Hierarchical Self-Organized Learning in Agent-Based Modeling of the MAPK Signaling Pathway

Abbas Sarraf Shirazi, Sebastian von Mammen and Christian Jacob[†]
Deptartment of Computer Science, Faculty of Science
[†]Deptartment of Biochemistry & Molecular Biology, Faculty of Medicine
University of Calgary, Canada

***Abstract*—In this paper, we present a self-organized approach to automatically identify and create hierarchies of cooperative agents. Once a group of cooperative agents is found, a higher-order agent is created which in turn learns the group behaviour. This way, the number of agents and thus the complexity of the multiagent system will be reduced, as one agent emulates the behaviour of several agents. Our proposed method of creating hierarchies captures the dynamics of a multiagent system by adaptively creating and breaking down hierarchies of agents as the simulation proceeds. Experimental results on two MAPK signaling pathways suggest that the proposed approach is suitable in stable systems while periodic systems still need further investigations.**

I. Introduction

Computer simulation is an important means to investigate the properties of biological models. Simulations based on numerical methods reveal the change of an entity in the system over time or other measures. Unlike numerical methods, cellular automata simulations consider spatial location of entities in the model. In agent-based modeling (ABM), each entity is considered to be an agent in a multiagent system (MAS) that interacts locally with its neighbours as well as the environment [1], [2]. The ABM can also take geometrical properties like shape into account. Furthermore, the result of local interactions among agents often leads to the emergence of collective behaviours which is not reflected in the behaviour of individual agents. The human body is a good example of such a phenomenon. Starting inside a cell, there is DNA defining the behaviour of a cell. As the scale increases, cells work together to form tissues, tissues form organs, and organs drive the human body.

A signaling pathway is a cascade of biochemical reactions which results in changing the concentration of substrates inside a cell [3]. The MAPK signaling pathway is one of the most documented pathways in the literature in which an extra stimulus regulates cellular activities, such as gene expression, mitosis, etc [4]. A typical agent-based model of the MAPK signaling pathway might consist of substrates as agents. The result of agents interactions is the change in their concentrations. Thus, the change of concentration of a substrate over time could be considered as an emergent property of such a system.

In this work, we present a multiagent approach to model the MAPK signaling pathway. In our proposed method, a higher-order agent learns and subsumes the behaviour of a group of agents that exhibit similar behaviours ("adoption"). Since the dynamics of the system might change over the course of a simulation, the learned behaviour might not be valid any more after some time. In this case, the learned hierarchy must collapse and the individual agents must be again released to the system ("release"). The process of adoption and release happens recursively yielding dynamic hierarchical structures. How and when to build hierarchies, how to learn the behaviour of lower-level agents, and how to monitor the validity of higher-order agents are issues that we aim to address in this research.

The remainder of this paper is organized as follows. Section 2 reviews related work in the field of multiscale and multiagent modeling of biological systems. Section 3 presents the details of our proposed method. Section 4 reports on the experiments conducted to demonstrate the performance of the proposed method. Finally, concluding remarks are presented in Section 5.

II. Related Work

Von Mammen *et al.* [5] conceptualize the idea of self-organized middle-out abstraction in multiagent systems. In particular, they identify different actions required by an *observing* agent to build hierarchies of agents. They conjecture that motif detection algorithms in time series analysis will be useful in finding recurrent patterns to be abstracted and learned by higher-order agents.

Amigoni and Schiaffonati [3] review three multiagent approaches to model the MAPK signaling pathway. The first approach [6], models each intracellular component as an agent using a blackboard mechanism to interact with other agents. In the second approach [4], agents are the biochemical reactions while the approach proposed in [7] defines an agent to be a molecular entity communicating to other agents through biochemical reactions. It is also noted that although multiagent modeling of biological systems allows to associate each biological entity to an agent type, the lack of experimental data makes it hard to validate such systems [3].

A modularization approach for the MAPK signaling pathway is presented in [8]. It is based on the graph theory in which a node with the maximum number of neighbours in the biological interaction network is found first. Further expansion of this node into a subgraph results in a module. This approach has two drawbacks: (1) it assumes that the graph of the network is known beforehand, and (2) the analysis is static.

Recently, with the advances in technology, there is more focus on multiscale modeling in the literature. In [9], different scales for modeling the human body (cell, tissue, organ) and their link to real-world applications like tumor growth therapy is explained. In [10], a software framework for multiscale model integration and simulation is proposed; however, no specific modeling technique is described. Lavelle *et al.* [11] identify challenges ahead of biological multiscale modeling. For each challenge, they list issues to be addressed along with the significance of each challenge. Although there are a few physical multiscale models proposed in the literature (e.g. CPM [12], Synergetics [13], and Heterogeneous Multiscale Methods [14]), there is no universally adopted theoretical or computational framework for the assembly of multiscale biological models [15].

Bassingthwaighte *et al.* identify six steps for developing multiscale models [16]: (1) defining the model as its highest level of resolution, (2) designing reduced-form modules, (3) determining the range of validity of the reduced form modules, (4) monitoring the variables of the system, (5) replacing higher resolution models with reduced form modules, and finally, (6) validating the performance of the multiscale model against available real data. They further identify issues that must be addressed by any attempt to multiscale modeling. Examples of these issues are parameter identification of closed-loop systems, the identification of input-output delays, and the imposition of known constraints. Their work is among very few attempts to identify challenges ahead of multiscale modeling from a computer science perspective.

In our previous work [17], we proposed a multiagent approach to model the MAPK signaling pathway. Our proposed system is similar to [7] in that every molecular entity is an agent. We further extended [7] by introducing a self-organized approach to learn group behaviours of agents using neural networks. This way, we reduce the complexity of the multiagent system by replacing individual agents with higher-order agents which exhibit similar behaviour. We showed that the transition between scales can be smooth by creating, learning, and validating hierarchies of agents.

III. Hierarchical Middle-Out Learning in a Multiagent System

The most prominent feature of a multiagent system (MAS) is the absence of a top-down control unit. Agents in a MAS interact with their neighbours to perform a task. The nature of agent interactions depends on their given task and varies from information sharing to physical contact. In this paper, we are interested in numerical data exchange in which a pair of agents exchange a numerical value which in turn results in a change of internal states of both agents.

One of the problems with a MAS is high computational costs which makes it inefficient in the presence of a large number of agents. Another problem arises when a multi-agent multiscale system is modelled. Since the dynamics and complexity at each scale are different, a link between two consecutive scales should be established and maintained throughout the simulation. Abstraction is the key to find such a link. The idea of abstraction is that a group of agents that exhibit similar behaviours is subsumed by a higher-order agent with a similar behaviour but reduced computational cost.

In our proposed approach, we aim to create, integrate and break down hierarchies of agents dynamically as the simulation proceeds. How to create hierarchies, how to learn a group behaviour, and how to monitor the validity of a hierarchy are among the issues that we address in this section.

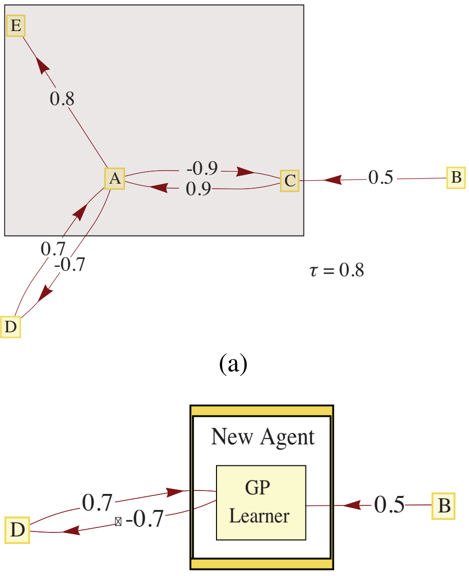
A. Creating Hierarchies of Agents

In our system, each agent builds and updates an *interaction graph* while it interacts with its neighbours. It also keeps track of its interaction values in an *interaction history*. The weight of an edge in the interaction graph is equal to the correlation coefficient with that neighbour. A correlation coefficient between two statistical variables indicates their linear dependancy. A zero correlation coefficient means that two variables are independent, while +1 or -1 shows highly correlated variables. The more two variables are correlated, the more similar their function is. In case there is a series of *n* measurements of agents *s* and *t* in the form of s_i and t_i, where $i = 1, 2, ..., N$, their correlation coefficient (ρ_{st}) is defined as follows:

$$\rho_{st} = \frac{\sum_{i=1}^{N}(s_i - \bar{s})(t_i - \bar{t})}{(n-1)\sigma_s \sigma_t}$$

where $\bar{s}$ and $\bar{t}$ are the mean values of, and σ_s and σ_t are standard deviations of *s* and *t*, respectively.

Having a local weighted graph, each agent then periodically checks if its correlation coefficient with each neighbour is greater than some threshold (τ_{edge}). If so, they form a new agent and repeat this process to identify a cluster of agents that are highly correlated (Algorithm 1). Figure 1 shows an example in which *Agent A* finds *Agent C* and *Agent E*, and they form a higher-order agent. The set of new neighbours is the union of all neighbours of the underlying agents. Having formed such a hierarchy, the next step is to train this new agent, so that it learns and emulates the group behaviour of its underlying agents.



(a)

(b)

Fig. 1: Example of an interaction graph. The edges denote the correlation coefficients. (a) *Agent A*, *Agent C*, and *Agent E* form a hierarchy, (b) The new neighbours of this hierarchy are *Agent B* and *Agent D*.

Algorithm 1 Hierarchy Indentification & Creation

```
$m = current\_agent$;
$Agent\ new\_agent$; {A higher-order agent}
$Queue\ q$;
$q.Enqueue(m)$;
$new\_agent.Add(m)$;
while $!q.empty()$ do
    $Agent\ head = q.Dequeue()$;
    for all $Agent\ s$ in $head$ do
        for all $Agent\ t$ in $s.Neighbours()$ do
            if $|\rho_{st}| \geq \tau_{edge}$ then
                $new\_agent.Add(t)$;
                $q.Enqueue(t)$;
            end if
        end for
    end for
end while
{saving current hierarchy configuration (discussed in section III-C}
$new\_agent.SaveConfig()$;
```

B. Learning the Group Behaviour

Once a higher-order agent is created, it has to subsume the behaviour of its children by learning to emulate their behaviour. Having a set of input/output values in the interaction history of each individual agent, any supervised learning algorithm can be utilized to learn the agents' behaviour. Unlike [17] in which a feed-forward neural network has been used, we use genetic programming (GP) [18] to find a data model for each child inside a higher-order agent. Using a heuristic learning algorithm like GP enables us to control the speed of learning as well. Furthermore, we can also utilize parallel processing to find each GP model.

We include four main operations (*, /, +, -) in the function set of GP. As for the terminal set, in addition to numerical constants, there are two design choices. The first one would be to assign external nodes with incoming edges and ignore internal connections (*Agent D* in Fig. 1). An alternative approach that we take in this paper is to consider internal nodes as well. This way, a GP model has more data elements to be trained with.

C. Monitoring the Validity of Modules

Once a new higher-order agent is created and trained, it subsumes the behaviour of its underlying children. Due to the dynamic behaviour of the system, at some point, the new agent might show invalid behaviours. To address this issue, we check the validity of each higher-order agent periodically. Nonetheless, we need an indicator to compare the current and expected behaviour of the higher-order agent. A heuristic indicator are the previous correlation coefficients of the underlying agent graphs before they are subsumed by the higher-order agents (ρ'_{st}). According to Algorithm 2, we compare the current correlation coefficients of the agent to previous values for each individual agent. If the difference is larger than some threshold (τ_{valid}), we consider the new agent invalid and consequently break its hierarchy down into its previous configuration. Here, unlike [17] which uses non-hierarchical learning, we save all hierarchies when we identify a new agent in order to use them here as we dissolve a higher-order agent back to the simulation (Fig. 2).

Algorithm 2 Validity Monitoring

```
$m = current\_agent$;
$needToBreak = false$;

for all $Agent\ s$ in $m$ do
    for all $Agent\ t$ in $s.Neighbours()$ do
        if $|\rho_{st} - \rho'_{st}| > \tau_{valid}$ then
            $needToBreak = true$;
            $break$;
        end if
    end for
end for

if $needToBreak$ then
    $simulation.remove(m)$;
    for all $Agent\ s$ in $m.hierarchy()$ do
        {$s$ can be an individual or a higher order agent}
        $simulation.add(s)$;
    end for
end if
```

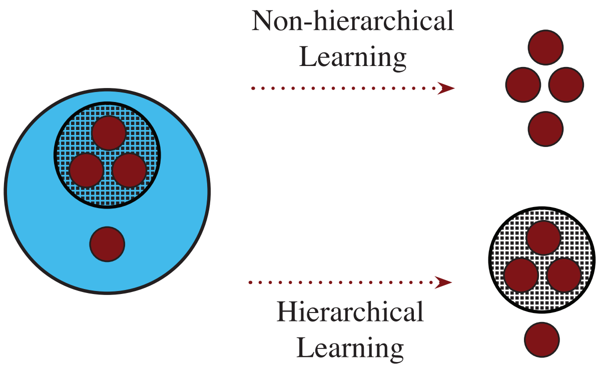


Fig. 2: An example in which one hierarchy breaks down to four individual agents in non-hierarchical learning. In hierarchical learning, as we use it, this hierarchy breaks down into the previous configuration (one higher-order agent and one individual agent).

IV. Experiments on the MAPK Signaling Pathway

Our proposed algorithm can be employed in any multiagent system where agents cooperate by exchanging quantized information. Signaling pathways are such ideal candidates in which an external stimulus causes a cascade of biochemical reactions, which in turn results in changing the concentration of substrates. In the MAPK signaling pathway [19], a hypothetical enzyme E1 stimulates the cell and results in an increase in production of MAPK-PP enzyme (Fig. 3(a)). In another model [20], a negative feedback loop causes sustained oscillations in the production of MAPK-PP (Fig. 3(b)).

Starting with a diagram of a signaling pathway, the next step is to find the update formula for each substrate. Numerical differential equation solvers are used to calculate the concentrations of substrates in the system over time. For example, the update formula for the change of concentration for MAPK-PP is as follows:

$$d[MAPK-PP]/dt = v_8 - v_9$$

$$v_8 = \frac{k_8 \cdot [MKK-PP] \cdot [MAPK-P]}{K_8 + [MAPK-P]}$$

$$v_9 = \frac{V_9 \cdot [MAPK-PP]}{K_9 + [MAPK-PP]}$$

where k_8, K_8, V_9, and K_9 are constants and $[X]$ is the current concentration of substrate X[1].

A. Transforming the MAPK Signalling Pathway to an Agent-Based Model

Similar to [7], we define each substrate in a signaling pathway to be an independent entity which is loosely defined as an agent. In our proposed model, each agent interacts with those substrates that appear in its update formula. This way, we can find a local interaction graph for each agent. Fig. 4 shows

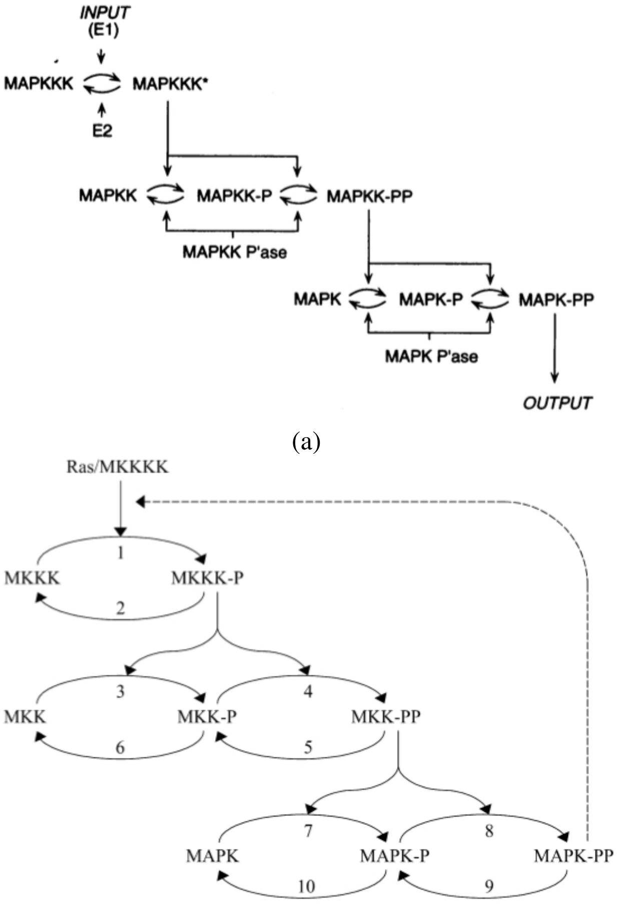


Fig. 3: The MAPK signaling pathway in which an external stimulus E1 initiates a cascade of biochemical reactions (a) Without feedback (from [19]), and (b) With a negative feedback (from [20]).

the complete interaction graphs for the signaling pathways of Fig. 3. Note that Fig. 4(b) has an extra edge compared to Fig. 4(a) which is the negative feedback from MKKK to MAPK-PP.

We validated the performance of our proposed method by conducting a series of experiments on both MAPK models. In all experiments, we use standard genetic programming [18] to learn a model of agent group behaviour. In addition, there are six more parameters to be manually tuned in each experiment which are summarized in Table I. We observe the system for some time (t_{wait}) before starting to create hierarchies within a time interval (Δ_{find}). The size of the *interaction history* is denoted by $hist\_size$. We wait at least for $hist\_size$ time steps to get enough data to learn a group behaviour. Once a hierarchy has been found, we constantly monitor its validity in predefined intervals ($\Delta_{monitor}$). A hierarchy is valid as long as its correlation coefficients with its neighbours do not vary too much with regards to those of individual agents (τ_{valid}). As described in Algorithm 1, if the absolute value of an edge

[1] The complete set of update equations can be found in [19] and [20].

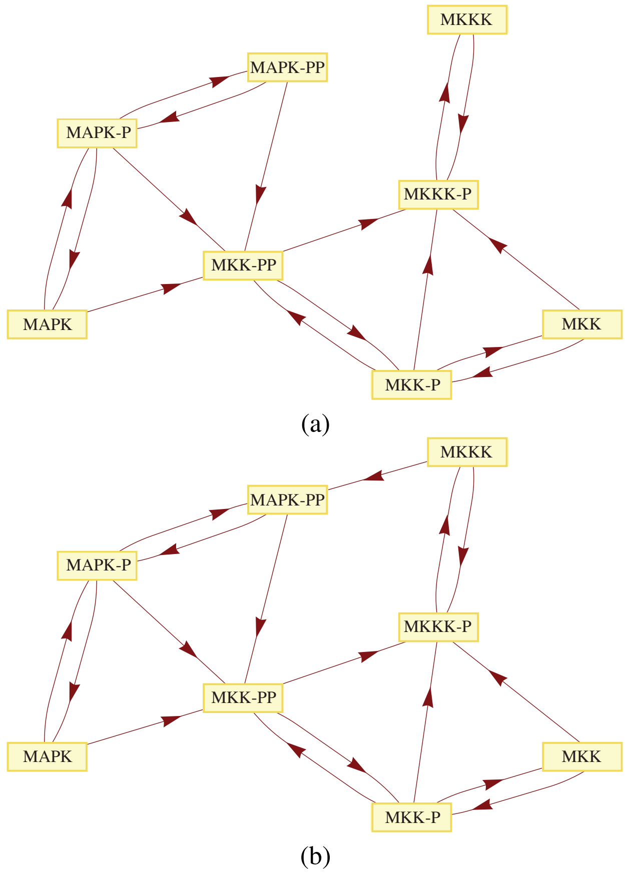


Fig. 4: Agents interaction graph for the MAPK signaling pathways of Fig. 3.

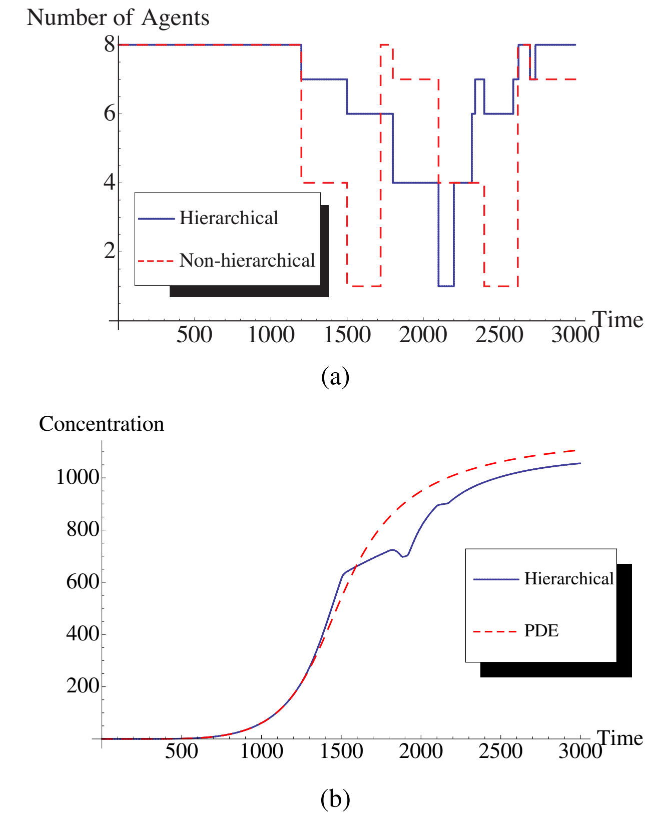


Fig. 5: Results for the first MAPK pathway model of Fig. 3(a). (a) Number of agents (solid line: our hierarchical approach, dashed line: non-hierarchical approach proposed in [17]), (b) Concentration of MAPK-PP.

in an agent's *interaction graph* is greater than some threshold (τ_{edge}), a new hierarchy will be formed.

Fig. 5(a) compares the performance of our proposed hierarchical approach to our previous non-hierarchical approach reported in [17]. Having the same t_{wait}, they both start finding hierarchies after $t = 1200$. Our previous approach reduces the number of agents more quickly. However, when a higher-order agent contains a larger number of individual agents, the probability of that agent to become invalid is higher. Therefore, when an all-encompassing agent in the system breaks down, due to the non-hierarchical nature of that approach, all individual agents are released back into the system, hence making a periodic pattern emerge. In case of hierarchical learning, when a higher-order agent breaks down, it will release the previous hierarchy back to the system. For example, after the current hierarchy with a single agent breaks down at $t = 2200$, there will be 4 agents in the system again (compared to 8 in the case of [17]). Fig. 5(b) shows that the final concentration successfully resembles that of the PDE solver.

As Fig. 6(a) shows, the proposed approach and our previous approach performed similarly on the second MAPK pathway. The number of spikes in both approaches suggests that neither learning methods make a significant difference in case there is periodicity in the system. In particular, a short validity period of both approaches is the result of using the correlation coefficient as an indicator to measure how close two agents work together. Since the correlation coefficient varies from -1 to $+1$ over a periodic signal, it fails to capture the similarity of two agents in a periodic system. This result suggests that we have to look for other indicators when we have a periodic system.

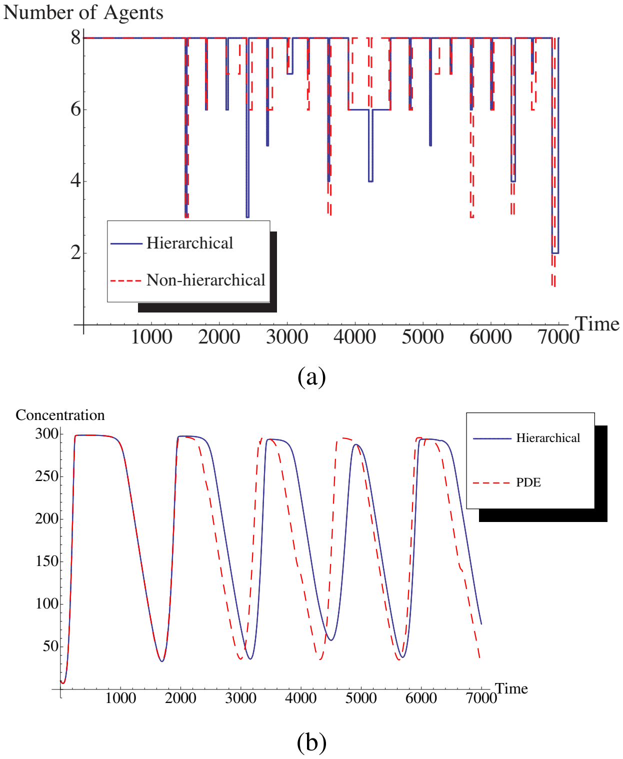
V. Conclusion

In this paper, we extended our previous work by introducing a self-organized hierarchical learning concept using genetic programming. We showed that by using the correlation coefficient, we can identify and create hierarchies of agents thus reducing the complexity of the multiagent environment while emulating individual behaviours. Although we demonstrated our algorithm in a case in which agents exchange numerical values, we believe that our proposed approach could be utilized in any multiagent system provided that a correlation coefficient between each pair of agents could be deduced from their interactions.

In order to apply our proposed method to a wider range of multiagent settings, we need to look for other measures of agent cooperation. Another reason to look for other measures

TABLE I: Model Parameters

Parameter Name	Symbol	Value in Experiment 1	Value in Experiment 2
Delay before finding hiearchies	t_{wait}	1200	1500
Hierarchy finding interval	Δ_{find}	300	300
Monitoring interval	$\Delta_{monitor}$	20	20
Validity threshold	τ_{valid}	0.1	0.1
Edge threshold	τ_{edge}	0.99	0.7
Interaction history size	$hist\_size$	1000	1000



(a)

(b)

Fig. 6: Results for the second MAPK pathway model of Fig. 3(b). (a) Number of agents (solid line: our hierarchical approach, dashed line: non-hierarchical approach proposed in [17]), (b) Concentration of MAPK-PP.

is the inability of the correlation coefficient to capture periodic systems effectively. In this research, we focused only on a multiagent problem in which agents have rather simple behaviour. In more advanced problems in which agents have several behaviours, we also need to identify what to learn, too. Altogether, this work is among very few attempts to find an algorithmic framework to address the challenge of complexity reduction in an agent-based system. Our proposed approach can serve as the first step to address the reduction of complexity in highly complex systems.

References

[1] C. Jacob, A. Barbasiewicz, and G. Tsui, “Swarms and genes: Exploring λ-switch gene regulation through swarm intelligence,” in *CEC 2006, IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada*, 2006.

[2] C. Jacob and I. Burleigh, “Biomolecular swarms: An agent-based model of the lactose operon,” *Natural Computing*, vol. 3, no. 4, pp. 361–376, December 2004.

[3] F. Amigoni and V. Schiaffonati, “Multiagent-based simulation in biology a critical analysis,” *Model-Based Reasoning in Science, Technology, and Medicine*, vol. 64, pp. 179–191, Jun 2007.

[4] G. Desmeulles, G. Querrec, P. Redou, S. Kerdelo, L. Misery, V. Rodin, and J. Tisseau, “The virtual reality applied to biology understanding: The in virtuo experimentation,” *Expert Syst Appl*, vol. 30, no. 1, pp. 82–92, Jan 2006.

[5] S. von Mammen, J.-P. Stehöfer, J. Denzinger, and C. Jacob, “Self-organized middle-out abstraction,” in *IWSOS 2011: 5th International Workshop on Self-Organizing Systems*, 2011, pp. 26–31.

[6] P. P. Gonzalez, M. Cardenas, D. Camacho, A. Franyuti, O. Rosas, and J. Lagunez-Otero, “Cellulat: an agent-based intracellular signalling model,” *Biosystems*, vol. 68, no. 2-3, pp. 171–185, 2003.

[7] S. Khan, R. Makkena, F. McGeary, K. Decker, W. Gillis, and C. Schmidt, “A multi-agent system for the quantitative simulation of biological networks,” in *AAMAS ’03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM, 2003, pp. 385–392. [Online]. Available: http://dx.doi.org/10.1145/860575.860637

[8] L. Nayak and R. K. De, “An algorithm for modularization of mapk and calcium signaling pathways: comparative analysis among different species,” *J Biomed Inform*, vol. 40, no. 6, pp. 726–49, Dec 2007.

[9] M. Martins, S. F. Jr, and M. Vilela, “Multiscale models for biological systems,” *Current Opinion in Colloid & Interface Science*, vol. 15, no. 1-2, pp. 18 – 23, 2010.

[10] E. Z. Erson and M. C. Cavuşoğlu, “A software framework for multiscale and multilevel physiological model integration and simulation,” *Conf Proc IEEE Eng Med Biol Soc*, vol. 2008, pp. 5449–53, Jan 2008.

[11] C. Lavelle, H. Berry, G. Beslon, F. Ginelli, J.-L. Giavitto, Z. Kapoula, A. L. Bivic, N. Peyrieras, O. Radulescu, A. Six, V. Thomas-Vaslin, and P. Bourgine, “From molecules to organisms: towards multiscale integrated models of biological systems,” *Theoretical Biology Insights*, vol. 1, pp. 13–22, 2008.

[12] R. M. Merks and J. A. Glazier, “A cell-centered approach to developmental biology,” *Physica A: Statistical Mechanics and its Applications*, vol. 352, no. 1, pp. 113 – 130, 2005.

[13] H. Haken, *Synergetics : an introduction : monequilibrium phase transitions and self-organization in physics, chemistry and biology / Hermann Haken*. Springer-Verlag, Berlin :, 1977.

[14] E. Weinan, B. Engquist, X. Li, W. Ren, and E. Vanden-Eijnden, “Heterogeneous Multiscale Methods: A Review,” *Communications in Computational Physics*, vol. 2, no. 3, pp. 367–450, Jun. 2007.

[15] D. C. Walker and J. Southgate, “The virtual cell–a candidate coordinator for ’middle-out’ modelling of biological systems,” *Briefings in Bioinformatics*, vol. 10, no. 4, pp. 450–461, Jul 2009.

[16] J. Bassingthwaighte, H. Chizeck, and L. Atlas, “Strategies and tactics in multiscale modeling of cell-to-organ systems,” *Proceedings of the IEEE*, vol. 94, no. 4, pp. 819 – 831, Apr 2006.

[17] A. Sarraf Shirazi, S. von Mammen, and C. Jacob, “Adaptive modularization of the mapk signaling pathway using the multiagent paradigm,” in *Parallel Problem Solving from Nature – PPSN XI*. Springer, 2011, vol. 6239, pp. 401–410.

[18] J. R. Koza, “Genetic programming as a means for programming computers by natural selection,” *Statistics and Computing*, vol. 4, pp. 87–112, 1994.

[19] C. Y. Huang and J. E. Ferrell, “Ultrasensitivity in the mitogen-activated protein kinase cascade,” *Proc Natl Acad Sci USA*, vol. 93, no. 19, pp. 10 078–83, Sep 1996.

[20] B. N. Kholodenko, “Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades,” *Eur J Biochem*, vol. 267, no. 6, pp. 1583–8, Mar 2000.