Ikonen, Teemu; Mostafaei, Hossein; Ye, Yixin; Bernal, David; Grossmann, Ignacio; Harjunkoski, Iiro

Large-scale selective maintenance optimization using bathtub-shaped failure rates

Published in:
Computers and Chemical Engineering

DOI:
10.1016/j.compchemeng.2020.106876

Published: 04/08/2020

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Ikonen, T., Mostafaei, H., Ye, Y., Bernal, D., Grossmann, I., & Harjunkoski, I. (2020). Large-scale selective maintenance optimization using bathtub-shaped failure rates. *Computers and Chemical Engineering* , *139*, Article 106876. https://doi.org/10.1016/j.compchemeng.2020.106876

Large-scale selective maintenance optimization using bathtub-shaped failure rates

Teemu J. Ikonen[a], Hossein Mostafaei[a], Yixin Ye[b], David E. Bernal[b], Ignacio E. Grossmann[b], Iiro Harjunkoski[a,c,*]

[a]*Aalto University, School of Chemical Engineering, Kemistintie 1, 02150 Espoo, Finland*
[b]*Carnegie Mellon University, Department of Chemical Engineering, Pittsburgh, PA 15213, US*
[c]*ABB Corporate Research, Wallstadter Str. 59, 68526 Ladenburg, Germany*

Abstract

Engineering systems are typically maintained during planned, or unplanned, downtimes in between operation periods. If the duration of the downtime or the budget of the maintenance is an active constraint, all desired maintenance actions cannot be conducted. Seeking of the optimal subset of maintenance actions is referred to as *selective maintenance optimization*. In this work, we link the statistical analysis of lifetime data into selective maintenance optimization, focusing on datasets with bathtub-shaped failure rates. We also propose two improvements to the efficiency of mixed integer non-linear programming (MINLP)-based selective maintenance optimization. The first is the preclusion of component replacements that, due to the infant mortality period of the component, reduce the reliability. The second is the convexification of two MINLP models, involving only replacement, or replacement and repair, actions. The improvements enable our MINLP-based methods to tackle large-scale selective maintenance optimization problems with up to 700 to 1000 system components.

Keywords: reliability, optimization, selective maintenance, component replacement, component repair, failure rate

1. Introduction

Industrial plants should ideally be robust and reliable in continuous operation. Unexpected component failures at the plant may cause costly disruptions to the operation. In order to avoid disruptions, the operators of the plant schedule major shutdowns, enabling maintenance operations to be conducted for the components (e.g., electrical drives, pumps, and fans) of the plant. As these shutdowns are expensive, both in terms of direct maintenance costs and lost production time, the maintenance operations that are performed during a shutdown should be carefully selected. Such decision-making is challenging because a modern industrial plant may consist of hundreds – or even thousands – of individual components with various levels of criticality.

Selective maintenance, first introduced by Rice et al. (1998), aims at finding the optimal subset of maintenance actions to be performed for a multicomponent system. In single-objective optimization, the objective is to maximize the reliability of the system for the next operation window, subject to maintenance duration and/or cost constraints, or vice versa. Alternatively, the reliability maximization and the maintenance duration and/or cost minimization can be considered as a multiobjective optimization problem, the solution of which yields a Pareto optimal set of solutions, representing the best trade-offs between the objectives. Selective maintenance has been applied to various fields, ranging from aircraft maintenance (in between flight operations) to maintenance shutdowns of large industrial plants. The connecting factor in

*Corresponding author.
Email address: iiro.harjunkoski@de.abb.com (I. Harjunkoski)

these applications is that the system has predefined operating windows, and maintenance actions can only be conducted in between the windows.

After the pioneering work by Rice et al. (1998), several improvements and extensions have been reported in the selective maintenance literature. Cassady et al. (2001a) extended the approach in two ways. First, they considered components with time-dependent failure rates by characterizing the component lifetimes using the Weibull distribution (Weibull, 1951). Second, they expanded the selection of maintenance actions to 1) the minimal repair of a failed component, 2) the replacement of a failed component and 3) the replacement of a functioning component[1]. Cassady et al. (2001b) extended the problem definition to permit systems with any component arrangement. Rajagopalan & Cassady (2006) improved the efficiency of the original solution strategy by Rice et al. (1998), i.e. a total enumeration strategy, by more than two orders of magnitude by four individual improvements (which include, for example, defining upper and lower bounds for the variables and the objective function). Khatab et al. (2007) proposed two heuristic search algorithms, which iteratively add repair actions for failed components, having the highest improvement in the reliability of the system, until the cost or time constraint is saturated. Lust et al. (2009) proposed a heuristic search algorithm that is able to assign both minimal repair and replacement actions to the components. The authors report that, on a set of six optimization problems, their heuristic search algorithm yields solutions, the reliabilities of which are at most 3.71% worse than the optimal reliability. The benefit of heuristic algorithms is that they can quickly find a good solution. Lust et al. (2009) also applied the branch-and-bound and tabu search (Glover, 1989) methods to the selective maintenance problem.

Galante & Passannanti (2009) proposed a variation of the algorithm by Kettelle Jr (1962), which is capable to identify non-dominated maintenance decision vectors in the space of reliability and maintenance cost for serial systems. The variation extends the algorithm to serial-parallel systems. The authors applied the modified algorithm to a large-scale preventive maintenance optimization of ship components. Certa et al. (2011) extended the method by Galante & Passannanti (2009) to be suitable for multi-objective selective maintenance optimization, in which the objectives are to minimize the cost and duration of maintenance actions subject to a minimum reliability requirement.

Recently, further extensions have been proposed in the literature, in order to improve the relevance to industrial applications. In reality, possible maintenance actions may not be restricted to minimal repair or replacement, but to include also intermediate choices, i.e. imperfect maintenance. Liu & Huang (2010) extended the selection of actions in selective maintenance by relating the cost of the maintenance action to its quality via the Kijima type II model (Kijima et al., 1988; Kijima, 1989), in which a maintenance action reduces the (virtual) age of the component by the factor c from the range $[0, 1]$. Zhu et al. (2011) included an intermediate (imperfect) maintenance action as an addition to minimal repair and replacement by modeling both the age reduction factor and the hazard rate increase, the information of which was assumed to be known. Pandey et al. (2013a) also proposed an imperfect maintenance model that considers both the component age reduction and hazard rate adjustment. The abovementioned selective imperfect maintenance models limit the state of the system components to be only binary (i.e., working or failed). Pandey et al. (2013b) proposed a multi-state selective imperfect maintenance model, which relaxes this limitation, and allows the components to have also intermediate performance levels. Khatab et al. (2018) included the assignment of repair personnel to maintenance actions in their selective maintenance optimization model. Diallo et al. (2018) extended the problem definition from serial-parallel systems to serial n-out-of-k systems, i.e. a stage is functioning if n out of k components are functioning. Diallo et al. (2019) combined the selective maintenance optimization of serial n-out-of-k systems and the assignment of repair personnel.

In the aforementioned studies, the reliability objective, or constraint, corresponds to a single operation window. Maillart et al. (2009) used stochastic dynamic programming to solve optimization problems with two and infinite operation windows. However, they conclude that the (computationally most expensive) model with infinite number of operation windows yields only minimal improvement in the expected number of successful missions in comparison to models with a one and two operation windows. Long-term selective maintenance optimization has been modeled as a discrete-time Markov chain (Iyoob et al., 2006) and Markov

[1] Rice et al. (1998) only considered the second of these maintenance actions.

decision process (Liu et al., 2020). Ye et al. (2019) proposed an optimization model, based on a continuous-time Markov chain, for the design of reliable systems. Schneider & Cassady (2004, 2015) developed selective maintenance optimization models for a fleet of systems. The latter study includes a linearization of a fleet-level selective maintenance model.

Real-life optimization problems are often formulated using parameter values that involve uncertainty. Recently, studies considering uncertainty have been published in the selective mantenance optimization literature. Khatab et al. (2017) were the first to model uncertainty related to the break and mission durations are a stochastic optimization problem. Liu et al. (2018) proposed a sequence planning model for selective maintenance optimization under ucertainty in the break duration. Khatab & Aghezzaf (2016) and Duan et al. (2018) developed models for selective maintenance optimization with imperfect maintenance actions under uncertainty in the age reduction factors.

In the context of chemical plant reliability, Amaran et al. (2015) formulated a mixed-integer linear programming (MILP) model for long-term turnaround planning of integrated chemical sites, allowing the shutdowns of the sites to occur at different times. Their model is linear because it is defined based on the minimum maintenance frequency for each component, instead of the system reliability. Maintenance operations have also been included into scheduling models of chemical processes. Biondi et al. (2017) included the degradation of plant components into an MILP process scheduling model based on the state-task network (Kondili et al., 1993). They model degradation to reduce the maximum capacity of the components, or to restrict their operation modes. Vieira et al. (2017) incorporated the planning of maintenance operations into a resource-task network (Pantelides, 1994) based scheduling model, tailored for biopharmaceutical processes. Wu et al. (2020) integrated maintenance tasks into a scheduling model based on the general precedence formulation (Méndez et al., 2001). They model the component degradation to be depend on the sequence of multiple-grade batch runs.

The component lifetimes in the selective maintenance literature are commonly assumed to follow either the exponential or Weibull distributions (Cao et al., 2018). In the case of the former, the underlying assumption is that the failure rates are constant. Thus, only corrective maintenance actions for failed components are sensible; the replacement of a functioning component would have no influence on the system reliability. The Weibull distribution, on the other hand, can be used to describe components with increasing, constant or decreasing failure rates. However, the distribution is not suitable for modeling non-monotone failure rates.

In reality, many engineering components have a non-monotone bathtub-shaped failure rate, which is a combination of decreasing infant mortality rate, a constant random failure rate and an increasing failure rate due to degradation. A wide range of parametric distributions has been proposed in the literature to model such failure rates. Our aim here is only to provide a brief overview of these distributions. Mudholkar & Srivastava (1993) proposed an exponentiated Weibull distribution. Xie et al. (2002) also proposed an extension to the Weibull distribution that is flexible to model bathtub-shaped failure rates. El-Gohary et al. (2013) proposed a generalized Gompertz distribution (Gompertz, 1825). All the above mentioned distributions have three parameters. Sarhan & Apaloo (2013) proposed a four-parameter model that is a generalization of both models by Xie et al. (2002) and El-Gohary et al. (2013). Jiang (2013) proposed a new three-parameter finite support model, and showed evidence that finite support models yield good fits to data with bathtub-shaped failure rates. The model parameters are typically fitted to the data by the maximum likelihood method, or by the maximum spacing method (see the paper by Jiang (2013)).

In a recent review paper, Cao et al. (2018) stress the lack of data-driven approaches in selective maintenance literature. For the operators of the plant, the starting point for selective maintenance is typically some, perhaps limited, dataset of component lifetimes. However, in the corresponding literature, the aspect of data availability is often omitted, and the starting point is typically defined as a given lifetime distribution with arbitrarily chosen parameters. As Cao et al. (2018) point out, the linking of lifetime data with corresponding distribution parameters, in the context of selective maintenance, has not been discussed in the literature. *Therefore, as the first contribution of this paper, we link the statistical analysis of component lifetime data to the selective maintenance. More specifically, we study two lifetime datasets with bathtub-shaped failure rate distributions (Aarset, 1987; Meeker & Escobar, 1998), and use the failure rate models by Sarhan & Apaloo (2013) and Jiang (2013).*

When considering only a single maintenance break, the selective maintenance decision-making can be formulated as a mixed-integer nonlinear programming (MINLP)[2]. The algebraic term of the reliability of a serial-parallel system involves products of decision variables, which typically results in a non-convex MINLP problem. Recently, Ye et al. (2018) presented a convexified form of the reliability algebraic term. However, instead of selective maintenance, their work considered the reliability design of a new chemical plant. The convexified model is guaranteed to find the global optimum with a non-global MINLP solver. The authors showed that the solution time of their convexified model, using the non-global solver DICOPT (Viswanathan & Grossmann, 1990), was around half of that of the nonconvex model, using the global solver BARON (Tawarmalani & Sahinidis, 2005), for an example problem containing 42 binary variables. Further, in order to also avoid non-linearity, Diallo et al. (2018) proposed a two-stage approach, in which they first transform the problem into multi-dimensional multiple-choice knapsack problem and then solve it using MILP.

As far as we are able to ascertain, the largest selective maintenance problems reported, and solved to optimality, in the literature[3] have 200 system components (Galante & Passannanti, 2009), if only one maintenance action (e.g. replacement or repair) is considered, and 28 system components (Lust et al., 2009) if two (or more) maintenance actions are considered. It is also worth noticing that, regarding the latter category, Diallo et al. (2018) studied a problem with slightly fewer system components (23) – but report roughly three orders of magnitude smaller computational time than Lust et al. (2009), due to the linearization approach mentioned above. Evolutionary algorithms, as well as other heuristic approaches, provide an alternative solution method to tackle large-scale selective maintenance problems. However, with these approaches, the optimality of the solution cannot be guaranteed.

In order to improve the efficiency of selective maintenance optimization for industrial-scale problems, while still guaranteeing the optimality of the solution, the second contribution of this paper is two concurrently applicable improvements to the efficiency of MINLP based optimization. First, our statistical analysis shows that the component-specific reliability is reduced if the age of the component and the next planned operation window are within certain limits. This reduction is caused by the infant mortality period of new components. We preclude component replacements in such cases by variable preassignments, which reduces the size of the decision space. Second, we modify the aforementioned convexification of the reliability expression by Ye et al. (2018) to be applicable to selective maintenance optimization with replacement action. Further, we also derive the corresponding convexification applicable to selective maintenance optimization with both replacement and repair actions.

Nomenclature

Sets

K	Set of stages
I_k	All partitions of J_k into three subsets
J_k	Set of parallel units in stage k
S	A subset of J_k
$S_{k,m}$	Subset m of J_k
$\mathbb{S}_k$	Power set of J_k: $\mathbb{S}_k = \{S \mid S \subseteq J_k\}$
$S^{\text{x}}_{k,i}$	Repair subset of J_k on stage k
$S^{\text{y}}_{k,i}$	Replacement subset of J_k on stage k

Indices

k	Stage

[2]The reader may wish to consult papers by Grossmann (2002) and Belotti et al. (2013) for general reviews of MINLP, and Kronqvist et al. (2019) for a review of solution methods for convex MINLP problems.

[3]We consider here only studies with a single maintenance break.

i	A ternary partition of J_k
j	Parallel unit
q	Budget level
m	A subset of J_k

Parameters

$\alpha, \beta, \gamma, \eta, \lambda, k_{\mathrm{w}}$	Failure model parameters (used as variables in failure model fitting)
$\alpha_{j,k,i}$	Ternary parameter indicating to which subset (no action, repair, replacement) unit j at stage k belongs in partition i
$\omega_{j,k,m}$	Binary parameter indicating if unit j at stage k belongs to the m^{th} subset of $\mathbb{S}_k$
a	Age of a component
$a_{k,j}$	Age of component j at stage k
d_i	Binary parameter indicating if experiment i ended in failure
$F_{k,j}$	Binary parameter indicating if component j at stage k is functioning before the maintenance shutdown
$F(t)$	Cumulative failure function
$h(t)$	Failure rate
$R(t)$	Reliability function
$R^0_{k,j}$	Reliability of the component j at stage k, if the component is not replaced or repaired
$R^{\mathrm{x}}_{k,j}$	Reliability of the unit j at stage k, if the component is repaired
$R^{\mathrm{y}}_{k,j}$	Reliability of the unit j at stage k, if the component is replaced
$\Delta R^{\mathrm{x}}_{k,j}$	Improvement in the reliability of the unit j at stage k, if the component is repaired ($\Delta R^{\mathrm{x}}_{k,j} = R^{\mathrm{x}}_{k,j} - R^0_{k,j}$)
$\Delta R^{\mathrm{y}}_{k,j}$	Improvement in the reliability of the unit j at stage k, if the component is replaced ($\Delta R^{\mathrm{y}}_{k,j} = R^{\mathrm{y}}_{k,j} - R^0_{k,j}$)
$c^{\mathrm{y}}_{k,j}$	Cost of replacing unit j at stage k
$c^{\mathrm{x}}_{k,j}$	Cost of repairing unit j at stage k
$c_{\mathrm{budget},q}$	Cost upper bound for the maintenance at the budget level q of the ϵ-constraint method
c_{person}	Cost of hiring a maintenance person
n	Number of components in a system
$\hat{n}$	Number of samples in a dataset
T_{break}	The duration of the maintenance break
t	Time
t_i	End time of experiment i (may also be the failure time, see parameter d_i)
$t^{\mathrm{y}}_{k,j}$	Replacement duration of unit j at stage k
$t^{\mathrm{x}}_{k,j}$	Repair duration of unit j at stage k
t_{w}	Next operation window
$\mathcal{L}$	Likelihood function

Variables

c_{tot}	Total cost of the maintenance operations
p	Number of maintenance personnel involved in the maintenance operations
R'_k	Reliability of stage k
R_{sys}	System reliability
$\tilde{R}_{\mathrm{sys}}$	Logarithm of the system reliability
T_{break}	Duration of the maintenance break
T_{sum}	Sum of maintenance action durations
$w_{k,i}$	Binary variable used in the Convex Replacement-Repair (CRR) model
$x_{k,j}$	Binary variable defining whether unit j at stage k is repaired

$y_{k,j}$ Binary variable defining whether unit j at stage k is replaced
$z_{k,m}$ Binary variable used in the Convex Replacement (CR) model

2. Data analysis

Each component in an engineering system has an underlying failure rate, which is rarely explicitly known. However, the operators of the system can obtain implicit observations of the failure rate by collecting the lifetime data of the components while operating the system, or by running accelerated lifetime tests on individual components. The collected data are then fed to statistical models, in order to estimate the failure rate of the component.

As indicated in the introduction, the scope of this work is on lifetime datasets with bathtub-shaped failure rates. We have chosen to use two datasets reported by Aarset (1987) (Table 1) and Meeker & Escobar (1998) (Table 2), which we refer to as Datasets 1 and 2, respectively. Dataset 1 has no censored data points, whereas Dataset 2 is right-censored at 300 time units, which means that, even if no failure has occurred, the experiment is terminated at this point. The reasoning to choose these datasets is that their underlying distributions are bathtub-shaped, and they are widely studied in the literature. In addition, they represent cases with (Meeker & Escobar, 1998) and without (Aarset, 1987) right-sensored data.

Table 1: Lifetime dataset 1 (Aarset, 1987), consisting of failure times of 50 components. The dataset is a one-dimensional array, reported on multiple lines.

0.1	0.2	1	1	1	1	1	2
3	6	7	11	12	18	18	18
18	18	21	32	36	40	45	46
47	50	55	60	63	63	67	67
67	67	72	75	79	82	82	83
84	84	84	85	85	85	85	85
86	86						

Table 2: Lifetime dataset 2 (Meeker & Escobar, 1998), consisting of failure times of 30 components. Sign '+' indicates that the data point is right-censored.

2	10	13	23	23	28
30	65	80	88	106	143
147	173	181	212	245	247
261	266	275	293	300^+	300^+
300^+	300^+	300^+	300^+	300^+	300^+

Despite the scope being at bathtub-shaped failure rates, the failure models and optimization methods we discuss herein are also applicable to lifetime datasets with monotonically increasing failure rates[4]. If the failure rate is constant (i.e. the exponential lifetime distribution) or monotonically decreasing, the replacement action become irrelevant. The reason is that the replacement of a functioning component is not sensible, as, in this case, the actions would not improve the reliability of the component. The reader may wish to consult the paper by Aarset (1987), for a statistical method of identifying whether a dataset has a bathtub-shaped, or monotonically increasing or decreasing failure rate.

In the next two subsections, we present the algebraic equations of the failure rate $h(t)$ and cumulative failure function $F(t)$ of the aforementioned failure models by Jiang (2013) and Sarhan & Apaloo (2013). We

[4]With the caveat that, in this case, the aforementioned decision-space reduction by variable preassignments is no longer relevant.

have chosen to use these models because they are reported to yield good fits to datasets having a bathtub-shaped failure rate, in comparison to other models reported in the literature[5]. In addition, they represent two different classes of bathtub-shaped failure models; the former represents the class with finite support, whereas the latter the class of models with infinite support.

2.1. Failure rate model by Jiang (2013)

The failure rate $h(t)$ and cumulative failure function $F(t)$ of the model by Jiang (2013), having three adjustable model parameters, are defined as

$$\begin{cases} h(t) = \dfrac{\beta}{t+\eta} + \dfrac{1}{\gamma - t} & (1) \\[2ex] F(t) = 1 - \dfrac{1 - t/\gamma}{(1 + t/\eta)^{\beta}}, \qquad t < \gamma, & (2) \end{cases}$$

where β, γ, and η are the adjustable model parameters, defined to be positive $(\beta, \eta, \gamma > 0)$. The model is defined so that when $t \to \gamma$ the failure rate $h(t)$ approaches infinity, i.e. the finite support. The author indicates that the feature enables the model to adapt to failure models with a rapidly increasing failure rate during the wear-out phase.

2.2. Failure rate model by Sarhan & Apaloo (2013)

Sarhan & Apaloo (2013) define their model, which they refer to as the exponentiated modified Weibull extension distribution, to be a generalization of three models: the generalized Gompertz distribution (El-Gohary et al., 2013), the modified Weibull extension distribution (Xie et al., 2002) and the exponentiated Weibull distribution (Mudholkar & Srivastava, 1993). The model involves four adjustable parameters, and its failure rate $h(t)$ and cumulative failure function $F(t)$ are

$$\begin{cases} h(t) = \dfrac{\lambda\beta\gamma\left(\frac{t}{\alpha}\right)^{\beta-1} e^{(t/\alpha)^{\beta} + \lambda\alpha(1 - e^{(t/\alpha)^{\beta}})}}{[1 - e^{\lambda\alpha(1-e^{(t/\alpha)^{\beta}})}]^{1-\gamma} + e^{\lambda\alpha(1-e^{(t/\alpha)^{\beta}})} - 1} & (3) \\[2ex] F(t) = [1 - e^{\lambda\alpha(1-e^{(t/\alpha)^{\beta}})}]^{\gamma}, \qquad t \geq 0, & (4) \end{cases}$$

where λ, α, β and γ are the adjustable model parameters, also defined to be positive $(\lambda, \alpha, \beta, \gamma > 0)$.

Fitting a failure model to a dataset means seeking the model parameters that minimize, or maximize, a predefined goodness-of-fit measure. In the next section, we collect optimized parameters for both failure models on Datasets 1 and 2. We will also visualize Eqs. 1 to 4, using the optimized model parameters, at the end of the section (Figs. 1 and 2).

2.3. Fitting failure models to data

Sarhan & Apaloo (2013) use the maximum log-likelihood estimate to determine the model parameters, and report them for both Datasets 1 and 2. However, when analyzing Dataset 2, they assume that the right-censored values are actual failure times. Jiang (2013) reports the optimized model parameters for Dataset 2, which she determines based on the maximum log-likelihood estimate, but does not analyze Dataset 1. Thus, we lack the optimized parameters for the failure model by Jiang (2013) on Dataset 1 and the failure model by Sarhan & Apaloo (2013) on Dataset 2.

The fitting of bathtub-shaped failure models to data is well-established in the literature (see, for example, the papers by Xie et al. (2002), Jiang (2013), Sarhan & Apaloo (2013), and El-Gohary et al. (2013)). In the following, we briefly describe the log-likelihood function and the optimization approach that we used to obtain the missing parameter values.

[5]See Table 5 in the paper by Jiang (2013) and Tables 2 and 5 in the paper by Sarhan & Apaloo (2013) for the listing of the benchmark models.

The likelihood function of a failure model is defined as

$$\mathcal{L} = \prod_{i=1}^{\hat{n}} h(t_i)^{d_i} R(t_i), \tag{5}$$

where $\hat{n}$ is the number of points, and parameter d_i indicates whether component i has failed at time t_i. Further, $R(t_i)$ is the reliability of the component, i.e. the probability of the component being functioning at time $t = t_i$, given that it is new and functioning at time $t = 0$. The reliability $R(t)$ is the complement probability of the cumulative failure function $F(t)$:

$$R(t) = 1 - F(t). \tag{6}$$

The log likelihood of a failure model is then

$$\log \mathcal{L} = \sum_{i=1}^{\hat{n}} [d_i \log h(t_i) + \log R(t_i)]. \tag{7}$$

We tune the parameters of the failure model by Jiang (2013) on Dataset 1 by maximizing the extended maximum spacing[6], and the parameters of the failure model by Sarhan & Apaloo (2013) on Dataset 2 by maximizing the log-likelihood (Eq. 7). In order to solve the optimization problems, we use SLSQP (sequential least squares programming) (Kraft, 1988) as the optimization method, and initialize the optimization runs from 100 randomized starting points. The optimized model parameters and the corresponding log-likelihoods for Dataset 1 are listed in Table 3, in which the parameters for the failure model by Jiang (2013) are obtained from her paper and for that of Sarhan & Apaloo (2013) by the multi-start SLSQP approach. The corresponding values for Dataset 2 are listed in Table 4, in which the parameters for the failure model by Sarhan & Apaloo (2013) are obtained from their paper and for that of Jiang (2013) by the multi-start SLSQP approach. In both tables, we list, as a reference, the model parameters and log-likelihoods of exponential and Weibull distributions, which we also generate by the multi-start SLSQP approach. The adjustable parameter of the exponential distribution is λ and those of Weibull distribution are λ and k_{w}.

Table 3: Optimized model parameters for the studied failure models on Dataset 1 (Aarset, 1987). The methods used for training are the maximum a log-likelihood estimate (MLE) and extended maximum spacing method (EMSM).

model	method	trained parameters	$\log \mathcal{L}$
exponential	MLE	$\lambda = 45.686$	-241.09
Weibull	MLE	$\lambda = 44.913$, $k_{\mathrm{w}} = 0.94904$	-241.00
Jiang (2013)	EMSM	$\beta = 3.3588\text{e-}2$, $\gamma = 88.201$, $\eta = 0.13517$	-217.60
Sarhan & Apaloo (2013)	MLE	$\alpha = 49.05$, $\beta = 3.148$, $\gamma = 0.145$, $\lambda = 7.181\text{e-}5$	-213.86[i]

[i] The values are from the paper by Sarhan & Apaloo (2013).

On Dataset 1, the log likelihood of the trained model by Sarhan & Apaloo (2013) is higher than that of the trained model by Jiang (2013), indicating a better fit to the dataset. Similar results are obtained on Dataset 2, although the margin is very small. Further statistical assessment of which of these trained models has the best fit for the datasets falls outside the scope of this work. Suitable metrics for this are for example the Akaike information criterion (AIC) (Akaike, 1974) and the Kolmogorov-Smirnov test (Massey Jr,

[6] According to Jiang (2013), the maximum (log-)likelihood estimate may not be suitable for failure models with finite support, because the parameter defining the upper bound of the support can converge to the largest non-censored datapoint, in the case of which the log-likelihood approaches infinity (typically, this behavior is not seen if the largest observation is right-censored). The use of extended maximum spacing as the cost function avoids this problem (see the paper by Jiang (2013) for further information).

Table 4: Optimized model parameters for the studied failure models on Dataset 2 (Meeker & Escobar, 1998). See Table 3 for explanations of the cost functions.

model	method	trained parameters	$\log \mathcal{L}$
exponential	MLE	$\lambda = 241.41$	-142.70
Weibull	MLE	$\lambda = 242.59$, $k_{\mathrm{w}} = 0.92679$	-142.62
Jiang (2013)	MLE	$\beta = 6.6737\text{e-}2$, $\gamma = 452.35$, $\eta = 9.5118$	-141.36[ii]
Sarhan & Apaloo (2013)	MLE	$\alpha = 260.19$, $\beta = 4.3280$, $\gamma = 0.14848$, $\lambda = 9.5159\text{e-}5$	-141.23

[ii] The values are from the paper by Jiang (2013).

1951). As we indicated in the introduction, we have chosen the two bathtub-shaped failure models as representative models from the literature, and show also the distributions obtained by the exponential and Weibull distribution on the same datasets. The latter two are the most commonly used distributions in selective maintenance optimization (Cao et al., 2018). In the remainder of this section, we highlight the differences in the failure rate $h(t)$ and cumulative failure function $F(t)$ of the trained models.

Figures 1(a) and 1(b) visualize the failure rate $h(t)$ and cumulative failure function $F(t)$, respectively, of the trained models on Datasets 1. Figure 1(b) also shows the empirical failure distribution function. The failure rate of the models by Sarhan & Apaloo (2013) and Jiang (2013) have fundamentally the same shape, and in both models, the failure rate increases rapidly at around $t = 75$. However, during the mid-life period ($t \approx 10 \ldots 65$) of the component, the model by Sarhan & Apaloo (2013) predicts a lower failure rate than the model by Jiang (2013). When looking at the cumulative failure distribution, the model by Sarhan & Apaloo (2013) follows the empirical distribution closer than the model by Jiang (2013). Arguably, this is due to the additional flexibility provided by the additional model parameter. The cumulative failure distributions of the exponential and Weibull distributions are nearly identical because the trained model parameter k_{w} of the Weibull distribution is close to unity.

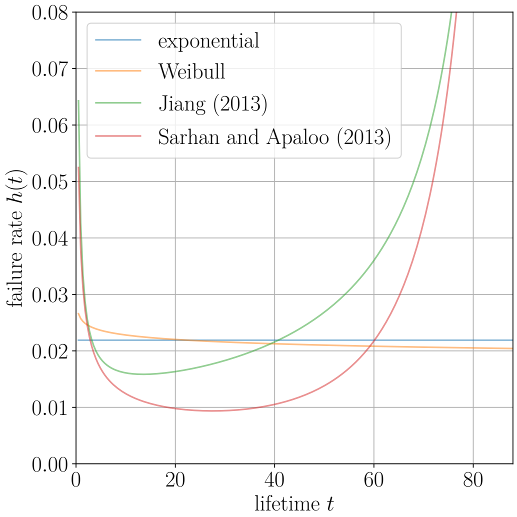
Figures 2(a) and 2(b) visualize the corresponding information of the trained models on Dataset 2. In this case, all four models predict fairly similar failure behavior before a lifetime of around $t = 300$ (Figure 2(b)), at which point the remaining functioning components are right-censored. Beyond $t = 300$, the trained model by Sarhan & Apaloo (2013) is the most pessimistic about the length of the remaining lifetime. This can be clearly seen in the rapidly increasing failure rate. The trained model by Jiang (2013) is also more pessimistic about the remaining lifetime than the trained exponential and Weibull distributions. The reason is that the latter two do not capture the underlying increasing failure rate in the dataset. It is also worth noticing that during the mid-life period of the component ($t \approx 25 \ldots 175$) the trained model by Sarhan & Apaloo (2013) predicts a lower failure rate than that by Jiang (2013), which further predicts a lower failure rate than the trained exponential and Weibull distributions (Figure 2(a)).

2.4. Maintenance actions

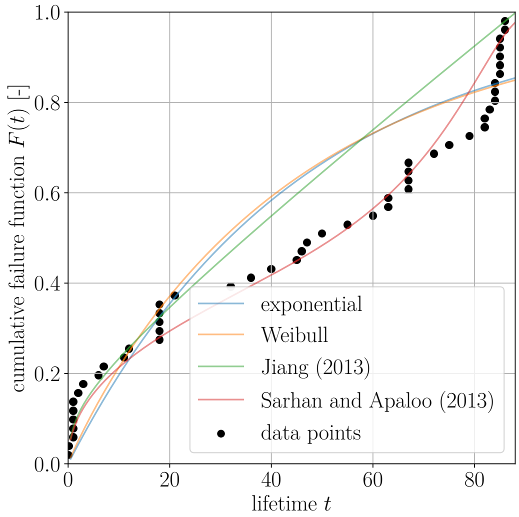
In this section, we transform the trained failure models into a format that can be used as an input for selective maintenance optimization. The selective maintenance actions we consider in this work are 1) minimal repair of a failed component, 2) replacement of a failed component and 3) replacement of a functioning component.

Let us now consider a component j located at stage k in a system of components (see Figure 3 as an example arrangement). The system is functioning if at least one component j at every stage $1 \ldots |K|$ is functioning, where K is the set of stages. Otherwise, the system is failed. We indicate the state of the component (k, j) at the start of the maintenance break by the binary parameter $F_{k,j}$, such that if $F_{k,j} = 1$ the component is functioning.

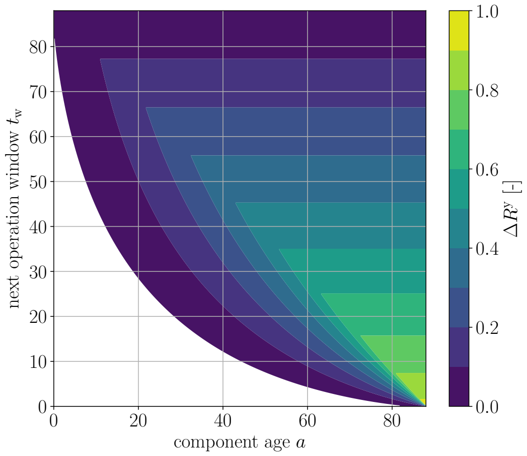
For the sake of simplicity, we assume in this work that all components of the system have an identical failure behavior. Nevertheless, the approaches we propose in this work do not rely on this assumption. We make this assumption in order to enable easy generation and reporting of results on large-scale problems



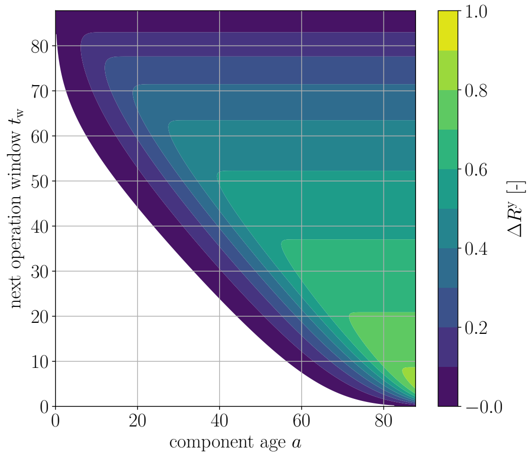
(a) Failure rate $h(t)$



(b) Cumulative failure function $F(t)$

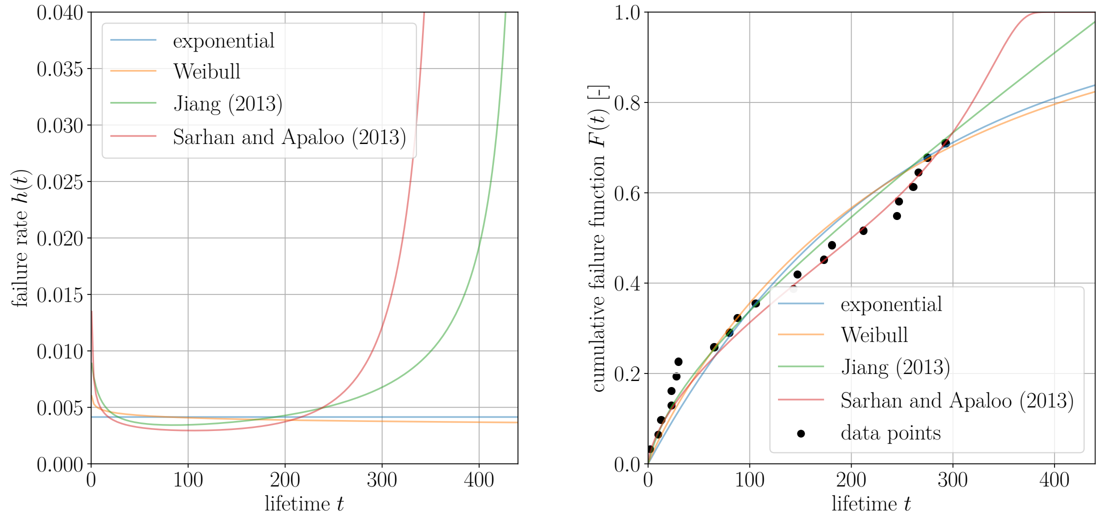


(c) Improvement in reliability based on the failure model by Jiang (2013).



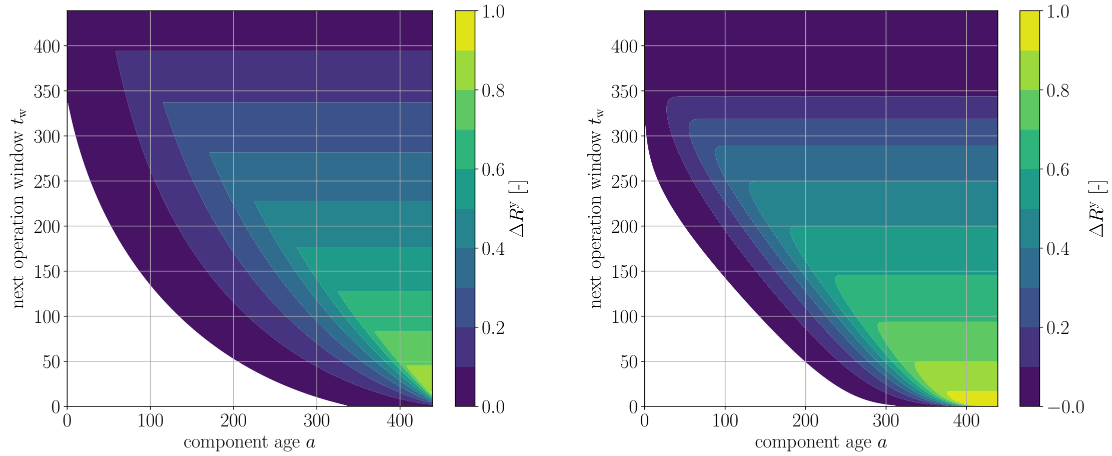
(d) Improvement in reliability based on the failure model by Sarhan & Apaloo (2013).

Figure 1: Fitting of failure models to Dataset 1 (Aarset, 1987). Subfigures (**c**) and (**d**) are contour plots of the reliability improvement if a functioning component (k, j) is replaced, $\Delta R^{\mathrm{y}}_{k,j}$ (Eq. 13). In the white regions of the plot, the improvement is negative, which means that the replacement is not sensible.



(a) Failure rate $h(t)$

(b) Cumulative failure function $F(t)$



(c) Improvement in reliability: model by (Jiang, 2013)

(d) Improvement in reliability: model by (Sarhan & Apaloo, 2013)

Figure 2: Fitting of failure models to Dataset 2 (Meeker & Escobar, 1998). Subfigures (**c**) and (**d**) are contour plots of the reliability improvement if a functioning component (k, j) is replaced, $\Delta R^{y}_{k,j}$ (Eq. 13). In the white regions of the plot, the improvement is negative, which means that the replacement is not sensible.

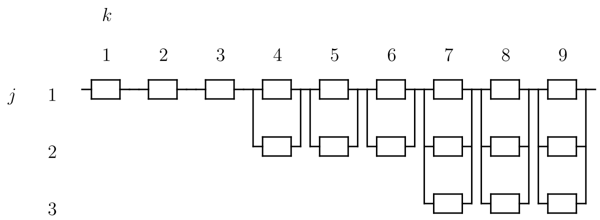


Figure 3: An example arrangement of plant stages k. Here, stages $k = \{1, 2, 3\}$ only have a single component (no redundancy), and stages $k = \{4, 5, 6\}$ and $k = \{7, 8, 9\}$ have two and three parallel components, respectively.

with varying number of components. Subject to the availability of relevant failure data, systems with non-identical failure behavior can be modeled by conducting the data analysis individually for each component.

The three above described maintenance actions were first considered by Cassady et al. (2001a), who modeled the component reliabilities using the Weibull distribution. The authors correctly state that, in an ideal case where the time (and cost) constraints are not active,

- a failed component should be replaced, if its shape parameter $k_{\mathrm{w}} > 1$,
- a failed component should be minimally repaired, if its shape parameter $k_{\mathrm{w}} \leq 1$,
- a functioning component should be replaced, if its shape parameter $k_{\mathrm{w}} > 1$, and
- no maintenance action should be assigned to a functioning component, if its shape parameter $k_{\mathrm{w}} \leq 1$.

In the previous section, we saw that, for both Datasets 1 and 2, the trained shape parameter of the Weibull distribution $k_{\mathrm{w}} < 1$. This means that the failure rate is predicted to be decreasing, and thus, ideally, all failed components should be minimally repaired and no maintenance action should be assigned to functioning components. The failure model would never recommend replacing a component, as it does not identify the wear-out periods in the datasets. The same always applies the exponential distribution, which by definition has a constant failure rate. This highlights the importance of identifying the type of failure behavior in datasets, and accordingly using a relevant failure model.

The choice of maintenance action affects the reliability of the component during the next operation window. We define these reliabilities using the conditional reliability

$$R(a + t_w \mid a) = \frac{R(a + t_w)}{R(a)}, \tag{8}$$

where t_{w} is the length of the next operation window and a is the age of the component at the start of the maintenance break. The conditional reliability $R(a + t_{\mathrm{w}} \mid a)$ is the probability of a component being functioning at age $a + t_{\mathrm{w}}$, taken that it was functioning at age a.

Thus, the resulting reliabilities of the component (k, j) are

$$\begin{cases} R^0_{k,j} = R(a_{k,j} + t_{\mathrm{w}} \mid a_{k,j}) F_{k,j} & (9) \\ R^{\mathrm{x}}_{k,j} = R(a_{k,j} + t_{\mathrm{w}} \mid a_{k,j}) & (10) \\ R^{\mathrm{y}}_{k,j} = R(t_{\mathrm{w}} \mid 0) & (11) \end{cases}$$

where $R^0_{k,j}$, $R^{\mathrm{x}}_{k,j}$, and $R^{\mathrm{y}}_{k,j}$ correspond to situations where no maintenance action is assigned to the component, the component is repaired or the component is replaced, respectively. Throughout the equations of this work, we denote repair and replacement actions by letters 'x' and 'y', respectively. For the sake of easier notation later in this work, we define two new parameters

$$\begin{cases} \Delta R^{\mathrm{x}}_{k,j} = R^{\mathrm{x}}_{k,j} - R^0_{k,j} = R(a_{k,j} + t_{\mathrm{w}} \mid a_{k,j}) & (12) \\ \Delta R^{\mathrm{y}}_{k,j} = R^{\mathrm{y}}_{k,j} - R^0_{k,j} = R(t_{\mathrm{w}} \mid 0) - R(a_{k,j} + t_{\mathrm{w}} \mid a_{k,j}) F_{k,j}, & (13) \end{cases}$$

the former of which defines the change in the reliability if the component is repaired and the latter of which the corresponding change if the component is replaced. In the former equation, the term $R^0_{k,j} = 0$, as only a failed component can be repaired ($F_{k,j} = 0$).

As reliability is always nonnegative, $\Delta R^{\mathrm{x}}_{k,j} \geq 0$ and $\Delta R^{\mathrm{y}}_{k,j} \geq 0$ for all failed components. Interestingly, when the failure rate of a functioning component (k, j) is bathtub-shaped, its $\Delta R^{\mathrm{y}}_{k,j}$ may be either positive or negative. Figures 1(c) and 1(d) depict the contour plots of $\Delta R^{\mathrm{y}}_{k,j}$ in a space of the component age a and the length of next operation window t_{w} using the trained failure models by Jiang (2013) and Sarhan & Apaloo (2013), respectively, on the Dataset 1. Figures 1(c) and 1(d) depict the corresponding plots on Dataset 2. On both datasets, general appearances of the plots are similar, despite being generated by different failure models. The clearest visible difference is the different shape of the top left corner of the isocurves. The corner is sharp when using the failure model by Jiang (2013) and smooth when using that by Sarhan & Apaloo (2013). The reason is that the former model has finite and the latter infinite support.

Finally, we wish to highlight the region of negative $\Delta R^{\mathrm{y}}_{k,j}$ in the bottom left corner of the plots on both datasets (indicated by the white color). Replacing a (functioning) component lying in this region is not sensible because the action would reduce its reliability. This behavior is caused by the infant mortality period of the component having a bathtub-shaped failure rate. In section 5.1, we will exploit this observation by preassigning binary variables corresponding to such components to zero, in order to reduce the decision space of the optimization problem.

3. Mathematical models

In this section, we define two mathematical models for selective maintenance optimization. In the first (Section 3.1), the maintenance actions are restricted to replacement only, whereas the second (Section 3.2) includes both replacement and minimal repair. Replacement and repair actions on the component j at the stage k are modeled as binary variables $y_{k,j}$ and $x_{k,j}$, respectively. An action (replacement or repair) is conducted, if the corresponding binary variable equals one. We here define the two models separately because, later in Section 5, we will convexify both of them, and examine their applicability to large-scale problems.

3.1. Non-convex replacement model

Let us start with the replacement model, and consider a stage k in the system of $|K|$ parallel stages. The stage k is functioning if at least one of its $|J_k|$ components is functioning. Therefore, its reliability is

$$R'_k = 1 - \prod_{j \in J_k} (1 - R^0_{k,j}(1 - y_{k,j}) - R^{\mathrm{y}}_{k,j} y_{k,j}), \quad k \in K, \tag{14}$$

where $R^{\mathrm{y}}_{k,j}$ and $R^0_{k,j}$ are the alternative reliabilities of the component (k, j) during the next operation window if the component is or is not replaced, respectively. These parameters were defined in Eqs. 11 and 9, respectively. Using Eq. 13, Equation 14 simplifies into

$$R'_k = 1 - \prod_{j \in J_k} (1 - R^0_{k,j} - \Delta R^{\mathrm{y}}_{k,j} y_{k,j}), \quad k \in K. \tag{15}$$

As the system consists of $|K|$ stages in series, its reliability is

$$R_{\mathrm{sys}} = \prod_{k \in K} R'_k. \tag{16}$$

By definition, selective maintenance optimization features constraints that limit the number of maintenance actions that can be performed. In the literature, the two most commonly considered constraints are time and cost budgets. We define our model here by considering a situation where the replacement of a component (k, j) incurs the cost $c^{\mathrm{y}}_{k,j}$ and requires a working time $t^{\mathrm{y}}_{k,j}$ by a maintenance person. The number of personnel assigned to the maintenance break is an integer variable [7] p. We model the total duration required to perform the maintenance actions as the variable T_{sum}, defined as

$$T_{\text{sum}} = \sum_{k \in K} \sum_{j \in J_k} t^{\text{y}}_{k,j} y_{k,j}. \tag{17}$$

The total duration of the maintenance break is constrained to T_{break}. Thus, in order to finish all maintenance actions in time, the number of maintenance personnel p needs to satisfy constraint

$$T_{\text{sum}} \leq T_{\text{break}} p. \tag{18}$$

The total cost then defined as

$$c_{\text{tot}} = \sum_{k \in K} \sum_{j \in J_k} c^{\text{y}}_{k,j} y_{k,j} + c_{\text{person}} p, \tag{19}$$

where c_{person} is the cost of involving one maintenance person in the maintenance break.

For the operators planning the maintenance actions for the system, it is beneficial to know the trade-off between the conflicting maximum systems reliability R_{sys} and the minimum total cost c_{tot}. This trade-off can be determined by solving the bi-objective MINLP optimization problem, defined as

$$\begin{aligned} \max_{\mathbf{y},p} \quad & R_{\text{sys}}, -c_{\text{tot}} \\ \text{subject to} \quad & \text{Eqs. 15 - 19.} \end{aligned} \tag{20}$$

We solve this optimization problem by the ϵ-constraint method (Haimes et al., 1971), by transforming the minimization of the total cost into the following iteratively-relaxed constraint

$$c_{\text{tot}} \leq c_{\text{budget},q}, \tag{21}$$

where $c_{\text{budget,q}}$ is the cost upper bound of the budget level q. At each budget level q, we then solve the MINLP optimization problem

$$\begin{aligned} \max_{\mathbf{y},p} \quad & R_{\text{sys},q} \\ \text{subject to} \quad & \text{Eqs. 15 - 19, 21,} \end{aligned} \tag{NCR}$$

which we refer to, later in this work, as the Non-Convex Replacement (NCR) model.

3.2. Non-convex replacement-repair model

In this section, we define the second model, involving both replacement and minimal repair actions. Using the two actions, the reliability of stage k is defined as

$$R'_k = 1 - \prod_{j \in J_k} (1 - R^0_{k,j}(1 - y_{k,j} - x_{k,j}) - R^{\text{y}}_{k,j} y_{k,j} - R^{\text{x}}_{k,j} x_{k,j}), \quad k \in K, \tag{22}$$

where $R^0_{k,j}$, $R^{\text{x}}_{k,j}$ and $R^{\text{y}}_{k,j}$ are the alternative reliabilities of the component (k, j), depending on the assigned maintenance action. These reliabilities were defined in Eqs. 9, 10 and 11, respectively. Again, using the changes in the reliabilities (in this case, Eqs. 12 and 13), the equation simplifies into

$$R'_k = 1 - \prod_{j \in J_k} (1 - R^0_{k,j} - \Delta R^{\text{y}}_{k,j} y_{k,j} - \Delta R^{\text{x}}_{k,j} x_{k,j}), \quad k \in K. \tag{23}$$

During one maintenance break, the component (k, j) can only be either replaced or repaired, and repaired only if it is failed at the start of the maintenance break. Accordingly, we here use the following constraints, defined by Cassady et al. (2001a):

$$\begin{cases} y_{k,j} + x_{k,j} \leq 1, \quad k \in K, j \in J_k & (24) \\ F_{k,j} + x_{k,j} \leq 1, \quad k \in K, j \in J_k. & (25) \end{cases}$$

[7] This variable can also be used to represent the number of maintenance teams, or any other unit of workforce.

In order to define the corresponding cost model, we define two new parameters $c^{\text{x}}_{k,j}$ and $t^{\text{x}}_{k,j}$, which are the cost and duration of repairing the component (k, j), respectively. The total duration of performing all maintenance actions is then

$$T_{\text{sum}} = \sum_{k \in K} \sum_{j \in J_k} t^{\text{y}}_{k,j} y_{k,j} + \sum_{k \in K} \sum_{j \in J_k} t^{\text{x}}_{k,j} x_{k,j}, \tag{26}$$

and the total cost

$$c_{\text{tot}} = \sum_{k \in K} \sum_{j \in J_k} c^{\text{y}}_{k,j} y_{k,j} + \sum_{k \in K} \sum_{j \in J_k} c^{\text{x}}_{k,j} x_{k,j} + c_{\text{person}} p. \tag{27}$$

As a summary, at every budget level q of the ϵ-constraint method, we solve the following MINLP optimization problem:

$$\begin{aligned} \max_{\mathbf{x},\mathbf{y},p} \quad & R_{\text{sys},q} \\ \text{subject to} \quad & \text{Eqs. 16, 18, 21, 23 - 27,} \end{aligned} \tag{NCRR}$$

which we refer to as the Non-Convex Replacement-Repair (NCRR) model.

4. Illustrative examples

Before considering any large-scale problems, let us here define and examine an illustrative small-scale example problem. We solve the problem in Section 4.1 using the non-convex replacement and replacement-repair models, defined in Sections 3.1 and 3.2, respectively, and highlight the differences in between the obtained results (Section 4.2). Finally, in Section 4.3, we demonstrate, using the latter model, the differences in the final Pareto optimal solutions when the reliability parameters are based on the same failure data but determined using the two different bathtub-shaped failure models.

4.1. Optimization problem

We define our example system to comprise five different types of components, the cost and replacement/repair durations of which are presented in Table 5. Component types I and II represent those that are relatively cheap to replace/repair but are located in inconvenient locations, i.e. accessing them requires unbuilding other components of the system. Component types III to V represent those that are the opposite.

Table 5: Component catalog (items I - V).

Component type	I	II	III	IV	V
cost of replacement c^{y} [kEUR]	1	3	5	7	8
cost of repair c^{x} [kEUR]	0.5	0.3	1.4	1	2
duration of replacement t^{y} [h]	30	10	5	7	8
duration of repair t^{x} [h]	20	5	2	5	3

The example system has the component arrangement shown in Figure 3 consisting of three single, three double and three triple stages of components in series, i.e. $|J_1| \dots |J_3| = 1$, $|J_4| \dots |J_6| = 2$, $|J_7| \dots |J_9| = 3$. Table 6 lists the component types lying at each location in the arrangement, which we generated by drawing them randomly from the component catalog (Table 5).

Table 6: Component types of the system.

		stage k								
component type		1	2	3	4	5	6	7	8	9
unit j	1	I	IV	IV	V	IV	II	I	I	I
	2				I	III	V	III	I	IV
	3							I	V	II

We assume that all components have a failure behavior equivalent to that yielding the Dataset 1, and use the trained model by Sarhan & Apaloo (2013) to predict the reliability parameters $R^{0}_{k,j}$, $R^{\mathrm{x}}_{k,j}$ and $R^{\mathrm{y}}_{k,j}$. Dataset 1 does not have units in its original reference (Aarset, 1987). In order to place our illustrative example into a reasonable time-scale, we assume that the lifetimes of Dataset 1 are months. Further, we define the length of the next operation window to be $t_{\mathrm{w}} = 10$ months, the length of the maintenance break to be $T_{\mathrm{break}} = 50$ h, and the cost of hiring a maintenance person to be $c_{\mathrm{person}} = 4$ kEUR. The age distribution of the components is drawn randomly from the range of $\{10, 20, \ldots, 70\}$ months. In addition, four out of 18 components in the system are failed at the start of the maintenance break. Table 7 shows the age distribution, as well as the failed components, in the system.

Table 7: Ages of the components at the start of the maintenance break. Failed components ($F_{k,j} = 0$) are indicated by crosses.

		stage k								
age $a_{k,j}$ [month]		1	2	3	4	5	6	7	8	9
unit j	1	20	50	$\cancel{70}$	70	10	10	70	60	20
	2				$\cancel{70}$	30	40	$\cancel{40}$	40	$\cancel{10}$
	3							40	70	40

4.2. Results from the replacement and replacement-repair models

We generate the Pareto front of solutions to the illustrative example by starting from the total budget $c_{\mathrm{budget},1} = 0$ kEUR, and iteratively increasing it by 0.5 kEUR until $c_{\mathrm{budget},110} = 54.5$ kEUR. At each budget level, we solve Models NCR and NCRR by the global MINLP solver BARON 18.5.8 (Tawarmalani & Sahinidis, 2005), using the relative optimality criterion of 10^{-6}. Table 8 summarizes the size of the MINLP optimization problem when solving it by the two models.

Table 8: The number of variables and constraints in Models NCR and NCRR for the optimization problem defined in Section 4.1, and the average CPU time when solving the models by BARON.

model	variables			constraints	average CPU time [s]
	binary	integer	scalar		
NCR	18	1	12	14	0.01
NCRR	36	1	12	50	0.01

Figure 4 presents the solutions obtained by iteratively solving Models NCR and NCRR, as well as illustrations of representative solutions (duplicate and dominated solutions are filtered). Both sets of solutions are Pareto optimal to their own optimization problems. However, if we examine them all as solutions to Model NCRR, only two solutions obtained by solving Model NCR are Pareto optimal (representative solutions (1) and (5)). The gap between the two frontiers demonstrates the general improvement in the solutions when including the repair action in the model. In the literature, Liu & Huang (2010) obtained a similar result when comparing models with and without imperfect maintenance actions.

Representative solution (1) is the trivial solution where no maintenance actions are performed. Representative solutions (2) and (4) are those with the lowest total cost c_{tot} while still yielding a functioning system ($R_{\mathrm{sys}} > 0$) after the maintenance break, obtained by Models NCRR and NCR, respectively. The obvious difference is that in the former the failed component $(3, 1)$ repaired, whereas in the latter it is replaced. On the other hand, representative solutions (6) and (7) are those with the highest system reliability R_{sys}. It is worth noticing that in these solutions none of the functioning components younger than 50 months is replaced. The reason for this is that, for all of these components, the parameter $\Delta R^{\mathrm{y}}_{k,j} < 0$ (see Figure 1).

As a reference, we also generate results by a slightly modified version of the heuristic search algorithm by Lust et al. (2009), which we have implemented in Python. In this case, all solutions the algorithm yields for Models NCR and NCRR are Pareto optimal. However, because of its additive way of constructing the solutions, the algorithm cannot find all solutions lying at the Pareto fronts.

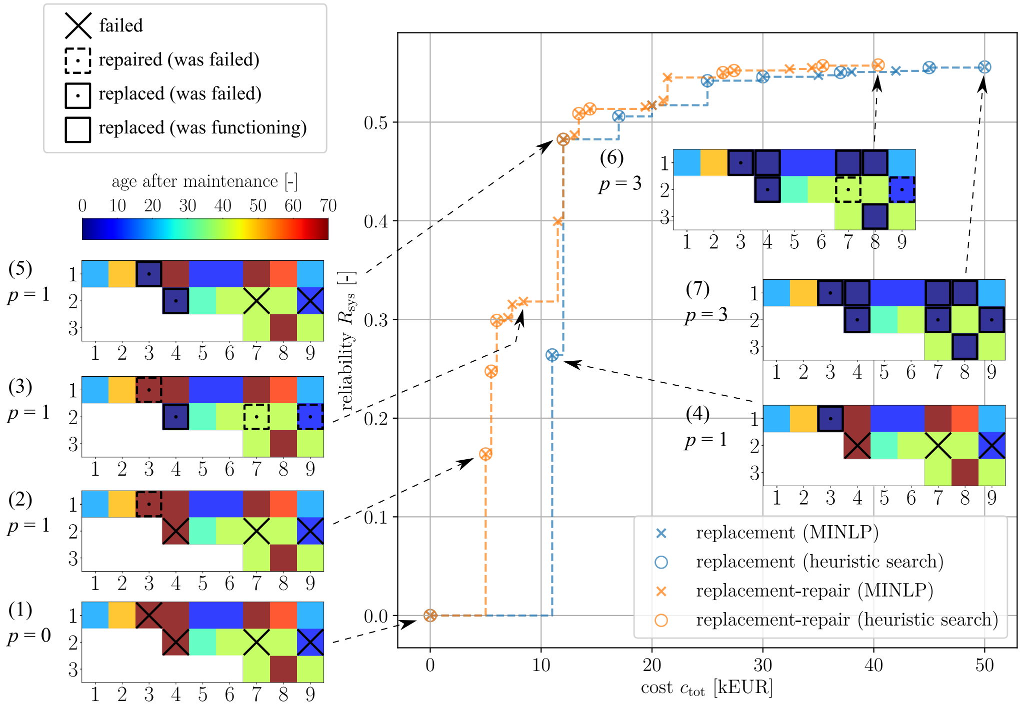


Figure 4: Pareto fronts obtained from the non-convex replacement (NCR) and replacement-repair (NCRR) models. The plot visualizes also representative solutions (1)-(7) from the Pareto front, showing their the maintenance actions, the resulting component age distribution (after the maintenance operations) and the number of maintenance personnel involved, p.

4.3. Decision-making based on different failure models

In the previous section, we used reliability parameters predicted by the trained failure model by Sarhan & Apaloo (2013). The purpose of this section is to examine how different the decision-making of maintenance actions is if the reliability parameters are instead predicted by the other bathtub-shaped failure model, i.e. the one by Jiang (2013).

Figure 5(a) shows the comparison of results obtained by solving the illustrative example, defined in Section 4.1, by Model NCRR using the reliability parameter predictions from the two different failure models (i.e., those by Sarhan & Apaloo (2013) and Jiang (2013)). Starting from the bottom left corner of both Pareto fronts, the first 12 solutions are the same, regardless of the different reliability parameter predictions used as inputs. Representative solution (2) shows the maintenance actions and the resulting component age distribution of the 12$^{\text{th}}$ solution. Pairwise, the solutions have the same total cost c_{tot} (and are therefore vertically aligned), but different system reliability R_{sys}. Representative solution (2) has the system reliability of $R_{\text{sys}} = 0.5134$, if determined by the reliability parameters from the failure model by Sarhan & Apaloo (2013), and $R_{\text{sys}} = 0.3617$, in the case of those from the failure model by Jiang (2013). The reason for the difference is that the trained model by Jiang (2013) predicts a higher failure rate during the mid-life period ($t \approx 10 \ldots 65$) of the components than the model by Sarhan & Apaloo (2013), see Figure 1(a). Representative solutions (3) and (4) are different (see the actions assigned for components (7,1) (8,1)).

The solutions at the top right end of the Pareto fronts (representative solutions (5) and (6)) maximize the system reliability R_{sys} for the next operation window. The solutions are otherwise the same, but in (5) the component (2,1) is not replaced, whereas in (6) it is replaced. The reason for this difference is that the parameter $\Delta R^{\text{y}}_{2,1}$ is negative when determined by the failure model by Sarhan & Apaloo (2013), and positive when determined by that by Jiang (2013) (see the point ($a = 50$, $t_{\text{w}} = 10$) in Figures 1(c) and 1(d)).

In order to further examine the differences in the decision-making, we defined another illustrative example problem, which is the same as the one in Section 4.1, but with the following changes. First, the components are assumed to have a failure behavior equivalent to that yielding the Dataset 2. Second, we create a new randomized instance of the component type arrangement (Table 9). Third, the length of the next operation window is changed to $t_\mathrm{w} = 60$ months, and we draw the component ages (randomly) from the range of [60, 120, ..., 300] months. Table 10 shows the randomized age distribution and failed components.

Table 9: Component arrangement of the system (the second instance).

		stage								
component type		1	2	3	4	5	6	7	8	9
unit	1	III	II	V	I	I	I	I	IV	II
	2				III	I	V	IV	I	IV
	3							II	V	IV

Table 10: Ages of the components at the start of the maintenance break (the second instance). Failed components ($F_{k,j} = 0$) are indicated by crosses.

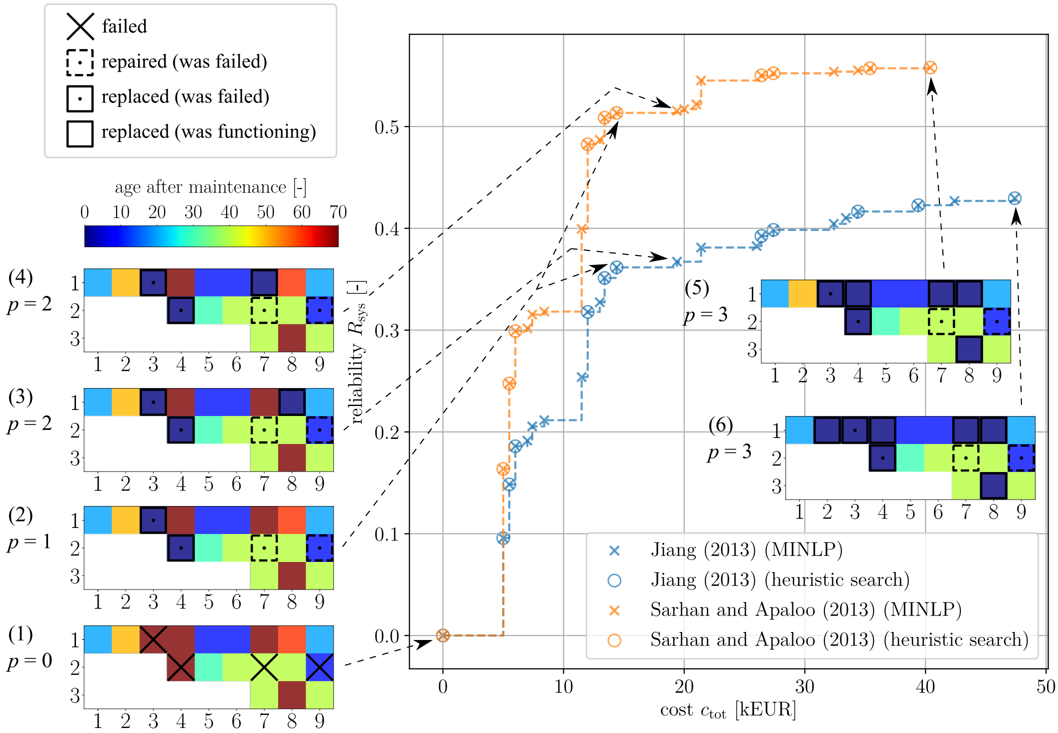
		stage								
age a [month]		1	2	3	4	5	6	7	8	9
unit	1	120	300	60	180	240	~~240~~	~~120~~	60	300
	2				~~60~~	240	240	60	240	60
	3							240	~~240~~	240

Figure 5(b) shows the obtained Pareto fronts with the two different failure models. Unlike the previously studied system instance, this instance is functioning at the start of the maintenance break. If no maintenance is performed during the break (representative solution (1)), the system has predicted reliabilities of $R_\mathrm{sys} = 0.1682$ (Jiang, 2013) and $R_\mathrm{sys} = 0.0370$ (Sarhan & Apaloo, 2013). The reason for the difference is that the system has many relatively old components, for which the failure model by Sarhan & Apaloo (2013) predicts significantly higher failure rates than the model by Jiang (2013), see Figure 2(a). On the other hand, for the representative solution (3), laying at the other extremes of the Pareto fronts, the predicted system reliabilities are in the opposite order: $R_\mathrm{sys} = 0.4567$ (Sarhan & Apaloo, 2013) and $R_\mathrm{sys} = 0.4058$ (Jiang, 2013). In this case, the system has relatively young components, which then mostly operate in their mid-life period ($t \approx 25 \ldots 175$) during the next operation window. In Section 2.3, we pointed out that the failure model by Sarhan & Apaloo (2013) predicts a lower failure rate than the model by Jiang (2013) for components in their mid-life period, which explains the difference in the system reliabilities. Solutions on the Pareto fronts between representative solutions (2) and (3) are pairwise the same.

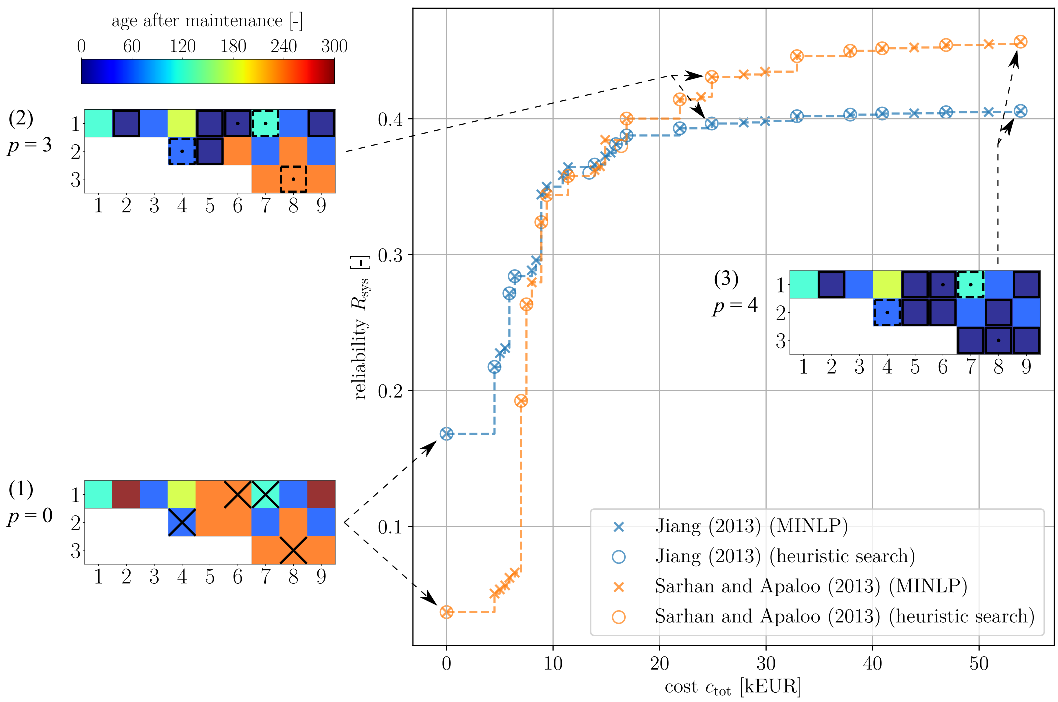
As a summary, we here made a comparison between the two representative bathtub-shaped failure models, which both compare well against other models in the literature. We observe that, if these models are used to predict reliability parameters based on the same lifetime dataset, significantly different system reliability predictions are obtained, and the decisions of the maintenance actions are also partially different. This highlights the importance of carefully choosing a relevant failure model, and tuning its model parameters, for a given lifetime dataset.

5. Large-scale selective maintenance optimization

As already mentioned in the introduction, the number of individual replaceable/repairable components in a real industrial system (e.g. a chemical production plant, power plant or ship) is in the order of hundreds, or even thousands – far beyond the size of the illustrative example. Earlier, in Section 1, we listed the largest selective maintenance optimization problems reported, and optimally solved, in the literature. In this section, we investigate two improvements to the MINLP model formulations, in order to reduce the computational cost of such large-scale problems while still guaranteeing the global optimality.



(a) Dataset 1



(b) Dataset 2

Figure 5: Comparison of the obtained results in the illustrative example when using the reliability parameters from the failure models by Jiang (2013) and Sarhan & Apaloo (2013). The results are generated by solving the non-convex replacement-repair (NCRR) model. As the failure models yield different reliability parameters the results are Pareto fronts of different optimization problems.

5.1. Variable preassignment

In Section 2.4, we derived parameters $\Delta R^{\mathrm{y}}_{k,j}$ and $\Delta R^{\mathrm{x}}_{k,j}$, indicating the changes in the reliability of the component (k, j) if being replaced or repaired, respectively. We observed that, if the failure model has a bathtub-shaped failure rate, the parameter $\Delta R^{\mathrm{y}}_{k,j}$ (Eq. 13) is negative in a certain region in the space of component age a and next operation window t_{w}, see Figures 1(c), 1(d), 2(c) and 2(d). This means that replacing such (functioning) component is not sensible because it would undesirably reduce the system reliability. Such actions, although being possible, were correctly not included in any of the Pareto optimal solutions of the illustrative example (Section 4.1).

Thus, as the first improvement, we define preassignments that preclude replacements, a priori known to reduce the system reliability, from the decision space:

$$y_{k,j} = 0, \quad \forall k, j \in \{(k, j)|\Delta R^{\mathrm{y}}_{k,j} \leq 0\}. \tag{28}$$

In general, reducing the size of the decision space is likely to reduce the computational effort of solving the optimization problem.

5.2. Convexification of the replacement model

Solving Models NCR and NCRR with optimality guarantees requires a global optimization method[8], because of the non-convex algebraic equations (Eqs. 15 and 23) defining the system reliability. Convexification of these equations would enable the models to be solved with a non-global MINLP method, such as the Generalized Benders Decomposition (Geoffrion, 1972), the Outer-approximation (Duran & Grossmann, 1986), or the Extended Cutting Plane (Westerlund & Pettersson, 1995) method. These methods are, in general, computationally less intensive than global optimization methods (Kronqvist et al., 2019). Therefore, we convexify, in this section and Section 5.4, both Models NCR and NCRR, respectively. This is the latter of our two investigated improvements.

Let us start with the non-convex replacement (NCR) model. The objective function R_{sys}, defined in Eq. 16, is the product of the stage reliabilities $R'_k, k \in K$. As each of these reliabilities include multi-linear terms (Eq. 15), the objective function is nonlinear and non-convex. Ye et al. (2018) proposed a linearization of a constraint nearly equal to Eq. 15, which enables the convexification of the objective function. They conduct the linearization by first expanding the products of linear terms in Eq. 15 into summations of multi-linear terms, and then linearizing the resulting multi-linear terms. However, in their model, the constraint equivalent to Eq. 15 does not include term $-R^0_{k,j}$. In the following, we describe the convexification proposed by Ye et al. (2018) and highlight the difference caused by the additional term.

Let us first expand the product of linear terms in Eq. 15 into the summation of multi-linear terms. In order to enable the expansion, we denote the power set of J_k by $\mathbb{S}_k = \{S|S \subseteq J_k\}$. As an example, if stage $k = 1$ consists of three parallel units, the power set $\mathbb{S}_1 = \{\varnothing, \{1\}, \{2\}, \{1, 2\}, \{3\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$. Further, we denote the m^{th} set of $\mathbb{S}_k$ (i.e. the m^{th} subset of J_k) by $S_{k,m}$. Using the newly defined sets, Eq. 15 can be expanded into

$$\begin{aligned} R'_k &= 1 - \prod_{j \in J_k} (1 - R^0_{k,j} - \Delta R^{\mathrm{y}}_{k,j} y_{k,j}) \\ &= 1 - \sum_{S_{k,m} \in \mathbb{S}_k} \Big(\prod_{j \in S_{k,m}} (-\Delta R^{\mathrm{y}}_{k,j} y_{k,j}) \prod_{j \in J_k \setminus S_{k,m}} (1 - R^0_{k,j}) \Big), \quad k \in K. \end{aligned} \tag{29}$$

In the model by Ye et al. (2018), the term $-R^0_{k,j}$ is absent, and therefore the last product becomes unity, which simplifies the equation. As the additional term is present in our case, this simplification cannot be performed.

The above mentioned power set $\mathbb{S}_k$ of J_k can be systematically generated for any finite number of parallel units by the equation

$$\omega_{j,k,m} = \left\lfloor \frac{\mathrm{mod}(m - 1, 2^j)}{2^{j-1}} \right\rfloor, \quad k \in K, \tag{30}$$

[8] In the illustrative example, we used the global MINLP solver BARON.

where the binary parameter $\omega_{j,k,m}$ defines whether unit j at stage k belongs to the m^{th} set of $\mathbb{S}_k$.

Next, we describe the linearization of Eq. 29. First, we introduce a new binary variable $z_{k,m}$, defined as

$$z_{k,m} = \prod_{j \in S_{k,m}} y_{k,j}, \quad k \in K, S_{k,m} \in \mathbb{S}_k. \tag{31}$$

The following logic propositions hold for $z_{k,m}$ (Glover & Woolsey, 1974)

$$\begin{aligned} &z_{k,m} \Leftrightarrow (\bigwedge_{j \in S_{k,m}} y_{k,j}), && k \in K, S_{k,m} \in \mathbb{S}_k, S_{k,m} \neq \varnothing \\ &z_{k,m} = 1, && k \in K, S_{k,m} = \varnothing. \end{aligned} \tag{32}$$

Raman & Grossmann (1991) reformulated these conditions into the following two linear inequalities

$$z_{k,m} \leq y_{k,j}, \quad k \in K, j \in S_{k,m}, S_{k,m} \in \mathbb{S}_k, S_{k,m} \neq \varnothing \tag{33}$$

$$z_{k,m} \geq \sum_{j \in S_{k,m}} y_{k,j} - |S_{k,m}| + 1, \quad k \in K, S_{k,m} \in \mathbb{S}_k. \tag{34}$$

Using Eq. 31, the linearized form of Eq. 29 becomes

$$\begin{aligned} R'_k &= 1 - \sum_{S_{k,m} \in \mathbb{S}_k} \Big(\prod_{j \in S_{k,m}} (-\Delta R^{\text{y}}_{k,j} y_{k,j}) \prod_{j \in J_k \setminus S_{k,m}} (1 - R^0_{k,j}) \Big) \\ &= 1 - \sum_{S_{k,m} \in \mathbb{S}_k} \Big(\prod_{j \in S_{k,m}} y_{k,j} \prod_{j \in S_{k,m}} -\Delta R^{\text{y}}_{k,j} \prod_{j \in J_k \setminus S_{k,m}} (1 - R^0_{k,j}) \Big) \\ &= 1 - \sum_{S_{k,m} \in \mathbb{S}_k} \Big(z_{k,m} \prod_{j \in S_{k,m}} -\Delta R^{\text{y}}_{k,j} \prod_{j \in J_k \setminus S_{k,m}} (1 - R^0_{k,j}) \Big), \qquad k \in K. \end{aligned} \tag{35}$$

Finally, the original objective function (Eq. 16) can be replaced by its logarithm:

$$\tilde{R}_{\text{sys}} = \ln R_{\text{sys}} = \ln\Big(\prod_{k \in K} R'_k \Big) = \sum_{k \in K} \ln R'_k. \tag{36}$$

As logarithmic functions are always monotonic, maximizing $\tilde{R}_{\text{sys}}$ is equivalent to maximizing R_{sys}. Each term in the above summation (Eq. 35) is concave, and thus the new objective function is also concave. Maximizing a concave function is equivalent to minimizing a convex function.

The nonlinear equality constraint in Eq. 36 still has a non-convex feasible region. Nevertheless, as the left hand side of the constraint, $\tilde{R}_{\text{sys}}$, is our objective function (of the maximization type), we can relax the constraint to be an inequality constraint (less than or equal to)

$$\tilde{R}_{\text{sys}} \leq \sum_{k \in K} \ln R'_k. \tag{37}$$

As each term $\ln R'_k, k \in K$ is concave, the inequality constraint has a convex feasible region. Thus, the Convex Replacement (CR) model, which is a convex MINLP, is

$$\begin{aligned} \max_{\mathbf{y},p} \quad & \tilde{R}_{\text{sys},q} \\ \text{subject to} \quad & \text{Eqs. 17 - 19, 21, 33 - 35, 37.} \end{aligned} \tag{CR}$$

In Eq. 37, terms $\ln R'_k$, $k \in K$ approach infinity when $R'_k \to 0$. In order to avoid numerical problems, we define a lower bound of 10^{-8} for variables R'_k, $k \in K$ when implementing Model CR (this also applies later to models CRR and CRR2).

5.3. Results: replacement models

Let us now investigate the efficiency, as well as the goodness of the obtained solutions, when solving Models NCR and CR by global and non-global solvers on large-scale problems. Moreover, we study whether the inclusion of the preassignment (Eq. 28) improves the efficiency.

We study ten large-scale selective maintenance optimization problems, having $n = \{100, 200, \ldots, 1000\}$ components. In order to facilitate an easy generation of similar problems with a varying number of components, we define a basic arrangement of 100 components (Figure 6), in which $|J_1|, |J_2| = 1$, $|J_3| \ldots |J_8| = 2$, $|J_9| \ldots |J_{18}| = 3$, and $|J_{19}| \ldots |J_{32}| = 4$. This is the arrangement of the optimization problem with 100 components. We generate the component arrangements of the problems with $n \geq 200$ by aligning multiple basic arrangements in series. As an example, the optimization problem with 300 components consists of three of these basic arrangements and has, therefore, six stages with a single component, 18 stages with two parallel components, and so on.

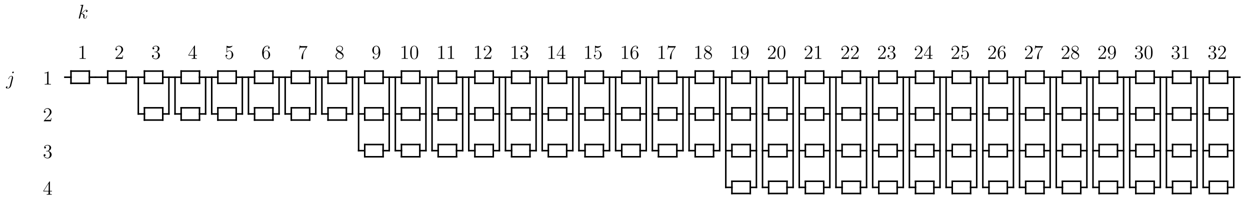


Figure 6: A basic arrangement of 100 components, used to define large-scale selective maintenance optimization problems. The arrangements consisting of 200 to 1000 components are generated by aligning two to ten, respectively, of these basic arrangements in series.

We again draw component types to the arrangements randomly from a component catalog, which we have here extended to consist of ten types (those listed in both Tables 6 and 11). We assume that the components have a failure behavior equivalent to that of Dataset 2, and use the failure model by Sarhan & Apaloo (2013) to generate the reliability parameters. Here, the cost of involving a maintenance person $c_{\text{person}} = 4$ kEUR, the duration of the maintenance break $T_{\text{break}} = 100$ h, and the planned next operation window $t_{\text{w}} = 30$ months. We draw component ages randomly from the range of $\{30, 60, \ldots, 330\}$ months, and choose randomly 20% of the components to be failed prior to the maintenance break.

Table 11: Component catalog (items VI - X).

Component type	VI	VII	VIII	IX	X
cost of replacement c^{y} [kEUR]	2	5.5	7.5	10	12
cost of replacement c^{x} [kEUR]	1.5	2	1	6	4
duration of replacement t^{y} [h]	5	7	11	8	12
duration of repair t^{x} [h]	9	2	5	15	6

When using the MINLP models, we approximate the Pareto front by solving optimization problems corresponding to 100 budget levels of the ϵ-constraint methods, such that $c_{\text{budget},100}$ is 2% more than that of the solution where all sensible replacements (for which $\Delta R^{\text{y}}_{k,j} > 0$) are conducted.

We solve Model NCR using both the global solver BARON 18.5.8 (with the relative optimality criterion of 10^{-6}) and the non-global solver DICOPT 2[9] (Bernal et al., 2019). It is to be noticed that the latter may not yield the global optimum for Model NCR. We solve Model CR using DICOPT 2. For brevity, we refer to DICOPT 2 simply as DICOPT in the remainder of this paper. As the model is convex, also the non-global solver is guaranteed to find the global optimum. The MINLP models are implemented in GAMS 25.1.3 software (GAMS Development Corporation, 2018). For each budget level of the ϵ-constraint method,

[9] In this work, unless otherwise stated, we use DICOPT 2 with solver parameters: *stop* 1, *infeasder* 1.

we define an upper computational time limit of 3600 seconds. Moreover, we generate reference results by the slightly modified version of the heuristic search algorithm by Lust et al. (2009). All results are generated on Intel(R) Core(TM) i5-7300U processor.

When using DICOPT, we use CONOPT 3.17I (Drud, 1994) as the nonlinear programming solver. As the corresponding mixed integer programming (MIP) solver, we tested both CPLEX 12.8.0.0 (IBM, 2018) and GUROBI 8.1.0 (Gurobi Optimization, LLC, 2019). Without the preassignment, CPLEX was more efficient than GUROBI on all ten optimization problem instances, and, with preassignment, on seven out of ten optimization problem instances. A detailed comparison of the computational times is presented in Appendix A. The differences in the optimized system reliability R_{sys} (when using the different MIP solvers) were insignificant, within 0.0134%. Therefore, we here report the results generated with CPLEX as the MIP solver, and use the same MIP solver later in Section 5.5.

Regarding the results, we monitor the relative differences in the optimized system reliability R_{sys} and the required computational time. These results are listed in Tables 12 and 13, respectively. Further, Figure 7 shows a graphical representation of the computational times. Experiments with each component arrangement size involve solving optimization problems with a number of budget levels, i.e. 100 when solving Model NCR or CR by DICOPT or BARON and a problem-specific number when using the heuristic search by Lust et al. (2009). Therefore, in order to enable a fair comparison, we report the average values of the relative differences in the optimized system reliability and computational times across the budget levels.

Table 12: The average relative difference (%) in the optimized system reliability R_{sys}. The reference results are those obtained by solving Model CR with DICOPT with the preassignment.

n	NCR / BARON	NCR / BARON / preassign.	CR / DICOPT	CR / DICOPT / preassign.	NCR / DICOPT /	NCR / DICOPT / preassign.	heuristic search (Lust et al., 2009)
100	0.0000	0.0000	0.0000	0.0000	-13.9622	-13.8441	-0.5772
200	-0.0000	-0.0006	0.0000	0.0000	-18.5247	-0.4511	-0.6166
300	0.0504	0.0505	0.0000	0.0000	-47.7391	-1.0091	-1.0515
400	0.0038	0.0034	0.0000	0.0000	-90.5637	-1.5098	-0.9297
500	0.0230	0.0262	0.0134	0.0000	-97.6209	-1.7325	-1.3383
600	0.0156	0.0123	-0.0071	0.0000	-97.7227	-10.2231	-1.3073
700	-0.0204	0.0082	-0.0000	0.0000	-99.9996	-67.7729	-1.4197
800	-0.1512	0.0003	-0.0010	0.0000	-99.9861	-9.9585	-1.3513
900	-	-	0.0001	0.0000	-100.0000	-100.0000	-1.1251
1000	-	-	0.0000	0.0000	-100.0000	-98.8919	-1.1022

Table 13: The average computational times (s) to generate one of the solutions approximating the Pareto front. The results are listed for optimization problems with varying number of components, n.

n	NCR / BARON	NCR / BARON / preassign.	CR / DICOPT	CR / DICOPT / preassign.	NCR / DICOPT /	NCR / DICOPT / preassign.	heuristic search (Lust et al., 2009)
100	0.35	0.27	0.27	0.08	0.02	0.02	0.01
200	0.80	0.65	1.35	0.29	0.03	0.03	0.05
300	1.49	1.03	1.30	0.30	0.04	0.04	0.09
400	40.46	3.94	3.42	0.26	0.03	0.04	0.16
500	20.13	6.21	6.02	0.62	0.02	0.06	0.27
600	121.16	92.54	9.29	0.65	0.02	0.06	0.41
700	360.53	60.68	11.99	0.97	0.02	0.06	0.52
800	1081.33	709.32	17.05	1.57	0.01	0.07	0.71
900	-	-	24.61	1.43	0.02	0.03	0.88
1000	-	-	122.11	2.73	0.08	0.02	1.09

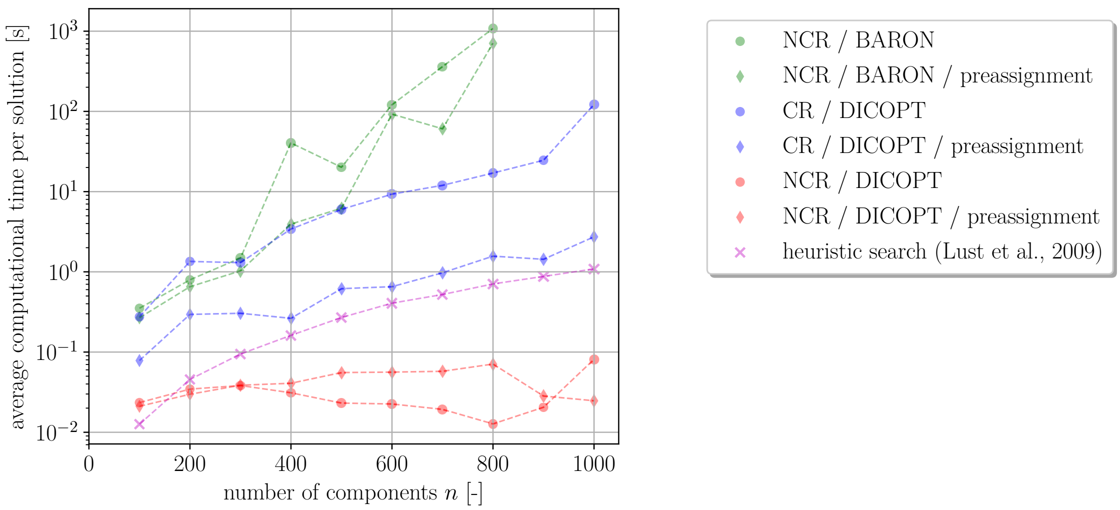


Figure 7: Average computational times to generate one solution lying at the (approximated) Pareto front. NCR and CR are abbreviations of the Non-Convex and Convex Replacement models. The dashed lines are added to the plot for better visualization. They do not represent values in between the points.

The relative differences in the system reliability are reported with respect to those obtained by solving Model CR by DICOPT with preassignment. When the system reliability R_{sys} is close to zero, even a very small absolute difference in the obtained results would cause a major relative difference, misleading the interpretation of the results. Therefore, we have filtered the lower end of the Pareto fronts where the reference system reliability $R_{\text{sys}} < 0.001$. Consequently, results of at most 10 out of 100 budget levels were filtered, which occurred on the system arrangement of $n = 1000$ components.

The system reliabilities obtained by solving Model NCR by BARON and Model CR by DICOPT, with or without preassignment, were on average within 0.1512% from each other (Table 12). This value occurs when comparing the results of solving Model NCR by BARON without the preassignment and the reference method on the problem involving 800 components (the worse results are obtained by the former approach). On this problem instance, the former approach is terminated prematurely on 17 out of 100 optimization runs, due to the computational timeout of 3600 s, which explains the sub-optimality of the results. The remaining results are on average within 0.0504% from each other.

The system reliabilities obtained by the heuristic search were, on average, 1.08% lower than the reference results. When solving NCR by DICOPT with or without preassignment, the algorithm has a tendency to converge to solutions where no replacements are conducted, which results in significantly lower system reliabilities. Consequently, the obtained reliabilities are, on average, 0.4511 to 100% lower that the reference results.

Both when solving Model NCR by BARON or Model CR by DICOPT, the inclusion of the preassignment, in general, reduces the required computational time. The reduction is more significant in the case of the latter, for which the difference is an order of magnitude, or more, for problems with ≥ 400 components. With preassignment, solving Model CR by DICOPT requires on average less computational time than solving Model NCR by BARON in all of the 10 studied component arrangements. On problems with ≥ 400 components, the difference is an order of magnitude or more, whereas, on smaller problems, it is around a factor of two. Without preassignment, solving Model CR by DICOPT requires on average less computational time than solving Model NCR by BARON for problems involving ≥ 400 components. For problems with less than 400 components, the computational times are similar.

At 800 components, the average computational time of solving a single budget level of Model NCR by BARON with or without preassignment is around 1000 s, which means that approximating the Pareto front by 100 budget levels requires around 27.8 h. Therefore, as the computational time presumably increases further, we have not solved Model NCR by BARON for problems involving more than 800 components.

The heuristic search algorithm by Lust et al. (2009) and solving Model NCR by DICOPT require less computational time than the above discussed approaches. However, these approaches do not necessarily yield the global optimum (Table 12). Moreover, solving Model NCR by DICOPT fails to find a solution other than the trivial solution (involving no maintenance actions) on many problem instances.

5.4. Convexification of the replacement-repair model

In Section 5.2, we convexified Model NCR by reformulating the multi-linear terms in Eq. 15. In this section, we reformulate Model NCRR by following the same principle. First, we revisit the original formulation of Eq. 23, defined as

$$R'_k = 1 - \prod_{j \in J_k} (1 - R^0_{k,j} - \Delta R^{\mathrm{y}}_{k,j} y_{k,j} - \Delta R^{\mathrm{x}}_{k,j} x_{k,j}), \quad \forall k \in K$$

In order to get rid of the multilinearity from the production, we define I_k as the index set of all possible partitions of J_k into the three subsets of repair ($S^{\mathrm{x}}_{k,i}$), replacement ($S^{\mathrm{y}}_{k,i}$), and no action ($J_k \setminus (S^{\mathrm{x}}_{k,i} \cup S^{\mathrm{y}}_{k,i})$):

$$I_k = \{i | S^x_{k,i} \subseteq J_k, S^{\mathrm{y}}_{k,i} \subseteq J_k, S^{\mathrm{x}}_{k,i} \cap S^{\mathrm{y}}_{k,i} = \emptyset\}, \quad \forall k \in K \tag{38}$$

Resembling the linearizing efforts for the replacement-only case, the partitions can be ordered with respect to ternary numbers. Table 14 shows the labeling for a stage with two units, where the indicator $\alpha_{j,k,i}$ in row j and column i being equal to 0 means that unit j belongs to the no-action subset in partition i, while 1 means repair, and 2 means replacement. For example, the two values in column $i = 6$ are 1 and 2, which put together to form 12, the ternary form of 5=6-1. The general formula for $\alpha_{j,k,i}$ is

$$\alpha_{j,k,i} = \left\lfloor \frac{\mathrm{mod}\ (i-1, 3^{|J_k|-j+1})}{3^{|J_k|-j}} \right\rfloor \quad \forall j \in J_k, k \in K, i \in I_k. \tag{39}$$

Table 14: An enumeration of set partitions.

	$\alpha_{j,k,i}$	$i \in I_k$: 1	2	3	4	5	6	7	8	9
$j \in J_k$	1	0	0	0	1	1	1	2	2	2
	2	0	1	2	0	1	2	0	1	2

Based on Eq. 38, the original formulation (Eq. 23) can be unfolded as

$$R'_k = 1 - \sum_{i \in I_k} \prod_{j \in J_k \setminus (S^{\mathrm{x}}_{k,i} \cup S^{\mathrm{y}}_{k,i})} (1 - R^0_{k,j}) \prod_{j \in S^{\mathrm{y}}_{k,i}} (-\Delta R^{\mathrm{y}}_{k,j} y_{k,j}) \prod_{j \in S^{\mathrm{x}}_{k,i}} (-\Delta R^{\mathrm{x}}_{k,j} x_{k,j}), \quad \forall k \in K. \tag{40}$$

Now, we introduce new binary variables $w_{k,i}$ and let

$$w_{k,i} = \prod_{j \in S^{\mathrm{x}}_{k,i}} x_{k,j} \prod_{j \in S^{\mathrm{y}}_{k,i}} y_{k,j}, \quad \forall i \in I_k, k \in K, \tag{41}$$

with which the formulation in Eq. 40 can be written as

$$R'_k = 1 - \sum_{i \in I_k} w_{k,i} [\prod_{j \in J_k \setminus (S^{\mathrm{x}}_{k,i} \cup S^{\mathrm{y}}_{k,i})} (1 - R^0_{k,j}) \prod_{j \in S^{\mathrm{y}}_{k,i}} (-\Delta R^{\mathrm{y}}_{k,j}) \prod_{j \in S^{\mathrm{x}}_{k,i}} (-\Delta R^{\mathrm{x}}_{k,j})], \quad \forall k \in K. \tag{42}$$

The multilinear term in Eq. 41 can be transformed into the following linear inequalities:

$$w_{k,i} \le x_{k,j}, \quad \forall i \in I_k, j \in S^{\mathrm{x}}_{k,i}, k \in K \tag{43}$$

$$w_{k,i} \leq y_{k,j}, \quad \forall i \in I_k, j \in S^{\mathrm{y}}_{k,i}, k \in K \tag{44}$$

$$w_{k,i} \geq \sum_{j \in S^{\mathrm{x}}_{k,i}} x_{k,j} + \sum_{j \in S^{\mathrm{y}}_{k,i}} y_{k,j} - |S^{\mathrm{x}}_{k,i}| - |S^{\mathrm{y}}_{k,i}| + 1, \quad \forall i \in I_k, k \in K. \tag{45}$$

Note that if $S^{\mathrm{x}}_{k,i}$ and/or $S^{\mathrm{y}}_{k,i}$ are empty sets, Eqs. 43 and/or 44 are redundant. For example, for $i \in I_k$ such that $S^{\mathrm{x}}_{k,i} = \varnothing$ and $S^{\mathrm{y}}_{k,i} = \varnothing$, $w_{k,i} \equiv 1$.

With that, we present the Convex Replacement-Repair model (CRR) as follows:

$$\begin{array}{ll} \max\limits_{\mathbf{x},\mathbf{y},p} & \tilde{R}_{\mathrm{sys},q} \\ \text{subject to} & \text{Eqs. 18, 21, 24 - 27, 37, 42 - 45.} \end{array} \tag{CRR}$$

Another way to express Eq. 41 as linear inequalities involving the new binary variables $w_{k,i}$ and the original ones $x_{k,j}$ and $y_{k,j}$ is shown in Appendix B. We refer to this model as the alternative Convex Replacement-Repair model (CRR2). The alternative model involves more binary variables, but has fewer inequalities for the cases where $|J_k| \geq 3$, and is tighter than the model presented in this section. However, based on our results, which we will presented in the next section, the average computational times of the two models seem similar. Due to this reason, and for the sake of readability, we have moved the description of the alternative model to the supplementary material.

5.5. *Results: replacement-repair models*

In this section, we return to the selective maintenance optimization problems, presented in Section 5.3, and solve them using replacement-repair models. We use both the non-convex Model NCRR and convexified Models CRR and CRR2, and, as a reference, the heuristic search by Lust et al. (2009). Again, we solve the non-convex model by both BARON and DICOPT, and convex models by DICOPT. All results are generated with and without the preassignment (Eq. 28). In Section 5.3, we studied ten optimization problem instances. As the computational cost of replacement-repair models is, in general, higher than that of replacement models, we report results only for the first seven optimization problem instances, involving 100 to 700 components.

Table 15 lists the average relative differences in the optimized system reliability R_{sys} in the obtained results, in which the reference results are those obtained by solving Model CRR by DICOPT with preassignment. The average relative differences obtained by solving Model NCRR by BARON with the preassignment and Models CRR and CRR2 by DICOPT with the preassignment are within 0.0413%. Solving Models CRR and CRR2 by DICOPT without preassignment is computationally expensive, and we were therefore only able to generate results for problem instances involving up to 400 components. However, for this problem size, the results were already on average 32.32 and 28.28%, respectively, worse than the reference results, due to multiple premature terminations caused by reaching the computational time limit.

The results obtained by solving Model NCRR by BARON without the preassignment were, on average, up to 0.6225% worse than the reference results. This occurred on the problem instance involving 700 components. In this case, eight out of 100 optimization runs were terminated due to the computational time limit of 3600 s, which seems to be the main reason causing sub-optimality in the results. Regarding the non-global optimization methods, the heuristic search and solving Model NCRR by DICOPT with the preassignment yield results that are on average 2.21 and 2.28%, respectively, worse than the corresponding reference results. Solving the non-convex Model NCRR by DICOPT with the preassignment yields more robust results than solving non-convex Model NCR with the same approach (Section 5.3). However, without the preassigment, the optimization runs again often converge to the trivial solution, involving no maintenance actions.

Table 16 shows the average computational times of tested approaches on the seven optimization problem instances. Figure 8 shows a graphical representation of the same results. As we already indicated in Section 5.4, the computational times of solving CRR and CRR2 by DICOPT with or without the preassignment

Table 15: The average relative difference (%) in the optimized system reliability R_{sys}. The reference results are those obtained by solving Model CRR with DICOPT with the preassignment.

number of components n	NCRR / BARON	NCRR / BARON / preassign.	CRR / DICOPT	CRR / DICOPT / preassign.	CRR2 / DICOPT /	CRR2 / DICOPT / preassign.	NCRR / DICOPT /	NCRR / DICOPT / preassign.	heuristic search (Lust et al., 2009)
100	-0.0103	0.0000	0.0000	0.0000	0.0000	0.0000	-5.1079	-3.1708	-1.3722
200	-0.0344	-0.0001	0.0000	0.0000	0.0000	0.0000	-24.4838	-0.0601	-0.5163
300	-0.1343	0.0413	-0.0000	0.0000	-0.0000	0.0000	-70.1091	-0.0467	-1.2762
400	-0.1201	0.0051	-32.3218	0.0000	-28.2843	0.0000	-73.6336	-0.0193	-1.2242
500	-0.1314	0.0003	-	0.0000	-	-0.0035	-96.7744	-2.2350	-7.1207
600	-0.1471	0.0323	-	0.0000	-	0.0000	-97.5206	-3.1210	-2.4324
700	-0.6225	-0.0104	-	0.0000	-	0.0000	-99.8366	-7.2751	-1.5015

are similar. In both cases, the inclusion of the preassignment reduces the average computational time by around an order of magnitude. When solving Model NCRR by BARON, the inclusion of the preassignment, in general, slightly enhances the efficiency; however, the opposite result is obtained on problem instances having 300 and 700 components.

Table 16: The average computational times (s) to generate one of the solutions (approximating) the Pareto front.

number of components n	NCRR / BARON	NCRR / BARON / preassign.	CRR1 / DICOPT	CRR1 / DICOPT / preassign.	CRR2 / DICOPT /	CRR2 / DICOPT / preassign.	NCRR / DICOPT /	NCRR / DICOPT / preassign.	heuristic search (Lust et al., 2009)
100	0.49	0.48	4.34	0.36	3.47	0.43	0.04	0.03	0.04
200	1.24	1.09	172.23	11.48	80.05	9.64	0.06	0.06	0.14
300	3.06	2.48	79.06	3.97	43.30	4.72	0.09	0.08	0.30
400	23.06	50.15	1027.87	6.46	1101.91	7.73	0.08	0.09	0.52
500	32.70	13.25	-	134.72	-	117.44	0.06	0.12	0.86
600	259.19	120.63	-	22.56	-	25.66	0.07	0.14	1.44
700	651.98	711.82	-	67.68	-	71.85	0.08	0.15	1.72

Opposite to the replacement models (Section 5.5), here the convexification does not improve the efficiency in the studied problem size range. With the preassignment, the solution times of solving the convex Models CRR and CRR2 by DICOPT are similar to solving the non-convex Model NCRR by BARON. Without the preassignment, the former approaches have worse efficiency than the latter. Regarding the non-global approaches, solving Model NCRR by DICOPT with the preassignment[10] requires less computational time than the heuristic search.

Finally, let us examine the results of a representative problem instance, containing 300 components. Figure 9 visualizes the obtained discretized Pareto fronts using both Models CR and CRR, as well as representative solutions on the Pareto fronts. Despite the larger scale, similar features are also visible here as earlier in the results of the illustrative example (Figure 4). First, as Model CRR includes both the replacement and repair actions, its Pareto optimal solutions dominate those of Model CR, where only the replacement action is included. Second, in representative solution (5), all failed components and functioning components, for which $\Delta R_{\mathrm{y}} > 0$, are replaced. Third, representative solution (4) is otherwise the same as representative solution (5), but all failed components, for which $\Delta R_{\mathrm{y}} < 0$, are repaired, instead of being replaced.

5.6. Additional remarks

Warm start. When generating the Pareto front using the ϵ-constraint method, the final solution of budget level q is a feasible and presumably good initial solution for the optimization run at budget level $q+1$. Let us refer to this initialization strategy as the *warm start*. In integer programming, a good initial solution has the potential to improve the solution efficiency, as regions with less fit objective function values can be eliminated from the search space early in the process. We tested the warm start with some solution approaches, but it provided only minor or no improvement to the solution efficiency.

[10] Comparing the same approach without the preassignment is irrelevant because the approach is not robust (see Table 15).

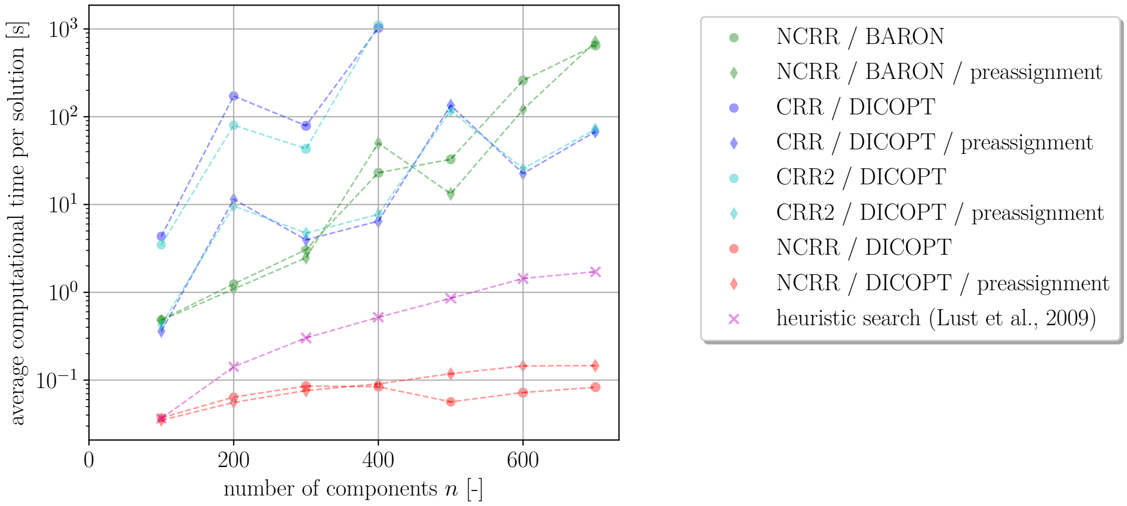


Figure 8: Average computational times to generate one solution lying at the (approximated) Pareto front. NCRR is abbreviation of the Non-Convex Replacement-Repair models, and CRR and CRR2 for the first and second Convex Replacement models.

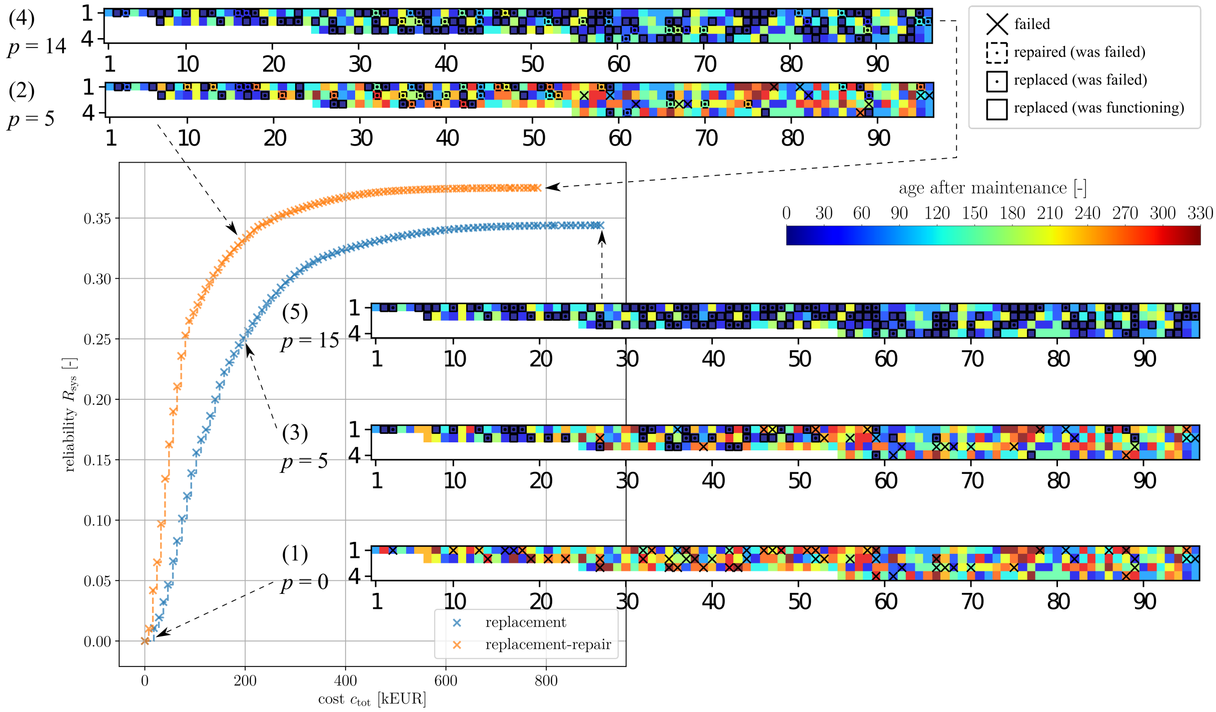


Figure 9: The discretized Pareto fronts of the problem with 300 components, using both Models CR (replacement) and CRR (replacement-repair). Representative solutions are plotted along with the number of maintenance personnel p.

The augmented penalty in DICOPT. The system reliabilities obtained solving the non-convex Models NCR and NCRR by BARON are, on average, at most 0.0505 and 0.0413%, respectively, higher than solving the corresponding convex Models CR and CRR by DICOPT (all with the preassignment). Both occurred with the problem instance of 300 components (see Tables 13 and 16). As DICOPT is expected to solve convex non-linear optimization problems to the optimality (within the machine precision), we further investigated the reason for sub-optimal results on these two problems instances.

DICOPT is a combination of the Outer-approximation method, equality relaxation and augmented penalty (Viswanathan & Grossmann, 1990). Two reasons lead to DICOPT yielding suboptimal results on some problem instances: the slack variables that the augmented penalty introduces to the optimization problem and the 'no-good' cuts introduced at each iteration to remove previously found solutions. In the augmented penalty approach, the linear outer-approximations of the nonlinear constraints are relaxed by adding a positive slack to each new inequality. These slacks are minimized together with the original objective function by penalizing its weighted sum, using as a coefficient 1000 (controlled by the solver parameter *weight*) times the marginal of the original nonlinear constraint. In the case that the original constraint is not active, then its corresponding slack variable is not included in the objective, and in the limit where the inequality is defining the subproblem solution, the penalization in the objective tends to transform the relaxed cut into a hard constraint. This heuristic has been useful for obtaining feasible solutions to non-convex MINLP problems but may lead to convergence to suboptimal solutions for convex MINLP problems if the optimal solution of the objective function plus the penalized objective is not the same as the one of the original objective function. The 'no-good' cuts or integer cuts introduce an extra inequality in terms of the integer variables and their values in previous iterations such that any solution yielding the same values for the integer variables is infeasible. If the integer variables in the problem are exclusively binary, the 'no-good' cut is a single inequality, while with general integer variables like the ones involved in these problem it is required to add up to two extra variables per 'no-good' cut, per integer variable (see Appendix 1 of Ref. Bernal et al. (2019)). This 'no-good' cuts are not required for convex MINLP problems, but are used as a heuristic for non-convex MINLP problems.

When we solved the two problem instances with 300 components by DICOPT with the augmented penalty suppressed[11] the average relative differences were only -0.0003 and 0.00005%, respectively, which supports our hypothesis. Finding these issues led to the introduction of new solver options for DICOPT[12] and the improvement of the solver, where the suboptimality issue was resolved. The improvements on DICOPT are available in GAMS since version 29.1.

6. Discussion

In this work, we convexified both the replacement and replacement-repair models NCR and NCRR, respectively. In the former, the solution efficiency improved, whereas, in the latter, it became worse without the preassignment and was similar with the preassignment. Presumably, the main reason for this is that the number of new binary variables, introduced by the convexification, increases at different rates in these models with respect to the number of components in the system. The number of new binary variables is $\sum_{k\in K} 2^{|J_k|}$ in Model CR, $\sum_{k\in K} 3^{|J_k|}$ in Model CRR, and $\sum_{k\in K} 3^{|J_k|} + 2\sum_{k\in K} 2^{|J_k|}$ in Model CRR2, where $|J_k|$ is the number of parallel components at stage k of the system.

Except for the preassigment, the approaches we present in this work are also applicable to systems where the components have an increasing failure rate. The preassignment is not applicable because for such components the improvement in reliability if replaced, ΔR_{y}, is always positive. The results where the preassigment is not used provide a rough indication of the efficiency of the approaches on such systems. It is also worth noticing that, when the failure rates are bathtub-shaped, the length of the next operation window t_{w} and the age distribution of the components are likely to have an effect on how much the preassigment improves the efficiency. Presumably, the more components lie in the non-sensible region (Figs. 1(c), 1(d), 2(c) and 2(d)), the more the preassigment enhances the efficiency.

[11]Using solver parameters: *stop* 1, *infeasder* 1, *feaspump* 1, *fp_cutoffdecr* 1e-6, *fp_iterlimit* 100, *fp_stalllimit* 100, *fp_integercuts* 0, *fp_softcuts* 0.

[12]If the solver parameter *weight* is set above 1E20, the augmented penalty approach is not used and no slack variables are introduced.

In comparison to the literature (see Section 1), we have expanded the largest problems reported, and solved to the optimality, from 200 to 1000 components in the case of one maintenance action and from 28 to 700 in the case of two maintenance actions. In our experiments, the average computational time per a solution in the former is 122.11 s (Model CR solved by DICOPT) and in the latter 652.0 s (Model NCRR solved by BARON). Here, we have listed the average computational times without preassignment because it would not be applicable to the optimization problems in the reference studies (these studies do not consider bathtub-shaped failure rates).

From the industrial perspective, our approaches facilitate the optimal decision-making of maintenance actions on engineering systems with more repairable or replaceable components than the earlier reported approaches. Alternatively, the extended number of components can be used to increase the detail of the maintenance actions. For example, instead of modeling the maintenance of an electrical motor as a whole, its individual repairable and/or replaceable subcomponents (e.g. the fan, insulation, and bearings) can be modeled individually. Based on the results, our recommendation for selective maintenance problems, involving only one maintenance action, is to use the convex Model CR, or its variation, and solve it by DICOPT. For problems involving two maintenance actions, our recommendation is to use the non-convex Model NCRR and solve it by BARON. If global optimality is not required, the heuristic search algorithm by Lust et al. (2009) is a robust choice, which in our experiments yielded for the replacement and replacement-repair models, on average, 1.08 and 2.21% sub-optimal results, respectively, with shorter computational time than global optimization approaches.

In the literature, the level of detail in selective maintenance optimization models has been expanded via features, such as imperfect maintenance (Liu & Huang, 2010), repair personnel assignment (Khatab et al., 2018), and serial n-out-of-k systems (Diallo et al., 2018) (see the introduction for more details). Models with more than two maintenance actions (i.e., including at least one imperfect maintenance action) could also be convexified using similar approaches to those we presented in this work. However, this would further increase the number of additional binary variables. Considering the solution strategies we have studied in this work, presumably, a better strategy for such models is to solve the original non-convex model by BARON. This prediction is based on an assumption that the trend in the relative performance between solving a non-convex model by BARON and a corresponding convex model by DICOPT remains the same when the number of maintenance action types is more than two.

Regarding the repair personnel assignment, we have included the total number of maintenance personnel as a binary variable p in our models (see Eqs. 18 and 19/27). However, for the sake of simplicity, our models do not explicitly assign individual maintenance persons to maintenance tasks. Nevertheless, such assignment (see the paper by Khatab et al. (2018)) could also be included in the convexified models via the inclusion of a new index r in variables $y_{k,j}$ and $x_{k,j}$, assigning the maintenance person r to the task (k, j), and the modification of Eq. 31 or 41 to have summations over the index r. Future work should investigate whether models describing serial n-out-of-k systems can be convexified in a similar way as those in this work describing serial-parallel systems. The preassignment of insensible component replacements is also applicable to models with of these features, as long as the components have a bathtub-shaped failure rate.

Finally, in this work, we have assumed that the failure data of components are available. Moreover, we assumed that the components of the system have identical failure behavior. In reality, system components, especially those located at different system stages, are likely to have different failure behavior. Collecting the failure data of all different component types in the system is a challenging and time-consuming task, especially if accelerated lifetime tests are not applicable. For example, the lifetime of pumps or drives, used in a chemical production plant, may be more than ten years. Therefore, collecting a dataset extensive enough, in terms of both the number of data points and the right-censoring time, for selective maintenance optimization may take many years. In some cases, the information of the failure rate may be obtained from the component supplier. However, this information might be based on an experiment conducted in a different operating environment or limited to only the warranty period of the component.

In Section 4.3, we demonstrated how optimal maintenance actions differ already if the failure rates are determined from the same dataset using different bathtub-shaped failure models. Future work should investigate what is the sensitivity of maintenance decisions to the number of data points in the dataset, and how long the components in the dataset should be operated (i.e. at what age a data point may be right-censored).

7. Conclusions

In this paper, we first linked bathtub-shaped failure rate models to selective maintenance optimization. Our sensitivity study shows that even if we start from the same failure data, but use different bathtub-shaped failure rate models (Jiang, 2013; Sarhan & Apaloo, 2013) (which, in the literature, are both considered to be suitable for the failure datasets studied in this work), the objective function space changes such that clearly different selective maintenance decisions become optimal. This highlights the importance of carefully fitting a suitable failure model to the failure data.

Second, in order to enhance the solution efficiency, we convexified selective maintenance optimization models, including 1) only replacement or 2) both replacement and repair actions. Moreover, we derived a preassignment of variables corresponding to components, the replacement of which would undesirably reduce the component-specific reliability (the reduction is caused by the infant mortality period of the bathtub-shaped failure rate). Such components can be identified prior to the optimization procedure using our data analysis method. In our experiments, the inclusion of the preassignment in the convexified models CR, CRR and CRR2 reduced the solution time by roughly an order of magnitude when using the non-global solver DICOPT. When solving non-convex Models NCR and NCRR by the global solver BARON, we observed, in general, similar behavior but with smaller reduction in the computational times. With the preassignment, solving the convexified replacement Model CR by DICOPT requires significantly less computational time than solving the equivalent non-convex Model NCR by BARON – the difference being an order of magnitude or more for problems involving ≥ 400 components. In the corresponding comparison of the models including also the repair action, the convexification did not reduce the computational time but the times were similar.

We demonstrated the approaches presented in this paper on selective maintenance optimization problems consisting of up to 1000 system components, when only the replacement action is included, and up to 700 system components, when both replacement and repair actions are included.

Acknoledgements

Financial support from the Academy of Finland, through project SINGPRO (Decision No. 313466), and the Center for Advanced Process Decision-making (CAPD) at Carnegie Mellon University are gratefully acknowledged. In addition, the authors would like to thank Simo Säynevirta, Juha Alamäki and Olli Alkkiomäki from ABB Finland for their valuable industrial perspective to the optimization problem definition.

Appendix A. Comparison of the solution times with CPLEX and GUROBI

Table A.17 lists the numerical results when solving the Model CR by DICOPT with CPLEX 12.8.0.0 or GUROBI 8.1.0 as the MIP solver. The lower solution times with and without the preassignment are highlighted by the bold font.

Appendix B. Alternative convexification of the non-convex replacement-repair model (NCRR)

Let us start by defining M_k as the index set of the subsets of J_k:

$$M_k = \{m | S_{k,m} \subseteq J_k\}, \quad \forall k \in K$$

We introduce new binary variables $u_{k,m}$ and $v_{k,m}$, such that

$$u_{k,m} = \prod_{j \in S_{k,m}} x_{k,j}, \quad \forall m \in M_k, k \in K \tag{B.1}$$

Table A.17: Comparison of average solution times, in seconds, of Model CR using DICOPT with CPLEX or GUROBI as the MIP solver.

	without preassignment		with preassignment	
n	CPLEX	GUROBI	CPLEX	GUROBI
100	**0.27**	0.40	0.08	**0.07**
200	**1.35**	2.24	**0.29**	0.48
300	**1.30**	2.53	**0.30**	0.35
400	**3.42**	5.60	0.26	**0.20**
500	**6.02**	10.32	**0.62**	0.76
600	**9.29**	27.93	0.65	0.65
700	**11.99**	47.64	**0.97**	1.63
800	**17.05**	60.16	**1.57**	2.83
900	**24.61**	89.83	**1.43**	2.29
1000	**122.11**	141.03	**2.73**	3.91

$$v_{k,m} = \prod_{j \in S_{k,m}} y_{k,j}, \quad \forall m \in M_k, k \in K. \tag{B.2}$$

Based on the definition of $w_{k,i}$ in Eq. 41, we have

$$w_{k,i} = u_{k,m} v_{k,m'}, \quad \forall i \in I_k, S_{k,m} = S^x_{k,i}, S_{k,m'} = S^y_{k,i}, k \in K \tag{B.3}$$

Therefore, alternatively, we can use of the relationships described in Eqs. B.1, B.2 and B.3 and transform them into the following linear inequalities:

$$u_{k,m} \leq x_{k,j}, \quad \forall m \in M_k, j \in S_{k,m}, k \in K \tag{B.4}$$

$$u_{k,m} \geq \sum_{j \in S_{k,m}} x_{k,j} - |S_{k,m}| + 1, \quad \forall m \in M_k, k \in K \tag{B.5}$$

$$v_{k,m} \leq y_{k,j}, \quad \forall m \in M_k, j \in S_{k,m}, k \in K \tag{B.6}$$

$$v_{k,m} \geq \sum_{j \in S_{k,m}} y_{k,j} - |S_{k,m}| + 1, \quad \forall m \in M_k, k \in K \tag{B.7}$$

$$w_i \leq u_{k,m}, \quad \forall i \in I_k, S_{k,m} = S^x_{k,i}, k \in K \tag{B.8}$$

$$w_i \leq v_{k,m}, \quad \forall i \in I_k, S_{k,m} = S^y_{k,i}, k \in K \tag{B.9}$$

$$w_i \geq u_{k,m} + v_{k,m'} - 1, \quad \forall i \in I_k, S_{k,m} = S^x_{k,i}, S_{k,m'} = S^y_{k,i}, k \in K \tag{B.10}$$

Similar to the formulation in Section 5.4, constraints over empty sets do not apply. For example, for $m \in M_k$ such that $S_{k,m} = \varnothing$, we have both $u_{k,m} \equiv 1$ and $v_{k,m} \equiv 1$.

The alternative Convex Replacement-Repair model (CRR2) is defined as

$$\begin{aligned} \max_{\mathbf{x},\mathbf{y},p} \quad & \tilde{R}_{\text{sys},q} \\ \text{subject to} \quad & \text{Eqs. 18, 21, 24 - 27, 37, 42, B.4 - B.10.} \end{aligned} \tag{CRR2}$$

The corresponding summation of (B.4) and (B.8) implies (43). The corresponding summation of (B.6) and (B.9) implies (44). The corresponding summation of (B.5), (B.7), and (B.10) implies (45). Therefore, the linear relaxation of (B.4) - (B.10) is at least as tight as that of (43) - (45). In other words, any point (integral or fractional) that satisfies (B.4) - (B.10) will satisfy (43) - (45).

References

Aarset, M. V. (1987). How to identify a bathtub hazard rate. *IEEE Transactions on Reliability*, *36*, 106–108. doi:https://doi.org/10.1109/TR.1987.5222310.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*, 716–723. doi:https://doi.org/10.1109/TAC.1974.1100705.

Amaran, S., Sahinidis, N. V., Sharda, B., Morrison, M., Bury, S. J., Miller, S., & Wassick, J. M. (2015). Long-term turnaround planning for integrated chemical sites. *Computers & Chemical Engineering*, *72*, 145–158. doi:https://doi.org/10.1016/j.compchemeng.2014.08.003.

Belotti, P., Kirches, C., Leyffer, S., Linderoth, J., Luedtke, J., & Mahajan, A. (2013). Mixed-integer nonlinear optimization. *Acta Numerica*, *22*, 1–131. doi:https://doi.org/10.1017/S0962492913000032.

Bernal, D. E., Vigerske, S., Trespalacios, F., & Grossmann, I. E. (2019). Improving the performance of DICOPT in convex MINLP problems using a feasibility pump. *Optimization Methods and Software*, (pp. 1–20). doi:https://doi.org/10.1080/10556788.2019.1641498.

Biondi, M., Sand, G., & Harjunkoski, I. (2017). Optimization of multipurpose process plant operations: A multi-time-scale maintenance and production scheduling approach. *Computers & Chemical Engineering*, *99*, 325–339. doi:https://doi.org/10.1016/j.compchemeng.2017.01.007.

Cao, W., Jia, X., Hu, Q., Zhao, J., & Wu, Y. (2018). A literature review on selective maintenance for multi-unit systems. *Quality and Reliability Engineering International*, *34*, 824–845. doi:https://doi.org/10.1002/qre.2293.

Cassady, C. R., Murdock Jr, W. P., & Pohl, E. A. (2001a). Selective maintenance for support equipment involving multiple maintenance actions. *European Journal of Operational Research*, *129*, 252–258. doi:https://doi.org/10.1016/S0377-2217(00)00222-8.

Cassady, C. R., Pohl, E. A., & Murdock Jr, W. P. (2001b). Selective maintenance modeling for industrial systems. *Journal of Quality in Maintenance Engineering*, *7*, 104–117. doi:https://doi.org/10.1108/13552510110397412.

Certa, A., Galante, G., Lupo, T., & Passannanti, G. (2011). Determination of pareto frontier in multi-objective maintenance optimization. *Reliability Engineering & System Safety*, *96*, 861–867. doi:https://doi.org/10.1016/j.ress.2010.12.019.

Diallo, C., Venkatadri, U., Khatab, A., & Liu, Z. (2018). Optimal selective maintenance decisions for large serial k-out-of-n: G systems under imperfect maintenance. *Reliability Engineering & System Safety*, *175*, 234–245. doi:https://doi.org/10.1016/j.ress.2018.03.023.

Diallo, C., Venkatadri, U., Khatab, A., Liu, Z., & Aghezzaf, E.-H. (2019). Optimal joint selective imperfect maintenance and multiple repairpersons assignment strategy for complex multicomponent systems. *International Journal of Production Research*, *57*, 4098–4117. doi:https://doi.org/10.1080/00207543.2018.1505060.

Drud, A. S. (1994). CONOPT – a large-scale GRG code. *ORSA Journal on computing*, *6*, 207–216. doi:http://doi.org/10.1287/ijoc.6.2.207.

Duan, C., Deng, C., Gharaei, A., Wu, J., & Wang, B. (2018). Selective maintenance scheduling under stochastic maintenance quality with multiple maintenance actions. *International Journal of Production Research*, *56*, 7160–7178. doi:https://doi.org/10.1080/00207543.2018.1436789.

Duran, M. A., & Grossmann, I. E. (1986). An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical programming*, *36*, 307–339. doi:https://doi.org/10.1007/BF02592064.

El-Gohary, A., Alshamrani, A., & Al-Otaibi, A. N. (2013). The generalized gompertz distribution. *Applied Mathematical Modelling*, *37*, 13–24. doi:https://doi.org/10.1016/j.apm.2011.05.017.

Galante, G., & Passannanti, G. (2009). An exact algorithm for preventive maintenance planning of series–parallel systems. *Reliability Engineering & System Safety*, *94*, 1517–1525. doi:https://doi.org/10.1016/j.ress.2009.02.009.

GAMS Development Corporation (2018). General Algebraic Modeling System (GAMS) Release 25.1.3. Fairfax, VA, USA. URL: http://www.gams.com/.

Geoffrion, A. M. (1972). Generalized benders decomposition. *Journal of optimization theory and applications*, *10*, 237–260. doi:https://doi.org/10.1007/BF00934810.

Glover, F. (1989). Tabu search–part I. *ORSA Journal on computing*, *1*, 190–206. doi:https://doi.org/10.1287/ijoc.1.3.190.

Glover, F., & Woolsey, E. (1974). Converting the 0-1 polynomial programming problem to a 0-1 linear program. *Operations research*, *22*, 180–182. doi:https://doi.org/10.1287/opre.22.1.180.

Gompertz, B. (1825). On the nature of the function expressive of the law of human mortality, and on a new mode of determining the value of life contingencies. *Philosophical transactions of the Royal Society of London*, *115*, 513–583. doi:https://doi.org/10.1098/rstl.1825.0026.

Grossmann, I. E. (2002). Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and engineering*, *3*, 227–252. doi:https://doi.org/10.1023/A:1021039126272.

Gurobi Optimization, LLC (2019). Gurobi optimizer reference manual, version 8.1.

Haimes, Y. V., Lasdon, L. S., & Wismer, D. A. (1971). On a bicriterion formation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man and Cybernetics*, (pp. 296–297). doi:10.1109/TSMC.1971.4308298.

IBM (2018). IBM ILOG CPLEX optimization studio, version 12.8.

Iyoob, I. M., Cassady, C. R., & Pohl, E. A. (2006). Establishing maintenance resource levels using selective maintenance. *The Engineering Economist*, *51*, 99–114. doi:https://doi.org/10.1080/00137910600695627.

Jiang, R. (2013). A new bathtub curve model with a finite support. *Reliability Engineering & System Safety*, *119*, 44–51. doi:https://doi.org/10.1016/j.ress.2013.05.019.

Kettelle Jr, J. D. (1962). Least-cost allocations of reliability investment. *Operations Research*, *10*, 249–265. doi:https://doi.org/10.1287/opre.10.2.249.

Khatab, A., & Aghezzaf, E.-H. (2016). Selective maintenance optimization when quality of imperfect maintenance actions are stochastic. *Reliability engineering & system safety*, *150*, 182–189. doi:https://doi.org/10.1016/j.ress.2016.01.026.

Khatab, A., Aghezzaf, E.-H., Djelloul, I., & Sari, Z. (2017). Selective maintenance optimization for systems operating missions and scheduled breaks with stochastic durations. *Journal of Manufacturing Systems*, *43*, 168–177. doi:https://doi.org/10.1016/j.jmsy.2017.03.005.

Khatab, A., Ait-Kadi, D., & Nourelfath, M. (2007). Heuristic-based methods for solving the selective maintenance problem in series-parallel systems. In *International Conference on Industrial Engineering and Systems Management, Beijing, China*. URL: https://www.airitilibrary.com/Publication/alDetailedMesh?docid=WFHYXW318730.

Khatab, A., Diallo, C., Venkatadri, U., Liu, Z., & Aghezzaf, E.-H. (2018). Optimization of the joint selective maintenance and repairperson assignment problem under imperfect maintenance. *Computers & Industrial Engineering*, *125*, 413–422. doi:https://doi.org/10.1016/j.cie.2018.09.012.

Kijima, M. (1989). Some results for repairable systems with general repair. *Journal of Applied probability*, *26*, 89–102. doi:https://doi.org/10.2307/3214319.

Kijima, M., Morimura, H., & Suzuki, Y. (1988). Periodical replacement problem without assuming minimal repair. *European Journal of Operational Research*, *37*, 194–203. doi:https://doi.org/10.1016/0377-2217(88)90329-3.

Kondili, E., Pantelides, C. C., & Sargent, R. W. H. (1993). A general algorithm for short-term scheduling of batch operations-–I. MILP formulation. *Computers & Chemical Engineering*, *17*, 211–227. doi:https://doi.org/10.1016/0098-1354(93)80015-f.

Kraft, D. (1988). A software package for sequential quadratic programming. *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt*, .

Kronqvist, J., Bernal, D. E., Lundell, A., & Grossmann, I. E. (2019). A review and comparison of solvers for convex MINLP. *Optimization and Engineering*, *20*, 397–455. doi:https://doi.org/10.1007/s11081-018-9411-8.

Liu, Y., Chen, Y., & Jiang, T. (2018). On sequence planning for selective maintenance of multi-state systems under stochastic maintenance durations. *European Journal of Operational Research*, *268*, 113–127. doi:https://doi.org/10.1016/j.ejor.2017.12.036.

Liu, Y., Chen, Y., & Jiang, T. (2020). Dynamic selective maintenance optimization for multi-state systems over a finite horizon: A deep reinforcement learning approach. *European Journal of Operational Research*, *283*, 166–181. doi:https://doi.org/10.1016/j.ejor.2019.10.049.

Liu, Y., & Huang, H.-Z. (2010). Optimal selective maintenance strategy for multi-state systems under imperfect maintenance. *IEEE Transactions on Reliability*, *59*, 356–367. doi:https://doi.org/10.1109/TR.2010.2046798.

Lust, T., Roux, O., & Riane, F. (2009). Exact and heuristic methods for the selective maintenance problem. *European Journal of Operational Research*, *197*, 1166–1177. doi:https://doi.org/10.1016/j.ejor.2008.03.047.

Maillart, L. M., Cassady, C. R., Rainwater, C., & Schneider, K. (2009). Selective maintenance decision-making over extended planning horizons. *IEEE Transactions on Reliability*, *58*, 462–469. doi:https://doi.org/10.1109/TR.2009.2026689.

Massey Jr, F. J. (1951). The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, *46*, 68–78. doi:https://doi.org/10.1080/01621459.1951.10500769.

Meeker, W. Q., & Escobar, L. A. (1998). *Statistical methods for reliability data*. John Wiley & Sons.

Méndez, C. A., Henning, G. P., & Cerdá, J. (2001). An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities. *Computers & Chemical Engineering*, *25*, 701–711. doi:https://doi.org/10.1016/s0098-1354(01)00671-8.

Mudholkar, G. S., & Srivastava, D. K. (1993). Exponentiated weibull family for analyzing bathtub failure-rate data. *IEEE transactions on reliability*, *42*, 299–302. doi:https://doi.org/10.1109/24.229504.

Pandey, M., Zuo, M. J., & Moghaddass, R. (2013a). Selective maintenance modeling for a multistate system with multistate components under imperfect maintenance. *Iie Transactions*, *45*, 1221–1234. doi:https://doi.org/10.1080/0740817x.2012.761371.

Pandey, M., Zuo, M. J., Moghaddass, R., & Tiwari, M. K. (2013b). Selective maintenance for binary systems under imperfect repair. *Reliability Engineering & System Safety*, *113*, 42–51. doi:https://doi.org/10.1016/j.ress.2012.12.009.

Pantelides, C. C. (1994). Unified frameworks for optimal process planning and scheduling. In *Proceedings on the second conference on foundations of computer aided operations* (pp. 253–274). Cache Publications New York.

Rajagopalan, R., & Cassady, C. R. (2006). An improved selective maintenance solution approach. *Journal of Quality in Maintenance Engineering*, *12*, 172–185. doi:https://doi.org/10.1108/13552510610667183.

Raman, R., & Grossmann, I. E. (1991). Relation between MILP modelling and logical inference for chemical process synthesis. *Computers & Chemical Engineering*, *15*, 73–84. doi:https://doi.org/10.1016/0098-1354(91)87007-V.

Rice, W. F., Cassady, C. R., & Nachlas, J. A. (1998). Optimal maintenance plans under limited maintenance time. In *Proceedings of the seventh industrial engineering research conference* (pp. 1–3).

Sarhan, A. M., & Apaloo, J. (2013). Exponentiated modified weibull extension distribution. *Reliability Engineering & System Safety*, *112*, 137–144. doi:https://doi.org/10.1016/j.ress.2012.10.013.

Schneider, K., & Cassady, C. R. (2004). Fleet performance under selective maintenance. In *Annual Symposium Reliability and Maintainability, 2004-RAMS* (pp. 571–576). IEEE. doi:https://doi.org/10.1109/rams.2004.1285508.

Schneider, K., & Cassady, C. R. (2015). Evaluation and comparison of alternative fleet-level selective maintenance models. *Reliability Engineering & System Safety*, *134*, 178–187. doi:https://doi.org/10.1016/j.ress.2014.10.017.

Tawarmalani, M., & Sahinidis, N. V. (2005). A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, *103*, 225–249. doi:https://doi.org/10.1007/s10107-005-0581-8.

Vieira, M., Pinto-Varela, T., & Barbosa-Póvoa, A. P. (2017). Production and maintenance planning optimisation in biopharmaceutical processes under performance decay using a continuous-time formulation: A multi-objective approach. *Computers & Chemical Engineering*, *107*, 111–139. doi:https://doi.org/10.1016/j.compchemeng.2017.03.008.

Viswanathan, J., & Grossmann, I. E. (1990). A combined penalty function and outer-approximation method for MINLP optimization. *Computers & Chemical Engineering*, *14*, 769–782. doi:https://doi.org/10.1016/0098-1354(90)87085-4.

Weibull, W. (1951). Statistical distribution function of wide applicability. *Journal of applied mechanics*, *18*, 293–297.

Westerlund, T., & Pettersson, F. (1995). An extended cutting plane method for solving convex MINLP problems. *Computers & Chemical Engineering*, *19*, 131–136. doi:https://doi.org/10.1016/0098-1354(95)87027-x.

Wu, O., Dalle Ave, G., Harjunkoski, I., Bouaswaig, A., Schneider, S. M., Roth, M., & Imsland, L. (2020). Optimal production and maintenance scheduling for a multiproduct batch plant considering degradation. *Computers & Chemical Engineering*, (p. 106734). doi:https://doi.org/10.1016/j.compchemeng.2020.106734.

Xie, M., Tang, Y., & Goh, T. N. (2002). A modified weibull extension with bathtub-shaped failure rate function. *Reliability Engineering & System Safety*, *76*, 279–285. doi:https://doi.org/10.1016/s0951-8320(02)00022-4.

Ye, Y., Grossmann, I. E., & Pinto, J. M. (2018). Mixed-integer nonlinear programming models for optimal design of reliable chemical plants. *Computers & Chemical Engineering*, *116*, 3–16. doi:https://doi.org/10.1016/j.compchemeng.2017.08.013.

Ye, Y., Grossmann, I. E., Pinto, J. M., & Ramaswamy, S. (2019). Modeling for reliability optimization of system design and maintenance based on markov chain theory. *Computers & Chemical Engineering*, *124*, 381–404. doi:https://doi.org/10.1016/j.compchemeng.2019.02.016.

Zhu, H., Liu, F., Shao, X., Liu, Q., & Deng, Y. (2011). A cost-based selective maintenance decision-making method for machining line. *Quality and Reliability Engineering International*, *27*, 191–201. doi:https://doi.org/10.1002/qre.1111.