

Combining SAN and P-Graphs for the Analysis and Optimization of Industrial Processes

Riccardo Bernini*, Andrea Bondavalli[†], Paolo Lollini[†], Leonardo Montecchi[†]

*Blue Reply s.r.l. – Milano, Italy

r.bernini@reply.it

[†]University of Firenze, Dipartimento di Matematica e Informatica – Firenze, Italy

{bondavalli, lollini, lmontecchi}@unifi.it

Abstract—Several formalisms and techniques have been introduced in the literature for the purpose of modeling and evaluation of complex systems. Each of them has its strengths and weaknesses, which also depend on the purpose of the evaluation. In this paper we propose the integration of two different formalisms in a single framework for the modeling, validation, and optimization of production-supply problems. In particular, the proposed framework combines Process Graphs (P-Graphs) as the modeling formalism, and Stochastic Activity Networks (SAN) for the analysis and optimization. The integration proposed in this paper extends the capabilities of the P-Graph formalism to include performance and dependability metrics in the optimization process, without hampering the modeling convenience of P-Graphs. The proposed approach is applied to a case study of the optimization of a power supply network.

1. Introduction

Several formalisms and techniques have been introduced in the literature for the purpose of modeling and evaluation of complex systems [1]. Each of them has its strengths and weaknesses, which also depend on the target and purpose of the evaluation. It is therefore difficult to think that a single formalism can be useful for all the stages and tasks of system planning and development. This is even more relevant in modern systems, which are becoming complex, heterogeneous and interconnected. Actually, the benefit of combining different formalisms in an integrated approach has been largely recognized both at theoretical level [2], and at the practical level, with the introduction of multi-formalism tools [3].

One of the tasks for which several formalisms have been devised is the evaluation and planning domain, in which different system architectures are compared with respect to some criteria. In this paper we combine two different formalisms in a single approach for the modeling, validation, and optimization of production-supply problems. In particular, the framework will use *Process Graphs* (P-Graphs, [4]) as the modeling formalisms and *Stochastic Activity Networks* (SAN, [5]) for the analysis and optimization.

The integration of the two formalisms aims at combining their beneficial features: ease of modeling for P-Graphs and versatility for SAN, while reducing their drawbacks, which consist in the limited evaluation possibilities of P-Graphs, and the need of specific knowledge for developing SAN models. The focus is on Process Network Synthesis problems, which are introduced later.

The paper is organized as follows. Section 2 introduces Process Network Synthesis problems and the P-Graph formalism; Section 3 introduces Stochastic Activity Networks. The approach to process network synthesis that we propose, combining the two formalisms, is described in Section 4. An example application of the proposed methodology is reported in Section 5. Related work is discussed in Section 6, while conclusions are drawn in Section 7.

2. Process Network Synthesis and P-Graphs

2.1. PNS Problems

The definition of the Process Network Synthesis (PNS) [4] problem was motivated by production problems in chemical engineering, in which the objective was to determine an optimal allocation of resources, planning of resources and flows, and management of operating units, thus optimizing the production.

Such problems were initially solved using techniques based on heuristics, dynamic programming, random search, or conventional graphs algorithms, which were efficient only for small scale problems, since the number of possible alternative structures results to be exponential. To overcome such limitations, a new way to formalize such problems was introduced, through the Process Graphs (P-Graphs) formalism [6]. P-Graphs have been later used in other domains, including critical infrastructures like transportation and power grid.

Let M be a given finite set of all materials that have to be involved in the synthesis of a process system; it may be just a set of names, or a set of vectors including characteristics of such materials. A PNS problem involving materials in M is defined by a triple (P, R, O) , where $P \neq \emptyset$ is the

set of products, $R \neq \emptyset$ is the set of raw materials, and $O \neq \emptyset$ is the set of operating units. The problem, which is NP-hard combinatorial problem [7], consists in finding the structure which, using the available operating units (O), is able to produce the desired materials (P) from the specific raw materials (R), and minimizes the cost of the resulting processing network [8], [9].

The basic relations between the above introduced sets are the following:

$$P \subseteq M, \quad R \subseteq M, \quad M \cap O = \emptyset \quad (1)$$

and

$$O \subseteq \mathcal{P}(M) \times \mathcal{P}(M), \quad (2)$$

where $\mathcal{P}(M)$ is the power set of M .

2.2. P-Graphs

A convenient formalism to model PNS problems are P-Graphs [10]. A P-Graph is a directed bipartite graph whose vertices are materials and operating units. Formally [11], given two finite sets M and O , such that Equation 2 holds, a P-Graph is defined to be the pair (M, O) . The vertices of the graph are elements of $V = M \cup O$, while the arcs of the graphs are elements of $A = A_1 \cup A_2$, where:

$$\begin{aligned} A_1 &= \{(x, y) \mid y = (\alpha, \beta) \in O \wedge x \in \alpha\}, \\ A_2 &= \{(y, x) \mid y = (\alpha, \beta) \in O \wedge x \in \beta\}. \end{aligned} \quad (3)$$

A particular kind of materials are *raw materials* $R \subset M$ and *products* $P \subset M$. Raw materials constitute the input of the process, and products are the output produced by the process. The remaining materials are called *intermediate materials*.

An operating unit $(\alpha, \beta) \in O$ consumes its input materials $x_i \in \alpha$ and produces its output materials $x_j \in \beta$. More precisely, being $a: A \rightarrow \mathbb{R}^+$ a function that gives the cardinality of each arc in the P-Graph, the operating unit $\gamma = (\alpha, \beta) \in O$ needs $a(x_i, \gamma)$ units of its input material $x_i, \forall x_i \in \alpha$ its input materials $x_i \in \alpha$, in order to produce $a(\gamma, x_j)$ units of the output material $x_j, \forall x_j \in \beta$. Graphically, operating units are represented as horizontal bars and materials as solid circles; materials from different categories (raw, product, intermediate) are sometimes depicted with circles having different decorations. A graphical example of a P-Graph is depicted in Figure 1 [12].

The synthesis of a processing network consists in choosing the best possible solution (i.e., arrangement of available operating units) among a number of options. This optimisation task has several possible objectives, the most common one being the system profit to be maximised.

Depending on the objective function, the following additional properties are added to operating units and materials. For each operating unit $o \in O$, $fc(o)$ is a fixed cost for including the unit in the solution, and $pc(o)$ is a proportional cost for each unit of time that the unit is working. For each raw material $r \in R$, c_r is the supplying cost for the material, U_r is an upper bound on the available quantity of a raw material. For each product $p \in P$, g_p is the gain associated

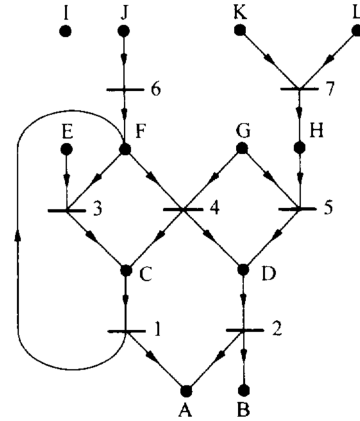


Figure 1: Example P-Graph (M, O) , with materials $M = \{A, \dots, L\}$ and operating units $O = \{1, \dots, 7\}$ [12].

with producing a final product, e.g., the price at which it is sold, and L_p is a lower bound that specifies the minimum quantity of the product that must be produced by the process.

In general, the entire search space of PNS problems is exponential with respect to the size of the problem, and would consist of $2^{|O|} - 1$ different solution-structures. However, the use of P-Graphs and the associated mathematical theory allows computational costs to solve PNS problems to be sensibly reduced. Such reduction of computational costs is due to the fact that not every possible combinatorial structure is an admissible solution. If admissible solutions can be identified, then the search for the optimal structure can be limited to those structures.

2.3. Solution Process

The mathematical framework behind P-Graphs consists of three algorithms that are applied to solve the PNS problem: the *Maximal Structure Generation (MSG)* [10], the *Solution-Structure Generation (SSG)* [13], and the *Accelerated Branch and Bound (ABB)* [12] algorithms.

Admissible solutions must satisfy a set of structural properties of the corresponding P-Graph, which are formalized through five axioms [10]. The set of admissible solution-structures is denoted as $\mathcal{S}(P, R, O)$; this set is closed with respect to the union. The super-structure (i.e., the union) of those solutions is called *maximal structure* for the problem, and it is denoted as $\mu(P, R, O)$. Therefore, the maximal structure is still an admissible solution of the problem. Such structure is generated using the Maximal Structure Generation (MSG) algorithm, which is the first step of the solution process.

The next step consists in the application of the Solution-Structure Generation (SSG) algorithm, which isolates and enumerates, from the maximal structure, all the admissible solutions to the PNS problem.

The optimal solution of the problem is generated by the Accelerated Branch and Bound (ABB) algorithm. General branch-and-bound methods are inefficient at solving

the mixed-integer programming (MIP) problem underlying a process synthesis problem [14]. The ABB algorithm searches for the optimal solution in an efficient way, taking into account of combinatorial properties of solution-structures, and properties ensured by structural axioms of P-Graphs.

3. Stochastic Activity Networks

Stochastic Activity Networks (SAN) can be considered an extension of Stochastic Petri Nets (SPN) [15], a widely used formalism for performance and dependability evaluation of complex systems.

A formal definition of SAN was given by Sanders and Meyer in [5]. We recall here the basic definitions provided in that paper. An Activity Network (AN) is defined as an eight-tuple [5]

$$AN = (P, A, I, O, \gamma, \tau, \iota, o),$$

where: P is a finite set of places; A is a finite set of activities; I is a finite set of input gates; O is a finite set of output gates; $\gamma: A \rightarrow \mathbb{N}^+$ specifies the number of cases for each activity; $\tau: A \rightarrow \{\text{timed, instantaneous}\}$ specifies the type of each activity; $\iota: I \rightarrow A$ maps input gates to activities; $o: O \rightarrow \{(a, c) \mid a \in A \text{ and } c \in \{1, 2, \dots, \gamma(a)\}\}$ maps output gates to cases of activities. If S is a set of places ($S \subseteq P$), a *marking* of S is a mapping $\mu: S \rightarrow \mathbb{N}$. Similarly, the set of possible markings of S is the set of functions $M_S = \{\mu \mid \mu: S \rightarrow \mathbb{N}\}$. The marking of a SAN defines its current state.

An input gate is defined to be a triple, (G, e, f) , where $G \subseteq P$ is the set of input places associated with the gate, $e: M_G \rightarrow \{0, 1\}$ is the enabling predicate of the gate, and $f: M_G \rightarrow M_G$ is the input function of the gate. An output gate is a pair, (G, f) , where $G \subseteq P$ is the set of output places associated with the gate, and $f: M_G \rightarrow M_G$ is the output function of the gate.

A Stochastic Activity Network (SAN) is formed by adding stochastic information, through functions C , F , and G . where C is the probability distribution of case selections, F is probability distribution functions of activity delay times, and G describes the sets of reactivation markings for each possible marking [5].

Formally:

$$SAN = (P, A, I, O, \gamma, \tau, \iota, o, \mu_0, C, F, G).$$

Informally, the number of tokens in each SAN place define the current state of the model. Activities, based on predicates specified in input gates can become enabled; if an activity stays continuously enabled it will fire after a stochastic delay. When an activity fires, it can modify the current state of the model. Different kinds of metrics can be defined on SAN models by means of reward variables, which define the amount of reward that is accumulated for the permanence in specific states, or for the firing of specific activities. Reward variables can be evaluated using numerical methods or by discrete-event simulation, using the solvers provided with the Möbius tool [3].

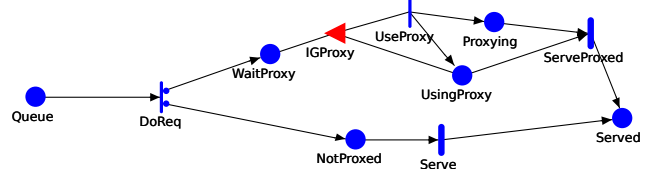


Figure 2: Example SAN model of a proxied web service [16].

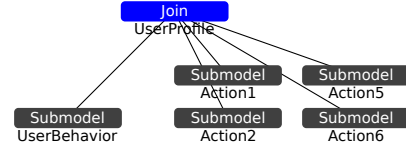


Figure 3: Composition of SAN models using the Rep/Join formalism.

Graphically, SAN use a notation similar to Petri Nets: activities are represented by vertical bars (thick bars for timed activities, thin bars for immediate ones), and places by circles. Input gates are denoted by red triangles, and output gates by black triangles; cases of activities are represented as small circles attached to the activity. An example SAN model of a web service using a proxy is depicted in Figure 2 [16]. Extensions supported by the Möbius tool include *extended places*, which can contain values of different types as tokens, including structured datatypes, instead of just an integer number of tokens like ordinary places.

SAN models can be combined together to form more complex models, using the Rep/Join formalism [17] (Figure 3). This feature is often exploited to improve the modularity of the models (e.g., see [18], [19], [20]), possibly using automated generation and composition approaches [21].

4. Combining P-Graphs and SANs

As discussed in Section 2, using P-Graphs makes it possible to model PNS problems conveniently, and at the same, through the underlying theory, to analyze the structural properties and find an optimal structure considering the nominal behavior of the production process. However, the solutions produced by the analysis of P-Graphs take into account only the nominal attributes associated with materials, products, and operating units under analysis, without performing more precise behavioral analysis of individual solution-structures, thus introducing some limitations and simplifications.

More precisely, such limitations consist in not taking into account possible interactions and interdependencies between process network components, and in a partial representation of time-related concepts. In fact, each operating unit has a fixed cost, a cost proportional to its usage, and produces a certain amount of material working during the year, but there is no information on how such time is actually distributed. Interactions such as possible waiting due to the absence of input materials, failures, or other events are thus disregarded. As a consequence, this also prevents a precise

estimation of the time required by the solution-structure in order to produce the amount of products requested by the given PNS problem.

We propose to overcome such limitations by combining the P-Graphs framework with the SAN formalism, as we describe in the rest of this chapter.

4.1. The Proposed Workflow

The basic idea is to use P-Graphs as modeling formalism, and then produce, using an automated transformation, the SAN model to be used for the analysis only. The size of the resulting SAN model, as well as the number of evaluations that need to be performed, is kept to a minimum by applying algorithms of the P-Graph theory.

This combines P-Graphs and SAN and exploits their best characteristics. P-Graph allow PNS problems to be modeled in a very convenient way; however, as previously mentioned, the underlying theory has some limitations in the phenomena that can be analyzed. More rich analyses can be performed through the use of SANs, which however are less easy to adopt for the modeling process, and are not supported by a sound theory allowing to enumerate all the possible admissible subnets (according to some criteria).

The main steps of this approach are described in the following; within square brackets we specify if the step mainly relies on P-Graphs or SAN concepts. Given a PNS problem [4] (P, R, O):

- 1) **[P-Graphs]** The PNS problem is modeled using the P-Graphs formalism; thanks to their simplicity the problem is easily modeled.
- 2) **[P-Graphs]** The MSG [10] algorithm is applied to the problem, obtaining the maximal solution-structure, i.e., the union of all the solution-structures that satisfy the structural axioms.
- 3) **[SAN]** Using a homomorphism that we define in this paper (Section 4.2), the maximal solution-structure is transformed into a SAN model representing detailed properties and behavior of operating units and materials. The resulting model includes specific variables that, depending on their value, enable or disable entire branches of the maximal solution-structure.
- 4) **[SAN]** Reward variables and structures are defined on the resulting SAN, in order to evaluate the measures of interest, i.e., the metrics based on which the solution-structures should be compared.
- 5) **[P-Graphs]** The SSG [13] algorithms is applied to the maximal solution-structure, in order to isolate all the individual admissible solution-structures.
- 6) **[SAN]** Instead of applying the ABB algorithm [12] to obtain the optimal solution (or the n sub-optimal ones), the SAN model obtained in Step 3 is configured to be able to represent all the admissible solution-structures obtained in Step 5. This means that, for each solution-structure, ad-hoc settings for the fork variables are devised, so that branches

not existing in the current solution-structure are disabled in the SAN model.

- 7) **[SAN]** Finally, following the enumeration produced by the SSG algorithm, admissible solution-structures are evaluated using the SAN model. The optimal one is devised by analyzing the obtained valued for the metrics of interest.

In the following we detail the transformation that generates the SAN model of Step 3.

4.2. P-Graph to SAN Transformation

It should be noted that models of both formalisms, P-Graphs and SANs, can be represented as bipartite graphs, which suggests that a correspondence can be established between the two formalisms. The most obvious mapping between the two formalisms is to define an homomorphism between places and materials, and between transitions and operating units, as formally defined by the authors of [22].

However, in this paper we do not want to just define a correspondence between the two formalisms, but rather to extend the possible analysis that can be performed on a P-Graph model, by combining it with a SAN model, so that more rich analyses can be performed, taking into account more detailed behavior of materials and operating units.

The idea is this to define a new *homomorphism* (or, using MDE [23] terminology, a *model transformation*) between the two formalisms, which is defined in the rest of this section. Such homomorphism is used to automatically derive a SAN model corresponding to the maximal solution-structure, and able to represent all the individual admissible solution-structures.

We consider P-Graph nodes to have the following *nominal* attributes:

- *Raw materials*: type; unit of measurement; available quantity; cost.
- *Final products*: type; unit of measurement; requested quantity; reward.
- *Operating units*: upper bound and lower bound of utilization capacity; fixed costs; proportional costs; input and output rate (and related units of measurement) of materials; utilization factor; amount of used material and produced material within a solution.

Since we want to extend the possible analysis types, we aim to be able to take into account also for the following aspects in the resulting SAN model:

- Execution time of operating units.
- Failure rate of operating units.
- Recovery rate (in case of failure) of operating units.
- Possibly, the impact due to limited capacity of stockage, in terms of interruption of the production of operating units.
- Concurrent execution of operating units in parallel branches (if admitted by the maximal solution).
- Interactions between the different operating units composing a solution.

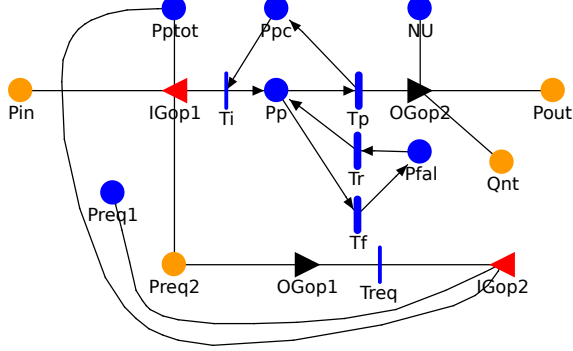


Figure 4: Structure of the SAN sub-net generated for each operating unit.

The homomorphism we define in the following considers two main cases: *materials*, which are directly mapped to *places* of the SAN formalism, and *operating units*, which instead are mapped to whole subnets (i.e., models) defined in the SAN formalism.

Given a P-Graph (M, O) , the corresponding SAN model can be defined through the injective homomorphism (monomorphism)

$$\mathcal{H}: M \cup O \rightarrow \mathcal{P} \cup \mathcal{S}, \quad (4)$$

where \mathcal{P} is a set of places and \mathcal{S} is the set of all the possible SAN models. Moreover, \mathcal{P} is the union of all the possible places belonging to SAN models in \mathcal{S} , i.e.:

$$\mathcal{P} = \bigcup_{((P, A, I, O, \gamma, \tau, \ell, o), \mu_0, C, F, G) \in \mathcal{S}} P, \quad (5)$$

For each $x_i \in M \cup O$, the function \mathcal{H} is defined as:

$$\mathcal{H}(x_i) = \begin{cases} P_i, \text{ with } P_i \in \mathcal{P} & \text{if } x_i \in M, \\ \gamma(x_i), \text{ with } \gamma(x_i) \in \mathcal{S} & \text{if } x_i \in O, \end{cases} \quad (6)$$

where $\gamma: O \rightarrow \mathcal{S}$ is a function that, considering the values of attributes associated with operating units, generates a corresponding SAN subnet. Such SAN subnet should follow a precise structure that is detailed in the next section.

4.3. SAN Subnet for Operating Units

As described in the previous section, each operating unit in the P-Graph is transformed to a specific SAN subnet that models its properties and behavior. Elements of such SAN subnet are reported in the following, with a brief description of the P-Graph characteristics, or extended aspects, they allow to be modeled. A graphical representation of the subnet is depicted in Figure 4. Places depicted in yellow are extended places (see Section 3).

- Places P_{in} and P_{out} model respectively containers for input and output materials of the operating unit. The choice of using extended places is due to the possibility to handle non-integer quantities. In case the unit has multiple input or output materials, those are represented as multiple places, which are still

connected to the input gate $IGop1$ (for input materials) or output gate $OGop2$ (for output materials).

- Places P_{ptot} and P_{pc} keep track of the maximum utilization capacity and the current utilization of the operating unit. A token in P_{pc} represents the case in which the operating unit is free, otherwise the unit is in use. Place P_{ptot} keeps track of the maximum utilization capacity of the individual operating unit, while the number of tokens in P_{pc} keeps track of the utilization capacity of different operating units that are identically distributed in the production, failure and recovery times.
- Places P_{req1} and P_{req2} are used to model a production request. When place P_{req1} is not empty a production request is active; the amount of material to be produced is given by the number of tokens in place P_{req2} . If the operating unit can be used (place P_{NU}), and the amount of input material is enough, the marking of place P_{req2} gives the amount of material to be produced.
- Place Qnt keeps track of the quantity of material that is produced, which is used to calculate the utilization factor with which the operating unit is being used.
- Place P_{fal} models the failure of the operating unit.
- Place P_p models an operating unit in production state.
- Place P_{NU} is used to distinguish between usable and unusable operating units. A token in this place means that the operating unit cannot be used. An operating unit cannot be used if there is not enough input material, and the producers of such materials are not usable as well, or if the unit has exhausted its producing capability and it is not currently operating.
- The immediate activity T_i models the operating unit entering in operational state, which means that the quantity of material needed for production is subtracted to the amount of available input material. Such activity is enabled if: i) there is enough amount of input materials (place P_{in}), ii) there is an active production request (place P_{req2}), iii) the operating unit is not already operating (place P_{op}), and iv) its maximum utilization capacity has not been exhausted (place P_{ptot}).
- The immediate activity T_{req} models taking over of a production request received from the outside, i.e., from another operating unit if the unit produces an intermediate material, or from an user if the operating unit produces a final product.
- The timed activity T_p models the execution of the operating unit, and its production time. This timed activity, as well as the following ones, can follow any probability distribution; they are not necessarily restricted to the exponential distribution.
- The timed activity T_f models the occurrence of a failure of the operating unit.
- The timed activity T_r models the restore of functionality of the operating unit, following to the occurrence of a failure.

4.4. Composing the Maximal Structure

Thanks to the homomorphism and transformation defined in the previous sections, a maximal solution-structure, modeled with a P-Graph, can be transformed in the corresponding SAN model that models the detailed behavior of operating units.

The SAN model corresponding to the maximal solution-structure is obtained as follows:

- For each *operating unit* $x_i \in O$, the function $\mathcal{H}(x_i)$ is defined so that a SAN subnet like the one in Section 4.3 is obtained.
- For each *raw material* $x_i \in R$, the function $\mathcal{H}(x_i)$ is defined so that the material is mapped to place P_{in} of the operating unit(s) that use that material as input material.
- For each *final product* $x_i \in P$, the function $\mathcal{H}(x_i)$ is defined so that the product is mapped to the P_{out} place of the operating unit(s) that produce that product. An additional place is introduced to model the quantity of the final product that is requested.
- For each *intermediate material* $x_i \in M \setminus (P \cup R)$, the function $\mathcal{H}(x_i)$ is defined so that the material is mapped to the P_{out} place of the operating unit(s) that produce that material; or, equivalently, to the P_{in} place of the operating unit(s) that use that material as input.

Then, after having transformed every single element $x_i \in M \cup O$ to $H(x_i)$, the generated subnets are composed using the Rep/Join formalism [17], to obtain a SAN model that represents the maximal structure. The topology of connections between the generated SAN submodels follows the connection paths of the maximal structure itself. Connections between submodels occur on places representing materials, and in particular intermediate materials: places P_{out} of operating units that generate a given material are connected with the corresponding P_{in} place of operating units that use that material as input. Similarly, if two or more operating units use the same raw material as input, or produce the same final product as output, then the corresponding P_{in} (or P_{out}) places are connected together.

The procedure described so far generates a SAN model able to represent the maximal structure. However, we want to be able to represent any of the solution-structure generated by the SSG algorithm. To accomplish this task, we add a set of variables, called *branch enabling variables*, to the resulting SAN model, so that specific parts of the model can be enabled or disabled at will. In particular, a branch enabling variable F_i is added for every operating unit at every fork (or at every join) of the production flow. Based on the value of such variables, which is given by the actual solution-structure to be represented, entire branches of the production system can be disabled or ordering introduced between them.

Adopting such approach stems from the need to produce results that are consistent with the original P-Graphs framework. In fact, P-Graphs algorithms assume that certain

operating units can be excluded from the final solution-structure, and that input materials can be directed with higher priority to a specific operating unit with respect to another.

4.5. Specification of Metrics

In order for the approach to be useful, the generated SAN model should support, in addition to the extended analyses, also the evaluation of the basic metrics that can be evaluated using the P-Graph theory. With the SAN formalism, metrics are specified using reward structures. In this section we describe the reward structures that permit the evaluation of the basic metrics that can be evaluated using P-Graph theory.

The main metric that is typically evaluated by the ABB algorithm is the *profit*. Its implementation in the SAN model is a composed metric, obtained as a combination of the following performance variables. In the following, \mathcal{T} is the set of activities of the SAN model, \mathcal{RS} is the reachability set of the SAN model, i.e., the set of reachable states, $R(m)$ is the reward associated with being in marking (state) m , and $C(a)$ is the reward associated with the firing of activity a .

- Q_t^k : *Total cost for raw material k*. This is an interval of time variable, returning the total cost for using raw material k up to time t . It uses the following reward structure:

$$\begin{aligned} R(m) &= 0, & \forall m \in \mathcal{RS}, \\ C(a) &= \begin{cases} c_k \cdot n & \text{if } a = T_i^k, \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \quad (7)$$

where T_i^k is an activity that takes n tokens from an input place representing material m_k , and c_k is the unit cost of that material.

- U_t^k : *Utilization factor of operating unit k*. This is an instant of time variable, returning the current utilization factor of the operating unit. It uses the following reward structure:

$$\begin{aligned} R(m) &= \begin{cases} \#m & \text{if } m = P_{tot}, \\ 0 & \text{otherwise,} \end{cases} \\ C(a) &= 0, & \forall a \in \mathcal{T}. \end{aligned} \quad (8)$$

- P_t : *Cost of the production process*. This is an interval of time variable, returning the total cost of the production process up to time t . It uses the following reward structure:

$$\begin{aligned} R(m) &= 0, & \forall m \in \mathcal{RS}, \\ C(a) &= \begin{cases} pc(o_i) & \text{if } a = T_p^i, \forall o_i \in O, \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \quad (9)$$

where $pc(o_i)$ is the proportional cost for operating unit o_i , and T_p^i is the activity corresponding to the execution of a unit of production for the operating unit, for all the operating units in the model.

- G_t : *Gain for final products*. This is an instant of time variable, returning the total gain obtained from the production of final products, up to time t . It uses the following reward structure:

$$R(m) = \begin{cases} \#m \cdot g_k & \text{if } m = P_{out}^k, \forall m_k \in P \\ 0 & \text{otherwise,} \end{cases}$$

$$C(a) = 0, \quad \forall a \in \mathcal{T}, \quad (10)$$

where P_{out}^k is the place corresponding to product m_k , and g_k is the gain associated with producing that material; P is the set of all products.

The final value for the *profit* metric is thus obtained as:

$$Profit_t = G_t - \left(\sum_{k=R} Q_t^k + P_t + \sum_{o \in O' \subseteq O} fc(o) \right), \quad (11)$$

where t is the time at which the profit is evaluated, and $O' \subseteq O$ is the set of the operating units included in the analyzed solution-structure.

Other metrics that can be obtained using the P-Graph theory can be easily obtained starting from the above ones, or with little modifications to their reward structures, and are thus not included here.

5. Case Study: Optimization of a Power Grid

To show the benefits of our approach, we now apply the proposed workflow to a case study of the optimization of a power grid. The case study presented here is an extension of the one analyzed by the authors of [14], in which a network for the distribution of electricity and heat was analyzed, with particular attention to the environmental sustainability of the structure. The network is designed using P-Graphs, trying to minimize costs and keeping the environmental impact low. The problem is expressed through the specification of consumption, production capacity, cost, environmental impact, and *emergy* [24], of available materials and of available technologies to fulfill the annual request for electricity and heat of the considered district.

The optimization function aims at i) maximizing the profit and ii) minimizing the ecological footprint (EF) indicator [25], an indicator of environmental sustainability.

5.1. The Problem and Initial Results

In this section we extend the analysis performed in [14] using the proposed approach. The extended analyses will take into account additional metrics related to the reliability and performance of the obtained production system.

The maximal structure for such problem (obtained by the application of the MSG algorithm in Step 2) is shown in Figure 5 [14]. In addition to the usual sources of electrical power and heat, the considered district also contains renewable power sources, like corn silage, grass silage, corn straw, and wood. Different technologies are available for energy

TABLE 1: Additional properties used for performability analysis (time in hours).

Unit	Production Time	MTTF	MTTR
Electricity Feeder	0.05	24	2
Natural Gas Burning	0.0005	48	3
Biogas CHP Corn	0.01	36	0.16
Biogas CHP Grass	0.02	36	2
Biogas Prod. Corn	0.001	72	0.007
Corn Silage Prod.	0.01	28	0.1
Grass Silage Prod.	0.01	36	1
Biogas Burning	1	12	0.5

conversion, like biogas plants, biogas combined heat and power (CHP), gas burners, pelletizers, and burners. Cost, ecological footprint, and emergy are modeled as materials, so that upper/lower bounds can be established and the space of possible solutions reduced.

The application of the SSG algorithm results in 21 solution-structures [14], of which one (Structure 13) is the reference structure, i.e., the structure currently in operation.

With respect to the structure which is currently deployed, other structures that make use of renewable energy sources allow the costs and/or the ecological footprint to be reduced (Figure 6). For example, with Structure 7 costs are reduced by about 5%, with also a reduction of 29% in the ecological footprint. Structure 10 would also be an improvement, reducing costs by about 3% and reducing the ecological footprint by 16%. For a higher reduction of the ecological footprint (by 78%) Structure 16 could be adopted, at the price of an increase of costs by 1.5%.

5.2. Analyses Extensions with SAN

Thanks to the approach we proposed in the previous sections, we can complement such results with more detailed analyses, taking into account for random failure of processing units and subsequent repairs, the time needed to units to satisfy requests, and the possibility of the process network to work in degraded mode, i.e., with some of the operating units non working due to failures.

Such extended analysis allow the decision maker to have a wider view of the existing tradeoffs between the admissible solution-structures. For example, the optimal solution with respect to costs could not be the most convenient one, if the failure probability is too high, or if it takes too much time to actually fulfill request for the needed amount of products.

To analyze those aspects it is necessary to know the additional properties of operational units, and in particular the *mean production time*, the *mean time to failure* (MTTF), and the *mean time to repair* (MTTR). The adopted values for such properties are reported in (Table 1). Unfortunately it was not possible to recover real data for this parameters; values in the table are thus provided as example data, to demonstrate the applicability of the approach. Additionally, we assumed that the requests for power and heat follow an exponential distribution with a mean request time of 5 hours.

Based on this additional information, we can now evaluate additional metrics, namely:

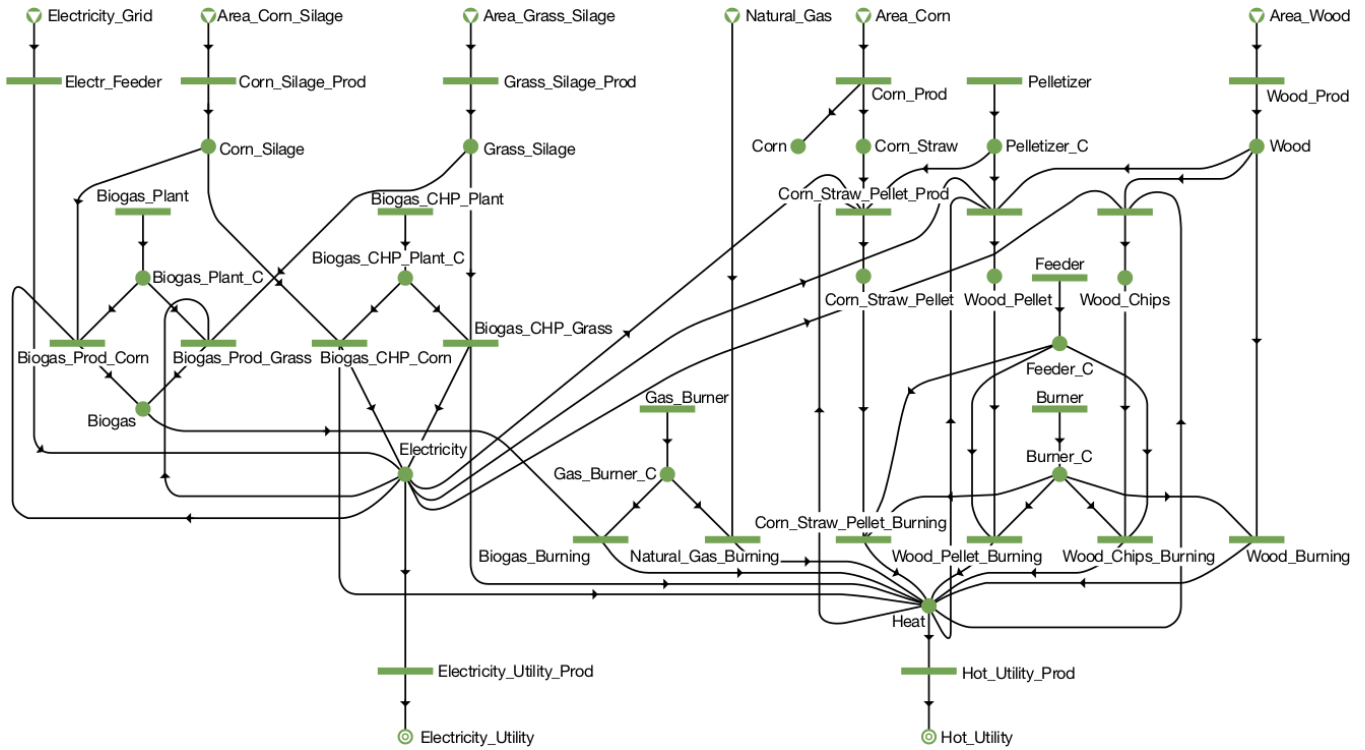


Figure 5: Maximal structure of the power grid [14].

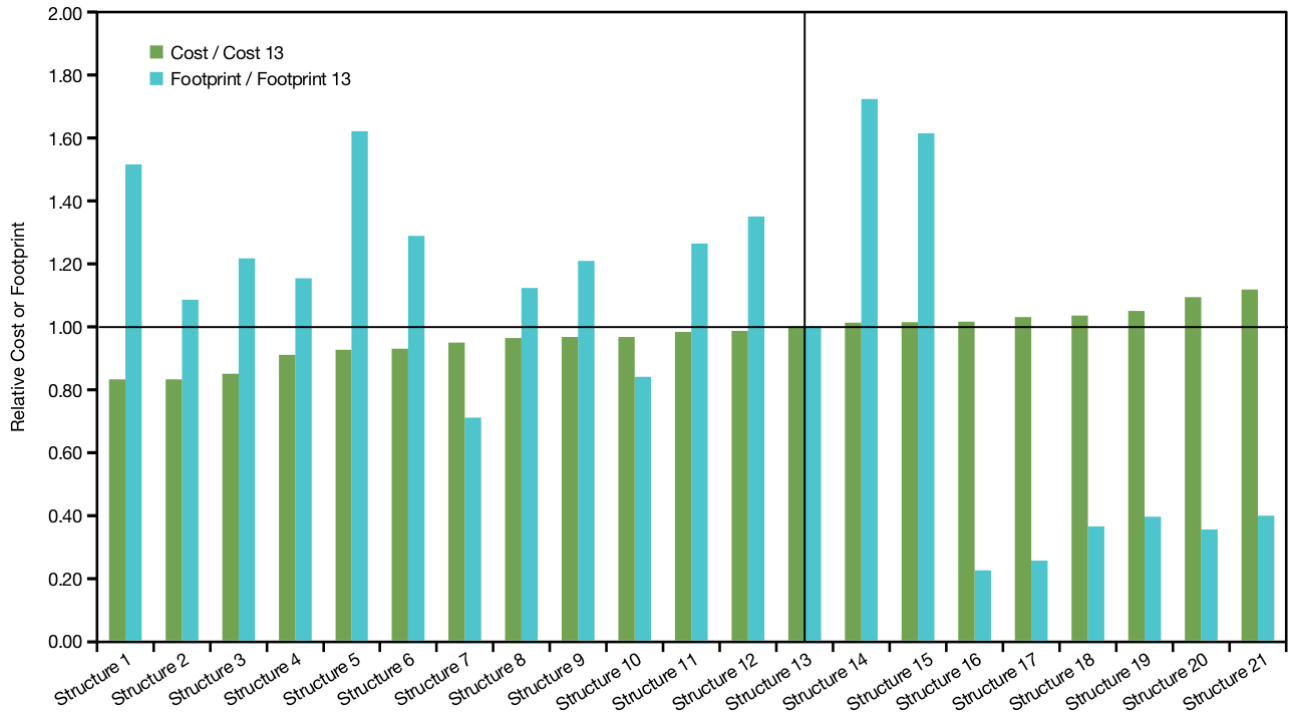


Figure 6: Comparison of the admissible solution-structures according to the original evaluation based on P-Graphs only [14]. Structure 13 is the reference structure, being the one currently deployed and operating.

- *Mission Time*, i.e., the time required to the solution-structure to produce the requested quantity of products;
- *Mean Time To Failure* (MTTF);
- *Mean Time To Repair* (MTTR);
- *Mean Time To First Failure* (MTTFF).

Ideally, such metrics can be computed for all the admissible solution-structures reported in Figure 6. However, we limit here our discussion only to Structure 13, the currently deployed one, and those that could actually provide an improvement with respect to costs and ecological footprint, i.e., Structures 7, 10, and 16. The transformation of Section 4 is then applied to the P-Graph of Figure 5, and then each of these structures is modeled by properly assigned values to the F_i branch enabling variables in the obtained SAN model corresponding to the maximal solution-structure.

We have evaluated metrics using the discrete-event simulator provided with the Möbius framework, using at least 1000 simulation batches and a confidence level of 95%. Metrics that were originally evaluated in [14] have been evaluated again using SAN, to demonstrate the consistency of the proposed approach. The obtained values are reported in Table 2; for each dimension, the solution-structure with the best value is highlighted in bold.

The evaluation of the extended set of metrics, using the proposed approach, allow result-structures to be better characterized according to multiple dimensions. Referring to the analyzed example it is possible to highlight a number of interesting aspects. First of all, it should be noted that results for the two metrics used in the original P-Graphs analysis, cost and ecological footprint, are consistent with the ones obtained with P-Graphs analysis [14]. For example, the cost with Structure 16 is slightly increased, but the ecological footprint is sensibly reduced, which is consistent with the results obtained in [14] (Figure 6).

According to the original results, Structure 16 was the most ecological one, and required only a small cost increment, thus making its implementation very appealing if reducing the footprint was a priority. However, the new extended analysis now suggest that this structure requires a much longer time to satisfy requests: three times worse with respect to the reference structure. Furthermore, it also has a very low MTTF, possibly resulting in availability problems.

Similar comments can be addressed towards the reference structure, Structure 13, which has limits concerning reliability and average production time: improvements are possible also on these dimensions. The two structures that provide the greatest improvements with respect to costs, Structure 7 and Structure 10, would also improve the overall production time (mission time). However, the two solutions have very different values for the MTTF, MTTR, and MTFF properties, with Structure 7 being much more reliable. In fact, the MTTF is more than twice larger, and the MTTR is about 4 times smaller.

This leads us to conclude that the best tradeoff among all the solutions to the PNS problem is to implement Structure 7. which would improve all the target metrics with respect

to the currently deployed system (Structure 13). It should be noted that this result becomes evident only after applying the extended analyses using SAN, through the method proposed in this paper.

6. Related Work

Model-based evaluation [1] has been largely applied for the evaluation of different kinds of systems and properties. In particular, models based on Stochastic Petri Nets have been applied to many domains, including for example multiprocessor systems [18], power-grids [26], tele-immersive applications [19], and many others. Initially introduced for the synthesis of chemical engineering processes, P-Graphs [4] have later been applied to a variety of other domains as well, including the optimization of a distribution of power and heat, like the example extended in this paper [14], transportation [27], and even the planning of evacuation routes [28].

Previously, other work in the literature investigated similarities between P-Graphs and Petri Nets. One of the first work in such direction is reported in [22], where the authors investigate how the the specific combinatorial features of PNS problems can be applied for developing more efficient mathematical methods for the analysis of Petri Nets. Such direction was then expanded in [29], where P-Graphs and Petri Nets form part of a mathematical framework focused on specifying and solving optimal trajectory problems (i.e., sequences of transition firings such that, from the initial marking, the net reaches a specific marking). Both these work focus on applying P-Graph algorithms to models specified using Petri Nets. Conversely, the approach we propose in this paper applies evaluation techniques for Stochastic Petri Nets on models initially specified using P-Graphs. To the best of our knowledge, investigations in this direction have been limited.

The approach that we adopt in this paper can be seen as a model transformation approach [30], in which the source model are P-Graphs, and the target are Stochastic Activity Networks. Model transformation has been applied to the analysis of performance and dependability problems, most often as an automated way to analyze models specified with the UML language, e.g. see [31]. While model transformation has been extensively applied to the evaluation of UML models, we are not aware of other proposals that apply model-transformations to generate Stochastic Petri Net models representing a PNS problem specified with P-Graphs.

7. Concluding Remarks

In this paper we proposed a new framework for addressing Process Network Synthesis problems. The framework combines the simplicity of P-Graphs, which are used for the specification of the problem, and for identifying a set of admissible solutions, with the flexibility and power of SAN, which are used for detailed analysis of measures of interest. Such combination enables the evaluation of additional

TABLE 2: Measures evaluated for the discussed structures (time in hours). For each dimension, the solution-structure with the best value is highlighted in bold.

Structure	Cost	Footprint	Mission Time	MTTF	MTTR	MTTFF
S ₇	543.71	489.81	2655.14	433.75	0.16	397.04
S ₁₀	554.01	580.01	2514.31	188.43	0.86	158.27
S ₁₃	572.96	690.61	3052.09	117.35	2.03	63.09
S ₁₆	581.31	154.24	10590.96	16.28	0.50	231.32

dimensions, including dependability and performance, thus allowing an expanded analysis of tradeoffs. The integration, which can be automated, can be implemented using model transformation technologies.

The proposed approach has been applied to a case study of a network for the distribution of electricity and heat. The analysis shown consistent results with those originally obtained with P-Graphs only, and provided new insights on the other admissible solution-structures, thus demonstrating the usefulness of the approach. As future work, we aim to implement the proposed transformation as an extension of tools currently available for the manipulation of P-Graphs. At the same time, limitations of the proposed approach should be investigated. In particular, solutions to mitigate state-space explosion need to be devised. The nature of models generated from PNS problems, in which for example operating units may be temporarily disabled due to lack of materials, suggests the investigation of decomposition techniques [32].

Acknowledgment

This work has been partially supported by the Joint Program Initiative (JPI) Urban Europe via the IRENE project, by the European FP7-ICT-2013-10-610535 AMADEOS project and by the DEVASSES project, funded by European Union’s Seventh Framework Programme under grant agreement PIRSES-GA-2013-612569.

References

- [1] D. M. Nicol, W. H. Sanders, and K. S. Trivedi, “Model-based evaluation: from dependability to security,” *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 48–65, Jan.–March 2004.
- [2] A. Bondavalli, O. Hamouda, M. Kaánchez, P. Lollini, I. Majzik, and H.-P. Schwefel, “The hidden holistic approach for the analysis of large critical mobile systems,” *IEEE Transactions on Mobile Computing*, vol. 10, no. 6, pp. 783–796, June 2011.
- [3] T. Courtney, S. Gaonkar, K. Keefe, E. Rozier, and W. Sanders, “Möbius 2.3: An extensible tool for dependability, security, and performance evaluation of large and complex system models,” in *39th IEEE/IFIP International Conference on Dependable Systems Networks (DSN’09)*, Estoril, Lisbon, Portugal, 2009, pp. 353–358.
- [4] F. Friedler, L. T. Fan, and B. Imreh, “Process network synthesis: Problem definition,” *Networks*, vol. 31, no. 2, pp. 119–124, 1998.
- [5] W. H. Sanders and J. F. Meyer, “Stochastic activity networks: formal definitions and concepts,” in *Lectures on formal methods and performance analysis*. Springer-Verlag New York, Inc., 2002, pp. 315–343.
- [6] P. Varbanov and F. Friedler, “P-graph methodology for cost-effective reduction of carbon emissions involving fuel cell combined cycles,” *Applied Thermal Engineering*, vol. 28, no. 16, pp. 2020–2029, 2008.
- [7] M. Brendel, F. Friedler, and L. Fan, “Combinatorial foundation for logical formulation in process network synthesis,” *Computers Chemical Engineering*, vol. 24, no. 8, pp. 1859–1864, 2000.
- [8] B. M. Beamon, “Designing the green supply chain,” *Logistics Information Management*, pp. 332–342, 1999.
- [9] M. J. Meixell and V. B. Gargeya, “Global supply chain design: A literature review and critique,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 41, no. 6, pp. 531–550, 2005.
- [10] F. Friedler, K. Tarján, Y. Huang, and L. Fan, “Graph-theoretic approach to process synthesis: axioms and theorems,” *Chemical Engineering Science*, vol. 47, no. 8, pp. 1973–1988, 1992.
- [11] F. Friedler, K. Tarjan, Y. Huang, and L. Fan, “Graph-theoretic approach to process synthesis: Polynomial algorithm for maximal structure generation,” *Computers Chemical Engineering*, vol. 17, no. 9, pp. 929–942, 1993.
- [12] F. Friedler, J. B. Varga, and L. T. Fan, “Decision-mapping: A tool for consistent and complete decisions in process synthesis,” *Chemical Engineering Science*, vol. 50, no. 11, pp. 1755–1768, 1995.
- [13] F. Friedler, K. Tarjan, Y. Huang, and L. Fan, “Combinatorial algorithms for process synthesis,” *Computers Chemical Engineering*, vol. 16, Supplement 1, pp. S313 – S320, 1992, european Symposium on Computer Aided Process Engineering – 1, 23rd European Symposium of the Working Party on Computer Aided Process Engineering, 463rd Event of the European Federation of Chemical Engineers (EFChE).
- [14] B. Bertok, H. Cabezas, I. Heckl, and F. Friedler, “Use the p-graph framework to design supply chains for sustainability,” *CEP Magazine*, 2015.
- [15] G. Ciardo, R. German, and C. Lindemann, “A characterization of the stochastic process underlying a stochastic petri net,” *IEEE Transactions on Software Engineering*, vol. 20, no. 7, pp. 506–515, Jul 1994.
- [16] L. Montecchi, N. Nostro, A. Ceccarelli, G. Vella, A. Caruso, and A. Bondavalli, “Model-based evaluation of scalability and security tradeoffs: a case study on a multi-service platform,” *Electronic Notes in Theoretical Computer Science*, vol. 310, pp. 113–133, 2015.
- [17] W. H. Sanders and J. F. Meyer, “Reduced base model construction methods for stochastic activity networks,” *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 1, pp. 25–36, January 1991.
- [18] M. Rabah and K. Kanoun, “Performability evaluation of multipurpose multiprocessor systems: the “separation of concerns” approach,” *IEEE Transactions on Computers*, vol. 52, no. 2, pp. 223–236, 2003.
- [19] N. R. Veeraragavan, L. Montecchi, N. Nostro, R. Vitenberg, H. Meling, and A. Bondavalli, “Modeling que in dependable tele-immersive applications: A case study of world opera,” *IEEE Transactions on Parallel and Distributed Systems*, 2016, To appear.
- [20] K. Kanoun and M. Ortalo-Borrel, “Fault-tolerant system dependability-explicit modeling of hardware and software component-interactions,” *IEEE Transactions on Reliability*, vol. 49, no. 4, pp. 363–376, 2000.

- [21] L. Montecchi, P. Lollini, and A. Bondavalli, "A DSL-Supported Workflow for the Automated Assembly of Large Performability Models," in *Proceedings of the 10th European Dependable Computing Conference*, 2014.
- [22] S. Gyapay, A. Pataricza, J. Sziray, and F. Friedler, "Petri net-based optimization of production systems," in *Proc. of the 6th IEEE International Conference on Intelligent Engineering Systems (INES-2002)*, Opatija, Croatia, May 26–28 2002.
- [23] D. C. Schmidt, "Guest editor's introduction: Model-driven engineering," *Computer*, vol. 39, no. 2, pp. 25–31, February 2006.
- [24] P. Vassallo, S. Bastianoni, I. Beiso, R. Ridolfi, and M. Fabiano, "Emergy analysis for the environmental sustainability of an inshore fish farming system," *Ecological Indicators*, vol. 7, no. 2, pp. 290–298, 2007.
- [25] W. Rees and M. Wackernagel, *Our ecological footprint*. Gabriola Island, BC and Stony Creek, CT: New Society Publishers, 1996.
- [26] S. Chiaradonna, F. Di Giandomenico, and P. Lollini, "Definition, implementation and application of a model-based framework for analyzing interdependencies in electric power systems," *International Journal of Critical Infrastructure Protection*, vol. 4, no. 1, pp. 24–40, 2011.
- [27] M. Barany, B. Bertok, Z. Kovacs, F. Friedler, and L. T. Fan, "Solving vehicle assignment problems by process-network synthesis to minimize cost and environmental impact of transportation," *Clean Technologies and Environmental Policy*, vol. 13, no. 4, pp. 637–642, 2011.
- [28] J. García-Ojeda, B. Bertok, F. Friedler, and L. Fan, "Building-evacuation-route planning via time-expanded process-network synthesis," *Fire Safety Journal*, vol. 61, pp. 338–347, 2013.
- [29] S. Varró-Gyapay, "Trajectory set approximation for optimization and verification of it systems," Ph.D. dissertation, Budapest University of Technology and Economics, Department of Measurement and Information Systems, September 2014.
- [30] K. Czarnecki and S. Helsen, "Classification of model transformation approaches," in *Proceedings of the 18th Annual SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*. Anaheim, CA, USA: ACM, 2003.
- [31] L. Montecchi, P. Lollini, and A. Bondavalli, "Towards a MDE Transformation Workflow for Dependability Analysis," in *Proceedings of the 16th IEEE International Conference on Engineering of Complex Computer Systems*, 2011, pp. 157–166.
- [32] P. Lollini, A. Bondavalli, and F. Di Giandomenico, "A decomposition-based modeling framework for complex systems," *IEEE Transactions on Reliability*, vol. 58, no. 1, pp. 20–33, March 2009.