Computers & Industrial Engineering 70 (2014) 176–182

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Congestion-aware dynamic routing in automated material handling systems

Kelly Bartlett [a,*], Junho Lee [a], Shabbir Ahmed [a], George Nemhauser [a], Joel Sokol [a], Byungsoo Na [b]

[a] Georgia Institute of Technology, 755 Ferst Drive, NW, Atlanta, GA 30332-0205, USA
[b] Division of Business Administration, Korea University, 2511 Sejong-ro, Sejong City, 339-700, Republic of Korea

ARTICLE INFO

Article history:
Received 22 November 2013
Received in revised form 31 January 2014
Accepted 1 February 2014
Available online 10 February 2014

Keywords:
Semiconductor
Material handling
Real-time
Dynamic vehicle routing
Simulation
Congestion

ABSTRACT

In automated material handling systems (AMHS), such as those used to transport wafers in semiconductor manufacturing facilities, vehicular congestion leads to transport delay and reduced production efficiency. Through the use of a high-fidelity simulation, we demonstrate a congestion-aware dynamic routing strategy that efficiently reroutes vehicles as congestion status changes. Steady-state routing performance moderately improves, the frequency of heavy congestion is significantly reduced, and the system recovers from vehicle breakdowns more efficiently.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

Semiconductor manufacturing is a highly complex, re-entrant process in which wafers, packaged in cartridges, are transported through hundreds of processing steps in a clean-room facility. Due to the size, value, and sensitive nature of the contents being transported, cartridges move through a facility on an automated material handling system (AMHS) commonly consisting of track-based Overhead Hoist Transport (OHT) vehicles. Severe congestion due to inefficient OHT routing and/or OHT equipment failure can quickly escalate into large-scale production delay. This motivates the need for routing algorithms that efficiently handle both steady-state congestion and exceptions such as OHT failure. Prior to implementing new vehicle control methods, proposed changes are tested on proprietary simulation software that integrates production scheduling and material handling. Justifying such testing requires evidence that the proposed changes will result in increased efficiency and/or reduced cost.

Semiconductor fabrication facilities are commonly organized into bays containing machines of the same type. In the type of facility that we consider, bays are connected via a center loop and an outer loop and each bay has two entrances and two exits. When a cartridge completes processing at one step, a transfer request is generated and a centralized control system assigns an OHT vehicle to transport it. If no vehicle is available, the system adds the request to a queue for requests awaiting assignment. Upon assignment, the vehicle moves to the current location of the cartridge, unloads the cartridge from the machine into the vehicle, moves to either the next machine in the processing sequence or to a storage location, and loads the cartridge into the next machine or storage location. Control of vehicle movement is complicated by physical characteristics such as differing velocities on straight and curved edges, hardware considerations that require deceleration, and policies that govern the movement of idle vehicles.

In practice, the route that a vehicle travels is selected based on static factors such as distance, maximum travel velocity on an edge, and the location of loading/unloading points. All requests for transport between a specific origin and destination will use the same route regardless of current congestion location and intensity and follow this route regardless of how the system changes. If congestion develops on the intended path, the vehicle will still travel directly into the congestion, both delaying itself and making the congestion worse. We define this as *static routing*. Due to the structure of semiconductor manufacturing facilities, static routing often causes significant congestion on the center loop including deadlock congestion requiring manual intervention to resolve. In

* Corresponding author. Tel.: +1 301 481 3260.
E-mail addresses: kelly.bartlett@gatech.edu (K. Bartlett), junholee@gatech.edu (J. Lee), sahmed@isye.gatech.edu (S. Ahmed), george.nemhauser@isye.gatech.edu (G. Nemhauser), jsokol@isye.gatech.edu (J. Sokol), byungsoo.na@korea.ac.kr (B. Na).

http://dx.doi.org/10.1016/j.cie.2014.02.002
0360-8352/© 2014 Elsevier Ltd. All rights reserved.

order to distribute traffic and reduce the occurrence of deadlock, engineers use ad hoc congestion penalties. The method is not systematic, does not easily accommodate changes in production sequence and product mix, and does not respond quickly to changing congestion characteristics.

To more effectively route vehicles and reduce congestion, a vehicle must consider the current and future routes of all other vehicles. If all requests were known in advance, large-scale integer programming could be used for a priori route optimization. Due to processing time uncertainty and flexible production scheduling, request times, origins, and destinations are not known in advance. Thus, this is an online, real-time vehicle routing problem. We define *dynamic routing* as an approach that allows a vehicle to alter its path in response to congestion by continually learning about the state of the system and adapting based on the new information. In the literature, congestion-aware dynamic routing methods are proposed but they are computationally intense and their effectiveness is demonstrated only on small-scale simulations with fewer than 50 vehicles.

In this paper, we propose a congestion-aware dynamic routing approach that uses an easily implementable congestion metric and is computationally efficient enough to be used in practice. We demonstrate its effectiveness in a high fidelity simulation of vehicle movement in a prototype facility. Steady-state performance is moderately improved, the frequency of heavy congestion is significantly reduced, and the system is much better able to recover from exceptions such as vehicle breakdowns. Even though transfer requests are generated at the rate of 1–2 per second, the dynamic routing algorithm is fast enough so that no delays occur in its simulated implementation.

In the remainder of this paper, Section 2 briefly discusses relevant literature, Section 3 describes our approach, Section 4 provides detail on our simulation, and Section 5 presents results. Section 6 concludes.

2. Literature

Limited work has been done in AMHS vehicle routing because the transition to unified systems from segregated systems is relatively recent. In unified systems, a vehicle moves throughout the entire network as opposed to being confined to one specific area as it is in the older segregated design. Selection of routes is more complex in unified systems. In terms of congestion-aware dynamic routing, (Yang, Cheng, Chiang, & Fu, 2008) reroutes vehicles at each diverging intersection by selecting a route from a candidate set using the number of vehicles currently traveling along a route as a measure of congestion. It demonstrates a moderate improvement in steady-state metrics in a small simulation but does not discuss the occurrence of heavy congestion, deadlock, or vehicle breakdown. Patents (Gaskins, Mariano, & Sparrow, 2001 and Huang, Chen, Yu, & Yu, 2008) also propose route selection using pre-determined candidate route sets. Both cover cases where vehicles are and are not rerouted while in progress and do not present results. Patent (Gaskins et al., 2001) uses the number of vehicles currently traveling along a route as a measure of congestion and (Huang et al., 2008) does not propose a specific metric.

Time-windows based (Kim & Tanchoco, 1991; Kim & Tanchoco, 1993; Huang, Palekar, & Kapoor, 1993; Smolic-Rocak, Bogdan, Kovacic, & Petrovic, 2010; Chen, Sun, Dai, Tao, & Liu, 2013), incremental route planning (Tanchoco & Taghaboni-Dutta, 1995), hierarchical simulation (Seifert, Kay, & Wilson, 1998), petri-nets (Nishi & Maeno, 2010; Nishi & Tanaka, 2012), zone-control (Ho & Liao, 2009), and agent-based (Lau & Woo, 2008) methods have been proposed for various types of congestion-aware dynamic routing in various types of general automated guided vehicle (AGV) networks with few vehicles. AGV is a general term used to describe any automated vehicle. General AGV methods are not always applicable to OHT systems because they typically consider smaller networks with a more general structure. The bay-based structure of OHT systems limits the existence of alternate paths.

3. Approach

We model a facility as a directed network. *Nodes* correspond to loading/unloading ports and rail intersection points and *edges* to rail segments connecting nodes. All rails are unidirectional. Each edge is assigned a *weight* that represents the estimated traversal time for that edge. For some purposes, we consider the *intersection point network* where nodes represent only rail intersection points, not ports. We use the term *diverging node* to indicate a node that has more than one outgoing edge. Note that in OHT networks, the total number of incoming plus outgoing edges associated with each node is typically no more than three.

We propose a congestion-aware dynamic routing method that uses continually updating edge weights computed via exponential smoothing. For each edge, we store an edge weight that represents the estimated traversal time. When a vehicle passes a node, the system records the time that it took for that vehicle to traverse that edge and updates the estimated edge traversal time using exponential smoothing, $t_{ij}^{n} = (1-\lambda)t_{ij}^{n-1} + \lambda l_{ij}^{n}$, where t_{ij}^{n} is the edge weight for edge (i,j) after the nth traversal, l_{ij}^{n} is the nth traversal time, and λ is a parameter indicating the sensitivity of the estimated edge traversal time to new information.

Before any vehicles traverse an edge, we set λ equal to the length of the edge divided by the maximum travel velocity on that edge. With a small value of λ, estimated edge traversal times approach the long-run average. Good values of λ may vary with layout and operational characteristics affecting the likelihood of heavy congestion or deadlock. We use one value of λ across the whole system but could use edge-specific values of λ. For example, it may be preferable to use larger values of λ for areas of the network with more frequent congestion and smaller for areas that are sparsely used.

Previously studied measures of congestion, such as the number of vehicles currently moving on a route, represent an instantaneous view of system congestion. By incorporating recent information in addition to current information, a more robust representation can be achieved. Our method also incorporates expected delay due to layout characteristics such as likely deceleration behind vehicles entering curved edges.

In our proposed method, vehicles are routed and rerouted using node-based lookup tables. Traditionally, a vehicle's route is stored as a vehicle attribute. At each diverging node the vehicle selects its next node from the stored route. To route dynamically, a vehicle recalculates its route at each diverging node to ensure that its preferred route has not changed. If we have R requests during a T second simulation time and each route has an average of D diverging nodes, the total number of route calculations will be $R*(D+1)$, which may not be evenly distributed over time.

Our proposed method stores a lookup table at each diverging node containing the next node in the route to each destination. Tables are updated periodically over time at a fixed time interval. As a vehicle approaches the diverging node, it looks up its next node in the table. For example, in Fig. 1 a vehicle is approaching node I and has a destination of node Z. The vehicle looks in the table associated with node I in the row for destination Z. The table indicates that the vehicle should next go to node A based on the current congestion status. Note that each vehicle with a particular destination approaching a given node between two subsequent table updates will follow the same route.

Download English Version:

https://daneshyari.com/en/article/1134005

Download Persian Version:

https://daneshyari.com/article/1134005

Daneshyari.com