Creating Intelligent Agents: Combining Agent-Based Modeling with Machine Learning

Dale Brearcliffe and Andrew Crooks

October 14, 2020

Creating Intelligent Agents: Combining Agent-Based Modeling with Machine Learning

Dale K. Brearcliffe[1][0000-0003-0142-000X] and Andrew Crooks[2][0000-0002-5034-6654]

[1] George Mason University, Fairfax VA 22030, USA
[2] University at Buffalo, Buffalo NY 14261
dbrearcl@gmu.edu & atcrooks@buffalo.edu

Abstract. Over the last two decades, with advances in computational availability and power, we have seen a rapid increase in the development and use of Machine Learning (ML) solutions applied to a wide range of applications including their use within agent-based models. However, little attention has been given to how different ML methods alter the simulation results. Within this paper, we discuss how ML methods have been utilized within agent-based models and explore how different methods affect the results. We do this by extending the Sugarscape model to include three ML methods (evolutionary computing, and two reinforcement learning algorithms (i.e., Q-Learning, and State→Action→Reward→State→Action (SARSA)). We pit these ML methods against each other and the normal functioning of the rule-based method (Rule M) in pairwise combat. Our results demonstrate ML methods can be integrated into agent-based models, that learning does not always mean better results, and that agent attributes considered important to the modeler might not be to the agent. Our paper's contribution to the field of agent-based modeling is not only to show how previous researchers have used ML but also to directly compare and contrast how different ML methods used in the same model impact the simulation outcome. Since this is rarely discussed, doing so will help bring awareness to researchers who are considering using intelligent agents to improve their models.

Keywords: Agent-Based Modeling, Evolutionary Computing, Machine Learning, Reinforcement Learning, Sugarscape.

1 Introduction

Advances in computational availability and power have permitted a rapid increase in the development and use of machine learning (ML) solutions in a wide variety of applications (Batty et al., 2012). With this growth there is growing interest utilizing ML within agent-based models as it has the potential to move from simple agents to complex ones (see Crooks et al., 2020; Rand, 2006). Broadly speaking, agent-based modeling can be considered to have three major steps: 1) design of the model, 2) execution of the model, and 3) evaluation of the model. Machine learning techniques have been applied to all three of these phases (see Abdulkareem et al., 2019). For example, Kavak et al. (2018) used ML to derive parameter values for an agent-based model such as human mobility. Others have used ML during the running of the model, allowing agents to learn from past experiences and make more informed decisions via reinforcement learning (Ramchandani et al., 2017). With respect to using ML algorithms to analyze model outputs, Heppenstall et al. (2007) used a genetic algorithm to validate outcomes of an agent-based model which simulates the retail petrol market.

In the ML community at large, it is common to compare different approaches and take the one that gives the best result (e.g. Yuan et al., 2019), but this is not the case within the social simulation community. While there is growing interest in ML within the agent-based modeling community, little has been written with respect to why one ML method was chosen over another, or how the simulation results might be different if different ML methods are used. The purpose of this paper is therefore to demonstrate the integration of three machine learning methods into a well-known agent based model (i.e., the Sugarscape model (Epstein and Axtell, 1996)) and to demonstrate how different methods alter the outcomes of a simulation. In the remainder of this paper, we first provide a brief overview of Machine Learning (ML) and its utilization within agent-based models (Section 2) before presenting our methodology in Section 3. Section 4 shows the results of different ML methods and Section 5 concludes the paper with some thoughts on future directions on using machine learning with agent-based models.

2 Background

Machine learning, a subfield of artificial intelligence, is a large subject area and to cover all types of methods is beyond the scope of this paper (interested readers are referred to Russell and Norvig, 2016). Therefore, this paper focuses on two major types of ML: Evolutionary Computing (EC), and Reinforcement Learning (RL). Scope is further limited to agent-based models that support social science research and enhance the capability of individual agents during the execution of the model. In contrast, ML used specifically to optimize model parameters (e.g., Junges and Klügl, 2011; Ma et al., 2014) or explore model outputs (e.g., Heppenstall et al., 2007) are not considered within this paper since these processes enhance the modeler rather than the model.

Our rationale for focusing on learning within the agents echoes that of Samuel (1959) who coined the phrase "*Machine Learning*" for a computer program capable of improving itself "*to play a better game of checkers than can be played by the person who wrote the program*." The ability to exceed its original program is a key facet of ML. Feedback to decisions made by the program, based on the program's observations of its environment, allows it to adjust its internal representations to make better future decisions (Russell and Norvig, 2016). Thus, the computer improves its ability to accomplish a task or achieve a goal.

One key element of agent-based models is that of the agent's ability to learn from past decisions and improve their decision making. Many agent-based models utilize simple rules (e.g. Crooks, 2010; Schelling, 1971) that are capable of creating complex, system level results. The simple rules, however, do not always provide the desired results. For example, Macy and Flache (2002) stated that the focus of modeling cooperation might need to move away from population level dynamics to, "*cognitive dynamics at the level of the individual.*" Also Devezer et al. (2019) noted that a limitation of their agent-based model was the inability of their simple agents to learn and remember previous decisions. To correct this, agents may need a capability to create dynamic rules that adapt to individual experiences: a memory of previous decisions and their consequences. Without experience to guide them, agents may be reduced to random decisions or encounters. People largely do not conduct themselves randomly (Kennedy, 2012), so incorporating experiences may lead to more accurate social science models.

In order for agents to learn we can turn to two of the most widely used ML methods, that of EC and RL. With respect to EC, probably the most well known subclass is that of Genetic Algorithms (GAs). GAs were first introduced by Holland (1975) and provide a means for the future agent behaviors to change when successful agents share their knowledge, represented as a feature set (genes), with their offspring. Offspring agents are created by selecting successful parents, combining portions of the parent's features, and then randomly changing a small percentage of the offspring's features (Reproduction, Crossover, and Mutation). Over time the agent Collective moves toward an optimal solution. Holland's (1975) original structures have been adapted from a variety of fields as shown in Table 1. Another subclass of EC is Genetic Programming (GP) which was introduced by Koza (1992) and is similar to GA, the focus, however, is on code segments that are available for reproduction, crossover, and mutation rather than agent features (i.e., attributes). Holland (1975) and Koza's (1992) use of the word "genetic" to describe their ideas show the link to biological evolution. Social scientists first questioned whether it was valid to employ biological evolution concepts to social processes (see Chattoe-Brown, 1998). This initial skepticism has been overcome and EC has been used within agent-based models to study a variety of social problems ranging from economics, social dynamics to that of teamwork (see Appendix A and Table A1 and Revay and Cioffi-Revilla (2018) for further details).

Table 1: A historical look at the fields of study utilizing adaptive structures (operators + structure) and performance measures which inspired the beginning of GA study (adapted from Holland, 1975).

Field	Structure	Operators (Processes)	Performance Measure
Genetics	Chromosomes	Mutation, recombination, etc.	Fitness
Economic planning	Mixes of goods	Production activities	Utility
Control	Policies	Bayes's rule, successive approximation, etc.	Error functions
Physiological psychology	Cell assemblies	Synapse modification	Performance rate
Game theory	Strategies	Rules for iterative approximation of optimal strategy	Payoff
Artificial Intelligence	Programs	"Learning" rates	Comparative efficiency

As noted above, another widely used approach in ML is RL, which originated from behavioral psychology and neuroscience (Sutton and Barto, 2018). RL provides a framework for agents to learn by sequential interaction with their environment and other agents in much the same way humans learn. These agents develop decision-making sequences that maximize a reward for a future goal. The frameworks do not require any prior knowledge, although such knowledge is sometimes included (e.g., the transfer of agent knowledge between models (Takadama et al., 2008)), and agents can learn by playing against themselves (e.g. a Markov game with teams of competing agents (Nowé et al., 2006)). Agents decide to move from a current state to a new state using a type of learned Markov decision process (Sutton and Barto, 2018). Similar to ML, RL is not a single methodology, but encompasses a number of approaches. Table 2 provides examples of RL types that we have found to be used in agent-based modeling (see Appendix B for more details), along with abbreviations used in this paper. This is not, however, a full list of RL methods, and such a list and detailed explanation of these are beyond the scope of this paper. We refer readers to Sutton and Barto (2018) for further examples. As with EC, RL has gained interest in the agent-based modeling community to explore a wide variety of applications from economics to the social dynamics of societies (examples of which are presented Table A2).

Table 2: Examples of different reinforcement learning types.

Reinforcement Learning Type	**Description**	**Reference**
Bush–Mosteller (BM)	A type of statistical learning where a predictive function is derived from data.	Bush & Mosteller (1955)
Learning Automata (LA)	Simple algorithm operating in a stochastic environment where agents can improve their actions during operation.	Narendra & Thathachar (1974)
Q-Learning (QL)	Learns a policy, expressed as a matrix of values for states and actions, thus telling an agent what to do in different circumstances. It does not require a model of the problem to be solved. (State→Action→Reward)	Watkins (1989)
State→Action→Reward →State→Action (SARSA)	Extends Q-Learning by also considering the future selected state-action. Uses a model it builds.	Rummery & Niranjan (1994)
Temporal-Difference (TD)	Learns from experience without an environmental model and updates estimates before final outcome is known.	Sutton (1988)

3 Methodology

In order to demonstrate the differences between rule-based and learning agents, and show how learning agents can be integrated into an agent-based model, we adapted the "*Sugarscape 2 Constant Growback*" (Li and Wilensky, 2009) model, which is included in the NetLogo (Wilensky, 1999) models library. Our rationale for choosing this model was that it is well known within the social sciences, and the purpose of this paper was not to solve or explore social issues, but to test the usability of EC and RL within an agent-based model. Using a well-known model seemed the logical choice. Sugarscape (Epstein & Axtell, 1996) has been used to demonstrate migration, trade, wealth inequality, disease processes, sex, culture, and conflict. It is on conflict, in the form of combat that we focus on within this paper as will be shown with the results in Section 4. Within the Sugarscape model there are two groups of agents (i.e., red and blue), where individual agents are allowed to move to the nearest unoccupied location with the most sugar and consume sugar at the location, metabolize sugar, and die if it runs out of sugar. Our model adds combat with agents that can attack and retreat. The basic logic of the model is presented in Figure 1 while Table 3 specifies the differences between the "original" Sugarscape model and the one presented in this paper. Much of the model is similar to the original model but with the additions to support EC and RL that are highlighted in Table 4. Due to word limitations, readers are referred to supplementary material provided at https://tinyurl.com/ML-Agents. At this link the model presented in this paper along with a full description of it following the Overview, Design concepts, and Details (ODD, Grimm et al., 2020) protocol can be found. We do this to allow others to replicate the results and adapt the ML methods for their own applications if they so desire.

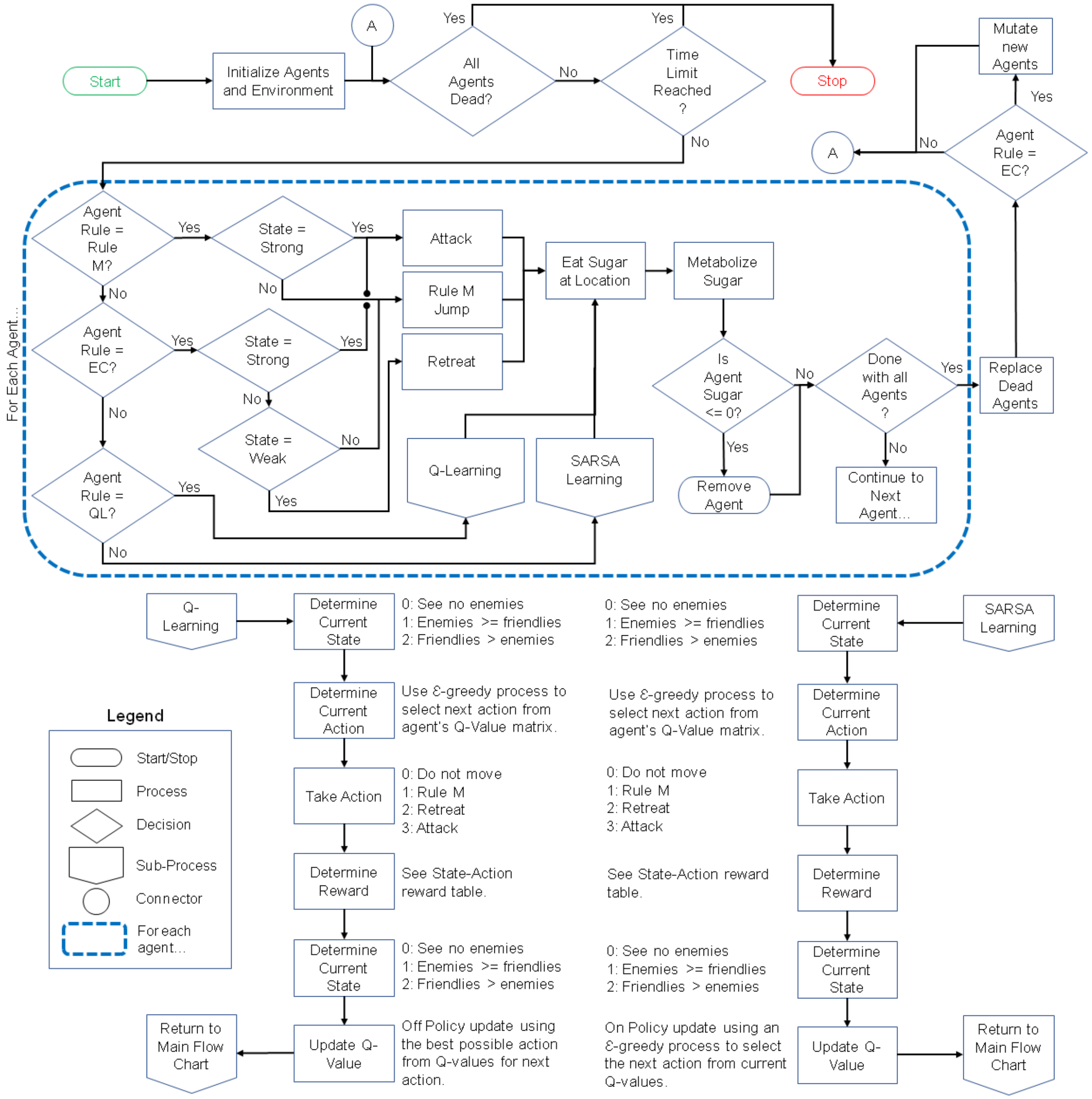

Figure 1: Model execution flowchart.

As the focus of this paper is to demonstrate, compare, and contrast different ML methods for agent learning, we now provide a brief description of the ML utilized within this paper, along with the original model's rule-based method, Rule M. We have created agent learning types (i.e., Q-Learning, SARSA, and EC) and the actions that can be taken are shown in Figure 1. Figure 1 also shows how these sub-models are positioned in the process flow and below we provide a brief discussion of each one (further details can be found in the online supplementary material):

- **Rule M** type agents use explicit knowledge in the form of a priori rules imposed by the modeler to determine actions (i.e., as in the original model) given its state. These agents do not learn.
- **EC type** agents use explicit knowledge in the form of a priori rules imposed by the modeler to determine actions given its state. The EC agents learn as a Collective (i.e., red or blue groups) using evolutionary computing (Holland, 1975). As EC agents die and are replaced, the new agent's metabolism and vision

attribute values are initialized based on those agents in their Collective that have the most wealth rather than a random value. The mean vision and metabolism attributes for the Collective converge to a local maximum that represents a best choice for wealth accumulation.

- **Q-Learning and SARSA** type agents uses reinforcement learning (Sutton, 1988; Sutton and Barto, 2018) to gather tacit knowledge over time and store their experience in a Q-Value (Quality Value) matrix. Initially, these agents explore their world randomly while updating their experience. As they age, they increasingly use their experience to make decisions until doing so 95% of the time for the remainder of their life as shown in Figure 2. In a sense, as the agents get older, they explore less and follow their tacit knowledge more. The difference between these two agent types is how they update their experience. Q-Learning agents make off-policy updates that allow them to follow their existing policy in their Q-Value matrix, and then update their experience by looking outside the policy, focusing on the best available reward. The SARSA agents make on-policy updates following and updating their experience using their existing policy. For a full description, see Sutton and Barto (2018).

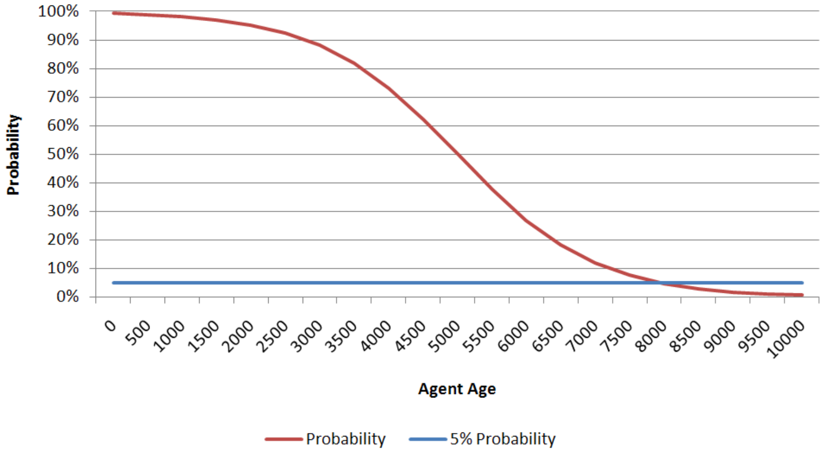

Figure 2: Age based exploration probability for reinforcement learning agents. The exploration probability ranges from 100% to an artificial lower bound of 5% chosen to ensure agents continue to explore during their entire life.

Table 3: Differences between the "original" Sugarscape and the ML models developed for this paper.

Attribute	**Original Base Model**	**ML Model**	**Attribute Type**
Collectives (Breeds)	No	Group A and Group B	Fixed
Discount Rate (Gamma)	No	0.9	Hyper Parameter
Epsilon-Greedy	No	$\epsilon = \frac{1}{1 + e^{\frac{t-5000}{1000}}}$	Fixed
Evolutionary Computing	No	Yes	Rule
Geographic Sugar Distribution	Fixed	Fixed	Fixed
Grow Back Rate (Sugar)	1	[0.1,1.0], Increment of 0.1	Parameter
Initial Population	[10,1000], Increment of 10	[1,400], Increment of 1	Parameter
Initial Sugar per Agent	[5,25], Increment of 1	[5,25], Increment of 1	Parameter
Learning Rate (Lambda)	No	0.8	Hyper Parameter
Metabolism	[1,4], Increment of 1	[1,4], Increment of 1	Parameter
Q-Learning	No	Yes	Rule
Reward	No	Yes	Fixed
Rule M	Yes	Yes	Rule
Rule R	No	Yes	Rule
SARSA Learning	No	Yes	Rule
Time Limit	No	20,001	Fixed
Vision	[1,6], Increment of 1	[1,6], Increment of 1	Parameter
Vision Neighborhood	von Neumann	von Neumann	Fixed

Table 4: List of model attributes and their descriptions.

Attribute	Description
Collectives (Breeds)	Agents are divided into two groups that can compete with each other.
Discount Rate (Gamma)	In reinforcement learning this value diminishes the impact of future rewards.
Epsilon-Greedy	In reinforcement learning, this changes the probability of exploration versus exploitation of time.
Evolutionary Computing	A rule that creates a new agent via crossover of the Vision and Metabolism attributes between the two agents with the most sugar.
Geographic Sugar Distribution	Provided as a text file that is read into the simulation. Shows the maximum and starting amount of sugar on each grid of the spatial map.
Grow Back Rate (Sugar)	The amount of sugar that grows back, up to the maximum allowed, on each grid, each tick.
Initial Population	The starting number of agents for each breed.
Initial Sugar per Agent	The amount of sugar each agent starts with selected from a uniform distribution.
Learning Rate (Lambda)	The smaller the number, the slower the learning rate.
Metabolism	The amount of sugar each agent consumes each time period.
Q-Learning	A rule that permits off-policy learning.
Reward	In reinforcement learning, what an agent receives for taking an action while in some state.
Rule M	Move to the nearest unoccupied location with the most sugar.
Rule R	Provides a replacement agent for each breed when an agent has died.
SARSA Learning	A rule that permits on-policy learning.
Ticks	Discrete temporal intervals of equal value that increase monotonically during model execution. Each Tick has no equivalency to wall clock time.
Time Limit	The number of ticks at which the simulation will halt.
Vision	The distance, in grids, an agent can see.
Vision Neighborhood	von Neumann - Can see along the four cardinal directions.

4 Results

Before presenting the results for the model, we need to mention our efforts for verification. Here we refer to verification as the process of checking that the model matches its design. This included code walkthroughs and parameter sensitivity testing. These tests ensured that we made no logical errors in the translation of the model into code, and that there were no programming errors. For example, during testing we noticed that the original model used a non-symmetric sugar topology that created noticeable differences in some outcomes when testing the model, specifically when changing the rule sets between two groups. To reduce this bias, a modified sugar topology was created which was symmetrical across the major axis running from the top left to bottom right. In addition, the torus feature was removed eliminating the ability of the agents to move off the edge of the map and appear on the opposing side. This modified sugar topology accompanies the model and is available at https://tinyurl.com/ML-Agents.

Now turning to ascertainment of how the different four rule methodologies (i.e., Rule M, Evolutionary Computing, Q-Learning, and SARSA) interplay with each other, Table 5 shows the initial settings used for testing while Figures 3 to 9 show pairwise comparisons. Readers should note that the Y-axes for each data graphic in the figures use a base 10 log scale while the X-axes represents 20,001 time intervals (0-20,000 ticks) and the results presented are the mean of 50 model runs.

Table 5. Initial settings for all model runs.

Attribute	Initial Value	Remark
Corner Start	On	Breeds will start in bottom left and top right corners.
Combat	On	Agents will attack.
Initial Population	400	Maximum agents for corner start.
Replace Dead	On	Generate replacements for dead agents. If Rule EC is used, replacements will have their Vision and Metabolism created from top two, high sugar agents for their breed.
Sugar Growback Rate	1.0	Note: A setting of 0.6 or less "stresses" the agents with low resources and encourages them to leave their corners.

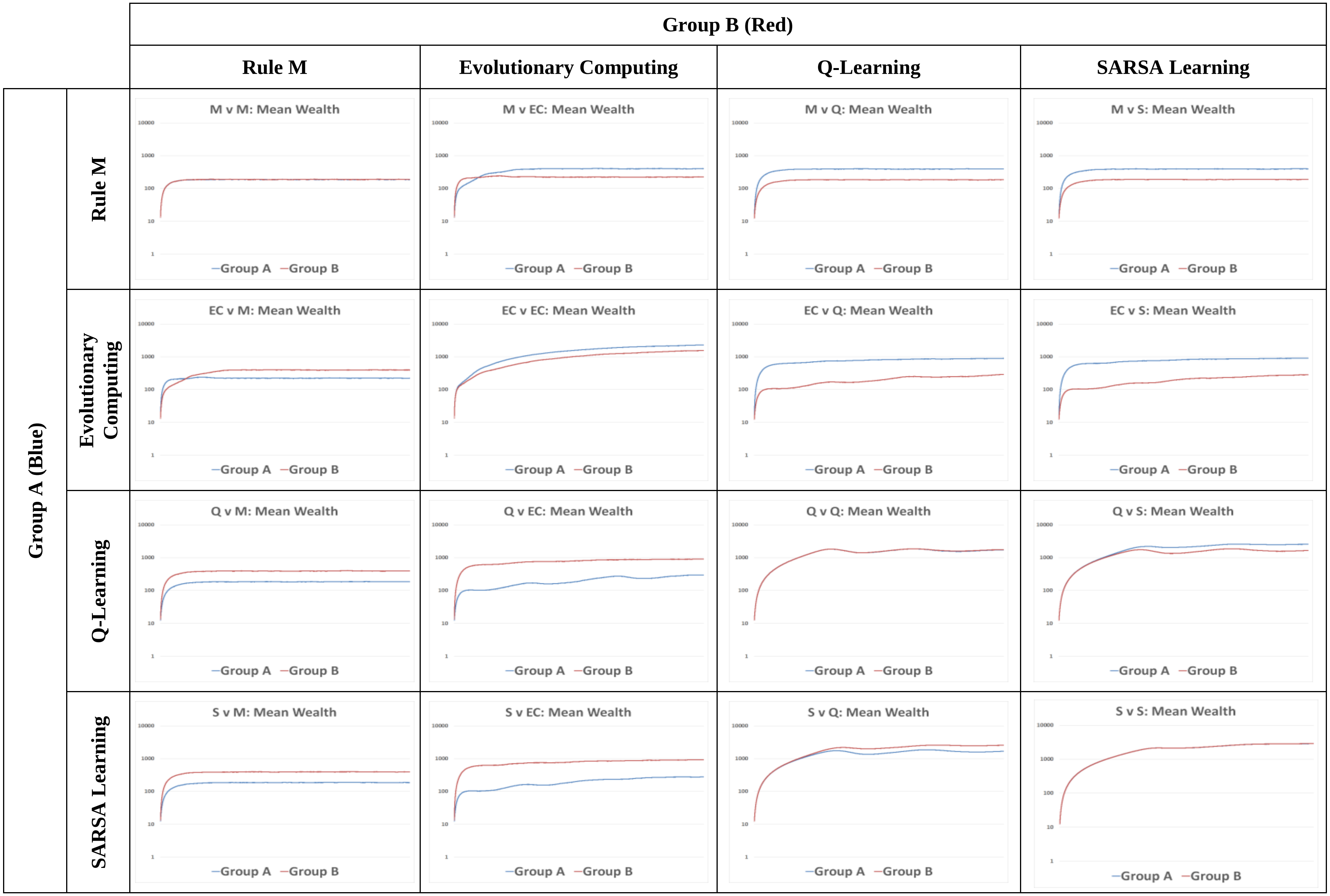

Figure 3: Mean result for wealth for all rule combinations (50 model runs).

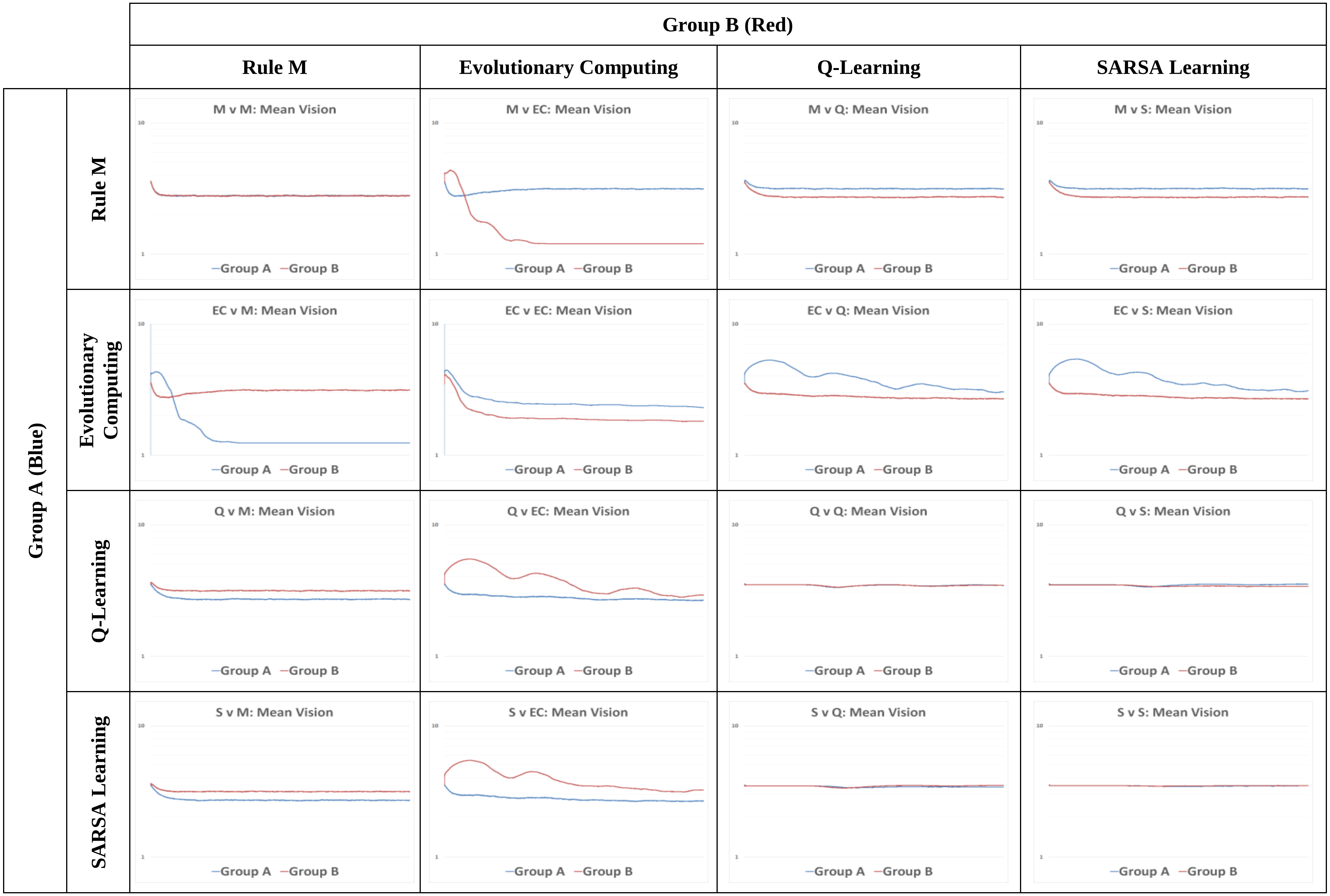

Figure 4: Mean result for vision for all rule combinations (50 model runs).

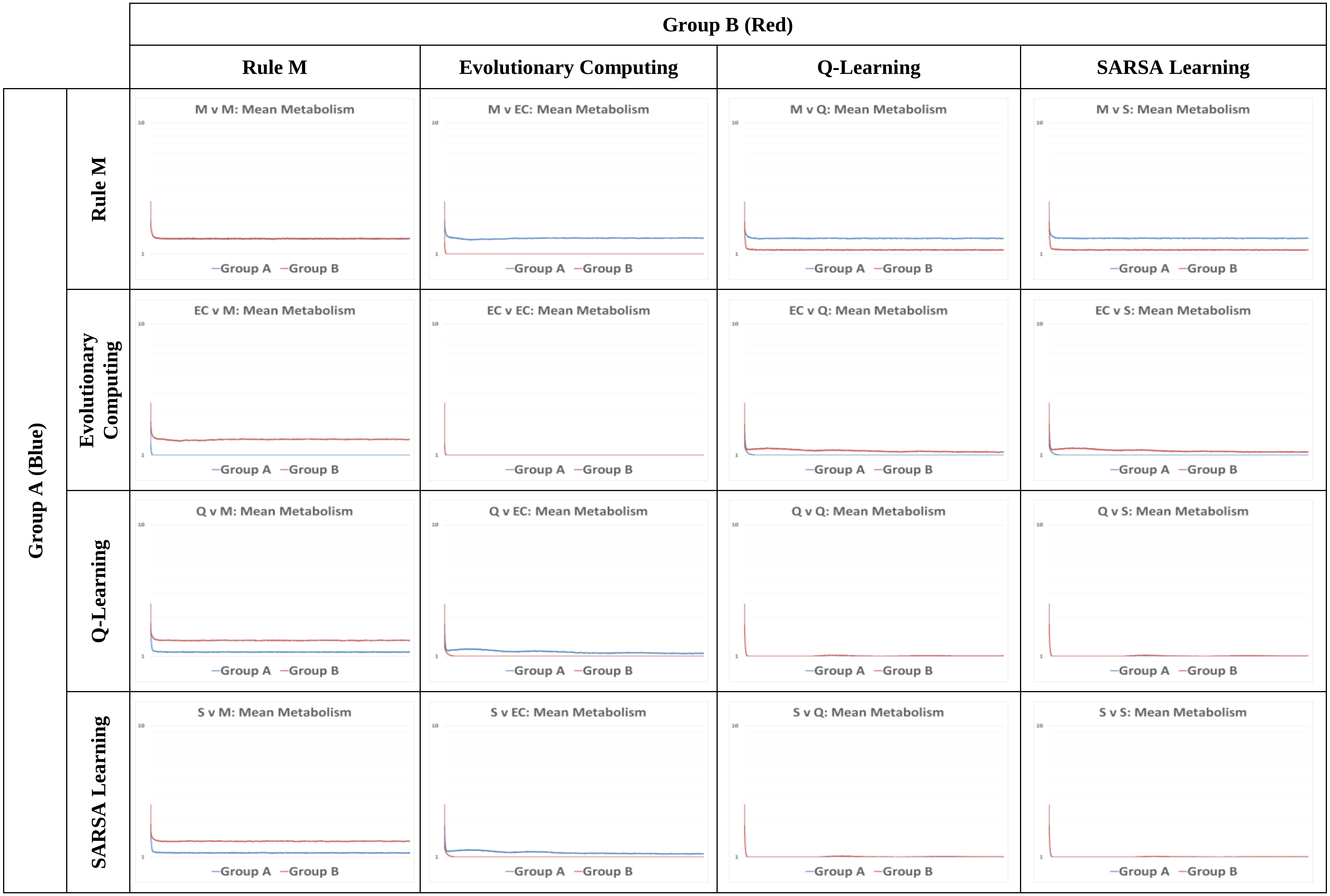

Figure 5: Mean result for metabolism for all rule combinations (50 model runs).

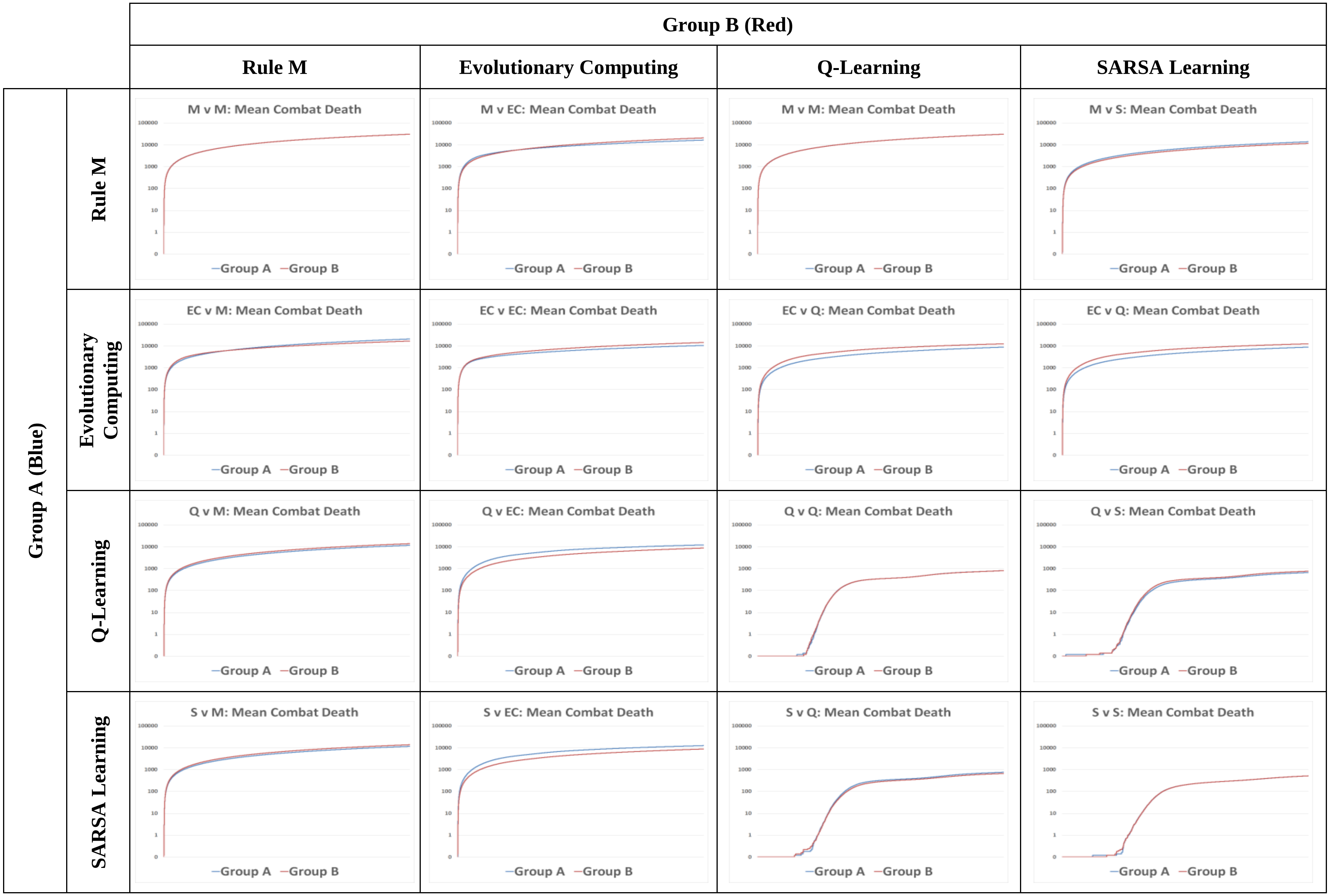

Figure 6: Mean result for cumulative combat deaths for all rule combinations (50 model runs).

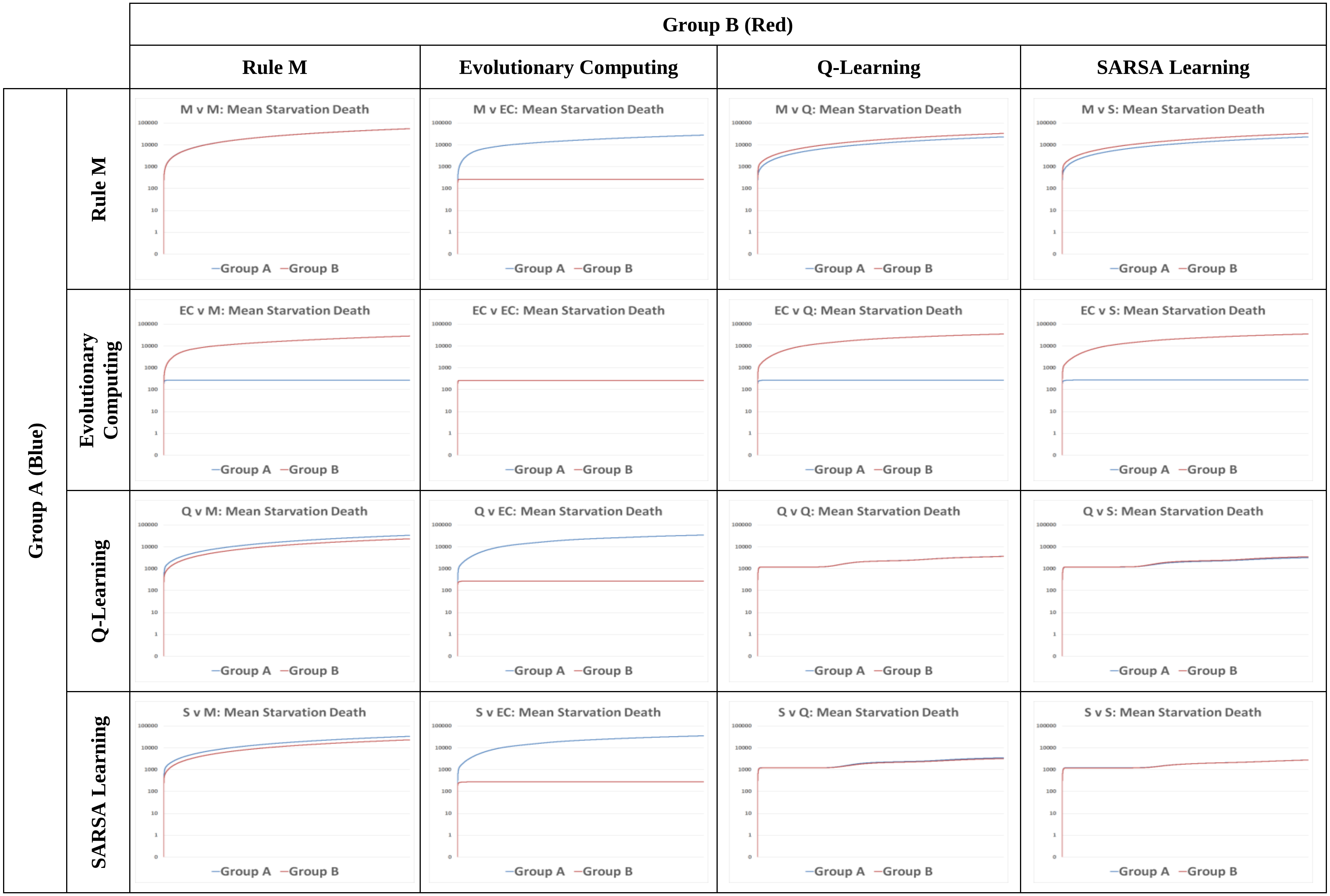

Fig 7. Mean result for cumulative starvation deaths for all rule combinations (50 model runs).

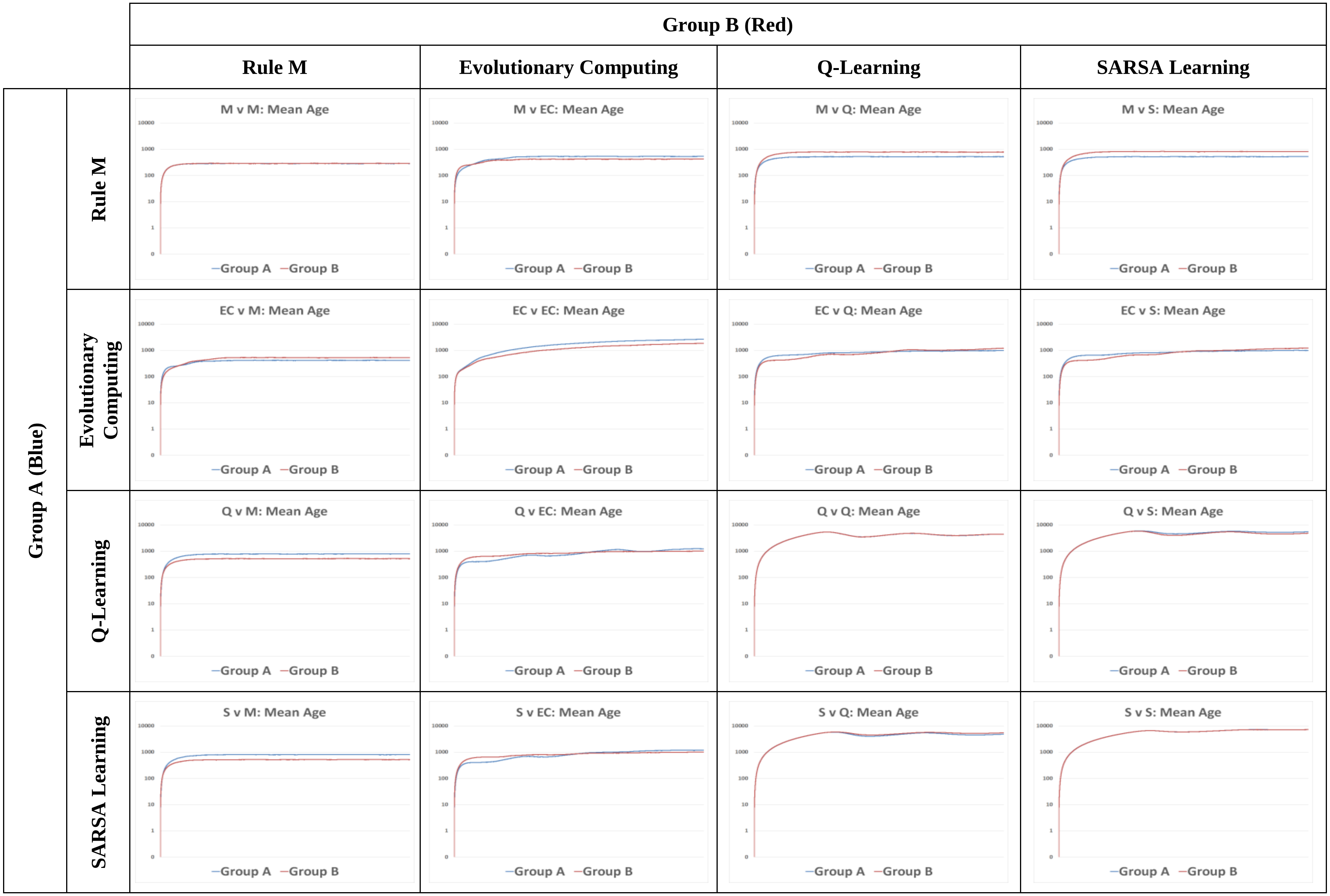

Figure 8: Mean result for age for all rule combinations (50 model runs).

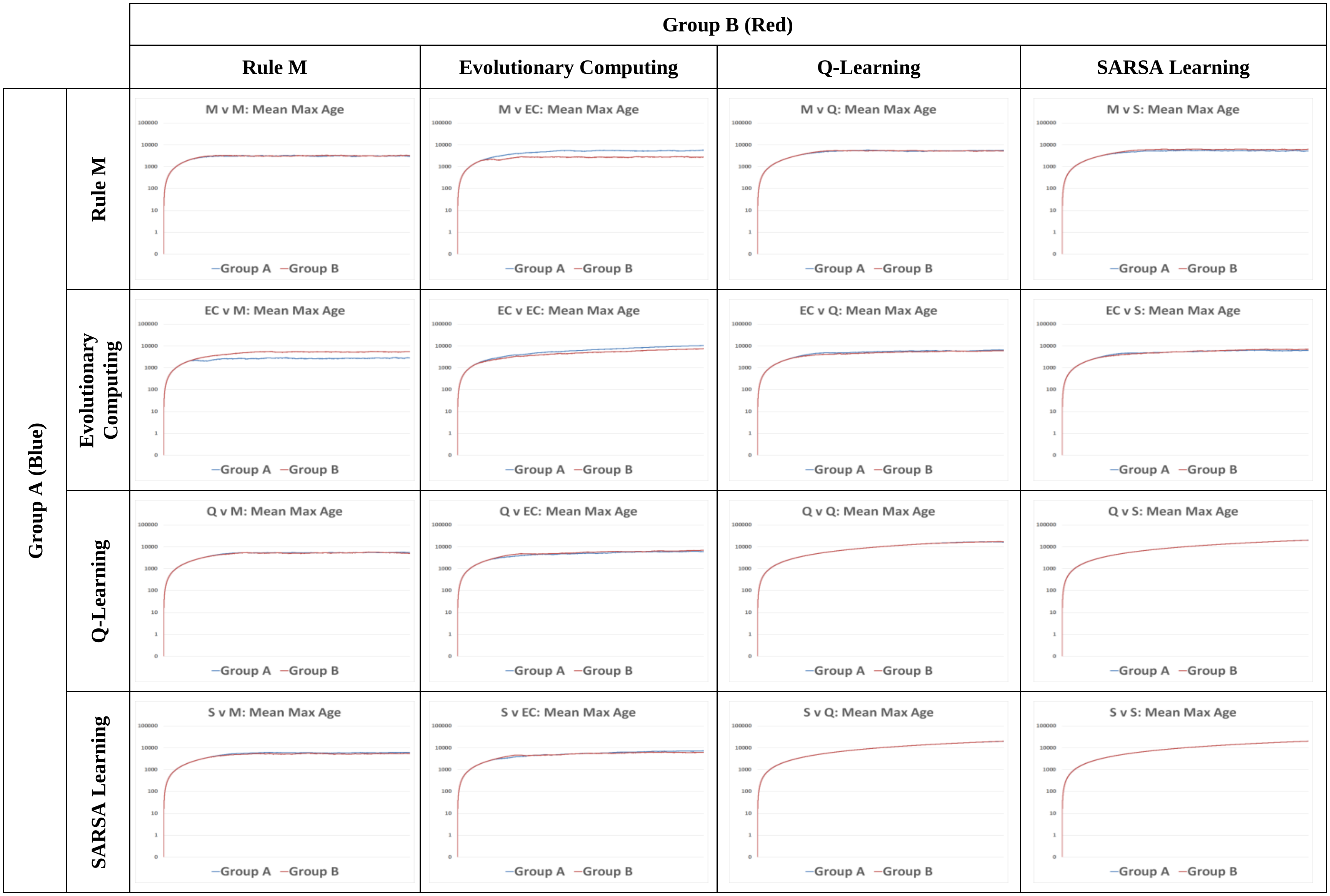

Figure 9: Mean result for maximum age for all rule combinations (50 model runs).

As can be seen in Figures 3 to 9 there is a general similarity across the main diagonal of each of the data graphics (i.e., when comparing the same rule methods) and this is to be expected, as there is no difference between the two competing groups other than their name (i.e., A and B). However, EC versus EC was the one exception with Group A (Blue) always having the same or slightly better outcomes than Group B (Red). We investigated this and we found it was not due to activation order or incorrect code. Within Sugarscape, there are two main functions of the agents within the model. The first is to collect sugar and this is reflected in the mean wealth that is shown in Figure 3. Agents utilizing Rule M, the original methodology, always accumulate more sugar against one of the three learning methodologies. Rule M also maintained a higher vision distance (as shown in Figure 4) and a slightly higher metabolism (Figure 5). EC learning maintained a significant advantage in wealth (sugar) accumulation over the two RL methods (Figure 3) and does so with a near equal metabolism (Figure 5). It does this with a much higher vision distance that declines over time (Figure 4). A longer run time might have ultimately eliminated this advantage, as the end state is only known to 20,001 ticks. Pitting the two RL methods against each other shows a small advantage of Q-Learning's ability to go outside its own policy (i.e., off policy) as the two groups shifted from exploration to policy as can be seen in Figure 3. Both maintained almost identical high vision distances (Figure 4) and low metabolisms (Figure 5). Compared to the other methods, the agents utilizing the SARSA method do not accumulate so much wealth (Figure 3).

The second main function of this Sugarscape model is conflict in the form of combat and here there are no clear winners. The EC method accumulated fewer combat deaths when opposing the two RL methods yet had almost equal combat deaths when opposing the Rule M group (Figure 6). This is possibly attributable to the EC group maintaining a higher vision distance (Figure 4) against the RL group, enabling better retreat and attack decisions. Rule M, however, was able to maintain a very high vision distance relative to EC (Figure 4). The two RL methodologies fighting against each other showed a noticeable initial period with very few combat deaths (Figure 6).

It should also be noted that in the current version of the model, death causes all sugar and combat experience to be lost by the dying agent. The EC method was noticeably superior to the other three methodologies in finding sufficient sugar to metabolize and not starve (Figure 7). In addition, mean age (Figure 8) and mean maximum age (Figure 9) are indicators of experience and sugar gathering opportunities. As RL agents age their individual Q-Value tables reflect their accumulated combat experience and EC agents holding the most sugar become replacement agent templates. All methodologies demonstrated a decrease in mean metabolism (Figure 5), a positive outcome given a high value could lead to an agent starving to death. Agents utilizing the EC method routinely saw their mean vision decrease over time (Figure 4). This was an unexpected evolution as superior vision can lead to better sugar mining, defensive actions (retreat in the face of superior numbers), and offensive actions (attacks against an outnumbered opponent). It is possible that combat deaths left mostly low sugar holders and they were low because they lacked vision, however, further analysis is needed to explain this phenomenon. While not reported in the figures, we would like to note that different ML methodologies resulted in variations in model run times. This difference was expected as ML methods can be computationally intensive. For example, using Rule M versus Rule M simulation as a baseline, the SARSA versus SARSA simulation could take as much as seven times as long to complete a single run.

5 Summary

Both agent-based modeling and ML have seen growth over the last two decades, along with the integration of ML within agent-based models (as discussed in Section 2). Seldom have agent-based modelers applied different ML methods to the same model and then reported how simulation outcomes differ. This paper does just that. By adapting the Sugarscape model with three types of learning agents (i.e. EC, Q-Learning and SARSA) and contrasting them with Rule M from the original model we found that different learning methods do have an impact on the simulation outcome (as discussed in Section 4). While our aim was not to prove which learning method was superior, but rather to provide a way to compare and contrast them, it was surprising that Rule M agents were able to accumulate more wealth than the ML agents. One possible reason for this is that with the ML agents, we did not specifically attempt to tune the simulation. For example, we did not experiment with altering the hyper parameters of the RL models (see Table 3). With that being said, our results do suggest that modelers who wish to use ML methods for agent learning should pay particular attention and justify why they chose one method over another.

The question our results bring up, is it worth the effort to add ML to agent-based models? Creating a model using agents has always required a firm grasp of the simulation's goal, but ML requires additional thought. In traditional rule-based models (e.g. Rule M), the agents' actions are designed specifically to match situations the modeler anticipates. For example, if the agent sees sugar, move to the location with the sugar and consume it. While with ML this is different, taking RL as an example, the modeler needs to consider a range of actions an agent may be tasked to perform and the states they may be in because of these actions (as discussed in Section 2). Conversely, the state the agent is in can be used to determine what actions could take place. Rewards can be used to tempt the agent mathematically for each state-action combination. Positive rewards create an inclination to repeat the action, while negative rewards should be like touching a hot stove for the first time. Taking the Sugarscape model presented in this paper, combat was an important part of this demonstration and agents are dependent on the presence of their enemy (i.e., the other group) for learning. RL agents that do not encounter the enemy, never learn to attack or retreat. EC agents evolve based on representative agents with the most accumulated sugar. High sugar EC agents that are killed by the enemy cannot pass their attributes onto new agents. For both types of learning agents, enemy actions have consequences beyond a single combat action. However, going back to the question posed at the beginning of this paragraph, we would argue that the answer is yes, it is worth adding ML to agent-based models, if the goal is unachievable by simpler methods or if you want the agents to learn from good and bad experiences. Another consideration when it comes to using ML is the increase in computational resources needed for them. Today's era of big data and cloud-based computing has increased the availability of such resources, but the cost is still a limitation and wasted time cannot be recovered. For example, our RL agents needed time to explore actions before beginning to exploit them and SASRA took seven times longer (wall clock time) to execute than Rule M over the same number of time steps (ticks).

While the methods used within this paper are well established within the ML field, it paves the way to exploring newer ML methodologies in the social science community. Looking ahead, using deep RL and combining ML methodologies in the same agent is worthy of future research. It is not surprising that others (Sutton and Barto, 2018) have combined RL and an artificial neural network (ANN). The ANN can act as a function to reduce a large number of inputs to a smaller set that focuses on critical features. In a different approach, EC could be used as a means for two successful RL agents to pass their knowledge to a new agent. Another area of further work would be to explore other well know agent-based models and explore the impact that ML would have on their end states. With all this being said, our paper's contribution to the field of agent-based modeling is not only to show how previous researchers have used ML but also to directly compare and contrast how different ML methods used in the same model impact the simulation outcome which is rarely discussed, thus helps bring awareness to researchers who are considering using intelligent agents to improve their models.

References

Abdulkareem, S.H., Mustafa, Y.T., Augustijn, E.-W. and Filatova, T. (2019), 'Bayesian Networks for Spatial Learning: A Workflow on Using Limited Survey Data for Intelligent Learning in Spatial Agent-based Models', *Geoinformatica*, 23(2): 243-268.

Arifovic, J. (1994), 'Genetic Algorithm Learning and the Cobweb Model', *Journal of Economic Dynamics and Control*, 18(1): 3-28.

Batty, M., Axhausen, K.W., Giannotti, F., Pozdnoukhov, A., Bazzani, A., Wachowicz, M., Ouzounis, G. and Portugali, Y. (2012), 'Smart Cities of the Future', *The European Physical Journal Special Topics*, 214(1): 481-518.

Bush, R.R. and Mosteller, F. (1955), *Stochastic Models for Learning*, John Wiley & Sons, Oxford, UK.

Chattoe-Brown, E. (1998), 'Just How (Un)realistic Are Evolutionary Algorithms As Representations of Social Processes?' *Journal of Artificial Societies and Social Simulation*, 1(3): 2, Available at http://jasss.soc.surrey.ac.uk/1/3/2.html.

Chen, S.-H. and Yeh, C.-H. (1996), 'Genetic Programming Learning and the Cobweb Model', in Angeline, P. and Kinnear, K.E. (eds.), *Advances in Genetic Programming 2*, MIT Press, Cambridge, MA, pp. 443-466.

Chmura, T. and Pitz, T. (2007), 'An Extended Reinforcement Algorithm for Estimation of Human Behaviour in Experimental Congestion Games', *Journal of Artificial Societies and Social Simulation*, 10(2): 1, Available at http://jasss.soc.surrey.ac.uk/10/2/1.html.

Christensen, K. and Sasaki, Y. (2008), 'Agent-Based Emergency Evacuation Simulation with Individuals with Disabilities in the Population', *Journal of Artificial Societies and Social Simulation*, 11(3): 9, Available at http://jasss.soc.surrey.ac.uk/11/3/9.html.

Claus, C. and Boutilier, C. (1998), 'The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems', in Buchanan, B.G. (ed.), *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, AAAI, Madison, WI, pp. 746-752.

Clempner, J.B., 20(2). (2017), 'A Game Theory Model for Manipulation Based on Machiavellianism: Moral and Ethical Behavior', *Journal of Artificial Societies and Social Simulation,* 20(2): 12, Available at http://jasss.soc.surrey.ac.uk/20/2/12.html.

Crooks, A.T. (2010), 'Constructing and Implementing an Agent-Based Model of Residential Segregation through Vector GIS', *International Journal of GIS*, 24(5): 661-675.

Crooks, A.T., Heppenstall, A., Malleson, N. and Manley, E. (2020), 'Agent-Based Modeling and the City: A Gallery of Applications', in Shi, W., Goodchild, M., Batty, M. and Kwan, M.-P. (eds.), *Urban Informatics*, Springer, New York, NY.

Curran, D. and O'Riordan, C. (2007), 'Cultural Learning in a Dynamic Environment: An Analysis of Both Fitness and Diversity in Populations of Neural Network Agents', *Journal of Artificial Societies and Social Simulation*, 10(4): 3, Available at http://jasss.soc.surrey.ac.uk/10/4/3.html.

Devezer, B., Nardin, L.G., Baumgaertner, B. and Buzbas, E.O. (2019), 'Scientific Discovery in a Model-Centric Framework: Reproducibility, Innovation, and Epistemic Diversity', *PLoS ONE*, 14(5): e0216125.

Dionne, S.D., Sayama, H. and Yammarino, F.J. (2019), 'Diversity and Social Network Structure in Collective Decision Making: Evolutionary Perspectives with Agent-Based Simulations', *Complexity,* 2019: 7591072.

Edmonds, B. (1999), 'Gossip, Sexual Recombination and the El Farol Bar: Modelling the Emergence of Heterogeneity', *Journal of Artificial Societies and Social Simulation,* 2(3): 2, Available at http://jasss.soc.surrey.ac.uk/2/3/2.html.

Epstein, J.M. and Axtell, R. (1996), *Growing Artificial Societies: Social Science from the Bottom Up,* MIT Press, Cambridge, MA.

Fischer, I. (2003), 'Evolutionary Development and Learning: Two Facets of Strategy Generation', *Journal of Artificial Societies and Social Simulation,* 6(1): 7, Available at http://jasss.soc.surrey.ac.uk/6/1/7.html.

Francisco, T. and Jorge dos Reis, G.M. (2008), 'Evolving Predator and Prey Behaviours with Co-evolution Using Genetic Programming and Decision Trees', in Keijzer, M. (ed.), *Proceedings of the 10th Annual Conference Companion on Genetic and Evolutionary Computation*, ACM, Atlanta GA, pp. 1893-1900.

Grefenstette, J.J. (1992), 'The Evolution of Strategies for Multiagent Environments', *Adaptive Behavior*, 1(1): 65-90.

Grimm, V., Railsback, S.F., Vincenot, C.E., Berger, U., Gallagher, C., DeAngelis, D.L., Edmonds, B., Ge, J., Giske, J., Groeneveld, J., Johnston, A.S.A., Milles, A., Nabe-Nielsen, J., Polhill, J.G., Radchuk, V., Rohw, M.-S., Stillman, R.A., Thiele, J.C. and Ayll, D. (2020), 'The ODD Protocol for Describing Agent-Based and Other Simulation Models: A Second Update to Improve Clarity, Replication, and Structural Realism ', *Journal of Artificial Societies & Social Simulation,* 23(2): 7, Available at http://jasss.soc.surrey.ac.uk/23/2/7.html.

Hao, J. and Leung, H.-F. (2013), 'The Dynamics of Reinforcement Social Learning in Cooperative Multiagent Systems', in Rossi, F. (ed.), *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, AAAI Press, Beijing, China, pp. 184-190.

Haynes, T. and Sen, S. (1996), 'Evolving Behavioral Strategies in Predators and Prey', in Weiß, G. and Sen, S. (eds.), *International Joint Conference on Artificial Intelligence*, Springer, Montréal, Canada, pp. 113-126.

Heppenstall, A.J., Evans, A.J. and Birkin, M.H. (2007), 'Genetic Algorithm Optimisation of a Multi-Agent System for Simulating a Retail Market', *Environment and Planning B,* 34(6): 1051-1070.

Holland, J.H. (1975), *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, MI.

Hsu, W.H. and Gustafson, S.M. (2002), 'Genetic Programming and Multi-Agent Layered Learning by Reinforcements', in Langdon, W.B., Cantú-Paz, E., K., M., Roy, R. and Davis, D. (eds.), *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, ACM, New York, NY, pp. 764-771.

Izquierdo, S.S., Izquierdo, L.R. and Gotts, N.M. (2008), 'Reinforcement Learning Dynamics in Social Dilemmas', *Journal of Artificial Societies and Social Simulation,* 11(2): 1, Available at http://jasss.soc.surrey.ac.uk/11/2/1.html.

Jang, J., Ju, X., Ryu, U. and Om, H. (2019), 'Coevolutionary Characteristics of Knowledge Diffusion and Knowledge Network Structures: A GA-ABM Model', *Journal of Artificial Societies and Social Simulation,* 22(3): 3, Available at http://jasss.soc.surrey.ac.uk/22/3/3.html.

Jim, K. and Giles, C.L. (2000), 'Talking Helps: Evolving Communicating Agents for the Predator-Prey Pursuit Problem', *Artificial Life*, 6(3): 237-254.

Junges, R. and Klügl, F. (2011), 'Evolution for Modeling: A Genetic Programming Framework for Sesam', in Krasnogor, N. and Lanzi, P.L. (eds.), *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation*, ACM, Dublin, Ireland, pp. 551-558.

Kavak, H., Padilla, J.J., Lynch, C.J. and Diallo, S.Y. (2018), 'Big Data, Agents, and Machine Learning: Towards a Data-driven Agent-based Modeling Approach', *Proceedings of the Spring Simulation Multiconference*, ACM, Baltimore, MD, pp. 12.

Kennedy, W. (2012), 'Modelling Human Behaviour in Agent-Based Models', in Heppenstall, A., Crooks, A.T., See, L.M. and Batty, M. (eds.), *Agent-based Models of Geographical Systems*, Springer, New York, NY, pp. 167-180.

Klüver, J. and Stoica, C. (2003), 'Simulations of Group Dynamics with Different Models', *Journal of Artificial Societies and Social Simulation*, 6(4): 8, Available at http://jasss.soc.surrey.ac.uk/6/4/8.html.

Koza, J.R. (1992), *Genetic Programming*, MIT Press, Cambridge, MA.

Kunz, J. (2011), 'Group-Level Exploration and Exploitation: A Computer Simulation-Based Analysis', *Journal of Artificial Societies and Social Simulation*, 14(4): 18, Available at http://jasss.soc.surrey.ac.uk/14/4/18.html.

Li, J. and Wilensky, U. (2009), NetLogo Sugarscape 2 Constant Growback, Available at https://ccl.northwestern.edu/netlogo/models/Sugarscape2ConstantGrowback [Accessed on July 28th, 2020].

Ma, T., Zhao, J., Xiang, S., Zhu, Y. and Liu, P. (2014), 'An Agent-Based Training System for Optimizing the Layout of AFVs' Initial Filling Stations', *Journal of Artificial Societies and Social Simulation*, 17(4): 6, Available at http://jasss.soc.surrey.ac.uk/17/4/6.html.

Macy, M.W. and Flache, A. (2002), 'Learning Dynamics in Social Dilemmas', *Proceedings of the National Academy of Sciences*, 99(3): 7229-7236.

Mahadevan, S. and Connell, J. (1992), 'Automatic Programming of Behavior-Based Robots Using Reinforcement Learning', *Artificial intelligence*, 55(2-3): 311-365.

Manson, S.M. (2005), 'Agent-based Modeling and Genetic Programming for Modeling Land Change in the Southern Yucatan Peninsular Region of Mexico', *Agriculture, Ecosystems & Environment*, 111(1-4): 47-62.

Narendra, K.S. and Thathachar, M.A. (1974), 'Learning Automata - A Survey', *IEEE Transactions on Systems, Man, and Cybernetics*, 4: 323-334.

Nowé, A., Verbeeck, K. and Peeters, M. (2006), 'Learning Automata as a Basis for Multi Agent Reinforcement Learning', in Tuyls, K., Jan't Hoen, P., Verbeeck, K. and Sen, S. (eds.), *International Workshop on Learning and Adaption in Multi-Agent Systems*, Springer, Utrecht, The Netherlands, pp. 71-85.

Ramchandani, P., Paich, M. and Rao, A. (2017), 'Incorporating Learning into Decision Making in Agent Based Models', in Oliveira, E., Gama, J., Vale, Z. and Cardoso, H.L. (eds.), *Progress in Artificial Intelligence: Proceedings of the 18th EPIA Conference on Artificial Intelligence*, Springer, Porto, Portugal, pp. 789-800.

Rand, W. (2006), 'Machine Learning Meets Agent-based Modeling: When Not to Go to a Bar', in Sallach, D., Macal, C.M. and North, M.J. (eds.), *Proceedings of the Agent 2006 Conference on Social Agents: Results and Prospects*, University of Chicago and Argonne National Laboratory, Chicago, IL, pp. 51-59.

Revay, P. and Cioffi-Revilla, C. (2018), 'Survey of Evolutionary Computation Methods in Social Agent-based Modeling Studies', *Journal of Computational Social Science*, 1: 115-146.

Rummery, G.A. and Niranjan, M. (1994), *On-Line Q-Learning Using Connectionist Systems*, Technical Report CUED/F-INFENG/TR 166, University of Cambridge, Department of Engineering, Cambridge, UK.

Russell, S.J. and Norvig, P. (2016), *Artificial Intelligence: A Modern Approach*, Pearson Education Limited, Harlow, England.

Sallans, B., Pfister, A., Karatzoglou, A. and Dorffner, G. (2003), 'Simulation and Validation of an Integrated Markets Model', *Journal of Artificial Societies and Social Simulation*, 6(4): 2, Available at http://jasss.soc.surrey.ac.uk/6/4/2.html.

Samuel, A.L. (1959), 'Some Studies in Machine Learning Using the Game of Checkers', *IBM Journal of Research and Development*, 3(3): 210-229.

Schelling, T.C. (1971), 'Dynamic Models of Segregation', *Journal of Mathematical Sociology*, 1(1): 143-186.

Sutton, R.S. (1988), 'Learning to Predict by the Methods of Temporal Differences', *Machine Learning*, 3(1): 9-44.

Sutton, R.S. and Barto, A.G. (2018), *Reinforcement Learning: An Introduction (2nd Edition)*, MIT Press, Cambridge, MA.

Takadama, K., Kawai, T. and Koyama, Y. (2008), 'Micro- and Macro-Level Validation in Agent-based Simulation: Reproduction of Human-like Behaviours and Thinking in a Sequential Bargaining Game', *Journal of Artificial Societies and Social Simulation*, 11(2): 9, Available at http://jasss.soc.surrey.ac.uk/11/2/9.html.

Tan, M. (1993), 'Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents', *Proceedings of the Tenth International Conference on Machine Learning*, ACM, Amherst, MA, pp. 330-337.

Tanabe, S. and Masuda, N. (2012), 'Evolution of Cooperation Facilitated by Reinforcement Learning with Adaptive Aspiration Levels', *Journal of Theoretical Biology*, 293: 151-160.

Tanev, I. and Shimohara, K. (2003), 'On Role of Implicit Interaction and Explicit Communications in Emergence of Social Behavior in Continuous Predators-Prey Pursuit Problem', in Cantu-Paz, E., Foster, J.A., Deb, K., Davis, L.D., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Standish, R., Kendall, G., Wilson, S., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A.C., Dowsland, K.A., Jonoska, N. and Miller, J. (eds.), *Proceedings of the 2003 Conference on Genetic and Evolutionary Computation*, Springer, Chicago, IL, pp. 74-85.

Vila, X. (2008), 'A Model-To-Model Analysis of Bertrand Competition', *Journal of Artificial Societies and Social Simulation*, 11(2): 11, Available at http://jasss.soc.surrey.ac.uk/11/2/11.html.

Wall, F. (2018), 'Emergence of Task Formation in Organizations: Balancing Units' Competence and Capacity', *Journal of Artificial Societies and Social Simulation*, 21(2): 6, Available at http://jasss.soc.surrey.ac.uk/21/2/6.html.

Watkins, C.J. (1989), *Learning from Delayed Rewards*, PhD Thesis, King's College, London, UK.

Wilensky, U. (1999), *NetLogo*. *http://ccl.northwestern.edu/netlogo*, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Wolpert, D.H., Wheeler, K.R. and Tumer, K. (1999), 'General Principles of Learning-based Multi-agent Systems', in Etzioni, O., Müller, J.P. and Bradshaw, J.M. (eds.), *Proceedings of the Third Annual Conference on Autonomous Agents*, ACM, Seattle, WA, pp. 77-83.

Xianyu, B. (2010), 'Social Preference, Incomplete Information, and the Evolution of Ultimatum Game in the Small World Networks: An Agent-Based Approach', *Journal of Artificial Societies and Social Simulation*, 13(2): 7, Available at http://jasss.soc.surrey.ac.uk/13/2/7.html.

Yamamoto, H., Okada, I., Taguch, T. and Muto, M. (2019), 'Effect of Voluntary Participation on an Alternating and a Simultaneous Prisoner's Dilemma', *Physical Review E*, 100(3): 032304.

Yuan, X., Schuchard, R. and Crooks, A.T. (2019), Examining Emergent Communities and Detecting Social Bots within the Polarized Online Vaccination Debate in Twitter, Social Media + Society. Available at https://doi.org/10.1177/2056305119865465.

Zschache, J. (2017), 'The Explanation of Social Conventions by Melioration Learning', *Journal of Artificial Societies and Social Simulation*, 20(3): 1, Available at http://jasss.soc.surrey.ac.uk/20/3/1.html.

Appendix A: Examples of Utilizing Evolutionary Computing within Agent-based Models to Study Social Issues.

In this appendix, readers will find a selection of papers that have utilized EC within agent-based models to study social issues. Before presenting Table A1, we provide a brief explanation of column headings to aid the reader:

- **Author**: The author(s) and year of publication.
- **Application**: All papers fall under the category of Sociology. The sub-category within sociology is listed.
- **Focus**: The author's intent in writing the paper leans either toward solving or exploring some sociology issue or toward demonstrating how an EC methodology could be applied.
- **Entity**: The entity types the agents represent in the model.
- **Behavior**: The method used by each agent to model human behavior. This is an extension of Kennedy's (2012) typology of mathematical behavior of human agents. All behaviors are mathematical and a type of Machine Learning (ML). The types can be evolutionary computing/evolutionary algorithm (EC/EA), evolutionary computing/evolutionary programming (EC/EP), artificial neural network (ANN), or reinforcement learning (RL).
- **Agent Scale**: The number of agents used in the model shown in a power of ten notations.
- Spatial Scale: The spatial size used by the model, if any. Network is a special spatial scale where there is no distance, but there is adjacency.

- **Time Scale**: Shown as interval or ordinal from Steven's (1946) typology. Interval means the temporal distance between units is identical. The unit can be explicit (years) or abstract (steps). Ordinal means the distance between units is undefined but is monotonic and usually increasing (generation).

Table A1: A selection of applications utilizing evolutionary computing to study social science applications.

Author	Application	Focus[1]	Entity	Behavior	Agent Scale	Spatial Scale	Time Scale
Arifovic (1994)	Economics	S	Firm	ML/EC/EA	10^0	None	Interval, Steps (10,000)
Chen & Yeh (1996)	Economics	M	Firm	ML/EC/EP	10^2	None	Interval, Steps, Generations (1,000)
Curran & O'Riordan (2007)	Social Dynamics	S	Individual	ML/EC/EA ML/ANN	10^1	Network	Interval, Steps (400)
Dionne et al. (2019)	Social Dynamics	S	Individual	ML/EC/EA	10^2	Network	Interval, Steps (60)
Edmonds (1999)	Social Dynamics	S	Individual	ML/EC/EP	10^0	None	Interval, Weeks (100)
Fischer (2003)	Social Dynamics	M	Individual	ML/EC/EA	10^0	None	Interval, Steps (10,000)
Francisco and Jorge dos Reis, (2008)	Teamwork	M	Individual	ML/EC/EP	10^2	Interval, Grid (5,000x5,000)	Ordinal (100)
Grefenstette (1992)	Teamwork	M	Individual	ML/EC/EP	10^0	None	Interval, Steps, Generations (100)
Haynes & Sen (1996)	Teamwork	M	Individual	ML/EC/EP	10^0	Interval, Grid (30x30)	Interval, Steps (100)
Hsu & Gustafson (2002)	Teamwork	M	Individual	ML/EC/EP	10^3	Interval, Grid (20x20)	Interval, Steps (400)
Jang et al. (2019)	Social Dynamics	S	Individual, Organization	ML/EC/EA	10^2	Network	Interval
Jim & Giles (2000)	Teamwork	M	Individual	ML/EC/EA	10^2	Interval, Grid (30x30)	Interval, Steps (5,000)
Klüver & Stoica (2003)	Social Dynamics	M	Individual	ML/EC/EA ML/ANN	10^0	Network	Interval
Kunz (2011)	Social Dynamics	S	Group	ML/EC/EA	10^1	None	Interval, Steps (200)
Manson (2005)	Economics	S	Household, Organization	ML/EC/EP	10^1	Interval, Grid (cell 28.5^2 m)	Interval, Years (9)
Tanev & Shimohara (2003)	Teamwork	M	Individual	ML/EC/EP	10^0	Interval, Grid [mm] (1,600x1,000)	Interval, Half-Second (600)
Vila (2008)	Economics	S & M	Individual	ML/EC/EA	10^3	None	Interval, Steps (500)
Xianyu (2010)	Social Dynamics	S	Individual	ML/EC/EA ML/RL	10^3	Network	Interval, Steps (5,000)
Yamamoto et al. (2019)	Social Dynamics	S	Individual	ML/EC/EA	10^2	None	Interval, Steps [200 Rounds x 200 Generations] (40,000)

[1] M is for Methodology and S is for Solve

Appendix 2: Examples of Utilizing Reinforcement Learning within Agent-based Models to Study Social Issues.

In this appendix, readers will find a selection of papers that have utilized RL within agent-based models to study social issues. To assist with interpretation of Table A2, the column headings are the same as in Table A1 with the exception of behavior. Here behavior refers to the method used by each agent to model human behavior. This is an extension of Kennedy's (2012) typology of mathematical behavior of human agents. All behaviors are mathematical and a type of Machine Learning (ML). The types can one of the following *Q-Learning* (QL), *Temporal-Difference* (TD), *Bush-Mostellar* (BM), *State→Action→Reward→State→Action* (SARSA), and *Learning Automata* (LA).

TableA2: A selection of applications utilizing reinforcement learning to study social science applications.

Author	Application	Focus[2]	Entity	Behavior	Agent Scale	Spatial Scale	Time Scale
Chmura & Pitz (2007)	Social Dynamics	S	Individual	ML/RL/QL	10^1	None	Interval, Steps (500)
Christensen & Sasaki (2008)	Social Dynamics	S	Individual	ML/RL/TD	10^1	Square Meters (4,000)	Interval, 25 Milliseconds (4,880)
Claus & Boutilier (1998)	Social Dynamics	M	Individual	ML/RL/QL	10^0	None	Interval, Steps (2,500)
Clempner (2017)	Social Dynamics	M	Individual	ML/RL/TD	10^0	None	Interval, Steps (Unknown)
Hao & Leung (2013)	Social Dynamics	S	Individual	ML/RL/QL	10^2	None	Interval, Rounds (5,000)
Izquierdo et al. (2008)	Social Dynamics	M	Individual	ML/RL/BM	10^0	None	Interval, Steps (1,000,000)
Junges & Klügl (2011)	Social Dynamics	M	Individual	ML/RL/QL	10^0	Continuous, Meters (20x30)	Interval, Half-Second (50,000)
Macy & Flache (2002)	Social Dynamics	M	Individual	ML/RL/BM	10^0	None	Interval, Steps (500)
Mahadevan & Connell (1992)	Task Managing	M	Individual	ML/RL/QL	10^0	Interval, Grid (37x19)	Interval, Steps (2,000)
Nowé et al. (2006)	Social Dynamics	M	Individual	ML/RL/LA	10^0	None	Interval, Steps (500)
Ramchandani et al. (2017)	Economics	M	Individual	ML/RL/QL	10^3	None	Interval, Months (540)
Sallans et al. (2003)	Economics	S	Firms, Traders, & Consumers	ML/RL/ SARSA	Not Known	None	Interval, Steps (4,500)
Takadama et al. (2008)	Social Dynamics	M	Individual	ML/RL/QL	10^0	None	Interval, Steps (10,000,000)
Tan (1993)	Teamwork	M	Individual	ML/RL/QL	10^0	Interval, Grid (10x10)	Interval, Steps (10,000)
Tanabe & Masuda (2012)	Social Dynamics	M	Individual	ML/RL/BM	10^3	None	Interval, Steps (200)
Wall (2018)	Social Dynamics	S	Organization Unit	ML/RL/BM	10^0	Network	Interval, Steps (Unknown)
Wolpert et al. (1999)	Social Dynamics	M	Individual	ML/RL/QL	10^2	Network	Interval, Weeks (4,000)
Zschache (2017)	Social Dynamics	M	Individual	ML/RL/QL	10^4	Network	Interval, Steps (1,000)

[2] M is for Methodology and S is for Solve