Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

EVALUATING PARALLELIZATION STRATEGIES FOR LARGE-SCALE INDIVIDUAL-BASED INFECTIOUS DISEASE SIMULATIONS

Johannes Ponge
Dennis Horstkemper
Bernd Hellingrath

Department of Information Systems
University of Münster
Leonardo-Campus 3
Münster, NRW 48149, GERMANY

Lukas Bayer
Wolfgang Bock

Department of Mathematics
RPTU Kaiserslautern-Landau
Gottlieb-Daimler-Straße 48
Kaiserslautern, RP 67663, GERMANY

André Karch

Institute of Epidemiology & Social Medicine
University of Münster
Domagkstraße 3
Münster, NRW 48149, GERMANY

ABSTRACT

Individual-based models (IBMs) of infectious disease dynamics with full-country populations often suffer from high runtimes. While there are approaches to parallelize simulations, many prominent epidemic models exhibit single-core implementations, suggesting a lack of consensus among the research community on whether parallelization is desirable or achievable. Rising demands in model scope and complexity, however, imply that performance will continue to be a bottleneck. In this paper, we discuss the requirements and challenges of parallel IBMs in general and the German Epidemic Micro-Simulation System (GEMS) in particular. While the exploitation of unique model characteristics can yield significant performance improvement potential, parallelization strategies generally necessitate trade-offs in either hardware requirements, model fidelity, or implementation complexity. Therefore, the selection of parallelization strategies requires a comprehensive assessment. We present a point-based evaluation scheme to assess the potential of parallelization strategies as our main contribution and exemplify its application in the context of GEMS.

1 INTRODUCTION AND BACKGROUND

The COVID-19 pandemic has demonstrated that individual-based infectious disease models can yield a profound contribution to public health decision-making and intervention planning as they allow for modeling complex social interactions and individual behavior (Lorig et al. 2021). Several models such as the GEPOC model (Bicher et al. 2020) or the ICM model (Niedzielewski et al. 2022) even simulate the population of entire countries to account for regional demographic heterogeneity and interregional mobility which pose a significant benefit over ordinary differential equation-based infection models. The models cover the entire populations of Austria (~8 million individuals) and Poland (~38 million individuals).

The recently founded OptimAgent consortium, a team of 14 German and international research institutions and partners funded by the German Federal Ministry of Education and Research, is currently working on GEMS. GEMS is intended to be a flexible individual-based infectious disease modeling

framework that covers the full German population (roughly 84 million). The framework is meant to provide model-driven public health decision support enabling the evaluation of non-pharmaceutical intervention measures to contact-communicable infectious disease outbreaks. GEMS should be easily adaptable to newly emerging pathogens, epidemic-, and endemic scenarios. The foundation of the microsimulation is a realistic synthetic population, including features such as age, sex, preexisting health conditions, household compositions, and assignments to schools and workplaces. Moreover, the model includes realistic age-dependent contact structures, individual behavioral adaptation mechanisms (e.g., as a reaction to an infection or the emergence of collective behavior like mask-wearing), realistic disease progressions, and comprehensive intervention planning.

Given the size and complexity of the model and anticipating its application as a (timely) decision support tool during infectious disease outbreaks, the simulation runtime will be one of the most critical success factors. Especially the simulation of human mobility and individual trajectories requires extensive computational resources, as shown in agent-based mobility models like the much-cited MATSim (Axhausen et al. 2016). Although respiratory viral infections are enforced by social contacts in close physical proximity, the driving factors are not necessarily random outdoor (open-air) encounters. A 2021 review study finds that indoor transmissions for COVID-19 were more than 18 times more likely than outdoor transmissions (Bulfone et al. 2021). Thus, a simulation can assume an epidemiologically relevant connection of individuals (e.g., via the same workplace) without moving them to these physical locations. We rather situate them in so-called settings (households, workplaces, schools, etc.) in which they might interact. Commuting and long-distance trips can hereby be modeled by mapping individuals to a transportation hub setting (e.g., train stations or airports) or sampling interregional contact candidates based on travel data. Therefore, GEMS can be considered as an individual-based microsimulation model with a superimposed network structure.

However, in its current state, this model architecture is limited to the execution within a single thread. In contrast, a parallelized execution may enable a speed up of the currently planned implementation, also leaving performance headroom for future extensions and to achieve scalability. There is a large body of literature discussing various strategies to parallelize individual-based models (Fujimoto 1990; Parry and Bithell 2012; Fachada et al. 2017) and even in the context of infectious disease simulations. For example, Perumalla and Seal (2012) achieve significant performance improvements using *an Optimistic Discrete Event Simulation* approach (discussed in more detail in Chapter 2.4) on a 65,536-core Cray supercomputer. Yet, many large-scale individual-based models that played an active role in supporting European COVID-response strategies such as Bicher et al. (2020) in Austria, Adamik et al. (2020) in Poland, and Ferguson et al. (2020) in the UK, do not parallelize individual simulation runs. Preceding discussions with select members of Austrian and Polish teams suggested that the principal reasons against single-run parallelization were that the legacy code of established models would require substantial refactoring and revalidation, parallelization would compromise model flexibility, and a parallelized architecture would impair the model's ability to be executable on less-performant hardware, effectively enforcing simulations to be run on a supercomputer exclusively. These facilities might not be permanently available. However, the discussions also suggested that single-run parallelization is likely inevitable for a model the size of GEMS (~84 million individuals), combined with its anticipated complexity due to the expected runtime.

Summarizing, the preceding arguments show, that the selection of parallelization strategies necessitates the careful consideration of factors unrelated to the mere parallel performance improvements, such as the required compromises in model flexibility, single-core performance, or expected implementation complexity. Thus, a more comprehensive evaluation is required. In this paper, we therefore perform a systematic evaluation of four candidate parallelization strategies for the intended individual-based network model (three approaches stem from literature while a fourth approach has been proposed by us), which we introduce in Chapter 2. We deliberately exclude GPU-based approaches as the model complexity will likely impede effective vectorization and the expected model size will exceed standard VRAM sizes. As the main contribution of this paper, we introduce a point-based grading scheme assessing the presented parallelization strategies with respect to seven evaluation criteria (including the aforementioned

implications for modeling and development), exemplify it through an application on GEMS and explain our results in detail in Chapter 3. Due to the early stage of the research and application field, the evaluation scheme represents a first high-level analysis. We discuss our findings and offer a conclusion on our results in Chapter 4. While we do perform this evaluation for the GEMS model, we argue that our results are transferable to other individual-based epidemiological models and that our evaluation scheme can be applied in various contexts where the potential of parallelization strategies needs to be evaluated.

2 PARALLELIZATION STRATEGIES

GEMS makes use of model elements following the commonly used SEIR Model (c.f. Bjørnstad et al. 2020). Thus, individuals can have the following states: *Susceptible* (individual can be infected), *Exposed* (individual is infected, but not infectious), *Infectious* (individual can infect others) and *Recovered* (individual is not infectious and immune). GEMS considers pathogens that will be transmitted via contacts between individuals. Only the transition of individuals from *Susceptible* to *Exposed* is dependent on a contact, while an individual will otherwise transition through all states on its own. Therefore, we investigate the parallelization of the simulation of infectious contacts.

In existing approaches for individual-based models (IBM), predominantly two different types of handling times during the simulation are employed: On the one hand, the states of all individuals can be calculated and synchronized after fixed-interval time-steps. We hereby call this approach fixed-interval spatial IBMs. On the other hand, the states of the individuals can be calculated and synchronized after dynamic-interval time-steps. The dynamic-interval lengths are then calculated as the time spent between the occurrence of two events in the system. These approaches are known as discrete-event simulation (DES) (Grimm and Railsback 2005). It is possible to combine both approaches, but such combinations do not find common applications within epidemic simulation models and are therefore not considered here.

Generally, DES are seen as advantageous, as the dynamic time-intervals promise overall less calculation by skipping time-steps, in which no significant state changes of the individuals occur. This increase in performance is however gained through a more complicated implementation. Furthermore, it is oftentimes not sufficient to separate the calculation by cores within a DES. Instead, separate memories are also allocated, whereby the combination of cores and this memory is depicted as a logical process (LP) (Fujimoto 1990). A further downside of DES approaches is that multiple events may occur within a single timestep, making it impossible to analyze sequence-dependent implications of these events. However, this is not of interest for GEMS, as individuals usually only become *Infectious* several timesteps after being infected. Thus, the sequence of the infections is not changing the simulation output. The increased effort in implementation does not seem worthwhile for GEMS, as the high number of considered individuals (~84 million) will likely cause events to be created in each time-step that would be considered within a fixed-interval spatial IBM anyway As DES approaches are more commonly used in literature, it is nevertheless relevant to analyze the parallelization approaches employed within these.

In the following, we will introduce three parallelization approaches that have been employed in other epidemics-related microsimulations (Rakowski et al. 2010; Perumella & Seal 2012): Conservative Discrete Event Simulation (*CDES*), Optimistic Discrete Event Simulation (*ODES*), and Generative Turns Fixed-Time Intervals (*GTFTI*). While not all considered simulation models published their parallelization approaches, we were made aware of the used approaches through interviews with the respective simulation model developers. Generally, our descriptions are based on two survey papers depicting the state-of-the-art parallelization in IBMs (Fachada et al. 2017; Parry & Bithell 2012). Subsequently, we synthesize a fourth approach, that we consider promising for employment within GEMS: The a*ctive setting* approach for *fixed-time-intervals* (**ASFTI** – marked bold as our own development).

2.1 Discrete Event Simulation

Synchronizing the states between individuals is particularly computationally expensive when considering DES approaches, as the localized memory states within each LP need to be adapted as well. Hence, the

approaches for the parallelization of DES commonly aim to reduce the need for such synchronizations. Generally, DES parallelization approaches can be classified as either Conservative or Optimistic. Both approaches in turn can be classified as environmental-parallel, as they parallelize the simulation of a subset of the overall environment in which the individuals operate.

Naron and Wasserkrug (2007) studied the parallelization of DES-based agent-based models in the context of infectious diseases, making use of a Conservative approach utilizing time-windows. These time-windows specify when LPs can safely process events with a synchronization step. In their work Naron and Wasserkrug (2007) assume that most contacts that an individual may have occur within tightly defined geographical regions. Therefore, they argue events are distributed according to these geographical regions, i.e., every LP in their approach only handles the events of a certain geographical region.

Figure 1 shows the concept of Conservative DES (*CDES*). The LPs handle events separated by geographical regions considering all individuals mainly operating within this region for a time window W_n, where events are safe to process. Afterwards, a synchronization step is executed (possibly causing infections between LPs) before continuing with the next time window W_{n+1}. Hereby, *contacts* between individuals of different LPs can only occur during this synchronization step.

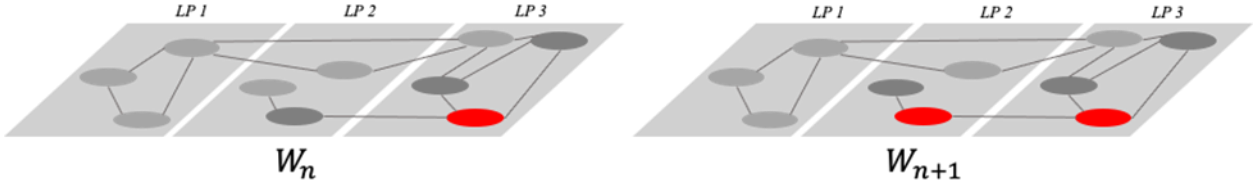

Figure 1: Conservative discrete event simulation.

In contrast to the *CDES*, the *Optimistic DES* (*ODES*) approach initially allows for state changes between LPs at any given time, but only synchronizes these at predefined time-intervals. Hence, the different LPs may assume conflicting states whenever *contacts* between LPs occur, which must be retroactively repaired to ensure a consistent behavior of the overall simulation model. For this purpose, rollback mechanisms are employed. Hereby, a commonly used rollback mechanism is the Time Warp protocol, first introduced by Jefferson (1985), which assigns each LP its own virtual time. A rollback of an LP occurs when mismatching states are recognized. When executing a rollback, every event (and hence *contact*) that occurred during the simulation is reset back to a time point before the mismatching states occurred. Therefore, each LP must keep a history of the preceding model states in memory, requiring further memory capacities. Figure 2 demonstrates the *Optimistic* approach. Every LP is calculated for its own virtual time T_i, where the timesteps can change independently from each other. Furthermore, when the global time of the simulation model advances (usually by defining the time, in which no further rollbacks can occur anymore), the virtual times will have changed by an independent increment k_i.

There are several variations of the Time Warp protocol for certain model types. We focus on spatial IBMs, for which Görür et al. (2019) proposed a variant of the Time Warp mechanism to reduce unnecessary rollbacks, improving the *ODES* approach. Furthermore, analogous to the *CDES* approach, every LP can be separated by geographical regions to reduce the *contacts* that occur between LPs.

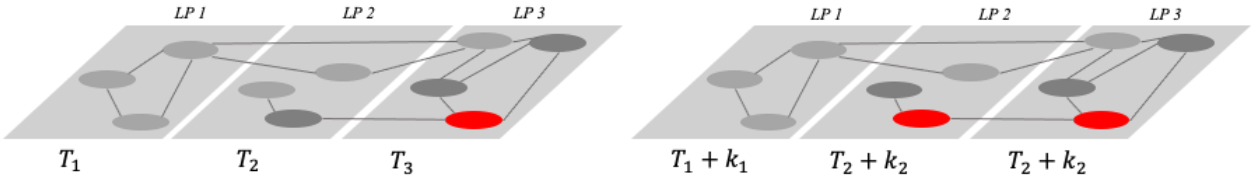

Figure 2: Optimistic discrete event simulation.

2.2 Fixed-interval Spatial Individual-based Modeling

Unlike in DES, where all identified approaches follow an *environmental-parallel* approach, Fixed-interval spatial IBMs are commonly also following an *agent-parallel* approach (Parry and Bithell 2012). Hereby, the individuals themselves are distributed to be handled by different cores, as opposed to the geographical regions in an *environmental-parallel* approach. We refer to the *agent-parallel* approach as *Generative Turns Fixed-Time Intervals* (*GTFTI*), where fixed-time-intervals represent the times in which the individuals states are updated. To reduce the computational effort, a separation of the individuals into *Infected* and *Susceptible* is performed. It is only needed to calculate the state change caused by one of these subsets of all individuals to be able to determine the outcome of all *contacts*. Therefore, we need to decide which subset is more computationally optimal to simulate.

When simulating all infected individuals, one initially only must simulate a very low number of individuals, as indicated within Figure 3. Here, in a timestep T_n only one individual is *Infected* and causes a second infection within the timestep T_{n+1}. However, when a high number of individuals is *Infected*, this results in a situation in which an individual that is *Susceptible* may be infected by several other individuals within the same timestep. When these individuals are processed on different cores, this also causes extensive synchronization efforts. This synchronization can be performed by having a copy of the model's state at a time T_n for every core, in which the outcomes of the different *contacts* calculated by the core are documented. These copies would be manipulated by the different cores and the state of the model for time T_{n+1} has to be obtained by synthesizing these copies into an overall state.

Alternatively, the parallelization could occur by simulating the *Susceptible* individuals. In a situation with only a few *Infected* individuals (as seen in Figure 3) this would initially cause a much higher computational effort. We can then calculate whether each *Susceptible* individual would become infected in each timestep. During a timestep, we would distribute the *Susceptible* individuals to different cores and calculate the risk of infection for each individual. Changing the state of a *Susceptible* individual during a time step T_n however could also change the infection probability of other *Susceptible* individuals. This issue can be handled by introducing a single individual copy of the overall state of the model, that represents an intermediate state between two time-steps and is continuously updated with the results of the calculations for each *Susceptible* individual. When all calculations have been performed, this copy represents the new state of the overall model after calculating the time step T_n. An alternative approach to update the IBM without the creation of a copy would be to calculate the infection probability on each core at time T_n, while only updating the status of the individuals at time T_{n+1}. As fewer copies of the overall network need to be handled when considering the simulation of *Susceptible* individuals (the synchronization issue, that a *Susceptible* individual gets infected from multiple *Infected* individuals on different cores does not occur in this case), it is overall less computationally expensive compared to simulating the *Infected* individuals, despite having to perform calculations for more individuals overall.

Figure 3: Model states for the generative turns approach for fixed-time-intervals.

As the presentation of DES-based approaches has shown, a high reduction of contacts affecting multiple cores can be achieved by performing an *environmental-parallel* assignment of individuals to cores. Hence, we introduce such an approach for *fixed-interval spatial IBMs* as well. This approach has been further extended by Bicher et al. (2020), who define specific "rooms", for which subsets of a population can be defined. Hereby, we call such rooms a *setting*. *Settings* can in turn be characterized by different types, e.g., households, workplaces, and schools. Furthermore, *settings* of the same type are disjoint, i.e., every

individual is at most assigned towards one setting of each type. The parallelization can thus be performed by computing infections inside *settings* of the same type in parallel for all *setting* types.

As we focus on the simulation of infectious contacts, we consider the notion of "active" *settings* within a parallelization strategy we have developed ourselves. We call a *setting* "active", if an infection could potentially happen inside it (i.e. it contains at least one infectious individual). Restricting the simulation to *active settings* therefore reduces the computational complexity. The a*ctive setting* approach for *fixed-time-intervals* (**ASFTI**) is depicted in Figure 4 with the *active settings* being highlighted by dashed white borders. The model state at time T_n depicts one *active setting* in the school layer and one *active setting* at the household layer, as both settings contain an infected individual. At time T_{n+1} a *setting* in the workplace layer becomes *active* because of a new infection that occurred within the *setting* on the household layer, affecting an individual that is a member of both settings. Multiple *active settings* of the same layer can be distributed and simulated on different cores due to the disjointed assignment of individuals to these *settings*.

As previously mentioned, this type of parallelization can be considered as an "*environment-parallel*" approach as regarded by Parry and Bithell (2012). They identified two main problems with such approaches, namely load balancing and non-local interactions. If infections occur outside of *settings*, these infections have to be simulated additionally and require communication between *settings*. While this poses a potential bottleneck for the *active settings* approach, if most of the infections happen in *settings*, sporadic infections become minimal and possibly negligible. This requires however the need to divide the environment into a sufficiently high number of *settings*. Furthermore, load-balancing issues can be addressed through reducing the variance of the number of individuals assigned to each setting.

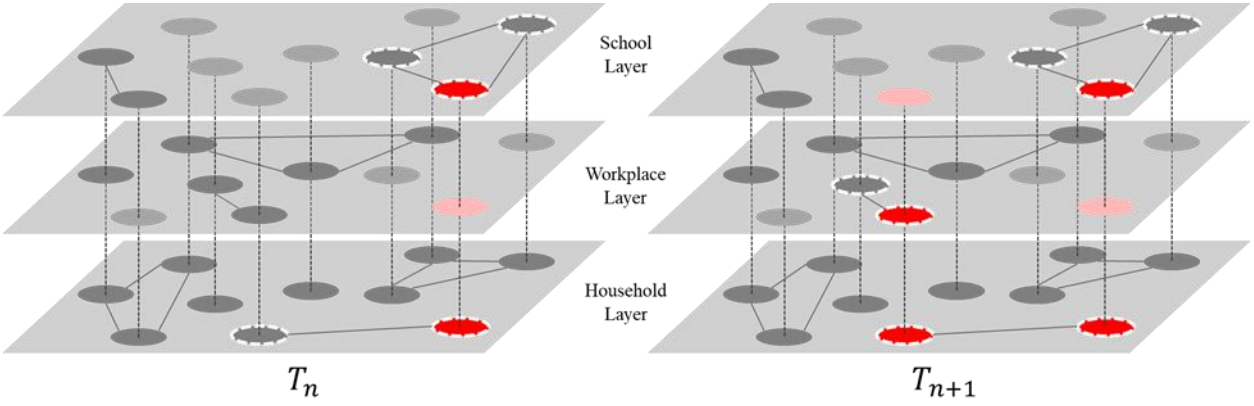

Figure 4: Active settings in a fixed-time-interval IBM.

3 EVALUATION

3.1 Evaluation Criteria

In order to evaluate the parallelization strategies, we introduce seven quality criteria: Three performance-related (Single-core Performance, Load-Balancing, Memory Efficiency), three model-related (Model Flexibility, Synchronization Overhead, Scalability), and one development-related criterion (Implementation Complexity) as seen in Table 1. These criteria were collected during a two-day systematic requirement analysis workshop among the OptimAgent project consortium where the intended application of the GEMS model and its anticipated conceptualization were discussed in detail.

3.2 Evaluation Scheme

As implementing functionally equivalent full-scale models for each of the four strategies is not realistically feasible, the parallelization potential according to the seven introduced evaluation criteria is assessed based

on careful theoretical considerations. Supporting this process, we present a point-based rating scheme, ranging from 1 (worst case) to 3 (best case). Performance criteria will achieve three points if there is little to no loss in expected runtime or memory efficiency compared to a non-parallel implementation. Two points suggest a reasonable efficiency loss that can be covered by adding capable hardware (e.g. an additional memory requirement of <100% would be considered reasonable). One point indicates a non-acceptable loss in expected performance such as a doubled required memory or severe non-preventable runtime performance losses. Among the model quality criteria, three points denote the best-case scenario which refers to no losses in flexibility (compared to the non-parallel implementation), no synchronization overhead, and full scalability with additional hardware. Two points indicate reasonable compromises in model flexibility without rendering the results unusable. A synchronization overhead might be present, but realistically coverable by means of adding computing resources. In terms of scalability, a two-point rating suggests that more hardware might incur more synchronization overhead, potentially even to a point where added resources and added overhead break even.

Table 1: Evaluation criteria.

Class	**Criterion**	**Description**
Perfor-mance	Single-core Perfor-mance	The GEMS framework is designed to run on clusters and home computers, acknowledging scenarios where users run simulations on a single core due to hardware scarcity or the need for concurrent Monte Carlo-like simulations. The Single-core Performance criterion estimates the runtime of a parallelized model on a single core. It is thus mainly an "overhead estimator".
	Load Balancing	The Load Balancing criterion assesses the ability to efficiently distribute the workload across all cores depending on the parallelization strategies.
	Memory Efficiency	The Memory Efficiency criterion assesses the expected memory overhead due to parallelization.
Model	Model Flexibility	Efficient parallelization generally requires disjunctive synchronized workloads. Partitioning a model (e.g. into geographical areas) or discretizing time (e.g. into daily timesteps) can inflict certain assumptions about the general model structure. The Model Flexibility criterion assesses the impact of such compromises induced by the proposed parallelization strategies.
	Synchroni-zation Overhead	To reliably simulate infection chains across computing nodes, synchronization is of utmost importance. The Synchronization Overhead criterion assesses the resources required to achieve such synchronization.
	Scalability	Adding computing resources in a parallel environment generally increases overhead. The Scalability criterion assesses the strategies' ability to effectively reduce runtime by adding more resources.
Devel-opment	Implemen-tation Complexity	Besides the theoretical considerations, developing parallel models is generally a challenging endeavor. The Implementation Complexity criterion assesses the severity of additionally required efforts to implement the proposed strategies.

One point in each of the categories suggests non-acceptable compromises, more overhead than performance benefits, or no scalability. Lastly, the implementation complexity criterion is evaluated with three points if

development efforts can be kept to a minimum (which includes the parallelization of certain loops, taking care of concurrent data logging, parallel random number generators, and other non-preventable features). Two points indicate reasonable effort that needs to go into code development, compared to a non-parallel implementation. One point suggests that parallelizing the code will be the major task during development and abstract the overall project from infectious disease modeling to parallel simulation development.

3.3 Results

Generative Turns Fixed Time-Interval. The *GTFTI* approach appears to be the most straightforward parallelization concept with the least expected implementation complexity (3 points). As a new iteration of the overall model is generated in each turn, solely based on the current state, we assume that very little synchronization overhead is incurred (3 points). Given that infections are evaluated from the susceptible individuals' perspective, infection routines are purely passive. Individuals are being infected rather than actively infecting. This feature allows for near-optimal segmentation for parallelization purposes as no infection can be triggered in a part of the model that is outside the computing core's assigned partition. As the approach is turn-based, overhead might be incurred due to certain cores waiting for others to finish. However, we assume this overhead can be kept to a minimum as the approach allows for very good load balancing (3 points). Since infection events are evaluated for all individuals in each turn, the overall collection of individuals can be distributed to the available cores. Minor deviation of core runtimes might occur if a particular core handles more actual infections as this usually triggers a logging event which can involve "costly" data-writing. While evaluating infections for all susceptible individuals in each turn yields excellent load balancing, it heavily impacts single-core performance (1 point). Assuming that most of the individuals will not be infected in any given turn, a large part of the calculations and thus the induced runtime will not have an impact on the simulation outcome. The strategy will require a considerable minimum number of computing cores just to break even with a non-parallel variant that handles infections more efficiently. However, due to the little synchronization overhead, we assume a very high potential for scalability (3 points). We rated model flexibility reasonable (2 points) as the turn-based nature necessarily induces fixed-time-intervals (e.g., a day). These discrete units impose certain model characteristics by assuming disease properties (such as the incubation period) are bound to be expressed in integer multiples of the step length. A 36-hour incubation period could not directly be simulated in a day-time-step model. However, as time steps can be chosen to match the model, these constraints are acceptable. A significant shortcoming of this strategy is related to memory efficiency (1 point). Since each turn results in a copy of the model, *GTFTI* requires at least twice the amount of available memory as a non-parallel implementation.

Active Settings Fixed-Time-Interval. The **ASFTI** approach deviates from the *GTFTI* in the assumption that all infections happen in self-contained settings. This prerequisite enforces a disjunct subnetwork structure across the setting types (households, schools, workplaces, and others) which allows for the parallel simulation of settings of the same type. Moreover, it provides the option to only consider settings that potentially host an infection event (active settings). In contrast to the passive *GTFTI* approach, this strategy avoids simulating a large number of non-infectious contacts without triggering infections in a partition of the model that is handled by another core. Infections can only happen within the current setting. All setting-related individuals are handled by the same core. The approach dictates that all agents are assigned to strictly disjunct settings (of the same type). No agent can be assigned to two households or two workplaces. We assessed this as a reasonable assumption in terms of model flexibility (2 points) as it is always possible to add more setting types, adding complexity to the overall network. We rate the implementation complexity as reasonable (2 points) as it suggests considerable refactoring of the model structure compared to the naive non-parallel approach. As the list of active settings can be easily assigned to computing cores, we consider the potential for load balancing very good (3 points) as well as the expected scalability (3 points). After each turn, there will be a necessary consolidation procedure that activates or deactivates settings with agents that became infectious during the current turn but in other settings (an infection at work will activate the agent's household) introducing a certain degree of synchronization overhead (2 points). This overhead will also be present in a non-parallel simulation run and thus most likely

results in noticeably reduced but acceptable single-core performance compared to a naive non-parallel approach (2 points). Implementing multiple layers of subnetworks for the same agents will lead to a larger overall model size and thus reduced but acceptable memory efficiency (2 points).

Conservative Discrete Event Simulation. Interestingly, the two discrete event simulation approaches appear to excel in contrasting categories compared to the fixed-time-interval variants. The *CDES* approach in particular will likely be very memory efficient (3 points) as it does not require a model copy or additional model layers for parallelization. The events that will have to be stored temporarily until synchronization is established are likely negligible compared to the overall model size. The strategy also retains the same level of model flexibility compared to its non-parallel counterpart (3 points) as the principal model structure is not compromised to facilitate parallelization. We assume single-core performance to be not considerably lower than the non-parallel implementation (3 points) as the introduced overhead mainly arises from core synchronization. On a single core, there is no synchronization backlog. This, however, suggests that added resources result in more synchronization overhead (2 points). As the model is sliced into segments that are assigned to LPs, more LPs will result in smaller model segments. Thus, more segments necessarily increase the need for synchronization. We would even expect that there is a natural limit for parallelization where the runtime benefit of one further attached LP and the additional synchronization overhead break even. Therefore, we assess scalability as reasonable but not optimal (2 points). Efficient load balancing will hardly be possible to achieve (1 point). The apriori model segmentation (e.g. based on geographical regions) does not take the actual workload into account. Considering that the *CDES* approach only simulates infectious agents, a fair load balancing could only be achieved if the infectious agents were evenly distributed geographically. Local infection hotspots (which are a prevalent scenario in infectious disease modeling) would cause high workloads for the core taking care of the respective geographical region while others idle (Naron and Wasserkrug 2007). The same problem applies when partitioning the model along other pivotal dimensions such as social network subgroups.

Optimistic Discrete Event Simulation. The *ODES* approach differs from the *CDES* variant in that synchronization is done upon the occurrence of divergences via messages between LPs and subsequent rollbacks of events, and not in continuous synchronizations. In the context of the proposed infectious disease model (Chapter 2.4), however, it suffers from the same load balancing issues as the *CDES* approach (1 point). Local infection hotspots might heavily occupy a single LPs while others idle. As the *ODES* variant does not have a periodic synchronization, we must assume that this can lead to significant deviations in simulation time between LPs. The update messages between LPs may then trigger extensive event rollbacks. Even worse, they might trigger recursive rollbacks, resulting in a system where anti-messages chase their initial messages indefinitely (Vee and Hsu 1999). Thus, we expect the synchronization overhead to be severe (1 point). The proposed rollback system requires that each LP stores a comprehensive list of local events that impacted the model state. A fair level of memory efficiency will hardly be achievable (1 point). While it would be theoretically feasible to dispose of events that are definitely out-of-scope (e.g., by tracking the slowest core's internal simulation clock time), this garbage-collecting process would induce further overhead. In terms of model flexibility and single-core performance, the *ODES* approach matches with the *CDES* (3 points each). The same applies to its scalability which we rate as reasonable following the *CDES* arguments (2 points). We expect this approach to yield the highest implementation complexity (1 point) due to the complex event management (Jafer et al. 2013). While the literature provides many suggestions to improve this strategy (e.g., by means of the previously mentioned memory management), optimizing the parallelization strategy would likely be the dominant task in a simulation project.

The results of our analysis are summarized in Table 2. They suggest that each parallelization strategy excels in a different set of evaluation criteria. *GTFTI*, **ASFTI**, and *CDES* achieve 16 points each, topping the *ODES* approach by four points. However, as the categories are unweighted, their relative importance in a particular project may depend on the available hardware, experience, and manpower.

Figure 5 visualizes the potential tradeoffs. With unweighted criteria, the **ASFTI** approach appears to be a good compromise as it achieves at least two points in every category. As we define a one-point rating as "non-acceptable", it would remain as the only "completely acceptable" approach. However, in a scenario

where the model is intended to run exclusively on high-performance clusters, single-core performance, and memory efficiency might be negligible. In that case, the *GTFTI* strategy is a prominent candidate as it achieves the maximum points in four out of the seven criteria. Thus, we suggest that the rating scheme might require slight adaptations in other use cases. The two discrete event simulation variants both suggest poor load balancing which is generally considered a necessity for effective parallelization. We therefore argue that *CDES* and *ODES* are no feasible candidate strategies although both strategies yield the highest potential to retain model flexibility, especially as their step-free approach does not enforce integer multiples of the step lengths for disease parameters such as the incubation period. We also observe that memory efficiency has the lowest accumulated rating acknowledging the well-known dilemma of runtime performance and memory efficiency generally depicting conflicting goals. All strategies achieve at least two points for the model flexibility and scalability criteria.

Table 2: Evaluation results.

Criteria/ Strategy	**Sg.-Core Perf.**	**Load Balanc.**	**Memory Eff.**	**Model Flex.**	**Sync. Overh.**	**Scala-bility**	**Implem. Comp.**	***Sum***
GTFTI	1	3	1	2	3	3	3	*16*
ASFTI	2	3	2	2	2	3	2	*16*
CDES	3	1	3	3	2	2	2	*16*
ODES	3	1	1	3	1	2	1	*12*
Sum	*9*	*8*	*7*	*10*	*8*	*10*	*8*	

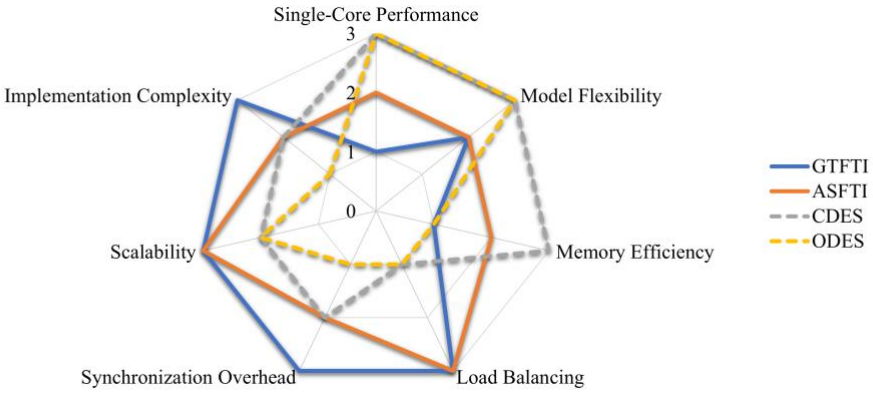

Figure 5: Visualization of tradeoffs.

4 DISCUSSION AND CONCLUSION

The fact that three out of the four parallelization strategies achieve the same score but excel in different categories supports the notion of IBM parallelization being a controversial issue and highly use-case specific. The infectious disease application context allows for certain assumptions about the model which largely influence the anticipated performance improvement potential. Especially the aspect of only actively simulating infectious individuals and their immediate surroundings drastically cuts the expected workload in most of the scenarios. At the same time, this feature inflicts severe load-balancing challenges for the evaluated parallel discrete event simulation strategies which are otherwise very prominent approaches in IBM parallelization literature (Fujimoto 1990; Perumalla and Seal 2012). Generally, we observe that parallelization strategies tailored towards multicore platforms (such as the Active Setting Fixed-Time-Interval approach) suggest fewer compromises and thus better parallelization potential over strategies that are parallelized ex-post. This perception coincides with the discussions we had with the European modelers. Many models are not parallelized because they were not designed to be parallelized.

Naturally, there are limitations to our work. We focused on the conceptual potential of different parallelization strategies as it is not within the scope of this work to implement four full-fledged functionally equivalent simulation systems. Therefore, we can only make well-reasoned assumptions about the actual performance of the evaluated strategies. Moreover, when going into the actual realization, further implementation-related criteria such as data transfer- and storage efficiency, programming skills, debugging, and testing abilities as well as hardware heterogeneity might become critical success factors. Consequently, future research should include such a comparative implementation of parallelization approaches to identify further relevant characteristics, fleshing out the evaluation scheme further. A generalization of the research results toward other kinds of microsimulations could be conducted as well.

Our work contributes to the ongoing debate about the potential of parallelizing large-scale infectious disease microsimulations. We have provided an evaluation framework, which also considers factors beyond the mere parallelization itself, focusing on the implication of the modelling of the underlying decisions. Furthermore, we advocate that our evaluation methodology is applicable in a variety of contexts, enabling practitioners and researchers to select parallelization strategies for other individual-based epidemiological models. For our specific use-case, we elaborated on the importance of scalable individual-based infectious disease model architectures and thoroughly discussed the concomitant challenges. We then briefly presented the German Epidemic Micro-Simulation System (GEMS) and proposed four applicable parallelization strategies for such a model. Furthermore, we introduced a systematic point-based approach to evaluating the potential of candidate parallelization strategies. We applied the schema to the four identified candidate strategies with respect to the GEMS model. From the results, we conclude that parallelizing large epidemiologic microsimulations remains a complex issue and that the choice of parallelization strategies is not only a matter of mere performance improvements. Naturally, it necessitates tradeoffs, not only regarding the utilization of resources (performance vs. memory efficiency) but also the implied compromises in terms of model flexibility (e.g. accepting certain abstractions to partition the model into concurrently computable sections), and compromises regarding the enforced application environment (e.g. always requiring a computing cluster). As the *ASFTI* strategy is the only approach with no 1-point rating across the evaluation criteria, we conclude that it is the most promising candidate for GEMS.

ACKNOWLEDGMENTS

This work is part of the OptimAgent project funded by the German Federal Ministry of Education and Research (BMBF); project 031L0299. We also want to thank our advisory board members Franciszek Rakowski (University of Warsaw), Tyll Krüger (Wroclaw University of Science and Technology), and Niki Popper (DEXHELPP, Austria) for sharing their long-standing experience in building large-scale individual-based infectious disease models with us and providing invaluable feedback for our concepts.

REFERENCES

Adamik, B., M. Bawiec, V. Bezborodov, W. Bock, M. Bodych, K.P. Burgard, T. Götz, T. Krueger, A. Migalska, B. Pabjan, T. Ożański, E. Rafajłowicz, E. Skubalska-Rafajłowicz, S. Ryfczyńska, E. Szczurek, and P. Szymański. 2020. "Mitigation and Herd Immunity Strategy for COVID-19 is Likely to Fail". MedRxiv, 2020-03.

Axhausen, K. W., A. Horni, and K. Nagel 2016. "*The Multi-Agent Transport Simulation MATSim*". London: Ubiquity Press.

Bicher, M. R., C. Rippinger, C. Urach, D. Brunmeir, U. Siebert and N. Popper. 2020. "Agent-Based Simulation for Evaluation of Contact-Tracing Policies against the Spread of Sars-Cov-2". *MedRxiv* 10(2020.05):12-20098970.

Bjørnstad, O. N., K. Shea, M. Krzywinski, and N. Altman. 2020. "The SEIRS Model for Infectious Disease Dynamics". *Nature Methods* 17(6): 557-559.

Bulfone, T. C., M. Malekinejad, G. W. Rutherford and N. Razani. 2021. "Outdoor Transmission of SARS-CoV-2 and Other Respiratory Viruses: A Systematic Review". *The Journal of Infectious Diseases* 223(4): 550-561

Fachada, N., V. V. Lopes, R. C. Martins et al. 2017. "Parallelization Strategies for Spatial Agent-Based Models". *International Journal of Parallel Programming* 45: 449–481.

Ferguson, N., D. Laydon, G. Nedjati Gilani, N. Imai, K. Ainslie, M. Baguelin, S. Bhatia, A. Boonyasiri, Z. U. L. M. A. Cucunuba Perez, G. Cuomo-Dannenburg and A. Dighe. 2020. "Report 9: Impact of Non-Pharmaceutical Interventions (NPIs) to Reduce COVID19 Mortality and Healthcare Demand". Imperial College, London.

Fujimoto, R. M. 1990. “Parallel Discrete Event Simulation”. *Communications of the Association for Computing Machinery* 33(10): 30–53.

Grimm, V., and S. F. Railsback. 2005. “A Conceptual Framework for Designing Individual-Based Models”. In *Individual-Based Modeling and Ecology*, edited by V. Grimm, and S.F. Railsback, 71–121. Princeton: Princeton University Press.

Görür, B. K., K. M. İmre, H. Oğuztüzün, and L. Yilmaz. 2019. “Predetermined Rollbacks: An Extension to Time Warp for Spatially Parallel Agent-Based Simulation”. *Simulation Modelling Practice and Theory* 95: 60-77.

Jafer, S., Q. Liu, and G. Wainer. 2013. “Synchronization Methods in Parallel and Distributed Discrete-Event Simulation”. *Simulation Modelling Practice and Theory* 30: 54-73.

Jefferson, D. R. 1985. “Virtual time”. *Association for Computing Machinery Transactions on Programming Languages and Systems* 7(3): 404–425.

Lorig, F., E. Johansson, and P. Davidsson. 2021. “Agent-Based Social Simulation of the COVID-19 Pandemic: A Systematic Review”. *Journal of Artificial Societies and Social Simulation* 24(3).

Naron, S., and S. Wasserkrug. 2007. "Utilizing Model Characteristics to Obtain Efficient Parallelization in the Context of Agent-Based Epidemiological Models". In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 1555-1561. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Niedzielewski, K., J. M. Nowosielski, R. P. Bartczuk, F. Dreger, Ł. Górski, M. Gruziel-Słomka, A. Kaczorek, J. Kisielewski, B. Krupa, A. Moszyński, M. Radwan, M. Semeniuk, J. Zieliński, and F. Rakowski. 2022. "The Overview, Design Concepts and Details Protocol of ICM Epidemiological Model (PDYN 1.5)." *Design Concepts and Details Protocol of ICM Epidemiological Model (PDYN 1.5)* (February 21, 2022).

Parry, H.R., and M. Bithell. 2012. “Large Scale Agent-Based Modelling: A Review and Guidelines for Model Scaling”. In *Agent-Based Models of Geographical Systems*, edited by Heppenstall, A., Crooks, A., See, L., Batty, M., 271-308. Dordrecht: Springer.

Perumalla, K. S., and S. K. Seal. 2012. “Discrete Event Modeling and Massively Parallel Execution of Epidemic Outbreak Phenomena”. *Simulation* 88(7): 768-783.

Rakowski, F., M. Gruziel, L. Bieniasz-Krzywiec, and J. P. Radomski. 2010. “Influenza Epidemic Spread Simulation for Poland — a Large Scale, Individual Based Model Study”. *Physica A: Statistical Mechanics and its Applications* 389(16): 3149-3165.

Rao, D.M. 2016. “Efficient Parallel Simulation of Spatially-Explicit Agent-Based Epidemiological Models”. *Journal of Parallel and Distributed Computing* 93–94: 102-119.

Vee, V. Y., and W. J. Hsu. 1999. “Parallel Discrete Event Simulation: A Survey”. Technical Report, Nanyang Technological University.

AUTHOR BIOGRAPHIES

JOHANNES PONGE is a Research Assistant at the Chair for Information Systems and Supply Chain Management at the University of Münster, Germany. He holds an M.Sc. in Information Systems. His work focuses on the development of simulation-based decision support systems for infectious disease mitigation and intervention. He is a developer of the German Epidemic Micro-Simulation System (GEMS). His email address is johannes.ponge@ercis.uni-muenster.de

LUKAS BAYER is a Research Assistant at the Technomathematics Group at the RPTU Kaiserslautern-Landau, Germany. He holds an M.Ed. in Computer Science and Mathematics and is one of the researchers developing the German Epidemic Micro-Simulation System (GEMS) as part of the OptimAgent project. His email address is lbayer@rptu.de

DENNIS HORSTKEMPER is a Research Assistant at the Chair for Information Systems and Supply Chain Management at the University of Münster, Germany. He holds a Ph.D. in Information Systems. His research work focuses on agent-based production planning and control in the context of Industrie 4.0. He is a developer of the German Epidemic Micro-Simulation System (GEMS). His email address is dennis.horstkemper@ercis.uni-muenster.de

WOLFGANG BOCK is a Senior Lecturer at the Technomathematics Group at the RPTU Kaiserslautern-Landau, Germany. He holds a Ph.D. in Mathematics and is PI of the developing group of the German Epidemic Micro-Simulation System (GEMS). His research focuses on complex systems and mathematical epidemiology. His email address is bockw@rptu.de

BERND HELLINGRATH is a Professor and Head of the Chair for Information Systems and Supply Chain Management at the University of Münster, Germany. His research interests deal with the broader area of modeling and simulating with a distinct focus on the context of crisis-management and humanitarian logistics. His email address is bernd.hellingrath@ercis.uni-muenster.de

ANDRÉ KARCH is a Professor of Clinical Epidemiology, and the Head of the Clinical Epidemiology Unit at the University of Münster, Germany. He holds an MD and an MSc in Epidemiology. His research in the field of infectious disease epidemiology focuses on dynamic transmission models for public health decision-making. His email address is akarch@uni-muenster.de