Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Event-Based Optimization for Dispatching Policies in Material Handling Systems of General Assembly Lines

Yanjia Zhao, *Student Member*, *IEEE*, Qianchuan Zhao, *Member*, *IEEE*, Qing-Shan Jia, *Member*, *IEEE*, Xiaohong Guan, *Fellow*, *IEEE*, and Xi-Ren Cao, *Fellow*, *IEEE*

***Abstract*—A material handling (MH) system of a general assembly line dispatching parts from inventory to working buffers could be complicated and costly to operate. Generally it is extremely difficult to find the optimal dispatching policy due to the complicated system dynamics and the large problem size. In this paper, we formulate the dispatching problem as a Markov decision process (MDP), and use event-based optimization framework to overcome the difficulty caused by problem dimensionality and size. By exploiting the problem structures, we focus on responding to certain events instead of all state transitions, so that the number of aggregated potential function (i.e., value function) is scaled to the square of the system size despite of the exponential growth of the state space. This effectively reduces the computational requirements to a level that is acceptable in practice. We then develop a sample path based algorithm to estimate the potentials, and implement a gradient-based policy optimization procedure. Numerical results demonstrate that the policies obtained by the event-based optimization approach significantly outperform the current dispatching method in production.**

I. INTRODUCTION

GENERAL Assembly (GA) is one of the most important steps and almost the last step in production [1][2][3]. A schematic layout of GA is shown in Fig. 1. During GA, parts are assembled through a series of assembling stations. At each station operators assemble certain types of parts onto the semi-product, which is then delivered to the next station. This procedure continues until all the parts are assembled and a final product is produced. Since each station consumes certain types of parts during the assembly, the replenishment of these parts are provided by a material handling (MH) system [4]. The MH system delivers the parts from the inventory at central docking area to the lineside buffers at assembling stations. In order to have smooth production, part delivery should be in time to prevent "starvation" of the assembly line. On the other hand, in fact, how to dispatch MH system is extremely important to production efficiency since 20-50% of manufacturing costs may be related to material handling [5].

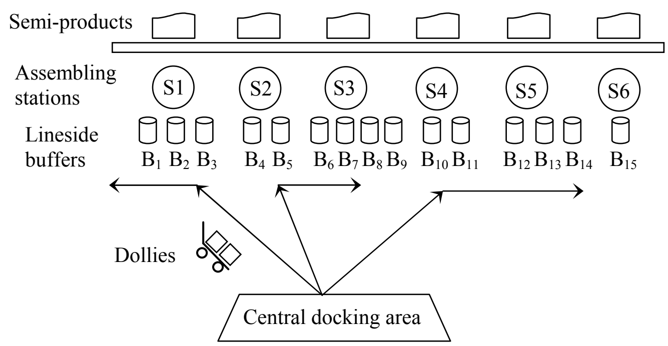

Fig. 1. Sketched general assembly line with material handling.

The dynamic transition of this MH system is triggered by the events including operation completion, part delivery, etc. It can be considered as a discrete event dynamic system (DEDS). The MH system follows a dispatching policy to determine when to send out the driver, which buffers to serve in each trip, and the serving sequences. The dispatching cost consists of three parts: the starving penalty of lineside buffers, the transportation cost and the inventory cost. We want to find the optimal dispatching policy.

The above dispatching problem is very difficult in the following senses:

1) Uncertainty. The major uncertainty in this MH system is the part consumption rates of the lineside buffers. The parts requirements are different for the products with different options, etc. This is the typical case in mixed assembly lines where different types of products are mixed and assembled in one line. This is the common practice in automotive industry [1][2][3]. Due to the requirements on high accuracy and flexibility, most operations in this GA line are manual, leading to an even larger variation in the consumption rate.
2) Policy space. A dispatching policy determines when to serve which buffers in what order. The policy space increases exponentially with the problem size determined by the number of buffers and the number of parts in the buffers, etc., and could become extremely large for the problems with practical sizes. In fact, a driver would supply dozens of buffers with sizes ranging

This work is partially supported by a contract between Tsinghua University and General Motor Corporation, and NSFC (60574067, 60736027, 60721003, 60704008), 863 High Tech Development Plan (2007AA04Z154), NCET program (NCET-04-0094), the Program of Introducing Talents of Discipline to Universities (the 111 International Collaboration Project of China), and the New Faculty Funding for Doctoral Program (20070003110).

Yanjia Zhao, Qianchuan Zhao, Qing-Shan Jia, Xiaohong Guan and Xi-Ren Cao are with Center for Intelligent and Networked Systems, Department of Automation, TNLIST, Tsinghua University, Beijing 100084, China.

Xiaohong Guan is also with SKLMS Lab and MOE KLINNS Lab, Xian Jiaotong Univeristy, Xian China.

Xi-Ren Cao is also with the Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology.

Corresponding author: zhaoyj@mails.tsinghua.edu.cn

from tens to hundreds. With little structure information of the policy space, it is in general computationally intractable to find the optimal policy by enumeration.

3) Policy evaluation. Due to the uncertainties and complicated dynamics in MH systems, it is very difficult to obtain a closed form expression for evaluating the performance of a dispatching policy [1][6]. Usually time consuming Monte Carlo simulation is the only way.

The above difficulties, together with the economic impact, attract a lot of research on dispatching policies for MH systems in the past decades. Most work focuses on moving semi-products along the serial line, assuming that the initial inventories of parts at lineside buffers are infinite [1][4][5]. Various dispatching rules [7][8] and heuristic algorithms [9][10] are well studies, based on simulation models easy to implement [11]. However, the systems with finite inventory at lineside buffers did not get enough attentions, and stochastic and dynamic dispatching problems of real-world MH systems were generally not formulated and well studied.

Markov decision process (MDP) are used to characterize sequential decision problems with Markovian properties [12][13]. However, two well-known difficulties, the large state and action spaces, prevent policy and value iterations used in traditional approaches. There are many efforts to overcome these difficulties, such as neuro-dynamic programming [14] (also known as adaptive or approximate dynamic programming [15]), state aggregation [16], time aggregation [17], and action aggregation [23]. Nevertheless, there are still no systematic formulations and solutions for the above MH dispatching problem.

In this paper, we consider the MH system of the GA line in a practical car manufacturing system. There are three salient features of our research:

1) we focus on supplying parts from the central docking area to the lineside buffers with finite inventory sizes;

2) we have a systematical MDP formulation for the stochastic and dynamic dispatching problem of a MH system;

3) we develop an Event-Based Optimization (EBO) approach [18][19] to address the aforementioned difficulties in traditional MDP approaches.

Our EBO approach is based on the structure of the MH system. By exploiting the problem structures, we focus on policies responding to certain events instead of all state transitions, so that the number of aggregated potential function (i.e., value function) is scaled to the square of the system size despite of the exponential growth of the state space [20][21]. This effectively reduces the computational requirements to a level that is acceptable in practice. We then develop a sample path based algorithm to estimate the potentials, and implement a gradient-based policy optimization procedure. Numerical results demonstrate that the policies obtained by the event-based optimization approach significantly outperform the current dispatching method in production.

II. Problem Formulation

This section presents a discrete MDP formulation for a 2-dolly MH system of a GA line. To simplify the discussion, we assume the following assumptions throughout the paper.

A1. There is one driver with a 2-dolly train in the MH system, i.e., the driver can supply two buffers at most in one trip.

A2. The number of parts consumed at each buffer at each unit time has the Bernoulli distribution, and the average consumption rate is a constant, which may be different for different buffers.

A3. To simplify the expression, we assume that the inventory level of buffers will be increased within the same time unit as the replenishment action is taken.

These assumptions have been approved by our industry partners according to the practical requirements in factories.

A. Notations

N, i number and index of lineside buffers, $i = 1, 2, \ldots, N$.

$T(i, j)$ travel time from point i to j, where $i, j = 1, 2, \ldots, N$ denote the lineside buffers; $i, j = 0$ denotes the central docking area.

C_i inventory capacity (i.e., the size) of lineside buffer i.

Q_i supplying quantity of parts for buffer i in one dolly. That is, if the driver uses one dolly of the train to supply buffer i, the amount of parts hold in the dolly is predetermined by the content of one package, i.e., Q_i.

U_i average usage rate of parts at buffer i, i.e., the average amount of parts consumed at buffer i at each time unit.

l decision epoch of the MDP formulation, $l = 0, 1, 2, \cdots$. The conveyor moves step by step in every unit time, transferring semi-products from one station to the next. We choose the decision epoch at each time unit, i.e., the time when the conveyor moves.

$n_{l,i}$ inventory level of buffer i at time l.

$M_{l,i}$ a random variable indicating the consumption amount of parts at lineside buffer i between time l and l+1.

y_l status of the driver at time l, i.e., how many time units left for the driver to come back to the central docking area. For example, $y_l = 0$ denotes that the driver is idle at time l; $y_l = 3$ shows that the driver is on trip at time l and will be back to central docking area at time l+3.

Y upper bound for total travel time of all possible trips.

$\vec{s}_l$ state vector of the Markov system at decision epoch l,

$$\vec{s}_l \triangleq (n_{l,1}, n_{l,2}, \cdots, n_{l,N}, y_l)\,. \quad (1)$$

$\mathcal{S}$ state space of the Markov system, which is defined as

$$\mathcal{S} \triangleq \{all\ \ \vec{s}_l :\ 0 \le n_{l,i} \le C_i, 0 \le y_l \le Y, n_{l,i}, y_l \in \mathbb{Z}\} \quad (2)$$

$\vec{a}_l$ action vector of the Markov system at epoch l,

$$\vec{a}_l \triangleq (a_l, a_l')\,, \quad (3)$$

where a_l and a_l' denote the index of the buffer to supply with the first and second dolly respectively at epoch l, a_l, $a_l' = 0, 1, \cdots, N$. $a_l' = 0$ means that the second dolly is not used at epoch l. Obviously, $a_l' = 0$ if $a_l = 0$, indicating a virtual trip supplying no buffers.

$\mathcal{A}$ action space of the Markov system, defined as

$$\mathcal{A} \triangleq \{all\ (a,a') : a,a' = 0,1,\cdots,N\}. \quad (4)$$

$\mathcal{A}(\vec{s}_l)$ set of all possible actions at decision epoch l with system state $\vec{s}_l$. $a_l = a_l' = 0$, if $y_l > 0$, since the driver can only be sent out when he/she is available at central docking area; a_l, $a_l' = 0, 1, \cdots, N$, if $y_l = 0$.

$\mathcal{L}$ dispatching policy of the MH system, which is a mapping form state space $\mathcal{S}$ to action space $\mathcal{A}$. We only consider the stationary Markov policies here.

$f_l(\vec{s}_l,\vec{a}_l)$ cost function at epoch l with state $\vec{s}_l$ and action $\vec{a}_l$.

$\eta^{\mathcal{L}}$ long-run average cost under policy $\mathcal{L}$.

B. Markov Decision Model

This problem can be formulated as an infinite-horizon discrete-time Markov decision process.

System dynamics

Based on the system state and the dispatching actions at decision epoch l, the state transition is:

$$n_{l+1,i} = \max\{\min\{n_{l,i} + Q_i \cdot (I_{(a_l=i)} + I_{(a'_l=i)}), C_i\} - M_{l,i}, 0\} \quad (5)$$

$$y_{l+1} = \max\{y_l + \lceil T(0,a_l) + T(a_l,a'_l) + T(a'_l,0) \rceil - 1, 0\} \quad (6)$$

where $I_{(\bullet)}$ is an indicator function which is defined as $I_{(\bullet)} = 1$ (or 0) if logic expression ($\bullet$) is true (or false); $\lceil \bullet \rceil$ is a ceiling function rounding a number upwards. With assumption A2, we have the following probability functions:

$$P(M_{l,i} = 1) = U_i,\ P(M_{l,i} = 0) = 1 - U_i, \text{ with } 0 < U_i < 1. \quad (7)$$

From the above description of the system dynamics, we can get transition probabilities.

Cost Structure

The cost function at epoch l contains three aspects: the starving penalty of lineside buffers, the transportation cost, and the inventory cost at buffers, which is calculated as:

$$f_l(\vec{s}_l,\vec{a}_l) = w_1 \cdot \sum_{i=1}^{N} I_{(n_{l,i}=0)} + w_2 \cdot I_{(y_l>0)} + w_3 \cdot \sum_{i=1}^{N} n_{l,i}, \quad (8)$$

where $w_1 \gg w_2 \gg w_3$. The priority is predetermined by our industry partners according to practical requirements.

To save the average dispatching cost of the MH system over a long period, the problem is formulated as to minimize the long-run average cost. That is:

$$\min_{\mathcal{L}} \ \eta^{\mathcal{L}} = \lim_{L\to\infty} \left\{ \frac{1}{L} E\left(\sum_{l=0}^{L-1} f_l(\vec{s}_l, \mathcal{L}(\vec{s}_l)) \right) \right\}. \quad (9)$$

From the above formulation, the challenge of large problem size is obvious. The size of the state space is $|\mathcal{S}| = \Pi_{i=1}^{N}(C_i + 1)\cdot(Y+1)$, where $|\bullet|$ indicates the cardinality of the set argument. The size of the action space is $|\mathcal{A}| = N^2$. With typical data from a practical system, the size of the state space is larger than 10^{20}, and the size of the action space is larger than 10^2. Thus the size of the traditional stated-based policy space, $|\mathcal{A}|^{|\mathcal{S}|}$, is extremely huge, which makes it difficult to optimize the dispatching policies through traditional approaches.

III. Event-Based Optimization

To address the above challenge of large problem size, this section presents a novel Event-Based Optimization (EBO) approach to optimize the dispatching policies in a MH system of a GA line. The terms used in this section follows the traditions in EBO literatures [18][19].

A. Events and event-based policies

Based on the problem structure, two most urgent buffers are more likely to be supplied in each trip since there are only two dollies for the driver in the MH system. For buffer i at decision epoch l, we define the estimated remaining life $x_{l,i}$ as the expected length of time it can maintain without supplying:

$$x_{l,i} = \left\lfloor \frac{n_{l,i}}{U_i} - T(0,i) \right\rfloor. \quad (10)$$

We sort the series of estimated remaining life ascendingly as $x_{l,(1)}$, $x_{l,(2)}$, …, $x_{l,(N)}$, and focus on the first two numbers, i.e., $x_{l,(1)}$, $x_{l,(2)}$, when making decisions.

An *event* $e_0(x_{(1)}, x_{(2)})$ is defined as a set of state transitions that the driver becomes idle and the first and second shortest estimated remaining lives turn into $x_{(1)}$ and $x_{(2)}$, respectively, where subscript (1) and (2) denote the indices of the two most urgent buffers. After the observation of an event $e_0(x_{(1)}, x_{(2)})$ at time l, there are three types of actions in event-based policies: action ($a_l = (1)$, $a_l' = (2)$) dispatches the driver to supply both of the buffers (1) and (2); action ($a_l = (1)$, $a_l' = 0$) only supplies buffer (1); action ($a_l = a_l' = 0$) supplies neither of them. These three types of actions are denoted as a_{+2}, a_{+1} and a_{+0} hereafter.

In event-based policies, we can take an action only when one of the events happens, and the actions are simplified. The size of action space becomes $|\mathcal{A}'| = 3$, and the size of the event space is $|\mathcal{E}| = N(N-1)\cdot C^2$, where $C = \max\{C_i, i = 1, 2, …, N\}$. Still take the typical data from the same practical system as before, we have $|\mathcal{E}| = 10^4$. Comparing with previous $|\mathcal{S}| = 10^{20}$ and $|\mathcal{A}| = 10^2$, the size of the event-based policy space, $|\mathcal{A}'|^{|\mathcal{E}|}$, has been greatly reduced. It is important to note that the size of event space is scale to square of the system size despite of the exponential growth of the state space. Although the event-based policies cannot guarantee to preserve the optimality as a tradeoff, it is consistent with intuitions of the problem structure. Numerical results in Section IV demonstrate its effectiveness and efficiency when dealing with large scale practical problems.

B. Performance sensitivity formula

A parameterized event-based policy γ can be denoted this way: $\gamma_1(x_{(1)}, x_{(2)})$ is the probability of taking action a_{+1} when event $e_0(x_{(1)}, x_{(2)})$ happens; $\gamma_2(x_{(1)}, x_{(2)})$ is the probability of taking a_{+2}; 1- $\gamma_1(x_{(1)}, x_{(2)})$ - $\gamma_2(x_{(1)}, x_{(2)})$ is the probability of a_{+0}. Let $\pi(e_0(x_{(1)},x_{(2)}))$ denote the steady-state probability of $e_0(x_{(1)}, x_{(2)})$. Let $\pi(\vec{s}\ |e_0(x_{(1)}, x_{(2)}))$ be the conditional steady-state probability that system state is $\vec{s}$ when $e_0(x_{(1)}, x_{(2)})$ happens.

Following the construction method in references [18][19], we can get the performance sensitivity formulas. Here we denote the original policy as **γ** and the perturbed policy as **γ'**. On a perturbed sample path of policy **γ'** with $L\gg1$ transitions, there are approximately $L\pi'(e_0(x_{(1)}, x_{(2)}))$ transitions at which event $e_0(x_{(1)}, x_{(2)})$ happens. Among them there are $L\pi'(e_0(x_{(1)}, x_{(2)}))\pi'(\vec{s}\ |e_0(x_{(1)}, x_{(2)}))$ transitions from state $\vec{s}$. At these points, the probability that the system transits from state $\vec{s}$ to $\vec{s}_{+2}$ is $\gamma'_2(x_{(1)}, x_{(2)})$. However it would transit to states $\vec{s}_{+1}$ or $\vec{s}_{+0}$ with probability $\gamma_1(x_{(1)}, x_{(2)})$ or probability 1- $\gamma_1(x_{(1)}, x_{(2)})$ - $\gamma_2(x_{(1)}, x_{(2)})$ respectively, on the original sample path where policy **γ** is taken. Here $\vec{s}_{+2}$, $\vec{s}_{+1}$ and $\vec{s}_{+0}$ are states after action a_{+2}, a_{+1} and a_{+0} are taken at state $\vec{s}$, respectively. Therefore, after an event $e_0(x_{(1)}, x_{(2)})$ happens with the system state $\vec{s}$, two types of jumps may happen; the probabilities of jumps from $\vec{s}_{+1}$ to $\vec{s}_{+2}$ and from $\vec{s}_{+0}$ to $\vec{s}_{+2}$ are determined by (11) and (12), respectively.

$$\gamma'_2(x_{(1)},x_{(2)})\cdot\gamma_1(x_{(1)},x_{(2)})\ ; \tag{11}$$

$$\gamma'_2(x_{(1)},x_{(2)})\cdot(1-\gamma_1(x_{(1)},x_{(2)})-\gamma_2(x_{(1)},x_{(2)}))\ . \tag{12}$$

Similarly, the probabilities of jumps from $\vec{s}_{+2}$ to $\vec{s}_{+1}$, from $\vec{s}_{+0}$ to $\vec{s}_{+1}$, from $\vec{s}_{+2}$ to $\vec{s}_{+0}$, and from $\vec{s}_{+1}$ to $\vec{s}_{+0}$ are determined by (13)-(16), respectively:

$$\gamma'_1(x_{(1)},x_{(2)})\cdot\gamma_2(x_{(1)},x_{(2)})\ , \tag{13}$$

$$\gamma'_1(x_{(1)},x_{(2)})\cdot(1-\gamma_1(x_{(1)},x_{(2)})-\gamma_2(x_{(1)},x_{(2)}))\ , \tag{14}$$

$$(1-\gamma'_1(x_{(1)},x_{(2)})-\gamma'_2(x_{(1)},x_{(2)}))\cdot\gamma_2(x_{(1)},x_{(2)})\ , \tag{15}$$

$$(1-\gamma'_1(x_{(1)},x_{(2)})-\gamma'_2(x_{(1)},x_{(2)}))\cdot\gamma_1(x_{(1)},x_{(2)})\ . \tag{16}$$

Each jump from state $\vec{s}$ to $\vec{s}\,'$ contributes to performance difference ΔF_L an amount measured by the realization factor $d(\vec{s},\vec{s}\,')=g(\vec{s}\,')-g(\vec{s})$. Finally, we add up the effects due to all the jumps through (11) to (16) together and obtain:

$$\begin{aligned}E(\Delta F_L)&=E(F'_L)-E(F_L)\\&=\sum_{e_0(x_{(1)},x_{(2)})\in\mathcal{E}}\ \sum_{\vec{s}\in I[e_0(x_{(1)},x_{(2)})]} L\pi'(e_0(x_{(1)},x_{(2)}))\pi'(\vec{s}\mid e_0(x_{(1)},x_{(2)}))\cdot\{g(\vec{s}_{+2})\\&(\gamma'_2(x_{(1)},x_{(2)})-\gamma_2(x_{(1)},x_{(2)}))+g(\vec{s}_{+1})(\gamma'_1(x_{(1)},x_{(2)})-\gamma_1(x_{(1)},x_{(2)}))\\&-g(\vec{s}_{+0})(\gamma'_1(x_{(1)},x_{(2)})-\gamma_1(x_{(1)},x_{(2)})+\gamma'_2(x_{(1)},x_{(2)})-\gamma_2(x_{(1)},x_{(2)}))\}\end{aligned} \tag{17}$$

where $\mathcal{E}$ is the event space; $I[e_0(x_{(1)}, x_{(2)})]$ denotes the input set of event $e_0(x_{(1)}, x_{(2)})$, which is defined as the set of states when event $e_0(x_{(1)}, x_{(2)})$ happens. Dividing both sides of equation (17) with L and letting $L\to\infty$, we get:

$$\begin{aligned}\eta'-\eta&=\sum_{e_0(x_{(1)},x_{(2)})\in\mathcal{E}}\ \sum_{\vec{s}\in I[e_0(x_{(1)},x_{(2)})]}\pi'(e_0(x_{(1)},x_{(2)}))\pi'(\vec{s}\mid e_0(x_{(1)},x_{(2)}))\cdot\\&\{g(\vec{s}_{+2})(\gamma'_2(x_{(1)},x_{(2)})-\gamma_2(x_{(1)},x_{(2)}))+g(\vec{s}_{+1})(\gamma'_1(x_{(1)},x_{(2)})-\gamma_1(x_{(1)},x_{(2)}))\\&-g(\vec{s}_{+0})(\gamma'_1(x_{(1)},x_{(2)})-\gamma_1(x_{(1)},x_{(2)})+\gamma'_2(x_{(1)},x_{(2)})-\gamma_2(x_{(1)},x_{(2)}))\}\end{aligned} \tag{18}$$

From (18), we have the sensitivity formulas:

$$\frac{\partial\eta}{\partial\gamma_1(x_{(1)},x_{(2)})}=\pi(e_0(x_{(1)},x_{(2)}))[\tilde{g}_{+1}(e_0(x_{(1)},x_{(2)}))-\tilde{g}_{+0}(e_0(x_{(1)},x_{(2)}))] \tag{19}$$

$$\frac{\partial\eta}{\partial\gamma_2(x_{(1)},x_{(2)})}=\pi(e_0(x_{(1)},x_{(2)}))[\tilde{g}_{+2}(e_0(x_{(1)},x_{(2)}))-\tilde{g}_{+0}(e_0(x_{(1)},x_{(2)}))] \tag{20}$$

This is the derivatives with respect to the dispatching probabilities with a give event, where the aggregated potential functions denote potential cost-to-go if taking some action when an event happens:

$$\tilde{g}_{+0}(e_0(x_{(1)},x_{(2)}))=\sum_{\vec{s}\in I(e_0(x_{(1)},x_{(2)}))}\pi(\vec{s}\mid e_0(x_{(1)},x_{(2)}))\cdot g(\vec{s}_{+0}) \tag{21}$$

$$\tilde{g}_{+1}(e_0(x_{(1)},x_{(2)}))=\sum_{\vec{s}\in I(e_0(x_{(1)},x_{(2)}))}\pi(\vec{s}\mid e_0(x_{(1)},x_{(2)}))\cdot g(\vec{s}_{+1}) \tag{22}$$

$$\tilde{g}_{+2}(e_0(x_{(1)},x_{(2)}))=\sum_{\vec{s}\in I(e_0(x_{(1)},x_{(2)}))}\pi(\vec{s}\mid e_0(x_{(1)},x_{(2)}))\cdot g(\vec{s}_{+2}) \tag{23}$$

C. Sample-path-based estimation

The aggregated potential functions for events in (21)-(23) can be estimated based on a sample path of the original system. As an example, detailed estimation method for $\tilde{g}_{+0}(e_0(x_{(1)},x_{(2)}))$ in (21) is explained here. Consider an original sample path:

$$\{\vec{s}_0,\vec{s}_1,\cdots,\vec{s}_L\}\ \text{with } L\gg1. \tag{24}$$

Denote the sequence of the time instants at which action a_{+0} is taken when the event $e_0(x_{(1)}, x_{(2)})$ is observed on the sample path as $l_1, l_2, \ldots, l_{L(x(1),x(2),+0)}$. Next, we group the set

$$\mathcal{T}_{x(1),x(2),+0} := \{l_k,\ k = 1,2,\ \ldots,\ L(x_{(1)}, x_{(2)}, +0)\} \tag{25}$$

into sub-groups

$$\mathcal{T}_{x(1),x(2),+0}=\bigcup_{\vec{s}\in I[e_0(x_{(1)},x_{(2)})]}\mathcal{T}_{\vec{s},+0}\ , \tag{26}$$

such that before the action a_{+0} is taken at $l\in\mathcal{T}_{\vec{s},+0}$, the system state is $\vec{s}$ when event $e_0(x_{(1)}, x_{(2)})$ is observed. Let $L_{\vec{s},+0}$ be the number of instants in set $\mathcal{T}_{\vec{s},+0}$. Then we have:

$$L(x_{(1)}, x_{(2)}, +0)=\sum_{\vec{s}\in I[e_0(x_{(1)},x_{(2)})]}L_{\vec{s},+0}\ . \tag{27}$$

Choose a large integer K. Set

$$g_k=\sum_{l=k}^{k+K}f(\vec{s}_l)\ . \tag{28}$$

From the above definitions, we have

$$\begin{aligned}&\frac{1}{L(x_{(1)},x_{(2)},+0)}\sum_{k=1}^{L(x_{(1)},x_{(2)},+0)}g_{l_k}\\&=\sum_{\vec{s}\in I[e_0(x_{(1)},x_{(2)})]}\frac{L_{\vec{s},+0}}{L(x_{(1)},x_{(2)},+0)}\cdot\left[\frac{1}{L_{\vec{s},+0}}\sum_{k\in\mathcal{T}_{\vec{s},+0}}g_{l_k}\right].\end{aligned} \tag{29}$$

By definition, we have

$$\lim_{K\to\infty}\lim_{L_{\vec{s},+0}\to\infty}\frac{1}{L_{\vec{s},+0}}\sum_{k\in\mathcal{T}_{\vec{s},+0}}g_{l_k}=g(\vec{s}_{+0})\ ; \tag{30}$$

$$\lim_{L(x_{(1)},x_{(2)},+0)\to\infty}\frac{L_{\vec{s},+0}}{L(x_{(1)},x_{(2)},+0)}=\pi(\vec{s}\mid e_0(x_{(1)},x_{(2)}))\ . \tag{31}$$

Thus, with equations (28) - (31), we can estimate the aggregated potential $\tilde{g}_{+0}(e_0(x_{(1)},x_{(2)}))$ based on the sample path this way:

$$\lim_{L(x_{(1)},x_{(2)},+0)\to\infty} \frac{1}{L(x_{(1)},x_{(2)},+0)} \sum_{k=1}^{L(x_{(1)},x_{(2)},+0)} g_{l_k} = \tilde{g}_{+0}(e_0(x_{(1)},x_{(2)})) \quad (32)$$

Similarly, we can develop the sample path based estimation algorithm for aggregated potential functions $\tilde{g}_{+1}(e_0(x_{(1)},x_{(2)}))$ and $\tilde{g}_{+2}(e_0(x_{(1)},x_{(2)}))$ in (22) and (23).

D. Performance optimization

Gradient-based optimization procedure can be developed based on the performance sensitivity formulas (19) and (20) as Box 1 shows.

The gradient-based policy optimization procedure:

- Step 1: *initialization*. Randomly pick an initial policy $\mathcal{L}^0$ from the event-based policy space, set $j = 0$.
- Step 2: *policy evaluation*. Let $\mathcal{L}^j = \{\gamma_1(x_{(1)},x_{(2)}), \gamma_2(x_{(1)},x_{(2)}); e_0(x_{(1)},x_{(2)}) \in \mathcal{E}\}$ denote the policy used in iteration j. Obtain the aggregated potential functions for events in (21) (22) (23) by estimating on a sample path of the system under policy $\mathcal{L}^j$ as formula (32) explains.
- Step 3: *policy improvement*. Obtain the performance derivatives over policies with any event $e_0(x_{(1)}, x_{(2)})$ through formulas (19) and (20). Update the policies to $\mathcal{L}^{j+1}$ according to the routine gradient based optimization approach (such as hill climbing) with steepest descent direction and constant stepsize.
- Step 4: *stopping criterion*. If $\left|\eta^{\mathcal{L}^{j+1}} - \eta^{\mathcal{L}^{j}}\right| > \epsilon$, let $j = j+1$ and go to step 2; otherwise, stop.

Box 1. The gradient-based policy optimization procedure with EBO.

Similar to the standard gradient-based optimization method, such as hill climbing, this novel EBO approach with the performance sensitivity formulas is guaranteed to converge to local optimal. Additionally, the following distinguished features make our EBO approach outstanding.

1) The potential functions are aggregated with events and the number of potentials to be estimated is reduced significantly. With aggregation, the number of potential functions of events to be estimated scales to the square of the system size, while the number of system states grows exponentially in the system size. Thus the policy space is reduced and significant computations are saved.

2) The aggregated potential functions can be estimated on a sample path under original policy, since they depend only on the original policy as (21)-(23) show. Note that estimation of aggregated potential functions of events requires the same computation and has the same accuracy as estimation of potential functions of states [18].

3) In our EBO approach, the potential aggregation is carried out by directly using the structure property of the system, and avoids the tedious effort in finding and storing the large transition probability matrix.

IV. Numerical Results

A. Comparison with practical method in industry

The practical dispatching policy currently applied in automotive industry is the so called Reorder-Point (RP) policy. r denotes a predefined threshold (*reorder point*) of remaining life, i.e., the time length a buffer can maintain without replenishment. When the remaining life of buffer i at decision epoch l is less than this threshold, i.e., $n_{l,i}/U_i < r$, it generates a request to the driver. The driver serves the requests with the First Come First Serve (FCFS) principle. If there is more than one request when the driver is dispatched, use both the dollies in one trip; otherwise, use only one dolly.

Based on our previous work on building a simulation platform for a practical MH system of a GA line in automotive industry [22], here we test EBO approach with 16 sets of typical and representative data from a world-famous automotive manufacturer. We compare its performance with the current used reorder point (RP) policy in factories and an optimized RP policy with optimal r^* obtained through enumeration. The numerical results are illustrated in table I.

TABLE I, Performance Comparison with Practical Data

Data set index	Current RP	Optimized RP	EBO	Saving %
#1	15185.2	15050.7	13837.5	8.06
#2	60487.5	59835.9	55213.2	7.73
#3	10697.9	10658.6	9985.47	6.32
#4	29061.5	28243.2	25367.2	10.2
#5	17617.6	17202.4	16949.7	1.47
#6	9694.37	9692.04	9180.24	5.28
#7	13501.8	13325.2	12551.6	5.81
#8	11881.5	11728.2	9867.74	15.9
#9	12006.8	11742.2	11045.8	5.93
#10	6125.65	5737.96	5670.71	1.17
#11	10506.7	9914.34	9580.43	3.37
#12	21272.7	19987.3	16499.4	17.5
#13	9416.74	9269.58	8518.74	8.10
#14	15781.6	15441.9	12868.5	16.7
#15	13891.5	13460.3	12401.5	7.87
#16	7106.89	6668.03	6105.7	8.43
Mean				8.11

In this table (and following table II), the columns "Current RP", "Optimized RP" and "EBO" indicate the system performance under the RP policy with factory used r, RP policy with optimized r and the policy obtained from EBO approach, respectively; the column "saving" is relative difference between our approach and the optimized RP i.e., (Optimized RP – EBO) / Optimized RP*100%. A maximum iteration number is previously setup as 200 in order to make the experiment within computational acceptable time.

From table I we can see that the system saves about 8.11% costs on average with the dispatching policy exploited by EBO approach, comparing with the optimized threshold policy in automotive industry. This result demonstrates the effectiveness and efficiency of the EBO approach.

B. Testing results of parameters sensitivities

In order to test the parameter sensitivities of our EBO approach, we use the practical parameters from industry field as the base-data, and randomly generate the testing-data within a given range. First, we uniformly generate x' within a range [0.5*x*, 1.5*x*], where *x*' denotes the testing parameter, and *x* indicates the practical parameter in data set #1. The parameters here consists of $T(i, j)$, C_i, Q_i, and U_i. Then we compare the performance of EBO approach with the currently used RP policy in industry, and show the results in table II.

TABLE II, PERFORMANCE COMPARISON WITH RANDOM PARAMETERS

Test index	CURRENT RP	OPTIMIZED RP	EBO	Saving %
1	157085	152598	146814	3.79
2	267668	260623	249158	4.40
3	19497.3	19453.7	18021.9	7.36
4	13865.8	13125.7	12613.2	3.90
5	17738.7	17491.8	16959.5	3.04
6	10486.9	10410.2	9764.67	6.20
7	10146.3	9750.75	9443.62	3.15
8	12978.4	12721.4	12109	4.81
9	13574.5	12458.6	12078.4	3.05
10	12755.5	11637.9	11531.1	0.92
Mean				4.06
Var				3.22

In this experiment, we repeat the tests for 10 repetitions with different parameters uniformly and randomly generated. Table II shows that the performance of policies obtained from the EBO approach outperforms that of RP policy in industry with optimized parameters obtained through enumeration. This result demonstrates that EBO approach is not sensitive to the system parameters and can be extendedly applied in various systems robustly and effectively.

V. CONCLUSIONS

This paper models and addresses a new optimization problem for dispatching a MH system of a GA line in a practical car manufacturing system. An event-based optimization approach is first developed based on specific features of the system to meet the challenges of huge state and action space. The overall optimization framework and our approach are effective and computationally efficient, and are verified by the numerical testing results in comparison with the dispatching method currently used in automotive industry. This paper provides a convincing application of the event-based optimization framework and shed new insight on overcoming the curse of dimensionality in traditional MDP based optimization approaches.

Although the event-based policies do not guarantee the optimality, we believe the gap between our solution and the optimal one is not large since our approach is consistent with the problem structures and intuitions. Investigating this gap is the next step of our work. Our future work also includes the extension of the approach to the system with more than two dollies and the case where drivers can take actions in continuous time.

ACKNOWLEDGMENT

The authors would like to thank Prof. Yu-Chi Ho, Prof. Leyuan Shi, Mr. Chaobo Yan, Mr. Yankai Xu, Mr. Tao Sun and Dr. Li Xia for their helpful suggestions. The authors also would like to thank our industry partners for their data and financial supports. Additionally, the authors wish to thank the anonymous reviewers for their insightful comments.

REFERENCES

[1] J. Li, and S. M. Meerkov, *Production Systems Engineering*, (Preliminary Edition). WingSpan Press, Livermore, CA, 2007.

[2] S. B. Gershwin, *Manufacturing systems engineering*. Englewood Cliffs, N.J. : PTR Prentice Hall, 1994.

[3] G. M. Buxey and D. Sadjadi, "Simulation studies of conveyor-paced assembly lines with buffer capacity," *International Journal of Production Research*, vol. 14(5), pp. 607-624, 1976,.

[4] Y.-L. Chang, R. S. Sullivan, and J. R. Wtlson, "Using SLAM to design the material handling system of a flexible manufacturing system," *International Journal of Production Research*, vol. 24(1), pp. 15-26, 1986.

[5] A.-V. Ardavan, and L. Gilbert, "Loop based facility planning and material handling," *European Journal of Operational Research*, vol. 164(1), pp. 1-11, 2005.

[6] A. Buzacott and J. G. Shantikumar, *Stochastic Models of Manufacturing Systems*, Englewood Cliffs, N.J. : Prentice Hall, 1993.

[7] J. R. Montoya-Torres, "A literature survey on the design approaches and operational issues of automated wafer-transport systems for wafer fabs," *Production Planning and Control*, vol. 17(7), pp. 648-663, 2006.

[8] E. B. Hoff, and B. R. Sarker, "An overview of path design and dispatching methods for automated guided vehicles," *Integrated Manufacturing Systems*, vol. 9(5), pp 296–307, 1998.

[9] A. N. Haq, T. Karthikeyan and M. Dinesh, "Scheduling decisions in FMS using a heuristic approach," *Int J Adv Manuf Technol*, vol. 22, pp. 374–379, 2003.

[10] S. K. Das and L. Spasovic, "Scheduling material handling vehicles in a container terminal," *Production Planning & Control*, vol. 14(7), pp. 623–633, 2003.

[11] F. T. S. Chan, H. K. Chan and H. C. W. Lau, "The State of the Art in Simulation Study on FMS Scheduling: A Comprehensive Survey," *Int J Adv Manuf Technol*, vol. 19, pp. 830–849, 2002.

[12] M. L. Puterman, *Markov Decision Process: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, 1994.

[13] X.-R. Cao, "A Unified Approach to Markov Decision Problems and Performance Sensitivity Analysis," *Automatica*, vol. 36, pp. 771-774, 2000.

[14] D. P. Bertsekas, and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena, Scientific, 1996

[15] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley-Interscience, 2007.

[16] Z. Ren, B. H. Krogh, "State aggregation in Markov decision processes," In: *Proceedings of the 41st IEEE Conference on Decision and Control*, pp. 3819-3824, 2002.

[17] X.-R. Cao, Z. Ren, S. Bhatnagar, M. Fu, and Marcus S, "A time aggregation approach to Markov decision processes," *Automatica*, vol. 38(6), pp. 929-943, 2002.

[18] X.-R. Cao, *Stochastic Learning and Optimization - A Sensitivity-Based Approach*. New York: Springer, 2007.

[19] X.-R. Cao, "Basic Ideas for Event-Based Optimization of Markov Systems," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 15, pp. 169-197, 2005.

[20] X.-R. Cao, "The Relation Among Potentials, Perturbation Analysis, Markov Decision, Processes, and Other Topics," *Journal of Discrete Event Dynamic Systems*, vol. 8, pp. 71-87, 1998.

[21] X.-R. Cao and H. F. Chen, "Perturbation realization, potentials and sensitivity analysis of Markov processes," *IEEE Trans. on Automat. Control*, vol. 42, pp. 1382-1393, 1997.

[22] Y. Zhao, C. Yan, Q. Zhao, N. Huang, J. Li and X. Guan, "Efficient Simulation for Serial Production lines based on Aggregated Event-Scheduling," In: *Proceedings of the IEEE Conference on Automation Science and Engineering*, pp. 406-411, 2008.

[23] L. Xia, Q. Zhao, and Q. Jia, "A Structure Property of Optimal Policies for Maintenance Problems with Safety-Critical Components," *IEEE Transactions on Automation Science and Engineering*, vol.5(3), pp. 519-531, 2008.