Activity-Based Modeling and Simulation
© Presses Universitaires Blaise-Pascal, 2010

Exploiting Spatial-temporal Heterogeneity for Agent-based Simulation of Pedestrian Crowd Behavior

Fasheng QIU[1], Xiaolin HU[1]

Abstract – *Agent-based simulation is widely used to simulate pedestrian crowd behavior. These simulations typically are implemented in a discrete time manner, where each agent decides its movement in every time step, independent of the fact that agents may move in different speeds. The non-uniform movements of agents result in a crowd system's spatial and temporal heterogeneity, which can be better exploited using a discrete event model where an agent's decision making is triggered by the changes of its external environment and/or its internal states. Motivated by the above observation, this paper systematically studies the discrete time and discrete event models of agent-based crowd behavior simulation and compares their performance results. The experiment results show that the discrete event model is able to track the crowd system's "activities" in both space and time, and thus leads to more computation efficient simulations. This work builds a ground for performance analysis for large scale agent-based crowd behavior simulations.*

Keywords – Crowd behavior simulation, Spatial-temporal heterogeneity, space resolution, discrete event modeling, discrete time modeling.

1. Department of Computer Science, Georgia State University, 34 Peachtree Street, Suite 1450, Atlanta, GA 30303.

Introduction

Agent based simulation has been widely used in simulating complex social phenomena [Davidsson, 2000], like the crowd behaviors. Many pedestrian crowd models have been developed, see, e.g., [Helbing *et al.*, 2002; Qiu and Hu, 2010]. All of them are discrete time models and featured with a virtual environment and multiple pedestrian agents. The simulations proceed in a discrete time manner, where each agent performs a decision making to decide its next movement in each time step. Such a discrete time approach makes it difficult to exploit the crowd system's spatial and temporal heterogeneity resulting from agents' non-uniform movements. To give an example, consider two agents situated in a large environment where one agent moves 10 times slower than the other. In a discrete time simulation, both agents make a movement decision in every time step. This is independent of the agent's speed. However, since one agent moves much slower than the other, intuitively one would think that the slower agent does not need to make movement decisions as frequently as the fast one. Thus a more computation efficient way of simulating the two agents is to allow the fast agent to make movement decisions more frequently and the slow agent to make movement decisions less frequently. In pedestrian crowd behavior simulations with realistic human-like behaviors, agents typically have non-uniform movements due to different individual characteristics such as moving speed, personality, the psychological states (e.g., panic, non-panic), and other factors. These non-uniform movements result in spatial and temporal heterogeneity in terms of agents' movement and decision making. Thus it is desirable to explore such heterogeneity for more computation efficient simulations.

In this paper, we present a discrete event approach to model and simulate pedestrian crowd and compare it with the discrete time model. The discrete event model uses a concept of "space resolution", which defines the threshold of an agent's position change in the environment, to decide the frequency of an agent's decision making. With the space resolution, an agent's position change less than the space resolution threshold does not trigger its decision making. Nor does it trigger the message passing from the agent to others. As a result, agents that move slowly make movement decisions less frequently than the fast agents. This concept of "space resolution" is derived directly from the quantization and activity concepts presented in [Zeigler *et al.*, 2004]. The value of the space resolution has significant impacts on the crowd behavior simulation. On one hand, the larger the space resolution is, the less frequently agents make decisions, and thus the more efficient the simulation is. On the other hand, the space resolution means that an agent does not update its position until its position change bypasses the space resolution threshold. This introduces position errors in the crowd simulation. The larger the space resolution is, the larger the position errors are, and thus the less accuracy the simulation is. We point out that similar kind of relationships also exists in a discrete time model, whose efficiency and precision depend on the value of the time step. A main effort of this work is to establish a formal "fair-comparison" rule that quantifies the position errors of both the discrete time and discrete event models, and conduct experiments from different aspects to compare the two. Note that both the space resolution in the discrete event model and the time step in the discrete time model are global variables shared by all agents. In this work, we use the number of decision making as an indicator of simulation performance. This is based on the observation that an agent's decision making usually involves complex logics, and thus accounts for the most significant part of computation in a simulation. We carry out this work based on the DEVS [Zeigler *et al.*, 2000] modeling and simulation framework, in particular the DEVSJAVA environment[2]. The DEVS framework was chosen due to its formal formalism and its capability of modeling both the discrete time and discrete event models. Nevertheless, we note that the model design and the conclusions drawn in this research are general and do not rely on the DEVS framework.

The remainder of this paper is organized as follows. Section 1 introduces some related work of the pedestrian crowd simulation, and discrete event models. Section 2 presents the crowd system, the discrete event agent and discrete time agent models of the crowd system. Section 3 presents a quantitative analysis of the two models. Section 4 shows experiment results from three experiments. Finally we conclude this work and propose some future directions.

1. Related work

Pedestrian crowd simulation has been studied for a long time. A well known model is Helbing's physics and social force model [Helbing *et al.*, 2002] where the behavior is described as the vector addition of the separate force

2. www.acims.arizona.edu.

terms reflecting different environmental influences. This model has successfully simulated several important features of crowd behavior, such as lane formation in crowds with opposite walking direction, oscillations of the crowd passing direction at a bottleneck, alternating collective patterns of motion at intersections and so on. Kaup's work [Kaup and Clarke, 2007] extends Helbing's model to produce more realistic behavior of an individual pedestrian under panic or non-panic conditions.

Crowd behavior is also simulated for studying emergency evacuations and safe egress. [Pan *et al.*, 2006] developed a prototype system to study some emergent human and social behaviors, such as competitive, queuing, and herding behaviors, during emergency evacuations. The work of [Pelechano *et al.*, 2007] provided a High-Density Autonomous Crowds (HiDAC) system which can be used to simulate a wide variety of emergent behaviors such as lane formation, pushing behavior and so on, by applying a set of psychological and geometrical rules with a social and physical forces model. S.R. Musse [Musse and Thalmann, 2001] proposed a hierarchy crowd model with three different ways for controlling human behaviors. The work of [Qiu and Hu, 2010] provided a common framework to model the group structures in pedestrian crowds and several group structured are demonstrated. [Seck *et al.*, 2005] proposed a dynamic personality filter which is used to model more realistic human behavior.

Most of these crowd behavior models adopt a discrete time based simulation, which is featured with a series of discrete time steps. In each time step, each pedestrian agent makes a decision to decide the next movement, based on its internal states and external environment. Such a discrete time based approach treats the crowd a uniform entity, which makes it difficult to exploit the heterogeneity of the crowd system. Thus, this work proposes a discrete event based approach to achieve more computation efficient simulation. To our knowledge, there are several works focusing on building agent based social system on the discrete event based paradigm. Dubiel's work [Dubiel and Tsimhoni, 2005] integrated ABS with DES to model humans traveling freely through a real-world problem from the theme park industry. [Zaft and Zeigler, 2002] developed a Sugarscape-style artificial society based on the Discrete Event System Specification (DEVS) formalism. However, little literature has been reported about exploring the effect of spatial and temporal heterogeneity on the agents' decision making and communication. This is what we want to achieve in this paper.

2. Discrete Event and Discrete Time Model of Crowd Behavior Simulation

A crowd system contains a simulated virtual world and multiple pedestrian agents. Before introducing the details of the discrete event and discrete time agent models studied in this paper, we briefly describe the major parameters of agents and the environment. In the following sections, to save space, the discrete event based agent model and the discrete time based agent model will be stated as *DES model* and *DTS model* respectively.

A crowd consists of a virtual environment, obstacles and multiple pedestrian agents. In this paper, the virtual environment is a rectangle area which is measured with a *width* and *length*. There are two categories of agents, *Obstacles* and *Virtual pedestrians*. *Obstacles* are stationary rectangle objects in the environment that block agent's movement. *Virtual pedestrians* are a set of autonomous pedestrian agents that move in the environment while avoiding collisions with obstacles and other agents. Each pedestrian agent is featured with a perception model and behavior control model. The perception model defines an elliptical area in front of the agent where the agent can perceive obstacles and other nearby agents. The behavior control model uses a bio-inspired behavior control architecture which decides the agent's movement (see [Qiu and Hu, 2008] for more details). In this work, each pedestrian agent has two behaviors:

1. *Move:* This behavior is used to simulate the casual movement of each agent. An agent moves to pre-defined destinations or randomly generated destinations in a sequential order. The moving path is the shortest path from the current position to the destination. And when a destination is reached, the agent moves to the next destination. In this work, the sequence of destinations is defined explicitly to ensure both the *DES* and *DTS* models use the same set of destinations for the purpose of fair comparison between them.
2. *Avoid*: This behavior is used to simulate the obstacle avoidance in the movement. When an agent is within a predefined minimum distance from the nearest neighbor agent or obstacle, it will stay away from it. The action of this behavior is as follows: if the agent is on the left side of the avoiding object, it turns right with an angle; otherwise, it turns left. In this process, a basic "collision prediction" subroutine is used to predict if the current computing agent will collide with other agents

once the turn is finished. If the subroutine returns true, the agent will try other angles recursively.

Each pedestrian agent has an *ID* that defines the global unique identification of the agent, and *Speed* that is the agent's moving speed. Two more important parameters of each agent are *SpaceResolution* and *TimeStep.* For the *DES* model, *SpaceResolution* defines the position change threshold of the agents. As described above, the larger the space resolution is, the less frequent the agents' decision making is. For the *DTS* model, *TimeStep* defines the time step of the simulation. It also affects the decision making frequency of the agents. The larger the time step is, the less frequent the agents' decision making is. As will be discussed later, both the space resolution and time step introduce position errors in the crowd simulation.

2.1. The Discrete Time Model

The *DTS* model is straightforward to understand. At every time step, each agent checks its environment (for example, if a destination is reached or if there are other agents in nearby locations), makes a decision to decide its next movement (e.g., move forward, or move sideways to avoid collision), and then carries out the movement for this time step. This will change the agent's position. Thus at the next time step, the agent goes through the same sequence again to check its environment, make a decision, and carry out the movement.

Each agent is implemented as a *DEVS* atomic model and has two states "decision_making" and "update_position". At the "decision_making" state, the agent checks its environment and makes a decision to choose a movement action. After that, the agent transits to the "update_position" state where its position is updated. The above procedure is performed for each time step. The procedure is implemented in the internal transition function *deltint()* of an agent. The pseudo-code is shown in Fig. 1.

```
procedure deltint()
  if the agent's current state is "decision_making"
        check the environment;
        Action a ← perform a decision making;
        holdIn("update_position", Timestep);
  else if the agent's current state us "update_position"
        update position based on section a;
        holdIn("decision_making", 0.0);
  end if.
end procedure deltint.
```

Figure 1. Internal transition function of the DTS model.

2.2. The Discrete Event Model

Unlike the *DTS* model, an agent in the *DES* model does not make a decision at every time step. Instead, the decision making is based on the "change" of the environment and/or the "change" of the agent's own position (according to the space resolution). Whenever such a change happens, an agent checks its environment and makes a decision to choose a movement action. Based on that action, it calculates the duration (space resolution divided by the agent's current moving speed) of the action, and after that duration elapses it performs the action (*move* or *avoid*) to update its position. This means that an agent does not update its position unless the position change equals to the space resolution. When an agent updates its position, it also notifies its nearby agents because this represents a "change" in the environment for those agents. Once those agents receive the message, they carry out the same sequence, *i.e.*, check environment, make a decision, perform action, and notify nearby agents, as described above. Note that because agents constantly move, the couplings between an agent and its nearby agents is set up dynamically based on agent's positions.

Each agent is modeled as a *DEVS* atomic model which includes an external transition function, an internal transition function, a confluent transition function (internal transition function first and then the external transition function) and an output function. These functions are used to decide the agent movements and inter-agent communication. Figure 2 shows the state transitions of an agent. In the "active" state, an agent checks its environment and makes a decision. In "move or avoid" state, the agent performs the action and carries out the movement. While in "message" state, the agent notifies the neighborhood agents about its new position.

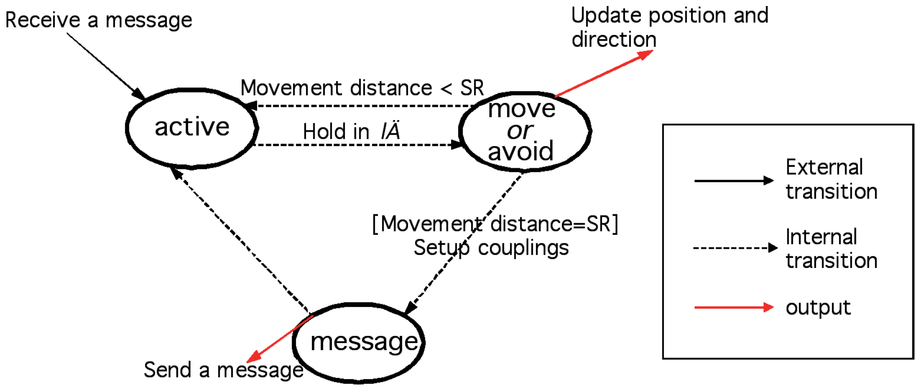


Figure 2. Agent state transition diagram.

Initially, the agent stays at the "active" state where the agent performs a decision making through the behavior control model to decide the next movement. The result is an action which indicates where the agent should move to. After holding in duration δ, the agent transits to "move or avoid" state where the agent moves to the new position. The agent then goes to the "message" state in order to send its new position information to the neighborhood agents if the movement distance is equal to the agent's space resolution *SR*. Before sending the position information, the couplings between the agent and the neighborhood agents are setup. And after the message is sent, the agent returns back to the "active" state. After each movement or when a new message is received from neighborhood agents, the agent performs a new decision making which continues the procedure mentioned above. The following functions implement this state transition diagram.

The external function is used to capture the messages sent from other agents. When a message is received, it transits to the "active" state. The procedure is shown in Fig. 3.

```
procedure deltext(e, x)
   if message received in x
        holdIn("active", 0.0);
   end if.
end procedure deltext.
```

Figure 3. External function.

In the internal function, the agent performs a decision making if the current state is "active". Otherwise, it performs the movement, updates its position and sends the new position to the neighborhood agents if the movement distance is greater than its space resolution. The procedure is shown in Fig. 4.

```
procedure deltint()
  if the agent's current state is "active"
        update the position;
        check the environment;
        Action a ← perform a decision making;
        holdIn(a.name, a.duration);
  else
        if the agent's current state is not "message"
              perform the action and update the position;
              if movement distance greater than SR
                    setup couplings with the neighborhood agents;
                    holdIn("message", 0.0);
                    return;
        else
              holdin("active", 0.0);
  end if.
end procedure deltint.
```

Figure 4. Internal function.

The output function is used to send messages to the neighborhood agents if the current state is "message". The procedure is described in Fig. 5.

```
procedure output()
  if the agent's current state is "message"
        send a message to the coupled agents;
  end if.
end procedure output.
```

Figure 5. Output function.

3. Analysis of the DES and DTS Models

Both the *DES* and *DTS* model introduce imprecision, also referred to as *position error* in this paper, in modeling agents' position updates. For the *DTS* model, an agent's position will not be updated until the time step is reached. Within a time step, an agent's position is considered as unchanged. For the *DES* model, an agent's position will not be updated until the space resolution threshold is reached. Any position change within the space resolution threshold is not captured. This section analyzes the position error introduced by the *DES* and *DTS* models and studies their relationship. The goal is to build a ground for comparing the *DES* and *DTS* models and showing how the *DES* model can exploit the spatial-temporal heterogeneity of the crowd system.

We compare both the *DTS* and *DES* models with an analytic model for an agent's position update. Assume there are n agents in the crowd and the simulation is running over the time base [$t1$, $t2$], where $t1$ is the starting time and $t2$ is the ending time. Assume the position at time $t1$ is known of all agents. This position is also called *initial position*. Using the analytic model, an agent j ($1 \le j \le n$)'s position at time t ($t1 \le t \le t2$) is calculated through Eq. 1.

$$\vec{p}_{t,j} = \vec{p}_{t1,j} + \int_{t1}^{t} \vec{v}_{t,j}\, dt \qquad (1)$$

In both *DES* and *DTS*, an agent's position is updated discretely. Eq. 2 and Eq. 3 represent the position update of the *DTS* and *DES* models respectively, where Δt is the time step of the DTS simulation and *SR* is the space resolution of the DES simulation. In the *DTS* model, Eq. 2 shows that before the time step t is reached, the agent j's position $\vec{p}_{t,j}$ is not changed. Thus, compared with the analytical model, between the time step t - 1and t, there is an error of agent j's position. Similarly, in the *DES* model, Eq. 3 shows that an agent's position will not be updated until the space resolution threshold is reached. Note that in Eq. 3, P_{t-1} should be interpreted as the agent's previous position, instead of the position at time t - 1.

$$\vec{p}_{t,j} = \begin{cases} \vec{p}_{t-1,j} & \text{if } \mathrm{t < t - 1 + \Delta}t \\ \vec{p}_{t-1,j} + \int_{t-1}^{t} \vec{v}_{t,j}\, dt & \text{if } \mathrm{t = t - 1 + \Delta t} \end{cases} \qquad (2)$$

$$\vec{p}_{t,j} = \begin{cases} \vec{p}_{t-1,j} & \text{if} \left| \int_{t-1}^{t} \vec{v}_{t,j} dt \right| < SR \\ \vec{p}_{t-1,j} + \int_{t-1}^{t} \vec{v}_{t,j} dt & \text{if} \left| \int_{t-1}^{t} \vec{v}_{t,j} dt \right| = SR \end{cases} \quad (3)$$

In order to make a fair comparison between *DES* and *DTS* models, the following condition should be satisfied: ***The maximum position error in both DES and DTS models is same.*** In *DTS* model, from Eq. 2 the maximum position error is $\int_{t-1}^{t-1+\Delta t} \vec{v}_{\max} dt$. Here $v_{\max}$ is the maximum moving speed among all agents. While in *DES* model, from Eq. 3 the maximum error is the space resolution *SR* of agents. Thus, Eq. 4 holds when the *DES* and *DTS* model are compared.

$$SR = \left| \int_{t-1}^{t-1+\Delta t} \vec{v}_{\max} dt \right| \quad (4)$$

When the moving speed of an agent is constant during a time step, Eq. 4 can be simplified to Eq. 5 shown below. In the following, we use *TS* to represent the time step of the *DTS* model.

$$SR = V_{\max} * TS \quad (5)$$

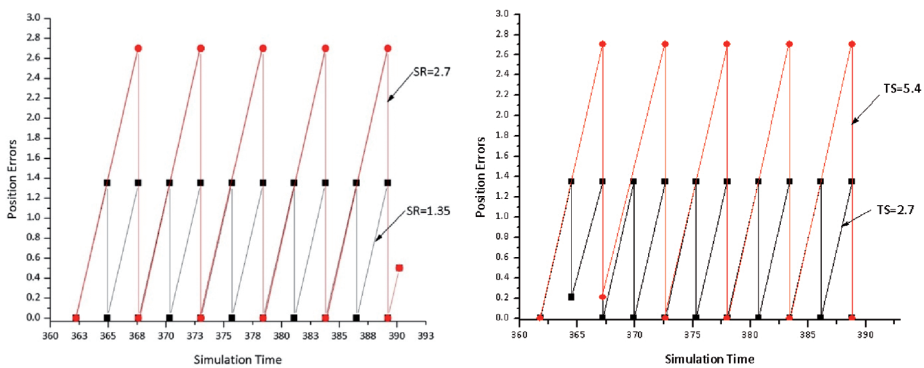
Eq. 5 is used as a basis in this work for comparing the *DES* and *DTS* models. In the next section, three experiments are used to compare the two models, and to show how the *DES* model can result in more efficient computation by exploiting the spatial-temporal heterogeneity of crowd behavior.

4. Experiment Results

This section compares the *DES* and *DTS* models from different aspects, including position errors, number of decision makings and execution time. It intends to show that the *DES* model can achieve a more efficient simulation than the *DTS* model through comparing the execution time and number of decision makings of both models under the fair comparison condition described in section 3. The first experiment compares the maximum position errors of both models. The second experiment shows that the computation in *DES* model is less than that of *DTS* model if there is a non-uniform movement in the crowd. And the last experiment further shows that the *DES* model is more efficient than the *DTS* model if there is speed heterogeneity in the crowd.

4.1. Experiment 1

In this experiment, position errors of both *DES* and *DTS* models are compared, and the number of decision makings is explored for the simulation with one agent whose moving speed is 0.5. The agent moves through a series of destinations which are pre-defined. When all destinations are arrived, the simulation stops and position errors and the number of decision makings are calculated. Here, the position errors are based on the comparison between the *DES*/*DTS* model and the analytic model (see Eq. 1). Fig. 6(a) and Fig. 6(b) show the position errors of the two models under two space resolutions *SR* and two time steps *TS*. As described in Section 3, to ensure fair comparisons between the two models, for a specific space resolution SR in a DES simulation, the corresponding time step TS in a *DTS* simulation is calculated as TS = SR/v, where v is the agent's moving speed (0.5 in this experiment).



(a) DES model (b) DES model

Figure 6. Position errors in *DES* model and DTS model (SR = 1.35 and 2.7, correspondingly TS = 2.7 and 5.4).

Fig. 6(a) shows the agent position errors in *DES* model under two space resolutions 1.35 and 2.7. X-axis represents the simulation time. Y-axis indicates the position error at different time. For *SR* = 1.35 the agent updates its position at time 2.7*N (N = 1, 2, 3…) since the moving speed is 0.5. And the position error is increasing linearly between two position updates. Similarly, for SR = 2.7, the agent updates its position at time 5.4*N (N = 1, 2, 3…). Fig. 6(a) shows that the greater the space resolution, the greater the position error the agent will have, and the less the simulation accuracy is. Fig. 6(b) shows the position errors in the *DTS* model under two time steps 2.7 and 5.4. It shows that the position error increases linearly between two time steps. For the time step 2.7, the maximum position error is 1.35. Note that when the agent approaches a destination (*i.e.* the time step 364.5), there exists position error because in our implementation if the agent is near the destination within a specified distance range, we consider that the agent has arrived at that destination. Fig. 6 confirms that both the *DES* and *DTS* model introduce position errors in agents' position update. To make a fair comparison, the maximum position error in both models should be the same. The *DES* model with *SR* = 2.7 and the *DTS* model with *TS* = 5.4 can be fairly compared since the maximum error in both models is same. Similarly, the *DES* model with *SR* = 1.35 and the *DTS* model with *TS* = 2.7 can be fairly compared because of the same maximum error.

Besides the maximum position error, the space resolution of a DES model also affects the number of decision makings the agent will perform. Fig. 7 shows the relationship between space resolution and the number of decision making the agent has performed in the *DES* model for different space resolutions. Here the number of decision makings calculated as the number of times the agent is in the "active" state since a decision making is performed whenever the agent is at that state (see Section 2 for more details).

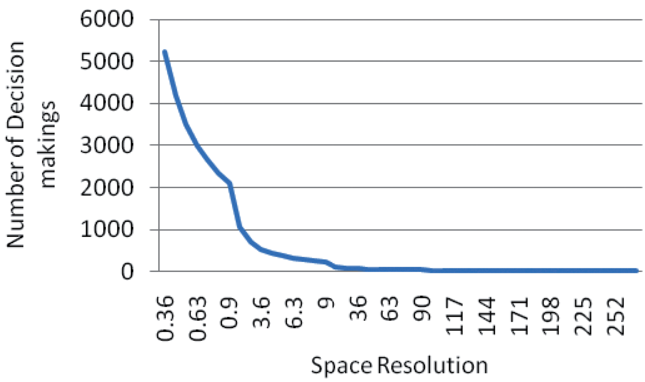


Figure 7. Relationship between space resolution and number of decision makings of *DES* model.

In Fig. 7, X-axis represents the space resolution and Y-axis represents the number of decision makings the agent has performed to traverse all predefined destinations. The number of decision makings decreases as the space resolution increases. This is because the greater the space resolution, the longer distance the agent can move in one step. Since the total distance the agent shall traverse is the same among different space resolutions, the longer distance the agent can move in one step, the fewer decision makings the agent needs to perform to finish all these destinations. As a result, one can increase space resolution to reduce the number of decision makings for more efficient computation. However, as shown by Fig. 6, the greater the space resolution, the greater the position errors are. Thus, one needs to consider the tradeoff between computation efficiency (number of decision making) and simulation accuracy (position error) and choose the space resolution in a balanced way.

4.2. Experiment 2

In experiment 1 the relationship between the number of decision makings and space resolution is explored for the simulation with one agent. Experiment 2 further compares the number of decision makings in both models for both *uniform* (*i.e.* agents with same moving speed) and *non-uniform* (*i.e.* agents with different moving speed) pedestrian crowds. The goal is to show that the number of decision makings in *DES* model is fewer than that of *DTS* model if the agents have *non-uniform* movements. In other words, the computation in *DES* model is less than that of *DTS* model, thus the *DES* model is more efficient than the *DTS* model. Otherwise, if the crowd has a *uniform* movement, the number of decision makings in *DES* model is the same as that of *DTS* model.

Three agents are used in this environment. The simulation time is 120000. Two cases are experimented. One is a *uniform* crowd where all agents have the same moving speed 0.5; the other is a *non-uniformed* crowd where agents have different moving speeds (0.005, 0.05 and 0.5 respectively). Fig. 8 shows the relationship between space resolution (*SR*) (or time step *TS*) and the average number of decision makings of the *uniform* crowd. The results of the *non-uniform* crowd are shown in Fig. 9. To ensure a fair comparison, *TS* is *2* SR* since the maximum moving speed of the agents is 0.5. The average number of decision makings is the total number of decision makings during the simulation divided by the number of agents.

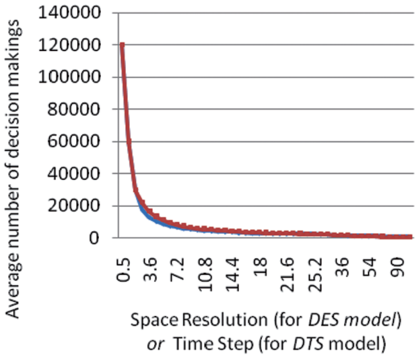


Figure 8. SR/TS and decision makings (same speed).

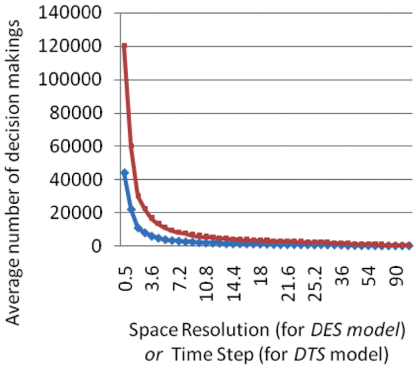


Figure 9. SR/TS and decision makings (different speed).

Fig. 8 shows that agents of *DES* and *DTS* models perform almost the same number of decision makings for all space resolutions. This is because in *DES* model, each agent in each time step will perform a decision making which is the same as the case in *DTS* model. However, for the *non-uniform* crowd (Fig. 9), *DES* model performs fewer decision makings than the *DTS* model. In other words, for the *non-uniform* crowd, under the fair comparison condition where two simulations have the same maximum position error, the *DES* model is computation more efficient than the *DTS* model. This is because in the *DTS* model, all agents need to make a decision in every time step. Thus a slow agent makes the same number of decisions as a fast agent. However, in a *DES* model the slow agent makes less number of decisions than a fast agent. To better see this, Fig. 10 shows in the case of *non-uniform* crowd, the relationship between space resolutions (or time steps) and the number of decision makings of each individual agent. Each agent in *DTS* model performs the same number of decision makings since each agent performs a decision making in each time step (shown by the curve indicated by "*DTS*"). The other three curves represent the three agents in *DES* model. As can be seen, the faster the agent moves, the larger number of decision makings the agent performs, since the faster agent performs decision makings more frequently than the slower ones. Fig. 9 and Fig. 10 show that in a *non-uniform* crowd, the *DES* model requires less computation than the *DTS* model. This will be further illustrated in Experiment 3.

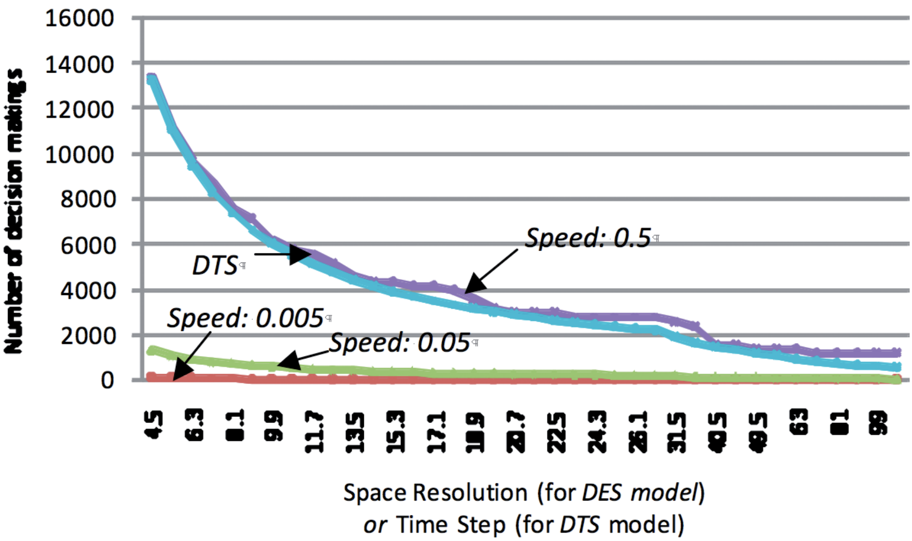


Figure 10. SR/TS and individual agents' decision makings (different speed).

4.3. Experiment 3

As an important research problem, *evacuation* of pedestrian crowds under emergent situations such as fire alarms has been studied for a long time [Helbing *et al.*, 2002]. Thus, in this experiment, we study the effect of space resolutions on decision makings and system "activities" under emergent situations. Here, activities refer to the distribution of the crowd system's decision makings in the environment. A snapshot of an emergent evacuation is shown in Fig. 11. In the simulation, agents move randomly in the environment and after a specified time, an emergency starts and all agents escape out of the environment with their speeds doubled. The system's spatial "activities" are shown in Fig. 12. As can be seen, more "activities" are distributed near the exit area since more decision makings are performed under the emergent situation and all agents escape from the exit with their speed increased.

Besides spatial heterogeneity, this system's activities also exhibit temporal heterogeneity. This is illustrated in Fig. 13, which shows the average number of decision makings of 64 simulation time intervals (*i.e.* interval *i* is [100**i*, 100*(*i*+1))) of a simulation. The simulation contains a non-uniform crowd with 20 agents where the moving speeds 0.00005 and 0.005

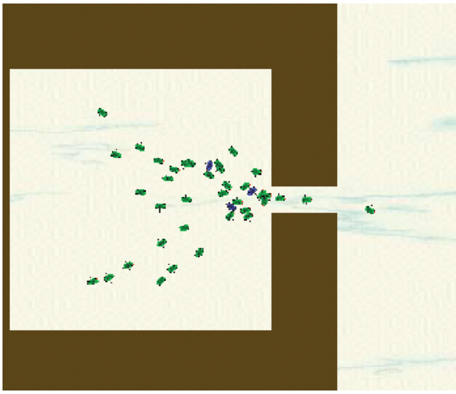


Figure 11. Simulation of emergent evacuation.

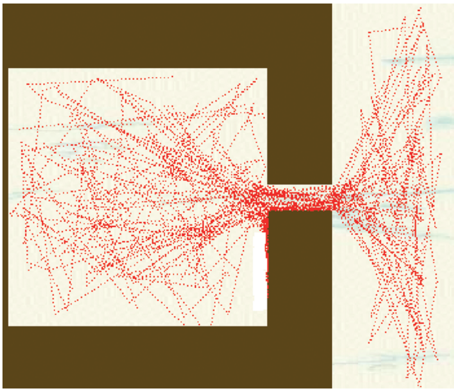


Figure 12. Illustration of system activities.

are distributed uniformly among 10 agents and the other 10 agents' moving speeds are 0.5. Note that these are agents' initial moving speeds. Agents' moving speed increase after emergency (described below). The simulation time is 6400 and the emergency start time is 1500.5. Space resolution *SR* is 0.5 and time step *TS* is 1.0. The average number of decision makings in each time interval is calculated as the total number of decision makings in that interval divided by the number of agents. Curve 1 shows the average number of decision makings of the *DTS* model for different time intervals. Since the number of decision makings of the *DTS* model dependents only on the time step, it is not changed for either the normal or emergent situations. Curve 2-4 show the distribution of the average number of decision makings of the *DES* model. Curve 2 and 3 are for the emergent cases where each agent gets a 100X and 50X speed increase respectively as along as its speed does not exceed 0.5. This constraint is used to keep a same maximum position error for both models, thus ensure a fair comparison. Curve 2 and 3 show the average number of decision makings increases after the emergency because of the speed increase. The more the speed increases, the larger number of decision makings. Curve 4 shows the average number of decision makings in each time interval for a simulation without emergency. In this case, the average number of decision making maintains the same throughout the simulation.

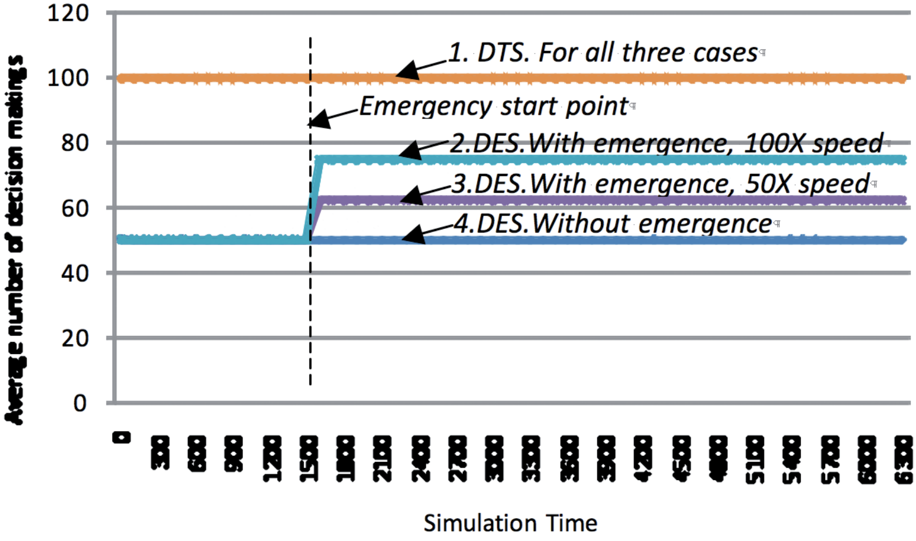


Figure 13. The distribution of the average number of decision makings.

To verify the computation advantage of the DES model as compared to the DTS model, we measure the execution time of both models for both the *uniform* and *non-uniform crowd*. We run crowd behavior simulation with 100 agents. Agents' initial speed in the *uniform crowd* is 0.05 and when an emergency occurs each agent doubles its speed. In the *non-uniform crowd*, the initial speed of each agent is generated randomly within the range [0.05, 0.5]. Similarly, when an emergency occurs each agent doubles its speed. The simulation time is 6400 and the emergency occurs at the time 500.5. Fig. 14 and Fig. 15 show the average number of decision makings and execution time for the *uniform crowd* and *non-uniform crowd* respectively. "*NoEmerg*" and "*Emerg*" stand for the normal case and emergent case respectively. "*DM*" stands for the average number of decision makings and "*Time*" stands for the execution time. The left-side Y-axis shows the execution time for each spatial resolution in milliseconds. And the right-side Y-axis shows the corresponding average number of decision makings. Fig. 14 shows that agents in both models perform almost same number of decision makings in either the normal case or the emergent case. Because of this, there is no big difference of the execution time of both models under both the normal and emergent cases. For the non-uniform crowd (Fig. 15), the *DES* model is more efficient than the *DTS* model in both cases since the execution time of the *DES* model is less than that of the *DTS* model, resulting from the less number of

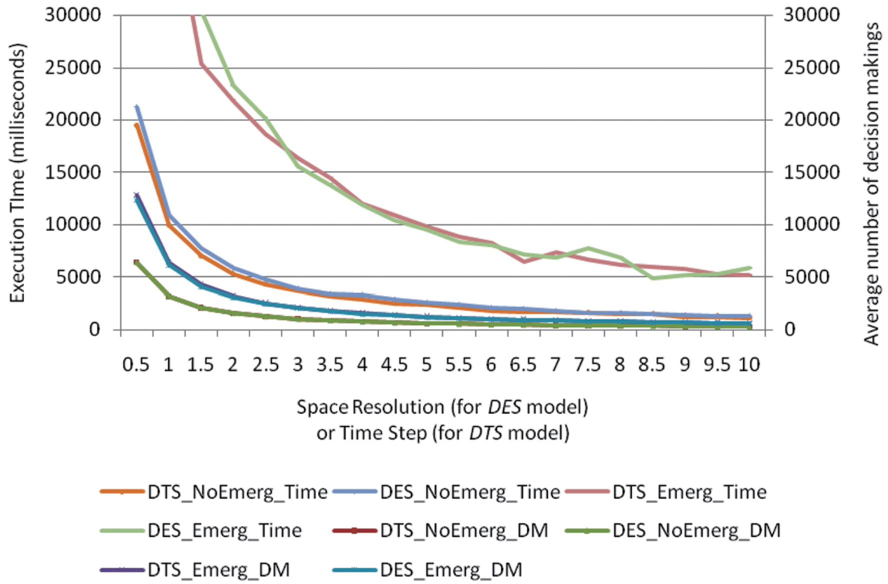


Figure 14. Execution time of *uniform crowd* in emergent evacuation.

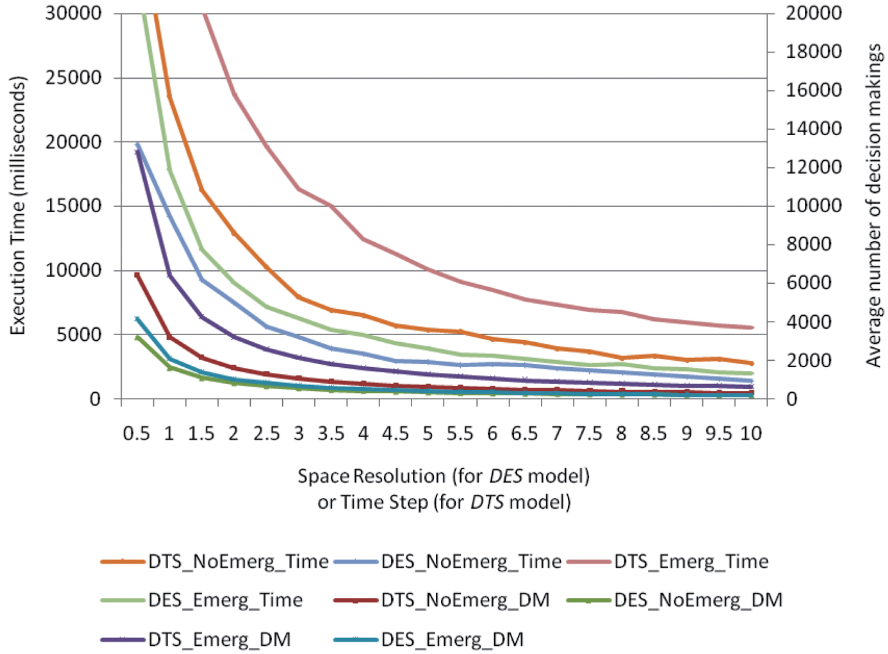


Figure 15. Execution time of *non-uniform crowd* in emergent evacuation.

decision makings performed in the *DES* model. Both Fig. 14 and Fig. 15 show that in the emergent case, agents perform more decision makings than the normal case. And thus, in the emergent case, the execution time is more than that of the normal case. This is due to the speed increase after the emergency starts. The faster the agents move the more decision makings they will perform (see section 2 for more details).

Conclusion and future work

This paper presents a discrete event model for simulating pedestrian crowd to exploit the crowd system's spatial-temporal heterogeneity resulting from agents' non-uniform movements. Both discrete event and discrete time models are considered and their performance results are compared. The experiment results show that the *DES* model can achieve better performance results (fewer decision makings and less execution time) than the *DTS* model for the non-uniform crowd. This is because the discrete event model is able to track the crowd system's "activities" in both space and time, and thus lead to more computation efficient simulations.

Future work will be carried out from several aspects. First, besides the decision makings, space heterogeneity also affects the frequency of agent interactions, such as the number of message passing between the agents. Thus one future work is to explore how the *DES* model affects the message passing among agents. Second, we will carry out more research to study how the *DES* model can affect the position error for more complex simulations with more agents. Third, we will study approaches of choosing an appropriate space resolution for a specific agent based simulation, and applying the proposed *DES* approach to other general agent based systems.

References

[Davidsson, 2000] – Davidsson P., Multi Agent Based Simulation: Beyond social simulation. *Multi Agent Based Simulation*, Vol. 1979, 2000, pp. 97-107.

[Dubiel and Tsimhoni, 2005] – Dubiel B. and Tsimhoni O., Integrating agent based modeling into a discrete event simulation. *In: Proceedings of the 2005 Winter Simulation Conference*, 2005, pp. 1029-1037.

[Helbing *et al.*, 2002] – Helbing D., Farkás I. J., Molnár P. and Vicsek T., Simulation of pedestrian crowds in normal and evacuation situations. *In:* Schreckenberg M. and Sharma S. D. (eds.), *Pedestrian and Evacuation Dynamics*, Berlin, Springer, 2002, pp. 21-58.

[Kaup and Clarke, 2007] – Kaup D. J. and Clarke T. L., Crowd Dynamics Simulation Research. *In: Proceedings of 16th Conference on Behavior Representation in Modeling and Simulation*, 2007, pp. 173-180.

[Musse and Thalmann, 2001] – Musse S. R. and Thalmann D., Hierarchical model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 7, No. 2, 2001, pp. 152-164.

[Pan *et al.*, 2006] – Pan X., Han C. S., Law K. H. and Latombe J. C., A Computational Framework to Simulate Human and Social Behaviors during Emergency Evacuations. *Blume Center News*, The John A. Blume Earthquake Engineering Center, Stanford University, No. 44, 2006.

[Pelechano *et al.*, 2007] – Pelechano N., Allbeck J. and Badler N., Controlling Individual Agents in High-Density Crowd Simulation. *ACM SIGGRAPH / Eurographics Symposium on Computer Animation (San Diego, 2007)*, pp. 99-108.

[Qiu and Hu, 2008] – Qiu F. and Hu X., BehaviorSim: A Learning Environment for Behavior based Agent. *In: Proc. The 10th International Conference on the SIMULATN OF ADAPTIVE BEHAVIOR (SAB'08)*, 2008, pp. 499-508.

[Qiu and Hu, 2010] – Qiu F. and Hu X., Modeling Group Structure in Pedestrian Crowds Simulation. *In: Simulation Modeling Practice and Theory (SIMPAT)*, Vol. 18, issue 2, February 2010, pp. 190-205.

[Seck *et al.*, 2005] – Seck M., Frydman C., Giambiasi N., Ören T. I. and Yilmaz L., Use of a Dynamic Personality Filter in Discrete Event Simulation of Human Behavior under Stress and Fatigue. *In: Proceedings of the 1st International Conference on Augmented Cognition*, 2005.

[Zaft and Zeigler, 2002] – Zaft G. C. and Zeigler B. P., Discrete Event Simulation and Social Science: The XeriScape Artificial Society. *In: Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics*, 2002.

[Zeigler *et al.*, 2004] – Zeigler B. P., Jammalamadaka R. and Akerkar S. R., Continuity and Change (Activity) Are Fundamentally Related in DEVS Simulation of Continuous Systems. *Simulation and Planning AIS'04*, 2004, pp. 1-13.

[Zeigler *et al.*, 2000] –Zeigler B. P., Kim T. G. and Praehofer H., *Theory of Modeling and Simulation*, Academic Press, 2000 (2 ed.).