Mixed Integer Nonlinear Programming for Material Supply Management

Vu A. Huynh*
Computation for Design and Optimization, Massachusetts Institute of Technology, Cambridge, MA 02139, vuhuynh@mit.edu

Maokai Lin
Computation for Design and Optimization, Massachusetts Institute of Technology, Cambridge, MA 02139, lmk@mit.edu

Joongwoo Brian Park
Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, jwbpark@mit.edu

Chuan Shi
Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, chuanshi@mit.edu

In this paper, we present the material supply optimization for a factory using mixed integer nonlinear programming. Our goal is to minimize the labor force for the factory under production rate constraints. Due to the lack of powerful solvers, we introduce a sub-optimization problem with a simplified model in which the number of labors is fixed, instead of considering the number of labors as a decision variable. We solve our model by the MINLP solver and verify our results using 3D simulation. Obtained results show that we reduce the labor force by 22.5%, and thus, save $23,760 per month in labor cost for the factory.

Key words: nonlinear programming, mixed integer programming, material supply optimization

1. Introduction

Mixed integer nonlinear programming (MINLP) represents a powerful framework for mathematically modeling many optimization problems that involve discrete and continuous variables. The use of MINLP is a natural approach of formulating problems where it is necessary to simultaneously optimize the system structure (discrete) and parameters (continuous). MINLP has been used in various applications, including the process industry and the financial, engineering, management science and operations research. It includes problems in process flow sheets, portfolio selection, batch processing in chemical engineering (consisting of mixing, reaction, and centrifuge separation), and optimal design of gas or water transmission networks. Other areas of interest include

* Authors’ names in alphabetical order of last name. Authors contributed equally to this work.
the automobile, aircraft, and VLSI manufacturing areas [1][2].

In our paper, we approach material supply management with MINLP method. There has been various analysis on supply management with network optimization, integer programming and linear programming [3][15][16]. Unlike traditional inventory stock management with linear approaches, with MINLP approach, we have showed in detail the analysis of labor and time managing in the factory with more realistic models.

The most basic form of an MINLP problem when represented in algebraic form is as follows:

$$\min Z = f(x, y)$$

$$s.t. g_j(x, j) \leq 0, j \in J$$

$$x \in X, y \in Y$$

where $f()$, $g()$ are convex, differentiable functions, $J$ is the index set of inequalities, and $x$ and $y$ are the continuous and discrete variables, respectively. The set $X$ is commonly assumed to be a convex compact set, e.g. $X = \{x | x \in \mathbb{R}^n, Dx \leq d, x^L \leq x \leq x^U\}$. The discrete set $Y$ corresponds to a polyhedral set of integer points, $Y = \{y | y \in \mathbb{Z}^m, Ay \leq a\}$, and in most applications is restricted to $\{0, 1\}$ values, $y \in \{0, 1\}^m$. In most applications of interest the objective and constraint functions $f()$, $g()$ are linear in $y$ (e.g. fixed cost charges and logic constraints). MINLP problems are precisely so difficult to solve, because they combine all the difficulties of both of their subclasses: the combinatorial nature of mixed integer programs (MIP) and the difficulty in solving nonconvex (and even convex) nonlinear programs (NLP). Because subclasses MIP and NLP are among the class of theoretically difficult problems (NP-complete), so it is not surprising that solving MINLP can be a challenging and daring venture. Methods that have addressed the solution of problem (1) include the branch and bound method [4–8], Generalized Benders Decomposition [9], Outer-Approximation [10–12], LP/NLP based branch and bound [13], and Extended Cutting Plane Method [14].

2. Problem Statement

The optimization problem in our project arises from an internal logistics system in a factory that manufactures optical disc drivers. In the factory, materials are stored in rows of racks in the first
floor. The process, in which labors collect the materials into carts according to the need of assembly lines in the second floor, includes walking time and nonlinear material-handling time varying with the weights and the volumes of the materials. Since there are dozens of types of the materials, a cart can only carry several of them. Our goal is to minimize the labor force (or equivalently, labor cost) subject to the production rate requirements of the assembly lines in the second floor. That is, we have to assure that the assembly lines get enough supply in every manufacturing cycle. We verify obtained optimization results through simulations.

3. Problem Formulation

According to the problem statement in Section 2, we want to minimize the labor force, or the labor cost, subject to the production rate requirements of the assembly lines. In order to do so, our approach is to optimize the task allocation for each labor. If we can do this, we can best utilize every labor to reduce the number of labors to the minimal, yet required, value. In the following subsections, we will first introduce our cost function. Then we will introduce necessary notations, following which we introduce two versions for our model. The first one is called an unfixed path model, while the second one is called a fixed path model. We elaborate on the reason that the unfixed path model is too complicated to solve and discuss why the fixed path model is reasonable to be adopted.

3.1. Cost Function Description

Our foremost consideration of the cost function is to minimize the number of labors. However, as we will introduce later, our final problem formulation turns out to be a mixed integer nonlinear model. Unfortunately, if the number of labors, say $N$, is also a decision variable, to the best of our knowledge, existing available solvers are unable to handle this model [19]. Thus, we eventually achieve our goal by introducing a sub-optimization model, in which we fix the number of labors. Hence, our approach is that we initially let the number of labors be 1. Then we solve the sub-optimization model (in which we optimize labor task allocation to check whether it can meet the production rate constraints), if there exists a feasible solution, we have reached our goal and the
optimal number of labors is 1. Otherwise, we increase the number of labors by 1 repetitively until
we finally reach a certain number of labors for which the sub-optimization problem has a feasible
solution for the first time. That certain number of labors will be a sub-optimal solution to our
problem.

Therefore, the cost function of our primal optimization model is

$$\min CN$$

where \( C \) is the constant cost per labor, and \( N \) is a decision variable standing for the number
of labors. And in our sub-optimization problem, since we fix the number of labors, our problem
becomes a feasibility problem.

3.2. Notations

We introduce necessary notations to formulate the problem in this section.

1. \( N \): A decision variable specifies the number of labors.

2. \( t \): A decision variable specifies material supply time per round.

3. \( x_{ij} = \{0,1\}, \forall i = 1,\cdots,N; \ j = 1,\cdots,K \): Decision variables identify whether the \( i \)th labor picks
up the \( j \)th material. Suppose we totally have \( N \) labors and \( K \) materials, \( x_{ij} = 1 \) means that the
\( i \)th labor will pick up the \( j \)th material in each round. Otherwise, he will not do so.

4. \( q_{ij}, \forall i = 1,\cdots,N; \ j = 1,\cdots,K \): Decision variables specify the quantity of the \( j \)th material
picked up by the \( i \)th labor, if any. There is a logical correspondence between \( x_{ij} \) and \( q_{ij} \), which is
\( q_{ij} > 0 \) if and only if \( x_{ij} = 1 \). Otherwise, \( q_{ij} = 0 \). This will be guaranteed by the following constraint

$$0 \leq q_{ij} \leq x_{ij}$$

5. \( T \): A constant denotes manufacturing cycle, 1 hour, or 3600 seconds.

6. \( C \): A constant denotes cost per labor in 1 hour: 15 dollars.

7. \( w_i, i = 1,\cdots,K \): Constants specify the unit weight of the \( i \)th material. Data will be provided
in section 4.
8. $v_i, i = 1, \cdots, K$: Constants specify the unit volume of the $i$th material. Data will be provided in section 4.

9. $D$: A constant specify total walking distance for a labor to go through all racks during each round: 130 meters.

10. $M_i, i = 1, \cdots, K$: Constants specify total amount of the $i$th material required during one operation period. Data will be provided in section 4.

11. $W$: A constant specifies the weight capacity of the carts: 200 kg.

12. $V$: A constant specifies the volume capacity of the cart: 1 m$^3$.

13. $\bar{v}$: A constant specifies the required walking speed of the labors set by the factory. 1.2 m/s.

3.3. Model Scenario

We convert our goal, the optimization of labor task allocation, into a mathematical model in this section. Let us clarify our scenario first. Since we assign various tasks for different labors, we want them to go to specific racks directly to pick up materials. This will save some walking time for labors. We initially formulate our model under this consideration. It, however, turns out that this version of our model is too hard to be solved by any commercial solver, and unfortunately, our knowledge in nonlinear programming so far is not enough to solve it by ourselves. Therefore, we finally simplify our model by making a reasonable assumption that during each round, every labor has to go through all racks. That is, the path for each labor to go through in each round is fixed. This seems a little unconvincing at a first glance, but after some calculations, labor’s walking time along the path during each period is less than 2 minutes, which is much less than the material handling time as we will show in our solutions in section 4. Hence, in the following subsections, we first present our initial unfixed path model in order to provide readers the idea of the general optimization problem. Then, we concentrate on the simplified fixed path model. We solve the simplified model in section 4.
3.4. Unfixed Path Model

We present our initial model, in which we consider the unfixed path option for each labor. In this case, we expect that each labor only walks between the racks from which he needs to pick up the materials directly. For example, if a labor needs to pick up material 1 and 5. He should go to rack 1 for material 1 and then go to rack 5 for material 5 directly, rather than going through rack 2, 3, and 4. In order to realize this, we introduce additional decision variables, which are not included in the notations section:

\[ P_{i,j,g} = \{0, 1\}, \forall i = 1, \ldots, N; j = 1, \ldots, K, g = 1, \ldots, K \]: Identifying whether, for the \( i \)th labor, the \( j \)th material is picked up at the \( g \)th order. For example, \( P_{1,5,1} = 1 \) means that, for the 1st labor, material 5 is picked up as the 1st material of him. Namely, the 1st labor doesn’t pick up any of the first 4 materials, which means he goes to the 5th rack (for material 5) from the origin directly.

Besides, there are some constraints for \( P_{i,j,g} \):

1. For a fixed pair of \( i \) and \( g \), say \( \{i^*, g^*\} \), \( P_{i^*,j,g^*} \) could equal to 1 for no more than one \( j \). For instance, in the example above, \( P_{1,5,1} = 1 \) indicates that material 5 is to be picked up as the first material for the \( i \)th labor. So materials other than the 5th material cannot be picked up as the first material by that labor. Then \( P_{1,j,1} = 0, \forall j = 1, \ldots, K, j \neq 5 \). Moreover, for the \( i^* \)th labor, for example, if he finally picks up 3 kinds of materials, then for \( g \geq 4 \), all \( P_{i^*,j,g} = 0, \forall j \), because no material will be picked up for the \( i^* \)th labor at an order greater than or equal to 4. Therefore, combining these two considerations, we have the following constraint for \( P_{i,j,g} \):

\[
\sum_{j=1}^{K} P_{i,j,g} \leq 1, \forall i = 1, \ldots, N; g = 1, \ldots, K (4)
\]

2. Next, we show the logical relationship between \( x_{ij} \) and \( P_{i,j,g} \), and how we realize it by mathematical expressions. Suppose that, for the \( i^* \)th labor, he will pick up the \( j^* \)th material. Then we know that \( x_{i^*,j^*} = 1 \). The order, say \( g^* \), at which he picks up material \( j^* \), equals to the sum of all \( x_{i^*m}, \forall m = 1, j^* \), or

\[
g^* = \sum_{m=1}^{j^*} x_{i^*m} (5)
\]
Then $P_{i^*,j^*,g^*} = 1$, and for $g \neq g^*$, we must have $P_{i^*,j^*,g} = 0$. Hence, for $x_{i^*,j^*} = 1$, which means the $i^*$th labor picks up the $j^*$th material, if $g = g^*$, we want $P_{i^*,j^*,g^*} = 1$; otherwise $P_{i^*,j^*,g} = 0$.

On the other hand, if $x_{i^*,j^*} = 0$, which means that the $i^*$th labor will not pick up the $j^*$ material, then for all $g$, we must have $P_{i^*,j^*,g} = 0$. Considering all the logical analysis above, we establish the following two constraints for $P_{i,j,g}$ and $x_{ij}$, which convert the logical relationship into mathematical expressions.

\begin{equation}
P_{i,j,g} \left[ x_{ij} \left( \sum_{m=1}^{j} x_{im} \right) - g \right] = 0, \forall i = 1, \cdots, N; j, g = 1, \cdots, K \tag{6}
\end{equation}

\begin{equation}
1 - P_{i,j,g} \leq \left| x_{ij} \left( \sum_{m=1}^{j} x_{im} \right) - g \right|, \forall i = 1, \cdots, N; j, g = 1, \cdots, K \tag{7}
\end{equation}

Before we use the decision variable $P_{i,j,g}$ to realize the unfixed path for each labor mathematically, we need to state other decision variables, which are used to calculated the distance between any two materials in $Y$ axis.

\[
\Delta Y_{j_1,j_2}, \forall j_1, j_2 \in \{1, \cdots, K\}:
\]

Difference in $Y$ coordinate between material $j_1$ and $j_2$. To explain this clearly, we provide Figure 1. In Figure 1, identical rectangles stand for racks. Please note that the figure just provides the idea of why we need $\Delta Y_{j_1,j_2}$. The layout of racks in the figure is not necessary the same with actual layout in that factory. The dashed line and the solid line are two sample paths (the dashed dot line can be seen as accessorl paths by which labors go to their 1st rack and go back to the scanner).

Let us consider the dashed line path first. For this case, a particular labor needs to pick up material 1 and material 5. Let $O_j(x)$ and $O_j(y)$ denote the $X$ coordinate and $Y$ coordinate of material $j$, respectively. Therefore, for the dashed path, the distance between the two materials is calculated as

\begin{equation}
d_{1,5} = |O_1(x) - O_5(x)| + L \tag{8}
\end{equation}

rather than

\begin{equation}
d_{1,5} = |O_1(x) - O_5(x)| + |O_1(y) - O_5(y)| = |O_1(x) - O_5(x)| \tag{9}
\end{equation}
Let us then consider the solid path, for which the cart needs to pick up material 18 and material 7. Then the distance between the two materials is calculated as

\[ d_{7,18} = |O_{7}(x) - O_{18}(x)| + |O_{7}(y) - O_{18}(y)| \]  

(10)

From the analysis above we see that for material \( j_1 \) and \( j_2 \) in different rows, their distance is
given by

\[ d_{j_1,j_2} = |O_{j_1}(x) - O_{j_2}(x)| + |O_{j_1}(y) - O_{j_2}(y)| \tag{11} \]

while for material \( i \) and \( j \) in the same row, their distance is given by

\[ d_{j_1,j_2} = |O_{j_1}(x) - O_{j_2}(x)| + L \tag{12} \]

This is why we introduce the decision variable \( \Delta Y_{j_1,j_2} \), and then for two materials \( j_1 \) and \( j_2 \), their distance is given by

\[ d_{j_1,j_2} = |O_{j_1}(x) - O_{j_2}(x)| + \Delta Y_{j_1,j_2} \tag{13} \]

where \( \Delta Y_{j_1,j_2} = \max(L, |O_{j_1}(y) - O_{j_2}(y)|) \). We further convert the max condition into the three expressions (constraints) below

\[ \Delta Y_{j_1,j_2} \geq L \tag{14} \]

\[ \Delta Y_{j_1,j_2} \geq |O_{j_1}(y) - O_{j_2}(y)| \tag{15} \]

\[ (\Delta Y_{j_1,j_2} - |O_{j_1}(y) - O_{j_2}(y)|)(\Delta Y_{j_1,j_2} - L) = 0 \tag{16} \]

With our necessary description above, we are ready to move on showing how we formulate the material supply time per round for each labor. Considering the unfixed path for each labor, then we have

\[ \sum_{j=1}^{K} 1.62(v_j q_{ij} + 0.00637 w_j q_{ij})^{0.5} + \sum_{g=1}^{K-1} \sum_{j_1=1}^{K} \sum_{j_2=1}^{K} P_{i,j_1,g} P_{i,j_2,g+1} (|O_{j_1}(x) - O_{j_2}(x)| + \Delta Y_{j_1,j_2}) / \hat{v} \leq t \tag{17} \]

In the inequality above, the first part \( \sum_{j=1}^{K} 1.62(v_j q_{ij} + 0.00637 w_j q_{ij})^{0.5} \) is the material handling time for the \( i \)th labor in each round. Parameters in the expression are generated from experiments.

The second part \( \sum_{g=1}^{K-1} \sum_{j_1=1}^{K} \sum_{j_2=1}^{K} P_{i,j_1,g} P_{i,j_2,g+1} (|O_{j_1}(x) - O_{j_2}(x)| + \Delta Y_{j_1,j_2}) / \hat{v} \) is the walking time for the \( i \)th labor going through all materials he has to pick up. For example, for the 1st labor, if he needs to pick up material 3, 5, and 10 at the order of 1, 2, and 3, respectively. Then we know that \( P_{1,3,1} = 1, P_{1,5,2} = 1, \) and \( P_{1,10,3} = 1, \) while all other \( P_{1,j,g} = 0. \) Thus, in the expression above, when \( g = 1, \) for all combination of \( j_1, j_2 \in \{1, \cdots, K\} \), we will only have

\[ P_{1,3,1} P_{1,5,2} = 1 \tag{18} \]
this tells us that the 1st labor moves from material 3 to 5 directly. Similarly, when $g = 2$, for all combination of $j_1, j_2 \in \{1, \cdots, K\}$, we will only have

$$P_{1,5,2}P_{1,10,3} = 1$$  \hspace{1cm} (19)

this tells us that he moves from material 5 to 10. For all $g \geq 4$, since all $P_{1,j,g} = 0$, that labor doesn’t need to pick up any more materials.

In addition, in this constraint, we do not consider the time the labor spent to move from the origin to the first material he needs to pick up, and the time the labor spent to move from the last material he needs to pick up to the scanner. We can see the origin and the scanner as two pseudo-racks for which each labor has to go through. We can simply realize that by adding $x_{i0}$ and $x_{iK+1}$ for each labor.

It turns out that Equation (4), (6), (7), and (14) − (17) form a set of constraints that make the unfixed path model very difficult. Since we are unable to solve it at this moment, we eventually simplify our model to the fixed path model and introduce it in the following section.

\subsection*{3.5. Fixed Path Model}

Since we fix the path for all labors, we will no longer need the decision variable $P_{i,j,g}$. Therefore, we will not have the set of constraints mentioned above. Instead, the distance of the path for labors to go through becomes a constant, which is denoted by $D$. And thus, the walking time ($\frac{D}{v}$) becomes a constant as well.

\subsubsection*{3.5.1. Cost function} We have presented our cost function in section 3.1. We simply restate it here for integrity.

$$\min CN$$ \hspace{1cm} (20)

\subsubsection*{3.5.2. Constraints} We introduce our constraints one by one in this section.

1. Logical relationship in $x_{ij}$ and $q_{ij}$:

$$0 \leq q_{ij} \leq x_{ij} \infty$$ \hspace{1cm} (21)

We have indicated this in section 3.2.
2. Material supply time per round constraint for all labors:

\[
\sum_{j=1}^{K} 1.62(v_j q_{ij} + 0.00637 w_j q_{ij})^{0.5} + \frac{D}{\varpi} \leq t, \forall i = 1, \ldots, N
\]  

(22)

In the inequality above, the first part is the material handling time for the \(i\)th labor in each round as before, while the second part \(\frac{D}{\varpi}\) is labor’s walking time through the fixed path in each round.

3. Weight capacity constraint of the cart:

\[
\sum_{j=1}^{K} w_j q_{ij} \leq W, \forall i = 1, \ldots, N
\]  

(23)

The total weight of materials picked up by a certain labor in each round should satisfy the weight capacity constraint of the cart.

4. Volume capacity constraint of the cart:

\[
\sum_{j=1}^{K} v_j q_{ij} \leq V, \forall i = 1, \ldots, N
\]  

(24)

The total volume of materials picked up by a certain labor in each round should satisfy the volume capacity constraint of the cart.

5. Material demand constraint for all materials

\[
M_j \leq \left( \sum_{i=1}^{N} q_{ij} \right) \left\lceil \frac{T}{t} \right\rceil \leq 1.05M_j, \forall j = 1, \ldots, K
\]  

(25)

For each material, total supply amount during each manufacturing cycle by all labors should meet its demand while should not surpass too much to avoid any inventory cost.

So far, we describe our cost function and all constraints. The problem is therefore a mixed integer nonlinear programming problem. We next introduce our solution mechanism.

4. Solution Mechanism

In this session, we solve the fixed path model. We model the problem in AMPL language and adopt a mixed-integer with non-linear constraints solver called MINLP to obtain optimal solutions.
4.1. Model in AMPL

```
set MATERIAL; set CART;
param demand{MATERIAL} >= 0;
param capacity{CART} >= 0;
param cartVolume{CART} >= 0;
param volume{MATERIAL} >= 0;
param weight{MATERIAL} >= 0;
param length >= 0;
param operationTime >= 0;
param velocity >= 0.5;
param laborCost >= 0;
param largenumber:= 100;
param epsi:= 0.0001;
param distance:= 10;
var x{CART,MATERIAL} binary;
var q{CART,MATERIAL} >= 0 integer;
var t:=distance/velocity, <= operationTime;
var labor{CART} >= 0; #binary
var numberofround >= 1 integer;
minimize Total_Cost:
laborCost*(sum{i in CART} labor[i]);
subject to rounds1:
  numberofround >= operationTime/t-1;
subject to rounds2:
  numberofround <= operationTime/t;
subject to forced_constraint1{i in CART}:
  labor[i]=1;
subject to q_constraint1{i in CART,j in MATERIAL}:
  q[i,j]<=x[i,j]*largenumber;
subject to time_constraint{i in CART}:
  (sum{j in MATERIAL} 1.62*(volume[j]*q[i,j]+ 0.00637*weight[j]*q[i,j] + epsi)0.5) + distance/velocity <= t;
subject to capacity_constraint{i in CART}:
  sum{j in MATERIAL} weight[j]*q[i,j] <= capacity[i];
subject to cartvolume_constraint{i in CART}:
  sum{j in MATERIAL} volume[j]*q[i,j] <= cartVolume[i];
subject to demand1_constraint{j in MATERIAL}:
  (sum{i in CART} q[i,j])*numberofround>=demand[j];
subject to demand2_constraint{j in MATERIAL}:
  (sum{i in CART} q[i,j])*numberofround<=demand[j]*1.05;
```

In this model, we straightly follow our problem formulation in Section 3.3 with some special notes.

1. The forced_constraint1 assigns each labor for each cart in every guess of the number of labors.

2. The variable numberofround is enforced to be \( \lceil \frac{T}{t} \rceil \) in constraint rounds1 and rounds2.
4.2. MINLP Solver

Solving a mixed-integer programming with non-linear constraints is generally hard. The MINLP solver [18] relaxes the problem at hand into successively close non-linear problems. In this process, the MINLP solver uses outer approximation methods (OA), branch and bound (BB), and generalize Bender’s decomposition (GBD). These approaches provide the globally optimal solution under generalized convexity. In our approach, however, after many attempts to solve the full model in which we see the number of labors itself a decision variable, the solver could not provide any solutions. Hence, we fix the number of labors for each trial and solve the sub-optimization model with this assumption.

4.3. Data in the Model

We totally have 15 materials. Their unit weights, unit volumes, and demand in each manufacturing cycle are provided in Table 1.

<table>
<thead>
<tr>
<th>Material</th>
<th>unit weight (g)</th>
<th>unit volume (cm$^3$)</th>
<th>demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptor spindle motor</td>
<td>132</td>
<td>3</td>
<td>7020</td>
</tr>
<tr>
<td>Chassis loader</td>
<td>78</td>
<td>9</td>
<td>12325</td>
</tr>
<tr>
<td>Tray loader</td>
<td>55</td>
<td>11</td>
<td>7011</td>
</tr>
<tr>
<td>Plate up,down</td>
<td>40</td>
<td>3</td>
<td>7011</td>
</tr>
<tr>
<td>Frame traverse</td>
<td>120</td>
<td>3</td>
<td>7011</td>
</tr>
<tr>
<td>Assembly panel</td>
<td>67</td>
<td>16</td>
<td>7011</td>
</tr>
<tr>
<td>Door</td>
<td>33</td>
<td>11</td>
<td>7011</td>
</tr>
<tr>
<td>Plastic round cover</td>
<td>58</td>
<td>3</td>
<td>7011</td>
</tr>
<tr>
<td>Plate traverse</td>
<td>280</td>
<td>5</td>
<td>7011</td>
</tr>
<tr>
<td>Cover top</td>
<td>273</td>
<td>8</td>
<td>7011</td>
</tr>
<tr>
<td>Cover bottom</td>
<td>116</td>
<td>7</td>
<td>14023</td>
</tr>
<tr>
<td>Weight balance</td>
<td>227</td>
<td>3</td>
<td>14023</td>
</tr>
<tr>
<td>Bubble bag</td>
<td>5</td>
<td>1</td>
<td>7011</td>
</tr>
<tr>
<td>Cushion</td>
<td>20</td>
<td>45</td>
<td>1402</td>
</tr>
<tr>
<td>Paper box</td>
<td>37</td>
<td>18</td>
<td>701</td>
</tr>
</tbody>
</table>
4.4. Optimal Solutions

We finally solve the problem by the MINLP solver and the minimal (optimal) number of labor is 31. Values of all $x_{ij}, \forall i = 1, \cdots, 31; j = 1, \cdots, 15$ are summarized in Table 2, in which the horizon indices are for materials while the vertical indices are for labors. Values of all $q_{ij}, \forall i = 1, \cdots, 31; j = 1, \cdots, 15$ are summarized in Table 5, in which the horizon indices are for materials while the vertical indices are for labors. Besides, the value of the cost function is 3,100. The material supply time, $t$, is 651.755 seconds. And the number of rounds for all labors to go during each manufacturing cycle is 5.

4.5. Result Comparison

Before our work on this project, that factory recruit 40 labors for the 1st floor for material supply. Those labors work in the way that each of them takes care of a certain kind of carts, which means each labor only picks up certain kinds of materials. Such a strategy is convenient for the manager, but rather the best way from the point of view of saving cost. That is why our goal is to minimize the number of labors while maintaining necessary production rate of the assembly lines in the 2nd floor. We also solve the case in which the number of labors is 40 using our model. From the solutions, which is not included in this report, we see that all labors are assigned less work than the sub-optimal solution of 31 labors. This means the factory does not fully utilize all its labors in its initial labor task allocation. The percentage of labor force we reduce for that factory is

$$ P = \frac{40 - 31}{40} \times 100\% = 22.5\% $$

(26)

And the total saving per month in labor cost can be calculated as

$$ S = (40 - 31) \times 15 \times 8 \times 22 = $23,760 $$

(27)

where we assume that labor cost per hour is $15, and each labor works 8 hours a day, 22 days a month.
Table 2 \( x_{ij} \) in the optimal solutions

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>27</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>31</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

5. Simulation

Since it is too complex for the mathematical model to capture the stochastic factors in the system, such as the randomness of material scanning time, material handling time, the uncertainty of the elevators’ arrival time, etc, we build up a 3D simulation of the factory, to verify the model without
Table 3 $q_{ij}$ in the optimal solutions

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>35</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>12</td>
<td>11</td>
<td>22</td>
<td>17</td>
<td>17</td>
<td>22</td>
<td>54</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>18</td>
<td>37</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>82</td>
<td>4</td>
<td>0</td>
<td>27</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>86</td>
<td>100</td>
<td>79</td>
<td>85</td>
<td>3</td>
<td>0</td>
<td>82</td>
<td>62</td>
<td>89</td>
<td>6</td>
<td>100</td>
<td>100</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>84</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>78</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>44</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>88</td>
<td>37</td>
<td>0</td>
<td>7</td>
<td>3</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>90</td>
<td>100</td>
<td>94</td>
<td>39</td>
<td>39</td>
<td>89</td>
<td>35</td>
<td>89</td>
<td>89</td>
<td>100</td>
<td>100</td>
<td>89</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>90</td>
<td>100</td>
<td>86</td>
<td>0</td>
<td>89</td>
<td>84</td>
<td>89</td>
<td>86</td>
<td>89</td>
<td>100</td>
<td>100</td>
<td>89</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>90</td>
<td>13</td>
<td>84</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>95</td>
<td>92</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>38</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>89</td>
<td>90</td>
<td>6</td>
<td>34</td>
<td>100</td>
<td>100</td>
<td>98</td>
<td>84</td>
<td>3</td>
<td>83</td>
<td>100</td>
<td>100</td>
<td>82</td>
<td>78</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>98</td>
<td>9</td>
<td>89</td>
<td>89</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>100</td>
<td>87</td>
<td>100</td>
<td>100</td>
<td>84</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>100</td>
<td>9</td>
<td>37</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>87</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>87</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>9</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>89</td>
<td>35</td>
<td>2</td>
<td>100</td>
<td>89</td>
<td>12</td>
<td>87</td>
<td>88</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>99</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>100</td>
<td>58</td>
<td>74</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>76</td>
<td>100</td>
<td>100</td>
<td>8</td>
<td>0</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>35</td>
<td>90</td>
<td>90</td>
<td>100</td>
<td>83</td>
<td>100</td>
<td>90</td>
<td>9</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>84</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>89</td>
<td>13</td>
<td>89</td>
<td>89</td>
<td>100</td>
<td>89</td>
<td>90</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>100</td>
<td>100</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>90</td>
<td>89</td>
<td>45</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>98</td>
<td>84</td>
<td>3</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>89</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>100</td>
<td>100</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>0</td>
<td>85</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>100</td>
<td>100</td>
<td>89</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>100</td>
<td>90</td>
<td>9</td>
<td>100</td>
<td>100</td>
<td>89</td>
<td>11</td>
<td>100</td>
<td>89</td>
<td>89</td>
<td>100</td>
<td>100</td>
<td>89</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>100</td>
<td>85</td>
<td>7</td>
<td>11</td>
<td>90</td>
<td>100</td>
<td>0</td>
<td>64</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>70</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>7</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>100</td>
<td>90</td>
<td>89</td>
<td>85</td>
<td>89</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>89</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>87</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>5</td>
<td>100</td>
<td>7</td>
<td>0</td>
<td>21</td>
<td>0</td>
<td>0</td>
<td>32</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>82</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>84</td>
<td>100</td>
<td>76</td>
<td>100</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>84</td>
<td>0</td>
<td>87</td>
<td>100</td>
<td>100</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>27</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>71</td>
<td>0</td>
<td>8</td>
<td>100</td>
<td>100</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>100</td>
<td>89</td>
<td>100</td>
<td>89</td>
<td>89</td>
<td>100</td>
<td>83</td>
<td>10</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>54</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>0</td>
<td>87</td>
<td>100</td>
<td>86</td>
<td>86</td>
<td>85</td>
<td>9</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>86</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>100</td>
<td>90</td>
<td>89</td>
<td>84</td>
<td>73</td>
<td>55</td>
<td>99</td>
<td>90</td>
<td>100</td>
<td>89</td>
<td>100</td>
<td>100</td>
<td>86</td>
<td>96</td>
<td>14</td>
</tr>
<tr>
<td>31</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>5</td>
<td>79</td>
<td>0</td>
<td>100</td>
<td>27</td>
<td>24</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

losing the important randomness factors in the system.

The simulation is based on the real data and internal logistic process in the factory. The process, which is much more comprehensive and complex than the model, is shown in figure 2:

The detailed pictures of each step in the logistic process are shown in the following figures:

In the simulation, the walking speed is deterministic, which is 1.2m/s, while the material handling
time, the scanning time, the time of putting materials onto the assembly lines and the time of unloading the empty boxes in the special area are normally distributed (with the tails on both sides truncated) with the following parameters:

During the simulation, we focus mainly on the material lacking rate of assembly lines and the busy/idle rate of the labors when a fixing number of labors is provided. We tested the original
mode (every worker is in charge of 3 carts) and the new mode (no cart assignments for labors) with 30 to 35 workers. We obtain the data in the following table from the simulation.

From the simulation result, we see that although 40 people are used in the original mode, there
still exists the risk of stopping assembly lines. Meanwhile, the idle rate of the workers is incredibly high (more than 70 percent!). This is a good matching of what we saw in the factory: the labors always have nothing to do and just chat with each other in most of the time. In the new mode, however, with 40 workers, we can almost guarantee that the assembly lines will not stop for lacking of materials. The idle rate is also brought down to about 60 percent.

We are not satisfied with the idle rate of the labors, however, since the rate of lacking materials is 0, we can further reduce the labors. From the simulation results, we can draw the conclusion that as the decreasing of labors, the rate of lacking materials at the assembly lines increases, and a big gap appears when reducing the workers from 34 to 33. Recall that this interesting phenomenon also appears in the queueing system, in which the congestion increases significantly when the busy rate of servers steps over a certain threshold. We show this analogy in figure 12 and 13.
An additional interesting thing we observed in the simulation is that when the number of labors is greater than 37, most of the carts stay upstairs, which means that the bottleneck is at the second floor. While when the number of labors is less than 31, most of the carts stop at the first floor. When the number of labors ranges from 31 to 37, the carts are approximately equally distributed in both floors in long run. Thus, the result, which coincides with the conclusion of the mathematical model, is that the number of labors should be chosen between 31 and 37.

In sum, based on the result from the simulation, if the rate of lacking materials is highly unacceptable, we can adopt the plan of 35 workers, while if the rate of about 1 percent is acceptable, we shall consider 34 workers. However, plans with fewer workers are not favorable since high material lacking rate on the assembly lines will be observed. This result (34 or 35) workers is not far away from the result of our model (about 31 workers). Realizing that many stochastic factors are considered in the simulation, we shall not be surprised that we have a higher requirement of labor force compared to the result obtained from the model. Moreover, studying the analogy of the result of our simulation to the queueing system is a potential direction for our future research.

6. Conclusions

In this research, we have considered the problem of optimizing operations in the factory. In particular, we have formulated the problem under the assumption that all labors follow the same path
to pick up materials. We solve this model to get a sub-optimal solution iteratively by guessing the number of required labors. Furthermore, with the aid of the simulation, we have visualized the impact of each factor on the operations of the factory. The results show that our current approach has yielded a significant improvement on the factory operations and has saved substantial expenses for the factory at the same time.

From the current formulation and approach, there are several research directions to tackle in the future. First, the study of the searching phase for the number of required labors needs more investigations to accelerate this process. Second, the more complicated model (briefly described in section 3.4), in which the optimal path for each labor is a decision variable, can be extended based on the current model. The theoretical background to solve this model is highly complex and will be the main research direction in our future work.

Acknowledgments
The authors gratefully acknowledge the discussion with Prof. Georgia Perakis, who helped the authors to make necessary simplification to the model.

References


http://www.gamsworld.org/minlp/siagopt.pdf