An efficient method for optimal design of large-scale integrated chemical production sites with endogenous uncertainty

Sebastian Terrazas-Moreno [a], Ignacio E. Grossmann [a,*], John M. Wassick [b], Scott J. Bury [b], Naoko Akiya [b]

[a] Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, United States
[b] The Dow Chemical Company, Midland, MI 48674, United States

ABSTRACT

Integrated sites are tightly interconnected networks of large-scale chemical processes. Given the large-scale network structure of these sites, disruptions in any of its nodes, or individual chemical processes, can propagate and disrupt the operation of the whole network. Random process failures that reduce or shut down production capacity are among the most common disruptions. The impact of such disruptive events can be mitigated by adding parallel units and/or intermediate storage. In this paper, we address the design of large-scale, integrated sites considering random process failures. In a previous work (Terrazas-Moreno et al., 2010), we proposed a novel mixed-integer linear programming (MILP) model to maximize the average production capacity of an integrated site while minimizing the required capital investment. The present work deals with the solution of large-scale problem instances for which a strategy is proposed that consists of two elements. On one hand, we use Benders decomposition to overcome the combinatorial complexity of the MILP model. On the other hand, we exploit discrete-rate simulation tools to obtain a relevant reduced sample of failure scenarios or states. We first illustrate this strategy in a small example. Next, we address an industrial case study where we use a detailed simulation model to assess the quality of the design obtained from the MILP model.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

The optimal design and operation of integrated production networks is a current and future opportunity in the chemical process industry. For instance, The Dow Chemical Company owns Texas Operations, an integrated site that manufactures 21% of the company's products sold globally (Wassick, 2009). BASF's site in Ludwigshafen is another example of a large integrated production system with over 200 production plants (BASF, 2010). Both of these sites began as smaller manufacturing facilities and grew in capacity and complexity over many decades. In contrast with the gradual integration of these heritage sites, recent strategic initiatives require the grassroots design of very large integrated process networks. The joint venture between Saudi Aramco and The Dow Chemical Company to construct and operate a world-scale chemical and plastic production complex in Saudi Arabia is an example of such an initiative (Dow, 2007).

These integrated sites feature different interconnected process networks. A failure event that reduces the production rate of any of the processes can propagate throughout the network. Some events are planned, for example, preventive maintenance of major

plant components; the remaining failure events occur at random, requiring corrective maintenance. The focus of this work is on the second type of events, namely, failure modes that decrease production capacity and that occur at random times with random repair durations. The industrial significance of this problem is illustrated in a paper by Miller, Owens, and Deans (2006) from The Dow Chemical Company. These authors explain the benefit of designing reliability into manufacturing systems and illustrate the scope of the involvement of Reliability–Availability–Maintainability (RAM) teams during the design of large-scale manufacturing systems. A note on terminology; availability is the ratio of uptime to total time or the fraction of time the unit is producing product, reliability is the probability of a unit or piece of equipment being in an up state at a particular time.

To study these stochastic failures, computer simulations are commonly used to test the effect of design alternatives in the availability or effective capacity of integrated systems. This approach is applicable to an integrated chemical facility. As expected for an integrated system, the increasing number of design degrees of freedom and the increasing number of stochastic inputs quickly increases the difficulty and time to search the design space, requiring more computing and modeling resources. In addition, the successful ranking and selection of options is often challenging to do cleanly when the search space is large. We believe that mathematical programming techniques can be used in

conjunction with process simulation tools to provide an efficient tool for improving the design of integrated sites by significantly reducing the design space. This new space can be then thoroughly explored and validated via simulation. Additional opportunities for applying optimization to the solution of the design, planning, and scheduling of integrated sites can be found in Wassick (2009).

In our previous work (Terrazas-Moreno, Grossmann, Wassick, & Bury, 2010), we proposed a mixed-integer linear programming (MILP) model for the optimal design of integrated sites subject to random failures and random supply and demand. The optimization criteria were the maximization of the probability of meeting customer demands and the minimization of capital investment. To be a practical and useful tool for design teams, this challenging model must be solved efficiently. A literature review of optimization approaches for related problems, which includes the works by Davies and Swartz (2008), Pistikopoulos, Thomaidis, Melin, and Ierapetritou (1996), Pistikopoulos, Vassiliados, Arvela, and Papageorgiou (2001), and Straub and Grossmann (1990, 1993), can be found in that paper.

In the present paper, we propose a novel algorithm based on Benders decomposition (BD) (Benders, 1962) to solve large-scale instances of the MILP model mentioned above. There is rich literature on the application of BD to the design of process systems under uncertainty. Most of these applications model uncertainty using a stochastic programming (SP) representation and apply variations of the BD algorithm. The standard decomposition technique is referred to as L-shaped decomposition (Van Slyke & Wets, 1969) in the stochastic programming literature. Straub and Grossmann (1993) proposed a nonlinear programming (NLP) model for maximizing the feasible operating region of a network with uncertain process parameters and used Generalized Benders Decomposition (GBD) (Geoffrion, 1972) to solve this problem. A similar approach was proposed by Pistikopoulos (1995) and applied by Ierapetritou, Acevedo, and Pistikopoulos (1996) and Acevedo and Pistikopoulos (1998) as a general algorithmic technique for solving a class of problems defined as process design and operations under uncertainty. More recently, Liu, Fan, and Ordonez (2009) addressed the design of reliable transportation networks subject to unpredictable natural disasters using Generalized Benders Decomposition (GBD).

The work by Santoso, Ahmed, Goetschalckx, and Shapiro (2005) is related to the model and algorithm we present in this paper, although they deal with exogenous uncertainties as opposed to endogenous uncertainties in our case. The paper by Santoso et al. (2005) proposes a two-stage stochastic programming (SP) model to optimize the design of a supply chain network under uncertainty. The authors consider in their MILP model uncertainty in parameters such as processing cost, raw material supply, finished product demand, and processing capacity of manufacturing facilities. These uncertain parameters are discretized in order to build scenarios with different combinations of parameter values. The resulting number of scenarios can be huge for realistic problem instances. The paper proposed an algorithm based on BD where the 1st stage network design variables are considered complicating. An interesting feature of this paper is that Benders decomposition is enhanced with convergence accelerating techniques based on three ideas. The first is adding constraints besides the usual dual cuts to the master problem that can be derived as strengthened dual cuts or from constraints expressed in terms of variables in the master problem that were redundant and not included in the full space model. The second idea is a heuristic for finding good feasible solutions. The third is a trust region algorithm that prevents the master problem from oscillating wildly in the first iterations.

All of the implementations of BD and GBD mentioned above correspond to SP problems with *exogenous* uncertainties. That is, the stochastic process is independent of design decisions (Jonsbraten, Wets, & Woodruff, 1998). If we exclude exogenous uncertainties

like demand and raw material supply, then the source of remaining uncertainties are random process failures, which are dependent on design decisions. This implicitly assumes that when a unit is "up" that it operates at the design rates. The number and selection of parallel processing units in the network are not known *a priori*, i.e., they are decision variables. Since the only processes where random failures can occur are those that are selected from the superstructure, the realizations of uncertainties are also a function of the decision variables. This type of uncertainty, defined as *endogenous*, significantly increases the complexity of the problem as well as the computational resources required for solving it. This paper proposes a novel implementation of Benders decomposition for two-stage stochastic programming (SP) problems with endogenous uncertainties. The technique we propose partly overcomes the combinatorial explosion in problem size that occurs with non-anticipativity constraints required in this type of problems (Jonsbraten, Wets, & Woodruff, 1998).

In the following sections, we present the problem of optimal design of integrated sites and its representation as a two-stage MILP stochastic programming problem. Some important properties of the problem are given, and the decomposition approach that exploits these properties is presented. Finally, we test the methodology with two numerical examples, one of them being an industrial case study. As part of the solution of this case study, we analyze a scenario reduction technique, and we report the results of simulating the operation of the Pareto-optimal designs with a commercial simulation tool, ExtendSim® (Imagine That Inc., 2010).

2. Problem statement

In this section, we describe the problem of optimal design of highly available integrated sites.

The following list of given data, degrees of freedom, optimization criteria, and assumptions is reproduced with minor changes from Terrazas-Moreno et al. (2010).

Given are:

- The superstructure of an integrated site with allowable parallel production units in each plant and intermediate storage tanks.
- A set of materials that the plants consume and produce.
- Mass balance coefficients for all units in the superstructure.
- The maximum capacity that can be assigned to each unit in the integrated site.
- Maximum supply of raw materials and maximum demand of finished products.
- Number of failure modes, production rate loss as a result of each failure, and the time between failure (TBF) and time to repair (TTR) per failure mode, either as probability distributions or mean values (MTBF and MTTR, respectively).
- A cost function that relates design decisions with capital investment.

The problem is to determine:

- The number of production units for each plant.
- The capacity of each unit.
- Sizes of intermediate storage between plants.
- For each state, material flows between plants and rate of accumulation or depletion of material in storage.

The objective is to determine the set of Pareto-optimal solutions that:

- Maximize the average production rate at which the integrated site supplies chemicals to an external market.

• Minimize the capital investment.

An interesting aspect of our approach is the development of a set of discrete states that correspond to all possible combinations of failure modes in the integrated site and a set of frequency and duration equations that allows us to calculate the mean residence time *mrt* and frequency of encounters *fr* of all possible states in the system, based only on the knowledge of the *MTTR* and *MTTF* of the units in the superstructure (Billinton & Allan, 1992).

The following assumptions and simplifications are made:

1. Random failures are independent events.
2. The cost function is represented by piece-wise linear approximations.
3. The production units in the plants are dedicated single product continuous processes.

Multiple failures in real integrated systems can be causally related. If this is the case, our first assumption will result in inaccurate statistical information of the discrete states described above. In practice, this is rare except at the equipment component level, which is at a level of detail finer than the failures considered in this work. Consequently, most reliability simulation is carried out assuming independent failure modes. In this sense, the independence assumption made in this work will not add any more inaccuracy than what is regularly used in industrial simulations. Statistically uncorrelated events allow us to determine discrete state probabilities, residence times, frequency, etc. using standard frequency and duration calculations (Billinton & Allan, 1992). Using a piecewise linear approximation of the objective function preserves linearity in the problem. The number of piecewise functions can be increased or decreased depending on the level of precision required in cost calculations. Finally, the third assumption limits the applicability of the approach to systems that can be modeled as continuous dedicated plants. Multiproduct plants would require scheduling considerations and a system of multiple storage tanks.

3. Overview of solution strategy

The solution approach we use in this paper integrates simulation and optimization tools. The simulation is built as a discrete rate model in ExtendSim software (Imagine That Inc., 2010). In a discrete rate model, flow rates and tank levels are updated only when needed, by an internal linear programming (LP) solver at each discrete event. In our problem, events are failures and repairs that change the rate of flow of materials among plants and changes in tank level status (e.g. full, empty, high, low). The optimization step involves a mixed-integer stochastic programming representation of the integrated site and exploits state-of-the-art computational technology for solving mixed-integer programs. The main idea of stochastic programming (SP) is to build a set of scenarios that consists of discrete realizations of uncertainties in the problem, and then optimizing the expected value of the objective criterion over all possible scenarios. In our problem, the scenarios correspond to the discrete failure states.

Combining the two technologies (simulation and optimization) exploits their complementary advantages. Discrete rate simulation is able to represent the operation of the integrated site in great detail. In fact, these types of models have been used at The Dow Chemical Company to simulate real manufacturing systems and validate them against actual operating data. Each simulation run, however, requires that design variables be fixed. Searching for an optimal design requires enumeration techniques that are time consuming and provide no guarantee of finding the optimal solution. By building and solving mixed-integer linear programming (MILP)
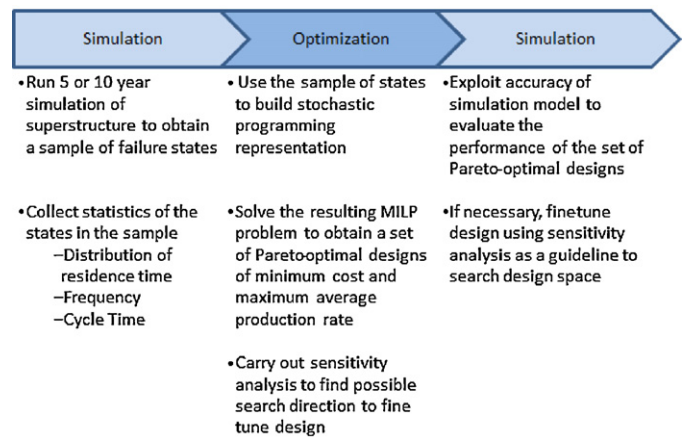

Simulation	Optimization	Simulation
• Run 5 or 10 year simulation of superstructure to obtain a sample of failure states	• Use the sample of states to build stochastic programming representation	• Exploit accuracy of simulation model to evaluate the performance of the set of Pareto-optimal designs
• Collect statistics of the states in the sample —Distribution of residence time —Frequency —Cycle Time	• Solve the resulting MILP problem to obtain a set of Paretooptimal designs of minimum cost and maximum average production rate	• If necessary, finetune design using sensitivity analysis as a guideline to search design space
	• Carry out sensitivity analysis to find possible search direction to fine tune design	

Fig. 1. Summary of solution strategy using simulation and optimization tools.

models, one can find a guaranteed optimal solution using state-of-the-art MILP solvers such as CPLEX (ILOG, 2011). MILP models, however, are equation-oriented: the objective function and operational constraints have to be expressed in algebraic terms. In order to do so, the modeler is required to make certain simplifications of real systems. We use optimization to find a set of Pareto-optimal designs using an algebraic model of the system and simulation to evaluate in more detail the performance of these candidate designs. If the number of scenarios (discrete states) used in the optimization model is impractically large, as is the case in when there is a large number of possible failure modes, we also use the simulation model to generate a sample of the most representative scenarios. Finally, we determine some sensitivity analysis derived from the optimal solution to provide guidelines for fine-tuning the design using the simulation model. Fig. 1 summarizes the solution process described above.

The following sections describe in more detail each of the elements in the simulation and optimization solution approach. First, we develop the MILP model of the integrated sites and explain how the resulting formulation corresponds to a two-stage stochastic programming (SP) problem with endogenous uncertainties. The SP formulation relies on constructing a set of failure scenarios that can be impractically large for the problem at hand. To overcome the computational complexity involved in solving large instances of the problem, we propose a novel decomposition algorithm based on Benders decomposition. Next, we describe the discrete-rate model used to simulate the integrated site. Finally, we present two case studies: an academic example and an industrial case study. The first example is meant to illustrate the results of the optimization approach. Since the number of failure modes is small, we can enumerate all discrete failure states, and there is no need for a simulation model to obtain a representative sample of states. We use this small example to introduce the methodology for sensitivity analysis. Since the industrial case study is significantly larger, it requires the simulation–optimization–simulation sequence depicted in Fig. 1. We use this example to explain the methodology for constructing a sample of failure states.

An important note to keep in mind is that in the SP representation, the discrete states that correspond to combinations of failure modes are called scenarios. For our purposes, the terms states and scenarios are equivalent, but the latter is used in some of the remaining sections in order to be consistent with the SP literature.

4. Mixed-integer linear programming model

The starting point for the mathematical model of the integrated site is a superstructure that corresponds to a network of
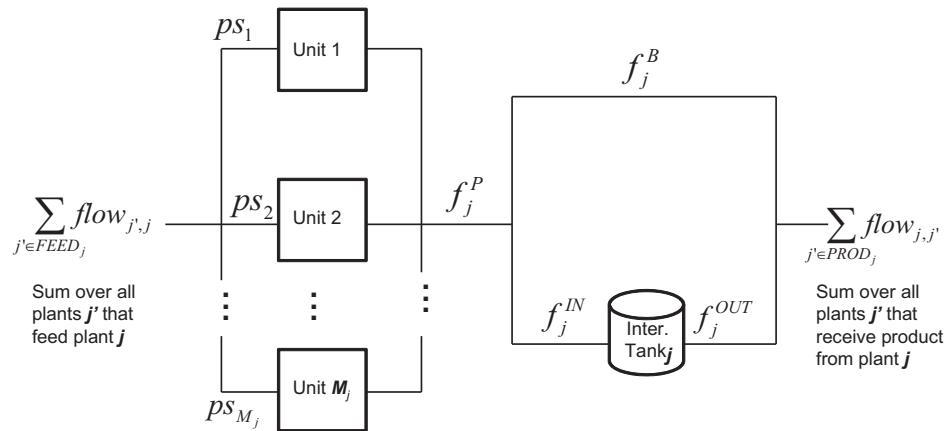
Fig. 2. Building block for plant j in the integrated site.

processes and storage tanks. Each of the nodes in the network has the structure shown in Fig. 2. The variables are described in the nomenclature section in Appendix A.

In previous work (Terrazas-Moreno et al., 2010), a mixed-integer linear programming (MILP) formulation for solving this problem was proposed in which exogenous uncertainties in supply and demand were also included. The most important features of the formulation are: (i) a superstructure that contains all potential parallel units and storage tanks in all plants in the process network, (ii) a state-space representation of the integrated site where each discrete state corresponds to a combination of simultaneous failure modes, and (iii) a model of intermediate storage, based on the concept of random walks (Heyman & Sobel, 1982), to determine the average and variance of the levels of material in the storage tanks as a function of the network design. The transitions between states (random process failures and repairs) follow the behavior of a continuous time Markov chain (Heyman & Sobel, 1982). This approach allows the calculation of the mean and variance of the time spent in each state, the frequency of visits to each of them, and the probabilities of finding the integrated site in any state using statistical information from historical reliability data of existing processes that resemble the ones postulated in the superstructure.

All of the above elements are integrated in an MILP formulation that maximizes the expected stochastic flexibility [E(SF)] of finished products and minimizes the capital investment required by the network design (Terrazas-Moreno et al., 2010). The bi-criterion optimization problem is solved using the ε-constrained method (Ehrgott, 2005). The degrees of freedom are the selection of units from the superstructure, the size and location of intermediate storage tanks, and the production capacity of the plants. An important difference between our previous paper and the present one is that here we maximize average production rate instead of expected stochastic flexibility (Straub & Grossmann, 1990). The system resides during a portion of the operating horizon in states where some of the components are affected by failure modes that temporarily decrease production rate. Maximizing average production rate over a long operating time involves designing the system so that the effect of random failures is minimized and, therefore, has a similar effect on the system design as the objective of maximizing E(SF). Computation of E(SF) relies on the criterion of whether or not the system can match the demand rate in each of the discrete failure states and fails to capture the difference between a state where the production rate is slightly less than the demand rate from one where the entire system is shut down. The objective of maximizing long-term average production rate better matches industrial design criteria than the maximization of E(SF). Appendix of this paper contains the complete mathematical formulation. Details of

the model and a description of each of the constraints can be found in Terrazas-Moreno et al. (2010).

5. Stochastic programming representation

We model the problem of optimal design of an integrated site (IS) as a two-stage stochastic mixed-integer linear program. The vector of first stage design decisions d includes binary variables to represent the selection of production units from a superstructure and continuous variables, such as production unit capacities and storage tank sizes. Stage two decisions only involve continuous variables. A number of failure modes contained in a set L can occur in the production units *of the superstructure* at random times. This fact introduces endogenous uncertainty to the operation of the IS. Furthermore, this uncertainty is of a discrete nature (whether or not failure $\ell \in L$ occurs) and can be modeled using a parameter y^ℓ that is 0 if failure ℓ occurs and 1 otherwise. The probability of a failure ℓ occurring at any point in time is $prob^\ell$. We define a vector $y = \{y^\ell\}$ where $\ell = 1, \ldots, |L|$; this vector has zeros in the positions corresponding to failures occurring simultaneously at any given time in the IS. There is a finite number of possible 0–1 combinations for the vector y. Each of these combinations defines a scenario in set S. Therefore, each scenario $s \in S$ corresponds to an instance of the vector y, and it can be represented as $y_s = \{y_s^\ell\}$, $\ell = 1, \ldots, |L|$. Assuming independent probabilities of failures, the probability associated with each scenario is $p_s = \prod_{\ell:y_s^\ell=0} prob^\ell \prod_{\ell:y_s^\ell=1}(1 - prob^\ell)$. The second stage variables $x^s \in \Re^q$ are used to model material flows in the integrated site (IS) for each scenario $s \in S$.

Remark

Set S contains all possible failure scenarios in the superstructure, but each flowsheet selection defines a subset of relevant scenarios. The scenarios in S can be aggregated – several scenarios can be projected into a single one – to derive any relevant subset for any possible flowsheet selection.

The following mixed-integer linear programming (MILP) problem is the deterministic equivalent of the stochastic optimal design problem. It also corresponds to a compact representation of the model presented in Appendix A. In the MILP model below, the variables x_s represents the vector of second-stage decision variables defined in the model in Appendix A. These variables are $flow_{j,j',n,s}$, $f_{j,n,s}^P$, $f_{j,n,s}^B$, $f_{j,n,s}^{IN}$, $f_{j,n,s}^{OUT}$, $ps_{m,s}$, $\delta_{j,n}^s$. The variable d is a vector of first-stage design variables that corresponds to the following variables in Appendix A: $\overline{inv}_{j,n}$, $\tilde{sd}_{j,n}$, $v_{j,n}$, pc_m, z_m.

$$Max \sum_{s \in S} p_s c_s^T x_s - pen^T sl \tag{1a}$$

s.t.

$$A_s^1 x_s + B_s^1 d \leq c^1 \quad \forall s \in S \tag{1b}$$

$$A_s^2 x_s \leq diag(B_s^2 d(e - y_s)^T) \quad \forall s \in S \tag{1c}$$

$$\sum_{s \in S} F_s x_s + G d - sl \leq h \tag{1d}$$

$$C_3 d \leq Capital \tag{1e}$$

$$\begin{bmatrix} B(d)_{s,s'} \\ x_s = x_{s'} \end{bmatrix} \vee \begin{bmatrix} \neg B(d)_{s,s'} \\ x_s \geq 0 \\ x_{s'} \geq 0 \end{bmatrix} \quad \forall s \in S, s' \in S, s' < s, (s, s') \in NA \tag{D1}$$

$$d \in D, x^s \in \Re^q, sl \in \Re_+, \quad \text{where} \quad D = \{d | d_i = 0, 1,$$

$$i = 1, i = 1, \ldots, p, d_i \in \Re_+^r, i = p + 1, \ldots, r\} \tag{1h}$$

$$B(d)_{s,s'} = \{True, False\} \quad \forall(s, s') \in S^2$$

The objective function (1a) represents the maximization of the average flow of product to external consumers. The coefficient pen^T penalizes the slack variables sl that are introduced in constraint (1d) for guaranteeing feasibility of the subproblems and accelerating the convergence of our implementation of Benders decomposition. The penalty term is large enough to enforce the slack variable to become equal to zero at the optimal solution. In the examples we solve in this paper, we use a value of 10 for the penalty term. Constraints (1b) and (1c) represent the mass balances in the superstructure. Constraint (1c) imposes reductions on the production rates of the plants in the integrated site, according to the active failure modes in state s; it will have terms of the form: $ps_{m,s} \leq pc_m[1 - (1 - y_\ell^s)(rc_\ell)]$ (refer to constraint (A11) in Appendix A). The notation $diag(X)$ corresponds to the diagonal elements of the matrix X; and e is a unitary vector. Constraint (1d) is a compact representation of the intermediate storage model we proposed in previous work (Terrazas-Moreno et al., 2010). Constraint (1e) corresponds to the ε-constraint for the bi-criterion optimization problem since it restricts the cost of a flowsheet with design variables d to be less than or equal to the maximum available investment, $Capital$. Disjunction (D1) establishes a relationship between the operating variables of scenario s and s'. The Boolean variable $B(d)_{s,s'}$ is true if scenarios s and s' are indistinguishable in the flowsheet defined by d. It is important to notice that (D1) is not defined for every possible pair (s',s), but only for those pairs that fulfill two conditions: (a) by symmetry of the constraint set, the first condition is $s' < s$; (b) that pair (s',s) should belong to set NA, which is made up of all pairs that differ in the value of exactly one element of the vector y^s. These properties are explained in detail by Goel and Grossmann (2006). We can represent (D1) using inequality constraints and a binary variable $\alpha_{s,s'}(d)$ in (1f) and (1g) (Raman & Grossmann, 1994), which yields the following MILP problem:

$$Max \sum_{s \in S} p_s c_s^T x_s - pen^T sl$$

s.t.

$$A_s^1 x_s + B_s^1 d \leq c^1 \quad \forall s \in S$$

$$A_s^2 x_s \leq diag(B_s^2 d(e - y_s)^T) \quad \forall s \in S$$

$$\sum_s F_s x_s + G d - sl \leq h \tag{1d'}$$

$$C_3 d \leq Capital$$

$$x_s \leq x_{s'} + M\alpha_{s,s'}(d) \quad \forall s \in S, \quad s' \in S, \quad s' < s, (s, s') \in NA \tag{1f}$$

$$x_s \geq x_{s'} - M\alpha_{s,s'}(d) \quad \forall s \in S, \quad s' \in S, \quad s' < s, (s, s') \in NA \tag{1g}$$

$$d \in D, x^s \in \Re_+^q, sl \in \Re_+ \quad \text{where} \quad D = \{d | d_i = 0, 1,$$

$$i = 1, \ldots, p, d_i \in \Re^r, i = p + 1, \ldots, r\} \tag{1h'}$$

$$\alpha_{s,s'}(d) = 0, 1 \quad \forall(s, s') \in S \times S$$

Inequalities (1f) and (1g) are non-anticipativity constraints, where the constant M is a large number that renders these inequalities to be redundant for $\alpha_{s,s'}(d) = 1$. The term $\alpha_{s,s'}(d)$ is a function of the integrated site flowsheet and is defined as follows:

$$\alpha_{s,s'}(d) = \begin{cases} 0 & \text{if scenarios } s \text{ and } s' \text{ are indistinguishable in} \\ & \text{the network topology defined by } d \\ 1 & \text{otherwise} \end{cases}$$

For instance, if design d does not include unit m from the superstructure, then states s',s that differ only on whether or not unit m has failed are considered indistinguishable. The explicit function for $\alpha_{s,s'}(d)$ is as follows:

$$\alpha_{s,s'}(d) = \sum_{m \in M} \sum_{\ell \in L_m} z_m \sigma_\ell^{s,s'} \quad \forall j \in J, \quad s \in S, \quad s' \in S, \quad s > s', (s, s') \in NA \tag{1i}$$

where $\sigma_\ell^{s,s'}$ is a problem *parameter* that can be derived from the vector y_ℓ^s. The vector y_ℓ^s is a fixed parameter defined in a previous section to be 1 if failure ℓ does not occur as part of state s, and 0 otherwise. Let the parameter $\sigma_\ell^{s,s'}$ be defined as below.

$$\sigma_\ell^{s,s'} = \max\{y_\ell^s - y_\ell^{s'}, (1 - y_\ell^s) - (1 - y_\ell^{s'})\} \quad \forall \ell \in L,$$

$$s \in S, \quad s' \in S, \quad s > s'(s, s') \in NA \tag{1j}$$

In Eq. (1j) $\sigma_\ell^{s,s'}$ is set to one if states s and s' are distinguishable with respect to failure ℓ; that is, if the failure occurs in one state and not in the other.

Remarks

1. The state space S usually has high dimensionality, so that the number of constraints defined by $\forall s \in S, s' \in S, s' < s, (s, s') \in NA$ can become computationally intractable.
2. For a fixed design $\hat{d}$, constraints (1f) and (1g) can be solved outside of the optimization problem (1). In this case, we can aggregate all indistinguishable scenarios a priori, and generate a *reduced* set S_B.
3. Problem (1) with fixed $\hat{d}$ and a reduced set S_B of scenarios corresponds to a linear programming (LP) problem. Since this LP problem excludes constraints (1f) and (1g), it results in a decrease of orders of magnitude in the number of constraints when compared against the full two-stage MILP stochastic problem.

6. Decomposition algorithm

6.1. Basic idea

The algorithm we propose in this section results in a significant reduction of the number of non-anticipativity (NA) constraints that are initially considered in problem (1). This algorithm can be used (as we do in our numerical examples) with other existing modeling techniques to reduce the number of NA constraints as in Goel and Grossmann (2006). NA constraints are required only because the optimal design is a degree of freedom. The failure scenarios are built considering all units in the *superstructure*, but not necessarily all of these units will be part of the optimal network topology. When two or more failure scenarios are different only with respect to a
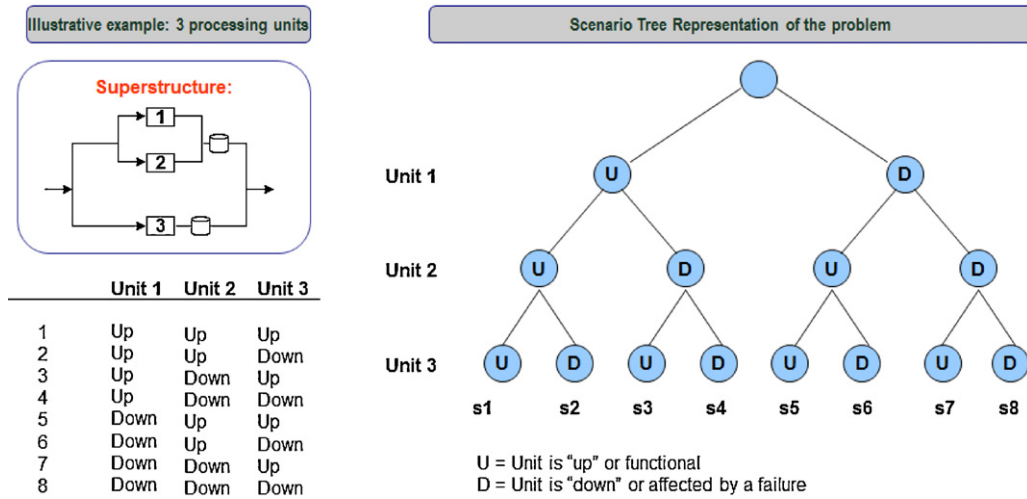
Fig. 3. Scenario tree representation of failure scenarios.

failure in units that are not part of the network, a non-anticipativity constraint has to be activated to make those scenarios identical.

Figs. 3 and 4 illustrate this point for a three unit example.

The basic idea of the algorithm is to iteratively solve for the design (flowsheet) in a problem with only a few scenarios, and then solve the rest of the problem in a reduced space where only scenarios relevant to the fixed flowsheet are considered. Since only failures relevant to installed units are considered in the second step, there is no need for NA constraints. We use the basic concept of Benders decomposition to obtain the flowsheet in the master problem and to solve the rest of the scenarios in the subproblem. Our contribution to the method is that the subproblem is solved in a reduced space where there is a limited number of scenarios and no (or only very few) non-anticipativity constraints.

Definitions

According to what we have defined so far in the paper, we have the following sets:

$L = \{\ell : \ell \text{ is a failure mode in the superstructure}\}$

$S = \{s : s \text{ represents a combination of failure modes in } L\}$

$y^s = \{y_\ell^s : y_\ell^s = 0 \text{ for active failure mode, 1 otherwise,}$
$\quad \ell = 1, \ldots, |L|\}$

We now include the following definitions

$\bar{L}_k \subseteq L$ is a subset of failures relevant to a network topology k

$S_B^k = \{s : s \text{ represent a combination of failure modes in } \bar{L}_k\}$

$S_M = \{s : \text{subset of failure modes for master problem}\}, \quad S_M \subseteq S$

$S_M^C := \text{Complement of } S_M$

Finally, we define a subset of the Cartesian product as:

$FN_k = \{(s', s) : y_\ell^{s'} = y_\ell^s, \quad \forall \ell \in \bar{L}_k\} \subseteq S \times S_B^k$
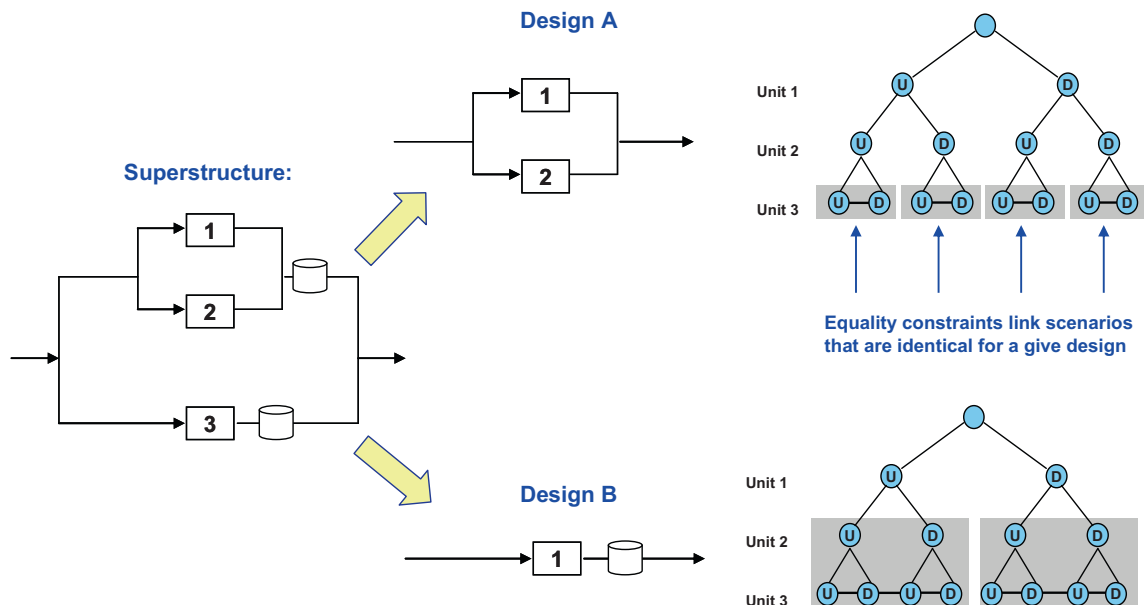

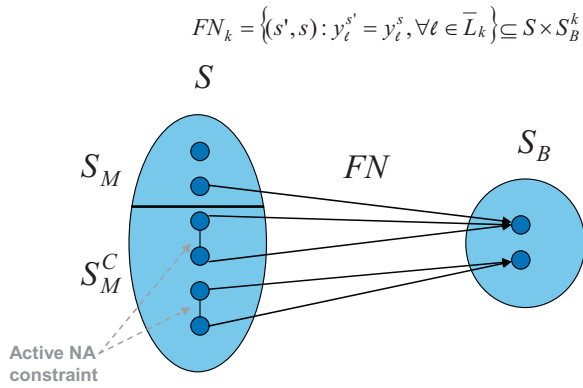**Fig. 4.** NA constraints are required since flowsheet structure is a degree of freedom.

$$FN_k = \left\{ (s',s) : y_\ell^{s'} = y_\ell^s, \forall \ell \in \bar{L}_k \right\} \subseteq S \times S_B^k$$

Fig. 5. Function FN_k.

Fig. 7. Reduced scenarios in S_B^k and their probabilities p_s^k for fixed unit 1.

The pair (s',s) represents all the duplets in the Cartesian product $(S \times S_B^k)$. For each of these duplets there is a corresponding pair $y^{s'}, y^s$. Recall that y^s is a vector with as many elements as failure modes $\ell \in L$, and that y_ℓ^s is a vector with as many elements as failure modes in $\ell \in L_k$, where $|L_k| \leq |L|$ since there are more failures in the units of the superstructure that in the units corresponding to a particular topology k, which is conformed of a subset of the units in the superstructure. If for failures relevant to topology k, that is $\ell \in L_k$ $y_\ell^{s'} = y_\ell^s$, scenarios s' and s are equivalent with respect to failures relevant to topology k. That is, looking only at the units in topology k, scenarios s' and s, have the same combination of active failure modes. If this condition is met, (s',s) are members of FN_k.

In fact, FN_k defines a function $FN_k : S \to S_B^k$ that maps between the sets S and S_B^k, since for every element $s' \in S$ there is only one corresponding element in $s \in S_B^k$. This relationship operates in the following way: taking any element in $s' \in S$ we look for the element $s \in S_B^k$ such that $(s', s) \in FN_k$.

The function $FN_k : S \to S_B^k$ can be used to project the different scenarios in S that are identical with respect to the failures in L_k onto one scenario in S_B^k. This use of FN_k is illustrated in Fig. 5.

7. Properties of the reduced set S_B^k

In the above section, S_M represents the failure modes considered in the master problem, while S_B^k was defined as a set of scenarios that include all the relevant failure mode combinations for a fixed network topology k. The main idea behind the decomposition approach presented in this paper is to solve the Benders subproblem in the reduced space of S_B^k. For instance, let the superstructure of an integrated site have two units. Each unit has one failure mode,
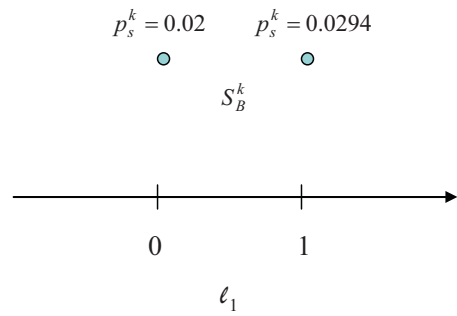
so $L = \{\ell_1, \ell_2\}$. Each vector $y_s = \{y_s^{\ell_1}, y_s^{\ell_2}\}$ describes a scenario $s \in S$ where $S = \{\{1,1\}, \{1,0\}, \{0,1\}, \{0,0\}\}$. Recall that a 0 in the vector y_s denotes the occurrence of a failure in scenario s. Assuming the probability of failure is small, the Benders algorithm is set up so that $S_M = \{\{1,1\}\}$ and its complement $S_M^C = \{\{1, 0\}, \{0, 1\}, \{0, 0\}\}$. If in a given iteration the flowsheet obtained from the solution of the master problem includes only one unit, we would define the reduced set $\bar{L}_k = \{\ell_1\}$, so that $S_B^k = \{\{1\}, \{0\}\}$. In this case scenario $\{1\}$ in S_B^k is the projection of $\{1,0\}$ from S onto S_B^k. Scenario $\{0\}$ is the projection of $\{0,1\}, \{0,0\}$.

An important property that we require of S_B^k is that the sum of the probabilities of the scenarios in the reduced set must be equal to the summation of the probabilities in the original set, i.e., $\sum_{s \in S_B^k} p_s^k = \sum_{s \in S_M^C} p_s$. We illustrate how to compute the probabilities p_s, $\forall s \in S_B^k$ in order to satisfy this condition using the example of a two-unit superstructure. There is one failure mode per unit in each of the two units, where $prob^{\ell 1} = 0.02$ and $prob^{\ell 2} = 0.03$. Fig. 6 shows the combinations of failure modes for scenarios in S and their corresponding probabilities. It also shows the partitioning of S into S_M and S_M^C.

Once more, assume that at a given iteration of the Benders algorithm only unit 1 is chosen from the superstructure, and we wish to solve the subproblem in the projected space S_B^k. There are two states in S_B^k: $\{1\}$ and $\{0\}$ corresponding to the functional and failed states of unit 1. The first of these, $\{1\}$, is the projection of $\{1,0\}$; the second, $\{0\}$, is the projection of $\{0,1\}, \{0,0\}$. The probabilities of the reduced states $\{1\}$ and $\{0\}$ are equal to the sum of the probabilities of the projected states $\{1,0\}$ and $\{0,1\}, \{0,0\}$. The probability of the state $\{1\}$ is equal to the probability of $\{1,0\}$, and the probability of $\{0\}$ is the sum of the probabilities of $\{0,1\}, \{0,0\}$. Fig. 7 shows the reduced scenarios and their corresponding probabilities for fixed unit 1. It can be verified that each of the probabilities in the scenarios in Fig. 7 correspond to the addition of probabilities of the
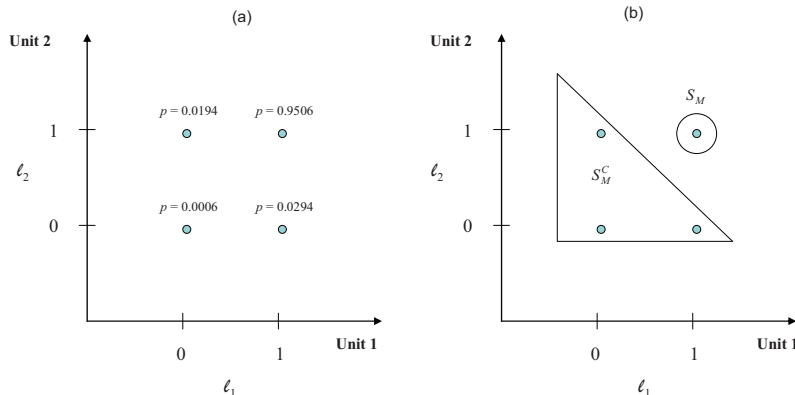

Fig. 6. Scenarios and their partitioning in illustrative example.

first and second column of Fig. 6(a), considering only the states in S_M^C. We label the probability in the reduced space as p_s^k. In general, $p_{s'}^k = \sum_{s \in A} p_s$, where $A = \{s : y_s^\ell = y_{s'}^\ell \quad \forall l \in \bar{L}, \quad s \in S_M^C\}$.

7.1. Proposed algorithm

Step 1

Define a maximum number of iterations $K_{\max}$ and the tolerance of the problem ε. Set the counter $K = 1$, and the initial value for the lower bound $LB = -\infty$.

Select S_M as the set of scenarios with largest probability p_s (i.e. $p_s \geq \gamma$, where γ is threshold value).

Step 2

Solve the master problem (M) as defined below:

$$Max \; Constant + \varepsilon\eta \tag{M1}$$

s.t.

$$\eta \leq \sum_{s \in S_M} p_s c_s^T x_s + \sum_{s \in \bar{S}_B^k} (u_{1s}^k(c^1 - B_s^{1^k}d) + u_{2s}^k diag(B_s^{2^k}d(e - y_s)^T))$$

$$+ v^k(h - Gd - \sum_{s \in S_M} F_s x_s) + \sum_{s \in S_B^k s':s' \in S_M,} \sum_{(s,s') \in FN_k \cap NA} w_{s,s'}^k x_{s'},$$

$$K \neq 1, \quad k = 1, \ldots, K - 1 \tag{M2}$$

$$\eta \geq LB + \varepsilon \tag{M3}$$

$$\eta \leq \sum_{s \in S_M} p_s c_s^T x_s + \sum_{s \notin S_M} p_s c_s^T x_s^{UB} \tag{M4}$$

$$A_s^1 x_s + B_s^1 d \leq c^1 \quad \forall s \in S_M \tag{M5}$$

$$A_s^2 x_s \leq diag(B_s^2 d(e - y_s)^T) \quad \forall s \in S_M \tag{M6}$$

$$\sum_{s \in S_M} F_s x_s + Gd \leq h - \sum_{s \in S_M} F_s x_s^{L0} \tag{M7}$$

$$C_3 d \leq Capital \tag{M8}$$

$$x_s \leq x_{s'} + M\alpha_{s,s'}(d) \quad \forall s \in S_M, \quad s' \in S_M, \quad s' < s, \quad (s, s') \in NA \tag{M9}$$

$$x_s \geq x_{s'} - M\alpha_{s,s'}(d) \quad \forall s \in S_M, \quad s' \in S_M, \quad s' < s, \quad (s, s') \in NA \tag{M10}$$

$$d \in D, x^s \in \Re_+^q, sl \in \Re_+ \quad where \quad D = \{d|d_i = 0, 1,$$

$$i = 1, \ldots, p, d_i \in \Re^r, i = p + 1, \ldots, r\} \tag{M11}$$

$$\alpha_{s,s'} = 0, 1 \quad \forall (s, s') \in S \times S$$

where u_{1s}^k, u_{2s}^k, v^k, and w_s^k are dual variables arising in the subproblems defined in Step 4.

The master problem (M) is the bottleneck of the decomposition algorithm. To speed up the convergence of (M), we have set it up as a feasibility problem instead of a rigorous optimization problem. The value of the constant term in the objective function (M1) is of the same order of magnitude as the optimal solution to the full space problem. This value is easy to calculate since it is possible to know the productivity if no failures were present in the integrated site. Then we solve (M) using a loose tolerance in the MILP solver. The tolerance that we refer to here is the gap between the upper and lower bounds *of the branch and bound method* used to solve the MILP. It is not to be confused with the tolerance defined for the Benders decomposition algorithm. For instance, if the tolerance of the Benders decomposition, which is ε in the nomenclature of this paper, is set to 2%, we can allow a gap of 5% for the branch and bound method. In (M1) the term *Constant* is much larger than the term $\varepsilon\eta$, so that (M) will converge once a feasible solution is found. Although, as we have just said "the term $\varepsilon\eta$ is comparatively small",

keeping it is important since it improves the quality of the feasible solution found. Eqs. (M2) and (M3) constrain the variable η to be less than the dominant Benders cut but greater than a valid lower bound plus the convergence tolerance ε. Constraint (M4) has been added to enforce a valid upper bound on the objective function of the master problem. Note that the cardinalities of the sets of constraints (M9) and (M10) are much smaller than the cardinalities of (1f) and (1g). The solution to this problem yields the optimal values of the decision variables of the master problem at iteration k: $\hat{d}^k$ and $\hat{x}_s^k \; \forall s \in S_M$.

Termination criterion: Constraints (M5)–(M11) can always be trivially satisfied by not installing any unit from the superstructure and setting to zero the internal flows within the integrated site. The only possibility for (M) to be infeasible is if η cannot satisfy constraints (M2)–(M4). This is the case only if the upper bounds for η set by constraints (M2) and (M4) are lower than the lower bound set by constraint (M3). Therefore, *the algorithm is terminated as soon as (M) is infeasible*, which in turn guarantees that the lower bound LB is within ε-tolerance of the optimal solution to the full space problem.

Step 3

Select the sets $\bar{L}_k$ and S_B^k as defined above. Use the function FN_k to compute the coefficients in the reduced space of S_B^k:

$$p_s^k = \sum_{s':s' \in S_M^C, \; (s',s) \in FN_k} p_{s'} \quad \forall s \in S_B^k \tag{}$$

The calculation of this coefficient has been explained in the previous section.*Step 4*

Solve the subproblem (B) of the Benders decomposition using the reduced scenario set S_B^k.

$$Max \sum_{s \in S_B^k} p_s^k c_s^T x_s^k + \sum_{s \in S_M} p_s c_s^T \hat{x}_s^k - pen^T sl \tag{B1}$$

s.t.

$$A_s^1 x_s \leq c^1 - B_s^1 \hat{d}^k \quad \forall s \in S_B^k \tag{B2}$$

$$A_s^2 x_s \leq diag(B_s^2 \hat{d}^k(e - y_s)^T) \quad \forall s \in S_B^k \tag{B3}$$

$$\sum_{s \in S_B^k} F_s x_s - sl \leq h - G\hat{d}^k - \sum_{s \in S_M} F_s \hat{x}_s^k \tag{B4}$$

$$x_s = \hat{x}_{s'}^k \quad \forall s \in S_B^k, \quad s' \in S_M, \quad (s, s') \in FN_k \cap NA \tag{B5}$$

$$x^s \in \Re_+^q, \quad sl \in \Re_q^+ \tag{B6}$$

The non-negative slack variables sl in (B4) ensure feasibility of the subproblem. A special note must be made regarding constraint (B5). The reduced set S_B^k is introduced in order to eliminate the need for non-anticipativity constraints (1f) and (1g) among the scenarios in S_M^C, but there are still indistinguishable pairs of scenarios (s,s') where s belongs to S_M^C and s' to S_M. The claim is that the set of constraints in (B5) is of significantly smaller cardinality than (1f) and (1g).

After solving subproblem (B), we construct the Benders cut using the objective function of the dual of (B):

$$\eta \leq \sum_{s \in S_M} p_s c_s^T x_s + \sum_{s \in \bar{S}_B^k} (u_{1s}^k(e - B_s^{1^k}d) + u_{2s}^k diag(B_s^{2^k}dy_s^T))$$

$$+ v^k \left(h - Gd - \sum_{s \in S_M} F_s x_s\right) + \sum_{s \in S_B^k s':s' \in S_M,} \sum_{(s,s') \in FN_k \cap NA} w_{s,s'}^k x_{s'}$$

where u_{1s}^k, u_{2s}^k, v^k, and w^k are the optimal dual multipliers of constraints (B2)–(B5) at iteration k.
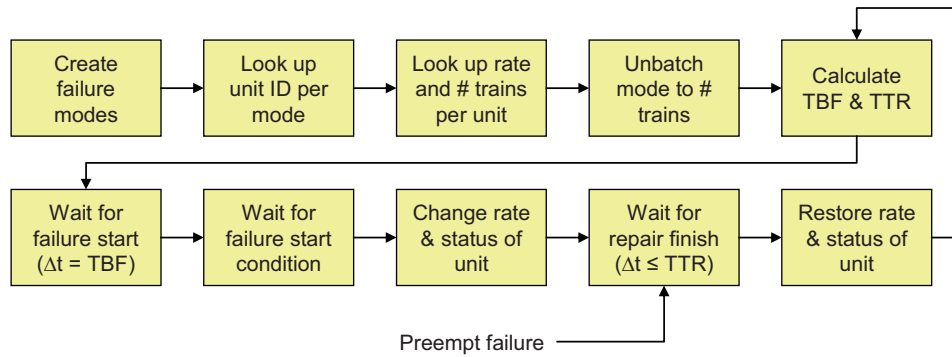
Fig. 8. Simulation logic for unplanned downtimes.

The value of the lower bound (*LB*) is updated if the optimal value of (B) is greater than the incumbent *LB*.

If $K = K_{\max}$, the algorithm stops. Otherwise, set $K = K + 1$, and go to *Step 2*.

8. Discrete rate simulation model

The simulation begins with the generation of discrete items, each representing a unique failure mode. Each plant in the integrated site has one or more failure modes. For each failure mode, we look up which plant is impacted by the failure. We then look up the number of parallel units for that plant and the production rate per unit. These parameters are assigned to the failure mode as attributes so the look-up is necessary only once at the beginning. Each item is unbatched to N items, where N = number of parallel units. The underlying assumption is that if multiple parallel units exist in a plant, these units fail independently of each other. For each failure mode, its time between failure (TBF) and time to repair (TTR) are calculated from distributions developed from available data. All failure modes wait for failure start, for the duration corresponding to their respective TBF. Then they wait in a queue, and each is released only when the affected unit/train is running (no ongoing failure or turnaround). When a failure mode is released from this queue, the rate and status of the affected unit is updated. All failure modes wait for repair to finish, for the duration corresponding to their respective TTR or less (only if the repair is pre-empted by a simulation logic that synchronizes certain maintenance activities). After the repair is finished for a given failure mode, the rate and status of the affected unit/train are restored. The TBF and TTR are recalculated, and the process is repeated. Fig. 8 summarizes the simulation logic for modeling unplanned downtimes resulting from failures.

A failure mode can result in the rate of zero for a complete shutdown or between zero and maximum unit capacity for rate loss. If there is an ongoing failure at a unit such that the current rate is zero, the above mentioned queue prevents any further failure modes for that unit to be activated, the assumption being that a down plant cannot fail any more. If there is an ongoing failure at a unit such that the current rate is not zero, then additional failure is possible. During the simulation, multiple failure modes can be in progress at the same time. The lowest rate of all failure modes in progress is chosen as the rate of a particular unit. The existence of more than one parallel unit per plant is accounted for in the updating of this rate input, depending on the status of all units. For example, if one unit is up and the other is down, then the rate input would be half the plant capacity.
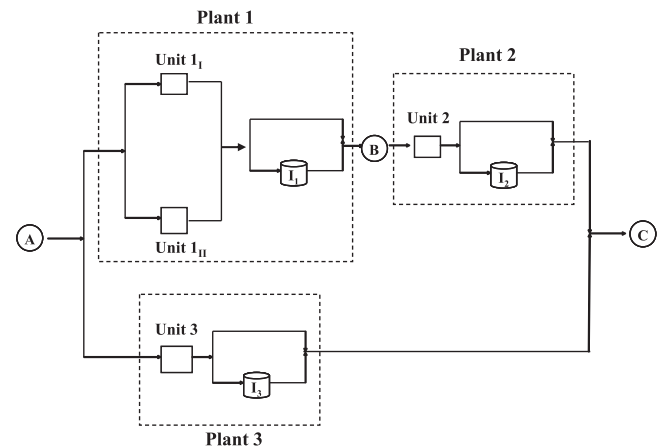

Fig. 9. Integrated Site for the production of C from A.

## 9. Application to the design of integrated sites (IS)

9.1. Illustrative example

We use the small example shown in Fig. 9 that was presented in our previous work (Terrazas-Moreno et al., 2010) to illustrate the proposed algorithm. The model and process data required to solve this example are available in Appendix A. We solve the full space version of the problem and then decompose it using the proposed algorithm for different values of capital investment in order to obtain a set of Pareto-optimal solutions. All results were obtained using the MILP solver CPLEX version 12.1, running on GAMS 23.3, with a 2.8 GHz Intel Pentium 4 processor and 2.5 GB RAM.

The Pareto-optimal solutions are shown in Fig. 10. Two specific network structures are shown for the Pareto-optimal points A and B in Fig. 11.

The network in Fig. 11(a) is designed to operate relying on process 3 and a large storage tank. In fact, 2.25 tons is the upper bound we set for the volume of the storage tank after plant 3. The network configuration that corresponds to point B in Fig. 10 belongs to a section of the optimal Pareto set where only marginal improvements in average production rate are achieved at the expense of large additional investment. Thus, the large spare capacities, the redundant units, and the relatively large storage tank in Fig. 11(b).

Tables 1 and 2 contain the problem sizes and the results that correspond to the full space model and to the decomposition strategy. Fig. 12 shows the convergence of the decomposition algorithm in 14 iterations for one value of capital investment (*Capital* = $60 MM). The lower bound before iteration 4 is a large negative number as a result of the slack variable in the objective function and Eq. (B4) having a large value. Without the slack variable the subproblem
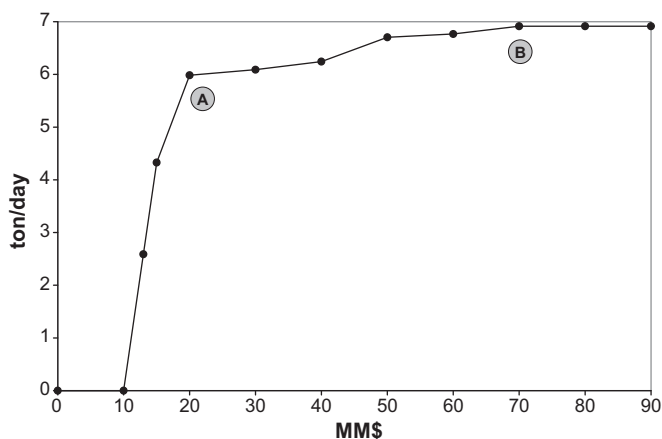
Fig. 10. Pareto-set of optimal solutions for Example 1. (a) Configuration A. (b) Configuration B.

Table 1
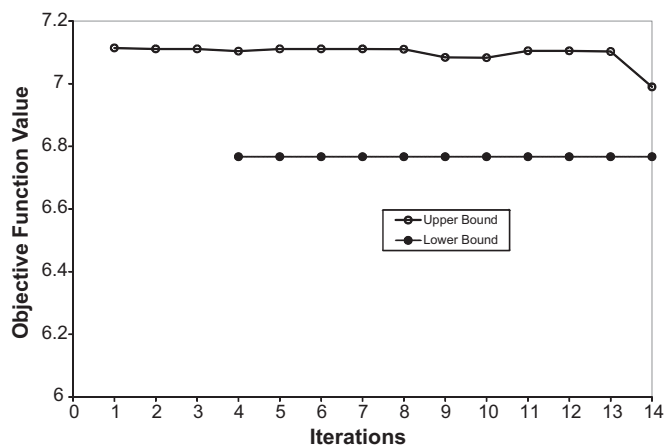Statistics of illustrative example.

	Discrete variables	Cont. variables	Constraints
Full space	4	1557	1670
Master problem[a]	4	502	370
Subproblem[b]	0	265	177

[a] Master problem at iteration 1.
[b] Subproblem at iteration 1 for 13 MM USD of capital investment.

Table 2
Results of illustrative example.

| Capital investment (MM USD)	Optimal solution (ton/day)	
	Full space	Sub problem (2% tolerance)
13	2.61	2.59
15	4.35	4.33
20	6.10	5.99
30	6.10	6.09
40	6.25	6.25
50	6.76	6.71
60	6.77	6.77
70	6.92	6.92
80	6.92	6.92
90	6.92	6.92

Fig. 12. Iterations of master problem and sub problem in the illustrative example for $60 MM of investment.

would have been infeasible. It is interesting to notice that, even for this small example made up of 4 processing units, the number of constraints in the full space model is significantly larger than the sum of the constraints in the master problem and subproblem (see Table 1). This is a consequence of the reduction in the number of non-anticipativity constraints in (1f) and (1g), and in the number of scenarios in the subproblem. Table 2 shows the results obtained by the full space and Benders decomposition methods for different values of capacity investments. The example is so small that each Pareto-optimal solution can be obtained in a fraction of a second of CPU time either using the decomposition algorithm or directly solving the full space model.

9.1.1. Sensitivity analysis

The Pareto-optimal solution that corresponds to point A in Fig. 10 represents an inflection point in the Pareto-optimal front. Due to the fact that further marginal increments in average production rate require large sums of extra capital investment, an industrial design team would be interested in understanding the factors that limit the performance of this design. For example, it could be more cost effective trying to improve the reliability

(a) Configuration A
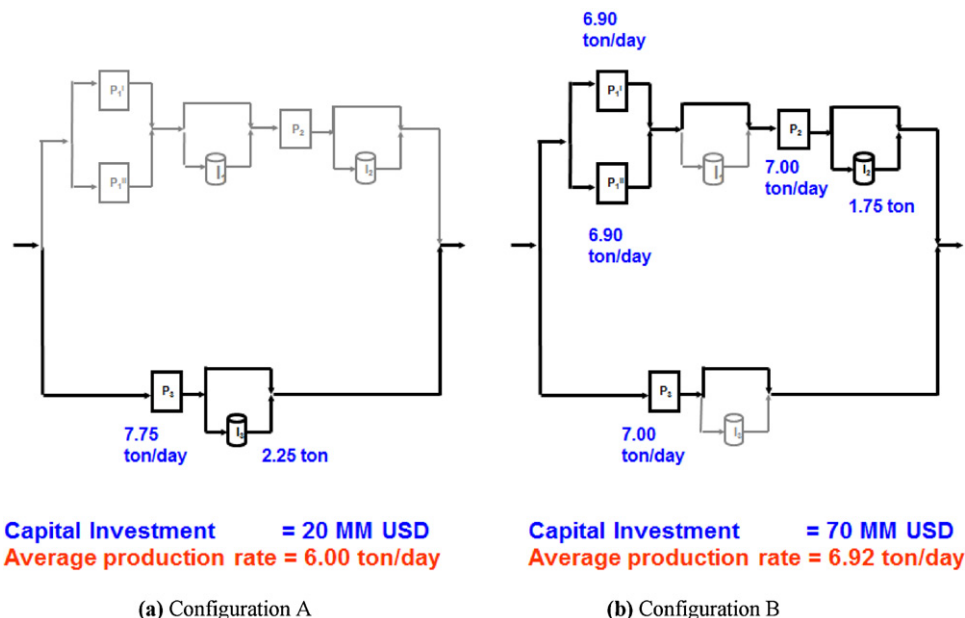
(b) Configuration B

Fig. 11. Network configurations for two Pareto-optimal solutions.

characteristics of key components in the design, rather than adding parallel production trains or increasing the capacity of existing ones. It could also be the case that small changes in the volume of storage tanks or capacity of production units can yield significant improvements in average production rate. This information can be very useful with the optimization–simulation approach proposed in this paper. These teams could use it to search efficiently the design space around one or more Pareto-optimal solutions with discrete-rate simulation. The sensitivity analysis described below has the objective of providing this type of information to such a design team, so that it can be used as a guideline to fine-tune the design in a cost-effective way.

The first step to carry out the sensitivity analysis is to fix the selection of units in the superstructure to that in the Pareto-optimal point of interest. We then construct the set of scenarios relevant to this selection in the same manner as when setting up the subproblem in the decomposition algorithm. Using this collection of scenarios as set *S*, and *with a fixed selection of units* (fixed z_m), we solve the full space problem described in Appendix A (Eqs. (A1)–(A22)) with the following two additional constraints.

$$pc_m^* \leq pc_m \leq pc_m^* \quad \forall m \in M \tag{S1}$$

$$v_{j,n}^* \leq v_{j,n} \leq v_{j,n}^* \quad \forall j \in J, \quad n \in N \tag{S2}$$

where pc_m^* and $v_{j,n}^*$ are the capacity of unit m and the volume of the tank for product n after plant j in the Pareto-optimal design being analyzed. Any solver for LP problems, such as CPLEX, provides the reduced costs at the optimal solution. The reduced costs, which are the dual variables at the active bounds, correspond to the derivatives of the objective function with respect to perturbations on the right-hand sides of each of the constraints in the optimization formulation (Chvatal, 1983). By reading reduced costs of constraints (S1) and (S2), we obtain the values of the following derivatives:

$$\frac{\partial(\text{Average Production Rate})}{\partial(pc_m^*)} \quad \forall m \in M \tag{S3}$$

and

$$\frac{\partial(\text{Average Production Rate})}{\partial(v_{j,n}^*)} \quad \forall j \in J, \quad n \in N \tag{S4}$$

These derivatives show the sensitivity of the optimal solution for marginal increments in the design variables. In this way, the relative magnitudes of the sensitivities can be used as guidelines for fine-tuning the design using simulation tools.

Next, we are interested in finding the derivatives of the objective function with respect to the probabilities of each failure mode, that is,

$$\frac{\partial(\text{Average Production Rate})}{\partial(p_\ell)} \quad \forall \ell \in L \tag{S5}$$

This information can be used to determine key failure modes and look for ways to improve the reliability characteristics of the corresponding components in the design. The probabilities of being in an operational state with respect to failure ℓ, p_ℓ, do not appear explicitly in the MILP model defined by Eqs. (A1)–(A22). However, the probability of each failures scenario (discrete state), $prob^s$, which is a function of the probabilities of independent failure modes, does appear in constraints (A1), (A12) and (A13). Note that we assume that $vrt^s fr^s = prob^s$ in constraint (A13), which is true when failures follow exponential distributions but an approximation in any other case. Knowing this, we carry out the following computations where *APR* stands for average production rate:

$$\frac{\partial(APR)}{\partial(p_\ell)} = \sum_{s \in S} \frac{\partial(APR)}{\partial(prob^s)} \frac{\partial(prob^s)}{\partial(p_\ell)}, \tag{S6}$$

Table 3
Sensitivity analysis of design corresponding to \$20 MM.[a]

$\partial(APR)/\partial(v_{3,c}^*)$	~0
$\partial(APR)/\partial(pc_3^*)$	~0
$\partial(APR)/\partial(p_3)$	6.98

[a] This design only includes unit 3 from the superstructure.

where

$$\frac{\partial(APR)}{\partial(prob^s)} = \sum_{j \in J} \sum_{n \in N} \left[\frac{\partial(APR)}{\partial(A12_{j,n})} \frac{\partial(A12_{j,n})}{\partial(prob^s)} + \frac{\partial(APR)}{\partial(A13_{j,n})} \frac{\partial(A13_{j,n})}{\partial(prob^s)} \right]$$
$$+ \frac{\partial(A1)}{\partial(prob^s)} \tag{S7}$$

and

$$\frac{\partial(prob^s)}{\partial(p_\ell)} = \prod_{\ell': \{y_{\ell'}^s = 1\}, \ \ell' \neq \ell} p_{\ell'} \prod_{\ell': \{y_{\ell'}^s = 0\}} (1 - p_{\ell'}) \quad \text{if} \ y_\ell^s = 1 \tag{S8}$$

or

$$\frac{\partial(prob^s)}{\partial(p_\ell)} = \prod_{\ell': \{y_{\ell'}^s = 1\}} p_{\ell'} \prod_{\ell': \{y_{\ell'}^s = 0\}, \ \ell' \neq \ell} (1 - p_{\ell'}) \quad \text{if} \ y_\ell^s = 0 \tag{S9}$$

In Eq. (S7), $A12$ and $A13$ stand for the right-hand sides of Eqs. (A12) and (A13) in Appendix A, and $A1$ corresponds to the objective function. The values $(\partial(APR)/(\partial(A12_{j,n}))$ and $(\partial(APR)/(\partial(A13_{j,n}))$ are the dual variables of the corresponding constraints, and they are obtained from the output of an LP solver. The partial derivatives $(\partial(A12_{j,n})/(\partial(prob^s))$, $(\partial(A13_{j,n})/(\partial(prob^s))$ and $(\partial(A1)/(\partial(prob^s))$ are obtained analytically from constraints (A12) and (A13) and the objective function (A1). Finally, Eqs. (S8) and (S9) are the result of differentiating the function $prob^s = \prod_{\ell: \{y_\ell^s = 1\}} p_{\ell'} \prod_{\ell: \{y_\ell^s = 0\}} (1 - p_{\ell'})$ that determines the probability of each failure state as a combination of the probabilities of the independent failure modes.

Table 3 contains the results of the sensitivity analysis of the design labeled as A in Fig. 10. *Note*: Design A only involves plant 3.

The results in Table 3 indicate that marginal changes in the design variables around their optimal values have little effect on the objective function. In fact, this is consistent with the small slope after point A in Fig. 10. In contrast, increasing the probability of being in an operational state of plant 3 has a large potential impact on the average production rate. To give the value of the derivative $(\partial(APR)/(\partial(p_3))$ a more tangible meaning, we can calculate the effect of a 5% increase in the ratio of MTBF over MTTR for plant 3. After some algebraic manipulation of the expression $p_3 = (MTBF_3/(MTBF_3 + MTTR_3))$ we get:

$$\Delta APR \ (5\% \ \textit{increase in MTTF/MTTR})$$
$$= \left[\frac{1.05}{1 + 0.05 p_\ell} - 1 \right] \frac{\partial(APR)}{\partial(p_3)} = 0.044$$

Thus, an increase of 5% on the mentioned availability could result in the design corresponding to \$20 MM going from 6.095 mass/h to 6.139 mass/h. This option would be preferred over increasing the capital investment from \$20 MM to over \$30 MM.

10. Large-scale example

This section describes the computational results of the proposed algorithm for solving an industrial-sized process network. Fig. 13 shows the 9 processing plants that constitute this network. Each of the plants in this integrated site represents the production of a chemical that can be shipped to external markets or used as raw material in a downstream process. In the latter case, the integrated
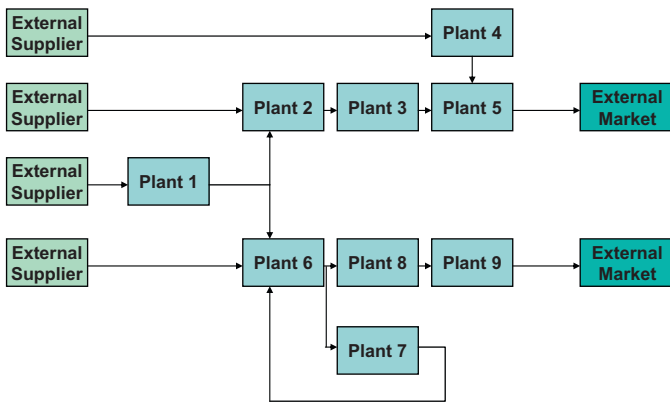
Fig. 13. Industrial integrated site in large-scale example.

site could be part of a very large chemical production site made up of several "smaller" integrated sites. In this example we use a concave cost function (the common six-tenths rule (Biegler, Grossmann, & Westerberg, 1997) for capturing the effect of different processing capacities on the capital investment required by processing units. We use a piece-wise linear approximation in order to keep the linearity of the model.

Each of the 9 plants that constitute the integrated site is modeled as a continuous process with one or several inputs and one output. We postulate a superstructure with two parallel production units in each plant and a storage tank after each plant except after plants 5 and 9, where the corresponding product cannot be stored. Each unit in the superstructure is subjected to different partial (decreased capacity) and total random failure modes. The distribution of the failure times is exponential, while the repair times follow a normal distribution. The total number of failures in the superstructure is 198.

Solving this industrial case study with the formulation in Appendix A using the decomposition framework presents two main challenges. One of them is the dimensionality: the resulting space for the units *in the superstructure* consists of 2^{198} discrete states (scenarios). Even with the decomposition algorithm, we are not able to handle this problem size. The second challenge is that the normal distribution of the repair times causes the variance of the residence time in each state to be different than that calculated assuming an exponential distribution. As indicated in our previous work (Terrazas-Moreno et al., 2010), the asymptotic values of the probabilities of the states and the mean residence time can be obtained analytically for any type of distribution as long as we have the mean failure and repair times. Unfortunately, this is not the case for the variance of the residence times. We overcome these two challenges by building a discrete event simulation model that we can use to collect a sample of scenarios. This technique is basically a Monte Carlo sampling procedure.

10.1. Sampling of scenarios and validation of designs using the discrete event simulation model

One use of the simulation model is to build a sample of scenarios for our problem. As the simulation runs, it generates a list of items (failures) and their attributes (time between failures and time to repair). It also registers a generation time for each of the items (failures). Scenarios (states) in the problem at hand are described by a set of failures occurring simultaneously in the integrated site. From the list of items mentioned above, we can construct a list of scenarios or states and their durations. Appendix B contains the details of the methodology to construct the list of scenarios from the list of failure items created in discrete event simulation. Since

the scenarios are generated randomly, we can consider the method as a Monte Carlo sampling. The number of scenarios is finite, and a few of them are highly more likely than the rest, so we can expect many repetitions of the same scenario as we sample. The sampling method allows us to calculate probabilities as well as mean and variance of residence time in each scenario. The size of the sample can be determined by the desired level of accuracy in the solution to our two-stage stochastic programming problem. The variance estimator of the solution to a stochastic programming problem is given by Shapiro and Homem-de-Mello (2000):

$$S(n) = \sqrt{\frac{\sum_{s=1}^{n}(E[obj] - obj_s)^2}{n - 1}} \qquad (2)$$

where *n* is the number of scenarios and *obj* is the objective value. Let $z_{\alpha/2}$ be the normal standard deviate (corresponding to a normal distribution with zero-mean and unitary standard deviation) for a confidence interval of $1 - \alpha$. In other words, $\Pr(z \le z_{\alpha/2}) = 1 - \alpha/2$, for $z \sim N(0, 1)$. For a confidence interval of 95%, $z_{\alpha/2} = 1.96$. To obtain the exact number of scenarios we follow the steps in You, Wassick, and Grossmann (2009):

(1) Obtain a sample for one year of operation to approximate $S(n)$.
(2) Define $z_{\alpha/2}$ and a desired interval H so that the solution to the stochastic programming problem is within an interval of $[obj_{sample} - H, obj_{sample} + H]$ with confidence of $1 - \alpha$.
(3) Determine the size of the sample as:

$$N = \left[\frac{z_{\alpha/2}S(n)}{H}\right]^2 \qquad (3)$$

For $H = 2.5$ mass/h (recall that the objective function is the expected throughput of the integrated site) and a confidence level of 95% we need a sample size of 2878 scenarios. We found that by simulating 10 years of plant operation we can construct this sample. Intuitively, one would expect that excluding scenarios that are not relevant after 10 years of plant has a minimal effect on the optimal design of the integrated site.

10.2. Numerical result

We simulated 10 year of operation of the integrated site and obtained a sample of 2973 scenarios. The statistics corresponding to the 10 most frequently encountered scenarios are shown in Table 4.

Table 4 contains some valuable information. For instance, from the first row we know that the integrated site will operate without any active failure mode around 23% of the time and that a failure will occur approximately every 28 h. Another important piece of information is that 4 out of the remaining 9 most common failure scenarios are due to some failure mode in plant 8.

After obtaining the sample of failure scenarios, we attempted to solve the problem in full space of 2973 scenarios (without decomposition) and using the proposed decomposition algorithm in a workstation with a 2.40 GHz Intel Core2 Quad CPU processor and 8 GB RAM. The results were obtained with the MILP solver CPLEX 12.1.0 running in GAMS 23.3.

The set of Pareto-optimal solutions in Fig. 14 was obtained first by solving the full space model which required 31 h. In contrast, using the decomposition algorithm with a 2% convergence tolerance, the Pareto curve can be obtained in 9 h. Table 5 shows the comparison in terms of CPU times for both approaches. An important note is that the solution times in the table do not include the time GAMS required to generate the model, which was nearly as long as the solution time for the full space model and not more than a few minutes with the decomposition algorithm. Table 6 shows the problem sizes of the decomposed and full space formulations.

Table 4
Statistics of most probable failure scenarios.

Scenario	Probability	Mean residence time (h)	Frequency of visits ($1/(h \times 10^3)$)	Plants in failure mode
1	0.231	28	8.0	None
2	0.022	23	0.9	Plant 8
3	0.020	20	1.0	Plant 8
4	0.020	25	0.8	Plant 8
5	0.019	27	0.7	Plant 2
6	0.013	18	0.7	Plant 2
7	0.012	21	0.6	Plant 5
8	0.010	25	0.4	Pant 9
9	0.009	21	0.4	Plant 8
10	0.008	28	0.3	Plant 9

Table 5
Performances of decomposition algorithm and full space solution using sample of scenarios.

Capital investment (k USD)	Full space		Decomposition algorithm		
	Solution (mass/h)	CPU time (s)	Upper bound (mass/h)	Lower bound (mass/h)	CPU time (s)
$100,000	5.20	1,052	5.26	5.16	1,324
$150,000	46.83	4,980	47.28	46.35	6,431
$200,000	62.12	15,376	62.52	61.29	3,817
$220,000	68.30	23,271	69.27	67.91	3,813
$240,000	70.77	12,645	71.66	70.26	4,277
$260,000	72.31	21,611	72.97	71.54	3,308
$280,000	73.27	18,082	72.95	71.52	5,109
$300,000	73.97	15,245	74.92	73.46	3,080
	Total	31.18 h		Total	8.66 h

Finally, Table 7 shows the details of three of the Pareto optimal network configurations in Fig. 14.

10.3. Sensitivity analysis

We carry out a sensitivity analysis around the design corresponding to $220 MM, following the same procedure outlined in Example 1. The analysis predicts that there is no marginal benefit in increasing the size of the storage tanks and that the design fine-tuning should focus on small increases on the capacity of plants 1

Table 6
Problem size corresponding to industrial case study.

	Discrete variables	Cont. variables	Constraints
Full space	144	4,753,154	1,864,942
Master problem[a]	144	599,136	203,574
Subproblem	0	4,183,206	877,194

[a] Master problem at iteration 1.

Table 7
Optimal designs for different capital investments using a random sample of scenarios.

Plant	200 MM USD Average Production Rate 61.29 mass/h			220 MM USD Average Production Rate 67.91 mass/h		
	Production units	Capacity per unit (ton/h)	Storage size (ton)	Production units	Capacity per unit (ton/h)	Storage size (ton)
1	1	74.8	–	1	88.2	–
2	1	77.1	–	1	87.8	–
3	1	58.7	–	1	64.4	3750
4	1	37.5	–	1	42.8	–
5	1	121.7	–	1	126.4	–
6	1	105.23	–	1	154.1	–
7	1	86.0	–	1	117.6	446
8	1	28.4	–	1	42.4	9
9	1	44.1	–	1	67.6	–

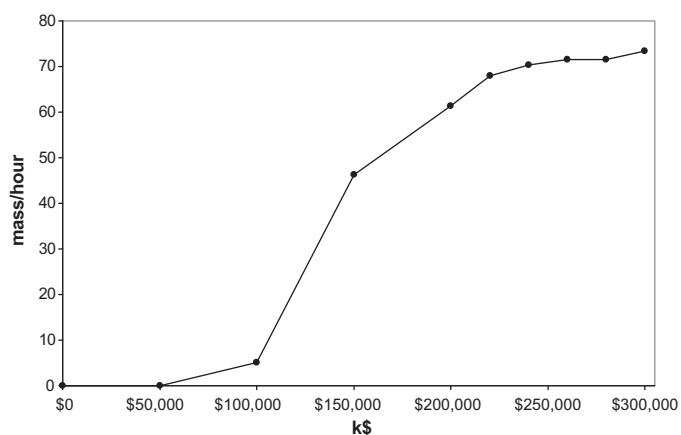| Plant	260 MM USD Average Production Rate 71.62 mass/h		
	Production units	Capacity per unit (ton/h)	Storage size (ton)
1	2	73.9	–
2	1	96.9	–
3	1	76.3	3750.0
4	1	42.8	3210.0
5	1	135.91	–
6	1	137.7	–
7	1	111.0	7770.0
8	1	42.4	93
9	2	52.0	–

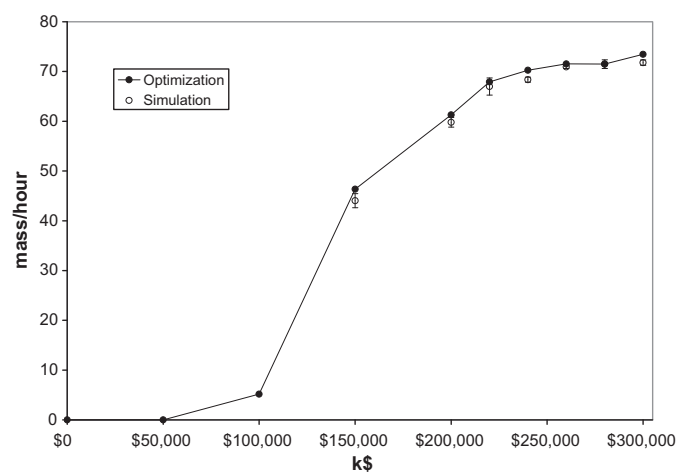**Fig. 14.** Pareto-optimal solutions for Example 2 using a random sample of 2973 scenarios.

Fig. 15. Comparison of simulation results vs. optimization results for some Pareto-optimal designs.

and 9. The sensitivity analysis with respect to probabilities in failure modes revealed that improving by 5% the ratio of MTBF over MTTR of one of the failure modes in either plant 2 or plant 8 could increase the average production rate from 68 ton/h to close to 70 ton/h. This is consistent with the data in Table 4 where it can be seen that the most frequent failure modes occur in plant 8 and plant 2.

10.4. Simulation of the Pareto-optimal designs

We use the same simulation model previously used for obtaining the sample of failure modes to evaluate the performance of the Pareto-optimal designs from the MILP optimization model. As we have indicated before, the simulation model is able to reproduce the real system to a greater extent than the optimization model, given the simplifications that have to be assumed to develop the MILP model. Therefore, simulating the design obtained in the optimization step allows us to observe the average production rate (objective function of the optimization model) of the integrated site under more realistic assumptions than those incorporated in the constraints of the MILP model. Since we perform three simulations per configuration, we report average and standard deviation for each metric. Fig. 15 shows the Pareto-optimal curve obtained through optimization and the corresponding performance fixing the network configuration and capacities in simulation runs. The error bars above and below the points that correspond to simulation are set to one standard deviation obtained when running

each point three times. As can be seen, the agreement between the optimization and simulation models is very good.

11. Conclusions

In this paper, we have addressed the problem of integrated site design under uncertainty as a two-stage MILP stochastic programming with endogenous uncertainties. In order to overcome the exponential growth in the number of scenarios required for modeling uncertainty, we proposed a decomposition algorithm based on Benders decomposition. Our main contribution is to exploit the problem structure in a way that allows the solution of Benders subproblems in a reduced space. The result is that the number of scenarios required is much smaller than with the full space problem. One of the main advantages of the proposed method is that the decomposed model requires significantly fewer non-anticipativity constraints.

The solution approach was tested in two case studies, one of them being an industrial processing network in which Monte Carlo Sampling was used to reduce the number of states. In the second case study, the decomposition algorithm reduces the solution time for the complete Pareto-set from 31 to 9 h. The integration with discrete rate simulation for validating/refining results increases the likelihood that the algorithmic method presented here will be accepted as a computational tool in an industrial setting.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.compchemeng.2011.10.005.

References

Acevedo, J. & Pistikopoulos, E. N. (1998). Stochastic optimization based algorithms for process synthesis under uncertainty. *Computers and Chemical Engineering*, *22*, 647–671.

BASF. (2010). http://www.basf.com/group/corporate/en/investor-relations/basfbrief/verbund/ludwigshafen-site Accessed on May 6th, 2010.

Benders, J. F. (1962). Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematik*, *4*, 238–252.

Biegler, L. T., Grossmann, I. E. & Westerberg, A. W. (1997). *Systematic methods of chemical process design*. New Jersey: Prentice Hall.

Billinton, R. & Allan, R. N. (1992). *Reliability evaluation of engineering systems*. New York: Plenum Press.

Chvatal, C. (1983). *Linear programming*. New York: W.H. Freeman and Company.

Davies, K. M., & Swartz, C. L. E. (2008). MILP formulations for optimal steady-state buffer level and flexible maintenance scheduling. MASc Thesis. McMaster University.

Dow. (2007). http://news.dow.com/dow_news/corporate/2007/20070712a.htm Accessed on May 6th 2010.

Ehrgott, M. (2005). *Multicriteria optimization*. Heidelberg: Springer Berlin.

Geoffrion, A. M. (1972). Generalized Benders decomposition. *JOTA*, *10*, 237–260.

Goel, V. & Grossmann, I. E. (2006). A class of stochastic programs with decision dependent uncertainty. *Mathematical Programming (Ser. B)*, *108*, 355–394.

Heyman, D. P. & Sobel, M. J. (1982). *Stochastic models in operations research* New York: McGraw-Hill.

Ierapetritou, M. G., Acevedo, J. & Pistikopoulos, E. N. (1996). An optimization approach for process engineering problems under uncertainty. *Computers and Chemical Engineering*, *20*, 703–709.

ILOG. (2011). http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/ Accessed on January 15th 2011.

Imagine That Inc. (2010). http://www.extendsim.com/index.html Accessed on March 23, 2011.

Jonsbraten, T. W., Wets, R. J. B. & Woodruff, D. L. (1998). A class of stochastic programs with decision dependent random elements. *Annals of Operations Research*, *82*, 83–106.

Liu, C., Fan, Y. & Ordonez, F. (2009). A two-stage stochastic programming model for transportation network protection. *Computers and Operations Research*, *36*(5), 1582–1590.

Miller, S., Owens, J. & Deans, D. (2006). From requirements development to verification – Designing reliability into a large scale chemical manufacturing system. In *Reliability and maintainability symposium, 2006. RAM'06 annual* (pp. 641–648).

Pistikopoulos, E. N. (1995). Uncertainty in process design and operations. *Computers and Chemical Engineering*, *19*, S553–S563.

Pistikopoulos, E. N., Thomaidis, T. V., Melin, A. & Ierapetritou, M. G. (1996). Flexibility, reliability and maintenance considerations in batch plant design under uncertainty. *Computers and Chemical Engineering*, *20*, S1209–S1214.

Pistikopoulos, E. N., Vassiliados, C. G., Arvela, J. & Papageorgiou, L. G. (2001). Interaction of maintenance and production planning for multipurpose process plants – A system effectiveness approach. *Industrial and Engineering Chemistry Research*, *40*, 3195–3207.

Raman, R. & Grossmann, I. E. (1994). Modeling and computational techniques for logic based integer programming. *Computers and Chemical Engineering*, *18*, 563–578.vadjust

Santoso, T., Ahmed, S., Goetschalckx, M. & Shapiro, A. (2005). A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research*, *167*, 96–115.

Shapiro, A. & Homem-de-Mello, T. (2000). On rate of convergence of Monte Carlo approximations of stochastic programs. *SIAM Journal on Optimization*, *11*, 70–86.

Straub, D. A. & Grossmann, I. E. (1990). Integrated stochastic metric of flexibility for systems with discrete state and continuous parameter uncertainties. *Computers and Chemical Engineering*, *14*(9), 967–985.

Straub, D. A. & Grossmann, I. E. (1993). Design optimization of stochastic flexibility. *Computers and Chemical Engineering*, *17*(4), 339–354.

Terrazas-Moreno, S., Grossmann, I. E., Wassick, J. M. & Bury, S. J. (2010). Optimal design of reliable integrated chemical production sites. *Computers and Chemical Engineering*, *34*(12), 1919–1936.

Van Slyke, R. & Wets, R. J. B. (1969). L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, *17*, 638–663.

Wassick, J. M. (2009). Enterprise-wide optimization in an integrated chemical complex. *Computers and Chemical Engineering*, *33*(12), 1950–1963.

You, F., Wassick, J. M. & Grossmann, I. E. (2009). Risk Management for Global Supply Chain Planning under Uncertainty: Models and Algorithms. *AIChE Journal*, *55*, 931–946.