Transdisciplinarity and the Future of Engineering
B.R. Moser et al. (Eds.)

doi:10.3233/ATDE220663

Preliminary Evolutionary Network Model for Efficient Collaboration in Systems-of-Systems

Myron BOYD[1], Thomas MAZZUCHI, and Shahram SARKANI
The George Washington University, Department of Engineering Management and Systems Engineering, Washington, USA

Abstract. Collaborative systems-of-systems (CSoSs) are defined by the Systems Engineering Body of Knowledge as groups of constituent systems that voluntarily work with each other toward a common goal. As the complexity, sociotechnical interactions, and cooperation of real systems increases, so too does our need to understand how to design and manage collaboration across disciplines. An agent-based model is developed that combines network evolution mechanisms with evolutionary game theory to simulate CSoSs. Collaboration efficiency (CE) is introduced as a metric by which collaboration may be measured and performance compared. Cost and strategy parameters of constituent systems are tested via CSoS model simulation to develop insights into best collaboration practices for CSoSs. Results suggest a reactive collaboration strategy or a reinforcement algorithm-based strategy produce the highest CE under certain conditions. Applicable to systems in sociotechnical enterprises, logistics, energy, infrastructure, and more, this research can improve the design and operation of any CSoS.

Keywords. System-of-Systems, Collaboration, Game Theory, Network Evolution, Agent-Based Model

Introduction

Systems across industries are growing larger and more complex. When groups of systems work together, a system-of-systems (SoS) is created. The SoS demonstrates new properties that emerge from the constituent subsystems interacting with each other [1]. However, not all SoSs are the same. Systems engineers often classify SoSs into four major categories depending on characteristics of the SoS: directed, acknowledged, collaborative, and virtual [2]. Various SoSs seen in industry can be described by one of these four types [3]. Collaborative SoSs (CSoSs) are increasingly important in areas such as energy, logistics, business organization, and policy, as complex decisions are made by both humans and machines. In CSoSs, independent constituent systems that often span across industries and disciplines, collaborate by working togther and sharing resources for the success of the SoS without a centralized authority directing operations [4], [5]. Collaboration is an intuitive component of human-based systems and social research, but it is less obvious for entirely technical systems and research. Nonetheless,

[1] Corresponding Author, Mail: myronsboyd@gwmail.gwu.edu.

computer and data-driven design and management is already being employed in engineered systems [3], [5]. A better understanding of the collaborative nature of CSoSs may yield insight into better engineering and operation of constituent systems.

Thus far, research into understanding CSoSs has developed several approaches [6]-[8]. Game theory has been used to represent collaborative behavior. Network and graph theory has been used to describe the structure of SoSs, illustrating relationships among constituents as parts of a whole [9]. Agent-based models (ABM) are often used to show SoS-level behavior and emergence [10]. Sill more recently, evolutionary game theory and network evolution have been used to show system dynamics and decision-making over time, which joins neatly with agent-based model simulations [11]-[13].

However, strategies available to constituent systems to improve collaboration in a CSoS are not readily available, nor even a metric by which collaboration in a complex system can be measured. A gauge of collaboration, and by extension, CSoS effectiveness, is developed in this research. The metric of *collaboration efficiency* is introduced and demonstrated using an ABM of CSoSs. This metric is used as a performance indicator of collaboration and tests of different parameters embodied in the constituent systems of a CSoS are performed. Best strategies under certain conditions are indentified. Future work will expand on the parameters tested to suggest best collaboration practices to follow when entities participate in CSoSs.

1. Background

1.1. Collaboration by Evolutionary Games

Game theory is a mathematical approach to describing interactions among individual agents. Interactions are modeled as stylized games with rules that describe a certain situation of interest. One class of interactions involves collaborative games (*cooperative games* in game theory terminology), where there is an option for two agents to either cooperate with each other or not. Exploring the nature of cooperation, Axelrod developed the iterated Prisoner's Dilemma tournament which gave insight into the best strategies to play that game [14]. Iterating a collaborative game contributed to what is now evolutionary game theory, which captures changes in the choices made by each player in a game, known as their *strategies*.

The dichotomy of collaboration or defection (not collaborating) has led to two primary areas of study in SoSs. The first uses custom games to validate characteristics that define a collaborative SoS [15], [16]. However, the use of custom games requires more empirical validation [17]. The second use of game theory in SoSs involves evolving games in concert with network theory to demonstrate how structure affects and is affected by collaboration. Studies in this area have identified strategies and network parameters that affect the evolution of the system [18], [19]. This group of research has typically used the Prisoner's Dilemma game as a canonical example of the collaborative game. However, while the Prisoner's Dilemma provides demonstrates why rational actors may not collaborate while acting in their own interest, the Stag Hunt is another stylized collaboration game that demonstrates a trade-off in social trust, that is, the rational choice to collaborate or not depends on what the other player (or players) is believed will do [20]. The Prisoner's Dilemma has one Nash equilibrium when both players defect. The Stag Hunt has two pure strategy Nash equilibria, one where both players defect, and another where both players cooperate [19]. As such, the socially collaborative game is considered more appropriate for studying collaborative behavior in a group of systems in this study. A general payoff matrix for a two-player collaborative game is shown in Table 1 using the standard Reward, Punishment, Sucker, Temptation (R, P, S, T) format [14]. The first letter in each outcome is the Player 1 choice and the second letter represents the Player 2 choice. The Reward payoff occurs when both players collaborate, the Punishment payoff occurs when both players defect, the Sucker payoff is received by a player that collaborates when the other defects, and the Temptation payoff is received by a player that defects when the other collaborates. The order of magnitudes for each payoff defines the type of collaborative game. The order for Stag Hunts is R > T ≥ P > S.

Table 1. Generalized collaborative game payoff matrix in R, P, S, T format.

		Player 2	
		Collaborate	*Defect*
Player 1	*Collaborate*	R, R	S, T
	Defect	T, S	P, P

1.2. SoS Network Structure

CSoSs are often difficult to clearly visualize because different constituent systems must have relationships with many other systems and have properties that define both the systems themselves and the links between them. Each constituent may be different from the others and may follow different rules and values, all of which must be captured in a SoS model. Networks have been proposed as a quantitative representation of SoSs [18], [21]. CSoSs are suited to network representation as each constituent has some level of autonomy and decision-making which shapes the structure of the network [9].

While an SoS may be defined once by a static network, *behavior* of the whole SoS may be observed in evolving networks [21]. The actions and rules of individual constituents are the building blocks that produce emergent properties, such as collaboration, at the network level. Thus, network theory can be leveraged to analyze and understand CSoSs.

Limited research has considered the network-wide effects of evolving collaboration among individual nodes. Hierarchies and network shapes have been identified under certain collaborative conditions, as well as distinct groupings of constituents that tend to act similarly [13], [21]. However, there does not appear to be accepted metrics by which collaboration in networks can be measured, compared, or predicted.

1.3. Agent-Based Modeling

Collaborative games can represent decision-making on the constituent system level and networks can represent the CSoS as a whole. But these theories will fail to capture the emergence of collaboration over time without being evolutionary. The literature supports using ABM and simulation to execute the approaches described previously [10], [18]. ABM has its origins in social choice mechanisms with Schelling's segregation model [22]. It has since been applied to a wide variety of complex behaviors in systems and system-like settings [17], [21]. The CSoS fits the description of an ABM as having a variety of constituent systems, represented as the agents, which interact with each other according to rules and requirements defined at the constituent level, represented as agent behavior. Allowing the constituent system agents to voluntarily interact with each other and observing SoS-wide effects is achieved through ABM simulation. ABM can be computationally expensive to implement on large scales but is an appropriate method of quantitatively describing complex systems and emergent behaviors [23].

2. Methodology

Predicting collaboration in a CSoS and testing constituent system parameters depends on a model of an SoS that provides the required quantities of collaborations and interactions. The primary components of the proposed model are collaboration modeled by evolutionary game theory, SoS structure modeled by network theory, and prediction by agent-based model simulation. The model is informed by a set of defined inputs that are drawn from a few basic parameters of a generalized CSoS.

1. Cost factor: value that defines the cost of an interaction.
2. Payoff matrix: defines the stylized game played, which follows the Stag Hunt game in this study, but any other two-player cooperative game payoffs may be used.
3. Number of time steps: sets the length of the simulation.

With these inputs, the collaborative behavior of the SoS may be simulated and collaboration efficiency calculated.

2.1. Collaboration Efficiency Metric

For CSoSs to maximize effectiveness, collaboration among constituents is a principal function. While prior work has aimed to demonstrate how collaboration functions within an SoS, this research builds on previous theory to provide a useful metric by which SoS collaboration can be measured, herein named *collaboration efficiency* (CE). The CE metric attempts to represent the level of effective collaboration between constituent systems among all opportunities for two constituents to collaborate. When two constituent systems interact with each other in a CSoS, they have the opportunity to collaborate with each other or not. CE is defined as the number of cases when both constituents collaborate with each other, divided by the total number of interactions that occur between two constituents, as shown in Equation 1. This metric fills a need to provide insight into the collaborative nature of an SoS and allows comparison among systems and configurations to support stakeholder decision-making [3].

$$CE = \frac{Number\ of\ fully\ collaborative\ interactions}{All\ interactions\ that\ occur} \tag{1}$$

While the CE is proposed as a measure of collaboration in a CSoS, its use may not be immediately obvious for constituent systems. By definition, CSoSs do not have a central authority over the whole. Each constituent system participates voluntarily towards a group goal. However, constituent systems must balance personal gain against the performance of the CSoS. Collaboration often involves a risk that the return on collaboration is not worth the effort. Therefore, constituent systems must make decisions to collaborate or not with other constituents, often with limited knowledge of others' decisions.

2.2. Stag Hunt Game

The games implemented in the agent interactions of this model are Stag Hunts to keep the focus on collaboration rather than individual gain, as suggested by the literature [13]. A key component of this model is the strategy employed by each constituent player in each game interaction. There have been many strategies proposed for playing collaborative games [18]. For this study, a selection of different strategy types is assessed. The strategies chosen for testing are listed, categorized, described, and interpreted in Table 2 [24], [25].

Table 2. Collaboration strategy list and interpretations.

Category	Categorical Description	Game Theory Strategy	Interpretation	Strategy Description
Pure	Simple, deterministic strategies defined by a mission.	*Always Collaborate*	Pure altruism	Collaborate no matter what.
		Always Defect	Pure self-interest	Do not collaborate (defect).
		Alternator	Defined mix of altruism and self-interest	Collaborate then defect every other turn.
Stochastic	Simple strategies informed by a specific probability.	*Random*	Random chance	Randomly choose to collaborate with probability 0.5.
Reactive	Strategies that attempt to balance collaboration and competition by taking advantage of opportunities when they appear. These strategies have a memory of one previous game.	*Tit for Tat*	Trusting copycat	Collaborate, then repeat the opponent's move from the last round.
		Suspicious Tit for Tat	Distrustful copycat	Defect, then repeat the opponent's move from the last round.
		Pavlov	Win stay, lose shift	Collaborate first, then collaborate if last round payoff was R or T, and defect otherwise.
Adaptive	Adaptation attempts to lengthen the memory of reactive strategies and include all previous games. An intuition is built out of the past experience.	*Adaptor*	Experienced intuition	Every past game played adjusts an internal state according to a simple function. The internal state defines the probability of collaboration.
Learning	A stochastic behavior function is developed based on feedback provided from past games using learning algorithms.	*Q Learner*	Blind reinforcement	A simple, model-free reinforcement learning algorithm takes the payoffs of every previous game and attempts to choose collaboration or defection to improve payoff.

The scope of strategies to test was developed based on literature for the best performing and most descriptive strategies already researched by the evolutionary game theory community. Pure strategies are not necessarily descriptive of real systems but are useful for understanding and comparison purposes. Similarly, the *Random* strategy is not necessarily useful in life, but can represent unknown strategies from one player's point of view. The *Random* strategy is treated as the baseline comparison for other strategies in this research.

The reactive, adaptive, and learning strategies are all increasingly nuanced representations of actual behaviors observed in the real world or that have performed the best in terms of payoff in previous research [24]. It should be noted that the vast majority of evolutionary game strategy research has looked at the Prisoner's Dilemma game. The Stag Hunt game is used in this research as a more appropriate approximator of collaborative interactions [13]. However, little research exists to suggest a best strategy for this game type, so the available library of strategies is carried over and investigated herein.

2.3. Network Representation

A network representation of the SoS arises out of the interactions between each modeled constituent. Each constituent is considered to interact with each other one at a time. That is, a network *edge* links exactly two constituent nodes. An interaction in this model is represented by a network edge. Thus, for each constituent, multiple games may have to be played to account for all interactions that exist at a given point time.

Not all constituents are necessarily always linked to each other. To capture evolution of the network structure, constituents will initially connect and interact with each other by random chance. Thereafter, interactions will be maintained for another game if and only if the payoff of each constituent from the first game is greater than the cost to make the edge. This mechanism was designed to follow the natural cost-benefit consideration that a rational agent voluntarily participating in the CSoS might use. New interactions after initialization are formed by random chance only if two constituents have not interacted for five consecutive time steps to allow new interactions to periodically form over time. In the case of a real CSoS, changes in sentiment, leadership, marketplace, environmental factors, and more can influence the decision to try new interactions. The function that defines new starting new interactions may be an area of study for future research.

2.4. Model Simulation

A variety of different tools exist for constructing and simulating ABMs, but the collaborative SoS model presented is developed directly in the Python programming language. Python is heavily used in scientific and data analysis for its high-level readability and extensibility. To streamline the development process, the Axelrod and NetworkX code libraries are used to run games in each interaction and represent the model as a network, respectively [25], [26]. Both libraries have been supported and validated with previous research.

In each simulation run, a comma separated values file is read to memory to define agents, and initial parameters are defined in code, including the number of simulation time steps, cost factor, and whether a seed is to be used for replicable simulations. Then a data table of placeholders for each possible interaction between every agent is initialized. A loop fills the data table by calculating whether an interaction exists for each combination and determines results of games played for the valid interactions during each time step. The interaction configuration of the network is updated each time step depending on whether the results of each game played were higher than the cost to play. When the simulation loop has processed all time steps, the number of true collaborations, i.e., all games where both constituent agents chose to collaborate, is counted and divided by the total number of valid interactions to yield the collaboration efficiency. One collaboration efficiency value is a result of each simulation run.

The results presented were calculated from simulations run on a consumer desktop computer. Each run lasted an average of 50.38 ± 2.02 seconds running in a single-threaded Python environment on a multicore processor at a clock speed of 4.6 GHz.

3. Initial Results

The test CSoS consists of five constituent systems with a basic set of parameters. The Stag Hunt payoff matrix used is given in Table 3. The remaining constituents were given the *Random* strategy. The baseline *Random* strategy is used to represent an unknown strategy from the point of view of the test agent. Because a CSoS has little to no central control, a real world constituent system would likely not have omniscience of the collaboration strategies of other constituents. The *Random* strategy represents this limited knowledge.

Table 3. Stag Hunt payoff matrix.

		Constituent 2	
		Collaborate	*Defect*
Constituent 1	*Collaborate*	5, 5	1, 3
	Defect	3, 1	3, 3

Simulating the CSoS ABM produces a few results of interest. Each run of 100 time steps results in one possible CSoS. The CE of each interaction is displayed so that constituent-level effects can be observed. The CSoS result from one test run is given in Fig. 2. The *Pavlov* test case is shown, whereby agent A follows the *Pavlov* strategy, and the remaining agents are *Random*. The thickness of each line corresponds to the CE value printed on the line. The CSoS average CE for this single test simulation is 28.1% even though the CE of individual interactions varies.

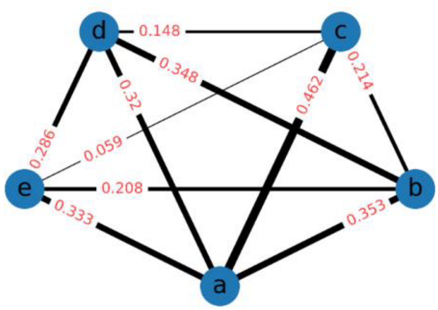

Figure 1. Network model of five-constituent CSoS.

A benefit of using the *Random* strategy for the baseline is allowing a basic verification of the model simulation by testing a model with all *Random* strategies. The expected collaboration efficiency from this case is ¼, or 0.25, because there is one result where both constituent collaborate out of the four possible outcomes of each game. If two constituents have a 0.50 probability to choose to collaborate, then the chance they will both collaborate is $(0.5)^2$ or 0.25. The simulation of the five *Random* constituents does produce this result, as seen in Fig. 2, where the CSoS CE is plotted over each time step. Six 100-step tests are overlaid to demonstrate that the simulations do converge.

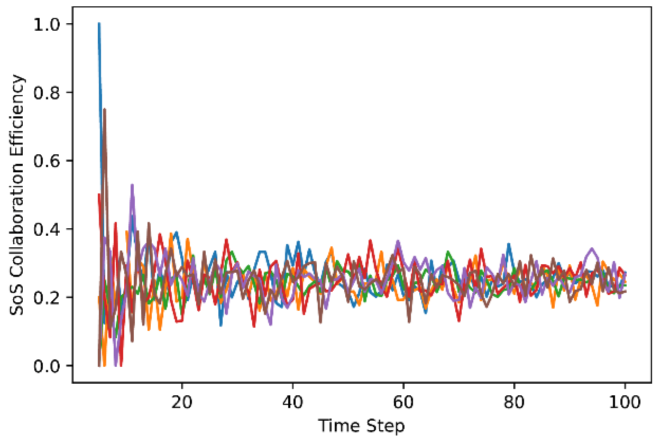

Figure 2. All-*Random* constituent systems converges to an expected CE of 0.25.

The full test results of all collaboration strategies listed earlier are presented in Table 4. Each strategy is also tested at a different cost value. The range of cost values tested was chosen to fall between the payoffs of each possible outcome of the Stag Hunt games. Testing was performed by varying the collaboration strategy of one constituent among the five, simulating the network of constituents for 100 steps, and repeating the simulation 50 times to observe if the CE of the CSoS appears to consistently converge to a specific value. All test cases converged to an average CE over time. The results are plotted in Fig. 3 for easier visualization. The 95% confidence limit for each test case based on 50 samples is shown with error bars. The 0.25 CE all-*Random* baseline is drawn over the results to show which collaboration strategies performed better and worse.

Table 4. Collaboration efficiency test results.

Strategy Category	Test Strategy	CSoS CE	95% CI
Pure	*Always Collaborate*	0.349	± 0.0016
	Always Defect	0.150	± 0.0014
	Alternator	0.251	± 0.0013
Stochastic	*Random*	0.249	± 0.0016
Reactive	*Tit for Tat*	0.251	± 0.0020
	Suspicious Tit for Tat	0.247	± 0.0017
	Pavlov	0.296	± 0.0014
Adaptive	*Adaptor*	0.215	± 0.0020
Learning	*Q Learner*	0.340	± 0.0017

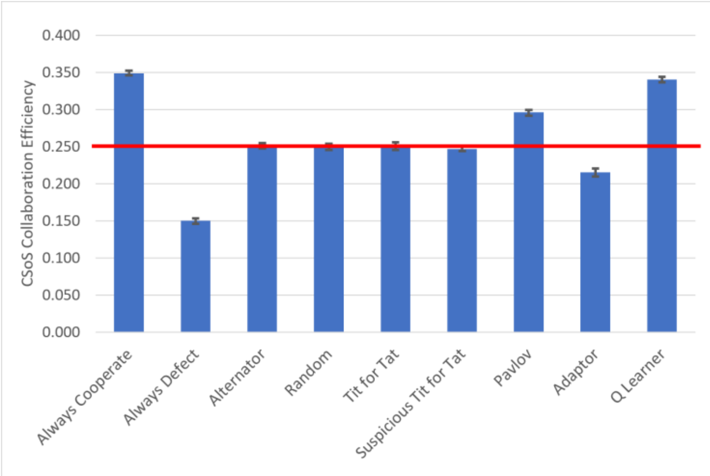

Figure 3. Five-constituent CSoS collaboration strategy results compared to *Random* baseline at 0.25 CE.

4. Conclusions and Future Work

This research so far establishes a metric to measure collaboration in a CSoS and develops a simulation to test constituent system parameters against it. An ABM constructs a CSoS with a network structure and set of input parameters, simulates collaboration decisions among constituent systems, and reports CSoS collaboration efficiency. The simulation model produces networks and CEs that are expected for simple constituent configurations. A constituent that always chooses to collaborate produces the highest CE, which is an expected but trivial solution. The *Pavlov* and *Q Learner* appear to be the most effective, non-trivial collaboration strategies for the tested set of parameters. The remaining test strategies are either worse than a random baseline or are not significantly different from it. These results suggest a decision-making heuristic that a constituent system may use to improve CSoS effectiveness. If a constituent participates in a small CSoS with a finite number of other constituents, has competitive costs to interact with other constituents, and will receive better returns for collaboration than not, then the constituent might choose to follow the prescribed *Pavlov* strategy for collaboration decisions or develop a learning algorithim such as a *Q Learner* model over time to inform their decisions.

However, the CSoS model presented has limitations. The cost function per interaction is not sophisticated and depends on a single input parameter. Competition among constituents is not addressed. Additional attributes may also be used to define environmental pressures, such as political, social, or technological, on realistic constituent operations and decision-making. Future work will attempt to address each of these shortcomings by building additional functionality into the ABM and testing a larger set of CSoS parameters. As with any simulation model, there remains a need to validate the results against real-world system dynamics. Regardless, both the collaboration efficiency metric presented, and collaboration strategies identified, may be useful to constituent systems in CSoSs and may encourage collaboration analysis in real systems across industry and business.

5. References

[1] M. Henshaw, J. Dahmann, and B. Lawson, 2009, *Systems of Systems (SoS) – SEBoK*, https://www.sebokwiki.org/wiki/Systems_of_Systems_(SoS), accessed: 28.02.2022.
[2] M.W. Maier, Architecting principles for systems-of-systems, *Systems Engineering*, Vol. 1, No. 4, 1998, pp. 267–284.
[3] B. Collins, S. Doskey, and J. Moreland, Modeling the Convergence of Collaborative Systems of Systems: A Quantitative Case Study, *Systems Engineering*, Vol. 20, No. 4, 2017, pp. 357–378.
[4] J. Axelsson, Game theory applications in systems-of-systems engineering: A literature review and synthesis, *Procedia Computer Science*, Vol. 153, 2019, pp. 154–165.
[5] P. Svenson, F. Reichenberg, and J. Axelsson, A design method for collaborative systems of systems applied to Metropolitan Multi-Mode Transport System, In: *16th International System of Systems Engineering Conference (SoSE)*, Vasteras, Sweden, 2021, DOI: 10.1109/SOSE52739.2021.9497488.
[6] W.C. Baldwin, *Modeling a Decision Mechanism for the Formation of Collaborative System of Systems*, PhD Dissertation, Stevens Institute of Technology, 2010.
[7] H.R. Darabi, and M. Mansouri, The Role of Competition and Collaboration in Influencing the Level of Autonomy and Belonging in System of Systems, *IEEE Systems Journal*, Vol. 7, No. 4, 2013, pp. 520–527.
[8] M.J. DiMario, J.T. Boardman, and B.J. Sauser, System of Systems Collaborative Formation, *IEEE Systems Journal*, Vol. 3, No. 3, 2009, pp. 360–368.
[9] C. Fan, and A. Mostafavi, Metanetwork Framework for Performance Analysis of Disaster Management System-of-Systems, *IEEE Systems Journal*, Vol. 14, No. 1, 2020, pp. 1265–1276.
[10] W.C. Baldwin, J.T. Boardman, and B.J. Sauser, Expanding a System of Systems Model with the Schelling Segregation Model, *Systems Research and Behavioral Science*, Vol. 30, No. 1, 2012, pp. 65–75.
[11] J.M. Pacheco, A. Traulsen, and M.A. Nowak, Coevolution of Strategy and Structure in Complex Networks with Dynamical Linking, *Physical Review Letters*, Vol. 97, No. 25, 2006.
[12] B. Wu, D. Zhou, F. Fu, Q. Luo, L. Wang, and A. Traulsen, Evolution of Cooperation on Stochastic Dynamical Networks, *PLoS ONE*, Vol. 5, No. 6, 2010, pp. e11187.
[13] M. Starnini, A. Sánchez, J. Poncela, and Y. Moreno, Coordination and growth: the Stag Hunt game on evolutionary networks, *Journal of Statistical Mechanics: Theory and Experiment*, Vol. 2011, No. 05, 2011, pp. P05008.
[14] R. Axelrod, *The Evolution of Cooperation*, Basic Books, New York, 1984.
[15] W.C. Baldwin, T. Ben-Zvi, and B.J. Sauser, Formation of Collaborative System of Systems Through Belonging Choice Mechanisms, *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, Vol. 42, No. 4, 2012, pp. 793–801.
[16] G. Ye, and B. Ge, System-of-Systems Design Optimization Based on Coevolutionary Game, In: *5th International Conference on Electromechanical Control Technology and Transportation (ICECTT)*, Nanchang, China, 2020, pp. 391–398.
[17] W.C. Baldwin, B.J. Sauser, and J. Boardman, Revisiting "The Meaning of Of" as a Theory for Collaborative System of Systems, *IEEE Systems Journal*, Vol. 11, No. 4, 2015, pp. 1–12.
[18] D.A. Gianetto, and B. Heydari, Catalysts of Cooperation in System of Systems: The Role of Diversity and Network Structure, *IEEE Systems Journal*, Vol. 9, No. 1, 2015, pp. 303–311.
[19] P.T. Grogan, K. Ho, A. Golkar, and O.L. de Weck, Multi-Actor Value Modeling for Federated Systems, *IEEE Systems Journal*, Vol. 12, No. 2, 2018, pp. 1193–1202.
[20] B. Skyrms, *The Stag Hunt*, Proceedings and Addresses of the American Philosophical Association 2001, Vol. 75 (2), pp.31 – 41.
[21] B. Heydari, and K. Dalili, Emergence of Modularity in System of Systems: Complex Networks in Heterogeneous Environments, *IEEE Systems Journal*, Vol. 9, No. 1, 2015, pp. 223–231.
[22] T.C. Schelling, Dynamic models of segregation, *The Journal of Mathematical Sociology*, Vol. 1, No. 2, 1971, pp. 143–186.
[23] L. John, G.B. John, M. Parker, B.J. Sauser, and J.P. Wade, Self-Organizing Cooperative Dynamics in Government Extended Enterprises, *IEEE Systems Journal*, Vol. 12, No. 3, 2018, pp. 2905–2916.
[24] C. Hauert, and O. Stenull, Simple Adaptive Strategy Wins the Prisoner's Dilemma, *Journal of Theoretical Biology*, Vol. 218, No. 3, 2002, pp. 261–72.
[25] V. Knight, 2015, *Axelrod 4.12.0 documentation*, https://axelrod.readthedocs.io/en/stable/index.html, accessed: 28.02.2022.
[26] Hagberg, D. Schult, and P. Swart, Exploring Network Structure, Dynamics, and Function using NetworkX, In: *Proceedings of the 7th Python in Science Conference (SciPy 2008)*, Pasadena, CA, 2008, pp. 11–15.