Computers and Chemical Engineering 142 (2020) 107052

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

Integrated optimization of design, storage sizing, and maintenance policy as a Markov decision process considering varying failure rates[☆]

Yixin Ye[a], Ignacio E. Grossmann[a,*], Jose M. Pinto[b], Sivaraman Ramaswamy[b]

[a] Department of Chemical Engineering, Carnegie Melon University, Pittsburgh, PA 15232, United States
[b] Business and Supply Chain Optimization, Linde.digital, Linde plc, Tonawanda, NY 14150, United States

ARTICLE INFO

Article history:
Received 22 May 2020
Revised 22 July 2020
Accepted 6 August 2020
Available online 9 August 2020

Keywords:
Reliability designs
Maintenance
Optimization
Markov decision process
MINLP

ABSTRACT

Various strategies can be applied to improve reliability at certain costs, including equipment redundancy, product storage, and maintenance, which gives rise to the problem of optimally allocating the reliability improvement costs among the strategies and balancing them against the potential loss due to unavailabilities. Motivated by the reliability concerns of air separation units, we use Markov Decision Process to model the stochastic dynamic decision making process of condition-based maintenance assuming bathtub shaped failure rate curves of single units, which is then embedded into a non-convex MINLP (DMP) that considers the trade-off among all the decisions. An initial attempt to directly solve the MINLP (DMP) for a mid-sized problem with several global solvers reveals severe computational difficulties. In response, we propose a custom two-phase algorithm that greatly reduces the required computation effort. The algorithm also shows consistent performance over randomly generated problems around the original example of 4 processing stages and problems of larger sizes.

1. Introduction

Process reliability is important to chemical plants, as it directly impacts the availability of the end product, and thus the profitability. In particular, what motivated this work is the reliability of air separation units that supply gas products to designated customers through pipelines, which is at an even higher stake since the interruption of the pipeline supply will also result in production interruption at the customer site.

If we focus on the system design and the reliability of the process itself, there is a clear trade-off between higher expected availability of the process and higher capital investment for backup units. A few works have looked into this point. Thomaidis and Pistikopoulos (1994, 1995) incorporate the reliability index of each unit as part of process design optimization considering flexibility and reliability. Aguilar et al. (2008) address reliability in utility plant design and operation by considering a few pre-specified redundancy selection alternatives and failure scenarios. Ye et al. (2018) propose a general mixed-integer programming framework for the optimal selection of redundant units.

Another strategy to improve product availability for air separation units is to provide buffer storage of the liquified products, which can be evaporated to sustain the pipeline supply when the air separation units fails. This strategy also incurs costs of building the tanks and maintaining the stock, which increases with the size of the storage tanks (Terrazas-Moreno et al., 2010). Our previous work (Ye et al., 2020) has looked into the exact modeling of the stochastic process of liquid storage consumption.

Quantifying and optimizing the maintenance efforts after the commissioning of a plant is a well recognized topic in industry (Van Rijn, 1987) as well as in academia. Tan and Kramer (1997) present a general framework for preventive maintenance optimization utilizing Monte Carlo Simulation and genetic algorithms overcoming certain drawbacks of analytics based methods and Markov based methods. With regards to timing and resource allocation of maintenance tasks, Pistikopoulos et al. (2001) optimize maintenance alongside with normal production tasks. Cheung et al. (2004) address a short-term scheduling problem of plant shutdown, overhaul, inspection and startup actions within one plant. Amaran et al. (2016) focus on optimizing the maintenance turnaround planning of integrated chemical sites for minimum cost accounting for workload and manpower uncertainties. Achkar et al. (2019) optimize the scheduling of general maintenance tasks on oil and gas wells and surface facilities.

[☆] This paper is dedicated to Prof. Sebastian Engell for his outstanding contributions and leadership in the area of Process Systems Engineering.
[*] corresponding author.
E-mail address: grossmann@cmu.edu (I.E. Grossmann).

https://doi.org/10.1016/j.compchemeng.2020.107052
0098-1354/© 2020 Elsevier Ltd. All rights reserved.

Nomenclature

Indexes

k	Processing stage
h	unit selection of single stage
p	product
n	storage size option
s, s'	state
a	action

Sets

K	processing stages
H_k	unit selections of stage k
P	product kinds, i.e. oxygen, nitrogen, etc.
N_p	storage size options for product p
$S_{k,h}$	possible states of stage k with unit selection h
$A(s)$	possible actions of state s

Parameters

$C^U_{k,h}$	the investment cost of unit selection h at stage k
$C^T_{p,n}$	the investment cost of tank size n for product p
$P_{k,h}(s, a, s')$	the probability of transitioning from state s to state s' if take action a in stage k with unit selection h
$R^{idv}_{k,h}(s,a,s')$	the instant operational cost of transitioning from state s to state s' if take action a in stage k with unit selection h
$R^{idv\_pen}_{k,h}(s,a,s')$	the instant penalty incurred cost of transitioning from state s to state s' if take action a in stage k with unit selection h
$(R^{pen}_{k,h}(s,a,s'))^L$	the lower bound of $R^{pen}_{k,h}(s,a,s')$
$t^{res}_{k,h}(s,a)$	the expected residence time of state s if take action a in stage k with unit selection h

Variables

$z_{k,h}$	binary variable that indicates the selection of design h at stage k
$x_{p,n}$	binary variable that indicates the selection of storage size n for product p
$y_{p,n,k,h}$	binary variable that indicates the selection of storage size n for product p and design h at stage k at the same time
$w_{k,h}(s, a)$	binary variable that indicates the selection of action a at state s of stage k with unit selection h
$v_{k,h}(s)$	the value of state s of stage k with unit selection h
$\pi_{k,h}(s)$	the stationary probability of state s of stage k with unit selection h
$\pi^{sub}_{k,h}(s,a)$	the disaggregated stationary probability of state s of stage k with unit selection h when choosing action a
$R_{k,h}(s, a, s')$	the instant cost of transitioning from state s to state s' if take action a in stage k with unit selection h
$R^{pen}_{k,h}(s,a,s')$	the instant penalty incurred cost of transitioning from state s to state s' if take action a in stage k with unit selection h that is discounted by $t\_ratio^A_k$ based on $R^{idv\_pen}_{k,h}(s,a,s')$
$R\_y^{pen}_{k,h}(s,a,s')$	the aggregated variable of $y_{p,n,k,h}R^{pen}_{k,h}(s,a,s')$
$t\_ratio^A_k$	the portion of time state k being available

However, there have only been limited number of works reported on the simultaneous optimization of design and maintenance schedule (Vassiliadis and Pistikopoulos, 2001; Nápoles-Rivera et al., 2013; Liang and Chang, 2008; Godoy et al., 2015; Wibisono et al., 2014), especially ones that model the stochastic process of failure and maintenance in detail (Redutskiy, 2017). Another example of the latter is our previous work (Ye et al., 2019), where the process is modeled with a Markov Chain, where the failure rates are considered constant for the entire horizon, and are subject to the inspection frequency decision. It was shown in the paper that incorporating maintenance decisions into the economic trade-off can have major impact on the overall cost. As an improvement, in this work, we incorporate the more realistic assumption that the failure rates of single units vary with time, and model the condition-based maintenance decisions as the action policy of Markov Decision Process, which largely refers to Amari et al. (2006), while the latter work only focuses on one unit and has no design component.

Markov Decision Process is a one-step look-ahead framework for dynamic decision making under uncertainty that was first proposed by Bellman (1957). It has since received wide interest and has had broad applications (White, 1993), such as inventory management (Ahiska et al., 2013; Yin et al., 2002), planning and scheduling (Shin et al., 2017), investment (Bäuerle and Rieder, 2011), and maintenance (Byon and Ding, 2010; Chen and Trivedi, 2005), as well as the recently rising reinforcement learning area (Powell, 2004), where MDP is used as the basic framework for describing the behavior of various systems including human beings. The optimality condition of a Markov Decision Process is given by the Bellman equation. Basic ways of finding the solution that satisfy the Bellman equation include linear programming, policy iteration and value iteration. In this work, we employ a slightly modified version of the linear programming formulation, which is mixed-integer and has a more specific objective function to suit our need of simultaneous design and operations optimization.

In Section 2, we describe the problem scope including the decisions to make and their correspondence to the modeling components. Section 3 presents the mathematical model, where we start by introducing the basics of Markov Decision Process in Section 3.1. Then, in Section 3.2, we propose the MINLP model as described in the last paragraph. In order to keep the model tractable, each processing stage is modeled as an MDP (Markov Decision Process), and the stage interdependency is captured via functional relationships of the MDP parameters. In Section 3.3, we describe the interaction between the MDPs of all the processing stages and its impact on the entire system. In Section 4.1, we propose to exactly linearize the bilinear terms in the model. Furthermore, in Section 4.2, we show a reformulation of the objective function that helps to tighten the objective bounds.

In Section 5, we present an example with 4 processing stages to discuss how to determine the basic parameters of the MDPs based on the reliability and price specifications of individual units, and show that directly solving the original MINLP model faces considerable computational difficulties. It motivates us in Section 6 to propose an algorithm with two phases: Enumeration and Bounding, and Rewards Iteration. The objective reformulation proposed in Section 4.2 also helps with the proof of a proposition that supports the validity of the algorithm. Finally in Section 7, we show that the proposed algorithm is efficient on the example introduced in Section 5 as well as on 20 problems randomly generated around it. The performance of the algorithm on larger problems is slightly less stable as reflected by solving another group of 20 randomly generated problems with 6 processing stages.

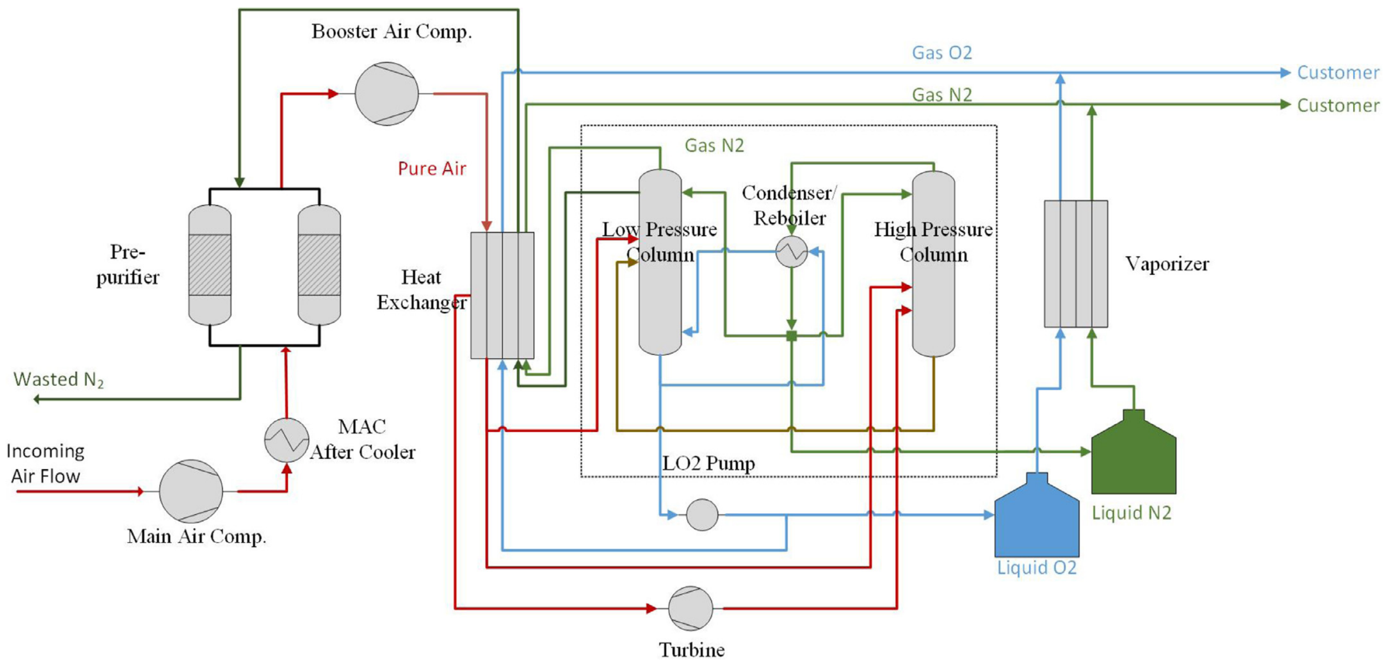


Fig. 1. Air separation process.

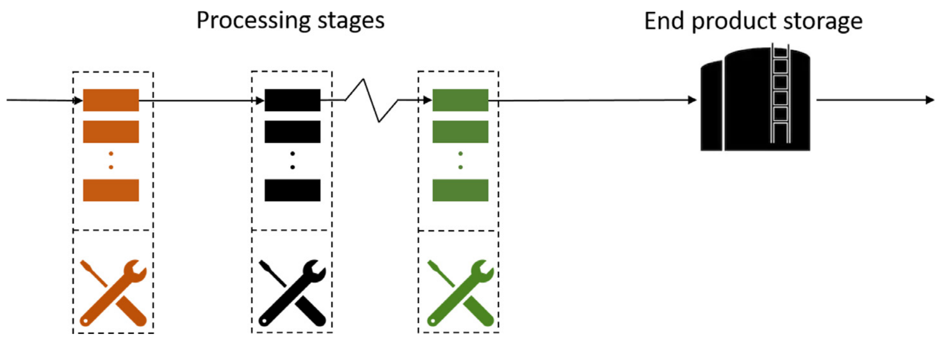


Fig. 2. Conceptual modeling structure.

2. Problem statement

The decisions to make include the selection of redundant units and sizes of storage tanks, as well as basic maintenance policies for a chemical process. As our motivating example, the general flowsheet of an air separation unit is shown in Fig. 1, where the failure of one stage results in the failure of the entire process. Following common practice of the air separation industry (Linde, plc, 2020; Chart, 2020), the sizes of the storage tanks will be selected from a few discrete standard options. If more redundant units are selected for critical processing stages, such as compressors, the plant will be less likely to fail. Also, as discussed in the introduction, the liquid oxygen and liquid nitrogen can be evaporated to sustain the pipeline supply when the air separation units fails, and the larger the storage tanks are, the longer are the downtime periods that can be covered. Furthermore, in the operation phase, placing more efforts into maintenance can increase the process reliability. Each pipeline interruption will incur a fixed amount of penalty, which is to be balanced against the costs of the above strategies to increase availability. To address this problem, we propose a mixed-integer programming model based on a Markov Decision Process framework. Fig. 2 shows the conceptual modeling structure. Assuming that for a process, the base flowsheet is given, the exact design and maintenance policy decisions are described in Sections 2.1 and 2.2, respectively.

2.1. Design decisions

There are two major design decisions:

- The number and selection of redundant units of different prices and reliability specifications for each processing stage. The binary variable $z_{k,h} = 1$ indicates that in processing stage k (e.g. compressor stage), design h (e.g. compressor 1 and compressor 3 each of 100% capacity) is selected.
- The sizes of end product storage tanks. Binary variable $x_{p,n} = 1$ means that for product p (e.g. Nitrogen), tank size n (e.g. 100,000 gallon) is selected.

2.2. Maintenance policy decisions

The bathtub curve shown in Fig. 3(a) is widely accepted on how the failure rate of a unit varies with time (Henley and Kumamoto, 1981; Barlow and Proschan, 1975, etc.). We propose to capture the bathtub-like deterioration/failure process of the equipment and the condition-based maintenance with a discrete-time Markov Decision Process. As shown in Fig. 3(b), the bathtub curve is discretized into three "working states" of the unit: 1. Infant, 2. Stable, and 3. Worn-out. When a unit is not working, there are three other possible states: 4. Stand-by, 5. Stopped, and 6. Failed. At each state, there is a finite number of possible actions: Inspect-T_m^{ins} (which

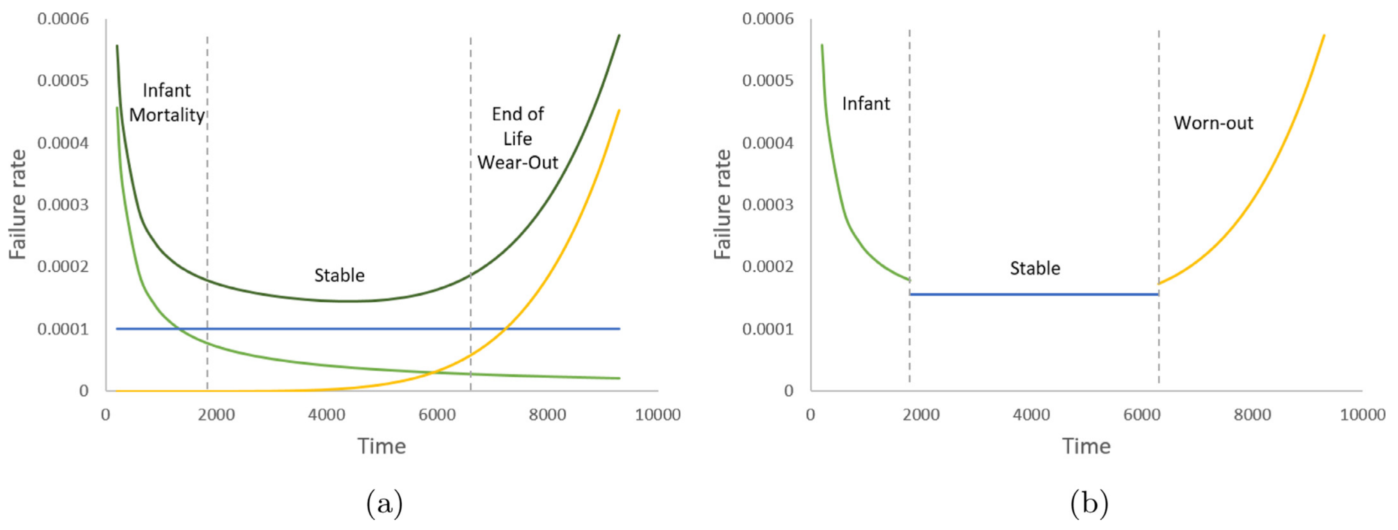


Fig. 3. The bathtub curve.

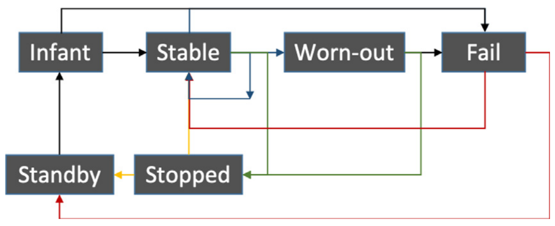


Fig. 4. State space and action space of single units maintenance policy.

means to carry out inspection after T_m^{ins} days. $m \in M$ is the index set of possible inspection intervals), Stop, Maintain, or Repair. One and only one action is to be assigned to each state, which is the main decision to make in terms of the maintenance policy. Fig. 4 shows the state space and action space of a processing stage with only one unit. When in the Infant state, no action will be taken except for waiting for the unit to either fail or proceed to the Stable state (blue arcs). When in the Stable state and the Worn-out state, the unit can be stopped (green arcs), which leads to state Stopped with probability 1. From the Stable state, the action can also be Inspect-T_m^{ins}, $m \in M$ (blue arcs), in which case, the next state will be revealed at the next inspection, and the probability of going to each of the next possible states depends on the inspection interval T_m^{ins}. When in the Stopped state or the Failed state, the only applicable actions are to maintain (the yellow arc) or to repair (the red arc), respectively, which leads back to the Stable state if the unit is needed instantly, or to the Stand-by state if other units in the stage is working properly. With one unit, the Stand-by state will always be skipped. When there are multiple units, actions at the Stand-by state are by no choice and denoted as None, and it will lead back to the Infant state with probability 1. That is because the only transition out of a Stand-by state is to the Infant state, and is always accompanied by the stopping or failing of other units, which are already accounted for. The Stand-by state being followed by the Infant state accounts for the higher likelihood of failure of the units after a period of idling.

3. Mathematical model

As mentioned above, we propose to model the process with MDP (Markov Decision Process), as preparation for which the state space representation is formulated as in the example shown in Fig. 4. Arguably, the exact modeling of the system with multiple processing stages would require construction of the entire state space and state-action pairs, which can become intractable as the number of stages and potential redundant units increase, as shown in Table 1. Therefore, we take a simplified approach to model each processing stage as an MDP (Markov Decision Process), and capture the stage interdependency via functional relationships of the MDP parameters. In Section 3.1, we introduce the basic principles of MDP, especially the optimal condition and how to compute the stationary distribution of the reduced Markov Chain given the optimal actions, which constitute the theoretical foundation of this paper and the mathematical prgramming model afterwards. In Section 3.2, we derive the optimization formulation of the MDP of each stage. In Section 3.3, the interaction between the MDP of each processing stage is modeled. Then in Section 4, we propose to improve the model formulation by linearizing the bilinear terms into two constraints, and reformulating the objective function into an expression with tighter lower bounds for relaxations that arise when dropping the discrete requirements. However, we will show later that directly solving the original MINLP model still faces considerable computational difficulties, which motivates us to propose a customized two-phase algorithm. The objective reformulation proposed in Section 4 also helps with the proofs of the supporting propositions of the algorithm.

Table 1
Numbers of possible states in systems of different sizes.

	SMALL	MEDIUM	LARGE
Number of stages	2	3	4
Number of potential units per stage	2	3	4
Number of possible states	425	430,000	3,576,336,546

3.1. Markov decision process and the optimality condition

A Markov Chain describes the stochastic transitioning behavior of a system among a finite set of states. On top of that, a Markov Decision Process allows the decision maker to choose an action at each state to control the system transitioning behavior. At the same time, an instant reward R (could be positive or negative) is associated with certain action at certain state and the next state. For example, in Fig. 5, compared to the Markov Chain on the left, in the Markov Decision Process shown on the right, actions $A1$ or $A2$ could result in different transition probabilities P between state S1 and S2, and the rewards R depend on current state, the action, and the next state.

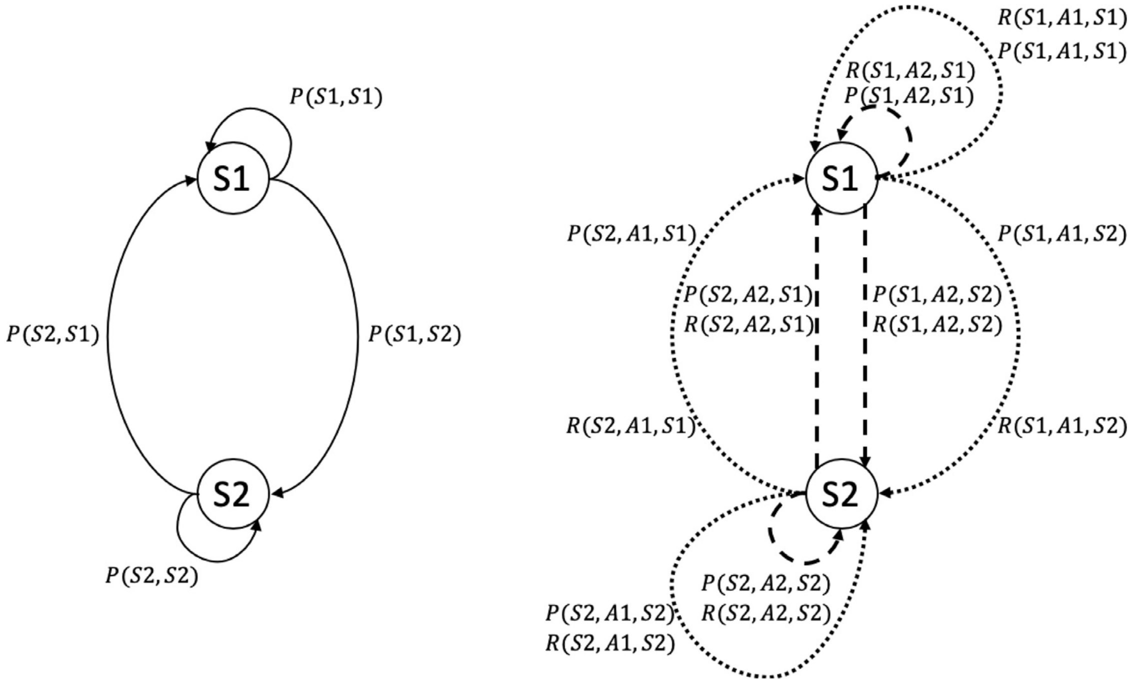


Fig. 5. Markov Chain (left) vs. Markov Decision Process (right).

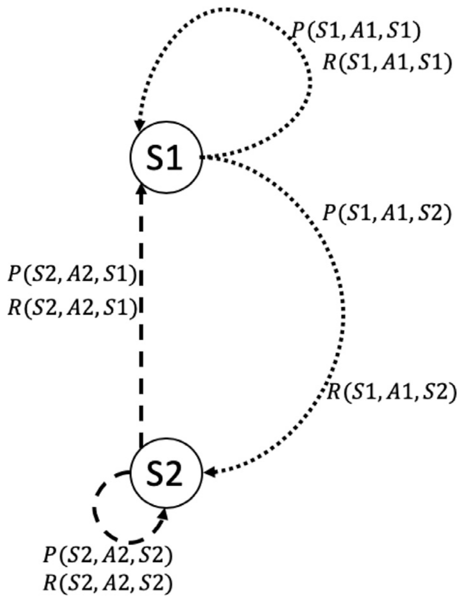


Fig. 6. Reduced transition diagram.

A policy δ of a Markov Decision Process is a projection from its state space S to the action space A.

$\delta : S \to A$

For the example in Fig. 5, a possible policy is $\delta(S1) = A1, \delta(S2) = A2$, in which case the transition diagram becomes as shown in Fig. 6, which reduces back to a Markov Chain.

Under a certain policy δ, a value function is defined for each state s as in Eq. (1). The first term is the reward associated with s and its designated action according to δ. The second term is the weighted sum of the value function of the next possible states $s' \in S$ discounted by a future factor γ. In this paper, we assign a 10% discount factor for γ.

$$V_s(\delta) = \sum_{s' \in S} P(s, \delta(s), s') \cdot (R(s, \delta(s), s') + \gamma \cdot V_{s'}(\delta)), \quad s \in S \quad (1)$$

With the reward function R, the transition matrix P and the discount factor γ given, the optimal policy for maximum overall value V_s^* is defined as in Eq. (2), where $a \in A_s$ is the set of possible actions for state s.

$$V_s^* = \max_{a \in A_s} \sum_{s' \in S} P(s, a, s') \cdot (R(s, a, s') + \gamma \cdot V_{s'}^*), \quad s \in S \quad (2)$$

In our case, the instant rewards $R(s, a, s')$ are the negative of the costs. Therefore, we let $U_s = -V_s$ and flip the sign of the conventional expression. We then remove the negative sign before $R(s, a, s')$, and let $R(s, a, s')$ be the instant costs of going from state s to state s' by action a. We still denote it as instant rewards to distinguish it from the breakdown cost parameters. With that, (3) is equivalent to (2).

$$U_s^* = \min_{a \in A_s} \sum_{s' \in S} P(s, a, s') \cdot (R(s, a, s') + \gamma \cdot U_{s'}^*), \quad s \in S \quad (3)$$

The equivalent linear programming model is shown in (4) (d'Epenoux, 1963; Puterman, 2014), where c_s are arbitrary positive weights. With given $P(s, a, s')$ and $R(s, a, s')$, problem (4) solves for the optimal value of each state and the corresponding policy. Notice that if the optimal policy selects action a for state s, then the optimal dual variable of constraint (s, a) is greater than 0; otherwise it equals 0.

$$\begin{aligned} \max \quad & \textstyle\sum_{s \in S} c_s U_s \\ \text{s.t.} \quad & U_s \le \textstyle\sum_{s' \in S} P(s, a, s') \cdot (R(s, a, s') + \gamma \cdot U_{s'}), \quad s \in S, a \in A_s \end{aligned} \quad (4)$$

As mentioned above, the weights in the objective function, c_s, do not affect the solution of the optimal actions in the above LP model. But in order for the objective function to reflect the operational cost, the weights are set equal to the stationary probability distribution π_s^* of the reduced Markov Chain under the optimal policy. π_s^* satisfies the following constraints.

$$\sum_{s\in S}\pi_s^* = 1 \tag{5}$$

$$\sum_{s\in S}\pi_s^* P(s, a^*(s), s') = \pi_s^*, \quad \forall s' \in S \tag{6}$$

Below we show how to enforce the LP formulation minimizing operational cost (4), as well as the constraints (5) and (6) in the overall optimization model.

3.2. Mixed integer programming formulation based on the MDP optimal condition of each stage

As indicated in the problem statement, we index the processing stages by k, the potential designs of individual stages by h, the products by p, and the storage tank sizes by n. The binary variable $z_{k,h}$ indicates the selection of design h of stage k, while the binary variable $x_{p,n}$ indicates the selection of tank size n for product p. These binary variables satisfy Eqs. (7) and (8).

$$\sum_{h\in H_k} z_{k,h} = 1, \quad \forall k \in K \tag{7}$$

$$\sum_{n\in N_p} x_{p,n} = 1. \quad \forall p \in P \tag{8}$$

Based on the above, we define the binary variable $y_{p,n,k,h}$ that indicates the overall design decision, which satisfy the logical relationship (9), where the logical clauses are true when design h of stage k and size n of product p are selected simultaneously.

$$Y_{p,n,k,h} \iff Z_{k,h} \wedge X_{p,n}, \quad \forall p \in P, n \in N_p, k \in K, h \in H_k \tag{9}$$

Another way of expressing the above logical relationship is (10) and (11), which can be directly translated into the algebraic constraints (12) and (13) that are tighter than those translated from (9) (see Castro et al., 2008 for proof).

$$\bigvee_{n\in N_p} Y_{p,n,k,h} \iff Z_{k,h}, \quad \forall k \in K, h \in H_k, p \in P \tag{10}$$

$$\bigvee_{h\in H_k} Y_{p,n,k,h} \iff X_{p,n}, \quad \forall p \in P, n \in N_p, k \in K \tag{11}$$

$$\sum_{n\in N_p} y_{p,n,k,h} = z_{k,h}, \quad \forall k \in K, h \in H_k, p \in P \tag{12}$$

$$\sum_{h\in H_k} y_{p,n,k,h} = x_{p,n}, \quad \forall p \in P, n \in N_p, k \in K \tag{13}$$

The state space of design h in stage k is denoted as $S_{k,h}$. Similarly, the transition probability matrix is $P_{k,h}(s, a, s')$, $s, s' \in S_{k,h}$, $a \in A(s)$, and the reward matrix is $R_{p,n,k,h}(s, a, s')$, $s, s' \in S_{k,h}$, $a \in A(s)$. The constraint stated in the general formulation (4) is adapted into constraint (14) that involves the design decision $y_{p,n,k,h}$. If $\sum_{p\in P, n\in N_p} y_{p,n,k,h} = 0$, it means that design h in stage k is not selected, and then the first term on the right-hand-side of (4) is zero, making $v_{k,h}(s) = 0, s \in S_{k,h}$ a feasible solution. Considering that in the objective function below, $v_{k,h}(s)$ are being minimized with non-negative weights, $v_{k,h}(s) = 0, s \in S_{k,h}$ is part of the optimal solution.

$$\begin{aligned} v_{k,h}(s) \le & \sum_{s'\in S_{k,h}} \Big(P_{k,h}(s, a, s') \cdot \sum_{p\in P, n\in N_p} y_{p,n,k,h} R_{p,n,k,h}(s, a, s')\Big) \\ & + \gamma \sum_{s'\in S_{k,h}} P_{k,h}(s, a, s') \cdot v_{k,h}(s'), \quad \forall k \in K, h \in H_k, s \in S_{k,h}, a \in A(s) \end{aligned} \tag{14}$$

The objective function is shown in (15), which minimizes the capital cost of the selected processing units ($C^U_{k,h}$) and storage tanks ($C^T_{p,n}$), plus the weighted state values of the MDPs ($\pi_{k,h} \cdot v_{k,h}$), which takes into consideration both the operational costs and the reliability values.

$$\min \sum_{p\in P, n\in N_p} x_{p,n} C^T_{p,n} + \sum_{k\in K, h\in H_k} z_{k,h} C^U_{k,h} + \sum_{k\in K, h\in H_k} \sum_{s\in S_{k,h}} \pi_{k,h}(s) v_{k,h}(s) \tag{15}$$

Notice that in the original LP form shown in (4), the weighted sum of the value functions ($v_{k,h}(s)$ in the model) are maximized in the objective function, which enforces the equality at the action that yields the minimum value. However, as shown in (15), the weighted sum of the value functions have to be minimized together with the investment costs. Therefore, we need another set of constraints to make sure that $v_{k,h}(s)$ does not become unbounded. In disjuction (16), Boolean variable $W_{k,h}(s, a)$ is true if action a is the optimal action for state s, in which case it enforces that $v_{k,h}(s)$ be greater than or equal to the right hand side.

$$\left[\begin{array}{c} W_{k,h}(s,a) \\ v_{k,h}(s) \ge \sum_{s'\in S_{k,h}} \big(P_{k,h}(s,a,s') \cdot \sum_{p\in P, n\in N_p} y_{p,n,k,h} R_{p,n,k,h}(s,a,s')\big) \\ +\gamma \sum_{s'\in S_{k,h}} P_{k,h}(s,a,s') \cdot v_{k,h}(s') \end{array}\right] \vee \big[\neg W_{k,h}(s,a)\big] \tag{16}$$

It can be translated via Big-M reformulation into the algebraic constraint (17).

$$\begin{aligned} & M(1 - w_{k,h}(s,a)) + v_{k,h}(s) \\ & \ge \sum_{s'\in S_{k,h}} \Big(P_{k,h}(s,a,s') \cdot \sum_{p\in P, n\in N_p} y_{p,n,k,h} R_{p,n,k,h}(s,a,s')\Big) \\ & + \gamma \sum_{s'\in S_{k,h}} P_{k,h}(s,a,s') \cdot v_{k,h}(s'), \quad \forall k \in K, h \in H_k, s \in S_{k,h}, a \in A(s) \end{aligned} \tag{17}$$

Logic condition (18) requires that if design h is not selected, meaning $z_{k,h} = 0$, then any action for that design is not valid. Constraint (19) is the corresponding algebraic equation of (18).

$$\bigvee_{a\in A_s} W_{k,h}(s,a) \iff Z_{k,h}, \quad \forall k \in K, h \in H_k, s \in S_{k,h} \tag{18}$$

$$\sum_{a\in A_s} w_{k,h}(s,a) = z_{k,h}, \quad \forall k \in K, h \in H_k, s \in S_{k,h} \tag{19}$$

$\pi_{k,h}(s)$ is the stationary probability of state s in the state space of the MDP of design h in stage k, which satisfies (20), where $a^*(s)$ is the optimal action at state s.

$$\sum_{s\in S_{k,h}} \pi_{k,h}(s) P_{k,h}(s, a^*(s), s') = \pi_{k,h}(s'), \quad \forall k \in K, h \in H_k, s' \in S_{k,h} \tag{20}$$

To represent (20) by algebraic inequalities, we define $\pi^{sub}_{k,h}(s,a)$ as the disaggregated variables of $\pi_{k,h}(s)$ with regard to action a as shown in (21), whose valued are related to the Boolean variables $W_{k,h}(s, a)$ as shown in (22), which translates into algebraic constraint (23).

$$\sum_{a\in A_s} \pi^{sub}_{k,h}(s,a) = \pi_{k,h}(s), \quad \forall k \in K, h \in H_k \tag{21}$$

$$\left[\begin{array}{c} W_{k,h}(s,a) \\ \pi^{sub}_{k,h}(s,a) \le 1 \end{array}\right] \vee \left[\begin{array}{c} \neg W_{k,h}(s,a) \\ \pi^{sub}_{k,h}(s,a) = 0 \end{array}\right] \tag{22}$$

$$\pi^{sub}_{k,h}(s,a) \le w_{k,h}(s,a), \quad \forall k \in K, h \in H_k, s \in S_{k,h}, a \in A_s \tag{23}$$

Given the relationship shown in (23), (21) can be substituted into (20), which yields (24).

$$\sum_{s\in S_{k,h}}\sum_{a\in A_s}\pi^{sub}_{k,h}(s,a)P_{k,h}(s,a,s')=\pi_{k,h}(s'),\quad \forall k\in K, h\in H_k, s'\in S_{k,h} \tag{24}$$

Furthermore, as shown in (22), we require that $\pi_{k,h}(s)$ are 0 if design h is not selected for stage k, and that $\pi_{k,h}(s)$ sum up to 1 if otherwise. The condition can be expressed with the algebraic constraint shown in (23)

$$\begin{bmatrix} Z_{k,h}\\ \sum_{s\in S_{k,h}}\pi_{k,h}(s)=1\end{bmatrix}\vee\begin{bmatrix}\neg Z_{k,h}\\ \sum_{s\in S_{k,h}}\pi_{k,h}(s)=0\end{bmatrix} \tag{25}$$

$$\sum_{s\in S_{k,h}}\pi_{k,h}(s)=z_{k,h},\quad \forall k\in K, h\in H_k \tag{26}$$

3.3. Stage interdependency

As mentioned in Section 2, the processing stages impact each other's performance. Especially, a stage k can only transition into the next state when the other stages are not in the Failed state. Therefore, as shown in Eq. (27), the reward of transitioning from state s into another state s', $R_{p,n,k,h}(s,a,s')$, is to be corrected with the portion of time the other states being available, based on the reward parameter when the stage k stands alone, $R^{idv}_{k,h}(s,a,s')$ and $R^{idv\_pen}_{p,n,k,h}(s,a,s')$. Notice here $R^{idv}_{k,h}(s,a,s')$ has to be evenly distributed to each product.

$$R_{p,n,k,h}(s,a,s')=R^{idv}_{k,h}(s,a,s')\cdot\frac{1}{|P|}+R^{idv\_pen}_{p,n,k,h}(s,a,s')\cdot\prod_{l\in K,l\neq k}t\_ratio^A_l,$$
$$\forall p\in P, n\in N_p, k\in K, h\in H_k, s\in S_{k,h}, a\in A(s), s'\in S_{k,h} \tag{27}$$

where $t\_ratio^A_k$ equals to the weighted probability of available states divided by the probability weighted time length of all states of stage k, as shown in (28), where t^{res} is the expected residence time in state s if taking action a.

$$t\_ratio^A_k=\frac{\sum_{h\in H_k}\sum_{s\in S_{k,h},s\notin S^f_{k,h},a\in A_s}\pi^{sub}_{k,h}(s,a)t^{res}_{k,h}(s,a)}{\sum_{h\in H_k}\sum_{s\in S_{k,h},a\in A_s}\pi^{sub}_{k,h}(s,a)t^{res}_{k,h}(s,a)},\ k\in K \tag{28}$$

With that, the basic non-convex MINLP model minimizes the total cost (15) subject to constraints (7), (8), (12)–(14), (17), (19), and (21)–(28), where (14) and (17) involve bilinear terms of binary variables $y_{p,n,k,h}$ and continuous variables $R^{pen}_{p,n,k,h}$, (27) involves multi-linear terms of continuous variables $t\_ratio^A_k$, and (28) is a linear-fractional constraint.

4. Reformulations

As will be shown in this section, the model can be improved through several reformulation steps. In Section 4.1, we perform an exact linearization of the bilinear terms in the model, and in Section 4.2, we propose to reformulate the objective function taking advantage of the functional relationships specified by the constraints.

4.1. Standard linearization of the bilinear constraints

We use Glover's linearization scheme (Glover, 1975), a special case of the McCormick Envelopes (McCormick, 1976), to linearize Eqs. (14) and (17), which also involve the reformulation of Eq. (27). We first introduce a new variable $R^{pen}_{p,n,k,h}(s,a,s')$ and rewrite the three constraints mentioned above as follows in (29)–(31) where variable $R_{p,n,k,h}(s,a,s')$ is eliminated.

$$v_{k,h}(s)\le\sum_{s'\in S_{k,h}}P_{k,h}(s,a,s')\cdot\Big(z_{k,h}R^{idv}_{k,h}(s,a,s')+\sum_{p\in P}\sum_{n\in N_p}y_{p,n,k,h}R^{pen}_{p,n,k,h}(s,a,s')\Big)$$
$$+\gamma\sum_{s'\in S_{k,h}}P_{k,h}(s,a,s')\cdot v_{k,h}(s'),\quad\forall k\in K, h\in H_k, s\in S_{k,h}, a\in A(s) \tag{29}$$

$$M(1-w_{k,h}(s,a))+v_{k,h}(s)$$
$$\ge\sum_{s'\in S_{k,h}}P_{k,h}(s,a,s')\cdot\Big(z_{k,h}R^{idv}_{k,h}(s,a,s')+\sum_{p\in P}\sum_{n\in N_p}y_{p,n,k,h}R^{pen}_{p,n,k,h}(s,a,s')\Big)$$
$$+\gamma\sum_{s'\in S_{k,h}}P_{k,h}(s,a,s')\cdot v_{k,h}(s'),\quad\forall k\in K, h\in H_k, s\in S_{k,h}, a\in A(s) \tag{30}$$

$$R^{pen}_{p,n,k,h}(s,a,s')=R^{idv\_pen}_{p,n,k,h}(s,a,s')\cdot\prod_{l\in K,l\neq k}t\_ratio^A_l,$$
$$\forall p\in P, n\in N_p, k\in K, h\in H_k, s\in S_{k,h}, a\in A(s), s'\in S_{k,h} \tag{31}$$

We then replace the bilinear term $y_{p,n,k,h}R^{pen}_{p,n,k,h}(s,a,s')$ with the new variable $R\_y^{pen}_{p,n,k,h}(s,a,s')$ as shown in (32) and (33), and use Eqs. (34)–(37) to require that they be equal to each other.

$$v_{k,h}(s)\le\sum_{s'\in S_{k,h}}P_{k,h}(s,a,s')\cdot\Big(z_{k,h}R^{idv}_{k,h}(s,a,s')+\sum_{p\in P}\sum_{n\in N_p}R\_y^{pen}_{p,n,k,h}(s,a,s')\Big)$$
$$+\gamma\sum_{s'\in S_{k,h}}P_{k,h}(s,a,s')\cdot v_{k,h}(s'),\quad\forall k\in K, h\in H_k, s\in S_{k,h}, a\in A(s) \tag{32}$$

$$M(1-w_{k,h}(s,a))+v_{k,h}(s)\ge\sum_{s'\in S_{k,h}}P_{k,h}(s,a,s')$$
$$\cdot\Big(z_{k,h}R^{idv}_{k,h}(s,a,s')+\sum_{p\in P}\sum_{n\in N_p}R\_y^{pen}_{p,n,k,h}(s,a,s')\Big)$$
$$+\gamma\sum_{s'\in S_{k,h}}P_{k,h}(s,a,s')\cdot v_{k,h}(s'),\quad\forall k\in K, h\in H_k, s\in S_{k,h}, a\in A(s) \tag{33}$$

$$R\_y^{pen}_{p,n,k,h}(s,a,s')\le y_{p,n,k,h}R^{idv\_pen}_{p,n,k,h}(s,a,s'),$$
$$\forall p\in P, n\in N_p, k\in K, h\in H_k, s\in S_{k,h}, a\in A(s), s'\in S_{k,h} \tag{34}$$

$$R\_y^{pen}_{p,n,k,h}(s,a,s')\le R^{pen}_{p,n,k,h}(s,a,s')+(y_{p,n,k,h}-1)(R^{pen}_{p,n,k,h}(s,a,s'))^L,$$
$$\forall p\in P, n\in N_p, k\in K, h\in H_k, s\in S_{k,h}, a\in A(s), s'\in S_{k,h} \tag{35}$$

$$R\_y^{pen}_{p,n,k,h}(s,a,s')\ge y_{p,n,k,h}(R^{pen}_{p,n,k,h}(s,a,s'))^L,$$
$$\forall p\in P, n\in N_p, k\in K, h\in H_k, s\in S_{k,h}, a\in A(s), s'\in S_{k,h} \tag{36}$$

$$R\_y^{pen}_{p,n,k,h}(s,a,s')\ge R^{pen}_{p,n,k,h}(s,a,s')+(y_{p,n,k,h}-1)R^{idv\_pen}_{p,n,k,h}(s,a,s'),$$
$$\forall p\in P, n\in N_p, k\in K, h\in H_k, s\in S_{k,h}, a\in A(s), s'\in S_{k,h} \tag{37}$$

The element-wise lower bounds of $R^{pen}_{p,n,k,h}(s,a,s')$, $(R^{pen}_{p,n,k,h}(s,a,s'))^L$, are found by first solving for the lowest possible portion of available time $(t\_ratio^A_k)^L$ of each stage k as shown in problem (38), and apply (31) as shown in (39). Notice that all the constraints of problem (38) are linear, and the objective function is linear fractional, which is nonlinear but pseudo-convex and pseudo-concave (Avriel, 2003). Pseudo-convex function has the property of a convex function with respect to finding its local minima. Therefore, when we relax the integrality requirement in problem (38), it becomes a convex NLP.

$$(t\_ratio^A_k)^L=\min\ \frac{\sum_{h\in H_k}\sum_{s\in S_{k,h},s\notin S^f_{k,h},a\in A_s}\pi^{sub}_{k,h}(s,a)t^{res}_{k,h}(s,a)}{\sum_{h\in H_k}\sum_{s\in S_{k,h},a\in A_s}\pi^{sub}_{k,h}(s,a)t^{res}_{k,h}(s,a)} \tag{38}$$

$$\text{s.t.}\ \sum_{h\in H_k}z_{k,h}=1$$

$$\sum_{a\in A_s} w_{k,h}(s,a) = z_{k,h}, \quad \forall h\in H_k, s\in S_{k,h}$$
$$\sum_{s\in S_{k,h}} \pi_{k,h}(s) = z_{k,h}, \quad \forall h\in H_k$$
$$\sum_{a\in A_s} \pi^{sub}_{k,h}(s,a) = \pi_{k,h}(s), \quad \forall h\in H_k$$
$$\pi^{sub}_{k,h}(s,a) \le w_{k,h}(s,a), \quad \forall h\in H_k, s\in S_{k,h}, a\in A_s$$
$$\sum_{s\in S_{k,h}}\sum_{a\in A_s} \pi^{sub}_{k,h}(s,a)P_{k,h}(s,a,s') = \pi_{k,h}(s'), \quad \forall h\in H_k, s'\in S_{k,h}$$

$$(R^{pen}_{p,n,k,h}(s,a,s'))^L = R^{idv\_pen}_{p,n,k,h}(s,a,s')\cdot \prod_{l\in K, l\neq k}((t\_ratio^A_k))^L,$$
$$\forall p\in P, n\in N_p, k\in K, h\in H_k, s\in S_{k,h}, a\in A(s), s'\in S_{k,h} \quad (39)$$

4.2. Reformulation of the objective function

In this section we show a valid reformulation of the objective function (15) that potentially provides tighter lower bounds, that is derived based on the MDP optimal conditions implied by the constraints. Thus, the MINLP (referred to as (DMP)) consists of the objective function (40) that represents net present value of the system and constraints (7), (8), (12), (13), (19), (21)–(23), (28), and (31)–(37), with linear fractional functions in (28), multi-linear terms in (31) and bilinear terms in (40), all of which cannot be exactly linearized.

$$\min_{x,y,z,w,\pi,\pi^{sub},\pi^a,v,R^{pen}} \sum_{p\in P,n\in N_p} x_{p,n}C^T_{p,n} + \sum_{k\in K,h\in H_k} z_{k,h}C^U_{k,h}$$
$$+\tfrac{1}{1-\lambda}\sum_{k\in K,h\in H_k}\sum_{s\in S_{k,h},a\in A_s}\pi^{sub}_{k,h}(s,a) \quad (40)$$
$$\cdot\sum_{s'\in S_{k,h}} P_{k,h}(s,a,s')\left(R^{idv}_{k,h}(s,a,s') + \sum_{p\in P}\sum_{n\in N_p} R\_y^{pen}_{p,n,k,h}(s,a,s')\right)$$

In Appendix A, we prove in Proposition 1 that the new objective function in (40) has equal value as the previous one in (15) when all the constraints are satisfied. The new objective function in (40) has two important properties. First, comparing to the original objective function in (15) which also contains the sum of bilinear terms, the new objective function breaks down to more terms, which gives rise to potentially stronger lower bounds when being directly relaxed (dropping the discrete requirement). Also, one part of the bilinear terms in the new objective function is the complicating variable $R\_y^{pen}_{p,n,k,h}$ that reflects the stage interactions, so when the complicating variable is fixed, the objective function becomes linear.

5. Illustrative example

The motivating example of air separation unit has a few critical processing stages, such as main air compressor, pre-purifier, booster air compressor, and the LO2 pump. In this example, three redundancies are considered for each processing stages, and the pre-purifier stage has to have at least 2 units to function. The superstructure is shown in Fig. 7. There are two products, O2 and N2. The reliability data being used are modified based on actual data. Stable failure rates range from 0.0001 to 0.001 times per day. The Stable phase lasts for 3 years. Repair times are 24 h. Capital cost of each unit range from \$30k to \$150k. Maintenance times are 6 h. Repair costs range from \$4k to \$20k per time. Maintenance costs range from \$2k to \$10k per time.

Table 2 shows the penalty rates and pipeline flow rates used in the model. Table 3 shows the standard tank size options in terms of number of days of demand it can cover, and the corresponding costs normalized based on the costs of the smallest tanks. In Section 5.1, we will show how the equipment and contract specifications shown above are transformed into the parameters used in the MINLP described in Sections 3 and 4. In Section 5.2, we show the preliminary computational results for directly solving the problem.

Table 2
Profitability parameters.

	c^{fail}_p (k\$ per outage)	δ_p (k gallon per day)
LO2	2400	48
LN2	2000	60

Table 3
Tank sizes and costs.

Tank sizes (days)	2	8	14	20	30
Normalized cost of LO2 tank	1	4.3	7.8	11.3	17.3
Normalized cost of LN2 tank	1	4.3	7.8	11.3	17.3

5.1. Transforming the specifications into the MDP parameters

The transition probability matrix $P_{k,h}(s,a,s')$, $s, s'\in S_{k,h}$, $a\in A(s)$ and the reward matrix $R_{k,h}(s,a,s')$, $s, s'\in S_{k,h}$, $a\in A(s)$ are the key parameters in the MDPs. In general, we extract the transition probabilities from the Weibull distributions of individual units, and the reward parameters from the operational costs and the unavailability losses. Below we show an example of how to obtain the MDP parameters for a single unit in a single stage, where the stage index k and design index h are dropped.

As stated in Section 2.2, we discretize the bathtub curve into 3 states, Infant, Stable, and Worn-out (Fig. 3(b)). For any unit in any stage, Stop action leads to Stopped state with probability 1. Also, Maintain action is the only option in Stopped state, and Repair action is the only option in Failed state, which leads to Infant state with probability 1.

$$P(\text{Stable},\text{Stop},\text{Stopped}) = 1 \quad (41)$$

$$P(\text{Worn-out},\text{Stop},\text{Stopped}) = 1 \quad (42)$$

$$P(\text{Stopped},\text{Maintain},\text{Stable}) = 1 \quad (43)$$

$$P(\text{Failed},\text{Repair},\text{Stable}) = 1 \quad (44)$$

For the working states on the bathtub curve, we assume Weibull distributions whose cumulative distributive function is denoted by $F(x;\lambda,\beta) = 1 - e^{-(\lambda x)^\beta}$. For Infant state, the shape factor is β_1. For Worn-out state, the shape factor is β_3. For Stable state, the shape factor is $\beta_2 = 1$, where the Weibull distribution reduces to exponential distribution. $\{T^{ins}_m, m\in M\}$ is the set of possible inspection intervals, where m is the index for inspection intervals. T^{stable} is the length of stable period of the subject unit. T^{infant} is the length of infant period. T^{worn} is the length of worn-out period.

$$P(\text{Infant},\text{None},\text{Failed}) = F(T^{infant};\lambda,\beta_1) \quad (45)$$

$$P(\text{Infant},\text{None},\text{Stable}) = 1 - F(T^{infant};\lambda,\beta_1) \quad (46)$$

$$P(\text{Stable},\text{Inspect-}T^{ins}_m,\text{Stable}) = e^{-\lambda\cdot T^{ins}_m}\cdot\left(1 - \frac{T^{ins}_m}{T^{stable}}\right), \quad \forall m\in M \quad (47)$$

$$P(\text{Stable},\text{Inspect-}T^{ins}_m,\text{Worn-out}) = e^{-\lambda\cdot T^{ins}_m}\cdot\frac{T^{ins}_m}{T^{stable}}, \forall m\in M \quad (48)$$

$$P(\text{Stable},\text{Inspect-}T^{ins}_m,\text{Failed}) = 1 - e^{-\lambda\cdot T^{ins}_m}, \forall m\in M \quad (49)$$

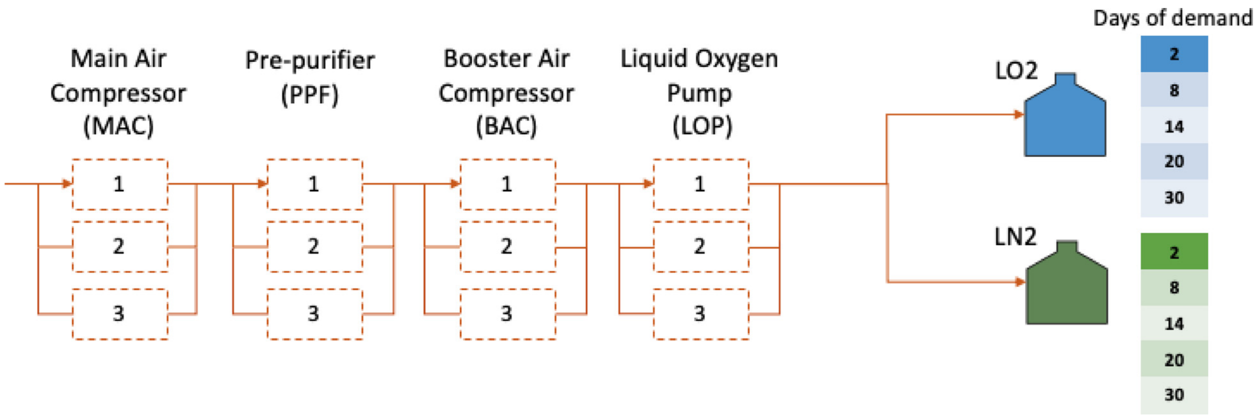


Fig. 7. Superstructure of the ASU example

$$P(\text{Worn-out, None, Failed}) = F(T^{infant} + T^{stable} + T^{worn}; \lambda, \beta_3) \tag{50}$$

$$P(\text{Worn-out, None, Worn-out}) = 1 - F(T^{infant} + T^{stable} + T^{worn}; \lambda, \beta_3) \tag{51}$$

The expected residence time in state s when taking action a is denoted as $t^{res}(s, a)$. For example, for Stable state, the next point of observation is either the next inspection depending on the choice of inspection interval, or the failure before the inspection. Therefore, as shown in (52), the expected residence time in a Stable state is the first moment of the exponential lifetime distribution within the inspection interval T_m^{ins}, plus T_m^{ins} multiplied with the cumulative probability of not failing within T_m^{ins}.

$$t^{res}(\text{Stable, Inspect-}T_m^{ins}) = \int_0^{T_m^{ins}} t \cdot \lambda e^{-\lambda t} dt + e^{-\lambda T_m^{ins}} \cdot T_m^{ins} = \frac{1}{\lambda}(1 - e^{-\lambda T_m^{ins}}) \tag{52}$$

Similarly, for Infant and Worn-out states, the expected residence times are calculated as shown in (53) and (54) based on the Weibull lifetime distributions. $\gamma(s, x) = \int_0^x t^{s-1} e^{-t} dt$ is the incomplete gamma function.

$$\begin{aligned} t^{res}(\text{Infant, None}) &= \int_0^{T^{infant}} t \cdot \beta_1 \lambda^{\beta_1} t^{\beta_1 - 1} e^{-(\lambda t)^{\beta_1}} dt + e^{-(\lambda T^{infant})^{\beta_1}} \cdot T^{infant} \\ &= \frac{1}{\lambda} \cdot \gamma(\frac{1}{\beta_1} + 1, (\lambda T^{infant})^{\beta_1}) + e^{-(\lambda T^{infant})^{\beta_1}} \cdot T^{infant} \end{aligned} \tag{53}$$

$$\begin{aligned} t^{res}(\text{Worn-out, None}) &= \int_{T^{infant}+T^{stable}}^{T^{infant}+T^{stable}+T^{worn}} t \cdot \beta_3 \lambda^{\beta_3} t^{\beta_3 - 1} e^{-(\lambda t)^{\beta_3}} dt \\ &+ e^{-\left(\lambda\left(T^{infant}+T^{stable}+T^{worn}\right)\right)^{\beta_3}} \cdot T^{worn} \end{aligned} \tag{54}$$

$$\begin{aligned} &= \frac{1}{\lambda} \cdot [\gamma\left(\frac{1}{\beta_3} + 1, \left(\lambda\left(T^{infant} + T^{stable} + T^{worn}\right)\right)^{\beta_3}\right) \\ &-\gamma\left(\frac{1}{\beta_3} + 1, \left(\lambda\left(T^{infant} + T^{stable}\right)\right)^{\beta_3}\right)] \\ &+ e^{-\left(\lambda\left(T^{infant}+T^{stable}+T^{worn}\right)\right)^{\beta_3}} \cdot T^{worn} \end{aligned} \tag{55}$$

The basic instant reward of transitioning from state s to state s' by taking action a is denoted as $R^{idv}(s, a, s')$, which is the corresponding operational cost of taking Inspection, Maintenance or Repair actions.

$$R^{idv}(s, \text{Inspect-}T_m^{ins}, s') = c^{ins}, \quad \forall s' \notin S^f, m \in M \tag{56}$$

$$R^{idv}(s, \text{Maintain}, s') = c^{main}, \quad \forall s' \notin S^f \tag{57}$$

$$R^{idv}(s, \text{Repair}, s') = c^{repair}, \quad \forall s' \notin S^f \tag{58}$$

The penalty instant reward is 0 if s' is not a failure state.

$$R_{p,n}^{idv\_pen}(s, a, s') = 0, \quad \forall p \in P, n \in N_p, s' \notin S^f \tag{59}$$

If the destination state is Failed or Stopped, then the penalty instant reward is the outage penalty associated with the repair/maintenance rate of the failure scenario, $\mu^r(s)/\mu^m(s)$, the storage size of each product $V_{p,n}$, and the consumption rate of each product δ_p, as shown in Eqs. (60) and (61).

$$R_{p,n}^{idv\_pen}(s, a, s') = c_p^{fail} \cdot e^{-\frac{V_{p,n}}{\delta_p \mu^r(s')}}, \quad \forall p \in P, n \in N_p, s' \in S^f \tag{60}$$

$$R_{p,n}^{idv\_pen}(s, \text{Stop}, s') = c_p^{fail} \cdot e^{-\frac{V_{p,n}}{\delta_p \mu^m(s')}}, \quad \forall p \in P, n \in N_p, s' \in S^f \tag{61}$$

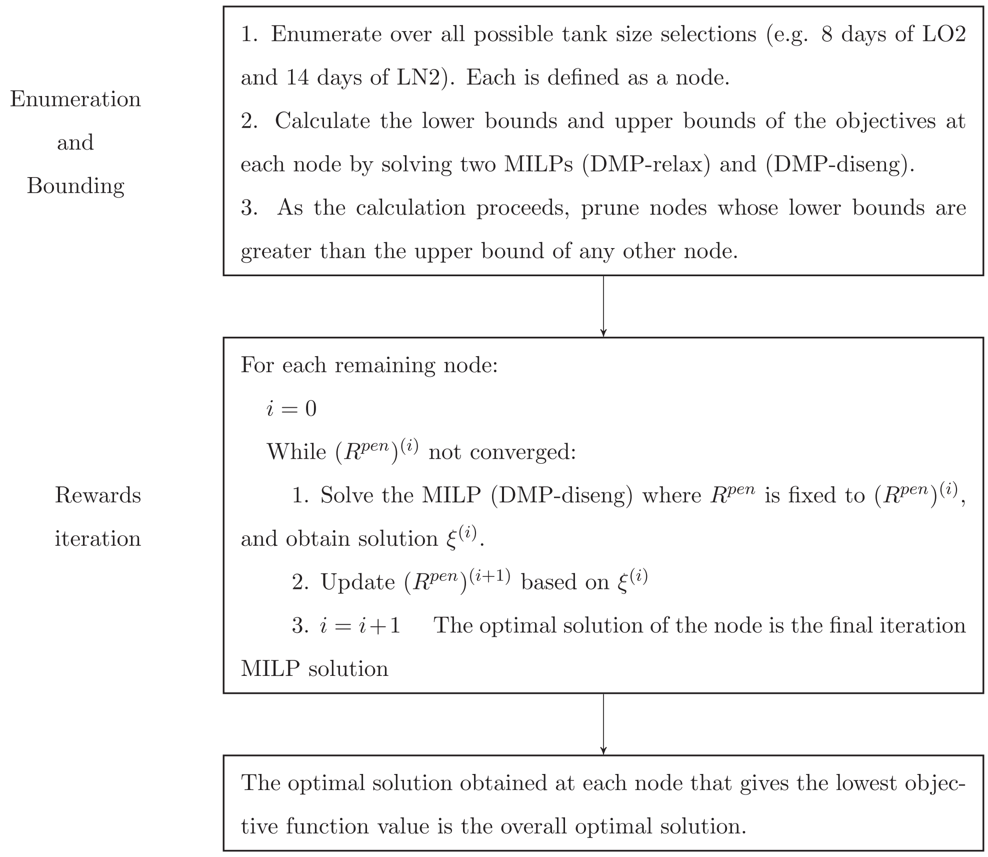
5.2. Preliminary computational results

The MINLP model (DMP) for the system in Fig. 7 has 39,555 equations and 47,559 variables with 1762 binary variables. A first attempt is made to solve the MINLP model (DMP) with the objective function (40) and constraints (7), (8), (12), (13), (19), (21)–(23), and (31)–(37) using the global solver SCIP 6.0 on GAMS 26.1.0 (Intel® Xeon® CPU X5650 @ 2.67 GHz). It only reached a solution with a gap of 21.08% after 100,000 CPUs (27.8 Hr), which is better than the performance of the global solvers BARON 17.4.1 and Antigone 1.1.

In order to overcome the computational difficulty, we propose below an algorithm that takes advantage of the problem structure.

6. A two-phase algorithm

Considering that the decision regarding storage tank sizes has a rather small search space and an impact on the entire system, i.e., the storage tanks are ”shared” by the processing stages, we propose an algorithm with two major execution phases. The first phase is called Enumeration and Bounding, where we exhaustively screen all the possible tank size decision nodes by solving two MILPs (denoted as (DMP-relax) and (DMP-diseng)) at each node for the respective objective function bounds, and prune all the nodes with lower bounds greater than the upper bound of any other node. The second phase is called Rewards Iteration, which is



Enumeration and Bounding

1. Enumerate over all possible tank size selections (e.g. 8 days of LO2 and 14 days of LN2). Each is defined as a node.
2. Calculate the lower bounds and upper bounds of the objectives at each node by solving two MILPs (DMP-relax) and (DMP-diseng).
3. As the calculation proceeds, prune nodes whose lower bounds are greater than the upper bound of any other node.

Rewards iteration

For each remaining node:
$i = 0$
While $(R^{pen})^{(i)}$ not converged:
1. Solve the MILP (DMP-diseng) where R^{pen} is fixed to $(R^{pen})^{(i)}$, and obtain solution $\xi^{(i)}$.
2. Update $(R^{pen})^{(i+1)}$ based on $\xi^{(i)}$
3. $i = i+1$ The optimal solution of the node is the final iteration MILP solution

The optimal solution obtained at each node that gives the lowest objective function value is the overall optimal solution.

Fig. 8. Algorithm overview.

carried out for each of the remaining nodes: A sequence of MILPs (DMP-diseng) with iterative reward parameters are solved until the decisions converge. Fig. 8 provides a more detailed description of the algorithm.

In Section 6.1, we introduce the bounding problems and prove that the bounds are valid. In Section 6.2, we explain the iterative procedure for solving each node.

6.1. Enumeration and bounding

In this section, we show that valid upper and lower bounds of the objective function with certain storage tank sizes can be obtained by solving two MILP models where R^{pen} is fixed to its upper and lower bounds, respectively, and we describe how to determine these variable bounds.

First, let us represent the MINLP (DMP) with the compact form (62). For simplicity, we let ξ stand for the binary and continuous variables other than storage tank selection variables x and the penalty reward R^{pen} subject to the multilinear constraint (31). Therefore, $\xi = (y, z, w, \pi, \pi^{sub}, t\_ratio^{A}, \nu)$, and we let ($x^*$, $(R^{pen})^*$, ξ^*) be the true optimizer of the problem.

$$\min_{x\in X, 0\le R^{pen}, \le R^{idv\_pen} \xi\in\Xi} \{f(x, R^{pen}, \xi) \mid g(x, R^{pen}, \xi) \le 0\} \tag{62}$$

For certain node of storage tank size $\hat{x}$, we let the $(\hat{R}^{pen}, \hat{\xi})$ be the minimizer of the constrained problem as shown in (63)

$$(\hat{R}^{pen}, \hat{\xi}) = \arg \min_{0\le R^{pen} \le R^{idv\_pen}, \xi\in\Xi} \{f(\hat{x}, R^{pen}, \xi) \mid g(\hat{x}, R^{pen}, \xi) \le 0\} \tag{63}$$

At each node of certain storage tank size $\hat{x}$, if we go further by removing the multilinear constraint (31) that engage R^{pen} of each stage with the stationary probability distribution of other stages, and fixing R^{pen} to certain valid values $\tilde{R}^{pen}$, the MINLP model (DMP) will reduce to an MILP, which we call (DMP-diseng). The compact form of (DMP-diseng) is written in (64).

$$\min_{\xi\in\Xi}\{c(\hat{x}, \tilde{R}^{pen}) \cdot \xi \mid A^{diseng} \cdot \xi \le b^{diseng}(\hat{x}, \tilde{R}^{pen})\} \tag{64}$$

By definition, we have (65) and (66).

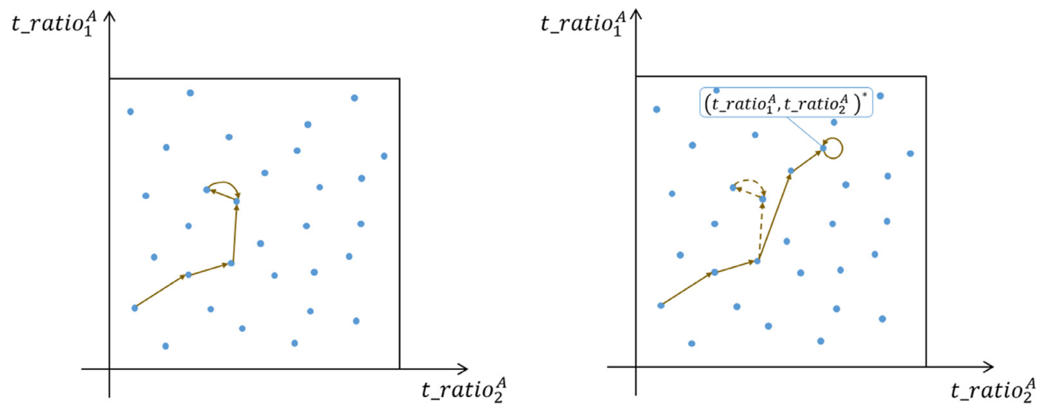
$$f(x^*, (R^{pen})^*, \xi^*) = \min_{\xi\in\Xi}\{c(x^*, (R^{pen})^*) \cdot \xi \mid A^{diseng} \cdot \xi \le b^{diseng}(x^*, (R^{pen})^*)\} \tag{65}$$

$$f(\hat{x}, \hat{R}^{pen}, \hat{\xi}) = \min_{\xi\in\Xi}\{c(\hat{x}, \hat{R}^{pen}) \cdot \xi \mid A^{diseng} \cdot \xi \le b^{diseng}(\hat{x}, \hat{R}^{pen})\} \tag{66}$$

Based on the MILP model (DMP-diseng), a small relaxation can be derived to form another MILP model (DMP-relax). If we have the element-wise lower and upper bounds of R^{pen}: $(R^{pen})^U$ and $(R^{pen})^L$, the lower bound and upper bound of the objective function $f(\hat{x}, \hat{R}^{pen}, \hat{\xi})$ at each node of storage tank size $\hat{x}$ can be obtained by solving (DMP-relax) and (DMP-diseng) as shown in (67) and (68). It is to be noticed that in (DMP-diseng) (68), the $\tilde{R}^{pen}$ in the objective function and the constraints are fixed to different values. In Appendix B, we will show the exact forms of (DMP-diseng) and (DMP-relax), and prove that inequalities (67) and (68) hold with respect to the problem nature.

$$\min_{\xi\in\Xi}\{c(\hat{x}, (R^{pen})^L) \cdot \xi \mid A^{relax} \cdot \xi \le b^{relax}(\hat{x}, (R^{pen})^L)\} \le f(\hat{x}, \hat{R}^{pen}, \hat{\xi}) \tag{67}$$

$$\min_{\xi\in\Xi}\{c(\hat{x}, (R^{pen})^U) \cdot \xi \mid A^{diseng} \cdot \xi \le b^{diseng}(\hat{x}, (R^{pen})^L)\} \ge f(\hat{x}, \hat{R}^{pen}, \hat{\xi}) \tag{68}$$



(a) Iteration path fallen into a loop

(b) Iteration path converges to a stationary point

Fig. 9. Iteration path illustration.

The element-wise upper bounds of R^{pen}: $(R^{pen})^U$, are set as the original penalty parameters R^{idv_pen}. The element-wise lower bounds of R^{pen}: $(R^{pen})^L$, are set as $(R^{pen}_{p,n,k,h}(s,a,s'))^L$ which are first derived in Section 4.1.

The algorithm of the Enumeration and Bounding phase is described in Algorithm 1.

Algorithm 1:

```
for $i \le \prod_{p\in P} N_p$ do
    Solve the lower bounding MILP (MDP-relax) at node $i$ where $x = \hat{x}_i$
    $LB_i = \min_{\xi\in\Xi}\{c(\hat{x}_i, (R^{pen})^L)\cdot\xi \mid A^{relax}\cdot\xi \le b^{relax}(\hat{x}_i, (R^{pen})^L)\}$
    if $\exists j \le i$ s. t. $UB_j \le LB_i$ then
        Prune node $i$
    else
        Solve the upper bounding MILP (MDP-diseng) at node $i$ where $x = \hat{x}_i$
        $UB_i = \min_{\xi\in\Xi}\{c(\hat{x}_i, (R^{pen})^U)\cdot\xi \mid A^{diseng}\cdot\xi \le b^{diseng}(\hat{x}_i, (R^{pen})^L)\}$
        if $\exists j \le i$ s. t. $UB_i \le LB_j$ then
            Prune node $j$
    end
end
```

6.2. Rewards iteration

For each node with certain storage tank size selection $\hat{x}$ that was not pruned in the bounding step, a rewards iteration algorithm that guarantees convergence is performed. We first illustrate the algorithm on a case with two stages ($k = 1, 2$) as shown in Fig. 9(a) and (b).

Each dot in Fig. 9(a) stands for a pair of availability values $(t\_ratio_1^A, t\_ratio_2^A)$ that is the result of a distinct combination of redundancy selection and maintenance policy (z_1, z_2, w_1, w_2) of the two stages. The number of these combinations is geometric with regard to the number of processing stages and design alternatives, but is finite.

Starting from an initial point $(t\_ratio_1^A, t\_ratio_2^A)^{(0)}$, we first substitute it into (31) and calculate $(R_1^{pen}, R_2^{pen})^{(0)}$, then solve the MILP (DMP-diseng) defined in (64) with $\tilde{R}^{pen}$ fixed to $(R_1^{pen}, R_2^{pen})^{(0)}$ to obtain the optimal design and maintenance policy in this case, $(z_1, z_2, w_1, w_2)^{(1)}$, which corresponds to a next blue point that the arrow leads to as shown in Fig. 9(a). The exact form of the MILP (DMP-diseng) is shown in Appendix B. As the iterations of the algorithm proceed, it is possible to get trapped into loops, which is also shown in Fig. 9(a). In that case the optimization step will be repeated at the point before the loop with the solutions corresponds to points in the loop excluded by integer cuts. As shown in Fig. 9(b), the algorithm stops when it converges to a stationary point $(t\_ratio_1^A, t\_ratio_2^A)$, where the optimization step leads back to itself. It is worth mentioning that the loop points are to be excluded from the feasible region of the MILP (DMP-diseng) only once when re-optimizing at the previous point to explore a different path, but not for future iterations, as the stationary property of a point is only proven when the optimization step leads back to itself among all the points without exception.

The algorithm is guaranteed to converge, because as discussed above, the number of the dots is finite, and the global optimum, which is a stationary point, is among them. As there can be several stationary points, the algorithm does not guarantee global optimality. However, as shown in Section 6.1, the lower bounds obtained in phase 1 of the algorithm are rigorous. We will also show later that for the examples we consider, the algorithm converges quickly to optimal or near optimal solutions. Algorithm 2 gives a detailed description of the algorithm. The convergence criterion depends on w, the binary action selection variables, instead of R^{pen}, because the same action selection will lead to the exact same probability distribution π and R^{pen}, and w as binary variables suffer less from computing precision issues.

The Rewards Iteration of the candidate nodes should be carried out in parallel, as the stationary point of a node is a valid upper bound for the node. Therefore, when a stationary point is found, any other node whose lower bound is greater that the objective value of this point should stop Rewards Iteration and be pruned.

7. Additional examples

7.1. Illustrative example revisited

In this section, we solve the example problem shown in Section 5 again with the proposed algorithm to minimize the total cost in (40). The MILPs are solved with CPLEX 12.8.1.1 in Pyomo

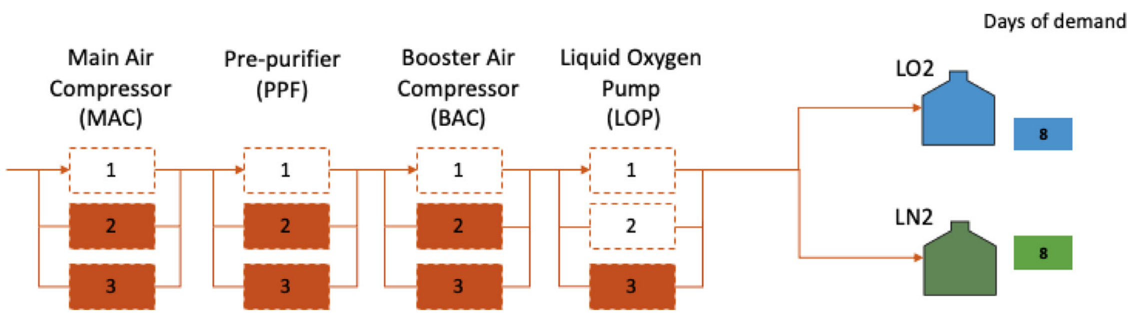


Fig. 10. Optimal design of the ASU example.

Algorithm 2: Reward parameters iteration.

```
Initialize: $i := 1$, $(R^{pen})^{(0)} := (R^{pen})^L$
$\xi^{(1)} := \arg\min_{\xi\in\Xi} \{c(\hat{x}, (R^{pen})^{(0)}) \cdot \xi \mid A\xi \le b(\hat{x}, (R^{pen})^{(0)})\}$
$(R^{pen})^{(1)} := R^{idv\_pen} \cdot \prod_{l\in K, l\neq k} (t\_ratio^A_k)^{(1)}$
while $w^{(i)} \neq w^{(i-1)}$ do
    if $\exists j \le i-1$ s. t. $w^{(j)} = w^{(i)}$ then
        $\xi^{(i)} := \arg\min_{\xi\in\Xi} \{c(\hat{x}, (R^{pen})^{(j-1)}) \cdot \xi \mid A\xi \le$
        $b(\hat{x}, (R^{pen})^{(j-1)}), \xi \neq \xi^{(n)}, j \le n \le i\}$
        $(R^{pen})^{(i)} := R^{idv\_pen} \cdot \prod_{l\in K, l\neq k} (t\_ratio^A_k)^{(i)}$
    else
        $\xi^{(i+1)} := \arg\min_{\xi\in\Xi} \{c(\hat{x}, (R^{pen})^{(i)}) \cdot \xi \mid A\xi \le$
        $b(\hat{x}, (R^{pen})^{(i)})\}$
        $(R^{pen})^{(i+1)} := R^{idv\_pen} \cdot \prod_{l\in K, l\neq k} (t\_ratio^A_k)^{(i+1)}$
        $i := i + 1$
    end
end
```

5.6.9, and the algorithm is implemented with Python 3.6 on Intel(R) Core(TM) i5 @ 1.60 GHz. Table 4 shows the results of the Enumeration and Bounding phase, where the storage size selection nodes are examined by rows left to right and top to bottom. 412.58 CPU seconds are spent in this phase. The nodes that are only showing LBs are pruned because their respective LBs are greater than the UB(s) of at least one node examined before them. The nodes where both LBs and UBs are calculated are pruned by the node(s) examined after them. For example, the node with 2 days of LO2 and 2 days of LN2 is pruned after the node with 8 days of LO2 and 2 days of LN2 is examined, which is later also pruned by the node with 8 days of LO2 and 8 days of LN2. It is guaranteed that the global optimal solution of the problem lies in the only highlighted node.

The Rewards Iteration phase at the node with 8 days of LO2 and 8 days of LN2 finds the stationary point after 1 iteration (28.44 CPUs), the objective value of which is 896.89.

To validate the results of the Rewards Iteration, the MINLP (DMP) restricted to the two nodes are also solved with the global solver BARON 17.4.1 on GAMS 24.8.5 platform to the global optimum of 896.21. It is confirmed that the decision variable values of the stationary solutions are the same as the ones obtained by BARON. The small difference in the objective values are likely due to computing precision issues. Therefore, it is guaranteed that the solution at the node with 8 days of LO2 and 8 days of LN2 is the global optimal solution.

Fig. 10 shows the optimal design, which selects the 2 cheapest units for the first three stages, and the cheapest one unit for LOP. The total capital cost is \$766k, and the expected operational cost is \$130.89k by the Rewards Iteration, and \$130.21k by directly solving the node with BARON.

Table 5 shows the maintenance policy for the booster air compressor (BAC) as an example, which is the same as that of the main air compressor (MAC). The table has four main columns. In the first column are the states with no unit being maintained or repaired. In the second and the third columns are the states with one unit being maintained or repaired. In the last column are the states where no unit is available. With the other redundancy on standby, the best action for the Stable state is to be inspected once a year. With the other unit Failed or Stopped, the best action for the Stable state is to be inspected every half month.

7.2. Demonstration of the algorithm's efficiency

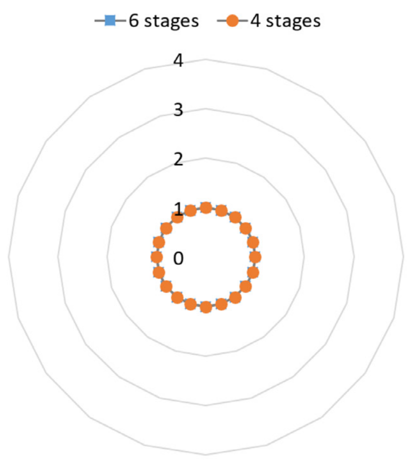
In order to test the efficiency of the algorithm, we randomly generate 20 problems around the example shown above. In particular, the reliability parameters and randomly perturbed within the range of $\pm$ 10%. Another group of 20 problems of 6 stages are randomly generated in the similar fashion and solved. The two additional stages are generated with data of similar orders of magnitudes. The computational statistics are shown in Fig. 11(a) and (b). It can be seen that the bounding step can prune most of the nodes for the problems of 4 stages (Fig. 11(a)), and the rewards iterations tend to converge fast at the remaining nodes. However, for the 6 stage problems (Fig. 11(b)), generally more than half of the nodes

Table 4
The results of bounds computation.

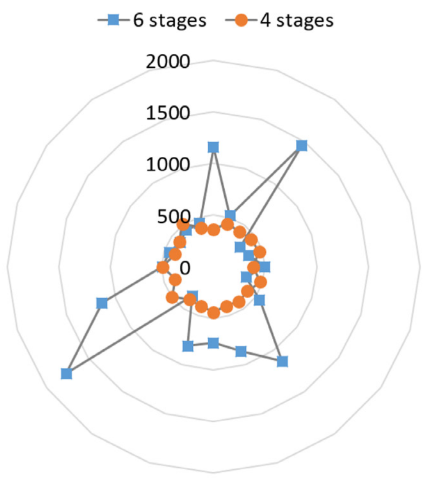
	2 days of LN2		8 days of LN2		14 days of LN2		20 days of LN2		30 days of LN2	
	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB
2 days of LO2	959.45	962.23	977.78		1045.93		1130.72		1279.27	
8 days of LO2	958.66	961.02	896.20	896.90	963.61		1048.37		1196.91	
14 days of LO2	1030.75		968.27		1035.17		1119.90		1268.45	
20 days of LO2	1123.26		1060.76		1127.65		1174.10		1303.80	
30 days of LO2	1287.11		1224.62		1289.57		1313.49		1433.93	

Table 5
Optimal action for the two units in the BAC stage.

	State	Action	State	Action	State	Action	State	Action
unit 2	Infant	None	Stable	Inspect-14	Stopped	Maintain	Failed	Repair
unit 3	Standby	None	Failed	Repair	Stable	Inspect-14	Failed	Repair
unit 2	Stable	Inspect-365	Infant	None	Failed	Repair	Failed	Repair
unit 3	Standby	None	Stopped	Maintain	Infant	None	Stopped	Maintain
unit 2	Worn-out	Stop	Infant	None	Worn-out	None	Stopped	Maintain
unit 3	Standby	None	Failed	Repair	Stopped	Maintain	Failed	Repair
unit 2	Standby	None	Worn-out	None	Failed	Repair		
unit 3	Stable	Inspect-365	Failed	Repair	Worn-out	None		
unit 2	Standby	None	Stopped	Maintain	Stopped	Maintain		
unit 3	Infant	None	Infant	None	Worn-out	None		
unit 2	Standby	None	Failed	Repair	Stable	Inspect-14		
unit 3	Worn-out	Stop	Stable	Inspect-14	Stopped	Maintain		



(a) Number of nodes to solve for each randomly generated problem



(b) Total CPUs for each randomly generated problem

Fig. 11. Computational results of randomly generated cases of two different sizes.

have to go through the Reward Iteration phase, and the CPU times tend to fluctuate more.

8. Conclusion

This paper considers redundancy selection, storage tank size selection as well as basic maintenance policies for a chemical process at the conceptual design phase. Markov Decision Process is used as the fundamental framework to model the stochastic dynamic decision making process of condition-based maintenance. We embed the optimal condition of Markov Decision Processes and the stationary probability distribution conditions of the reduced Markov Chain into an MINLP (DMP) that considers the economic trade-off among all major decisions. In order to make the model more solvable, we propose a standard linearization for the bilinear terms of binary variables and continuous variables, and a reformulation of the objective function that potentially provides a stronger relaxation of the objective.

An example based on the reliable design of an air separation unit is used to demonstrate how to extract the model parameters from the raw data. We attempted to solve the MINLP (DMP) directly with several global solvers and found that they would not be solved in reasonable amount of time. Therefore, we propose an algorithm that consists of two phases, Enumeration and Bounding, and Rewards Iteration. The validity of the bounding is based on the reformulation of the MDP objective function introduced earlier in the paper. Resolving the example shows that the two-phase algorithm greatly reduces the required computational effort. The algorithm also has consistent performance over 20 randomly generated problems around the original example of 4 processing stages. Another group of 20 random problems of 6 processing stages are also solved and show good computational results.

Credit author statement

Yixin Ye was the main author who did most of the writing. Ignacio Grossmann supervised her research, while Jose Pinto and Sivaraman Ramaswamy provided both data, directions and feedback on this work.

Declaration of Competing Interest

No conflict of interest.

Acknowledgment

The authors would like to acknowledge the financial supports from National Science Foundation under grant CBET-1705372, Linde plc, and Center for Advanced Process Decision-making at Carnegie Mellon University.

Appendix A. Objective function reformulation

Proposition 1. *The following equation holds*

$$\sum_{k\in K, h\in H_k} \sum_{s\in S_{k,h}} \pi_{k,h}(s) \nu_{k,h}(s)$$

$$= \frac{1}{1-\gamma} \cdot \sum_{k\in K, h\in H_k} \sum_{s\in S_{k,h}, a\in A_s} \pi_{k,h}^{sub}(s,a)$$

$$\cdot \sum_{s'\in S_{k,h}} P_{k,h}(s,a,s') \left(R_{k,h}^{idv}(s,a,s') + \sum_{p\in P} \sum_{n\in N_p} R\_y_{p,n,k,h}^{pen}(s,a,s') \right) \quad \text{(A.1)}$$

when constraints (12), (13), (19), (21)–(23), and (31)–(37) are satisfied.

Proof. First we focus on the optimal condition of a certain Markov Decision Process based on the notation of Section 3.1, where U_s^* is the optimal value of state s, and $a^*(s)$ is the optimal action of state s.:

$$U_s^* = \sum_{s'\in S} P(s, a^*(s), s') \cdot (R(s, a^*(s), s') + \gamma \cdot U_{s'}^*), \quad s \in S \quad \text{(A.2)}$$

Eq. (A.2) has the matrix form shown in (A.3), where $\mathbf{P}^*(s,s') = P(s, a^*(s), s')$, and similarly for $\mathbf{R}^*(s,s')$.

$$(1-\gamma)\mathbf{U}^* = \gamma(\mathbf{P}^* - \mathbf{I})\mathbf{U}^* + diag(\mathbf{P}^*(\mathbf{R}^*)^{\mathrm{T}}) \quad \text{(A.3)}$$

Next we rewrite Eq. (20) in the matrix form (A.4), which specifies the conditions that the stationary distribution $\boldsymbol{\pi}$ has to satisfy.

$$\pi^{\mathrm{T}}(\mathbf{P}^* - \mathbf{I}) = 0 \quad \text{(A.4)}$$

By left multiplying (A.3) with $\boldsymbol{\pi}^{\mathrm{T}}$ and substituting (A.4) into it, we obtain (A.5)

$$\pi^{\mathrm{T}}\mathbf{U}^* = \frac{1}{1-\gamma}\pi^{\mathrm{T}} diag(\mathbf{P}^*(\mathbf{R}^*)^{\mathrm{T}}) \quad \text{(A.5)}$$

Eq. (A.1) trivially holds for the stage designs that are not selected, where both sides of the equation are equal to 0. For the designs h that are selected for the respective stage k, when constraints (12), (13), (19), (21)–(23), and (27)–(37) are satisfied, (A.3) and (A.4) hold. Therefore, Eq. (A.5) holds, where the right-hand side with the subscripts h, k has the expression shown in (A.6), where $n^*(p)$ are the indices of the selected storage sizes of product p.

$$\frac{1}{1-\gamma} \sum_{k\in K, h\in H_k} \sum_{s\in S_{k,h}} \pi_{k,h}(s) \sum_{s'\in S_{k,h}} P_{k,h}(s, a^*(s), s')$$

$$\times \sum_{p\in P} R_{p,n^*(p),k,h}(s, a^*(s), s') \quad \text{(A.6)}$$

Since Eqs. (21) and (23) require that $\pi_{k,h}^{sub}(s, a^*(s)) = \pi_{k,h}(s)$, and that $\pi_{k,h}^{sub}(s,a) = 0$ if $a \neq a^*(s)$. (A.6) is equal to (A.7) when (21) and (23) hold.

$$\frac{1}{1-\gamma} \sum_{k\in K, h\in H_k} \sum_{s\in S_{k,h}, a\in A_s} \pi_{k,h}^{sub}(s,a) \sum_{s'\in S_{k,h}} P_{k,h}(s,a,s')$$

$$\times \sum_{p\in P} R_{p,n^*(p),k,h}(s,a,s') \quad \text{(A.7)}$$

As defined in Section 4.1, $R\_y_{p,n,k,h}^{pen}(s,a,s')$ is zero for non-selected storage sizes. Therefore, (A.7) can be further written as (A.8), which is the right hand side of (A.1).

$$\frac{1}{1-\gamma} \sum_{k\in K, h\in H_k} \sum_{s\in S_{k,h}, a\in A_s} \pi_{k,h}^{sub}(s,a) \sum_{s'\in S_{k,h}} P_{k,h}(s,a,s') (R_{k,h}^{idv}(s,a,s')$$

$$+ \sum_{p\in P} \sum_{n\in N_p} R\_y_{p,n,k,h}^{pen}(s,a,s')) \quad \text{(A.8)}$$

Thus, the proposition is proved. □

Appendix B. Objective bounding regarding key parameters

As shown in Section 6, the lower and upper bounds of the optimum of the MINLP (MDP) at each node of storage tank selection $\hat{x}$, $f(\hat{x}, \hat{R}^{pen}, \hat{\xi})$, are obtained by solving two MILPs:

Lower bounding MILP (DMP-relax):

$$\min_{\xi\in\Xi}\{c(\hat{x}, (R^{pen})^L) \cdot \xi \mid A^{relax} \cdot \xi \leq b^{relax}(\hat{x}, (R^{pen})^L)\} \quad \text{(B.1)}$$

Upper bounding MILP (DMP-diseng):

$$\min_{\xi\in\Xi}\{c(\hat{x}, (R^{pen})^U) \cdot \xi \mid A^{diseng} \cdot \xi \leq b^{diseng}(\hat{x}, (R^{pen})^L)\} \quad \text{(B.2)}$$

In the following, we will display their exact formulation, where $\tilde{R}^{pen}$ can be replaced with $(R^{pen})^L$ or $(R^{pen})^U$ depending on the needs, and prove in Proposition 2 and 3 that the MILPs provide valid bounds. (B.3) shows the exact form of $c(\hat{x}, \tilde{R}^{pen}) \cdot \xi$.

$$c(\hat{x}, \tilde{R}^{pen}) \cdot \xi = \sum_{k\in K, h\in H_k} z_{k,h} C_{k,h}^U$$

$$+ \frac{1}{1-\gamma} \sum_{k\in K, h\in H_k} \sum_{s\in S_{k,h}, a\in A_s} \pi_{k,h}^{sub}(s,a)$$

$$\times \sum_{s'\in S_{k,h}} P_{k,h}(s,a,s')(R_{k,h}^{idv}(s,a,s') + \sum_{p\in P} \sum_{n\in N_p} \hat{x}_{p,n} \cdot \tilde{R}_{p,n,k,h}^{pen}(s,a,s')) \quad \text{(B.3)}$$

$A^{relax} \cdot \xi \leq b(\hat{x}, \tilde{R}^{pen})$ consists of (7), (21)–(23), (B.5) and (B.6). $A^{diseng} \cdot \xi \leq b(\hat{x}, \tilde{R}^{pen})$ consists of (7), (21)–(23), (B.4) and (B.6).

$$v_{k,h}(s) \leq \sum_{s'\in S_{k,h}} (P_{k,h}(s,a,s')$$

$$\cdot (R_{k,h}^{idv}(s,a,s') + \sum_{p\in P} \sum_{n\in N_p} z_{k,h}\hat{x}_{p,n}\tilde{R}_{p,n,k,h}^{pen}(s,a,s')))$$

$$+ \gamma \sum_{s'\in S_{k,h}} P_{k,h}(s,a,s') \cdot v_{k,h}(s'), \quad \forall k \in K, h \in H_k, s \in S_{k,h}, a \in A(s) \quad \text{(B.4)}$$

$$M(w_{k,h}(s,a) - 1) + v_{k,h}(s) \leq \sum_{s'\in S_{k,h}} (P_{k,h}(s,a,s')$$

$$\cdot (R_{k,h}^{idv}(s,a,s') + \sum_{p\in P} \sum_{n\in N_p} z_{k,h}\hat{x}_{p,n}\tilde{R}_{p,n,k,h}^{pen}(s,a,s')))$$

$$+ \gamma \sum_{s'\in S_{k,h}} P_{k,h}(s,a,s') \cdot v_{k,h}(s'), \quad \forall k \in K, h \in H_k, s \in S_{k,h}, a \in A(s) \quad \text{(B.5)}$$

$$M(1 - w_{k,h}(s,a)) + v_{k,h}(s) \geq \sum_{s'\in S_{k,h}} (P_{k,h}(s,a,s')$$

$$\cdot (R_{k,h}^{idv}(s,a,s') + \sum_{p\in P} \sum_{n\in N_p} z_{k,h}\hat{x}_{p,n}\tilde{R}_{p,n,k,h}^{pen}(s,a,s')))$$

$$+ \gamma \sum_{s'\in S_{k,h}} P_{k,h}(s,a,s') \cdot v_{k,h}(s'), \quad \forall k \in K, h \in H_k, s \in S_{k,h}, a \in A(s) \quad \text{(B.6)}$$

Proposition 2. *If $(R^{pen})^L \preceq \hat{R}^{pen}$, then (B.1) is a valid lower bound of $f(\hat{x}, \hat{R}^{pen}, \hat{\xi})$.*

Proof. As shown in Section 6, assuming that $\hat{R}^{pen}$ is known, the result of (B.7) is $f(\hat{x}, \hat{R}^{pen}, \hat{\xi})$.

$$\min_{\xi\in\Xi}\{c(\hat{x}, \hat{R}^{pen}) \cdot \xi \mid A^{diseng} \cdot \xi \leq b^{diseng}(\hat{x}, \hat{R}^{pen})\}. \quad \text{(B.7)}$$

As a constraint in (B.1), (B.5) relaxes the requirement that the state value should be less than or equal to the right-hand-side corresponding to every possible action, and only requires that it be no greater than the selected action. That makes all the possible action decisions (possible solution of variable w) feasible to (B.1), which automatically includes the action decision $\hat{w}$ corresponding to the optimal solution of (B.7), $\hat{\xi}$.

Arguably, substituting $\hat{w}$ into (B.1) will lead to different state values than $\hat{v}$, but $\hat{\pi}^{sub}$ will stay the same, as certain action decision $\hat{w}$ leads to unique reduced Markov Processes for each stage, and unique stationary probability distributions $\hat{\pi}$ and $\hat{\pi}^{sub}$. Since only $\hat{\pi}^{sub}$ goes into the objective function, we can still insert $\hat{\pi}^{sub}$ as part of $\hat{\xi}$ into the objective function of (B.1) as a feasible solution, which by the definition of optimality is no less than the optimum of (B.1), and let (B.8) hold with a little abuse of the notation.

$$c(\hat{x}, (R^{pen})^L) \cdot \hat{\xi} \geq \min_{\xi \in \Xi}\{c(\hat{x}, (R^{pen})^L) \cdot \xi \mid A^{relax} \cdot \xi \leq b^{relax}(\hat{x}, (R^{pen})^L)\} \tag{B.8}$$

On the other hand, since $(R^{pen})^L \preceq \hat{R}^{pen}$, (B.9) holds given the exact formulation of the objective function shown in (B.3).

$$c(\hat{x}, (R^{pen})^L) \cdot \hat{\xi} \leq c(\hat{x}, \hat{R}^{pen}) \cdot \hat{\xi} \equiv f(\hat{x}, \hat{R}^{pen}, \hat{\xi}) \tag{B.9}$$

Therefore, we have (B.10) and the proposition is proved.

$$f(\hat{x}, \hat{R}^{pen}, \hat{\xi}) \geq \min_{\xi \in \Xi}\{c(\hat{x}, (R^{pen})^L) \cdot \xi \mid A^{relax} \cdot \xi \leq b^{relax}(\hat{x}, (R^{pen})^L)\} \tag{B.10}$$

□

Proposition 3. *If $(R^{pen})^L \preceq \hat{R}^{pen} \preceq (R^{pen})^U$, then (B.2) is a valid upper bound of $f(\hat{x}, \hat{R}^{pen}, \hat{\xi})$.*

Proof. As shown in Section 6, assuming that $\hat{R}^{pen}$ is known, the result of (B.7) is $f(\hat{x}, \hat{R}^{pen}, \hat{\xi})$. Also, we denote ξ^L as the optimal solution of (B.2).

Constraints (B.4) and (B.6) are the only constraints in (B.2) that contains R^{pen}. Since $(R^{pen})^L \preceq \hat{R}^{pen}$, ξ^L which satisfies (B.4) in $A^{diseng} \cdot \xi \leq b^{diseng}(\hat{x}, (R^{pen})^L)$ also satisfies (B.4) in $A^{diseng} \cdot \xi \leq b^{diseng}(\hat{x}, \hat{R}^{pen})$. As for (B.6), since exactly one action is selected for each state, v is always the unique solution of a square linear system and self-bounded by the parameters. Therefore, the big M can be made large enough such that (B.6) is always satisfied.

Therefore, ξ^L is a feasible solution of (B.7), and gives an objective value no less than the optimum as shown in (B.11)

$$c(\hat{x}, \hat{R}^{pen}) \cdot \xi^L \geq c(\hat{x}, \hat{R}^{pen}) \cdot \hat{\xi} \equiv f(\hat{x}, \hat{R}^{pen}, \hat{\xi}) \tag{B.11}$$

On the other hand, since $\hat{R}^{pen} \preceq (R^{pen})^U$, (B.12) holds given the exact formulation of the objective function shown in (B.3).

$$\begin{aligned} c(\hat{x}, \hat{R}^{pen}) \cdot \xi^L &\leq c(\hat{x}, (R^{pen})^U) \\ \cdot \xi^L &\equiv \min_{\xi \in \Xi}\{c(\hat{x}, (R^{pen})^U) \cdot \xi \mid A^{diseng} \cdot \xi \leq b^{diseng}(\hat{x}, (R^{pen})^L)\} \end{aligned} \tag{B.12}$$

Therefore, we have (B.13) and the proposition is proved.

$$f(\hat{x}, \hat{R}^{pen}, \hat{\xi}) \leq \min_{\xi \in \Xi}\{c(\hat{x}, (R^{pen})^U) \cdot \xi \mid A^{diseng} \cdot \xi \leq b^{diseng}(\hat{x}, (R^{pen})^L)\} \tag{B.13}$$

□

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.compchemeng.2020.107052.

References

Achkar, V.G., Cafaro, V.G., Mendez, C.A., Cafaro, D.C., 2019. Discrete-time MILP formulation for the optimal scheduling of maintenance tasks on oil and gas production assets. Ind. Eng. Chem. Res. 58 (19), 8231–8245.

Aguilar, O., Kim, J.-K., Perry, S., Smith, R., 2008. Availability and reliability considerations in the design and optimisation of flexible utility systems. Chem. Eng. Sci. 63 (14), 3569–3584.

Ahiska, S.S., Appaji, S.R., King, R.E., Warsing Jr., D.P., 2013. A Markov decision process-based policy characterization approach for a stochastic inventory control problem with unreliable sourcing. Int. J. Prod. Econ. 144 (2), 485–496.

Amaran, S., Zhang, T., Sahinidis, N.V., Sharda, B., Bury, S.J., 2016. Medium-term maintenance turnaround planning under uncertainty for integrated chemical sites. Comput. Chem. Eng. 84, 422–433.

Amari, S.V., McLaughlin, L., Pham, H., 2006. Cost-effective condition-based maintenance using Markov decision processes. In: RAMS'06. Annual Reliability and Maintainability Symposium, 2006. IEEE, pp. 464–469.

Avriel, M., 2003. Nonlinear Programming: Analysis and Methods. Courier Corporation.

Barlow, R.E., Proschan, F., 1975. Statistical theory of reliability and life testing: probability models. Technical Report. Florida State Univ Tallahassee.

Bäuerle, N., Rieder, U., 2011. Markov Decision Processes with Applications to Finance. Springer Science & Business Media.

Bellman, R., 1957. A Markovian decision process. J. Math. Mech. 6 (5), 679–684.

Byon, E., Ding, Y., 2010. Season-dependent condition-based maintenance for a wind turbine using a partially observed Markov decision process. IEEE Trans. Power Syst. 25 (4), 1823–1834.

Castro, P.M., Erdirik-Dogan, M., Grossmann, I.E., 2008. Simultaneous batching and scheduling of single stage batch plants with parallel units. AIChE J. 54 (1), 183–193.

Chart, 2020. Standard cryogenic tanks range. https://www.chartindustries.com/Industry/Industry-Products/Bulk-Storage-Tanks/Standard-Bulk-Tanks. Accessed 9-July-2020.

Chen, D., Trivedi, K.S., 2005. Optimization for condition-based maintenance with semi-Markov decision process. Reliab. Eng. Syst. Saf. 90 (1), 25–29.

Cheung, K.-Y., Hui, C.-W., Sakamoto, H., Hirata, K., O'Young, L., 2004. Short-term site-wide maintenance scheduling. Comput. Chem. Eng. 28 (1), 91–102.

d'Epenoux, F., 1963. A probabilistic production and inventory problem. Manag. Sci. 10 (1), 98–108.

Glover, F., 1975. Improved linear integer programming formulations of nonlinear integer problems. Manag. Sci. 22 (4), 455–460.

Godoy, E., Benz, S., Scenna, N., 2015. An optimization model for evaluating the economic impact of availability and maintenance notions during the synthesis and design of a power plant. Comput. Chem. Eng. 75, 135–154.

Henley, E.J., Kumamoto, H., 1981. Reliability Engineering and Risk Assessment, 568. Prentice-Hall Englewood Cliffs (NJ).

Liang, K.-H., Chang, C.-T., 2008. A simultaneous optimization approach to generate design specifications and maintenance policies for the multilayer protective systems in chemical processes. Ind. Eng. Chem. Res. 47 (15), 5543–5555.

Linde, plc, 2020. Cryogenic tanks. https://www.linde-engineering.com/en/plant_components/cryogenic_tanks/index.html. Accessed 9-July-2020.

McCormick, G.P., 1976. Computability of global solutions to factorable nonconvex programs: part I—Iconvex underestimating problems. Math. Program. 10 (1), 147–175.

Nápoles-Rivera, F., Serna-González, M., Bin-Mahfouz, A., Jiménez-Gutiérrez, A., El-Halwagi, M.M., Marı, J., et al., 2013. Simultaneous optimization of energy management, biocide dosing and maintenance scheduling of thermally integrated facilities. Energy Convers. Manag. 68, 177–192.

Pistikopoulos, E.N., Vassiliadis, C.G., Arvela, J., Papageorgiou, L.G., 2001. Interactions of maintenance and production planning for multipurpose process plants a system effectiveness approach. Ind. Eng. Chem. Res. 40 (14), 3195–3207.

Powell, W.B., 2004. Handbook of Learning and Approximate Dynamic Programming, 2. John Wiley & Sons.

Puterman, M.L., 2014. Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons.

Redutskiy, Y., 2017. Optimization of safety instrumented system design and maintenance frequency for oil and gas industry processes. Manag. Prod. Eng. Rev. 8.

Shin, J., Lee, J.H., Realff, M.J., 2017. Operational planning and optimal sizing of microgrid considering multi-scale wind uncertainty. Appl. Energy 195, 616–633.

Tan, J.S., Kramer, M.A., 1997. A general framework for preventive maintenance optimization in chemical process operations. Comput. Chem. Eng. 21 (12), 1451–1469.

Terrazas-Moreno, S., Grossmann, I.E., Wassick, J.M., Bury, S.J., 2010. Optimal design of reliable integrated chemical production sites. Comput. Chem. Eng. 34 (12), 1919–1936.

Thomaidis, T., Pistikopoulos, E., 1994. Integration of flexibility, reliability and maintenance in process synthesis and design. Comput. Chem. Eng. 18, S259–S263.

Thomaidis, T.V., Pistikopoulos, E., 1995. Optimal design of flexible and reliable process systems. IEEE Trans. Reliab. 44 (2), 243–250.

Van Rijn, C., 1987. A systems engineering approach to reliability, availability, and maintenance. In: Conf. on Foundation of Computer Aided Operations, FOCAP-O, pp. 221–251.

Vassiliadis, C., Pistikopoulos, E., 2001. Maintenance scheduling and process optimization under uncertainty. Comput. Chem. Eng. 25 (2–3), 217–236.

White, D.J., 1993. A survey of applications of Markov decision processes. J. Oper. Res. Soc. 44 (11), 1073–1096.

Wibisono, E., Adi, V.S.K., Chang, C.-T., 2014. Model based approach to identify optimal system structures and maintenance policies for safety interlocks with time-varying failure rates. Ind. Eng. Chem. Res. 53 (11), 4398–4412.

Ye, Y., Grossmann, I.E., Pinto, J., Ramaswamy, S., 2020. Integrated redundancy and storage design optimization for reliable ASU based on Markov chain—A game theoretic solution. Ind. Eng. Chem. Res. 59 (6), 2491–2504.

Ye, Y., Grossmann, I.E., Pinto, J.M., 2018. Mixed-integer nonlinear programming models for optimal design of reliable chemical plants. Comput. Chem. Eng. 116, 3–16.

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. Comput. Chem. Eng. 124, 381–404.

Yin, K.K., Liu, H., Johnson, N.E., 2002. Markovian inventory policy with application to the paper industry. Comput. Chem. Eng. 26 (10), 1399–1413.