Proceedings of the 1995 Winter Simulation Conference
ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman

ACHIEVING O(N) IN SIMULATING THE BILLIARDS PROBLEM IN DISCRETE-EVENT SIMULATION

Gary Harless
Ralph V. Rogers

Department of Industrial Engineering and Management Systems
University of Central Florida
Orlando, FL 32816, USA.

ABSTRACT

This paper identifies underlying issues associate with simulating those classes of problems which require both arbitary spatial and temporal precision and which must deal the with the complexities of a multitude of asynchronous pair-wise interactions occuring among a dynamic non-uniform distribution of numerous spatial components. The principal issue of interest discussed focuses on a proposed simulation modeling methodology which dynamically sectors the trajectory space based on the number of spatial objects occupying a portion of the trajectory space (i.e. object space density). That is, the trajectory space is divided into sectors of various sizes such that each sector contains no more than some specified number of spatial components. The authors demonstrate that with such a dynamic sectoring methodology a theorical reduction in the total number of pair-wise comparisons required during each time advancement can be achieved. Additionally, the theoretical computational complexity associated with identifying spatial conflicts will be better than $\mathbf{O(N^2)}$ for a non-uniform distribution of **N** spatial objects.

1 INTRODUCTION

Current discrete-event simulation methodologies do not adequately represent the spatial relationships present in many physical systems. Current methodologies are very robust for studying the temporal aspects of a system such as hourly throughput, average delay, average queue length, and maximum queue length. However, both spatial and temporal issues characterize many of the questions surrounding today's complex systems. Spatial components of these complex systems are characterized by the independent continuous movement of entities through time and space with the exception of discrete asynchronous instances of pair-wise interactions (Lubachevsky 1991).

Current methodologies for modeling systems containing continuous entity movement result in a trade-off between spatial precision, temporal precision, and computational efficiency. Spatial and temporal precision can be achieved at the expense of computational efficiency. Conversely, temporal precision and computational efficiency can be achieved at the expense of spatial precision.

Some researchers have used methodologies which incorporate sectoring in an attempt to increase computational efficiency. Sectoring involves sub-diving the trajectory space into sectors of equal size. Each spatial component is compared against other spatial components within the same sector and adjacent sectors. This strategy does reduce the total number of pair-wise comparisons required *assuming* that the spatial components are uniformly distributed throughout the trajectory space. Indeed, this has been the assumption in most previous studies (Goldberg 1984; Beckman et al. 1988; Cleary 1990; Lubachevsky 1991). However, in most complex systems of interest a clustering of spatial components occurs resulting in a non-uniform distribution of components. Examples include battlefield simulations where a clustering of components occurs at the point of conflict, air traffic control simulations where clustering occurs around major air fields, and maritime simulations where ships cluster around ports.

This paper addresses the issue of computional complexity for those classes of problems which require both arbitary spatial and temporal precision and which must deal the with the complexities of a multitude of asynchronous pair-wise interactions occuring among a dynamic non-uniform distribution of numerous spatial components. Preliminary studies suggest an approach which could reduce the computional complexity to close to **O(N)**. This approach dynamically sectors the trajectory space based on the number of spatial objects occupying a portion of the trajectory space (i.e. object space density). This paper discusses this general approach the presents the results of the studies.

2 BACKGROUND

The specific objective of the authors' research has been to design and develop an efficient discrete-event simulation methodology for modeling real systems which involve the movement and interaction of spatial objects. Simulating the billiards or coliliding puck model represents a prototypical problem for this class of complex systems. A billiards simulation requires comparisons between all objects which have the potential to collide with one another to determine if a collision is going to occur. The main computational expense associated with a billards simulation is scheduling of future collisions.

Conceptually, these systems can be easily modeled by having each individual component plan it's next event by querying every other component in the trajectory space to determine the next event of interest. This simple approach is referred to as the *naive* approach to simulating spatial interactions among **N** spatial objects. In the simulation of billiards, for example, the *naive* approach advances the global state of the billiards from collision to collision. At the point in time t_i of each collision the states of all **N** balls are examined and updated. This approach suffers from three separate problems. First, each collision is repeatedly scheduled an order of **N** times until the collision actually occurs. Second, at the point of a collision t_i most balls are not participating in collisions, but are still checked. Lastly, **N - 1** comparisons must be made by each ball to determine it's next collision resulting in a computational complexity of $\mathbf{O(N^2)}$ (Lubachevsky 1991). A high level flow chart is presented in Figure 1, which illustrates the simulation process associated with the *naive* approach.

We can estimate the computational cost of the process by stepping through the flow chart. At each step the estimated number of operations required is provided in bold type. Additionally, we assume that every ball is involved in a collision with another ball. Working from the inner most loop we can obtain a rough estimate of the computational complexity associated with the approach. The inner most loop contains a total of seven operations assuming a collision. The next loop contains two operations. Therefore, we can estimate the cost associated with this approach to be $\mathbf{7 \times N \times 2 \times N = (14 \times N^2)}$ or $\mathbf{O(N^2)}$.

3 FIXED SECTORING

To improve the performance of the *naive* approach current methodologies divide the trajectory space into sectors of equal dimension which are no smaller than the simulated components (Rogers 1993; Lubachevsky 1991; Hontales 1989; Cleary 1990; Beckman 1988; Goldberg 1984). Depending on the sectoring method employed components are either compared with other components in the same sector or to both components in the same sector and to components in directly adjacent sectors. Adjacent sectors are checked to account for balls which overlap sector boundaries. If adjacent sectors are not checked then any ball which overlaps more than one sector must be maintained on the list of balls occupying each of the sectors it overlaps. Assuming that components are distributed in a random uniform fashion, the theoretical computational complexity is reduced to approximately $\mathbf{O}(\boldsymbol{n}^2)$ where $\boldsymbol{n}$ is equal to **N** divided by the number of sectors **K** (i.e. $\boldsymbol{n} = \mathbf{N/K}$). Where **N** is the total number of balls. Figure 2 presents a high level flow chart of the process associated with a simulation utilizing sectoring.

As with the previous example, we can estimate the computational cost of the process by stepping through the flow chart. At each step the estimated number of operations required is provided in bold type. Again we assume the worst case (i.e. every ball collides with another ball) and working from the inner most loop we can obtain a rough estimate of the computational complexity associated with this approach. The two inner most loops contain a total of seven operations assuming every ball is involved in a collision. The next two loops contain two operations. Therefore, we can estimate the cost associated with this approach to be $\mathbf{7 \times \boldsymbol{n} \times 9 \times 2 \times \boldsymbol{n} \times \boldsymbol{K} = (126 \times \boldsymbol{n}^2 \times \boldsymbol{K})}$ or $\mathbf{O}(\boldsymbol{n}^2)$ since $\boldsymbol{K}$ is a constant.

The major limitation of sectoring is the assumption that components are distributed uniformly throughout the trajectory space. In most systems of interest some type of dynamic clustering of components occurs resulting in a non-uniform distribution. For example, a simulation model of the air traffic control system of the United States would have greater densities of airplanes in the vicinity of major airports like Chicago O'Hare, Atlanta and Dallas-Fort Worth. In the worst case the clustering of components causes thecomputational complexity to approach the $\mathbf{O(N^2)}$ complexity of the *naive* approach.

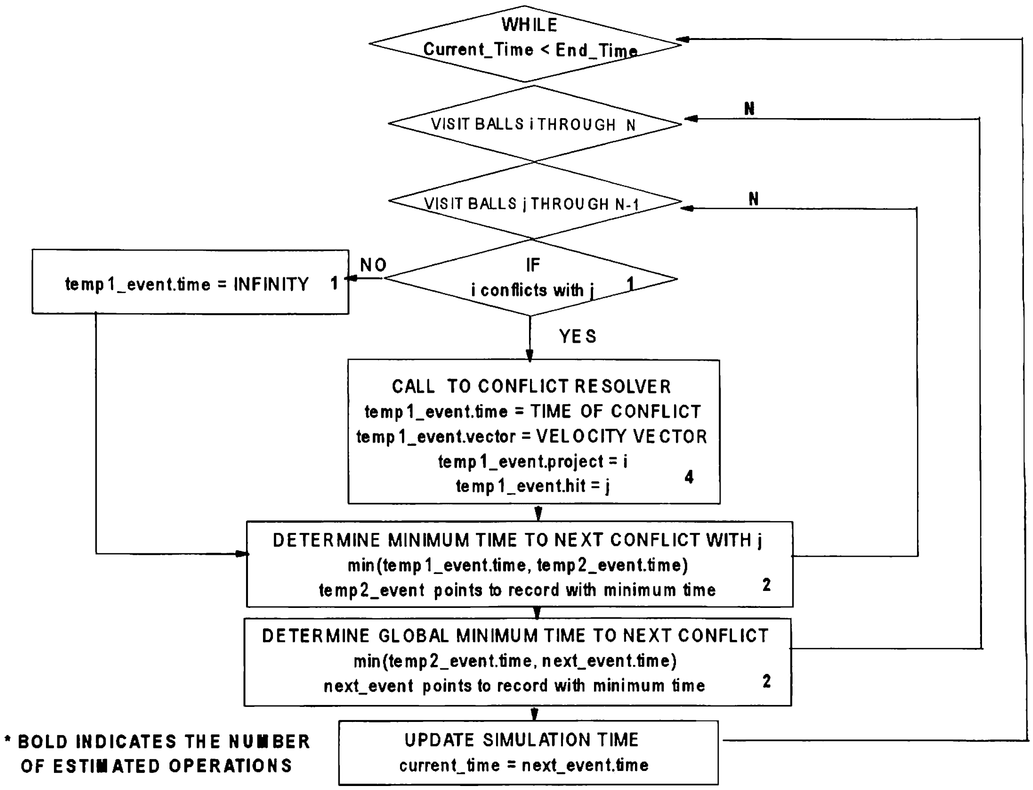

Figure 1: Flow Chart *Naive* Approach

4 DYNAMIC SECTORING

The benefits of sectoring can be extended to a non-uniform distribution of spatial objects if we are able to dynamically sub-divide the trajectory space based on the number of spatial objects occupying a portion of the trajectory space. Figure 3 illustrates the benefits a. dynamic sectoring method. If spatial objects are uniformly distributed throughout the trajectory space the sectoring approach performs well. Although, when spatial objects tend to cluster sectoring provides little benefit. However, if the sector where the clustering occurs is further sub-divided then the benefits of sectoring can be extended to a non-uniform distribution of spatial objects. A high level flow chart is presented in Figure 4 which illustrates the processes associated with the proposed dynamic sectoring approach

As with the previous example, we can estimate the computational cost of the process by stepping through a flow chart. Again we assume the worst case (i.e. every ball collides with another ball) and working from the inner most loop we can obtain a rough estimate of the computational complexity associated with the approach. Like the fixed sectoring method the two inner most loops contain a total of seven operations and the next two loops contain two operations each. At this point dynamic sectoring and fixed sectoring are equivalent with an estimated cost associated of $7 \times n \times 9 \times 2 \times n \times K = (126 \times n^2 \times K)$ or $O(n^2)$ since K is a constant. However, the dynamic sectoring method requires that sectors are sub-divided until no more than some pre-specified number of balls t occupies any one sector. This requires that the trajectory space is sub-divide c times. Where the value of c is directly proportional to the total number of balls N and the maximum number of balls t allowed per sector. Therefore, we can estimate the cost associated with this approach to be $((7 \times n \times 9 \times 2 \times n \times K) + c) = ((126 \times n^2 \times K) + c)$ or $((126 \times t^2 \times K) + c)$, since $n = t$, for each step of the clock.

5 A COMPARATIVE EXAMPLE

To illustrate the advantages of each method they will be compared using an example problem. Note that the flow charts presented do not include all the operations that would need to occur if the approach were actually

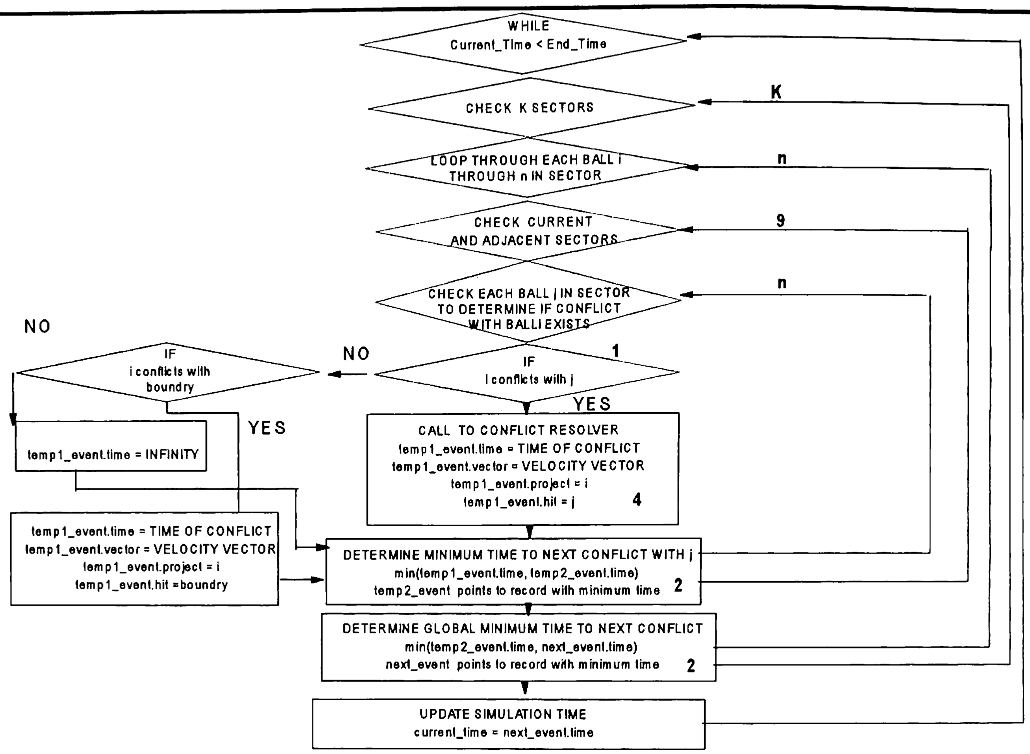

Figure 2: Flow Chart For Sectoring Approach

implemented in computer code. The comparisons are simply rough estimates of the operations required for each approach. The estimated number of operations is based on a worst case assumption. In other words, a collision occurs every time a ball moves. This assumption facilitates some consistency for the sake of comparing the different approaches. Given below is theestimated number of operations determined from the flow chart for each approach:

- *Naive* approach: $(9 \times N^2)$
- Fixed sectoring: $(126 \times n^2 \times K)$
- Dynamic sectoring: $((126 \times t^2 \times K) + c)$

where:

N = Total number of balls.
n = Number of balls in each sector.
t = Maximum number of balls allowed in each sector.
K = Total number of sectors.
c = Total number of sector sub-divisions.

An example problem will be used for the purpose of making comparisons. Initially we will assume that the balls are uniformly distributed such that there is an equal number of balls in each sector. Assume that we have 256 balls on a billiards table divided into 64 sectors (i.e. N = 256, $n = 4$, $t = 4$, $K = 64$, $c = 21$).

- *Naive* approach:
  $(9 \times N^2) = (9 \times 256^2) = 589{,}842$

- Fixed sectoring:
  $(126 \times n^2 \times K) = (126 \times 4^2 \times 64) = 129{,}024$

- Dynamic sectoring:
  $((126 \times t^2 \times K) + c) = ((126 \times 4^2 \times 64) + 21) = 129{,}045$

As this example illustrates both fixed sectoring and dynamic sectoring provide better efficiency when compared to the *naive* approach. The two sectoring approaches provide approximately equivalent efficiency when balls are distributed uniformly and an equal number of sectors are used. Now assume that we have the same number of balls, but the balls are clustered into a small area the size of a single sector.

- *Naive* approach:
  $9 \times N^2) = (9 \times 256^2) = 589{,}842$

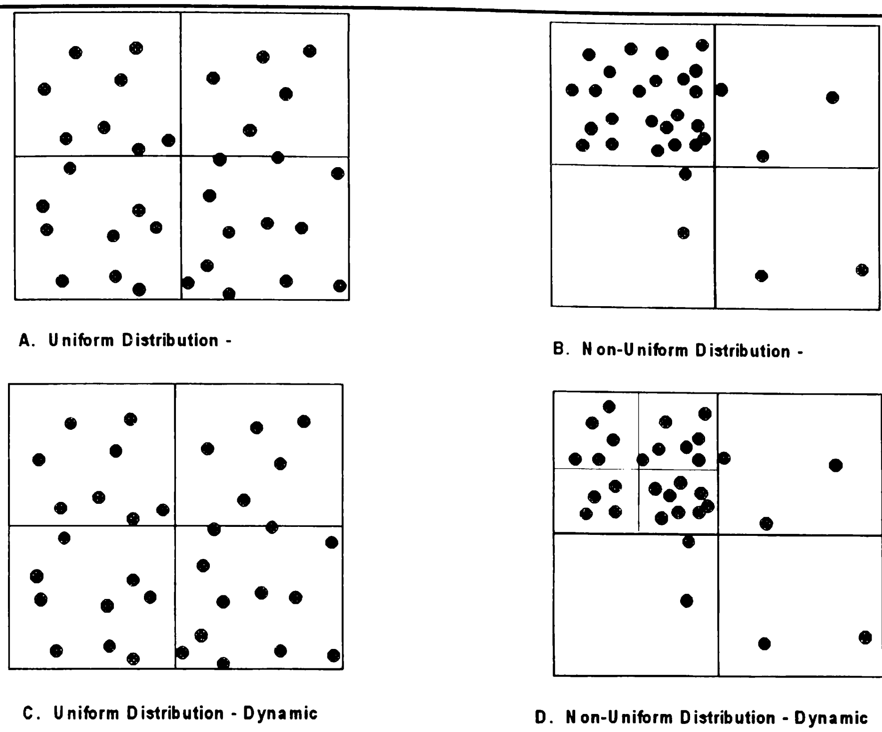

Figure 3: Examples Of Sectoring Approaches

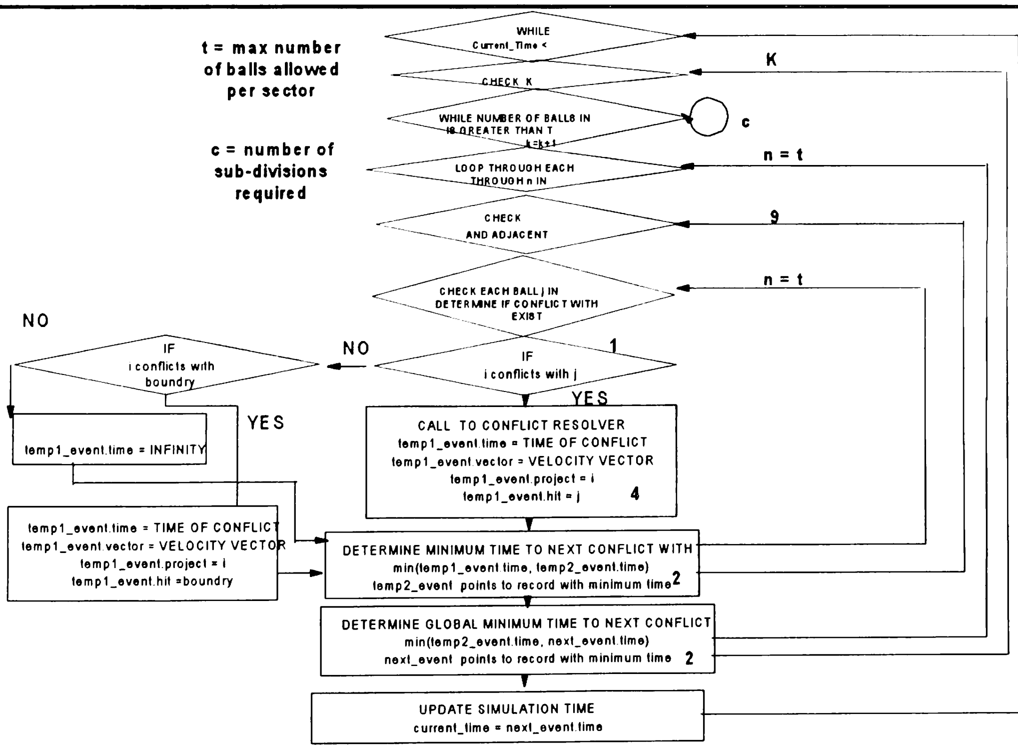

Figure 4: Flow Chart Of Dynamic Sectoring Approach

- Fixed sectoring:
  $((9 \times N^2) + K) = ((9 \times 256^2) + 64) = 589{,}888$

- Dynamic sectoring:
  $((126 \times t^2 \times K) + c) = ((126 \times 4^2 \times 64) + 51) = 129{,}075$

In this example the *naive* and fixed sectoring approaches both perform poorly. The dynamic sectoring approach performance is almost equivalent to the situation involving a uniform distribution of spatial objects. The results from the example problem demonstrate the potential of the proposed dynamic sectoring approach.

6 SUMMARY

The analysis presented suggests that an efficient discrete-event simulation methodology for modeling systems characterized by the independent continuous movement of entities through time and space can be developed based on dynamic sectoring. Clearly, a major concern is the overhead necessary to maintain a dynamic sectoring scheme. Efforts are continuing by the authors on implementation strategies for dynamic sectoring which keep overhead to acceptable levels. In particular, strategies based on object-oriented modeling and tree based data structures in lew of linked-lists are under current study and development. Early results are promising and efforts are continuing.

REFERENCES

Beckman, B. et. al. 1988. In *The Proceeding of the 1988 SCS Multiconference, Simulation Series* (Society for Computer Simulation, San Diego, CA.). Vol. 20, No. 3, p. 119-125.

Cleary, J.G. 1990. In *The Proceedings of the 1990 SCS Multiconference, Simulation Series* (Society for Computer Simulation, San Diego, CA.). Vol. 22, No. 1, p. 219-224.

Goldberg, A.P. 1984. *Object-Oriented Simulation of Pool Ball Motion*. Master's Thesis, University of California at Los Angeles, Los Angeles, CA.

Hontales, B. Beckman, et. al. 1989. In The *Proceedings* of the *1989 SCS Multiconference, Simulation Series* (Society for Computer Simulation, San Diego, CA.). Vol. 21, No. 2, p. 3-8.

Lubachevsky, B.D. 1991. How to Simulate Billiards and Similar Systems, *Journal of Computational Physics*. 94: p. 255-283.

Rogers, Ralph. V. 1993. *TATSS Final Report* (unpublished), Department of Industrial Engineering and Management Systems, University of Central Florida, Orlando, FL.

AUTHOR BIOGRAPHIES

MAJOR GARY HARLESS, U.S. Army is currently a Ph.D. student in the Industrial Engineering and Mangement Systems Department at the University of Central Florida. He received his M.S. in Industrial Engineering from the Texas A&M University in 1989 and his B.S. in Business Adminstration from Utah State University in 1981. His current research interests are in the area of discrete-event simulation. Prior to attending the University of Central Florida, Major Harless was assigned to TRADOC Analysis Command, at Fort Leavenworth, Kanas as an operations analyst.

RALPH V. ROGERS is Interim Chair and Associate Professor of Industrial Engineering and Management Systems at the University of Central Florida He received his Ph.D. in Systems Engineering from the University of Virginia. Dr. Rogers's research interests include object-oriented discrete-event simulation; knowledge-based discrete-event simulation; intelligent tutoring systems; and systems engineering. He is a m-ember of IEEE, IEEE-SMC Society, IIE, and SCS.