Computers & Operations Research 144 (2022) 105813

Contents lists available at ScienceDirect

Computers and Operations Research

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

Optimizing multiple qualifications of products on non-identical parallel machines

Antoine Perraudat [a,b,*], Stéphane Dauzère-Pérès [a,c], Philippe Vialletelle [b]

[a] Mines Saint-Etienne, Univ Clermont Auvergne, CNRS, UMR 6158 LIMOS, CMP, Department of Manufacturing Sciences and Logistics, 880 Route de Mimet, F-13120 Gardanne, France
[b] STMicroelectronics Crolles, 850 Rue Jean Monnet, F-38920 Crolles, France
[c] Department of Accounting and Operations Management, BI Norwegian Business School, 0484 Oslo, Norway

ARTICLE INFO

Keywords:
Flexible manufacturing systems
Qualification management
Non-identical parallel machines
Optimization
Heuristics

ABSTRACT

In some manufacturing contexts, such as semiconductor manufacturing, machines must be qualified, or eligible, to process a product, and machines cannot be qualified for all products. This paper investigates the problem of optimizing a given number of new qualifications of products to machines to maximize a flexibility measure that evaluates the balance of the qualification configuration of a work center in terms of utilization rate of machines on a set of non-identical parallel machines. Motivated by empirical observations, new solution approaches, notably inspired by heuristics for discrete location problems and based on the analysis of dual variables, are proposed and compared on industrial data from a semiconductor manufacturing facility and on randomly instances. The use of dual variables leads to heuristics that are effective both in terms of solution quality and computational time. The best proposed approach is currently used in the decision support system of a semiconductor manufacturing facility.

1. Introduction

In some manufacturing contexts with a large variety of products, machines performing the same type of operations are not always qualified (also called eligible in the literature) to process all products. This is in particular the case in semiconductor manufacturing, where operations on many different products (wafers in front-end manufacturing facilities and integrated circuits in back-end manufacturing facilities) at different stages of their manufacturing process need to be performed in different work centers. Each work center may include more than 100 machines that are usually non identical, *i.e.* the process time on a machine differs from one product to another. This is because, to apply a manufacturing process on a given product, a machine must follow a *recipe* that defines for instance the pressure, the temperature conditions and the chemicals that must be used. Recipes can be very different from one product to another, leading to very different process times. In manufacturing facilities with a large variety of products, the recipe count can be of several hundreds in a single work center. In the remainder of the paper, we use the term *product* and not *recipe*.

Moreover, before being allowed to run a product, a machine must undergo a *product-to-machine qualification* procedure. Hence, machines cannot process all products, *i.e.* a machine is only qualified for a limited number of products. For new machines or products, qualification procedures are expensive, time-consuming, can be energy-consuming and may sometimes take up to several months. Once the qualification procedure is completed, the machine is *qualified* for the product. To maximize manufacturing performances, in particular in terms of throughput and cycle time, an efficient design and *follow-up* of the qualification configuration of each work center is required (see *e.g.*, Johnzén et al. (2007, 2011); Kabak et al. (2013); Rowshannahad et al. (2015); Chang and Dong (2017) and Kopp et al. (2018)).

A machine qualified for a product does not remain qualified throughout its operation in the factory. Qualifications are dynamic, *i.e.*, time-varying. A machine no longer qualified for a product is said to be *disqualified* for the product. A *disqualification* can occur for different reasons, in particular because of unexpected events, *e.g.* a consumable becomes empty, or following a (scheduled or unscheduled) maintenance operation. A product can also be disqualified on a machine because the product has not been processed for a long time (qualification time window, Obeid et al. (2014) and Kopp et al. (2016)). Contrary to the initial qualification procedures, *re-qualification* procedures are usually less expensive, time-consuming and energy-consuming. Moreover, the machine is not necessarily down and can run products that are still qualified. In a work center, disqualifications can be frequent and have

* Corresponding author at: STMicroelectronics Crolles, 850 Rue Jean Monnet, F-38920 Crolles, France.
E-mail addresses: antoine.perraudat@emse.fr (A. Perraudat), dauzere-peres@emse.fr (S. Dauzère-Pérès), philippe.vialletelle@st.com (P. Vialletelle).

https://doi.org/10.1016/j.cor.2022.105813
Received 27 December 2020; Received in revised form 9 March 2022; Accepted 23 March 2022
Available online 14 April 2022
0305-0548/© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

serious consequences on factory performances, if they are not managed properly or anticipated (Kopp et al., 2018).

In this paper, we are pursuing the work of Johnzén et al. (2011) and Rowshannahad et al. (2015) on qualification management on non-identical parallel machines to optimize flexibility of work centers. Johnzén et al. (2011) propose a nonlinear qualification management optimization model to determine a single optimal qualification for a "time" flexibility measure that evaluates how balanced are the workloads between the machines. This work is extended in Rowshannahad et al. (2015) by considering the finite production capacity of machines to a capacitated time flexibility measure that evaluates how balanced are the utilization rates of the machines. To our knowledge, the qualification management optimization problem to optimize the capacitated "time" flexibility measure of Rowshannahad et al. (2015) with multiple qualifications has never been considered.

In this paper, we put ourselves in the shoes of a work center manager who must decide the best re-qualifications to perform. Therefore, we propose and evaluate new efficient optimization approaches that determine in real time, *i.e.* in small computational times, the best re-qualifications of products in a work center with non-identical parallel machines. The number of re-qualifications and the product quantities to process are given, and the objective is to maximize the capacitated "time" flexibility measure proposed by Rowshannahad et al. (2015). The most relevant approach has been implemented in an operational decision support system, which determines and proposes effective qualification plans to work center managers twenty minutes before every shift (every 8 h).

The paper is organized as follows. In Section 2, the literature on qualification management is reviewed. Our problem is formalized as a Mixed Integer NonLinear Program (MINLP) in Section 3, and solution approaches are proposed in Section 4. In Section 5, respectively Section 6, numerical results on industrial data, respectively randomly generated instances, are presented and discussed. Finally, we conclude and give perspectives in Section 7.

2. Literature review

From a general standpoint, little work has been done on qualification management to improve the manufacturing performances of work centers in semiconductor manufacturing. This can be explained by the fact that the semiconductor industry is a complex process industry and, because qualification take time and can be expensive, changing qualifications or adding costly qualifications may have not been of great importance in the past. However, with the normalization and development of custom products, with the short life cycles of products, and because of the fierce competition, manufacturers are more prone to change or add new qualifications on machines to keep or increase their competitive advantage (Johnzén et al., 2007). In the remainder of the section, closely related works are reviewed.

2.1. Assessing the qualification setting of a work center

The literature has studied the definition of Key Performance Indicators (KPIs) to measure the quality of the qualification configuration of a work center and to guide qualification decisions, in particular for short-term operational decisions. Most KPIs in the literature concern flexibility measures, and mathematical models are also introduced to optimize the KPIs.

Johnzén et al. (2011) propose "WIP", "time" and "toolset" flexibility measures. The "WIP", standing for Work-In-Process, flexibility measure evaluates how balanced are the workloads, not in number of time units but in number of product units, between the machines of the work center. Similarly, the "time" flexibility measure evaluates the balance of the workloads on the machines in number of time units. The "toolset" flexibility measure evaluates the risk of having too many products with a small number of qualified machines. A system flexibility measure is also introduced, which is a weighted sum of the three flexibility measures. Flexibility measures are used to identify bottlenecks, the lack of flexibility and to assess the impact of a qualification or disqualification on the performance of a work center. Johnzén et al. (2011) propose a nonlinear qualification management optimization model to determine a single optimal qualification for "WIP" and "time" flexibility measures. Rowshannahad et al. (2015) extends the uncapacitated time flexibility measure proposed by Johnzén et al. (2011) to a capacitated time flexibility measures by considering the finite production capacity of each machine. In this case, the utilization rate of the machine is considered instead of its workload. No solution approach is proposed to solve the multi-qualification version of the capacitated qualification management problem of Rowshannahad et al. (2015).

Rowshannahad and Dauzère-Pérès (2013) extend the "time" flexibility measure by considering batch size constraint. Rowshannahad et al. (2014) propose another measure to assess the utilization variability between machines in a work center. Numerical experiments show that reducing the utilization variability between machines with additional qualifications significantly improves the utilization balance. Finally, Pianne et al. (2016) introduce ideal and potential flexibility measures, and also consider the work center robustness.

More recently, Perraudat et al. (2019) propose a bilevel optimization approach partly based on the capacitated time flexibility measure introduced in Rowshannahad et al. (2015). The utilization balance of the machines is optimized in the follower problem, as in Rowshannahad et al. (2015), and then the throughput of the work center is computed from the utilization rates of the machines. Perraudat et al. (2019) compare single and multi-period settings and conclude that considering multiple periods may lead to more relevant qualification decisions due to production variability. Optimizing the time flexibility measure is then also a way to optimize the throughput.

2.2. Decision support systems

Interestingly, although the literature is rare on qualification management, there exist cooperation projects on qualification management between academics and semiconductor manufacturers (Leachman et al., 2002; Johnzén et al., 2009; Liao et al., 2017). Leachman et al. (2002) present a project and a decision support system (DSS) that enabled a wafer manufacturing facility to significantly reduce the mean cycle time and make substantial savings. A key element for this success was the preparation of the right qualifications with respect to the production plan. Johnzén et al. (2009) describe a qualification management software that implements the WIP, time, toolset and system flexibility measures to recommend a single qualification decision to work center managers. Finally, Liao et al. (2017) consider a strategic qualification management problem that consists in adding, or modifying, qualifications to product sites in order to improve on time deliveries. The project and the DSS are described in the paper. The qualification management problem is modeled as a MILP. A greedy heuristic is used to solve the model and recommend new qualifications.

2.3. Contributions and practical relevance

In this paper, we are pursuing the work of Johnzén et al. (2011) and Rowshannahad et al. (2015) on qualification management on non-identical parallel machines. To our knowledge, no efficient solution approach has been proposed to solve the time flexibility measure with multiple qualifications and finite production capacity. The time flexibility measure has several practical applications. It can be used to identify poorly balanced work centers, therefore identify bottleneck work centers and machines. Maximizing the time flexibility measure can be used to increase the throughput by maximizing the machine utilization balance. Note that our objective differs and is complementary to the one in Christ et al. (2019), where the utilization balance is optimized for a fixed set of qualifications, usually both the qualified and qualifiable pairs (product, machine). Our approach aims at proposing new qualifications at the operational level to improve the work center capacity.

We are particularly interested in embedding the optimization approach in a Decision Support System, where numerous scenarios should be evaluated before taking a final decision. Highlighting critical qualifications helps to improve manufacturing performances, and short computational times are necessary to solve the optimization problem. In the remainder of the paper, re-qualifications will be referred as qualifications for the sake of simplicity, as solution approaches cannot only be used for determining re-qualifications on short horizons (one day to a few weeks) but also to determine new qualifications on long horizons (a few months).

3. Problem definition and analysis

Let us consider a work center of M non-identical parallel machines which must process R different products with a strictly positive demand. Machines are non-identical, both in terms of qualifications and throughput rates. More precisely, machines are unrelated, *i.e.* there is no machine that is systematically faster than another machine for all products. Machines performing the same type of operations were most often not acquired together, and thus belong to different generations. In addition, machines do not have the same core competencies, *i.e.* all machines do not process the same types of products. A machine can only process qualified products, and a qualifiable product can be processed on a machine if it is already qualified. The qualification matrix between products and machines is known, and each product has a throughput rate on the machines on which it is qualifiable. Each machine has a finite capacity, which can be different from other machines. Among the qualifiable pairs (product, machine) not already qualified, the objective is to determine a qualification plan consisting of k new feasible qualifications in order to maximize the capacitated time flexibility measure.

The capacitated time flexibility measure, F^{Time}_{Capa}, evaluates the balance of the qualification configuration of a work center in terms of utilization rates of machines. F^{Time}_{Capa} is between 0% and 100% and enables a decision maker to evaluate potential productivity gains induced by qualifiable pairs from an initial situation. Concretely, maximizing the time flexibility measure improves the utilization balance of the machines, and therefore improves productivity as more products can be produced in less time. A better utilization balance of the machines means a better throughput and less backlog.

Finally, note that we are not interested in detailed scheduling decisions, as we focus on optimizing the utilization balance of the machines.

3.1. Problem modeling

The notations used in the paper are listed below.

Sets and indices:

$\mathcal{R} = \{1, \dots, r, \dots, R\}$,

$\mathcal{M} = \{1, \dots, m, \dots, M\}$,

$\mathcal{Q}^1 = \{(r, m) \mid q_{r,m} = 1\}$,

$\mathcal{Q}^2 = \{(r, m) \mid q_{r,m} = 2\}$.

Parameters:

$q_{r,m} \in \{0, 1, 2\}$: Is equal to 1 if machine m is currently qualified for product r, is equal to 2 if machine m is qualifiable for product r, and is equal to 0 if machine m cannot be qualified for product r,

k: Number of new qualifications,

$a_{r,m}$: Throughput rate (in number of products per hour) of product r on machine m,

c_m: Production availability or capacity (in hours) of machine m,

d_r: Quantity of product r to produce,

γ: Utilization balancing parameter, which is strictly greater than 1.

Variables:

$Y_{r,m} \in \{0, 1\}$: Is equal to 1 if product r should be qualified on machine m, and is equal to 0 otherwise,

U_m: Utilization rate of machine m,

$P_{r,m}$: Quantity of product r assigned to machine m.

Let us introduce the following optimization problem:

$$f_1(\boldsymbol{q}, k, \boldsymbol{a}, \boldsymbol{c}, \boldsymbol{d}, \gamma) = \min \quad \sum_m (U_m)^\gamma \tag{1}$$

$$\text{s.t.} \quad \sum_{r,m \mid (r,m) \in \mathcal{Q}^2} Y_{r,m} \leq k \tag{2}$$

$$U_m = \frac{1}{c_m} \sum_{r \mid (r,m) \in \mathcal{Q}^1} \frac{P_{r,m}}{a_{r,m}} \quad \forall m \tag{3}$$

$$\sum_{m \mid (r,m) \in \mathcal{Q}^1} P_{r,m} = d_r \quad \forall r \tag{4}$$

$$P_{r,m} \leq d_r \quad \forall (r, m) \in \mathcal{Q}^1 \tag{5}$$

$$P_{r,m} \leq d_r Y_{r,m} \quad \forall (r, m) \in \mathcal{Q}^2 \tag{6}$$

$$P_{r,m} \geq 0 \quad \forall (r, m) \in \mathcal{Q}^1 \vee \mathcal{Q}^2 \tag{7}$$

$$Y_{r,m} \in \{0, 1\} \quad \forall (r, m) \in \mathcal{Q}^2 \tag{8}$$

The objective function (1) aims at maximizing the utilization balance of the machines, i.e. at minimizing the sum of the utilization rates of machines as defined in Constraints (3). Constraint (2) limits the number of new qualifications, *i.e.* the size of the optimized qualification plan, to at most k. Constraints (3) compute the utilization rate for each machine in the work center. In this paper, the machine utilization rate should be understood as the "implied" machine utilization rate by the product quantities assigned to the machine. A machine utilization rate is not necessarily lower than or equal to 1 if the machine cannot process all its assigned product quantities on the horizon. Constraints (4) ensure that the demand of each product is fully assigned to the machines. Constraints (5)–(6) ensure that machine m can only process product r if r is currently qualified on m ($q_{r,m} = 1$) or is both qualifiable and proposed to be qualified ($q_{r,m} = 2$ and $Y_{r,m} = 1$). Note that the dual prices of Constraints (5)–(6) indicate the potential gain in terms of utilization balance (Bazaraa et al., 2013), and will be used in some of the heuristics proposed in Section 4. Finally, Constraints (7) are the non-negativity constraints, and Constraints (8) are the binary constraints.

Extending Rowshannahad et al. (2015), the capacitated time flexibility measure F^{Time}_{Capa} is equal to $\frac{f_1(\boldsymbol{q},\infty,\boldsymbol{a},\boldsymbol{c},\boldsymbol{d},\gamma)}{f_1(\boldsymbol{q},k,\boldsymbol{a},\boldsymbol{c},\boldsymbol{d},\gamma)} \in [0\%, 100\%]$. As $f_1(\boldsymbol{q}, \infty, \boldsymbol{a}, \boldsymbol{c}, \boldsymbol{d}, \gamma)$ is a constant term (computed by solving a nonlinear optimization problem) because all possible qualification decisions are made, maximizing F^{Time}_{Capa} then requires to solve a Mixed Integer Non-Linear (MINLP) optimization problem.

Let us discuss below some important characteristics of our problem:

- All qualifications require the same cost and time. This assumption comes from work center managers that can hardly differentiate between re-qualifications at the operational level. On a longer horizon of several weeks or months, where new qualifications need to be planned, considering different costs and times for qualifications would be relevant, although the information might not be easy to obtain.
- Demand and production capacity varying over time and disqualifications are not considered. This is because the problem is solved regularly, once every shift of 8 h for the next 24 h, and the qualifications are frequently updated given the current disqualifications and a new estimate of the quantities of products to process. Including disqualifications and time varying demand and production capacity in the problem on a longer planning horizon is left for future research.

3.2. Illustrative example: Influence of γ

γ is a critical parameter in the capacitated time flexibility measure that measures the distance between the current utilization balance and the ideal utilization balance of the machines. Rowshannahad et al. (2015) recommend adjusting γ according to the real workload distribution in the shop floor, for instance by using historical data. In the considered manufacturing system, for an horizon of 24 h, $\gamma = 4$ to $\gamma = 6$ are appropriate values. In the remainder of the paper, γ is set to 4.

In general, increasing γ leads to an increase of the total process time of machines and to a decrease of the maximum process time of machines (see Rowshannahad et al. (2015)). When $\gamma = 1$, the objective function does not maximize the utilization balance of the machines, and instead only leads to the allocation of each product on its fastest qualified machine, which is not the goal in practice.

Let us consider the following example to illustrate the influence of γ on the utilization rates of machines. Consider a work center consisting of four machines and seven products and the following parameters:

$$q = \begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 1 & 1 & 2 \\ 0 & 1 & 1 & 2 \\ 1 & 2 & 2 & 0 \\ 2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}, \quad a = \begin{pmatrix} 1 & 0 & 0.2 & 0 \\ 0 & 0.8 & 0.2 & 0.8 \\ 0 & 0.2 & 0.8 & 0.7 \\ 1 & 0.1 & 0.8 & 0 \\ 0.5 & 0 & 0.2 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 1 \end{pmatrix}$$

$$d = \begin{pmatrix} 100 & 200 & 200 & 100 & 100 & 100 & 300 \end{pmatrix},$$
$$c = \begin{pmatrix} 300 & 200 & 200 & 300 \end{pmatrix}$$

Fig. 1 illustrates the influence of γ on the utilization rates of the machines. For instance, when $\gamma = 1$, machine 4 is never used whereas, when $\gamma = 6$, the utilization rate of machine 4 is larger than the one of machine 3.

3.3. Computational complexity

Determining optimal qualification plans to maximize F^{Time}_{Capa}, or equivalently to minimize f_1, is complex as the throughput rates significantly vary from one product to another and from one machine to another, and the numbers of products and machines are large. Moreover, the effect of multiple additional qualifications on the utilization balance of the machines is difficult to capture as an initially overloaded machine can become less loaded than an initially underloaded machine after several qualifications.

Johnzén (2009) shows that optimizing the "WIP" flexibility measure is a strongly NP-Hard problem by reduction from the 3-partition problem (Garey and Johnson, 1979). The proof is based on the proof given in Aubry et al. (2008) for the Minimum Cost Load Balanced Configuration Problem (MCLBCP). Optimizing the "WIP" flexibility measure is a special case of our problem, even when $a_{r,m} = a\ \forall r,\ \forall m$, and $c_m = 1\ \forall m$.

The studied optimization is NP-Hard. In addition, we want to tackle large scale industrial instances (see Section 5.1). Efficient solution approaches must thus be designed to propose effective qualification plans that can be used by work center managers in factories.

3.4. Outer linearization algorithm for solving the nonlinear program

In this paper, solving the continuous relaxation (or when $k = 0$) of the MINLP (1)–(8) is performed by using an outer linearization algorithm, which can also be interpreted as a tangential approximation.

The outer linearization algorithm is motivated by the fact that the nonlinearity only comes from the objective function. Hence, the objective function is separable on the decision variables U_m, and it is possible to give realistic bounds to U_m. The outer linearization algorithm is used in all the solution approaches proposed in Section 4.

Consider Fig. 2 for a given machine m, which illustrates how $f(U_m) = (U_m)^\gamma$ can be linearized using outer linearization. Outer linearization constraints of $f(U_m) = (U_m)^\gamma$ are given for $U_m = 0.5$ and $U_m = 1.0$. At u_o, the outer linearization equation is equal to $u_o^\gamma + \gamma u_o^{\gamma-1}(U_m - u_o)$. By adding a sufficient number of outer linearization constraints, the continuous relaxation (or when $k = 0$) of the MINLP (1)–(8) can be solved. Nevertheless, adding all possible outer linearization constraints is unpractical, as it will lead to adding an infinite number of constraints. Adding the most relevant outer linearization constraints is therefore critical to quickly solve the MINLP.

The outer linearization is performed for all machines separately. Consider that $O_m = \max_{o \in \mathcal{O}_m}(u_o^\gamma + \gamma u_o^{\gamma-1}(U_m - u_o))$, where $\mathcal{O}_m$ is the set of outer linearization points for machine m. Intuitively, O_m represents the value of $(U_m)^\gamma$ when it is linearized by outer linearization. The objective function (1) then becomes $\min \sum_m O_m$, where $O_m \geq u_o^\gamma + \gamma u_o^{\gamma-1}(U_m - u_o)$ $\forall m$, $\forall o \in \mathcal{O}_m$. The linear program (9)–(11) below provides a lower bound on the objective function:

$$\min \quad \sum_m O_m \tag{9}$$

$$\text{s.t.} \quad O_m \geq u_o^\gamma + \gamma u_o^{\gamma-1}(U_m - u_o) \qquad \forall m, \forall o \in \mathcal{O}_m \tag{10}$$

$$(2)\text{–}(8) \tag{11}$$

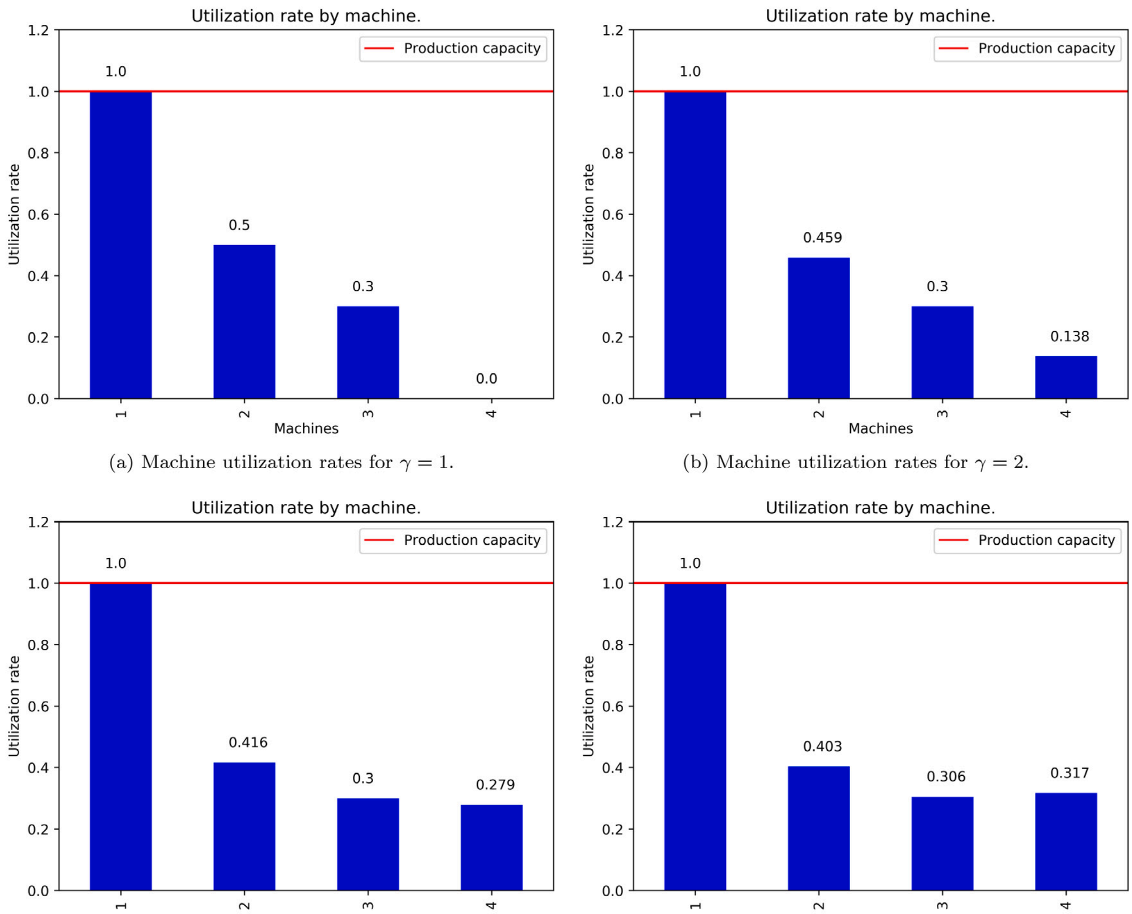
Eq. (9) is the objective function. Constraints (10) are the outer linearization constraints. Constraints (11) are the qualification constraints, the utilization rate computation constraints, and the constraints ensuring that the total demand of products must be assigned to qualified machines.

First, each set $\mathcal{O}_m$ is initialized with $0 \leq u \leq 8$. This is because, in industrial data and by experience, it is very unlikely for U_m to be larger than 8, even in a factory subject to high production variability. Once the linear program (9)–(11) is solved, $\boldsymbol{U}$ can be extracted from the incumbent solution to compute an upper bound on the objective function $\sum_m (U_m)^\gamma$. Then, additional outer linearization constraints are added to the sets $\mathcal{O}_m$ $\forall m$ until the stopping condition, *i.e.* a small relative gap ϵ between the lower and upper bounds, is met. The outer linearization is detailed in Algorithm 1.

Algorithm 1 Outer linearization algorithm.

procedure OUTER LINEARIZATION ALGORITHM
  $u_{min} \leftarrow 0$
  $u_{max} \leftarrow 8$
  $u_{step} \leftarrow 0.1$
  for $m = 1$ to M **do**
    $u_o \leftarrow u_{min}$
    while $u_o \leq u_{max}$ **do**
      $\mathcal{O}_m \leftarrow \mathcal{O}_m \cup u_o$
      $u_o \leftarrow u_o + u_{step}$
    end while
  end for
  gap $\leftarrow \infty$
  while gap $> \epsilon$ **do**
    Solve Linear Program (9)–(11) and compute $LB \leftarrow \sum_m O_m$
    $\boldsymbol{L} \leftarrow \boldsymbol{U}$
    $UB \leftarrow \sum_m (L_m)^\gamma$
    gap $\leftarrow \frac{UB-LB}{LB}$
    for $m = 1$ to M **do**
      $\mathcal{O}_m \leftarrow \mathcal{O}_m \cup L_m$
    end for
  end while
end procedure

For $\gamma = 4$, a gap of 1.10^{-4} and the values of u_{min}, u_{max} and u_{step} in Algorithm 1, empirical observations on the industrial instances of Section 5 show that the algorithm converges in less than ten iterations.



(a) Machine utilization rates for $\gamma = 1$.

(b) Machine utilization rates for $\gamma = 2$.

(c) Machine utilization rates for $\gamma = 4$.

(d) Machine utilization rates for $\gamma = 6$.

Fig. 1. Influence of γ on the utilization rates of machines.

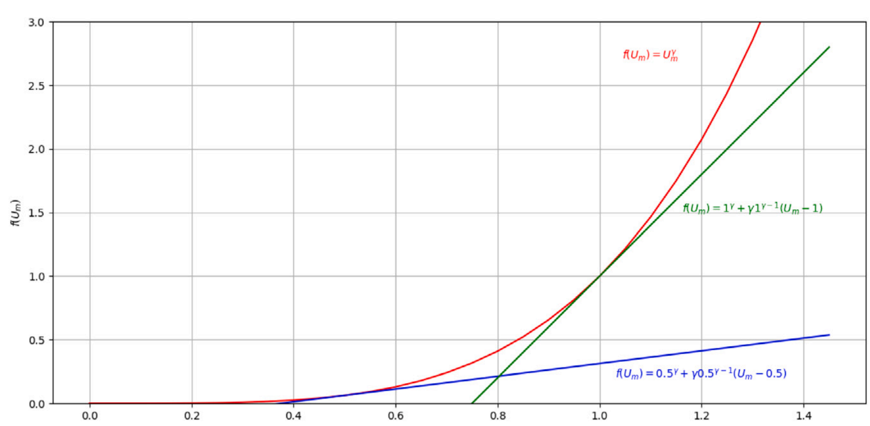


Fig. 2. Outer linearization example for $f(U_m) = (U_m)^\gamma$ for machine m.

Comparing solution approaches to solve nonlinear programs could be valuable, but is beyond the scope of this study and is left for future research.

Note that, if $\gamma = 1$, then the outer linearization algorithm is unnecessary as the objective function is linear. In this case, minimizing $\sum_m U_m$ subject to (2)–(8) is sufficient. However, as already discussed in Section 3.2, when $\gamma = 1$, the objective function does not maximize the utilization balance of the machines, and each product will be assigned to its fastest qualified machine.

4. Solution approaches

In this section, new solution approaches are proposed to solve the optimization problem with multiple qualifications formalized in Section 3.

4.1. Constructive greedy heuristic

The first proposed algorithm is a greedy heuristic, which is inspired by the "ADD" heuristics for discrete location problems (Daskin, 2011).

The pseudo code of the algorithm can be found in Algorithm 2. The greedy heuristic returns a set of qualifications $\mathcal{B}$ and f^* the value of the objective function f_1 associated to $\mathcal{B}$.

The greedy heuristic is a constructive heuristic that, at each iteration, selects the single best qualification (r^*, m^*), among all possible qualifications, that minimizes the nonlinear objective function f_1. At the end of each iteration, $\mathcal{B}$ is updated such that $\mathcal{B} = \mathcal{B} \cup (r^*, m^*)$.

Selecting the single best qualification (r^*, m^*) is performed by iterating through all possible qualifications in the qualification matrix. For each new candidate qualification (r, m) that is not already in $\mathcal{B}$, the outer linearization algorithm is run for $k = 0$ and a temporary qualification matrix $\boldsymbol{q}'$, where $q'_{r',m'} = q_{r',m'}\ \forall (r', m') \neq (r, m), \notin \mathcal{B}$, and $q'_{r,m} = 1$. The procedure is between lines 6 and 18 in Algorithm 2. The procedure is repeated until no new candidate qualification can be found, because either $|\mathcal{B}| = k$ or $\mathcal{B}$ includes all possible qualifications, Algorithm 2 returns $\mathcal{B}$. For some instances, $|\mathcal{B}|$ can therefore be smaller than k.

Algorithm 2 Greedy heuristic ($\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ)

Input data: $\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ
Output data: $\mathcal{B}$, f^*

1: **procedure** Greedy heuristic
2: $\mathcal{B} \leftarrow \infty$
3: $f^* \leftarrow \infty$
4: **for** $i = 1$ to k **do**
5: $(r^*, m^*) \leftarrow \infty$
6: **for** $r = 1$ to R
7: **for** $m = 1$ to M **do**
8: **if** $q_{r,m} = 2$ and **then** $(r,m) \notin \mathcal{B}$ **do**
9: $\boldsymbol{q'} \leftarrow \boldsymbol{q}$
10: $q'_{r,m} \leftarrow 1$
11: $f' \leftarrow f_1(\boldsymbol{q'}, 0, \boldsymbol{a}, \boldsymbol{c}, \boldsymbol{d}, \gamma)$
12: **if** $f' < f^*$ **then**
13: $(r^*, m^*) \leftarrow (r,\ m)$
14: $f^* \leftarrow f'$
15: **end if**
16: **end if**
17: **end for**
18: **end for**
19: **if** $(r^*, m^*) \neq \infty$ **then**
20: $q_{r^*,m^*} \leftarrow 1$
21: $\mathcal{B} \leftarrow \mathcal{B} \cup (r^*, m^*)$
22: **else**
23: **return** $\mathcal{B}$, f^*
24: **end if**
25: **end for**
26: **return** $\mathcal{B}$, f^*
27: **end procedure**

4.2. Local search

The local search heuristic is inspired by the "ADD–REMOVE" heuristics for discrete location problems (Daskin, 2011), and its pseudo code can be found in Algorithm 3. Similarly to the greedy heuristic of Section 4.1, the local search heuristic returns a set of qualifications $\mathcal{B}$ and f^*, the value of the objective function f_1 associated to $\mathcal{B}$.

The first step consists in determining a feasible qualification plan $\mathcal{B}'$ of value f^* with the greedy heuristic. The local search heuristic then removes one qualification at a time from $\mathcal{B}'$ and tries to swap it with a better qualification. The heuristic terminates when there is no longer a qualification that improves the objective function.

More formally, two indices i and j are introduced: i keeps track of the qualification that must be replaced, *i.e.* of the ith qualification to swap in $\mathcal{B}'$, and j keeps track of the number of swaps that are tried without improvement. Both indices i and j are initialized to 0. At each iteration of the local search heuristic, i is incremented. A subset $\mathcal{B}''$ of $\mathcal{B}'$ such that $\mathcal{B}'' = \mathcal{B}' \setminus \mathcal{B}'_i$ is used to remove one qualification from $\mathcal{B}'$, the ith qualification, to swap it with hopefully a better qualification. Algorithm 2 is then run with $q'_{r,m} = q_{r,m}\ \forall (r, m)$, and $q'_{r,m} = 1\ \forall (r, m) \in \mathcal{B}''$ and $k = 1$. At this step, Algorithm 2 returns a new (potentially) best set of qualifications $\mathcal{B}'''$ of value f^{**} such that $|\mathcal{B}'''| = 1$. If $f^{**} < f^*$, then an improving qualification has been found, j is then set to 0 and $\mathcal{B}' = \mathcal{B}'' \cup \mathcal{B}'''$. Otherwise, j is incremented. When $i = |\mathcal{B}'|$, then i is set back to 0 to avoid accessing elements that do not exist in $\mathcal{B}'$. When $j = |\mathcal{B}'|$, then the local search heuristic terminates because this means that all qualifications in $\mathcal{B}'$ were unsuccessfully swapped.

Algorithm 3 Local search ($\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ)

Input data: $\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ
Output data: $\mathcal{B}$, f^*

procedure Local search
$\mathcal{B}'$, $f^* \leftarrow$ Greedy Heuristic ($\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ)
$j \leftarrow 0$
$i \leftarrow 0$
$n \leftarrow \min(|\mathcal{B}'|, k)$
while $j \neq n$ **do**
$\mathcal{B}'' \leftarrow \mathcal{B}' \setminus \mathcal{B}'_i$
$\boldsymbol{q'} \leftarrow \boldsymbol{q}$
$q'_{r',m'} = 1\ \forall (r', m') \in \mathcal{B}''$
$i \leftarrow i + 1$
$\mathcal{B}'''$, $f^{**} \leftarrow$ Greedy Heuristic ($\boldsymbol{q'}$, 1, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ)
if $f^{**} < f^*$ **then**
$\mathcal{B}' \leftarrow \mathcal{B}'' \cup \mathcal{B}'''$
$f^* \leftarrow f^{**}$
$j \leftarrow 0$
else
$j \leftarrow j + 1$
end if
if $i = n$ **then**
$i \leftarrow 0$
end if
end while
return $\mathcal{B}'$, f^*
end procedure

4.3. Dual prices

Although heuristics presented in Sections 4.1 and 4.2 are starting points to determine good qualification plans, the number of qualifications to evaluate from one iteration to another can be substantial when the number of products and machines are large. On industrial instances, a few thousand qualifications have to be evaluated, which is not acceptable when short computational times are required. Given the problem structure and the nature of the data, we know from practical (industrial) experience that only a restricted set of qualifiable pairs (product, machine) can lead to valuable qualification plans in terms of utilization balance.

For instance, let us consider the illustrative example in Section 3.2 when $\gamma = 4$. The initial utilization balance is presented in Fig. 1. Machine 1 is critical (*i.e.* $U_1 = 1.0$), while other machines are under-loaded (*i.e.* $U_2 = 0.416 < 1.0$, $U_3 = 0.300 < 1.0$, and $U_4 = 0.279 < 1.0$). Adding new qualifications to machine 1 is probably irrelevant in terms of utilization balance, because machine 1 would be even more loaded. Therefore, in this example, the search of the optimal qualifications can potentially be restricted to machines 2, 3, and 4. All possible qualifications could be tested for the example presented in Fig. 1 as the number of products and the number of machines are small. However, because many products could be qualified on many machines in industrial data, evaluating all the possible qualifications is most often too time-consuming when short computational times are required.

To identify the most promising products and machines, and therefore to reduce the number of qualifications from one iteration to another, the dual prices of the relevant constraints of the following reformulation (when $k = 0$) of the optimization model (1)–(8) can be used, where $X_{r,m}$ is the ratio of the total quantity of product r assigned to machine m:

$$f_2(\boldsymbol{q},\boldsymbol{a},\boldsymbol{c},\boldsymbol{d},\gamma) = \min \quad \sum_m (U_m)^\gamma \tag{12}$$

$$\text{s.t.} \quad U_m = \frac{1}{c_m} \sum_{r|(r,m)\in Q^1} \frac{X_{r,m} d_r}{a_{r,m}} \quad \forall m \tag{13}$$

$$\sum_{m|(r,m)\in Q^1} X_{r,m} = 1 \quad \forall r \tag{14}$$

$$X_{r,m} \le 1 \quad \forall (r,m) \in Q^1 \tag{15}$$

$$X_{r,m} \le 0 \quad \forall (r,m) \in Q^2 \tag{16}$$

$$X_{r,m} \ge 0 \quad \forall (r,m) \in Q^1 \vee Q^2 \tag{17}$$

The objective function (12) aims at balancing the utilization rates of the machines. Constraints (13) compute the utilization rate of each machine in the work center. Constraints (14) ensure that the demand of each product is fully assigned to the machines. Constraints (15) and (16) ensure that machine m can only process product r if it is qualified on m. Finally, Constraints (7) are the non-negativity constraints for variables $X_{r,m}$.

The optimization model is close to the initial model (1)–(8), but has some significant differences. First, allocation variables $X_{r,m}$ are defined as the ratio of the quantity of product r that is assigned to machine m. Second, the constraints imposing that the current qualifications are satisfied are differentiated. With these modifications, before any qualification decision, this optimization model can be solved and the dual variable of each constraint (16) can be analyzed. The dual variable can then be interpreted as an approximation of the gain on the nonlinear objective function f_2 if product r is qualified on machine m, as dual variables can be interpreted as "the marginal rate of change in the objective function with respect to perturbations in the right-hand side of a constraint" (Bazaraa et al., 2013). f_2 would become $\sum_m (U_m)^\gamma + \lambda_{r,m}$, where $\lambda_{r,m}$ is the dual variable for the pair (r,m) of Constraint (16). Analyzing the value of $\lambda_{r,m}$ for each pair (r,m), when $q_{r,m} = 2$, allows the most promising qualification decisions *to be ranked*. The values of the dual variables associated to Constraints (16) provided by the solver we used (Lougee-Heimer, 2003, Löhndorf (2016)) were all negative or equal to 0. This makes sense as adding qualifications cannot increase the objective function. The smallest negative value indicates the most promising qualification. Note that $f_2(\boldsymbol{q},\boldsymbol{a},\boldsymbol{c},\boldsymbol{d},\gamma) = f_1(\boldsymbol{q},0,\boldsymbol{a},\boldsymbol{c},\boldsymbol{d},\gamma)$.

By embedding the use of the dual variables in the greedy heuristic, instead of testing every possible qualification at each iteration, the search space can be greatly reduced to the N most promising qualifications. For instance, at each iteration of the greedy heuristic, instead of testing 800 qualifications, only $N = 10$ are tested. If the qualifications are tested in parallel, N can be limited to the number of cores of the CPU. If, at a given iteration of the greedy heuristic, more than N dual variables have the same value, the first ones in the list are arbitrarily selected. The pseudo code of the greedy heuristic with dual variables is provided in Algorithm 4. The same principle can be applied to the local search.

Algorithm 4 is very close to Algorithm 2. The main difference is that selecting the single best qualification (r^*, m^*) at each iteration is no longer performed by iterating through all possible qualifications in the qualification matrix, but by iterating through the most promising qualifications. The set $\mathcal{C}$ of most promising qualifications in Algorithm 4 is identified by running Algorithm 5. The first step consists in solving (12)–(17) and creating a set $\mathcal{L}$ that includes all subsets of possible qualifications with the values of the dual variables associated to Constraints (16). $\mathcal{L}$ is then sorted by ascending order of $\lambda_{r,m}$, and $\mathcal{C}$ then consists of the first $\min(|L|, N)$ from $\mathcal{L}$.

Algorithm 4 Greedy heuristic with dual variables ($\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ, N)

Input data: $\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ, N
Output data: $\mathcal{B}$, f^*

procedure GREEDY HEURISTIC WITH DUAL VARIABLES
  $\mathcal{B} \leftarrow \infty$
  $f^* \leftarrow \infty$
  for $i = 1$ to k **do**
    $(r^*,m^*) \leftarrow \infty$
    $\mathcal{C} \leftarrow$ Identification of most promising qualifications ($\boldsymbol{q}$, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ, N)
    for $j = 1$ to $|\mathcal{C}|$ **do**
      $(r,m) \leftarrow \mathcal{C}_j.(r,m)$
      $\boldsymbol{q}' \leftarrow \boldsymbol{q}$
      $q'_{r,m} \leftarrow 1$
      $f' \leftarrow f_1(\boldsymbol{q}', 0, \boldsymbol{a}, \boldsymbol{c}, \boldsymbol{d}, \gamma)$
      if $f' < f^*$ **then**
        $(r^*, m^*) \leftarrow (r, m)$
        $f^* \leftarrow f'$
      end if
    end for
    if $(r*, m*) \neq \infty$ **then**
      $q_{r^*,m^*} \leftarrow 1$
      $\mathcal{B} \leftarrow \mathcal{B} \cup (r^*, m^*)$
    else
      return $\mathcal{B}$, f^*
    end if
  end for
  return $\mathcal{B}$, f^*
end procedure

Algorithm 5 Identification of most promising qualifications ($\boldsymbol{q}$, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ, N)

Input data: $\boldsymbol{q}$, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ
Output data: $\mathcal{C}$

procedure IDENTIFICATION OF MOST PROMISING QUALIFICATIONS
  $\mathcal{C} \leftarrow \infty$
  $\mathcal{L} \leftarrow \{\{(r,m), \lambda_{r,m}\} \in$ Constraint (16) $|$ $q_{r,m} = 2\}$ after solving (12)–(17).
  Sort $\mathcal{L}$ by ascending order of dual variables $\lambda_{r,m}$
  $\mathcal{C} \leftarrow$ first $\min(|L|, N)$ elements from $\mathcal{L}$
  return $\mathcal{C}$
end procedure

The local search heuristic that relies on the dual variables is very similar to the local search heuristic of Section 4.2. Instead of running the greedy heuristic (Algorithm 2) at each iteration for $k = 1$, the greedy heuristic with dual variables (Algorithm 4 is executed. The pseudo code is given in Algorithm 6.

Another "Instantaneous" Greedy Heuristic (IGH) can be designed by using dual variables in a more straightforward way. IGH builds a feasible qualification plan $\mathcal{B}$ with the k new qualifications associated to the k smallest dual variables identified after running Algorithm 5. Contrary to the greedy heuristic in Algorithm 4, IGH is not an iterative procedure since the k qualifications are taken just after the dual variables are computed. The pseudo code of the instantaneous greedy heuristic can be found in Algorithm 7.

4.4. Greedy randomized adaptive search procedure (GRASP)

A greedy randomized adaptive search procedure (GRASP) (Feo and Resende, 1989) is proposed that also relies on the dual variables. This is motivated by three remarks: (1) The greedy heuristic with dual

Algorithm 6 Local search with dual variables ($\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ, N)

Input data: $\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ
Output data: $\mathcal{B}$, f^*

procedure LOCAL SEARCH
  $\mathcal{B}'$, $f^* \leftarrow$ Greedy heuristic with dual variables ($\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ, N)
  $j \leftarrow 0$
  $i \leftarrow 0$
  $n \leftarrow \min(|\mathcal{B}'|, k)$
  while $j \neq n$ **do**
    $\mathcal{B}'' \leftarrow \mathcal{B}' \setminus \mathcal{B}'_i$
    $\boldsymbol{q'} \leftarrow \boldsymbol{q}$
    $q'_{r',m'} = 1\ \forall (r', m') \in \mathcal{B}''$
    $i \leftarrow i + 1$
    $\mathcal{B}'''$, $f^{**} \leftarrow$ Greedy heuristic with dual variables ($\boldsymbol{q'}$, 1, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ, N)
    if $f^{**} < f^*$ **then**
      $\mathcal{B}' \leftarrow \mathcal{B}'' \cup \mathcal{B}'''$
      $f^* \leftarrow f^{**}$
      $j \leftarrow 0$
    else
      $j \leftarrow j + 1$
    end if
    if $i = n$ **then**
      $i \leftarrow 0$
    end if
  end while
  return $\mathcal{B}'$, f^*
end procedure

Algorithm 7 Instantaneous Greedy Heuristic($\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ)

Input data: $\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ
Output data: $\mathcal{B}$, f^*

procedure INSTANTANEOUS GREEDY HEURISTIC($\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ)
  $\mathcal{B} \leftarrow \infty$
  $f^* \leftarrow \infty$
  $\boldsymbol{q'} \leftarrow \boldsymbol{q}$
  $C \leftarrow$ Identification of most promising qualifications ($\boldsymbol{q}$, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ, ∞)
  for $j = 1$ to $\min(k, |C|)$ **do**
    $(r, m) \leftarrow C_j.(r, m)$
    $q'_{r,m} \leftarrow 1$
  end for
  $f^* \leftarrow f_1(\boldsymbol{q'}, 0, \boldsymbol{a}, \boldsymbol{c}, \boldsymbol{d}, \gamma)$
  return $\mathcal{B}$, f^*
end procedure

prices (Algorithm 4) is relatively fast in the computational experiments, (2) The associated local search heuristic (Algorithm 6) does not always reach the maximum allowed computational time, and (3) The GRASP has also been successfully applied to difficult problems such as scheduling problems (e.g. Knopp et al. (2017) and Yepes-Borrero et al. (2021))

A GRASP is a metaheuristic where the construction of an initial solution, typically in a greedy manner, at each iteration is randomized. Initial solutions are then improved with a local search. The proposed GRASP is presented in Algorithm 8. The construction of an initial solution with at most k qualifications is performed by running Algorithm 9: The initial set of qualifications is built iteratively. At each iteration, Algorithm 5 is executed to identify a set C with at most the N most promising qualifications based on the value of the dual variables of Constraint (16). C can sometimes be empty if k is larger than the number of possible qualifications and if all possible qualifications are already made. If C is empty, Algorithm 9 immediately returns the best solution found so far as no new qualification can be made. If C is not empty, one element C_j among $\min(N, |C|)$ is then randomly selected from C and added to the initial set of qualifications $\mathcal{B}'$. $\mathcal{B}'$ is then improved by running the local search heuristic that uses the dual variables (Algorithm 6). The GRASP is stopped, in our case, when the allowed computational time is reached.

Algorithm 8 GRASP ($\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ, N)

Input data: $\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ, N
Output data: $\mathcal{B}$, f^*

procedure GRASP
  $\mathcal{B} \leftarrow \infty$
  $f^* \leftarrow \infty$
  $s \leftarrow 0$
  while $s = 0$ **do**
    $\mathcal{B}'$, $f' \leftarrow$ Greedy randomized heuristic ($\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ, N)
    $\boldsymbol{q'} \leftarrow \boldsymbol{q}$
    $q'_{r',m'} = 1\ \forall (r', m') \in \mathcal{B}'$
    $\mathcal{B}'$, $f' \leftarrow$ Local search with dual variables ($\boldsymbol{q'}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ, N)
    if $f' < f^*$ **then**
      $\mathcal{B} \leftarrow \mathcal{B}'$
      $f^* \leftarrow f'$
    end if
    $s = 1$ if allowed computational time is reached
  end while
  return $\mathcal{B}$, f^*
end procedure

Algorithm 9 Greedy randomized heuristic ($\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ, N)

Input data: $\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ, N
Output data: $\mathcal{B}$, f^*

procedure GREEDY RANDOMIZED HEURISTIC
  $\mathcal{B} \leftarrow \infty$
  $f^* \leftarrow \infty$
  $\boldsymbol{q'} \leftarrow \boldsymbol{q}$
  for $i = 1$ to k **do**
    $(r^*, m^*) \leftarrow \infty$
    $C \leftarrow$ Identification of most promising qualifications ($\boldsymbol{q}$, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ, N)
    Randomly select one element C_j from C
    $(r^*, m^*) \leftarrow C_j.(r, m)$
    if $(r^*, m^*) \neq \infty$ **then**
      $q_{r^*,m^*} \leftarrow 1$
      $\mathcal{B} \leftarrow \mathcal{B} \cup (r^*, m^*)$
    else
      return $\mathcal{B}$, f^*
    end if
  end for
  $f^* \leftarrow f_1(\boldsymbol{q'}, 0, \boldsymbol{a}, \boldsymbol{c}, \boldsymbol{d}, \gamma)$
  return $\mathcal{B}$, f^*
end procedure

4.5. Branch and bound

A branch and bound solution approach, in particular a best first approach, is also investigated to compare the solutions of the heuristic approaches to optimal solutions that can be obtained on some instances.

Branching is performed on the qualification decision variable $Y_{r,m}$ that is the closest to one but not binary. Bounding is performed by solving the continuous relaxation of the optimization model (1)–(8). A priority queue $\mathcal{Q}$ on the smallest lower bound is implemented to explore the tree. Finally, as explained in the hypothesis, a feasible solution can be quickly generated by running Algorithm 7. The pseudo code of the branch and bound algorithm is provided in Algorithm 10. Note that the lower bound is also updated by computing the smallest lower bound among all active nodes in $\mathcal{Q}$, as it can be used to further closing gaps (Bixby et al., 1999).

Algorithm 10 Branch and bound algorithm ($\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ)

procedure BRANCH AND BOUND ALGORITHM ($\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ)
  $\mathcal{B}$, $f^* \leftarrow$ Instantaneous Greedy Heuristic($\boldsymbol{q}$, k, $\boldsymbol{a}$, $\boldsymbol{c}$, $\boldsymbol{d}$, γ)
  $UB \leftarrow f^*$
  $\mathcal{Q} \leftarrow \infty$
  $\boldsymbol{Y} \leftarrow$ argmin (1)–(8) when relaxing binary constraints
  $LB \leftarrow f_1(\boldsymbol{Y}, 0, \boldsymbol{a}, \boldsymbol{c}, \boldsymbol{d}, \gamma)$
  $\mathcal{Q} \leftarrow \mathcal{Q} \cup (\boldsymbol{Y}, LB)$
  while $\mathcal{Q} \neq \infty$ or $\frac{UB-LB}{LB} > \epsilon$ **do**
    Take a node A ($\boldsymbol{Y'}$, f') off $\mathcal{Q}$
    if $\boldsymbol{Y'}$ binary and $f' \leq UB$ **then**
      $\mathcal{B} \leftarrow$ Qualifications from $\boldsymbol{Y'}$
      $UB \leftarrow f'$
    end if
    if $\boldsymbol{Y'}$ non binary **then**
      Let (r', m') be the largest non binary variable in $\boldsymbol{Y'}$
      $\boldsymbol{Y}_0 \leftarrow$ argmin (1)–(8) when relaxing binary constraints and $Y_{r',m'} = 0$
      $\boldsymbol{Y}_1 \leftarrow$ argmin (1)–(8) when relaxing binary constraints and $Y_{r',m'} = 1$
      if $f_1(\boldsymbol{Y}_0, 0, \boldsymbol{a}, \boldsymbol{c}, \boldsymbol{d}, \gamma) \geq UB$ **then**
        Prune node A
      else
        $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\boldsymbol{Y}_0, f_1(\boldsymbol{Y}_0, 0, \boldsymbol{a}, \boldsymbol{c}, \boldsymbol{d}, \gamma)\}$
      end if
      if $f_1(\boldsymbol{Y}_1, 0, \boldsymbol{a}, \boldsymbol{c}, \boldsymbol{d}, \gamma) \geq UB$ **then**
        Prune node A
      else
        $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\boldsymbol{Y}_1, f_1(\boldsymbol{Y}_1, 0, \boldsymbol{a}, \boldsymbol{c}, \boldsymbol{d}, \gamma)\}$
      end if
      $LB \leftarrow$ Smallest f' among all nodes A ($\boldsymbol{Y'}$, f') in $\mathcal{Q}$
    end if
  end while
  return $\mathcal{B}$, UB
end procedure

5. Computational study: Industrial instances

In this section, the solution approaches presented in Section 4 are compared on industrial instances. The objective is to determine the most suited solution approaches by work center given the required small computational time (a few minutes at most). Instances used in the computational study are characterized in Section 5.1, and the design of experiments is presented in Section 5.2. The main findings are discussed in Section 5.3, while the detailed numerical results can be found in Section 5.4.

5.1. Instance characterization

The computational study is performed by using historical data extracted from the most advanced production facility of STMicroelectronics located in Crolles, France. The facility is characterized by shifting bottleneck work centers, a large number of products, frequent product mix changes, high production variability, frequent disqualifications and large machine utilization rates. Four different work centers are studied (see Table 1). Work center A is an ion implantation work center where the fabrication process consists in doping products with ions. Work center B is a dry etching work center where the fabrication process consists in removing matter from the products. Work center C is a dielectric work center where the fabrication process consists in making deposits of isolation films. Finally, work center D is a metallization work center where the fabrication process consists in deposits of conductive layers on the surface of the products.

Table 1
Work centers, process types and instances.

Work center	Process type	Appendix
A	Ion implantation	Appendix A
B	Dry etching	Appendix B
C	Dielectric	Appendix C
D	Metallization	Appendix D

The four work centers are of different nature and account for nearly 40% of machines in the considered production facility (more than 600 machines in total). Moreover, many of the work centers not considered in the numerical experiments are not interesting because of their high flexibility and thus no qualification is proposed. Consequently, we believe the selected set of work centers is relevant to assess the performance of the solution approaches in a semiconductor manufacturing facility.

In total, 24 instances are used by work center to compare the solution approaches, and the production quantities and capacities for one day in each work center are used. Instances were retrieved in 2019. The instances used for the computational study are described in the appendix. For confidentiality reasons, industrial instances cannot be fully detailed as they may contain critical information. The following indicators are reported in Appendix:

- The coefficient of variation $\frac{\sigma(\boldsymbol{d})}{\overline{\boldsymbol{d}}}$, where $\overline{\boldsymbol{d}}$ is the mean demand of products, $\sigma(\boldsymbol{d})$ the standard deviation of the demand of products, the ratio $\frac{\boldsymbol{d}^-}{\overline{\boldsymbol{d}}}$, where $\boldsymbol{d}^-$ is the minimum demand over all products, and the ratio $\frac{\boldsymbol{d}^+}{\overline{\boldsymbol{d}}}$, where $\boldsymbol{d}^+$ is the maximum demand over all products,
- The coefficient of variation $\frac{\sigma(\boldsymbol{c})}{\overline{\boldsymbol{c}}}$, where $\overline{\boldsymbol{c}}$ is the mean production capacity of machines, and $\sigma(\boldsymbol{c})$ the standard deviation of the production capacity of machines, the ratio $\frac{\boldsymbol{c}^-}{\overline{\boldsymbol{c}}}$, where $\boldsymbol{c}^-$ is the minimum capacity over all machines, and the ratio $\frac{\boldsymbol{c}^+}{\overline{\boldsymbol{c}}}$, where $\boldsymbol{c}^+$ is the maximum capacity over all machines,
- The coefficient of variation $\frac{\sigma(\boldsymbol{a})}{\overline{\boldsymbol{a}}}$, where $\overline{\boldsymbol{a}}$ is the mean throughput of products on machines of initial and possible qualifications, and $\sigma(\boldsymbol{a})$ the standard deviation of the throughput of products on machines of initial and possible qualifications, the ratio $\frac{\boldsymbol{a}^-}{\overline{\boldsymbol{c}}}$, where $\boldsymbol{a}^-$ is the minimum throughput of products on machines, and the ratio $\frac{\boldsymbol{a}^+}{\overline{\boldsymbol{c}}}$, where $\boldsymbol{a}^+$ is the maximum throughput of products on machines,
- The number of initial qualification rates, and the possible qualification rates.

These indicators could be further used to generate new instances. A procedure is presented and numerical experiments are performed on randomly generated instances in Section 6. Note that it is reasonable that, given a work center, the number of products and machines do not vary much from one instance to another. This is because machines are very expensive and thus products, even if new ones are introduced, will go through the same work centers with approximately the same degree of reentrancy as previous products of the same technology node.

Two metrics are presented by instance to compare the solution approaches: The relative gain (%) on the utilization balance of the machines with respect to the initial qualification configuration and the computational time (in seconds). Numerical results are not detailed instance by instance to limit the length of the paper. More precisely, the relative gain (%) is equal to $\frac{f_1(\boldsymbol{q},0,\boldsymbol{a},\boldsymbol{c},\boldsymbol{d},\gamma)-f_1(\boldsymbol{q},k,\boldsymbol{a},\boldsymbol{c},\boldsymbol{d},\gamma)}{f_1(\boldsymbol{q},0,\boldsymbol{a},\boldsymbol{c},\boldsymbol{d},\gamma)} \times 100$ when k qualifications are proposed.

Table 2
Solution approaches tested in the computational study.

Algorithm	Dual prices	Short name	Reference section
Greedy heuristic	Off	GH	4.1
Local search	Off	LS	4.2
Greedy heuristic	On	GHDP	4.3
Local search	On	LSDP	4.3
Instantaneous Greedy heuristic (branch and bound)	On	IGH	4.3
Branch and Bound	–	B&B	4.5
Greedy Randomized Adaptive Search Procedure	On	GRASP	4.4

5.2. Design of experiments

In the computational study, the horizon is 24 h. Following the discussion in Section 3.2, γ is set to 4. The outer linearization algorithm is stopped when a relative gap lower than 0.0001 is reached. Each iteration of the outer linearization algorithm is solved by CLP, which is an open source solver (Lougee-Heimer, 2003; Löhndorf, 2016). Dual variables are then computed with CLP when the outer linearization algorithm is stopped. All solution approaches are implemented in Java 8 on a computer with an Intel(R) Xeon(R) CPU E3-1240 v5 @3.50 GHz with 4 cores and 32 GB of RAM. Note that all solution approaches are parallelized, including the Branch and Bound algorithm. As discussed in Section 4.3, the maximum number of qualification plans that are simultaneously evaluated is equal to the number of logical threads, *e.g.* 8 on the computer we used. Hence, N is set to 8 in all our approaches. For instance, 8 qualification plans are tested in parallel in the greedy heuristic of Section 4.1. In B&B, we set an optimality gap, *i.e.* $\frac{UB-LB}{LB}$, of 0.0001. If B&B is running but the gap is lower than 0.0001, then B&B is stopped and the best solution found so far is considered as numerically optimal.

Solution approaches are compared for a number of qualifications $k \in \{1,2,3,4,5,6,7,8,40,100\}$. We study all values between 1 and 8 because, in most cases, it is unnecessary to make a larger number of qualifications to significantly improve the utilization balance of the machines. In other words, the three best qualifications lead to better increase on the utilization balance of the machines than the following three best qualifications, even if the utilization balance of the machines still improves. In addition, in practice, only a limited number of qualifications is usually allowed on 24 h. Larger values of k, *i.e.* 40 and 100, are studied to evaluate the performances of solution approaches in a limited computational time.

Solution approaches are executed for the four work centers presented in Table 1. Two maximum computational times are considered: 30 s and 180 s (3 min). In addition, two initial qualification configurations are studied:

First qualification configuration. It consists in taking the industrial qualification matrix as is to test our approaches for real-life qualification configurations.

Second qualification configuration. We are also interested in testing our approaches for more extreme cases. This configuration consists in making qualifiable the qualifications that are currently not qualifiable (*i.e.* when $q_{r,m} = 0$). For each machine, the associated throughput for these cases is set to the mean throughput over other initially qualified and qualifiable machines. The density of the qualification matrix is then close to 100%. Considering this configuration is interesting for at least two reasons. The first one is to study the limit of the solutions approaches when the problem sizes increase. The second reason is practical and related to medium-term or long-term qualification management. Although $q_{r,m} = 0$ means that product r cannot be currently qualified on machine m, investigating if conducting a non-existing time-costly qualification is still relevant, in particular when new products or new machines are introduced.

The solution approaches are presented in Table 2. They are summarized by their name and whether dual prices are used. In total, six different solution approaches are compared to generate a qualification plan for short-term qualification management. For the sake of presentation, short names are given to the solution approaches (see Table 2) to present the numerical results in Section 5.4.

5.3. Main findings

Numerical results in Section 5.4 show that all algorithms do not perform equally. Generally, GH and LS are irrelevant because GHDP and LSDP determine qualification plans of similar or better quality in smaller computational times. However, depending on the work center, the qualification configuration and the computational budget, the other solution approaches are valuable to a certain extent:

- There is no solution approach that systematically outperforms all other solution approaches in all experiments,
- Restricting the search space by using the dual prices is in most cases relevant both in terms of solution quality and computational time,
- For work center B and both qualification configurations, GHDP, LSDP and GRASP provide qualification plans that are of similar quality in terms of gain.
- For a very small computational budget, instantaneous or of a few seconds, allowed in the Decision Support System, IGH is the most suitable approach, in particular for $k > 1$, because the computational time is independent of k, no matter the work center and the qualification configuration. However, a qualification plan determined by IGH may be of poor quality compared to GHDP, because one machine could inappropriately be overqualified at the expense of other machines. Therefore, a qualification plan may need manual rework by work center managers in the Decision Support System.
- For the first qualification configuration and small work centers or work centers with a small number of possible qualifications, B&B is particularly suitable as it can determine optimal solutions in less than 180 s. B&B is also suitable for the second qualification configuration when $k \leq 3$.
- For the first qualification configuration and large work centers or work centers with a large number of possible qualifications, using GHDP seems the best policy. GHDP determines solutions that are close to the optimal solutions determined by B&B. LSDP and GRASP are only slightly better than GHDP but could be considered if work center managers accept larger computational times, which can be the case for large work centers such as work centers B and C.
- For the second qualification configuration, GRASP seems to be the best approach as it always outperforms GHDP and in most cases also outperforms LSDP. GRASP determines solutions that are close to the optimal solutions.
- This study shows that, although an optimization problem can be NP-Hard, studying the nature of the data is essential to design efficient solution approaches. For manufacturing facilities with a large product variety, using dual variables to guide the solution approach is shown to be effective and efficient for different types of work centers and qualification configurations.

- The gains between the first and second qualification configurations are very different. This shows that machines that cannot be qualified for some products, *i.e.* such that $q_{r,m} = 0$ in the first configuration, could potentially lead to substantial improvements for the work center in terms of utilization balance of the machines. This may be worth to investigate, and to check if these forbidden qualifications could actually be made, *i.e.* whether the associated $q_{r,m} = 0$ in the first configuration could be changed to $q_{r,m} = 2$.

5.4. Detailed numerical results

Solution approaches were actually only executed for 180 s, but all evaluated sets of qualifications were kept while the approaches were running. Computational times are reported as follows:

- If the total computational time is larger than or equal to 180 s, then the solution approach, both for a computational time limit of 30 and 180 s, does not terminate on time. In this case, 180 s, respectively 30 s, is reported for a computational time limit of 180 s, respectively 30 s.
- If the computational is smaller than 180 s but larger than 30 s, then the total computational time is reported for a computational time limit of 180 s, but 30 s are reported for a computational time limit of 30 s.
- If the computational time is smaller than or equal to 30 s, then the total computational time is reported for both computational time limits.

A computational time is associated to each reported solution. For a computational time limit of 180 s, respectively 30 s, only solutions that could be reached before 180 s, respectively 30 s, are kept in the numerical results. Finally, the gain associated to the best set of qualifications is reported in the numerical experiments for each computation time limit and each k.

5.4.1. First qualification configuration

For each k and each work center, Table 3, respectively Table 4, shows the numerical results for a computational time limit of 30 s, respectively 180 s. Table 5 provides details on the Branch and Bound algorithm for the first qualification configuration such as the initial relaxation gap at the root node, the final relaxation gap when the algorithm stops, the total number of explored nodes and the number of instances where the optimal solution is found. Note that, when an optimal solution is found, a gap of 0% is reported.

Depending on the work center, all solution approaches may not determine satisfactory qualification plans, in particular GH and LS compared to GHDP and LSP. For instance, for a computational time limit of 30 s, GHDP performs better on average than GH from $k = 6$. The mean gain with GH is equal to 6.9% whereas the mean gain with GHDP is equal to 7.0%. The larger k, the larger the difference between GHDP and GH. This is due to the fact that, although the mean run time of GH is equal to 24.8 s, on several instances GH cannot find a complete qualification plan because it reaches the computational time limit. This is confirmed by experiments for $k = 7$ and $k = 8$. This shows that, for a small computational time limit, using dual variables is valuable. For a computational time limit of 180 s, GH actually performs slightly better on average than GHDP for $k = 6$ and $k = 7$. This is because the dual variables are only indicative of the marginal increase in the objective function. However, when $k = 40$ or $k = 100$, GHDP determines better qualification plans than GH because GH reaches the computational time limit.

Generally, GHDP determines solutions that are close to the optimal solutions, and can even outperform B&B for work centers B and C. On average, the benefit of LSDP is very limited for a substantial increase of the computational time. On average, LSDP only slightly improves (at most by 0.1%) the mean gain of GHDP. GRASP determines slightly better solutions than GHDP and LSDP. For instance, for $k \in \{40, 100\}$ and work center C, the mean gain determined by GRASP is equal to 18.2% whereas the mean gain determined by LSDP is equal to 17.4%. Similar observations can be made for work center B. However, in general, the difference between LSDP and GRASP do not exceed 0.1%.

Note that, GHDP, LSDP and GRASP do not necessarily determine optimal solutions for $k = 1$, although close to optimal solutions are obtained. This is because dual variables are only indicative, and do not necessarily guarantee that the marginal rate can be fully reached. Also, it is possible that several dual variables have the same value but, in practice, does not lead to the same gain on the utilization balance of the machines. In addition, there could be more than N dual variables with the same value but only N are considered.

For work centers A and D, where the number of possible qualifications is the smallest among the studied work centers, B&B determine optimal solutions for all instances in less than 30 s when $k \leq 8$, and for 23 out of the 24 instances when $k = 40$ or 100. All optimal solutions are determined for a computational time limit of 180 s (see Table 5). The mean computational time to reach optimal solutions do not exceed a few seconds. For work centers B and D, B&B is better than any other approach for $k \in \{1, 2, 3\}$, but is outperformed by either LSDP or GRASP for larger values of k. In most cases, when B&B is not the best solution approach, GRASP determines better solutions, which are always close to the optimal solutions.

It is interesting to observe that, for $k = 100$, fewer nodes are explored by B&B than when $k = 40$, which can be counter intuitive because more combinations should be tested. However, as the number of qualifications increases, almost all relevant qualification decisions are already binary in the continuous relaxation at the root node (due to the nature of data), and thus considered in the initial feasible qualification plan determined by IGH. Hence, the required branching effort is reduced because the resulting number of "choices" is smaller. Similarly, almost all relevant qualifications are determined by using the k largest dual variables. Therefore, on industrial data, as soon as k exceeds a few qualifications, even if the optimization problem is NP-Hard, the theoretical combinatorial aspect of the problem fades.

Numerical results show that, for small work centers or work centers with a limited number of possible qualifications, B&B performs better than other solution approaches both in terms of solution quality and computational times. For instance, in practice, B&B should be used for work centers A and D. This also shows that using empirical observations and dual variables, which are part of the B&B solution approach, is relevant for these work centers. For larger work centers, numerical results show that B&B is outperformed by LSDP and GRASP which should be preferred. In particular, as work centers get larger, larger computational time limits seem acceptable. In this case, GRASP is probably the best solution approach.

5.4.2. Second qualification configuration

For each value of k and each work center, Table 6, respectively Table 7, shows the numerical results for a computational time limit of 30 s, respectively for a computational time limit of 180 s. Table 8 provides details on the Branch and Bound algorithm for the second qualification configuration such as the initial relaxation gap at the root node, the final relaxation gap when the algorithm stops, the total number of explored nodes and the number of instances where the optimal solution is found.

GH and LS always propose unsatisfactory qualification plans, whether the computational time limit is 30 or 180 s, even for $k = 1$, and are always outperformed by GHDP, LSDP and GRASP. When $k = 1$, GH may determine qualification plans close in terms of quality to the qualification plans of GHDP. However, such a quality in the qualification plans is almost obtained "by chance" because the computational time limit of 30 or 180 s is always reached and good solutions are among the first ones evaluated. Such differences are due to the significant combinatorial explosion. For instance, consider instance 1 of work center B. There are 768 products and 162 machines (see B). The initial

Table 3
Numerical results for a computational time limit of 30 s and the first qualification configuration. Cells in italic, respectively bold, indicate the smallest, respectively the largest, gain value by k and work center.

Work center	k	GH		LS		GHDP		LSDP		GRASP		IGH		B&B	
		Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)
A	1	2.7	7.5	2.7	7.5	2.7	0.8	2.7	0.8	2.7	30.0	*2.1*	0.2	**2.7**	0.2
	2	4.1	7.5	4.1	16.6	4.1	0.8	4.1	2.0	4.1	30.0	*3.4*	0.2	**4.2**	0.3
	3	5.1	11.7	5.1	25.1	5.1	1.4	5.1	4.6	5.1	30.0	*4.5*	0.2	**5.1**	0.4
	4	5.9	16.2	5.9	28.5	5.9	2.1	5.9	6.1	5.9	30.0	*5.1*	0.2	**5.9**	0.4
	5	6.5	21.1	6.5	29.7	6.5	4.1	6.6	7.6	6.6	30.0	*5.7*	0.2	**6.6**	0.5
	6	6.9	24.8	6.9	30.0	7.0	4.6	7.1	9.0	7.1	30.0	*6.0*	0.2	**7.1**	0.5
	7	7.0	27.0	7.0	30.0	7.4	5.0	7.5	9.8	7.5	30.0	*6.3*	0.2	**7.5**	0.6
	8	7.1	28.3	7.1	30.0	7.8	6.1	7.8	12.0	7.8	30.0	*6.7*	0.2	**7.8**	0.8
	40	*7.2*	30.0	7.2	30.0	10.4	24.5	10.4	30.0	**10.4**	30.0	9.6	0.2	10.4	4.1
	100	*7.2*	30.0	7.2	30.0	10.5	29.9	10.5	30.0	10.7	30.0	10.6	0.2	**10.8**	2.2
B	1	15.4	30.0	15.4	30.0	15.8	4.4	15.8	4.4	15.8	30.0	*15.1*	2.3	**15.9**	10.1
	2	*15.4*	30.0	15.4	30.0	20.8	4.5	20.8	9.1	20.8	30.0	17.6	2.3	**20.9**	17.5
	3	*15.4*	30.0	15.4	30.0	23.0	6.8	23.1	14.2	23.2	30.0	18.9	2.3	**23.2**	24.9
	4	*15.4*	30.0	15.4	30.0	24.6	9.0	**24.7**	18.7	24.7	30.0	19.9	2.4	24.1	28.8
	5	*15.4*	30.0	15.4	30.0	25.6	11.2	**25.8**	24.2	25.8	30.0	20.4	2.3	22.9	29.6
	6	*15.4*	30.0	15.4	30.0	26.5	13.4	**26.7**	27.3	26.6	30.0	20.9	2.3	22.3	29.6
	7	*15.4*	30.0	15.4	30.0	27.2	15.4	**27.3**	29.9	27.2	30.0	21.6	2.3	21.6	30.0
	8	*15.4*	30.0	15.4	30.0	27.7	17.8	**27.7**	30.0	27.7	30.0	21.9	2.5	21.9	30.0
	40	*15.4*	30.0	15.4	30.0	28.7	30.0	28.7	30.0	**29.1**	30.0	25.9	2.4	25.9	30.0
	100	*15.4*	30.0	15.4	30.0	28.7	30.0	28.7	30.0	**29.1**	30.0	28.3	2.6	28.3	30.0
C	1	7.4	30.0	7.4	30.0	7.4	1.5	7.4	1.5	7.4	30.0	*6.7*	0.6	**7.4**	3.1
	2	8.9	30.0	8.9	30.0	10.4	1.5	10.4	5.2	10.5	30.0	*8.1*	0.6	**10.5**	7.8
	3	*8.8*	30.0	8.8	30.0	12.1	2.4	12.1	6.9	**12.1**	30.0	8.9	0.6	12.1	15.3
	4	*8.9*	30.0	8.9	30.0	13.2	4.8	13.3	8.5	**13.3**	30.0	9.3	0.6	13.0	21.2
	5	*9.0*	30.0	9.0	30.0	14.1	5.9	14.1	11.6	**14.1**	30.0	9.8	0.6	13.3	25.4
	6	*8.9*	30.0	8.9	30.0	14.7	6.8	14.7	15.2	**14.7**	30.0	10.0	0.6	12.8	25.7
	7	*8.8*	30.0	8.8	30.0	15.2	7.8	15.2	18.7	**15.2**	30.0	10.8	0.6	12.9	27.5
	8	*8.8*	30.0	8.8	30.0	15.6	8.3	15.6	27.5	**15.7**	30.0	11.8	0.6	13.7	28.0
	40	*9.0*	30.0	9.0	30.0	17.4	30.0	17.4	30.0	**18.2**	30.0	16.2	0.6	16.2	30.0
	100	*8.9*	30.0	8.9	30.0	17.4	30.0	17.4	30.0	**18.3**	30.0	17.7	0.7	17.8	27.5
D	1	3.7	0.9	3.7	0.9	3.7	0.7	3.7	0.7	3.7	30.0	*3.7*	0.1	**3.7**	0.2
	2	5.1	0.9	5.1	2.2	5.1	0.7	5.1	1.9	5.1	30.0	*4.5*	0.1	**5.1**	0.4
	3	5.8	1.5	5.8	4.6	5.8	1.3	5.8	4.3	5.8	30.0	*5.0*	0.1	**5.8**	0.5
	4	6.3	2.1	6.3	5.6	6.3	1.9	6.3	5.5	6.3	30.0	*5.4*	0.1	**6.3**	0.9
	5	6.6	3.1	6.6	7.6	6.6	2.8	6.6	7.3	6.6	30.0	*5.8*	0.1	**6.6**	1.2
	6	6.8	4.8	6.8	9.3	6.8	4.4	6.8	8.3	6.8	30.0	*6.1*	0.1	**6.8**	1.3
	7	6.9	5.2	6.9	11.0	6.9	5.1	6.9	10.7	6.9	30.0	*6.3*	0.1	**6.9**	1.3
	8	7.0	6.2	7.0	20.1	7.0	5.6	7.0	18.2	7.0	30.0	*6.5*	0.1	**7.0**	1.4
	40	*7.5*	30.0	7.5	30.0	7.5	21.3	7.5	27.3	7.5	30.0	7.5	0.1	**7.5**	0.3
	100	*7.5*	30.0	7.5	30.0	7.5	21.7	7.5	27.6	7.5	30.0	7.5	0.1	**7.5**	0.1

number of qualifications is equal to 3,975. For the second qualification configuration, this means that the total number of qualifiable pairs (product, machine) is equal to $768 \times 162 - 3,975 = 120,441$. 120,441 qualifications cannot be evaluated in 30 or 180 s. The use of dual variables is therefore particularly relevant to restrict the search space to the N most promising qualifications. Doing so immunizes GHDP, LSDP and GRASP against the increase in the number of qualifiable products on each machine.

Contrary to the first qualification configuration, B&B performs poorly on a large number of experiments. For work center A, B&B is relevant until approximately $k = 5$ where it is outperformed by GHDP, LSDP or GRASP. For work center B, B&B is relevant until approximately $k = 7$ where it is outperformed by GHDP, LSDP or GRASP. For work centers B and C, B&B is outperformed by GHDP, LSDP or GRASP as soon as $k = 3$. Contrary to the first qualification configuration, the poor performance of B&B can be explained by the fact that empirical observations that motivate B&B do not longer hold and cause a combinatorial explosion. For instance, many qualification decisions are relevant and the continuous relaxation may no longer be strong. Many qualifications can be relevant to improve the utilization balance of the machines and the qualification matrix is now dense. Another reason that explains why B&B performs worst on the second qualification configuration than in the first qualification configuration is the fact that the linear relaxation is more computationally expensive. For instance, consider work center B and a computational time limit of 180 s. Several hundreds nodes could be explored for the first qualification configuration (see Table 5) whereas no more than 30 nodes can be explored for the second qualification configuration (see Table 8).

Although the mean run time is still very small, less than 1 s for work centers A, C, and D and less than 3 s for work center B, IGH is less relevant to determine qualification plans in the second qualification configuration than in the first qualification configuration. IGH is far from the best solution found by other solution approaches because many dual variables that rank among the best ones when assessing the initial situation often correspond to the same product, or the same machine. In practice, qualifying the same product, or the same machine, many times is irrelevant to efficiently improve the utilization balance of the machines.

LSDP improves the initial qualification plan determined by GHDP more in the second qualification configuration than in the first qualification configuration. The improvement can reach more than 1%, as shown in Table 7 for work center A and $k = 8$.

From a general perspective, GRASP is the best solution approach because it always outperforms GHDP and in most cases also outperforms LSDP. GRASP also determines the best solutions for most values of k for work centers B and C. And when GRASP is not the best solution approach, it is still close to the best found solution, even for experiments for work centers A and D where B&B could determine optimal solutions. LSDP also determines satisfactory qualifications plans, but which in general are of lower quality than the qualifications plans determined by GRASP. Thus, GRASP seems to be most relevant approach to tackle the

Table 4
Numerical results for a computational time limit of 180 s and the first qualification configuration. Cells in italic, respectively bold, indicate the smallest, respectively the largest, gain value by k and work center.

Work center	k	GH		LS		GHDP		LSDP		GRASP		IGH		B&B	
		Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)
A	1	2.7	7.5	2.7	7.5	2.7	0.8	2.7	0.8	2.7	180.0	*2.1*	0.2	**2.7**	0.2
	2	4.1	7.5	4.1	16.7	4.1	0.8	4.1	2.0	4.1	180.0	*3.4*	0.2	**4.2**	0.3
	3	5.1	11.7	5.1	30.2	5.1	1.4	5.1	4.6	5.1	180.0	*4.5*	0.2	**5.1**	0.4
	4	5.9	16.5	5.9	50.6	5.9	2.1	5.9	6.1	5.9	180.0	*5.1*	0.2	**5.9**	0.4
	5	6.5	22.0	6.6	63.7	6.5	4.1	6.6	7.6	6.6	180.0	*5.7*	0.2	**6.6**	0.5
	6	7.1	29.2	7.1	83.6	7.0	4.6	7.1	9.0	7.1	180.0	*6.0*	0.2	**7.1**	0.5
	7	7.5	37.4	7.5	103.5	7.4	5.0	7.5	9.8	7.5	180.0	*6.3*	0.2	**7.5**	0.6
	8	7.8	44.3	7.8	116.0	7.8	6.1	7.8	12.0	7.8	180.0	*6.7*	0.2	**7.8**	0.8
	40	9.7	174.0	9.7	180.0	10.4	24.5	10.4	65.0	**10.4**	180.0	9.6	0.2	10.4	5.8
	100	9.6	180.0	9.6	180.0	10.8	54.2	10.8	148.1	10.8	180.0	10.6	0.2	**10.8**	2.2
B	1	15.9	180.0	15.9	180.0	15.8	4.4	15.8	4.4	15.8	180.0	*15.1*	2.3	**15.9**	11.9
	2	15.9	180.0	15.9	180.0	20.8	4.5	20.8	9.1	20.9	180.0	17.6	2.3	**20.9**	32.1
	3	15.9	180.0	15.9	180.0	23.0	6.8	23.1	14.2	23.2	180.0	18.9	2.3	**23.2**	77.9
	4	15.9	180.0	15.9	180.0	24.6	9.0	**24.7**	18.7	24.8	180.0	19.9	2.4	24.8	121.7
	5	15.9	180.0	15.9	180.0	25.6	11.2	**25.8**	24.5	25.9	180.0	20.4	2.3	25.2	160.6
	6	15.9	180.0	15.9	180.0	26.5	13.4	**26.7**	29.3	26.7	180.0	20.9	2.3	25.5	166.7
	7	15.9	180.0	15.9	180.0	27.2	15.4	**27.3**	35.6	27.3	180.0	21.6	2.3	25.7	175.3
	8	15.9	180.0	15.9	180.0	27.7	17.8	**27.8**	42.8	27.8	180.0	21.9	2.5	24.5	177.0
	40	15.9	180.0	15.9	180.0	29.5	89.5	29.5	180.0	**29.5**	180.0	25.9	2.4	25.9	180.0
	100	15.9	180.0	15.9	180.0	29.6	180.0	29.6	180.0	**29.6**	180.0	28.3	2.6	28.3	180.0
C	1	7.4	87.1	7.4	87.1	7.4	1.5	7.4	1.5	7.4	180.0	*6.7*	0.6	**7.4**	3.1
	2	10.4	86.8	10.5	173.9	10.4	1.5	10.4	5.2	10.5	180.0	*8.1*	0.6	**10.5**	10.4
	3	12.1	144.7	12.1	180.0	12.1	2.4	12.1	6.9	**12.1**	180.0	8.9	0.6	12.1	34.8
	4	12.7	177.0	12.7	180.0	13.2	4.8	13.3	8.5	**13.3**	180.0	9.3	0.6	13.2	71.8
	5	12.9	178.4	12.9	180.0	14.1	5.9	14.1	12.0	**14.1**	180.0	9.8	0.6	13.5	91.5
	6	12.9	180.0	12.9	180.0	14.7	6.8	14.7	15.2	**14.8**	180.0	10.0	0.6	14.0	114.9
	7	12.9	180.0	12.9	180.0	15.2	7.8	15.2	20.9	**15.2**	180.0	10.8	0.6	14.2	135.5
	8	12.9	180.0	12.9	180.0	15.6	8.3	15.6	35.2	**15.7**	180.0	11.8	0.6	14.8	138.7
	40	12.9	180.0	12.9	180.0	18.2	114.3	18.2	180.0	**18.2**	180.0	16.2	0.6	16.2	180.0
	100	12.9	180.0	12.9	180.0	18.3	180.0	18.3	180.0	**18.3**	180.0	17.7	0.7	18.3	44.4
D	1	3.7	0.9	3.7	0.9	3.7	0.7	3.7	0.7	3.7	180.0	*3.7*	0.1	**3.7**	0.2
	2	5.1	0.9	5.1	2.2	5.1	0.7	5.1	1.9	5.1	180.0	*4.5*	0.1	**5.1**	0.4
	3	5.8	1.5	5.8	4.6	5.8	1.3	5.8	4.3	5.8	180.0	*5.0*	0.1	**5.8**	0.5
	4	6.3	2.1	6.3	5.6	6.3	1.9	6.3	5.5	6.3	180.0	*5.4*	0.1	**6.3**	0.9
	5	6.6	3.1	6.6	7.6	6.6	2.8	6.6	7.3	6.6	180.0	*5.8*	0.1	**6.6**	1.2
	6	6.8	4.8	6.8	9.3	6.8	4.4	6.8	8.3	6.8	180.0	*6.1*	0.1	**6.8**	1.3
	7	6.9	5.2	6.9	11.0	6.9	5.1	6.9	10.7	6.9	180.0	*6.3*	0.1	**6.9**	1.3
	8	7.0	6.2	7.0	20.9	7.0	5.6	7.0	18.5	7.0	180.0	*6.5*	0.1	**7.0**	1.4
	40	7.5	115.5	7.5	168.3	7.5	33.8	7.5	68.1	7.5	180.0	7.5	0.1	**7.5**	0.3
	100	7.5	146.1	7.5	164.9	7.5	36.0	7.5	68.8	7.5	180.0	7.5	0.1	**7.5**	0.1

studied optimization problem on very large scale industrial instances, even for a small computational budget.

Another interesting conclusion that can be drawn from these numerical experiments is that the gain between the first and second qualification configurations are very different. Consider $k = 1$ where the optimal solution is found for all instances by B&B. For the first qualification configuration, the mean gain is equal to 2.7% whereas it is equal to 15.4% for the second qualification configuration. The difference is significant. This shows that machines that cannot be qualified for some products, *i.e.* such that $q_{r,m} = 0$ in the first configuration, could potentially lead to substantial improvements for the work center in terms of utilization balance of the machines. This may be worth to investigate, and to check if these forbidden qualifications could actually be made, *i.e.* whether the associated $q_{r,m} = 0$ in the first configuration could be changed to $q_{r,m} = 2$.

Note that if many dual variables have the same value, the solution approaches that are based on dual variables lose quality if a restricted number of qualifications is tested at each iteration. However, numerical results show that this loss is not substantial and does not seem to depend on the number of products R and machines M. If the loss was significant, the number of qualifications tested at each iteration in GHDP, LSDP or GRASP could be increased to overcome the loss of quality.

6. Computational study: Random instances

In this section, additional numerical experiments are performed on 96 randomly generated instances to further compare and validate the proposed solution approaches. The design of experiments is similar to the one in Section 5.2. The procedure to generate the random instances is detailed in Section 6.1 while, in Section 6.2, the main findings are discussed. Finally, Section 6.3 analyzes the numerical results in more details.

6.1. Instance generation

To generate random instances, the industrial data are used as a baseline. We proceed as follows: First, the demand and throughput are randomly generated. Then, initial and possible qualifications are randomly generated. Finally, the capacity is computed from the demand and throughput and but also randomly generated.

For each of the 96 randomly generated instances, each of the following "hyperparameters" are randomly selected from one the 96 industrial instances: The number of products *and* machines, $\frac{d^-}{\bar{d}}$, $\frac{d^+}{\bar{d}}$, $\frac{\sigma(d)}{\bar{d}}$, $\frac{a^-}{\bar{a}}$, $\frac{a^+}{\bar{a}}$, $\frac{\sigma(a)}{\bar{a}}$, $\frac{c^-}{\bar{c}}$, $\frac{c^+}{\bar{c}}$, $\frac{\sigma(c)}{\bar{c}}$, the initial qualification rate, and the possible qualification rate. Selecting these parameters is critical as they are used to randomly generate $\boldsymbol{c}$, $\boldsymbol{d}$, $\boldsymbol{q}$, and $\boldsymbol{a}$. Each hyperparameter, for

Table 5
Details of the branch and bound solution approach for the first qualification configuration.

Work center	k	30 seconds				180 seconds			
		Initial Gap (%)	Final Gap (%)	Nodes	Number of optimal solutions	Initial Gap (%)	Final Gap (%)	Nodes	Number of optimal solutions
A	1	0.59	0.00	0.3	24	0.59	0.00	0.3	24
	2	0.89	0.00	0.8	24	0.89	0.00	0.8	24
	3	0.75	0.00	2.1	24	0.75	0.00	2.1	24
	4	0.92	0.00	2.3	24	0.92	0.00	2.3	24
	5	0.99	0.00	4.5	24	0.99	0.00	4.5	24
	6	1.22	0.00	6.6	24	1.22	0.00	6.6	24
	7	1.30	0.00	6.9	24	1.30	0.00	6.9	24
	8	1.26	0.00	10.5	24	1.26	0.00	10.5	24
	40	0.96	0.08	201.3	23	0.96	0.00	303.8	24
	100	0.20	0.00	81.9	23	0.20	0.00	84.4	24
B	1	5.75	2.31	6.8	23	5.75	2.28	8.8	24
	2	8.89	0.67	21.2	15	8.89	0.27	43.6	22
	3	10.33	0.91	38.3	9	10.33	0.23	124.6	18
	4	11.30	2.20	54.7	3	11.30	0.46	235.5	13
	5	12.01	5.91	65.8	1	12.01	1.62	359.1	7
	6	12.16	9.03	67.0	1	12.16	2.46	415.7	3
	7	11.75	11.28	70.2	0	11.75	3.34	459.5	1
	8	11.85	11.54	70.2	0	11.85	5.55	483.3	1
	40	7.05	7.05	70.0	0	7.05	7.05	503.5	0
	100	2.59	2.59	70.5	0	2.59	2.59	499.9	0
C	1	2.98	0.02	7.1	24	2.98	0.02	7.1	24
	2	4.47	0.05	48.9	22	4.47	0.02	71.9	24
	3	5.29	0.15	125.6	19	5.29	0.03	332.2	22
	4	6.24	0.69	222.4	11	6.24	0.21	808.0	17
	5	6.54	1.29	326.2	9	6.54	0.97	1187.2	16
	6	7.07	2.69	344.3	6	7.07	1.04	1615.5	12
	7	6.68	3.46	383.0	4	6.68	1.51	2022.7	8
	8	5.90	2.90	410.5	4	5.90	1.32	2120.8	7
	40	2.67	2.67	453.6	0	2.67	2.67	3018.2	0
	100	0.77	0.66	402.9	7	0.77	0.00	669.9	24
D	1	0.26	0.00	1.1	24	0.26	0.00	1.1	24
	2	0.84	0.00	3.2	24	0.84	0.00	3.2	24
	3	1.08	0.00	10.3	24	1.08	0.00	10.3	24
	4	1.08	0.00	27.1	24	1.08	0.00	27.1	24
	5	1.01	0.00	63.7	24	1.01	0.00	63.7	24
	6	0.86	0.00	71.9	24	0.86	0.00	71.9	24
	7	0.79	0.00	74.2	24	0.79	0.00	74.2	24
	8	0.67	0.00	75.2	24	0.67	0.00	75.2	24
	40	0.03	0.00	11.5	24	0.03	0.00	11.5	24
	100	0.03	0.00	0.0	24	0.03	0.00	0.0	24

instance $\frac{c^-}{\bar{c}}$ and $\frac{c^+}{\bar{c}}$, is randomly selected from one the 96 industrial instances. Note that, even in the case where all the "hyperparameters" would be drawn from the same industrial instance, the resulting instance would still be different from the industrial instance as $\boldsymbol{c}$, $\boldsymbol{d}$, $\boldsymbol{q}$, and $\boldsymbol{a}$ are randomly generated.

Generate the demand and the throughput. It is assumed that the demand follows a normal distribution of mean $\overline{\boldsymbol{d}}$, standard deviation $\sigma(\boldsymbol{d})$. A value is then generated with $\min(\boldsymbol{d}^+, \max(\overline{\boldsymbol{d}}+\sigma(\boldsymbol{d})\mathcal{N}(0.0, 1.0), \boldsymbol{d}^-))$, where $\mathcal{G}(0.0, 1.0)$ is a random value drawn from the normal law that has a mean 0.0 and standard deviation 1.0. It is assumed that $\overline{\boldsymbol{d}} = 100$ so that $\sigma(\boldsymbol{d})$, $\boldsymbol{d}^-$, and $\boldsymbol{d}^+$ can be generated from the industrial data as only ratios are provided in Appendix A. Similarly, it is assumed the throughput follows a normal distribution of mean $\overline{\boldsymbol{a}}$ and standard deviation $\sigma(\boldsymbol{a})$. A value is then generated with $\min(\boldsymbol{a}^+, \max(\overline{\boldsymbol{a}}+\sigma(\boldsymbol{a})\mathcal{G}(0.0, 1.0), \boldsymbol{a}^-))$. It is also assumed that $\overline{\boldsymbol{a}} = 100$ so that $\sigma(\boldsymbol{a})$, $\boldsymbol{a}^-$, and $\boldsymbol{a}^+$ can be generated from the industrial data.

Generate the initial and possible qualifications. First, for each recipe, a machine is randomly selected to be initially qualified for the recipe. Similarly, for each machine, a recipe is randomly selected to be initially qualified on the machine. Then, additional initial and possible qualifications are added based on the initial qualification and possible qualification rates of the random instances. Let us define v_1 as the initial qualification rate and v_2 as the possible qualification rate of the industrial instance. A list of all couples (product, machine) is created then shuffled. The first v_1 couples (product, machine) of the list are selected and will be initially qualified, and the following v_2 couples (product, machine) are selected as possible qualifications.

Generate the capacity. The capacity is also assumed to follow a normal distribution of mean $\overline{\boldsymbol{c}}$ and standard deviation $\sigma(\boldsymbol{c})$. A value is then generated with $\min(\boldsymbol{c}^+, \max(\overline{\boldsymbol{c}} + \sigma(\boldsymbol{c})\mathcal{G}(0.0, 1.0), \boldsymbol{c}^-))$. $\overline{\boldsymbol{c}}$ is first determined from the demand and throughput: $\overline{\boldsymbol{c}} = \frac{\sum_r (d_r \times \min_m(a_{r,m}))}{90\% \times M}$ assuming that the initial mean capacity utilization rate is equal to 90%. In general, generating the capacity this way does not ensure that the initial mean utilization rate is equal to 90% due to the randomness of the instance generation procedure. Let us define $\boldsymbol{c}^0$ as the value of the capacities generated so far. To ensure that the mean utilization rate is equal to 90%, Algorithm 1 is run on the generated instance where the capacity is $\boldsymbol{c}^0$. The real utilization rate of machine m is then equal to c_m^1, where c_m^1 is the optimized utilization rate of machine m computed with Algorithm 1. The final capacity value of each machine m is then defined as $\frac{c_m^0}{\frac{90\% \times M}{\sum_m (c_m^1)}}$.

6.2. Main findings

Similarities between the industrial and randomly generated instances can be observed, in particular the fact that restricting the search space by using the dual prices is relevant both in terms of solution quality and computational time. However, small differences with the main findings of the results for the industrial instances can be also observed:

Table 6
Numerical results for a computational time limit of 30 s and the second qualification configuration. Cells in italic, respectively bold, indicate the smallest, respectively the largest, gain value by k and work center.

Work center	k	GH		LS		GHDP		LSDP		GRASP		IGH		B&B	
		Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)
A	1	11.6	30.0	11.6	30.0	15.4	0.9	15.4	0.9	15.4	30.0	*13.7*	0.2	**15.4**	1.0
	2	11.6	30.0	11.6	30.0	23.5	0.9	24.7	2.5	25.1	30.0	*15.3*	0.2	**25.1**	2.0
	3	11.6	30.0	11.6	30.0	30.7	1.5	31.7	5.0	32.1	30.0	*16.1*	0.2	**32.1**	3.8
	4	11.6	30.0	11.6	30.0	35.3	2.1	36.3	7.0	37.3	30.0	*16.4*	0.2	**37.3**	10.4
	5	11.6	30.0	11.6	30.0	38.9	4.4	40.2	8.9	41.2	30.0	*17.4*	0.2	**40.4**	19.1
	6	11.6	30.0	11.6	30.0	42.2	4.5	43.2	10.4	44.1	30.0	*18.8*	0.2	**44.2**	22.9
	7	11.6	30.0	11.6	30.0	44.5	5.5	46.1	12.2	47.0	30.0	*19.9*	0.2	**43.6**	27.2
	8	11.6	30.0	11.6	30.0	46.7	6.5	48.6	14.9	49.1	30.0	*20.2*	0.2	**40.3**	30.0
	40	11.6	30.0	11.6	30.0	60.4	24.8	60.6	30.0	**60.8**	30.0	43.1	0.3	43.1	30.0
	100	11.6	30.0	11.6	30.0	61.0	30.0	61.0	30.0	62.4	30.0	53.0	0.3	**54.6**	28.0
B	1	0.8	30.0	0.8	30.0	35.3	7.2	35.3	7.2	35.3	30.0	*32.3*	2.9	**35.2**	29.4
	2	0.8	30.0	0.8	30.0	44.5	7.1	44.8	12.7	45.4	30.0	34.6	2.7	**36.2**	30.0
	3	0.8	30.0	0.8	30.0	50.8	9.9	51.8	20.2	51.8	30.0	35.2	2.7	**35.2**	30.0
	4	0.8	30.0	0.8	30.0	55.7	12.4	**56.5**	25.7	56.7	30.0	35.3	2.8	35.3	30.0
	5	0.8	30.0	0.8	30.0	59.5	15.3	**61.3**	29.7	61.0	30.0	35.3	2.8	35.3	30.0
	6	0.8	30.0	0.8	30.0	63.4	17.8	**64.3**	30.0	64.2	30.0	35.3	2.9	35.3	30.0
	7	0.8	30.0	0.8	30.0	65.8	20.7	**66.7**	30.0	66.4	30.0	35.3	2.8	35.3	30.0
	8	0.8	30.0	0.8	30.0	68.3	23.1	**69.0**	30.0	68.4	30.0	35.3	2.9	35.3	30.0
	40	0.8	30.0	0.8	30.0	72.2	30.0	72.2	30.0	**77.4**	30.0	35.9	2.8	35.9	30.0
	100	0.8	30.0	0.8	30.0	72.0	30.0	72.0	30.0	**77.3**	30.0	37.1	3.0	37.1	30.0
C	1	7.2	30.0	7.2	30.0	34.5	1.8	34.5	1.8	34.5	30.0	*33.6*	0.8	**34.5**	25.2
	2	7.2	30.0	7.2	30.0	46.5	2.1	46.5	6.1	46.5	30.0	*34.6*	0.8	**47.7**	27.5
	3	7.2	30.0	7.2	30.0	53.6	3.0	54.0	7.9	**54.0**	30.0	34.8	0.8	54.5	30.0
	4	7.2	30.0	7.2	30.0	59.2	5.6	59.4	10.2	**59.6**	30.0	35.0	0.8	49.7	30.0
	5	7.2	30.0	7.2	30.0	62.8	6.7	62.9	13.7	**62.9**	30.0	35.1	0.8	38.8	30.0
	6	7.2	30.0	7.2	30.0	65.6	7.9	65.7	19.2	**65.8**	30.0	35.1	0.8	35.1	30.0
	7	7.2	30.0	7.2	30.0	67.9	8.8	68.1	21.5	**68.1**	30.0	35.1	0.8	35.1	30.0
	8	7.2	30.0	7.2	30.0	69.6	9.5	70.0	28.4	**69.8**	30.0	35.1	0.8	35.1	30.0
	40	7.2	30.0	7.2	30.0	77.5	30.0	77.5	30.0	**82.2**	30.0	37.8	0.8	37.8	30.0
	100	7.2	30.0	7.2	30.0	77.5	30.0	77.5	30.0	**83.6**	30.0	53.0	0.9	53.0	30.0
D	1	32.6	22.7	32.6	22.7	32.4	0.8	32.4	0.8	32.4	30.0	*24.3*	0.1	**32.6**	0.7
	2	44.0	23.0	44.0	29.7	42.9	0.8	43.0	2.1	43.6	30.0	*33.7*	0.1	**44.2**	1.0
	3	48.0	29.1	48.0	30.0	49.5	1.6	50.0	5.2	50.1	30.0	*34.5*	0.1	**50.2**	1.5
	4	48.5	29.9	48.5	30.0	52.4	2.1	53.3	7.2	53.6	30.0	*35.1*	0.1	**53.9**	4.1
	5	49.0	30.0	49.0	30.0	54.8	2.8	55.3	9.9	56.1	30.0	*35.2*	0.1	**56.4**	11.4
	6	49.0	30.0	49.0	30.0	56.6	4.5	57.7	17.0	57.9	30.0	*35.8*	0.1	**58.3**	17.9
	7	49.0	30.0	49.0	30.0	58.2	5.2	59.2	22.4	59.4	30.0	*38.7*	0.1	**59.8**	21.5
	8	49.0	30.0	49.0	30.0	59.2	6.2	60.3	25.5	60.5	30.0	*39.3*	0.1	**58.7**	25.3
	40	49.0	30.0	49.0	30.0	64.2	30.0	64.2	30.0	66.7	30.0	55.3	0.2	**55.3**	30.0
	100	49.0	30.0	49.0	30.0	64.3	30.0	64.3	30.0	67.6	30.0	62.2	0.2	**67.2**	12.0

1. For the first qualification configuration, using GHDP does not seem to be the best policy. LSDP and GRASP largely outperform GHDP even for $k = 1$,
2. For the first qualification configuration, LSDP, GRASP and B&B outperform other solution approaches for both computational time limits,
3. LSDP is more successful to improve solutions determined by GHDP than in industrial instances,
4. For the first qualification configuration, in particular small values of k, LSDP and GRASP provide near optimal solutions and perform slightly better than B&B for large values of k,
5. For the second qualification configuration, when $k < 4$, B&B is the best solution approach,
6. For the second qualification configuration, LSDP and GRASP provide fewer near optimal solutions but remain the best solution approaches when $k > 4$, Increasing N could help determine better solutions but at the cost of an increased computational time.

6.3. Detailed numerical results

6.3.1. First qualification configuration

For each value of k, Table 9, respectively Table 10, shows the numerical results for a computational time limit of 30 s, respectively for a computational time limit of 180 s. Table 11 provides details on the Branch and Bound algorithm for the first qualification configuration such as the initial relaxation gap at the root node, the final relaxation gap when the algorithm stops, the total number of explored nodes and the number of instances where the optimal solution is found.

Results on random instances are similar to the results obtained for the industrial instances but small differences can be observed.

For a computational time limit of 30 s, GH and GHDP perform similarly when k is not too large, typically when $k \leq 8$. However, GHDP is often much faster to reach the same quality of solutions as GH. When $k \leq 8$, GHDP is between 2 and 5 times faster than GH. For a computational time limit of 180 s, GHDP performs slightly better than GH and is between 1.5 and 20 times faster than GH. LS is always outperformed by LSDP for all values of k and LSDP always finished before LS. IGH is also outperformed by GHDP, LSDP and GRASP.

For a computational time limit of 30 s, B&B determines all optimal solutions when $k = 1$. For a computational time limit of 180 s, B&B determines all optimal solutions when $k \leq 4$ and, when $k > 4$, determines an optimal solution for more than 89% of the instances.

It is interesting to note that LSDP and GRASP provide near optimal solutions when $k < 4$ as the differences between LSDP, GRASP and B&B are almost unnoticeable in terms of gain, less than 0.1%. B&B finds solutions faster than LSDP and GRASP. When B&B does not provide optimal solutions, GRASP and LSDP often perform a bit better than B&B. GRASP performs better than B&B when $k \in \{5, 6, 40\}$, and LSDP performs better than B&B when $k \in \{5, 6\}$.

Table 7
Numerical results for a computational time limit of 180 s and the second qualification configuration. Cells in italic, respectively bold, indicate the smallest, respectively the largest, gain value by k and work center.

Work center	k	GH		LS		GHDP		LSDP		GRASP		IGH		B&B	
		Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)
A	1	15.0	180.0	15.0	180.0	15.4	0.9	15.4	0.9	15.4	180.0	*13.7*	0.2	**15.4**	1.0
	2	15.0	180.0	15.0	180.0	23.5	0.9	24.7	2.5	25.1	180.0	*15.3*	0.2	**25.1**	2.0
	3	15.0	180.0	15.0	180.0	30.7	1.5	31.7	5.0	32.1	180.0	*16.1*	0.2	**32.1**	3.8
	4	15.0	180.0	15.0	180.0	35.3	2.1	36.3	7.0	37.3	180.0	*16.4*	0.2	**37.3**	18.6
	5	15.0	180.0	15.0	180.0	38.9	4.4	40.2	8.9	41.4	180.0	*17.4*	0.2	**41.4**	38.5
	6	15.0	180.0	15.0	180.0	42.2	4.5	43.2	10.4	44.6	180.0	*18.8*	0.2	**44.7**	54.6
	7	15.0	180.0	15.0	180.0	44.5	5.5	46.1	12.2	47.2	180.0	*19.9*	0.2	**47.3**	91.2
	8	15.0	180.0	15.0	180.0	46.7	6.5	48.6	14.9	49.3	180.0	*20.2*	0.2	**48.9**	153.4
	40	15.0	180.0	15.0	180.0	60.4	24.8	61.0	93.4	**61.0**	180.0	43.1	0.3	43.1	180.0
	100	15.0	180.0	15.0	180.0	62.4	59.9	62.5	164.9	62.5	180.0	53.0	0.3	**54.6**	146.8
B	1	3.1	180.0	3.1	180.0	35.3	7.2	35.3	7.2	35.3	180.0	*32.3*	2.9	**36.0**	166.1
	2	3.1	180.0	3.1	180.0	44.5	7.1	44.8	12.7	45.6	180.0	34.6	2.7	**46.5**	180.0
	3	3.1	180.0	3.1	180.0	50.8	9.9	51.8	20.2	52.6	180.0	35.2	2.7	**53.4**	180.0
	4	3.1	180.0	3.1	180.0	55.7	12.4	**56.5**	26.7	57.6	180.0	35.3	2.8	52.4	180.0
	5	3.1	180.0	3.1	180.0	59.5	15.3	**61.5**	37.4	61.9	180.0	35.3	2.8	35.3	180.0
	6	3.1	180.0	3.1	180.0	63.4	17.8	**64.5**	45.0	64.8	180.0	35.3	2.9	35.3	180.0
	7	3.1	180.0	3.1	180.0	65.8	20.7	**67.0**	57.9	67.5	180.0	35.3	2.8	35.3	180.0
	8	3.1	180.0	3.1	180.0	68.3	23.1	**69.3**	60.5	70.1	180.0	35.3	2.9	35.3	180.0
	40	3.1	180.0	3.1	180.0	88.1	117.4	88.7	180.0	**88.8**	180.0	35.9	2.8	35.9	180.0
	100	3.1	180.0	3.1	180.0	90.3	180.0	90.3	180.0	**91.4**	180.0	37.1	3.0	37.1	180.0
C	1	8.8	180.0	8.8	180.0	34.5	1.8	34.5	1.8	34.5	180.0	*33.6*	0.8	**34.6**	76.0
	2	8.8	180.0	8.8	180.0	46.5	2.1	46.5	6.1	46.5	180.0	*34.6*	0.8	**47.8**	148.7
	3	8.8	180.0	8.8	180.0	53.6	3.0	54.0	7.9	**54.0**	180.0	34.8	0.8	54.7	180.0
	4	8.8	180.0	8.8	180.0	59.2	5.6	59.4	10.2	**59.6**	180.0	35.0	0.8	56.8	180.0
	5	8.8	180.0	8.8	180.0	62.8	6.7	62.9	14.0	**63.1**	180.0	35.1	0.8	45.4	180.0
	6	8.8	180.0	8.8	180.0	65.6	7.9	65.7	21.6	**66.1**	180.0	35.1	0.8	37.5	180.0
	7	8.8	180.0	8.8	180.0	67.9	8.8	68.1	30.4	**68.4**	180.0	35.1	0.8	36.5	180.0
	8	8.8	180.0	8.8	180.0	69.6	9.5	70.0	39.8	**70.3**	180.0	35.1	0.8	35.1	180.0
	40	8.8	180.0	8.8	180.0	82.3	117.5	82.6	180.0	**82.6**	180.0	37.8	0.8	37.8	180.0
	100	8.8	180.0	8.8	180.0	83.4	180.0	83.4	180.0	**84.7**	180.0	53.0	0.9	53.0	180.0
D	1	32.6	23.6	32.6	23.6	32.4	0.8	32.4	0.8	32.4	180.0	*24.3*	0.1	**32.6**	0.7
	2	44.2	24.3	44.2	58.2	42.9	0.8	43.0	2.1	43.6	180.0	*33.7*	0.1	**44.2**	1.0
	3	50.1	39.1	50.2	92.5	49.5	1.6	50.0	5.2	50.1	180.0	*34.5*	0.1	**50.2**	1.5
	4	53.7	58.0	53.7	119.6	52.4	2.1	53.3	7.2	53.7	180.0	*35.1*	0.1	**53.9**	4.1
	5	56.2	64.9	56.3	142.5	54.8	2.8	55.3	9.9	56.2	180.0	*35.2*	0.1	**56.4**	20.4
	6	58.0	83.8	58.1	161.3	56.6	4.5	57.7	19.1	58.1	180.0	*35.8*	0.1	**58.3**	56.5
	7	59.5	98.8	59.6	176.0	58.2	5.2	59.2	27.5	59.6	180.0	*38.7*	0.1	**59.8**	101.9
	8	60.5	115.5	60.6	177.5	59.2	6.2	60.3	37.7	60.7	180.0	*39.3*	0.1	**60.9**	117.9
	40	63.5	180.0	63.5	180.0	66.6	95.4	66.8	180.0	66.8	180.0	55.3	0.2	**55.9**	176.0
	100	63.5	180.0	63.5	180.0	67.3	180.0	67.3	180.0	67.6	180.0	62.2	0.2	**67.6**	18.2

Note also that, contrary to the industrial instances, LSDP (and GRASP) can significantly improve GHDP. When the differences between GHDP and LSDP is of about 0.1% in most cases, LSDP can improve solutions determined by GHDP by more than 3%.

6.3.2. Second qualification configuration

For each value of k, Table 12, respectively Table 13, shows the numerical results for a computational time limit of 30 s, respectively for a computational time limit of 180 s. Table 14 provides details on the Branch and Bound algorithm for the second qualification configuration such as the initial relaxation gap at the root node, the final relaxation gap when the algorithm stops, the total number of explored nodes and the number of instances where the optimal solution is found.

Similarly to what can be observed for the industrial instances, GH and LS always propose unsatisfactory qualification plans, whether the computational time limit is 30 or 180 s, even for $k = 1$, and are always outperformed by GHDP, LSDP, GRASP and B&B. Similarly to the industrial instances, for the second qualification configuration, there are tens of thousands of qualifications to evaluate for a single iteration of GH. This cannot be done in a few seconds. The use of dual variables is therefore particularly relevant to restrict the search space.

B&B is the best solution approaches as long as $k < 4$, although not all solutions are optimal. For instance, consider $k = 1$. For the first qualification configuration, B&B determines an optimal solution for all the instances. For the second qualification configuration, only 83 optimal solutions are determined for 96 instances. The number of optimal solutions quickly drops as k increases.

Contrary to the first qualification configuration, LSDP and GRASP do not always provide near optimal solutions when $k < 4$. The difference in gains between GRASP and B&B varies between 1% and 2%. The fact that LSDP and GRASP no longer provide as many near optimal solutions can be explained by N, which is the parameter driving the number of qualifications to evaluate at each iteration. A similar explanation can be given for the first qualification matrix when $k = 1$, as LSDP improves solutions found by GHDP. In this case, increasing N may lead to better solutions. When $k > 4$, LSDP and GRASP both outperform B&B.

LSDP and GRASP still provide much better solutions than GH, LS, GHDP and IGH. First, let us compare GHDP with GH and LS. The solutions of GHDP are between 1.5 and 5 times better and are determined much faster than the solutions determined by GH and LS. Contrary to the first qualification configuration, LSDP only slightly improves the quality of the qualification plans. For all values of k and both computational time limits, GRASP provides better results on average in terms of gain than LSDP.

7. Conclusions and perspectives

In this paper, we propose new solution approaches to determine optimized qualification plans in work centers with non-identical parallel machines to maximize the capacitated time flexibility measure

Table 8
Details of the branch and bound solution approach for the second qualification configuration.

Work center	k	30 seconds				180 seconds			
		Initial Gap (%)	Final Gap (%)	Nodes	Number of optimal solutions	Initial Gap (%)	Final Gap (%)	Nodes	Number of optimal solutions
A	1	2.04	0.09	1.8	24	2.04	0.09	1.8	24
	2	13.78	0.03	8.1	24	13.78	0.03	8.1	24
	3	25.15	0.01	21.5	24	25.15	0.01	21.5	24
	4	35.62	0.04	82.8	21	35.62	0.03	158.4	23
	5	44.27	1.76	182.5	16	44.27	0.01	368.0	22
	6	51.13	0.69	257.3	11	51.13	0.01	537.6	22
	7	57.49	7.15	343.9	5	57.49	0.02	925.0	20
	8	64.39	19.21	406.9	0	64.39	0.67	1827.6	10
	40	60.79	60.79	423.8	0	60.79	60.79	2762.5	0
	100	32.32	28.31	396.1	5	32.32	28.31	2205.3	5
B	1	10.95	4.72	1.6	1	10.95	3.38	12.5	3
	2	30.71	27.42	1.5	0	30.71	5.36	23.4	0
	3	51.04	51.04	1.3	0	51.04	6.25	21.1	0
	4	71.36	71.36	1.2	0	71.36	23.19	19.8	0
	5	91.01	91.01	1.1	0	91.01	91.00	17.5	0
	6	111.19	111.19	1.0	0	111.19	111.19	16.5	0
	7	131.72	131.72	1.0	0	131.72	131.72	16.0	0
	8	152.43	152.43	1.0	0	152.43	152.43	15.2	0
	40	696.11	696.11	1.0	0	696.11	696.11	11.5	0
	100	890.51	890.51	1.0	0	890.51	890.50	16.6	0
C	1	7.42	5.13	13.9	6	7.42	0.75	45.0	21
	2	40.85	5.00	33.7	3	40.85	3.18	230.8	6
	3	71.12	7.38	38.4	0	71.12	5.65	284.5	0
	4	98.69	52.48	38.5	0	98.69	27.31	284.1	0
	5	124.84	116.86	37.5	0	124.84	90.17	287.0	0
	6	149.89	149.77	35.8	0	149.89	143.86	279.8	0
	7	172.98	172.90	34.0	0	172.98	169.77	272.7	0
	8	194.12	194.08	32.8	0	194.12	193.94	263.1	0
	40	406.03	406.02	33.5	0	406.03	406.02	259.0	0
	100	297.11	297.11	38.0	0	297.11	297.11	289.7	0
D	1	23.90	0.30	3.0	24	23.90	0.30	3.0	24
	2	30.69	0.23	6.5	24	30.69	0.23	6.5	24
	3	47.25	0.03	15.8	24	47.25	0.03	15.8	24
	4	60.59	0.03	78.6	24	60.59	0.03	78.6	24
	5	72.96	0.15	272.8	20	72.96	0.00	433.1	24
	6	83.21	0.45	508.2	14	83.21	0.07	1285.8	19
	7	78.77	0.73	691.6	9	78.77	0.24	2338.3	13
	8	85.50	5.18	873.3	7	85.50	0.40	3071.7	10
	40	49.10	49.10	1262.4	0	49.10	48.07	7329.0	1
	100	20.38	0.97	462.8	23	20.38	0.00	739.2	23

Table 9
Numerical results for a computational time limit of 30 s and the first qualification configuration. Cells in italic, respectively bold, indicate the smallest, respectively the largest, gain value by k.

k	GH		LS		GHDP		LSDP		GRASP		IGH		B&B	
	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)
1	*14.9*	15.5	*14.9*	15.5	*14.9*	3.5	16.4	3.5	16.4	30.0	15.9	1.1	**16.4**	2.0
2	*18.6*	15.5	*18.6*	18.3	*18.6*	3.5	21.6	6.9	21.6	30.0	18.3	1.1	**21.6**	3.0
3	*20.7*	16.9	20.7	20.3	*20.7*	5.6	24.0	10.1	24.1	30.0	20.4	1.1	**24.1**	4.7
4	*21.7*	18.5	21.7	21.7	*21.7*	6.7	25.6	12.0	25.6	30.0	21.9	1.1	**25.6**	5.5
5	*22.5*	19.7	22.5	23.1	*22.5*	8.6	26.5	13.9	**26.6**	30.0	23.1	1.1	26.1	6.7
6	*22.8*	20.2	22.9	24.1	*22.8*	10.0	27.3	15.8	**27.3**	30.0	24.1	1.1	26.9	6.9
7	*23.2*	21.1	23.2	25.4	*23.2*	10.9	27.8	17.5	27.8	30.0	24.7	1.1	**27.8**	7.6
8	*23.5*	21.7	23.5	26.6	*23.5*	11.9	28.3	20.0	28.3	30.0	25.3	1.1	**28.3**	7.7
40	*24.5*	29.5	*24.5*	30.0	*24.5*	27.2	30.9	29.7	**31.6**	30.0	30.8	1.1	31.4	11.3
100	*24.7*	29.9	*24.7*	30.0	*24.7*	28.2	31.0	29.7	32.1	30.0	32.1	1.2	**32.2**	10.4

proposed in Rowshannahad et al. (2015). In particular, dual prices are used to derive heuristics that are quickly guided towards good solutions. The proposed approaches are compared on industrial data on four different work centers, covering a significant number of machines in the considered semiconductor manufacturing facility, and two different qualification configurations. The proposed approaches are also compared on instances randomly generated using parameters taken from the industrial data. The approaches relying on dual variable provide very good solutions. Because the four work centers are of different nature, we expect the approaches to be effective on the remaining work centers in the production facility. Recommendations are finally provided. The approaches are now embedded in a decision support system that determines and proposes effective qualification plans to work center managers twenty minutes before every shift (every 8 h). The decision support is used to enhance their decision process and better manage work centers.

We believe the following perspectives are worth investigating in the future:

Table 10
Numerical results for a computational time limit of 180 s and the first qualification configuration. Cells in italic, respectively bold, indicate the smallest, respectively the largest, gain value by k.

k	GH		LS		GHDP		LSDP		GRASP		IGH		B&B	
	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)
1	*16.2*	71.7	*16.2*	71.7	16.4	3.5	16.4	3.5	16.4	180.0	15.9	1.1	**16.4**	2.0
2	*20.7*	72.9	20.7	83.6	21.6	3.5	21.6	7.5	21.6	180.0	18.3	1.1	**21.6**	3.2
3	*22.7*	77.5	22.7	89.0	24.0	5.7	24.1	12.0	24.1	180.0	20.4	1.1	**24.1**	5.6
4	*24.0*	83.8	24.0	96.6	25.5	7.4	25.6	15.5	25.6	180.0	21.9	1.1	**25.6**	8.5
5	*24.7*	86.5	24.8	99.1	26.5	9.8	26.6	19.4	26.6	180.0	23.1	1.1	**26.6**	12.3
6	*25.3*	88.8	25.4	102.0	27.3	11.9	27.3	24.3	27.3	180.0	24.1	1.1	**27.4**	13.8
7	*25.8*	91.9	25.8	106.1	27.8	13.5	27.9	29.3	27.9	180.0	24.7	1.1	**27.9**	16.9
8	*26.2*	95.2	26.2	109.2	28.3	15.3	28.3	36.2	28.4	180.0	25.3	1.1	**28.4**	20.4
40	*28.5*	136.3	28.5	158.0	31.8	80.2	31.8	124.8	**31.9**	180.0	30.8	1.1	31.4	59.9
100	*28.5*	158.1	28.5	167.8	32.4	109.0	32.4	133.1	**32.7**	180.0	32.1	1.2	32.3	49.1

Table 11
Details of the branch and bound solution approach for the first qualification configuration.

k	30 seconds				180 seconds			
	Initial Gap (%)	Final Gap (%)	Nodes	Number of optimal solutions	Initial Gap (%)	Final Gap (%)	Nodes	Number of of optimal solutions
1	3.06	0.67	1.0	96	3.06	0.67	1.0	96
2	12.75	0.04	2.7	95	12.75	0.04	2.8	96
3	19.75	0.04	6.7	91	19.75	0.01	7.4	96
4	25.65	0.07	10.2	89	25.65	0.01	13.0	96
5	38.33	9.44	18.1	86	38.33	0.06	36.5	95
6	48.56	10.07	18.8	82	48.56	0.00	27.8	94
7	58.39	0.25	24.4	82	58.39	0.01	46.1	93
8	41.63	0.23	29.6	78	41.63	0.10	59.1	92
40	6.80	5.20	129.2	64	6.80	5.06	598.8	67
100	3.44	2.76	117.3	69	3.44	2.62	399.3	73

Table 12
Numerical results for a computational time limit of 30 s and the second qualification configuration. Cells in italic, respectively bold, indicate the smallest, respectively the largest, gain value by k.

k	GH		LS		GHDP		LSDP		GRASP		IGH		B&B	
	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)
1	*12.9*	30.0	*12.9*	30.0	20.2	4.3	20.2	4.3	20.2	30.0	20.0	1.2	**21.3**	9.8
2	*12.9*	30.0	*12.9*	30.0	28.1	4.3	28.2	7.7	28.3	30.0	20.2	1.2	**30.5**	13.5
3	*12.9*	30.0	*12.9*	30.0	33.1	6.0	33.5	11.3	33.6	30.0	20.2	1.2	**35.3**	15.1
4	*12.9*	30.0	*12.9*	30.0	35.9	7.6	36.4	13.6	**36.8**	30.0	20.3	1.2	36.6	17.0
5	*12.9*	30.0	*12.9*	30.0	38.3	9.1	38.7	15.1	**39.1**	30.0	20.6	1.2	35.8	18.0
6	*12.9*	30.0	*12.9*	30.0	40.0	10.6	40.7	17.9	**41.0**	30.0	20.8	1.2	35.3	18.4
7	*12.9*	30.0	*12.9*	30.0	41.3	11.6	42.3	20.9	**42.7**	30.0	21.0	1.2	34.2	19.0
8	*12.9*	30.0	*12.9*	30.0	42.7	12.6	43.5	22.9	**43.9**	30.0	21.4	1.2	33.4	19.6
40	*12.9*	30.0	*12.9*	30.0	53.1	28.4	53.2	30.0	**56.2**	30.0	31.1	1.3	39.5	23.8
100	*12.9*	30.0	*12.9*	30.0	53.5	30.0	53.5	30.0	**60.3**	30.0	42.0	1.3	44.1	27.0

Table 13
Numerical results for a computational time limit of 180 s and the second qualification configuration. Cells in italic, respectively bold, indicate the smallest, respectively the largest, gain value by k.

k	GH		LS		GHDP		LSDP		GRASP		IGH		B&B	
	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)	Gain (%)	CPU (s)
1	*14.2*	161.5	*14.2*	161.5	20.2	4.3	20.2	4.3	20.2	180.0	20.0	1.2	**21.4**	35.6
2	*17.0*	162.1	17.0	174.4	28.1	4.3	28.2	8.3	28.3	180.0	20.2	1.2	**30.6**	50.1
3	*18.0*	170.3	18.0	180.0	33.1	6.2	33.5	13.4	33.7	180.0	20.2	1.2	**35.6**	61.9
4	*18.4*	175.6	18.4	180.0	36.0	8.2	36.4	17.4	36.9	180.0	20.3	1.2	**38.0**	76.4
5	*18.1*	178.7	18.1	180.0	38.3	10.4	38.8	21.9	**39.3**	180.0	20.6	1.2	37.1	81.5
6	*18.5*	179.7	*18.5*	180.0	40.1	12.6	40.8	29.3	**41.3**	180.0	20.8	1.2	37.0	88.2
7	*18.9*	180.0	*18.9*	180.0	41.4	14.3	42.3	37.4	**42.9**	180.0	21.0	1.2	36.3	93.2
8	*18.9*	180.0	*18.9*	180.0	42.8	16.2	43.7	46.0	**44.3**	180.0	21.4	1.2	37.0	97.6
40	*18.9*	180.0	*18.9*	180.0	56.6	87.4	57.0	152.7	**57.4**	180.0	31.1	1.3	43.5	120.8
100	*18.9*	180.0	*18.9*	180.0	60.8	138.5	61.0	179.5	**62.6**	180.0	42.0	1.3	46.5	145.7

Table 14
Details of the branch and bound solution approach for the second qualification configuration.

k	30 seconds				180 seconds			
	Initial Gap (%)	Final Gap (%)	Nodes	Number of optimal solutions	Initial Gap (%)	Final Gap (%)	Nodes	Number of optimal solutions
1	8.74%	5.16%	7.8	74	8.74%	5.00%	14.7	83
2	46.71%	5.47%	41.6	63	46.71%	4.98%	85.0	76
3	93.52%	5.14%	53.5	57	93.52%	4.18%	184.5	70
4	144.62%	44.07%	81.6	52	144.62%	31.43%	273.5	63
5	205.70%	102.45%	85.3	49	205.70%	90.50%	345.0	58
6	288.74%	175.31%	90.5	42	288.74%	157.57%	403.1	56
7	401.80%	283.44%	97.0	41	401.80%	264.68%	443.2	51
8	549.76%	443.59%	102.7	39	549.76%	403.78%	481.3	49
40	29727.94%	29343.32%	142.1	28	29727.94%	28920.18%	598.2	37
100	130259.70%	129512.62%	151.4	14	130259.70%	128722.32%	781.3	23

Table A.15
Instances for work center A: $\overline{\boldsymbol{d}}$ is the mean demand of products, $\sigma(\boldsymbol{d})$ the standard deviation of the demand of products, $\boldsymbol{d}^+$ the maximum demand of products, $\boldsymbol{d}^-$ the minimum demand over all products, $\overline{\boldsymbol{c}}$ the mean production capacity of machines, $\sigma(\boldsymbol{c})$ the standard deviation of the production capacity of machines, $\boldsymbol{c}^+$ the maximum capacity over all machines, $\boldsymbol{c}^-$ the minimum capacity over all machines, $\overline{\boldsymbol{a}}$ the mean throughput of products on machines of initial and possible qualifications, $\boldsymbol{a}^+$ the maximum throughput of products on machines of initial and possible qualifications, and $\boldsymbol{a}^-$ the minimum throughput of products on machines of initial and possible qualifications, and $\sigma(\boldsymbol{a})$ the standard deviation of the throughput of products on machines of initial and possible qualifications. The initial qualification rate (%) is the number of entries equal to 1 in the matrix $\boldsymbol{q}$ divided by $R \times M$, and the qualifiable rate (%) is the number of entries equal to 2 in the matrix $\boldsymbol{q}$ divided by $R \times M$.

Instance	R	M	$\frac{d^-}{\overline{d}}$	$\frac{d^+}{\overline{d}}$	$\frac{\sigma(d)}{\overline{d}}$	$\frac{a^-}{\overline{a}}$	$\frac{a^+}{\overline{a}}$	$\frac{\sigma(a)}{\overline{a}}$	$\frac{c^-}{\overline{c}}$	$\frac{c^+}{\overline{c}}$	$\frac{\sigma(c)}{\overline{c}}$	Initial qualification rate (%)	Possible qualification rate (%)
1	660	14	0.016	8.049	1.015	0.055	8.042	0.574	0.838	1.167	0.109	25.130%	5.400%
2	650	14	0.015	8.039	1.062	0.055	5.308	0.561	0.836	1.208	0.106	25.000%	5.396%
3	667	15	0.015	7.493	1.033	0.045	5.547	0.576	0.808	1.167	0.121	24.048%	4.018%
4	642	14	0.014	7.457	1.092	0.045	5.528	0.570	0.815	1.156	0.097	25.567%	4.105%
5	548	15	0.015	6.328	0.999	0.046	5.157	0.562	0.803	1.160	0.120	24.684%	5.876%
6	538	15	0.014	6.016	1.008	0.045	5.076	0.575	0.828	1.196	0.107	24.845%	5.799%
7	532	14	0.014	6.330	1.029	0.090	5.035	0.553	0.803	1.160	0.118	26.705%	6.002%
8	542	14	0.016	6.243	1.005	0.093	2.493	0.549	0.792	1.145	0.113	28.268%	4.942%
9	569	14	0.016	6.231	1.006	0.059	2.466	0.556	0.824	1.190	0.117	26.048%	5.938%
10	565	14	0.016	6.612	0.999	0.059	2.456	0.558	0.797	1.151	0.113	27.155%	5.815%
11	563	14	0.032	8.416	1.041	0.077	2.423	0.558	0.803	1.160	0.123	27.138%	6.141%
12	585	14	0.033	7.357	1.019	0.057	2.419	0.555	0.824	1.190	0.120	26.984%	6.105%
13	578	14	0.015	7.365	0.950	0.058	2.461	0.569	0.817	1.181	0.120	27.533%	5.252%
14	602	14	0.015	7.631	1.002	0.045	2.411	0.564	0.806	1.164	0.119	28.049%	4.461%
15	590	15	0.016	10.089	1.071	0.044	2.402	0.549	0.805	1.163	0.119	25.763%	4.226%
16	578	14	0.016	10.113	1.085	0.069	6.426	0.563	0.818	1.182	0.119	27.558%	5.042%
17	632	14	0.016	8.525	1.113	0.044	6.384	0.548	0.807	1.165	0.102	27.238%	5.120%
18	631	14	0.015	8.107	1.161	0.070	6.455	0.552	0.821	1.186	0.120	27.383%	5.173%
19	604	14	0.014	7.984	1.175	0.079	2.484	0.557	0.808	1.146	0.109	26.703%	6.055%
20	582	14	0.013	7.074	1.192	0.077	2.446	0.558	0.796	1.149	0.113	26.338%	6.112%
21	558	15	0.015	7.634	1.133	0.079	2.501	0.566	0.792	1.144	0.107	24.886%	5.783%
22	564	14	0.015	8.133	1.052	0.079	2.503	0.561	0.794	1.147	0.111	26.722%	5.990%
23	588	14	0.013	6.242	1.096	0.071	2.458	0.552	0.826	1.172	0.108	25.948%	6.353%
24	601	15	0.013	7.395	1.203	0.071	2.447	0.555	0.820	1.143	0.099	24.326%	5.768%

- Some parameters might be subject to uncertainty, such as product quantities and machines capacities, and designing robust qualification plans should be an interesting research avenue,
- Workload variables are continuous but, in practice, some machines run product quantities by batches. Hence, the consideration of batching constraints could be explored as in Rowshannahad and Dauzère-Pérès (2013),
- An outer linearization algorithm is used to solve nonlinear programs. Other algorithms, such as active-set methods or sequential quadratic methods (Rowshannahad et al., 2015) could be compared to the outer linearization algorithm to further reduce computational times,
- Solution approaches could be compared on data from other factories to further validate the relevance of the dual variable solution approaches,
- We assumed in this work that each qualification has the same cost, which makes sense at the operational level. However, considering different qualification costs when decisions are taken for the next weeks or months could be relevant,
- Studying the effect of disqualifications on the compromise between qualification costs and utilization balance can also be relevant,
- Considering time-varying demand and production capacity on a longer planning horizon is interesting, but makes sense at a different decision level than the one considered in this paper,
- It would be relevant to study the robustness of solution approaches, *e.g.* under what conditions using dual prices does not provide good solutions.

CRediT authorship contribution statement

Antoine Perraudat: Investigation, Conceptualization, Methodology, Formal analysis, Software, Writing – review & editing. **Stéphane Dauzère-Pérès:** Investigation, Conceptualization, Methodology, Supervision, Writing – review & editing, Funding acquisition. **Philippe Vialletelle:** Investigation, Conceptualization, Supervision, Funding acquisition.

Table B.16
Instances for work center A: $\overline{d}$ is the mean demand of products, $\sigma(d)$ the standard deviation of the demand of products, d^+ the maximum demand over all products, d^- the minimum demand over all products, $\overline{c}$ the mean production capacity of machines, $\sigma(c)$ the standard deviation of the production capacity of machines, c^+ the maximum capacity over all machines, c^- the minimum capacity over all machines, $\overline{a}$ the mean throughput of products on machines of initial and possible qualifications, a^+ the maximum throughput of products on machines of initial and possible qualifications, and a^- the minimum throughput of products on machines of initial and possible qualifications, and $\sigma(a)$ the standard deviation of the throughput of products on machines of initial and possible qualifications. The initial qualification rate (%) is the number of entries equal to 1 in the matrix q divided by $R \times M$, and the qualifiable rate (%) is the number of entries equal to 2 in the matrix q divided by $R \times M$.

Instance	R	M	$\frac{d^-}{\overline{d}}$	$\frac{d^+}{\overline{d}}$	$\frac{\sigma(d)}{\overline{d}}$	$\frac{a^-}{\overline{a}}$	$\frac{a^+}{\overline{a}}$	$\frac{\sigma(a)}{\overline{a}}$	$\frac{c^-}{\overline{c}}$	$\frac{c^+}{\overline{c}}$	$\frac{\sigma(c)}{\overline{c}}$	Initial qualification rate (%)	Possible qualification rate (%)
1	786	168	0.015	10.051	1.300	0.076	45.635	1.423	0.890	1.049	0.037	3.010%	0.924%
2	760	168	0.014	9.937	1.264	0.078	46.882	1.452	0.840	1.053	0.039	3.022%	0.931%
3	806	168	0.014	13.364	1.263	0.072	43.423	1.809	0.891	1.050	0.037	3.052%	0.969%
4	785	168	0.014	12.204	1.282	0.073	43.802	1.715	0.839	1.052	0.039	3.044%	0.961%
5	767	168	0.014	9.993	1.233	0.078	91.173	1.993	0.841	1.054	0.040	2.894%	0.979%
6	767	168	0.014	10.024	1.185	0.079	91.885	2.000	0.841	1.054	0.040	2.930%	0.944%
7	780	168	0.014	9.622	1.128	0.079	92.418	2.001	0.840	1.053	0.040	2.898%	0.982%
8	809	168	0.015	10.299	1.159	0.079	47.644	1.661	0.840	1.053	0.040	2.860%	1.023%
9	821	168	0.015	7.667	1.102	0.078	46.954	1.643	0.892	1.051	0.038	2.888%	0.977%
10	814	169	0.015	8.696	1.120	0.078	46.622	1.639	0.838	1.050	0.038	2.758%	1.092%
11	781	168	0.015	7.303	1.173	0.081	48.617	1.692	0.838	1.050	0.039	2.895%	1.023%
12	793	168	0.016	6.987	1.168	0.081	48.671	1.696	0.839	1.051	0.040	2.866%	1.028%
13	793	168	0.015	9.071	1.180	0.079	47.235	1.574	0.839	1.051	0.038	2.797%	1.099%
14	783	168	0.015	10.039	1.197	0.074	44.531	1.760	0.839	1.051	0.039	2.842%	1.022%
15	798	168	0.015	8.690	1.211	0.079	47.491	1.588	0.839	1.052	0.039	2.846%	1.018%
16	790	168	0.015	7.908	1.198	0.078	46.642	1.582	0.837	1.049	0.038	2.884%	0.995%
17	795	168	0.015	9.697	1.167	0.077	46.349	1.564	0.839	1.052	0.038	2.880%	1.013%
18	794	169	0.015	12.039	1.213	0.077	46.596	1.569	0.889	1.048	0.036	2.820%	1.023%
19	791	168	0.016	19.706	1.395	0.075	45.164	1.533	0.839	1.051	0.040	2.823%	1.026%
20	767	168	0.014	14.196	1.425	0.075	45.252	1.540	0.838	1.051	0.040	2.875%	1.004%
21	752	169	0.015	15.569	1.459	0.083	50.205	1.500	0.839	1.051	0.040	2.882%	1.028%
22	753	168	0.015	11.377	1.249	0.078	90.358	1.916	0.839	1.052	0.038	2.892%	1.022%
23	779	168	0.016	12.244	1.297	0.080	48.302	1.604	0.891	1.050	0.037	2.854%	1.104%
24	762	168	0.015	14.442	1.327	0.077	46.596	1.577	0.838	1.050	0.039	2.843%	1.051%

Table C.17
Instances for work center A: $\overline{d}$ is the mean demand of products, $\sigma(d)$ the standard deviation of the demand of products, d^+ the maximum demand over all products, d^- the minimum demand over all products, $\overline{c}$ the mean production capacity of machines, $\sigma(c)$ the standard deviation of the production capacity of machines, c^+ the maximum capacity over all machines, c^- the minimum capacity over all machines, $\overline{a}$ the mean throughput of products on machines of initial and possible qualifications, a^+ the maximum throughput of products on machines of initial and possible qualifications, and a^- the minimum throughput of products on machines of initial and possible qualifications, and $\sigma(a)$ the standard deviation of the throughput of products on machines of initial and possible qualifications. The initial qualification rate (%) is the number of entries equal to 1 in the matrix q divided by $R \times M$, and the qualifiable rate (%) is the number of entries equal to 2 in the matrix q divided by $R \times M$.

Instance	R	M	$\frac{d^-}{\overline{d}}$	$\frac{d^+}{\overline{d}}$	$\frac{\sigma(d)}{\overline{d}}$	$\frac{a^-}{\overline{a}}$	$\frac{a^+}{\overline{a}}$	$\frac{\sigma(a)}{\overline{a}}$	$\frac{c^-}{\overline{c}}$	$\frac{c^+}{\overline{c}}$	$\frac{\sigma(c)}{\overline{c}}$	Initial qualification rate (%)	Possible qualification rate (%)
1	589	69	0.014	11.901	1.367	0.051	46.437	2.336	0.630	1.273	0.166	5.472%	1.821%
2	579	69	0.014	13.678	1.498	0.052	31.064	2.239	0.631	1.274	0.159	5.384%	1.867%
3	565	69	0.014	12.955	1.509	0.053	24.722	2.125	0.624	1.260	0.151	5.502%	1.857%
4	556	70	0.014	13.073	1.473	0.055	20.587	2.000	0.633	1.279	0.154	5.447%	1.775%
5	540	70	0.016	8.046	1.172	0.075	23.157	2.287	0.638	1.290	0.156	5.172%	1.849%
6	550	69	0.016	13.858	1.219	0.080	24.640	2.238	0.628	1.269	0.159	5.270%	1.900%
7	537	69	0.016	14.502	1.241	0.078	24.208	2.303	0.636	1.286	0.165	5.090%	1.924%
8	554	69	0.016	14.869	1.214	0.080	24.852	2.211	0.629	1.271	0.165	5.148%	1.870%
9	513	69	0.016	14.488	1.238	0.076	19.033	2.244	0.628	1.269	0.160	5.150%	1.865%
10	516	69	0.016	14.774	1.325	0.076	19.114	2.237	0.638	1.288	0.159	5.081%	2.005%
11	579	69	0.017	15.549	1.265	0.052	24.292	2.193	0.639	1.291	0.161	5.136%	1.997%
12	568	70	0.017	13.335	1.230	0.053	24.832	2.149	0.633	1.279	0.165	5.179%	1.901%
13	487	69	0.017	14.478	1.312	0.052	24.224	2.272	0.632	1.277	0.161	5.074%	1.976%
14	501	69	0.018	8.896	1.124	0.051	23.811	2.225	0.677	1.239	0.153	5.091%	1.970%
15	506	69	0.018	6.760	1.052	0.050	23.346	2.312	0.627	1.266	0.154	5.058%	1.988%
16	494	69	0.017	5.983	1.087	0.050	23.350	2.307	0.642	1.296	0.163	5.090%	2.004%
17	543	70	0.018	10.123	1.220	0.047	21.915	2.366	0.629	1.270	0.158	5.017%	1.978%
18	516	69	0.017	11.783	1.294	0.050	23.148	2.272	0.636	1.286	0.163	5.123%	1.938%
19	540	69	0.016	9.877	1.166	0.052	19.719	2.147	0.632	1.276	0.164	5.360%	1.688%
20	497	70	0.015	8.935	1.189	0.053	24.293	2.180	0.634	1.281	0.163	5.185%	1.771%
21	523	69	0.016	10.767	1.253	0.048	22.443	2.233	0.634	1.280	0.159	5.495%	1.649%
22	496	69	0.015	12.566	1.322	0.044	53.408	2.740	0.634	1.281	0.157	5.508%	1.622%
23	531	69	0.015	11.915	1.295	0.041	49.669	2.774	0.630	1.273	0.166	5.273%	1.850%
24	530	70	0.015	10.193	1.318	0.046	54.768	2.769	0.623	1.258	0.153	5.261%	1.771%

Acknowledgments

This work has been partly funded by the ANRT (Association Nationale de la Recherche et de la Technologie) through CIFRE funds (PhD number 2017/0750) and a cooperation contract between STMicroelectronics and Mines Saint-Etienne. This work has also received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737459 (project Productive4.0). This Joint Undertaking received support from the European

Table D.18
Instances for work center A: $\overline{d}$ is the mean demand of products, $\sigma(\boldsymbol{d})$ the standard deviation of the demand of products, $\boldsymbol{d}^{+}$ the maximum demand over all products, $\boldsymbol{d}^{-}$ the minimum demand over all products, $\overline{c}$ the mean production capacity of machines, $\sigma(\boldsymbol{c})$ the standard deviation of the production capacity of machines, $\boldsymbol{c}^{+}$ the maximum capacity over all machines, $\boldsymbol{c}^{-}$ the minimum capacity over all machines, $\overline{a}$ the mean throughput of products on machines of initial and possible qualifications, $\boldsymbol{a}^{+}$ the maximum throughput of products on machines of initial and possible qualifications, and $\boldsymbol{a}^{-}$ the minimum throughput of products on machines of initial and possible qualifications, and $\sigma(\boldsymbol{a})$ the standard deviation of the throughput of products on machines of initial and possible qualifications. The initial qualification rate (%) is the number of entries equal to 1 in the matrix $\boldsymbol{q}$ divided by $R \times M$, and the qualifiable rate (%) is the number of entries equal to 2 in the matrix $\boldsymbol{q}$ divided by $R \times M$.

Instance	R	M	$\frac{d^-}{\overline{d}}$	$\frac{d^+}{\overline{d}}$	$\frac{\sigma(d)}{\overline{d}}$	$\frac{a^-}{\overline{a}}$	$\frac{a^+}{\overline{a}}$	$\frac{\sigma(a)}{\overline{a}}$	$\frac{c^-}{\overline{c}}$	$\frac{c^+}{\overline{c}}$	$\frac{\sigma(c)}{\overline{c}}$	Initial qualification rate (%)	Possible qualification rate (%)
1	228	21	0.016	12.971	1.687	0.050	1.590	0.310	0.467	1.204	0.150	15.539%	1.838%
2	219	21	0.016	13.621	1.636	0.049	1.581	0.313	0.845	1.167	0.082	15.482%	1.957%
3	229	21	0.016	12.096	1.375	0.101	1.614	0.317	0.851	1.174	0.088	15.325%	1.539%
4	235	21	0.017	11.186	1.398	0.102	1.626	0.319	0.835	1.153	0.078	15.299%	1.479%
5	226	21	0.019	7.692	1.210	0.103	1.718	0.336	0.464	1.197	0.142	14.834%	1.622%
6	226	21	0.018	8.326	1.190	0.050	1.608	0.320	0.844	1.165	0.088	15.381%	1.686%
7	222	21	0.018	6.394	1.117	0.050	1.607	0.330	0.854	1.162	0.068	14.972%	1.780%
8	228	21	0.018	5.438	1.093	0.050	1.599	0.321	0.469	1.209	0.144	14.724%	2.109%
9	221	22	0.019	5.574	1.063	0.050	1.609	0.332	0.467	1.204	0.139	14.500%	1.666%
10	223	22	0.019	6.494	1.083	0.050	1.608	0.332	0.465	1.200	0.141	13.596%	2.059%
11	240	22	0.020	8.341	1.223	0.051	1.618	0.338	0.854	1.178	0.084	12.784%	2.917%
12	237	21	0.020	8.342	1.219	0.051	1.624	0.337	0.846	1.167	0.075	14.627%	1.869%
13	218	21	0.018	6.959	1.204	0.051	1.642	0.326	0.459	1.183	0.143	15.138%	1.573%
14	213	21	0.018	7.271	1.132	0.050	1.631	0.323	0.465	1.200	0.147	15.336%	1.543%
15	215	21	0.018	6.585	1.163	0.050	1.641	0.329	0.848	1.171	0.089	15.150%	1.550%
16	215	21	0.018	6.360	1.129	0.050	1.616	0.319	0.835	1.152	0.077	15.216%	1.639%
17	223	21	0.018	7.625	1.166	0.101	1.635	0.329	0.467	1.205	0.142	14.820%	1.708%
18	219	21	0.019	6.590	1.171	0.101	1.645	0.338	0.464	1.196	0.146	14.808%	1.805%
19	214	21	0.017	7.056	1.115	0.101	1.687	0.329	0.466	1.202	0.145	15.198%	1.602%
20	207	21	0.019	6.958	1.127	0.100	1.677	0.325	0.465	1.199	0.149	15.183%	1.610%
21	223	21	0.020	8.120	1.162	0.051	1.708	0.335	0.462	1.190	0.142	14.606%	1.815%
22	241	21	0.020	10.274	1.206	0.050	1.683	0.329	0.462	1.190	0.142	15.076%	1.680%
23	215	21	0.018	8.144	1.195	0.050	1.685	0.339	0.467	1.204	0.150	14.862%	1.949%
24	224	21	0.019	9.036	1.166	0.051	1.692	0.340	0.461	1.189	0.147	14.881%	1.786%

Unions Horizon 2020 research and innovation program and Germany, Austria, France, Czech Republic, Netherlands, Belgium, Spain, Greece, Sweden, Italy, Ireland, Poland, Hungary, Portugal, Denmark, Finland, Luxembourg, Norway, Turkey.

The authors would also like to thank the three anonymous referees and the editors whose help, remarks and suggestions improved the quality of the article.

Appendix A. Instances for work center A

See Table A.15.

Appendix B. Instances for work center B

See Table B.16.

Appendix C. Instances for work center C

See Table C.17.

Appendix D. Instances for work center D

See Table D.18.

References

Aubry, A., Rossi, A., Espinouse, M.-L., Jacomino, M., 2008. Minimizing setup costs for parallel multi-purpose machines under load-balancing constraint. European J. Oper. Res. 187 (3), 1115–1125.

Bazaraa, M.S., Sherali, H.D., Shetty, C.M., 2013. Nonlinear Programming: Theory and Algorithms. John Wiley & Sons.

Bixby, E.R., Fenelon, M., Gu, Z., Rothberg, E., Wunderling, R., 1999. MIP: Theory and practice—closing the gap. In: IFIP Conference on System Modeling and Optimization. Springer, pp. 19–49.

Chang, X., Dong, M., 2017. Stochastic programming for qualification management of parallel machines in semiconductor manufacturing. Comput. Oper. Res. 79, 49–59.

Christ, Q., Dauzère-Pérès, S., Lepelletier, G., 2019. An iterated min-max procedure for practical workload balancing on non-identical parallel machines in manufacturing systems. European J. Oper. Res. 279 (2), 419–428.

Daskin, M.S., 2011. Network and Discrete Location: Models, Algorithms, and Applications. John Wiley & Sons.

Feo, T.A., Resende, M.G., 1989. A probabilistic heuristic for a computationally difficult set covering problem. Oper. Res. Lett. 8 (2), 67–71.

Garey, M.R., Johnson, D.S., 1979. Computers and Intractability. Vol. 174, San Francisco: freeman.

Johnzén, C., 2009. Modeling and optimizing flexible capacity allocation in semiconductor manufacturing. (Ph.D. thesis). Ecole Nationale Supérieure des Mines de Saint-Etienne.

Johnzén, C., Dauzère-Pérès, S., Vialletelle, P., 2011. Flexibility measures for qualification management in wafer fabs. Prod. Plan. Control 22 (1), 81–90.

Johnzén, C., Dauzère-Pérès, S., Vialletelle, P., Yugma, C., 2007. Importance of qualification management for wafer fabs. In: 2007 IEEE/SEMI Advanced Semiconductor Manufacturing Conference. IEEE, pp. 166–169.

Johnzén, C., Dauzère-Pérès, S., Vialletelle, P., Yugma, C., 2009. Optimizing flexibility and equipment utilization through qualification management. In: 2009 IEEE/SEMI Advanced Semiconductor Manufacturing Conference. IEEE, pp. 137–140.

Kabak, K.E., Heavey, C., Corbett, V., Byrne, P., 2013. Impact of recipe restrictions on photolithography toolsets in anASIC fabrication environment. IEEE Trans. Semicond. Manuf. 26 (1), 53–68.

Knopp, S., Dauzère-Pérès, S., Yugma, C., 2017. A batch-oblivious approach for complex job-shop scheduling problems. European J. Oper. Res. 263 (1), 50–61.

Kopp, D., Mönch, L., Pabst, D., Stehli, M., 2016. An optimization model for qualification management in wafer fabs. In: Proceedings of the 2016 Winter Simulation Conference. IEEE Press, pp. 2610–2620.

Kopp, D., Mönch, L., Pabst, D., Stehli, M., 2018. Simulation-based performance assessment of tool requalification strategies in wafer fabs. In: 2018 IEEE 14th International Conference on Automation Science and Engineering. CASE, IEEE, pp. 638–641.

Leachman, R.C., Kang, J., Lin, V., 2002. SLIM: Short cycle time and low inventory in manufacturing at samsung electronics. Interfaces 32 (1), 61–77.

Liao, B., Yano, C.A., Esturi, S., 2017. Optimizing site qualification across the supply network at western digital. Interfaces 47 (4), 305–319.

Löhndorf, N., 2016. Java interface for the CLP solver, (Accessed in 2017). https://github.com/quantego/clp-java.

Lougee-Heimer, R., 2003. The common optimization interface for operations research: Promoting open-source software in the operations research community. IBM J. Res. Dev. 47 (1), 57–66.

Obeid, A., Dauzère-Pérès, S., Yugma, C., 2014. Scheduling job families on non-identical parallel machines with time constraints. Ann. Oper. Res. 213 (1), 221–234.

Perraudat, A., Dauzère-Pérès, S., Vialletelle, P., 2019. Evaluating the impact of dynamic qualification management in semiconductor manufacturing. In: 2019 Winter Simulation Conference. WSC, IEEE, pp. 2336–2347.

Pianne, A., Rivero, L., Dauzère-Pérès, S., Vialletelle, P., 2016. Ideal and potential flexibility measures for qualification management in semiconductor manufacturing. In: Winter Simulation Conference. WSC, 2016, IEEE, pp. 2621–2632.

Rowshannahad, M., Dauzère-Pérès, S., 2013. Qualification management with batch size constraint. In: 2013 Winter Simulations Conference. WSC, IEEE, pp. 3707–3718.

Rowshannahad, M., Dauzère-Pérès, S., Cassini, B., 2014. Qualification management to reduce workload variability in semiconductor manufacturing. In: Proceedings of the 2014 Winter Simulation Conference. IEEE Press, pp. 2434–2443.

Rowshannahad, M., Dauzère-Pérès, S., Cassini, B., 2015. Capacitated qualification management in semiconductor manufacturing. Omega 54, 50–59.

Yepes-Borrero, J.C., Perea, F., Ruiz, R., Villa, F., 2021. Bi-objective parallel machine scheduling with additional resources during setups. European J. Oper. Res. 292 (2), 443–455.