*Proceedings of the 2017 Winter Simulation Conference*
*W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.*

# SCHEDULING OF DRONE-BASED MATERIAL TRANSFER SYSTEM IN SEMICONDUCTOR MANUFACTURING

Andy Ham

Industrial and Systems Engineering
Liberty University
1971 University Blvd.
Lynchburg, VA 24515, USA

DJ Kim

Central Operations Team
Micron Technology Inc,
9600 Godwin Dr.
Manassas, VA 20110, USA

## ABSTRACT

The idea of deploying unmanned aerial vehicles, also known as drones, for delivery in logistics operations has inspired this research. One conceivable scenario is to use a drone to transfer jobs between locations in a future semiconductor factory. Each job might be characterized by origin, destination, priority, and precedence-relationship. In particular, the precedence-relationship occurs when drones are competing for limited number of ports (similar to helicopter landing platform). The objective is to minimize the maximum completion time of all delivery jobs performed by a fleet of drones. Two exact approaches are presented: a mixed integer programming and a constraint programming, and tested for real-time perspective with problem instances up to 50-drone and 100-job.

## 1 INTRODUCTION

A semiconductor fabrication line (fab hereafter) has been always emphasizing a cycle time reduction in order to cope with a rapidly changing market demand. The efforts include deploying advanced scheduling/dispatching systems, merging processing steps, eliminating unnecessary steps, adjusting batching size, utilizing chamber machines, designing efficient fab layout, and so on. Now, material handling (or transfer) system (MHS or MTS), once considered as non-critical, has started receiving an intensive attention since a job travels several miles to visit hundreds of machine in a fab, relying on MTS. Furthermore, a fab, where its expensive clean-room facilities demand a zero footprint, has been motivated to utilize an air-space for MTS. Therefore, overhead transfer system (OHT), wherein vehicles travel on the rail mounted near the ceiling to transfer jobs, was devised. Ironically speaking, OHT already utilizes an air-space to transfer jobs, just like drone will do. Replacing OHT with drone would create many benefits. One of obvious is tearing down rails and vehicles, which opens up an whole new way of designing a fab. A high tower-like fab can be foreseen when drone replaces OHT. Another is a faster delivery time since drone flies.

Regarding vehicle allocation problem, industries have used simple dispatching approaches (Wang et al. 2016; Lin et al. 2001). Considering AMHS as a supporting mechanical system for main production machines in a fab seems to contribute to the reason. However, in this coming drone era, simple dispatching approaches will lose its ground. Instead, a detailed scheduling system, that simultaneously orchestras a fleet of drones and a set of jobs in real-time, will be demanded.

## 2 MATERIAL TRANSFER DRONE SCHEDULING PROBLEM IN FAB

Inspired by Amazon's significant expansion of its army of warehouse robots and the adaptation of drones for final-mile small parcel delivery, Ham (2017) mixed these two emerging technologies and proposed a drone-powered material transfer system in a warehouse, wherein jobs could be picked up by drone, transferred, and dropped off directly to shipping cartons. Ham termed this the material transfer drone scheduling problem (MTDSP). This paper applies his proposed model in a similar problem encountered in a fab.

The problem may be formally defined as follows (Ham, 2017). Let $J$ represent the set of jobs $\{1, \ldots, n\}$ where $n$ denote the number of jobs (lots to deliver). Let $M$ represent the set of locations $\{1, \ldots, m\}$, where $m$ denotes numbers of collective locations including machines and stockers. Then let $D$ represent the set of drones $\{1, \ldots, \kappa^d\}$, where $\kappa^d$ denotes numbers of parallel drones. Let $\iota^d$ represent the last known location of each drone $d$. Also, let $pkup_j(drop_j)$ denote the pickup (drop) locations of job $j$.
The MTDSP for the fab can be defined on a directed graph $G = (M, A)$, where $M = \{1, \ldots, m\}$ is the set of nodes and $A$ is the set of arcs. For $pkup_j, drop_j \in M$, the arc exists between them. The time required for the drone to travel between nodes is given by $\tau^t_{m,\hat{m}}$. The travel-time is assumed to be given with a high accuracy. A time window $\left[\underline{w_j}, \overline{w_j}\right]$ is associated with each job $j$, where $\underline{w_j}$ and $\overline{w_j}$ represent the earliest start (EST) and the latest completion time (LCT), respectively, which indirectly reflect the job priority. Further, a set of jobs could have a precedence-relationship because some of ports of machine permit drone's access one at a time due to limited space (similar to helicopter landing platform). Therefore, a pickup (unloading) must occur prior to a drop (loading) when a job preoccupies the port. The objective is to schedule parallel drones to minimize the maximum completion time of all jobs, while satisfying both precedence-relationship and job-priority expressed as time-window.

Table 1 represents a sample problem with 6-job. Each job has origin, destination, precedence-relationship, and time-window. For instance, $j_1$ is currently located at $m_1$, and must be delivered to $m_2$, within the given time-window. Also, $j_1$ must complete after $j_2$ completes since $j_2$ must be first picked up since $m_2$ has only one port. Also, there are two parallel drones, which are hovering over $m_3$ and $m_2$, respectively. Figure 1 pictures such a system with the transfer-time between locations.

Table 1: A sample MTDSP in a fab with 6-job.

| Job | from-loc(pkup) | to-loc (drop) | Precedence | ECT | LCT |
|---|---|---|---|---|---|
| $j_1$ | $m_1$ | $m_2$ | after $j_2$ | 0 | 120 |
| $j_2$ | $m_2$ | *stk* | | 0 | 150 |
| $j_3$ | $m_3$ | $m_1$ | | 30 | 60 |
| $j_4$ | $m_4$ | *stk* | | 0 | 90 |
| $j_5$ | *stk* | $m_3$ | after $j_3$ | 0 | 180 |
| $j_6$ | *stk* | $m_4$ | after $j_4$ | 0 | 240 |

Two exact approaches are proposed by Ham (2017): a mixed integer programming and a constraint programming. This paper only contains the proposed CP formulation. For the MIP formulation, refer to his paper. It is worth mentioning that CP formulation is very different with MIP formulation. Furthermore, there is no standard in CP formulation. Namely, it varies to each CP package, unlike a similar MIP formulation (Ham and Cakici, 2016). Therefore, this paper will try to formulate the model using generic keywords and syntaxes as we refer to the CP formulations by Laborie 2009, Ghédira 2013, Goel *et al*. 2015, and IBM ILOG CPOptimizer (IBM, 2015).

The proposed model is built upon the following three decision variables:

$Itv_j$ *interval* variable representing the $j$-th job with the size of $\tau^t_{pkup_j,drop_j}$

$ItvAlt_{j,d}$ *optional interval* variable representing the $j$-th job of drone $d$.

$Seq_d$ collection of interval variables ($Itv_j$) assigned to drone $d$.

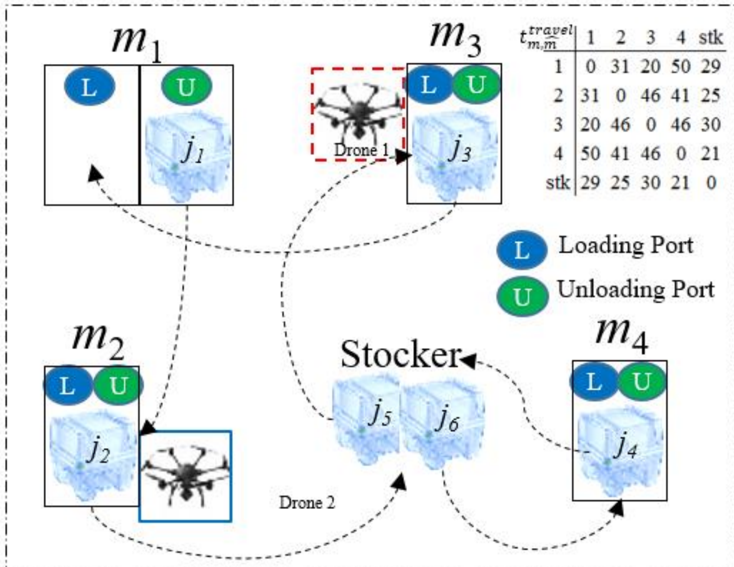


Figure 1: A tour representation of the sample problem with 2-drone.

An *interval* variable represents an interval of time during which, for example, a task occurs. An *interval* variable can represent optional tasks to be scheduled. The domain of an *interval* variable *x* is a subset of $\{Absent\} \cup \{[s, e)| \; s, e \in \mathbb{Z}, s \leq e\}$. As any decision variable in an optimization problem, an *interval* variable *x* is said to be fixed if its domain is reduced to a singleton. When an *interval* is present, *s* represents the start time, and *e* the end time. Note each job interval ($Itv_j$) has the size of its transfer-time between $pkup_j$ and $drop_j$. Then, the MTDSP is concisely modeled in CP by Ham (2017) as follows:

$$minimize \; max \left\{ \boldsymbol{endOf}\left(Itv_j\right)\right\} \tag{1}$$

$$\boldsymbol{Alternative}\left(Itv_j, ItvAlt_{j,d}\right): \; Itv_j \rightarrow j \leftarrow ItvAlt_{j,d} \tag{2}$$

$$\boldsymbol{noOverlap}\left(Seq_d, \tau^t_{m\hat{m}}\right) \; \forall d \in D \tag{3}$$

$$\begin{aligned} &\boldsymbol{type} \text{ function } \vartheta \left(Seq_d, ItvAlt_{j,d}: \; Seq_d \rightarrow d \leftarrow ItvAlt_{j,d}\right) \\ &initialize \left[\vartheta\left(Seq_d, ItvAlt_{j,d}\right)\right] = \iota^d \quad, \forall d \end{aligned} \tag{4}$$

$$\boldsymbol{endBeforeEnd}\left(Itv_j, Itv_{\hat{j}}\right) \quad \forall j, \hat{j} = \; ebe_j \tag{5}$$

$$\underline{w_j} \; \leq \boldsymbol{startOf}\left(Itv_j\right) \quad \forall j \in J \tag{6}$$

$$\boldsymbol{endOf}\left(Itv_j\right) \leq \overline{w_j} \quad \forall j \in J \tag{7}$$

The integer expression *endOf*(*j*) represents the end of *interval* variable *j* whenever the *interval* variable is present (otherwise, its value is 0 by default). Now, the objective function (1) seeks to minimize the maximum completion time of all jobs. Let *alternative*(*J*, {*D*}) constraint prescribe an exclusive alternative relationship among {*D*}. Namely, if *interval J* is present then exactly one of *intervals* {*D*} is present and it is synchronized together with this chosen one (same start/end value). Constraint (2) ensures that each job is assigned to exactly one drone in our model. Next, let *noOverlap*[*Seq*, $\boldsymbol{\Delta}$] constraint on a sequence variable *Seq* state that the sequence defines a chain of non-overlapping intervals, and any interval in the chain is constrained to end before the start of the next interval in the chain with the minimal distance $\boldsymbol{\Delta}$. Constraint (3) requires drone to observe the sequence-dependent transfer-time between locations. Note each job interval ($Itv_j$) already considers the size of its transfer-time between $pkup_j$ and $drop_j$. However, there is another transfer, flying to origin location ($pkup_{\vec{j}}$), which is equivalent to $\tau^{t}_{drop_{\overleftarrow{j}},\, pkup_{\vec{j}}}$ when $\overleftarrow{j}$ is immediately preceding $\vec{j}$ in the sequence of drone. Constraint (4) links the last-known location of drone to the beginning of sequence variable in order to consider the initial location of drone. Constraint (5) ensures the precedence-relationship between jobs. The parameter, $ebe_j$, holds the precedence-relationship. Finally, Constraints (6) and (7) ensure the time-window: earliest start time and latest completion time.

Figure 2 represents an optimal schedule for the earlier sample problem with 6-job and 2-drone. The drone 1 starts its tour with $j_3$. Note there is no initial transfer time for the drone since the drone is currently hovering over $m_3$ where $j_3$ is located. After the drone delivers $j_3$ from $m_3$ to $m_1$ at 50, the done next picks up $j_1$ at the same location, drops $j_1$ at $m_2$ at 81, then flies to stocker for picking up $j_6$ at 106, and finally delivers $j_6$ to $m_4$ at 127. The drone 2 follows a similar tour.

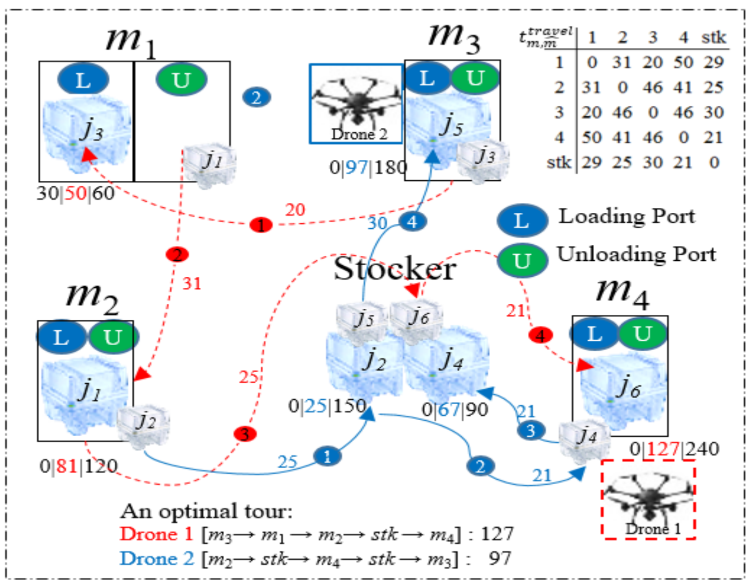


Figure 2: An optimal schedule of the sample problem with 6-job and 2-drone.

Although CP has excelled most notably in scheduling applications (Baptiste et al., 2012), the performance of the original CP formulation can be further improved by using variable orderings heuristics (VOH). In the CP search strategy, the order of the search phases in the array is important. The variables will be instantiated phase by phase starting by the first phase of the array. There are some models where giving such an order can have a dramatic impact on the solution time. Beck et al. (2004) discussed that the "likelihood of finding a solution" could be treated probabilistically. For a given decision there is some probability over all possible subsequent decisions that the choice will lead to a solution. They demonstrated that different variable ordering heuristics do exhibit different levels of promise, wherein the promise is the ability to make choices that lead to a solution.

For the purposes of our investigation, we implement instantiation orderings over variable groups defined as: $\mathcal{V} \coloneqq \{ItvAlt_{j,d}, Seq_d\}$, similar to the method proposed by Booth et al. (2016). Within these variable groups we investigate orderings on the sets. Problem variables not included in the selected subset will be instantiated after those selected. For instance, the variables $Seq_d$ will be firstly instantiated when

$\boldsymbol{searchPhase}(Seq_d)$ is added in CP model. In experiments, $\boldsymbol{CP^{voh}_{default}}(\boldsymbol{CP^{voh}_{jd}}, \boldsymbol{CP^{voh}_{seq}})$ represent the CP search with the variable instantiation strategy for *default*, $ItvAlt_{j,d}$, and $Seq_d$, respectively.

## 3 COMPUTATIONAL EXPERIMENTS

In this section, the effectiveness of the proposed model is compared. MIP, CP and flow control models are all coded in IBM OPL 12.7.0 on a personal computer with an Intel Core i5-3537 @ 2.5 Ghz processor and 8 GB RAM.

For the MTDSP, a total of 60 test problem instances are randomly generated. The instances are divided to six different sizes: 5, 10, 25, 50, 75, and 100 customers, with 10 replications for each. Depending on job-size, different sizes of drones (1-50) are assumed. In particular, the 5 (10, 25, 50, 75, 100) jobs are tested with 1 and 2 (3 and 5, 6 and 13, 13 and 25, 18 and 38, 25 and 50) drones, configuring each drone handles 2 jobs, respectively, on average. Note each job contains two transfers: flying to origin and flying to destination, which makes a total of 4 transfers per drone, respectively, on average. Finally, each instance is tested with the five different models: $\boldsymbol{CP^{voh}_{default,}}$ $\boldsymbol{CP^{voh}_{jd,}}$ and $\boldsymbol{CP^{voh}_{seq}}$, which leads to a total of 600 runs as shown in Table 2. In addition, best feasible solutions founded in 1, 3, 5, 10, and 15 s runtime are collected in order to understand the capability of the proposed model from the real-time scheduling perspective.

Table 2. Experimental Design for MTDSP in Fab.

| Factors | Levels | Sizes |
|---|---|---|
| Jobs | 6 | 5, 10, 25, 50, 75, 100 |
| Models | 3 | $CP^{voh}_{default}, CP^{voh}_{j,d}, CP^{voh}_{seq}$ |
| Runs | 6×3×10 reps. | 180 |

The experiment assumes that every job has the precedence-relationship with another. All customers are distributed across an 80-meter square region. An initial location of each drone is set to be equal to its index for simplicity's sake ($\iota^d = d$). All the test instances and CP log files are located at the following space: https://drive.google.com/open?id=0B85VSacgqRfTYUE5Q1BNemw0bE0.

Table 3 summarizes the computational results of test problem instances of MTDSP. The table reports count of feasible solution, proven optimality, and gap against best solution found, according to different job-size, models, and runtimes. Columns 1 (2) show different job-sizes and models. Columns 3-7 (8-12, 13-17) record the count of feasible solution, the count of proven optimality, and the gap against best solution found for the different runtimes, respectively. $\boldsymbol{CP^{voh}_{j,d}}$ successfully proved optimality of all test instances (5, 10, 25, 50, 75, 100 jobs) within 5 s.

Table 3. Count of feasible solution, proven optimality, and gap against best solution found, according to different job-size, models, and runtimes.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Count of feasible solutions found | | | | | Count of optimal solutions found | | | | | Gap against best solution found | | | | |
| Jobs | Models | 1 s | 3 s | 5 s | 10 s | 15 s | 1 s | 3 s | 5 s | 10 s | 15 s | 1 s | 3 s | 5 s | 10 s | 15 s |
| 5 | $\boldsymbol{CP^{voh}_{default}}$ | 10 | 10 | 10 | 10 | 10 | 6 | 7 | 7 | 9 | 10 | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| | $\boldsymbol{CP^{voh}_{seq}}$ | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| | $\boldsymbol{CP^{voh}_{j,d}}$ | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| 10 | $\boldsymbol{CP^{voh}_{default}}$ | 10 | 10 | 10 | 10 | 10 | 5 | 5 | 5 | 5 | 5 | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| | $\boldsymbol{CP^{voh}_{seq}}$ | 10 | 10 | 10 | 10 | 10 | 7 | 10 | 10 | 10 | 10 | 0.1% | 0.0% | 0.0% | 0.0% | 0.0% |
| | $\boldsymbol{CP^{voh}_{j,d}}$ | 10 | 10 | 10 | 10 | 10 | 6 | 10 | 10 | 10 | 10 | 0.3% | 0.0% | 0.0% | 0.0% | 0.0% |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 25 | $CP^{voh}_{default}$ | 10 | 10 | 10 | 10 | 10 | 9 | 9 | 9 | 10 | 10 | 0.8% | 0.7% | 0.3% | 0.0% | 0.0% |
| | $CP^{voh}_{seq}$ | 10 | 10 | 10 | 10 | 10 | 8 | 9 | 10 | 10 | 10 | 1.4% | 0.3% | 0.0% | 0.0% | 0.0% |
| | $CP^{voh}_{j,d}$ | 10 | 10 | 10 | 10 | 10 | 9 | 9 | 10 | 10 | 10 | 0.9% | 0.8% | 0.0% | 0.0% | 0.0% |
| 50 | $CP^{voh}_{default}$ | 9 | 10 | 10 | 10 | 10 | 7 | 10 | 10 | 10 | 10 | 0.1% | 0.0% | 0.0% | 0.0% | 0.0% |
| | $CP^{voh}_{seq}$ | 10 | 10 | 10 | 10 | 10 | 0 | 8 | 8 | 10 | 10 | 99.5% | 1.2% | 0.2% | 0.0% | 0.0% |
| | $CP^{voh}_{j,d}$ | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| 75 | $CP^{voh}_{default}$ | 0 | 10 | 10 | 10 | 10 | 0 | 10 | 10 | 10 | 10 | nf | 0.0% | 0.0% | 0.0% | 0.0% |
| | $CP^{voh}_{seq}$ | 10 | 10 | 10 | 10 | 10 | 0 | 0 | 2 | 8 | 9 | 255.1% | 61.6% | 11.7% | 0.3% | 0.1% |
| | $CP^{voh}_{j,d}$ | 10 | 10 | 10 | 10 | 10 | 4 | 10 | 10 | 10 | 10 | 3.2% | 0.0% | 0.0% | 0.0% | 0.0% |
| 100 | $CP^{voh}_{default}$ | 0 | 8 | 10 | 10 | 10 | 0 | 0 | 8 | 9 | 10 | nf | 5.2% | 0.8% | 0.0% | 0.0% |
| | $CP^{voh}_{seq}$ | 0 | 10 | 10 | 10 | 10 | 0 | 0 | 0 | 0 | 6 | nf | 159.3% | 77.1% | 23.3% | 3.8% |
| | $CP^{voh}_{j,d}$ | 0 | 10 | 10 | 10 | 10 | 0 | 5 | 10 | 10 | 10 | nf | 2.7% | 0.0% | 0.0% | 0.0% |

*nf indicates no feasible solution was found in one of the instances at least.*

## 4 CONCLUSION

Inspired by Amazon's significant expansion of its army of warehouse robots and the adaptation of drones for final-mile small parcel delivery, this study has mixed these two emerging technologies and proposed a drone-powered material transfer system in a fab, wherein jobs could be picked up by drone, transferred, and dropped off. Contrary to OHT which utilizes rail mounted near the ceiling to transfer jobs via vehicles, this study proposes a delivery of jobs via drone, tearing down traditional rails and vehicles, and opens up an whole new way of designing a fab. A high tower-like fab can be foreseen when drone replaces OHT. For this coming drone era, a detailed scheduling system, that simultaneously orchestras a fleet of drones and a set of jobs in real-time, is studied. A job might be characterized by origin, destination, priority, and precedence-relationship. In particular, the precedence-relationship occurs when drones are competing for limited number of ports. The computational study demonstrates the proposed $\boldsymbol{CP^{voh}_{j,d}}$ impressively proved optimality of all test instances (5, 10, 25, 50, 75, 100 jobs) within 5 s when each drone is loaded with 4 transfers on average. As this is the first paper to address the use of a fleet of drones in a fab, several potential areas can be foreseen for future research:

**Zone control**: The zone control method, segmenting flow paths into zones (Berman et al., 2003) and dedicating a subset of drones to each zone, can be introduced to decompose the original problem, if the real-time performance of the proposed method becomes problematic, especially when there are very large number of jobs and drones, that must be concurrently scheduled.

**Unrelated parallel drones**: In this paper, identical parallel drones are assumed, but a drone might have its variants: one has only one carrier and another has two carriers. The drone equipped with two carriers will be able to greatly reduce a delivery-time, by picking up and dropping off jobs at one stop.

**Simulation study**: It will be very interesting to quantify the difference between a traditional OHT-based transfer and this drone-based transfer, in terms of delivery time and cost. A drone is obviously much faster than a OHT vehicle. Furthermore, OHT system costs multi-billion dollar capital investment, whereas a drone costs much less.

## REFERENCES

Baptiste, P., Le Pape, C., and Nuijten, W. 2012. "Constraint-based scheduling: applying constraint programming to scheduling problems (Vol. 39). *Springer Science & Business Media*.

Beck, J. C., Prosser, P., and Wallace, R. J. 2004. "Variable ordering heuristics show promise." In *Proceedings of the International Conference on Principles and Practice of Constraint Programming*, 711-715. Springer Berlin Heidelberg.

Berman, S., Edan, Y., and Jamshidi, M. 2003. "Navigation of decentralized autonomous automatic guided vehicles in material handling." *IEEE Transactions on Robotics and Automation* 19(4):743-749.

Booth, K. E., Nejat, G., and Beck, J. C. 2016. "A Constraint Programming Approach to Multi-Robot Task Allocation and Scheduling in Retirement Homes." In *Proceedings International Conference on Principles and Practice of Constraint Programming* , 539-555. Springer International Publishing.

Ghédira, K, 2013. *Constraint Satisfaction Problems: CSP Formalisms and Techniques.* John Wiley & Sons.

Goel, V., Slusky, M., van Hoeve, W. J., Furman, K. C., and Shao, Y. 2015. "Constraint programming for LNG ship scheduling and inventory management." *European Journal of Operational Research* 241(3):662-673.

Ham, A. M., and Cakici, E. 2016. "Flexible job shop scheduling problem with parallel batch processing machines: MIP and CP approaches." *Computers & Industrial Engineering* 102:160-165.

Ham, A. M. 2017. "Scheduling of Drone-Based Material Transfer System in a Warehouse: MIP and CP approaches." *submitted for publication.*

IBM Software, 2015. IBM ILOG CPLEX Optimization Studio V12.6.3, *IBM.*

Laborie, P. 2009. "IBM ILOG CP Optimizer for detailed scheduling illustrated on three problems." In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, 148–162.

Lin, J. T., Wang, F. K., and Yen, P. Y. 2001. "Simulation analysis of dispatching rules for an automated interbay material handling system in wafer fab." *International Journal of Production Research* 39(6):1221-1238.

Wang, C. N., Lee, Y. H., Hsu, H. P., and Dang, V. H. 2016. "The heuristic preemptive dispatching rule using stocker for conveyor-based material handling system of 450-mm wafer fabrication." *The International Journal of Advanced Manufacturing Technology* 5(88):2179-2189.

## AUTHOR BIOGRAPHIES

**ANDY M. HAM** is an Associate Professor of Industrial & Systems Engineering at Liberty University. He previously worked for Samsung Electronics, ILOG, and AMD/Globalfoundries. He holds a Ph.D. in Industrial Engineering from Arizona State University and Masters in Operations Research from University of Texas at Austin. His research focuses on real-time scheduling of drones, robots, vehicles, machines, and humans in the fields of smart factory focusing on semiconductor and logistics focusing on taxi and parcel-delivery industry, via constraint programming approach. His email address is mham@liberty.edu.

**DJ KIM** is a Senior Member of Technical Staff in Operations Central Team at Micron Technology Inc., He received MBA from Georgetown University and M.S. in Engineering Management from Queensland University of Technology, Australia and has performed about 19 years of semiconductor simulation, scheduling, and modeling experiences. His e-mail is djkim@micron.com.