Technical report 08-027

Centralized versus decentralized route choice control in DCV-based baggage handling systems*

A. Tarău, B. De Schutter, and H. Hellendoorn

If you want to cite this report, please use the following reference instead:
A. Tarău, B. De Schutter, and H. Hellendoorn, "Centralized versus decentralized route choice control in DCV-based baggage handling systems," *Proceedings of the 2009 IEEE International Conference on Networking, Sensing and Control*, Okayama, Japan, pp. 334–339, Mar. 2009.

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

Centralized versus decentralized route choice control in DCV-based baggage handling systems

Alina Tarău, Bart De Schutter, and Hans Hellendoorn

Abstract— **The process of handling baggage in an airport is time-critical. Currently, the fastest way to transport the luggage is to use destination coded vehicles (DCVs). These vehicles transport the bags at high speed on a "mini" railway network. The route of each DCV has to be computed in order to optimize the performance of the system. In this paper we determine an event-based model of a DCV-based baggage handling system and we compare centralized and decentralized control methods for routing the DCVs through the network. The proposed centralized control methods are optimal control and model predictive control. Due to the large computational effort required, we also analyze a fully decentralized control approach. In this case, each junction has its own local controller for positioning the switch into the junction or out of it. The considered control methods are compared for several scenarios. Results indicate that optimal control becomes intractable when a large stream of bags has to be handled, while MPC can still be used to suboptimally solve the problem. On the other hand, the decentralized control method usually gives worse results than those obtained when using MPC, but with very low computation time.**

I. INTRODUCTION

The baggage handling system of an airport plays a decisive role in the airport's efficiency and comfort, which are among the most important factors that determine the airport's ability to attract new airlines or to stay a major airline hub.

The baggage handling system is performing successfully if all the bags are transported to the corresponding end point[1] before the plane has to be loaded. So, the process is time-critical. The faster the transportation is performed, the more efficient the baggage handling system is. However, due to multiple planes departing at nearby time instants, and due to the limited number of end points, a plane is allocated to an end point only with a given amount of time before its departure. Hence, the baggage handling system works optimally if all the bags arrive at their given end point in a specific time interval.

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.

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 assignment of each bag (and implicitly the switch control of each junction), velocity control of each DCV, line balancing (route assignment of each empty DCV), and prevention of buffer overflows.

Here we focus on the route choice control of the DCVs. In the literature, the route assignment problem has been addressed in e.g. [1], [2].

The goal of this paper is to compare the centralized and decentralized route choice control of each DCV in the baggage handling system by implementing advanced control methods such as optimal control, model predictive control, and a fast heuristic approach.

The paper is organized as follows. In Section II, the baggage handling process using DCVs is described, and afterwards, a continuous-time event-driven model of the system is presented. In Section IV, first several control approaches are proposed for computing the optimal route of each DCV transporting a bag in a *centralized* manner. Afterwards, in order to implement the *decentralized* control [3], we propose several heuristic rules for determining the position of the switches leading into and out of a junction. These rules depend on the weighted static and dynamic priorities of the bags transported by DCVs on the incoming links, on the weighted shortest time path to destination, and on the weighted density of the DCVs on the outgoing links. The weighting parameters are calibrated. The analysis of the simulation results and the comparison of the proposed control methods are elaborated in Section V. Finally, in Section VI, conclusions are drawn and the future directions are presented.

II. EVENT-DRIVEN MODEL

A. Operation of the system

The baggage handling process begins after the bags have passed the check-in. Then they enter the conveyor network, being routed to loading conveyors towards loading stations. Depending on the availability of empty DCVs, at each loading station a queue of bags may be formed. In this paper we focus on the transporting-using-DCVs part of the process as sketched in Figure 1. The baggage handling

All authors are with Faculty of Mechanical, Maritime and Materials Engineering, Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands a.n.tarau@tudelft.nl, b@deschutter.info, j.hellendoorn@tudelft.nl

Bart De Schutter is also with the Marine and Transport Technology department of Delft University of Technology.

[1]An end point is the place where the bags are lined up, waiting to be loaded onto the plane.

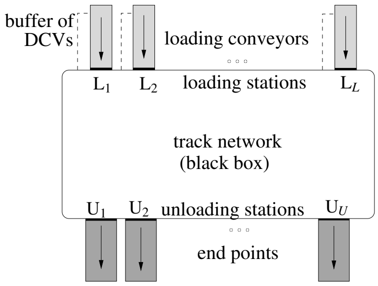


Fig. 1. Baggage handling system using DCVs.

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 optimal route of each DCV has to be computed subject to operational and safety constraints such that the system optimum is ensured.

We consider a baggage handling system with L loading stations and U unloading stations as depicted in Figure 1. Accordingly, we have L FIFO (First In First Out) buffers of bags waiting to enter the system.

B. Modeling assumptions

Later on we will use the model for on-line model-based control. So, in order to obtain a balanced trade-off between a detailed model that requires large computation time and a fast simulation we make the following assumptions:

A_1: a sufficient number of DCVs are present in the system so that when a bag is at the loading station there is a DCV ready for transporting it.

A_2: the "mini" railway network has single-direction tracks.

A_3: each junction has maximum 2 incoming links and 2 outgoing links.

A_4: a route switch at a junction can be performed in a negligible time span.

A_5: the speed of a DCV is piecewise constant.

A_6: the end points have capacity large enough that no buffer overflow can occur.

A_7: the destinations to which the bags have to be transported are allocated to the end points when the process starts.

Since we consider the line balancing problem solved, these assumptions are reasonable and give a good approximation of the real baggage handling system.

C. Model

There are four types of events that can occur:

- loading a new bag into the system.
- unloading a bag that arrives at the corresponding end point.
- updating the position of the route switch of
  1) a junction's switch-in.
  2) a junction's switch-out.
- updating the speed of a DCV.

The model of the baggage handling system is an event-driven one consisting of a continuous part describing the movement of the individual vehicles transporting the bags through the network, and of the discrete events listed above.

We consider that the network has S junctions S_s, $s = 1, \cdots, S$. Let DCV_i be the DCV that transports the ith bag that entered the system and $X_{\mathrm{current}}(t)$ the number of bags that entered the baggage handling system up to the current time instant. Then, the model of the baggage handling system is given by the algorithm below.

Algorithm 1. Model of the baggage handling system

1: $t \leftarrow t_0$
2: **while** there are bags to be handled **do**
3:   **for** $\ell = 1$ to L **do**
4:     $\delta_{\mathrm{load}}(\ell) \leftarrow$ time that will pass until the next loading event of L_ℓ
5:   **end for**
6:   **for** $\ell = 1$ to U **do**
7:     $\delta_{\mathrm{unload}}(\ell) \leftarrow$ time that will pass until the next unloading event of U_ℓ
8:   **end for**
9:   **for** $s = 1$ to S **do**
10:     $\delta_{\mathrm{switch\_in}}(s) \leftarrow$ time that will pass until the next switch-in event of S_s
11:     $\delta_{\mathrm{switch\_out}}(s) \leftarrow$ time that will pass until the next switch-out event of S_s
12:   **end for**
13:   **for** $i = 1$ to $X_{\mathrm{current}}(t)$ **do**
14:     **if** bag i is not at an end point **then**
15:       $\delta_{\mathrm{speed\_update}}(i) \leftarrow$ time that will pass until the next speed-update event of DCV_i
16:     **end if**
17:   **end for**
18:   $\delta_{\min} \leftarrow \min\big(\min_{\ell=1,\cdots,L} \delta_{\mathrm{load}}(\ell), \min_{\ell=1,\cdots,U} \delta_{\mathrm{unload}}(\ell), \min_{s=1,\cdots,S} \delta_{\mathrm{switch\_in}}(s), \min_{s=1,\cdots,S} \delta_{\mathrm{switch\_out}}(s), \min_{i=1,\cdots,X_{\mathrm{current}}(t)} \delta_{\mathrm{speed\_update}}(i)\big)$
19:   $t \leftarrow t + \delta_{\min}$
20:   update the state of the system
21:   **if** $\delta_{\min} = \min_{i=1,\cdots,X_{\mathrm{current}}(t)} \delta_{\mathrm{speed\_update}}(i)$ **then**
22:     update the speed of the DCV_i
23:   **end if**
24: **end while**

If multiple events occur at the same time, then we take all these events into account when updating the state of the system at step 20.

D. Operational constraints

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 traffic jam at a loading station, then no loading event can occur at that loading station.

C_3: a DCV can transport only one bag.

C_4: a switch at a junction changes its position after minimum δ_{x} time units in order to avoid chattering.

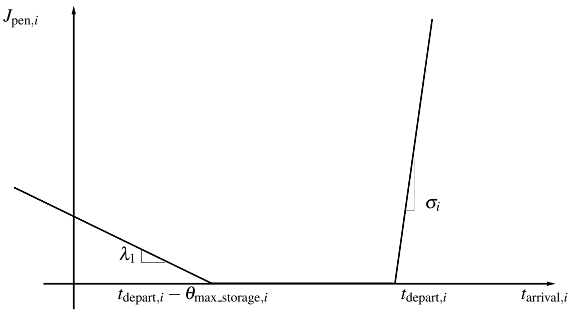


Fig. 2. Objective function $J_{\text{pen},i}$.

III. PERFORMANCE INDEX

We now define the performance index J that will be used in this paper.

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 bag i is to penalize the overdue time and the additional storage time. So, as sketched in Figure 2,

$$\begin{aligned} J_{\text{pen},i} = \ & \sigma_i \max(0, t_{\text{arrival},i} - t_{\text{depart},i}) + \\ & \lambda_1 \max(0, t_{\text{depart},i} - \theta_{\text{max\_storage},i} - t_{\text{arrival},i}) \end{aligned}$$

where $t_{\text{arrival},i}$ is the time instant when the bag i arrives at its corresponding end point, $t_{\text{depart},i}$ is the time instant when the end point closes, σ_i is the static priority[2] of the bag i, and $\theta_{\text{max\_storage},i}$ is the maximum possible time interval for which the end point of bag i is open for that specific flight. The weighting parameter $\lambda_1 < 1$ represents the relative cost of 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 yield difficulty for many optimization algorithms. To get some additional gradient we could also include the dwell time, resulting in:

$$J_i = J_{\text{pen},i} + \lambda_2 t_{\text{dwell},i}$$

where λ_2 is a small weight factor ($\lambda_2 \ll \lambda_1$).

The final performance index is given by $J_{\text{tot}} = \sum_{i=1}^{X} J_i$, where i is the index of the transported bag.

IV. CONTROL APPROACHES

A. Velocity control

In this paper we assume that the velocity of each DCV is always at its maximum 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.

[2]The static priority is the flight priority, bounded between 1 and 10, priority 1 being the lowest priority that a bag can have.

B. Centralized route control

In this paper we consider several centralized control approaches that determine the route of each DCV such as finite-horizon optimal control and model predictive control.

1) Optimal control: Several methods for solving dynamic optimization problems have been developed. The optimal control problem consists of finding the time-varying control law $u(\cdot)$ for a given system such that a performance index $J(u(\cdot))$ is optimized while satisfying the operational constraints imposed by the model, see e.g. [4].

The performance index J is influenced by the route that each bag takes. Assuming that there are R possible routes named $1\,2\cdots,R$, the route of DCV$_i$ is $r(i)$, $i = 1,2,\cdots,X$ with X the number of bags that enter the track network. Then the route sequence is represented by $\mathbf{r} = [r(1)\, r(2) \cdots r(X)]^{\text{T}}$. The piecewise constant speed profile of the DCV$_i$ is defined as $v_i : \{0,1,\cdots,N_i\} \rightarrow [0, v_{\max}]$ where N_i represents the number of speed-update events of DCV$_i$ that are performed from the loading station up to its corresponding end point.

Then the optimal control problem is defined as follows:

P1: $\min_{\mathbf{r}} J_{\text{tot}}(\mathbf{r}, \mathscr{V})$
subject to
the system dynamics
operational constraints

where the tuple $\mathscr{V} = (\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_X)$ with $\mathbf{v}_i = [v_i(0)\, v_i(1) \cdots v_i(N_i)]^{\text{T}}$ for $i = 1,2,\cdots,X$.

But, computing the optimal route of each DCV transporting bags through the network so as to minimize the performance index J_{tot} requires extremely high computational effort. In practice, the problem P1 becomes intractable when the number of possible routes and the number of bags to be transported are large.

2) Model predictive control: In order to make a trade-off between the optimality and the time required to compute the optimal route of each DCV transporting bags, model predictive control (MPC) is introduced.

Model predictive control is an on-line control design method that uses the receding horizon principle, see e.g. [5].

In the basic MPC approach, given a prediction horizon N_{p} and a control horizon N_{c} with $N_{\text{c}} \leq N_{\text{p}}$, at time step k, the future control sequence $u(k|k), \cdots, u(k+N_{\text{c}}-1|k)$ is computed by solving a discrete-time optimization problem over a period $[k, k+N_{\text{p}}]$ so that a cost criterion J is optimized subject to constraints on the inputs and outputs. The input signal is typically assumed to become constant beyond the control horizon i.e. $u(k+j|k) = u(k+N_{\text{c}}-1|k)$ for $j \geq N_{\text{c}}$. 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 $k+1$ 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. Also, computing the control $u(k|k)$ consists of determining the route of DCV$_k$.

In this variant of MPC the prediction horizon corresponds to the number of bags that we let to enter the track network. The control horizon is equal to the prediction horizon ($N_\mathrm{p} = N_\mathrm{c} = N$) since, in this case, the control horizon constraint cannot be applied. This happens due to the fact that the DCVs transporting the bags do not always have the same destination, and, therefore, assigning them the same route obviously implies suboptimal performance. At step k, where k is the number of bags in the network, the control $u(j|k) = r(k+j)$ for $j = 1,2,\cdots,N$ is computed such that the performance index $J_{\mathrm{tot},N}$ over the next N bags that enter the track network is minimized.

The MPC optimization problem at bag step k is defined as follows:

P2: $\min_{\mathbf{r}(k)} J_{\mathrm{tot},N}(\mathbf{r}(k), \mathscr{V}(k))$
subject to
the system dynamics
operational constraints

where, at step k, $\mathbf{r}(k) = [r(k+1)\ r(k+2) \cdots r(k+N)]^\mathrm{T}$ is the future route sequence and $\mathscr{V}(k) = (\mathbf{v}_{k+1}, \mathbf{v}_{k+2}, \cdots, \mathbf{v}_{k+N})$ with $\mathbf{v}_{k+i} = [v_{k+i}(0)\ v_{k+i}(1) \cdots v_{k+i}(N_i)]^\mathrm{T}$ for $i = 1,2,\cdots,N$ is the future velocity profile for the next N bags entering the network.

Only the first control $r(k+1)$ will be applied. Given the state of the system after applying the MPC control, a new optimization will be solved over the prediction horizon.

The main advantage of MPC consists in a smaller computation time than the one needed when using optimal control. Even more, the route of each DCV may be computed online. However, this happens at the cost of a suboptimal performance of the baggage handling system.

3) Optimization methods: In order to solve the optimization problems presented in the previous subsections, the route for each DCV has to be determined. The route is represented by an integer value. Therefore, to solve any of the optimization problems P1 or P2 one might use *mixed-integer* algorithms such as branch and bound methods or *genetic* algorithms see e.g. [6], [7].

C. Decentralized route control

In order to lower even more the computation time of solving our route choice problem, in this section the route of each DCV is controlled by the baggage handling system in a decentralized way. So, each switch is locally controlled based on heuristic rules as presented in the next subsections.

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_current.

1) Control of the switch-in: Recall that each switch into a junction has maximum 2 incoming links indexed by $l \in \{1,2\}$ as sketched in Figure 3.

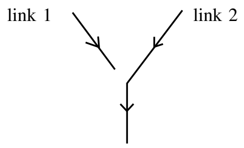


Fig. 3. Incoming links at a junction. The switch-in positioned on link 2.

For a junction S_s, with $s \in \{1,2,...,S\}$, we define the following variables:

- $\Gamma_l(s)$ is the set of bags transported on the incoming link $l \in \{1,2\}$ of junction S_s.
- $a_{s,l} = \sum_{i \in \Gamma_l(s)} \sigma_i$.
- $b_{s,l} = \sum_{i \in \Gamma_l(s)} \frac{c_{i,l}}{d_i}$ with $c_{i,s}$ the time required to cover the shortest distance from the current position of bag i to its destination in case of no congestion and average speed, and d_i the maximum time left to bag i to spend in the system. The variable d_i is defined as $d_i \leftarrow t_{\mathrm{depart},i} - t_\mathrm{current}$ if $t_{\mathrm{depart},i} - t_\mathrm{current} > 0$ and $d_i \leftarrow t_{\mathrm{new\_flight},i} - t_\mathrm{current}$ if $t_{\mathrm{depart},i} - t_\mathrm{current} \leq 0$ with $t_{\mathrm{new\_flight},i}$ the time instant when, for bag i, a new flight is associated to an end point.

Furthermore, in order to determine the next position of the switch-in at junction S_s for $s = 1,2,...,S$, we compute the performance measure $p_{l,\mathrm{switch\_in}}(s)$ for $l = 1,2$. This performance measure takes into account the static and dynamic priorities of the bags transported by DCVs on the incoming link l of the junction S_s, and the current position of the switch-in at junction S_s (due to the operational constraint C_4). The weighting parameters $w_{\mathrm{st\_priority}}$, $w_{\mathrm{dyn\_priority}}$, and $w_{\mathrm{switch\_in}}$ can be calibrated as explained in Section IV-C.3.

We denote $\alpha = 1$ if the switch is positioned on the incoming link 1 and $\alpha = 2$ if the switch is positioned on the incoming link 2. In Algorithm 2, $I_\alpha \leftarrow 0$ if $\alpha = 1$ and $I_s \leftarrow 1$ if $\alpha = 2$.

The control of the switch-in at junction S_s, with $s \in \{1,2,...,S\}$ is given in the Algorithm 2 below.

Algorithm 2. Control of switch-in at junction S_s

$t_{\mathrm{switch\_in}}(s) \leftarrow \infty$
while there are bags traveling towards junction S_s **do**
  $p_{1,\mathrm{switch\_in}}(s) \leftarrow w_{\mathrm{st\_priority}} a_{s,1} + w_{\mathrm{dyn\_priority}} b_{s,1} - w_{\mathrm{switch\_in}} \delta_\mathrm{x} I_s$
  $p_{2,\mathrm{switch\_in}}(s) \leftarrow w_{\mathrm{st\_priority}} a_{s,2} + w_{\mathrm{dyn\_priority}} b_{s,2} - w_{\mathrm{switch\_in}} \delta_\mathrm{x} (1 - I_s)$
  if $(p_{1,\mathrm{switch\_in}}(s) > p_{2,\mathrm{switch\_in}}(s)$ **and** $I_s = 1)$ **or** $(p_{2,\mathrm{switch\_in}}(s) > p_{1,\mathrm{switch\_in}}(s)$ **and** $I_s = 0)$ **then**
    $t_{\mathrm{switch\_in}}(s) \leftarrow$ the time when the junction S_s's switch-in changes its position
  end if
end while

2) Control of the switch-out: Recall that also each switch out of a junction has maximum 2 outgoing links indexed by $l \in \{1,2\}$ as sketched in Figure 4.

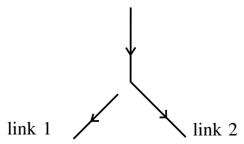


Fig. 4. Outgoing links at a junction. The switch-out positioned on link 2.

In order to determine the control of the switch-out, for each DCV passing the junction S_s, for $s = 1,2,...,S$ we compute the new performance measure $p_{l,\text{switch\_out}}(s)$ for $l = 1,2$ that takes into account the density $\rho_{l,s}$ of DCVs on the outgoing link l of junction S_s, the shortest time τ_s from junction S_s to the corresponding end point (for an average speed), and the current position of the outgoing switch (due to the operational constraint C_4) as shown in Algorithm 3 below. The weighting parameters w_{time}, w_{density}, and $w_{\text{switch\_out}}$ can be calibrated as explained in the next subsection.

We also denote $\beta = 1$ if the switch-out is positioned on the outgoing link 1 and $\beta = 2$ if the switch-out is positioned on link 2. In Algorithm 3, $I_\beta \leftarrow 0$ if $\beta = 1$ and $O_s \leftarrow 1$ if $\beta = 2$.

Algorithm 3. Control of switch-out at junction S_s

$t_{\text{switch\_out}}(s) \leftarrow \infty$
while there are bags at junction S_s **do**
  $p_{1,\text{switch\_out}}(s) \leftarrow w_{\text{density}}\rho_{1,s} + w_{\text{time}}\tau_s + w_{\text{switch\_out}}\delta_{\text{x}} O_s$
  $p_{2,\text{switch\_out}}(s) \leftarrow w_{\text{density}}\rho_{2,s} + w_{\text{time}}\tau_s + w_{\text{switch\_out}}\delta_{\text{x}} (1 - O_s)$
  if $(p_{1,\text{switch\_out}}(s) < p_{2,\text{switch\_out}}(s)$ **and** $O_s = 1)$ **or** $(p_{2,\text{switch\_out}}(s) < p_{1,\text{switch\_out}}(s)$ **and** $O_s = 0)$ **then**
    $t_{\text{switch\_out}}(s) \leftarrow$ the time when the junction S_s's switch-out changes its position
  end if
end while

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

$$\text{P3:} \quad \min_{\mathbf{w}} \frac{1}{N_{\text{scenario}}} \sum_{j=1}^{N_{\text{scenario}}} J_{\text{tot},j}(\mathbf{w})$$

subject to
the system dynamics depending on $\mathbf{w}$
operational constraints

where $\mathbf{w} = [w_{\text{st\_priority}}\ w_{\text{dyn\_priority}}\ w_{\text{time}}\ w_{\text{density}}\ w_{\text{switch\_in}}\ w_{\text{switch\_out}}]^{\text{T}}$ and N_{scenario} is the number of scenarios over which the calibration is performed.

So as to solve the optimization problem P3, one might also use e.g. *pattern search* or *simulated annealing* algorithms, see e.g. [8], [9].

V. Results

A. Set-up

We consider the network of tracks depicted in Figure 5 with one loading station, one unloading station, and four 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 are indicated in Figure 5.

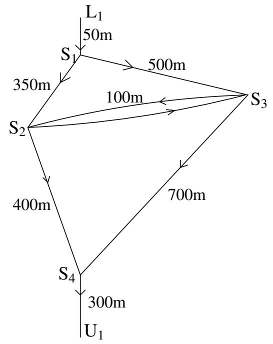


Fig. 5. Case study for a DCV-based baggage handling system.

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.

B. Scenarios

For the calibration of the weighting parameters we have defined 27 scenarios, each consisting of a stream of 200 bags, the arrival at the loading station of each bag being dynamically assigned.

Three demand profiles have been considered. Their approximation is illustrated in Figure 6.

We have also considered 3 different initial states of the system, namely:

- Init_1: 20 DCVs transporting bags are on the link from S_1 to S_2, in the first 60 m. These DCVs have priority 10.
- Init_2: the same DCVs as at Init_1 and, additionally, 10 DCVs transporting bags are on the link from S_2 to S_4, and 10 DCVs are on the link from S_3 to S_4. The bags transported by these DCVs have priority 1 and are located in the first 20 m of each link.
- Init_3: 10 DCVs transporting bags are on the link from S_1 to S_2 and 10 more DCVs are on the link from S_1 to S_3. They transport bags with priority 15 and respectively 1, being located in the first 20 m of each link.

The departure time of the bags is first considered to be the same for all the bags (there is only one plane onto which they have to be loaded). Afterwards, we consider that the group of bags transported by DCVs through the network before t_0 have an earlier departure time than the group of bags that arrive at the loading station after t_0. Finally, 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 end point if maximum speed of the DCV

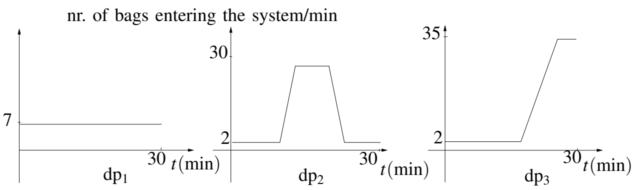


Fig. 6. Demand profile.

is continuously used), and respectively more relaxed. So, in total we consider $3 \cdot 3 \cdot 2 \cdot 2 = 36$ scenarios.

For comparing the control methods we have used different samples of the demand profiles than the ones considered for calibrating the weighting parameters of the decentralized route choice control.

C. Analysis

To solve the optimization problems P1 and P2 we have chosen the *genetic* algorithms (bitstring population) with multiple runs since simulations show that *mixed-integer* algorithms require more computation time than *genetic* algorithms without having as result a smaller performance index. Also, the weighting factors of P3 have been optimized over all the considered scenarios using *simulated annealing* algorithms with multiple initial points. This optimization technique has been chosen since the optimization is performed off-line, the technique giving the best performance.

Based on simulations we now compare, for the same scenarios (where new samples of demand profiles have been used), the proposed control methods. In Figure 7 we plot the results obtained when using MPC and the fully decentralized approach. The best performance of the baggage handling system would be obtained using optimal control. However, in our experiments, for optimal control the computation time[3] would exceed 10^6 s. Therefore, these results have not been plotted in Figure 7.

Results indicate that MPC involves a trade-off between computation time and optimality, the performance being influenced by the prediction horizon. Increasing the prediction horizon the performance improves, but at the cost of higher computation time. On the other hand, the decentralized approach performs very fast, but, usually, the results are worse than the ones obtained when using MPC.

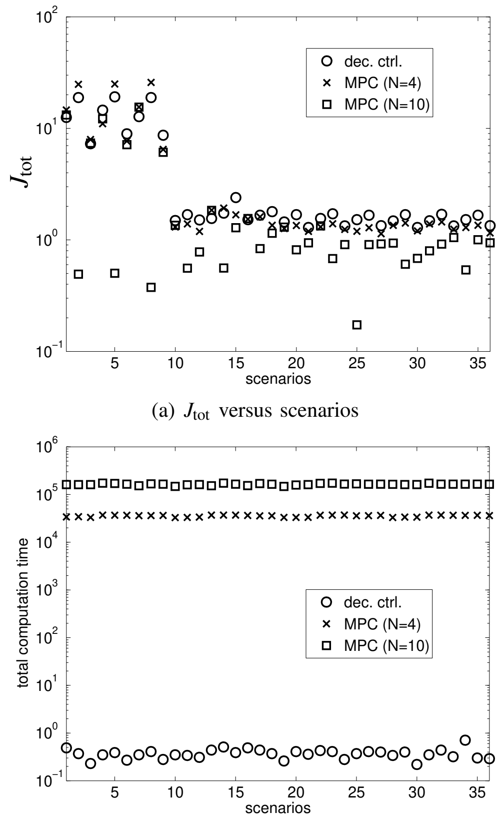
VI. Conclusions and future work

In this paper we have considered the baggage handling process in large airports using destination coded vehicles (DCVs) running at high speed on a “mini” railway network, together with the main control problems of the baggage handling systems. A fast event-driven model of the continuous-time baggage handling process has been determined. We have compared the centralized and decentralized route choice control of the baggage handling systems using DCVs.

Theoretically, the best performance is obtained using centralized optimal control. However, centralized optimal control is not tractable in practice due to the very high computational efforts. Centralized MPC involves a trade-off between computation time and optimality, the performance being influenced by the prediction horizon. Moreover, the decentralized approach performs very fast, but usually, the results are worse than those obtained when using MPC.

In future work we will include communication of the local control between the neighboring junctions, and verify the benefit obtained by looking farther in the system using distributed control.

[3]The simulations were performed on a 3.0 GHz P4 with 1 GB RAM.



(a) J_{tot} versus scenarios

(b) computation time versus scenarios

Fig. 7. Comparison of the proposed control approaches.

Acknowledgment

Research supported by the STW-VIDI project “Multi-Agent Control of Large-Scale Hybrid Systems”, by the BSIK project “Next Generation Infrastructures” (NGI), by the Transport Research Centre Delft, by the Delft Research Centre Next Generation Infrastructures, and by the European STREP project “Hierarchical and Distributed Model Predictive Control” (HD-MPC).

References

[1] T. Friesz, J. Luque, R. Tobin, and B. Wie, “Dynamic network traffic assignment considered as a continuous time optimal control problem,” *Operations Research*, vol. 37, no. 6, pp. 893–901, 1989.

[2] M. Duinkerken, J. Ottjes, and G. Lodewijks, “Comparison of routing strategies for AGV systems using simulation,” in *Proceedings of the 2006 Winter Simulation Conference*, Monterey, California, 2006, pp. 1523–1530.

[3] D. Siljak, *Decentralized Control of Complex Systems*. San Diego, California: Academic Press, INC, 1991.

[4] F. Lewis, *Optimal Control*. New York, USA: John Wiley & Sons, Inc., 1986.

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

[6] C. Floudas, *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. New York, USA: Oxford University Press, 1995.

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

[8] C. Audet and J. Dennis, “Analysis of generalized pattern searches,” *SIAM Journal on Optimization*, vol. 13, no. 3, pp. 889—-903, 2003.

[9] K. Dowsland, “Simulated annealing,” in *Modern heuristic techniques for combinatorial problems*, C. Reeves, Ed. New York, USA: John Wiley & Sons, Inc., 1993, ch. 2, pp. 20–69.