Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

HEURISTICS FOR ORDER-LOT PEGGING IN MULTI-FAB SETTINGS

Lars Mönch

Department of Mathematics and Computer Science
University of Hagen
Universitätsstraße 1
Hagen, 58097, GERMANY

Liji Shen

Supply Chain Management Group
WHU - Otto Beisheim School of Management
Burgplatz 2
Vallendar, 56179, GERMANY

John W. Fowler

Department of Supply Chain Management
Main Campus, PO BOX 874706
Arizona State University
Tempe, AZ 85287-4706, USA

ABSTRACT

In this paper, we study order-lot pegging problems in semiconductor supply chains. The problem deals with assigning already released lots to orders and with planning wafer releases to fulfill orders if there are not enough lots. The objective is to minimize the total tardiness of the orders. We propose a mixed integer linear programming (MILP) formulation for this problem. Moreover, we design a simple heuristic based on list scheduling and a biased random key genetic algorithm (BRKGA). Computational experiments based on problem instances from the literature for the single-fab case and newly proposed instances for the multi-fab setting are conducted. The results demonstrate that the BRKGA approach is able to determine high-quality solutions in a short amount of computing time.

1 INTRODUCTION

Semiconductor wafer fabrication facilities (wafer fabs) belong to the most complex existing manufacturing systems. Integrated circuits (ICs) are produced on thin discs (wafers) made from silicon or gallium arsenide. Each wafer fab contains hundreds of complicated machines, some of them are extremely expensive. The routes of the individual products may contain up to 800 operations, i.e. process steps, for the most advanced technologies. Lots, groups of wafers that travel together through a wafer fab, are the moving entities in wafer fabs. The cycle time, i.e. the time span between the release of material and its emergence as final product is up to 10 weeks in wafer fabs (Mönch et al. 2013). After the wafer fabrication step in wafer fabs, wafers are sent to sort facilities where ICs that do not meet the quality requirements are identified. The probed wafers are then sent to assembly where they are cut into individual ICs and the good ones are put into a package to allow connections with higher level devices. Finally, the packaged ICs are tested and labeled. The production of ICs takes place in semiconductor supply chains which might contain dozens of wafer fabs, sort facilities, and assembly and test (A/T) facilities.

In this paper we discuss a planning problem for foundries. Foundries manufacture ICs for a wide range of customers in varying quantities on a common manufacturing process. They typically operate following a make-to-order (MTO) strategy. The foundries business model is important in driving technological developments (Li, Huang, and Chen 2011).

The planning problem studied in the present paper deals with assigning wafer fabrication lots to specific customer orders. In addition, it determines the amount of wafers to be released into the wafer fabs if the already existing lots are not enough. This assignment activity is called order-lot pegging. It is a short-term planning problem that belongs to the demand fulfillment function in semiconductor supply chains (Mönch et al. 2018a; Mönch et al. 2018b). A single-fab version of the order-lot pegging (OLP) problem is studied by Kim and Lim (2012). Several heuristics based on dispatching rules and a simulated annealing (SA) scheme are proposed. In the present paper, we extend the OLP problem to a multi-fab version, abbreviated by MF-OLP. We propose a simple heuristic based on a dispatching rule and a more sophisticated one based on a BRKGA.

The rest of the paper is organized as follows. The problem is described in the next section. This includes a discussion of related work. The proposed heuristics are described in Section 3. The results of the conducted computational experiments are presented and analyzed in Section 4. Conclusions and future research directions are discussed in Section 5.

2 PROBLEM SETTING AND ANALYSIS

2.1 Problem Statement

We assume that we have m wafer fabs that run in parallel and have identical capabilities. A given set of N orders have to be satisfied from already released lots and newly released wafers from these wafer fabs during a planning horizon that consists of T equidistant periods. The overall setting is shown in Figure 1.

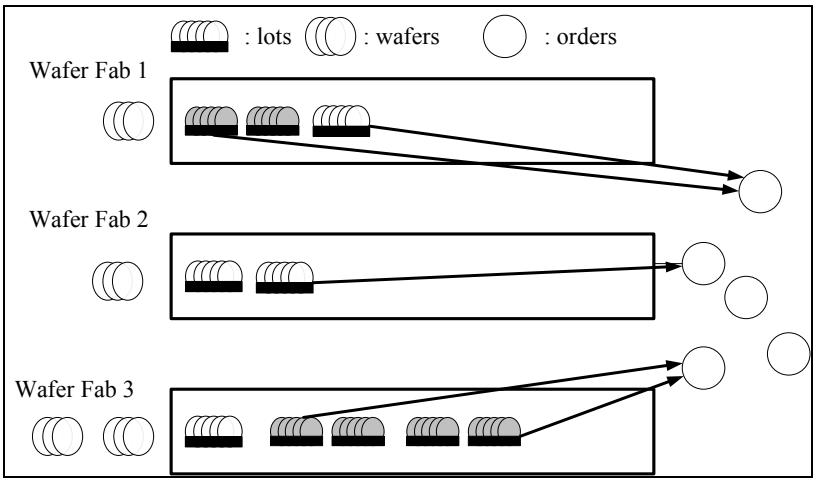

Figure 1: Overall problem setting.

Order i consists of q_i wafers. It has a due date d_i. There are $L_k, k = 1,\ldots,m$ already released lots in wafer fab k. Each lot l of wafer fab k has a remaining lead time of r_{lk} where the lead time is an estimate of the cycle time. Moreover, the number of wafers in lot l in wafer fab k is w_{lk}. Overall, we have L lots across all wafer fabs. There are compatibility matrices $A \in IR^{N \times L_k}$ whose entries $a_{il}^{(k)}$ are 1 if lot l of wafer fab k can be used to fulfill order i and zero otherwise.

We assume that orders can only be fulfilled from lots and wafers of the same wafer fab, i.e., if a first lot from a specific wafer fab is pegged to the order, then all the remaining required lots and wafers have to be from this wafer fab. This assumption is mainly justified by traceability reasons. Since the capacity of a wafer fab is finite, only Q_{tk} wafers can be launched in period t into wafer fab k. The lead time for newly released wafers to fulfill order i in wafer fab is s_{ik}.

The tardiness of order i is given by $T_i := \max(C_i - d_i, 0)$, where C_i is the completion time of order i. We are interested in minimizing the total tardiness (TT) of the orders which is defined by $TT := \sum_{i=1}^{N} T_i$ .This on-time delivery-related measure is important in a foundry setting. Note that the MF-OLP problem is somehow between production planning where release quantities are determined and scheduling where sequencing decisions are determined. However, the capacity modeling is less detailed in the MF-OLP problem compared to production planning where the available capacity is represented at the tool group level.

Since the MF-OLP problem contains the OLP problem which is NP-hard (cf. Kim and Lim 2012) as a special case, the MF-OLP problem is also NP-hard. Hence, we will design and test efficient heuristics in the present paper.

2.2 Related Work

There is a fairly small body of literature that deals with assigning lots to customer orders. The first stream of research is related to MTO, whereas the second stream deals with approaches for make-to-stock (MTS) situations. We start by discussing related work for the MTS case. The problem of assigning different sized lots to customer orders of different sizes in a single assembly facility is considered by Knutson et al. (1999). The objectives are maximizing the number of ICs sent to customers and the number of orders delivered on time and minimizing the excess inventory. For traceability reasons, a lot can only be used in a single order. Heuristics inspired by bin-packing are proposed. Additional heuristics for this problem are designed and evaluated in Fowler et al. (2000) and Carlyle et al. (2001). A lot-to-order matching problem for multiple product classes as a result of binning is considered by Boushell et al. (2008). A similar problem that looks at under- or overfilling customer orders in the face of uncertain lot sizes is studied by Ng et al. (2010). A robust optimization approach is proposed. A generalized version of this problem is studied by Sun et al. (2011). Downward product substitution is allowed when demand exceeds supply. All the studied lot-to-order matching problems are different from the MF-OLP problem since due dates are not considered in some cases and only a single assembly and test facility is assumed.

Next, we continue with the MTO case. Hard and soft pegging strategies are proposed by Bang et al. (2005) and Kim et al. (2008). Hard pegging refers to the situation where a lot is assigned to a single customer order and cannot be reassigned. Soft pegging, in contrast, allows repegging, for instance, when important orders arrive or due to machine breakdowns. It is shown by means of simulation studies that soft pegging approaches are able to significantly outperform hard pegging strategies under several experimental conditions. However, only a single wafer fab is considered in these papers. The OLP problem is studied for the first time by Kim et al. (2010). An MILP is formulated. Moreover, several simple, but fast heuristics based on the Earliest Due Date (EDD) dispatching rule are proposed and tested. Additional dispatching rules for this problem are designed by Kim et al. (2015). More efficient solution approaches for this problem are proposed by Kim and Lim (2012). This paper is the most pertinent previous work for the present paper. But again only a single wafer fab is considered which does not match with the foundry situation. This limitation is tackled in the present paper.

2.3 MILP Formulation and Problem Analysis

Next, we provide an MILP formulation which extends the MILP model formulated by Kim and Lim (2012) to the foundry situation. The following indices and sets are applied:

- i : order index, $i = 1,\ldots,N$
- l : lot index, $l = 1,\ldots,L_k$, $k = 1,\ldots,m$
- k : wafer fab index, $k = 1,\ldots,m$
- t : period index, $t = 1,\ldots,T$.

The model is based on the following parameters:

- N : number of orders
- L_k : number of lots in wafer fab k
- m : number of wafer fabs
- q_i : quantity of order i (in wafers)
- w_{lk} : number of wafers in lot l in wafer fab k
- d_i : due date of order i
- r_{lk} : remaining lead time of lot l in wafer fab k (in periods)
- s_{ik} : lead time of newly released wafers for order i into wafer fab k (in periods)
- Q_{tk} : maximum number of wafer that can be released in period t in wafer fab k
- $a_{il}^{(k)}$: 1, if lot l of wafer fab k can be used to satisfy order i, 0, otherwise.

The following decision variables are used in the model:

- x_{ilk} : number of wafers of lot l of wafer k fab assigned to order i
- y_{itk} : number of wafers to be released in period t in wafer fab k to fulfill order i
- z_{ilk} : 1, if lot l of wafer fab k is used to satisfy order i, 0, otherwise
- u_{itk} : 1, if wafers are released in period t in wafer fab k to satisfy order i, 0, otherwise
- f_{ik} : 1, if lots and/or wafers of wafer k are used to satisfy order i, 0, otherwise
- C_i : completion time of order i
- T_i : tardiness of order i.

The model itself can be formulated as follows:

$$\min \sum_{i=1}^{N} T_i \tag{1}$$

subject to

$$\sum_{k=1}^{m}\left(\sum_{l=1}^{L_k} x_{ilk} + \sum_{t=1}^{T} y_{itk}\right) = q_i \qquad i = 1,\ldots,N \tag{2}$$

$$\sum_{i=1}^{N} x_{ilk} \le w_{lk} \qquad k = 1,\ldots,m,\ l = 1,\ldots,L_k \tag{3}$$

$$\sum_{i=1}^{N} y_{itk} \le Q_{tk} \qquad t = 1,\ldots,T,\ k = 1,\ldots,m \tag{4}$$

$$x_{ilk} \le w_{lk} z_{ilk} \qquad k = 1,\ldots,m,\ l = 1,\ldots,L_k,\ i = 1,\ldots,N \tag{5}$$

$$y_{itk} \le Q_{tk} u_{itk} \qquad k = 1,\ldots,m,\ i = 1,\ldots,N,\ t = 1,\ldots,T \tag{6}$$

$$\sum_{k=1}^{m} f_{ik} = 1 \qquad i = 1,\ldots,N \tag{7}$$

$$u_{itk} \le f_{ik} \qquad i = 1,\ldots,N,\ t = 1,\ldots,T,\ k = 1,\ldots,m \tag{8}$$

$$z_{ilk} \le f_{ik} \qquad i = 1,\ldots,N,\ k = 1,\ldots,m,\ l = 1,\ldots,L_k \tag{9}$$

$$z_{ilk} \le a_{il}^{(k)} \qquad i = 1,\ldots,N,\ k = 1,\ldots,m,\ l = 1,\ldots,L_k \tag{10}$$

$$r_{lk} z_{ilk} \le C_i \qquad i = 1,\ldots,N,\ k = 1,\ldots,m,\ l = 1,\ldots,L_k \tag{11}$$

$$(t + s_{ik})u_{itk} \le C_i \qquad i = 1,\ldots,N,\ k = 1,\ldots,m,\ t = 1,\ldots,T \tag{12}$$

$$C_i - d_i \le T_i \qquad i = 1,\ldots,N \tag{13}$$

$$0 \le T_i,\ 0 \le C_i,\ 0 \le x_{ilk},\ 0 \le y_{itk} \qquad i = 1,\ldots,N,\ k = 1,\ldots,m,\ l = 1,\ldots,L_k,\ t = 1,\ldots,T \tag{14}$$

$$z_{ilk} \in \{0,1\},\ u_{itk} \in \{0,1\},\ f_{ik} \in \{0,1\} \qquad i = 1,\ldots,N,\ k = 1,\ldots,m,\ l = 1,\ldots,L_k,\ t = 1,\ldots,T. \tag{15}$$

The objective function (1) is the TT of all orders. The constraint set (2) ensures that each order is satisfied with wafers from already released lots and, if necessary, with newly released wafers. It is expressed by constraint set (3) that the number of wafers belonging to each lot is respected during the pegging process. Constraint set (4) ensures that the total number of wafers to be released into a fab does not exceed the maximum number of wafers that can be released in a given period. Constraint set (5) models that a lot is pegged to an order if at least a single wafer of this lot is assigned to that order. The same is expressed for newly released wafers in a period by constraint set (6). It is modeled by constraint set (7) that each order is assigned to exactly one wafer fab. The constraints (7), (8), and (9) ensure that only already released lots and newly released wafers of the same wafer fab can be used to fulfill an order. The compatibility of lots to orders is respected by constraint set (10). The tardiness of individual orders is calculated by the constraint sets (11), (12), and (13). The constraints (14) and (15) model the range of the decision variables.

Next, we recall the notation of an order split from Kim and Lim (2012) for solutions of OLP problem instances. Roughly speaking, an order i is split by order j if two lots or new wafer releases are used to satisfy i with (remaining) lead times $p_1 < p_2$ but there is another lot or wafer release that is used to fulfill order j with (remaining) lead time of p_3, $p_1 < p_3 < p_2$. In addition, the following second split situation is also possible. Orders i and j are both satisfied by two lots or wafer releases with (remaining) lead times $p_1 < p_2$. It is easy to see that in both situations an order split does not improve the TT value for the OLP problem. Since we assume due to traceability reasons that only lots and wafers from a single wafer fab can be used to fulfill an order, split orders are also not beneficial for the MF-OLP problem.

The notion of incompactness of a solution of an instance of the OLP problem is also introduced in Kim and Lim (2012). A solution is called incompact if there is a wafer that remains being unassigned to any order although it can be used to satisfy another order to which a wafer with longer (remaining) lead time is already assigned. It is shown by Kim and Lim (2012) that an incompact assignment does to improve the TT value of an assignment. The so-called compact pegging method proposed by Kim and Lim (2012) starts from a given order sequence and assigns in this sequence first lots and then wafers to each order (see also Subsection 3.1 of the present paper). It is shown that there exits an optimal order sequence that can be used to determine an optimal solution of the OLP problem by the compact pegging method.

It can be shown by counter examples that in the multi-fab setting (globally) incompact solutions exist that are optimal. This behavior is a result of the traceability condition. But of course, it is still valid that the incompactness with respect to a single wafer fab does not improve the partial solution for this wafer fab. This is called local incompactness in contrast to global incompactness. Therefore, we can concentrate on determining appropriate assignments of orders to wafer fabs and apply then the compact pegging method individually for the orders that belong to a single wafer fab.

3 ALGORITHMS TO SOLVE THE ORDER-LOT PEGGING PROBLEM

3.1 Reference Heuristics

The fairly simple reference heuristic is based on assigning orders to wafer fabs and then applying the compact pegging method to the order sets for each individual wafer fab. We refer to this heuristic as MF-EDD since it is based on an EDD sorting of the orders. Figures 2 and 3 show the procedure.

1: **Initialize:** Sort the orders with respect to the EDD dispatching rule in non-decreasing order. The resulting list is L. Sort the lots in each wafer fab with respect to the remaining lead time r_{lk} in non-decreasing order. Let S be the set of all already considered orders. Initialize $S \leftarrow \emptyset$.
2: **Repeat** until $|S| = N$
3: Let order i be the first element of L.
4: **Repeat** for all wafer fabs k
5: Determine the available lot (if there is any) with the smallest r_{lk} value which is compatible with order i, i.e. $a_{il}^{(k)} = 1$.
6: **End Repeat**
7: If there is no lot available in any wafer fab, randomly select a wafer fab to which order i is assigned, otherwise assign order i to the wafer fab $k^* \leftarrow \text{argmin}_k r_{lk}$. Let l^* be the corresponding lot. Mark l^* as unavailable for further iterations.
8: **Update:** $S \leftarrow S \cup \{i\}$ and $L \leftarrow L \setminus \{l^*\}$.
9: **End Repeat**
10: **Repeat** for all wafer fabs k
11: Apply the compact pegging method to all orders assigned in Steps 2-9 to wafer fab k.
12: **End Repeat**

Figure 2: Procedure MF-EDD.

For the sake of completeness, we briefly recall the compact pegging method from Kim and Lim (2012). We suppress the wafer fab index since this procedure is only for the orders of a single wafer fab.

1: **Initialize:** $x_{il} = 0$ and $y_{it} = 0$ for $i = 1, \ldots, N$, $l = 1, \ldots, L$, $t = 1, \ldots, T$.
2: **Repeat** for all positions in the order sequence
3: Let i be the current order. Find the lot l with the smallest remaining lead time in the wafer fab which is compatible with order i, i.e. $a_{il} = 1$, and which is unconsumed. If such a lot does not exist, go to Step 4. Otherwise, assign lot l to order i. If $q_i \le w_l$ then update $x_{il} \leftarrow x_{il} + q_i$, $w_l \leftarrow w_l - q_i$, and go to Step 2. Otherwise, set $x_{il} \leftarrow x_{il} + w_l$, $q_i \leftarrow q_i - w_l$, and repeat this step.
4: **Determine** the earliest period when new wafers can be launched into the wafer fab. Let i be this period. If $q_i \le Q_t$ then update $y_{il} \leftarrow y_{il} + q_i$, $Q_t \leftarrow Q_t - q_i$, and go to Step 2. Otherwise, set $y_{il} \leftarrow y_{il} + Q_t$, $q_i \leftarrow q_i - Q_t$, and repeat this step.

Figure 3: Procedure Compact Pegging Method.

Note that the MF-EDD procedure assigns orders to wafer fabs and sequences them for each wafer fab. Based on these sequences, the compact pegging method determines a partial pegging plan for each wafer fab using the lots of the fab. For a single wafer fab, the MF-EDD procedure is equivalent to the EDD rule from Kim and Lim (2012).

3.2 Metaheuristic Approach

The MF-EDD procedure is similar to a list scheduling heuristic in parallel machine scheduling, i.e., its outcome depends on the given order sequence and the rule which selects wafer fab for the next order to be

assigned. It is desirable to have a heuristic which avoids such dependencies and is more flexible with respect to assignment and sequencing decisions.

GAs are often used for hard combinatorial optimization problems. A GA maintains a solution set, a so-called population. GAs work iteratively where an iteration corresponds to a generation. Reproduction and mutation procedures are applied to the individuals of the previous generation to obtain the new generation. Typically, only the fittest individuals are selected. GAs using a random key representation are proposed by Bean (1994). The resulting RKGAs are able to deal with sequencing and assignment decisions. Chromosomes are represented in RKGAs as vectors of randomly generated real numbers. A decoder is used to associate the chromosome with a solution of the corresponding optimization problem. The random-key vectors are typically sorted to determine a sequence.

Starting from a randomly chosen population of random-key vectors, the fitness of the chromosomes of the population is determined by a decoder. The population consists of a small set of elite individuals and the remaining set of non-elite individuals. Individuals that belong to the elite set have large fitness values. The elite individuals are copied unchanged into the next generation. RKGAs use mutation based on immigration, i.e., mutants that are generated in the same manner as individuals of the initial population are placed into the population. The remaining individuals of the population of the next generation are found by crossover.

BRKGAs differ from conventional RKGAs with respect to the way parents are chosen for mating. A chromosome of this set is generated in BRKGAs by combining a randomly chosen element from the elite set with one from the non-elite set that is randomly selected (cf. Gonçalves and Resende 2011), whereas two individuals are randomly chosen from the population in RKGAs. A parameterized uniform crossover is used in BRKGAs. A biased coin is tossed for each gene to determine which parent will contribute to the allele. The probability of choosing the parent from the elite set is $\rho_e > 0.5$. Considering multiple populations that evolve independently and change elite chromosomes from time to time is another method to improve the convergence behavior of the BRKGA (cf. Gonçalves and Resende 2011).

Next, the encoding and decoding scheme will be described. We are interested in assigning N orders to m wafer fabs and determine order sequences for each single wafer fab. A chromosome is therefore coded as a vector

$$RK = \left[rk_1, rk_2, \ldots, rk_N\right] \tag{16}$$

of real numbers, where $rk_i \in (0,1),\ 1 \le i \le N$. Gene rk_i of the chromosome is related to order i.

The decoder is described next. To obtain assignments of an order to a wafer fab, we multiply each random key by m. The integer part $\lfloor m \cdot rk_i \rfloor$ then determines the wafer fab, where we determine the sequence by sorting all random-keys in non-decreasing order. We apply the compact pegging method to the order set that is assigned to an individual wafer fab to compute the TT value of a chromosome. To avoid infeasibilities due to a large number of order assignments to a single wafer fab in chromosomes, an artificial period $T+1$ with infinite capacity and huge $s_{i,T+1}$ values is added to penalize the resulting solutions.

4 COMPUTATIONAL EXPERIMENTS

4.1 Design of Experiments

We expect that the performance of the proposed algorithms depends on the number of orders N and the number of wafer fabs m. We use the instances from Kim and Lim (2012) for $m = 1$. Additional instances for $m = 2$ are generated for each instance with $m = 1$ by assigning each lot with probability 0.5 to one of the two wafer fabs. The design of experiments is summarized in Table 1.

Table 1: Design of experiments.

Factor	Level	Count
# of orders (N)	25, 50, 200, 400	4
# of wafer fabs (m)	1,2	2
Independent replications		20 for $N \le 50$ 10 for $N \ge 200$
Total number of instances		120

Overall, we consider 120 problem instances. The largest ones contain 400 orders and up to around 2500 lots. The instances with $N \le 50$ are called small-sized, the instances with $N = 200$ medium-sized, and the instances with $N = 400$ large-sized.

Two sets of experiments are conducted. In the first experiment, we are interested in benchmarking the two proposed heuristics using near-to-optimal solutions determined by the GUROBI solver for small-sized instances. The performance of the two heuristics is assessed for medium- and large-sized instances by comparing the TT values. Moreover, for $m = 1$ we use the best known TT values determined by the SA approach from Kim and Lim (2012). The BRKGA is repeated three times with different seeds to obtain statistically reasonable results. Average TT values of the three runs are reported. We use the relative performance measure

$$R(A_1, A_2) := TT(A_1)/TT(A_2) \tag{17}$$

for two algorithms A_1 and A_2. The notation $TT(A)$ indicates that algorithm A is used to compute the TT value. For the small-sized instances, we use $R(A, MILP)$ for $MF-EDD, BRKGA \in A$, whereas the $R(BRKGA, A)$ values with $MF-EDD, SA \in A$ are reported for medium- and large-sized problem instances in the second set of experiments.

4.2 Parameter Setting and Implementation Issues

The following parameter settings are used for the BRKGA. The population size is 200 in all experiments, we use 500 generations for the small- and medium-sized instances, while 1000 generations are applied for the large-sized instances. We use $\rho_e = 0.7$ as the probability of choosing a parent from the elite set. The fraction of the population to be replaced by mutants is $p_m = 0.1$. The fraction of the population that belongs to the elite set is 0.2. These values are found by recommendations from the literature (Bean 1994; Toso and Resende 2011) and some preliminary experiments with a small number of instances following a trial and error strategy. The MILP is able to solve only small-sized instances consisting of up to 50 orders and two fabs to optimality. The maximum computing time given is 60 minutes for an instance with 25 orders and 240 minutes for an instance with 50 orders, respectively.

The MILP instances are solved using GUROBI 9.0 on a PC with Intel Xeon CPU E5-2697 v3 with 2.60GHz and 128 GB RAM. The heuristics are coded using the C++ programming language. The brkgaAPI framework (Toso and Resende 2011; Toso and Resende 2015) is used to code the BRKGA. A PC with i7- 4810MQ CPU@2.80 GHz processor and 8GB RAM is used to carry out the performance assessment of the heuristics. The MILP instances are solved.

4.3 Computational Results

We start by showing the results for the first set of experiments in Table 2. The columns labeled by MF-EDD and BRKGA contain the $R(MF-EDD, MILP)$ and $R(BRKGA, MILP)$ values, respectively. The average computing time per instance of the BRKGA is less than 15 seconds for $N = 25$ and around 40 seconds for $N = 50$.

Table 2: Computational results for small-sized problem instances.

#orders/# wafer fabs	MF-EDD	BRKGA	#MILP instances solved to optimality
25/1	1.1167	1.0043	20
25/2	1.8327	1.0063	18
50/1	1.2696	1.0022	14
50/2	2.7952	1.0384	10

We see from Table 2 that the BRKGA is able to determine high-quality solutions. The MILP solver often computes solutions with proven optimality. For $m = 2$ and $N = 50$ the results are slightly worse. The MF-EDD procedure does not perform very well, especially for $m = 2$. Overall, it seems that the BRKGA is correctly implemented.

Next, we continue by presenting computational results for medium- and large-sized problem instances in Table 3. Here, the column labeled by SA refers to $R(BRKGA, SA)$ values. The average computing time per instance of the BRKGA is around seven minutes for $N = 200$ and up to 35 minutes for $N = 400$. The instances for $m = 2$ require sometimes longer computing times than the instances for $m = 1$ since the compact pegging method is more often applied.

Table 3: Computational results for medium and large-sized problem instances.

#orders/# wafer fabs	BRKGA	SA
200/1	0.7231	1.0028
200/2	0.3866	-
400/1	0.6592	1.0445
400/2	0.4226	-

We see from Table 3 that the BRKGA is able to significantly outperform the MF-EDD heuristic. Up to 35% improvements are possible for $m = 1$. For $m = 2$ up to 62% are possible. These improvements are in line with the bad performance of the MF-EDD heuristic for small-sized instances (see Table 2). We see that the BRKGA is able to compete with the SA scheme for medium-sized instances. Although much older hardware is used by Kim and Lim (2012) compared to the present setup which makes a direct comparison problematic, the BRKGA seems to be faster for medium-sized instances. The SA approach slightly outperforms the BRKGA for the large-sized instances. However, the computing time reported by Kim and Lim (2012) is on average almost two hours per instance, whereas the BRKGA requires less than 35 minutes for 1000 generations.

5 CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

In this paper, we discussed a pegging problem in a foundry situation. The studied problem belongs to the demand fulfillment function which is an underresearched area in semiconductor supply chains (Mönch et al. 2018b). An MILP formulation was proposed for the MF-OLP problem. Moreover, a heuristic based on dispatching rules and a BRKGA were proposed to tackle large-sized problem instances. The BRKGA assigns orders to wafer fabs where the order sequence is used to make the pegging decisions for each wafer fab. Based on the computational experiments, it turned out that the BRKGA is able to provide high-quality solutions within a short amount of computing time. It can compete with the SA algorithm from Kim and Lim (2012) for the OLP problem, a special case of the problem studied in this paper, for medium-sized instances.

There are several directions for future research. First of all, it seems possible to design BRKGA-type approaches that propose only order sequences which can be combined with fab assignment rules to assign

orders to wafer fabs. Since the search space of this approach is much smaller compared to the one of the BRKGA proposed in the present paper, we expect some advantage of such an algorithm.

Various generalizations of the MF-OLP problem seem to be possible. For instance, we can consider the situation that not all wafer fabs are qualified for all orders. Moreover, the production costs might be wafer fab-specific. It seems also possible to take into account the different transportation costs from each wafer fab to the customers. A related scheduling problems for parallel machines is studied by Mönch and Shen (2020). It is also interesting to replace the TT measure by the total weighted tardiness measure. It is not clear that the compact pegging method will still work for this more complicated performance measure.

ACKNOWLEDGMENTS

The authors would like to thank Jae-Gon Kim, University of Incheon, for providing his problem instances for the single-fab case. The research was partially supported by the iDev 4.0 project. The iDev40 project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 783163. The JU receives support from the European Union's Horizon 2020 research and innovation programme. It is co-funded by the consortium members, grants from Austria, Germany, Belgium, Italy, Spain and Romania. The first author gratefully acknowledges the provided financial support.

REFERENCES

Bang, J.-Y., K.-Y. An, Y.-D. Kim, and S.-K. Lim. 2005. "A Due-date Based Algorithm for Order-lot Pegging in a Semiconductor Wafer Fabrication Facility". In *Proceedings 3rd International Conference on Modeling and Analysis of Semiconductor .Manufacturing*, 175–180.

Bean, J. C. 1994. "Genetic Algorithms and Random Keys for Sequencing and Optimization". *ORSA Journal of Computing* 6: 154-160.

Boushell, T. G., J. W. Fowler, A. Keha, K. Knutson, and D. C. Montgomery. 2008. "Evaluation of Heuristics for a Class-constrained Lot-to-Order Matching Problem in Semiconductor Manufacturing". *International Journal of Production Research*, 46(12):4143-3166.

Carlyle, W., K. Knutson, and J. W. Fowler. 2001. "Bin Covering Algorithms in the Second Stage of the Lot to Order Matching Problem". *Journal of the Operational Research Society* 52(11): 1232-1243.

Fowler, J., K. Knutson, W. Carlyle. 2000. "Comparison and Evaluation of Lot-To-Order Matching Policies for a Semiconductor Assembly and Test Facility". *International Journal of Production Research* 38(8):1841-1853.

Gonçalves, J. F., and M. G. C. Resende. 2011. "Biased Random-key Genetic Algorithms for Combinatorial Optimization". *Journal of Heuristics*17(5):487–525.

Kim, Y.-D., J.-Y. Bang, K.-Y. An, and S.-K. Lim. 2008. "A Due-Date-Based Algorithm for Lot-Order Assignment in a Semiconductor Wafer Fabrication Facility". *IEEE Transactions on Semiconductor Manufacturing* 21(2):209-216.

Kim, J.-G., S.-K. Lim, S.-O. Shim, and S.-W. Choi. 2010. "Order-lot Pegging Heuristics for Minimizing Total Tardiness in a Semiconductor Wafer Fabrication Facility". In *Proceedings of the 2010 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 1224-1229.

Kim, J.-G., and S.-K. Lim. 2012. "Order-lot Pegging for Minimizing Total Tardiness in Semiconductor Wafer Fabrication Process". *Journal of the Operational Research Society* 63:1258-1270.

Kim, J.-G., S.-K. Lim, and J.-Y. Bang. 2015. "Lot-Order Assignment Applying Priority Rules for the Single-Machine Total Tardiness Scheduling with Nonnegative Time-Dependent Processing Times". *Mathematical Problems in Engineering* Article ID 434653.

Knutson, K., K. Kempf, J. W. Fowler, M. Carlyle. 1999. "Lot-to-Order Matching for a Semiconductor Assembly & Test Facility". *IIE Transactions* 31(11):1103-1111.

Li, Y. T., M. H. Huang, and D. Z. Chen. 2011. "Semiconductor Industry Value Chain: Characters' Technology Evolution". *Industrial Management & Data Systems* 111 (3): 370–390.

Mönch, L., J. W. Fowler, and S. J. Mason. 2013. *Production Planning and Control for Semiconductor Wafer Fabrication Facilities: Modeling, Analysis, and Systems*. New York: Springer.

Mönch, L., and L. Shen. 2020. "Parallel Machine Scheduling with the Total Weighted Delivery Time Performance Measure in Distributed Manufacturing". submitted for publication.

Mönch, L., R. Uzsoy, and J. W. Fowler. 2018a. "A Survey of Semiconductor Supply Chain Models Part I: Semiconductor Supply Chains and Strategic Network Design". *International Journal of Production Research* 56(13):4524-4545.

Mönch, L., R. Uzsoy, and J. W. Fowler. 2018b. "A Survey of Semiconductor Supply Chain Models Part III: Master Planning, Production Planning, and Demand Fulfillment". *International Journal of Production Research* 56(13):4524-4545.

Ng, T., Y. Sun, J. W. Fowler. 2010. "Semiconductor Lot Allocation Using Robust Optimization". *European Journal of Operational Research* 205(3):557-570.

Sun, Y., J. W. Fowler, D. Shunk. 2011. "Policies for Allocating Product Lots to Customer Orders in Semiconductor Manufacturing Supply Chains". *Production Planning & Control* 22(1):69-80.

Toso, R. F., and M. G. C. Resende 2011. A C++ Application Programming Interface for Biased Random-key Genetic Algorithms. http://mauricio.resende.info/doc/brkgaAPI.pdf. accessed 15th April 2020.

Toso, R. F., and M. G. C. Resende. 2015. "A C++ Application Programming Interface for Biased Random-key Genetic Algorithms". *Optimization Methods and Software* 30(1): 81-93.

AUTHOR BIOGRAPHIES

LARS MÖNCH is Professor in the Department of Mathematics and Computer Science at the University of Hagen, Germany. He received a master's degree in applied mathematics and a Ph.D. in the same subject from the University of Göttingen, Germany. His current research interests are in simulation-based production control of semiconductor wafer fabrication facilities, applied optimization and artificial intelligence applications in manufacturing, logistics, and service operations. He is a member of GI (German Chapter of the ACM), GOR (German Operations Research Society), and INFORMS. He is an Associate Editor for the *European Journal of Industrial Engineering*, *Journal of Simulation*, *Business & Information Systems Engineering*, *IEEE Robotics and Automation Letters*, *IEEE Transactions on Semiconductor Manufacturing*, and *IEEE Transactions on Automation Science and Engineering*. His email address is lars.moench@fernuni-hagen.de.

LIJI SHEN is a full professor at the WHU-Otto Beisheim School of Management, Germany. She received her Master's and Ph.D. degree at the Technical University Dresden in Germany. Her research interests include modern metaheuristics and intelligent search methods, mathematical optimization, and machine scheduling. She is a member of GOR (German Operations Research Society), APORS (Asia-Pacific Operations Research Society), and VHB (German Academic Association of Business Research). Her email address is liji.shen@whu.edu.

JOHN W. FOWLER is the Motorola Professor of Supply Chain Management at the Arizona State University. His research interests include discrete event simulation, deterministic scheduling, and multi-criteria decision making. He has published over 130 journal articles and over 100 conference papers. He was the Program Chair for the 2002 and 2008 Industrial Engineering Research Conferences and the 2008 Winter Simulation Conference (WSC). He recently served as Editor-in-Chief for *IIE Transactions on Healthcare Systems Engineering*. He is also an Editor of the *Journal of Simulation* and Associate Editor of *IEEE Transactions on Semiconductor Manufacturing* and *Journal of Scheduling*. He is a Fellow of the Institute of Industrial Engineers (IIE) and recently served as the IIE Vice President for Continuing Education, is a former INFORMS Vice President, and was on the WSC Board of Directors from 2005-2017. His email address is john.fowler@asu.edu.