Agent-Based Models (ABMs) are an emerging simulation paradigm for modeling complex systems, comprised of autonomous, possibly heterogeneous, interacting agents. The utility of ABMs lies in their ability to represent such complex systems as self-organizing networks of agents. Modeling and understanding the behavior of complex systems usually occurs at large and representative scales, and often obtaining and visualizing simulation results in real-time is critical.

The real-time requirement necessitates the use of in-memory computing, as it is difficult and challenging to handle the latency and unpredictability of disk accesses. Combining this observation with the scale requirement emphasizes the need to use parallel and distributed computing platforms, such as MPI-enabled CPU clusters. Consequently, the agent population must be "partitioned" across different CPUs in a cluster. Further, the typically high volume of interactions among agents can quickly become a significant bottleneck for real-time or large-scale simulations. The problem is exacerbated if the underlying ABM network is dynamic and the inter-process communication evolves over the course of the simulation. Therefore, it is critical to develop topology-aware partitioning mechanisms to support such large simulations.

In this dissertation, we demonstrate that distributed agent-based model simulations benefit from the use of graph-partitioning algorithms that involve a local, neighborhood-based perspective. Such methods do not rely on global accesses to the network and thus are more scalable. In addition, we propose two partitioning schemes that consider the bottom-up individual-centric nature of agent-based modeling. The first technique utilizes label-propagation community detection to partition the dynamic agent network of an ABM. We propose a latency-hiding, seamless integration of community detection in the dynamics of a distributed ABM. To achieve this integration, we exploit the similarity in the process flow patterns of a label-propagation community-detection algorithm and self-organizing ABMs.

In the second partitioning scheme, we apply a combination of the Guided Local Search (GLS) and Fast Local Search (FLS) metaheuristics in the context of graph partitioning. The main driving principle of GLS is the dynamic modification of the objective function to escape local optima. The algorithm augments the objective of a local search, thereby transforming the landscape structure and escaping a local optimum. FLS is a local search heuristic algorithm that is aimed at reducing the search space of the main search algorithm. It breaks down the space into sub-neighborhoods such that inactive sub-neighborhoods are removed from the search process. The combination of GLS and FLS allowed us to design a graph-partitioning algorithm that is both scalable and sensitive to the inherent modularity of real-world networks.
The public is welcome to attend.