Part Supply Method for Mixed-Model Assembly Lines with Decentralized Supermarkets

Jietao Dong, Linxuan Zhang*, and Tianyuan Xiao

Abstract: In-house part supply affects the efficiency of mixed-model assembly lines considerably. Hence, we propose a reliable Just-In-Time part supply strategy with the use of decentralized supermarkets. For a given production sequence and line layout, the proposed strategy schedules tow train routing and delivery problems jointly to minimize the number of employed town trains and the traveling time, while ensuring that stations never run out of parts. To solve this problem, a mathematical formulation is proposed for each sub-problem aiming at minimizing supply cost. Then, a dynamic programming algorithm for routing and a greedy algorithm for delivery are developed, both of which are of polynomial runtime. Finally, a computational study is implemented to validate the effectiveness of the strategy, and to investigate the effects of the delivery capacity of tow trains and storage capacity of stations on supply cost.

Key words: part supply; mixed-model assembly lines; supermarket; dynamic programming; greedy algorithm

1 Introduction

An ever-increasing variety of customer requirements has popularized the use of mixed-model assembly lines, in which several models of a basic product are assembled simultaneously. Even though this production mode facilitates rapid response to requirement changes, it poses great production management-related challenges[1]. Most existing research related to line optimization deals with either line balancing for determining line configuration[2, 3] or production sequencing for scheduling the model sequence[4, 5]. However, in-house part supply has become one of the major challenges in managing high-variant mixed-model assembly lines, and it affects the production efficiency considerably. For instance, an average-sized automobile plant processes more than 13,000 containers of parts carried by more than 400 trucks every day[6]. Thus, it is very important to develop a reliable Just-In-Time (JIT) part supply method, aimed mainly at delivering the right parts to the right stations in a timely manner to prevent line stoppages and station idleness owing to part shortage.

In general, a part supply system consists of a part storage sub-system and a part delivery sub-system. For storage, parts received from different suppliers are first held in a centralized warehouse called the receiving store[7]. Then, they are transferred closer to the line and stored in a decentralized storage area, called the supermarket[8]. This method can avoid long traveling distances, which may lead to inflexibility of part supply and high inventories at stations[9, 10]. For delivery, tow trains composed of a motorized tugger and several un-powered wagons carrying part bins are adopted[11]. These trains travel on predefined routes to deliver the required parts from the supermarket to various stations.

The overall part supply process of a mixed-model assembly line is shown in Fig. 1. A supermarket serves a nearby line segment using several tow trains. Once a...
supply order from a station is generated, the required parts are assembled in bins, and loaded on empty wagons for the next tour. In each tour, a tow train with filled bins for the assigned stations starts from the supermarket and visits each station as defined in the route schedule [6]. Beside each station, there is a line-side buffer called the shooter rack where the tugger docks and filled bins in the wagons are moved to the rack, and empty bins are withdrawn from the rack without interrupting assembly operations.

The implementation of part supply gives rise to some crucial decision problems [9] such as warehousing, routing, and scheduling. This paper attempts to optimize the supply process between the supermarket and the stations of a given line segment. First, a subset of stations that are to be visited and served in the flow direction of the line should be assigned to each tow train (routing problem). Then, the delivery tours need to be scheduled for each tow train, including the departure times, quantities, and types of parts to be loaded (delivery problem).

The routing and delivery problems are typically solved separately in practice, for example, Ref. [12] formulated tow train routing as a JIT capacitated vehicle routing problem and proposed a heuristic solution procedure. However, routing and delivery are heavily interdependent. This study intends to jointly deal with both problems and minimize the total supply cost, including the fixed cost of employing tow trains and delivery cost. This problem has been seldom addressed in the literature except Ref. [10], in which a similar problem was tackled by considering the routing and scheduling problems together and developing a nested dynamic programming algorithm to obtain fixed routes and cyclic as well as non-cyclic schedules. However, that study ignored the storage capacity of stations, delivery time constraints, and loading policies. In reality, the space is scare on the line-side buffer, and parts must be replenished before they are exhausted. Therefore, it is quite necessary to consider the constraints of storage capacity and delivery time. Moreover, the loading is a crucial sub-problem in practical delivery that determines the quantities and types of parts loaded for each tour. Considering loading improves the applicability of our research. To solve the interdependent routing and delivery problems, this paper presents mathematical formulations for each of them. Then, a dynamic programming algorithm is developed for routing, and a greedy algorithm is proposed for delivery. The dynamic programming algorithm can find the optimal solution within the polynomial runtime, and the performance of the greedy algorithm is validated by comparing the computational results with those of the Kanban-controlled system. Finally, the effects of the delivery capacity of tow trains and the storage capacity of stations on the supply cost are investigated through a computational study.

The rest of this paper is organized as follows: Section 2 describes the routing and delivery problems. In
Section 3, we propose an efficient solution procedure for the two problems. Section 4 describes our computational study and Section 5 summarizes the conclusions drawn in this study.

2 Problem Definition

2.1 Situations and assumptions

This paper deals jointly with the routing and delivery of the tow trains of a supermarket, which serves a given line segment in a mixed-model assembly line. Routing entails the assignment of stations to each tow train; then, the tow train visits the assigned stations along the flow direction of the line. To avoid congestion of trains in narrow aisles, only consecutive partitions of stations are permitted. Delivering involves determining the number of tours required to be made to serve the part demands of all stations and all production cycles, as well as specifying the departure time, and quantities and types of parts loaded on a train for each tour. Consequently, the number of tow trains and the total traveling time are expected to be minimized, and these two objectives are aggregated by a cost function.

Moreover, other assumptions related to this problem are introduced as follows:

- The line layout and production sequence are known with certainty so that the part demands for all stations and cycles can be determined easily.
- Each station is assumed to process a single part type for the sake of simplicity because a station processing multiple types of parts can be viewed as several dummy stations, each of which processes a single part type, and the distances between them are zero.
- Parts are supplied in bins of a predefined size, and the part demand of a single model does not exceed the quantity that can be stored in one bin. Moreover, only filled bins are delivered to stations.
- Both the maximum number of bins stored at a station and the maximum number of bins carried by a tow train are predefined.
- Because practical cycle times are usually much shorter than traveling times, the time unit is normalized to the equivalent length of the cycle time.
- Loading and unloading times are relatively short, so they are ignored.

2.2 Mathematical formulation

Based on the aforementioned assumptions and the notations in Table 1, a mathematical formulation can be developed for each problem.

Routing problem: For a supermarket and a line segment with stations \( n = 1, \ldots, N \), determine the number of tow trains and assign each tow train to several consecutive stations to minimize the total supply cost. The decision is represented as a vector \( X = (X_1, \ldots, X_{K+1}) \), where \( X_k \) denotes the leftmost station serviced by tow train \( k \), \( k = 1, \ldots, K \), and \( X_{K+1} \) denotes the leftmost station serviced by a dummy tow train. The routing problem is formulated as follows:

\[
\begin{align*}
\min F(X) &= \omega \cdot K + \nu \sum_{k=1}^{K} P^*(X_k, X_{k+1}-1) \cdot T_{X_k, X_{k+1}-1} \\
\text{subject to} & \quad X_{k+1} \geq X_k + 1, \forall k \in K \\
& \quad T_{X_k, X_{k+1}-1} = T_{X_k, X_{k+1}-1} + T_{X_k} + T_{X_{k+1}-1}, \forall k \in K
\end{align*}
\]

(1) Function (1) seeks to minimize the fixed cost of
employing tow trains and the delivery cost, where \( P^*(X_k, X_{k+1} - 1) \) represents the minimum number of tours that a tow train needs to satisfy the part demands of stations from \( X_k \) to \( (X_{k+1} - 1) \). The constraint expressed by Eq. (2) ensures that the line segments are ordered consistently from left to right and each tow train serves at least one station. Equation (3) denotes the completion time of a tour made by a tow train.

**Delivery problem:** For a tow train with assigned consecutive stations from \( l \) to \( r \), find the departure time \( (Y_p) \) and quantities of loaded bins \( (Z_{p,n}) \) for each tour to minimize objective function (4) subject to constraints (5)–(10):

\[
\min P(l, r) \tag{4}
\]

subject to

\[
\sum_{n=1}^{r} Z_{p,n} \leq b_{\text{max}}, \forall p \in P \tag{5}
\]

\[
y_p + T_{i,r} \leq Y_{p+1}, \forall p = 1, \ldots, P - 1 \tag{6}
\]

\[
Q_n \sum_{p \in P} Z_{p,n} + S_n \geq \sum_{i=1}^{m} D_{i,n}, \forall n = l, \ldots, r, \forall m \in M \tag{7}
\]

where \( P \) is the set of tours that have arrived at station \( n \) up to cycle \( m \), and \( P = \{ p \mid y_p + T'_i + T_{i,n} \leq m, p \in P \} \);

\[
\sum_{q=1}^{p} Z_{q,n} - \sum_{m \in M} D_{m,n} - S_n \leq a_{\text{max}}, \forall n = l, \ldots, r, \forall p \in P \tag{8}
\]

where \( M \) represents the set of cycles completed before tour \( p \) arrives at station \( n \), and \( M = \{ m \mid m \leq y_p + T'_i + T_{i,n} - 1, m \in M \} \);

\[
y_p \in \{1, 2, \ldots, M\}, \forall p \in P \tag{9}
\]

\[
Z_{p,n} \in \{0, 1, \ldots, a_{\text{max}}\}, \forall n = l, \ldots, r, \forall p \in P \tag{10}
\]

Function (4) aims to minimize the number of tours, which is equivalent to minimizing the total delivery cost. If no feasible solution exists, an infinite value is returned. Constraint (5) indicates that the quantity of bins loaded in a tour cannot exceed the delivery capacity. Constraint (6) ensures that a new tour is created after the end of the last tour. Constraint (7) ensures that the production is never stalled owing to part shortage. Constraint (8) declares that the storage capacity must be satisfied while replenishing parts. Constraints (9) and (10) impose restrictions on the decision variables.

### 3 Solution Procedure

#### 3.1 Dynamic programming algorithm for routing problem

Since the route of a new tow train is related only to its left tow train, the optimal solution of the routing problem can be obtained by combining the optimal solutions of its non-overlapping sub-problems. Therefore, the routing problem can be solved by a dynamic programming algorithm\[^{[10,14]}\], which operates with a forward recursion.

The partial supply cost of a tow train serving stations from \( l \) to \( r (l \leq r) \) is defined as follows:

\[
h(l, r) = \omega + v \cdot P^*(l, r) \cdot T_{l,r} \tag{11}
\]

where \( P^*(l, r) \) denotes the minimum number of tours for satisfying the demands of stations from \( l \) to \( r \). Suppose that \( H(r) \) represents the minimum supply cost of tow trains serving stations from \( 1 \) to \( r \), and \( H(0) \) is equal to zero. Then, the following recursion is performed repeatedly for \( r = 1, \ldots, N \):

\[
H(r) = \min_{1 \leq l \leq r} \{ H(l - 1) + h(l, r) \} \tag{12}
\]

Finally, \( H(N) \) is the minimum value of the objective function and the optimal partition is derived by simple backward recovery. The detailed procedure is presented in Algorithm 1. The time complexity of the recursion is \( O(N^2) \) and that of the backward recovery is \( O(N) \). Hence, the time complexity of the entire procedure is \( O(N^2) \).

#### 3.2 Greedy algorithm for delivery problem

To minimize the number of tours, the tow train is expected to be fully loaded in each tour. However, when it is known that some stations would run out of stock, a tour has to be made even if some capacity would be wasted. Therefore, a greedy algorithm\[^{[15]}\] is proposed, which makes a locally optimal decision at each step.

Since a tour may start at the beginning of any cycle, a few auxiliary variables are calculated to determine whether a new tour is needed. If \( P \) tours have been made before cycle \( m \) and a new tour starts at the beginning of cycle \( m \), the time this new tour arrives at station \( n (l \leq n \leq r) \) is

\[
I_n = m + T'_i + T_{i,n} - 1 \tag{13}
\]

The number of parts remaining at station \( n \) at time \( I_n \) is

\[
A_n = S_n + Q_n \cdot \sum_{p=1}^{P} Z_{p,n} - \sum_{i=1}^{\min(I_n, M)} D_{i,n} \tag{14}
\]
The number of empty bins at station \( n \) at time \( I_n \) is
\[
B_n = a_{\text{max}} - \lceil A_n/Q_n \rceil
\]
where \( \lceil \cdot \rceil \) represents the round toward positive infinity.

If this new tour starts in the next cycle, that is, \( m + 1 \), the number of parts remaining at station \( n \) at the time of arrival of the new tour is
\[
C_n = \begin{cases} 
A_n - D_{l+1,n}, & I_n < M; \\
A_n, & I_n \geq M
\end{cases}
\]
(16)

Then, the new tour starts in cycle \( m \) in two cases: (1) The tow train can be fully loaded, that is, \( \sum_{n=1}^{r} B_n \geq b_{\text{max}} \), the time interval from the last tour is not shorter than the tour completion time of a tour, that is, \( Y_P + T_{l,r} \leq m \), and the buffer parts are inadequate for the remaining production, that is, \( A(n) < \sum_{l=1}^{M} D_{m,n} \), \( \forall n \); or (2) a few stations will be out of stock if the new tour starts later, that is, \( \min_{n \in \{l, \ldots, r\}} C_n < 0 \), and \( Y_P + T_{l,r} \leq m \). In the first case, the loaded quantities are scheduled to ensure that the stock out happens as late as possible. In the latter case, the loaded quantities are arranged to ensure that all shooter racks can be filled when the tow train arrives. Moreover, the satisfaction of \( \min_{n \in \{l, \ldots, r\}} C_n < 0 \) and \( Y_P + T_{l,r} > m \) means that a tour must be made but the tow train has not arrived at the supermarket from the last tour. Hence, there is no feasible solution in this case, and the procedure ends by returning an infinite value.

If a tour starts after cycle \( (M - T_l) \), the production is completed when the tow train arrives at the first station, thus rendering the tour needless. The detailed procedure is presented in Algorithm 2. Furthermore, the time complexity of the procedure is \( O(M^2N) \).

Algorithm 2 The proposed greedy algorithm for delivering
\[
\text{Input:} \quad \text{The data set, first station (l), last station (r);} \\
\text{Output:} \quad \text{Minimum number of tours (P), departure times (Y), loaded quantities of part bins (Z);}
\]
\[
\begin{align*}
1: & \quad P \leftarrow 0 \\
2: & \quad Y_P \leftarrow -T_{l,r} \\
3: & \quad \text{for } m = 1, 2, \ldots, (M - T_l) \text{ do} \\
4: & \quad \text{Calculate } A_n, B_n, C_n, I_n \text{ for } n = l, \ldots, r \text{ according to } Eqs. (13), (14), (15) \text{ and } (16), \text{ respectively} \\
5: & \quad \text{if } \left( \sum_{n=1}^{r} B_n \geq b_{\text{max}} \right) \text{ and } A(n) < \sum_{l=1}^{M} D_{m,n}, \forall n \text{ then} \\
6: & \quad \text{if } Y_P + T_{l,r} > m \text{ then} \\
7: & \quad \text{if } \min_{n \in \{l, \ldots, r\}} C_n < 0 \text{ then} \\
8: & \quad \text{There is no feasible solution, and the procedure ends with returning an infinite value} \\
9: & \quad \text{end if} \\
10: & \quad \text{else} \\
11: & \quad P \leftarrow P + 1; //a new tour \\
12: & \quad Y_P \leftarrow m; \\
13: & \quad Z_{P,n} \leftarrow 0 \text{ for } n = l, \ldots, r \\
14: & \quad \text{end if} \\
15: & \quad \text{end if} \\
16: & \quad \text{if a new tour is generated then} \\
17: & \quad \text{if } \sum_{n=1}^{r} B_n \leq b_{\text{max}} \text{ then} \\
18: & \quad Z_{P,n} \leftarrow B_n \text{ for } n = l, \ldots, r \\
19: & \quad \text{else} \\
20: & \quad \text{for } j = 1, \ldots, (M - l) \text{ do} \\
21: & \quad \text{for } n = l, \ldots, r \text{ do} \\
22: & \quad \text{if } I_n < M \text{ then} \\
23: & \quad p_x \leftarrow A_n + Z_{P,n} \cdot Q_n \\
24: & \quad p_d \leftarrow \sum_{m=I_n+1}^{M} D_{m,n} \\
25: & \quad \text{if } p_x < p_d \text{ and } Z_{P,n} < B_n \text{ then} \\
26: & \quad Z_{P,n} \leftarrow Z_{P,n} + 1 \\
27: & \quad \text{Break while } \sum_{n=1}^{r} Z_{P,n} \leq b_{\text{max}} \\
28: & \quad \text{end if} \\
29: & \quad \text{end if} \\
30: & \quad \text{end for} \\
31: & \quad \text{end for} \\
32: & \quad \text{end if} \\
33: & \quad \text{end if} \\
34: & \quad \text{end for}
\end{align*}
\]
4 Computational Study

4.1 Experimental setting

Rarely have the routing and delivery problems been solved simultaneously in the literature. Hence, there is no existing benchmark data for the joint routing and delivery problem. Consequently, we generated an instance arising from a practical factory, and an assembly line with 50 stations and 5 models is introduced. The demands of different models for different parts and the bin capacities of different parts are presented in Table 2. Because the bin size is standardized and different parts may have different sizes, the bin capacity varies with part type. Furthermore, the locations of the supermarket and the stations are predefined, so the traveling time between each station and the supermarket can be obtained easily for a given route layout.

The part demand per station per cycle ($D_{m,n}$) is not given directly, but derived from the production sequence ($L$). In this section, the length of the sequence ($|L|$) takes different values, and the model type of each production in the sequence is selected randomly. The number of production cycles is defined as $M = |L| + N - 1$, and the part demand is calculated as follows:

$$D_{m,n} = \begin{cases} R_{L,m-n+1,n}, & \text{if } n \leq m \leq n + |L| - 1; \\ 0, & \text{otherwise} \end{cases}$$

(17)

where $L_i$ indicates the model type at the $i$-th position of the production sequence, and $R_{j,n}$ indicates the part demand of model $j$ for part $n$.

Furthermore, we investigated the effects of the storage capacity of stations ($a_{\text{max}}$) and the delivery capacity of tow trains ($b_{\text{max}}$) on supply cost. To this end, different scenarios with different values of $a_{\text{max}}$ and $b_{\text{max}}$ were considered. A wagon can carry 10 bins, and in a tow train with a higher delivery capacity, a greater number of wagons are pulled by the tugger. Hence, the fixed cost of a tow train ($w$) is defined as $w = 40 + 1.5 \cdot b_{\text{max}}$, which means that the cost of a tugger is 40 and the cost of a wagon is 15. Besides, the delivery cost of a tow train per unit traveling time ($v$) is 1, and the shooter racks of all stations are considered fully filled at the start of production.

4.2 Performance evaluation

Since the dynamic programming algorithm is capable of finding the optimal solution in polynomial runtime, the performance of the part supply method relies largely on the Greedy Algorithm (GA). Therefore, the GA is used to solve the delivery schedules at a given route covering stations from $l$ to $r$; the obtained schedules specify the departure time, quantities and types of parts for each tour, consequently determining the delivery cost. To evaluate the performance of the GA, the experimental results were compared with those of the

| Part | $Q_n$ |
|------|-------|
| 1    | 12    |
| 2    | 20    |
| 3    | 11    |
| 4    | 19    |
| 5    | 13    |
| 6    | 16    |
| 7    | 11    |
| 8    | 10    |
| 9    | 11    |
| 10   | 13    |
| 11   | 19    |
| 12   | 9     |
| 13   | 18    |
| 14   | 20    |
| 15   | 14    |
| 16   | 15    |
| 17   | 16    |

Table 2 Bill of parts for each model and bin capacity for each part type.
Kanban-controlled system\cite{11}, in which the tow train circulates at given time intervals. If the maximum interval is less than the completion time for a tour, no feasible solution exists in the Kanban-controlled system. Since the completion time of a tour for a given route is constant, minimizing the number of tours is equivalent to minimizing the delivery cost, as indicated by Eq. (1). Therefore, the performances of the GA and the Kanban approach are compared in terms of the minimum number of tours.

In this subsection, the storage and delivery capacities are set to 8 and 30 bins, respectively. The leftmost station $(l)$ of the route is 1, and 16 problem instances are generated by varying the rightmost station $(r)$ from 4 to 10 and varying the length of the production sequence $(L)$ from 1000 to 4000.

Before running the algorithms, the bounds of the number of tours are estimated. The lower bound $(P_{\text{inf}})$ is calculated as

$$P_{\text{inf}} = \left\lfloor \frac{\sum_{n=1}^{r} [(\sum_{m=1}^{M} D_{m,n} - S_n)/Q_n]}{b_{\text{max}}} \right\rfloor$$

(18)

The super bound $(P_{\text{sup}})$ is calculated as

$$P_{\text{sup}} = \left\lceil \frac{M - T_l'}{T_{l,r}} \right\rceil$$

(19)

Both algorithms were coded in MATLAB and run on a x64 PC with an Intel Core i7 3.40 GHz CPU and 8 GB of RAM. The computational results are reported in Table 3, including the lower bound $(P_{\text{inf}})$, super bound $(P_{\text{sup}})$, minimum number of tours $(P_{\text{GA}})$ and runtime of the GA, and the minimum number of tours $(P_{\text{kb}})$ with the Kanban approach. It can be seen that the GA outperforms the Kanban approach in terms of the minimum number of tours, and the superiority of the GA is more significant when the route covers fewer stations. In scenarios with $r = 8$ or 10, the solution of the GA even reaches the lower bound. Moreover, it is evident that the GA is computationally efficient as the runtime increases very slowly with increasing the length of the production sequence. To conclude, the GA is very suitable for practical application.

### 4.3 Effects of storage capacity and delivery capacity on supply cost

In this subsection, the length of the production sequence was set to 2000, and the sequence was generated randomly. To investigate the effect of storage capacity on cost, three groups of tests were conducted, in which the delivery capacity was set to 30, 60, and 90 bins. In each test group, the storage capacity was varied from 5 to 19 bins in the increments of 2 bins. The computational results are shown in Fig. 2, and the average runtime of a single test is 253.43 s. It can be seen that increasing the storage capacity leads to a decrease in the delivery cost and an increase in the fixed cost. The delivery cost changes faster than the fixed cost, so the total cost decreases. It also corresponds to the ordinary rule that a higher storage capacity relaxes the corresponding constraints, thus affording a better solution. However, an excessively high storage capacity does not help a lot in decreasing the supply cost, but increases the cost of setting up stations. Hence, it is suggested that a suitable value should be set based on a multi-scenario analysis.

Similarly, three groups of tests were carried out by setting the storage capacity to 9, 12, and 15 bins in order to investigate the effect of delivery capacity on cost. In each test group, the delivery capacity was varied from 20 to 90 bins in the steps of 10 bins, and the computational results are shown in Fig. 3. The average runtime of a single test was 256.45 s. It can be inferred that by increasing the delivery capacity, the number of employed tow trains decreases and the unit cost of employing a tow train increases. As a result, the fixed cost remains nearly the same. Moreover, the number of tours decreases, so the delivery cost decreases significantly. However, an excessively high delivery capacity is also unsuitable, because the number

| Scenario | Bounds | GA | Kanban | Runtime |
|----------|--------|----|--------|---------|
| 10       | 4      | 35 | 35     | 0.15    | 42      |
| 2000     | 42     | 70 | 70     | 0.27    | 86      |
| 3000     | 106    | 135| 135    | 0.49    | 135     |
| 4000     | 140    | 189| 189    | 0.86    | 189     |
| 1        | 35     | 35 | 35     | 0.18    | 39      |
| 2000     | 71     | 71 | 71     | 0.60    | 90      |
| 3000     | 107    | 142| 142    | 1.13    | 142     |
| 4000     | 143    | 190| 190    | 1.70    | 190     |
| 10       | 45     | 45 | 45     | 0.47    | 46      |
| 2000     | 93     | 93 | 93     | 0.75    | 94      |
| 3000     | 141    | 141| 141    | 1.15    | 142     |
| 4000     | 187    | 187| 187    | 1.57    | 189     |
| 10       | 59     | 59 | 59     | 0.52    | 61      |
| 2000     | 120    | 121| 121    | 0.86    | 124     |
| 3000     | 180    | 180| 180    | 1.33    | 186     |
| 4000     | 242    | 242| 242    | 2.01    | 248     |
of tours cannot be reduced due to time constraints.

5 Conclusion

This paper simultaneously deals with the routing and delivery problems with the aim of developing a reliable JIT part supply method for high-variant mixed-model assembly lines. To this end, both problems were formulated mathematically with the objective of minimizing the total supply cost, including the fixed cost of employing tow trains and the delivery cost. Thereafter, a dynamic programming algorithm was proposed for routing, and a greedy algorithm was developed for delivery. Finally, the supply method was validated to be well suitable for practical application by comparing the experimental results of different approaches. Moreover, the effects of delivery capacity of tow trains and storage capacity of stations on the supply cost were investigated by conducting a computational study.

The work in this paper is still in progress. Because of numerous uncertain disturbances such as vehicle failures, equipment maintenance, defective parts, and rush orders, our future research will focus on dynamic part supply. Particularly, the forklifts with different delivery capacities will be considered because they can improve supply flexibility by taking different routes in different tours. Furthermore, the line inventory control is also a critical influence factor in terms of cost saving.

Acknowledgment

This research was supported in part by the National Key Technology Research and Development Program (No. 2012BAF15G01).

References

[1] J. Golz, R. Gujjula, H. O. Guenther, S. Rinderer, and M. Ziegler, Part feeding at high-variant mixed-model assembly lines, *Flexible Services and Manufacturing Journal*, vol. 24, no. 2, pp. 119–141, 2012.
[2] O. Battaia and A. Dolgui, A taxonomy of line balancing problems and their solution approaches, *International Journal of Production Economics*, vol. 142, no. 2, pp. 259–277, 2013.
[3] P. Sivasankaran and P. Shahabudeen, Literature review of assembly line balancing problems, *The International Journal of Advanced Manufacturing Technology*, vol. 73, nos. 9–12, pp. 1665–1694, 2014.
[4] N. Boysen, M. Fliedner, and A. Scholl, Sequencing mixed-model assembly lines: Survey, classification, and model
critique, *European Journal of Operational Research*, vol. 192, no. 2, pp. 349–373, 2009.

[5] J. Dong, L. Zhang, T. Xiao, and H. Mao, Balancing and sequencing of stochastic mixed-model assembly U-lines to minimise the expectation of work overload time, *International Journal of Production Research*, vol. 52, no. 24, pp. 7529–7548, 2014.

[6] D. Battini, N. Boysen, and S. Emde, Just-in-time supermarkets for part supply in the automobile industry, *Journal of Management Control*, vol. 24, no. 2, pp. 209–217, 2013.

[7] J. Gu, M. Goetschalckx, and L. F. McGinnis, Research on warehouse design and performance evaluation: A comprehensive review, *European Journal of Operational Research*, vol. 203, no. 3, pp. 539–549, 2010.

[8] D. Battini, M. Faccio, A. Persona, and F. Sgarbossa, "Supermarket warehouses": Stocking policies optimization in an assembly-to-order environment, *The International Journal of Advanced Manufacturing Technology*, vol. 50, nos. 5–8, pp. 775–788, 2010.

[9] N. Boysen, S. Emde, M. Hoeck, and M. Kauderer, Part logistics in the automotive industry: Decision problems, literature review and research agenda, *European Journal of Operational Research*, vol. 242, no. 1, pp. 107–120, 2015.

[10] S. Emde and N. Boysen, Optimally routing and scheduling tow trains for JIT-supply of mixed-model assembly lines, *European Journal of Operational Research*, vol. 217, no. 2, pp. 287–299, 2012.

[11] R. Domingo, R. Alvarez, M. Melodía Peña, and R. Calvo, Materials flow improvement in a lean assembly line: A case study, *Assembly Automation*, vol. 27, no. 2, pp. 141–147, 2007.

[12] B. S. Vaidyanathan, J. O. Matson, D. M. Miller, and J. E. Matson, A capacitated vehicle routing problem for just-in-time delivery, *IIE Transactions*, vol. 31, no. 11, pp. 1083–1092, 1999.

[13] Y. Q. Rao, M. C. Wang, K. P. Wang, and T. M. Wu, Scheduling a single vehicle in the just-in-time part supply for a mixed-model assembly line, *Computers & Operations Research*, vol. 40, no. 11, pp. 2599–2610, 2013.

[14] M. Held and R. M. Karp, A dynamic programming approach to sequencing problems, *Journal of the Society for Industrial and Applied Mathematics*, vol. 10, no. 1, pp. 196–210, 1962.

[15] J. Edmonds, Matroids and the greedy algorithm, *Mathematical Programming*, vol. 1, no. 1, pp. 127–136, 1971.

**Jietao Dong** is now a PhD candidate in Department of Automation of Tsinghua University. He received the bachelor degree from Beihang University in 2010. His main research interests include intelligent optimization, production scheduling and simulation. He has published 5 papers related with intelligent manufacturing in journals and major conferences. Till now he has participated in 3 scientific research projects.

**Linxuan Zhang** is an associate professor at State CIMS Engineering Research Center of Tsinghua University, dedicating to the study of virtual assembly and its related technology. He received the PhD degree from Tsinghua University in 1998. Concurrently he has more than 70 publications in books, journals, and major conferences, and he has participated in or presided over more than 30 scientific research projects. In 2010, he and his research group won the second prize of Science and Technology Progress of Education Ministry.

**Tianyuan Xiao** is a professor at State CIMS Engineering Research Center of Tsinghua University. He received the bachelor degree from Tsinghua University in 1970. He has been researching into system simulation, virtual manufacturing, computer integrated manufacturing (CIM), and networked manufacturing. Concurrently he has more than 200 publications in books, journals, and major conferences, and he won more than 10 awards in scientific research projects.