Computers and Chemical Engineering 60 (2014) 154–171

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

Computational methods for the simultaneous strategic planning of supply chains and batch chemical manufacturing sites

Gabriela Corsano[a], Gonzalo Guillén-Gosálbez[b,*], Jorge M. Montagna[a]

[a] INGAR – Instituto de Desarrollo y Diseño (CONICET-UTN), Avellaneda 3657, S3002 GJC Santa Fe, Argentina
[b] Departament d'Enginyeria Química, Universitat Rovira i Virgili, Av. Països Catalans, 26, Tarragona E-43007, Spain

ARTICLE INFO

Article history:
Received 7 March 2013
Received in revised form 20 August 2013
Accepted 3 September 2013
Available online xxx

Keywords:
Supply chain
Design and planning
Mixed-integer linear programming
Large-scale optimization
Decomposition methods

ABSTRACT

In this work we present efficient solution strategies for the task of designing supply chains with the explicit consideration of the detailed plant performance of the embedded facilities. Taking as a basis a mixed-integer linear programming (MILP) model introduced in a previous work, we propose three solution strategies that exploit the underlying mathematical structure: A bi-level algorithm, a Lagrangean decomposition method, and a hybrid approach that combines features from both of these two methods. Numerical results show that the bi-level method outperforms the others, leading to significant CPU savings when compared to the full space MILP.

1. Introduction

The concept of supply chain management (SCM), which appeared in the early 1990s, has recently raised a lot of interest since the opportunity of an integrated management of the supply chain (SC) can lead to significant economic benefits. In the context of Process Systems Engineering (PSE), the optimal integration of supply, manufacturing and distribution activities is the main goal of the emerging area known as enterprise-wide optimization (EWO), which as opposed to SCM, places more emphasis on the manufacturing stage (Grossmann, 2005). In the recent past, there have been many contributions in the area of SCM. Excellent reviews on the topic, with emphasis on supply chain design and planning using mathematical programming tools, can be found in the works by Melo, Nickel, and Saldanha-da-Gama (2009), Papageorgiou (2009) and Grossmann (2012).

1.1. Integration of decision levels in SCM

From a functional viewpoint, decisions made in SCM have been traditionally divided in three basic levels according to their temporal and spatial scale: strategic, tactical and operational. Several authors have recognized the importance of integrating decision levels in SCM as an effective manner to increase the overall profit (Goetschalckx, Vidal, & Dogan, 2002; Grossmann, 2004; McDonald & Reklaitis, 2004; Papageorgiou, 2009; Shah, 2005; Varma, Rekalitis, Blau, & Pekny, 2007), but very few contributions have been made in this field.

Particularly, the SC design and planning, SC planning and scheduling, and SC redesign and planning were traditionally treated as isolated areas (see Goetschalckx et al., 2002; Maravelias & Sung, 2009; Rungtusanatham & Forza, 2005). However, in the recent past, attempts have been made to deal with them in a simultaneous fashion. Following this approach, Lee, Lee, and Reklaitis (2000) studied the capacity expansion problem of multisite batch plants. Sundaramoorthy and Karimi (2004) presented an approach for new product introduction and planning in pharmaceutical supply chains. Guillén, Badell, Espuña, and Puigjaner (2006) integrated planning and scheduling decisions of chemical SCs taking into account financial management issues. The integration of financial considerations into SCM was also addressed by Longinidis and Georgiadis (2013) and Susarla and Karimi (2012). Amaro and Barbosa-Póvoa (2008) presented a modelling approach for the sequential planning and scheduling of SCs, while Cóccola, Zamarripa, Mendez, and Espuña (2013) addressed the integration of production and distribution tasks in multi-echelon supply chains.

As for the integration of spatial decisions in SCM, to the best of our knowledge, only a few papers have dealt with the integrated design of SCs along with the involved plants. The design of multi-product plants can be posed as the problem of selecting the processing units and their sizes so as to minimize a selected design

* Corresponding author.
E-mail address: gonzalo.guillen@urv.cat (G. Guillén-Gosálbez).

0098-1354/$ – see front matter © 2013 Published by Elsevier Ltd.
http://dx.doi.org/10.1016/j.compchemeng.2013.09.001

Nomenclature

Sets

I	set of products
J_l	set of stages of batch plant l
K	set of customer zones
L	set of production plants
M	set of warehouses
R	Set of raw materials
STF_{jl}	set of discrete sizes for units of stage j of plant l
SV_{jl}	set of discrete sizes for units of stage j of plant l

Indices

d	number of in phase units
g	discrete size for storage tank
G_{jl}	number of available discrete sizes for a tank of stage j of plant l
i	product
j	Stage
k	customer zone
l	Plant
m	Warehouse
n	number of out of phase units
p	discrete size for batch unit
P_{jl}	number of available discrete sizes for a unit of stage j of plant l

Parameters

C_{ann}	capital charge factor
Cd_{im}	warehouse maintenance cost coefficient
$Cdep_m$	warehouse installation cost coefficient
Cpl_l	plant installation cost coefficient
$Craw_{sr}$	raw material cost coefficient
$Cprod_{il}$	production cost coefficient
Ctd_{imk}	transportation cost
Ctp_{ilm}	transportation cost
$Ctraw_{srl}$	transportation cost
fp_{ril}	raw material conversion factor
H_l	time horizon of plant l
NB_{ijl}^{UP}	maximum number of batches of product i
NP_{jl}^{UP}	maximum number of in phase unit for stage j of plant l
NT_{jl}^{UP}	maximum number of out of phase unit for stage j of plant l
Q_{il}^{LO}	lower bound for product i in plant l
Q_{il}^{UP}	upper bound for product i in plant l
Q_m^{UP}	maximum warehouse capacity
Q_{sr}^{UP}	upper bound for raw material r at site s
SF_{ijl}	size factor of product i in stage j of plant l
ST_{ijl}	size factor of product i in tank of stage j of plant l
t_{ijl}	processing times for each product i in stage j of plant l
VF_{jpl}	discrete size for batch units in stage j of plant l
VTF_{jlg}	discrete size for storage tank in stage j of plant l
α_{jl}	cost coefficient for units of stage j of plant l
$\tilde{\alpha}_{jl}$	cost coefficient for tank of stage j of plant l
β_{jl}	cost exponent for units of stage j of plant l
$\tilde{\beta}_{jl}$	cost exponent for tank of stage j of plant l
ϕ	maximum ratio allowed between the number of batches of consecutive stages

Binary variables

ex_l	binary variable for plant allocation
v_{jpl}	binary variable that denotes if the units of stage j have size p
vt_{jpl}	binary variable that denotes if the units of stage j have size p
x_{jln}	binary variable for the number of out of phase parallel units of stage j of plant l
xx_{jld}	binary variable for the number of in phase parallel units of stage j of plant l
y_m	binary variable for warehouse allocation
z_{il}	binary variable for product selection in plant l

Continuous variables

e_{ijlpd}	variable that takes value Q_{il} if $v_{jlp} = 1$ and $xx_{jld} = 1$
f_{ijlg}	variable that takes value Q_{il} if $vt_{jlg} = 1$ and $xx_{jld} = 1$
Nb_{ijl}	number of batches of product i in stage j of plant l
NP_{jl}	number of units in phase in stage j of plant l
NT_{jl}	number of units out of phase in stage j of plant l
w_{ijln}	variable that represents the bilinear term $Nb_{ijl}\, x_{jln}$
Q_{il}	amount of product i in plant l
Q_{ilm}	amount of product i transported from plant l to warehouse m
Q_{imk}	amount of product i transported from warehouse m to customer zone k
Q_{sril}	amount of raw material r transported from site s to plant l for producing product i
T_{il}	total time for producing product i in plant l
V_j	continuous variable that denotes the size of a batch unit in stage j
λ_{ijgl}	variable that represents the bilinear term $vt_{jlg}\, Nb_{ijl}$
ρ_{jlpnd}	variable that represents the term $v_{jlp}\, x_{jln}\, xx_{jld}$

objective, while satisfying the minimum production requirements over a defined time of planning. The design of batch processes has been an active area of research in the PSE community over the last decades (see the review by Barbosa-Póvoa, 2007 for further details and references). In contrast, the incorporation of batch plant design into SCM has received little attention.

Corsano, Montagna, Iribarren, and Aguirre (2007) presented a detailed non linear programming (NLP) model for the design of a multisite plant complex, considering the integration between plants simultaneously with their optimal operation and production planning. Corsano, Vecchietti, and Montagna (2011) presented an MINLP optimization model for a sustainable design and performance analysis of sugar/ethanol SCs. A detailed model for ethanol plant design was embedded in the SC model in order to obtain the plant and SC designs simultaneously. In a more general work, Corsano and Montagna (2011) presented an MILP model for the simultaneous optimization of SCs and their batch plants, showing that the simultaneous optimization outperforms hierarchical approaches.

1.2. Solution approaches in SCM

Unfortunately, the integration of decisions leads to large-scale MILPs that are hard to solve, and whose complexity increases with the number of production plants, batch stages and/or final products. Therefore, it becomes evident that the integration of decision levels poses a major computational challenge.

The preferred modelling tool in SCM has been MILP. The motivation for this choice is that these formulations tend to be represented at a high level, and hence apply fairly simple representations of capacity that avoid nonlinearities and permit an easy adaptation to a wide range of industrial scenarios. These MILPs have been traditionally solved via branch and bound techniques, which in some cases have been applied in conjunction with other strategies such as Lagrangean (Graves, 1982), Benders (Spengler, Püchert, Penkuhn, & Rentz, 1997) and bi-level decomposition methods (Iyer & Grossmann, 1998).

The integration of decision making levels in SCM further increases the complexity of the modelling approach, as additional variables and constraints need to be defined to represent decisions of different nature. Hence, one of the main challenges of integrating decisions in SCM is the efficient solution of large-scale problems, which typically requires the use of decomposition strategies.

In this line, van den Heever, Grossmann, and Vasantharajan (2001) proposed a specialized heuristic algorithm based on the concept of Lagrangean decomposition for the long-term design and planning of offshore hydrocarbon field infrastructures with complex economic objectives. Jackson and Grossmann (2003) presented a multisite production planning and distribution formulation that included nonlinear process models to represent the production facilities that were solved with Lagrangean decomposition. Chen and Pinto (2008) developed several Lagrangean-based decomposition techniques for a continuous flexible process network model, including Lagrangean decomposition, Lagrangean relaxation, and Lagrangean/surrogate relaxation. You and Grossmann (2008) proposed a decomposition algorithm based on Lagrangean relaxation for solving the integrated stochastic inventory management and supply chain network design. Later, You, Grossmann, and Wassick (2011) developed a bi-level algorithm and a Lagrangean decomposition method to address the simultaneous capacity, production, and distribution planning of a multisite system.

Bi-level decomposition has also been used for solving SC design and planning problems. Iyer and Grossmann (1998) solved a MILP model for determining the optimal selection and expansion of processes over a long-range planning horizon using a rigorous bi-level decomposition algorithm, while Guillén-Gosálbez, Mele, and Grossmann (2010) applied this technique to the design of hydrogen supply chains for vehicle.

Despite these algorithmic developments, the use of decomposition strategies in the integrated design of SCs along with their embedded facilities has been quite scarce. Because of this, current full space approaches can only tackle problems that consider only a limited number of plants, depots and clients. In this work we fill this research gap by presenting several customized decomposition algorithms for this problem. The MILP model for the design of SCs presented in Corsano and Montagna (2011) that includes equations to model the performance of the batch plants of the network is taken as a basis to develop our algorithmic framework. Such a spatial integration of decision-making levels leads to a complex formulation that is hard to solve in reasonable computational time. This MILP becomes even more complex as the number of plants increases, mainly because of the presence of complicating constraints that are required to model the plant performance precisely. Hence, the main contribution of this work is the development of three tailored algorithms inspired on bi-level and Lagrangean decomposition schemes that exploit the problem structure, making it possible to tackle large-scale problems encountered in practice. Numerical results show that our approaches outperform standard branch and cut codes applied to the full space MILP.

The article is organized as follows. We introduce the problem of interest in the next section. We then present an MILP model that integrates decisions at different hierarchical levels. In Section 4, we introduce three decomposition strategies for the efficient solution of this MILP. In Section 5 the capabilities of the proposed methodologies are illustrated through some examples. Finally, the conclusions of the work are drawn in the last section of the paper.

2. Problem statement

In this work, we consider a generic SC like the one shown in Fig. 1.

Based on this representation, we formally state next the problem of interest. Given is a set of raw materials sites $s \in S$. Each raw material site s, has one or more types of raw materials r, $r \in R$, with limited capacity Q_{sr}^{UP}, to be delivered to plants $l \in L$, which operate over a time horizon H_l. Each multiproduct plant has a set of batch stages $j \in J_l$, for producing a set of products $i \in I$.

For each multiproduct batch plant, we consider in phase and out of phase unit duplication. The allocation of intermediate storage tanks between two batch stages is also considered. These can be allocated in $|J_l|-1$ positions in plant l, where position j is defined between batch stages j and $j+1$. A zero-wait (ZW) transfer policy is adopted between consecutive batch stages.

According to the usual unit procurement policy, a set SV_{jl} of P_{jl} discrete unit sizes, $SV_{jl} = \{VF_{jl1}, VF_{jl2}, \ldots, VF_{jlP_{jl}}\}$, is available for stage j in plant l. Similarly, a set of G_{jl} discrete sizes for storage tanks $STF_{jl} = \{VTF_{jl1}, VTF_{jl2}, \ldots, VTF_{jlG_{jl}}\}$ is available for position j in plant l.

Final products are delivered from plant l to several warehouses $m \in M$, each of them with a limited capacity Q_m^{UP}. Products are then transported from warehouse m to different customer zones $k \in K$, in order to satisfy a known product demand D_{ik}. Assuming that cost parameters associated to plants and warehouses installation, investment, production, distribution, raw materials and operation are known, the problem consists of determining simultaneously:

- The SC topology (nodes allocation).
- The SC planning (production rates and flows among nodes).
- The multiproduct batch plan design (plant configuration and unit sizes)

in order to fulfil the product demands with minimum cost. The cost function considers installation, investment, production, operation, and transportation costs.

3. Mathematical formulation

For the sake of completeness of this work, we describe in Appendix A section the MILP formulation presented by Corsano and Montagna (2011), which will be taken as a basis for the development of the decomposition strategies presented herein. For further details, see Corsano and Montagna (2011).

The overall MILP model for the strategic planning of SC and multiproduct batch plants design, namely (FP), can be expressed in compact form as follows:

$$\begin{aligned} &\min C_{total} = C_{inv} + C_{inst} + C_{oper} + C_{trans} \\ &\text{s.t. constraints} \quad (A1.1)-(A1.38) \end{aligned} \qquad (FP)$$

4. Solution methods

As previously mentioned, despite producing more accurate results, the simultaneous strategic planning of SCs and batch chemical manufacturing sites leads to complex problems. This application thus calls for decomposition strategies capable of reducing the computational burden while still providing near optimal solutions. Particularly, in this work we present three algorithms that are inspired on bi-level and Lagrangean decomposition

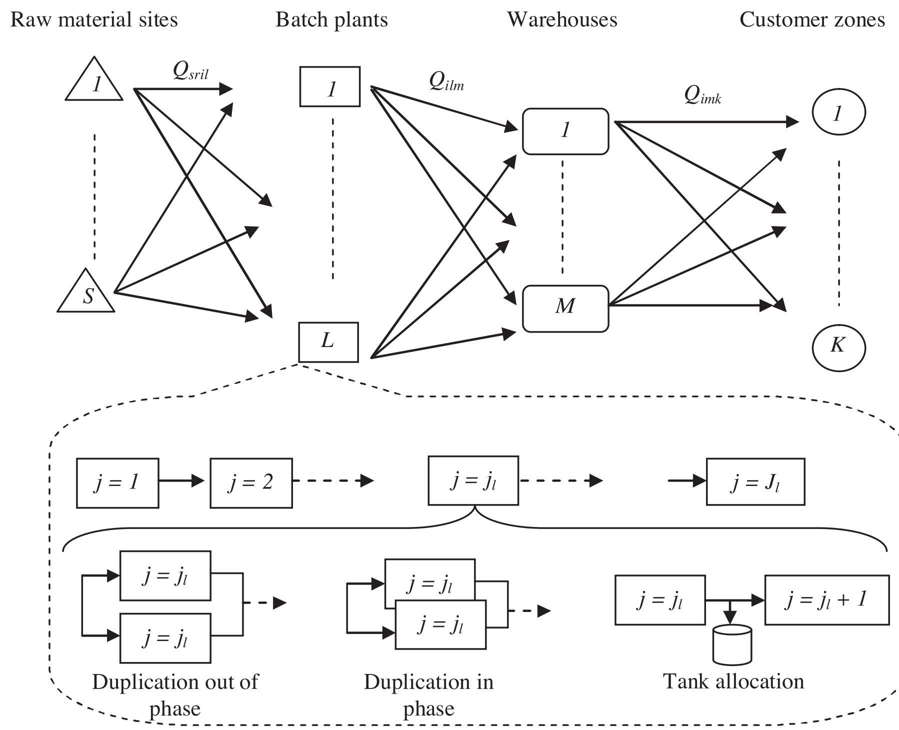

Fig. 1. Integrated SC and multiproduct batch plant design.

schemes. The ensuing sections provide details on each of them.

4.1. Bi-level decomposition

Bi-level algorithms decompose the original full space model into two subproblems at different hierarchical levels between which they iterate until a termination criterion is satisfied. Our proposed bi-level algorithm solves a lower bounding master problem (LBP), which is a relaxation of the full space problem (FP), to obtain a lower bound on the cost. This relaxation is constructed by dropping the integrality requirement of some binary variables that model the performance of the manufacturing sites in the full space model. The master problem provides as output a SC configuration that is optimal in the relaxed search space, but not necessarily in the full space model. Hence, the configuration obtained from (LBP) is fixed in the upper bounding problem (UBP), to obtain an upper bound on the total cost of the network and determine at the same time the values of those variables that were relaxed in (LBP).

Problems (LBP) and (UBP) are then solved iteratively until a termination criterion is reached. In every iteration a new integer cut is added to (LBP) in order to exclude from the search space SC configurations already explored in previous iterations. As iterations proceed, the difference between the best lower and upper bounds (i.e., optimality gap) decreases. Two termination criteria that tend to work well in practice are to stop either when the difference between the lower and upper bounds falls below a desired tolerance or when a maximum number of iterations is reached. We provide next details on how the (LBP) and (UBP) problems are constructed.

In the (LBP) problem, the binary variables that represent product selection in each plant, z_{il}, as well as those denoting the batch unit (v_{jlp}) and storage tank (vt_{jlg}) sizes are relaxed (treated as continuous variables rather than binaries). The rest of the variables and constraints remain unmodified in this problem. Because of these modifications, problem (LBP) is less complex in combinatorial terms than the original model (FP). More precisely, the number of binary variables in (LBP) is equal to $|L| + |M| + |J||L|NT_{jl}^{UP} + |J||L|NP_{jl}^{UP}$ whereas in (FP) is $|J||L|NT_{jl}^{UP} + |J||L|NP_{jl}^{UP} + |I||J| + |J||G||L| + |J||L||P|$.

Model (LBP) provides as output the subset of installed manufacturing plants and warehouses. Furthermore, since (LBP) is a relaxation of (FP) (i.e., its search space contains the search space of (FP)), it provides a rigorous valid lower bound on the global optimum of (FP). The aforementioned decision variables are then fixed in (UBP), whose solution provides a valid upper bound on the global optimum of (FP). Problem (UBP) corresponds to the original MILP model (FP), which is solved for only a subset of production plants and warehouses predicted by (LBP), thereby yielding a valid upper bound on the global optimum to (FP). The main advantage of (UBP) is that many tradeoffs, arising from the simultaneous optimization of conflicting variables, are avoided when the SC topology is fixed. Therefore, the problem complexity is greatly reduced and the computational expense is lowered.

Finally, (LBP) and (UBP) are solved iteratively until the bounds of each level converge within a specified tolerance. Model (LBP) is updated at each iteration by adding integer cuts (Balas & Jeroslow, 1972) in order to exclude solutions explored by the algorithm in previous iterations. These cuts are expressed by the following constraints (see Iyer & Grossmann, 1998, for further details):

$$\sum_{l\in WL_1^r} ex_l + \sum_{m\in WM_1^r} y_m - \sum_{l\in WL_0^r} ex_l - \sum_{m\in WM_0^r} y_m \leq \left|WL_1^{iter}\right| + \left|WM_1^{iter}\right| - 1 \quad (1)$$

where $WL_1^{iter} = \{l : ex_l = 1$ in the optimal solution of LBP at iteration $iter\}$, $WM_1^{iter} = \{m : y_m = 1$ in the optimal solution of LBP at iteration $iter\}$, $WL_0^{iter} = \{l : ex_l = 0$ in the optimal solution of LBP at iteration $iter\}$, $WM_0^{iter} = \{m : y_m = 0$ in the optimal solution of LBP at iteration $iter\}$.

In summary, the steps of the bi-level decomposition algorithm are the following (see Fig. 2):

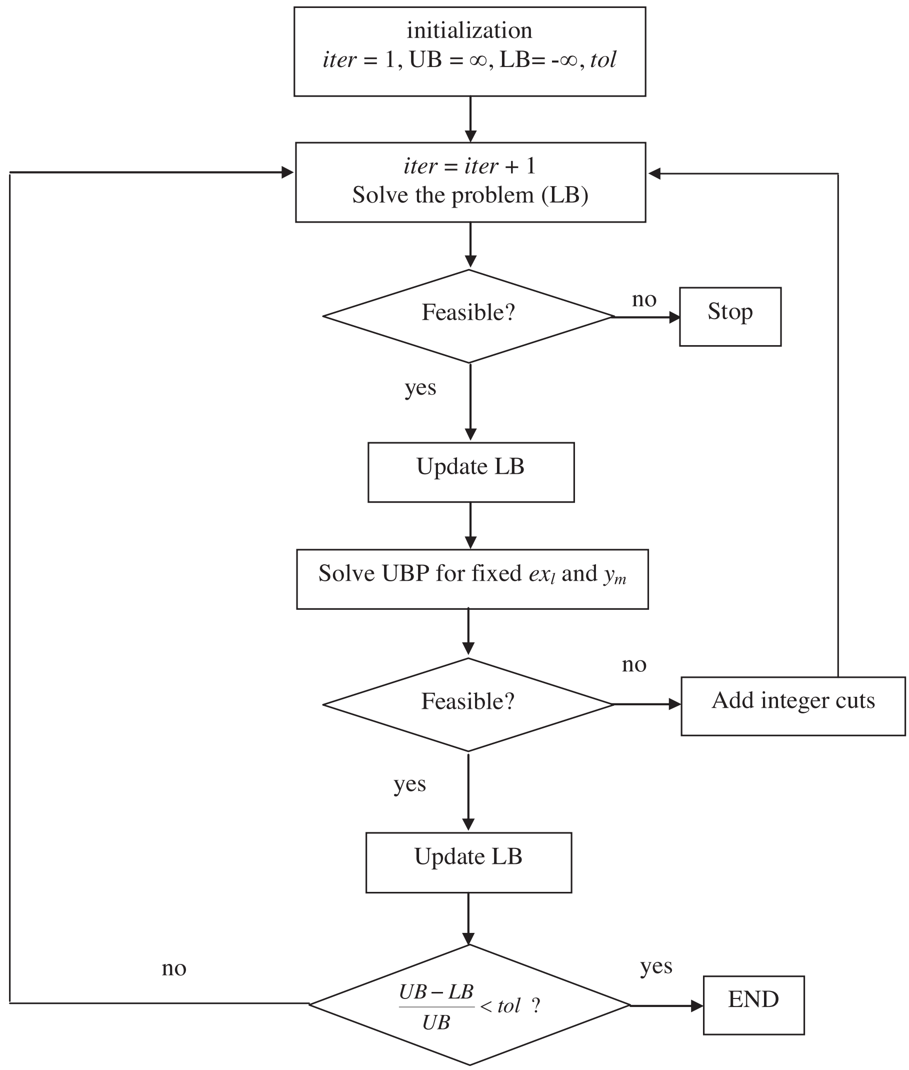

Fig. 2. Bi-level decomposition algorithm flowchart.

Step 1: Set iteration count $iter = 1$, lower bond $LB = -\infty$, and upper bound $UB = +\infty$, and tolerance error equal to *tol*.

Step 2: Set $iter = iter + 1$. Solve the MILP lower bounding master problem (LBP):

- If problem (LBP) is infeasible, then stop, there is no feasible solution to (FP).
- Otherwise, set the current lower bound $LB = C_{total}^{iter}$, and define
  $WL_1^{iter} = \{l : ex_l = 1$ in the optimal solution of LBP at iteration $iter\}$
  $WM_1^{iter} = \{m : y_m = 1$ in the optimal solution of LBP at iteration $iter\}$
  $WL_0^{iter} = \{l : ex_l = 0$ in the optimal solution of LBP at iteration $iter\}$
  $WM_0^{iter} = \{m : y_m = 0$ in the optimal solution of LBP at iteration $iter\}$
  where C_{total}^{iter} is the objective function value associated with the optimal solution of (LBP) in iteration *iter*.

Step 3: For fixed ex_l^{iter} and y_m^{iter}, solve problem (UBP):

- If problem (UBP) is infeasible, then add the following integer cut:
  $$\sum_{l\in WL_1^r} ex_l + \sum_{m\in WM_1^r} y_m - \sum_{l\in WL_0^r} ex_l - \sum_{m\in WM_0^r} y_m \le |WL_1^{iter}| + |WM_1^{iter}| - 1$$
- Otherwise, update the upper bound $UB = \min_{iter}\{UB^{iter}\}$ where UB^{iter} represents the objective function value associated with the optimal solution of problem (UBP) in iteration *iter*.

Step 4: check the convergence criteria:

- If $(UB - LB)/(UB) < tol$, then stop. The solution corresponding to *UB* (i.e., the solution of model (UBP) in the iteration with minimum cost) satisfies the termination criterion (i.e., it can be regarded as optimal within the predefined optimality gap).
- Otherwise, go to *step 2*.

An important remark is that the bi-level method is guaranteed to identify the global optimum for a sufficiently large number of iterations. This is because in every iteration we exclude one feasible solution from the master problem. Since all the solutions of the original problem are also feasible in the lower bounding problem, we would end up evaluating all of them during the execution of the algorithm if no termination criterion was used.

4.2. Lagrangean decomposition

This method relies on constructing a relaxation of (FP) (i.e., Lagrangean dual problem), obtained by dualizing (dropping) some "complicating" constraints of the model. There are different ways to construct the relaxed problem according to the type of constraints that are dropped. In the context of SCM, the two main approaches are spatial and temporal decomposition.

Particularly, as can be noted in the mathematical formulation in the Appendix A, Eqs. (A.1)–(A.5) and (A.11)–(A.35) are defined for production plants, while Eqs. (A.6)–(A.10) are expressed for warehouses. Therefore, the problem can be spatially separated in order to solve reduced models for the plants and warehouses embedded in the network separately. Hence, we propose to spatially decompose the SC nodes and dualize the interconnecting constraints that link production plants and warehouses through the mass balance equations, so they can be finally optimized individually.

To this end, the continuous variable that represents the flows between plants and warehouses is duplicated, so that one copy $\hat{Q}_{ilm}$ is defined for warehouse constraints, while the original Q_{ilm} variable is used for plants constraints. Hence, we add the following new constraints to the model:

$$Q_{ilm} = \hat{Q}_{ilm} \quad i \in I, \quad l \in L, \quad m \in M \tag{2}$$

Therefore, the affected constraints are rewritten in terms of the proper interconnection variables. Eq. (A.7) is expressed as:

$$\sum_{\substack{l \in L \\ i \in I}} \hat{Q}_{ilm} \le Q_m^{UP} y_m \quad m \in M \tag{3}$$

while Eq. (A.8) is rewritten as shown Eq. (4).

$$\sum_{l \in L} \hat{Q}_{ilm} = \sum_{k \in K} Q_{imk} \quad i \in I, \quad m \in M \tag{4}$$

We then construct the Lagrangean dual problem by dualizing Eq. (2), which is transferred to the objective function and multiplied by Lagrangean multipliers λ_{ilm}. Therefore the Lagrangean dual problem (LDP) takes the following form:

discussed later in the article, the resulting submodel (P1) tends to have a large size, which makes the Lagrangean decomposition less efficient than the bi-level algorithm.

After solving the Lagrangean dual for some values of the Lagrangean multipliers (note that this problem is decomposed into (P1) and (P2)), we obtain a lower bound on the total cost. We then need to generate an upper bound and a feasible solution to the original problem. To this end, we fix some variables of the original model according to the output of the dual model and then solve it in a reduced domain so as to yield a valid upper bound.

After obtaining an upper bound, we can generate new values for the Lagrangean multipliers and repeat the overall procedure until a termination criterion is satisfied. In our case, the multipliers are updated using the following equation:

$$\lambda_{ilm}^{iter+1} = \lambda_{ilm}^{iter} + Step^{iter}(\hat{Q}_{ilm}^{iter} - Q_{ilm}^{iter}) \quad \forall i \in I, \quad l \in L, \quad m \in M \tag{5}$$

where $Step^{iter}$ represents the step size and is calculated from the upper (UB) and lower (LB) bounds, the difference between the interconnection variables (Q_{ilm}^{iter} and $\hat{Q}_{ilm}^{iter}$), and a weigh parameter μ, as follows:

$$Step^{iter} = \frac{\mu(UB - LB)}{\sum_{i,l,m} (\hat{Q}_{ilm}^{iter} - Q_{ilm}^{iter})^2} \tag{6}$$

Note that because the model has binary variables, there will be always a duality gap (i.e., difference between the lower and upper

$$\begin{aligned}
\min C_{Lagrange} = {} & C_{ann}\left(\sum_{l \in L}\sum_{j \in J_l}\sum_{p=1}^{P_{jl}}\sum_{n=1}^{NT_{jl}^{UP}}\sum_{d=1}^{NP_{jl}^{UP}} \alpha_{jl} nd VF_{jlp}^{\beta_{jl}} \rho_{jlpnd} + \sum_{l \in L}\sum_{j \in J_l}\sum_{g=1}^{G_{jl}} \tilde{\alpha}_{jl} VTF_{jlg}^{\tilde{\beta}_{jl}} vt_{jlg}\right) + C_{ann}\left(\sum_{l \in L} Cpl_l ex_l + \sum_{m \in M} Cdep_m y_m\right) \\
& + \sum_{s \in S}\sum_{r \in R}\sum_{i \in I}\sum_{l \in L} Craw_{sr} Q_{sril} + \sum_{i \in I}\sum_{l \in L}\sum_{m \in M} Cd_{im}\hat{Q}_{ilm} + \sum_{i \in I}\sum_{l \in L} Cprod_{il} Q_{il} + \sum_{s \in S}\sum_{r \in R}\sum_{i \in I}\sum_{l \in L} Ctraw_{srl} Q_{sril} \\
& + \sum_{i \in I}\sum_{l \in L}\sum_{m \in M} Ctp_{ilm}\hat{Q}_{ilm} + \sum_{i \in I}\sum_{m \in M}\sum_{k \in K} Ctd_{imk} Q_{imk} + \sum_{i \in I}\sum_{l \in L}\sum_{m \in M} \lambda_{ilm}(\hat{Q}_{ilm} - Q_{ilm})
\end{aligned}$$

Subject to constraints (A.1)–(A.6), (A.9)–(A.34) and (3) and (4).

When the multipliers are fixed, model (LDP) is separable in the set of plants and warehouses. Note that the resulting submodels (P1) and (P2) are easier to solve, as they involve fewer variables and constraints:

$$\begin{aligned}
\min C_{P1} = {} & C_{ann}\left(\sum_{l \in L}\sum_{j \in J_l}\sum_{p=1}^{P_{jl}}\sum_{n=1}^{NT_{jl}^{UP}}\sum_{d=1}^{NP_{jl}^{UP}} \alpha_{jl} nd VF_{jlp}^{\beta_{jl}} \rho_{jlpnd} + \sum_{l \in L}\sum_{j \in J_l}\sum_{g=1}^{G_{jl}} \tilde{\alpha}_{jl} VTF_{jlg}^{\tilde{\beta}_{jl}} vt_{jlg}\right) + C_{ann}\sum_{l \in L} Cpl_l ex_l + \sum_{s \in S}\sum_{r \in R}\sum_{i \in I}\sum_{l \in L} Craw_{sr} Q_{sril} \\
& + \sum_{i \in I}\sum_{l \in L} Cprod_{il} Q_{il} + \sum_{s \in S}\sum_{r \in R}\sum_{i \in I}\sum_{l \in L} Ctraw_{srl} Q_{sril} - \sum_{i \in I}\sum_{l \in L}\sum_{m \in M} \lambda_{ilm} Q_{ilm}
\end{aligned} \tag{P1}$$

subject to Eqs. (A.1)–(A.6) and (A.11)–(A.34).

$$\min C_{P2} = C_{ann}\sum_{m \in M} Cdep_m y_m + \sum_{i \in I}\sum_{l \in L}\sum_{m \in M} Cd_{im}\hat{Q}_{ilm} + \sum_{i \in I}\sum_{l \in L}\sum_{m \in M} Ctp_{ilm}\hat{Q}_{ilm} + \sum_{i \in I}\sum_{m \in M}\sum_{k \in K} Ctd_{imk} Q_{imk} + \sum_{i \in I}\sum_{l \in L}\sum_{m \in M} \lambda_{ilm}\hat{Q}_{ilm} \tag{P2}$$

Subject to Eqs. (A.9) and (A.10) and (3) and (4).

Note that as oppose to other Lagrangean decomposition schemes applied to SCM that define one subproblem for each production plant (Jackson & Grossmann, 2003; You et al., 2011), we consider herein all the plants of the network together. The motivation for this is that there is a total demand that must be fulfilled and the allocation of orders to plants must be decided. Hence, if plants were approached separately, there would be the risk that the models for each production plant would render infeasible. It is therefore not convenient to treat the plants separately. As will be

bounds) and the overall procedure is not guaranteed to converge to the global optimum of the original problem.

The overall Lagrangean strategy comprises the following steps (see Fig. 3):

Step 1. Set initial multipliers λ_{ilm} = 0, upper bound UB = +∞, lower bound LB = −∞, $iter$ = 0, μ = constant (chosen between 0 and 2, often equal to 2), and tolerance error equal to tol (e.g., 10^{-2}).

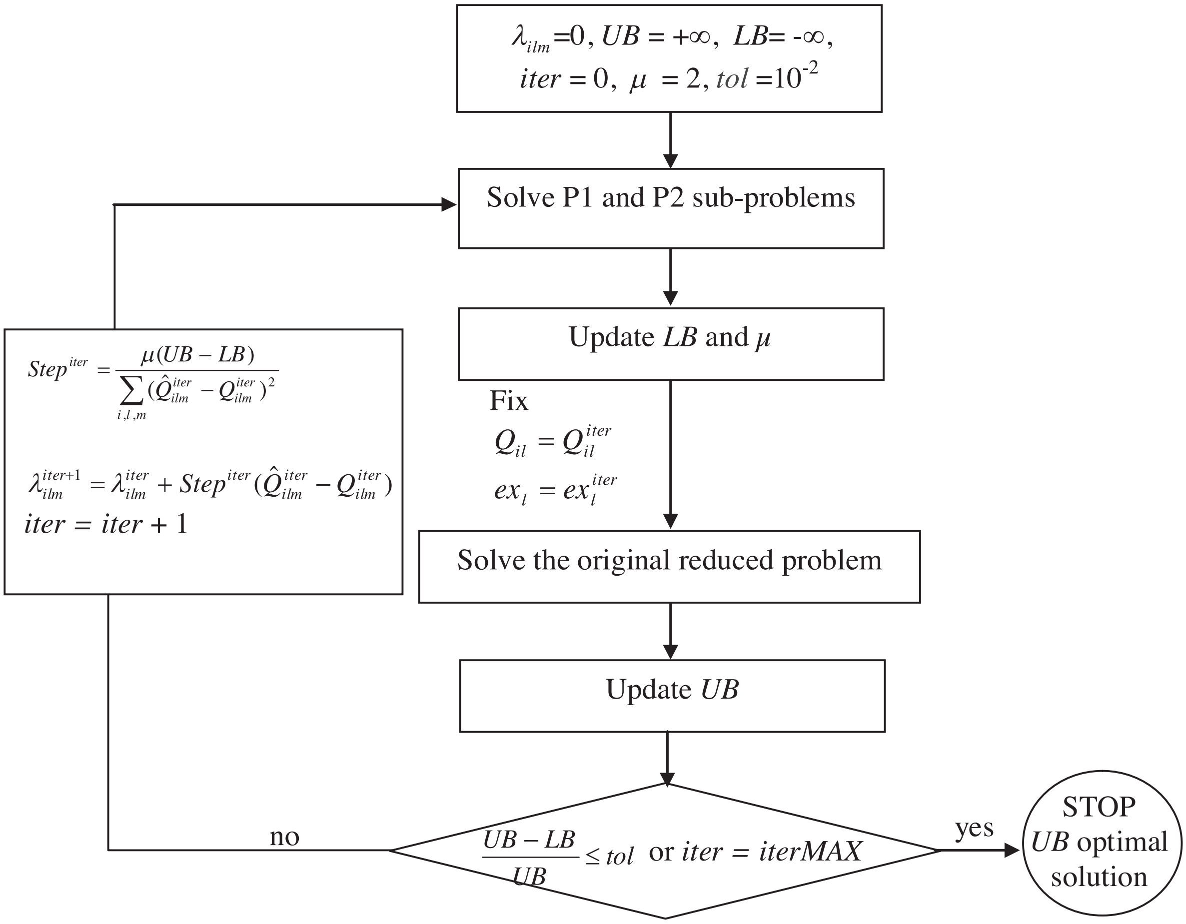

Fig. 3. Lagrangean decomposition algorithm flowchart.

Step 2. Solve models (P1) and (P2) for fixed λ_{ilm}. Let the optimal objective function of each problem be denoted by C_{P1}^{iter} and C_{P2}^{iter}, respectively. Store the optimal value of Q_{il} and ex_l (represented by Q_{il}^{iter} and ex_l^{iter}, respectively). Calculate $C_{Lagrange}^{iter}=C_{P1}^{iter}+C_{P2}^{iter}$.

If $C_{Lagrange}^{iter}>LB$, then update the lower bound, $LB=C_{Lagrange}^{iter}$. If no reduction of LB is performed after two iterations of the subgradient procedure, then halve the step length parameter by setting $\mu=\mu/2$.

Step 3. Fix binary variable ex_l and the production variables Q_{il} to the values obtained on (P1), ex_l^{iter} and Q_{il}^{iter}, respectively, and then solve the original problem (FP). If its optimal value C_{total}^{iter} is lower than the actual upper bound (i.e., $C_{total}^{iter}<UB$), then update the upper bound $UB=C_{total}^{iter}$, and store the current solution. Note that because of the way in which the upper bound is obtained, (FP) is always guaranteed to be feasible for any potential value of the decision variables generated by (P1).

Step 4. If $(UB-LB)/UB\le tol$ or the number of iterations has been reached, set UB as the optimal cost and the current stored solution as the optimal solution. Else, go to Step 5.

Step 5. Calculate the step size (Fisher, 1981), and update the multipliers (Fisher, 1981; Guignard & Kim, 1987):

$$Step^{iter}=\frac{\mu(UB-LB)}{\sum_{i,l,m}(\hat{Q}_{ilm}^{iter}-Q_{ilm}^{iter})^2}$$

$$\lambda_{ilm}^{iter+1}=\lambda_{ilm}^{iter}+Step^{iter}(\hat{Q}_{ilm}^{iter}-Q_{ilm}^{iter})\quad \forall i\in I,\quad l\in L,\quad m\in M$$

Assign $iter=iter+1$, and go to *Step 2*.

4.3. Hybrid strategic

A third strategy is proposed that combines the basic ideas of both Lagrangean and bi-level decomposition algorithms. The number of binary variables of (P1) solved at each iteration of the Lagrangean decomposition is similar to the number of binary variables of (FP), and so is the computational burden. To expedite the solution of (P1), we propose the following modifications:

- Subproblem (P1) is solved relaxing some of its binary variables, namely those that model the size of the units and tanks. This leads to lower computational burdens.
- The following integer cuts are added to (P1) in each iteration of the algorithm in order to avoid repeated solutions:

$$\sum_{l\in WL_1^r}ex_l-\sum_{l\in WL_0^r}ex_l\le\left|WL_1^{iter}\right|-1$$

where $WL_1^{iter}=\{l:ex_l=1$ in the optimal solution of $(P1)$ at iteration $iter\}$, $WL_0^{iter}=\{l:ex_l=0$ in the optimal solution of $(P1)$ at iteration $iter\}$

Reducing the number of binary variables in subproblem (P1), lowers in turn the computational burden. The production plants selected by (P1) are then fixed in the reduced model (FP). In order to avoid solutions explored in previous iterations, we add integer cuts to (P1), so it is guaranteed that the set of installed plants to be fixed in the reduced (FP) will be different at each iteration. The details of the algorithm are shown in Fig. 4.

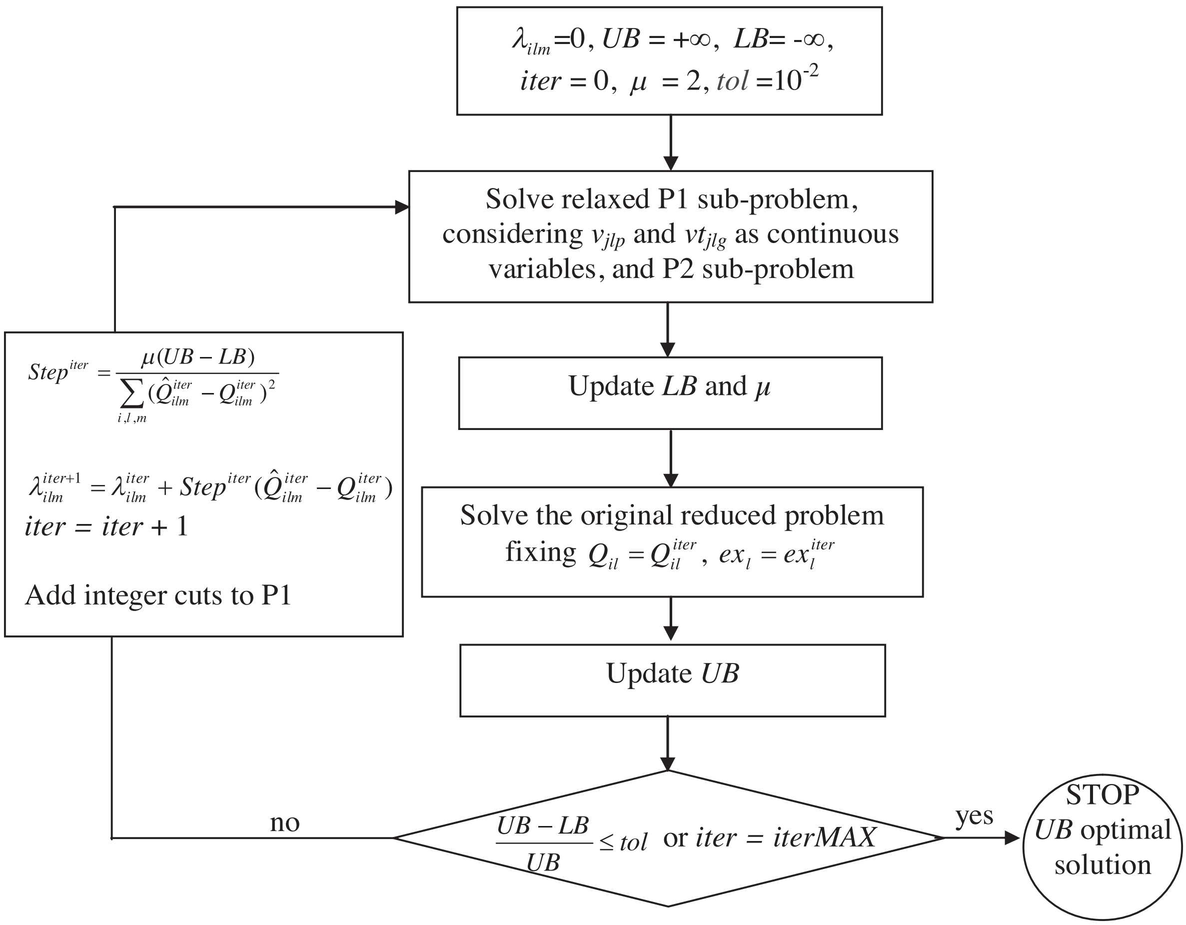

Fig. 4. Hybrid decomposition strategic flowchart.

It is worth to mention that model (P1) solved by the hybrid method is a relaxation of problem (P1) used in the Lagrangean decomposition. Therefore, the lower bound obtained in the former method is guaranteed to be lower than that generated by the Lagrangean decomposition. As a result, the Lagrangean decomposition provides a better relative gap (difference between the upper and lower bounds) than the hybrid strategy. In summary, the hybrid strategy leads to less iterations, while the Lagrangean algorithm produces lower relative gaps.

5. Numerical results

We solved six numerical examples to evaluate the performance of the proposed decomposition algorithms. The first and second examples are taken from Corsano and Montagna (2011). Examples 3, 4 and 5 have been generated from the first one by increasing the number of facilities, products, warehouses and customer zones. The aim here is to test the model behaviour for larger problem sizes. Finally, to construct the sixth example we have modified some model parameters values from the third example in order to show how the problem data affect the computational burden due to the existence of inherent tradeoffs.

All the examples were implemented and solved in GAMS (Rosenthal, 2008) on an Intel Core i5, 2.3 GHz. The CPLEX 12.1.0 solver was employed for solving the MILP problems. The number of continuous and binary variables and constraints strongly depends on the number of plants to be installed, the number of products to be produced, and the number of discrete options considered for the batch units and storage tanks sizes. Note that the computational complexity of the problem, and consequently the computational burden, grows with the number of binary variables. Moreover, due to some trade-offs involved in the decision-making problem, the model performance varies according to the problem data, as will be discussed in Example 6.

In all of the examples, the maximum number of iterations for the decomposition algorithms was set to 20, and the maximum time limit to 14,400 CPU s (4 h). In addition, the tolerance error (difference between bounds) for the bi-level, Lagrangean and hybrid algorithms was set to 1%. The reason for this is that the optimality gap might not be closed in these algorithms (even when running an infinite number of iterations). This is because the lower bounding problems are relaxations of the original model, and therefore might yield optimal values that differ from the global optimum of the original model. Note that these algorithms might identify the global optimum even if the optimality gap is fixed to 1% (the global optimal solution is identified, but global optimality cannot be proved). Hence, for a fair comparison, an optimality gap of 1% was also defined for CPLEX when solving the full space model (FP). Table 1 resumes the number of raw material sites, raw materials, production plants, products, warehouses, and customer zones considered in each example.

5.1. Example 1

This example is taken from Corsano and Montagna (2011). We consider a SC with 2 raw material sites that provide 3 different raw materials to 5 production plants. These plants produce 4 products through 3 batch stages. For each batch stage, we consider a set of 5 discrete sizes (300 l, 500 l, 750 l, 1000 l, and 1200 l), and the option of duplication of up to 2 units in phase or out of phase is

Table 1
Summary of elements considered in each example.

	# raw material sites	# raw materials	# production plants	# products	# warehouses	# customer zones
Example 1	2	3	5	4	3	3
Example 2	6	3	5	4	5	45
Example 3	2	3	10	4	10	3
Example 4	2	3	10	6	10	10
Example 5	2	3	15	6	10	10
Example 6	2	3	10	4	10	10

Table 2
Detailed list of costs for the optimal solution of Example 1.

Costs	For all proposed approaches
Investment	466,400
Production	116,300
Raw material	44,800
Warehouse	117,000
Distribution	275,000
Total ($/year)	1,019,500

allowed. Also, 3 different tank sizes are available: 3000 l, 5000 l, and 10,000 l. There are 3 types of warehouses available with a capacity equal to 1,050,000 kg, 1,250,000 kg, and 1,100,000 kg respectively, and 3 customer zones. All the model parameters are presented in Appendix B.

The full space problem (FP) involves 223 binary variables, 2082 continuous variables, and 2144 constraints. It takes 79.8 CPU seconds to compute the optimal solution with a 0% optimality gap. The minimum cost is equal to $1,019,543, and the optimal SC design involves the installation of one batch production plant. Table 2 shows a breakdown of the total cost for the global optimal solution, which is identified by the full space method (i.e., CPLEX) and also by the bi-level, Lagrangean and hybrid decomposition strategies. The optimal SC configuration and plant design are shown in Fig. 5. In this instance, the plant investment cost has a large contribution in the objective function, so the solution tries to minimize the use of units (and plants). From the logistic standpoint, the nodes are selected taking into account their capacity and the associated transportation costs. Hence, the optimal solution chooses site S2 to produce the necessary raw materials, and the final products are delivered from plant L4 to warehouses M1 and M3.

A detailed summary of the problem size is presented in the first row of Table 3, which displays also the same information for the remaining examples. The first column of this table shows the number of constraints, continuous and binary variables for the full space (FP) problem. The second column shows the same characteristics for the lower bounding (LBP) and upper bounding (UBP) subproblems of the bi-level decomposition algorithm. The third column displays the same information for problems (P1), (P2) and the reduced (FP) solved by the Lagrangean decomposition algorithm. The last column displays the problem size of the sub-problems solved by the hybrid method.

Table 3
Model characteristics for approaches in each example.

		(FP)	Bi-level		Lagrangean			Hybrid		
			(LBP)	(UBP)	(P1)	(P2)	Reduced FP	(P1)	(P2)	Reduced FP
Example 1	bin. var.	223	68	115	220	3	218	65	3	218
	cont. var.	2082	2082	2081	2039	104	2142	2039	104	2142
	constr.	2144	2484	2144	2114	35	2204	2114	35	2204
Example 2	bin. var.	225	145	220	220	5	220	160	5	220
	cont. var.	3229	3208	3228	2319	1010	3328	2319	1010	3328
	constr.	2317	3156	2336	3986	215	4296	3986	215	4300
Example 3	bin. var.	450	140	430	440	10	440	130	10	440
	cont. var.	4488	4488	4488	4354	535	4888	4354	535	4888
	constr.	4286	4287	4286	4214	77	4686	4214	77	4686
Example 4	bin. var.	470	140	450	460	10	460	130	10	460
	cont. var.	6608	6608	6608	5994	1215	7208	5994	1215	7208
	constr.	6054	6054	6054	5916	145	6654	5916	145	6654
Example 5	bin. var.	700	205	675	690	10	685	195	10	685
	cont. var.	9603	9603	9603	8989	1515	10,503	8989	1515	10,503
	constr.	9004	9005	9004	8866	145	9904	8866	145	9904
Example 6	bin. var.	450	140	430	440	10	440	130	10	440
	cont. var.	4768	4768	4768	4354	815	5168	4354	815	5168
	const.	4314	4315	4314	4214	105	4714	4214	105	4714

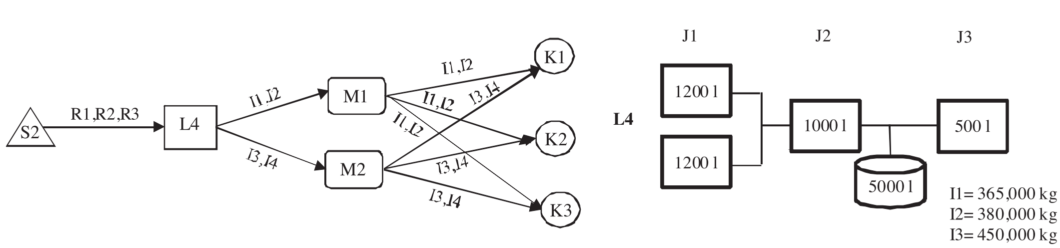

Fig. 5. Optimal SC and plant design for Example 1.

Table 4 shows the computational performance of the methods in all the examples. In the first row, the results of each strategy for Example 1 are shown. As observed, all of the methods identify the global optimum, but the (FP), Lagrangean and hybrid methods guarantee its global optimality (despite fixing an optimality gap of 1%).

It can be noted that models (LBP) and (UBP) used by the bi-level algorithm are smaller in size than model (FP). Hence, each iteration requires less time than the full space model. The algorithm converges in 5 iterations, and the total resolution time is very close to that corresponding to the (FP) model. The upper and lower bounds in each iteration are shown in Fig. S1 of the Supporting information. The algorithm stops after 5 iterations when the error between the bounds is equal to 0.7% (less than the fixed tolerance of 1%), identifying the global optimal solution at the upper bound.

The Lagrangean decomposition algorithm converges in 4 iterations and it takes more CPU time to arrive to the optimal solution than the rest of the methods. It is worth mentioning that the number of binary variables of (P1) is similar to the number of binary variables in (FP). Therefore, the major tradeoffs are evaluated in this subproblem and the computation burden of each iteration is similar to that associated with (FP). Fig. S2 of the supporting information section shows the upper and lower bounds in each iteration. As can be noted, the upper bound problem identifies the optimal solution in all the iterations. As a result, the same production plant (L4) and production amounts are fixed in the reduced FP in all the iterations.

The hybrid strategy needs three iterations to reach the optimal solution and takes 31.9 CPU s. As can be noted in Table 4, the lower and upper bounds cross each other. This is due to the use of integer cuts in the master problem that remove solutions explored in previous iterations. That means that the best solution identified so far is in turn the global optimum, since it is impossible to obtain any other solution with a better objective function value. Fig. S3 shows the upper and lower bounds in each iteration.

5.2. Example 2

Example 2 is taken from Corsano and Montagna (2011). This example considers six raw material sites, with three different raw materials in each site, five multiproduct batch plants with three stages each, where four products can be produced, five warehouses and 45 customer zones. The set of available batch units and storage tanks sizes are the same of the previous example. Unit duplication in and out of phase and tank allocation is considered. The model parameters are shown in Appendix B.

In the second row of Table 3 the model characteristics for the different approaches are reported. The optimal solution, which is identified by methods, selects two batch plants (L3 and L4) as shown in Fig. 6. The second row of Tables 3 and 4 shows the model characteristics and computational performance respectively. As observed, the three methods identify the global optimum, but only the bi-level one guarantees global optimality (despite fixing an optimality gap of 1%).

As can be noted from Table 4, the bi-level and hybrid algorithms perform better than the (FP) model (i.e., they lead to shorter computational time in only two and six iterations respectively). The Lagrangean algorithm shows a performance similar to that associated with the (FP). Fig. S4 shows the upper and lower bounds calculated by the Lagrangean method, while Fig. S5 shows the bounds obtained by the hybrid strategy. It can be noted from these figures that the hybrid method leads to fewer iterations compared to the Lagrangean algorithm due to the use of integer cuts.

Table 4
Computational performance for each approach in the presented examples.

	FP			Bi-level decomposition				Lagrangean spatial decomposition				Hybrid decomposition strategy			
	Solution	Gap	Time (s)	Solution	Gap	Time (s)	iter.	Solution	Gap	Time (s)	iter.	Solution	Gap	Time (s)	iter.
Example 1	UB = 1,019,543.2[a] LB = 1,019,543.2	0%[b]	79.8	UB = 1,019,543.2[a] LB = 1,012,543.6	0.7%	54.4	5	UB = 1,019,543.2[a] LB = 1,019,543.2	0%[b]	517.1	4	UB = 1,019,543.2[a] LB = 1,030,440.1	0%[b]	31.9	3
Example 2	UB = 1,926,396.1[a] LB = 1,907,133.3	0.99%	7342.05	UB = 1,926,396.1[a] LB = 1,970,720.9	0%[b]	324.2	2	UB = 1,926,396.1[a] LB = 1,910,971.6	0.8%	7623.4	14	UB = 1,926,396.1[a] LB = 1,914,976.7	0.6%	4913.6	6
Example 3	UB = 1,014,643.2[a] LB = 1,004,497.3	0.99%	14,244.4	UB = 1,014,643.2[a] LB = 1,005,993.6	0.9%	185.7	2	UB = 1,014,643.2[a] LB = 970,718.7	4.3%	14,400[c]	6	UB = 1,014,643.2[a] LB = 1,006,735.8	0.8%	775.71	8
Example 4	UB = 1,592,852.4 LB = 1,438,601.5	9.68%	14,400[c]	UB = 1,569,624.6[a] LB = 1,583,033.7	0%[b]	1839	9	UB = 1,599,279.2 LB = 1,389,259.6	13.1%	14,400[c]	4	UB = 1,628,520.3 LB = 1,430,749.3	12.1%	14,400[c]	15
Example 5	UB = 1,580,595 LB = 1,254,313.6	20.64%	14,400[c]	UB = 1,569,624.6[a] LB = 1,576,071.8	0%[b]	9022	7	UB = 1,589,898.6 LB = 1,445,003.8	9.4%	14,400[c]	4	UB = 1,590,001.1 LB = 1,331,471.6	16.3%	14,400[c]	9
Example 6	UB = 3,299,862.7 LB = 2,500,626.6	24.22%	14,400[c]	UB = 3,123,332.7[a] LB = 3,147,634.7	0%[b]	11,588	3	UB = 3,380,358.6 LB = 2,332,012.9	31%	14,400[c]	2	UB = 3,324,447.2 LB = 2,599,881.3	21.8%	14,400[c]	6

[a] Global optimum identified in the upper bound.
[b] The optimality gap is closed.
[c] The time limit is reached.

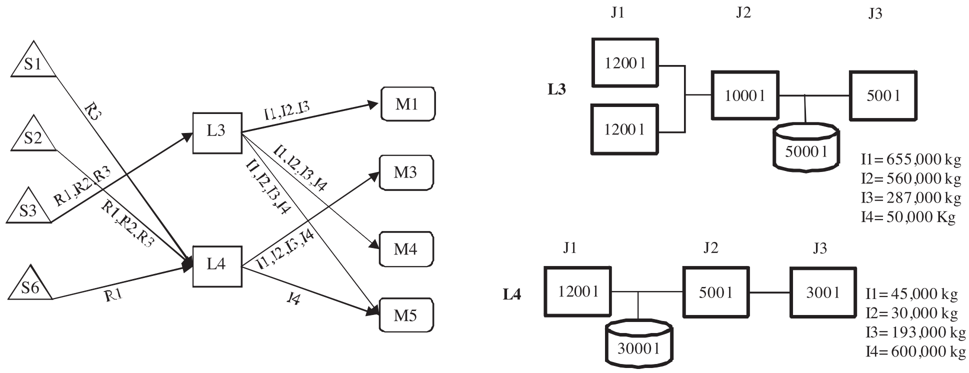

Fig. 6. Optimal SC and plant designs for Example 2.

5.3. Example 3

To test the proposed algorithm in large problems, this example considers 10 production plants and 10 warehouses. The number of customer zones and associated demand remain the same as in Example 1. The model data are presented in Appendix B. The third row of Table 3 shows the number of variables and constraints of each approach, while the third row of Table 4 displays their computational performance.

The (FP) method identifies a solution with the desired optimality gap after 14,244.4 CPU s, while the bi-level algorithm obtains the same optimal solution in only two iterations and 185.7 CPU s. The Lagrangean decomposition algorithm reaches the fixed time limit at the sixth iteration, and it is unable to close the gap below the desired tolerance. The lower bound obtained from the subgradient procedure improves slowly, and the overall algorithm convergence is rather slow. Fig. S6 shows the bounds produced by the Lagrangean algorithm at each iteration.

The hybrid algorithm identifies a solution within the desired tolerance in 8 iterations and 775.7 CPU s. Fig. S7 shows the bounds produced at each iteration.

All of the methods identify as best solution the one shown in Fig. 7. Fixing the optimality gap to 0% in the full space method, it can be verified that this solution is the global optimum. In this case, plant L9 is installed with a similar design as that obtained in Example 1 for plant L4. The total annual cost is equal to $1,014,643.2.

5.4. Example 4

In this case, the number of products to be produced in each plant is increased to 6, while the number of plants and warehouses remain equal to 10. The model parameters are shown in Appendix B.

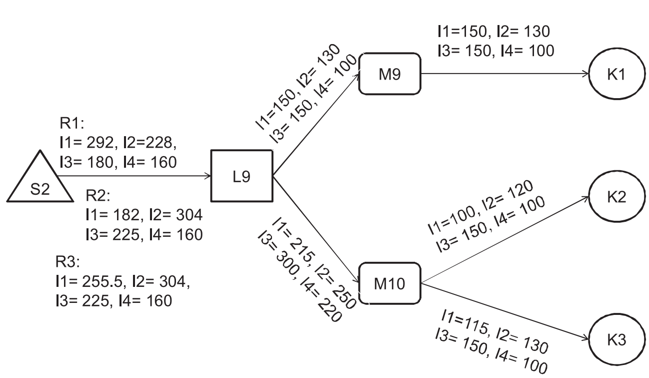

Fig. 7. Optimal SC design for Example 3 (amounts $\times 10^3$ kg), plant design similar to that obtained in Example 1 for plant L4.

The fourth row of Table 3 shows the model characteristics for this case, while the fourth row of Table 4 displays the computational performance for each approach. As noted, the bi-level decomposition approach is the only method that identifies the global optimal solution (in 9 iterations and 1839 CPU s). Fig. S8 shows the upper and lower bounds for each iteration in the bi-level decomposition algorithm, while Figs. S9 and S10 display the same information but for the Lagrangean and hybrid methods respectively. In these two last figures, it can be noted that none of these methods is able to close the optimality gap below the desired tolerance for the given time limit.

For this instance, the global optimum, which is obtained by the bi-level approach considers the installation of two plants (L1 and L9) and four warehouses (M1, M2, M3 and M8). Figure 8 shows SC and plant designs. The best solution obtained by the (FP) instals only one plant (L1) and three warehouses (M1, M2 and M3), while the solutions of the Lagrangean and hybrid approaches are similar to the bi-level solution in terms of network configuration, but the selected plants (L1 and L9) have different design and product assignments.

5.5. Example 5

In this example, the number of production plants is increased to 15, while the number of products to be produced, warehouses and customer zones remains as in the previous example. All the model parameters are displayed in Appendix B.

The fifth row of Tables 3 and 4 shows the model characteristics and computational performance, respectively. As can be noted from Table 3, this example shows the largest size due to the existence of major trade-offs concerning the number of plants to be used, their configuration, and the sizes of their equipment units. Again, the bi-level decomposition algorithm is the most efficient, being the only one that identifies the global optimal solution within the desired time limit. It takes 7 iterations and 9022 CPU s. for obtaining the optimal solution, while the remaining approaches are unable to close the optimality gap in the given time limit.

Figs. S11–S13 show the upper and lower bounds for each iteration in each of the proposed strategies. The hybrid strategy leads to more iterations than the Lagrangean method because it requires less computational time for solving each sub-problem. This is due to the fact that it relaxes some variables at the expense of yielding a lower bound lower than that obtained in Lagrangean methods, thereby providing an optimality gap worse than that of the Lagrangean algorithm.

The optimal SC and plant designs (which are obtained by the bi-level algorithm) are the same as that obtained for the previous example, i.e. it selects plants L1 and L9, and warehouses M1, M2, M3 and M8, as it is shown in Fig. 8. However, the SC design

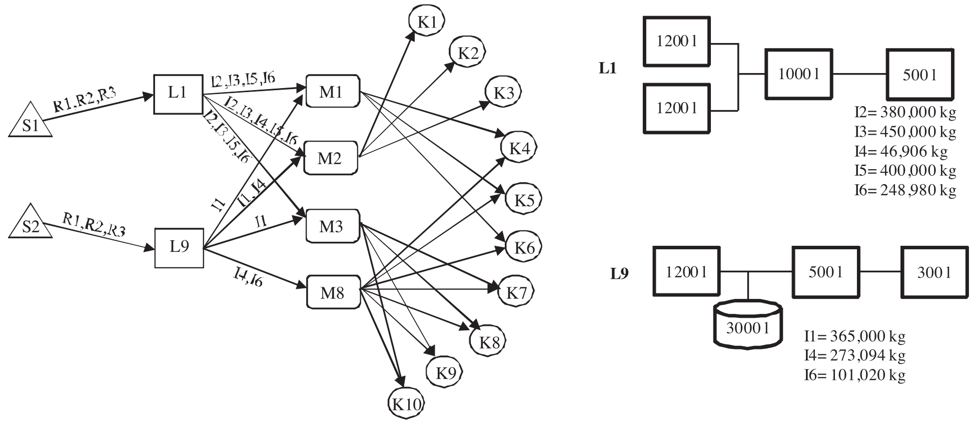

Fig. 8. Optimal SC and plant designs obtained by the bi-level algorithm for Examples 4 and 5.

changes in the best solutions found by the (FP), Lagrangean and hybrid methods. The optimal SC obtained by the full space problem selects plants L1 and L9, and warehouses M1, M2, M3, M8 and M10, while for the Lagrangean methods the solution selects plants L1 and L4, and warehouses M1, M2, M3 and M8. Finally, the hybrid methodology selects plants L1 and L8, and warehouses M1, M2 and M3.

5.6. Example 6

In this type of models where several decisions are jointly considered, the tradeoffs between different variables have a negative impact on the performance of the algorithm. In this example the customer zone demands are increased in order to assess the performance of the decomposition algorithms in more complex instances when the problem data lead to tradeoffs. We take as a basis Example 3, and then increase the number of customer zones to 10 and approximately double the total product demand of each product. The product demands and the transportation costs from warehouses to customer zones are shown in Appendix B, the rest of the model parameters are considered as in Example 3. Since the total product demand cannot be accommodated using only one batch plant, the model is forced to evaluate different alternatives, which increases its complexity.

As observed in the sixth row of Table 3, the number of binary variables is the same as in Example 3, since the addition of customer zones affects only the number of constraints and continuous variables. However, the (FP) model cannot be solved in 4 CPU hours (14,400 s). The best solution found within the fixed time limit shows a total cost equal to $3,299,862.7, which represents an optimality gap of 24.22%. This suboptimal solution considers the installation of plants L1, L8 and L9, and warehouses M1, M2, M3, M8 and M10. The SC and plant designs are shown in Fig. 9.

Again the bi-level decomposition algorithm is the most efficient one, identifying the global optimum after 11,588 CPU s and three iterations. The sixth row of Table 4 shows the computational performance for this example. The total cost is equal to $3,123,332.7. In the third iteration, the lower bound is greater than the upper bound due to the use of integer cuts, similarly as occurred with the hybrid approach in Example 1 and the bi-level decomposition method in Examples 2, 4 and 5. This implies that the best solution identified so far is in turn the global optimum, since it is impossible to obtain any other solution with a better objective function value. The optimal solution for the bi-level approach involves the establishment of plants L4, L6 and L9, and warehouses M1, M2, M3, M8 and M10. Fig. 10 shows the optimal SC and plant designs.

The Lagrangean decomposition algorithm leads to a suboptimal solution with a 31% of optimality gap after 4 CPU hours. The SC design opens plants L8 and L9, and warehouses M1, M2, M3, M8 and M10. The algorithm performs only two iterations before exceeding the time limit. In contrast, the hybrid approach executes 6 iterations in the same computational time and leads to a suboptimal solution with a 21.8% of optimality gap. The total cost for this solution is equal to $3,324,447.2, which is 6% worse than the total cost of the bi-level algorithm solution. In this case, the suboptimal SC configuration selects plants L1 and L9 and warehouses M1, M2, M3, M8 and M9, as it is shown in Fig. 11. It can be noted that two batch plants are selected, and the unit sizes for its stages

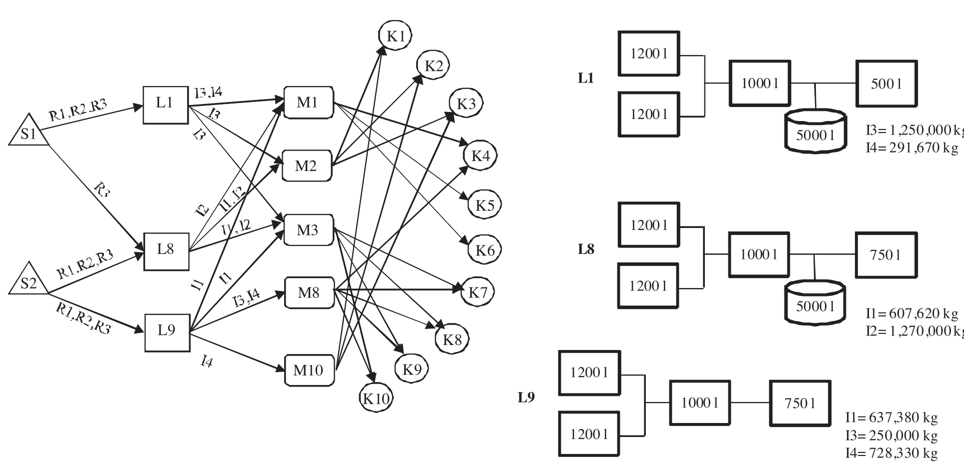

Fig. 9. Sub-optimal SC and plant designs of FP model for Example 6.

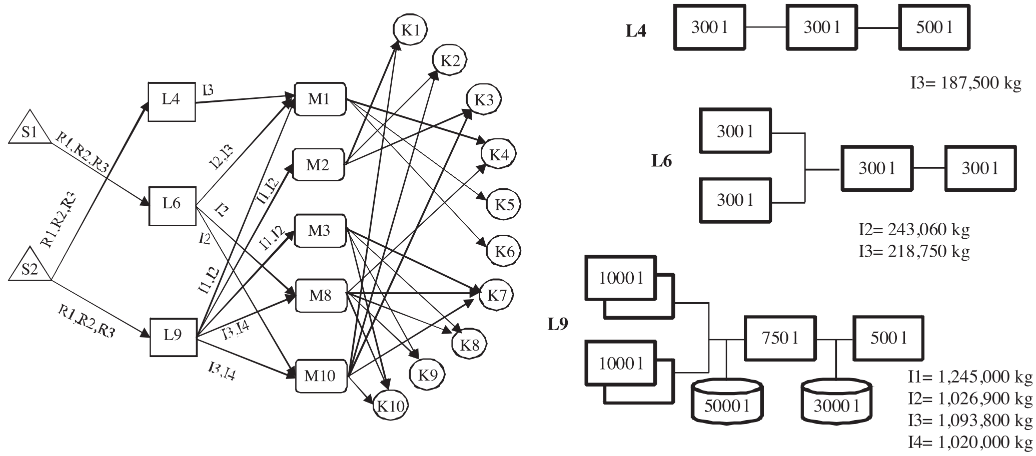

Fig. 10. Optimal SC and plant designs of bi-level model for Example 6.

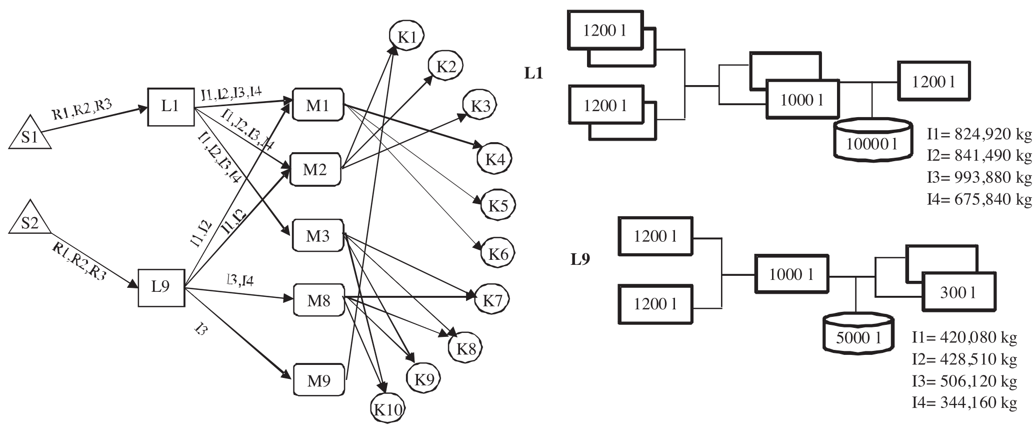

Fig. 11. Sub-optimal SC and plant designs of hybrid approach for Example 6.

are bigger than those selected in the optimal solution of the bi-level decomposition approach.

6. Conclusions

In this work we have addressed the simultaneous design of batch plants and strategic planning of supply chains. Taking as a basis an MILP that calculates decisions pertaining to different hierarchical levels in SCM (i.e., single site and multi-site design tasks), we have developed three decomposition algorithms that exploit the problem structure. Several case studies have been solved to test the capabilities of these numerical methods.

From these examples, it has been shown that the bi-level decomposition scheme is the best method in terms of quality of the final solution produced (i.e., it identifies the global optimum in all the examples) and time spent in its generation (it is the fastest to close the optimality gap in 5 of the 6 examples). This finding is consistent with other results published in the recent literature (You et al., 2011).

Regarding the Lagrangean decomposition, its performance is rather poor, as it takes the largest time to close the gap in 5 of the 6 examples, despite identifying the global optimum in three of the cases. This is because the subproblems have a large number of binary variables and several tradeoffs among decisions (product selection, plant configuration, unit sizing, etc.) must be evaluated therein.

On the other hand, the hybrid method reduces the number of binary variables in the subproblems. The hybrid method performs better than the Lagrangean method in all except one of the case studies (i.e., case 5), being the best in the smallest one and performing better than the full space branch and bound in cases 1, 2, 3, 5 and 6.

Appendix A. Mathematical modelling

The mathematical formulation for the supply chain and plant design optimization model introduced by Corsano and Montagna (2011) is summarized below. Note that the model neglects recycles between the SC entities. The symbols used for indices, sets, parameters and variables are described in the Nomenclature section.

A.1. SC network constraints

$$z_{il}Q_{il}^{LO} \le Q_{il} \le Q_{il}^{UP} z_{il} \quad i \in I, \quad l \in L \tag{A.1}$$

$$\sum_{i,l} Q_{sril} \le Q_{sr}^{UP} \quad s \in S, \quad r \in R \tag{A.2}$$

$$Q_{sril} \le Q_{sr}^{UP} z_{il} \quad s \in S, \quad r \in R, \quad i \in I, \quad l \in L \tag{A.3}$$

$$\sum_{s \in S} Q_{sril} = fp_{ril} Q_{il} \quad r \in R, \quad i \in I, \quad l \in L \tag{A.4}$$

$$z_{il} \le ex_l \quad i \in I, \quad l \in L \tag{A.5}$$

$$\sum_{m \in M} Q_{ilm} = Q_{il} \quad i \in I, \quad l \in L \tag{A.6}$$

$$\sum_{i,l} Q_{ilm} \le Q_m^{UP} y_m \quad m \in M \tag{A.7}$$

$$\sum_{l \in L} Q_{ilm} = \sum_{k \in K} Q_{imk} \quad i \in I, \quad m \in M \tag{A.8}$$

$$\sum_{i,k} Q_{imk} \le Q_m^{UP} y_m \quad m \in M \tag{A.9}$$

$$D_{ik} = \sum_{m \in M} Q_{imk} \quad i \in I, \quad k \in K \tag{A.10}$$

A.2. Batch plant design equations

$$NP_{jl} = \sum_{d=1}^{NP_{jl}^{UP}} d\,xx_{jld} \quad j \in J_l, \quad l \in L \tag{A.11}$$

$$\sum_{d=1}^{NP_{jl}^{UP}} xx_{jld} = 1 \quad j \in J_l, \quad l \in L \tag{A.12}$$

$$V_{jl} = \sum_{p=1}^{P_{jl}} \upsilon_{jlp} VF_{jlp} \quad j \in J_l, \quad l \in L \tag{A.13}$$

$$\sum_{p=1}^{P_{jl}} \upsilon_{jlp} = 1 \quad j \in J_l, \quad l \in L \tag{A.14}$$

$$Nb_{ijl} \ge \sum_{p,d} \frac{S_{ijl}}{VF_{jlp} d} e_{ijlpd} \quad i \in I, \quad j \in J_l, \quad l \in L \tag{A.15}$$

$$\sum_{d=1}^{NP_{jl}^{UP}} e_{ijlpd} \le Q_{il}^{UP} \upsilon_{jlp} \quad i \in I, \quad j \in J_l, \quad l \in L, \quad p = 1, .., P_{jl} \tag{A.16}$$

$$\sum_{p=1}^{P_{jl}} e_{ijlpd} \le Q_{il}^{UP} xx_{jld} \quad i \in I, \quad j \in J_l, \quad l \in L, \quad d = 1, .., NP_{jl}^{UP} \tag{A.17}$$

$$Q_{il} = \sum_{p,d} e_{ijlpd} \quad i \in I, \quad l \in L \tag{A.18}$$

$$\sum_{g=1}^{G_{jl}} \upsilon t_{jlg} = ex_l \quad j = 1, 2, \ldots, |J_l| - 1, \quad l \in L \tag{A.19}$$

$$Nb_{ijl} \ge 2 \sum_{g \ne 1} \frac{ST_{ijl}}{VTF_{jlg}} f_{ijlg} \quad i \in I, \quad j = 1, 2, \ldots, |J_l| - 1, \quad l \in L \tag{A.20}$$

$$Nb_{i,j+1,l} \ge 2 \sum_{g \ne 1} \frac{ST_{ijl}}{VTF_{jlg}} f_{ijlg} \quad i \in I, \quad j = 1, 2, \ldots, |J_l| - 1, \quad l \in L \tag{A.21}$$

$$f_{ijlg} \le Q_{il}^{UP} \upsilon t_{jlg} \quad i \in I, \quad j = 1, \ldots, |J_l| - 1, \quad l \in L, \quad g = 1, \ldots, G_{jl} \tag{A.22}$$

$$Q_{il} = \sum_{g=1}^{G_{jl}} f_{ijlg} \quad i \in I, \quad j = 1, 2, \ldots, |J_l| - 1, \quad l \in L \tag{A.23}$$

$$Nb_{ijl} + \left(\frac{1}{\phi} - 1\right) \sum_{g \ne 1} \lambda_{ijlg} \le Nb_{i,j+1,l} \le Nb_{ijl} + (\phi - 1) \sum_{g \ne 1} \lambda_{ijgl} \quad i \in I, \quad j = 1, 2, \ldots, \left|J_l\right| - 1, \quad l \in L \tag{A.24}$$

$$\lambda_{ijlg} \le Nb_{ijl}^{UP} \upsilon t_{jlg} \quad i \in I, \quad j = 1, \ldots, |J_l| - 1, \quad l \in L, \quad g = 1, \ldots, G_{jl} \tag{A.25}$$

$$Nb_{ijl} = \sum_{g=1}^{G_{jl}} \lambda_{ijlg} \quad i \in I, \quad j = 1, 2, .., |J_l| - 1, \quad l \in L \tag{A.26}$$

A.3. Timing constraints

$$\sum_{i=1}^{|I|} T_{il} \le H_l \quad l \in L \tag{A.27}$$

$$NT_{jl} = \sum_{n=1}^{NT_{jl}^{UP}} n x_{j\ln} \quad j \in J_l, \quad l \in L \tag{A.28}$$

$$\sum_{n=1}^{NT_{jl}^{UP}} x_{j\ln} = ex_l \quad j \in J_l, \quad l \in L \tag{A.29}$$

$$T_{il} \ge \sum_{n=1}^{NT_{jl}^{UP}} \frac{t_{ijl}}{n} w_{ij\ln} \quad i \in I, \quad j \in J_l, \quad l \in L \tag{A.30}$$

$$w_{ij\ln} \le Nb_{ijl}^{UP} x_{j\ln} \quad i \in I, \quad j \in J_l, \quad l \in L, \quad n = 1, \ldots, NT_{jl}^{UP} \tag{A.31}$$

$$Nb_{ijl} = \sum_{n=1}^{NT_{jl}^{UP}} w_{ij\ln} \quad i \in I, \quad j \in J_l, \quad l \in L \tag{A.32}$$

A.4. Objective function

$$\rho_{jlpnd} \ge \upsilon_{jlp} + x_{j\ln} + xx_{jld} - 2 \quad j \in J_l, \quad l \in L, \quad p = 1, \ldots, P_{jl}, \quad n = 1, .., NT_{jl}^{UP}, \quad d = 1, \ldots, NP_{jl}^{UP} \tag{A.33}$$

$$0 \le \rho_{jlpnd} \le 1 \quad j \in J_l, \quad l \in L, \quad p = 1, \ldots, P_{jl}, \quad n = 1, .., NT_{jl}^{UP}, \quad d = 1, \ldots, NP_{jl}^{UP} \tag{A.34}$$

$$C_{inv} = C_{ann} \left(\sum_{l \in L} \sum_{j \in J_l} \sum_{p=1}^{P_{jl}} \sum_{n=1}^{NT_{jl}^{UP}} \sum_{d=1}^{NP_{jl}^{UP}} \alpha_{jl} n d VF_{jlp}^{\beta_{jl}} \rho_{jlpnd} + \sum_{l \in L} \sum_{j \in J_l} \sum_{g=1}^{G_{jl}} \tilde{\alpha}_{jl} VTF_{jlg}^{\tilde{\beta}_{jl}} \upsilon t_{jlg} \right) \tag{A.35}$$

Table A.2.1
Raw material costs for Example 1.

	Distribution cost ($ kg^{-1})					Procurement cost ($ kg^{-1})		
	L1	L2	L3	L4	L5	R1	R2	R3
S1	0.02	0.1	0.08	0.06	0.08	0.02	0.02	0.01
S2	0.14	0.12	0.14	0.02	0.06	0.02	0.01	0.02

Table A.2.2
Batch plants parameters of Example 1 and 2.

	Size factors			Operating times (h)			Raw material conversion factor			Production cost ($ kg^{-1})				
	J1	J2	J3	J1	J2	J3	R1	R2	R3	L1	L2	L3	L4	L5
I1	0.9	0.6	0.4	14	5	7	0.8	0.5	0.7	0.12	0.18	0.12	0.06	0.12
I2	0.6	0.5	0.4	12	6	4	0.6	0.8	0.8	0.08	0.16	0.06	0.12	0.1
I3	0.7	0.5	0.4	16	8	5	0.4	0.5	0.5	0.12	0.14	0.14	0.08	0.14
I4	0.8	0.6	0.4	10	4	5	0.5	0.5	0.5	0.14	0.08	0.14	0.04	0.12

Table A.2.3
Batch plant investment cost parameters.

	Unit cost coefficient α_{jl} (annualized)					Unit exponent coefficient β_{jl}
	L1	L2	L3	L4	L5	For all the plants
J1	1620	2430	1350	1350	1890	0.6
J2	2160	1620	2160	1620	1890	0.6
J3	1890	2700	1890	1890	2430	0.7
Tanks	500	500	500	500	500	0.6

Table A.2.4
Distribution cost between plants and warehouses for Example 1 ($ kg^{-1}).

	M1				M2				M3			
	I1	I2	I3	I4	I1	I2	I3	I4	I1	I2	I3	I4
L1	0.1	0.17	0.05	0.05	0.2	0.1	0.15	0.15	0.23	0.16	0.11	0.11
L2	0.2	0.19	0.25	0.25	0.19	0.18	0.35	0.35	0.18	0.19	0.15	0.15
L3	0.2	0.18	0.25	0.25	0.18	0.15	0.25	0.25	0.15	0.08	0.15	0.18
L4	0.05	0.1	0.2	0.15	0.15	0.11	0.2	0.2	0.1	0.15	0.15	0.05
L5	0.2	0.18	0.25	0.25	0.2	0.15	0.25	0.25	0.15	0.15	0.08	0.08

Table A.2.5
Product demands and distribution cost from warehouses to customer zones for Example 1.

	Product demand (kg)				Distribution cost ($ kg^{-1})		
	I1	I2	I3	I4	M1	M2	M3
K1	150,000	130,000	150,000	100,000	0.08	0.09	0.09
K2	100,000	120,000	150,000	100,000	0.07	0.09	0.08
K3	115,000	130,000	150,000	120,000	0.06	0.07	0.05

Table A.2.6
Example 2. Distribution and raw material procurement costs.

	L1	L2	L3	L4	L5	R1	R2	R3
S1	0.6	0.5	0.4	0.3	0.2	0.2	0.2	0.1
S2	0.5	0.4	0.3	0.2	0.6	0.2	0.1	0.2
S3	0.4	0.3	0.2	0.6	0.5	0.1	0.1	0.1
S4	0.3	0.2	0.6	0.5	0.4	0.2	0.2	0.2
S5	0.2	0.6	0.5	0.4	0.3	0.3	0.3	0.2
S6	0.6	0.5	0.4	0.3	0.2	0.1	0.1	0.3

Table A.2.7
Example 2. Product demands in each customer zone.

	K1	K2-K15	K16	K17-K30	K31	K32-K39	K40	K41-K42	K43-K45
I1	15,000	15,000	40,000	15,000	40,000	15,000	15,000	10,000	10,000
I2	50,000	10,000	50,000	10,000	10,000	10,000	20,000	20,000	20,000
I3	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000	20,000
I4	15,000	15,000	15,000	15,000	15,000	15,000	15,000	10,000	10,000

Table A.2.8
Example 3. Batch plant investment cost parameters.

	Unit cost coefficient α_{jl} (annualized)										Unit exponent coef. β_{jl}
	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L1–L10
J1	1620	2430	1350	1350	1890	1620	2430	1350	1350	1890	0.6
J2	2160	1620	2160	1620	1890	2160	1620	2160	1620	1890	0.6
J3	1890	2700	1890	1890	2430	1890	2700	1890	1890	2430	0.7
Tanks	500	500	500	500	500	500	500	500	500	500	0.6

Table A.2.9
Example 3. Raw material costs.

	Distribution cost ($\$\,kg^{-1}$)										Procurement costs ($\$\,kg^{-1}$)		
	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	R1	R2	R3
S1	0.01	0.05	0.04	0.03	0.04	0.02	0.08	0.05	0.03	0.02	0.02	0.02	0.01
S2	0.07	0.06	0.07	0.01	0.03	0.06	0.02	0.04	0.01	0.06	0.02	0.01	0.02

Table A.2.10
Example 4. Product demands for each customer zone (amounts $\times 10^3$ kg).

Products	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10
I1	36.5	36.5	36.5	36.5	36.5	36.5	36.5	36.5	36.5	36.5
I2	38	38	38	38	38	38	38	38	38	38
I3	45	45	45	45	45	45	45	45	45	45
I4	32	32	32	32	32	32	32	32	32	32
I5	40	40	40	40	40	40	40	40	40	40
I6	35	35	35	35	35	35	35	35	35	35

Table A.2.11
Batch plants parameters of Example 4.

	Size factors			Operating times (h)			Raw material conversion factor			Production cost ($\$\,kg^{-1}$)									
	J1	J2	J3	J1	J2	J3	R1	R2	R3	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10
I1	0.9	0.6	0.4	14	5	7	0.8	0.5	0.7	0.12	0.18	0.12	0.06	0.12	0.12	0.18	0.12	0.06	0.12
I2	0.6	0.5	0.4	12	6	4	0.6	0.8	0.8	0.08	0.16	0.06	0.12	0.1	0.08	0.16	0.06	0.12	0.10
I3	0.7	0.5	0.4	16	8	5	0.4	0.5	0.5	0.12	0.14	0.14	0.08	0.14	0.12	0.14	0.14	0.08	0.14
I4	0.8	0.6	0.4	10	4	5	0.5	0.5	0.5	0.14	0.08	0.14	0.04	0.12	0.14	0.08	0.14	0.04	0.12
I5	0.7	0.6	0.5	10	8	6	0.5	0.6	0.7	0.06	0.08	0.06	0.06	0.08	0.1	0.1	0.1	0.1	0.1
I6	0.8	0.7	0.5	15	8	5	0.6	0.4	0.6	0.1	0.1	0.14	0.04	0.12	0.1	0.1	0.1	0.08	0.1

Table A.2.12
Distribution cost between plants and warehouses for Example 4 ($\$\,kg^{-1}$).

	M1–M4						M5–M7						M8–M10					
	I1	I2	I3	I4	I5	I6	I1	I2	I3	I4	I5	I6	I1	I2	I3	I4	I5	I6
L1-L2	0.1	0.17	0.05	0.05	0.05	0.05	0.2	0.1	0.15	0.15	0.15	0.15	0.23	0.16	0.11	0.11	0.11	0.11
L3-L4	0.2	0.19	0.25	0.25	0.25	0.25	0.19	0.18	0.35	0.35	0.35	0.35	0.18	0.19	0.15	0.15	0.15	0.15
L5-L6	0.2	0.18	0.25	0.25	0.25	0.25	0.18	0.15	0.25	0.25	0.25	0.25	0.15	0.08	0.15	0.18	0.18	0.18
L7-L8	0.05	0.1	0.2	0.15	0.15	0.15	0.15	0.11	0.2	0.2	0.2	0.20	0.1	0.15	0.15	0.05	0.05	0.05
L9-L10	0.2	0.18	0.25	0.25	0.25	0.25	0.2	0.15	0.25	0.25	0.25	0.25	0.15	0.15	0.08	0.08	0.08	0.08

Table A.2.13
Example 5. Batch plant investment cost parameters.

	Unit cost coefficient α_{jl} (annualized)															Unit exponent coefficient β_{jl}
	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15	L1–L15
J1	1620	2430	1350	1350	1890	1620	2430	1350	1350	1890	1620	2430	1350	1350	1890	0.6
J2	2160	1620	2160	1620	1890	2160	1620	2160	1620	1890	1620	2160	1620	1620	2160	0.6
J3	1890	2700	1890	1890	2430	1890	2700	1890	1890	2430	1890	2160	1890	1620	1890	0.7
Tanks	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	0.6

Table A.2.14
Distribution cost between plants and warehouses for Example 5 ($\$\,kg^{-1}$).

	M1–M4						M5–M7						M8–M10					
	I1	I2	I3	I4	I5	I6	I1	I2	I3	I4	I5	I6	I1	I2	I3	I4	I5	I6
L1-L2	0.1	0.17	0.05	0.05	0.05	0.05	0.2	0.1	0.15	0.15	0.15	0.15	0.23	0.16	0.11	0.11	0.11	0.11
L3-L4	0.2	0.19	0.25	0.25	0.25	0.25	0.19	0.18	0.35	0.35	0.35	0.35	0.18	0.19	0.15	0.15	0.15	0.15
L5-L6	0.2	0.18	0.25	0.25	0.25	0.25	0.18	0.15	0.25	0.25	0.25	0.25	0.15	0.08	0.15	0.18	0.18	0.18
L7-L8	0.05	0.1	0.2	0.15	0.15	0.15	0.15	0.11	0.2	0.2	0.2	0.20	0.1	0.15	0.15	0.05	0.05	0.05
L9-L15	0.2	0.18	0.25	0.25	0.25	0.25	0.2	0.15	0.25	0.25	0.25	0.25	0.15	0.15	0.08	0.08	0.08	0.08

Table A.2.15
Product demands for each customer zone of Example 6 (amounts x10^3 kg).

Products	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10
I1	150	100	115	150	100	115	150	100	115	150
I2	130	120	130	130	120	130	130	120	130	130
I3	150	150	150	150	150	150	150	150	150	150
I4	100	100	120	100	100	100	100	100	100	100

Table A.2.16
Distribution cost between warehouses and customer zones for all products for Example 6 ($\$\,kg^{-1}$).

	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10
M1	0.1	0.1	0.1	0.01	0.01	0.01	0.09	0.09	0.09	0.09
M2	0.01	0.01	0.01	0.1	0.1	0.1	0.09	0.09	0.09	0.09
M3	0.09	0.09	0.09	0.1	0.1	0.1	0.01	0.01	0.01	0.01
M4–M6	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
M7–M8	0.2	0.2	0.2	0.1	0.1	0.1	0.08	0.08	0.08	0.08
M9–M10	0.1	0.1	0.1	0.2	0.2	0.2	0.1	0.1	0.1	0.1

$$C_{inst} = C_{ann}\left(\sum_{l\in L} Cpl_l ex_l + \sum_{m\in M} Cdep_m y_m\right) \quad (A.36)$$

$$C_{oper} = \sum_{s\in S}\sum_{r\in R}\sum_{i\in I}\sum_{l\in L} Craw_{sr} Q_{sril} + \sum_{i\in I}\sum_{l\in L}\sum_{m\in M} Cd_{im} Q_{ilm} + \sum_{i\in I}\sum_{l\in L} Cprod_{il} Q_{il} \quad (A.37)$$

$$C_{trans} = \sum_{s\in S}\sum_{r\in R}\sum_{i\in I}\sum_{l\in L} Ctraw_{srl} Q_{sril} + \sum_{i\in I}\sum_{l\in L}\sum_{m\in M} Ctp_{ilm} Q_{ilm} + \sum_{i\in I}\sum_{m\in M}\sum_{k\in K} Ctd_{imk} Q_{imk} \quad (A.38)$$

Appendix B. Problem data

In this section the more relevant model parameters are reported. For space reason, not all are presented, but they are available for interesting readers.

For all instances, the production time horizon was taken equal to 7000 h.

For each batch stage, we consider a set of 5 discrete sizes: 300 l, 500 l, 750 l, 1000 l, and 1200 l; and 3 different tank sizes: 3000 l, 5000 l, and 10,000 l.

In Example 1, 3 warehouses are available with a capacity equal to 1,050,000 kg, 1,250,000 kg, and 1,100,000 kg. In Example 2, the warehouses are 5 with capacity 1,500,000, 750,000, 750,000, 750,000, 750,000 kg respectively, while 10 warehouses are available in Examples 3, 4, 5 and 6, with maximum capacity of 1,500,000 kg for M1–M3, 12,500,000 kg for M4–M6, 110,000,000 kg for M7–M10.

Plant installation cost is adopted as $7000, $15,000, $7000, $9000, and $12,000 for plants L1–L5 respectively in Examples 1 and 2. In Examples 3, 4 and 6 the plant installation cost is $7000 for plant L1, L3, L6 and L8; $15,000 for L2 and L7; $9000 for L4 and L9, and $12,000 for L5 and L10. For Example 5, the installation costs adopted are: $7000 for L1, L3, L6, L8, L13 and L14; $15,000 for L2 and L7; $9000 for L4, L9, L11; $12,000 for L5, L10 and L15; and $10,000 for L12.

The warehouses installation cost for Example 1 is taken as $1500 for M1, $2000 for M2 and $2500 for M3. In Example 2 the installation cost for warehouses is equal to $1500 for M1, $2000 for M2 and M4, $ for M3 and $3500 for M5, while for Examples 3–6 the costs are: $1500 for M1, M4, M7 and M10; $2000 for M2, M5, and M8; and $2500 for M3, M6 and M9.

Following, the data arranged in tables is presented: Tables A.2.1–A.2.16.

Appendix C. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.compchemeng.2013.09.001.

References

Amaro, A. C. S., & Barbosa-Póvoa, A. P. F. D. (2008). Supply chain management with optimal scheduling. *Industrial and Engineering Chemistry Research*, *47*, 116–132.

Balas, E., & Jeroslow, R. (1972). Canonical cuts on the unit hypercube. *SIAM Journal on Applied Mathematics*, *23*, 61–79.

Barbosa-Póvoa, A. P. (2007). A critical review on the design and retrofit of batch plants. *Computers and Chemical Engineering*, *31*, 833–855.

Chen, P., & Pinto, J. M. (2008). Lagrangean-based techniques for the supply chain management of flexible process networks. *Computers and Chemical Engineering*, *32*(11), 2505–2528.

Cóccola, M. E., Zamarripa, M., Mendez, C. A., & Espuña, A. (2013). Toward integrated production and distribution management in multi-echelon supply chains. *Computers and Chemical Engineering*, http://dx.doi.org/10.1016/j.compchemeng.2013.01.004

Corsano, G., Montagna, J. M., Iribarren, O. A., & Aguirre, P. A. (2007). Heuristic method for the optimal synthesis and design of batch plants considering mixed product campaigns. *Industrial and Engineering Chemistry Research*, *46*, 2769–2780.

Corsano, G., & Montagna, J. M. (2011). Mathematical modeling for simultaneous design of plant and supply chain in batch process industry. *Computers and Chemical Engineering*, *35*(1), 149–164.

Corsano, G., Vecchietti, A. R., & Montagna, J. M. (2011). Optimal design for sustainable bioethanol supply chain considering detailed plant performance model. *Computers and Chemical Engineering*, *35*, 1384–1398.

Fisher, M. L. (1981). The Lagrangean relaxation method for solving integer programming problems. *Management Science*, *27*(1), 18.

Goetschalckx, M., Vidal, C. J., & Dogan, K. (2002). Modeling and design of global logistics systems: A review of integrated strategic and tactical models and design algorithms. *European Journal of Operational Research*, *143*, 1–18.

Graves, S. C. (1982). Using Lagrangean techniques to solve hierarchical production planning problems. *Management Science*, *28*(3), 260–275.

Grossmann, I. E. (2004). Challenges in the new millennium: Product discovery and design, enterprise and supply chain optimization, global life cycle assessment. *Computers and Chemical Engineering*, *29*, 29–39.

Grossmann, I. E. (2005). Enterprise-wide optimization: A new frontier in process systems engineering. *AIChE Journal*, *51*(7), 1846–1857.

Grossmann, I. E. (2012). Advances in mathematical programming models for enterprise-wide optimization. *Computers and Chemical Engineering*, *47*, 2–18.

Guignard, M., & Kim, S. (1987). Lagrangean decomposition: A model yielding stronger Lagrangean bounds. *Mathematical Programming*, *39*, 215–228.

Guillén, G., Badell, M., Espuña, A., & Puigjaner, L. (2006). Simultaneous optimization of process operations and financial decisions to enhance the integrated planning/scheduling of chemical supply chains. *Computers and Chemical Engineering*, *30*, 421–436.

Guillén-Gosálbez, G., Mele, F., & Grossmann, I. E. (2010). A bi-criterion optimization approach for the design and planning of hydrogen supply chains for vehicle use. *AIChE Journal*, *56*(3), 650–667.

Iyer, R. R., & Grossmann, I. E. (1998). A bilevel decomposition algorithm for long-range planning of process networks. *Industrial and Engineering Chemistry Research*, *37*, 474–481.

Jackson, J. R., & Grossmann, I. E. (2003). Temporal decomposition scheme for non-linear multisite production planning and distribution models. *Industrial and Engineering Chemistry Research*, *42*(13), 3045–3055.

Lee, H.-K., Lee, I. B., & Reklaitis, G. V. (2000). Capacity expansion problem of multisite batch plants with production and distribution. *Computers and Chemical Engineering*, *24*, 1597–1602.

Longinidis, P., & Georgiadis, M. C. (2013). Managing the trade-offs between financial performance and credit solvency in the optimal design of supply chain networks under economic uncertainty. *Computers and Chemical Engineering*, *48*, 264–279.

Maravelias, C. T., & Sung, C. (2009). Integration of production planning and scheduling: Overview, challenges and opportunities. *Computers and Chemical Engineering*, *33*, 1919–1930.

McDonald, C. M., & Reklaitis, G. V. (2004). Design and operation of supply chains in support of business and financial strategies. In *Foundations of computer-aided process operations* Princeton, NJ, USA, July 11–16,

Melo, M. T., Nickel, S., & Saldanha-da-Gama, F. (2009). Facility location and supply chain management—A review. *European Journal of Operational Research*, *196*, 401–412.

Papageorgiou, L. G. (2009). Supply chain optimisation for the process industries: Advances and opportunities. *Computers and Chemical Engineering*, *33*(12), 1931–1938.

Rosenthal, R. E. (2008). *GAMSA: A user's manual*. Washington, DC: GAMS Development Corporation.

Rungtusanatham, M., & Forza, C. (2005). Coordinating product design, process design, and supply chain design decisions: Part A: Topic motivation, performance implications, and article review process. *Journal of Operations Management*, *23*(3–4), 257–265.

Shah, N. (2005). Process industry supply chains: Advances and challenges. *Computers and Chemical Engineering*, *29*, 1225–1235.

Spengler, T. H., Püchert, H., Penkuhn, T., & Rentz, O. (1997). Environmental integrated production and recycling management. *European Journal of Operational Research*, *97*, 308–326.

Sundaramoorthy, A., & Karimi, I. A. (2004). Planning in pharmaceutical supply chains with outsourcing and new product introductions. *Industrial and Engineering Chemistry Research*, *43*, 8293–8306.

Susarla, N., & Karimi, I. A. (2012). Integrated supply chain planning for multinational pharmaceutical enterprises. *Computers and Chemical Engineering*, *42*, 168–177.

van den Heever, S. A., Grossmann, I. E., & Vasantharajan, S. A. (2001). Lagrangean decomposition heuristic for the design and planning of offshore hydrocarbon field infrastructures with complex economic objectives. *Industrial and Engineering Chemistry Research*, *40*(13), 2857–2875.

Varma, V. A., Rekalitis, G. V., Blau, G. E., & Pekny, J. F. (2007). Enterprise-wide modelling and optimization—An overview of emerging research challenges and opportunities. *Computers and Chemical Engineering*, *31*, 692–711.

You, F., & Grossmann, I. E. (2008). Mixed-integer nonlinear programming models and algorithms for large-scale supply chain design with stochastic inventory management. *Industrial and Engineering Chemistry Research*, *47*, 7802–7817.

You, F., Grossmann, I. E., & Wassick, J. M. (2011). Multisite capacity, production, and distribution planning with reactor modifications: MILP model, bilevel decomposition algorithm versus Lagrangean decomposition scheme. *Industrial and Engineering Chemistry Research*, *50*, 4831–4849.