*Proceedings of the 2019 Winter Simulation Conference*
*N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.*

# TRANSFER ROBOT TASK SCHEDULING IN SEMICONDUCTOR MANUFACTURING

Andy Ham

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

## ABSTRACT

This paper studies a simultaneous scheduling of production and material transfer in a semiconductor manufacturing. The simultaneous scheduling approach has been recently adopted by warehouse operations, wherein transbots pick up jobs and deliver to pick-machines for processing that requires a simultaneous scheduling of jobs, transbots, and machines. However, both a large proportion of literature and real-world scheduling systems in semiconductor manufacturing consider only one side of the problem. We propose the most efficient solution for job shop scheduling problem (JSP) with transbots, significantly outperforming all other benchmark approaches in the literature.

## 1 INTRODUCTION

One of the urgent needs in a modern smart factory (Intel 2015) and warehouse (Amazon 2014; Alibaba 2017; Ocado 2018) is to integrate the scheduling of production-machines with transfer-robots (transbot hereafter), wherein a job must be transferred by a transbot between operations. Intel and most semiconductor manufacturers use transbots for point-to-point delivery as pictured on Figure 1(a). Amazon and Alibaba have employed transbots to deliver shelves from staging areas to pick-machines as pictured on Figures 1(b) and 1(c). Similarly, Ocado utilizes transbots to grab a crate, pull it up into their interior, and deliver to pick-machine as pictured on Figure 1(d). A robotic mobile fulfillment system (RMFS) is a new type of automated storage and part-to-picker order picking system, where transfer-task by transbot and and pick-task by robotic-arm are synchronized.


(a) Intel transbot (2015) (b) Amazon transbot (2014) (c) Alibaba transbot (2017) (d) Ocado transbot (2018)

Figure 1: Transbots for pick-up and drop-off tasks.

These two decisions (transbot-scheduling and machine-scheduling) are interrelated with each other and must be synchronized. However, the robotics community focuses on transbot-scheduling problems while the traditional operations research community studies on machine-scheduling problems as an independent effort. The effort of connecting these two isolated research streams has been started. One of the successful works has been noticed in a job shop scheduling with transbots ($JSP^{+transbots}$). The classical JSP schedules a set of jobs on a set of machines with the objective to minimize a maximum completion time over all jobs

(Cmax), subjected to the constraint that each job has an ordered set of operations, each of which must be processed on a predefined machine. In this new integrated approach, transbots perform a delivery task between two operations. Figure 2 shows the Gantt chart of the simultaneous scheduling of machine and transbot. For instance, J1 is transferred from stocker to M1 by V2. Then the same transbot V2 returns to stocker to pick up J5 and drops it off to M1. The contribution of this paper is to propose the most efficient solution for JSP$^{+transbots}$, significantly outperforming all other benchmark attempts in the literature.

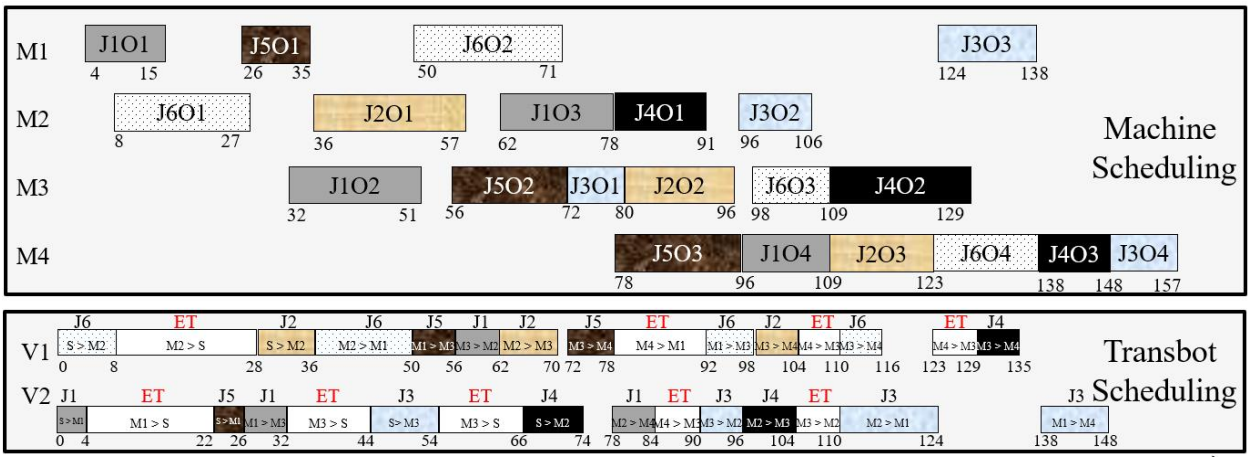


Figure 2. Gantt chart of the optimal solution for JSP EX104 instance (Bilge and Ulusoy, 1995).

## 2 LITERATURE REVIEW

Sawik (1996) noticed that neglecting transportation times at the tactical planning level, as well as lack of appropriate coordination between a schedule for transbots and machines, can have severe consequences. This can easily lead to bottlenecks in the system and its underutilization. Transportation times can also contribute to machine idle time if machines must wait for the delivery of the next part for processing. To bridge the gap, several researchers have demonstrated the benefits of coordinating transbot-schedule with machine-schedule called simultaneous scheduling (SS) as summarized in Table 1.

Table 1: The articles related JSP with transbots.

| Year | Author | Approach | nMachine | nTransbot | Publisher |
|---|---|---|---|---|---|
| 1995 | Bilge and Ulusoy | Heuristic | 4 | 2 | OR |
| 1996 | Sawik | Heuristic | 6 | 1 | MCM |
| 2004 | Sankar and Ponnambalam | GA | 4 | 1 | IEEE |
| 2006 | Khayat, Langevin, and Riopel | MIP/CP | 8 | 2 | EJOR |
| 2006 | Reddy and Rao | GA | 6 | 2 | IJAMT |
| 2008 | Deroussi, Gourgand, and Tchernev | SA | 4 | 2 | IJPR |
| 2009 | Caumond, Lacomme, Moukrim, and Tchernev | MIP | 4 | 1 | EJOR |
| 2009 | Yung, Ponnambalam, and Yogeswaran | ACO | 6 | 2 | IEEE |
| 2010 | Badr, Schmitt, and Göhner | Agent | 16 | 2 | IEEE |
| 2012 | Erol, Sahin, Baykasoglu, and Kaplanoglu | Agent | 8 | 2 | ASC |
| 2014 | Zeng, Tang, and Yan | NLP | 15 | 2 | ASC |
| 2016 | Baruwa and Piera | PN | 4 | 2 | IJPR |
| 2019 | This paper | CP | 4 | 2 | — |

Bilge and Ulusoy (1995) propose the most relevant benchmark instances on the JSP$^{+transbots}$ which have been used by several publications (Sankar and Ponnambalam 2004; Khayat et al.2006; Reddy and Rao 2006; Deroussi et al.2008; Erol et al. 2012; Zeng et al. 2014; Baruwa and Piera 2016). We use the same test instances to estimate the performance of the proposed model.

## 3 PROBLEM DESCRIPTION

Consider a job shop environment with transbots that consists of a set of machines ($m$, $M$), the load/unload (L/U) stocker where parts enter and leave the system, and a set of identical transbots ($v$, $V$) used for transportation of parts between two machines. There is a set of jobs ($j$, $J$) to be processed on one or more machines. Each job $j$ consists of an ordered sequence of steps $s$. We refer the combination of job and step as an operation ($O_{js}$). Each operation $O_{js}$ must be processed on a dedicated machine $\mu_{js} \in M$ without preemption in JSP environment. A machine can perform at most one operation at a time. Each machine has input and output (I/O) buffers in which parts are stored before and after processing. The buffers serve as pickup and delivery (P/D) points for the transbots. Parts visit different machines in the system for different operations which in turn generate P/D requests for the transbots. A transbot performs a transportation operation between any two operations $O_{js}$ and $O_{j,s+1}$, to move a job from the source machine $\mu_{js}$ to the destination machine $\mu_{j,s+1}$ for the next processing. Transbots perform two types of trips: a loaded trip and an empty trip (ET). A loaded trip is a delivery operation where the transbot moves a part from the output buffer of a machine $\mu_{js}$ to the input buffer of another machine $\mu_{j,s+1}$. In an empty trip, the transbot moves from an idle position at a machine $m$ without carrying a job in order to pick up a job waiting to be transferred from $\mu_{js}$ to $\mu_{j,s+1}$, where $m \neq \mu_{js}$. Let $t_{m,\hat{m}}$ represents the travel time between any two machines $m$ and $\hat{m}$. The travel times are job independent and machine dependent (Baruwa and Piera 2015). Figure 3 shows the four different layout configurations for the JSP$^{+transbots}$ proposed by Bilge and Ulusoy (1995). Each layout is composed of four machines, one L/U station (stocker), and two transbots.

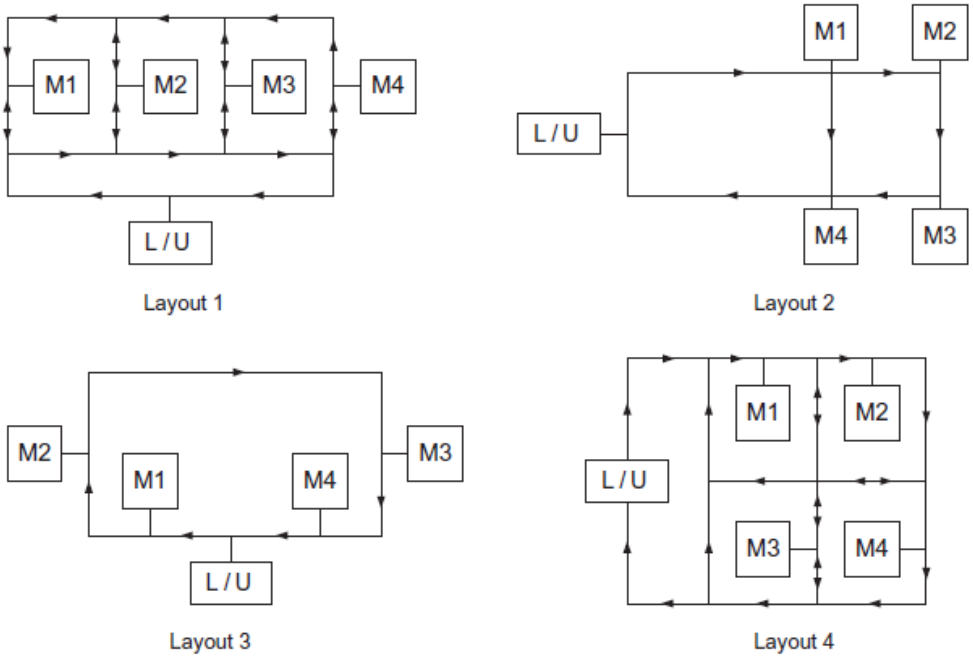


Figure 3. Layout configurations used in the test example (Bilge and Ulusoy 1995).

The simultaneous scheduling of machine and transbot (SSMT) problem is formulated as follows: Given the JSP$^{+transbots}$ environment described, determine the sequence of operations and the starting and completion times of each job on each machine, and the trips between machines together with the assignment of transportation tasks to vehicles according to the makespan minimization. The problem is formulated under the following assumptions given in Bilge and Ulusoy (1995):

- Machine operations and transbot trips are non-preemptive and there is sufficient I/O buffer space at each machine and L/U stocker to avoid deadlocks.
- Processing, loading and unloading times are deterministic and known.
- The number of transbots is known and they initially start from the stocker.
- Transbots have unit-capacity and they move along predetermined shortest paths, with the assumption of no delay due to congestion and travel times on each segment of the path are known.

## 4 METHODOLOGY

CP is an optimization paradigm that can handle a large number of constraint types that go beyond MIP-style linear constraints. While MIP focuses on the objective function and its optimality, CP focuses on satisfying the constraints and it is designed to find feasible solutions faster. CP is a suitable technique for solving sequencing and scheduling applications, feasibility problems, and highly constrained problems (Edis and Ozkarahan, 2011), outperforming MIP far exceedingly (Ham and Cakici 2016). The search within IBM ILOG CP Optimizer is equipped with the presolve functionality, some constraint propagation algorithms, the temporal linear relaxation used to guide the search and the two search space exploration strategies that are used concurrently: the Large Neighborhood Search (for producing good quality solutions) and Failure-Directed Search (for proving infeasibility or optimality). Large Neighborhood Search (LNS) is a component of CP Optimizer automatic search for scheduling consisting of successive relaxation and re-optimization phases. It operates by first finding an initial feasible solution. Then, a number of iterations are carried out, each iteration comprising a relaxation step followed by a re-optimization of the relaxed solution. This process continues until some condition is satisfied, typically, when the solution can be proved to be optimal or when a time limit is reached (Laborie, Rogerie, Shaw, and Vilím 2018). Because of space limitations, we recommend readers review Laborie and Rogerie (2008), Laborie, Rogerie, Shaw, and Vilím (2009), and IBM Software (2015).

In this SSMT problem, we need to explicitly model the transbot's task as well as the machine's task. We characterize each operation *o* by *j* (associated job id), *p* (order of operation *o* of job *j*), *pt* (processing time), *m* (machine ID at the current operation), *pm* (machine ID at the very preceding operation, stocker ID: if *o* is the first operation), and *isLast* (1: if step *o* is the last sequenced operation of job *j*, 0: o/w).
We model the SSMT problem with the following decision variables:

- $Pkup_{o\in O}$: interval variable representing pick-up task for operation *o* (interval for transbot to approach to a source where a job is completed at the preceding operation)
- $Drop_{o\in O}$: interval variable representing drop-off task for operation *o* (interval for transbot to move to a destination where a job will be processed)
- $JobOnMch_{o\in O}$: interval variable representing actual processing time of operation *o* on machine
- $Move4Pkup_{o\in O,v\in V}$: interval variable representing the assignment of each operation to one of the transbots for pick-up task
- $Move4Drop_{o\in O,v\in V}$: interval variable representing the assignment of each operation to one of the transbots for drop-off task
- $SeqMch_{m\in M} \leftarrow [JobOnMch_{o\in O:\, o.mch=m}]$: sequence variable representing all permutation sequences of the interval variables assigned to the sequence on each machine

- $SeqVeh_{v\in V} \leftarrow \left[Move4Pkup_{o\in O,v\in V} \cup Move4Drop_{o\in O,v\in V}\right]$ sequence variable representing all permutation sequences of the interval variables assigned to the sequence on each transbot

Then, the JSP$^{+\text{transbots}}$ can be formulated into CP as it follows:

| **CP1 formulation for JSP$^{+\text{transbots}}$ (Ham 2019)** | | |
|---|---|---|
| $Min\ Max_{o\in O}\{endOf(JobOnMachine_o)\}$ | | (1) |
| $endBeforeStart(JobOnMachine_o, JobOnMachine_{\hat{o}})$ | $\forall(o,\hat{o}) \in O: j(o) = j(\hat{o}) \wedge p(\hat{o}) = p(o) + 1$ | (2) |
| $endBeforeStart(JobOnMachine_o, Pkup_{\hat{o}})$ | $\forall(o,\hat{o}) \in O: j(o) = j(\hat{o}) \wedge p(\hat{o}) = p(o) + 1$ | (3) |
| $alternative(Pkup_o, [Move4Pkup_{ov}]_{v\in V})$ | $\forall o \in O$ | (4) |
| $alternative(Drop_o, [Move4Drop_{ov}]_{v\in V})$ | $\forall o \in O$ | (5) |
| $endBeforeStart\left(Pkup_o, Drop_o, t_{pm(o),m(o)}\right)$ | $\forall o \in O$ | (6) |
| $endBeforeStart(Drop_o, JobOnMachine_o)$ | $\forall o \in O$ | (7) |
| $endBeforeStart(Move4Pkup_{ov}, Move4Drop_{ov})$ | $\forall o \in O, v \in V$ | (8) |
| $endBeforeStart(Move4Drop_{ov}, JobOnMachine_o)$ | $\forall o \in O, v \in V$ | (9) |
| $prev(seqVeh_v, Move4Pkup_{ov}, Move4Drop_{ov})$ | $\forall o \in O, v \in V$ | (10) |
| $presenceOf(Move4Pkup_{ov}) = presenceOf(Move4Drop_{ov})$ | $\forall o \in O, v \in V$ | (11) |
| $noOverlap(SeqVeh_v, T)$ | $\forall v \in V$ | (12) |
| $noOverlap(SeqMch_m)$ | $\forall m \in M$ | (13) |
| $Interval\ JobOnMachine_o\ size\ pt(o), Pkup_o, Drop_o$ | $\forall o \in O$ | (14) |
| $Interval\ Move4Pkup_{ov}\ optional, Move4Drop_{ov}\ optional$ | $\forall o \in O, v \in V$ | (15) |
| $Sequence\ SeqVeh_v, SeqMch_m$ | $\forall v \in V, m \in M$ | (16) |

Objective (1) minimizes the time required to complete all operations (makespan). The expression *endOf*(*i*) represents the end of interval variable *i* whenever the interval variable is present (otherwise, its value is 0 by default). Constraints (2–3) ensure the precedence relation between the operations associated with each job. In particular, Constraint (3) enforces to complete a preceding operation ($o$) at a machine before starting pick-up task for the subsequent operation ($\hat{o}$) by transbot. Constraint (4) assigns each pick-up task to exactly one of transbots. Both intervals of $Pkup_{o\in O}$ and $Move4Pkup_{o\in O,v\in V}$ start and end together with this chosen one. Similarly, Constraint (5) assigns each drop-off task to exactly one of transbots. Both intervals of $Drop_{o\in O}$ and $Move4Drop_{o\in O,v\in V}$ start and end together. Constraint (6) ensures the precedence relation between pick-up and drop-off tasks associated with each operation. Constraint (7) makes sure to first complete a drop-off task by a transbot before being processed by a machine at each operation. Constraints (8–9) are redundant with Constraints (6–7). However, they help CP engine to escalate the search. Constraint (10) ensures that no other job can be ordered between consecutive pick-up and drop-off tasks on a given transbot at the same operation. Constraint (11) forces to employ the same transbot for pick-up and drop-off tasks at the same operation. Constraint (12) forces a transbot to perform at most one task at a time. The expression *noOverlap* (*s, T*) 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. This expression is typically useful for modeling disjunctive resources. If a transition distance matrix *T* is specified, it defines the minimal distance that must separate two consecutive intervals in the sequence. The matrix *T* is used herein to represent the location-dependent job transfer-time by transbot. Similarly, Constraint (13) forces a machine to perform at most one operation at a time. Finally, Constraints (14–16) define the CP decision variables. The specialized keywords of CP enable modelers to develop a concise code, compared to the MIP formulation, although a certain degree of ingenuity and insight into the problem is required to recognize how a problem can be formulated by CP.

## 5 COMPUTATIONAL STUDY

In this section, the effectiveness of the proposed model is examined. The CP and flow control models are all coded in IBM OPL 12.8.0 on a personal computer with an Intel® Core i7-4770 CPU with 16 GB of RAM. The Gantt-chart is created by CSPI ezDFS (2019).

The 82 test problems proposed by Bilge and Ulusoy (1995) are grouped into two sets. The first set contains 40 instances whose $t_{m,\hat{m}}/pt$ ratios are greater than 0.25, and the other set contains 42 instances with $t_{m,\hat{m}}/pt$ lower than 0.25. We adopted the first 40 instances in this study. Each instance code is designated with prefix EX followed by two digits that indicate the job set and the layout. Table 5 shows the performance comparison of CP with five off-line and two on-line scheduling algorithms for test problems with t/p > 0.25. The off-line approaches are: sliding time window heuristic (STW) (Bilge and Ulusoy 1995), two hybrid genetic algorithms, AGA (Abdelmaguid and Nassef 2010) and PGA (Reddy and Rao 2006), a hybrid local search with simulated annealing (SALS) (Deroussi, Gourgand, and Tchernev 2008), and anytime layered search (ALS) (Baruwa and Piera 2015). They reported that ALS found two new best known solutions and outperforms the other algorithms with the exception of EX103. The on-line algorithms are taken from Erol, Sahin, Baykasoglu, and Kaplanoglu (2012): a multi-agent system (MAS) and the shortest travelling distance (STD) rule. The CPU time is recorded in second.

Here the results are quite astounding. For instance, both CP1 and CP2 find an optimal of EX22 instantly while ALS takes 100 s. In summary, CP2 converges to optimal for all 40 problem instances very quickly. Most of them converge to optimal in a less than a second. The CP1 also finds optimal for all 40 problem instances, but it fails to converge to optimal except six instances. Furthermore, both CP models obtain the optimal solution for EX103 instance which has been failed by all other attempts in the literature. Lastly, when the simultaneous scheduling is compared to the traditional machine-only scheduling, the makespan is reduced by 10.69% in average.

Table 2. Performance comparison of CP with existing approaches for problems with t/p > 0.25.

| | STW | AGA | PGA | SALS | MAS | STD | ALS | | | CP1 | | | CP2 (Ham 2019) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | Obj | Obj | Obj | Obj | Obj | Obj | Obj | CPU | Converge | Obj | CPU | Converge | Obj | CPU | Converge |
| EX11 | 96 | 96 | 96 | 96 | 130 | 126 | **96** | 138.50 | 800.80 | 96 | 0.05 | — | **96** | 0.14 | 0.83 |
| EX12 | 82 | 82 | 82 | 82 | 98 | 104 | **82** | 39.20 | 800.60 | 82 | 0.03 | — | **82** | 0.04 | 0.44 |
| EX13 | 84 | 84 | 84 | 84 | 109 | 110 | **84** | 145.10 | 762.50 | 84 | 0.03 | — | **84** | 0.07 | 0.39 |
| EX14 | 108 | 103 | 103 | 103 | 168 | 164 | **103** | 510.20 | 832.80 | 103 | 0.38 | — | **103** | 0.92 | 1.26 |
| EX21 | 105 | 102 | 100 | 100 | 143 | 147 | 100 | 282.40 | — | 103 | 1.47 | — | **100** | 0.94 | 2.21 |
| EX22 | 80 | 76 | 76 | 76 | 86 | 104 | **76** | 100.50 | 666.20 | **76** | 0.05 | 0.06 | **76** | 0.16 | 0.17 |
| EX23 | 86 | 86 | 86 | 86 | 98 | 118 | 86 | 96.60 | — | 86 | 0.06 | — | **86** | 0.08 | 1.59 |
| EX24 | 116 | 108 | 108 | 108 | 169 | 172 | 108 | 475.90 | — | 108 | 0.79 | — | **108** | 2.22 | 5.02 |
| EX31 | 105 | 99 | 99 | 99 | 142 | 138 | 99 | 27.70 | — | 99 | 1.42 | — | **99** | 0.48 | 1.26 |
| EX32 | 88 | 85 | 85 | 85 | 114 | 116 | 85 | 44.90 | — | 85 | 0.03 | — | **85** | 0.31 | 0.49 |
| EX33 | 86 | 86 | 86 | 86 | 103 | 126 | 86 | 617.30 | — | 86 | 0.11 | — | **86** | 0.26 | 0.26 |
| EX34 | 116 | 111 | 111 | 111 | 167 | 182 | 111 | 414.90 | — | 111 | 4.59 | — | **111** | 1.23 | 2.08 |
| EX41 | 118 | 112 | 112 | 112 | 198 | 220 | 112 | 255.40 | — | 113 | 236.2 | — | **112** | 17.84 | 22.42 |
| EX42 | 93 | 88 | 87 | 87 | 129 | 151 | 87 | 268.70 | — | 87 | 3.85 | — | **87** | 1.61 | 3.88 |
| EX43 | 95 | 89 | 89 | 89 | 155 | 143 | 89 | 216.50 | — | 89 | 15.81 | — | **89** | 0.29 | 2.37 |
| EX44 | 126 | 126 | 126 | 121 | 242 | 247 | 121 | 452.00 | — | 126 | 6.69 | — | **121** | 2.98 | 7.60 |
| EX51 | 89 | 87 | 87 | 87 | 130 | 124 | **87** | 18.40 | 857.10 | 87 | 2.72 | — | **87** | 0.37 | 1.39 |
| EX52 | 69 | 69 | 69 | 69 | 98 | 101 | **69** | 98.70 | 869.70 | 69 | 2.48 | — | **69** | 0.46 | 0.46 |
| EX53 | 76 | 74 | 74 | 74 | 109 | 103 | **74** | 139.40 | 821.70 | 74 | 0.66 | — | **74** | 0.08 | 1.23 |
| EX54 | 99 | 96 | 96 | 96 | 168 | 168 | **96** | 223.20 | 885.00 | 96 | 0.1 | — | **96** | 1.29 | 2.18 |
| EX61 | 120 | 118 | 118 | 118 | 153 | 162 | 118 | 74.70 | — | 118 | 2.8 | — | **118** | 1.42 | 4.40 |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EX62 | 100 | 98 | 98 | 98 | 123 | 135 | 98 | 66.60 | — | **98** | 0.16 | 0.24 | **98** | 0.35 | 0.39 |
| EX63 | 104 | 104 | 103 | 103 | 128 | 143 | 103 | 902.60 | — | 103 | 0.27 | — | **103** | 0.65 | 0.66 |
| EX64 | 120 | 120 | 120 | 120 | 189 | 190 | 120 | 370.20 | — | 126 | 2.69 | — | **120** | 2.31 | 3.72 |
| EX71 | 119 | 115 | 111 | 111 | 129 | 143 | 111 | 549.30 | — | 114 | 0.88 | — | **111** | 54.82 | 406.28 |
| EX72 | 90 | 79 | 79 | 79 | 92 | 109 | 79 | 2303.30 | — | 79 | 2.61 | — | **79** | 0.57 | 1.22 |
| EX73 | 91 | 86 | 83 | 83 | 93 | 109 | 83 | 2403.30 | — | 83 | 5.59 | — | **83** | 1.77 | 2.58 |
| EX74 | 136 | 127 | 126 | 126 | 156 | 173 | 126 | 3598.00 | — | 126 | 254.99 | — | **126** | 125.53 | 792.88 |
| EX81 | 161 | 161 | 161 | 161 | 196 | 217 | 161 | 1300.60 | — | **161** | 0.1 | 0.11 | **161** | 0.06 | 0.07 |
| EX82 | 151 | 151 | 151 | 151 | 172 | 180 | 151 | 2.70 | — | **151** | 0.03 | 0.03 | **151** | 0.01 | 0.02 |
| EX83 | 153 | 153 | 153 | 153 | 172 | 182 | 153 | 9.30 | — | **153** | 0.03 | 0.04 | **153** | 0.01 | 0.01 |
| EX84 | 163 | 163 | 163 | 163 | 251 | 246 | 163 | 295.80 | — | **163** | 0.66 | 0.66 | **163** | 0.13 | 0.14 |
| EX91 | 120 | 118 | 116 | 116 | 178 | 163 | **116** | 57.00 | 2721.40 | 116 | 0.28 | — | **116** | 0.48 | 0.95 |
| EX92 | 104 | 104 | 102 | 102 | 123 | 128 | **102** | 284.00 | 2643.00 | 102 | 0.15 | — | **102** | 0.07 | 0.27 |
| EX93 | 110 | 106 | 105 | 105 | 119 | 132 | **105** | 54.10 | 2602.30 | 105 | 0.09 | — | **105** | 0.15 | 0.38 |
| EX94 | 125 | 122 | 122 | 120 | 181 | 190 | **120** | 1266.50 | 2867.30 | 120 | 4.53 | — | **120** | 0.58 | 1.30 |
| EX101 | 153 | 147 | 147 | 147 | 188 | 193 | 146 | 115.50 | — | 146 | 2.77 | — | **146** | 1.38 | 2.35 |
| EX102 | 139 | 136 | 135 | 135 | 154 | 164 | 135 | 3252.90 | — | 135 | 4.23 | — | **135** | 0.43 | 0.67 |
| EX103 | 143 | 141 | 139 | 138 | 158 | 180 | 139 | 66.60 | — | 137 | 8.19 | — | **137** | 1.05 | 1.30 |
| EX104 | 171 | 159 | 158 | 159 | 246 | 249 | 157 | 822.20 | — | 157 | 26.16 | — | **157** | 4.29 | 12.80 |
| Notes: Bold – Solution converged to optimal | | | | | | | | 559.02 | | | 14.87 | | | 5.70 | |

## 6 CONCLUSION

A simultaneous scheduling of production machine and material transfer robot in a job shop was studied, as factories and warehouses are predicted to soon adopt an automatic material transfer system powered by transbots. Here we propose a CP modeling for JSP$^{+\text{transbots}}$, significantly outperforming all other benchmark approaches in the literature. The success can be attributed to IBM CP Optimizer's recent application of machine learning techniques to portfolios of large neighborhoods and completion strategies in order to find the best combined method for the problem being solved. Two different CP models are proposed. The first model defines pick-up and drop-off as separate tasks, whereas the second merges these two transfer tasks into one by utilizing a customized transition distance matrix.

The future study can consider a fleet of transbots and a set of pick-stations in a warehouse. The most difficult challenge in this problem would be scalability. For instance, Ocado (2018) employs hundreds of transbots to fulfil 65,000 orders per week. In addition, synchronization of drop-off tasks by multiple transbots to arrive at the same pick-station with a minimum time gap for a given order is an additional complexity. As far as this author understands, Amazon (2014), Alibaba (2017), and Ocado (2018) are currently utilizing a rule-based approach to dispatch transbots to handle dynamic orders. The rule-based approach has a tunnel vision (Dabbas and Fowler 2003), lacking comprehensive view. Therefore, a near real-time scheduling approach can be a very interesting work for these companies.

## ACKNOWLEDGMENTS

This research was partially supported by CSPI Research Fund.

## REFERENCES

Abdelmaguid, T. F. and A. O. Nassef. 2010. "A Constructive Heuristic for the Integrated Scheduling of Machines and Multipleload Material Handling Equipment in Job Shops". *The International Journal of Advanced Manufacturing Technology* 46(9–12): 1239–1251.

Alibaba. 2017. "Inside Alibaba's smart warehouse staffed by robots" [Video file]. https://www.youtube.com/watch?v=FBl4Y55V2Z4, accessed 15th April 2019.

Amazon. 2014. Meet the robots making Amazon even faster [Video file]. https://www.youtube.com/watch?v=UtBa9yVZBJM, accessed 15th April 2019.

Badr, I., F. Schmitt, and P. Göhner. 2010. "Integrating transportation scheduling with production scheduling for FMS: An agent-based approach". In *2010 IEEE International Symposium on Industrial Electronics*, July 4th-7th, Bari, Italy, 3539-3544.

Baruwa, O. T. and M. A. Piera. 2015. "Identifying FMS Repetitive Patterns for Efficient Search-based Scheduling Algorithm: A Colored Petri Net Approach." *Journal of Manufacturing Systems* 35: 120–135.

Baruwa, O. T. and M. A. Piera. 2016. "A coloured Petri net-based hybrid heuristic search approach to simultaneous scheduling of machines and automated guided vehicles". *International Journal of Production Research* 54(16), 4773-4792.

Bilge, U. and G. Ulusoy. 1995. "A Time Window Approach to Simultaneous Scheduling of Machines and Material Handling System in an FMS." *Operations Research* 43 (6):1058–1070.

Caumond, A., P. Lacomme, A. Moukrim, and N. Tchernev. 2009. "An MILP for scheduling problems in an FMS with one vehicle". *European Journal of Operational Research* 199(3):706-722.

CSPI. 2019. "Production/Logistics Automation". http://www.cspi.co.kr/eng_html/business_01.php, accessed 15th April 2019.

Dabbas, R. M., and J. W. Fowler. 2003. "A new scheduling approach using combined dispatching criteria in wafer fabs". *IEEE Transactions on semiconductor manufacturing*, 16(3), 501-510.

Deroussi, L., M. Gourgand, and N. Tchernev. 2008. "A Simple Metaheuristic Approach to the Simultaneous Scheduling of Machines and Automated Guided Vehicles." *International Journal of Production Research* 46 (8):2143–2164.

Edis, E. B. and I. Ozkarahan. 2011. "A combined integer/constraint programming approach to a resource-constrained parallel machine scheduling problem with machine eligibility restrictions". *Engineering Optimization* 43(2):135–157.

El Khayat, G., A. Langevin, and D. Riopel. 2006. "Integrated production and material handling scheduling using mathematical programming and constraint programming". *European Journal of Operational Research* 175(3):1818-1832.

Erol, R., C. Sahin, A. Baykasoglu, and V. Kaplanoglu. 2012. "A Multi-agent Based Approach to Dynamic Scheduling of Machines and Automated Guided Vehicles in Manufacturing Systems." *Applied Soft Computing* 12 (6): 1720–1732.

Fattahi, P., M. Mehrebad, and F. Jolai. 2007. "Mathematical modeling and heuristic approaches to flexible job shop scheduling problems", *Journal of Intelligent Manufacturing*. 18:331–342.

Gerkey, B. P. and M. J. Matarić. 2004. "A formal analysis and taxonomy of task allocation in multi-robot systems". *The International Journal of Robotics Research* 23(9):939-954.

Ham, A. 2019. "Transfer-Robot Task Scheduling in a Job Shop". *International Journal of Production Research* (to appear).

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

IBM Software. 2018. IBM ILOG CPLEX Optimization Studio V12.8.0.

Intel. 2015. The Most Sophisticated Manufacturing Process in the World [Video file]. https://www.youtube.com/watch?v=-KTKg0Y1snQ, accessed 15th April 2019.

Korsah, G. A., A. Stentz, and M. B. Dias. 2013. "A comprehensive taxonomy for multi-robot task allocation". *The International Journal of Robotics Research*, 32(12), 1495-1512.

Laborie, P. and J. Rogerie. 2008. "Reasoning with Conditional Time-Intervals". In *FLAIRS conference*, May 15th-17th, Florida, USA, 555–560.

Laborie, P., J. Rogerie, P. Shaw, and P. Vilim. 2009. "Reasoning with Conditional Time-Intervals. Part II: An Algebraical Model for Resources". In *FLAIRS conference*, May 19th-21st, Florida, USA, 201–206.

Laborie, P., J. Rogerie, P. Shaw, and P. Vilím. 2018. "IBM ILOG CP optimizer for scheduling". *Constraints* 23(2):210-250.

Nunes, E., M. Manner, H. Mitiche, and M. Gini. 2017. "A taxonomy for task allocation problems with temporal and ordering constraints". *Robotics and Autonomous Systems* 90:55-70.

Ocado. 2018. "Inside A Warehouse Where Thousands Of Robots Pack Groceries [Video file]". https://www.youtube.com/watch?v=4DKrcpa8Z_E, accessed 15th April 2019.

Reddy, B. S. P. and C. S. P. Rao. 2006. "A hybrid multi-objective GA for simultaneous scheduling of machines and AGVs in FMS". *The International Journal of Advanced Manufacturing Technology* 31(5-6):602-613.

Sankar, S. S. and S. G. Ponnambalam. 2004. "An intelligent integrated scheduling model for flexible manufacturing system". In *IEEE Conference on Robotics, Automation and Mechatronics*, December 1st-3th, Singapore, 1095-1100.

Sawik, T. (1996). "A multilevel machine and vehicle scheduling in a flexible manufacturing system". *Mathematical and computer modelling* 23(7): 45-57.

Yung, T. W., S. G. Ponnambalam, and M. Yogeswaran. 2009. "Multi-objective ACO for integrated scheduling of machines and material handling equipment in flexible manufacturing systems". In *IEEE International Conference on Automation Science and Engineering CASE 2009*, August 22nd-25th, Bangalore, India,304-309.

Zeng, C., J. Tang, and C. Yan. 2014. "Scheduling of no buffer job shop cells with blocking constraints and automated guided vehicles". *Applied Soft Computing* 24:1033-1046.

## AUTHOR BIOGRAPHIES

**ANDY HAM** received Ph.D. in industrial engineering from Arizona State University in 2009, and M.S. in OR/IE from University of Texas at Austin in 2000. He is currently working as a full professor in Industrial and Systems Engineering, Liberty University, Virginia, while he serves as a technical consultant for Berkshire Grey: AI and Robotics based omni-channel fulfillment company. Prior to the current position, he worked for Samsung Electronics, Samsung Austin Semiconductor, GlobalFoundries, AMD, and IBM/ILOG, in the areas of modeling, real-time dispatching, real-time scheduling, supply chain management, and decision analysis. His research is currently focusing on real-time scheduling of drones and robots in smart factories, smart warehouses, and logistics industry. His research has been published in peer-reviewed journals such as IEEE Transactions on Automation Science and Engineering, IEEE Transactions on Semiconductor Manufacturing, Transportation Research Part C: Emerging Technologies, Applied Mathematical Modelling, International Journal of Production Research, Computers and Industrial Engineering, etc. His email address is mham@liberty.edu.