Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

EXACT AND HEURISTIC ALGORITHMS FOR A BI-CRITERIA ORDER-LOT PEGGING PROBLEM IN A MULTI-FAB SETTING

Andreas Haspecker
Lars Mönch

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

ABSTRACT

We study an order-lot pegging problem 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 in the wafer fabs. The objectives are minimizing the total tardiness of the orders and minimizing the total cost. We are interested in computing the set of Pareto-optimal plans. Based on a mixed-integer linear programming (MILP) formulation, an $\boldsymbol{\epsilon}$ - constraint method is proposed for small-sized problem instances. Moreover, a non-dominated sorting genetic algorithm (NSGA)-II is designed for tackling larger problem instances within a reasonable amount of computing time. We perform computational experiments with the ε-constraint method for small-sized problem instances and with the NSGA-II scheme for small- and medium-sized problem instances.

1 INTRODUCTION

Semiconductor manufacturing belongs to the most complex existing manufacturing processes. Integrated circuits are manufactured on wafers, thin discs made from silicon or gallium arsenide. A single wafer fab contains hundreds of complicated and often extremely complex machines. The routes of the individual products may contain up to 800 process steps (operations) for advanced technologies. Lots are groups of wafers that travel together through a wafer fab. The time span between launching material and its emergence as final product in a wafer fab is up to 12 weeks (Mönch et al. 2013). Semiconductor manufacturing provides an extreme environment for production planning and control due to the sheer size of the fabrication facilities involved, the pervasive presence of different kinds of uncertainties, and the rapid pace of change (Chien et al. 2011).

In this paper, we study a short-term planning problem for semiconductor foundries belonging to the demand fulfillment function in semiconductor supply chains (Mönch et al. 2018a, Mönch et al. 2018b). Foundries manufacture ICs for a wide range of customers in varying quantities on a common manufacturing process. The make-to-order (MTO) strategy is typical for the operations of foundries.

The planning problem studied in the present paper extends the one from Mönch et al. (2020) towards a bi-criteria setting where minimizing variable and fixed costs is the second objective in addition to minimizing total tardiness (TT) of the orders. The problem deals with assigning wafer fabrication lots to given customer orders. The amount of wafers to be released into the wafer fabs is determined when the already existing lots in the wafer fabs are not enough. We refer to this assignment activity as order-lot pegging. A single-objective (TT), 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 for multiple wafer fabs and the TT performance measure towards a bi-criteria setting, abbreviated by BC-MF-OLP problem. We propose an

979-8-3503-6966-3/23/$31.00 ©2023 IEEE

exact $\boldsymbol{\epsilon}$-constraint method for small-sized problem instances and an NSGA-II scheme to tackle medium- and large-sized problem instances.

The paper is organized as follows. In the next section, we describe and analyze the problem at hand and discuss related work. An exact approach based on the ϵ-constraint method is established in Section 3. An NSGA-II scheme is proposed in Section 4. The results of computational experiments are presented and discussed in Section 5. Conclusions and future research directions are indicated in Section 6.

2 PROBLEM SETTING AND DISCUSSION OF RELATED WORK

2.1 Problem Description

We consider m wafer fabs with identical capabilities running in parallel. 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 periods of the same size. Order i has q_i wafers and a due date d_i. The quantities $L_k, k = 1, \dots, m$ are the number of 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. In addition, the number of wafers belonging to lot l in wafer fab k is w_{lk}. We have a total of L lots across all wafer fabs. There are compatibility matrices $A^{(k)} \in \mathbb{R}^{N \times L_k}$ whose entries $a_{il}^{(k)}$ are 1 if lot l of wafer fab k can be used to fulfill order i and zero otherwise. Variable costs v_{ik} per wafer and period are assumed for each order i in wafer fab k. In this research, we assume that the variable costs for all orders that are fulfilled by lots and newly released wafers of the same wafer fab are identical. Moreover, we consider fixed production costs p_{ik} for each order i in wafer fab k. The sum of fixed and variable costs of all orders is called total cost (TC).

Due to traceability reasons, orders can only be fulfilled from lots and wafers of the same wafer fab. This means that if a first lot from a wafer fab is pegged to a certain order, all the remaining required lots and wafers have to be from this wafer fab. 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 k is s_{ik}.

The tardiness of order i is $T_i := \max(C_i - d_i, 0)$, where C_i is the completion time of order i. We are interested in minimizing the TT value 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. Moreover, we are interested in minimizing the TC. The two objectives are in conflict. Small TT values require assigning the orders to wafer fabs where appropriate lots with a low remaining lead exist, whereas small TC values can be obtained when appropriate wafer fabs are chosen. Hence, we are interested in determining the set of all Pareto-optimal plans. A plan P is called Pareto-optimal if there does not exist another plan P' such that $TT(P') \le TT(P)$ and $TC(P') \le TC(P)$ and at least one of the two inequalities is strict.

2.2 Analysis

The notion of an order split is recalled from Kim and Lim (2012) for solutions of OLP problem instances for a single wafer fab. An order i is split by order j if two lots or new wafer releases, we refer to this as (wafer) resource, are used to satisfy i with (remaining) lead times $\delta_1 < \delta_2$, but there is another resource that is used to fulfill order j with (remaining) lead time of δ_3, $\delta_1 < \delta_3 < \delta_2$. In addition, the following second split situation is also possible. Orders i and j are both satisfied by two resources with (remaining) lead times $\delta_1 < \delta_2$. It is easy to see that in both situations an order split does not improve the TT value for the OLP problem.

For a single wafer fab, the variable costs of the orders depend on the remaining lead times of the lots and new wafer releases. Next, we will show that split orders are also not beneficial for the BC-MF-OLP problem.

Proposition 1 An order split does not improve the TC value of OLP problem instances in a single wafer fab setting.
Proof: We consider a solution S in which order i is split by order j. Another solution S' can be constructed on the one hand by reassigning wafers from the resource with remaining lead time δ_3 originally assigned to order j in S to order i. On the other hand, wafers from resource with remaining lead time of δ_1 originally assigned to order i in S are assigned to order j in S'. Due to $\delta_1 < \delta_3 < \delta_2$ the variable cost of the orders i, j in S'will be unchanged. The second split situation can be addressed by a similar exchange argument. ■

The incompactness notion of a solution of an instance of the OLP problem is also introduced by Kim and Lim (2012). A solution of an OLP instance is 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 not improve the TT value. Next, a similar statement will be shown for the TC value in a single-fab setting.

Proposition 2 An incompact assignment does not improve the TC value of OLP problem instances in a single wafer fab setting.
Proof: We consider an incompact solution S in which wafers in a resource k_1, i.e. a lot or newly launched wafers, exist that are not assigned to any order. According to the incompactness definition, there is an order i and a resource k_2 such that wafers of k_2 are used to satisfy order i. A solution S' can be obtained by assigning wafers in k_1 not assigned to any order to order i instead of the wafers of k_2 assigned to i in S. Since we have $\delta_{k_1} < \delta_{k_2}$ due to the definition of an incompact solution and the fact that the variable costs only depend on the remaining lead time, we obtain $TC(S') < TC(S)$. ■

The so-called compact pegging method proposed by Kim and Lim (2012) starts from a given order sequence and assigns in this sequence first compatible lots with the smallest remaining lead time and then wafers with the smallest lead time to each order. It is shown by Kim and Lim (2012) 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. The compact pegging method computes solutions that are compact and at the same time they do not contain split orders. Since the variable costs are identical for all orders per wafer fab, the assignment of orders to wafer fabs fully determines the TC. For a given order assignment to wafer fabs, a solution with the smallest TT value is a candidate for a Pareto-optimal solution. Such a solution can be determined by the compact pegging method. The nondominated solutions of the set of Pareto-optimal candidates provide the set of Pareto-optimal solutions. Note that for order- and wafer fab-dependent variable cost the compact pegging method applied to the orders of each wafer fab in general will not work anymore.

It is shown by Kim and Lim (2012) that even the single-fab OLP problem with TT objective function is NP-hard. Based on the fact that the generalized assignment problem is NP-hard (cf. Kellerer et al. 2004), it can be shown that the OLP problem with at least two wafer fabs and with TC performance measure is NP-hard. Hence, determining the set of Pareto-optimal plans for the BC-MF-OLP problem is NP-hard too (Ehrgott 2010). Hence, we have to look for efficient heuristics for large-sized problem instances in the present paper.

2.3 Discussion of Related Work

The problem of assigning lots to customer orders in semiconductor manufacturing is rarely discussed in the literature. There are two streams of research in this area. The first one deals with the make-to-stock (MTS) production strategy. The second one is related to the MTO production strategy. We start by the first stream. Knutson et al. (1999) study the problem of assigning lots of different sizes to customer orders in an assembly facility. Maximizing the number of integrated circuits that are sent to customers and the number of orders delivered on time and minimizing the excess inventory are the goals. Bin-packing inspired heuristics are designed. Fowler et al. (2000) and Carlyle et al. (2001) investigate similar problems and propose more heuristics. Boushell et al. (2008) consider a lot-to-order matching problem for multiple product classes resulting from binning. A somewhat related problem that considers under- or overfilling of customer orders when lot sizes are uncertain is tackled by Ng et al. (2010) using robust optimization. Sun et al. (2011) generalized this problem by allowing downward product substitution when demand exceeds supply. All the studied lot-to-order matching problems often do not consider due dates. Moreover, multiple facilities are not taken into account. Next, we look at the second stream. Bang et al. (2005) and Kim et al. (2008) study hard and soft pegging strategies. Hard pegging refers to the situation that a lot is assigned to a single customer order and cannot be reassigned. Soft pegging allows for event-driven repegging in the face of uncertainty. It is shown by using discrete simulation that soft pegging strategies are able to significantly improve hard pegging ones under several experimental conditions. However, only a single wafer fab and a single performance measure are considered in these papers.

The problem in the present paper extends the OLP problem of Kim et al. (2010). An MILP is established there. Several simple and fast heuristics based on the Earliest Due Date dispatching rule are proposed. Additional dispatching rules for this problem are studied by Kim et al. (2015). More efficient solution approaches for this problem are proposed by Kim and Lim (2012). Mönch et al. (2020) consider a multi-fab version of the OLP problem of Kim et al. (2010). They propose a biased random key GA. This paper is the most pertinent related work. But only a single criterion, namely minimizing TT, is considered. However, in real-world settings additional measures are of interest, especially when wafer fabs with different cost structure are considered. This situation is assumed in the present paper.

3 EXACT APPROACH

3.1 MILP

The MILP formulation for minimizing the TC and the TT values is presented next. It extends the formulation provided by Mönch et al. (2020) to the bi-criteria setting. 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 formulation is based on the following parameters:

N : number of orders

L_k : number of lots in wafer fab k

m : number of wafer fabs

p_{ik} : fixed production cost for order i in wafer fab k

v_{ik} : variable cost for order i in wafer fab k

T : length of the planning horizon

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 wafers 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 fab k 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

C_i : completion time of order i

T_i : tardiness of order i

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 fab k are used to satisfy order i, 0, otherwise.

The model itself can be formulated as follows:

$$\min\left(\sum_{i=1}^{N} T_i \, , \sum_{k=1}^{m}\sum_{i=1}^{N}\left(p_{ik}f_{ik} + v_{ik}\left\{ \sum_{l=1}^{L_k} r_{lk}x_{ilk} + \sum_{t=1}^{T}(t+s_{ik})y_{itk} \right\}\right)\right) \qquad (1)$$

subject to

$$\sum_{k=1}^{m} f_{ik} = 1 \qquad i=1,\ldots,N \qquad (2)$$

$$\sum_{k=1}^{m}\left(\sum_{l=1}^{L_k} x_{ilk} + \sum_{t=1}^{T} y_{itk} \right) = q_i \qquad i=1,\ldots,N \qquad (3)$$

$$\sum_{i=1}^{N} x_{ilk} \le w_{lk} \qquad k=1,\ldots,m,\; l=1,\ldots,L_k \qquad (4)$$

$$\sum_{i=1}^{N} y_{itk} \le Q_{tk} \qquad t=1,\ldots,T,\; k=1,\ldots,m \qquad (5)$$

$$x_{ilk} \le w_{lk} z_{ilk} \qquad k=1,\ldots,m, l=1,\ldots,L_k,\; i=1,\ldots,N \qquad (6)$$

$$y_{itk} \le Q_{tk} u_{itk} \qquad k=1,\ldots,m,\; i=1,\ldots,N,\; t=1,\ldots,T \qquad (7)$$

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

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

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

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

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

$$C_i - d_i \le T_i \qquad i=1,\ldots,N \qquad (13)$$

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

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

The first component of the objective function (1) is the TT value of all orders, the second one is the corresponding TC value. Constraint set (2) models that each order is assigned to exactly one wafer fab whereas constraint set (3) expresses that an order is satisfied with wafers from already launched lots and eventually with newly released wafers. Constraint set (4) indicates that the number of wafers belonging to each lot is respected during the pegging process. Constraints (5) ensure that the total number of wafers to be released into a wafer fab does not exceed the maximum number of wafers that can be released in a given period. Constraint set (6) 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 fact is expressed for newly released wafers in a period by constraint set (7). The constraints (2), (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 lot compatibility to orders is modeled by (10). The tardiness of individual orders is calculated by the constraint sets (11), (12), and (13). The range of the decision variables is modeled by the constraint sets (14) and (15).

3.2 ε - Constraint Method

The ε - constraint method is an approach that allows to compute a Pareto frontier (Ehrgott 2010). Instead of combining the different objectives into an integrated objective function, a single objective function is optimized in a certain step while the remaining objectives are transformed into constraints. An ε-constraint problem with two objectives $f_k, k = 1,2$ to be minimized is provided by:

$$\min f_k$$

subject to

$$f_j \le \varepsilon_j,\ j = 1,2,\ j \ne k,$$

where $\varepsilon \in \mathbb{R}^2$ is given. The application of the ε-constraint method to the MILP formulation (1)-(15) is shown next:

$$\min\left(E_{TT}\sum_{i=1}^{N} T_i + E_{TC}\sum_{k=1}^{m}\sum_{i=1}^{N}\left(p_{ik}f_{ik} + v_{ik}\left\{\sum_{l=1}^{L_k} r_{lk}x_{ilk} + \sum_{t=1}^{T}(t+s_{ik})y_{itk}\right\}\right)\right) \tag{16}$$

subject to

(2)-(15)

$$\left(1 - E_{TT}\right)\sum_{i=1}^{N} T_i \le \varepsilon_{TT}. \tag{17}$$

$$(1 - E_{TC})\sum_{k=1}^{m}\sum_{i=1}^{N}\left(p_{ik}f_{ik} + v_{ik}\left\{\sum_{l=1}^{L_k} r_{lk}x_{ilk} + \sum_{t=1}^{T}(t+s_{ik})y_{itk}\right\}\right) \le \varepsilon_{TC}. \tag{18}$$

The MILP has the parameters $E_{TT}, E_{TC} \in \{0,1\}$, $E_{TT} + E_{TC} = 1$, and $\varepsilon_{TT}, \varepsilon_{TC} \in \mathbb{R}$. For $E_{TT} = 1$ and $E_{TC} = 0$ the model pursues a TT minimization whereas the TC value is restricted to ε_{TC}. In the case of $E_{TT} = 0$ and $E_{TC} = 1$, the model strives for a minimization of the TC value. In this situation, the TT value of the plan is restricted to ε_{TT}.

The model (16)-(18) and (2)-(15) is iteratively solved to obtain the set of Pareto-optimal plans. It is assumed that all parameter values are integer. The first iteration starts with $E_{TT} = 1, E_{TC} = 0, \varepsilon_{TT} = 0$, and $\varepsilon_{TC} = M$ where M is a sufficient large number. The solution is a plan P with objective function value $TT(P)$ where the TC value is restricted to M. Afterwards, the MILP is solved a second time with

$E_{TT}=0, E_{TC}=1, \varepsilon_{TC}=0, \varepsilon_{TT}=TT(P)$ leading to a Pareto optimal schedule. The next iteration starts by $E_{TT}=0, E_{TC}=1, \varepsilon_{TC}=0$, and $\varepsilon_{TT}=TT(P)-1$. The TT and TC values are always integers due to the choice of the parameter values of the instances. Therefore, choosing $\varepsilon_{TT}=TT(P)-1$ is reasonable. This procedure is repeated until the MILP becomes infeasible for the parameters ε_{TC} and ε_{TT}. The set of all Pareto-optimal plans is obtained for instances with parameter values selected as described before. Note that the ε-constraint method will work only for small-sized instances using a reasonable amount of computing time since the BC-MF-OLP problem is an NP-hard combinatorial optimization problem.

4 METAHEURISTIC APPROACH

4.1 Overall Approach

A GA maintains a set of solutions where we refer to this set as population. GAs are iterative algorithms. A single iteration corresponds to a generation. Reproduction and mutation procedures are used to change the individuals of the current generation to form a new generation. It is likely that only the fittest individuals are selected for the new generation. In the present paper, we apply a NSGA-II procedure to tackle larger problem instances of the BC-MF-OLP problem. The NSGA-II approach proposed by Deb et al. (2002) is a GA tailored towards multi-criteria combinatorial optimization problems (Coello Coello and Lamont 2004). It ensures diversity of the population by exploiting information of solutions from the entire population. To do so, the set of solutions belonging to a population is sorted into distinct fronts of different domination levels within each GA iteration. The first front contains all solutions not dominated by any other solution. The second front contains those which are only dominated by solutions from the first front. This principle is repeated. The fitness value of an individual is determined by the front its solution belongs to. For solutions of the same front a crowded comparison operator is applied that assigns a higher fitness value to solutions in less crowded regions of the solution space. Two individuals of a population are randomly chosen and the one with higher fitness is chosen for crossover purposes (Michalewicz 1996), i.e., binary tournament selection is applied. Offspring are generated by recombination until the population size is doubled. An elitist strategy is used within the NSGA-II scheme. Individuals are inserted into the new population by non-increasing fitness values, i.e., solutions are accepted starting from the first front until the original population size is reached. Solutions from the least crowded regions of the solution space are preferred from the last front to be accepted. It is well-known (Deb and Goel 2001) that the performance of NSGA-II approaches can be improved by including local search to move the obtained Pareto frontier closer to the true Pareto front. In the next subsection, we will describe the encoding and decoding procedures for the NSGA-II scheme.

4.2 Tailored NSGA-II Scheme

GAs based on the random key representation are proposed by Bean (1994). These random-key GAs (RKGAs) are appropriate to support sequencing and assignment decisions. Chromosomes are represented as vectors of randomly generated real numbers from $(0,1)$. Problem-specific decoders must be designed to associate a chromosome with a solution of the optimization problem at hand. Sorting the random keys is required to compute a sequence. Starting from a randomly chosen population of random-key vectors, the fitness of the chromosomes is computed by a decoder that implements the objective function of the optimization problem. The population consists of a small set of elite individuals and a set of nonelite individuals. Individuals that belong to the elite set have small objective function values in the case of a minimization problem. The elite individuals are taken over unchanged into the next generation. A certain fraction of randomly generated mutants are placed into the population. The remaining individuals of the population of the next generation are found by crossover. Two individuals are randomly chosen from the population in a RKGA for crossover purposes. A parameterized uniform crossover is applied. 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 first chromosome is $\rho \geq 0.5$. The random-key representation is appropriate for the BC-MF-OLP problem since after an assignment of the orders to wafer fabs starting from an order sequence, it is enough to apply the compact pegging method from Kim and Lim (2012) as shown in Subsection 2.2.

The following encoding and decoding is therefore appropriate for the problem at hand. We have to assign N orders to m wafer fabs and must determine sequences of the orders for each single wafer fab to apply the compact pegging method which requires a sorted list of orders. A chromosome is given by the following random-key vector $RK = [rk_1, \ldots, rk_N]$ of real numbers, where $rk_i \in (0,1)$, $1 \leq i \leq N$, and gene rk_i represents order i. The decoder works as follows. Each random key is multiplied by the integer m to assign each order to a wafer fab. The integer part determines the wafer fab. The sequence of the orders assigned to wafer fab $1 \leq k \leq m$ is obtained by sorting the quantities $m \cdot rk_i - \lfloor m \cdot rk_i \rfloor$ in non-decreasing order. The compact pegging method is applied to the order sequence of each wafer fab to compute the TT and TC values 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. Next, we describe the applied local search procedure within the NSGA-II approach. An integrated objective function is used for a given plan P as follows:

$$F(P) := \left(w_1^S TT(P) + w_2^S TC(P)\right) / \left(w_1^S + w_2^S\right), \tag{19}$$

with weights $w_1^S := \dfrac{TT^{\max} - TT(P)}{TT^{\max} - TT^{\min}}$ and $w_2^S := \dfrac{TC^{\max} - TC(P)}{TC^{\max} - TC^{\min}}$. $f^{\max}$ and $f^{\min}$ are the maximum and minimum value of an objective function values f for the entire population. The local search (LS) procedure for $m = 2$ wafer fabs is based on the following two neighborhood structures:

LS procedure:

1. **Swap:** Randomly choose two different orders and exchange their position in the order sequence, i.e. exchange the corresponding random keys.
2. **Shift:** Randomly choose an order i. If $0 < rk_i \leq 1/2$ then use the random key $2(rk_i + 1/2)$, otherwise, use $2(rk_i - 1/2)$.

Swap and Shift moves are executed in a consecutive manner. Each move is evaluated with respect to the integrated objective function F (19) instead of using the TT and the TC measures separately. The LS procedure terminates after five moves do not lead to an improved value of the integrated objective function. The resulting NSGA-II scheme equipped with the LS is abbreviated by LS-NSGA-II in the rest of the paper.

5 COMPUTATIONAL EXPERIMENTS

5.1 Design of Experiments

We expect that the performance of the ϵ-constraint method and of the NSGA-II scheme depends on the number of orders N and the number of wafer fabs m. We reuse the instances provided by Kim and Lim (2012) for the single-fab case. Similar to Mönch et al. (2020), instances for $m = 2$ are generated for each of the single-fab instances by assigning each lot with equal probability to one of the two wafer fabs. Variable costs are set as $v_{ik} \equiv k$ for $i = 1, \ldots, N, k \in \{1,2\}$ whereas the fixed costs are taken from a discrete uniform distribution $p_{ik} \sim DU[100, 200, 300, 400, 500]$. The design of experiments is summarized in Table 1. Overall, we consider 20 problem instances in preliminary computational experiments.

In the first experiment, we are interested in assessing the computational tractability of the ϵ-constraint method. Here, we measure the computing time per problem instance. In a second experiment, we are interes-

Table 1: Design of experiments.

Factor	Level	Count
# of orders (N)	25, 50	2
# of wafer fabs (m)	2	1
Independent replications		10 for $N = 25$ 10 for $N = 50$
Total number of problem instances		20

ted in the performance of the NSGA-II approach, especially when the LS scheme is used. The proposed NSGA-II variants determine an approximation of the Pareto frontier. Instead of comparing the algorithms based on objective function values, quality measures are computed from the approximation sets (Van Veldhuizen 1999, Zitzler et al. 2003). Let Y_H be the set of solutions obtained by the heuristic H and Y_{true} be the related set of true Pareto-optimal solutions. The overall non-dominated vector generation (ONVG) measure is given by

$$ONVG(Y_H) := |Y_H|,$$

i.e., it is the number of non-dominated solutions obtained by H. In addition, we measure the number of solutions that Y_H and Y_{true} have in common. This measure is called the overall true nondominated vector generation (OTNVG)), given by

$$OTNVG := |\{y | y \in Y_H \cap Y_{true}\}|.$$

The overall non-dominated vector generation ratio (ONVGR) measure is obtained by:

$$ONVGR(Y_H, Y_{true}) := ONVG(Y_H)/ONVG(Y_{true}).$$

Moreover, the ratio of solutions in Y_H that are not in Y_{true}, is defined by

$$Error := \left(ONVG\left(Y_H\right) - OTNVG\right)/ONVG\left(Y_H\right).$$

A distance measure proposed by Jaszkiewicz (2004) is applied that computes the mean distance of solutions provided by H to the nearest solution of the true Pareto front. Therefore, we introduce the quantity:

$$d(y,\hat{y}) := \left((TT(y) - TT(\hat{y}))^2 / \left(TT^{\max} - TT^{\min}\right)^2 + (TC(y) - TC(\hat{y}))^2 / \left(TC^{\max} - TC^{\min}\right)^2\right)^{1/2}$$

for $y \in Y_H$ and $\hat{y} \in Y_{true}$ where f_k^{max} and f_k^{min} are the maximum and minimum of the k-th objective function component found among the solutions from Y_H and Y_{true}. The average distance of a solution from Y_H to the closest solution in Y_{true} is obtained by:

$$dist(Y_H, Y_{true}) := \sqrt{\sum_{y \in Y_H} \left(\min_{\hat{y} \in Y_{true}} d(y,\hat{y})\right)^2} \Big/ ONVG(Y_H).$$

All the computational experiments are carried out on a Intel® Core™ i7-7700 CPU 3.60GHz PC with 16GB RAM. The pure NSGA-II and the LS-NSGA-II are performed using a computing time of 30 and 60 minutes per instance for $N = 25$ and $N = 50$, respectively.

5.2 Parameter Setting and Implementation Issues

Some preliminary computational experimentation is carried out to set appropriate parameter values. A population size of 300 is used in the NSGA-II-type approaches. Moreover, 20% of the population are replaced by randomly chosen new elements as suggested by Bean (1994) for RKGAs. We use $\rho = 0.7$ for the uniform crossover. The NSGA-II approach is coded using the MOMHLIB++ (Jaszkiewicz 2023), a framework for multi-criteria evolutionary optimization written in the C++ programming language. The ϵ-constraint method is coded using again the C++ programming language and IBM CPLEX 12.7.1.

5.3 Results

The computing time per instance for the ϵ-constraint method ranges from 1 until 78 minutes for $N = 25$. Its average value is 7 minutes. Due to the high computational burden, only experiments for $N = 25$ are conducted. Even a computing time of several hours per instance is not sufficient to solve the instances for $N = 50$. Next, computational results for small-sized instances of the BC-MF-OLP problem for the NSGA-II variants are presented. This allows us to compare the heuristics with the true Pareto frontier Y_{true} obtained by the ε-constraint method. All heuristics are performed five times with different seeds for each instance since the NSGA-II variants contain stochastic elements. The set of solutions is then formed by these five replications where dominated solutions are removed. The results are shown in Table 2 where best results among comparable results are marked bold. Instead of reporting individually the performance measure values for all instances, we show the average values across the ten instances and the best and worst ONVGR and Error values.

Table 2: Computational results for $\boldsymbol{N = 25}$.

Instance	**ONVG**	**OTNVG**	**ONVGR**	**Error**	**dist**
NSGA-II					
best	-	-	0.850	0.567	0.0071
worst	-	-	0.374	0.887	0.0543
average	9.58	2.18	0.635	0.730	0.0267
LS-NSGA-II					
best	-	-	**1.000**	**0.020**	**0.0001**
worst	-	-	**0.730**	**0.349**	**0.0045**
average	**14.14**	**12.54**	**0.916**	**0.110**	**0.0023**

We observe from Table 2 that the LS-NSGA-II outperforms the pure NSGA-II for all performance measures. However, even the LS-NSGA-II is not able to find the entire true Pareto front. This behavior can be also observed in Figure 1 where the true Pareto front and the one obtained by the LS-NSGA-II are visualized. Next, we show the computational results for the instances with for $N = 50$ in Table 3. The true Pareto front is formed by all known solutions for this problem instance where dominated solutions have been removed. We observe again from Table 3 that the LS-NSGA-II outperforms the pure NSGA-II for all performance measures. Again even the LS-NSGA-II is not able to determine the entire true Pareto front. However, the values of the distance measure are fairly small.

6 CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

In this paper, we introduced and discussed a bi-criteria order-lot pegging problem arising in semiconductor supply chains. The $\varepsilon -$ constraint method was used for small-sized instances to compute the Pareto frontier where we assume that all problem data is integer. Moreover, a NSGA-II approach hybridized with local search based on the random-key representation proposed by Mönch et al. (2020) for the single-criterion version of the problem with the TT measure was designed. The NSGA-II approach only works for the special case that the variable cost only depends on the wafer fab but not on the order.

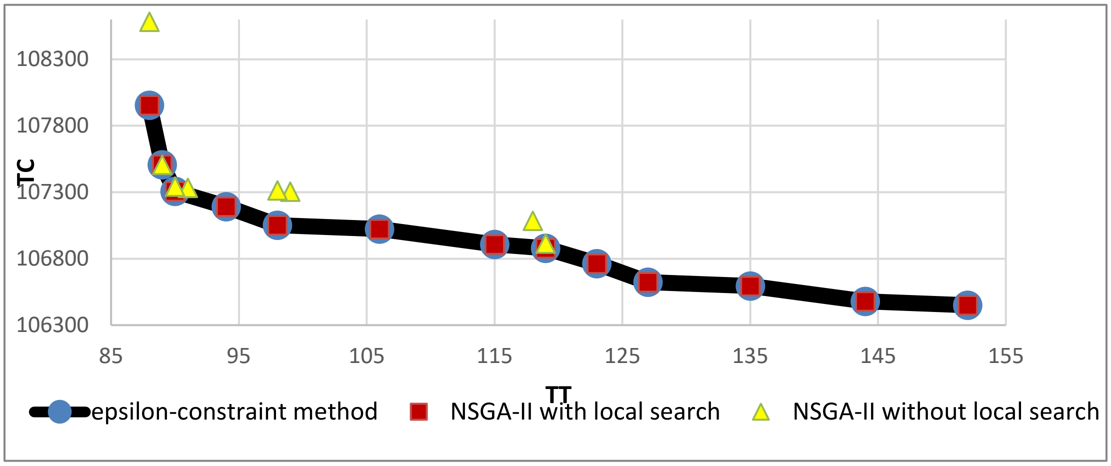

Figure 1: Pareto frontiers for a single problem instance.

Table 3: Computational results for $\boldsymbol{N = 50}$.

Instance	**ONVG**	**OTNVG**	**ONVGR**	**Error**	**dist**
NSGA-II					
best	-	-	**1.000**	0.983	0.0304
worst	-	-	0.600	1.000	0.1561
average	**12**	0.02	**0.824**	0.998	0.0845
LS-NSGA-II					
best	-	-	0.930	**0.483**	**0.0097**
worst	-	-	**0.615**	**0.779**	**0.0283**
average	10.72	**3.14**	0.716	**0.660**	**0.0173**

There are several directions for future research. First of all, we are interested in improving the proposed NSGA-II approach by designing a more appropriate local search procedure. More computational experiments with larger problem instances must be carried out to get a full picture of the performance of the proposed NSGA-II approach. Moreover, it is planned to extend the NSGA-II approach to the situation that the variable cost depends on the order and the wafer fab. We know from preliminary computational experiments with the $\boldsymbol{\epsilon}$-constraint method that solutions are possible that contain split orders, i.e., the compact pegging method and the representation method proposed in this paper will not work anymore.

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*, October 6th - 7th, Singapore, Singapore, 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.

Chien, C.-F., S. Dauzère-Pérès, H. Ehm J. W. Fowler, Z. Jiang, S. Krishnaswamy, L. Mönch, and R. Uzsoy. 2011**.** “Modeling and Analysis of Semiconductor Manufacturing in a Shrinking World: Challenges and Successes”. *European Journal of Industrial Engineering* 5(3): 254-271.

Coello Coello, C. A., and G. B. Lamont. 2004. “An Introduction to Multi-objective Evolutionary Algorithms and Their Applications”. In *Applications of Multi-Objective Evolutionary Algorithms*. edited by C. A. Coello Coello and G. B. Lamont, 1-28. Singapore: World Scientific.

Deb, K., and T. Goel. 2001. “A Hybrid Multi-Objective Evolutionary Approach to Engineering Shape Design”. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, LNCS 1993, March 7th - 9th, Zurich, Switzerland, 385-399.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. 2002. “A Fast and Elitist Genetic Algorithm: NSGA-II”. *IEEE Transactions on Evolutionary Computation* 6(2):182-197.

Ehrgott, M. 2010. *Multicriteria Optimization*. 2nd ed., New York: Springer.

Fowler, J., K. Knutson, and 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.

Jaszkiewicz, A. 2004. “Evaluation of Multiple Objective Metaheuristics”. In *Proceedings Metaheuristics for Multiobjective Optimization*. Lecture Notes in Economics and Mathematical Systems, 535, 65–89.

Jaszkiewicz, A. 2023. MOMHLIB++: Multiple Objective Metaheuristics Library in C++. https://github.com/derino/-maponoc/tree/master/libs/libmomh-1.91.3. Accessed 30th April 2023.

Kellerer, H., U. Pferschy, and D. Pisinger. 2004. *Knapsack Problems*. 1st ed., Berlin: Springer.

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)*, December 7th - 10th, Macao, China, 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*, Volume 2015, Article ID 434653, 1 - 11.

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

Michalewicz, Z. 1996. *Genetic Algorithms + Data Structures = Evolution Programs*. 3rd ed., Berlin: Springer.

Mönch, L., J. W. Fowler, and S. J. Mason. 2013. *Production Planning and Control for Semiconductor Wafer Fabrication Facilities: Modeling, Analysis, and Systems*. 1st ed., New York: Springer.

Mönch, L., L. Shen, and J. W. Fowler. 2020. “Heuristics for Order-lot Pegging in Multi-fab Settings”. In *Proceedings of the 2020 Winter Simulation Conference*, edited by K.-H. G. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, 1742-1752. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

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, and J. W. Fowler. 2010. “Semiconductor Lot Allocation Using Robust Optimization”. *European Journal of Operational Research* 205(3):557-570.

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

Van Veldhuizen, D. A. 1999. “Multiobjective Evolutionary Algorithms: Classifications, Analysis, and New Innovations”. Technical Report, Air Force Institute of Technology, Department of Electrical and Computer Engineering, Dayton, OH.

Zitzler, E., L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. 2003. “Performance Assessment of Multiobjective Optimizers: an Analysis and Review”. *IEEE Transactions on Evolutionary Computation* 7(1):117–132.

AUTHOR BIOGRAPHIES

ANDREAS HASPECKER is a teaching and research assistant and a master student at the Chair of Enterprise-wide Software Systems, University of Hagen. He received a bachelor degree in Information Systems from the University of Hagen, Germany. His research interests include applied optimization and metaheuristics for semiconductor manufacturing. He can be reached by email at andreas.haspecker@fernuni-hagen.de.

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. His email address is lars.moench@fernuni-hagen.de.