Computers and Chemical Engineering 62 (2014) 211–224

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

Optimal supply chain design and management over a multi-period horizon under demand uncertainty. Part II: A Lagrangean decomposition algorithm

Jiang Yongheng [a], Maria Analia Rodriguez [b], Iiro Harjunkoski [c], Ignacio E. Grossmann [d]

[a] Institute of Process Control Eng., Department of Automation, Tsinghua University, Beijing 100084, PR China
[b] INGAR (CONICET-UTN), Avellaneda 3657, Santa Fe 3000, Argentina
[c] ABB AG, Corporate Research Germany, Wallstadter Straße 59, 68526 Ladenburg, Germany
[d] Center for Advanced Process Decision-Making, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA

ARTICLE INFO

Article history:
Received 6 June 2013
Received in revised form
17 November 2013
Accepted 22 November 2013
Available online 7 December 2013

Keywords:
Supply chain
Lagrangean decomposition
Adaptive piecewise linearization

ABSTRACT

In Part I (Rodriguez, Vecchietti, Harjunkoski, & Grossmann, 2013), we proposed an optimization model to redesign the supply chain of spare parts industry under demand uncertainty from strategic and tactical perspectives in a planning horizon consisting of multiple time periods. To address large scale industrial problems, a Lagrangean scheme is proposed to decompose the MINLP of Part I according to the warehouses. The subproblems are first approximated by an adaptive piece-wise linearization scheme that provides lower bounds, and the MILP is further relaxed to an LP to improve solution efficiency while providing a valid lower bound. An initialization scheme is designed to obtain good initial Lagrange multipliers, which are scaled to accelerate the convergence. To obtain feasible solutions, an adaptive linearization scheme is also introduced. The results from an illustrative problem and two real world industrial problems show that the method can obtain optimal or near optimal solutions in modest computational times.

1. Introduction

In the spare parts industry, or more specifically the electric motor industry as was illustrated in part 1 (Rodriguez, Vecchietti, Harjunkoski, & Grossmann, 2013), there are some key issues that strongly influence the cost of the supply chain. One is that a low-level inventory is important (bound capital). Moreover, it is critical that a spare motor can be obtained as soon as possible since the motor is a key part of the customer plant. Tens or hundreds of different types of motors are required by the customers. Also, the criticality of a given unit can be very different. If the time requirement is very tight, it might be necessary to have some motors in stock at the customer sites. The main objective of the model is to optimally redesign supply chain to meet the demand with minimal costs involving decisions on where to place warehouses, which installed warehouses should be expanded or shutdown, as well as deciding the stock capacities, safety stocks required, and how to connect the different echelons of the supply chain in order to satisfy uncertain demand of motors. Due to the above features, the problem corresponds to a large scale MINLP problem that is very hard to solve.

Lagrangean decomposition has been successfully applied to large-scale mathematical programming problems (Wang, 2003). According to the problem structure, temporal and spatial decomposition can be adopted (Terrazas-Moreno, Trotter, & Grossmann, 2011). The subgradient optimization is a popular method for updating the multipliers in Lagrangean decomposition (Baker and Sheasby, 1999), although the convergence of the multipliers is the main challenge. Other contributions include methods for accelerating convergence through the use of subgradients (Baker and Sheasby, 1999; Fumero, 2001) and other strategies (Mouret, Grossmann, & Pestiaux, 2011; Buil, Piera, & Luh, 2012). In Terrazas-Moreno et al. (2011), an economic interpretation of the multipliers is given, which can benefit from the problem structure to accelerate the convergence. Considering that the dual problem is a high-dimensional nonlinear problem, the shape of its domain and contours is a key to accelerate the convergence, and the interpretation from an economic view may be helpful.

This paper is organized as follows. In Section 2, the model from Part I is reformulated. In Section 3, a decomposition scheme is proposed, and the methods to solve the subproblems, initialize and update the multipliers, and design of the feasibility problem are discussed. The results from an illustrative example and two real world industrial problems are shown and discussed in Section 4. Finally, some conclusions are drawn in Section 5.

0098-1354/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compchemeng.2013.11.014

Nomenclature

Indices

c	criticality levels of motors
i	factories
j	warehouses
k	end customers
p	standard units
s	special units
t	time periods

Sets

CT_{ks}	customers k that allow used repaired units to satisfy their demand of units
JF	subset of warehouses j that are already installed (fixed) at the beginning of the horizon planning
KSC_{ksc}	customers k that order special units s of criticality c
KT_{ks}	customers k that order tailor made units s
PS_{ps}	special units s belonging to standard unit p
SC	subset of warehouses j that can be also considered as repair workshops

Binary variables

u_{ikst}	if factory i produces and delivers tailor made unit s to end customer k in period t
v_{jkst}	if repair workshop j repairs special units s from customer k in period t
w_{it}	if factory i is installed in period t
w_{it}^{e}	if warehouse j is expanded in period t
w_{it}^{u}	if factory i is uninstalled (eliminated) in period t
x_{ijpt}	if factory i produces and delivers standard units p to warehouse j in period t
y_{jt}	if warehouse j is installed in period t
y_{jt}^{e}	if warehouse j is expanded in period t
y_{jt}^{u}	if warehouse j is uninstalled (eliminated) in period t
z_{jkt}	if warehouse j delivers units to customer k in period t
β_I	auxiliary variable for linearization of subproblems

Positive variables

l_{ksct}	net lead time of customer k for special unit s of criticality c in period t
l'_{jksct}	net lead time of customer k if special unit s of criticality c is provided by warehouse j in period t
m_{ksct}	net lead time of customer k for tailor made unit s of criticality c in period t
m'_{jksct}	net lead time of customer k if tailor made unit s of criticality c is provided by warehouse j in period t
TI_t	the total investment cost in period t
TIw_{tj}	the total investment costin new warehouse j in period t
TOF_t	the total operational fixed cost in period t
$TOFw_{tj}$	the total operational fixed cost in warehouse j in period t
TE_t	the total investment expansion cost in period t
TEw_{tj}	the total investment expansion cost in warehouse j in period t
TU_t	the total shutdown cost in period t
TUw_{tj}	the total shutdown cost in warehouse j in period t
TOV_t	the total variable cost in period t
$TOVw_{tj}$	the total variable cost in warehouse j in period t
TPV_t	the total variable cost in factories for the motors transported in period t
$TPVw_{tj}$	the total variable cost in factories for the motors transported to warehouse j in period t
TR_t	the repair cost in period t
TRw_{tj}	the repair cost in warehouse j in period t
TTF_t	the transportation cost from factories to warehouses and customer sites in period t
$TTFw_{tj}$	the transportation cost from factories to warehouse j in period t
$TTFc_{tT}$	the transportation cost from factories to customer sites in period t
TTW_t	the transportation cost in period t
$TTWw_{tj}$	the transportation cost from warehouse j in period t
TPC_t	the mean inventory cost at customer sites for the special motors from warehouse and tailor made motors in period t
$TPCw_{tj}$	the mean inventory cost at customer sites for the special motors from warehouse j in period t
$TPCc_{tT}$	the mean inventory cost at customer sites for the special motors from tailor made motors in period t
TSS_t	the summation of the safety stock cost in period t
$TSSw_{tj}$	the summation of the safety stock cost at warehouse j and customer sites for the special motors from warehouse j in period t

$TSSc_{tT}$ the safety stock cost at customer sites for tailor made motors in period t
TBT_t the lost sales stock cost for special motors in period t
$TBTw_{tj}$ the lost sales stock cost for special motors from warehouse j in period t
$\lambda z_{kt}; \lambda v_{skt}; \lambda c$ the positive Lagrange multipliers
$a_I, a'_I \gamma_1, \gamma_2$ auxiliary variables for linearization of subproblems
μ^{new}_{ijkpt} amount of demand of standard units p from customer k satisfied with new units from factory i and warehouse j
μ^{used}_{jkst} amount of demand of special units s from customer k satisfied with used units from repair workshop j
τ^{new}_{ikst} amount of demand of tailor made units p from customer k satisfied with new units from factory i
τ^{used}_{jkst} amount of demand of tailor made units s from customer k satisfied with used units from repair workshop j

Variables
$g = (gz^T_{kt}, gv^T_{skt}, gmuct^T_{kpt}, gtol^T_{ks}, gc^T)^T$ the subgradients of the Lagrangean function
x the general independent variable
x^p the solution of the previous iteration for linearization of feasibility problem
x^t the temporal variables for linearization of subproblems
y the general dependent variable
L the Lagrangean function
L_j the part for warehouse j of the Lagrangean function
L_r the remaining part of the Lagrangean function excluding L_j
$\lambda muct_{kpt}; \lambda tol_{kst}$ the Lagrange multipliers

Parameters
a the scalar for linearization of feasibility problem
$b1_{ks}$ unit annual lost sales cost for special unit s at customer k
$c1_{ij}$ unit transportation cost from factory i to warehouse j
$c2_{jk}$ unit transportation cost from warehouse j to customer k
$c3_{ik}$ unit transportation cost from factory i to customer k
$t2_{jkp}$ order processing time of customer k for standard unit p if it is served by warehouse j, including material handling time in k, transportation time from warehouse j to k, and inventory review period in the customer site
α_p production factor rate for standard unit p
$\alpha_z, \alpha_v, \alpha_t$ the scalars for subgradient scaling
μ_{ksct} mean demand of special units s of criticality c from customer k in period t
σ_{ksct} demand standard deviation of special units s of criticality c from customer k in period t
x^L, x^U the lower and upper bounds of x
x days in the year

2. The supply chain model reformulation

In order to design the decomposition algorithm, we reformulate the model from Part I to aggregate the terms in the objective and constraints according to the warehouses for which we consider potential selection, capacity expansion and shutdowns. In the reformulated model, we assume for simplicity that no factory expansion and shutdown are considered. That is, all the necessary factories are given with fixed capacities at the beginning of time horizon for the design of the supply chain.

Firstly, the cost terms (Eqs. (54), (56), (58), (60), (62)–(70) from Part I) are disaggregated in the objective function (Eq. (72) from Part I) in terms of the warehouses j, as follows.

$$TI_t = \sum_j TIw_{tj} \quad \forall t \tag{1}$$

where TIw_{tj} denotes the total investment cost in new warehouse j in period t.

$$TOF_t = \sum_j TOFw_{tj} \quad \forall t \tag{2}$$

where $TOFw_{tj}$ denotes the total operational fixed cost in warehouse j in period t.

$$TE_t = \sum_j TEw_{tj} \quad \forall t \tag{3}$$

where TEw_{tj} denotes the total investment expansion cost in warehouse j in period t.

$$TU_t = \sum_j TUw_{tj} \quad \forall t, \quad \forall t \tag{4}$$

where TUw_{tj} denotes the total shutdown cost in warehouse j in period t.

$$TOV_t = \sum_j TOVw_{tj} \quad \forall t \tag{5}$$

where TOV_{tj} denotes the total variable cost in warehouse j in period t.

$$TPV_t = \sum_j TPVw_{tj} \quad \forall t \tag{6}$$

where $TPVw_{tj}$ denotes the total variable cost in factories for the motors transported to warehouse j in period t.

$$TR_t = \sum_{j \in SC} TRw_{tj} \quad \forall t \tag{7}$$

where $TRVw_{tj}$ denotes the repair cost in warehouse j in period t.

$$TTF_t = \sum_{j \in SC} TTFw_{tj} + TTFc_{tT} \quad \forall t \tag{8}$$

where $TTFw_{tj}$ denotes the transportation cost from factories to warehouse j in period t, and $TTFc_{tT}$ denotes the transportation cost from factories to customer sites in period t.

$$TTW_t = \sum_{j \in SC} TTWw_{tj} \quad \forall t, \quad \forall t \tag{9}$$

where $TTFw_{tj}$ denotes the transportation cost from warehouse j in period t.

$$TPW_t = \sum_j TPWw_{tj} \quad \forall t \tag{10}$$

where $TPWw_{tj}$ denotes the mean inventory cost in warehouse j in period t.

$$TPC_t = \sum_j TPCw_{tj} + TPCc_{tT} \quad \forall t \tag{11}$$

where $TPCw_{tj}$ and $TPCc_{tT}$ denote the mean inventory cost at customer sites for the special motors from warehouse j and tailor made motors in period t, respectively.

$$\begin{aligned} TSS_t &= \sum_j \sum_p h1_{jp} \cdot ss_{jpt} + \sum_j \sum_k \sum_{s \notin KT_{KS}} \sum_{C \in KSC_{KSC}} h2_k \cdot \lambda 2_{ks} \cdot \sigma_{ksct} \cdot \sqrt{l'_{jksct}} + \sum_k \sum_{s \in KT_{KS}} \sum_{C \in KSC_{KSC}} h2_k \cdot \lambda 2_{ks} \cdot \sigma_{ksct} \cdot \sqrt{l_{ksct}} \\ &= \sum_j TSSw_{tj} + TSSc_{tT} \end{aligned} \tag{12}$$

where $TSSw_{tj}$ denotes the summation of the safety stock cost at warehouse j and customer sites for the special motors from warehouse j in period t, and $TSSc_{tT}$ denotes the safety stock cost at customer sites for tailor made motors in period t.

$$TBT_t = \sum_j \sum_k \sum_{S \notin KT_{ks}} \sum_{c \in KSC_{ksc}} b1_{ks} \cdot 0.45 \cdot \sigma_{ksct} \cdot \sqrt{l'_{jksct}} \cdot e^{\lambda 2_{ks}/-0.59} \cdot \chi \frac{z_{jkt}}{t2_{jks}} = \sum_j TBTw_{tj} \tag{13}$$

where $TBTw_{tj}$ denotes the lost sales stock cost for special motors from warehouse j in period t.

Eqs. (55), (57), (59) and (61) from Part I are not included since the factories are assumed to be given. We therefore include Eqs. (1)–(13) above and (71) from part I in the objective function.

Also, constraint (30) from part I can be rewritten as follows.

$$l'_{jksct} \geq s_{jpt} \cdot z_{jkt} + t2_{jkp} \cdot z_{jkt} - R_{ksc} \tag{14}$$

We consider Eq. (14) above and Eqs. (10)–(15), (18), (19), (24)–(29), (31)–(41), (52) and (53) from part I (Rodriguez et al., 2013) as the constraints of the reformulated MINLP model.

3. Lagrangean decomposition algorithm

3.1. Lagrangean decomposition steps

Based on the reformulation of the model, we decompose the problem by warehouses. This requires dualizing constraints (8) and (9) from Part I, as they couple the different warehouses by specifying that the summation of warehouses assigned to a certain customer not exceed one. Considering that the demand constraints and factory capacity constraints also couple the different warehouses, constraints (20)–(23) and (53) from Part I are also dualized.

Hence, the Lagrangean function is as follows.

$$
\begin{aligned}
L = {} & \frac{\sum_t TI_t + TOF_t + TE_t + TU_t + TOV_t + TPV_t + TR_t + TTF_t + TTW_t + TPW_t + TPC_t + TSS_t + TBT_t + TBS_t}{(1+ir)^t} \\
& + \sum_{kt}\left[\lambda z_{kt}\left(\sum_j z_{jkt} - 1\right)\right] + \sum_{skt}\left[\lambda \upsilon_{skt}\left(\sum_j \upsilon_{jkst} - 1\right)\right] + \sum_{kpt}\left[\lambda muct_{kpt}\left(\sum_i\sum_j \mu^{\text{new}}_{ijkpt}\right.\right. \\
& \left.\left. + \sum_{j\in sc}\mu^{\text{used}}_{jkpt} - \sum_{\substack{s\in ps_{ps}\\ s\in CT_{ks}}}\sum_{c\in ksc_{ksc}}\mu_{ksct}\right)\right] + \sum_{kpt}\left[\lambda muct_{kpt}\left(\sum_i\sum_j \mu^{\text{new}}_{ijkpt} - \sum_{\substack{s\in PS_{ps}\\ s\notin CT_{ks}}}\sum_{c\in KSC_{ksc}}\mu_{ksct}\right)\right] \\
& + \sum_{t,(k,s)\in KT_{ks}}\left[\lambda tol_{kst}\left(\sum_i \tau^{\text{new}}_{ikst} + \sum_{j\in SC}\tau^{\text{used}}_{jkst} - \sum_{C\in KSC_{ksc}}\mu_{ksct}\right)\right] \\
& + \sum_{t,(k,s)\notin KT_{ks}}\left[\lambda tol_{kst}\left(\sum_i \tau^{\text{new}}_{ikst} - \sum_{C\in ksc_{ksc}}\mu_{ksct}\right)\right] \\
& + \sum_{it}\left[\lambda c\left(\sum_j\sum_k\sum_p \mu^{\text{new}}_{ijkpt}\cdot\alpha_p - qf_{it}\right)\right] \underline{\underline{\Delta}} f(\lambda z_{kt}, \lambda\upsilon_{skt}, \lambda muct_{kpt}, \lambda tol_{ks}, \lambda c)
\end{aligned}
\tag{15}
$$

where $\lambda z_{kt} \geq 0$, $\lambda \upsilon_{skt} \geq 0$, $\lambda muct_{kpt}$, λtol_{kst} and $\lambda c \geq 0$ are the corresponding Lagrange multipliers.

According to Eqs. (1)–(13) from Part II above and (71) from Part I, Eq. (15) can be rewritten as follows

$$
\begin{aligned}
L = {} & \sum_j\left[\sum_t \frac{TIw_{tj} + TOFw_{tj} + TEw_{tj} + TUw_{tj} + TOVw_{tj} + TPVw_{tj} + TRw_{tj} + TTFw_{tj} + TTWw_{tj} + TPCw_{tj} + TSSw_{tj} + TBTw_{tj}}{(1+ir)^t}\right] \\
& + \sum_t \frac{(TTFc_{tT} + TSSc_{tT} + TPCc_{tT} + TBSc_t)}{(1+ir)^t} + \sum_j\sum_{kt}\lambda z_{kt} z_{jkt} - \sum_{kt}\lambda z_{kt} + \sum_j\sum_{skt}\lambda\upsilon_{skt}\upsilon_{jkst} - \sum_{skt}\lambda\upsilon_{skt} \\
& + \sum_j\sum_{kpt}\left[\lambda muct_{kpt}\left(\sum_i \mu^{\text{new}}_{ijkpt} + \mu^{\text{used}}_{jkpt}\big|_{j\in SC}\right)\right] - \sum_{kpt}\left[\lambda muct_{kpt}\left(\sum_{\substack{s\in PS_{ps}\\ s\in CT_{ks}}}\sum_{c\in KSC_{ksc}}\mu_{ksct}\right)\right] \\
& + \sum_j\sum_{kpt}\left(\lambda muct_{kpt}\sum_i \mu^{\text{new}}_{ijkpt}\right) - \sum_{kpt}\left(\lambda muct_{kpt}\sum_{\substack{s\in PS_{ps}\\ s\notin CT_{ks}}}\sum_{c\in KSC_{ksc}}\mu_{ksct}\right) + \sum_{j\in SC}\sum_{t,(k,s)\in KT_{ks}}\lambda tol_{kst}\,\tau^{\text{used}}_{jkst} \\
& + \sum_{t,(k,s)\in KT_{ks}}\left[\lambda tol_{kst}\left(\sum_i \tau^{\text{new}}_{ikst} - \sum_{c\in KSC_{ksc}}\mu_{ksct}\right)\right] + \sum_{t,(k,s)\notin KT_{ks}}\left[\lambda tol_{kst}\left(\sum_i \tau^{\text{new}}_{ikst} - \sum_{c\in KSC_{ksc}}\mu_{ksct}\right)\right] \\
& + \sum_j\sum_{it}\left[\lambda c\left(\sum_k\sum_p \mu^{\text{new}}_{ijkpt}\cdot\alpha_p\right)\right] - \sum_{it}\lambda c\, QP^{UP}_i
\end{aligned}
\tag{16}
$$

Defining for each warehouse j

$$
\begin{aligned}
L_j = {} & \left[\sum_t \frac{TIw_{tj} + TOFw_{tj} + TEw_{tj} + TUw_{tj} + TOVw_{tj} + TPVw_{tj} + TRw_{tj} + TTFw_{tj} + TTWw_{tj} + TPCw_{tj} + TSSw_{tj} + TBTw_{tj}}{(1+ir)^t}\right] \\
& + \sum_{kt}\lambda z_{kt} z_{jkt} + \sum_{skt}\lambda\upsilon_{skt}\upsilon_{jkst} + \sum_{kpt}\left[\lambda muct_{kpt}\left(\sum_i \mu^{\text{new}}_{ijkpt} + \mu^{\text{used}}_{jkpt}\big|_{j\in SC}\right)\right] + \sum_{kpt}\left(\lambda muct_{kpt}\sum_i \mu^{\text{new}}_{ijkpt}\right) \\
& + \sum_{t,(k,s)\in KT_{ks}}\lambda tol_{kst}\,\tau^{\text{used}}_{jkst}\big|_{j\in SC} + \sum_{it}\left[\lambda c\left(\sum_k\sum_p \mu^{\text{new}}_{ijkpt}\cdot\alpha_p\right)\right] \quad \forall j\in J
\end{aligned}
\tag{17}
$$

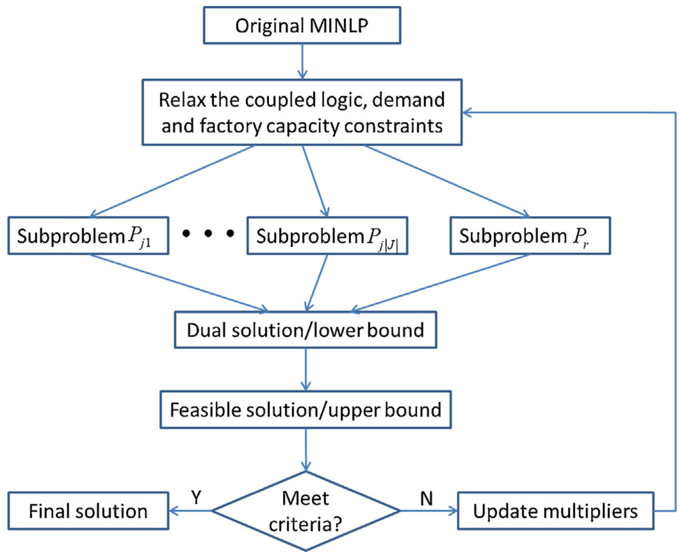

Fig. 1. Steps of Lagrangean decomposition algorithm.

and defining for the remaining cost terms (transportation cost for tailor made motors from factories to customers, safety stock cost at customer sites for tailor made motors, the mean inventory cost at customer sites for tailor made motors and lost sales cost for tailor made motors)and the penalty terms of the Lagrange multipliers except the ones involving variables z_{jkt}, v_{jkst}, μ^{new}_{ijkpt}, μ^{used}_{jkpt} and τ^{used}_{jkst}.

$$L_r = \sum_t \frac{(TTFc_{tT} + TSSc_{tT} + TPCc_t + TBSc_t)}{(1+ir)^t} - \sum_{kt} \lambda z_{kt} - \sum_{skt} \lambda v_{skt} - \sum_{kpt} \left[\lambda muct_{kpt} \left(\sum_{\substack{s \in PS_{ps} \\ s \in CT_{ks}}} \sum_{c \in KSC_{ksc}} \mu_{ksct} \right) \right]$$

$$- \sum_{kpt} \left(\lambda muct_{kpt} \sum_{\substack{s \in PS_{ps} \\ s \notin CT_{ks}}} \sum_{c \in KSC_{ksc}} \mu_{ksct} \right) + \sum_{t,(k,s) \in KT_{ks}} \left[\lambda tol_{kst} \left(\sum_i \tau^{new}_{ikst} - \sum_{c \in KSC_{ksc}} \mu_{ksct} \right) \right]$$

$$+ \sum_{t,(k,s) \notin KT_{ks}} \left[\lambda tol_{kst} \left(\sum_i \tau^{new}_{ikst} - \sum_{c \in KSC_{ksc}} \mu_{ksct} \right) \right] - \sum_{it} \lambda cQP^{UP}_i \tag{18}$$

Then,

$$L = \sum_j L_j + L_r \tag{19}$$

Thus, the Lagrangean function is decomposed into $|J| + 1$ terms as $L_j, \quad \forall j \in J$ and L_r. We can obtain the subproblems denoted by $P_j, \quad \forall j \in J$ and P_r, as follows.

$P_j: \quad \min \quad L_j$

subject to (11), (12), (14), (15), (18), (19), (24), (26)–(28), (31)–(41), (52) from Part I and (14) from this Part for given j.

$P_j: \quad \min \quad L_r$

subject to (10), (13), (25), (29) and (53) from Part I.

Using Lagrangean decomposition, the subproblems can be solved individually for a given set of Lagrange multipliers. The summation of the objective values then provides a lower bound of the primal problem, and the multipliers can be updated according to the solutions of the subproblems. The steps of the algorithm are shown in Fig. 1.

3.2. Solving the subproblems

Each subproblem $|J| + 1$ or $L_j, \quad \forall j \in J$ is also a large scale MINLP with nonlinear terms in the objective given by the square roots with positive coefficients. Considering a piecewise linear approximation of the square roots as shown in Fig. 2, each of subproblems reduces to an MILP that provides a lower bound to the MINLP.

For a square root term, $y = \sqrt{x}$, with x^L and x^U as the lower and upper bounds of continuous variable x respectively, we consider a temporal point x^t, with which the piecewise linear approximation is given by Eqs. (20)–(22)

$$x = \gamma_1 x^t + \beta_I \gamma_2 x^L + (1 - \beta_I) \gamma_2 x^U \tag{20}$$

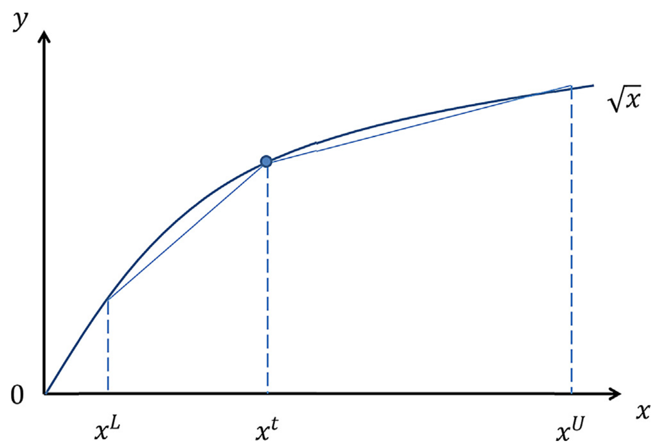

Fig. 2. Piecewise linear approximation of the nonlinear term of the subproblems.

$$\gamma_1 + \gamma_2 = 1 \tag{21}$$

$$y = \gamma_1\sqrt{x^t} + \beta_I\gamma_2\sqrt{x^L} + (1-\beta_I)\gamma_2\sqrt{x^U} \tag{22}$$

where γ_1 and γ_2 are positive continuous variables, β_I is a binary variable that indicates whether x lies between x^L and x^t.

There are bilinear terms $\beta_I\gamma_2$ in Eqs. (20) and (22). We introduce positive continuous variables a_I and a_I' as auxiliary variables, and constraints (23)–(25) as follows.

$$a_I + a_I' = \gamma_2 \tag{23}$$

$$a_I \le \beta_I \tag{24}$$

$$a_I' \le 1 - \beta_I \tag{25}$$

Then, $a_I = \beta_I\gamma_2$. Hence, Eqs. (20) and (22) can be rewritten as Eqs. (26) and (27).

$$x = \gamma_1 x^t + a_I x^L + \gamma_2 x^U - a_I x^U \tag{26}$$

$$y = \gamma_1\sqrt{x^t} + a_I\sqrt{x^L} + \gamma_2\sqrt{x^U} - a_I\sqrt{x^U} \tag{27}$$

Thus, the square root term $y = \sqrt{x}$ can be approximated with the linear Eqs. (21), (23)–(27).

We adopt an adaptive scheme to update the temporal point. In the first iteration, we take $x^t = (x^L + x^U/2)$, and in the following iterations, x^t is assigned with the solutions of the previous iteration. This is because the solutions of the previous iteration are close to the real solutions, and the linear approximation with the temporal point the original upper and lower bounds is relatively accurate near the temporal point. Then, we can expect that x^t's will converge to the optimal solution of the primal problem.

Furthermore, considering that the approximate problems are MILP problems with 0–1 binary and continuous variables, we consider an LP relaxation that can provide a lower bound to the MILP, which in turn is a lower bound to the original MINLP.

3.3. Feasibility scheme

A feasible solution is necessary to update the upper bound of the primal problem and provide a candidate solution. With the current Lagrange multipliers, we can obtain the solutions of the subproblems. But in general, the solutions are not feasible for the primal problem, especially constraints (8) and (9) from Part I are violated, and the value of z_{jkt} and v_{jkst} may not be integer. Therefore, to construct a feasible solution, we specify $z_{j'kt}$ and $v_{j''kst}$ with Algorithm *Specify*.Algorithm Specify

Start;
$NoOneU(k, t) = 1;$
$NoOneV(k, sp, t) = 1;$
$while(j \in J,$
$if(z(j, k, t) = \max_{j'} z(j', k, t) and\ NoOneU(k, t) = 1) = 1, then\ specify\ z(j, k, t)\ with\ 1;$
$if\ z.l(j, k, t),\ then\ NoOneU(k, t) = 0$
$if(vd.l(j, k, sp, t) = \max_{j'} vd.l(j', k, sp, t) and\ NoOneU(k, , sp, t) = 1),$
$then\ specify\ v(j, k, sp, t) with\ 1;$
$NoOneV(k, t)\$v.l(j, k, sp, t) = 0$
);
End;

In the algorithm, the initial values of z and v come from the corresponding solutions of the subproblems. The algorithm means that we specify z_{j^1kt} and v_{j^2kst} with 1 for $j^1 = \arg\max_j(z_{jkt})$ and $j^2 = \arg\max_j(v_{jkst})$ (if j^1 or j^2 is not unique, we take the first one by increasing order), and specify z_{jkt} $(j \ne j^1)$ and v_{jkst} $(j \ne j^2)$ with 0.

When z_{jkt} and v_{jkst} are specified with algorithm *Specify*, the feasibility problem reduces to an MINLP with binary variables, including x_{ijpt} and y_{jt}, y^e_{jt}, y^u_{jt}, and continuous variables. The nonlinear terms involve square root functions. To design an efficient feasibility scheme, we design another adaptive linear approximation scheme, which is shown in Fig. 3 with x^t_1 and x^t_2 updated iteratively.

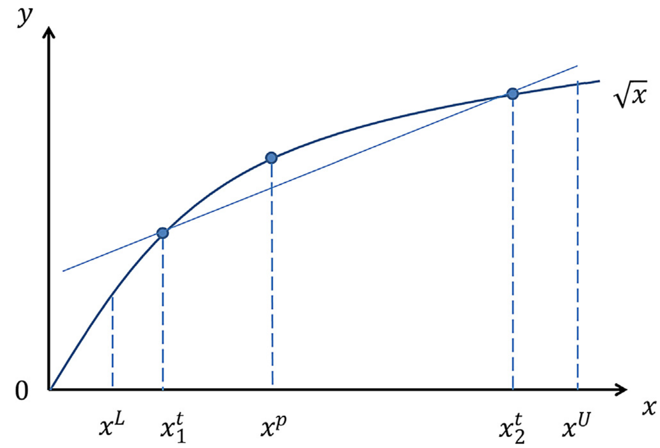

Fig. 3. Linear approximation of the nonlinear term of the feasible problem.

For a square root term, $y=\sqrt{x}$, with x^{L} and x^{U} as the lower and upper bounds of continuous variable x, respectively, we consider a pair of temporal points x_1^t and x_2^t. Then, the linear approximation is given by Eq. (28).

$$y=\frac{\sqrt{x_2^t}-\sqrt{x_1^t}}{x_2^t-x_1^t}(x-x_1^t)+\sqrt{x_1^t} \tag{28}$$

The pair of x_1^t and x_2^t are updated adaptively as in Eqs. (29) and (30).

$$\bar{x}_1^t=x_1^t+a(x^p-x_1^t) \tag{29}$$

$$\bar{x}_2^t=x_2^t+a(x^p-x_2^t) \tag{30}$$

where $\bar{x}_1^t$ and $\bar{x}_2^t$ are the updated points, x^p is the solution of the previous iteration, a is a scalar in (0,1), and we specify a with 0.05 in our algorithm. Thus, we can also expect that x_1^t and x_2^t will converge to the optimal solution from opposite directions. Hence, we obtain an MILP problem, denoted MILPFeas as an approximate feasibility problem. By solving MILPFeas, we can obtain a near optimal solution, according to which we calculate the exact objective with the original nonlinear function, which corresponds to an upper bound to the primal problem.

Here, for the feasibility problem, we use two temporal points to approximate the nonlinear terms, while we use only one temporal point to approximate the linear terms for the subproblems. The reason is that we have found from numerical experience that the solutions of the previous iteration are close to the real solutions. However, for the subproblems, two linear sections are needed to better approximate the nonlinear term.

3.4. Multipliers initialization and update

The appropriate multipliers of the demand constraints can help the solution of the subproblems to meet the demand. Furthermore, the Lagrange multipliers can be interpreted as the price of the products. Considering the Lagrangean function (15), if the demand is not met, a penalty will be incurred with the corresponding multipliers. At the same time, when the customer order is satisfied, the company has to incur in production cost, transportation cost, stock and repairing cost. Therefore, when the penalty and the cost reach a balance, the demand will be satisfied. In this way, we can estimate the multipliers of the demand constraints by calculating the unit cost of a feasible solution of the primal problem. Practically, we specify z_{jkt} and v_{jkst} arbitrarily to satisfy constraints (8) and (9) from Part I, and solve the feasibility problem. According to the solution, for each tuple (k, p, t) and (k, s, t), we calculate a total cost involving the corresponding terms in Eqs. (62)–(69) from Part I as Eqs. (31) and (32), then divide the cost by the corresponding demand. This is the average variable cost of the corresponding motor to meet the demand, the opposite of which we take as the initial value of the corresponding multiplier. The other multipliers are specified with 0 as the initial value.

$$\lambda muct_{kp0}=-\frac{\begin{pmatrix}\sum_i\sum_j\sum_p g_j\cdot\mu_{ijkpt}^{\mathrm{new}}\cdot\chi+\sum_i\sum_j\sum_p gp_i\cdot\mu_{ijkpt}^{\mathrm{new}}\cdot\chi+\sum_{j\in SC}\sum_p gr_{jp}\cdot\mu_{jkpt}^{\mathrm{used}}\cdot\chi+\sum_p\sum_i\sum_{j\in SC}\sum_p c1_{ij}\cdot\mu_{ijkpt}^{\mathrm{new}}\cdot\chi\\ +\sum_i\sum_j\sum_p c2_{jk}\cdot\mu_{ijkpt}^{\mathrm{new}}\cdot\chi+\sum_{j\in SC}2\cdot c2_{jk}\cdot\chi\cdot\sum_p\mu_{jkpt}^{\mathrm{used}}+\sum_i\sum_j\sum_p\theta1_{jp}\cdot\mu_{ijkpt}^{\mathrm{new}}\cdot t1_{ijp}\\ +\sum_i\sum_j\sum_p\theta2_{kp}\cdot\mu_{ijkpt}^{\mathrm{new}}\cdot t2_{jkp}+\sum_j\sum_p h1_{jp}\cdot ss_{jpt}+\sum_{s\notin KT_{ks}}\sum_{cnKSC_{ksc}}h2_k\cdot\lambda2_{ks}\cdot\sigma_{ksct}\cdot\sqrt{l_{ksct}}\end{pmatrix}}{\left(\sum_i\sum_j\sum_p\mu_{ijkpt}^{\mathrm{new}}+\sum_{jnSC}\sum_p\mu_{jkpt}^{\mathrm{used}}\right)} \tag{31}$$

$$\lambda tol_{kst0}=-\frac{\left(\sum_{j\in SC}\sum_{s\in KT_{ks}}gr'_{js}\cdot\tau_{jkst}^{\mathrm{used}}\cdot\chi+\sum_i\sum_k\sum_{s\in KT_{ks}}c3_{ik}\cdot\tau_{ikst}^{\mathrm{new}}\cdot\chi+\sum_{j\in SC}\sum_k 2\cdot c2_{jk}\cdot\chi\cdot\sum_{s\in KT_{ks}}\tau_{jkst}^{\mathrm{used}}+\sum_k\sum_{s\in KT_{ks}}\sum_{c\in KSC_{ksc}}h2_k\cdot\lambda2_{ks}\cdot\sigma_{ksct}\cdot\sqrt{m_{ksct}}\right)}{\left(\sum_i\tau_{ikst}^{\mathrm{new}}+\sum_{j\in SC}\tau_{jkst}^{\mathrm{used}}\right)} \tag{32}$$

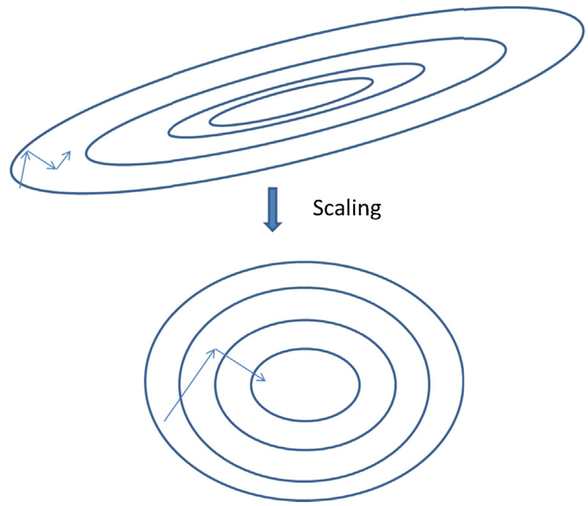

Fig. 4. Contours of the dual problem.

The subgradient optimization is a common method for updating the set of multipliers for the Lagrangean relaxation (Baker and Sheasby, 1999). In our problem, a scaling scheme is applied based on the fact that the multipliers of the demand constraints are equivalent to a unit cost. Furthermore, there are a large number of multipliers and their values range from the hundreds to the thousands. However, the subgradients of constraint (8) and (9) from Part I cannot exceed the number of warehouses (only 5 warehouses considered in our real world cases), and the corresponding multipliers are of the same order. This fact tends to make the contours of the dual problem long and narrow, as illustrated in Fig. 4, making the dual problem hard to converge. To overcome the problem, we scale all the multipliers to be the same order of magnitude by scaling so that the contours become near circles. The scaling scheme is as follows.

Recall that $L = f(\lambda z_{kt}, \lambda v_{skt}, \lambda mu_{ctkpt}, \lambda tol_{kst}, \lambda c)$, and the subgradients of f is $g = (gz_{kt}^T, gv_{skt}^T, gmuct_{kpt}^T, gtol_{ks}^T, gc^T)^T$.

Let $\lambda z_{kt} = \alpha_z \lambda z'_{kt}$, $\lambda v_{skt} = \alpha_v \lambda v'_{skt}$, $\lambda muct_{kpt} = \alpha_{mu} \lambda muct'_{kpt}$, $\lambda tol_{ks} = \alpha_t \lambda tol'_{ks}$ $\lambda c = \alpha_c \lambda c'$, where α_n's are positive scalars, we can rewrite the function as $L = f'(\lambda z'_{kt}, \lambda v'_{skt}, \lambda muct'_{kpt}, \lambda tol'_{ks}, \lambda c')$. Then the subgradients of f' is $g' = \left(\frac{1}{\alpha_z} gz_{kt}^T, \frac{1}{\alpha_v} gv_{skt}^T, \frac{1}{\alpha_{mu}} gmuct_{kpt}^T, \frac{1}{\alpha_t} gtol_{ks}^T, \frac{1}{\alpha_c} gc^T\right)^T$. To make all the multipliers of the same order, we specify α_v with 1, and α_z, α_{mu}, α_t, α_c with 10^x which are closest to the corresponding initial multipliers, where x is an integer, then the multipliers are scaled with Eqs. (33)–(37).

$$\lambda z'_{kt} = \frac{1}{\alpha_z} \lambda z_{kt} \tag{33}$$

$$\lambda v'_{skt} = \frac{1}{\alpha_v} \lambda v_{skt} \tag{34}$$

$$\lambda muct'_{kpt} = \frac{1}{\alpha_{mu}} \lambda muct_{kpt} \tag{35}$$

$$\lambda tol'_{ks} = \frac{1}{\alpha_t} \lambda tol_{ks} \tag{36}$$

$$\lambda c' = \frac{1}{\alpha_c} \lambda c \tag{37}$$

3.5. *Lagrangean decomposition algorithm*

In summary, the Lagrangean decomposition algorithm is as follows.

3.6. *Algorithm LD*

Step 1: Transform the original MINLP into an MILP, denoted *MILPWh*, by approximating the square root terms with piecewise linear ones according to Eqs. (21), (23)–(27), and relax all the binary variables to obtain an LP, denoted *LPWh*; Obtain subproblems P_j, $\forall j \in J$ and P_r of *LPWh* by relaxing constraints (8) and (9) from Part I;

Step 2: Transform the original MINLP into an MILP, denoted *MILPFeas*, by approximating the square root terms with a linear approximation according to Eq. (28);

Step 3: Specify z_{jkt} and v_{jkst} arbitrarily subject to constraints (8) and (9) from Part I, then solve *MILPWh*, and initialize the Lagrange multipliers $\lambda muct_{kp}$ and λtol_{kst} according to Eqs. (31) and (32) respectively, initialize the other Lagrange multipliers $\lambda z'_{kt}$, $\lambda v'_{skt}$, $\lambda muct'_{kpt}$, λ' to 0;

Step 4: Scale the Lagrange multipliers according to Eqs. (33)–(37);

Table 1
Size of the cases.

Component	Case 1	Case 2	Case 3
Factories	3	7	7
Warehouse candidates	7	5	5
Customers	27	27	27
Standard motors	32	32	99
Special motors	49	49	396
Criticality levels	4	4	4
Periods of time horizon	5	5	5

Table 2
Model statistics.

Item	Case 1	Case 2	Case 3
Number of constraints	326,151	430,707	1,471,287
Number of variables	190,414	236,853	807,713
Number of binary variables	14,444	16,339	47,654

Step 5: Solve subproblems $P_j, \quad \forall j \in J$ and P_r, obtain the dual objective l_{it} by summarizing the objectives of the subproblems, update the lower bound of the original MINLP with the summation of, denoted lup_{it}, where it denotes the iteration;

Step 6: Call *Algorithm Specify* to specify z_{jkt} and v_{jkst} according to the solutions of the subproblems, then solve *MILPFea*; Calculate the objective of the original MINLP according to Eqs. (1)–(13) from this second Part and (71) from part I using the solutions of *MILPFea*, update the upper bound of the original MINLP, denoted fup_{it} with it as the iteration;

Step 7: If the convergence criterion is satisfied, stop the algorithm; otherwise, update x^t in equations (26), (27) with the corresponding solutions of *MILPFea*, x^p in Eq. (28) using Eqs. (29) and (30), and update the scaled Lagrange multipliers using $(\lambda z'_{kt}, \lambda v'_{skt}, \lambda muct'_{kpt}, \lambda tol'_{ks}, \lambda c')_{it+1} = P_{+zvc}\left((\lambda z'_{kt}, \lambda v'_{skt}, \lambda muct'_{kpt}, \lambda tol'_{ks}, \lambda c')_{it+1} + (fup_{it} - l_{it})\frac{g'}{|g'|^2}\right)$, where $P_{+zvc}(.)$ means projection to the space with nonnegative $\lambda z'_{kt}, \lambda v'_{skt}, \lambda c'$, and go to step 5.

4. Results

The application of the proposed Lagrangean decomposition algorithm is shown in this section. All the cases are executed in GAMS 24.01 using a CPU Intel(R) Core(TM) i7 CPU 870@2.93GHz with RAM 12.0 Gb. We have run three cases for the supply chain described in Part I, with the number of the components in each echelon and the numbers of motors shown in Table 1. Furthermore, we assume that the number of factories is fixed, while for the warehouses we consider all the locations as potential ones whose capacities can be expanded or shutdown except that warehouse J1 is installed at the beginning of time horizon because it operates as a main warehouse. Case 1 is a relatively small illustrative problem, while cases 2 and 3 are based on real world industrial data. The model statistics of 3 single MINLP problems are shown in Table 2.Each of the models is solved with 5 algorithms. First, the model is solved with DICOPT as a single MINLP problem (MINLP). Then, the nonlinear terms are approximated by piecewise linearizations with 2 and 5 intervals respectively (MILP-2, MILP-5), and the MILP problem is solved by CPLEX. Next, the MINLP problem is solved by the approach proposed in Part I (AltNLPMILP). Finally, the problem is solved by the Lagrangean decomposition algorithm proposed in this Part II paper (LD).

Case 1. The objective values and CPU times required are shown in Table 3.

The iteration details of the LD results are shown in Figs. 5 and 6.

In Fig. 5, it can be seen that the gap between the lower and upper bound is reduced to 0.003% at iteration 7, and the CPU time required is about 130 s. In Table 3 it can be seen that the MINLP model cannot obtain any feasible solution after more than 3 h. The CPU times required by the MILP models increase quickly as the number of intervals grows, but the accuracy cannot be improved. The CPU time of AltNLPMILP is about 2 thirds as large as the one for MILP-2 with a slightly higher error (0.31% vs. 0.24%). The Lagrangean decomposition algorithm reaches the optimal solution faster in about 130 s. The convergence of several multipliers is illustrated in Fig. 6.

The capacity profiles of the 3 selected warehouses (J1, J3 and J7) are shown in Table 4. The cost details of the optimal solution obtained with the Lagrangean decomposition algorithm are shown in Figs. 7–10.

The variable costs of warehouses for new motors are illustrated in Fig. 7, where the x-axis indicates time period, while y-axis indicates variable costs in dollars. The stock cost, safety stock cost and repair cost of warehouses are illustrated in Figs. 8–10, respectively.

Case 2. The objective values and CPU times required for case 2 are shown in Table 5. The convergence of several multipliers is shown in Fig. 13

Table 3
Objective values and CPU time required by different algorithms for Case 1.

Name	Optimal objective ($)	Error (%)	CPU time (s)
MINLP	No feasible solution	-	12,159.84
MILP-2	5,747,911.87	0.24	683.67
MILP-5	5,748,118.91	0.24	10,086.24
AltNLPMILP	5,752,005.04	0.31	460.02
LD (30 iterations)	5,733,962.14	0	551.92
LD (7 iterations)	5,733,962.14	0.003%	130 (estimated)

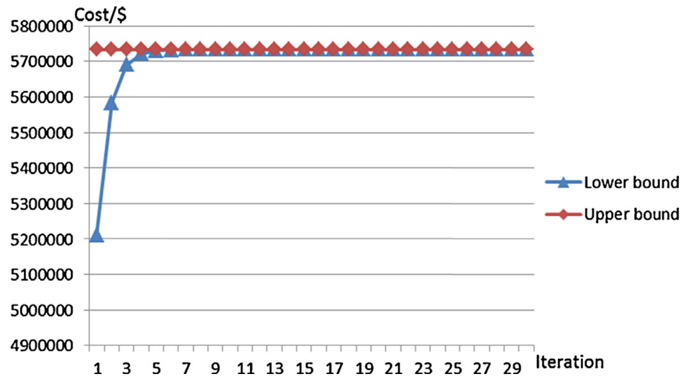

Fig. 5. Convergence of the lower and upper bound for Case 1.

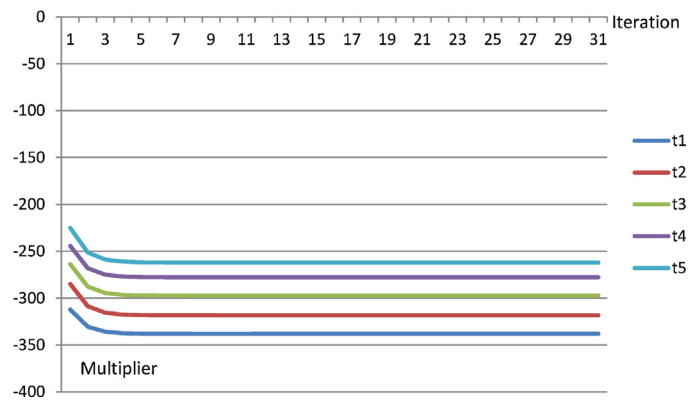

Fig. 6. Convergence of multipliers of demand constraints of motor $p1$ with criticality $k1$ at period t for Case 1.

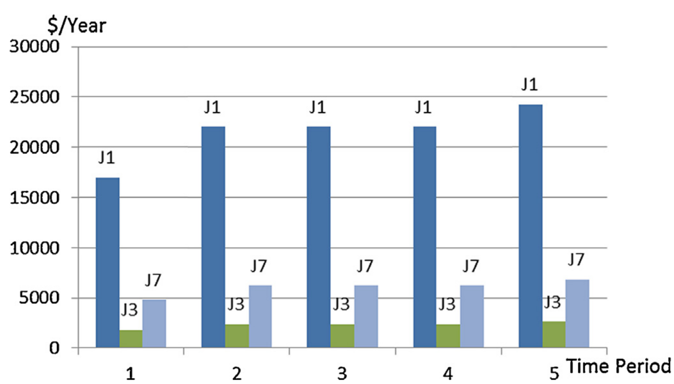

Fig. 7. Variable costs of warehouses for new motor for Case 1.

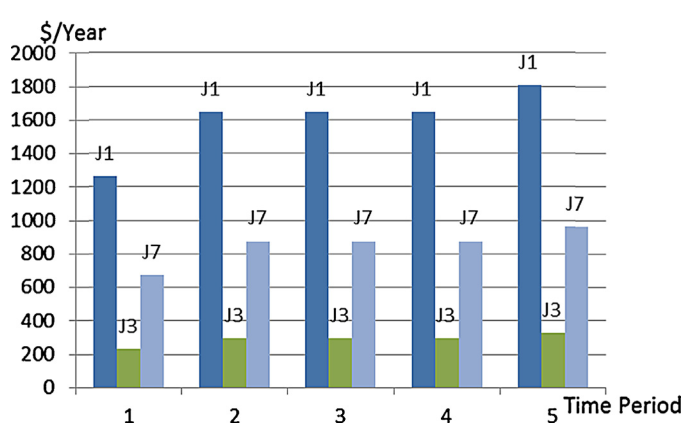

Fig. 8. Mean stock costs of warehouses for modifying for Case 1.

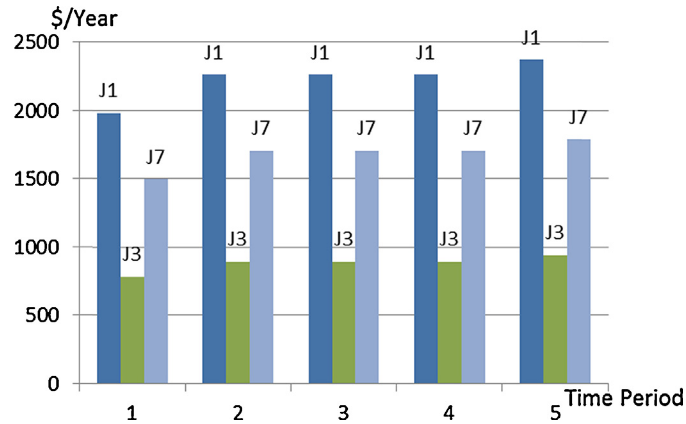

Fig. 9. Safety stock costs of warehouses for Case 1.

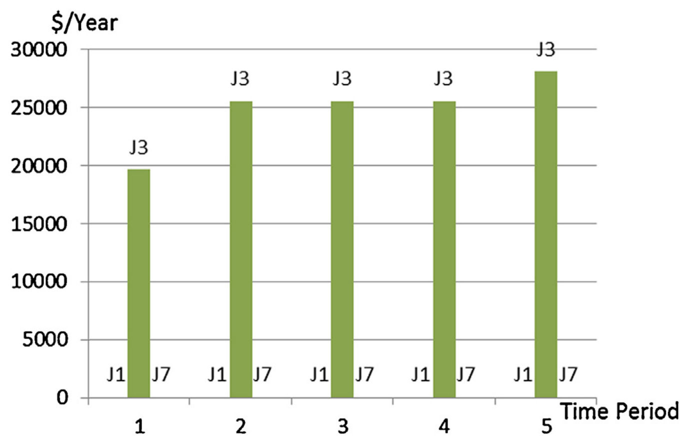

Fig. 10. Replace cost of warehouses for special motors for Case 1.

Table 4
Capacity profiles of warehouses for Case 1.

SKUs	Year 1	Year 2	Year 3	Year 4	Year 5
J1	2000	2000	2000	2000	2000
J3	40	40	40	40	40
J7	50	50	50	50	50

In this case, only J1, J3 and J7 are installed, where the initial capacity of J1 is much larger than the capacities of J3 and J7.

In Fig. 11, it can be seen that the gap between the lower and upper bound is reduced to 0.004% at iteration 8, and the estimated CPU time required is 162 s. There are similar trends as in case 1. The capacity profiles for case 2 of the 3 selected warehouses (J1, J2 and J3) are shown in Table 6 (Fig. 12).

Case 3. The objective values and CPU time required for case 3 are shown in Table 7. The convergence of several multipliers is shown in Fig. 14

In Fig. 12, it can be seen that the gap between the lower and upper bound is reduced to 0.003% at iteration 7, and the estimated CPU time required is 4696 s. There are similar trends as in case 1. Both MINLP and MILP-5 cannot find any feasible solution, and AltNLPMILP performs similar to MILP-2 with about twice the CPU time.

Table 5
Objective values and CPU time required by different algorithms for Case 2.

Name	Optimal objective ($)	Error (%)	CPU time (s)
MINLP	6,537,842.48	5.40	32,288.42
MILP-2	6,354,304.15	2.44	798.99
MILP-5	6,354,304.15	2.44	86,020.932
AltNLPMILP	6,358,672.25	2.51	530.11
LD (30 iterations)	6,202,732.33	0	607.58
LD (8 iterations)	6,202,732.33	0.004%	162 (estimated)

The iteration details of the LD results are shown in Figs. 11 and 12.

Table 6
Capacity profiles of warehouses for Case 2.

SKUs	Year 1	Year 2	Year 3	Year 4	Year 5
J1	50,000	50,000	50,000	50,000	50,000
J2	50	50	50	50	50
J3	50	50	50	50	50

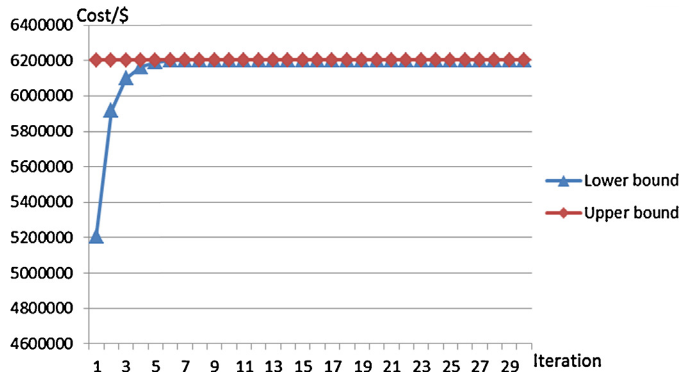

Fig. 11. Convergence of lower and upper bound for Case 2.

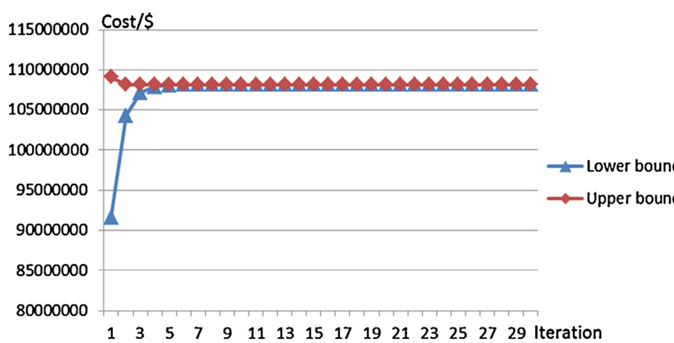

Fig. 12. Convergence of lower and upper bound for Case 3.

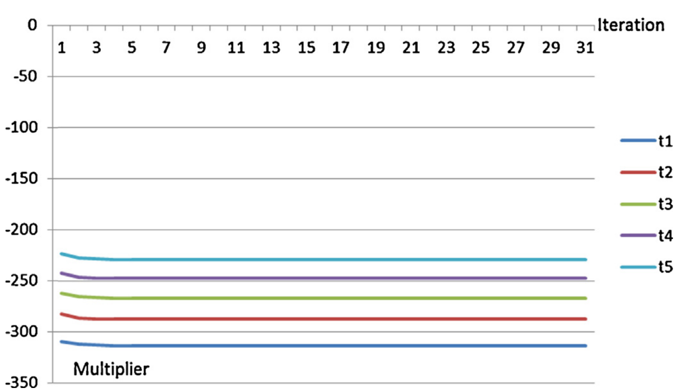

Fig. 13. Convergence of multipliers of demand constraints of motor $p1$ with criticality $k1$ at period t for Case 2.

Table 7
Objective values and CPU time required by different algorithms for Case 3.

Name	Optimal objective ($)	Error (%)	CPU time (s)
MINLP	No feasible solution	-	360,123.00
MILP-2	120,349,878.7579	11.25	10,871.29
MILP-5	No feasible solution	-	Out of memory
AltNLPMILP	120,657,481.7045	11.53	19,942.81
LD (30 iterations)	108,178,792.52	0	20,125.03
LD (7 iterations)	108,178,792.52	0.003%	4696 (estimated)

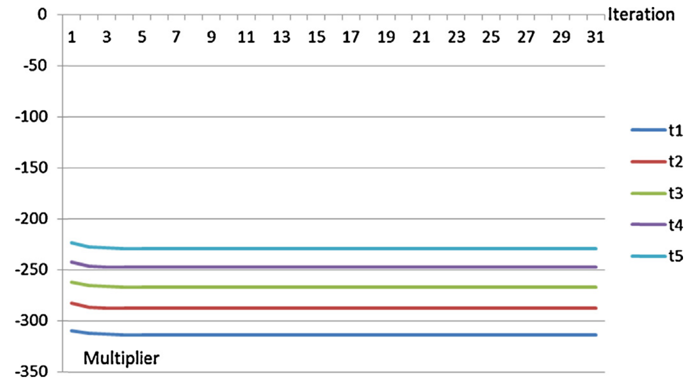

Fig. 14. Convergence of multipliers of demand constraints of motor $p1$ with criticality $k1$ at period t for Case 3.

From Figs. 5, 11 and 12, we can see that the optimal solutions are all obtained in the early iterations, namely iteration 1, 1, and 3 respectively. The reason is that due to the initialization step, the initial Lagrangean multipliers are close to the optimal one, therefore, the dualized binary variables obtained by the subproblems and algorithm *Specify* can reach their optimal values in the early iterations.

To summarize the three cases, we can conclude that the Lagrangean decomposition algorithm can obtain the optimal solution efficiently. The algorithm performs similarly on different scale cases, and as the problem scale increases, the advantage becomes more apparent, especially for highly constrained problems.

5. Conclusions

The supply chain of electric motor is complex due to many decisions, especially the reverse flows, which results in a large scale MINLP problem, whose number of variables and equations can range from thousands to millions. Therefore, the solution of this type of problem is a challenging task. Lagrangean decomposition is a popular method for large scale problems, but the decomposition scheme depends on the problem structure. In this paper, we decompose the problem by warehouses. Given that warehouses share the demands of customers and capacities of factories, the corresponding constraints have to be dualized simultaneously. As a consequence, there are a large number of Lagrange multipliers, which are quite different in scale. To accelerate the convergence, a scaling scheme has been proposed. Furthermore, considering that the multipliers can be interpreted in an economic sense, we design a method to estimate initial values for them. Another challenge for the decomposition method is that the sizes of the subproblems are still quite large involving nonlinear terms and binary variables. An adaptive piecewise linearization method is proposed to approximate the nonlinear terms. To obtain feasible solutions, another adaptive piece-wise linearization is also presented. The test results on illustrative and real world industrial problems show that the Lagrangean decomposition algorithm is effective and efficient, while the single MINLP is hard to solve and the MILP approximation is only computationally feasible with a few intervals. The AltNLPMILP of Part I performs similarly to the MILP approximation. The advantage of the proposed method is especially apparent for large scale and highly constrained problems. That is, if there are many motors to be dealt with in the supply chain and few potential warehouses to be selected.

Acknowledgments

The authors gratefully acknowledge financial support from ABB Corporation, the Center for Advanced Process Decision-MAKING, and from the National Natural Science Foundation of China (No. 61273039).

References

Baker, B. M., & Sheasby, J. (1999). Accelerating the convergence of subgradient optimization. *European Journal of Operational Research*, *117*, 136–144.
Buil, R., Piera, M. À., & Luh, P. B. (2012). Improvement of Lagrangian relaxation convergence for production scheduling. *IEEE Transactions on Automations Science and Engineering*, *9*(1), 137–147.
Fumero, F. (2001). A modified subgradient algorithm for Lagrangean relaxation. *Computers & Operations Research*, *28*, 33–52.
Rodriguez, M. A., Vecchietti, A. R., Harjunkoski, I., & Grossmann, I. E. (2013). Optimal supply chain design and management over a multi-period horizon under demand uncertainty. Part I: MINLP and MILP models. *Computers and Chemical Engineering* (in press).
Mouret, S., Grossmann, I. E., & Pestiaux, P. (2011). A new Lagrangian decomposition approach applied to the integration of refinery planning and crude-oil scheduling. *Computers and Chemical Engineering*, *35*, 2750–2766.
Terrazas-Moreno, S., Trotter, P. A., & Grossmann, I. E. (2011). Temporal and spatial Lagrangean decompositions in multi-site, multi-period production planning problems with sequence-dependent changeovers. *Computers and Chemical Engineering*, *35*, 2913–2928.
Wang, S.-H. (2003). An improved step size of subgradient algorithm for solving the Lagrangian relaxation problem. *Computers and Electrical Engineering*, *29*, 245–249.