Autonomous Swarms using Case-based Reasoning

Daniel O' Connor[1], Stelios Kapetanakis[2], Michael Floyd[3], Santiago Ontañon[4], Miltos Petridis[5]

[1] Keble College, Oxford University, Oxford, OX1 3PG, UK
[2] School of Computing, Engineering and Mathematics, University of Brighton, Moulsecoomb Campus, Lewes road, Brighton BN2 4GJ, UK
[3] Knexus Research Corporation; Springfield, VA, USA
[4] Department of Computer Science, Drexel University, Philadelphia, PA 19104, USA
[5] Department of Computing, University of Middlesex, The Burroughs, London NW4 4BT, UK
[1]daniel.oconnor@keble.ox.ac.uk;
[2]s.kapetanakis@brighton.ac.uk;
[3]michael.floyd@knexusresearch.com;
[4]santi@cs.drexel.edu;
[5]m.petridis@mdx.ac.uk

Abstract. This work presents a novel approach of simulating swarm computing behaviour in a sandbox environment where swarms of robots are challenged to fight against each other with a goal of "conquering" any environment bases. Swarm strategies are being used which are decided, modified and applied at run time. This work, although at its infancy, seems surprisingly applicable to several problems where combined artificial intelligence agents are challenged to generate innovative solutions and evaluate them prior to proposing or adopting the best possible one. This work is applicable in areas where AI should select fast enough within a range of available options under a multi-constraint, multi-objective mis-sion environment. Relevance to Business Process workflows is also presented and documented.

Keywords: CBR, Swarm Intelligence, Swarm Robotics, Multi-agent Systems, Real-time Strategy, Goal-driven Autonomy, Autonomous Computing

1 Introduction

Case-based reasoning (CBR) mechanics have traditionally been used as high-level rea-soning mechanisms in a variety of fields where a formal and distinct representation can describe a problem case adequately. The CBR paradigm of reasoning and learning [1] has been applied with success in finding solutions by reusing knowledge similarly to the human cognitive approach of recollecting past problems and adjust their solutions in a wide range of tasks.

Swarm Intelligence (SI) is the discipline of a collective behaviour of natural or artificial decentralized systems comprising many individuals that can govern themselves in a self-organized way. An SI system or colony usually has a population of simple units, we will refer to them as agents, that can communicate (interact) locally among each other and with/within their environment. Several examples of SI systems can be found in Natural Sciences, such as Biology, where decentralised species with no leadership or master control can demonstrate complex behaviours and intelligent global performance that is usually unknown or not possible to perform by any single individual. Several natural examples exist including bird flocking, ant colonisation, bacterial growth, animal herding, etc. Artificial Intelligence is mimicking such behaviours and several algorithms appear under SI or "SI applications in robotics".

SI algorithms may be categorised in Models of behaviour and/or metaheuristics as we will discuss in Section 2 of this work. SI can be substantially interesting in areas where multi-agent environments exist and collaborative reasoning should be applied based on simultaneous multi-sensory input from various sources. For example, this can be applied in drone swarms or Unmanned Ground Vehicle (UGV) squads that need to fulfil a versatile mission on an unknown terrain or engage in multi-objective, high-cost, high-risk missions.

This work presents a SI robot application, named RoboWars, which was developed for the needs of a real-world scenario simulator. RoboWars is a mechanical robot sim-ulation tool which can be adjusted based on real world requirements adhering to drone cases, UGV or any other autonomous mechanical application scenario provided by the user. Its mission is to be able to simulate realistic environment constraints and work as a benchmark for AI algorithm applications. This paper presents a Case-based Reason-ing application as it was applied on a variation of the Capture the Flag (CtF) game, demonstrating its applicability for versatile environment and algorithmic scenarios.

RoboWars has been used for educational purposes and at limited scale, however its applicability can be expanded significantly on open field scenarios. This paper investigates a simple mission using Case-based Reasoning and it is structured as follows: Section 2 will present the relevant work in CBR, SI and Business Workflow Scenarios; Section 3 will illustrate our case representation, environment mapping and research assumptions; Section 4 will show our system evaluation and results; and finally, Section 5 will discuss the future steps of this work and possible improvements.

2 Related Work

CBR works as a circular and problem-oriented, problem/solution-embedded process where experience supports learning [2]. CBR uses extensively any knowledge within its application domain and is based on a solid case representation and rigid similarity mechanics that allow its continuous 5-R step of retrieve, reuse, revise, review, retain.

CBR has been applied in a variety of domains with substantial success including recommender systems, business process workflows, medical domain, etc. CBR has a few examples in Swarm Intelligence applications with the most notable of Lorenzi et al. (2007) [3] in task allocation, Nouaouria and Boukadoum (2011) [4] in CBR retrieval optimisation, Ben Yahia et al. (2012) [5] in fuzzy CBR and particle swarm optimisation for decision making support and Teodorovic et al. (2013) [24] in ensemble CBR and

Bee colony optimisation for dose planning in cancer treatment. A lot of work on CBR relevant to swarm computing and complementary to our work can be seen in the fields of agent-based computing and games.

On the agent-based computing we can see the work of Floyd and Esfandiari (2011) on learning by observation [17], Sebestyénová on agent-based Decision Support systems [7], agent-based CBR for computation resource allocation within a cloud environ-ment [8], multi-based collaborative reasoning using CBR [9], ensemble CBR and multi-agents for collaborative management in supply chain [10] and distributed agent-based CBR for large scale operations [11].

CBR and games literature shows an extensive range of applications from Real-time strategical decisions [12] [13] [14], hybrid approaches combining CBR and Reinforcement Learning [15], CBR and real-time pathfinding [16] to automatic feature selection for robocup agents [18] and automatic CBR-game case generation(s) [19].

Close to our work is also the work on CBR and business process workflow monitor-ing, remedy finding and reasoning having several examples on temporal-spatial work-flows [20], [21], [22] and advanced path finding scenarios [23].

3 Environment

This section outlines the RoboWars simulator and scenarios, our case representation, and the strategies used by the swarm agents.

Environment Mapping

RoboWars simulator allows several usages based on the user requirements (real mission scenarios, experimental scenarios) and the user level (guest user, domain expert, devel-oper). Its users can load real maps of an area or opt in for the generation of a purely artificial environment. For this work we used several randomly generated maps simu-lating a capture "as many bases as possible" game. A simple description of the envi-ronment is the following: When a simulation is initiated, a random map is generated as shown in Figure 1, having an odd number of "bases".

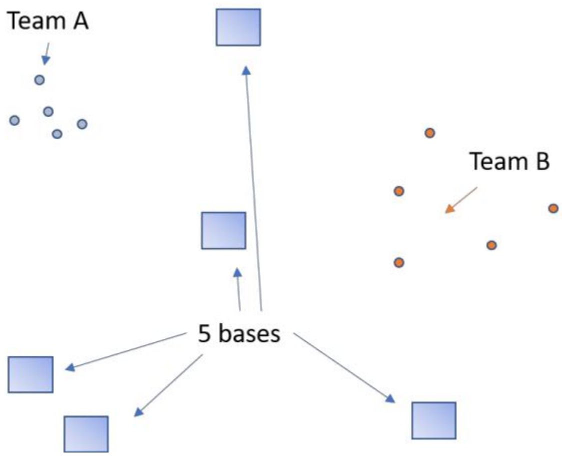

Fig. 1. Graphical representation of two (2) adversarial teams and random “bases”

The number of bases is between 3 and 7. Too few (i.e., < 3) or too many (i.e., > 7) bases were avoided since such configurations did not allow the agents to deploy their strategies efficiently (e.g., divide resources among the various bases) or the simulations took too long. Upon the successful generation of a map, two teams of 3 to 5 agents each were deployed on the map starting from opposite directions e.g. Team A on the East of the map whereas Team B on the West, North vs. South, etc. Each team’s mission was to capture as many bases as possible. Upon a successful capture each team was re-warded with a score bonus.

The simulation was over once all bases where conquered by a swarm or when a swarm had accumulated the highest score over a period. Any agent can conquer a base once it is available and undefended within 2 seconds. However, once a base has any kind of defence, to be conquered the enemy should aggregate a higher amount of force e.g. 4 agents vs. 3 to get it. In the scenarios, the agents do not destroy each other when they collide but instead push each other with a standard amount of force. For example, consider agents $_1$ and $_2$ from Team A and agents $_1{}'$ and $_2{}'$ from Team B. In a sce-nario where $_1$ comes across $_1{}'$ (top of Figure 2), each of them will push each other with force 1 . In a scenario where $_1$ comes across $_1{}'$ and $_2{}'$ (bottom of Figure 1), $_1$ will push them each with a force of 0.5 (i.e., the regular 1 force divided over two agents) whereas $_1{}'$ and $_2{}'$ will push back with a combined force of 2 .

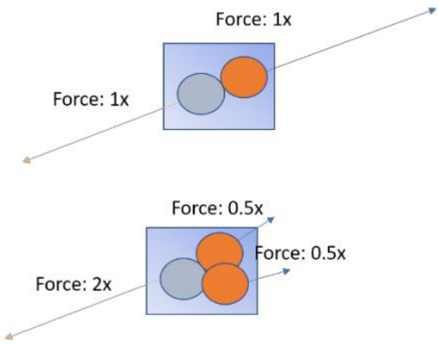

Fig. 2. Agent collisions scenarios

Case Representation

During each simulation, each team of robots has a simple mission with known success criteria. For example, missions could include: to capture and keep as many bases as possible regardless of the time required, to capture and keep as many bases as possible within a limited amount of time, or to constantly maintain a higher score than the op-ponent team. For simplicity, the mission chosen for this work was the last one: con-stantly maintain a higher score than the opponents. Each mission is regarded as an ex-ecuted workflow based on space and time and it is an instance of the overall business process which pertains to swarm domination (DOM) [6].

Since the agents are not destroyed (i.e., they push each other but do not damage each other), the difficulty and the complexity of the chosen strategies increased over time since:

1. swarm force does not increase/ overtime
2. no team has competitive advantage based on re-generation rate like in other games
3. each swarm should evolve its strategies in order to be able to win as such Each case is composed of: a team deployment over time, the actions of each team mem-ber, the team strategy, and the result of the strategy. A case $= \langle \ , \ , \ , \ \rangle$, where:

- is a set of continuous agent positions as captured per second over time. If is the position of agent at time , $= \{ \ {}_1^1, \dots, {}^1, \dots, {}_1, \dots, \}$ and con-tains $\times$ items.
- is a set of actions performed by the agents over time. If is the action performed by agent at time , , $= \{ \ {}_1^1, \dots, {}^1, \dots, {}_1, \dots, \}$ and contains $\times$ items.
- is the chosen strategy of the team
- is the actual result of using the case's strategy is a simulation

The set of available strategies comprised four options:

1. Attack the closest base
2. Attack any base
3. Defend all conquered bases
4. All agents defend one base

The set of strategies was chosen based on the mission criteria which were to max-imise the score of each team. All simulations have a maximum duration of 3 minutes, however simulations can end earlier if one team is judged to have “won”. If the score of Team A and Team B differ by more than 270 points ($| - | > 270$), the team with the higher score is said to have won that simulation.

4 Evaluation

For the evaluation of the case-based reasoning approach, three case bases were created: one for Team A, one for Team B, and a global case base which contained any “new” combination of strategies used by either of the teams. To create a baseline for our ex-periments, were ran to tackle the cold start problem of CBR (i.e., the generation of initial cases). In any of these “probe” status simulations, each team was randomly as-signed a strategy and performed agent allocations with the sole goal of completing a scenario round and learning how it’s chosen actions affected the outcome (i.e., perform exploration). A simple scenario was the following:

For Team A,
<u>random strategy chosen</u>: “attack closest base”,
<u>for all agents till the end of the simulation round</u>: from any position “attack closest base”
<u>chosen actions per agent</u>:
1 move to xposition 194, yposition 4
2 move to xposition 104, yposition 14
…
move to …
<u>result</u>: loss

We ran between 40-50 hours of simulations to generate the initial cases. These baseline simulations managed to learn cases that helped both teams to gain an initial understand-ing of the strategies and the implications of their actions. With these initial case bases, we ran an initial experiment where the teams compete against each other over an addi-tional 20 hours of simulations. Our observations were that the outcomes of these simu-lations converging to something like: “conquer a base” by accident and the “defend one base all together” strategy until the end of the round. This strategy seemed to provide the best results for any swarm.

To eliminate the case base bias, we allowed for an evolved model of choosing strat-egies where: teams could opt for different strategies over time based on the current score (e.g., if a swarm was noticing that its score was lower than its opponents it would attempt to change its strategy mode to acquire a higher score over time). Additionally, if a team was ahead of score it would attempt to maintain it by opting for a more risk-averse strategy. For this experiment the initial case bases seemed not sufficient, and the cold start training had to be repeated to have an appropriate set of cases for the swarms to choose from. We conducted 200 hours of simulations with longer simulation rounds (5 – 10 minutes each) to allow for a more comprehensive case base formulation. All the experiments afterwards contained 3-minute rounds with the swarms able to choose from any combination of strategies that would maximise their score, regardless of the time taken in the training period. This second experiment contained more than 60 hours of simulations allowing for a more comprehensive view of how swarms could behave over time while attempting to maximise their achieved scores. A few interesting obser-vations that came into light from this experiment were the following:

a) Swarms tended to reuse very often their "best" tactics. For example, a rapid succession between "attack any base" scenario and "defend our bases equally" seemed to work very well for a specific swarm and such sequences of strategies were heavily utilised across time.
b) Swarms were strongly biased upon their original training (their original case bases) and were slow enough to adjust their strategy sequences. This was expected to some extent due to the nature of CBR. However, interestingly once a better strategy sequence was achieved, it was rapidly evolving into the "most keen to use" one from the swarms.
c) Finally, since no specific time limits were defined in terms of how often a swarm can change strategy, there were rounds were a swarm could end up being "confused". In such cases a swarm would swiftly change strategy to achieve a better score and this change in decision would be changed again after a few seconds. This phenomenon was prevalent when the case base exceeded a few thousand sequences and we believe the equal "ranking" of cases made difficult to swarms to take the right decision.

Our final experiment was among the trained swarms and a new swarm, called the Golden swarm, which was trained with the global case base (i.e., the hybrid case base containing the novel cases from each case base). We ran additional simulations and observed several interesting outcomes. After several rounds of equal wins and losses two phenomena were observed as the most prevalent:

1) The Golden swarm tended to "exploit" the limitations of its opponent by resort-ing to cases that its opponent has never seen before and extreme scenarios that have probably come from a past swarm's initial training
2) The Golden swarm managed to "confuse" its opponent several times, by adopt-ing series of strategies its opponent has shown mediocre performance in the past.

5 Conclusions and Challenges

This work has investigated an interesting concept in CBR of swarm optimisation in robotics. A new simulator was developed to allow the simulation of UGVs and drones, and at the same time being able to apply different AI techniques and measure their outcomes and impact. For this work we have demonstrated several CBR vs. CBR evaluations, illustrating how a CBR system can evolve and be able to achieve superior per-formance based on its original training and after having several rounds of iterations with a worthy opponent. However, this was just a brief demonstration of what can be really achieved with the proposed tool and AI methodology.

Our future work will focus on overcoming the challenges we encountered, from both CBR and the Swarm design limitations. Our focus will be on redesigning and revaluat-ing any early steps and allow for a more advanced workflow representation and simi-larity finding e.g. consider each case's log of agent actions per second. We have ob-served cases where CBR seemed to restrict each swarm. In such cases and to allow for future evolution we are planning to investigate more appropriate techniques to allow for deviation in behaviour. Finally, more advanced robot formations and strategies and advanced teams and skill within the robots can provide a more realistic experience and adherence to real life scenarios.

Acknowledgements: We would like to thank UK Nuffield Research and the European Defence Agency (EDA) for providing support to the initial stages of this work.

References

1. David B. Leake. 2003. Case-based reasoning. In Encyclopedia of Computer Science (4th ed.), Anthony Ralston, Edwin D. Reilly, and David Hemmendinger (Eds.). John Wiley and Sons Ltd., Chichester, UK pp. 196-197.
2. Aamodt, A. & Plaza, E. (1994). Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Communications, 7(1), pp. 39–59.
3. Lorenzi, F., Scherer, D., Santos, D., de Oliveira Boschetti, D., Bazzan, A. (2007). Task al-location in case-based recommender systems: A swarm intelligence approach. Architectural Design of Multi-Agent Systems: Technologies and Techniques.
4. Nouaouria, N., Boukadoum, M. (2011). A Particle Swarm Optimization Approach for the Case Retrieval Stage in CBR. In: Bramer, M, Petridis, M and Hopgood, A. (eds) Research and Development in Intelligent Systems XXVII: Incorporating Applications and Innova-tions in Intelligent Systems XVIII Proceedings of AI-2010, The Thirtieth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, pp 209-222.
5. Ben Yahia, N., Bellamine, N., Ben Ghezala, H. (2012). Integrating fuzzy case-based reason-ing and particle swarm optimization to support decision making. International Journal of Computer Science Issues, 9 (3) pp. 117 – 124.
6. Vasta, M., Lee-Urban, S., Munoz-Avila, H. (2007). RETALIATE: Learning Winning Poli-cies in First-Person Shooter Games. In: Proceedings of the Seventeenth Innovative Applica-tions of Artificial Intelligence Conference (IAAI 2007) , AAAI Press, Menlo Park, pp. 1801-1806.
7. Sebestyénová, J. (2007). Case-based Reasoning in Agent-based Decision Support System In: Acta Polytechnica Hungarica, Vol. 4, No. 1, 2007, Ed. Andras Bako, Budapest Tech, pp. 127-138.

8. De la Prieta, F., Bajo, J., Corchado, J. M. (2016). A CBR Approach to Allocate Computational Resources Within a Cloud Platform, In: Intelligent Distributed Computing IX: Proceedings of the 9th International Symposium on Intelligent Distributed Computing - IDC'2015, pp. 75-84.
9. Manousakis-Kokorakis, V., Petridis, M., Kapetanakis, S. (2015). Collaborative Reasoning in Workflow Monitoring Using a Multi-Agent Architecture. Journal of Expert Update Vol. 15 (1) pp.37-47.
10. Fu, J., Fu, Y. (2012). Case-Based Reasoning and Multi-Agents for Cost Collaborative Management in Supply Chain. Int. Workshop Inf. Electronics, Procedia Eng. 29, pp. 1088-1098.
11. Agorgianitis, I., Petridis, M., Kapetanakis, S., Fish, A. (2016) Evaluating Distributed Meth-ods for CBR Systems for Monitoring Business Process Workflows. In proceeding of ICCBR 2016, Workshop on Reasoning about time in CBR, Atlanta, GA, October 28-November 2, 2016, pp.122-131.
12. Ontañón, S., Mishra, K., Sugandh, N., Ram, A. (2007) Case-Based Planning and Execution for Real-Time Strategy Games. in ICCBR 2007, Lecture Notes in Computer Science 4626, pp 164-178.
13. Santiago Ontañón (2012) Case Acquisition Strategies for Case-Based Reasoning in Real-Time Strategy Games. In FLAIRS 2012. AAAI Press
14. Mishra, K., Ontañón, S., Ram, A. (2008), Situation Assessment for Plan Retrieval in Real-Time Strategy Games. ECCBR-2008, Lecture Notes in Computer Science 5239, pp 355-369.
15. Wender, S., Watson, I. (2014) "Combining Case-Based Reasoning and Reinforcement Learning for Unit Navigation in Real-Time Strategy Game AI, In: Lamontagne, L. and Plaza, E. (eds) Case-Based Reasoning Research and Development: 22nd International Con-ference, ICCBR 2014, pp. 511-525 .
16. Bulitko, V., Bjornsson, Y., Lawrence, R.: Case-based subgoaling in real-time heuristic search for video game pathfinding. Journal of Artificial Intelligence Research 39, 269–300 (2010).
17. Floyd, M.W., Esfandiari, B. (2011). Building Learning by Observation Agents Using jLOAF. In Proceedings of Workshop on Case-Based Reasoning for Computer Games (held at the 19th International Conference on Case-Based Reasoning), Greenwich, England, UK, September 12-15, 37-41.
18. Acosta, E., Esfandiari, B., Floyd, M.W. (2010). Feature Selection for CBR in Imitation of RoboCup Agents: A Comparative Study. In Proceedings of the Workshop on Case-Based Reasoning for Computer Games (held at the 18th International Conference on Case-Based Reasoning), Alessandria, Italy, July 19-22, 25-34.
19. Floyd, M.W., Esfandiari, B. (2009). An Active Approach to Automatic Case Generation. In Proceedings of the 8th International Conference on Case-Based Reasoning, Seattle, Wash-ington, USA, July 20-23, 150-164. Springer.
20. Kapetanakis, S.,Petridis, M.,Knight, B.,Ma, J.,Bacon,L. : A Case Based Reasoning Approach for the Monitoring of Business Workflows, 18th International Conference on Case-Based Reasoning, ICCBR 2010, Alessandria, Italy, LNAI (2010)
21. Kapetanakis, S., Petridis, M., Knight, B., Ma, J.,Bacon, L.: Providing Explanations for the Intelligent Monitoring of Business Workflows Using Case-Based Reasoning, in workshop proceedings of ExACT-10 at ECAI 2010, Lisbon, Portugal (2010)
22. Kapetanakis, S., Petridis, M.: Evaluating a Case-Based Reasoning Architecture for the Intelligent Monitoring of Business Workflows, in Successful Case-based Reasoning Applications-2, S. Montani and L.C. Jain, Editors. 2014, Springer Berlin Heidelberg. p. 43-54.

23. Niu L., Zhuo G., An improved real algorithm for difficult path finding situation. Proceeding of the International Archives of the Photogrammetry, Remote Sensing and Spatial Infor-mation Sciences 2008 Volume 37. Beijing, China
24. Teodorovic, D., Šelmic, M., Mijatovic-Teodorovic, L. (2013). Combining case-based reasoning with Bee Colony Optimisation for dose planning in well differentiated thyroid cancer treatment. Expert Systems with Applications 40 (5), pp. 2147-2155.