

Exploring Agent-Based Simulations using Temporal Graphs

R. Jordan Crouser*
Tufts University

Jeremy G. Freeman †
Tufts University

Remco Chang‡
Tufts University

ABSTRACT

Agent-based simulation has become a key technique for modeling and simulating dynamic, complicated behaviors in social and behavioral sciences. Lacking the appropriate tools and support, it is difficult for social scientists to thoroughly analyze the results of these simulations. In this work, we capture the complex relationships between discrete simulation states by visualizing the data as a temporal graph. In collaboration with expert analysts, we identify two graph structures which capture important relationships between pivotal states in the simulation and their inevitable outcomes. Finally, we demonstrate the utility of these structures in the interactive analysis of a large-scale social science simulation of political power in present-day Thailand.

1 INTRODUCTION

Modeling dynamic, complex behaviors is a key component of research in political science. In recent years, agent-based simulation has become a vital technique for modeling and exploring these behaviors a country’s political patterns, which might indicate the imminence of civil unrest and help predict catastrophic events [6]. A complete exploration of the resulting simulation data could help social and political scientists understand nuanced social behaviors, find patterns of violence and socioeconomic repression, predict catastrophic events and better inform the decision-makers who influence global policy. Unfortunately, existing methods and tools only permit either a broad-overview analysis of the entire dataset, or an in-depth analysis of a single simulation run. Without the ability to interactively compare simulation runs and understand how they relate, it is extremely difficult to effectively interpret and analyze the results of these simulations.

We begin to address these needs by developing a theoretical framework and visualization techniques for analyzing complex dynamic behaviors using temporal graphs. In this graph, a vertex represents a unique configuration of the variables in the simulation, and a directed edge indicates a temporal transition between two states. We aggregate a large set of sample sequences from the agent-based simulation to build the graph. Each of these sequences can be thought of as a pathway through the “simulation space”, the set of reachable configurations in the high-dimensional parameter space. Substructures in the graph can be mapped to semantically meaningful relationships between simulation states, and we can develop efficient algorithms to support interactive analytical reasoning. In collaboration with expert analysts in political science, we identify several important relationships between states in the simulation. By supporting interactive exploration of these underlying structures, we can assist domain experts in uncovering important yet subtle patterns in their simulations.

*e-mail: rcrouse01@cs.tufts.edu

†e-mail: jeremy.freeman@cs.tufts.edu

‡e-mail: remco@cs.tufts.edu

2 GRAPHS IN VISUAL ANALYTICS

Visualization of graph structures is a well-studied area in computational science. For a survey, see work by Herman et al. [3]. However, the study of graphs in visual analytics has primarily been restricted to the study of entity relationships. There has been much interest in tools for visualizing these relationships in the context of social networks [2], protein interaction [4], and more. There is also some existing work in visualizing general dynamic graphs [1, 7] as well as recent interest in visualizing entity-relationship graphs with a temporal component in the area of information propagation through a network [5]. While these graphs encode a temporal component of the data they represent, to our knowledge there exists no research on the utilization of graph structures for the analysis of aggregated temporal simulation data.

3 METHODS

To evaluate the this framework and validate our claim that temporal graphs are useful in an interactive visual analysis environment for political scientists, we implemented a functional interactive prototype within Cytoscape [8], an existing graph visualization system. We then built a temporal graph of the results of a real-world political science simulation of political violence and power systems in present-day Thailand. This simulation models the interaction of 31 political identities, tracking their popularity, level of violence, protest, power and influence. The resulting graph contains 14,887 unique vertices and 59,000 edges. The graph formulation was then presented to a panel of 4 analysts for discussion and preliminary evaluation.

4 EVALUATION: FINDING MEANING THROUGH STRUCTURE

In the following sections, we describe several state relationships that are of interest to our collaborators in political science. While a formal proof is beyond the scope of this paper, our research has shown that each of these relationships can be efficiently computed on a standard desktop machine. All empirical results were obtained using an Intel Core 2 Quad 2.66Ghz desktop with 2GB of RAM running Windows XP.

4.1 Disaster Avoidance

One important task facing the analysts we interviewed is maintaining an awareness of threat scenarios and developing strategies for efficient, cost-effective disaster avoidance. Using filtering tools, the analysts can identify clusters that exhibit high levels of violence or states where a known oppressive regime is in power. Once these states have been located, it is of great interest to them to be able to detect vertices that inevitably lead into these situations, and to identify critical decision points preceding these paths (see Fig. 1). These states may appear non-threatening on the surface, leading traditional analysis to overlook them. However, they may represent a favorable balance of conditions for the rise to power of the oppressive organization or violent regime. By exploring the configurations on the periphery of the set, the analysts are able to make recommendations regarding early warning signs to watch for in the real world and to identify critical decision points that could help avert disaster.

To this end, we developed an iterative set-building algorithm to identify these state with a theoretical bound of $O(n^2)$ time and a

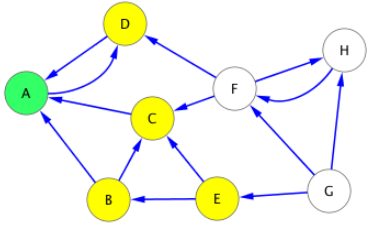


Figure 1: Every maximal path leaving a yellow vertex eventually passes through A. If we are trying to avoid reaching A, vertices F and G are critical decision points.

remarkably fast in-practice running time. To demonstrate the applicability of this algorithm for use in real-time analysis, we systematically ran our implementation on each of the vertices in the graph and tracked its running time. In all instances, the algorithm terminated in under *1ms*. This result provides strong evidence for the applicability of this algorithm to interactive data exploration using temporal graphs.

4.2 Forecasting Outcomes

Another important area of concern for our analysts is preparation for upcoming events. Using a graph built from their simulation data, it is helpful for them to be able to locate a state in the system that closely resembles current conditions and to then detect unavoidable future states. That is, given a starting vertex representing a configuration of the simulation, determine the set of inevitable outcomes. We can also define this relationship as the set of vertices that are visited along every path leading out of the starting vertex (see Fig. 2). Unfortunately, it can be shown that the problem of finding the intersection of all maximal paths in a graph is $\#P$ -complete¹, and is therefore not a feasible approach. A second possibility would be to determine the subgraph reachable from the starting position, and naïvely run the algorithm from the previous section on all vertices in the subgraph. This method could take up to $O(n^3)$ on a complete graph on n vertices. While this is certainly a dramatic improvement over the previous method, it is still not efficient enough to be run in real time.

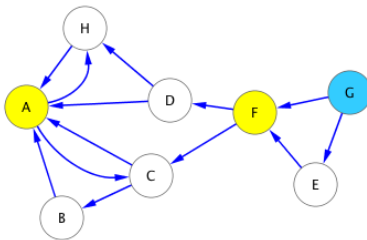


Figure 2: Every path leaving G passes eventually through each of the yellow vertices.

In order to be able to compute these values in real time, we developed a hybrid algorithm that leverages the benefits of both of these approaches. By first choosing a random maximal path and running the algorithm only on each vertex along the path rather than the entire subgraph, we perform a preliminary set reduction and may eliminate a large portion of the computation. The construction of a random maximal path may stop upon traversing any edge leading back to a vertex already in the path with full confidence that the path contains all the desired vertices. Our empirical results indicate that, in practice, randomly chosen maximal paths are overwhelmingly short, due perhaps to the small-world phenomena² exhibited

¹This reduction result is attributed to Andrew Winslow of Tufts University: awinslow@cs.tufts.edu

²In a *small-world network*, the average distance between two randomly

chosen vertices grows proportionally to the logarithm of the number of vertices in the network.

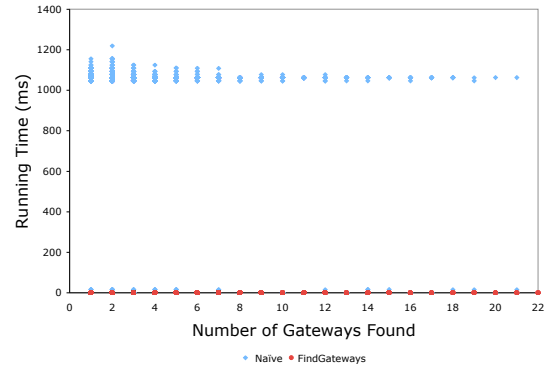


Figure 3: Running time of the random maximal path heuristic algorithm compared to a naïve application of the algorithm.

5 DISCUSSION AND CONCLUSIONS

In the above sections, we have made an argument for the utility of temporal graphs in the analysis of agent-based simulation data in political science, and have identified graph structures with strong semantic significance. It is our strong belief that this representation as well as the substructures and algorithms presented in this work are much more broadly applicable. In particular, we propose that this same formulation can be applied to other problems involving similarly large parameter spaces that can be explored in pieces. Interactive exploration of temporal graph representations can help analysts find relationships between simulation states as well as simulation runs, providing support for the development of a deeper understanding of the simulation and yielding more robust evidence for use in decision-making and policy development.

REFERENCES

- [1] S. Bender-deMoll and D. McFarland. The art and science of dynamic network visualization. *Journal of Social Structure*, 7(2), 2006.
- [2] J. Heer and D. Boyd. Vizster: Visualizing online social networks. 2005.
- [3] I. Herman, G. Melançon, and M. Marshall. Graph visualization and navigation in information visualization: A survey. *Visualization and Computer Graphics, IEEE Transactions on*, 6(1):24–43, 2000.
- [4] F. Iragne, M. Nikolski, B. Mathieu, D. Auber, and D. Sherman. ProViz: protein interaction visualization and exploration. *Bioinformatics*, 21(2):272, 2005.
- [5] D. Liben-Nowell and J. Kleinberg. Tracing information flow on a global scale using Internet chain-letter data. *Proceedings of the National Academy of Sciences*, 105(12):4633, 2008.
- [6] I. Lustick, B. Alcorn, M. Garces, and A. Ruvinsky. From Theory to Simulation: The Dynamic Political Hierarchy in Country Virtualization Models. In *American Political Science Association (APSA) 2010 Annual Meeting*, 2010.
- [7] J. Moody, D. McFarland, and S. Bender-deMoll. Dynamic network visualization1. *ajs*, 110(4):1206–41, 2005.
- [8] P. Shannon, A. Markiel, O. Ozier, N. Baliga, J. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498, 2003.

chosen vertices grows proportionally to the logarithm of the number of vertices in the network.