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

Dale K. Brearcliffe
PhD Student, Computational Social Science
Dr. Andrew Crooks
Professor, Buffalo University

GEORGE MASON UNIVERSITY

Agenda

- Introduction
- Background
- Methodology
- Results
- Conclusion

Introduction

- Paper's purpose:
  - Demonstrate the integration of three machine learning (ML) methods into a well-known agent based model, Sugarscape
  - Demonstrate how different methods alter the outcomes of a simulation
  - Answer the Question: Is it worth the effort to add machine learning to agent-based models?

Background

- We focus on two types of ML: Evolutionary Computing (EC), and Reinforcement Learning (RL)
- The 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
    - ML used specifically to optimize model parameters or explore model outputs are not considered within this paper since these processes enhance the modeler rather than the model

Background (Continued)

- 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."
- Most agent-based models utilize simple rules that are capable of creating complex, system level results
  - The simple rules, however, do not always provide the desired results

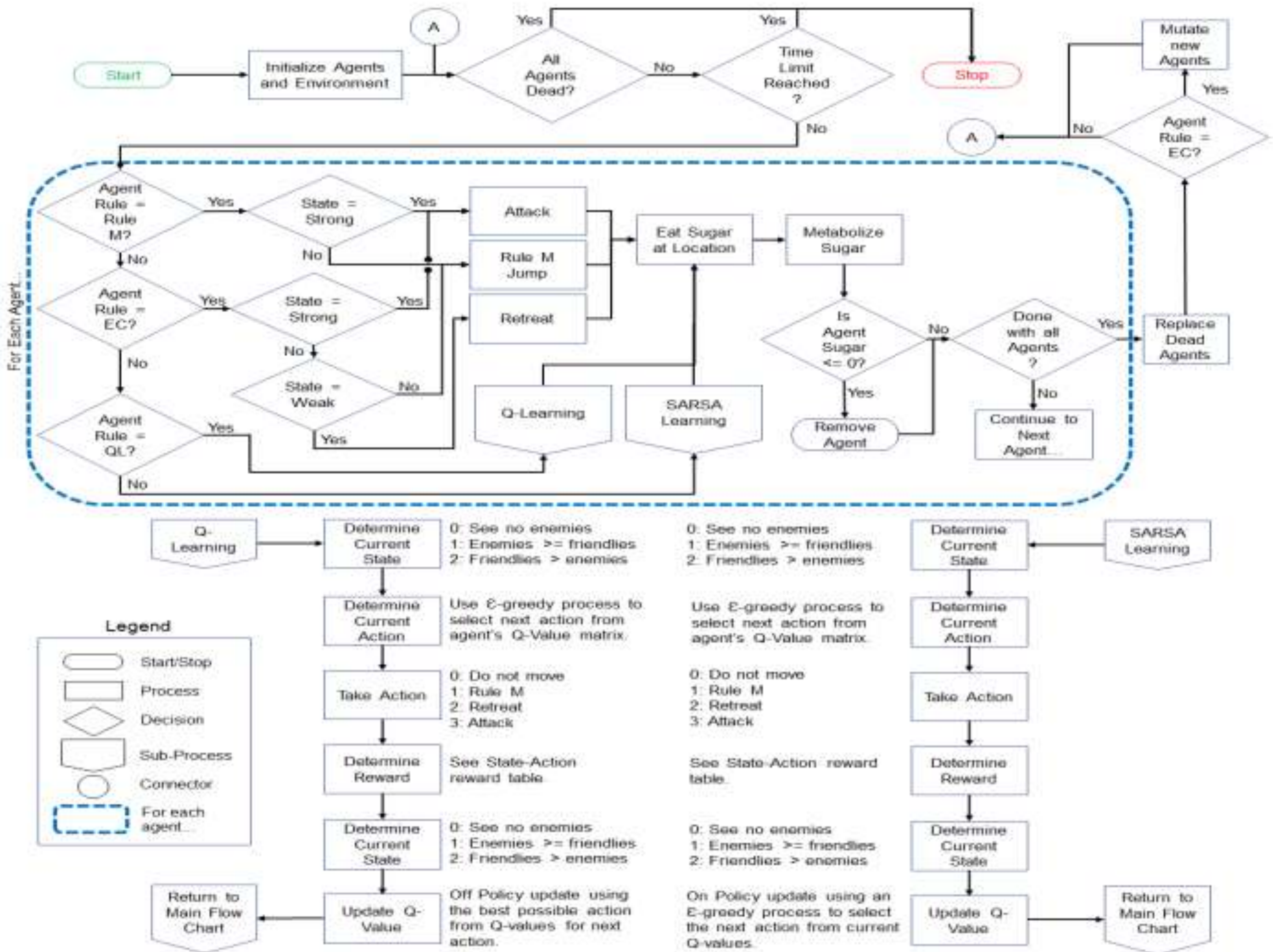
Methodology - Introduction

- 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" model, which is included in the NetLogo models library
  - Sugarscape 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

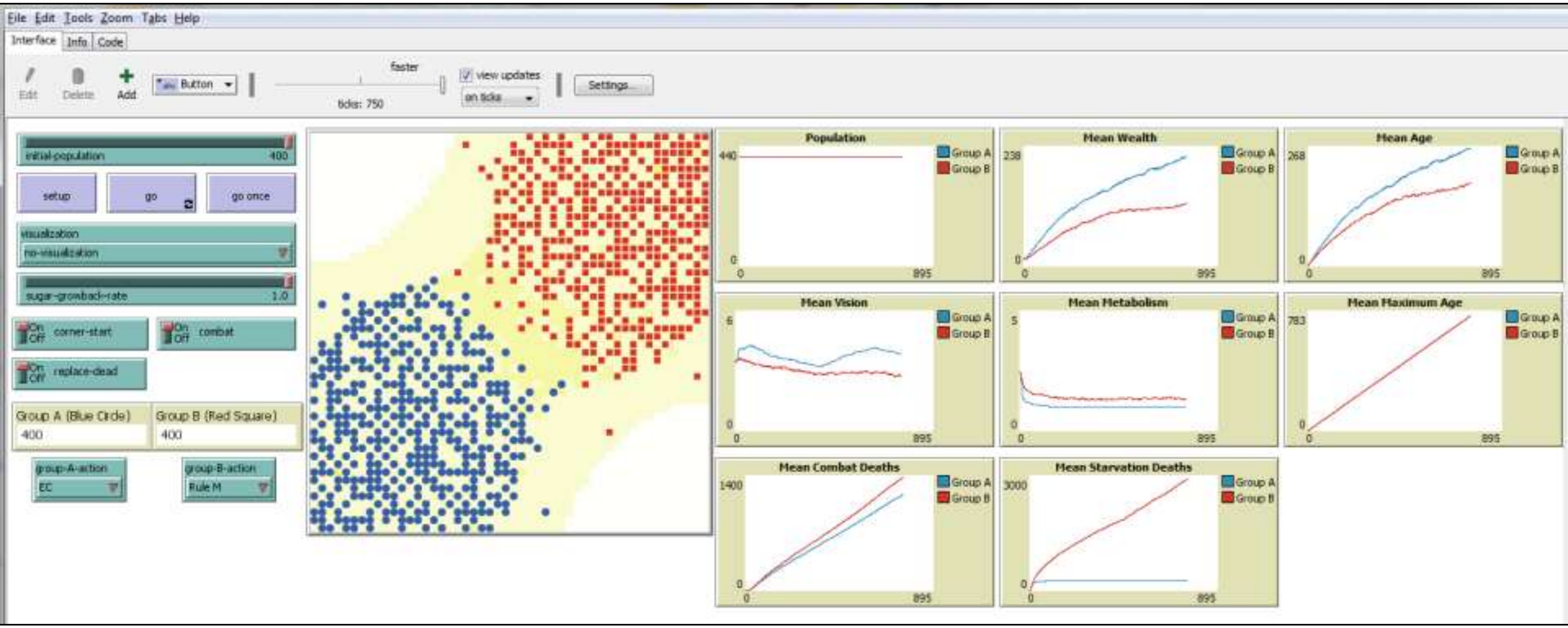
Methodology - Rationale

- Our rationale for choosing Sugarscape 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
    - 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 ran out of sugar
    - Our model adds combat with agents that can attack and retreat

Model Execution Flowchart



Model GUI (NetLogo Framework)



Methodology – Models (Rule M & EC)

- Rule M
  - Use explicit knowledge in the form of *a priori* rules imposed by the modeler - these agents do not learn and attributes do not change
- EC (Evolutionary Computing)
  - Similar to Rule M, but the agents learn as a collective when they die and are replaced using the wealthiest agents as templates for the new agent's metabolism and vision attribute values

Methodology – Models (Q-Learning)

- Q-Learning (Reinforcement Learning)
  - Begin with no explicit knowledge, relying on tacit knowledge accumulated and remembered over time
  - Progress over time from random explorations to experience based decisions
  - 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

Methodology – Models (SARSA)

- SARSA (Reinforcement Learning)
  - Begin with no explicit knowledge, relying on tacit knowledge accumulated and remembered over time
  - Progress over time from random explorations to experience based decisions
  - Make on-policy updates following and updating their experience using their existing policy
- See last slide for more information

Methodology – Model Parameters

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.

- Using the initial parameters in the table above, each pairwise comparison of models is run 50 times for 20,001 time steps
- Results are calculated for the mean of the 50 runs

Methodology – Model Runs

- Tracked attributes for each collective include population, wealth, vision distance, metabolism, combat deaths, starvation deaths, mean age, and maximum age
  - Population was stable with Replace Dead parameter set to ON
- Pairwise comparisons of the four model methodologies results in 16 comparisons for each tracked attribute

Results - Wealth

- 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 and a slightly higher metabolism
- EC learning maintained a significant advantage in wealth accumulation over the two RL methods and does so with a near equal metabolism
  - It does this with a much higher vision distance that declines over time

Results - Wealth (Continued)

- 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
  - Both maintained almost identical high vision distances and low metabolisms
- Compared to the other methods, the agents utilizing the SARSA method do not accumulate as much wealth

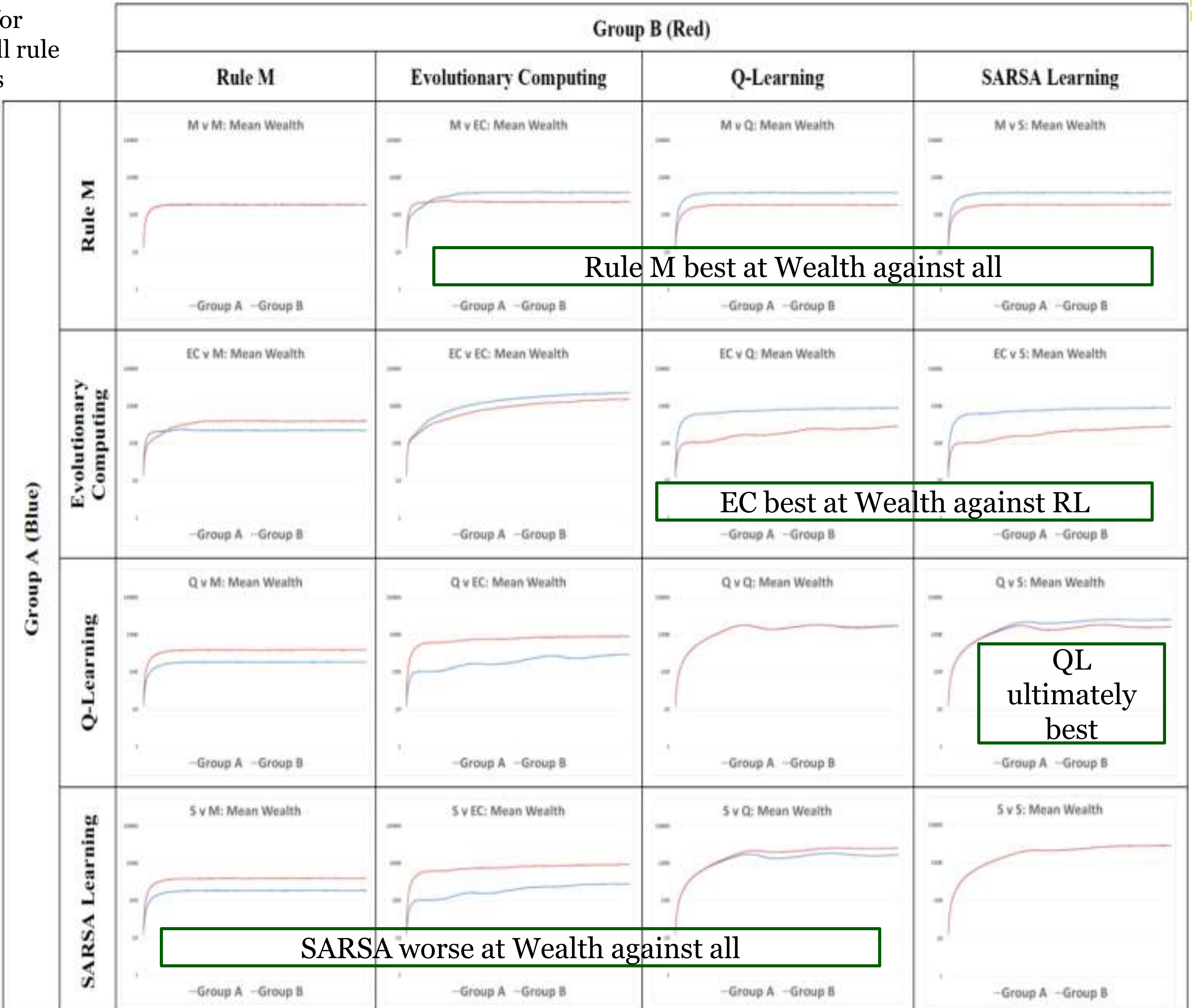
Results - Combat

- 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
  - This is possibly attributable to the EC group maintaining a higher vision distance 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
- The two RL methodologies fighting against each other showed a noticeable initial period with very few combat deaths

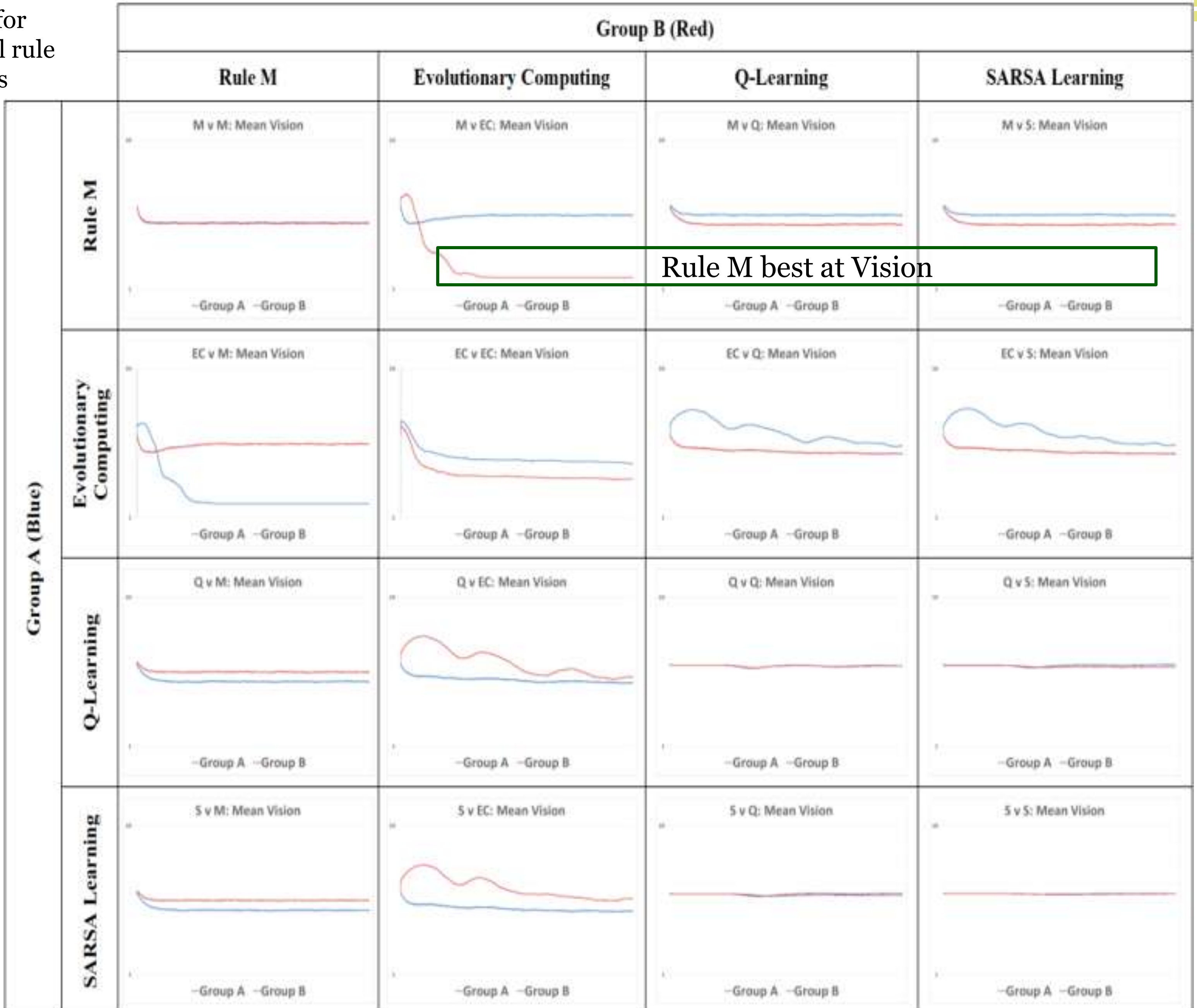
Results - Other

- The EC method was noticeably superior to the other three methodologies in finding sufficient sugar to metabolize and not starve
- All methodologies demonstrated a decrease in mean metabolism, 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
    - 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)

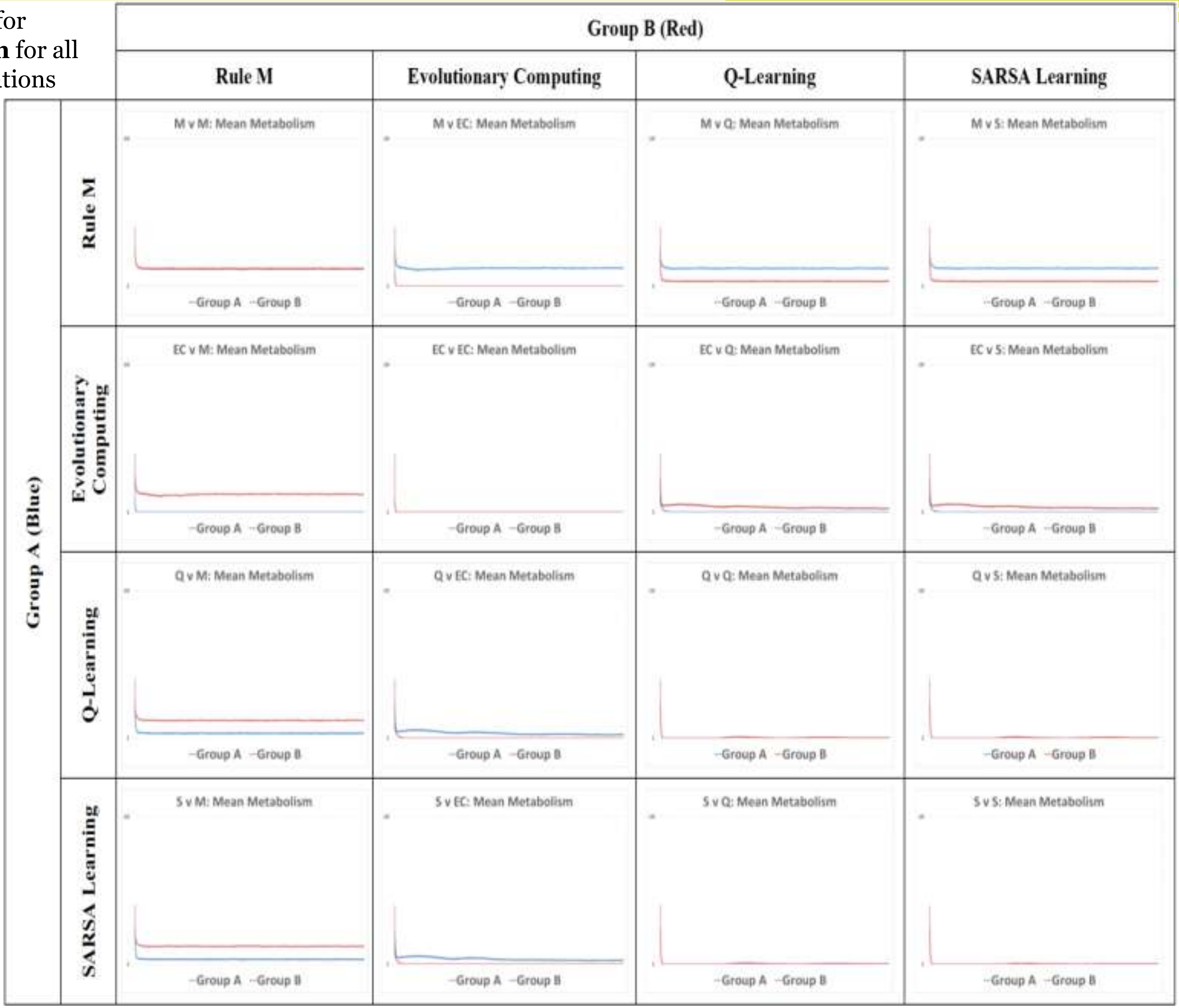
Mean result for **wealth** for all rule combinations



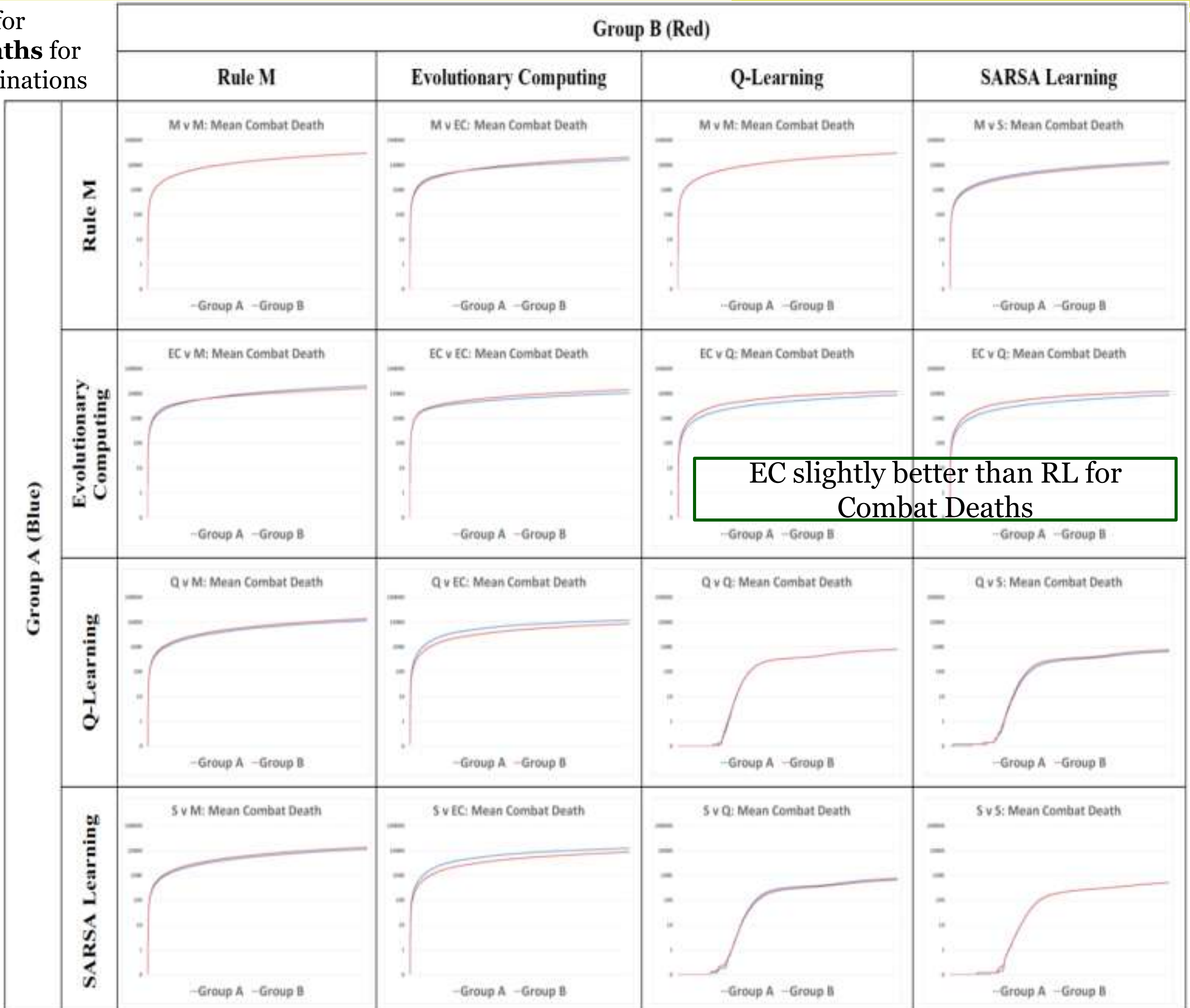
Mean result for **vision** for all rule combinations



Mean result for **metabolism** for all rule combinations

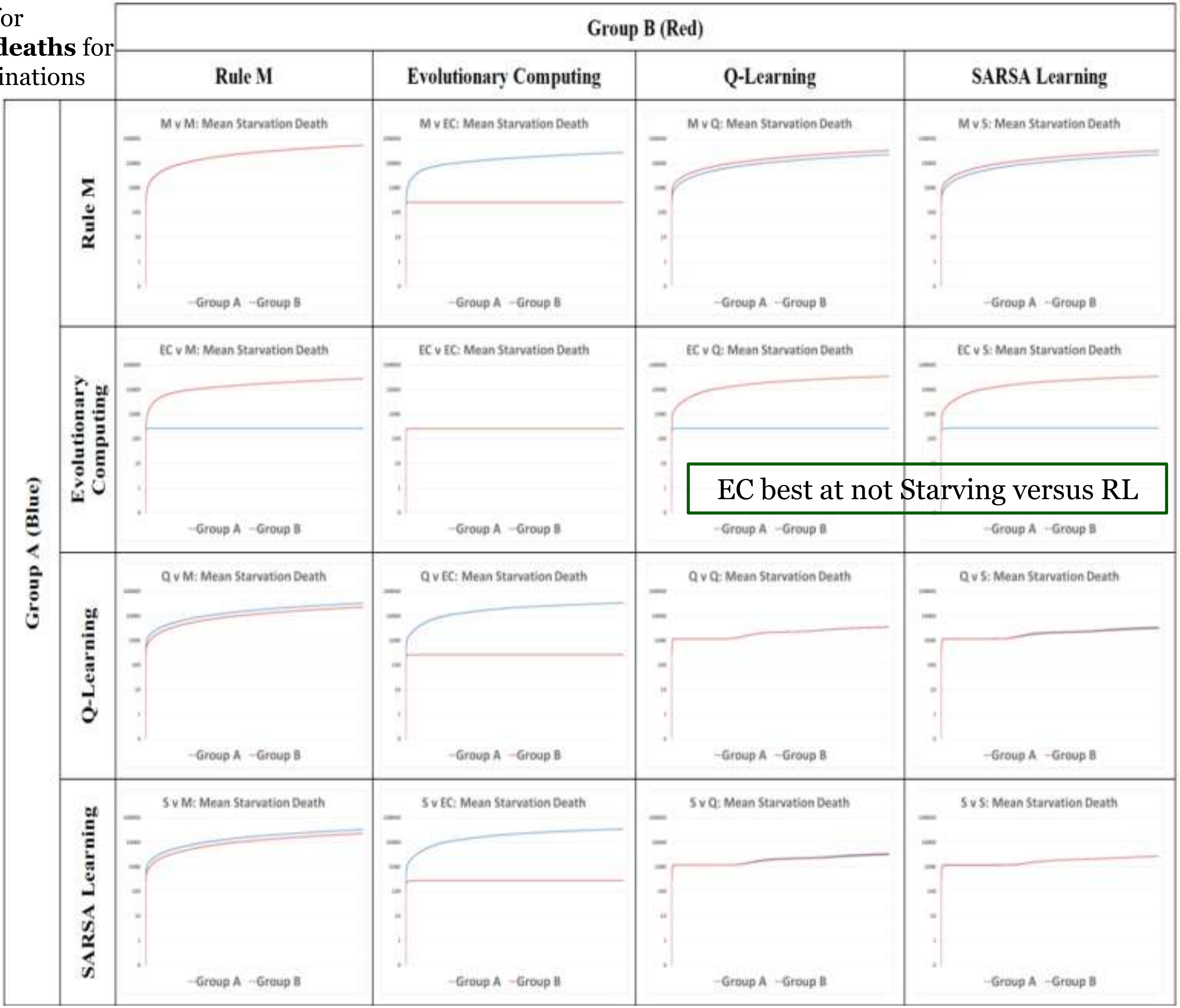


Mean result for **combat deaths** for all rule combinations

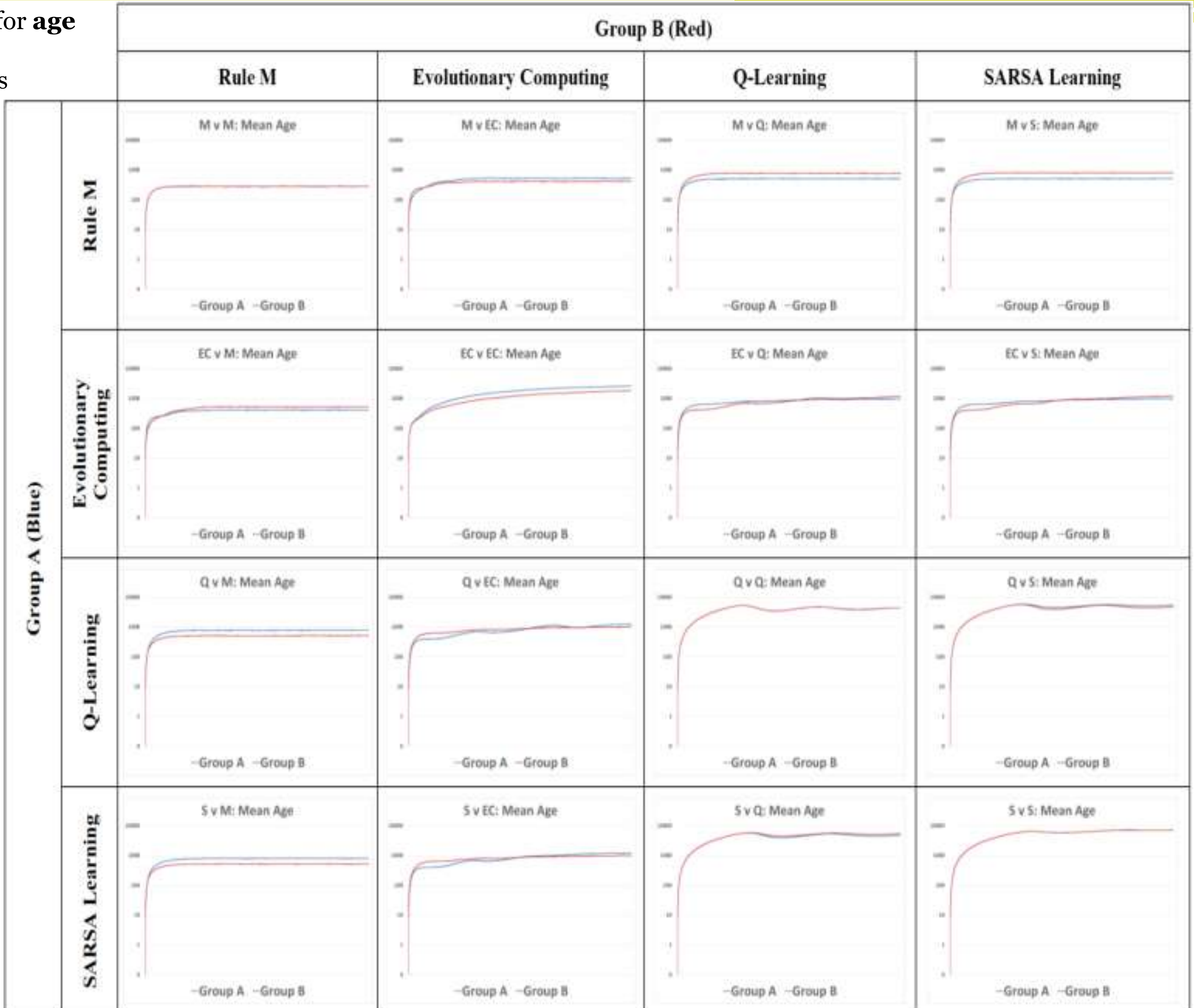


EC slightly better than RL for Combat Deaths

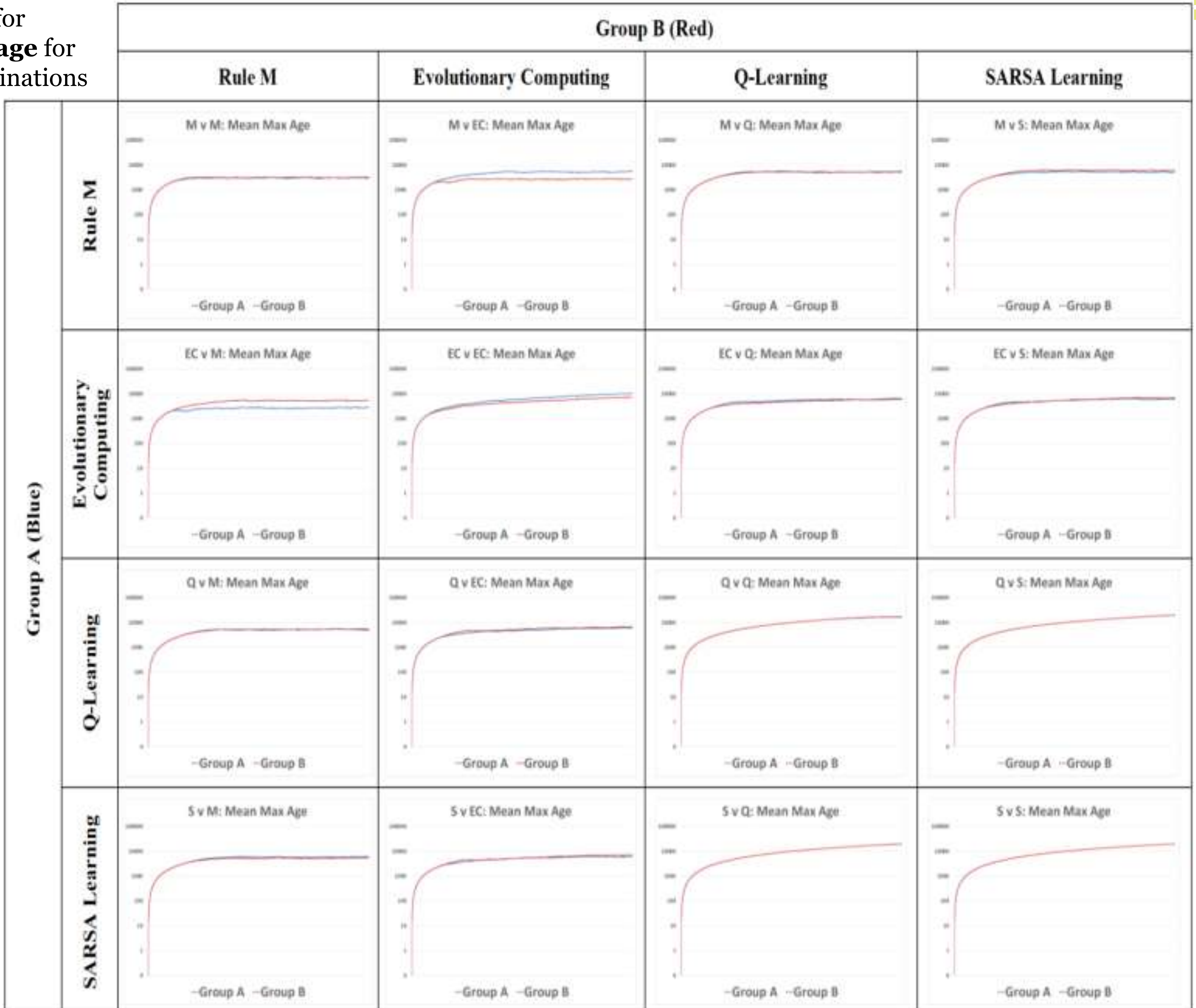
Mean result for **starvation deaths** for all rule combinations



Mean result for **age** for all rule combinations



Mean result for **maximum age** for all rule combinations



Conclusion

- We adapted the Sugarscape model with three types of learning agents (i.e. a EC, Q-Learning and SARSA) and contrasted them with Rule M from the original model
  - We found that different learning methods do have an impact on the simulation outcome
  - We were surprised that Rule M agents were able to accumulate more wealth than the ML agents
- Modelers who wish to use ML methods for agent learning should pay particular attention and justify why they chose one method over another

Conclusion (Continued)

- Is it worth the effort to add ML to agent-based models?
- ML requires additional thought
  - In traditional rule-based models, the agents' actions are designed specifically to match situations the modeler anticipates (explicit knowledge)
  - In RL, rewards are used to push the agent toward or away from each state-action combination letting the agent develop their own rules (tacit knowledge)

Conclusion (Continued)

- ML may require additional computer resources and time
  - The SARSA method took almost seven time longer to execute (wall clock) than Rule M
- ML is worth it if the goal is unachievable by simpler methods or if there is a need for the agents to learn from good and bad experiences (memory)

More information

- ODD[1], NetLogo[2] Model, and Symmetric Map
  - https://tinyurl.com/ML-Agents
- Reinforcement Learning
  - Sutton & Barto Book (2018) Reinforcement Learning: An Introduction
    - http://incompleteideas.net/book/the-book-2nd.html
  - David Silver Lecture #4 (2015)
    - https://www.youtube.com/watch?v=PnHCvfgC_ZA
- Dale Brearcliffe
  - http://tangledinfo.com/
  - dbrearcl@gmu.edu
- Dr. Andrew Crooks
  - https://www.gisagents.org/
  - atcrooks@buffalo.edu

[1] *The ODD Protocol for Describing Agent-Based and Other Simulation Models: A Second Update to Improve Clarity, Replication, and Structural Realism*. **Grimm et al.**, 2020, JASSS.
[2] **Wilensky, Uri**. NetLogo. Evanston : Center for Connected Learning and Computer-Based Modeling, Northwestern University, 1999.