6-2013

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

Jiang Yongheng
*Tsinghua University*

Maria Analia Rodriguez
*INGAR*

Iiro Harjunkoski
*ABB Corporate Research Center*

Ignacio E. Grossmann
*Carnegie Mellon University*, grossmann@cmu.edu



Published In
Computers and Chemical Engineering, 62, 211-224.

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

*Jiang Yongheng[1], Maria Analia Rodriguez[2],Iiro Harjunkoski[3] and Ignacio E. Grossmann[4]*

*[1]Institute of Process Control Eng., Department of Automation, Tsinghua University, Beijing, 100084, P.R. China*
*[2]INGAR (CONICET-UTN), Avellaneda 3657, Santa Fe 3000, Argentin*
*[3]ABB AG, Corporate Research Germany, Wallstadter Straße 59, 68526 Ladenburg, Germany*
*[4]Center for Advanced Process Decision-making, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15217, USA*

*Abstract*

In Part I (Rodriguez, et al., 2013 an optimization model was proposed to redesign the supply chain of spare parts industry under demand uncertainty in a specified planning horizon. To address large industrial problems, a Lagrangean scheme is proposed to decompose the MINLP of Part I according to the warehouses by dualizing the logic constraints that assign the warehouses to different customers, together with the demand constraints and factory capacity constraints. 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. 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.

Keywords: supply chain optimization, Lagrangean decomposition, adaptive piecewise linearization

## 1. Introduction

In the spare parts industry, or more specifically the electric motor industry as was illustrated in part 1 (Rodriguez et al, 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 many large-scale mathematical programming problems (**Wang, 2003**). According to the problem structure, temporal and spatial decomposition can be adopted (**Terrazas-Moreno, et al. 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 (**Buil, et al. 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 subproblems, initialize and update 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.

## 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 (Equations (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.

$$TI_t = \sum_j TI_{tj} \qquad \forall t \qquad (1)$$

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

$$TOF_t = \sum_j TOF_{tj} \qquad \forall t \qquad (2)$$

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

$$TE_t = \sum_j TE_{tj} \qquad \forall t \qquad (3)$$

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

$$TU_t = \sum_j TU_{tj} \ \forall t \qquad \forall t \qquad (4)$$

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

$$TOV_t = \sum_j TOV_{tj} \qquad \forall t \qquad (5)$$

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

$$TPV_t = \sum_j TPV_{tj} \qquad \forall t \qquad (6)$$

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

$$TR_t = \sum_{j \in SC} TR_{tj} \qquad \forall t \qquad (7)$$

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

$$TTF_t = \sum_{j \in SC} TTF_{tj} + TTF_{tT} \qquad \forall t \qquad (8)$$

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

$$TTW_t = \sum_{j \in SC} TTW_{tj} \, \forall t \qquad \forall t \qquad (9)$$

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

$$TPW_t = \sum_j TPW_{tj} \qquad \forall t \qquad (10)$$

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

$$TPC_t = \sum_j TPC_{tj} + TPC_{tT} \qquad \forall t \qquad (11)$$

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

$$TSS_t = \sum_j \sum_p h1_{jp} \cdot ss_{jpt} + \sum_j \sum_k \sum_{s \notin KT_{ks}} \sum_{c \in KSC_{ksc}} h2_k \cdot \lambda 2_{ks} \cdot \sigma_{ksct} \cdot \sqrt{l'_{jksct}} + \sum_k \sum_{s \in KT_{ks}} \sum_{c \in KSC_{ksc}} h2_k \cdot \lambda 2_{ks} \cdot \sigma_{ksct} \cdot \sqrt{m_{ksct}}$$

$$= \sum_j TSS_{tj} + TSS_{tT} \qquad (12)$$

where $TSS_{tj}$ 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 $TSS_{tT}$ denotes the safety stock cost at customer sites for tailor made motors in period $t$.

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

$$= \sum_j TBT_{tj} \qquad (13)$$

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

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

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

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

We consider equation (14) above and equations (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.

$$\begin{aligned}
&L \\
&= \sum_t \frac{TI_t + TOF_t + TE_t + TU_t + TOV_t + TPV_t + TR_t + TTF_t + TTW_t + TPW_t + TPC_t + TSS_t + TBT_t + TBS_t}{(1+ir)^t} \\
&+ \\
&\sum_{kt}\left[\lambda_{zkt}\left(\sum_j z_{jkt} - 1\right)\right] + \sum_{skt}\left[\lambda_{vskt}\left(\sum_j v_{jkst} - 1\right)\right] + \\
&\sum_{kpt}\left[\lambda_{muctkpt}\left(\sum_i \sum_j \mu^{new}_{ijkpt} + \sum_{j\in SC} \mu^{used}_{jkpt} - \sum_{\substack{s\in PS_{ps} \\ s\in CT_{ks}}} \sum_{c\in KSC_{ksc}} \mu_{ksct}\right)\right] + \\
&\sum_{kpt}\left[\lambda_{muctkpt}\left(\sum_i \sum_j \mu^{new}_{ijkpt} - \sum_{\substack{s\in PS_{ps} \\ s\notin CT_{ks}}} \sum_{c\in KSC_{ksc}} \mu_{ksct}\right)\right] + \\
&\sum_{t,(k,s)\in KT_{ks}}\left[\lambda_{tolkst}\left(\sum_i \tau^{new}_{ikst} + \sum_{j\in SC} \tau^{used}_{jkst} - \sum_{c\in KSC_{ksc}} \mu_{ksct}\right)\right] + \\
&\sum_{t,(k,s)\notin KT_{ks}}\left[\lambda_{tolkst}\left(\sum_i \tau^{new}_{ikst} - \sum_{c\in KSC_{ksc}} \mu_{ksct}\right)\right] + \\
&\sum_{it}\left[\lambda_c\left(\sum_j \sum_k \sum_p \mu^{new}_{ijkpt} \cdot \alpha_p - qf_{it}\right)\right] \\
&\triangleq f\left(\lambda_{zkt}, \lambda_{vskt}, \lambda_{muctkpt}, \lambda_{muctnkpt}, \lambda_{tolks}, \lambda_c\right)
\end{aligned} \quad (15)$$

where $\lambda_{zkt} \geq 0$, $\lambda_{vskt} \geq 0$, $\lambda_{muctkpt}$, $\lambda_{tolkst}$ and $\lambda_c \geq 0$ are the corresponding Lagrange multipliers.

According to equations (1)-(13) from Part II above and (71) from Part I, equation (15) can be rewritten as follows

$$L = \sum_j \left[\sum_t \frac{TI_{tj}+TOF_{tj}+TE_{tj}+TU_{tj}+TOV_{tj}+TPV_{tj}+TR_{tj}+TTF_{tj}+TTW_{tj}+TPC_{tj}+TSS_{tj}+TBT_{tj}}{(1+ir)^t}\right]$$

$$+\sum_t \frac{(TTF_{tT}+TSS_{tT}+TPC_{tT}+TBS_t)}{(1+ir)^t} +$$

$$\sum_j \sum_{kt} \lambda_{zkt} z_{jkt} - \sum_{kt} \lambda_{zkt} + \sum_j \sum_{skt} \lambda_{vskt} v_{jkst} - \sum_{skt} \lambda_{vskt} +$$

$$\sum_j \sum_{kpt} \left[\lambda_{muctkpt} \left(\sum_i \mu_{ijkpt}^{new} + \mu_{jkpt}^{used}\big|_{j\in SC}\right)\right] - \sum_{kpt} \left[\lambda_{muctkpt} \left(\sum_{\substack{s\in PS_{ps} \\ s\in CT_{ks}}} \sum_{c\in KSC_{ksc}} \mu_{ksct}\right)\right] +$$

$$\sum_j \sum_{kpt} \left(\lambda_{muctkpt} \sum_i \mu_{ijkpt}^{new}\right) - \sum_{kpt} \left(\lambda_{muctnkpt} \sum_{\substack{s\in PS_{ps} \\ s\notin CT_{ks}}} \sum_{c\in KSC_{ksc}} \mu_{ksct}\right) +$$

$$\sum_{j\in SC} \sum_{t,(k,s)\in KT_{ks}} \lambda_{tolkst} \tau_{jkst}^{used} + \sum_{t,(k,s)\in KT_{ks}} \left[\lambda_{tolkst} \left(\sum_i \tau_{ikst}^{new} - \sum_{c\in KSC_{ksc}} \mu_{ksct}\right)\right] +$$

$$\sum_{t,(k,s)\notin KT_{ks}} \left[\lambda_{tolkst} \left(\sum_i \tau_{ikst}^{new} - \sum_{c\in KSC_{ksc}} \mu_{ksct}\right)\right] +$$

$$\sum_j \sum_{it} \left[\lambda_c \left(\sum_k \sum_p \mu_{ijkpt}^{new} \cdot \alpha_p\right)\right] - \sum_{it} \lambda_c QP_i^{UP} \qquad (16)$$

Defining for each warehouse j

$$L_j = \left[\sum_t \frac{TI_{tj}+TOF_{tj}+TE_{tj}+TU_{tj}+TOV_{tj}+TPV_{tj}+TR_{tj}+TTF_{tj}+TTW_{tj}+TPC_{tj}+TSS_{tj}+TBT_{tj}}{(1+ir)^t}\right] +$$

$$\sum_{kt} \lambda_{zkt} z_{jkt} + \sum_{skt} \lambda_{vskt} v_{jkst} + \sum_{kpt} \left[\lambda_{muctkpt} \left(\sum_i \mu_{ijkpt}^{new} + \mu_{jkpt}^{used}\big|_{j\in SC}\right)\right] +$$

$$\sum_{kpt} \left(\lambda_{muctkpt} \sum_i \mu_{ijkpt}^{new}\right) + \sum_{t,(k,s)\in KT_{ks}} \lambda_{tolkst} \tau_{jkst}^{used} + \sum_{it} \left[\lambda_c \left(\sum_k \sum_p \mu_{ijkpt}^{new} \cdot \alpha_p\right)\right] \ \forall j \in J \qquad (17)$$

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 $z_{jkt}$ , $v_{jkst}$, $\mu_{ijkpt}^{new}$, $\mu_{jkpt}^{used}$ and $\tau_{jkst}^{used}$.

$$L_r = \sum_t \frac{(TTF_{tT}+TSS_{tT}+TPC_t+TBS_t)}{(1+ir)^t} - \sum_{kt} \lambda_{zkt} - \sum_{skt} \lambda_{vskt} - \sum_{kpt} \left[\lambda_{muctkpt} \left(\sum_{\substack{s\in PS_{ps} \\ s\in CT_{ks}}} \sum_{c\in KSC_{ksc}} \mu_{ksct}\right)\right] +$$

$$- \sum_{kpt} \left(\lambda_{muctnkpt} \sum_{\substack{s\in PS_{ps} \\ s\notin CT_{ks}}} \sum_{c\in KSC_{ksc}} \mu_{ksct}\right) + \sum_{t,(k,s)\in KT_{ks}} \left[\lambda_{tolkst} \left(\sum_i \tau_{ikst}^{new} - \sum_{c\in KSC_{ksc}} \mu_{ksct}\right)\right] +$$

$$\sum_{t,(k,s)\notin KT_{ks}} \left[\lambda_{tolkst} \left(\sum_i \tau_{ikst}^{new} - \sum_{c\in KSC_{ksc}} \mu_{ksct}\right)\right] - \sum_{it} \lambda_c QP_i^{UP} \qquad (18)$$

Then, $L = \sum_j L_j + L_r$ (19)

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

$$P_r: \quad \min \quad L_r$$

*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.

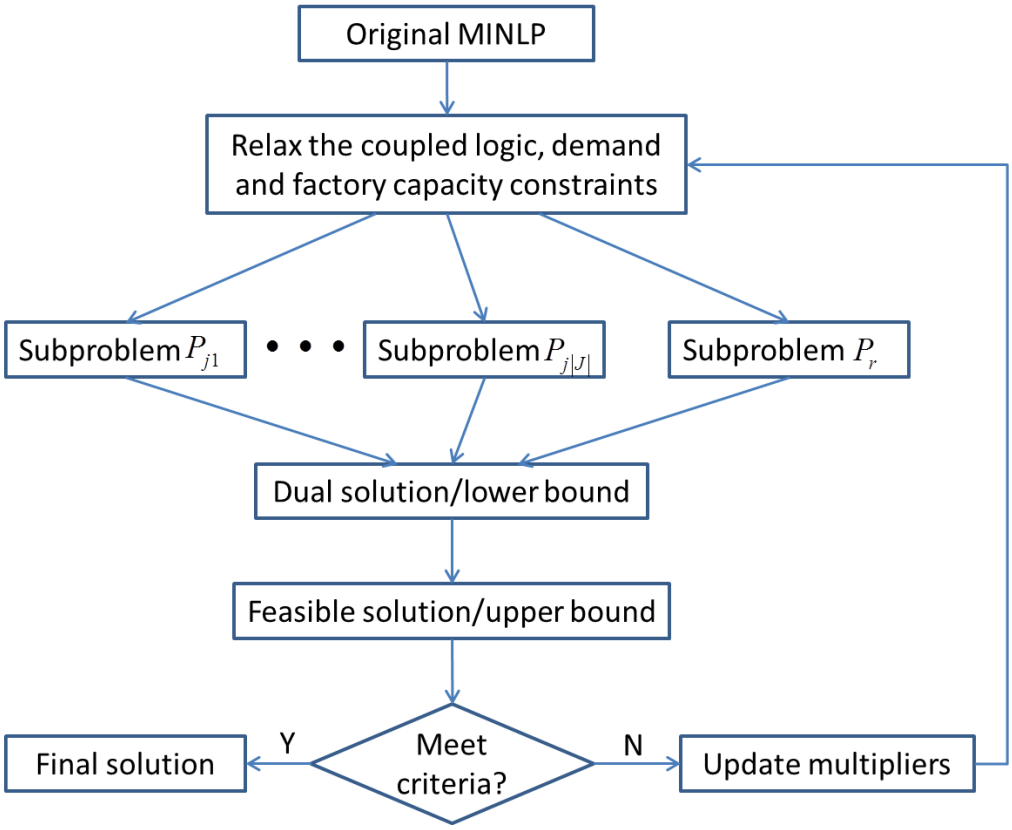


*Figure 1 Steps of Lagrangean decomposition algorithm*

### 3.2 Solving the subproblems

Each subproblem $L_j, j \in J$ or $L_r$ 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 equations (20)-(22)

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

$$\gamma_1 + \gamma_2 = 1 \quad (21)$$

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

where $\gamma_1$ and $\gamma_2$ are positive continuous variables, $\beta_I$ is a binary variable that indicates whether $x$ lies between $x^L$ and $x^t$.

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

$$a_I + a_I' = \gamma_2 \quad (23)$$

$$a_I \leq \beta_I \quad (24)$$

$$a_I' \leq 1 - \beta_I \quad (25)$$

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

$$x = \gamma_1 x^t + a_I x^L + \gamma_2 x^U - a_I x^U \quad (26)$$

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

Thus, the square root term $y = \sqrt{x}$ can be approximated with the linear equations (21), (26) and (27).

We adopt an adaptive scheme to update the temporal point. In the first iteration, we take $x^t = \frac{x^L + x^U}{2}$, and in the following iterations, $x^t$ is assigned with the solutions of the previous iteration. Then, we can expect that $x^{t\prime}$s will converge to the optimal solution of the primal problem.

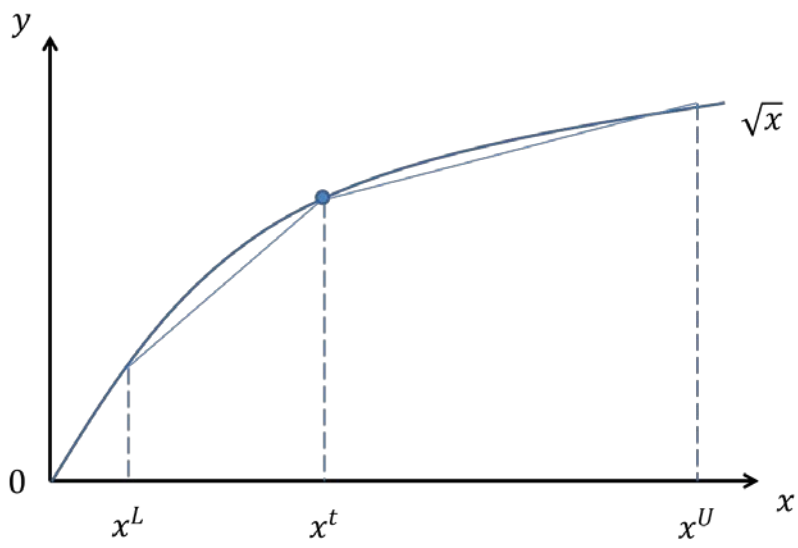


*Figure 2 Piecewise linear approximation of the nonlinear term of the subproblems*

Furthermore, considering that the approximate problems are MILP problem 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*;*
$NoOneU(k,t) = 1;$
$NoOneV(k,sp,t) = 1;$
$loop(j,$
$z.fx(j,k,t)\$\left(zd.l(j,k,t) = max_{j'} zd.l(j',k,t) and\ \ NoOneU(k,t)\right) = 1;$
$NoOneU(k,t)\$z.l(j,k,t) = 0;$
$v.fx(j,k,sp,t)\$\left(vd.l(j,k,sp,t) = max_{j'} vd.l(j',k,sp,t) and\ \ NoOneU(k,,sp,t)\right) = 1;$
$NoOneV(k,t)\$v.l(j,k,sp,t) = 0;$
);
End;

In the algorithm, $zd$ and $vd$ denote the corresponding solutions of the subproblems. The algorithm means that we specify $z_{j^1kt}$ and $v_{j^2kst}$ with 1 for $j^1 = argmax_j(z_{jkt})$ and $j^2 = argmax_j(v_{jkst})$ (if $j^1$ or $j^2$is not unique, we take the first one by increasing order), and specify $z_{jkt}$ $(j \neq j^1)$ and $v_{jkst}$ $(j \neq j^2)$ 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 $y_{jt}$, $y_{jt}^e$, $y_{jt}^u$, 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 $x_1^t$ and $x_2^t$ updated iteratively.

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 equation (28).

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

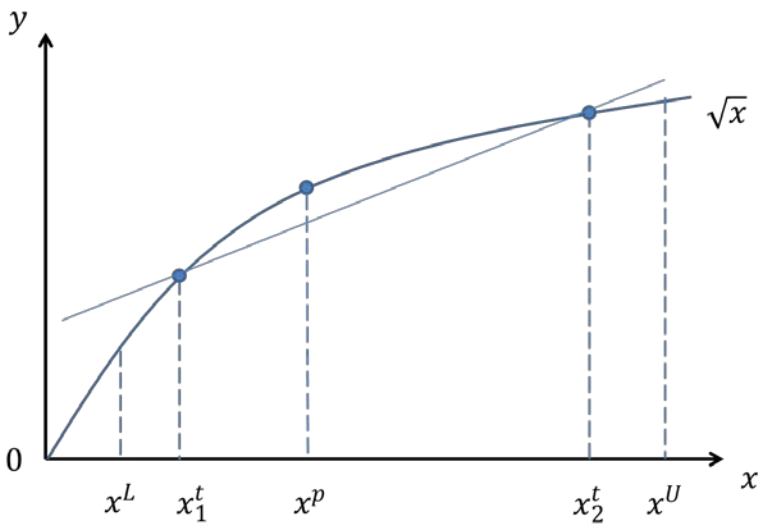


*Figure 3 Linear approximation of the nonlinear term of the feasible problem*

The pair of $x_1^t$ and $x_2^t$ are updated adaptively as in equations (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 $(0,1)$. 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.

### 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 equations(62)-(69) from Part I as equations (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.

$$\lambda_{muctkp0} = -\frac{\begin{pmatrix} \sum_i \sum_j \sum_p g_j \cdot \mu_{ijkpt}^{new} \cdot \chi + \sum_i \sum_j \sum_p gp_i \cdot \mu_{ijkpt}^{new} \cdot \chi \\ + \sum_{j \epsilon SC} \sum_p gr_{jp} \cdot \mu_{jkpt}^{used} \cdot \chi + \sum_i \sum_{j \epsilon SC} \sum_p c1_{ij} \cdot \mu_{ijkpt}^{new} \cdot \chi \\ + \sum_i \sum_j \sum_p c2_{jk} \cdot \mu_{ijkpt}^{new} \cdot \chi + \sum_{j \epsilon SC} 2 \cdot c2_{jk} \cdot \chi \cdot \sum_p \mu_{jkpt}^{used} \\ + \sum_i \sum_j \sum_p \theta 1_{jp} \cdot \mu_{ijkpt}^{new} \cdot t1_{ijp} + \sum_i \sum_j \sum_p \theta 2_{kp} \cdot \mu_{ijkpt}^{new} \cdot t2_{jkp} \\ + \sum_j \sum_p h1_{jp} \cdot ss_{jpt} + \sum_{s \notin KT_{ks}} \sum_{c \epsilon KSC_{ksc}} h2_k \cdot \lambda 2_{ks} \cdot \sigma_{ksct} \cdot \sqrt{l_{ksct}} \end{pmatrix}}{\begin{pmatrix} \sum_i \sum_j \sum_p \mu_{ijkpt}^{new} \\ + \sum_{j \epsilon SC} \sum_p \mu_{jkpt}^{used} \end{pmatrix}} \tag{31}$$

$$\lambda_{tolkst0} = -\frac{\begin{pmatrix} \sum_{j \epsilon SC} \sum_{s \epsilon KT_{ks}} gr'_{js} \cdot \tau_{jkst}^{used} \cdot \chi + \sum_i \sum_k \sum_{s \epsilon KT_{ks}} c3_{ik} \cdot \tau_{ikst}^{new} \cdot \chi \\ + \sum_{j \epsilon SC} \sum_k 2 \cdot c2_{jk} \cdot \chi \cdot \sum_{s \epsilon KT_{ks}} \tau_{jkst}^{used} \\ + \sum_k \sum_{s \epsilon KT_{ks}} \sum_{c \epsilon KSC_{ksc}} h2_k \cdot \lambda 2_{ks} \cdot \sigma_{ksct} \cdot \sqrt{m_{ksct}} \end{pmatrix}}{\begin{pmatrix} \sum_i \tau_{ikst}^{new} \\ + \sum_{j \epsilon SC} \tau_{jkst}^{used} \end{pmatrix}} \tag{32}$$

The subgradient optimization is a popular method to find a good 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. Due to the problem size, the values of multipliers can reach several hundreds to more than one thousand, and the corresponding subgradients are of the same order. However, the subgradients of constraint (8) and (9) from Part I cannot exceed the number of warehouses, 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 make all the multipliers of the same order by scaling, with which the contours become near circles. The scaling scheme is as follows.

Recall that $L = f(\lambda_{zkt}, \lambda_{vskt}, \lambda_{muctkpt}, \lambda_{tolkst}, \lambda_c)$, and the subgradients of $f$ is $g = (g_{zkt}^T, g_{vskt}^T, g_{muctkpt}^T, g_{tolks}^T, g_c^T)^T$.

Let $\lambda_{zkt} = \alpha_z \lambda'_{zkt}$, $\lambda_{vskt} = \alpha_v \lambda'_{vskt}$, $\lambda_{muctkpt} = \alpha_{mu} \lambda'_{muctkpt}$, $\lambda_{tolks} = \alpha_t \lambda'_{tolks}$, $\lambda_c = \alpha_c \lambda'_c$, where $\alpha_.$'s are positive scalars, we can rewrite the function as $L = f'(\lambda'_{zkt}, \lambda'_{vskt}, \lambda'_{muctkpt}, \lambda'_{tolks}, \lambda'_c)$. Then the subgradients of $f'$ is $g' = \left(\frac{1}{\alpha_z} g_{zkt}^T, \frac{1}{\alpha_v} g_{vskt}^T, \frac{1}{\alpha_{mu}} g_{muctkpt}^T, \frac{1}{\alpha_t} g_{tolks}^T, \frac{1}{\alpha_c} g_c^T\right)^T$. To make all the multipliers of the same order, we select right $\alpha_.$'s according to the initialization process, then the multipliers are scaled with equations (33)-(37).

$$\lambda'_{zkt} = \frac{1}{\alpha_z} \lambda_{zkt} \tag{33}$$

$$\lambda'_{vskt} = \frac{1}{\alpha_v} \lambda_{vskt} \tag{34}$$

$$\lambda'_{muctkpt} = \frac{1}{\alpha_{mu}} \lambda_{muctkpt} \tag{35}$$

$$\lambda'_{tolks} = \frac{1}{\alpha_t} \lambda_{tolks} \tag{36}$$

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

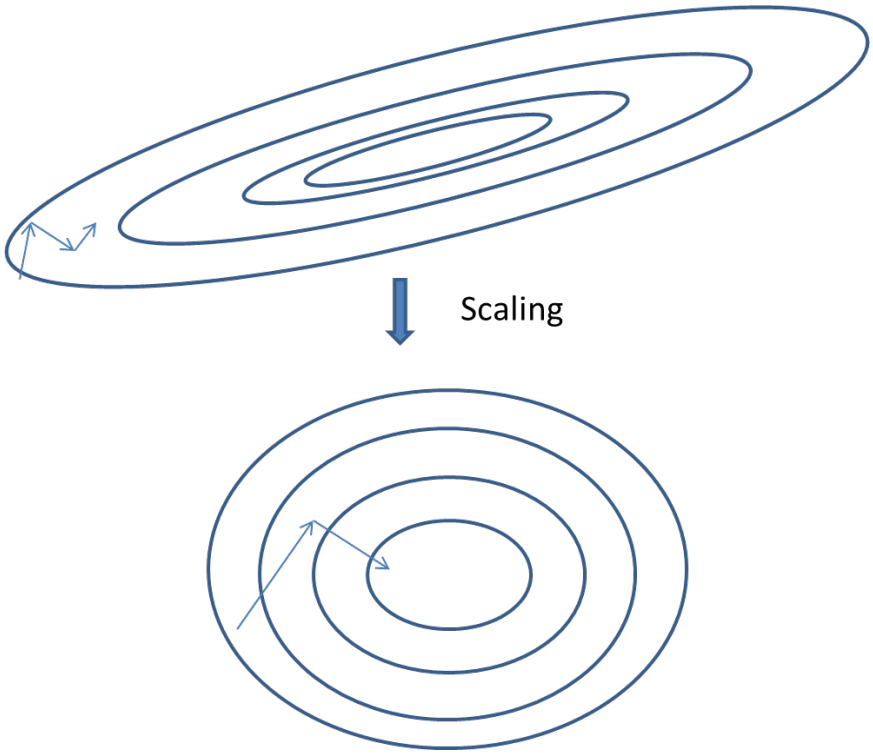


*Figure 4 Contours of the dual problem*

### 3.5 Lagrangean decomposition algorithm

In summary, the Lagrangean decomposition algorithm is given below.

*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 equations (21), (26) and (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 *MILPFea*, by approximating the square root terms with a linear approximation according to equation (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 $\lambda_{muctkp}$ and $\lambda_{tolkst}$ according to equations (31) and (32) respectively, initialize the other Lagrange multipliers $\lambda'_{zkt}, \lambda'_{vskt}, \lambda'_{muctkpt}, \lambda'_{c}$ to 0;

*Step 3*: Scale the Lagrange multipliers according to equations (33)-(37);

*Step 4*: Solve subproblems $P_j,\ \forall j \in J$ 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 5*: 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 equations (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 6:* If the convergence criterion is satisfied, stop the algorithm; otherwise, update $x^t$ in equations (20), (26), (27), $x^p$ in equation (28) and the scaled Lagrange multipliers $\left(\lambda'_{zkt}, \lambda'_{vskt}, \lambda'_{muctkpt}, \lambda'_{tolks}, \lambda'_{c}\right)_{it+1} = P_{+zvc}\left(\left(\lambda'_{zkt}, \lambda'_{vskt}, \lambda'_{muctkpt}, \lambda'_{tolks}, \lambda'_{c}\right)_{it+1} + (fup_{it} - l_{it})\frac{g'}{|g'|^2}\right)$, where $P_{+zvc}(\cdot)$ means projection to the space with nonnegative $\lambda'_{zkt}, \lambda'_{vskt}, \lambda'_{c}$, go to step 4.

### 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 12.0Gb. 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 an illustrative problem with relative

small scale, 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.

*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 |

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 linearization 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.

*Table 3 Objective values and CPU time required by different algorithms*

| Name | Optimal objective/$ | Error/% | CPU time (s) |
|---|---|---|---|
| MINLP | No feasible solution | — | 12,159.839 |
| MILP-2 | 5747911.872 | 0.24 | 683.674 |
| MILP-5 | 5748118.909 | 0.24 | 10,086.244 |
| AltNLPMILP | 5752005.0406 | 0.31 | 460.016 |
| LD (30 iterations) | 5733962.14 | 0 | 551.918 |

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

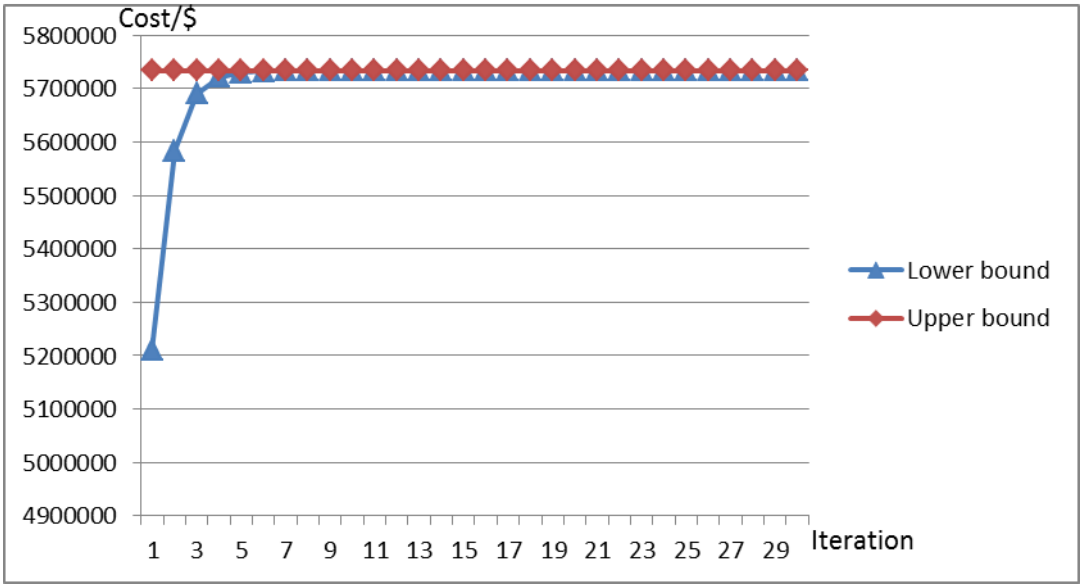


*Figure 5 Convergence of the lower and upper bound*

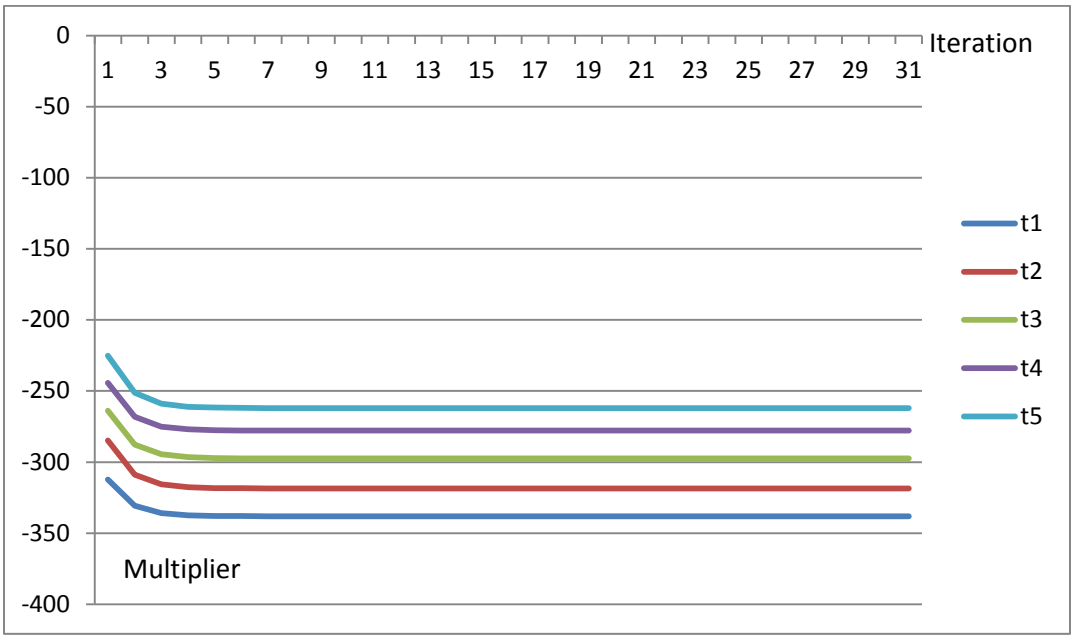


*Figure 6 Convergence of multipliers of demand constraints of motor p1 with criticality k1 at period t*

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 seconds. In Table 3 it can be seen that the MINLP model cannot obtain any feasible solution after more than 3 hours. The CPU time 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 long as MILP-2 with a slightly higher error (0.31% vs. 0.24%). The Lagrangean decomposition algorithm reaches the optimal solution faster in about 130 seconds. 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.

*Table 4 Capacity profiles of warehouses*

| /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.

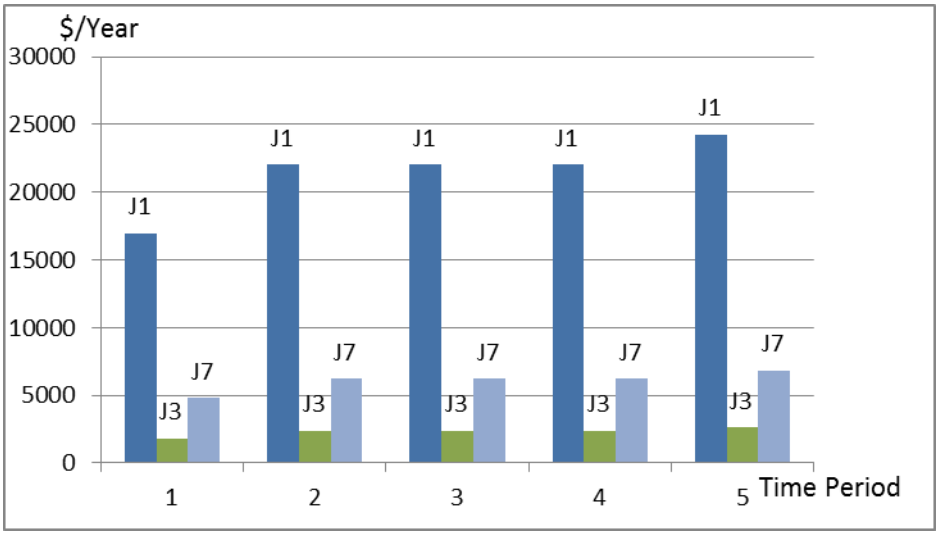


*Figure 7 Variable costs of warehouses for new motors*

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 Fig. 8-10, respectively.

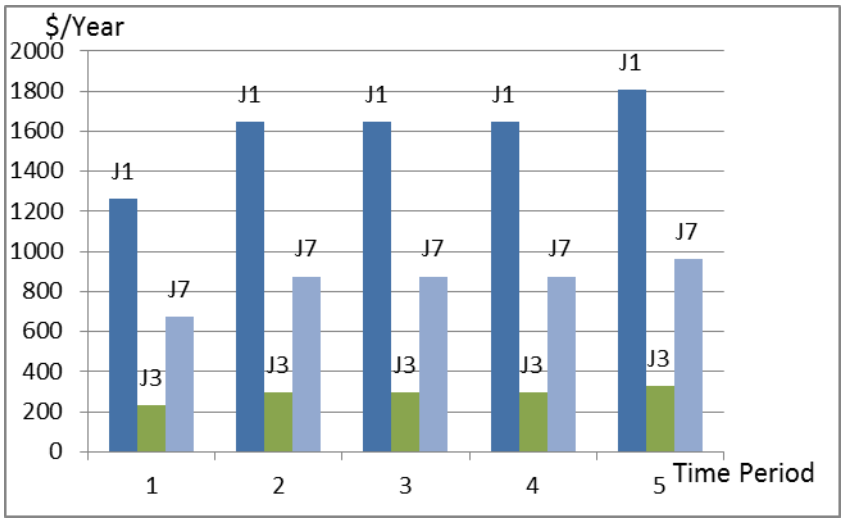


*Figure 8 Mean stock costs of warehouses for modifying*

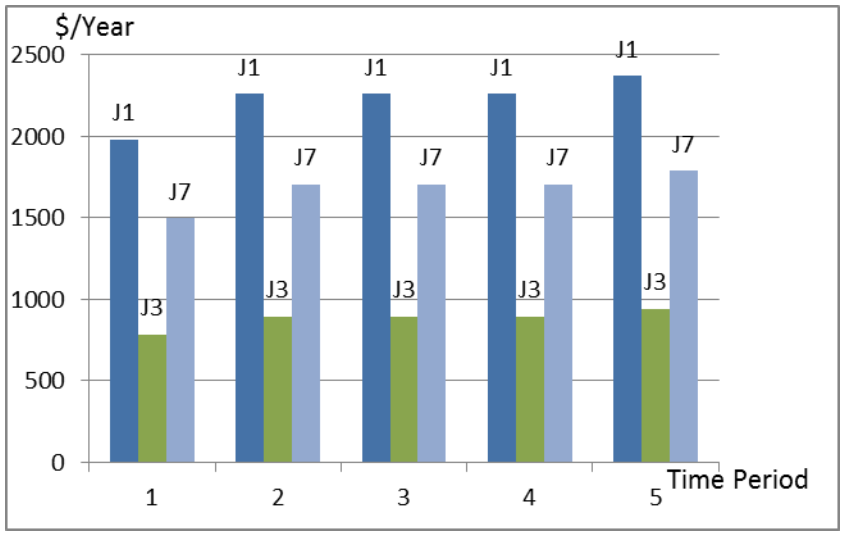


*Figure 9 Safety stock costs of warehouses*

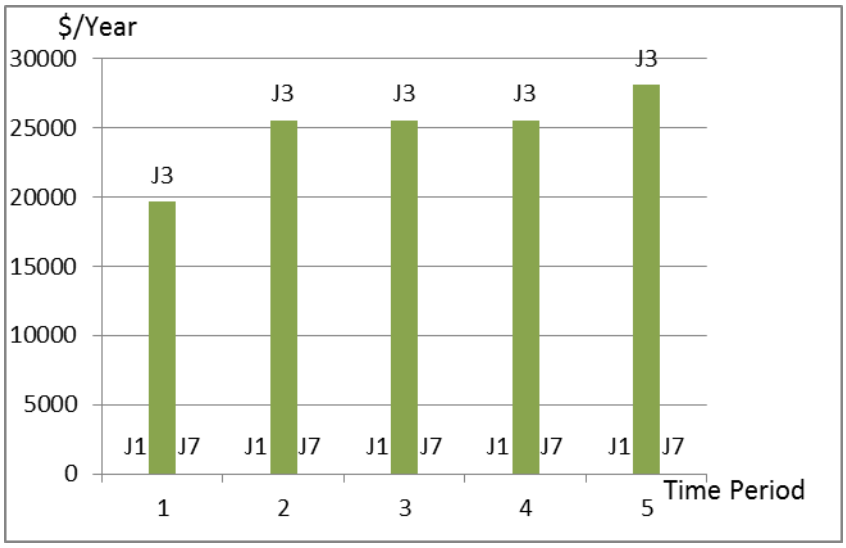


*Figure 10 Repair cost of warehouses for special motors*

*Case 2*

The objective values and CPU times required for case 2 are shown in Table 5.

*Table 5 Objective values and CPU time required by different algorithms*

| Name | Optimal objective/$ | Error/% | CPU time /s |
|---|---|---|---|
| MINLP | 6537842.475 | 5.40 | 32,288.415 |
| MILP-2 | 6354304.153 | 2.44 | 798.990 |
| MILP-5 | 6354304.153 | 2.44 | 86,020.932 |
| AltNLPMILP | 6358672.248 | 2.51 | 530.108 |
| LD (30 iterations) | 6202732.33 | 0 | 607.577 |

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

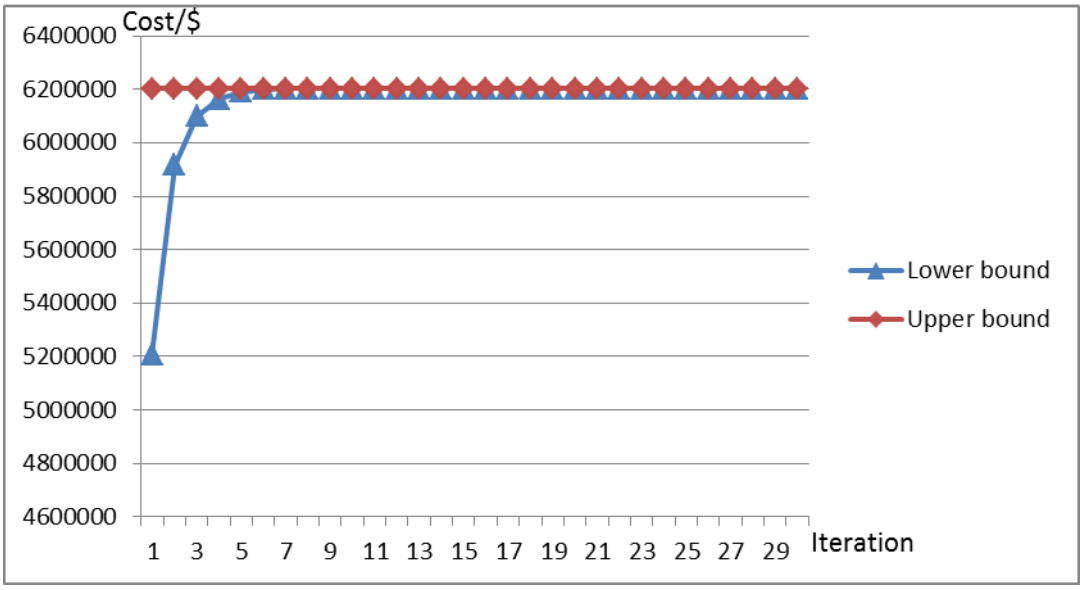


*Figure 11 Convergence of the lower and upper bound*

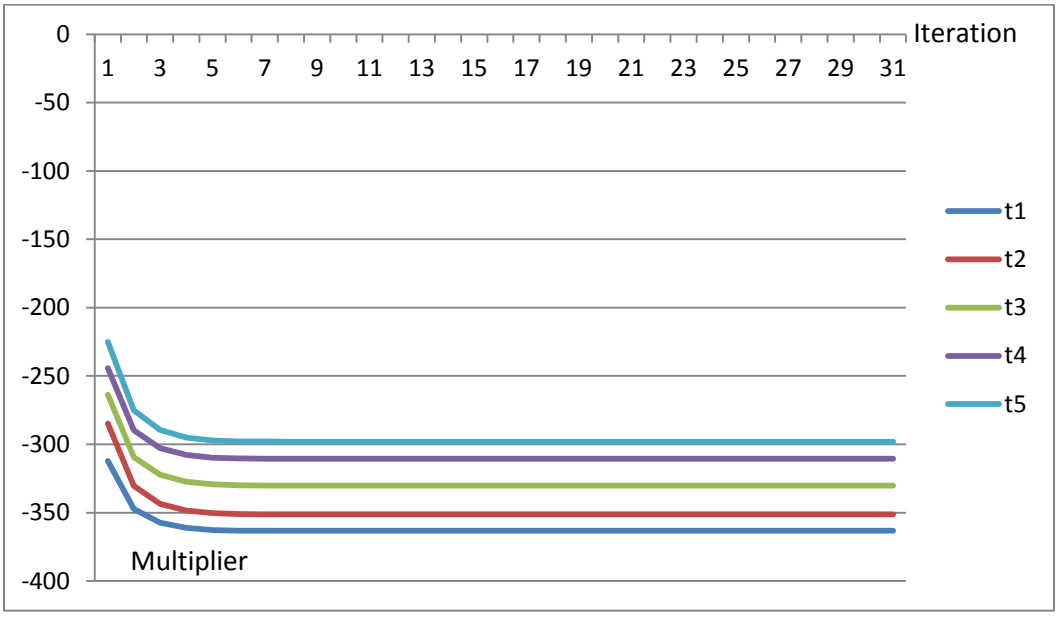


*Figure 12 Convergence of multipliers of demand constraints of motor p1 with criticality k1 at period t*

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 seconds. There are similar trend as in case 1.The capacity profiles for case 2 of the 3 selected warehouses (J1, J2 and J3) are shown in Table 6, the cost details of the optimal solution obtained with the Lagrangean decomposition algorithm are shown in Figs. 13-16.

*Table 6 Capacity profiles of warehouses*

| /SKUs | Year 1 | Year 2 | Year 3 | Year 4 | Year 5 |
|---|---|---|---|---|---|
| J1 | 50000 | 50000 | 50000 | 50000 | 50000 |
| J2 | 50 | 50 | 50 | 50 | 50 |
| J3 | 50 | 50 | 50 | 50 | 50 |

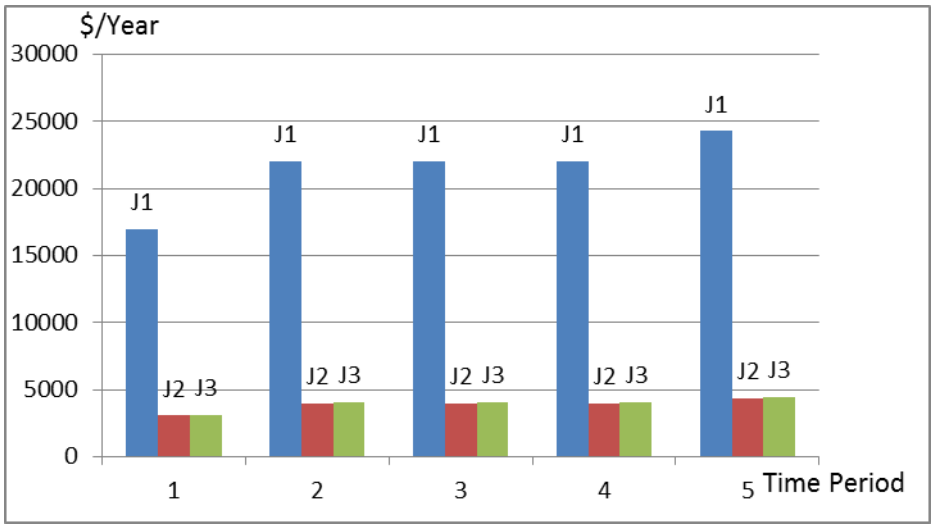


*Figure 13 Variable costs of warehouses for new motors*

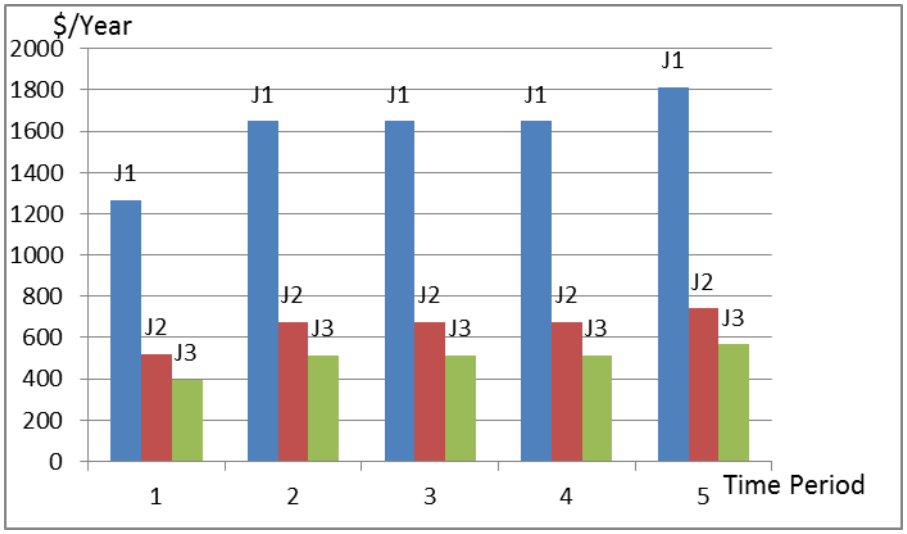


*Figure 14 Mean stock costs of warehouses for modifying*

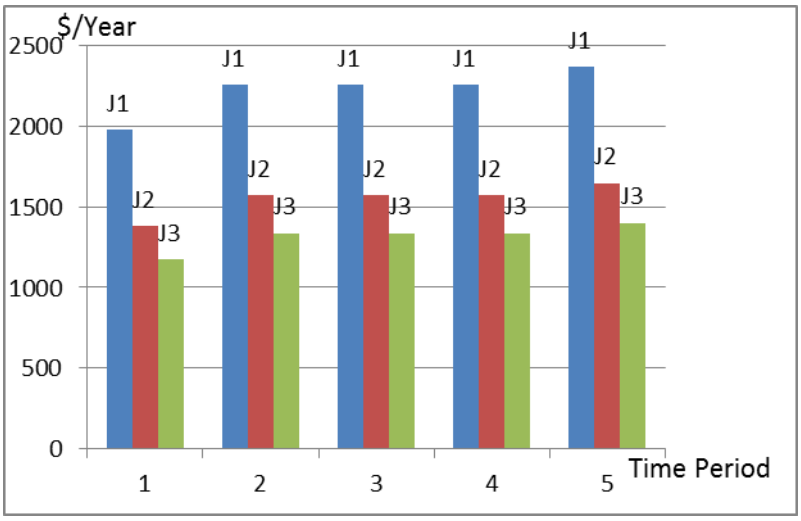


*Figure 15 Safety stock costs of warehouses*

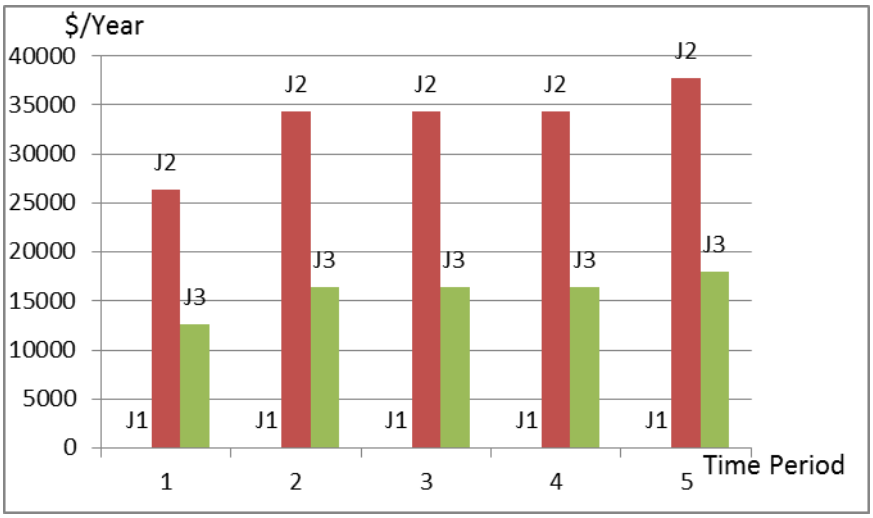


*Figure 16 Repair costs of warehouses for special motors*

*Case 3*

The objective values and CPU time required for case 3 are shown in Table 7.

*Table 7 Objective values and CPU time required by different algorithms*

| Name | Optimal objective/$ | Error/% | CPU time /s |
|---|---|---|---|
| MINLP | No feasible solution | — | 360,123.004 |
| MILP-2 | 120349878.7579 | 11.25 | 10,871.289 |
| MILP-5 | No feasible solution | — | out of memory |
| AltNLPMILP | 120657481.7045 | 11.53 | 19942.81 |
| LD (30 iterations) | 108178792.52 | 0 | 20125.03 |

The iteration details of the LD results are shown in Figs. 17 and 18.

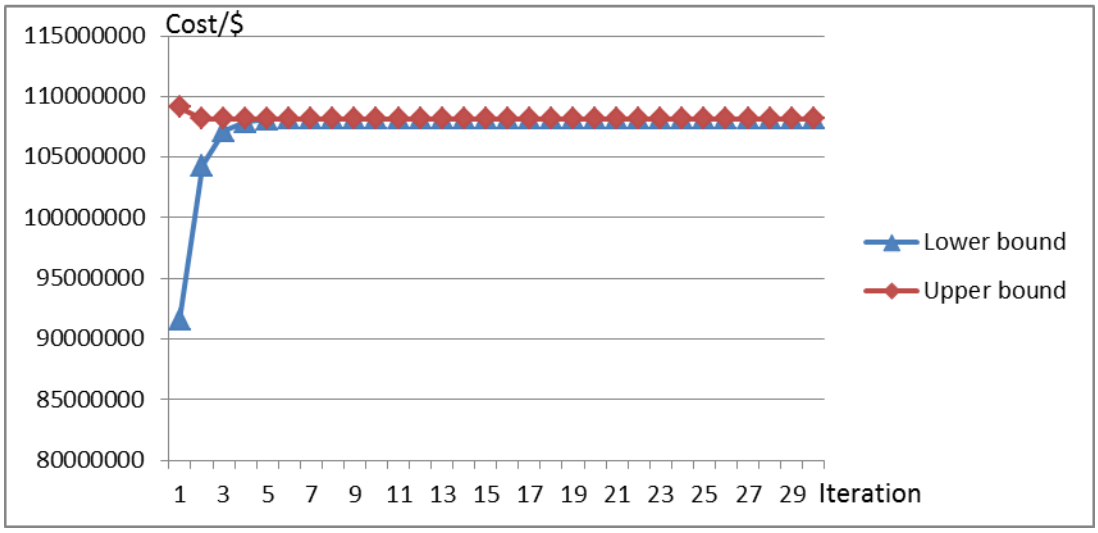


*Figure 17 Convergence of lower and upper bound*

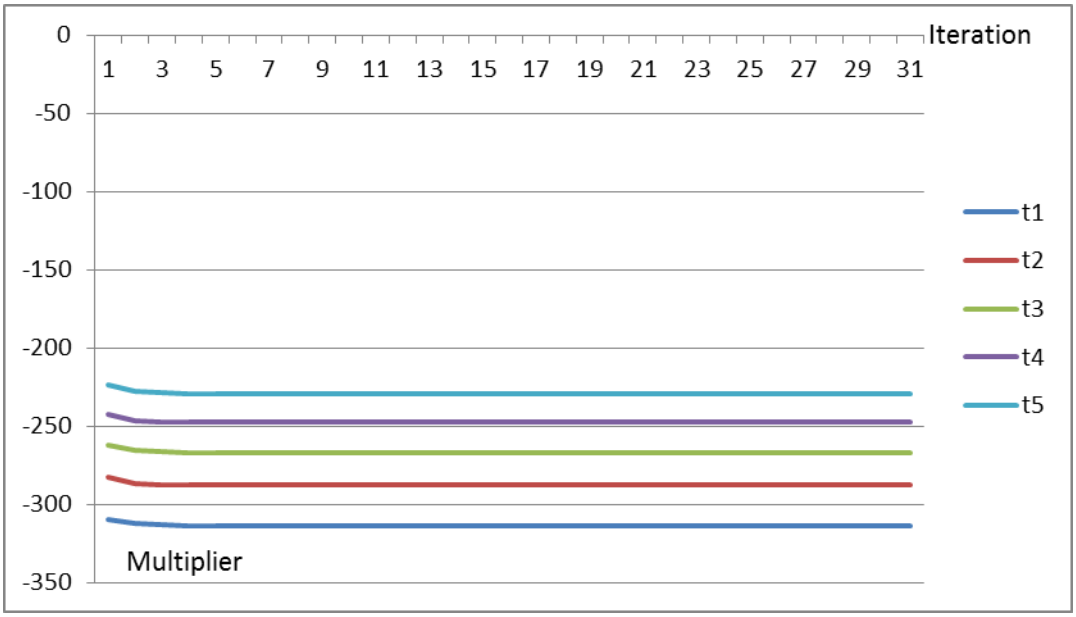


*Figure 18 Convergence of multipliers of demand constraints of motor p1 withcriticality k1 at period t*

In Fig. 17, 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 seconds. 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.

The capacity profiles for case 3 of warehouses (all the warehouses are selected) are shown in Table 8, the cost details of the optimal solution obtained with the Lagrangean decomposition algorithm are shown in Fig. 19-22. In this case, warehouse J5 is installed for repairing the used motors, no new motor is modified in warehouse J5.

*Table 8 Capacity profiles of warehouses*

| /SKUs | Year 1 | Year 2 | Year 3 | Year 4 | Year 5 |
|---|---|---|---|---|---|
| J1 | 50000 | 50000 | 50000 | 50000 | 50000 |
| J2 | 50 | 100 | 150 | 167.64 | 217.64 |
| J3 | 50 | 94.162 | 106.147 | 106.147 | 106.147 |
| J4 | 50 | 100 | 100 | 100 | 100 |
| J5 | 80 | 80 | 80 | 80 | 80 |

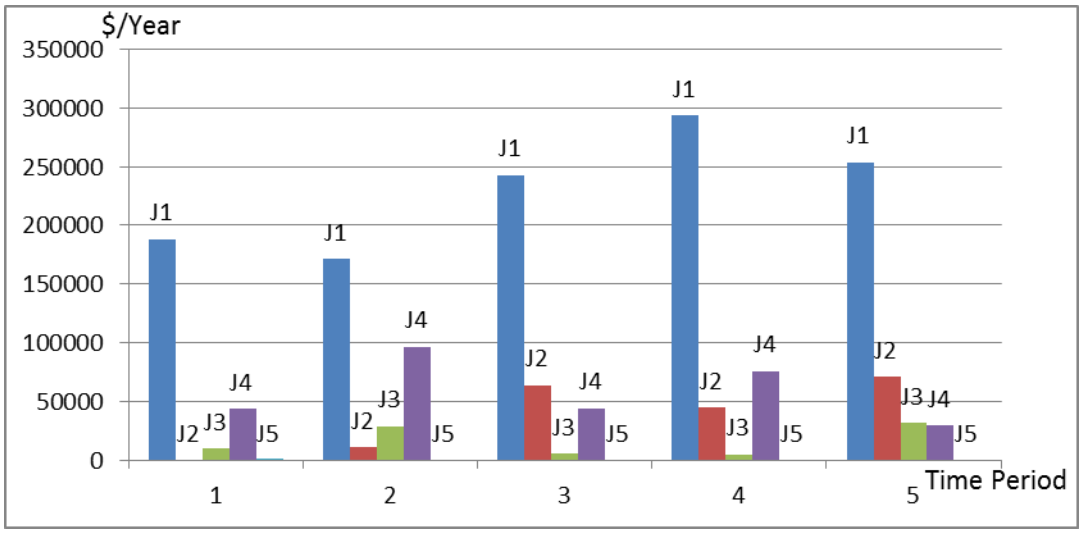


*Figure 19 Variable costs of warehouses for new motors*

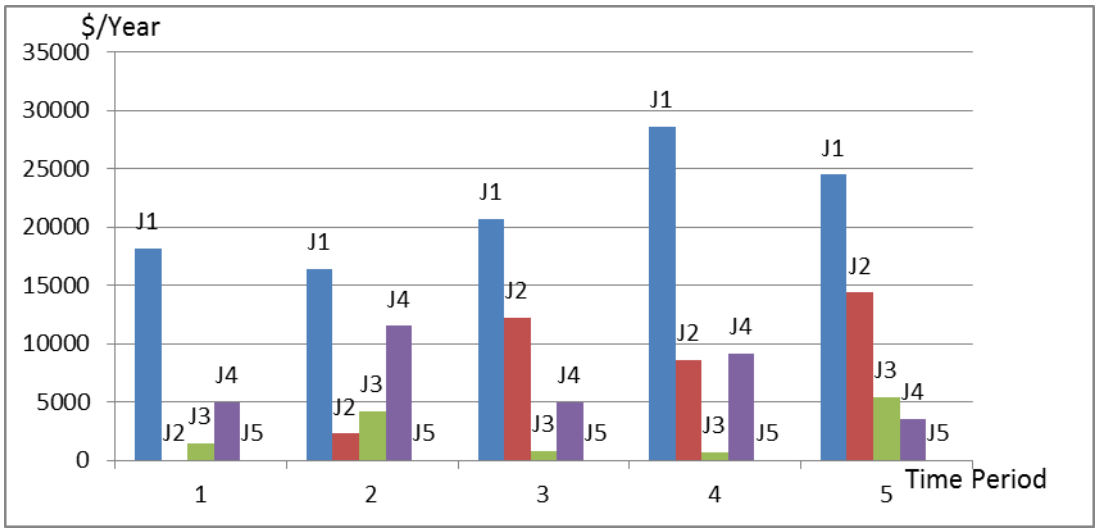


*Figure 20 Mean stock costs of warehouses for modifying*

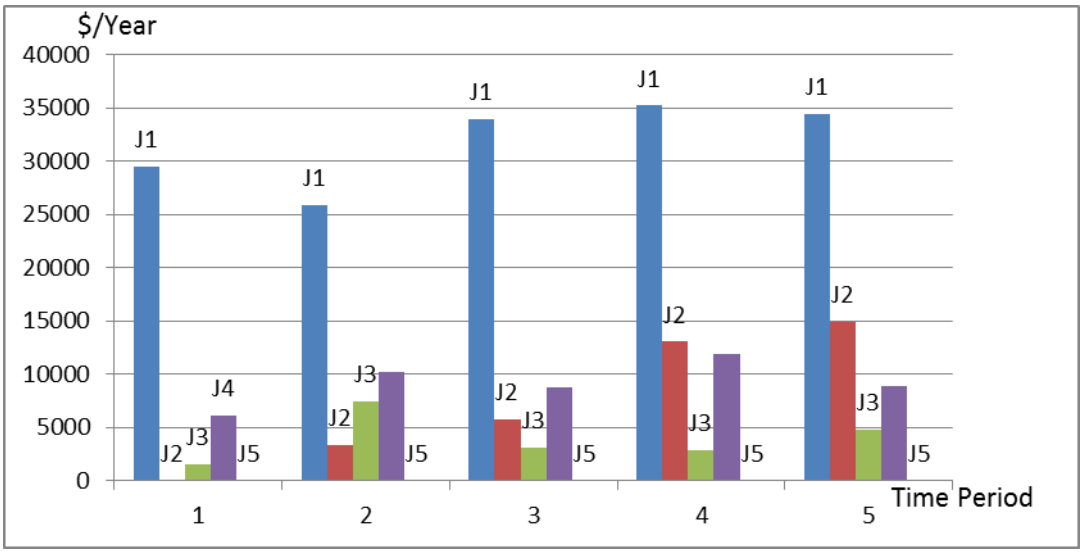


*Figure 21 Safety stock costs of warehouses*

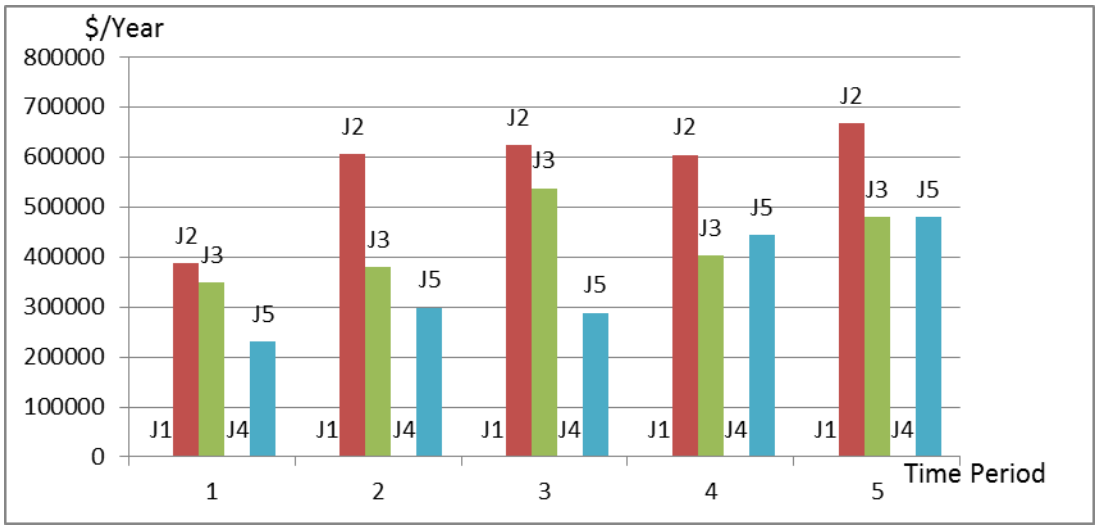


*Figure 22 Repair costs of warehouses for special motors*

To summarize the three cases, we can conclude that the Lagrangean decomposition algorithm can obtain the optimal solution efficiently. As the problem scale increases, the advantage becomes more apparent, especially for the problem with small feasible region.

## 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 from, the proposed Lagrangean decomposition method is more competitive.

**References**


Barrie M. Baker, Janice Sheasby. Accelerating the convergence of subgradient optimization. European Journal of Operational Research, 117: 136-144, 1999

Roman Buil, Miquel Àngel Piera, Peter B. Luh. Improvement of Lagrangian relaxation convergence for production scheduling. IEEE Transactions on Automations Science and Engineering, 9(1): 137-147, 2012

Francesca Fumero. A modified subgradient algorithm for Lagrangean relaxation. Computers & Operations Research, 28: 33-52, 2001

Grossmann, I.E. Enterprise-wide Optimization: A New Frontier in Process Systems Engineering. AIChE Journal, 51(7): 1846–1857, 2005

Maria Analia Rodriguez, Aldo R. Vecchietti, Iiro Harjunkoski and Ignacio E. Grossmann, Optimal supply chain design and management over a multi-period horizon under demand uncertainty. Part I: MINLP and MILP models, submitted for publication, 2013

M.T. Melo, S. Nickelc, F. Saldanha-da-Gama. Facility location and supply chain management - A review. European Journal of Operational Research 196: 401-412, 2009

Sylvain Mouret, Ignacio E. Grossmann, Pierre Pestiaux. A new Lagrangian decomposition approach applied to the integration of refinery planning and crude-oil scheduling. Computers and Chemical Engineering, 35: 2750- 2766, 2011

Sebastian Terrazas-Moreno, Philipp A. Trotter, Ignacio E. Grossmann. Temporal and spatial Lagrangean decompositions in multi-site, multi-period production planning problems with sequence-dependent changeovers. Computers and Chemical Engineering, 35: 2913- 2928, 2011

Shih-Ho Wang. An improved step size of subgradient algorithm for solving the Lagrangian relaxation problem. Computers and Electrical Engineering, 29: 245-249, 2003