

# Large Scale Agent Based Social Simulations with High Resolution Raster Inputs in Distributed HPC Environments



Sergiy Gogolenko

**Abstract** Agent-based modelling and simulation (ABMS) is an essential tool which allows to explore the role of social phenomena via computer simulation. Large scale social simulations of HPCs—also known as parallel and/or distributed agent-based simulation (PDABS)—play a key role in the emerging field of computational global systems sciences (GSS). Agent-based models (ABMs) in GSS are characterized by highly non-uniform spatial distribution of agents and importance of long distance social interactions. Over the last two decades, researchers proposed a number of approaches to effectively address these traits of ABMs. Such approaches are driven by data available for scientists. In many GSS applications, the data partially come in raster formats. Yet, case of raster inputs for GSS applications is barely studied in literature and not supported sufficiently in the state-of-the-art ABMS frameworks. In this paper, we propose a graph-based approach to represent ABMs with raster inputs on HPCs. This approach naturally leads to a space-relationships-based work partitioning strategy which allows to improve performance of PDABS.

## 1 Introduction

Large scale social simulations play a key role in the emerging field of computational global systems sciences (GSS) which deals with providing “scientific evidence to support policy-making, public action, and civic society” [1]. to provide scientific evidence to support policy-making, public action and civic society to collectively engage in societal action. “The behaviour of many social systems requires that they be modelled at the level of individual people” [2], which is usually achieved by agent-based modelling and simulation (ABMS). On the one hand, since GSS applications analyze society on global or country level, individual-based view on

---

S. Gogolenko (✉)  
High Performance Computing Center Stuttgart (HLRS), Stuttgart, Germany  
e-mail: [gogolenko@hlrs.de](mailto:gogolenko@hlrs.de)

the global systems naturally leads to computationally expensive large scale ABMS runs. On the other hand, the modeler obtains flexibility in addressing heterogeneity in agents, non-linearity in their responses, and other complex model assumptions. This flexibility enables ABMS to capture emergent social phenomena overlooked by macro- and meso-scale models [3], as well as to outperform conventional machine learning techniques in some cases [4].

At the same time, agent-based models (ABMs) for GSS have many peculiarities. In [5], authors list three major traits of ABMs in GSS, which differ them from ABMs encountered in other scientific domains such as computational biology or ecology. These are heterogeneity of agents, highly non-uniform spatial distribution of agents in the environment, and important role of long distance (social) interactions. Heterogeneity of agents reflects diversity of actors involved into GSS models. Non-uniform spatial distribution of agents is caused by urbanization processes, obstacles imposed by nature, etc. ABMs in GSS often encompass two types of communications—short distance communications representing interactions due to spatial proximity of agents, and long distance interactions standing for social relationships. In many cases, the latter dominate over the former.

Over the last two decades, researchers proposed a number of models and data structures to effectively address traits of ABMs in GSS on HPC clusters. These models and data structures are driven by data available for scientists. The data about environment may come in vector or raster format. ABMs with inputs in vector formats cover the majority of GSS use cases. Examples of recent developments focused on inputs in vector formats include, but not limited to hierarchical (tree-like) models [6, 7], directed probabilistic social networks [8], social contact networks [9], urban geo-social networks [10]. At the same time, in many applications the global system scientists use the data about ABM environment available in raster formats from popular open data sources like NASA’s Socioeconomic Data and Applications Center [11] and others. In this paper we propose an HPC compliant model and corresponding data structure for this situation.

The rest of the paper is organized as follows. Section 2 briefly reviews state-of-the-art HPC compliant ABMS software for global system scientists, as well as approaches to model spatial environment and distribute workload implemented in these tools. Section 3 presents graph-based HPC compliant model and corresponding data structure for ABMs with raster inputs and strong long distance interactions between agents. Section 4 illustrates performance of our solution on the toy benchmark implementing Axelrod’s model of dissemination of culture. Finally, Sect. 5 discusses conclusions and direction for further work.

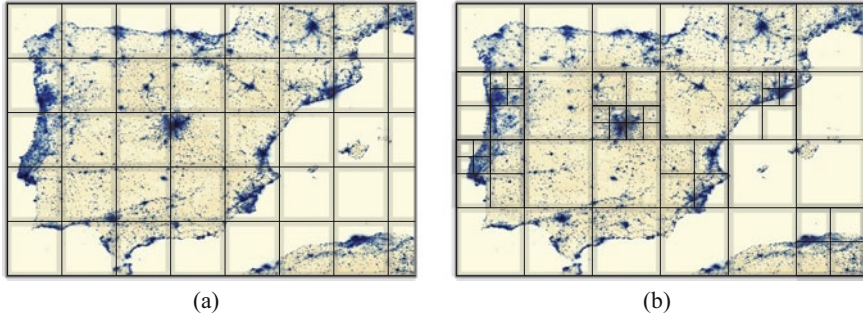
## 2 Previous Work

During the last decade, a number of HPC compliant ABMS codes were developed [12–14]. These developments vary from domain specific tools to general purpose ABMS frameworks.

Although domain specific tools usually target individual use cases, they address traits of particular GSS applications very effectively. Moreover, many ideas implemented in such tools have generic nature and can be applied to a broader number of use cases. The remarkable examples of the domain specific tools are FluTE [7], EpiFast [8], EpiSimdemics [9]. FluTE [7] represents the model of iterations in society as a multi-level tree. The root of the tree corresponds to the whole society, while the lower levels of the tree represent elements of society with finer granularity until reach the level of individual households as leaves. The probability of interactions between agents is dictated by the distance to the closest common parent. EpiSimdemics [9] implements a so-called social contact network (SCN) model. In SCN, society is represented by an affiliation (bipartite) graph with agents on one side and loci of their interactions (environment) on the other. This affiliation graph is accompanied with a schedule of interactions. Both SCN and tree-like models take into account only short range interactions between agents. Urban Geo-Social Network Model (UGSN) proposed in [10] addresses this limitation by further development of the SCN idea. It represents the society by SCN and additional multilayer network [15] of direct social connections between agents for modelling long distance communications.

Noticeable examples of the HPC compliant general purpose ABMS frameworks are RepastHPC [16], D-MASON [17], Flame-GPU [18], and Pandora [19]. Despite the wide choice, these frameworks fail to address all common traits of GSS applications effectively. Being implemented in Java, D-MASON has limited potential for use and porting on state-of-the-art large-scale HPC clusters. FlameHPC and Pandora lack proper support for simulation of social connections between agents. Although RepastHPC has formally all components required to build ABMs in GSS, it demands significant HPC expertise and advanced programming skills from the modeler (due to intricate and verbose API). Moreover, latest version of RepastHPC still ignores recent advances in high performance data structures such as new techniques for handling evolving graphs, modern fast implementations of hash tables, etc.

The common bottleneck for the majority of general-purpose ABMS frameworks is a naïve approach to model spatial environment and distribute workload. Many popular frameworks—including Flame-GPU, RepastHPC, and Pandora—model environment topology by cartesian grids. In 2D case, environment attributes are represented by dense matrices of the same size. Indices of the matrices correspond to the spatial coordinates and define locations of the grid vertices. During the distributed simulations, cartesian grid is split evenly between processes (Fig. 1a). This approach is often referred as uniform partitioning [20]. Since amount of computational work in agent based simulation step is proportional to the number of agents, uniform partitioning results in a significant load imbalance if agents are distributed very non-uniformly in space. As a result, this approach allows to reach reasonably good performance for many classical ABMS applications, but gives poor performance in situations with non-uniform spatial distribution of agents which is a case of GSS applications where agents are highly concentrated in the urban areas and sparsely distributed outside the settlements. D-MASON tackles this



**Fig. 1** Approaches to partition the rasters. (a) Uniform partitioning. (b) Non-uniform (tree codes based) partitioning

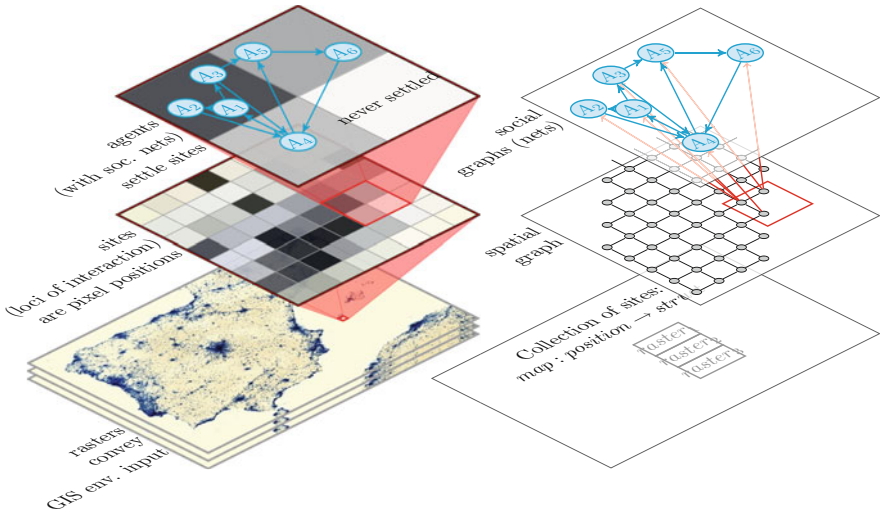
limitation by introducing a space-based non-uniform work partitioning approach based on tree codes (multi-scale meshes) [20]. This approach extends idea of the quad-tree Barnes-Hut algorithm, widely used in  $n$ -body simulations, to the agent-based models [21]. In particular, in [20], authors propose to use a so called bounded pseudo quad-tree (Fig. 1b). Even though the space-based work partitioning approach significantly reduces load imbalance, it does not take into account the situation when long distance communications play an important role in social simulations. As discussed in Sect. 1, the latter refers to the vast majority of GSS applications.

### 3 Graph-Based Model for Sparse Raster Inputs

#### Concept

Figure 2 illustrates typical organization of inputs for ABMs where data about environment comes in raster format. The information from the rasters can be combined into sparse spatial graph. Vertices of the spatial graph correspond to the non-empty pixels of the rasters and represent sites populated by agents. Each site is attributed with a tuple of pixel values in the corresponding position for all raster. Edges of the spatial graph stand for spatial proximity between sites and serve to model short distance communications. Agents are linked into a multilayer network of direct social connections. In addition, each agent is assigned to the site corresponding to its spatial location. This results into internal representation where agents linked into multilayer network of social connections  $G_A = (V_A, E_A)$  are mapped on the sites linked into spatial graph  $G_S = (V_S, E_S)$  (see right side of Fig. 2).

In order to keep workload balanced, the spatial graph should be distributed between processes taking into account the number of agents in sites, as well as short and long distance communications between agents. It can be achieved if we map multilayer network of social connections on the spatial graph to obtain



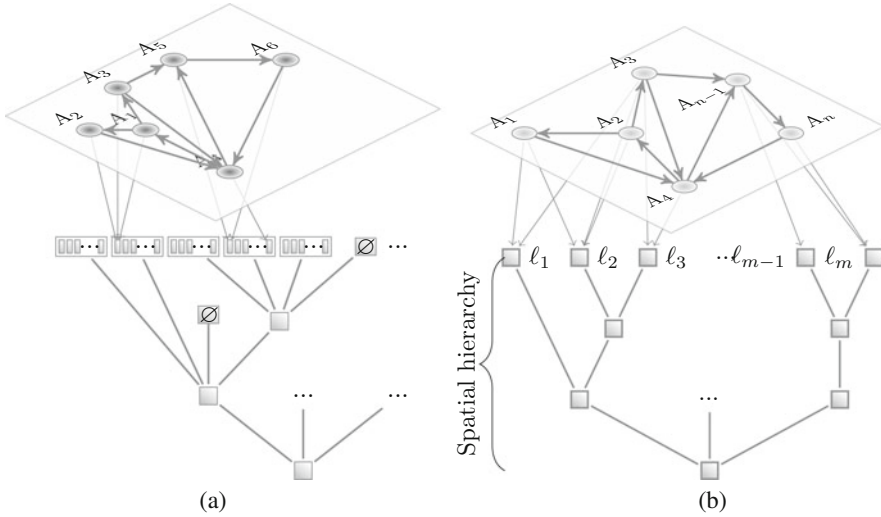
**Fig. 2** Internal representation of ABMs with raster inputs

a computational graph  $G_c = (V_c, E_c)$  with vertices  $v \in V_c$  corresponding to sites and edges  $e \in E_c$  corresponding to short and long distance communications. In this graph, weight of the vertex  $w_v$  equals the number of agents located at the corresponding site, while weight of the edge  $w_e$  between vertices equals the total number of agents in both vertices if sites are spatially connected or the number of social links between agents assigned to these vertices if sites are distant. Optimal partitioning of the computational graph gives balanced distribution of agents between processors.

However, if rasters have high resolution, this approach cannot be used directly in distributed HPC environments since the number of sites becomes too big to address them effectively and to perform a balanced partitioning of the computational graph. This obstacle can be overcome by grouping sites into the chunks and partitioning the computational graph built upon the chunks of sites instead of individual sites. Nevertheless, even spatial partitioning of the sites into chunks might significantly disbalance the number of agents assigned to chunks. The better way is to build chunks upon tree codes—quad-trees or bounded pseudo quad-trees—using approaches discussed in [20]. The latter leads to a graph-based model and corresponding data structure illustrated in Fig. 3a. Note that this model uses the data structure similar to the data structure behind combination of USGN model with hierarchical tree-like model (see Fig. 3b). According to the taxonomy of PDABS work partitioning strategies proposed in [20], the corresponding work partitioning approach belongs to the class of space-relationships-based strategies.

**Software for Implementation**

Neither of the data structures depicted in Fig. 3 can be implemented in existing HPC compliant ABMS frameworks without dramatic changes in their cores. In



**Fig. 3** Comparison of data structures of fine grained graph-based models designed for ABMs with sparse raster inputs and with vector inputs. **(a)** Graph-based with sparse raster inputs. **(b)** UGSN + hierarchical

**Table 1** Support of short distance and long distance interactions between agents in Pandora and RepastHPC

	Pandora	RepastHPC
<i>Social relationships representation</i>		
Type of model	None	Directed multilayer network
Evolving social graphs		+
Edge weights		+
Multiplicity		+
Implementation		Native (ptrs)
<i>Spatial component representation</i>		
Format	std::map of dense matrices	Grid projector with vector of dense matrices (value layers)
Boundary conditions	Von Neumann	Von Neumann, Moore
GIS support	Rasters with GDAL	None

order to support this claim, Table 1 compares the most advanced HPC compliant frameworks written in C++—Pandora and RepastHPC—with respect to coverage of features necessary to implement the approach discussed above. This comparison shows that Pandora does not support multilayer network of social connections, whereas RepastHPC has insufficient number of instruments to implement spatial graphs.

In order to implement the approach without ABMS frameworks, one needs a graph partitioning tool and a general purpose graph library, which provides functionality sufficient to model social relationships.

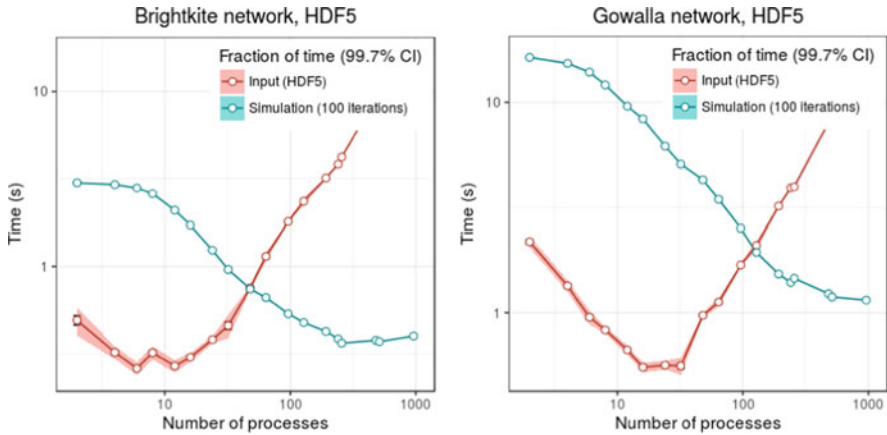
The most appropriate general purpose graph libraries are Snap[22], PBGL, and GraphLab/PowerGraph[23]. PBGL (Parallel Boost Graph Library) is a rather lightweight package which supports most of the features required to model multilayer network of social connections and spatial graph. Even though interface of PBGL is designed for the users with advanced CS-skills, VTK library provides easy-to-use wraps over native PBGL interfaces. PowerGraph is an advanced distributed framework which implements graph-based gather-apply-scatter (GAS) programming model[23]. On the one hand, concept of the graph-based ABM simulation maps perfectly on the GAS programming model. In particular, “apply” phase allows to specify behaviour of the agents, and “gather” phase allows to collect suitable information from neighbours. On the other hand, PowerGraph does not support dynamic changes in the graph structure (vertices removal, etc). The latter strongly limits potential use of PowerGraph for ABMS. Both PBGL and PowerGraph assume that all vertices must have the same attributes. Nevertheless, the effect of versatility in vertex attributes can be achieved with variant types.

The incomplete list of remarkable graph partitioning packages developed over the last decades includes PT-Scotch, ParMETIS, PaGrid, Chaco, JOSTLE, MiniMax, ParaPART, DRUM, etc. But two of them—METIS and Scotch—gained much more popularity than others and are often referred as load balancing tools of choice in sophisticated time-consuming parallel numerical simulations. While both packages fit well to the needs of graph-based approach, ParMETIS is preferable since it allows to repartition distributed graph dynamically.

## 4 Benchmark for a Proof-of-Concept Implementation

In order to assess performance of our solution, we prepared a toy benchmark that implements Axelrod’s model of dissemination of culture. This model was proposed in 1996 by R. Axelrod[24] and immediately gained broad popularity among social scientists. Nowadays, it is considered as one of the most well studied ABMs—both theoretically and empirically[25, 26],—which motivated us to choose Axelrod’s model for benchmarking.

The model defines agents and rules for their interactions as follows. Agents model individuals in culture dissemination process. Each agent is endowed with  $F$  integer attributes called cultural traits, which are meant to model different beliefs, opinions, and other properties of agents. The model allows only a limited number of values for each cultural trait  $f_i = (0, 1, \dots, q_i - 1)$ . In the dynamic step, each agent randomly selects one neighbour and the agent interacts with the neighbour with some probability proportional to the overlaps between the agent-neighbour pairs (the overlap is computed as a number of equal features). The interaction consists in assigning to one of the agent’s trait the value of its neighbour trait. In other words, these rules make interacting agents more similar, but the interaction happens more often if agents already share many traits and it never happens if agents have no trait in common. This suggests that Axelrod’s interaction rules allow



**Fig. 4** Scalability of Axelrod’s model implemented with the Amos framework on Hazelhen cluster

to model two cultural mechanisms—social influence and homophily. In order to fit Axelrod’s model to the graph-based approach discussed in Sect. 3, we slightly modified Axelrod’s notion of neighbours. In our implementation, we consider as neighbours all agents located at the same spatial site, as well as the agents that have direct links in the social graphs.

The benchmark was performed on the Hazelhen cluster at HLRS. Hazelhen is composed of CRAY XC40 nodes and has peak performance 7.4 Pflops. The cluster includes 41 Cray cascade cabinets with in total 7712 dual socket compute nodes. Each node is equipped with 2 12-core Intel Haswell E5-2680v3 CPUs and 128 GB of DDR4 RAM. The type of interconnect is Cray Aries. It uses Lustre storage and operates with the Cray Linux Environment. In our implementation, we used Snap 3 library as a back-end general-purpose graph library, and METIS as a graph partitioning tool. We compiled all components with GCC 6.4. Input files were pre-processed into CSV-format.

In our benchmarks, we used two networks from [22]—Brightkite and Gowala—for representing long-range interactions. The number of agents was artificially adjusted to the number of vertices in the networks. In order to create sites for their allocation, we used a  $240 \times 290$  pixel raster with a gridded population density heat map of the Faroe islands from NASA’s SEDAC. Agents were initialized with three cultural features each taking random values between 0 and 9.

Figure 4 summarizes results of the benchmark. Its subplots contain line chart with confidence intervals for measured elapsed times of data input and 100 simulation iterations of Axelrod’s model. In these plots, we compare performance of the simulation iterations against embarrassingly parallel data input on both social networks. By embarrassingly parallel CSV input, we mean a naïve embarrassingly parallel implementation of the CSV reader which assumes that the user split the input data and prepared CSV files for each MPI process separately. For both



networks, the scalability of the simulation iterations is better than the scalability of embarrassingly parallel input.

## 5 Conclusions and Further Work

The case of raster inputs for GSS applications is barely studied in literature and not supported sufficiently in the state-of-the-art ABMS frameworks. In this paper, we proposed a new graph-based model and corresponding data structure for efficient implementation of ABMs with raster inputs on HPC clusters. This model combines ideas of UGSN model, hierarchical models, and tree codes. State-of-the-art ABMS frameworks do not provide sufficient features to implement such model out-of-the-box. Nevertheless, we have shown that the model can be efficiently implemented with the general-purpose graph libraries and graph partitioning tools.

In the future, we plan to include support of this model in one of the ABMS frameworks, as well as to assess performance of our model on the large scale real world use cases.

## References

1. EC: Global systems science. Internet (2017). Accessed 24 Jun 2019
2. Dum, R., Johnson, J.: Global systems science and policy. In: *Non-equilibrium Social Science and Policy. Understanding Complex Systems*, pp. 209–225. Springer International Publishing, Cham (2017)
3. Farmer, J.D., Foley, D.: The economy needs agent-based modelling. *Nature* **460**(7256), 685–686 (2009)
4. Suleimenova, D., Bell, D., Groen, D.: A generalized simulation development approach for predicting refugee destinations. *Sci. Rep.* **7**(1), 13377 (2017)
5. Paolotti, D., Tizzoni, M., Edwards, M., Fürst, S., Geiges, A., Ireland, A., Schütze, F., Gesine, S.: D4.1-first report on pilot requirements. Deliverable 4.1, CoeGSS—Centre of Excellence for Global Systems Science (2016)
6. Germann, T.C., Kadau, K., Longini, I.M., Macken, C.A.: Mitigation strategies for pandemic influenza in the United States. *Proc. Natl Acad. Sci.* **103**(15), 5935–5940 (2006)
7. Chao, D.L., Halloran, M.E., Obenchain, V.J., Longini, I.M.: Flute, a publicly available stochastic influenza epidemic simulation model. *PLoS Comput. Biol.* **6**(1), e1000656 (2010)
8. Bisset, K.R., Chen, J., Feng, X., Anil Kumar, V.S., Marathe, M.V.: EpiFast. In: *Proceedings of the 23rd International Conference on Supercomputing - ICS '09* (2009)
9. Barrett, C.L., Bisset, K.R., Eubank, S.G., Feng, X., Marathe, M.V.: Episimdemics: an efficient algorithm for simulating the spread of infectious disease over large realistic social networks. In: *2008 SC: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–12 (2008). <https://dl.acm.org/doi/abs/10.5555/1413370.1413408>
10. Hristova, D., Williams, M.J., Musolesi, M., Panzarasa, P., Mascolo, C.: Measuring urban social diversity using interconnected geo-social networks. In: *Proceedings of the 25th International Conference on World Wide Web - WWW '16*, pp. 21–30 (2016)

11. CIESIN and CIAT. Gridded population of the world version 3 (GPWv3): Population density grids. Socioeconomic data and applications center (SEDAC). Internet (2019). Accessed 24 Aug 2019
12. Kravari, K., Bassiliades, N.: A survey of agent platforms. *J. Artif. Soc. Soc. Simul.* **18**(1), 11 (2015)
13. Abar, S., Theodoropoulos, G.K., Lemarinier, P., O'Hare, G.M.P.: Agent based modelling and simulation tools: a review of the state-of-art software. *Comput. Sci. Rev.* **24**, 13–33 (2017)
14. Rousset, A., Herrmann, B., Lang, C., Philippe, L.: A survey on parallel and distributed multi-agent systems for high performance computing simulations. *Comput. Sci. Rev.* **22**, 27–46 (2016)
15. Kivela, M., Arenas, A., Barthelemy, M., Gleeson, J.P., Moreno, Y., Porter, M.A.: Multilayer networks. *J. Complex Netw.* **2**(3), 203–271 (2014)
16. Macal, C., North, M., Pieper, G., Drugan, C.: Modeling: agent-based modeling and simulation for EXASCALE computing. *SciDAC Rev.* **8**, 34–41 (2008)
17. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., Balan, G.: Mason: a multiagent simulation environment. *Simulation* **81**(7), 517–527 (2005)
18. Richmond, P., Walker, D., Coakley, S., Romano, D.: High performance cellular level agent-based simulation with flame for the GPU. *Brief. Bioinform.* **11**(3), 334–347 (2010)
19. Rubio-Campillo, X.: Pandora: a versatile agent-based modelling platform for social simulation. In: *The Sixth International Conference on Advances in System Simulation, SIMUL 2014*, pp. 29–34 (2014)
20. Cordasco, G., Spagnuolo, C., Scarano, V.: Work partitioning on parallel and distributed agent-based simulation. In: *2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 1472–1481 (2017)
21. Barnes, J., Hut, P.: A hierarchical  $O(N \log N)$  force-calculation algorithm. *Nature* **324**(6096), 446–449 (1986)
22. Leskovec, J., Sosič, R.: Snap: a general-purpose network analysis and graph-mining library. *ACM Trans. Intell. Syst. Technol.* **8**(1), 1–20 (2016)
23. Gonzalez, J.E., Low, Y., Gu, H., Bickson, D., Guestrin, C.: Powergraph: distributed graph-parallel computation on natural graphs. In: *Presented as part of the 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 12)*, pp. 17–30. USENIX, Hollywood (2012)
24. Axelrod, R.: The dissemination of culture: a model with local convergence and global polarization. *J. Conf. Resolut.* **41**(2), 203–226 (1997)
25. Castellano, C., Fortunato, S., Loreto, V.: Variants of the Axelrod mode. in *statistical physics of social dynamics*. *Rev. Mod. Phys.* **81**(2), 591–646 (2009)
26. Lanchier, N.: The Axelrod model for the dissemination of culture revisited. *Ann. Appl. Probab.* **22**(2), 860–880 (2012)