Delft University of Technology

Delft Center for Systems and Control

Technical report 09-011

Route choice control of automated baggage handling systems*

A.N. Tarău, B. De Schutter, and J. Hellendoorn

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.dcsc.tudelft.nl

Route choice control of automated baggage handling systems

Submission date: October 21, 2008

Word count: 6207 words + (4 figures)*(250 words) + (1 table)*(250 words)= 7457 words

Authors:

A.N. Tarău, Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands, phone: +31-15-2782087, fax: +31-15-2786679, email: a.n.tarau@tudelft.nl

B. De Schutter, Delft Center for Systems and Control & Marine and Transport Technology, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands, phone: +31-15-2785113, fax: +31-15-2786679, email: b@deschutter.info

J. Hellendoorn, Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands, phone: +31-15-2789007, fax: +31-15-2786679, email: j.hellendoorn@tudelft.nl

Abstract
State-of-the-art baggage handling systems transport luggage in an automated way using destination coded vehicles (DCVs). These vehicles transport the bags at high speeds on a "mini" railway network. Currently, the networks are simple, with only a few junctions, since otherwise bottlenecks would be created at the junctions. This makes the system inefficient. In the research we conduct, more complex networks are considered. In order to optimize the performance of the system we develop and compare centralized and decentralized control methods that can be used to route the DCVs through the track network. The proposed centralized control method is model predictive control (MPC). Due to the large computation effort centralized MPC requires, decentralized MPC and a fast decentralized heuristic approach are also proposed. When implementing the decentralized approaches, each junction has its own local controller for positioning the switch going into the junction and the switch going out of the junction. In order to assess the advantages and disadvantages of centralized MPC, decentralized MPC, and the decentralized heuristic approach, we also discuss a simple benchmark case study. The considered control methods are compared for several scenarios. Results indicate that centralized MPC becomes intractable when a large stream of bags has to be handled, while decentralized MPC can still be used to suboptimally solve the problem. Moreover, the decentralized heuristic approach usually gives worse results than those obtained when using decentralized MPC, but with very low computation time.

1   INTRODUCTION

The baggage handling system of an airport is one of the most important factors that determines the airport's efficiency. The first objective of a baggage handling system is to transport all the checked-in or transfer bags to the corresponding end points before the planes have to be loaded. However, due to the airport's logistics, an end point is allocated to a plane only with a given amount of time before the departure of the plane. Hence, the baggage handling system performs optimally if each of the bags to be handled arrives at its given end point in specific time windows. In order to transport the bags in an automated way, a baggage handling system could incorporate technology such as scanners that scan the labels on each piece of luggage, baggage screening equipment for security scanning, networks of conveyors equipped with junctions that route the bags through the system, and destination coded vehicles (DCVs). A DCV is a metal cart with a plastic tub on top. These carts are mounted on tracks and propelled by linear induction motors. They transport the bags at high speed on a "mini" railway network (*1*).

Briefly, the main control problems of a baggage handling system are coordination and synchronization of the processing units (when loading the bags onto the system in order to avoid damaging the bags and blocking the system, or when unloading them to the corresponding end point), route choice control of each bag (and implicitly the switch control of each junction), velocity control of each DCV, line balancing (route assignment of each empty DCV such that all the loading stations have enough empty DCVs at any time instant), and prevention of buffer overflows (*2*). We assume that the correct loading and unloading of DCVs is ensured by low-level controllers. Also, there is a sufficient number of DCVs present in the system so that when a bag is at the loading station there is a DCV ready to transport it. Finally, we consider that the capacity of the end points is large enough that no buffer overflow can occur. So, in this paper we focus on the route choice and velocity control of DCVs.

The route assignment problem has been addressed in e.g. (*3, 4*). But, in our case we do not deal with a shortest-path or shortest-time problem, since we need the bags at their corresponding end point within a given time window. An attempt to solve the routing problem of DCVs transporting bags using an analogy of how data are transmitted via internet is presented in (*5*), but without presenting any experimental results. Also, in (*6*), a multi-agent approach for the control software of a DCV-based baggage handling system is presented. However, this multi-agent system deals with major challenges due to the extensive communication required. Therefore, the goal of our work is to develop and compare control approaches for route choice control of each DCV transporting a bag through the track network.

The paper is organized as follows. In Section 2, a continuous-time event-driven model of the system is presented. Afterwards, in Section 3, the global performance index is elaborated. In Section 4, we propose several control methods for computing the route of each DCV transporting a bag. First we propose a centralized approach, namely model predictive control (MPC). Due to the large computation effort that the centralized MPC requires, we propose also decentralized approaches as decentralized MPC and fast heuristics. The heuristic approach uses rules to determine the position of the switches leading into and out of a junction. These rules depend on the static and dynamic priorities of the bags transported by DCVs on the incoming links, on the optimal path to destination, and on the congestion of the DCVs on the outgoing links. The analysis of the simulation results and the comparison of the proposed control methods are elaborated in Section 5. Finally, in Section 6, conclusions are drawn and directions of future research are presented.

2   EVENT-DRIVEN MODEL

In this section we briefly recapitulate the event-driven model of a baggage handling system that we have developed in (*7*). To illustrate our approaches we consider a DCV-based baggage handling system as sketched in Figure 1. This system operates as follows: given a dynamic demand of bags (identified by their unique code) and a buffer of empty DCVs for each loading station, together with the network of tracks, the route of

each DCV (from a given loading station to the corresponding unloading station) has to be computed subject to operational and safety constraints such that the performance of the system is optimized.

The model of the baggage handling system we have developed in (7) consists of a continuous part describing the movement of the individual vehicles transporting the bags through the network, and of the following discrete events: loading a new bag onto a DCV, unloading a bag that arrives at its end point, updating the position of the switch going into a junction (called switch-in here after) and the position of a switch going out of a junction (called switch-out here after), and updating the speed of a DCV. According to the discrete-event model in (7), as long as there are bags to be handled, given the current state of the system (the positions of the DCVs in the network and the positions of the switch-in and switch-out of the junctions), we shift the current time to the next event time, take the appropriate action, and update the state of the system.

The operational constraints derived from the mechanical and design limitations of the system are the following:

- C_1: the speed of each DCV is bounded between 0 and v_{max}.

- C_2: a bag can be loaded onto a DCV only if there is an empty DCV under the loading station. This means that if there is a congestion at a loading station, then no loading event can occur at that loading station.

- C_3: the position of a switch at a junction can change after minimum τ_x time units in order to avoid the quick and repeated back and forth movements of the switch which may lead to mechanical damage.

3   GLOBAL PERFORMANCE INDEX

In order to compare later on the proposed control methods, in this section we define the global performance index J to be used.

Since the baggage handling system performs successfully if all the bags are transported to the corresponding end point before a given time instant, from a central point of view, the primary objective is the minimization of the overdue time. A secondary objective is the minimization of the additional storage time at the end point. This objective is required due to the intense utilization of the end points in a busy airport. Hence, one way to construct the objective function $J_{\text{pen},i}$ corresponding to the ith bag that entered the network, $i \in \{1, 2, \ldots, X\}$, is to penalize the overdue time and the additional storage time. Note that the objective function $J_{\text{pen},i}$ depends on the arrival time of DCV_i at its corresponding end point, and implicitly $J_{\text{pen},i}$ depends on the routes of all the X bags to be handled. So, we have:

$$J_{\text{pen},i}(t) = \sigma_i \max(0, t - t_{\text{load\_plane},i}) + \lambda_1 \max(0, t_{\text{load\_plane},i} - \theta_{\text{max\_storage},i} - t) \tag{1}$$

where $t_{\text{load\_plane},i}$ is the time instant when the end point closes and the bags are loaded into the plane, σ_i is the static priority of the bag on DCV_i (the flight priority), and $\theta_{\text{max\_storage},i}$ is the maximum possible time window for which the end point corresponding to DCV_i is open for that specific flight. The weighting parameter $\lambda_1 \leq 1$ represents the relative cost between buying additional storage space at the end points and the cost of customers that have their baggage delayed.

Note that the above performance function has some flat parts, which yields difficulty for many optimization algorithms. To get some additional gradient we could also include the dwell time, resulting in:

$$J_i(t) = J_{\text{pen},i}(t) + \lambda_2(t - t_{\text{entry},i}) \tag{2}$$

where $t_{\text{entry},i}$ denotes the time instant when the ith bag enters the baggage handling system (i.e. arrives at one of the loading stations) and λ_2 is a small weight factor ($\lambda_2 \ll \lambda_1$).

The final performance index is given by $J_{\text{tot}} = \sum_{i=1}^{X} J_i(t_{\text{arrival},i})$, where $t_{\text{arrival},i}$ is the time instant when DCV_i is unloaded.

4 CONTROL APPROACHES

In this section we determine the route of each DCV transporting a bag. We assume that the velocity of each DCV is always at its maximum, v_{max}, unless overruled by the local on-board collision avoidance controller. These collision avoidance controllers ensure a minimum safe distance between DCVs and also hold DCVs at switching points, if required.

4.1 Centralized model predictive control

Model predictive control is an on-line control design method (8) that uses the receding horizon principle. In the basic MPC approach, given a horizon N, at step k, the future control sequence $u(k+1), u(k+2), \ldots, u(k+N)$ is computed by solving a discrete-time optimization problem over a period $[t_k, t_k + T_s N]$, where $t_k = t_0 + kT_s$ with T_s the sampling time, so that a cost criterion is optimized subject to the operational constraints. MPC uses a receding horizon approach. So, after computing the optimal control sequence, only the first control sample is implemented, and subsequently the horizon is shifted. Next, the new state of the system is measured or estimated, and a new optimization problem at time step t_{k+1} is solved using this new information. In this way, also a feedback mechanism is introduced.

We define now a variant of MPC, where k is not a time index, but a bag index. In this context the horizon N corresponds to the number of bags that we let enter the track network after bag step k. Also, computing the control $u(k+j)$, for $j = 1, 2, \ldots, N$ consists in determining the route of DCV_{k+j}. Assume that there is a fixed number N_{routes} of possible routes from a loading station to an unloading station. The N_{routes} routes are named $1, 2 \ldots, N_{\text{routes}}$. Let $r(i) \in \{1, 2, \ldots, N_{\text{routes}}\}$ denote the route of DCV_i. Note that, at bag step k, the route is selected once for each DCV without being adjusted after the decision has been made. Now let $\mathbf{r}(k)$ denote the future route sequence for the next N bags entering the network after bag step k, $\mathbf{r}(k) = [r(k+1)\, r(k+2)\, \ldots\, r(k+N)]^{\text{T}}$.

The total performance function of the centralized MPC is defined as $J_{\text{tot CMPC}, k+N}(r(1), r(2), \ldots, r(k+N)) = \sum_{i=1}^{k+N} J_i(\hat{t}_{\text{arrival},i})$ where $\hat{t}_{\text{arrival},i}$ is the estimated arrival time of DCV_i depending on the routes of the first $k+N$ bags that entered the network. Accordingly, the MPC optimization problem at bag step k is defined as follows:

$$\text{P}_1: \quad \min_{\mathbf{r}(k)} J_{\text{tot CMPC}, k+N}(r(1), r(2), \ldots, r(k+N))$$
$$\text{subject to}$$
$$\text{the system dynamics}$$
$$\text{operational constraints}$$

Centralized MPC can compute on-line the route of each DCV in the network, but it requires large computation effort as will be illustrated in Section 5. Therefore, we will also propose some decentralized control approaches that offer a trade-off between the optimality of the system's performance and the time required to compute the solution.

Optimization methods At each bag step k, the future route sequence $r(k+1), r(k+2), \ldots, r(k+N)$ is computed by solving P_1 over a horizon of N bags so that $J_{\text{tot CMPC},N}(r(1), r(2), \ldots, r(k+N))$ is minimized subject to the model of the system and the operational constraints. So, the control has integer representation. Therefore, to solve the optimization problems P_1 one could use e.g. *genetic* algorithms, *simulated annealing*, or *tabu search* (9), (10), (11).

4.2   Decentralized model predictive control

In decentralized model predictive route choice control we consider local systems, each consisting of a junction S_s with $s \in \{1, 2, \ldots, S\}$, its incoming and its outgoing links (note that without loss of generality we assume that each switch into a junction has maximum 2 incoming links and maximum 2 outgoing links, both indexed by $l \in \{0, 1\}$). For the sake of simplicity of notation, in the remainder of this subsection, we will not explicitly indicate the subscript s for variables that refer to junction S_s.

Our prediction model is a simulation of the local system. In the prediction model we index the bags that successively cross S_s during the entire simulation period $[t_0, T_{\max}]$ as $b_1, b_2, \ldots, b_{N_{\text{bags}}}$, where N_{bags} is the number of bags that cross S_s during $[t_0, T_{\max}]$. The optimization is performed at every junction S_s over the next $N \le N_{\text{bags}}$ bags that pass the junction.

Every time a bag has crossed the junction we update the local control. So, assume that bag b_k has crossed S_s. The time instant at which this happens is denoted by t_{crt}. For the sake of simplicity of notation, we will not explicitly include the time argument when specifying the control laws and related variables since they always refer to the current time t_{crt}. Next we compute the control sequence $\mathbf{u}(k) = [u_{\text{sw\_in}}(k+1) \ldots u_{\text{sw\_in}}(k+N) \, u_{\text{sw\_out}}(k+1) \ldots u_{\text{sw\_out}}(k+N)]^{\mathrm{T}}$ corresponding to the next N bags $b_{k+1}, b_{k+2}, \ldots, b_{k+N}$ that will cross the junction by solving an optimization problem. The control decisions $u_{\text{sw\_in}}(k+j)$ for $j = 1, 2, \ldots, N$ of the switch into S_s determine the order in which the bags cross the junction and the corresponding time when bags b_{k+j} enter S_s. The control decisions $u_{\text{sw\_out}}(k+j)$ for $j = 1, 2, \ldots, N$ of the switch out of S_s influence the route that bag b_{k+j} will take.

When solving the optimization problem, we will use a local performance index $J_{\text{DMPC},N}$. The local performance index is computed via a simulation of the local system for the next N bags that will cross the junction. This goes as follows. At bag step $k \ge 0$, the initial state of the local system consists of the positions of the DCVs traveling in the local system and the positions of the switch-in and switch-out of S_s at the time instant when bag k crossed the junction (if $k = 0$ we consider the initial state of the local system at time instant t_0). Next, at bag step k, we compute the release rate of each outgoing link l for $l = 0, 1$. Let n_l denote the number of DCVs that left the outgoing link l within the time window $[t_{\text{crt}} - \tau_{\text{q}}, t_{\text{crt}}]$, of length τ_{q} time units. Then the fixed release rate of link l for the prediction model at bag step k is given by $\zeta_l = \dfrac{n_l}{\tau_{\text{q}}}$. However, for links that connect S_s with unloading stations, the release rate is by definition unbounded. The control of the switch-out $u_{\text{sw\_out}}(k+j)$ with $j \in \{1, 2, \ldots, N\}$ represents the position of switch-out when bag b_{k+j} will cross S_s, determining the next junction towards which bag b_{k+j} will travel. Let us call this junction $S_{\text{next},k+j}$ (note that in fact $S_{\text{next},k+j}$ is a function of $u_{\text{sw\_out}}(k+j)$). For each possible route $r \in \mathscr{R}_{\text{next},k+j}$, where $\mathscr{R}_{\text{next},k+j}$ is the set of routes from $S_{\text{next},k+j}$ to the corresponding end point of bag b_{k+j}, we estimate the time that the DCV transporting bag b_{k+j} needs to reach its end point via route r as follows:

$$\hat{t}_{\text{arrival},k+j,r} = t_{\text{leave local system},k+j} + \tau_{\text{approx},r}$$

where $t_{\text{leave local system},k+j}$ is the time instant (predicted by the local simulation model) at which bag b_{k+j} leaves the link $S_s \to S_{\text{next},k+j}$ for the release rate $\zeta_{u_{\text{sw\_out}}(k+j)}$, and $\tau_{\text{approx},r}$ is the static time period that the DCV carrying bag b_{k+j} would need to travel the route $r \in \mathscr{R}_{k+j,\text{next}}$ with $v_{\max}$. Then the local objective function $J_{\text{DMPC},N}(\mathbf{u}(k))$ is defined as

$$J_{\text{DMPC},N}(\mathbf{u}(k)) = \sum_{j=1}^{N} J_{k+j}(\hat{t}^*_{\text{arrival},k+j})$$

where $\hat{t}^*_{\text{arrival},k+j} = \underset{\{\hat{t}_{\text{arrival},k+j,r} \mid r \in \mathscr{R}_{\text{next},k+j}\}}{\arg\min} J_{k+j}(\hat{t}_{\text{arrival},k+j,r})$.

So, in decentralized route choice MPC, at step k, where k is the number of DCVs that passed junction S_s, the future control sequence of the switches $\mathbf{u}(k)$ is computed by solving an optimization problem

over a horizon of N bags so that the local performance function $J_{\text{DMPC},N}(\mathbf{u}(k))$ is optimized subject to the operational constraints. Accordingly, the MPC optimization problem at junction S_s and bag step k is defined as follows:

$$P_2: \quad \min_{\mathbf{u}(k)} J_{\text{DMPC},N}(\mathbf{u}(k))$$
$$\text{subject to}$$
$$\text{the local dynamics of the junction } S_s \text{ with its incoming and outgoing links}$$
$$\text{operational constraints}$$

After computing the future control only the first switch-in and switch-out control is applied. Given the state of the system after applying the MPC control, a new optimization will be solved over the considered horizon.

The main advantage of decentralized MPC consists in a smaller computation time than the one needed when using centralized control due to the fact that we now compute in parallel the solution of a smaller and simplified optimization problem.

Optimization methods   Recall that the control variables $u_{\text{sw\_in}}(k+j)$, $u_{\text{sw\_out}}(k+j)$ of the switch at junction S_s, represent the positions 0 or 1 that the switch-in and switch-out of S_s should have when the DCV carrying bag b_{k+j} will pass the junction. Hence, the control variable has an integer representation. In order to solve the optimization problem P_2 one can use integer optimization once more (see Section 4.1).

4.3   Heuristic approach

In order to lower the computation time for solving the switch control problem even more, in this section the route of each DCV is controlled using a fast heuristic approach. In this approach each toggle of the switch S_s with $s \in \{1, 2, \ldots, S\}$ is locally controlled based on heuristic rules as presented next. The control variables represent the time period after which the switch-in and respectively switch-out will change position.

We consider that the switch control is performed only based on local information regarding the flow of DCVs on the incoming and outgoing links. For the sake of simplicity of notation, we will not explicitly include the s subscript in the remainder of this subsection since we only describe the control of the switch-in and switch-out for one junction only. Also, the time argument will be omitted when denoting the control laws and related variables since they always refer to the current time t_{crt}.

4.3.1   Local heuristic control of the switch-in

For a junction S_s we define the following variables:

- Γ_l is the set of bags transported by DCVs that are traveling on the incoming link $l \in \{0, 1\}$ of junction S_s at current time t_{crt} (we do not consider any prediction or future time window),

- $a_l = \displaystyle\sum_{i \in \Gamma_l} \sigma_i,$

- $b_l = \displaystyle\sum_{i \in \Gamma_l} \frac{\hat{\delta}_i}{\delta_{\max,i}}$ with $\hat{\delta}_i$ the estimate of the actual time bag i requires to get from its current position to its destination in case of no congestion and average speed, and $\delta_{\max,i}$ is the maximum time left to bag i to spend in the system while still arriving at the plane on time. If bag i misses the flight, then the bag has to wait for a new plane with the same destination, and hence, a new departure time is assigned to bag i, and consequently a new loading time $t_{\text{new\_load\_plane},i}$ for bag i is considered. Then the variable

$\delta_{\text{max},i}$ is defined as

$$\delta_{\text{max},i} = \begin{cases} t_{\text{load\_plane},i} - t_{\text{crt}} & \text{if } t_{\text{load\_plane},i} > t_{\text{crt}} \\ t_{\text{new\_load\_plane},i} - t_{\text{crt}} & \text{if } t_{\text{load\_plane},i} \leq t_{\text{crt}}. \end{cases}$$

In order to determine the next position of the switch-in at junction S_s, every time a new bag enters the incoming link $l \in \{0, 1\}$ we compute the performance measure $p_{l,\text{sw\_in}}$ for $l = 0, 1$. This performance measure takes into account the static and dynamic priorities of the bags transported by DCVs on the link l, and the current position of the switch-in at junction S_s (due to the operational constraint C_3: in order to avoid chattering, the position of a switch at a junction can only change after minimum τ_x time units):

$$p_{0,\text{sw\_in}} = w_{\text{st\_priority}}a_0 + w_{\text{dyn\_priority}}b_0 - w_{\text{sw\_in}}\tau_x I_{\text{crt}}$$
$$p_{1,\text{sw\_in}} = w_{\text{st\_priority}}a_1 + w_{\text{dyn\_priority}}b_1 - w_{\text{sw\_in}}\tau_x(1 - I_{\text{crt}})$$

where I_{crt} denotes the current position of the switch-in at junction S_s (i.e. $I_{\text{crt}} = 0$ if the switch-in is positioned on the incoming link 0 and $I_{\text{crt}} = 1$ if the switch-in is positioned on the incoming link 1). The weighting parameters $w_{\text{st\_priority}}$, $w_{\text{dyn\_priority}}$, and $w_{\text{sw\_in}}$ can be calibrated as explained in Section 4.3.3.

Let $z_l \in \Gamma_l$ denote the index of the bag closest to S_s on incoming link l. The variable d_{z_l} denotes the distance between the current position of bag z_l and S_s and v_{z_l} denotes the current speed of DCV transporting bag z_l. Then we define $\tau_{\text{arrival\_at\_S}_s,l}$ as $\tau_{\text{arrival\_at\_S}_s,l} = \dfrac{d_{z_l}}{v_{z_l}}$ if $d_{z_l} > 0$, and $\tau_{\text{arrival\_at\_S}_s,l} = 0$ if $d_{z_l} = 0$.

The position of the switch-in at S_s is toggled only if $p_{0,\text{sw\_in}} > p_{1,\text{sw\_in}}$ and $I_{\text{crt}} = 1$ or if $p_{1,\text{sw\_in}} > p_{0,\text{sw\_in}}$ and $I_{\text{crt}} = 0$. If this is the case, then the current position of the switch-in is changed after $\tau_{\text{sw\_in},s} = \max(\tau_x - \tau_{\text{sw\_in\_prev},s}, \tau_{\text{arrival\_at\_S}_s,1-I_{\text{crt}}})$ time units where $\tau_{\text{sw\_in\_prev},s}$ is the time for which the switch-in at S_s has been in its current position.

4.3.2   Local heuristic control of the switch-out

Every time when a bag is at junction S_s we compute the control variable $\tau_{\text{sw\_out}}$ which represents the time period until the position of the switch-out has to be changed.

Assume that bag i is now at junction S_s. The time instant when this happens is denoted by t_{crt}. First we estimate the time that bag i needs to reach its end point taking into account the congestion of the outgoing links of S_s. Let us call $S_{\text{neighbor},l}$ the junction connected via the outgoing link l to S_s. If there is congestion on the outgoing link $l \in \{0, 1\}$ then there is a queue of DCVs on $d_{\text{cong},l,i} \leq d_{\text{link},l}$ distance units before $S_{\text{neighbor},l}$, as sketched in Figure 2, where $d_{\text{link},l}$ is the length of the outgoing link l at S_s.

Let $\ell_{\text{queue},l,i}$ denote the number of DCVs waiting in the queue on the outgoing link l at the current time instant. Then for each possible route $r \in \mathscr{R}_{\text{neighbor},l}$, where $\mathscr{R}_{\text{neighbor},l}$ is the set of routes from $S_{\text{neighbor},l}$ to $S_{\text{dest},i}$, the end point corresponding to bag i, we estimate the time that the DCV transporting bag i needs to reach its end point via the optimal route r as follows:

$$\hat{t}_{l,i,r} = t_{\text{crt}} + \frac{d_{\text{link},l} - d_{\text{cong},l,i}}{v_{\text{max}}} + \frac{\ell_{\text{queue},l,i} + 1}{\zeta_l} + \frac{d_{\text{route},l}}{v_{\text{max}}}$$

where $d_{\text{cong},l,i}$ is the length of the congested part of the link l before $S_{\text{neighbor},l}$, and $d_{\text{route},l}$ is the length of route r.

The optimal time $\hat{t}_{l,i}^*$ is defined as follows:

$$\hat{t}_{l,i}^* = \underset{\{\hat{t}_{l,i,r} \mid r \in \mathscr{R}_{\text{neighbor},l}\}}{\arg\min} \, J_i(\hat{t}_{l,i,r}).$$

Then we compute the cost criterion $c_{l,i,\text{sw\_out}}$ for $l = 0, 1$ that takes into account $\hat{t}^*_{l,i}$ and the current position O_{crt} of the outgoing switch (due to the operational constraint C_3):

$$c_{0,i,\text{sw\_out}} = w_{\text{pen}} J_i(\hat{t}^*_{0,i}) + w_{\text{sw\_out}} \tau_{\text{x}} O_{\text{crt}}$$
$$c_{1,i,\text{sw\_out}} = w_{\text{pen}} J_i(\hat{t}^*_{1,i}) + w_{\text{sw\_out}} \tau_{\text{x}} (1 - O_{\text{crt}})$$

The weighting parameters w_{pen} and $w_{\text{sw\_out}}$ can be calibrated as explained in Section 4.3.3.

The position of the switch-in at junction S_s is toggled only if $c_{0,i,\text{sw\_out}} < c_{1,i,\text{sw\_out}}$ and $O_{\text{crt}} = 1$ or if $c_{1,i,\text{sw\_out}} < c_{0,i,\text{sw\_out}}$ and $O_{\text{crt}} = 0$. If this is the case, then the switch-out is toggled after $\tau_{\text{sw\_out}} = \max(0, \tau_{\text{x}} - \tau_{\text{sw\_out\_prev}})$ where $\tau_{\text{sw\_out\_prev},s}$ is the time for which the switch-out at S_s has been in its current position.

4.3.3   Calibration

The calibration of the weighting parameters presented in the previous section will be done by solving the following optimization problem:

$$\text{P}_3: \quad \min_{\mathbf{w}} \sum_{j=1}^{N_{\text{scenario}}} J_{j,\text{tot}}(\mathbf{w})$$
$$\text{subject to}$$
$$\text{the system dynamics depending on } \mathbf{w}$$
$$\text{operational constraints}$$

where $\mathbf{w} = [w_{\text{st\_priority}} \; w_{\text{dyn\_priority}} \; w_{\text{pen}} \; w_{\text{sw\_in}} \; w_{\text{sw\_out}}]^{\text{T}}$, N_{scenario} is the number of scenarios over which the calibration is performed, and $J_{j,\text{tot}}$ is the total performance index corresponding to scenario j, with $j \in \{1, 2, \dots, N_{\text{scenario}}\}$.

The problem P_3 is a nonlinear, nonconvex optimization problem with continuous variables. So, in order to solve P_3, we have to use global optimization algorithms as multistart *sequential quadratic programming*, *pattern search*, see e.g. (*12*), (*13*), multistart *simulated annealing* algorithms, or *genetic* algorithms.

5   CASE STUDY

In this section we compare the proposed control methods based on simulation examples.

5.1   Set-up

We consider the network of tracks depicted in Figure 3 with 6 loading stations, 1 unloading station, and 10 junctions. We have considered this network because on the one hand it is simple, allowing an intuitive understanding of and insight in the operation of the system and the results of the control, and because on the other hand, it also contains all the relevant elements of a real set-up.

We assume that the velocity of each DCV varies between 0 m/s and 20 m/s. The lengths of the track segments and their corresponding free flow travel time (the time that a DCV needs to travel a track segment when there is no delay due to congestion) are indicated in Figure 3.

In order to faster assess the efficiency of our control method we assume that we do not start with an empty network but with a network already populated by DCVs transporting bags.

5.2   Scenarios

For the calibration of the weighting parameters we have defined 27 scenarios, each consisting of a stream of 120 bags.

We have considered 3 classes of demand profile called "dp_1", "dp_2", and "dp_3" here after. According to these 3 classes of demand profiles, the bags arrive at each loading station in the time interval $[t_0, t_0 + 100]$ s, the arrival time at a loading station being allocated randomly using uniform distribution ("dp_1"), normal distribution ("dp_2"), and, respectively, half normal distribution ("dp_3"). We also consider 3 different initial states of the system called "$Init_1$", "$Init_2$", and "$Init_3$", where 60 DCVs are already transporting bags in the network, running from loading stations $L_1, L_2, \ldots, L_6$ to junctions S_4, S_2, S_1, S_3, S_7, from S_1 to S_2, and from S_1 to S_3. Their position at t_0 and their static priorities are assigned randomly.

The bags to be handled can be organized in 2 groups of bags. Then let "group 1" consist of the bags that populate the DCV network before t_0 and "group 2" consist of the bags that enter the network after t_0. For a maximum storage period of 100 s, we examine both situations where the transportation of the bags is very tight (the last bag that enters the system can only arrive in time at the corresponding endpoint if the DCV travels the shortest route with maximum speed), and respectively more relaxed. We denote the scenarios according to Table 1.

For comparing the control methods we have used the same scenarios, but different samples of the demand profiles than those considered for calibrating the weighting parameters **w**.

5.3 Results

To solve the optimization problem P_1, P_2, and P_3 we have used *genetic* algorithms with multiple runs since numerical experiments show that for the given application, this technique gives a good performance with a small computation time.

Based on simulations we now compare, for the same scenarios, the proposed control methods. In Figure 4 we plot the results obtained when using the proposed control approaches.

Note that, as expected, the best performance of the system is obtained when using centralized switch control. However, centralized control becomes intractable in practice when the number of junctions is large due to the large computation time it requires. The results indicate that using decentralized MPC lowers the computation time, but not yet enough to obtain real-time control. Note that this computation time is obtained while we run the genetic algorithm 3 times on a single computer for solving one optimization problem. So, the computation time can be lowered even more if we perform parallel computation of the optimization or if we use faster computers. Note also that the implementation has been performed in Matlab. Hence, one may implement the proposed control in object coded programming languages and lower the computation time even more. However, as the simulations show, the decentralized approaches offer a balanced trade-off between optimality and computational effort.

6 CONCLUSIONS AND FUTURE WORK

In this paper we have considered the baggage handling process in large airports that use destination coded vehicles (DCVs) running at high speed on a "mini" railway network, together with the main control problems of a DCV-based baggage handling system. A fast event-driven model of the continuous-time baggage handling process has been determined. In this paper, we have compared approaches that control the route of each DCV transporting bags viz. centralized model predictive control (MPC), decentralized MPC, and a fast heuristic approach.

The results show that the best performance of the system is obtained using centralized control. However, centralized MPC is not tractable in practice due to the large computational effort. On the one hand the computational effort can be lowered by using decentralized MPC, due to the parallel computation. However, in order to obtain real-time results, one can e.g. parallelize the computation of the local control, implement the control method in object coded programming language, or reduce the prediction horizon or the time allowed for solving the local optimization at the cost of suboptimality. On the other hand, the

decentralized heuristic approach usually gives worse results than those obtained when using decentralized MPC, but with very low computation time. Hence, the decentralized approaches offer a balanced trade-off between optimality and computational effort.

In future work we will include communication and coordination between the neighboring junctions, and assess the benefits that could be obtained by using such distributed control. We will also include more complex dynamics of the system than those considered in this paper.

ACKNOWLEDGMENT

REFERENCES

[1] Vanderlande Industries, Baggage handling. URL `www.vanderlande.com`.

[2] de Neufville, R., The Baggage System at Denver: Prospects and Lessons. *Journal of Air Transport Management*, vol. 1, no. 4, 1994, pp. 229–236.

[3] Gang, H., J. Shang, and L. Vargas, A neural network model for the free-ranging AGV route-planning problem. *Journal of Intelligent Manufacturing*, vol. 7, no. 3, 1996, pp. 217–227.

[4] Kaufman, D., J. Nonis, and R. Smith, A mixed integer linear programming model for dynamic route guidance. *Transportation Research Part B: Methodological*, vol. 32, no. 6, 1998, pp. 431–440.

[5] Fay, A., Decentralized control strategies for transportation systems. In *International Conference on Control and Automation*, Budapest, Hungary, June 2005, pp. 898–903.

[6] Hallenborg, K., and Y. Demazeau, Dynamical Control in Large-Scale Material Handling Systems through Agent Technology. In *International Conference on Intelligent Agent Technology*, Hong Kong, China, December 2006, pp. 637–645.

[7] Tarău, A., B. De Schutter, and J. Hellendoorn, Travel time control of destination coded vehicles in baggage handling systems. In *IEEE International Conference on Control Applications*, San Antonio, Texas, September 2008, pp. 293–298.

[8] Maciejowski, J., *Predictive Control with Constraints*. Prentice Hall, Harlow, UK, 2002.

[9] Reeves, C., and J. Rowe, *Genetic Algorithms - Principles and Perspectives: A Guide to GA Theory*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[10] Dowsland, K., Simulated annealing. In *Modern Heuristic Techniques for Combinatorial Problems* (C. Reeves, editor), John Wiley & Sons, Inc., New York, USA, 1993, pp. 20–69.

[11] Glover, F., and F. Laguna, *Tabu Search*. Kluwer Academic Publishers, Norwell, MA, USA, 1997.

[12] Gill, P., W. Murray, and M. Wright, *Practical Optimization*. Academic Press, London, UK, 1981.

[13] Audet, C., and J. Dennis, Analysis of Generalized Pattern Searches. *SIAM Journal on Optimization*, vol. 13, no. 3, 2003, pp. 889—-903.
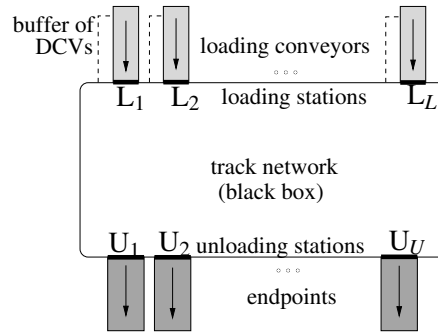
LIST OF TABLES

LIST OF FIGURES

TABLE 1 Considered Scenarios

| Scenario type	Departure time	Initial state	Demand profile	Scenario ID
Relaxed		Init_1	dp_1	1
			dp_2	2
	same		dp_3	3
time window group 1	departure time		dp_1	4
$[t_0+100, t_0+200]$		Init_2	dp_2	5
	$t_{\text{load\_plane,group 1}} = t_0 + 200$		dp_3	6
time window group 2	$t_{\text{load\_plane,group 2}} = t_0 + 200$		dp_1	7
$[t_0+100, t_0+200]$		Init_3	dp_2	8
			dp_3	9
Relaxed		Init_1	dp_1	10
			dp_2	11
	different		dp_3	12
time window group 1	departure time		dp_1	13
$[t_0, t_0+100]$		Init_2	dp_2	14
	$t_{\text{load\_plane,group 1}} = t_0 + 100$		dp_3	15
time window group 2	$t_{\text{load\_plane,group 1}} = t_0 + 200$		dp_1	16
$[t_0+100, t_0+200]$		Init_3	dp_2	17
			dp_3	18
Tight		Init_1	dp_1	19
			dp_2	20
	different		dp_3	21
time window group 1	departure time		dp_1	22
$[t_0, t_0+100]$		Init_2	dp_2	23
	$t_{\text{load\_plane,group 1}} = t_0 + 100$		dp_3	24
time window group 2	$t_{\text{load\_plane,group 1}} = t_0 + 170$		dp_1	25
$[t_0+70, t_0+170]$		Init_3	dp_2	26
			dp_3	27

FIGURE 1 Baggage handling system using DCVs
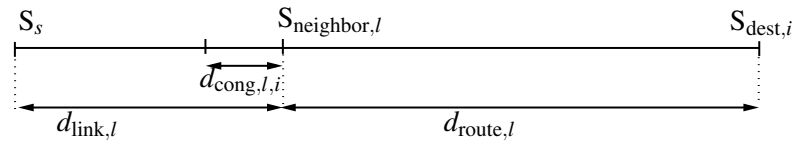
FIGURE 2 Congestion detection on link $l \in \{0, 1\}$
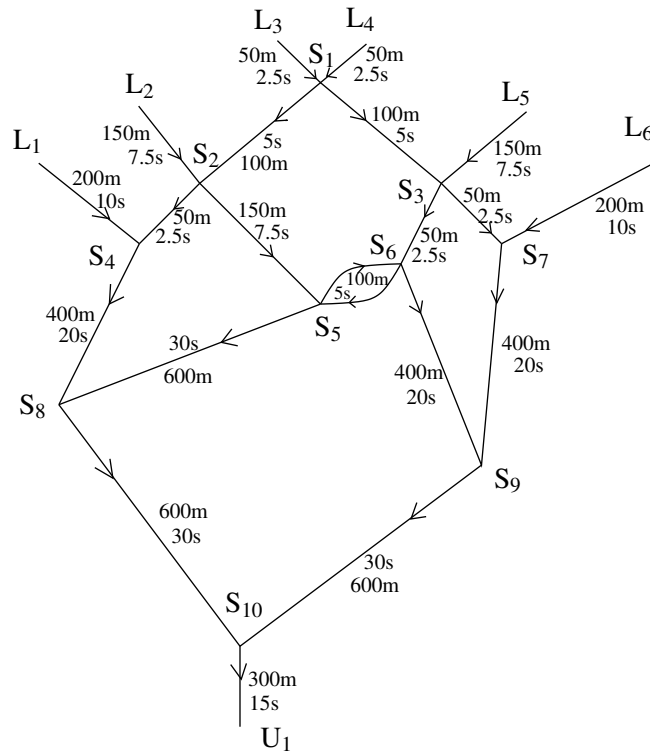
FIGURE 3 Case study for a DCV-based baggage handling system
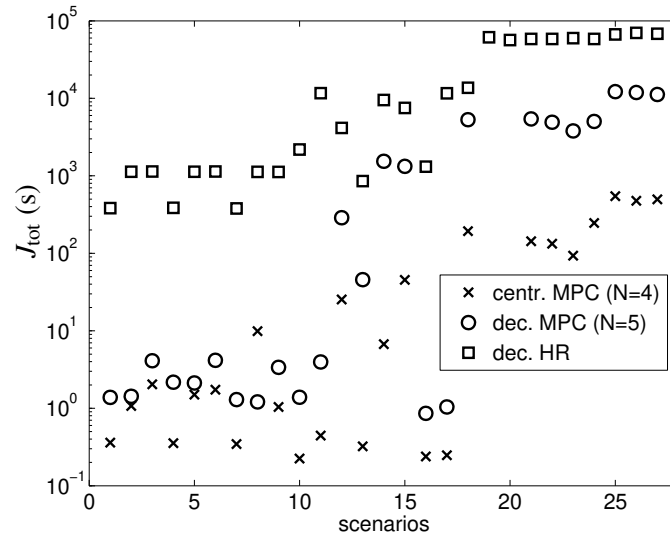
(a) J_{tot} versus scenarios
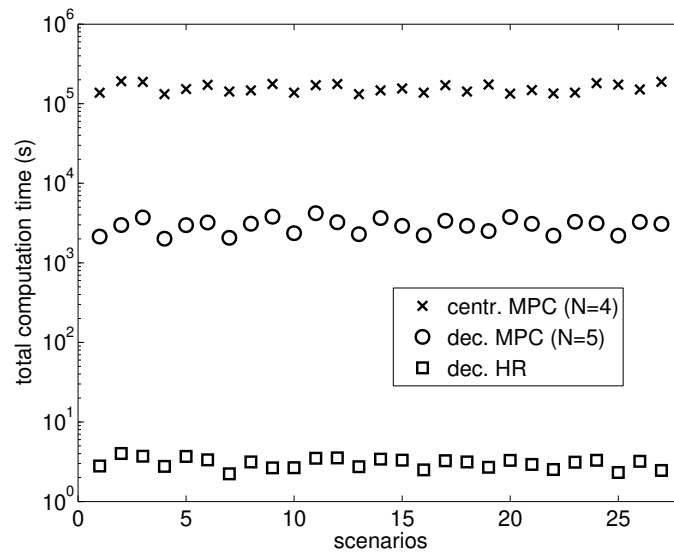
(b) computation time versus scenarios

FIGURE 4 Comparison of the proposed control approaches