ELSEVIER
Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 24 (2013) 143 – 151
Procedia
Computer Science

17th Asia Pacific Symposium on Intelligent and Evolutionary Systems, IES2013

Towards Believable Resource Gathering Behaviours in Real-time Strategy Games with a Memetic Ant Colony System

Xianshun Chen[a], Yew Soon Ong[a,*], Liang Feng[a], Meng Hiot Lim[b], Caishun Chen[a], Choon Sing Ho[a]

[a] *Multi-plAtform Game Innovation Centre (MAGIC), School of Computer Engineering, Nanyang Technological University, Singapore*
[b] *School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore*

Abstract

In this paper, the resource gathering problem in real-time strategy (RTS) games, is modeled as a path-finding problem where game agents responsible for gathering resources, also known as harvesters, are only equipped with the knowledge of its immediate surroundings and must gather knowledge about the dynamics of the navigation graph that it resides on by sharing information and cooperating with other agents in the game environment. This paper proposed the conceptual modeling of a memetic ant colony system (MACS) for *believable* resource gathering in RTS games. In the proposed MACS, the harvester's path-finding and resource gathering knowledge captured are extracted and represented as memes, which are internally encoded as state transition rules (memotype), and externally expressed as ant pheromone on the graph edge (sociotype). Through the inter-play between the memetic evolution and ant colony, harvesters as memetic automatons spawned from an ant colony are able to acquire increasing level of capability in exploring complex dynamic game environment and gathering resources in an adaptive manner, producing consistent and impressive resource gathering behaviors.

Selection and peer-review under responsibility of the Program Committee of IES2013

Keywords: Memetic Computing; Ant Colony; Resource Gathering; Path Finding

1. Introduction

With the advancements in graphics and physics engines reaching the stage of maturity, game researchers are now turning to artificial intelligence (AI) as an increasingly influential factor for enhancing the quality of user experiences and furthering the growth of game innovation and believability. At the same time, due to the dynamic nature and complexity of digital games, many challenges in the field have posed as interesting problems that can be addressed using artificial intelligence and computational intelligence (CI) technologies. One of these challenges is the dynamic path-finding and resource gathering problem commonly found in real-time strategy (RTS) games. In RTS games, the primary objective of the game is to build an army strong enough to destroy the opponent base without getting

* Corresponding author. Tel.: +65-6790-6448 ; fax: +65-6792-6559.
E-mail address: asysong@ntu.edu.sg

Selection and peer-review under responsibility of the Program Committee of IES2013
doi:10.1016/j.procs.2013.10.037

overrun by the enemies. In order to do that, players are supposed to build and strengthen their base by purchasing various upgrades and soldiers using the available game resources provided. Efficiency in gathering these scarce and limited resources is often one of the key factors that determines the eventual outcome of the game, especially when the location of these game resources are not known beforehand since they are often are randomly generated. This can be a challenging task for AI if the computer controlled players, often known as a bots, were to play on equal grounds with the human players, meaning to say that the bots are not equipped with complete information on the locations of the resources. Rather, they have to explore the map extensively in order to locate the resources. While players have come to expect increasing levels of AI sophistication in the resource gathering and path-finding behaviors, the modern commercial game environments for RTS games are becoming frequently non-deterministic with imperfect knowledge and a large number of rather dynamic variables. Furthermore, the dynamic path-finding and resource gathering behaviors in these games have to deal with the vast number of game agents frequently as well as the time synchronization among them in a large-scale persistent game world with the use of intensive computational resources. Consequently, dynamic path-finding and intelligent gathering of arbitrarily valued resources in RTS games represent a fundamental problem of high complexity, and with potential merits if appropriately addressed [1].

RTS games with particularly innovative AI for dynamic path-finding and resource gathering can stand out among competitors. Such innovative AI is related to the general principle of *believability*. As discussed in [2], believable agent is described as one that provides the illusion of life, and thus permits the audience's suspension of disbelief. A believable agent is usually characterized with human-like peculiarities, such as the capabilities to learn, to cooperate, to make mistakes, to adjust its own strategy in response to the opponent's actions. In this work, the RTS resource gathering problem to be solved here is first defined as a path-finding problem, while game agents that are responsible for gathering the resources shall be referred to as harvesters in this paper. It is assumed that harvesters have very limited knowledge of the digital game world, apart from its immediate or local surroundings. In contrast to the A-Star search, which requires complete information of the navigation graph and essentially cheats to know the resource location ahead of time, the harvester bots must now explore and discover resource locations by sharing information and cooperating with other harvesters in its surroundings. This clearly puts the harvester bots in more equal footings with the human players in the game play, hence providing greater believability in the resultant RTS game. Achieving cooperative behavior is often difficult, as maintaining shared information about the dynamics of agents in the game world can be complex. One approach is via explicit planning and search [3,4], which has been used to tackle the cooperative path-finding problem that require multiple agents to follow non-interfering paths from the current states to their respective goal states. The approach is however computationally expensive, though well-suited for handling complex navigation problems where movement is constrained. An alternative approach is based on swarm intelligence and the flow-field techniques. Among such an approach is the direction map proposed in [5,6], which presents a distributed approach whereby agents share information about the direction in which they traveled when passing through each checkpoint. The information on each checkpoint then serves to encourage agents that pass through the same location to travel in the same direction as past agents. The Ant Colony Optimization (ACO) proposed in [1] is another example on the use of swarm intelligence for RTS resource gathering. The distributed nature of swarm intelligence is well suited for maintaining shared information about the dynamics of agents in the game world. Nevertheless, the approach is slow in converging to the best path-finding solution as they are poor in efficiently concentrating around the best path-finding solutions. As a result, harvesters may be spending lots of time wandering and looking silly before they concentrate on the optimal paths [1]. One other limitation of such computational intelligence method is that its performance is highly sensitive to parameter tuning and hence leading to undesirable path-finding behaviors when the navigation graph is highly dynamic in nature, which is not acceptable to game developers. Furthermore, the current swarm-based cooperative path-finding approach including [5,6,1] are not designed for game environments that are based on navigation mesh [7], which have become the search space representation of choice for path-finding in digital games [8,9].

In this paper, our interest is on the social behavior of harvester bots in RTS resource gathering tasks. In particular, we propose a *M*emetic *A*nt *C*olony *S*ystem (MACS) for facilitating *believable* resource gathering behaviors on navigation graph in real-time strategy games. In the proposed system, the harvesters (i.e., ants) leave their base location (i.e., the ant colony nest), seek resources and drop pheromone on the way back to the nest. By following pheromone trails, the harvesters are assured a path to resources and back home. The more ants that travel the path with resources found, the stronger the pheromone grows [1]. While this path-finding search framework is based on Ant Colony Optimization

(ACO) and similar to that in [1], our proposed approach differs in three core manner. Firstly, in contrast to [1], our path-finding task focuses on navigation graphs, which is the typical search space representation for path-finding in digital games [8,9]. Secondly, an individual learning phase in the form of a partial A-Star search is carried out by the harvester whenever resources is located, which enhance the colony in converging to the best path-finding solutions more efficiently. Third, memetic computation [10], specifically memetic automaton and the process of imitation, are integrated with the ant colony learning mechanism to form a meme-centric framework that adapts agents' behaviors according to the dynamics of the navigation graph. This effectively eliminates the need for tedious parameter fine-tuning in ACO search framework.

Memetic computation is a paradigm that uses the notion of meme(s) as units of information encoded in computational representations for the purpose of problem-solving [10,11,12]. Meme(s) on its own is perceived as a form of structured knowledge, for example, in the form of recurrent patterns. The proposed MACS is a meme-centric Ant Colony System (ACS) framework in which the harvester's path-finding and resource gathering knowledge captured are extracted and represented as memes, which are internally encoded as the state transition rules (*internal logic* or *memotype*) and externally expressed as ant pheromone on the graph edge (*behavior* or *sociotype*). The *memotypes* exist within the mind universe of a harvester and serves as the *internal logic*s used by the harvester in selecting the next edge to travel when it reaches a particular junction of the navigation map. On the other hand, the *sociotypes*are expressed as the resultant pheromone deposited on the traveled edges by the harvester. These scents are then picked up by other harvesters subsequently reaching the same location, which may then imitate the previous harvester's *behavior* in selecting the next edge. While the sociotypes are being evolved by pheromone evaporation and deposition process within the Ant Colony System, the memetic evolution takes place at the same time within the harvesters's mind universe, whereby memotypes undergo selection, transmission and variation. Through the inter-play between the memetic evolution and ant colony, harvester bots or more precisely memetic automaton, are able to acquire increasing level of capability in exploring complex dynamic game environment and gathering resources in an adaptive manner, producing consistent and impressive resource gathering behaviors [11,13].

The rest of the paper is organized as follows: Section 2 begins with the detailed description of the RTS resource gathering problem to be solved in which harvesters are only equipped with the knowledge of its immediate surrounding. Subsequently, Section 3 discusses the design of the Memetic Ant Colony System, focusing on the core contributions towards facilitating *believable* resource gathering behaviours in RTS games, namely, memetic representation, ACS and individual learning, as well as memetic evolution. In Section 4, the empirical assessment of MACS is considered. Empirical results obtained show that MACS producing consistent and impressive resource gathering behaviors even when the harvesters are only given the limited knowledge of its immediate surroundings during resource gathering. Finally, the last section ends with a brief conclusion of the presented research.

2. RTS Resource Gathering Problem

Resource gathering is arguably one of the most fundamental AI task in RTS games. In the resource gathering problem we defined here, the harvester is only given the knowledge of its immediate surrounding. Suppose that the navigation graph is denoted as $G = (V, E)$, which consists of a set of vertices, $V = \{v_i\}$ and a set of edges $E = \{e_{ij}\}$ in which an edge $e_{ij} = (v_i, v_j)$ connect vertices v_i and v_j. We denote $G_k(t) \subset G$ as the partial graph visible to harvester $agt(k)$ at time step t, and we further denote $E_i = \{e_{ij} | e_{ij} = (v_i, v_j), e_{ij} \in E, v_j \in V\}$ and $V_i = \{v_j | v_j \in e_{ij}, e_{ij} \in E_i\}$ as the set of edges and vertices connecting to the vertex v_i. When harvester $agt(k)$ reaches vertex v_i at time step t, its visibility only extends up to V_i, E_i, i.e., $G_k(t) = (V_i, E_i)$. With such a limited visibility, the harvester $agt(k)$ must gather knowledge about the graph structure of the navigation map on which it explores by sharing information and cooperating with other harvesters in the game environment.

A harvester is only given the knowledge of its immediate surroundings in resource gathering. This constraint is important from the practical point of views in the context of RTS games. Firstly, modern game environments are becoming frequently nondeterministic with imperfect knowledge and large numbers of quite dynamic variables [14,15], e.g. the obstacles may change location such as sudden destruction of a bridge, and the resources and navigation mesh may also change dynamically. As a result, the creation of an AI resource gathering subsystem must be designed with minimum use of the preprocessed information (e.g. the full visibility of the entire navigation graph) in order to handle a fully dynamic navigation graph that deal with fast changes. Secondly, the dynamics of the world should also take into account that the game agents can change the states of the world, e.g., a resource location may become depleted due to harvesters gathering resources from it, or soldier units causing the terrain to change after bombardment on a piece of land. Moreover, a game scene can have many thousands of moving objects at peak times [15], the resource gathering behaviors in RTS games have to frequently handle the vast number of game agents as well as the time synchronization among them in a persistent game world which can undergo large-scale physical change affecting dozen to hundreds of characters. In addition, modern RTS usually features large-scale navigation graph that may undergo large physical change as well. Consequently, from the consideration of computational efficiency, it is important to minimize the amount of information to be processed by a single harvester during resource gathering, which can be done by processing only the local navigation knowledge. Thirdly, the desired illusion of intelligence manifests best when game agents emulate the behavior of a human player. That is, believable resource gathering behaviors of the harvester should be imperfect to a degree, being either fun and beatable or perhaps simply irrational, as human beings often are. The seemingly simple aspect of seeking and returning resources takes on new meaning in RTS games when now the computer controlled players have to play on equal grounds as a human player, without cheating and equipped with the same amount of incomplete map information.

3. Memetic Ant Colony System

In this section, we introduce the proposed memetic ant colony system for RTS resource gathering problem. In each generation g, a colony of ants (i.e., harvesters) is generated, in which each harvester $agt(k)$ carries a path memory G_k and a tabu list $tl(k)$. The path memory G_k records the edges and vertices that $agt(k)$ traveled up to the current time step and the tabu list $tl(k)$ records the edges recently traveled by $agt(k)$. $agt(k)$ also carries its memotype $M(k)$, which encodes a set of state transition rules that determine which edge e_{ij} to travel next when it reaches v_i.

The harvesters start by moving out of their base location v_0 (i.e., the ant colony nest) in search of resources. As each harvester $agt(k)$ has only a limited sight range $G_k(t) = (V_i, E_i)$ when it reaches v_i, the harvester must make use of its available memotype $M(k)$ to analyze the sociotype memes available in its surroundings. These sociotype memes have been deposited as pheromone on the edges E_i by other harvesters previously passing over v_i. $agt(k)$ then imitates the behavior of the harvesters who have successfully found a resource location or explore as far as possible to search for resources. After $agt(k)$ passes over e_{ij}, it expresses a sociotype meme into the pheromone trail τ_{ij} on e_{ij} and update its tabu memory $tl(k)$ to include e_{ij}, so that harvesters can avoid potentially dangerous paths or paths that have been recently traveled by other harvesters or itself, based on the pheromone that is laid on the edges as well as its tabu memory $tl(k)$.

After a harvester $agt(k)$ successfully finds a resource location v_I, the fitness of $M(k)$ is increased by 1, i.e., $fitness(k) = fitness(k) + 1$. Subsequently, an individual learning is performed in which $agt(k)$ pathfinds the shortest path $P_{best}(v_0, v_I)$ from v_0 to v_I using partial A-Star search on its path memory G_k, while sociotype memes are then deposited as pheromone on the edge set $\{e_{ij} | e_{ij} \in P_{best}(v_0, v_I)\}$. Periodically, the pheromone on the edges traveled by all harvesters are evaporated to encourage the harvesters in selecting paths that will lead them to already found resource locations as well as to encourage them in exploring edges that have not yet being uncovered, in the hope of finding new additional resources.

At the same time, when harvesters perform the ACS-like resource gathering activity, harvesters self-configure their memotype encoded transition rules by subjecting them in a memetic evolutionary cycle, where memotype undergo meme selection, transmission and variation in the harvester's mind universe according their fitness.

The MACS search scheme include three key aspects: i) The memotype and sociotype representations of memes in the MACS, ii) the ACS and A-Star individual learning mechanisms that promote the imitation of sociotype memes among harvesters, and iii) the memetic evolution mechanisms that govern the evolution of memotype encoded state transition rules.

3.1. Memetic Representation

In memetic computation, the memotype representation refers to knowledge or memes internal to the mind universal of an harvester, while sociotype refers to the social expression of a meme. In the proposed MACS, sociotype is expressed in the form of pheromone $\tau(r, s)$ that is deposited on the edge (r, s). On the other hand, the memotype representation of memes encode the state transition rules to provide a direct way in balancing between exploration of new edges and exploitation of *a priori* and accumulated knowledge about the path-finding to known resource locations. Structurally, the memotype representation $M(k)$ is a continuous-value string that encodes the following state transition rules:

$$M(k) = \boxed{T_k \mid \rho \mid l_k \mid Q_0 \mid a_{min} \mid b_{min}}$$

where T_k is the tabu tenure (i.e. the max length) of the tabu list $tl(k)$ carried by harvester $agt(k)$, *rho* is the local pheromone update parameter for $agt(k)$ to express its sociotype memes into the pheromone trail on the edge it just traveled. l_k is the lifetime of $agt(k)$, Q_0 is a parameter that balances between exploitation and exploration in the state transition rules. a_{min} and b_{min} are inhibit parameters that discourage harvesters from traveling to vertices that have been recently traveled by the harvester or potentially dangerous (e.g., obstacles ahead).

The memotype and sociotype memes in MACS work in a complementary nature. Memotype are internal logic blocks of an harvester when making decision on path-finding, while the sociotype are behaviors expressed by harvesters which are transmitted to other harvesters via pheromone. When reaching a vertex r, harvester $agt(k)$ may decide to imitate a particular successful harvester $agt(l)$ previously passing this vertex, and select the edge $agt(l)$ has traveled. The behavior imitation is accomplished via the memotype state transition rules and sociotype pheromone information as follows: a harvester $agt(k)$, whose current visited vertex is r, chooses the next vertex s from its surrounding vertices $V(r) = \{\mu|(r, \mu) \in E\}$ to move by applying a set of state transition rules. If $V(r)$ contains a resource location v_I, then $s = v_I$. Otherwise s is selected from $V(r)$ using the pseudo-random-proportional rule:

$$s = \begin{cases} arg\max_{\mu \in V(r)} P(\mu) & \text{if } p \le Q_0; \\ S & \text{if } p > Q_0. \end{cases} \tag{1}$$

In Eqn. 1, $P(\mu)$ denote the preference for selecting vertex $\mu \in V(r)$, p is a random number that is uniformly distributed in $(0, 1]$ and Q_0 is taken from $M(k)$. If $p \le Q_0$, the pseudo-random-proportional rule favors exploitation (s is chosen as the vertex such that edge (s, r) is the edge on which the preference $P(\mu = s)$ is the highest). If $p > Q_0$, the algorithm favors exploration and s is assigned a random variable S selected with the probability $P(S, r)$:

$$P(S, r) = \frac{P(S)}{\sum_{u \in V(r)} P(u)} \tag{2}$$

Eqn. 2 is known as the random-proportional rule in that it favors transitions toward vertices having higher preference $P(\mu)$.

For each neighboring vertex $\mu \in V(r)$, its preference $P(\mu)$ is given by:

$$P(\mu) = \begin{cases} \tau(r, \mu) & \mu \notin tl(k) \\ \tau_{min} * a_{min} & \mu \in tl(k) \end{cases} \tag{3}$$

where $\tau(r, \mu)$ is the pheromone on the edge (r, μ), and τ_{min} is the minimum pheromone value. $0 \le a_{min} \le 1$ is from $M(k)$. Eqn. 3 discourages the harvester $agt(k)$ from selecting a vertex that is in its tabu list $tl(k)$.

The state transition rules in Eqns. 1, 2, 3 for resource gathering differ from that of the ACS state transition rules in traditional TSP path planning in that there is a probability that the harvester $agt(k)$ will revisit a vertex which it has visited in the past, so as to explore its neighboring vertices again in hope of discovering a resource location. Furthermore, it does not assume a fully connected graph as in TSP path planning.

Initially, the pheromone level of any edge not traveled by any harvester is assumed to have a pheromone level of τ_0. After $agt(k)$ travels the edge (r, s), it expresses its temporal presence at the edge (r, s) via pheromone update as given by:

$$\tau(r, s) \leftarrow (1 - \rho) \cdot \tau(r, s) + \rho \cdot \tau_0 \tag{4}$$

where the local decay parameter $0 < \rho < 1$ is taken from the memotype $M(k)$ of harvester $agt(k)$. If $agt(k)$ dies as a result of an ambush at vertex s, it also communicates this acknowledgement on edge (r, s) via the pheromone update as given by:

$$\tau(r, s) \leftarrow \tau_{min} * b_{min} \tag{5}$$

where the pheromone inhibit parameter $0 < b_{min} < 1$ is taken from the memotype $M(k)$ of harvester $agt(k)$.

Eqns. 4 and 5 allow harvester to share knowledge via sociotype meme expression in terms of pheromone changes on the edge. Thus subsequent harvesters can avoid paths potentially dangerous, or recently traveled by other harvesters or by itself.

3.2. ACS and Individual Learning

Here we detail the Ant Colony System (ACS) process of pheromone evaporation and deposition as well as the individual learning involved in the resource gathering. After an harvester $agt(k)$ successfully finds a resource location v_I, the path-finding knowledge or sociotype meme can be transmitted to other harvesters via the global pheromone update, in which $agt(k)$ raises the pheromone level on the traversed edges stored in its path memory G_k. However, $agt(k)$ may have to take many unnecessary twist and turns during its search for the resource location. Thus it is important to locally refine its path memory G_k. The local refinement is performed through an individual learning stage in which $agt(k)$ path finds the shortest path $P_{best}(v_0, v_I)$ from v_0 to v_I using partial A-Star search on its path memory G_k. The A-Star search is partial as it is only restricted to the partial graph represented by G_k. Sociotype memes are then deposited as pheromone on the edge set $\{(r, s)|(r, s) \in P_{best}(v_0, v_I)\}$, using the global update rule as given by:

$$\tau(r, s) \leftarrow \tau_{max} \forall (r, s) \in P_{best}(v_0, v_I) \tag{6}$$

where τ_{max} is the maximum pheromone level.

Periodically, the pheromone on the edges traveled by all harvesters are evaporated, to encourage the harvesters in selecting paths that will lead them to the found resource locations

$$\tau(r, s) \leftarrow (1 - \alpha) * \tau(r, s), \forall (r, s) \in G_k, \forall k \tag{7}$$

It is worth noting that Eqn 7 only applies to edges that have been traversed by the harvesters, while the pheromone level of any untraversed edge is maintained at τ_0. This is to encourage harvesters in exploring edges that have not covered in the hope of finding more resources, particularly when the current resources found have been depleted or when the previously found paths to the resource locations are blocked.

3.3. Memetic Evolution

Many of the parameters within the memotype memes of an harvester specify the behavior of the state transition rules during path-finding. The role of memetic evolution is to auto-tune the behavior of the state transition rules so that the harvester can self-adapt to the dynamics of the navigation graph. The memetic evolution inter-plays with the ACS resource gathering described in Subsections 3.1 and 3.2. In the lifetime of an harvester $agt(k)$, each time it successfully finds a resource location, the fitness $fitness(k)$ of its currently associated memotype meme $M(k)$ is increased by 1, i.e., $fitness(k) = fitness(k) + 1$. This gives rise to a selection pressures among the memotype memes residing in the meme pool constituting memotypes memes being used by harvesters. In a memetic evolution cycle, these memotype memes undergo meme selection, transmission and variation according their fitness $fitness(k)$, which give rise to the new generations of memotype memes. In our current implementation, a binary tournament selection scheme is used to select the memotype memes either for path-finding or memetic reproduction. Redcliff Blending is used for memetic transmission in which a new meme is produced by inheriting properties from its parent memes. With respect to the memetic variation process, a normal distribution mutation is applied to heuristically modify the memetic materials.

The memetic evolution allows harvesters to adapt their state transition rules to the dynamics of a game world which can undergo fast and large-scale physical change affecting dozen to hundreds of game agents, by balancing between exploitation and exploration on the navigation graph.

4. Empirical Study

In this section, we evaluate the proposed memetic ant colony system for resource gathering on a simulated RTS environment, which is depicted in Fig. 1. In the simulated environment, the graph for resource gathering is randomly

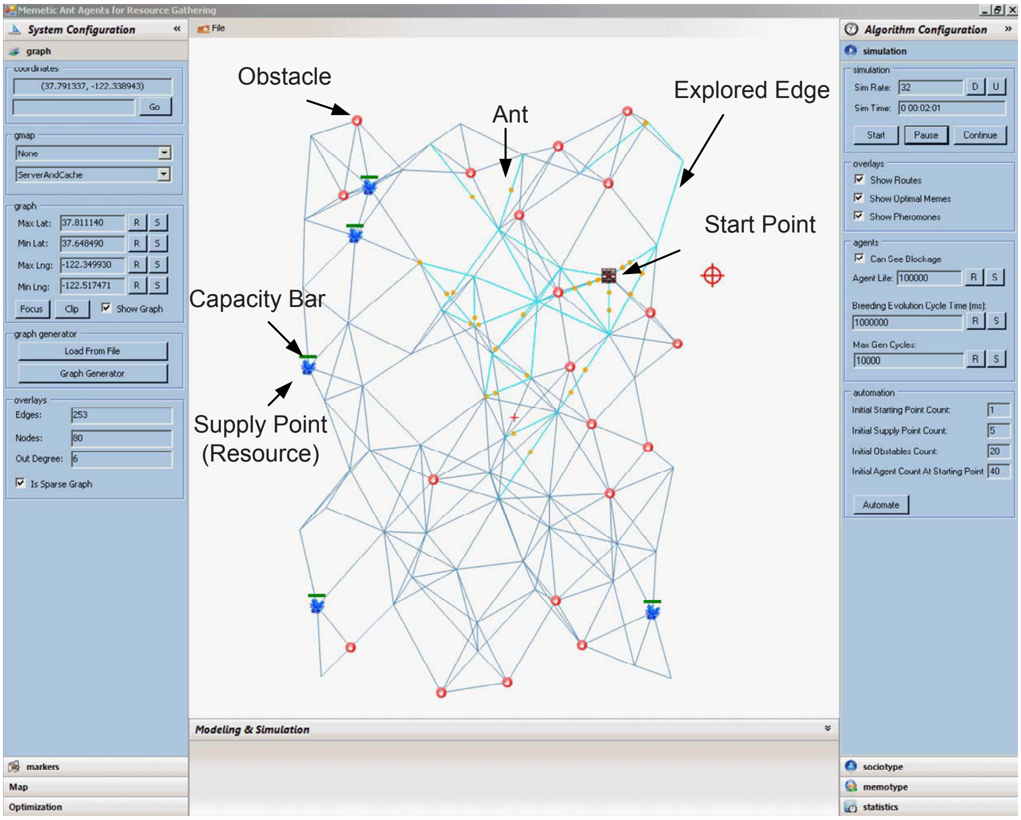


Fig. 1. Memetic Ant for Resource Gathering.

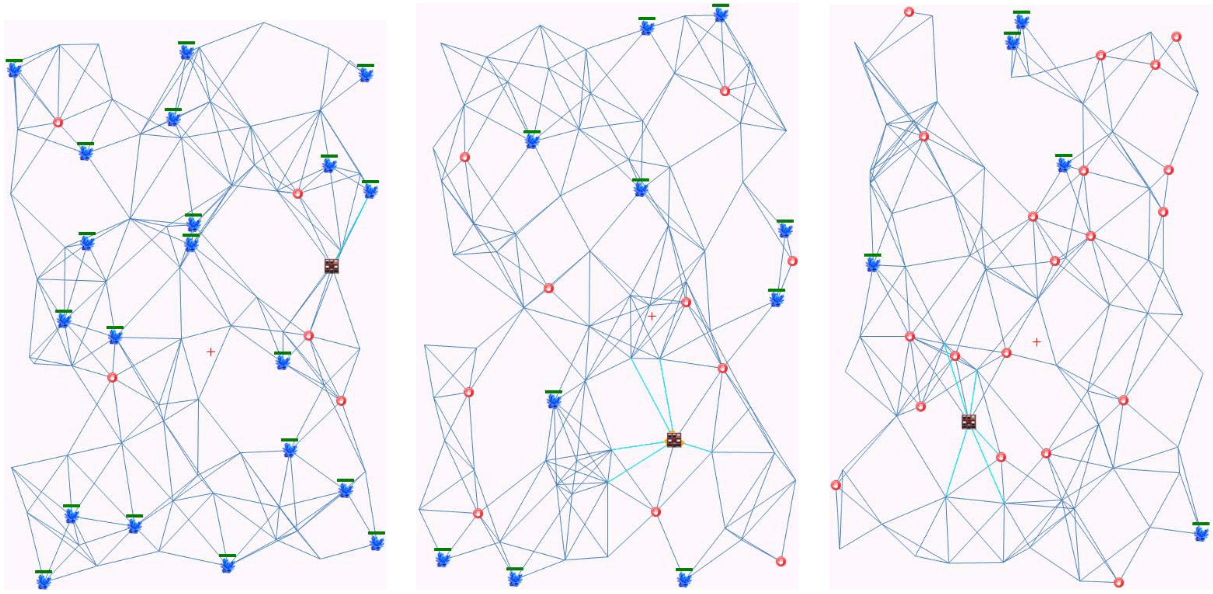
Table 1. Summarization of parameter configurations in plentiful supply, medium supply and little supply.

	Supply Point	*Obstacle*	*Memetic Ants*
Plentiful Supply	20	5	20
Medium Supply	10	10	20
Little Supply	5	20	20

generated. We can configure the number of start point, obstacles, supply points (resource) and ants at the right plane as shown in Fig. 1. Each supply point has a capacity as shown on its top, and the capacity bar will become empty when the supply point has been exhausted. The edge with highlighted color denotes the path that has been explored by the memetic ants. More importantly, the whole structure of the generated graph is non-familiar to the memetic ants, and only the immediately surrounding information is available to the respective memetic ants.

Further, three scenarios of resource gathering, namely plentiful supply, medium supply and little supply are studied in this experiment. In particular, for plentiful supply, extensive supply points are available to the memetic ants, and little obstacles existed on the graphs. Medium supply has equal size of supply points and obstacles on the navigation graph, while little supply holds lots of obstacles in the navigation graph but little supply points available to the memetic ants. The detailed parameter configuration of each scenario is summarized in Table 1. Fig. 2 presents the navigation graph and respective distribution of supply points and obstacles in each resource gathering scenario considered. Since the number of memetic ants is kept fixed in all the three scenarios, the difficulty of resource gathering task increases from plentiful to little supply.

All the simulated results of the proposed memetic ant colony system on plentiful supply, medium supply and little supply resource gathering scenarios are presented in Fig. 3. As can be observed, the resources in both easy (i.e., Fig.



(a) Plentiful Supply (b) Medium Supply (c) Little Supply

Fig. 2. Distributions of Supply of Plentiful Supply, Medium Supply and Little Supply Resource Gathering Scenarios.

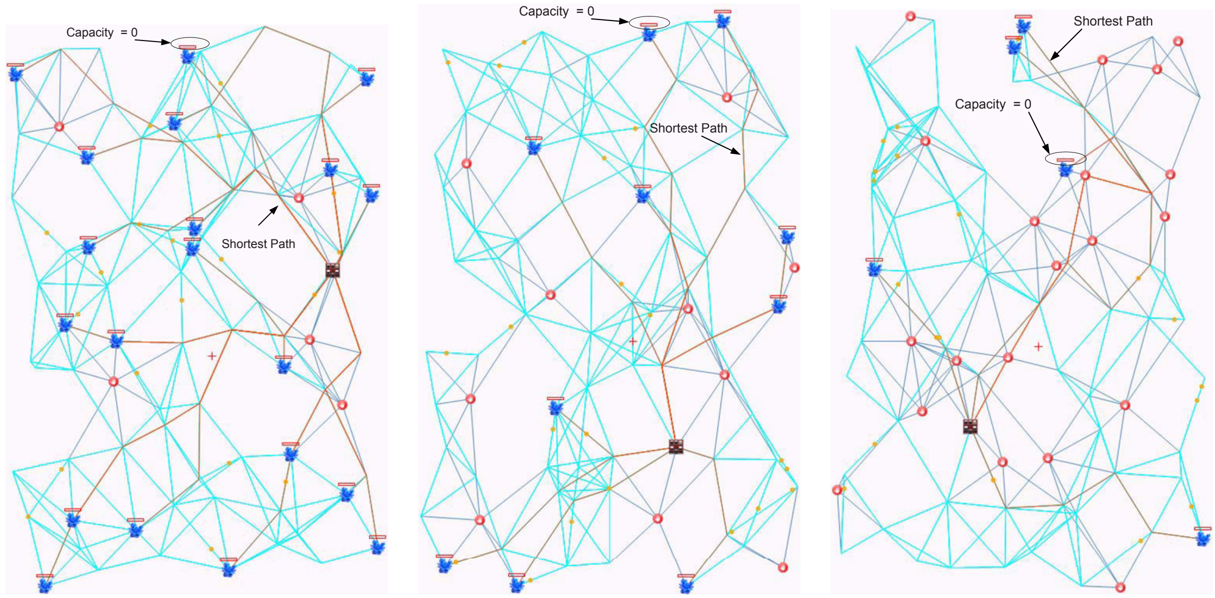
3(a) plentiful supply) and difficult (i.e., Fig. 3(c) little supply) resource gathering have all been successfully found by the memetic ants (i.e., *Capacity* = 0). The shortest paths from the start point to each supply point (resource) are all highlighted in red. It is worth noting that the graphes of the three resource gathering scenarios considered are all different, the consistent performance of the proposed MACS confirmed its effectiveness in path-finding when limited information of the navigation graph is available to the harvesters.

5. Conclusion

This paper proposes the conceptual modeling of a memetic ant colony system (MACS) for facilitating *believable* resource gathering behaviours in RTS games, where harvesters are equipped with only limited knowledge about its immediate surroundings. In the proposed MACS, the harvester's path-finding and resource gathering knowledge captured are extracted and represented as memes, which are internally encoded as state transition rules (memotype) and externally expressed as ant pheromone on the graph edge (sociotype). Through the inter-play between the memetic evolution and ant colony, harvesters as memetic automaton spawned from an ant colony are able to acquire increasing levels of capabilities in exploring the complex dynamic game environment and gathering resources in an adaptive manner. The empirical study shows that the proposed MACS is able to produce consistent and believable resource gathering behaviors even when only limited knowledge of its immediate surroundings is available in the process of resource gathering.

Acknowledgment

This research is supported by Multi-plAtform Game Innovation Centre (MAGIC), funded by the Singapore National Research Foundation under its IDM Futures Funding Initiative and administered by the Interactive & Digital Media Programme Office, Media Development Authority.



(a) Plentiful Supply (b) Medium Supply (c) Little Supply

Fig. 3. Paths found by The Proposed Memetic Ant Colony System in Plentiful Supply, Medium Supply and Little Supply Resource Gathering Scenarios.

References

1. J. Dunn, Ant colony organization for mmorpg and rts creature resource gathering, *AI Game Programming Wisdom* 3 (2005) 495–505.
2. I. Umarov, M. Mozgovoy, Believable and effective ai agents in virtual worlds: Current state and future perspectives, *International Journal of Gaming and Computer-Mediated Simulations (IJGCMS)* 4 (2) (2012) 37–59.
3. T. S. Standley, R. E. Korf, Complete algorithms for cooperative pathfinding problems, *in: IJCAI* 2011, pp. 668–673.
4. D. Silver, Cooperative pathfinding, *AI Game Programming Wisdom* 3 (2005) 99–111.
5. R. Jansen, N. Sturtevant, A new approach to cooperative pathfinding, *in: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 3, AAMAS '08, International Foundation for Autonomous Agents and Multiagent Systems*, Richland, SC, 2008, pp. 1401–1404.
6. M. R. Jansen, N. R. Sturtevant, Direction maps for cooperative pathfinding., *in: AIIDE'08*, 2008, pp. –1–1.
7. G. Snook, Simplified 3d movement and pathfinding using navigation meshes, *Game Programming Gems*, (2000) 188–204.
8. C. McAnlis, J. Stewart, ntrinsic detail in navigation mesh generation, *Game Programming Wisdom* 4 (2008) 95–112.
9. D. Hamm, Navigation mesh generation: An empirical approach, *Game Programming Wisdom* 4 (2008) 113–123.
10. X. Chen, Y.-S. Ong, M.-H. Lim, K. C. Tan, A multi-facet survey on memetic computation, *IEEE Trans. Evolutionary Computation* 15 (5) (2011) 591–607.
11. Y. S. Ong, M. H. Lim, X. S. Chen, Research frontier: Memetic computation - past, present & future, *IEEE Computational Intelligence Magazine* 5 (2) (2010) 24–36.
12. L. Feng, Y. S. Ong, I. W. H. Tsang, A. H. Tan, An evolutionary search paradigm that learns with past experiences, *IEEE World Congress on Computational Intelligence, Congress on Evolutionary Computation*, 1 – 8.
13. L. Feng, Y. S. Ong, A. H. Tan, X. S. Chen, Towards human-like social multi-agents with memetic automaton, *IEEE Congress on Evolutionary Computation* (2011) 1092–1099.
14. P. Marden, F. Smith, Dynamically updating a navigation mesh via efficient polygon subdivision, *Game Programming Wisdom* 4 (2008) 83–94.
15. R. Axelrod, Navigation graph generation in highly dynamic worlds, *AI Game Programming Wisdom* 4 (2008) 125–141.