Proceedings of the 2002 Winter Simulation Conference
E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.

DATA-BASED NODE PENALTIES IN A PATH-FINDING ALGORITHM IN AN AUTOMATED MATERIAL HANDLING SYSTEM

Miki Fukunari
Srinivas Rajanna
Robert J. Gaskins
Mary Ellen Sparrow

805 Middlesex Turnpike
Brooks-PRI Automation, Inc.
Billerica, MA 01821, U.S.A.

ABSTRACT

Increasing factory throughput is a critical issue in the semiconductor industry, and a quick transition of material to the next location in the automation system plays a significant role in increasing throughput. A dynamic path-finding algorithm for a vehicle-based automated material handling system (AMHS) is discussed in this paper. The dynamic path-finding algorithm uses distance between nodes, node penalties, and the number of vehicles queued to calculate the total cost of a path. This paper introduces the use of historical data from the AMHS and discusses how to effectively utilize such data in critical situations to improve overall AMHS performance.

1 INTRODUCTION

For the transition to 300mm wafers, more imposition is placed on automation systems to deliver material throughout a semiconductor factory (fab). Wafers must be transported to and from process tools as well as storage buffers (stockers). Thus, an automated material handling system (AMHS) becomes more complex in terms of control logic and layout – transport routes become longer and more alternate paths need to be considered. Alleviating traffic is imperative in such an environment.

An AMHS typically seeks the shortest path for routing a vehicle to a destination. The shortest-path problem is fundamental and applicable in a variety of fields (Ikeda et al. 1994) such as manufacturing systems, communication networks, and computer systems. When applying the shortest-path problem, the system can be considered as a queuing network with finite buffer capacity. Although the study of queuing networks with buffer capacity constraints has a long history, there is a dearth of results that are both definitive and useful in the area of AMHS (Kumar et al. 1996). Furthermore, the dynamic nature of an AMHS in a fab makes it more difficult to optimize the system.

The AMHS discussed in this paper consists of monorail track, material transport vehicles, and nodes connecting track segments. A node is a module where a vehicle is routed to a different track segment, loaded or unloaded, or stopped for battery charging. The AMHS controller dynamically routes a vehicle using real-time data such as the number of vehicles queued at a node and the node type (Gaskins et al. 2001). This real-time data is used to calculate a cost, or penalty, for crossing a node. However, real-time data is not sufficient to reflect heavy traffic congestion. Thus, historical data was introduced for node penalties called data-based node penalties in this paper.

Two approaches for determining node penalties were compared via simulation: node-type and data-based. The node-type approach uses a static node penalty based on the node type (load/unload node, routing node, etc) while the data-based approach uses historical data. The simulation results indicate the data-based approach yields significant improvements in certain high-traffic situations. The data-based approach helps alleviate excessive queuing by placing a larger penalty on the right node at the right time.

2 PROBLEM STATEMENT

In an AMHS in a semiconductor fab, vehicles tend to cluster around one busy node at certain times because of lot batching and frequent move requests at one specific node (Bahri and Gaskins 2000). For example, compare the two situations at an inter-floor bridge node shown in Figure 1. For the more crowded case in Figure 1, node penalties need to be larger because a vehicle has to wait not only for the three vehicles in the next segment but also for the two vehicles ahead of the next segment. This blockage has a significant impact on the system performance.

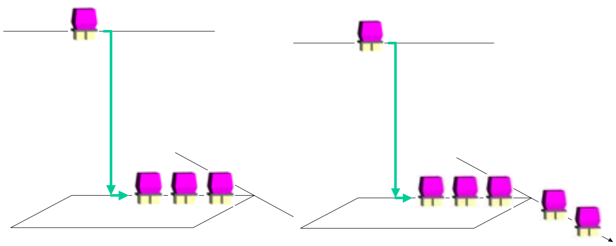

Figure 1: Vehicle Congestion

In order to avoid crowded spots, the node penalties need to be dynamic. Assigning larger penalties in busy areas at the right time alleviates congestion because it encourages other vehicles to take a different route thereby avoiding the traffic. The system controller re-evaluates a vehicle's route each time the vehicle reaches a point where more than one track segment can be taken (i.e., when a routing node is encountered) since the state of the system (i.e., traffic conditions) changes frequently in the AMHS. The system controller determines the shortest, or minimum cost, path between the vehicle's current node and destination node using the Dijkstra algorithm, which is a traditional algorithm to solve the shortest-path problem.

3 APPROACHES

The total cost of a path is determined by the distance between a source and a destination node, the number of vehicles queued at nodes in the path, and node penalties. The system controller chooses the path that has the least total cost. Two approaches were compared for node penalties: node-type and data-based. The node-type approach uses static node penalty based on a node's type while the data-based approach uses historical data obtained from the AMHS as it in operation.

3.1 Node-Type Approach

In the node-type approach, the node penalty is a constant based on a node's type. A node's type is determined primarily by its function (routing, loading/unloading, charging, etc.). The total cost to cross a node is the product of the number of vehicles currently queued at the node and the static penalty. For instance, the total penalty is χ if one vehicle is queued at a node, 2χ if two vehicles are queued, 3χ if three vehicles are queued, and so on where χ is the static penalty based on the node's type. This approach works well when the system experiences insignificant congestion. This approach, however, does not work well when heavy congestion occurs. Congestion often results when a large number of vehicle moves are attempted in same area within a short time. Congestion is more likely when vehicles are unable to queue due to limited space between nodes. It is critical that the system controller deal with this situation.

3.2 Data-Based Approach

In order to avoid congestion, node penalties are dynamically manipulated in the data-based approach. The data-based node penalty is calculated from data describing the time required for vehicles to cross a node over a recent period of time. For example, the total penalty is $\chi(t)$ if one vehicle is queued at a node, $2\chi(t)$ if two vehicles are queued, $3\chi(t)$ if three vehicles are queued, and so on where $\chi(t)$ is obtained from historical data on node crossing time. The penalty $\chi(t)$ varies with time. Since vehicle-crossing time varies due to changing traffic conditions, dynamic node penalties quickly reflect state changes. With the data-based approach, the system controller avoids routing a vehicle to a crowded area by giving nodes in that area a larger penalty at the right moment.

4 COMPARISON VIA SIMULATION

The two approaches for node penalties were evaluated through simulation. The node-penalty logic was included in the simulation model along with key elements of the AMHS including the system controller, vehicles, stockers, etc. An AeroTrak™-based AMHS was modeled. The AeroTrak transport system, from Brooks-PRI Automation, features vehicles traveling on an overhead monorail track. The vehicles transport wafer lots between stockers. AeroTrak also provides inter-floor bridging devices for moving vehicles (loaded or unloaded) between track on different levels of a multi-floored fab. A comprehensive AeroTrak simulator was developed using AutoMod™ software from Brooks-PRI Automation. The scope of this discrete-event simulator includes all physical components as well as the system controller. All routing and dispatching rules were essentially translated from the controller's source code to the AutoMod process language.

The two approaches were compared across a variety of layout types (multi-loop, multi-floor, etc.). When evaluating the two approaches for a given layout, all model input data and modeling assumptions remained the same but for the node-penalty approach. Ten days were simulated following a ten-hour warm-up period. Observations made during the warm-up period were discarded to eliminate initialization bias.

4.1 Comparison At Important Nodes

In a multi-floor system, the balancing of inter-floor bridge nodes is essential for good system performance. Because inter-floor moves usually require more time, traffic around the inter-floor bridge nodes has a significant impact on the overall system performance. In the simulated multi-floor

AMHS, unbalanced utilization of the bridge nodes was observed when the node-type approach was used. However, as can be seen in Table 1, this problem did not occur when the data-based approach was used.

Table 1: Simulation Results of Inter-Floor Bridge Nodes

Inter-Floor Node	Node Blockage (%)	
	Node-Type	Data-Based
VS1	1.18	0.57
VS2	0.74	0.36
VS3	0.14	0.65
VS4	0.56	0.39

The number of necessary inter-floor moves is the same for the two node-penalty approaches. Thus, the goal is not to reduce node blockage at all inter-floor bridge nodes, but rather to (1) minimize the number of empty vehicles traveling to inter-floor bridge nodes and to (2) share excess moves with other inter-floor bridge nodes. The simulation results indicate that if these two objectives are met, overall average node blockage will be reduced. In addition, node blockage at the inter-floor bridge nodes was more evenly distributed in the data-based approach. This is because the system controller chose a wider variety of paths, and therefore a wider variety of inter-floor bridge nodes, with the data-based approach.

4.2 Comparison Across Different Layouts

Multiple layouts were tested and the simulation results, as shown in Table 2, indicate improvements with the data-based approach when the AMHS suffers from long delivery times due to heavy congestion.

5 CONCLUSION

A path-finding algorithm is critical for automated material handling in a semiconductor fab. As move rate and/or layout complexity increase, the system performance is more greatly affected by the effectiveness of the vehicle routing algorithm. Having larger penalties for nodes in crammed areas can alleviate congestion. This would encourage other vehicles to take an alternative route. Two approaches for penalties in a path-finding algorithm were compared: node-type and data-based. Both approaches cooperate with queuing delays, incorporating the number of vehicles queued at a node. The node-type approach, however, does not perform well when the system suffers from heavy traffic, while the data-based approach shows better performance in such a case by feeding historical data back to the system controller. The result is that the data-based approach improves the system performance when significant congestion areas are found in the system.

Table 2: Simulation Results of Various Layouts

Layout	Loop	Type of Penalty	MPH	No of Cars	Delivery Time Avg	Transport Time Avg	% Car Moving
Inter-Floor	NA	Node-Type	350	46	11.92	4.72	40.79%
		Data-Based	**350**	**46**	**4.59**	**3.54**	**30.65%**
	Loop1	Node-Type	481	70	61.96	6.88	67.75%
		Data-Based	**481**	**70**	**9.52**	**6.28**	**62.66%**
	Loop2	Node-Type	192	16	5.46	3.06	50.76%
		Data-Based	**192**	**16**	**5.49**	**3.06**	**50.77%**
Ball-Room	NA	Node-Type	700	100	6.58	4.53	66.00%
		Data-Based	**700**	**100**	**6.54**	**4.48**	**66.00%**
	NA	Node-Type	515	45	5.87	3.37	60.10%
		Data-Based	**515**	**45**	**5.80**	**3.36**	**59.32%**
Multiple Loop	Loop1	Node-Type	209	44	7.54	5.41	50.37%
		Data-Based	**209**	**44**	**7.62**	**5.43**	**50.82%**
	Loop2	Node-Type	266	38	9.19	4.66	55.36%
		Data-Based	**266**	**38**	**9.06**	**4.66**	**55.01%**
	Loop3	Node-Type	189	44	10.48	7.62	60.29%
		Data-Based	**189**	**44**	**10.49**	**7.59**	**60.24%**
Center Aisle	NA	Node-Type	464	42	4.86	3.28	56.76%
		Data-Based	**464**	**42**	**4.89**	**3.27**	**56.54%**

REFERENCES

Bahri, N. and R. J. Gaskins. 2000. Automated Material Handling System Traffic Control By Means of Node Balancing. In *Proceedings of the 2000 Winter Simulation Conference*, ed. J.A. Joines, R.R. Barton, K. Kang, and P.A. Fishwick, 1344-1346. Piscataway, New Jersey: Institute of Electrical ad Electronics Engineers.

Gaskins, R. J., T. Mariano and M. Sparrow. 2001. Dynamic Traffic-Based Routing Algorithm. U.S. Patent #6,285,951.

Ikeda, T., M. Hsu, and H. Imai. 1994. A Fact Algorithm For Finding Better Routes By AI Search Techniques. In *Proceedings Of 1994 Vehicle Navigation & Information Systems Conference,* 291-296.

Kumar, S., R. Srikant, and P. R. Kumar. 1996. Bounding Blocking Probabilities And Throughput In Queueing Networks With Buffer Capacity Constraints. In *Proceedings of the 35th Conference on Decision and Control,* 779-784.

AUTHOR BIOGRAPHIES

MIKI FUKUNARI is a simulation engineer in the Automation Planning and Design group at Brooks-PRI Automation. She received her B.S. in Physics from Tokyo Metropolitan University in 1992 and her M.S. degree in Industrial Engineering from Arizona State University in 1999. She has been involved in the simulation modeling and analysis of semiconductor manufacturing over the last three years. Her current interests include simulation modeling methodologies and optimization algorithms in semiconductor manufacturing.

SRINIVAS RAJANNA is a simulation engineer in the Automation Planning and Design group at Brooks-PRI Automation. He received his B.S. in Mechanical Engineering from Bangalore University, India in 1995 and M.S. degree in Industrial Engineering from West Virginia University in 2000. He has been involved in the simulation modeling and analysis of a wide range of manufacturing systems over the last two years.

ROBERT J. GASKINS is a simulation engineer in the Automation Planning and Design group at Brooks-PRI Automation. He received his B.S. and M.S. degrees in Industrial Engineering from Purdue University. He has been involved in the simulation modeling and analysis of a wide range of manufacturing and communication systems over the last fifteen years. His current interests include modeling methodologies and the simulation/emulation of automated material handling systems.

MARY ELLEN SPARROW is a software developer in the Automation Material Handling Systems group at Brooks-PRI Automation. She received her B.S. in Geology with a concentration in Mathmetics from the University of Massachusetts in 1980 and her M.S. degree in Computer Science from University of New Hampshire in 1989. She has been involved in the software for overhead monorail systems used in semiconductor manufacturing over the last six years. Her current interests include traffic management and optimization algorithms to improve semiconductor manufacturing.