OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

MATH FAKULTÄT FÜR MATHEMATIK

Material Handling Tools for a Discrete Manufacturing System: A Comparison of Optimization and Simulation

Frank Werner
Fakultät für Mathematik
OvGU Magdeburg, Germany

(Joint work with Yanting Ni, Chengdu University, China)

OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

Overview of the Talk

- Introduction
- Literature Review
- Markov Decision Process Model
- Dynamic Programming Algorithm
- Numerical Experiments
- Conclusion

Introduction

Background:

- A material handling tool (MHT) is one of the essential components in a manufacturing system.
- MHTs are responsible for the transitions of the lots between the stations.
- The strategy of MHTs will impact the delivery rate, cycle time and WIP level.

Introduction

- A Markov decision process (MDP) will be applied to model the MHT system.

- A dynamic programming algorithm will be used to solve this problem.

Introduction

Two contributions are discussed in this paper:

- A systematic management method of MHTs under a discrete manufacturing will be developed using a Markov decision process. The quantified relationships between MHTs and WIP will be discussed within the constant WIP (CONWIP) methodology and constant demand.

- The dynamic MHT replenishment method of MHTs will be discussed within the theory of Little's law.

Literature Review

- Many approaches for analyzing the performance of MHTs have been proposed, etc.:
- Huang et al. (2011) study the vehicle allocation problem in a typical 300 mm wafer fabrication. They formulate it as a simulation-optimization problem and propose a conceptual framework to handle the problem.
- Chang et al. (2014) study the vehicle fleet sizing problem in semiconductor manufacturing and propose a formulation and a solution method to facilitate the determination of the optimal vehicle fleet size that minimizes the vehicle cost while satisfying time constraints.

Literature Review

- To overcome the shortcomings of simulation, some mathematical models are developed to quantify the parameters of a material handling system (MHS), such as a queuing theory model, queuing network model and a Markov chain model.
- Nazzal and McGinnis (2008) model a multi-vehicle material handling system as a closed-loop queuing network with finite buffers and general service times.
- Zhang et al. (2015) propose a modified Markov chain model to analyze and evaluate the performance of a closed-loop automated material handling system .

MDP Model

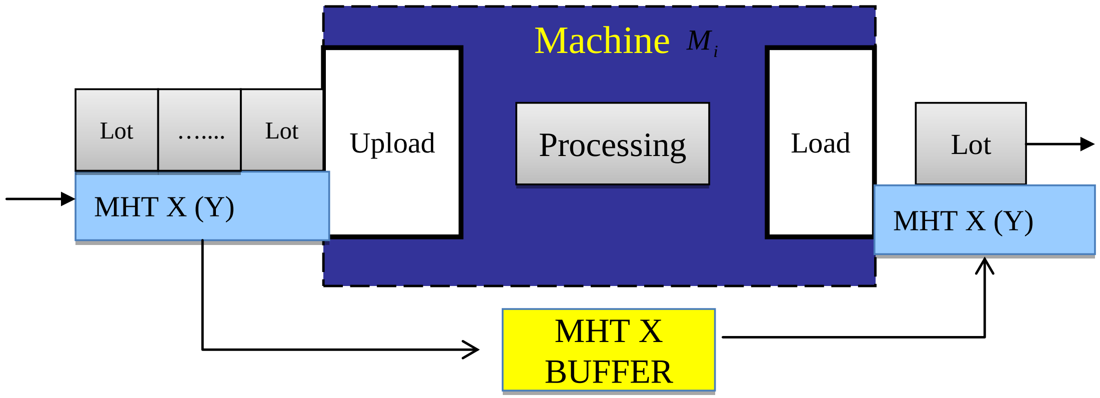
System Analysis

- In a discrete manufacturing factory, there exist many types of MHTs to carry the working lots between different stages.
- There might exist only two possible scenarios for each individual workstation - an MHT change or no change.

MDP Model

System Analysis

1.MHT no Change – **Single Cycle**



2.MHT Change – **Two-Loop Cycle**

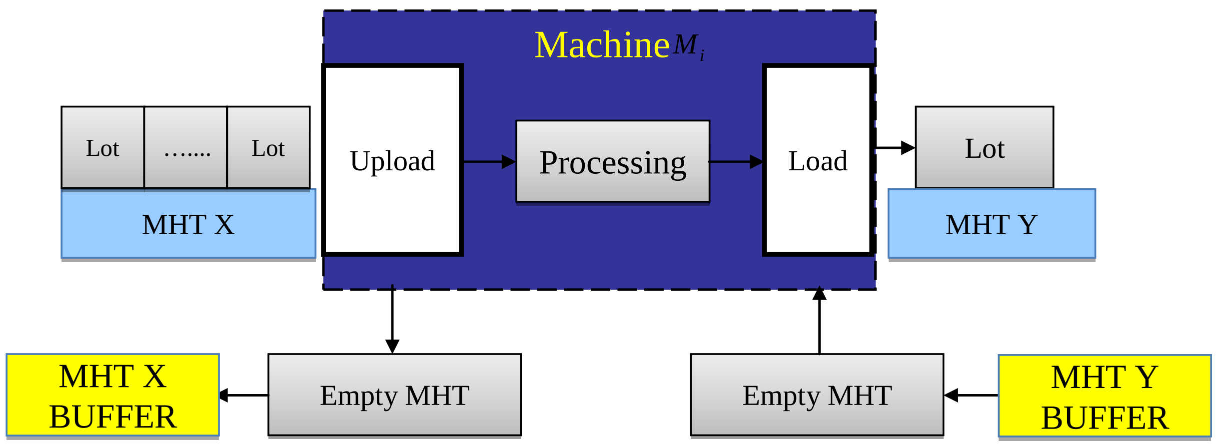


Figure 1:
MHT Cycle
Classification

MDP Model

Some basic notations:

i : Station i consisting of $M_i \geq 1$ machines

w_i : WIP quantity of station i

X : Total number of vehicles of type X

Y : Total number of vehicles of type Y

x_i : Number of vehicles X at the station i

y_i : Number of vehicles Y at the station i

MDP Model

Assumptions:

(1) The processing time at each station is constant, and production meets an M/G/1 queuing system.

(2) A lot arrives according to an exponential distribution with the associated parameter λ .

(3) Each MHT transports the lots based on the FIFO (first-in-first-out) rule.

(4) The loading time, the unloading time and the running speed of the vehicles have a deterministic value, and both acceleration and deceleration of vehicles are ignored.

MDP Model

<u>*Assumptions (cont'd)*</u>:

(5) The WIP quantity meets the CONWIP scenario and the desired WIP level is w^*.

(6) The route of the MHT at a specific work station for one specific product is fixed within the product design period.

(7) The delivery quantity is aligned with the demand of the master production schedule (MPS).

(8) Only one product is considered in this paper.

MDP Model

We use an MDP which can be described by the following 5-tuple: $(T, S, A, P_{trans}, v(SA))$

Here T describes the set of time moments, S denotes the state space, A describes the set of actions (policy set), P_{trans} gives the transition probabilities, and $v(SA)$ denotes the reward function for a solution SA described by a feasible sequence of states and actions. Subsequently, we describe the particular components more in detail.

MDP Model

Decision times T

Lots of production tasks will be released based on the numbers of available vehicles and the recycle status at each time, $t \in T = \{0, 1, 2, \ldots, L \mid L \in \mathrm{N}_{+}\}$ where L is the length of the defined production cycle.

MDP Model

Definition of the set of states S

The set of states S is composed of n sets $S_1, S_2, \dots, S_n$ .For stage i , representing station i , the set of states can be defined as $S_i = \left\{ s_i = \{x_i,\ y_i\} \middle| x_i \in \{0,1,2,\dots,X\}, y_i \in \{0,1,2,\dots,Y\} \right\}$, $i \in \{1,2,\dots,n\}$, where x_i and y_i denote the numbers of vehicles of type X and Y, respectively, at station i at a particular time t .

MDP Model

Definition of the set of actions A

At a decision moment t, the decision maker will take the action $a_i \in \{0,1\}$ and according to the transition probability $P_{trans} = P\left\{s_i^{'} \middle| s_i, a_i\right\}$ described subsequently, the numbers of vehicles of station i may change from state s_i at time t to state $s_i^{'}$ at time $t+1, i \in \{1, 2, ..., n\}$.

MDP Model

Definition of the set of actions A

The set $A=\{a_1,a_2,...,a_i,...,a_n | a_i \in \{0,1\}, i \in \{1,2,...,n\}\}$ is the production strategy set. According to the CONWIP methodology, if the WIP is higher than the desired value at station i , the station needs to stop running to avoid an excessive inventory, this means that the action $a_i=0$ is taken.

Otherwise the WIP is running normally according to first-in-first-out (FIFO) strategy, and the action $a_i=1$ is taken. At time $t \in T = \{0,1,2,...,L | L \in \mathrm{N}_+\}$, once a decision a_i has been taken, the lots will be released with the vehicle.

P_{trans}

MDP Model

State transition probabilities P_{trans}

(a) It is assumed that the vehicles arrive at station i according to a Poisson distribution with the mean arrival rate λ_i. Thus, the probability that k vehicles arrive is given by $P_i^a(k) = P(X = k) = \frac{\lambda_i^k e^{-\lambda_i}}{k!}$. In the production environment, λ_i is equal to the mean throughput of the preceding station $\overline{TH}_{i-1}$.

MDP Model

State transition probabilities P_{trans}

(b) It is assumed that the breakdown rate of work station i is q_i^d and thus, the probability of a breakdown of l machines of station i is $P_i^b(l) = \binom{M_i}{l}(q_i^d)^l(1\text{-}q_i^d)^{M_i-l}$,

where M_i is equal to the number of machines at station i.

MDP Model

State transition probabilities P_{trans}

(c) It may happen that abnormal lots are encountered, which will be cancelled. It is assumed that the lot cancellation rate is q_l^c, so the probability of the interruption of m lots is $P_i^c(m) = \binom{M_i}{m}(q_l^c)^m(1\text{-}q_l^c)^{M_i-m}$.

MDP Model

State transition probabilities P_{trans}

The state will change when new lots arrive, tasks are cancelled or a machine has a breakdown. These three events can separately occur and so the state transition probability is:

$$P_{trans}\left(s_i^{'} \middle| s_i, a_i, k, l, m\right) = P_i^a(k) \times P_i^b(l) \times P_i^c(m)$$

MDP Model

Reward Function v(SA)

The purpose of the vehicle management is to minimize ***the penalties for late deliveries*** of each product and to ***control the WIP level*** in the whole line within certain lower and upper limits. We can formulate the following optimization function as:

$$v(SA) = Max\,E[\sum_{t=0}^{L}\sum_{i=1}^{n} R_t(s_i, a_i) \,|\, s_i = (x_i, y_i), a_i \in \{0,1\}] \qquad (1)$$

MDP Model

Maximize the Reward Function v(SA)

s.t.

$$R_t(s_i, a_i) = e^{-\frac{\gamma}{D_t}}$$

$$\gamma = \sigma(D_t - \sum_{i=1}^{n} w_i)$$

The reward ratio $R_t(s_i, a_i) = e^{-\frac{\sigma(D_t - \sum_{i=1}^{n} w_i)}{D_t}}$ depending on the state s_i and action a_i which is used to maintain a rather constant WIP status (within lower and upper bounds), where $\sigma(D_t - \sum_{i=1}^{n} w_i)$ is the standard deviation to measure the offset-overflow or shortage between the demand D_t at time t and the overall WIP quantity $\sum_{i=1}^{n} w_i$ of all stations.

MDP Model (cont’d)

$$\sum_{i=1}^{n} x_i \le X$$

$$\sum_{i=1}^{n} y_i \le Y$$

The numbers of vehicles of type X and Y are not allowed to exceed the upper bounds X and Y .

$$\sum_{i=1}^{n} w_i \le \sum_{i=1}^{n} w_i^* = w^*$$

The total WIP quantity should be not greater than the total desired WIP level $\sum_{i=1}^{n} w_i^* = w^*$.

Dynamic Programming Algorithm

- The whole set of stages are grouped into 3 parts: ***a bottleneck group, a front group and a backend group.***

- The **CONWIP** methodology is used for ***the front group*** and the **FIFO** rule is used for ***backend group***.

Dynamic Programming Algorithm

Step 1: Initialization: Determine the n stages representing the stations (consisting of one or more machines) for the problem and the states to be considered in each stage. Here we can reduce the number of states at each stage since we maintain a WIP level within lower and upper bounds. The actions will be taken in stage i , $i \in \{1, 2, .., k, .., n\}$, for station i . To stage k , there is assigned s_k as the initial state, i.e., $S_k = \{s_k\}$. Both the front groups and the backend groups are initialized from stage k to make sure that the whole line WIP is controlled by the bottleneck station.

Dynamic Programming Algorithm

Step 2: Since the bottleneck station k is considered as the initial stage in this algorithm, we assign to action a_k and the WIP w_k the desired initial numbers. Then the reward value for any state $s_i = (x_i, y_i) \in S_i, i \in \{k-1, k-2, ..., 1\}$, of the front group can be determined by means of s_k in the next step.

Dynamic Programming Algorithm

Step 3: Evaluate the recurrence equations from stage $k-1$ to stage 1 and calculate the reward function value for each possible stage of the front group. Let $v_i(s_i, a_i)$ be the reward combination of station i when action a_i is taken for state s_i. The reward function for state s_i is given by

$$f_i^*(s_i) = \max\left\{ v_i(s_i, a_i(s_i)) + f_{i+1}^*(s_{i+1}) \,\middle|\, a_i(s_i) \in \{0,1\} \right\},$$

$i=k-1, k-2, \ldots, 1$.

Dynamic Programming Algorithm

Step 4: Evaluate the recurrence equations from stage k+1 to stage n and calculate the reward function value for each possible stage of the backend group. The reward function for state s_i is given by

$$f_i^*(s_i) = \max\left\{v_i(s_i, a_i(s_i)) + f_{i\text{-}1}^*(s_{i\text{-}1}) \,\middle|\, a_i(s_i) \in \{0,1\}\right\},$$

$i = k+1, k+2, \ldots, n$.

Dynamic Programming Algorithm

Step 5: Determine the states $s_1^* \in S_1$ and $s_n^* \in S_n$ with the maximal reward function values $f^*(s_1^*) = \max\left\{ f^*(s_1) \middle| s_1 \in S_1 \right\}$ and $f^*(s_n^*) = \max\left\{ f^*(s_n) \middle| s_n \in S_n \right\}$.

Dynamic Programming Algorithm

Combine the optimal solution $(s_1^*, a_1^*(s_1^*), s_2^*, a_2^*(s_2^*), \ldots, s_k^* = s_k)$ for the front group and the optimal solution for the backend group $(s_k^* = s_k, a_k^*(s_k^*), s_{k+1}^*, a_{k+1}^*(s_{k+1}^*) \ldots, s_n^*)$.

Accordingly, we can obtain an optimal state and action sequence

$$SA^t = (s_1^*, a_1^*(s_1^*), s_2^*, a_2^*(s_2^*), \ldots, s_k^*, a_k^*(s_k^*), s_{k+1}^*, a_{k+1}^*(s_{k+1}^*), \ldots, s_n^*)$$

for time t.

Dynamic Programming Algorithm

If such an optimal sequence S^t has been determined for each $t \in T$, the overall solution $(SA^0, SA^1, ..., SA^t, ..., SA^L)$ is obtained for the production cycle of length L.

Dynamic Programming Algorithm

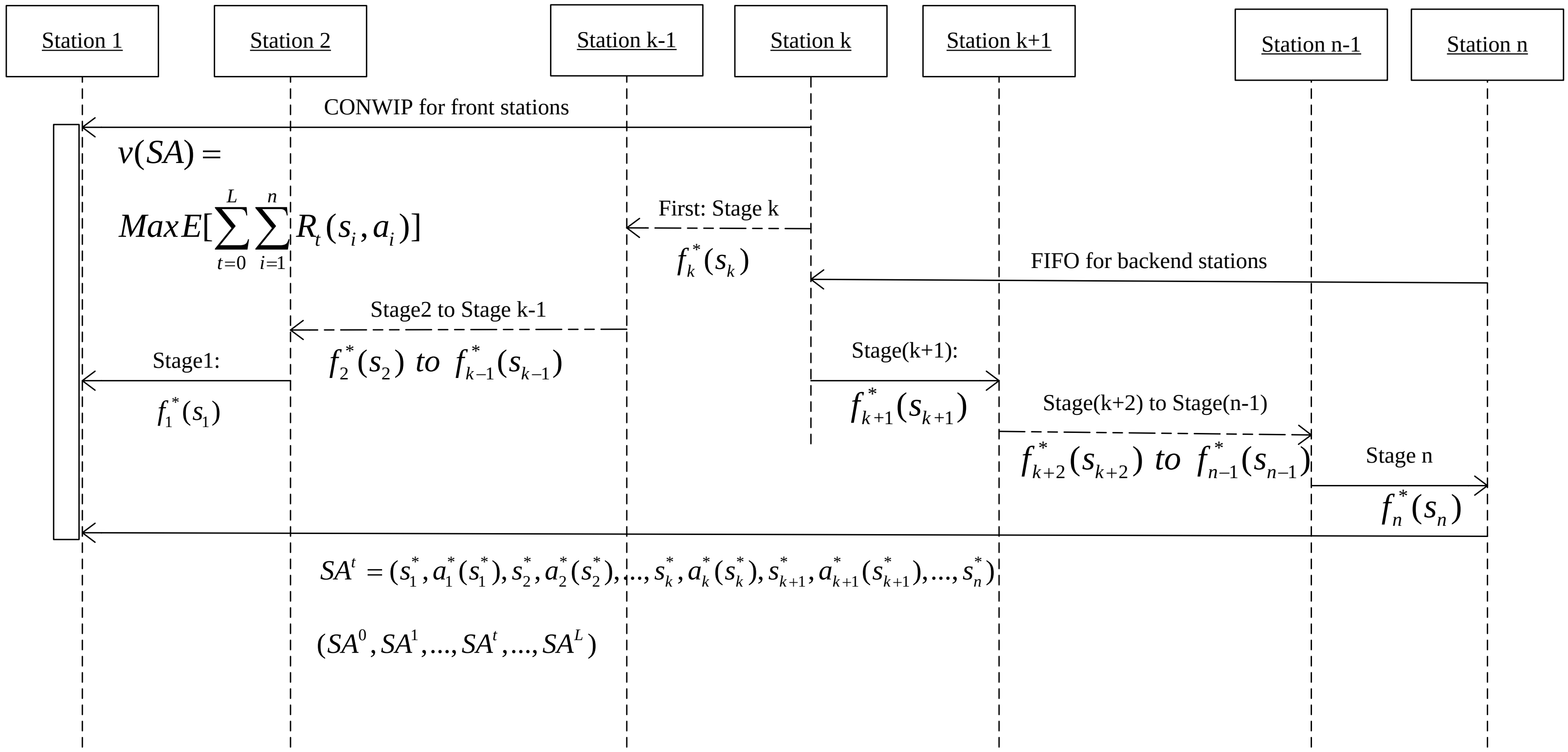


Figure 2: Sequence graph for dynamic programming

Experiments

We implemented our approach in a 300 mm semiconductor assembly and test factory and collected the required data for performing the experiments.

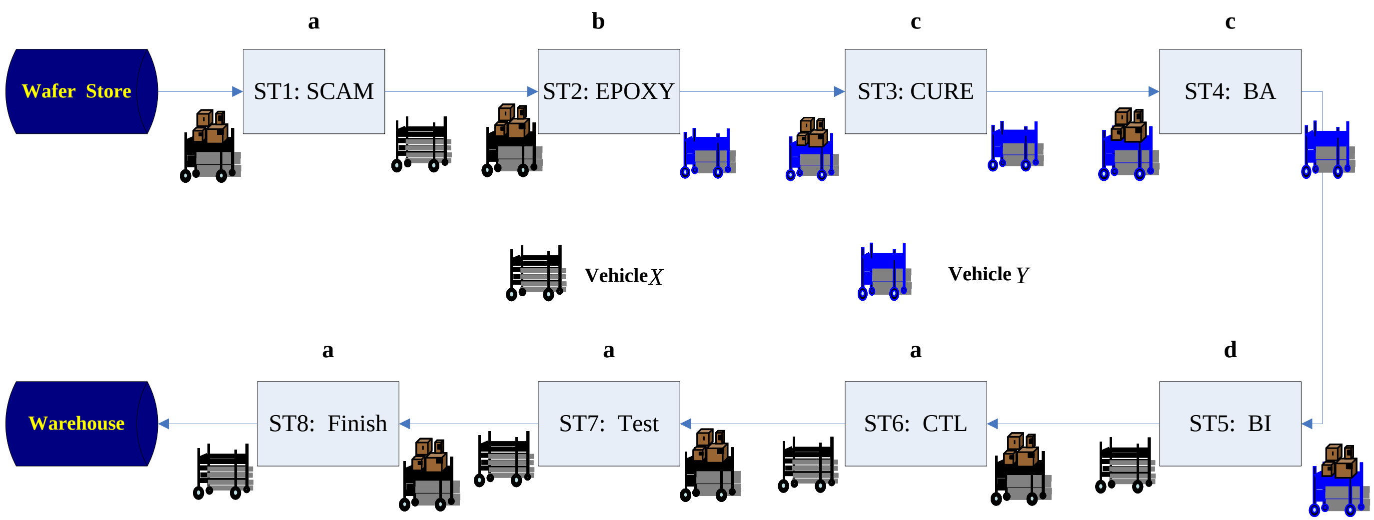


Figure 3: Workstation flow in the case factory

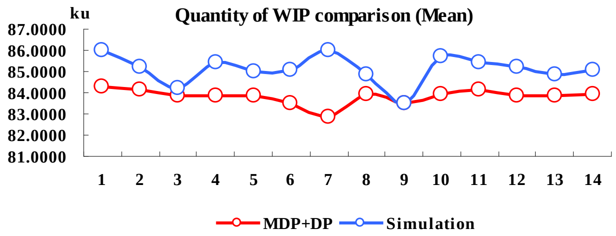
Experiments

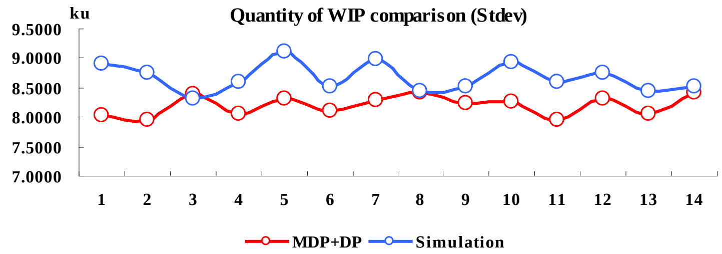
- Experiment 1: λ_1 =3.64 lots/hour

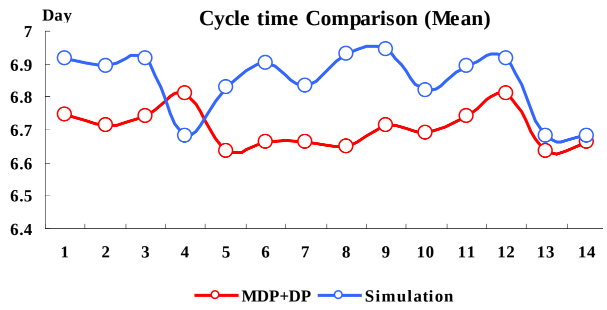
	WIP Quantity		Cycle Time		Vehicle Utilization	
	MDP+DP	Simulation	MDP+DP	Simulation	MDP+DP	Simulation
Exper1	84.289	85.976	6.748	6.917	0.818	0.709
Exper2	84.159	85.235	6.715	6.894	0.948	0.937
Exper3	83.854	84.201	6.741	6.917	0.897	0.918
Exper4	83.852	85.437	6.811	6.680	0.918	0.965
Exper5	83.825	85.035	6.637	6.828	0.993	0.812
Exper6	83.466	85.098	6.661	6.903	0.885	0.872
Exper7	82.889	86.026	6.662	6.835	0.946	0.852
Exper8	83.937	84.885	6.649	6.932	0.820	0.735
Exper9	83.508	83.496	6.713	6.943	0.992	0.893
Exper10	83.956	85.710	6.689	6.818	0.939	0.884
Exper11	84.163	85.429	6.741	6.894	0.855	0.884
Exper12	83.830	85.231	6.811	6.917	0.992	0.834
Exper13	83.860	84.885	6.637	6.680	0.749	0.970
Exper14	83.941	85.091	6.661	6.680	0.890	0.977
Deviation	-1.55%		-2.09%		2.58%	

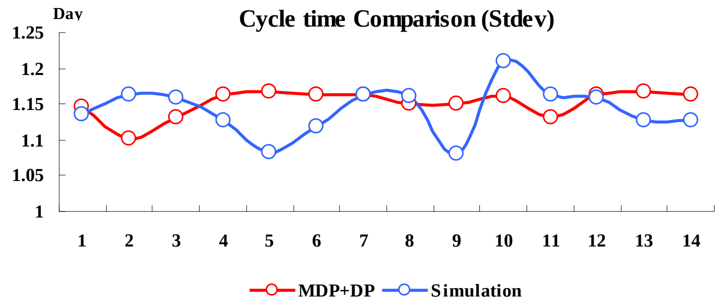
Experiments

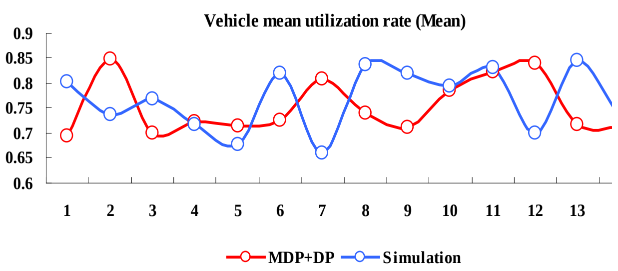
- Experiment 1: λ_1 =3.64 lots/hour

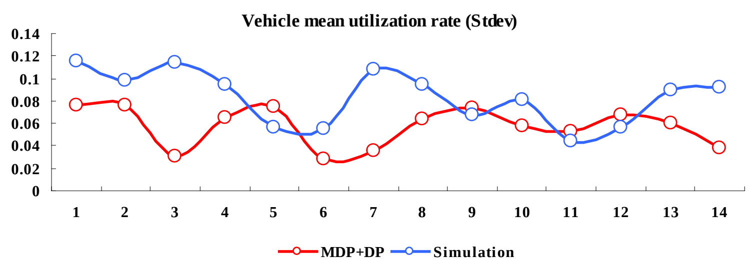












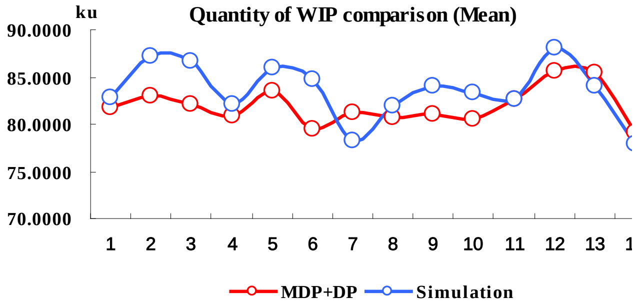
Experiments

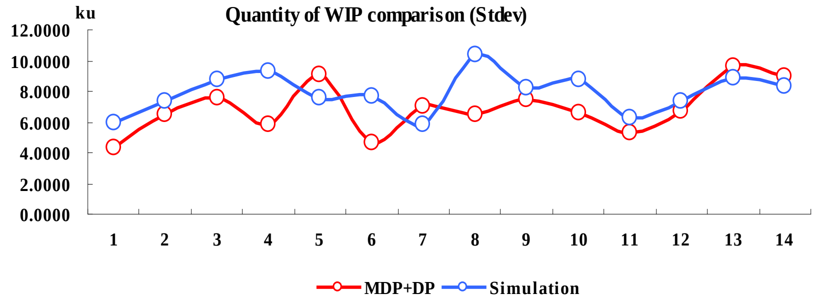
- Experiment 2: λ_2 =4.42 lots/hour

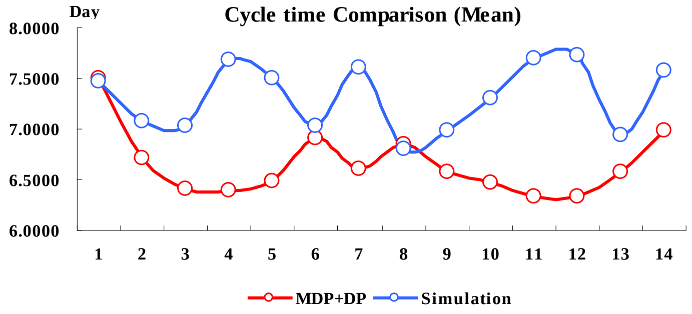
	WIP Quantity		Cycle Time		Vehicle Utilization	
	MDP+DP	Simulation	MDP+DP		MDP+DP	Simulation
Exper1	81.765	82.814	7.502	7.470	0.953	0.935
Exper2	83.015	87.271	6.705	7.075	0.910	0.862
Exper3	82.078	86.675	6.417	7.036	0.877	0.741
Exper4	80.854	82.100	6.399	7.685	0.836	0.710
Exper5	83.593	86.019	6.490	7.503	0.837	0.967
Exper6	79.487	84.660	6.911	7.035	0.858	0.869
Exper7	81.169	78.192	6.612	7.603	0.876	0.812
Exper8	80.695	82.003	6.854	6.809	0.931	0.728
Exper9	80.969	84.120	6.570	6.985	0.815	0.929
Exper10	80.533	83.253	6.470	7.300	0.889	0.710
Exper11	82.644	82.647	6.338	7.696	0.826	0.721
Exper12	85.585	88.120	6.339	7.732	0.875	0.705
Exper13	85.363	83.984	6.573	6.935	0.768	0.983
Exper14	79.082	77.828	6.992	7.577	0.875	0.720
Deviation	-2.00%		-10.21%		2.58%	

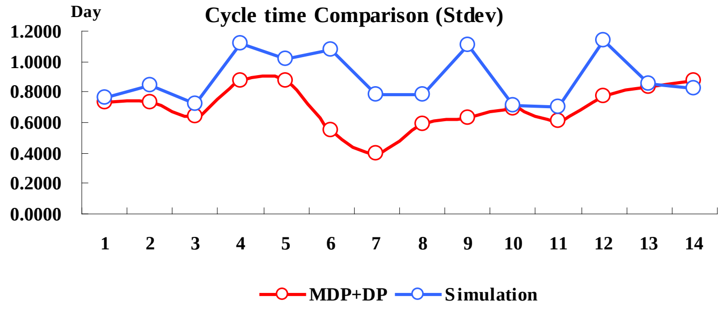
Experiments

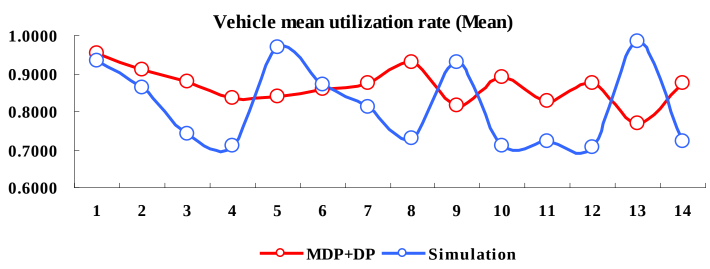
- Experiment 2: λ_2 =4.42 lots/hour

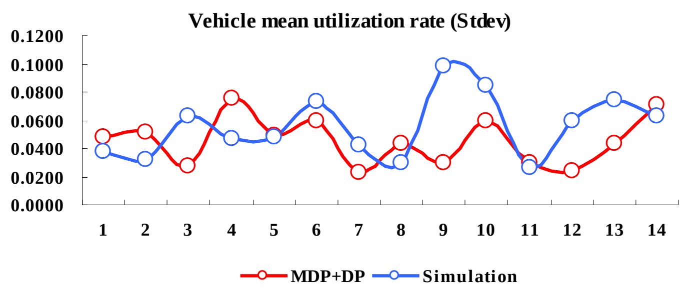












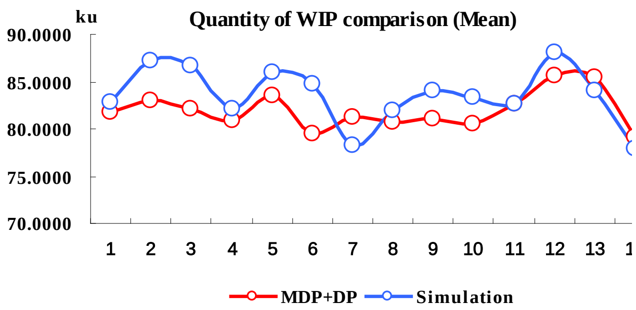
Experiments

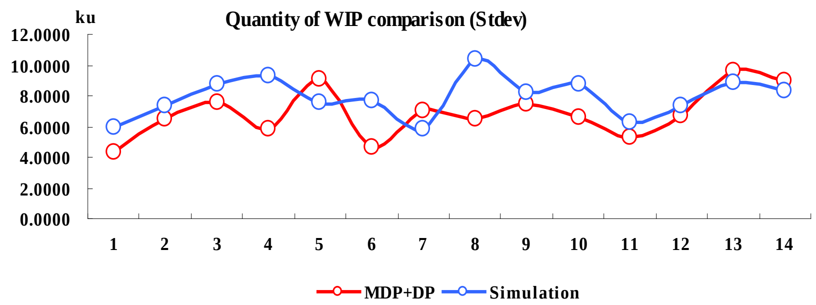
- Experiment 3: λ_3 =3.09 lots/hour

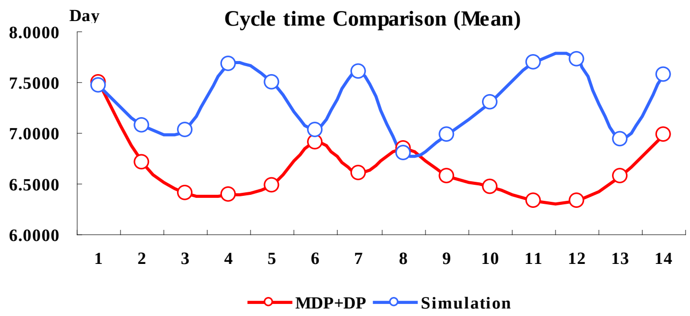
	WIP Quantity		Cycle Time		Vehicle Utilization	
	MDP+DP	Simulation	MDP+DP	Simulation	MDP+DP	Simulation
Expert1	92.777	95.486	8.126	8.444	0.818	0.709
Expert2	91.699	94.067	8.166	8.284	0.948	0.937
Expert3	86.558	95.105	8.134	8.407	0.897	0.918
Expert4	92.570	97.866	8.192	8.512	0.918	0.965
Expert5	90.923	92.830	8.206	8.429	0.993	0.812
Expert6	90.157	93.087	8.155	8.265	0.885	0.872
Expert7	87.713	95.363	8.161	8.292	0.946	0.852
Expert8	91.362	93.027	8.171	8.244	0.820	0.735
Expert9	93.254	93.764	8.211	8.390	0.992	0.893
Expert10	91.455	92.763	8.125	8.491	0.939	0.884
Expert11	89.936	95.199	8.203	8.236	0.855	0.884
Expert12	91.036	98.064	8.167	8.360	0.992	0.834
Expert13	94.786	90.904	8.238	8.496	0.749	0.970
Expert14	91.124	93.683	8.264	8.474	0.890	0.977
Deviation	-3.66%		-2.45%		2.58%	

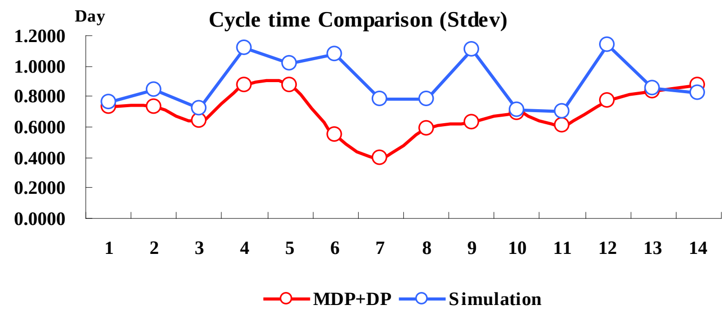
Experiments

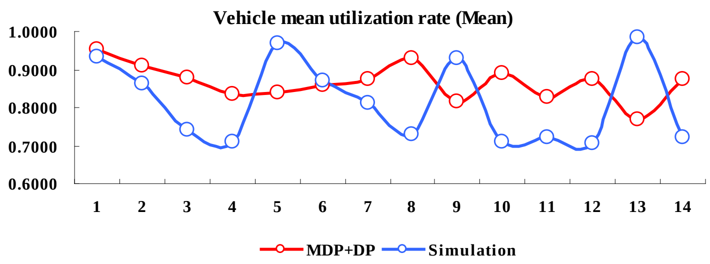
- Experiment 3: λ_3 =3.09 lots/hour

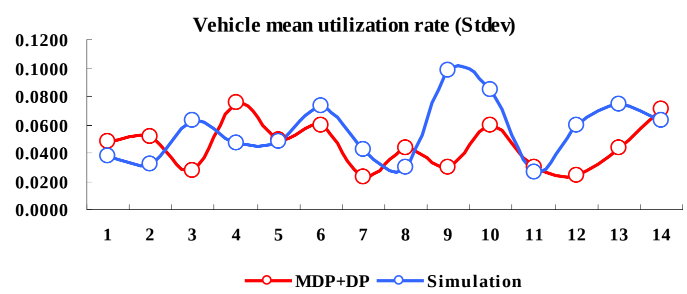












Conclusion

- The results of the experiments showed some improvements of the MDP+DP approach over simulation for the majority of the runs and confirmed that the proposed approach is both feasible and effective.

Future work

- A first extension is to generalize the model since we simplified the model by including only one product with several stations in contrast to real complex discrete manufacturing systems.

- An effective traceability method for the MHTs for the daily operations will be developed. In this way, we want to provide a **practical** method for manufacturing managers and supervisors.