al-Dabass, editors, *Computer Games: Artificial Intelligence, Design and Education (CGAIDE 2004)*, pages 389–396, Wolverhampton, UK, 2004. University of Wolverhampton.

[Russell and Norvig, 2003] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition, 2003.

[Sawyer, 2002] B. Sawyer. Serious games: Improving public policy through gamebased learning and simulation. *Foresight and Governance Project, Woodrow Wilson International Center for Scholars*, 2002.

[Scott, 2002] B. Scott. The illusion of intelligence. In S. Rabin, editor, *AI Game Programming Wisdom,* pages 16–20. Charles River Medias, 2002.

[Spronck *et al.*, 2003] Pieter Spronck, Ida Sprinkhuizen-Kuyper, and Eric Postma. Improving opponent intelligence through offline evolutionary learning. *International Journal of Intelligent Games and Simulation*, Vol.2, No.1:20–27, 2003.