Agents' Social Environment Rewarding System Algorithm - ASERSA

Budd McCrackn

Introduction

ASERSA is a concept that integrates socio-economic principles into an agent-based system where agents interact and develop based on variables such as wealth, influence, responsibility, self-esteem, willpower, ambition, competence, inspiration, and action level. The aim is to create a dynamic environment where agents not only optimize their individual goals but also contribute to the community.

This document develops mathematical equations that formally describe the ASERSA model, integrating foundational elements from the Dynamic Force Index Algorithm (DFIA). While DFIA provides the environment for calculating key attributes like volume, influence, and force, ASERSA builds upon these to model complex agent dynamics. We introduce non-linear relationships between variables to create more realistic and intricate interactions. Concrete examples and descriptions of visual elements are included to make the concept more tangible.

1. Definition of Agent Variables

For each agent X at time t, we define the following variables:

- $W_X(t)$: Wealth
- $I_X(t)$: Influence
- $AS_X(t)$: Agent Status
- $R_X(t)$: Responsibility
- $S_X(t)$: Self-Esteem
- $V_X(t)$: Willpower
- $A_X(t)$: Ambition
- $C_X(t)$: Competence
- $IN_X(t)$: Inspiration
- $AL_X(t)$: Action Level

Example: Let's assume that agent X starts with the following values at time $t = 0$:

- $W_X(0) = 50$ (average wealth)
- The remaining variables are initialized accordingly based on the functions below.

2. Dynamic Force Index Algorithm (DFIA)

The **Dynamic Force Index Algorithm (DFIA)** is a foundational component that creates an environment enabling the calculation of essential attributes such as volume, influence, and force. These attributes serve as the basis for ASERSA's equations representing responsibility, self-esteem, willpower, inspiration, ambition, competence, action level, and more. While DFIA operates independently, it seamlessly integrates with ASERSA to facilitate complex agent dynamics.

3. Non-Linear Causal Relationships between Variables

We model the causal relationships between the variables using non-linear functions to create more complex and realistic dynamics:

1. **Wealth Influences Influence:**

$$I_X(t) = I_{\max} \cdot \frac{1}{1 + e^{-k_1 (W_X(t) - W_0)}}$$

where:

- $I_{\max}$: Maximum influence
- k_1: Growth rate
- W_0: Wealth value at which influence is half of $I_{\max}$

Example: Assuming $I_{\max} = 100$, $k_1 = 0.1$, $W_0 = 50$, and $W_X(0) = 50$:

$$I_X(0) = 100 \cdot \frac{1}{1 + e^{-0.1(50-50)}} = 50$$

2. **Influence Affects Agent Status:**

$$AS_X(t) = k_2 \cdot I_X(t)^{\alpha}$$

where:

- k_2: Proportionality constant
- $\alpha > 1$: Exponent controlling non-linearity

Example: With $k_2 = 1$, $\alpha = 1.5$, and $I_X(0) = 50$:

$$AS_X(0) = 1 \cdot 50^{1.5} \approx 353.55$$

3. **Agent Status Affects Responsibility:**

$$R_X(t) = \frac{AF_X(t)}{SF_X(t)} \cdot R_{\text{opt}}$$

where:

- $AF_X(t)$: Agent's relative force
- $SF_X(t)$: Society's relative force
- R_{opt}: Optimal responsibility level

Example: With $AF_X(t) = 50$, $SF_X(t) = 150$, and $R_{\text{opt}} = 100$:

$$R_X(t) = \frac{50}{150} \cdot 100 \approx 33.33$$

4. **Responsibility Affects Self-Esteem:**

$$S_X(t) = \left(\frac{AS_X(t)}{SS_X(t) + AS_X(t)}\right)^2 \cdot S_{\text{opt}}$$

where:

- $SS_X(t)$: Society's relative status
- S_{opt}: Optimal self-esteem level

Example: Assuming $AS_X(t) = 353.55$, $SS_X(t) = 100$, and $S_{\text{opt}} = 100$:

$$S_X(t) = \left(\frac{353.55}{100 + 353.55}\right)^2 \cdot 100 \approx 77.74$$

5. **Self-Esteem Affects Willpower:**

$$V_X(t) = V_{\max} \cdot \frac{1}{1 + e^{-k_5(S_X(t) - S_0)}}$$

where:

- $V_{\max}$: Maximum willpower
- k_5: Growth rate
- S_0: Self-esteem value at which willpower is half of $V_{\max}$

Example: With $V_{\max} = 100$, $k_5 = 0.1$, $S_0 = 50$, and $S_X(t) = 77.74$:

$$V_X(t) = 100 \cdot \frac{1}{1 + e^{-0.1(77.74 - 50)}} \approx 99.98$$

6. **Willpower Affects Ambition:**

$$A_X(t) = k_6 \cdot \left(1 - e^{-\frac{IN_X(t)}{R_X(t)}}\right)$$

where:

- k_6: Proportionality constant
- $IN_X(t)$: Inspiration

Example: With $k_6 = 0.01$, $IN_X(t) = 19.874$, and $R_X(t) = 33.33$:

$$A_X(t) = 0.01 \cdot \left(1 - e^{-\frac{19.874}{33.33}}\right) \approx 0.01 \cdot (1 - e^{-0.596}) \approx 0.01 \cdot 0.450 \approx 0.0045$$

7. **Ambition Affects Competence:**

$$C_X(t+1) = C_X(t) + k_7 \cdot A_X(t) \cdot (C_{\max} - C_X(t)) + \kappa \cdot \left(\overline{C}_{\text{best}}(t) - C_X(t)\right)$$

where:

- k_7: Learning rate
- κ: Learning rate from best-performing agents
- $\overline{C}_{\text{best}}(t)$: Average competence of the best-performing agents
- $C_{\max}$: Maximum competence

Example: With $k_7 = 0.01$, $A_X(t) = 0.0045$, $C_{\max} = 100$, $C_X(t) = 20$, $\kappa = 0.1$, and $\overline{C}_{\text{best}}(t) = 80$:

$$C_X(t+1) = 20 + 0.01 \cdot 0.0045 \cdot (100 - 20) + 0.1 \cdot (80 - 20) \approx 20 + 0.0036 + 6 \approx 26.0036$$

8. **Competence Influences Inspiration:**

$$IN_X(t) = \begin{cases} \sqrt{\phi \cdot \frac{AI_X(t)}{SI_X(t)}} & \text{if } \phi \cdot \frac{AI_X(t)}{SI_X(t)} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

where:

- ϕ: Sensitivity to inspiration
- $AI_X(t)$: Agent's relative influence
- $SI_X(t)$: Society's relative influence

Example: With $\phi = 0.5$, $AI_X(t) = 11.11$, and $SI_X(t) = 50$:

$$IN_X(t) = \sqrt{0.5 \cdot \frac{11.11}{50}} \approx \sqrt{0.1111} \approx 0.333$$

9. **Inspiration and Willpower Affect Action Level:**

$$AL_X(t) = \psi \cdot \left((C_X(t) \cdot V_X(t) \cdot A_X(t))^{\frac{1}{3}}\right) \cdot \left(1 - e^{-\left((C_X(t) \cdot V_X(t) \cdot A_X(t))^{\frac{1}{3}}\right)}\right)$$

where:

- ψ: Proportionality constant

Example: With $\psi = 0.01$, $C_X(t) = 40.251$, $V_X(t) = 99.98$, and $A_X(t) = 0.0045$:

$$AL_X(t) = 0.01 \cdot \left((40.251 \cdot 99.98 \cdot 0.0045)^{\frac{1}{3}}\right) \cdot \left(1 - e^{-\left((40.251 \cdot 99.98 \cdot 0.0045)^{\frac{1}{3}}\right)}\right) \approx 0.01 \cdot 3.36 \cdot (1 - e^{-3.36}) \approx 0.01 \cdot 3.36 \cdot 0.967$$

4. Adaptive Reward Function in Reinforcement Learning

We define the reward function $r_X(t)$ with dynamically adjusting weights $\alpha(t)$, $\beta(t)$, and $\gamma(t)$:

$$r_X(t) = \alpha(t) \cdot \Delta W_X(t) + \beta(t) \cdot C_X^{\text{comm}}(t) + \gamma(t) \cdot \Delta AS_X(t)$$

where $\alpha(t)$, $\beta(t)$, and $\gamma(t)$ are weights that sum to 1 at each time t:

$$\alpha(t) + \beta(t) + \gamma(t) = 1$$

Individual Utility $\Delta W_X(t)$:

$$\Delta W_X(t) = W_X(t) - W_X(t-1)$$

Contribution to the Community $C_X^{\text{comm}}(t)$:

$$C_X^{\text{comm}}(t) = \tau_X(t) \cdot W_X(t)$$

Self-improvement $\Delta AS_X(t)$:

$$\Delta AS_X(t) = AS_X(t) - AS_X(t-1)$$

4.1. Dynamic Adjustment of Weights

The weights $\alpha(t)$, $\beta(t)$, and $\gamma(t)$ are updated dynamically based on the agent's performance and interactions over time.

Performance Measure:

We define a performance measure for the agent, such as a moving average of the rewards:

$$P_X(t) = (1 - \lambda) \cdot r_X(t) + \lambda \cdot P_X(t-1)$$

where $0 < \lambda < 1$ is a smoothing parameter.

Updating Weights:

The weights are updated using gradient ascent to maximize the performance measure:

$$\theta(t+1) = \theta(t) + \eta \cdot \delta(t) \cdot \frac{\partial r_X(t)}{\partial \theta}$$

where:

- $\theta(t) \in \{\alpha(t), \beta(t), \gamma(t)\}$
- η is the learning rate
- $\delta(t)$ is the temporal difference error:

$$\delta(t) = r_X(t) + \lambda \cdot P_X(t) - P_X(t-1)$$

Gradient Calculation:

$$\frac{\partial r_X(t)}{\partial \alpha(t)} = \Delta W_X(t)$$
$$\frac{\partial r_X(t)}{\partial \beta(t)} = C_X^{\text{comm}}(t)$$
$$\frac{\partial r_X(t)}{\partial \gamma(t)} = \Delta AS_X(t)$$

Normalization of Weights:
After updating, the weights are normalized to ensure they sum to 1:

$$\theta(t+1) = \frac{\theta(t+1)}{\alpha(t+1) + \beta(t+1) + \gamma(t+1)}$$

Example:
Assuming:

- Initial weights: $\alpha(0) = 0.4$, $\beta(0) = 0.3$, $\gamma(0) = 0.3$
- Learning rate: $\eta = 0.05$
- Smoothing parameter: $\lambda = 0.9$
- At time $t = 1$, we have:
  - $\Delta W_X(1) = 5$
  - $C_X^{\text{comm}}(1) = 2$
  - $\Delta AS_X(1) = 1$
  - $r_X(1) = 0.4 \cdot 5 + 0.3 \cdot 2 + 0.3 \cdot 1 = 2 + 0.6 + 0.3 = 2.9$
  - $P_X(1) = (1 - 0.9) \cdot 2.9 + 0.9 \cdot P_X(0)$

Assuming $P_X(0) = 2.9$:

$$P_X(1) = 0.1 \cdot 2.9 + 0.9 \cdot 2.9 = 2.9$$

Compute the temporal difference error:

$$\delta(1) = 2.9 + 0.9 \cdot 2.9 - 2.9 = 2.61$$

Update weights:

$$\alpha(1) = 0.4 + 0.05 \cdot 2.61 \cdot 5 = 0.4 + 0.6525 = 1.0525$$
$$\beta(1) = 0.3 + 0.05 \cdot 2.61 \cdot 2 = 0.3 + 0.261 = 0.561$$
$$\gamma(1) = 0.3 + 0.05 \cdot 2.61 \cdot 1 = 0.3 + 0.1305 = 0.4305$$

Normalize weights:

$$\text{Total} = 1.0525 + 0.561 + 0.4305 = 2.044$$

$$\alpha(1) = \frac{1.0525}{2.044} \approx 0.515$$
$$\beta(1) = \frac{0.561}{2.044} \approx 0.274$$
$$\gamma(1) = \frac{0.4305}{2.044} \approx 0.211$$

The updated weights reflect a higher emphasis on individual utility due to its higher contribution to the reward.

5. Tax and Redistribution Mechanism

To model the agents' responsibility to contribute to the community, ASERSA introduces a tax and redistribution system. The tax rate is dynamically calculated based on agents' wealth, status, and overall economic stability.

5.1. Tax Rate Calculation

The tax rate $\tau_X(t)$ for agent X at time t is defined as:

$$\tau_X(t) = \tau_{\max} \cdot \left(\omega_W \cdot \frac{W_X(t)}{W_{\text{total}}(t)} + \omega_{AS} \cdot \frac{AS_X(t)}{AS_{\text{opt}}} + \omega_E \cdot E(t) \right)$$

where:

- $\tau_{\max}$: Maximum tax rate
- ω_W, ω_{AS}, ω_E: Weights for wealth, agent status, and economic stability respectively, such that $\omega_W + \omega_{AS} + \omega_E = 1$
- $W_{\text{total}}(t) = \sum_X W_X(t)$: Total wealth of all agents at time t
- AS_{opt}: Optimal agent status level
- $E(t)$: Economic stability factor at time t (normalized between 0 and 1)

Explanation:

- **Wealth Factor**: Reflects the agent's share of the total wealth.
- **Agent Status Factor**: Higher status agents contribute more, reflecting their greater responsibility.
- **Economic Stability Factor**: Adjusts the tax rate based on the overall economic conditions. During economic downturns ($E(t)$ increases), taxes may increase to support societal needs.

Example: Assuming:

- $\tau_{\max} = 0.4$
- $\omega_W = 0.5$, $\omega_{AS} = 0.3$, $\omega_E = 0.2$
- $W_X(t) = 70$, $W_{\text{total}}(t) = 100$
- $AS_X(t) = 40$, $AS_{\text{opt}} = 100$
- $E(t) = 0.2$

Compute $\tau_X(t)$:

$$\tau_X(t) = 0.4 \cdot \left(0.5 \cdot \frac{70}{100} + 0.3 \cdot \frac{40}{100} + 0.2 \cdot 0.2\right) = 0.4 \cdot (0.35 + 0.12 + 0.04) = 0.4 \cdot 0.51 = 0.204$$

5.2. Tax Collection

The tax collected from agent X is:

$$\text{Tax}_X(t) = \tau_X(t) \cdot W_X(t)$$

Example: With $\tau_X(t) = 0.204$ and $W_X(t) = 70$:

$$\text{Tax}_X(t) = 0.204 \times 70 = 14.28$$

5.3. Redistribution Mechanism

The total tax collected $T(t)$ is redistributed to agents based on their relative deprivation.

$$T(t) = \sum_X \text{Tax}_X(t)$$

Each agent X_j receives a redistribution amount $D_{X_j}(t)$ calculated as:

$$D_{X_j}(t) = \frac{RD_{X_j}(t)^\theta}{\sum_{\text{all } k} RD_k(t)^\theta} \cdot T(t)$$

where:

- $RD_{X_j}(t) = \frac{W_{\text{avg}}(t) - W_{X_j}(t)}{W_{\text{avg}}(t)}$: Relative Deprivation Index
- $W_{\text{avg}}(t) = \frac{W_{\text{total}}(t)}{N}$: Average wealth of all agents
- $\theta \geq 1$: Redistribution sensitivity parameter

Explanation:

- Agents with lower wealth relative to the average receive higher redistribution.
- The parameter θ controls how strongly redistribution favors the most deprived agents.

Example: Assuming three agents:

- Agent 1: $W_1(t) = 100$
- Agent 2: $W_2(t) = 60$
- Agent 3: $W_3(t) = 20$
- $T(t) = 14.28 + 9.12 + 2.04 = 25.44$
- $W_{\text{avg}}(t) = \frac{100+60+20}{3} \approx 60$
- $RD_1(t) = \frac{60-100}{60} = -\frac{40}{60} \approx -0.6667$ (Not considered for redistribution)
- $RD_2(t) = \frac{60-60}{60} = 0$
- $RD_3(t) = \frac{60-20}{60} \approx 0.6667$
- $\theta = 2$

Only positive RD values are considered for redistribution:

$$D_3(t) = \frac{(0.6667)^2}{(0.6667)^2} \cdot 25.44 = 1 \cdot 25.44 = 25.44$$

Updated wealth:

- $W_1(t+1) = 100 + \Delta W_1(t) - 14.28$
- $W_2(t+1) = 60 + \Delta W_2(t) - 9.12$
- $W_3(t+1) = 20 + \Delta W_3(t) - 2.04 + 25.44 = 43.4 + \Delta W_3(t)$

6. Competence and Learning Mechanisms

Agents improve their competence based on ambition, inspiration, and by learning from the best-performing agents within the network.

6.1. Competence Update Equation

$$C_X(t+1) = C_X(t) + k_7 \cdot A_X(t) \cdot (C_{\max} - C_X(t)) + \kappa \cdot \left(\overline{C}_{\text{best}}(t) - C_X(t)\right)$$

where:

- k_7: Learning rate from self-improvement
- κ: Learning rate from best-performing agents
- $\overline{C}_{\text{best}}(t)$: Average competence of the best-performing agents
- $C_{\max}$: Maximum competence level

Example: Assuming $k_7 = 0.01$, $A_X(t) = 0.0045$, $C_{\max} = 100$, $C_X(t) = 20$, $\kappa = 0.1$, and $\overline{C}_{\text{best}}(t) = 80$:

$$C_X(t+1) = 20 + 0.01 \cdot 0.0045 \cdot (100 - 20) + 0.1 \cdot (80 - 20) \approx 20 + 0.0036 + 6 = 26.0036$$

6.2. Learning from Best-Performing Agents

Agents can enhance their competence by learning from peers with higher competence levels. This fosters a competitive yet collaborative environment.

Learning Mechanism:

$$C_X(t+1) = C_X(t) + k_7 \cdot A_X(t) \cdot (C_{\max} - C_X(t)) + \kappa \cdot \left(\overline{C}_{\text{best}}(t) - C_X(t)\right)$$

Example: With the same parameters as above, the agent's competence increases by both self-driven ambition and learning from peers.

7. Reward Function and Reinforcement Learning

The reward function guides agents in balancing personal wealth accumulation with community contributions and self-improvement. Reinforcement Learning (RL) is employed to adaptively adjust the weights $\alpha(t)$, $\beta(t)$, and $\gamma(t)$ to optimize cumulative rewards.

7.1. Reward Function

$$r_X(t) = \alpha(t) \cdot \Delta W_X(t) + \beta(t) \cdot C_X^{\text{comm}}(t) + \gamma(t) \cdot \Delta AS_X(t)$$

where:

- $\alpha(t)$, $\beta(t)$, $\gamma(t)$: Weights that sum to 1
- $\Delta W_X(t)$: Change in wealth
- $C_X^{\text{comm}}(t)$: Contribution to the community
- $\Delta AS_X(t)$: Change in agent status

7.2. Temporal Difference Error

$$\delta(t) = r_X(t) + \lambda \cdot P_X(t) - P_X(t-1)$$

where:

- λ: Smoothing parameter ($0 < \lambda < 1$)
- $P_X(t)$: Performance measure at time t

7.3. Weight Update Rules

$$\theta(t+1) = \theta(t) + \eta \cdot \delta(t) \cdot \frac{\partial r_X(t)}{\partial \theta}$$

where $\theta(t) \in \{\alpha(t), \beta(t), \gamma(t)\}$, and η is the learning rate.

Gradient Calculations:

$$\frac{\partial r_X(t)}{\partial \alpha(t)} = \Delta W_X(t)$$
$$\frac{\partial r_X(t)}{\partial \beta(t)} = C_X^{\text{comm}}(t)$$
$$\frac{\partial r_X(t)}{\partial \gamma(t)} = \Delta AS_X(t)$$

7.4. Normalization of Weights

After updating, normalize the weights to ensure they sum to 1:

$$\theta(t+1) = \frac{\theta(t+1)}{\alpha(t+1) + \beta(t+1) + \gamma(t+1)}$$

7.5. Example

Assuming:

- Initial weights: $\alpha(0) = 0.4$, $\beta(0) = 0.3$, $\gamma(0) = 0.3$
- Learning rate: $\eta = 0.05$
- Smoothing parameter: $\lambda = 0.9$
- At time $t = 1$:
  - $\Delta W_X(t) = 5$
  - $C_X^{\text{comm}}(t) = 2$
  - $\Delta AS_X(t) = 1$
  - $r_X(t) = 0.4 \cdot 5 + 0.3 \cdot 2 + 0.3 \cdot 1 = 2 + 0.6 + 0.3 = 2.9$
  - $P_X(t) = (1 - 0.9) \cdot 2.9 + 0.9 \cdot 2.9 = 2.9$

Compute the temporal difference error:

$$\delta(t) = 2.9 + 0.9 \cdot 2.9 - 2.9 = 2.61$$

Update weights:

$$\alpha(t+1) = 0.4 + 0.05 \cdot 2.61 \cdot 5 = 0.4 + 0.6525 = 1.0525$$
$$\beta(t+1) = 0.3 + 0.05 \cdot 2.61 \cdot 2 = 0.3 + 0.261 = 0.561$$
$$\gamma(t+1) = 0.3 + 0.05 \cdot 2.61 \cdot 1 = 0.3 + 0.1305 = 0.4305$$

Normalize weights:

$$\text{Total} = 1.0525 + 0.561 + 0.4305 = 2.044$$

$$\alpha(t+1) = \frac{1.0525}{2.044} \approx 0.515$$
$$\beta(t+1) = \frac{0.561}{2.044} \approx 0.274$$
$$\gamma(t+1) = \frac{0.4305}{2.044} \approx 0.211$$

The updated weights reflect a higher emphasis on individual utility due to its higher contribution to the reward.

8. Tax and Redistribution Mechanism

To model the agents' responsibility to contribute to the community, ASERSA introduces a dynamic tax and redistribution system based on wealth, agent status, and economic stability.

8.1. Tax Rate Calculation

The tax rate $\tau_X(t)$ for agent X at time t is defined as:

$$\tau_X(t) = \tau_{\max} \cdot \left(\omega_W \cdot \frac{W_X(t)}{W_{\text{total}}(t)} + \omega_{AS} \cdot \frac{AS_X(t)}{AS_{\text{opt}}} + \omega_E \cdot E(t) \right)$$

where:

- $\tau_{\max}$: Maximum tax rate
- ω_W, ω_{AS}, ω_E: Weights for wealth, agent status, and economic stability respectively, such that $\omega_W + \omega_{AS} + \omega_E = 1$
- $W_{\text{total}}(t) = \sum_X W_X(t)$: Total wealth of all agents at time t
- AS_{opt}: Optimal agent status level
- $E(t)$: Economic stability factor at time t (normalized between 0 and 1)

Explanation:

- **Wealth Factor**: Reflects the agent's share of the total wealth.
- **Agent Status Factor**: Higher status agents contribute more, reflecting their greater responsibility.
- **Economic Stability Factor**: Adjusts the tax rate based on overall economic conditions. During economic downturns ($E(t)$ increases), taxes may increase to support societal needs.

Example: Assuming:

- $\tau_{\max} = 0.4$
- $\omega_W = 0.5$, $\omega_{AS} = 0.3$, $\omega_E = 0.2$

- $W_X(t) = 70$, $W_{\text{total}}(t) = 100$
- $AS_X(t) = 40$, $AS_{\text{opt}} = 100$
- $E(t) = 0.2$

Compute $\tau_X(t)$:

$$\tau_X(t) = 0.4 \cdot \left(0.5 \cdot \frac{70}{100} + 0.3 \cdot \frac{40}{100} + 0.2 \cdot 0.2\right) = 0.4 \cdot (0.35 + 0.12 + 0.04) = 0.4 \cdot 0.51 = 0.204$$

8.2. Tax Collection

The tax collected from agent X is:

$$\text{Tax}_X(t) = \tau_X(t) \cdot W_X(t)$$

Example: With $\tau_X(t) = 0.204$ and $W_X(t) = 70$:

$$\text{Tax}_X(t) = 0.204 \times 70 = 14.28$$

8.3. Redistribution Mechanism

The total tax collected $T(t)$ is redistributed to agents based on their relative deprivation.

$$T(t) = \sum_X \text{Tax}_X(t)$$

Each agent X_j receives a redistribution amount $D_{X_j}(t)$ calculated as:

$$D_{X_j}(t) = \frac{RD_{X_j}(t)^\theta}{\sum_{\text{all } k} RD_k(t)^\theta} \cdot T(t)$$

where:

- $RD_{X_j}(t) = \frac{W_{\text{avg}}(t) - W_{X_j}(t)}{W_{\text{avg}}(t)}$: Relative Deprivation Index
- $W_{\text{avg}}(t) = \frac{W_{\text{total}}(t)}{N}$: Average wealth of all agents
- $\theta \geq 1$: Redistribution sensitivity parameter

Explanation:

- Agents with lower wealth relative to the average receive higher redistribution.
- The parameter θ controls how strongly redistribution favors the most deprived agents.

Example: Assuming three agents:

- Agent 1: $W_1(t) = 100$
- Agent 2: $W_2(t) = 60$

- Agent 3: $W_3(t) = 20$
- $T(t) = 14.28 + 9.12 + 2.04 = 25.44$
- $W_{\text{avg}}(t) = \frac{100+60+20}{3} \approx 60$
- $RD_1(t) = \frac{60-100}{60} = -\frac{40}{60} \approx -0.6667$ (Not considered for redistribution)
- $RD_2(t) = \frac{60-60}{60} = 0$
- $RD_3(t) = \frac{60-20}{60} \approx 0.6667$
- $\theta = 2$

Only positive RD values are considered for redistribution:

$$D_3(t) = \frac{(0.6667)^2}{(0.6667)^2} \cdot 25.44 = 1 \cdot 25.44 = 25.44$$

Updated Wealth:

- $W_1(t+1) = 100 + \Delta W_1(t) - 14.28$
- $W_2(t+1) = 60 + \Delta W_2(t) - 9.12$
- $W_3(t+1) = 20 + \Delta W_3(t) - 2.04 + 25.44 = 43.4 + \Delta W_3(t)$

9. Updating Competence and Learning

Agents improve their competence based on ambition, willpower, inspiration, and by learning from the best-performing agents within the network.

9.1. Competence Update Equation

$$C_X(t+1) = C_X(t) + k_7 \cdot A_X(t) \cdot (C_{\max} - C_X(t)) + \kappa \cdot \left(\overline{C}_{\text{best}}(t) - C_X(t)\right)$$

where:

- k_7: Learning rate from self-improvement
- κ: Learning rate from best-performing agents
- $\overline{C}_{\text{best}}(t)$: Average competence of the best-performing agents
- $C_{\max}$: Maximum competence level

Example: Assuming $k_7 = 0.01$, $A_X(t) = 0.0045$, $C_{\max} = 100$, $C_X(t) = 20$, $\kappa = 0.1$, and $\overline{C}_{\text{best}}(t) = 80$:

$$C_X(t+1) = 20 + 0.01 \cdot 0.0045 \cdot (100 - 20) + 0.1 \cdot (80 - 20) \approx 20 + 0.0036 + 6 = 26.0036$$

9.2. Learning from Best-Performing Agents

Agents enhance their competence by learning from peers with higher competence levels, fostering a competitive yet collaborative environment.

Learning Mechanism:

$$C_X(t+1) = C_X(t) + k_7 \cdot A_X(t) \cdot (C_{\max} - C_X(t)) + \kappa \cdot \left(\overline{C}_{\text{best}}(t) - C_X(t)\right)$$

Example: With the same parameters as above, the agent's competence increases by both self-driven ambition and learning from peers.

10. Network Effects in ASERSA

Network effects play a crucial role in ASERSA by influencing agents' competence and behavior through their interactions. The network structure determines how agents learn from each other and how influence propagates within the system.

10.1. Network Structure

- **Agent Network (G)**: A graph $G = (N, E)$ where N is the set of agents and E is the set of edges representing relationships between agents.
- **Edge Weights (w_{ij})**: Each edge e_{ij} has a weight w_{ij} indicating the strength and type of interaction between agent i and agent j.
  - $w_{ij} > 0$: Positive influence (cooperation, support).
  - $w_{ij} < 0$: Negative influence (competition, conflict).

10.2. Updating Competence with Network Effects

Agents update their competence not only based on their ambition and inspiration but also through interactions with their neighbors in the network.

$$C_X(t+1) = C_X(t) + k_7 \cdot A_X(t) \cdot (C_{\max} - C_X(t)) + \kappa \cdot \left(\overline{C}_{\text{net}}(t) - C_X(t)\right)$$

where:

- $\overline{C}_{\text{net}}(t)$: The weighted average competence of agent X's neighbors.

$$\overline{C}_{\text{net}}(t) = \frac{\sum_{j \in N_X} w_{Xj} \cdot C_j(t)}{\sum_{j \in N_X} |w_{Xj}|}$$

- N_X: The set of neighbors of agent X.
- w_{Xj}: The weight of the connection between agent X and neighbor j.

Example:

Suppose agent X has two neighbors, agent Y and agent Z, with the following competences and weights:

- $C_Y(t) = 70$, $w_{XY} = 0.8$ (positive relation)
- $C_Z(t) = 50$, $w_{XZ} = -0.5$ (negative relation)

Compute $\overline{C}_{\text{net}}(t)$:

$$\overline{C}_{\text{net}}(t) = \frac{0.8 \times 70 + (-0.5) \times 50}{0.8 + 0.5} = \frac{56 - 25}{1.3} = \frac{31}{1.3} \approx 23.85$$

Update competence for agent X:

$$C_X(t+1) = C_X(t) + 0.01 \cdot A_X(t) \cdot (100 - C_X(t)) + 0.1 \cdot (23.85 - C_X(t))$$

Assuming $C_X(t) = 40$, $A_X(t) = 9.87$:

$$C_X(t+1) = 40 + 0.01 \cdot 9.87 \cdot 60 + 0.1 \cdot (23.85 - 40) = 40 + 5.922 + (-1.615) \approx 44.307$$

10.3. Impact of Network Topology

The structure of the network significantly influences the dynamics of competence and influence:

- **Scale-Free Networks**: Promote the emergence of highly influential agents (hubs) who can significantly impact others.
- **Small-World Networks**: Facilitate rapid information and influence spread, enhancing collective learning.
- **Random Networks**: Offer uniform interaction opportunities, preventing the dominance of specific agents.

11. Overall Agent Update Rule

To integrate all the components, we establish a comprehensive update rule for each agent X at each time step t:

1. **Update Wealth:**

$$W_X(t+1) = W_X(t) + \Delta W_X(t) - \tau_X(t) \cdot W_X(t) + D_X(t)$$

2. **Update Influence:**

$$I_X(t+1) = I_{\max} \cdot \frac{1}{1 + e^{-k_1(W_X(t+1) - W_0)}}$$

3. **Update Agent Status:**

$$AS_X(t+1) = k_2 \cdot I_X(t+1)^{\alpha}$$

4. **Update Responsibility:**

$$R_X(t+1) = \frac{AF_X(t+1)}{SF_X(t+1)} \cdot R_{\text{opt}}$$

5. **Update Self-Esteem:**

$$S_X(t+1) = \left(\frac{AS_X(t+1)}{SS_X(t+1) + AS_X(t+1)} \right)^2 \cdot S_{\text{opt}}$$

6. **Update Willpower:**

$$V_X(t+1) = V_{\max} \cdot \frac{1}{1 + e^{-k_5(S_X(t+1) - S_0)}}$$

7. **Update Ambition:**

$$A_X(t+1) = k_6 \cdot \left(1 - e^{-\frac{IN_X(t+1)}{R_X(t+1)}} \right)$$

8. **Update Inspiration:**

$$IN_X(t+1) = \begin{cases} \sqrt{\phi \cdot \frac{AI_X(t+1)}{SI_X(t+1)}} & \text{if } \phi \cdot \frac{AI_X(t+1)}{SI_X(t+1)} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

9. **Update Action Level:**

$$AL_X(t+1) = \psi \cdot \left((C_X(t) \cdot V_X(t) \cdot A_X(t))^{\frac{1}{3}} \right) \cdot \left(1 - e^{-\left((C_X(t) \cdot V_X(t) \cdot A_X(t))^{\frac{1}{3}} \right)} \right)$$

10. **Update Competence:**

$$C_X(t+1) = C_X(t) + k_7 \cdot A_X(t) \cdot (C_{\max} - C_X(t)) + \kappa \cdot \left(\overline{C}_{\text{best}}(t) - C_X(t) \right)$$

11. **Calculate Reward:**

$$r_X(t) = \alpha(t) \cdot \Delta W_X(t) + \beta(t) \cdot C_X^{\text{comm}}(t) + \gamma(t) \cdot \Delta AS_X(t)$$

12. **Update Weights:**

$$\theta(t+1) = \theta(t) + \eta \cdot \delta(t) \cdot \frac{\partial r_X(t)}{\partial \theta}, \quad \theta \in \{\alpha, \beta, \gamma\}$$
$$\theta(t+1) = \frac{\theta(t+1)}{\alpha(t+1) + \beta(t+1) + \gamma(t+1)}$$

12. Tax and Redistribution Policies

ASERSA supports multiple taxation and redistribution policies to explore their impacts on agent dynamics and socio-economic outcomes.

12.1. Flat Tax Policy

A uniform tax rate applied equally across all agents and token types.

12.2. Universal Basic Income (UBI) Policy

Taxes are collected uniformly and redistributed equally among all agents, regardless of their wealth or status.

12.3. Progressive Tax Policy

Tax rates vary based on agents' wealth and status, favoring redistribution to less wealthy or lower-status agents.

12.4. Applying Tax Policies

The tax policy is applied as follows:

$$\text{Tax}_X(t) = \begin{cases} \tau_{\text{flat}} \cdot W_X(t) & \text{if policy is Flat Tax} \\ \tau_X(t) \cdot W_X(t) & \text{if policy is Progressive Tax} \\ \text{Calculated based on UBI rules} & \text{if policy is UBI} \end{cases}$$

After tax collection, redistribution mechanisms allocate $T(t)$ based on the selected policy.

13. Concluding Remarks

By integrating socio-economic variables, non-linear relationships, adaptive learning models, and dynamic tax and redistribution policies into the dynamics of the agents, ASERSA creates a system where agents not only focus on their individual goals but also adapt their behaviors based on their experiences and interactions with others.

The adaptive learning model allows agents to:

- Dynamically adjust their focus between personal gain, community contribution, and self-improvement.
- Learn from their environment and experiences, leading to more sophisticated decision-making processes.
- Contribute to a more complex and realistic simulation of socio-economic behaviors in an agent-based model.

This enriched model can be used to study various phenomena in socio-economic systems, such as wealth distribution, social mobility, and the impact of policies on individual and collective behaviors.