

Technical report 15-019

An integrated model predictive scheme for baggage-handling systems: Routing, line balancing, and empty-cart management*

Y. Zeinaly, B. De Schutter, and H. Hellendoorn

If you want to cite this report, please use the following reference instead:

Y. Zeinaly, B. De Schutter, and H. Hellendoorn, “An integrated model predictive scheme for baggage-handling systems: Routing, line balancing, and empty-cart management,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1536–1545, July 2015. doi:[10.1109/TCST.2014.2363135](https://doi.org/10.1109/TCST.2014.2363135)

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: <https://www.dsc.tudelft.nl>

* This report can also be downloaded via https://pub.bartdeschutter.org/abs/15_019.html

An Integrated Model Predictive Scheme for Baggage-Handling Systems: Routing, Line Balancing, and Empty-Cart Management

Yashar Zeinaly, Bart De Schutter, *Senior Member, IEEE*, and Hans Hellendoorn

Abstract

This paper proposes a new strategy for integrated control of baggage handling systems. Three main control issues in baggage handling systems, namely, routing and scheduling, empty-cart management, and line balancing, are identified and a combined control approach based on model predictive control is proposed to tackle these issues in an optimal way. It is shown that the control approach can be formulated as a linear programming problem, which can be solved very efficiently, and hence the proposed approach can be extended to large-scale baggage handling systems. We illustrate the applicability and performance of the proposed approach by a case study, and we compare the results with the state-of-the-art method currently used for baggage handling systems.

Keywords

baggage handling systems, routing and scheduling, line balancing, empty-cart management, model predictive control, linear programming, iterative linear programming.

I. INTRODUCTION

IN the past decade, modern baggage handling systems [1], [2] (BHS) have been implemented in large airports to accommodate the rising demand in air travel. Such baggage handling systems are controlled by state-of-the-art techniques that are mostly tailor-made for a specific layout. However, with increasing demand, it becomes necessary to increase the efficiency and reliability of the baggage handling systems by utilizing a systematic controller design approach. Such a control approach should optimize the performance of the baggage handling system in terms of reliability and costs. A modern baggage handling system is composed of the following components [1]: Destination coded vehicles (DCVs), which are high-speed vehicles powered by linear induction motors transporting the pieces of baggage between various locations in the system. Each DCV can carry only one piece of baggage. The term DCV refers to both carts move powered by linear induction motors and passive tubs on modular conveyor elements; loading stations, where the pieces of baggage are loaded onto DCVs after entering the system (either from the check-in desks or from the transfer flights); unloading stations, where the DCVs unload the pieces of baggage; the pieces of baggage are then transported to the planes; an early baggage storage (EBS), which is an automated storage/retrieval system used to temporarily store the loaded DCVs; a network of uni-directional tracks on which the DCVs travel. This network connects the loading stations, the unloading stations, and the EBS; and a switch controller at each junction that determines the path of the DCVs that pass through that junction.

Research supported by the EU Seventh Framework Programme [FP7/2007-2013] under grant agreement no. 257462 HYCON2 Network of Excellence.

Y. Zeinaly, B. De Schutter, and H. Hellendoorn are with Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands. y.zeinaly@tudelft.nl

From a high-level control perspective, there are three main control challenges related to the baggage handling systems [2]–[4], namely, i) routing and scheduling of DCVs, ii) line balancing, and iii) empty-cart management. The routing problem is the problem of routing loaded DCVs from the loading stations to the unloading stations or to the EBS and the problem of routing the DCVs from the EBS to the unloading stations. Line balancing is the problem of dynamically assigning empty DCVs located at the unloading stations to the loading stations. Closely related to line balancing is empty-cart management, which is the problem of routing empty DCVs from the unloading stations, through the network, to their assigned loading stations.

The control problems in baggage handling system can, in general, be related to operation scheduling, flow shop scheduling, and production scheduling [5], [6], or to predictive routing and flow control [7], [8]. Recently, the application of model predictive control (MPC) [9]–[11] to supply chain management has been studied in the literature [12]–[14]. In [7] the authors propose a two-level decision making process: long-term strategic level based on offline branch and bound optimization and short-term tactical level based on MPC. In [13], the authors propose a MPC scheme based on quadratic programming (QP) optimization, and [12] proposes an MPC approach based on mixed integer linear programming (MILP). Our proposed approach here differs from the literature in following ways: i) Unlike the supply chain networks where transport times are fixed or known functions of time, BHS involves queue-length-dependent transport delays, which needs special treatment when developing the model and defining the optimization problem. ii) To achieve timely arrival of DCVs within their time windows as well as possible, we introduce time-window constraints as soft constraints in the objective function. To this end, the objective function includes a time-weighted sum of DCV queues and DCV flows. In this way, we employ a different objective function from those in the literature. iii) For real-time control purposes, the proposed algorithm has to be computationally efficient for large-scale BHS. This makes MPC schemes with long prediction horizons based on MILP or nonlinear optimization practically unsuitable for large-scale BHS. Therefore, the solutions developed in the literature are not directly applicable to the baggage handling systems. To address this issue, we need to develop numerically-less-intensive schemes. In this paper we have opted for an LP-based approach.

The state-of-the-art control method for baggage handling systems addresses the routing problem by using look-up tables to control the switches at the junctions [1], [4]. These look-up tables are computed off-line for different system operation scenarios. However, such a control scheme cannot guarantee optimal performance of the system for complex network layouts. The current control method addresses the empty-cart management by decoupling the empty-DCV and loaded-DCV traffic flows using dedicated loops to transfer empty carts to unloading stations. Moreover, it relies on heuristics for line balancing. An immediate shortcoming of such a control system is that baggage handling systems have to adopt simple design layouts (e.g., composed of few loops). This limits the achievable performance of the system in terms of flexibility and baggage throughput.

Recently, it has been shown [3], [4], [15], [16] that a systematically designed control system can cope with complex network layouts. In [15], a multi-agent control strategy has been proposed for conveyor-based baggage handling systems, where each bag is dynamically assigned a path using the shortest path algorithm. The authors of [15] have shown that their algorithm outperforms the conventional control scheme used in practice, but they have not included optimal performance in their problem formulation. Model-based control of DCV-based baggage handling systems is considered in [3], where an automated way of learning routing rules has been proposed to solve the routing problem. In comparison with [3], we discuss optimality of our approach as well as on-time arrival of DCVs to the unloading stations. In [4], a solution has been proposed for the routing problem based on model

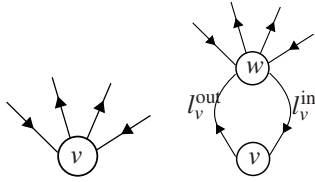


Fig. 1. Physical node v_1 , which represents a loading station, an unloading station, or the EBS (left) and its “extended” description (right)

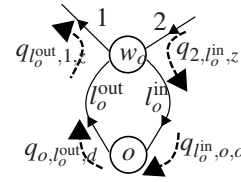


Fig. 2. Flow variables associated with an extended loading station connected to external links 1 and 2.

predictive control (MPC) [9]–[11]. In this approach, a dynamic sequence of optimal switch positions is assigned to each DCV in order to guide it along an optimal route to its destination. This approach guarantees optimal performance of the system, but it is computationally prohibitive for large-scale systems¹. A more computationally efficient MPC-based approach has been proposed in [17], which arrives at a mixed integer linear programming (MILP) formulation of the problem. In [18], it has been shown that the problem in [17] can in fact be recast as a linear programming (LP) problem, which can be solved efficiently for large-scale systems [19]. Nevertheless, to the best of our knowledge, the previous works have only focused on a particular control issue of the baggage handling systems. The aim of this paper is to propose an optimal integrated solution to routing and scheduling, empty-cart management, and line balancing.

We consider the following two criteria for an effective baggage handling system. First, the pieces of baggage should reach their assigned unloading stations within pre-specified time windows. Second, the cost of operating the system should be minimized. To achieve an overall optimal performance with respect to these criteria, in this paper, we propose a control scheme based on MPC that addresses the aforementioned control problems in *one integrated approach*. Moreover, based on the model we develop, we show that the resulting optimization problem is in general a nonlinear optimization problem the solution of which can be obtained using a general nonlinear programming (NLP) approach or a iterative linear programming (ILP) approach. We also propose a suboptimal solution based on linear programming. We then compare the performance of these three approaches to each other and to the state-of-the-art method used for BHS. In addition, we compare our proposed control approaches in terms of their computational complexity for the case study at hand.

The rest of this paper is organized as follows. In Section II, we develop the dynamical model. In Section III, we define the MPC optimization problem. A case study is presented in Section IV to illustrate the performance of the proposed control approach for a given scenario and finally, Section V concludes the paper.

II. DYNAMICAL MODEL

A. Notation and Assumptions

The baggage handling system network can be seen as a directed graph, where nodes of the graph are composed of loading stations, unloading stations, junctions, and the EBS, and where the links represent the tracks of the system. The relation between the graph representation and the real network is rather symbolic. Not all components of DCV-based BHS are shown on the graph. However, the most important components of BHS, namely, the loading stations, the junctions, the unloading stations, and

¹For instance, Amsterdam Schiphol airport operates 550 DCVs [1], and Denver International Airport operates about 4000 DCVs [2], which indicates real-life systems can indeed be large scale.

the EBS are present in our graph representation of the network. In our mathematical description of the system, as depicted in Fig. 1, we will replace physical loading stations, unloading stations, and the EBS with their “extended” description, which is a super node comprising two nodes, a unique virtual incoming link, and a unique virtual outgoing link. Please note that there is no virtual link associated with the junctions of the network, so the links connected to the junctions do represent the tracks. There are no DCV queues on the virtual links and the travel time on the virtual links represents the time needed for storing DCVs in their corresponding stacks or the time needed for loading pieces of baggage onto DCVs. Hence, each loading station and each unloading station, and the EBS has only one incoming link and one outgoing link. This considerably simplifies representation of the model. Hereafter, unless otherwise mentioned, we drop explicit reference to the “extended” prefix and use the term graph to refer to its extended version. The graph representation of the network is denoted as $G = (V, A)$, where $V = V_1 \cup V_2 \cup V_3 \cup \{v^*\}$ is the set of nodes composed of set V_1 associated with the loading stations, set V_2 associated with the intermediate nodes (i.e., junctions), set V_3 associated with the unloading stations, and the node v^* associated with the EBS. Moreover, A is the set of arcs composed of links, (i.e., physical tracks as well as virtual links) connecting the elements of V . In the sequel, we premise the following assumptions regarding configuration of the network:

- A1 Only loaded DCVs are dispatched from the loading stations.
- A2 The baggage queues at the unloading stations are ignored. This is because we assume either destination nodes have sufficient capacity or the pieces baggage are immediately transported to the planes upon arrival.
- A3 The movement of DCVs on the network is approximated by a continuous flow of DCVs.
- A4 The DCV travel time on each link is an integer multiple of the sampling time T_s .

Assumption A3 is necessary for tractability of the control problem. Even though the number of DCVs is an integer in reality, for a fairly large number of DCVs, the movement of DCVs can be approximated by continuous flows. This is not very restrictive as the computed flows can then be realized as well as possible by a lower-level control loop that determines the optimal switching pattern for the switch controllers at the junctions. Assumption A4 allows us to arrive at a linear discrete-time model of the system. In this setup, we control the flows of DCVs within the network. The flows are indexed based on their destinations, enabling us to distinguish between loaded DCVs and empty DCVs, and also loaded or empty DCVs with different destinations among themselves. The DCV flows with an index $o \in V_1$ refer to empty-DCV flows whereas the DCV flows with an index $d \in V_3$ refer to loaded-DCV flows. Consequently, partial DCV queues associated with different destinations occur along the links of the network. The total DCV queue length along a link is then given as the sum of such partial queue lengths. The system is composed of baggage and empty-DCV vertical queues at the loading stations, empty-DCV vertical queues at the unloading stations, loaded-DCV queues at the EBS, and empty and loaded DCV queues along the links of the network. Note that certain links of the network may carry both empty and loaded DCV flows. Moreover, the loading stations, unloading stations, EBS, the links have limited capacity. In our mathematical model, we make use of the following notation:

- For each node $v \in V$, L_v^{in} is the set of incoming links of v and L_v^{out} is the set of outgoing links of v .
- For each link $l = (v, w)$ of the network, $q_{l,p,z}$ is the flow from the end of link l to link $p \in L_w^{\text{out}}$, with destination z .
- For each destination z , L_z denotes to the set of links that are on some directed path to z .
- s_l is the length of link l . The speed of DCVs is denoted by v_{DCV} . Moreover T_s is the sampling time.

- For each $z \in V_3$, k_z^{open} and k_z^{close} mark, respectively, the beginning and the end of the time window of destination z , and $k_{v,z}^{\text{nom}}$ is the nominal travel time from $v \in V_1 \cup V_2 \cup \{v^*\}$ to z . Moreover, $k_{v,z}^1 = k_z^{\text{open}} - k_{v,z}^{\text{nom}}$ and $k_{v,z}^2 = k_z^{\text{close}} - k_{v,z}^{\text{nom}}$ are respectively the *relative* opening time-step and relative closing time-step of destination z as see from v .

B. Model Description

Now we will derive the dynamical model of baggage handling system in discrete time under the assumptions A1-A7. In the sequel, $x(k)$, $k = 0, 1, \dots$ denotes the value of x at time step k , and $\lceil x \rceil$ denotes the smallest integer bigger than or equal to x .

1) *Loading Stations*: For each loading station node o , let $l_o^{\text{in}} = (w_o, o)$ and $l_o^{\text{out}} = (o, w_o)$ respectively be the virtual incoming link and virtual outgoing link of o for some $w_o \in V \setminus V_1$. The control variables at each loading station are the flows of loaded DCVs, $q_{o,l_o^{\text{out}},d}$, from the DCV stack at o to l_o^{out} with destination $d \in V_3$, and the flow of empty DCVs, $q_{l_o^{\text{in}},o,o}$, from l_o^{in} to o with destination o . Fig. 2 illustrates how the flow variables are defined for the loading stations. For each outgoing link p of w_o , we need to impose the following constraints:

$$q_{l_o^{\text{out}},p,z}(k) = 0, \forall z \in V_1; \quad q_{l_o^{\text{out}},p,z}(k) = 0, \forall z \in V_3 \text{ s.t. } p \notin L_z; \quad q_{l_o^{\text{out}},p,z}(k) \geq 0, \forall z \in V_3 \text{ s.t. } p \in L_z. \quad (1)$$

For each incoming link p of w_o , we impose the following constraints:

$$q_{p,l_o^{\text{in}},z}(k) = 0, \forall z \in V_3; \quad q_{p,l_o^{\text{in}},z}(k) = 0, \forall z \in V_1 \setminus \{o\}; \quad q_{p,l_o^{\text{in}},z}(k) \geq 0, z = o. \quad (2)$$

Note that constraint (1) implies that the loading station o can only send loaded-DCV flows to the unloading stations that are reachable from o . Constraint (2) implies that the loading station o can only accept empty-DCV flows that are designated for o . The evolution of DCV stack at loading station o , $x_o(k)$, is then given by

$$x_o(k+1) = x_o(k) + T_s(q_{l_o^{\text{in}},o,o}(k) - \sum_{d \in V_3} q_{o,l_o^{\text{out}},d}(k)); \quad 0 \leq x_o(k) \leq x_{o,\max}, \quad (3)$$

where $x_{o,\max}$ is the maximum capacity of the DCV stack at loading station o . Let $x_{o,d}^{\text{bag}}(k)$ be the length of the baggage queue, with destination $d \in V_3$, at loading station o . Then, $x_{o,d}^{\text{bag}}(k)$ is described as

$$x_{o,d}^{\text{bag}}(k+1) = x_{o,d}^{\text{bag}}(k) + T_s(Q_{o,d}^{\text{bag}}(k) - q_{o,l_o^{\text{out}},d}(k)); \quad x_{o,d}^{\text{bag}}(k) \geq 0, \forall d \in V_3, \quad (4)$$

where $Q_{o,d}^{\text{bag}}(k)$ is the baggage demand at loading station o that needs to be transported to destination d . The total length of baggage queue at node o is given by $x_o^{\text{bag}}(k) = \sum_{d \in V_3} x_{o,d}^{\text{bag}}(k)$. In order to guarantee that there is no DCV queue along the virtual outgoing link l_o^{out} and the virtual incoming link l_o^{in} , their inflow and outflow must be set equal, or equivalently

$$q_{o,l_o^{\text{out}},d}(k) = \sum_{p \in L_{w_o}^{\text{out}}} q_{l_o^{\text{out}},p,d}(k + k_{l_o^{\text{out}}}^{\text{out}}), \forall d \in V_3; \quad q_{l_o^{\text{in}},o,o}(k + k_{l_o^{\text{in}}}^{\text{in}}) = \sum_{p \in L_{w_o}^{\text{in}}} q_{p,l_o^{\text{in}},o}(k), \quad (5)$$

where $k_{l_o^{\text{out}}}^{\text{out}}$ is the number of time steps required to load a piece of baggage onto the DCVs, and $k_{l_o^{\text{in}}}^{\text{in}}$ is the number of time steps that is required to store empty DCVs in the DCV stack.

2) *Unloading Stations*: For each unloading station node d , let $l_d^{\text{in}} = (w_d, d)$ and $l_d^{\text{out}} = (d, w_d)$ respectively be the virtual incoming link and virtual outgoing link of d for some $w_d \in V_2$. The control variables at each unloading station are the flows of empty DCVs, $q_{d,l_d^{\text{out}},o}$, from d to l_d^{out} with destination o , and the flows of loaded DCVs, $q_{l_d^{\text{in}},d,d}$, from l_d^{in} to d with destination d . For each outgoing link p of w_d , we impose the following constraints:

$$q_{l_d^{\text{out}},p,z}(k) = 0, \forall z \in V_3; \quad q_{l_d^{\text{out}},p,z}(k) = 0, \forall z \in V_1 \text{ s.t. } p \notin L_z; \quad q_{l_d^{\text{out}},p,z}(k) \geq 0, \forall z \in V_1 \text{ s.t. } p \in L_z. \quad (6)$$

For each incoming link p of w_d , we need to impose the following constraints:

$$q_{p,l_d^{\text{in}},z}(k) = 0, \forall z \in V_1; \quad q_{p,l_d^{\text{in}},z}(k) = 0, \forall z \in V_3 \setminus \{d\}; \quad q_{p,l_d^{\text{in}},z}(k) \geq 0, z = d. \quad (7)$$

Constraint (6) implies that unloading station d can only send empty-DCV flows to the loading stations that are reachable from d . Constraint (7) implies the unloading station d can only accept loaded-DCV flows the final destination of which is d . The evolution of the DCV stack at unloading station d is given by

$$x_d(k+1) = x_d(k) + T_s(q_{l_d^{\text{in}},d,d}(k) - \sum_{o \in V_1} q_{d,l_d^{\text{out}},o}(k)); \quad 0 \leq x(k) \leq x_{d,\text{max}}, \quad (8)$$

where $x_{d,\text{max}}$ is the maximum capacity of the DCV stack at unloading station d . Since, by definition, no DCV queues can occur along the virtual links of d , we have to set their inflow equal to their outflow, or equivalently

$$q_{d,l_d^{\text{out}},o}(k) = \sum_{p \in L_{w_d}^{\text{out}}} q_{l_d^{\text{out}},p,o}(k + k_d^{\text{out}}), \forall o \in V_1; \quad q_{l_d^{\text{in}},d,d}(k + k_d^{\text{in}}) = \sum_{p \in L_{w_d}^{\text{in}}} q_{p,l_d^{\text{in}},d}(k), \quad (9)$$

where k_d^{out} , and k_d^{in} are respectively the number of time steps that is required to release the DCVs stored in the DCV stack, and the number of time steps that is required to unload and store the DCVs in the DCV stack at the unloading station.

3) *EBS*: Let $l_{v^*}^{\text{out}} = (v^*, w^*)$ and $l_{v^*}^{\text{in}} = (w^*, v^*)$ be the virtual outgoing and incoming links of EBS, respectively. For the EBS node v^* and for each $d \in V_3$, the control variables are the outflows of loaded DCVs, $q_{v^*,l_{v^*}^{\text{out}},d}$, with destination d , and the inflows of loaded DCVs, $q_{l_{v^*}^{\text{in}},v^*,d}$, whose final destination is d . For each outgoing link p of w^* , we introduce the following constraints:

$$q_{l_{v^*}^{\text{out}},p,z}(k) = 0, \forall z \in V_1; \quad q_{l_{v^*}^{\text{out}},p,z}(k) = 0, \forall z \in V_3 \text{ s.t. } v^* \notin L_z; \quad q_{l_{v^*}^{\text{out}},p,z}(k) \geq 0, \forall z \in V_3 \text{ s.t. } v^* \in L_z. \quad (10)$$

For each incoming link p of w^* , we impose the following constraints:

$$q_{p,l_{v^*}^{\text{in}},z}(k) = 0, \forall z \in V_1; \quad q_{p,l_{v^*}^{\text{in}},z}(k) = 0, \forall z \in V_3 \text{ s.t. } v^* \notin L_z; \quad q_{p,l_{v^*}^{\text{in}},z}(k) \geq 0, \forall z \in V_3 \text{ s.t. } v^* \in L_z. \quad (11)$$

Constraints (10) and (11) jointly imply that the EBS cannot accept or send out empty-DCV flows, and that it can only receive and send loaded-DCV flows the final destination of which is reachable from the EBS. The evolution of the loaded-DCV queue lengths at the EBS with final destination $d \in V_3$ is given as

$$x_{v^*,d}(k+1) = x_{v^*,d}(k) + T_s(q_{l_{v^*}^{\text{in}},v^*,d}(k) - q_{v^*,l_{v^*}^{\text{out}},d}(k)); \quad x_{v^*,d}(k) \geq 0, \quad (12)$$

The total length of the DCV queues at the EBS is, therefore, given by $x_{v^*}(k) = \sum_{d \in V_3} x_{v^*,d}(k)$ with the constraint $x_{v^*}(k) \leq x_{v^*,\max}$, where x_{\max,v^*} is the maximum capacity of EBS. The following guarantee that no queues occur along the virtual links of EBS:

$$q_{l_{v^*},v^*,d}(k+k_{l_{v^*}}^{\text{in}}) = \sum_{p \in L_{w^*}^{\text{in}}} q_{p,l_{v^*},d}(k); \quad q_{l_{v^*},l_{v^*},d}(k) = \sum_{p \in L_{w^*}^{\text{out}}} q_{l_{v^*},p,d}(k+k_{l_{v^*}}^{\text{out}}), \quad (13)$$

where $k_{l_{v^*}}^{\text{in}}$, and $k_{l_{v^*}}^{\text{out}}$ are respectively the number of time steps that is required to store loaded DCVs in the EBS, and the number of time steps that is required to release loaded DCVs from the EBS.

4) *Links*: For each real link $l = (v, w)$ and for each $z \in V_1 \cup V_3$, the controls are the empty and loaded DCV flows, $q_{l,p,z}$, from the link l to each of its outgoing links p with destination z . For each $p \in L_w^{\text{out}}$, the flows of DCVs from l to p must satisfy

$$q_{l,p,z}(k) = 0, \quad \forall z \in V_1 \cup V_3 \text{ s.t. } p \notin L_z; \quad q_{l,p,z}(k) \geq 0, \text{ otherwise,} \quad (14)$$

which implies that the DCVs with final destination z can be sent from l to p if z is reachable from p . Let $F_{l,z}^{\text{in}}(k)$ be the sum of all DCV flows with destination z that enter link l , and let $F_{l,z}^{\text{out}}(k)$ be the sum of all DCV flows with destination z that leaves link l . We then have

$$F_{l,z}^{\text{in}}(k+k_l(k)) = \sum_{p \in L_v^{\text{in}}} q_{p,l,z}(k); \quad F_{l,z}^{\text{out}}(k) = \sum_{p \in L_w^{\text{out}}} q_{l,p,z}(k), \quad (15)$$

where $k_l(k)$ is the number of travel time steps for DCV on the link l given by $k_l(k) = \left\lceil \frac{s_l - x_l(k) l_{\text{DCV}}}{v_{\text{DCV}} T_s} \right\rceil$, where $x_l(k) = \sum_{z \in V_1 \cup V_3} x_{l,z}(k)$, $x_l(k) \leq x_{l,\max}$ is the total DCV queue length along link l with $x_{l,\max}$ being the maximum allowed queue length on link l and with $x_{l,z}(k)$ being the partial DCV queue length along link l associated with destination d described by

$$x_{l,z}(k+1) = x_{l,z}(k) + T_s (F_{l,z}^{\text{in}}(k) - F_{l,z}^{\text{out}}(k)); \quad x_{l,z}(k) \geq 0. \quad (16)$$

The equality and inequality constraints (1)-(16) define the set of feasible trajectories of the system. Let $x(k) \in \mathbb{R}^n$ be the state vector at time step k the elements of which are the baggage queue lengths at the loading stations; the empty-DCV queues at the loading stations, at the unloading stations, and on the links; the loaded-DCV queues on the links and in the EBS; the empty- and loaded-DCV flows at the past time steps, where the number of past time steps for which we need to store the flow value depends on the capacity of the link. Let $u(k) \in \mathbb{R}^{m_1}$ be the input vector the elements of which are empty- and loaded-DCV flows from the end of each link to its outgoing links; the inflow of empty DCVs from the virtual incoming link of loading stations to the loading stations and the outflow of empty DCVs from unloading stations to their virtual outgoing link; the flow of loaded DCVs to the EBS from its virtual incoming link, and from the EBS to its virtual outgoing link, and let $z(k) \in \mathbb{R}^{m_2}$ be the disturbance vector the elements of which are $Q_{o,d}(k)$ for all origin and destination pairs. Note that we consider the baggage demand $Q_{o,d}(k)$ as a disturbance since it is an exogenous input that is not fully measurable or predictable. We define the output vector $y(k) \in \mathbb{R}^p$ as the collection of queue, i.e., $x_o(k)$, $x_d(k)$, $x_{v^*,d}(k)$, $x_{l,z}(k)$, $x_{o,d}^{\text{bag}}(k)$. Then, the system description can be expressed in the form

$$x(k+1) = A(x(k))x(k) + Bu(k) + Gz(k); \quad y(k) = Cx(k); \quad Eu(k) = Fx(k); \quad 0 \leq y(k) \leq y_{\max}; \quad 0 \leq u(k) \leq u_{\max}, \quad (17)$$

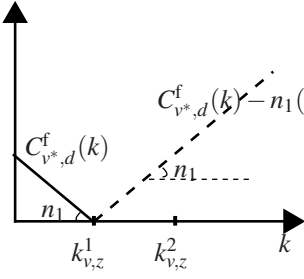


Fig. 3. Weighting functions used in the penalty terms associated with the EBS inflows (dashed line) and the EBS outflows (solid line).

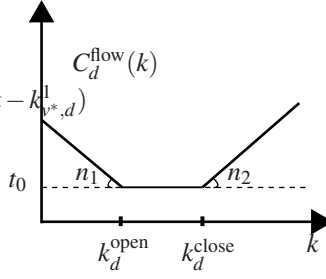


Fig. 4. Weighting function used in the penalty term associated with the inflows of the unloading stations.

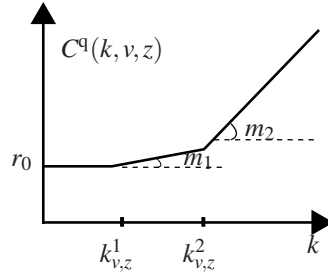


Fig. 5. The general weighting function for the queue lengths.

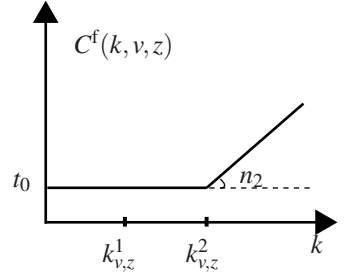


Fig. 6. The general weighting function for the flows

for properly defined matrices A , B , C , E , F , and G . Note that depending on the value of $x(k)$, certain elements of $A(x(k))$ have non-zero values. Therefore, (17) *does not* define a linear discrete-time system with linear constraints.

III. MPC PROBLEM FORMULATION

In this section, we use the dynamic model introduced in Section II within the context of MPC. At every time step, based on the current state of the system and a future prediction of baggage demands, a constrained finite horizon optimization problem will be solved yielding a sequence of optimal controls. According to the receding horizon policy, only the first step of this sequence is applied to the system, and this process is repeated at the next time step [9]–[11].

A. Objective Function

The aim of the control scheme is to assure delivery of pieces of baggage to the unloading stations within the given time windows. Imposing explicit constraints on delivery times would only be possible using a model that gives exact arrival times. However, due to its complexity, the resulting optimization problem would be intractable. Moreover, imposing explicit constraints on delivery times could lead to an infeasible optimization problem. Using the proposed flow model, the arrival time of DCVs to the unloading stations cannot be explicitly computed. Therefore, we include time-window constraints as a soft constraint in the objective function. Hence, the penalty functions penalize the flows and DCV queue lengths in such a way that the loaded DCVs are delivered to the unloading stations within their respective time windows as well as possible. Recall from Section II that $k_{v,z}^1$ and $k_{v,z}^2$ are respectively the relative opening and closing time steps of destination z . We define the following weighting functions:

$$C^q(k, v, z) = \begin{cases} r_0 & \text{if } k \leq k_{v,z}^1 \\ r_0 + m_1(k - k_{v,z}^1) & \text{if } k_{v,z}^1 < k \leq k_{v,z}^2 \\ r_0 + m_1(k_{v,z}^2 - k_{v,z}^1) + m_2(k - k_{v,z}^2) & \text{if } k > k_{v,z}^2, \end{cases} \quad C^f(k, v, z) = \begin{cases} s_0 & \text{if } k \leq k_{v,z}^2 \\ s_0 + n_2(k - k_{v,z}^2) & \text{if } k > k_{v,z}^2 \end{cases} \quad (18)$$

$$C_{v^*,d}^f(k) = \begin{cases} 0 & \text{if } k \leq k_{v^*,d}^1 \\ n_1(k - k_{v^*,d}^1) & \text{if } k > k_{v^*,d}^1 \end{cases}; \quad C_d^f(k) = \begin{cases} s_0 - n_1(k - k_d^{\text{open}}) & \text{if } k < k_d^{\text{open}} \\ s_0 & \text{if } k_d^{\text{open}} < k \leq k_d^{\text{close}} \\ s_0 + n_2(k - k_d^{\text{close}}) & \text{if } k > k_d^{\text{close}} \end{cases} \quad (19)$$

The weighting function $C^q(k, v, z)$ will be used in the penalty terms for the baggage queues in the loading stations, loaded DCV-queues in the EBS and along the links, and $C^f(k, v, z)$ will be used in the penalty terms associated with the loaded-DCV flows in the loading stations and along the links. These weighting functions are depicted in Fig. 5 and Fig. 6. The weighting function $C_{v^*,d}^{\text{flow}}(k)$, depicted in Fig. 3, will be used to penalize the loaded-DCV flows in and out of the EBS, and C_d^f , depicted in Fig. 4 will be used to penalize the empty-DCV outflows at unloading stations. In above, $r_0, m_2 > m_1, s_0, n_1$, and n_2 are strictly positive constants that determine the shape of the weighting function. Note that the relative magnitude of these constants is a design parameter that determines how much in-time delivery of DCVs is favored to energy consumption.

We will define the following penalty terms, which penalize the loaded-DCV flows and queues. For loading station o , the penalty terms at time step k associated with the baggage queues and the loaded-DCV flows are defined as:

$$J_{\text{LS}}^{\text{bag}}(k) = \sum_{o \in V_1} \sum_{d \in V_3} C^q(k, o, d) x_{o,d}^{\text{bag}}(k); \quad J_{\text{LS}}^{\text{flow}}(k) = \sum_{o \in V_1} \sum_{d \in V_3} C^f(k, o, d) q_{o,l_o^{\text{out}},d}(k). \quad (20)$$

For links $l = (v, w)$, the loaded-DCV queues and loaded-DCV flows are penalized as follows:

$$J_L^{\text{DCV}}(k) = \sum_{d \in V_3} \sum_{l=(v,w) \in L_d} C^q(k, w, d) x_{l,d}(k); \quad J_L^{\text{flow}}(k) = \sum_{d \in V_3} \sum_{l=(v,w) \in L_d} \left(C^f(k, w, d) \sum_{p \in L_w^{\text{out}}} q_{l,p,d}(k) \right), \quad (21)$$

For the EBS, penalty term associated with the loaded-DCV queues is defined as:

$$J_{\text{EBS}}^{\text{DCV}}(k) = \sum_{d \in V_3} (C^q(k, v^*, d) - r_0) x_{v^*,d}(k), \quad (22)$$

and the penalty terms associated with loaded-DCV flows in and out of the EBS are defined as:

$$J_{\text{EBS}}^{\text{inflow}}(k) = \sum_{d \in V_3} C_{v^*,d}^f(k) q_{l_{v^*,v^*,d}^{\text{in}}}(k); \quad J_{\text{EBS}}^{\text{outflow}}(k) = \sum_{d \in V_3} \left(C_{v^*,d}^f(k) - n_1(k - k_{v^*,d}^1) \right) q_{v^*,l_{v^*,d}^{\text{out}}}(k), \quad (23)$$

For the unloading stations d , the penalty term associated with the inflow of loaded DCVs to d is defined as:

$$J_{\text{US}}^{\text{flow}}(k) = \sum_{d \in V_3} C_d^f(k) q_{l_d^{\text{in}},d,d}(k). \quad (24)$$

One can observe that for $k \leq k_{o,d}^1$, as depicted in Fig. 5, we assign a constant weight r_0 to the baggage queues at the loading station and to the loaded-DCV queues along the links. With the choice of $s_0 \ll r_0$ this allows for early release of baggage, hence loaded-DCVs, into the network. These DCVs will move to the EBS since the loaded-DCV inflow of the unloading stations is highly penalized (see Fig. 4) and since the inflow of DCVs to the EBS inflicts no cost (see Fig. 3). Please note that, during this period, the DCVs will remain in the EBS since dispatching the DCVs is more expensive. For $k_{o,d}^1 < k \leq k_{o,d}^2$, the weighting functions for the baggage queues at the loading stations and for the loaded-DCV queues along the links and in the EBS increase

with the constant slope m_1 to have more loaded DCVs released in the network. The weight of the loaded-DCV inflows to the EBS increases with the constant slope n_1 to prevent DCVs from entering the EBS. Moreover the inflow of loaded DCVs to the unloading station involves no cost. Hence, the released DCVs arrive at the specified unloading stations.

For $k > k_{o,d}^2$ the weighting functions for the baggage queues at the loading stations and for the loaded-DCV queues along the links increase with the constant slope m_2 . In addition the weighting functions of the loaded-DCV flows along the links and into the unloading stations increases with the constant slope n_2 . Since the slope of the second part of the weighting functions is larger than the slope of the first part, the case of having loaded DCVs on the links, or having loaded-DCV flows arriving at the unloading stations during this time interval becomes expensive.

In order to take into account the energy consumption, we define the following penalty terms, which penalize the flow of empty DCVs. The cost of empty-DCV flows, $q_{l,p,o}(k)$ is given by $J^e(k) = t_1 \sum_{o \in V_1} \sum_{l=(v,w) \in L_o} \sum_{p \in L_w^{\text{out}}} q_{l,p,o}(k)$, where we assign a constant weight $t_1 > 0$ to empty DCV flows to avoid arbitrary empty DCV flow circulations, which is in accordance with our objective of minimizing the energy consumption. We define $J_1(k) = J_{\text{LS}}^{\text{bag}}(k) + J_{\text{LS}}^{\text{DCV}}(k) + J_{\text{L}}^{\text{DCV}}(k) + J_{\text{EBS}}^{\text{DCV}}(k)$ and $J_2(k) = J_{\text{LS}}^{\text{flow}}(k) + J_{\text{L}}^{\text{flow}}(k) + J_{\text{US}}^{\text{flow}}(k) + J_{\text{EBS}}^{\text{inflow}}(k) + J_{\text{EBS}}^{\text{outflow}}(k)$ respectively as the total cost of baggage stacks and loaded DCV queues and the total cost of loaded DCV flows. The total cost function at time step k over the prediction horizon N_p is then given by

$$J_{N_p}(k) = \frac{1}{J_{1,\text{nom}}} \sum_{j=1}^{N_p} J_1(k+j) + \frac{\alpha_1}{J_{2,\text{nom}}} \sum_{j=0}^{N_p-1} J_2(k+j) + \frac{\alpha_2}{J_{\text{nom}}^e(k)} \sum_{j=0}^{N_p-1} J^e(k)(k+j), \quad (25)$$

where $\alpha_1 > 0$, and $\alpha_2 > 0$ are constants indicating the relative importance of the respective component of the objective function, and where $J_{i,\text{nom}}$, $i = 1, 2$ is the nominal value² of $J_i(k)$.

B. Linear Programming Approach

In general, finding the optimal flow values and optimal queue lengths is a nonlinear programming problem. To arrive at a linear programming formulation we first make the following additional assumption:

A-LP The queue lengths remain constant over the prediction horizon.

Assumption A-LP enables us to arrive at a linear programming formulation of the optimization problem. Under A-LP, the first equation in (17) can be written as

$$x(k+j+1) = A(x(k))x(k+j) + Bu(k+j) + Gz(k+j), \quad (26)$$

for $j = 1, \dots, N_p - 1$, which, together with constraints of (17), defines a linear discrete time-invariant system with linear constraints. Since (25) is weighted sum of the state variables and the input variables, it can be expressed as $J_{N_p}(k) = F_0^T(k)z_{N_p}(k) + F_1^T(k)x(k) + F_2^T(k)u_{N_p}(k)$ by successive substitution in (26). Here, $z_{N_p}(k) = [z^T(k), \dots, z^T(k+N_p-1)]^T \in \mathbb{R}^{m_2 N_p}$ is the predicted disturbance vector, $u_{N_p}(k) = [u^T(k), \dots, u^T(k+N_p-1)]^T \in \mathbb{R}^{m_1 N_p}$ is the control input sequence and $F_0(k) \in \mathbb{R}^{m_2 N_p}$, $F_1(k) \in \mathbb{R}^n$, and $F_2(k) \in \mathbb{R}^{m_1 N_p}$ are coefficient vectors. Note that since the values of the weighting functions are known for $k, k+1, \dots$,

²The nominal value $J_{i,\text{nom}}$, $i = 1, 2$ can be computed by averaging over $J_i(k)$ for $k = 1, \dots, N$, where $J_i(k)$ is obtained based on simulating the system under the nominal input $u_{\text{nom}}(k)$ and the nominal baggage demand $z_{\text{nom}}(k)$, for $k = 1, \dots, N$.

$k + N_p - 1$, $F_0(k)$, $F_1(k)$, and $F_2(k)$ are fully determined at time step k . Therefore, at every time step k , we solve an optimization problem of the form

$$\min_{u_{N_p}(k)} F_2^T(k) u_{N_p}(k); \quad \text{subject to: } A^{\text{eq}}(k) u_{N_p}(k) = b^{\text{eq}}(k), \quad A^{\text{ineq}}(k) u_{N_p}(k) \leq b^{\text{ineq}}(k), \quad 0 \leq u_{N_p}(k) \leq u_{\max}, \quad (27)$$

where the matrices $A^{\text{eq}}(k)$, and $A^{\text{ineq}}(k)$ and the vectors $b^{\text{eq}}(k)$, and $b^{\text{ineq}}(k)$ are obtained based on (17), the current state of the system $x(k)$, and the predicted baggage demand $z_{N_p}(k) = [z^T(k), \dots, z^T(k + N_p - 1)]^T \in \mathbb{R}^{m_2 N_p}$. This is an LP problem, which, among others, can be solved efficiently using the simplex, active-set, or interior point methods [20].

C. Iterative Linear Programming Approach

The prediction model (26) can yield inaccurate predictions. To remedy this problem and still arrive at a tractable formulation for the optimization problem, we propose an iterative version of (49): at each iteration, a problem of form (27) is solved to find the optimal sequence of flow variables. This sequence is then used in the model (17) in forward simulation to compute the updated values for queue lengths resulting from the obtained flow values. The updated queue lengths are then used again in the LP problem formulation to find updated flow values. This process is repeated for a certain number of iterations. More formally, we replace the first equation in (17) by the prediction model

$$x_{(i)}(k + j + 1) = A(x_{(i-1)}(k + j)) x_{(i)}(k + j) + B u_{(i)}(k + j) + G z(k + j), \quad (28)$$

for $j = 0, \dots, N_p - 1$ with the initial conditions $x_{(0)}(k + j) = x(k)$ for $j = 0, \dots, N_p - 1$, $x_{(i)}(k) = x(k)$ for $i = 1, 2, \dots$, where i is the iteration index. Hence, at time step k , initializing the algorithm with the initial predicted state ³ $x_{N_p,(0)}(k) = [x^T(k), \dots, x^T(k)]^T$, the predicted state vector $x_{N_p,(i)}(k) = [x_{(i)}^T(k), \dots, x_{(i)}^T(k + N_p - 1)]^T$ at iteration i is computed based on the input sequence, $u_{N_p,(i)}(k) = [u_{(i)}^T(k), \dots, u_{(i)}^T(k + N_p - 1)]^T$ at iteration i , the predicted disturbance vector $z_{N_p}(k)$, and the predicted state vector at iteration $i - 1$. Using (28), the objective function at time step k in iteration i be expressed as $J_{N_p,i}(k) = F_{0,(i-1)}^T(k) z_{N_p}(k) + F_{1,(i-1)}^T(k) x(k) + F_{2,(i-1)}^T(k) u_{N_p,(i)}(k)$. Therefore, at time step k and in iteration i , we solve an optimization problem of the following form:

$$\min_{u_{N_p,(i)}(k)} F_{2,(i-1)}^T(k) u_{N_p,(i)}(k); \quad \text{subject to: } A_{(i-1)}^{\text{eq}}(k) u_{N_p,(i)}(k) = b_{(i-1)}^{\text{eq}}(k), \quad A_{(i-1)}^{\text{ineq}}(k) u_{N_p,(i)}(k) \leq b_{(i-1)}^{\text{ineq}}(k), \quad (29)$$

where the matrices $A_{(i-1)}^{\text{eq}}(k)$, and $A_{(i-1)}^{\text{ineq}}(k)$ and the vectors $b_{(i-1)}^{\text{eq}}(k)$, and $b_{(i-1)}^{\text{ineq}}(k)$ are obtained based on (28), and the constraints of (17) using the predicted state vector $x_{N_p,(i-1)}(k)$ at iteration $i - 1$, the current state of the system $x(k)$, and the predicted baggage demand $z_{N_p}(k)$. Note that even though we assume constant queue lengths at each inner iteration of (29), the variations in queue lengths over the prediction horizon are taken into account in the next inner iteration. Therefore, at the end of inner iterations, the computed flow values are the ones obtained having taken into account the effect of variations in the queue lengths over the prediction horizon.

³If the optimal input sequence from the previous time step is available, it can be used to compute the initial predicted state

D. Nonlinear Programming Approach

In this approach, without any simplifying assumptions, we make direct use of the model presented by (17) and the cost given by (25) in defining optimization problem of the following form:

$$\min_{u_{N_p}(k)=[u^T(k), \dots, u^T(k+N_p-1)]^T} J_{N_p}(k); \quad \text{subject to (17),} \quad (30)$$

which is a nonlinear nonconvex problem. Hence, one can use multi-start local nonlinear optimization algorithms (e.g., SQP algorithm or interior-point methods [20]) or global optimization methods (e.g., pattern search and genetic algorithms [21].)

IV. CASE STUDY

First, we introduce a state-of-the-art (SOA) method as is currently used for control of BHS. In this method, the flow of loaded DCVs at each junction is dynamically assigned to its outgoing links based on the projected deviation of delivery times from the beginning of time window of the destination. In this way, the DCVs are dynamically routed via links that make timely delivery possible while distributing the traffic in a smart way, hence avoiding congestion. The empty-DCV flows at each junction assigned based on projected travel times, and the baggage queues at the loading stations. Hence, more empty DCVs are sent via “faster” links to the loading station with longer baggage queues.

a) Routing and scheduling of loaded DCVs: at each junction of the network the flow from an incoming link l to an outgoing link p with destination d at current time $t = kT_s$ is obtained as

$$q_{l,p,d}(k) = \frac{1/|t - t_d^{\text{open}} - t_{l,p,d}(k)|}{\sum_j 1/|t - t_d^{\text{open}} - t_{l,j,d}(k)|} q_{l,\max},$$

where t_d^{open} is the opening time for the destination, $q_{l,\max}$ is the maximum flow of the link, and $t_{l,p,d}$ is the estimated travel time from the end of link l to destination d via link p . This travel time can be obtained using historical data or by computing $t_{l,p,d}(k) = t_p(k) + t_{l,p,d}^*$, where $t_{l,p,d}^*$ is the travel time from l to d via the shortest path and $t_p(k)$ is the estimated clearance time of the queues given by

$$t_p(k) = \frac{x_p(k)}{q_{l,\max}} + \frac{\sum_{j=0}^{n_p-1} F_p^{\text{in}}(k-j)(n_p-j)}{\sum_{j=0}^{n_p-1} F_p^{\text{in}}(k-j)} + \frac{T_s}{q_{l,\max}} \sum_{j=0}^{n_p-1} F_p^{\text{in}}(k-j), \quad (31)$$

where $x_p(k)$ is the total queue length at the end of link p , $F_p^{\text{in}}(k)$ is the total inflow of DCVs to link p , and $n_p T_s$ is the DCV travel time from the beginning to the end of p . The first term on the right hand side of the (31) is the clearance time of current DCV queue at the end of link p . The second term determines the average time that the past DCV flows that are currently traveling on link p need to reach the end of the link, and the last term determines the clearance time of these flows once they have reached the end of the link.

b) Routing and scheduling of empty DCVs: at each junction of the network, the flow of empty DCVs from an incoming link l to an outgoing link p is with destination o is given as

$$q_{l,p,o}(k) = \frac{x_o^{\text{bag}}}{\sum_{j \in V_1} x_j^{\text{bag}}} \frac{1/t_{l,p,o}(k)}{\sum_j 1/t_{l,j,o}(k)} q_{l,\max},$$

TABLE I. CLOSED-LOOP SIMULATION DETAILS

MPC Parameters					
N_p	$(k_{U_1}^{\text{open}}, k_{U_1}^{\text{close}}), (k_{U_2}^{\text{open}}, k_{U_2}^{\text{close}})$	$u_{\max} [\text{DCV/s}]$	$(x_{o,\max}, x_{d,\max}, x_{v^*,\max}, x_{x_l,\max}) [\text{DCV}]$	(α_1, α_2)	$(s_0, s_1, n_1, n_2), (r_0, m_1, m_2)$
8	(60, 90), (80, 120)	6	(50, 100, 120, 26)	(1, 1)	(1, 1, 2), (0.5, 1, 1, 2)
Closed-loop Simulation Parameters					
$T_s [\text{s}]$	$(x_{L_1}(0), x_{L_2}(0), x_{U_1}(0), x_{U_2}(0)) [\text{DCV}]$	N_{sim}	$v_{\text{DCV}} [\text{m/s}]$	$l_{\text{DCV}} [\text{m}]$	
0.5	(20, 49, 90, 0)	150	10	1.5	

where x_o^{bag} is the total baggage queue at loading station o , and $t_{l,p,d}$ is the estimated travel time from the end of link l to destination d that can be obtained using historical data or the aforementioned procedure via (31). For the network layout depicted in Fig. 8 with $V_1 = \{1, 9\}$, $V_3 = \{5, 13\}$, $V_2 = \{2, 3, 4, 6, 7, 10, 11, 12\}$, and $v^* = \{8\}$, we compare the performance of the SOA, LP-MPC, ILP-MPC, and NLP-MPC against each other in terms of MPC-in-the-loop optimal cost and the computational burden under the demand scenario depicted in Fig. 7 using the parameters listed in Table I. For the LP and ILP approaches, we use the CPLEX solver via TOMLAB toolbox for MATLAB, and for the NLP approach, we use the implementation of the interior-point algorithm in the MATLAB optimization toolbox. For each method, the total MPC-in-the-loop cost and the corresponding CPU time is listed in Table II for different prediction horizons using a dual core PC with Intel E8400 processor running at 3.00GHz and with 4GB of RAM. The reported CPU times are computed by averaging over the CPU times for all simulation time steps. It can be observed from Table II that while the computation time of the LP and ILP-MPC approaches are comparable to the SOA method, they outperform the SOA in terms of controller-in-the-loop cost. This can also be observed by comparing Fig. 13 with Fig. 12 which illustrate the controller-in-the-loop performance of the two methods for the baggage demand profile depicted in Fig. 7. While with the ILP-MPC with three iterations, 70% of the baggage demand for destination $U_1 = 5$ and 80% of the baggage demand for destination $U_2 = 13$ arrive within the respective time window of destination, with SOA method, only 10% of demand to $U_1 = 5$ and 20% of demand to $U_2 = 13$ arrive within the time windows. One can observe from Fig. 11 and from Table II that the NLP approach outperforms the LP and ILP approaches, but its computational burden increases very sharply for high values of prediction horizon. Moreover, the optimal cost of ILP approach converges to the NLP approach by increasing the number of iterations with far less computation effort. We also observe from Table II that the ILP approach outperforms the LP approach in terms of the closed-loop cost and that the difference between the closed-loop cost of the two increases for the increasing values of the prediction horizon. Fig. 9 compares the computation times of SOA, LP, and ILP methods for different values of N_p . It can be observed that the computation time of the LP approach is comparable with the SOA. More importantly, the computation time of LP and ILP methods increase linearly in the problem size. The computation time of ILP approach as a function of ILP iteration is depicted in Fig. 10.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have revisited the problem of dynamic routing and scheduling of DCVs in baggage handling systems. We have jointly addressed the main control problems of DCV-based baggage handling systems, namely, routing and scheduling of DCVs, line balancing, and empty cart management. Our derived model was used as the prediction model within the MPC framework. The objective function was defined so as to penalize the deviation of baggage delivery time at the unloading stations from pre-specified time windows, and the energy consumption. We formulated the MPC problem as a nonlinear programming

TABLE II. COMPARISON OF CLOSED-LOOP PERFORMANCE AND COMPUTATION TIMES

N_p	Opt. Scheme (Solver)	Iter./Multi-start Iter.	min. CPU Time [s]	Avg. CPU Time [s]	max. CPU Time	Optimal Cost
N/A	SOA	N/A	0.0015	0.0036	0.0411	16671.5
2	LP (CPLEX)	N/A	0.0056	0.0074	0.0411	8083.1
2	ILP (CPLEX)	2	0.0128	0.0237	0.7401	8083.1
2	ILP (CPLEX)	4	0.0249	0.0389	0.7416	8083.1
2	ILP (CPLEX)	6	0.0475	0.0530	0.0739	8083.1
2	NLP (interior-point)	3	99.6190	114.53	135.52	8083.1
4	LP (CPLEX)	N/A	0.0074	0.0094	0.0239	7982.1
4	ILP (CPLEX)	2	0.0150	0.0271	0.7664	7869.5
4	ILP (CPLEX)	4	0.0309	0.0445	0.7845	7820.5
4	ILP (CPLEX)	6	0.0453	0.0514	0.0913	7800.5
4	NLP (interior-point)	4	310.00	344.93	376.16	6813.4
6	LP (CPLEX)	N/A	0.0087	0.0197	0.6674	7471.3
6	ILP (CPLEX)	4	0.0375	0.0457	0.0967	5890.2
6	ILP (CPLEX)	8	0.0721	0.0852	0.1133	5574.2
6	NLP (interior-point)	8	419.62	450.15	486.42	4546.7
8	LP (CPLEX)	N/A	0.1212	0.1902	0.3299	5925.3
8	ILP (CPLEX)	6	0.8061	1.0185	1.4325	4949.9

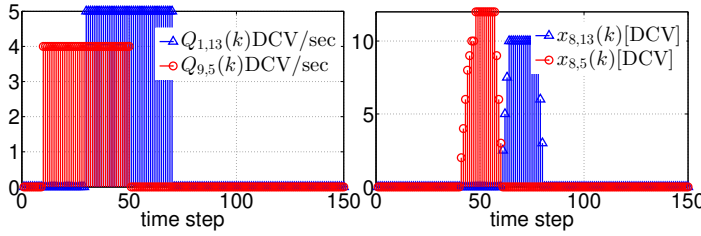


Fig. 7. The baggage demand (left) and number of DCVs in the EBS (right).

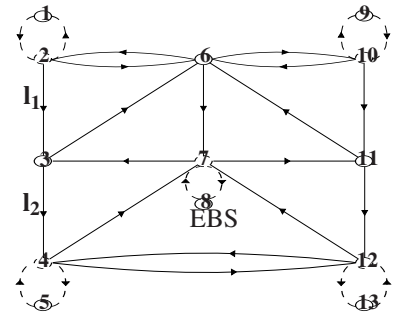
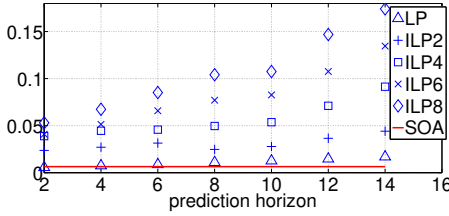
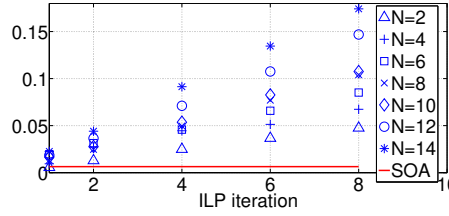
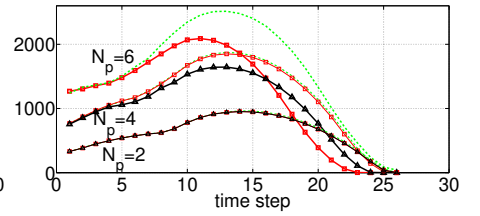
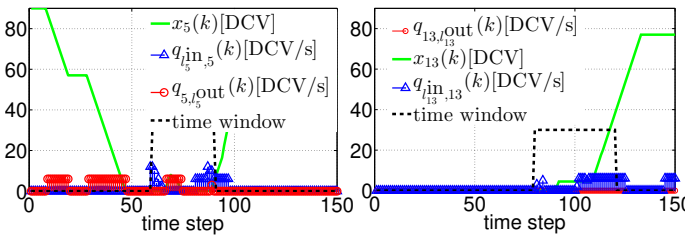
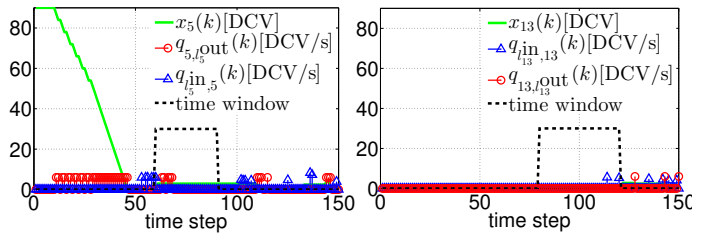
Fig. 8. Schematic layout of the case study network with loading stations $L_1 = 1$, and $L_2 = 9$ and unloading stations $U_1 = 5$, and $U_2 = 13$.Fig. 9. Computation time of LP and ILP in seconds as a function of N_p 

Fig. 10. Computation time of the ILP in seconds as a function of ILP iterations

Fig. 11. Predicted cost as a function of closed-loop time step using the LP approach (dashed line), the ILP approach (square markers), and the NLP approach (triangle markers) for different prediction horizons N_p .Fig. 12. ILP-MPC-based control. Triangle: total DCV inflow. Square: DCV outflow. Solid line: DCV queue for unloading stations (left) U_1 and (right) U_2 with the associated time window of destination.Fig. 13. SOA control. Triangle: total DCV inflow. Square: DCV outflow. Solid line: DCV queue for unloading stations (left) U_1 and (right) U_2 with the associated time window of destination.

(NLP) problem, which, for the case study, was computationally very inefficient for large prediction horizons (e.g., $N_p > 6$) due to the large-scale nature of the problem. Under some simplifying assumptions, we showed that the underlying optimization problem can be recast as a linear programming (LP) problem. However, the performance of the LP approach is suboptimal with respect to the NLP approach with significant performance loss for larger values of prediction horizons. To achieve a fast yet close-to-optimal solution, we have proposed an iterative linear programming (ILP) scheme that solves a sequence of LP problems to compute the optimal control sequence. We have illustrated that by using this method for an appropriate number of ILP iterations, it is possible to achieve a performance close to the NLP approach with significantly less computational burden. We also compared the performance of the our proposed approach to a state-of-the-art method (SOA) as is used in practice. We illustrated via simulations that our proposed method outperforms the SOA. The scalability of our approach was illustrated by numerical tests. Particularly, we showed that the required computation time increases linearly for increasing values of the prediction horizon and number of iterations.

For future work, we will compare the performance and computational requirements of our approach to an approach based on computing the exact travel time of each DCV in the network using a given distribution of passengers and baggage allowance for a predefined flight schedule. In addition, the trade-off between performance and computational efficiency of the ILP approach, especially for large-scale systems, will be analyzed in the future.

REFERENCES

- [1] Vanderlande Industries, “DCV SYSTEMS: Combining Speed, Security and Reliability,” Brochure, 2013. [Online]. Available: <http://www.vanderlande.com/en/Baggage-Handling.htm>
- [2] R. De Neufville, “The baggage system at Denver: Prospects and lessons,” *Journal of Air Transport Management*, vol. 1, pp. 229–236, 1994.
- [3] M. Johnstone, D. Creighton, and S. Nahavandi, “Status-based routing in baggage handling systems: Searching verses learning,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40, no. 2, pp. 189–200, 2010.
- [4] A. Tarău, B. De Schutter, and H. Hellendoorn, “Model-based control for route choice in automated baggage handling systems,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40, no. 3, pp. 341–351, May 2010.
- [5] L. Tang and X. Wang, “An improved particle swarm optimization algorithm for the hybrid flowshop scheduling to minimize total weighted completion time in process industry,” *IEEE Transactions on Control Systems Technology*, vol. 18, no. 6, pp. 1303–1314, 2010.
- [6] —, “A scatter search algorithm for a multistage production scheduling problem with blocking and semi-continuous batching machine,” *IEEE Transactions on Control Systems Technology*, vol. 19, no. 5, pp. 976–989, 2011.
- [7] A. Alessandri, C. Cervellera, and M. Gaggero, “Predictive control of container flows in maritime intermodal terminals,” *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1423–1431, 2013.
- [8] N. Lechevin, C. A. Rabbath, and M. Lauzon, “A decision policy for the routing and munitions management of multiplatforms of unmanned combat vehicles in adversarial urban environments,” *IEEE Transactions on Control Systems Technology*, vol. 17, no. 3, pp. 505–519, 2009.
- [9] J. Maciejowski, *Predictive Control with Constraints*. Harlow, England: Prentice Hall, 2002.
- [10] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control Theory and Design*. Nob Hill Publishing., 2009.
- [11] C. E. Garcia, D. M. Prett, and M. Morari, “Model predictive control: Theory and practice—A survey,” *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.
- [12] E. Perea-Lopez, B. Ydstie, and I. E. Grossmann, “A model predictive control strategy for supply chain optimization,” *Computers and Chemical Engineering*, vol. 27, no. 8-9, pp. 1201–1218, 2003.

- [13] W. Wang and D. Rivera, "Model predictive control for tactical decision-making in semiconductor manufacturing supply chain management," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 5, pp. 841–855, Sept 2008.
- [14] A. Alessandri, M. Gaggero, and F. Tonelli, "Min-max and predictive control for the management of distribution in supply chains," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 5, pp. 1075–1089, Sept 2011.
- [15] G. Black and V. Vyatkin, "Intelligent component-based automation of baggage handling systems with IEC 61499," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 2, pp. 337–351, 2010.
- [16] V. Vyatkin, "IEC 61499 as enabler of distributed and intelligent automation: State-of-the-art review," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 768–781, 2011.
- [17] A. Tarău, B. De Schutter, and H. Hellendoorn, "Hierarchical route choice control for baggage handling systems," in *Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems*, St. Louis, Missouri, USA, Oct. 2009, pp. 679–684.
- [18] Y. Zeinaly, B. De Schutter, and J. Hellendoorn, "An MPC scheme for routing problem in baggage handling systems: A linear programming approach," in *Proceedings of the 10th IEEE International Conference on Networking, Sensing and Control*, Paris, France, Apr. 2013.
- [19] A. Schrijver, *Theory of Linear and Integer Programming*. New York, NY, USA: John Wiley & Sons, Inc., 1986.
- [20] A. Antoniou and W. S. Lu, *Practical Optimization: Algorithms and Engineering Applications*. Berlin, Germany: Springer, 2007.
- [21] T. Weise, *Global Optimization Algorithms - Theory and Application*, 2nd ed. Self-Published, Jun. 26, 2009.