

Technical report 09-030

# **Predictive route choice control of destination coded vehicles with mixed integer linear programming optimization\***

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

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

A.N. Tarău, B. De Schutter, and J. Hellendoorn, "Predictive route choice control of destination coded vehicles with mixed integer linear programming optimization," *Proceedings of the 12th IFAC Symposium on Transportation Systems*, Redondo Beach, California, pp. 64–69, Sept. 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>

---

\*This report can also be downloaded via [https://pub.deschutter.info/abs/09\\_030.html](https://pub.deschutter.info/abs/09_030.html)

# Predictive route choice control of destination coded vehicles with mixed integer linear programming optimization

A.N. Tarău \* B. De Schutter \*\* J. Hellendoorn \*

\* *Delft Center for Systems and Control*  
*Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands*  
{a.n.tarau, j.hellendoorn}@tudelft.nl

\*\* *Marine and Transport Technology Department*  
*Delft University of Technology, The Netherlands, b@deschutter.info*

---

**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. In this paper we consider the problem of controlling the route of each DCV in the system. This is a nonlinear, nonconvex, mixed integer optimization problem. Nonlinear model predictive control (MPC) for mixed integer problems is usually very expensive in terms of computational effort. Therefore, in this paper we present an alternative approach for reducing the complexity of the computations by simplifying and approximating the nonlinear optimization problem by a mixed integer linear programming (MILP) problem. The advantage is that for MILP optimization problems solvers are available to allow us to efficiently compute the global optimal solution. The solution of the MILP problem can then be used as a good initial starting point for the original nonlinear optimization problem. To assess the performance of the proposed formulation of the MPC optimization problem, we consider a benchmark case study, the results being compared for several scenarios.

*Keywords:* Baggage handling systems, route choice control, model predictive control.

---

## 1. INTRODUCTION

Modern baggage handling systems in airports transport luggage at high speeds using destination coded vehicles (DCVs). These vehicles transport the bags at high speed on a “mini” railway network. Low-level controllers ensure the coordination and synchronization when loading a bag onto a DCV, in order to avoid damaging the bags or blocking the system, and when unloading it to the corresponding end point. Low-level controllers also compute the velocity of the DCVs such that collisions are avoided. Currently, the DCVs are routed through the system using routing schemes based on preferred routes. These routing schemes can be adapted to respond on the occurrence of predefined events. However, as argued by de Neufville (1994), the patterns of loads on the system are highly variable, depending on e.g. the season, time of the day, type of aircraft at each gate, number of passengers for each flight. Therefore, in the research we conduct we do not consider predefined preferred routes. Instead we develop advanced control methods to determine the optimal routing in case of dynamic demand.

For applications such as automated guided vehicles route planning or traffic route guidance, the route assignment problem has been addressed by e.g. Gang et al. (1996); Kaufman et al. (1998). 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. Fay (2005) solved the routing problem of DCVs transporting bags using an analogy of how data are transmitted via internet, but without presenting any experimental results. Also, Hallenborg and Demazeau (2006) present a multi-agent approach for the control

software of a DCV-based baggage handling system. However, this multi-agent system is faced with major challenges due to the extensive communication required. The goal of our work is to develop and compare efficient control approaches for route choice control of each DCV on the track network.

Theoretically, the maximum performance of such a DCV-based baggage handling systems would be obtained if one computes the optimal routes using optimal control (Lewis, 1986). However, as shown by Tarău et al. (2008), this control method becomes intractable in practice due to the heavy computation burden. Therefore, in order to make a trade-off between computational effort and optimality, in (Tarău et al., 2009), we have also implemented centralized and decentralized model predictive control (MPC), and also a decentralized heuristic approach. As the results confirmed, centralized MPC requires high computation time to determine a solution. The use of decentralized control lowers the computation time, but at the cost of suboptimality.

In this paper we investigate whether the computational effort required for computing the route of each DCV by using MPC can be lowered even more by using mixed integer linear programming (MILP). The large computation time obtained in previous work comes from solving the nonlinear, nonconvex, mixed integer optimization problems. Note that such problems may also have multiple local minima and are NP hard, and therefore, difficult to solve. So, in this paper we rewrite the route choice problem as an MILP problem for which efficient solvers are available. The solution of this MILP can then be

used as an initial starting point for the original nonlinear optimization problem.

The paper is organized as follows. Section 2 briefly introduces the concepts of MPC that will be later on used in solving the route choice problem. In Section 3, we briefly recapitulate an event-driven route choice model that we have developed (Tarău et al., 2008). Afterwards, in Section 4 we approximate the model by using MILP equivalences. Both the nonlinear and MILP model are then used to determine the route of DCVs using MPC. The analysis of the simulation results and the comparison of the proposed formulations are elaborated in Section 6. Finally, Section 7 draws the conclusions for the paper.

## 2. BACKGROUND

Since later on we will use model predictive control (MPC) for determining the routes of the DCVs in the network, in this section we briefly introduce the basic MPC concepts.

MPC is an on-line model-based predictive control design method (Maciejowski, 2002) that uses a receding horizon principle. As illustrated in Fig. 1, in the basic MPC approach, given a horizon  $N$ , at step  $k \geq 0$ , where  $k$  is integer valued, corresponding to the time instant  $t_k = kT_s$  with  $T_s$  the sampling time, the future control sequence  $u(k), u(k+1), \dots, u(k+N-1)$  is computed by solving a discrete-time optimization problem over the period  $[t_k, t_k + NT_s]$  so that a performance index defined over the considered period  $[t_k, t_k + NT_s]$  is optimized subject to the operational constraints. 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  $t_{k+1}$  is solved using this new information. In this way, a feedback mechanism is introduced.

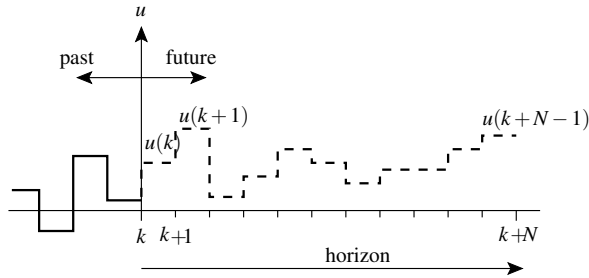


Fig. 1. Prediction horizon in MPC.

## 3. MODELS

### 3.1 System description and original model

In this section we briefly recapitulate the event-driven route choice model of a baggage handling system that we have developed in (Tarău et al., 2008).

The DCV-based baggage handling system operates as follows: given a demand of bags and 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 (Tarău et al., 2008) 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 switches into and out of a junction, and updating the speed of a DCV. The state of the system consists of the positions of the DCVs in the network and the positions of each switch of the network. According to the discrete-event model of (Tarău et al., 2008), as long as there are bags to be handled, given the current state, the system evolves as follows: 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: the speed of each DCV is bounded between 0 and  $v_{\max}$ , while a switch at a junction has to wait at least  $T_s$  time units between two consecutive switches in order to avoid the quick and repeated back and forth movements of the switch which may lead to mechanical damage.

### 3.2 Simplified route choice model

*Network* We represent the mini railway network that DCVs use to transport the luggage as a directed graph. Then the nodes via which the DCVs enter the network are called loading stations, the nodes via which the DCVs unload the transported bags are called unloading stations, while all other nodes in the network are called junctions. The section of track between two nodes is called track segment (or link). For each track segment a free-flow travel time is assigned. This free-flow travel time represents the time period that a DCV requires to travel through a track segment in case of no congestion, using, hence, maximum speed. In order to simplify the explanation of our approach we assume that the free-flow travel time of a link is always a multiple of  $T_s$ .

We assume without loss of generality that in our network each junction has maximum two incoming and maximum two outgoing links indexed by  $l \in \{0, 1\}$  as illustrated in Fig. 2. This assumption of a network corresponds to current practice in state-of-the-art baggage handling systems.

*Extra model assumptions* In order to transform the route choice problem into an MILP problem, we first simplify it by assuming the following:

- We only determine the position of the switches out of junctions. We do not control the position of the switches into junctions. For these switches we assume that low-level controllers are installed to toggle the position such that a DCV can enter the junction as soon as possible. This assumption lowers the computational complexity. Note however that an extension to also controlling the switch into the junction is straightforward.

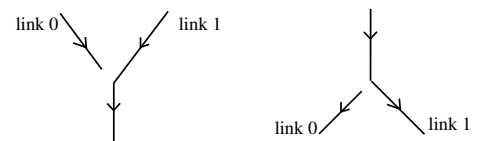


Fig. 2. Incoming and outgoing links at a junction. Both switches are positioned on link 1.

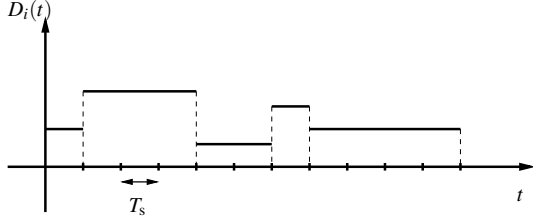


Fig. 3. Demand profile.

- The DCVs run with maximum speed along the track segment and, if necessary, they wait before crossing the junction in a vertical queue.
- The dynamic demand  $D_i$  of loading station  $L_i$ ,  $i \in \{1, \dots, L\}$ , where  $L$  is the number of loading stations, is approximated with a piecewise constant demand as illustrated in Fig. 3. The piecewise constant demand  $D_i$  has level changes occurring only at integer multiples of  $T_s$ . This is necessary in order to easily combine the time when a bag reaches a queue at a junction with the time when the demand changes. So, in the time interval  $[t_k, t_{k+1})$ , with  $t_k = kT_s$ , the demand is  $D_i(k)$ .

*Simplified model* In order to illustrate the derivation of the route choice model let us now consider the most complex cell a network can contain, which is depicted in Fig. 4 where junction  $S_r$  has two neighboring junctions  $S_s$  and  $S_p$  connected via its incoming links.

Next we present how the evolution of the queue length at junction  $S_r$  is determined.

The control time step for each junction in the network is  $T_s$ . So, at each step  $k \geq 0$  the control actions  $u_s(k)$  and  $u_p(k)$  are computed for junctions  $S_s$  and  $S_p$ . A control action at step  $k$  corresponds to the position of the switch on the outgoing link 0 or 1 of a junction during the period  $[t_k, t_{k+1})$ . So, at step  $k$  each of the control signals  $u_s(k)$  and  $u_p(k)$  is either 0 or 1.

Let  $q_r(k)$  denote the length of the queue at junction  $S_r$  at time step  $k$ . Recall that each link in the network has been assigned a given free-flow travel time. Let us denote the link between two nodes  $a$  and  $b$  as  $a \rightarrow b$ . Then, as illustrated in Fig. 4, the free-flow of the link  $S_s \rightarrow S_r$  is  $T_{sr}$  and the free-flow of the link  $S_p \rightarrow S_r$  is  $T_{pr}$ . Hence, the control signals  $u_s(k)$  and  $u_p(k)$  influence  $q_r$  after  $\frac{T_{sr}}{T_s}$  and respectively  $\frac{T_{pr}}{T_s}$  time steps.

The evolution of queue  $q_r$ , the length of which is always greater than or equal to 0, is given by:

$$q_r(k+1) = \max(0, f_r(k)) \quad (1)$$

where  $f_r(k)$  is defined as:

$$f_r(k) = q_r(k) + (I_r(k) - O_{\max})T_s$$

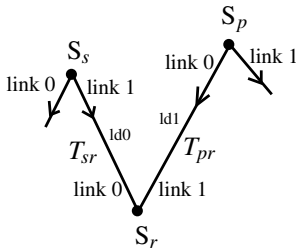


Fig. 4. Network elements.

with  $I_r(k)$  denoting the number of vehicles that enter junction  $S_r$  or the vertical queue at  $S_r$  during the period  $[t_k, t_{k+1})$  and  $O_{\max}$  the maximum outflow<sup>1</sup> of a junction.

The variable  $I_r(k)$  is defined as follows:

$$I_r(k) = u_s(k - \frac{T_{sr}}{T_s})O_s(k - \frac{T_{sr}}{T_s}) + (1 - u_p(k - \frac{T_{pr}}{T_s}))O_p(k - \frac{T_{pr}}{T_s}) \quad (2)$$

where  $O_s(k)$  and  $O_p(k)$  are the outflow of junction  $S_s$  and respectively  $S_p$  during  $[t_k, t_{k+1})$ . If  $k < 0$ , then  $O_j(k)$  is equal to 0 by definition.

The term  $u_s(k)O_s(k)$  represents the inflow<sup>2</sup> of the link  $S_s \rightarrow S_r$  at step  $k$  due to the control action  $u_s(k)$ . So, if  $u_s(k) = 0$  the inflow of the link  $S_s \rightarrow S_r$  at step  $k$  is 0. Similarly, the term  $(1 - u_p(k))O_p(k)$  represents inflow of the link  $S_p \rightarrow S_r$  at step  $k$ . Note that, in (2), these terms appear with a delay of  $\frac{T_{sr}}{T_s}$  and respectively  $\frac{T_{pr}}{T_s}$  time steps due to the free-flow of links  $S_s \rightarrow S_r$  and respectively  $S_p \rightarrow S_r$ .

For  $k \geq 0$  the outflow  $O_j(k)$  with  $j \in \{s, p\}$ , is defined as:

$$O_j(k) = \min\left(O_{\max}, \frac{q_j(k)}{T_s} + I_j(k)\right) \quad (3)$$

#### 4. MIXED INTEGER LINEAR PROGRAMMING

In this section we transform the model presented above using mixed integer linear programming (MILP) theory.

##### 4.1 Background

To remove the nonlinearities of (1)-(3) we will use the following equivalences, see (Bemporad and Morari, 1999), where  $f$  is a function defined on a bounded set  $X$  with upper and lower bounds  $M$  and  $m$  for the function values,  $\delta$  is a binary valued scalar variable,  $y$  is a real valued scalar variable, and  $\epsilon$  is a small tolerance (typically the machine precision):

**P1:**  $[f(x) \leq 0] \iff [\delta = 1]$  is true if and only if

$$\begin{cases} f(x) \leq M(1 - \delta) \\ f(x) \geq \epsilon + (m - \epsilon)\delta \end{cases}$$

**P2:**  $y = \delta f(x)$  is equivalent to

$$\begin{cases} y \leq M\delta \\ y \geq m\delta \\ y \leq f(x) - m(1 - \delta) \\ y \geq f(x) - M(1 - \delta) \end{cases}$$

The tolerance  $\epsilon$  is needed to transform a constraint of the form  $y > 0$  into  $y \geq 0$ , since in MILP problems only nonstrict inequalities are allowed.

##### 4.2 MILP model

In this section we use the MILP properties presented above in order to obtain an MILP model for the simplified route choice model given by equations (1)-(3).

<sup>1</sup> The outflow of a junction is defined as the number of vehicles that cross that junction per time unit.

<sup>2</sup> The inflow of a link equals the number of vehicles that entered that link per time unit.

We start by transforming (3) using Property **P1**. So, we introduce the binary variable  $\delta_j^{\text{out}}(k)$  with  $j \in \{s, p\}$  which equals 1 if and only if  $O_{\max} \leq \frac{q_j(k)}{T_s} + I_j(k)$ . Then we rewrite (3) as follows:

$$O_j(k) = \delta_j^{\text{out}}(k)O_{\max} + (1 - \delta_j^{\text{out}}(k))\left(\frac{q_j(k)}{T_s} + I_j(k)\right) \quad (4)$$

where the condition  $\delta^{\text{out}} = 1$  if and only if  $O_{\max} - \frac{q_j(k)}{T_s} - I_j(k) \leq 0$  is equivalent to (conform Property **P1**):

$$\begin{cases} \frac{q_j(k)}{T_s} + I_j(k) \geq O_{\max} \delta_j^{\text{out}}(k) \\ O_{\max} - \frac{q_j(k)}{T_s} - I_j(k) \geq \varepsilon + (O_{\max} - q_{\max} - I_{\max} - \varepsilon) \delta_j^{\text{out}}(k) \end{cases}$$

with  $q_{\max}$  the maximum possible length of the queue and  $I_{\max}$  the maximum possible value for  $I_j$  with  $j \in \{s, p\}$ .

But (4) is not yet linear, so, we use Property **P2** and introduce the real-valued scalar variable  $y_j^{\text{queue}}(k)$  such that:

$$y_j^{\text{queue}}(k) = \delta_j^{\text{out}}(k)q_j(k)$$

or equivalently:

$$\begin{cases} y_j^{\text{queue}}(k) \leq q_{\max} \delta_j^{\text{out}}(k) \\ y_j^{\text{queue}}(k) \geq 0 \\ y_j^{\text{queue}}(k) \leq q_j(k) \\ y_j^{\text{queue}}(k) \geq q_j(k) - q_{\max}(1 - \delta_j^{\text{out}}(k)) \end{cases} .$$

and the real-valued scalar variable  $y_j^{\text{inflow}}(k)$  such that:

$$y_j^{\text{inflow}}(k) = \delta_j^{\text{out}}(k)I_j(k)$$

or its equivalent set of inequalities of Property **P2** for  $f(x) = I_j(k)$ ,  $M = I_{\max}$ , and  $m = 0$ .

Hence, one obtains:

$$O_j(k) = \delta_j^{\text{out}}(k)O_{\max} + \frac{1}{T_s}q_j(k) + I_j(k) - \frac{1}{T_s}y_j^{\text{queue}}(k) - y_j^{\text{inflow}}(k)$$

which is linear.

Now, in order to transform (2), we introduce the extra variables  $y_{us}(k) = u_s(k)O_s(k)$  and  $y_{up}(k) = u_p(k)O_p(k)$  and the corresponding set of linear inequalities of Property **P2** for  $f(x) = O_s(k)$  and respectively  $f(x) = O_p(k)$ , with  $M = O_{\max}$ , and  $m = 0$ , and we obtain the linear equation:

$$I_r(k) = y_{us}(k - \frac{T_{sr}}{T_s}) + O_p(k - \frac{T_{pr}}{T_s}) - y_{up}(k - \frac{T_{pr}}{T_s}) \quad (5)$$

Finally, we want to transform (1). So, we introduce the binary variable  $\delta_r(k)$  which equals 1 if and only if  $f_r(k) \leq 0$  and we rewrite (1) as:

$$q_r(k+1) = (1 - \delta_r(k))f_r(k) \quad (6)$$

together with the set of linear inequalities of Property **P1** for  $M = q_{\max} + O_{\max}T_s$  and  $m = -O_{\max}T_s$ .

However (6) is not yet linear. Therefore, we introduce an additional variable  $y_r(k) = \delta_r(k)f_r(k)$  and the set of linear inequalities of Property **P2** for  $f(x) = f_r(k)$ ,  $M = q_{\max} + O_{\max}T_s$ , and  $m = -O_{\max}T_s$ , and we obtain:

$$q_r(k+1) = f_r(k) - y_r(k) \quad (7)$$

which is linear.

If we now collect all the variables for the model (i.e.  $q_r(k)$ ,  $f_r(k)$ ,  $I_r(k)$ ,  $y_r(k)$ ,  $y_{us}(k)$ ,  $y_{up}(k)$ ,  $y_s^{\text{queue}}(k)$ ,  $y_p^{\text{queue}}(k)$ ,  $y_s^{\text{inflow}}(k)$ ,  $y_p^{\text{inflow}}(k)$ ),

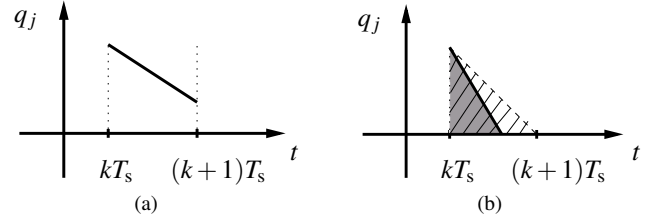


Fig. 5. Two situations for queue evolution.

$y_p^{\text{inflow}}(k)$ ,  $u_s(k)$ ,  $u_p(k)$ ,  $\delta_r(k)$ ,  $\delta_s^{\text{out}}(k)$ ,  $\delta_p^{\text{out}}(k)$ ) in one vector  $v(k)$ , we can express  $q_r(k+1)$  as an affine function of  $v(k)$ :

$$q_r(k+1) = av(k) + b$$

with a vector properly defined  $a$  and a scalar  $b$ , where  $v(k)$  satisfies a system of linear equations  $Cv(k) = e$  and linear inequalities  $Fv(k) \leq g$ , system which corresponds to the linear equations and constraints introduced above by the MILP transformations.

## 5. MODEL PREDICTIVE ROUTE CHOICE CONTROL

In this section we define the MPC optimization problem for both the nonlinear and the MILP case.

Recall that we want to assess the performance of MPC when using the original nonlinear model and when using the approximated MILP model. Therefore, the performance index should be linear or piecewise affine. In this paper we consider minimizing the total time spent in the queue for a network with  $S$  junctions. This performance index has been considered since the time spent in the queue  $\tau_j^{\text{queue}}$  at a junction  $S_j$ , with  $j \in \{1, 2, \dots, S\}$ , can be approximated to a linear one, see e.g. (van den Berg et al., 2008) for road traffic. However, note that the piecewise affine performance index used in (Tarău et al., 2009) can also be used after linearizing it using the MILP equivalences presented above.

When the length of a queue decreases during the sampling period  $T_s$ , we deal with the two situations sketched in Fig. 5 where at step  $k+1$  either the queue length  $q_j(k+1) > 0$  or where  $q_j$  becomes 0 before step  $k+1$ . Note that in this second case if the queue vanishes before step  $k+1$ , then  $q_j$  stays equal to 0 at least until  $t \geq (k+1)T_s$  since  $T_s$  is the sampling period, and both the demand and the control action are piecewise affine. Then, as in De Schutter (2002), we approximate the gray area under the curve of Fig. 5(b) with the dashed area  $A_j(k) = \frac{1}{2}(q_j(k) + q_j(k+1))T_s$ , a formula which holds also for Fig. 5(a). Then, the total time that the DCVs traveling toward junction  $S_j$  spend in the queue at junction  $S_j$  from the beginning of the simulation (step 0) until the last predicted time instant (step  $k+N-1$ ) is given by:

$$\tau_{j,k,N}^{\text{queue}} = \sum_{i=0}^{k+N-1} \frac{A_j(i)}{O_{\max}T_s} \quad (8)$$

Let  $J_{k,N} = \sum_{j=1}^S w_j \tau_{j,k,N}^{\text{queue}}$  denote the performance index at step  $k$ ,

for a prediction horizon  $N$ , where  $w_j$  is a weighting parameter that represents the penalization of DCVs waiting at junction  $S_j$ .

Then the nonlinear MPC optimization problem is defined as:

$$\min_{\mathbf{u}(k)} J_{k,N}(\mathbf{u}(k))$$

subject to  
the system dynamics  
operational constraints

where  $\mathbf{u}(k) = [u(k) u(k+1) \dots u(k+N-1)]^T$  with  $u(k) = [u_1(k) u_2(k) \dots u_S(k)]^T$ , while the time spent in the queue is determined via simulation.

In order to solve this mixed integer nonlinear optimization problem above one could use e.g. *genetic* algorithms, *simulated annealing*, or *tabu search* see e.g. Reeves and Rowe (2002), Dowsland (1993), and Glover and Laguna (1997).

Similarly, the linear MILP MPC optimization problem is defined as:

$$\min_{\mathbf{v}(k)} J_{k,N}(\mathbf{v}(k))$$
subject to  
MILP model  
operational constraints

where  $\mathbf{v}(k) = [v(k) v(k+1) \dots v(k+N-1)]^T$ , while the time spent in the queue is computed using (8).

To solve the MILP optimization problem one could use solvers such as CPLEX, Xpress-MP, GLPK, see e.g. (Atamtürk and Savelsbergh, 2005).

We expect that computing the route for each DCV in the network when solving the nonlinear optimization problem will give better performance than when solving the approximated MILP, but at the cost of much higher computational efforts. So, one could use MILP to compute a good initial point for the nonlinear optimization problem and this would reduce the computation time.

## 6. CASE STUDY

We consider as benchmark case study the network depicted in Fig. 6. This network consists of four loading stations and one unloading station connected via single direction track segments, where the free-flow travel time is provided for each link.

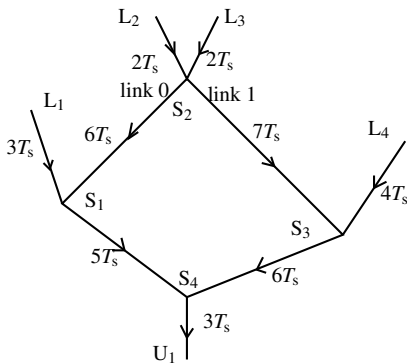


Fig. 6. Case study for a DCV-based baggage handling system.

Then the evolution of queue  $q_j$ , for  $j = 1, 2, 3, 4$  is given by:

$$q_j(k+1) = \max(0, f_j(k))$$

where  $f_j(k)$  is defined as follows:

$$f_1(k) = q_1(k) + (D_1(k-3) + (1 - u_2(k-6))O_2(k-6) - O_{\max})T_s$$

$$f_2(k) = q_2(k) + (D_2(k-2) + D_3(k-2) - O_{\max})T_s$$

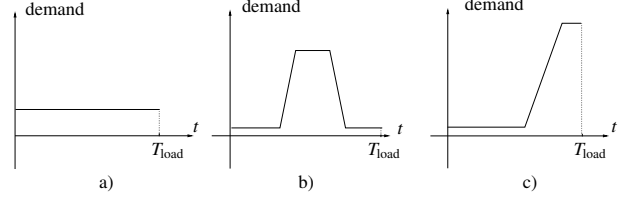


Fig. 7. Demand profile.

$$f_3(k) = q_3(k) + (D_4(k-4) + u_2(k-7)O_2(k-7) - O_{\max})T_s$$

$$f_4(k) = q_4(k) + (O_1(k-5) + O_3(k-6) - O_{\max})T_s$$

To compare the results we have considered 18 scenarios where 460 bags have to be handled for different initial states of the system, queues on different links, and different weighting parameters. For these scenarios we consider that the bags arrive at loading stations according to the three different classes of demand profiles sketched in Fig. 7, where  $T_{load}$  is the total loading time.

Let us now compare the results obtained when using the proposed predictive control method with different formulations of the optimization problem.

To solve the original mixed integer nonlinear MPC optimization problem we have chosen a *genetic* algorithm with multiple runs since simulations show that this optimization technique gives good performance with the shortest computation time. For solving the MILP optimization we have used the CPLEX solver. As prediction horizon we have considered  $N = 10$  for all MPC optimization problems. Note that we have chosen this horizon since simulations show that this value gives acceptable computational effort and performance index for all problem formulations.

Based on simulations we now compare, for the given scenarios, the results obtained for the proposed formulations of the optimization problem. The results of the simulations are reported in Fig. 8 where the total performance of the system is defined as

$$J = \sum_{j=1}^S \sum_{i \in \Lambda} w_j \tau_{i,j}^{\text{queue}}$$

with  $\Lambda$  the set of bags that wait at junction  $S_j$  during the simulation, and  $\tau_{i,j}^{\text{queue}}$  the real time that bag  $i$  spends in the queue at junction  $j$  while being transported to its corresponding end point. These results confirm that computing the route choice using the original nonlinear formulation for the MPC optimization problem gives typically better performance than using the MILP formulation, but at the cost of higher computational effort. However, it can be noted that for some scenarios, the use of MILP formulation results in better performance. This happens due to the fact that the prediction horizon is not sufficiently large. But, increasing the prediction horizon will result in increasing the computational effort even more.

Recall that we have used a *genetic* algorithm for solving the original nonlinear MPC optimization problem. But, genetic algorithms do not allow a given initial guess, therefore, to further reduce the computational effort, at each MPC step, we have solved the MILP optimization problem and we have used this solution as a feasible initial guess for computing a solution of the original nonlinear MPC problem with *simulated annealing*. As illustrated in Fig. 8, the results confirm that this last method offers a good trade-off between performance and computational effort.

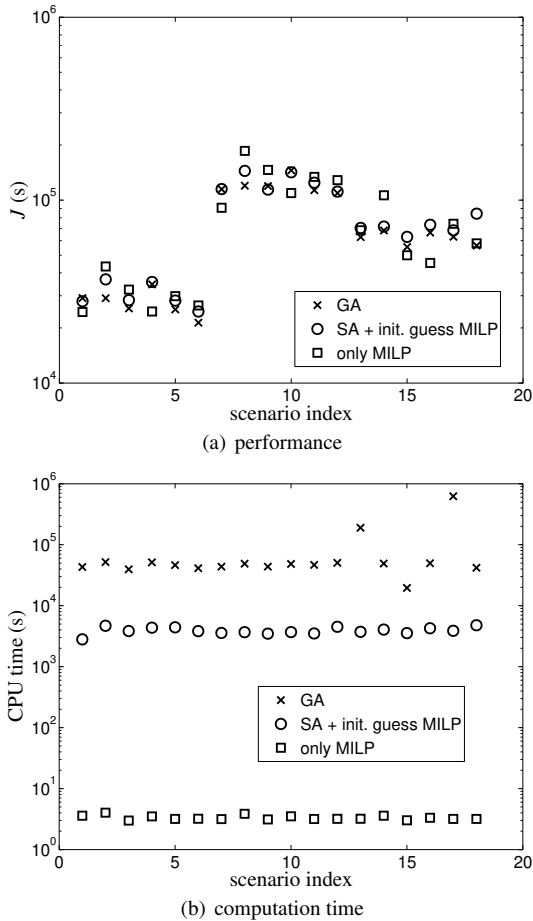


Fig. 8. Comparison of the proposed approaches.

## 7. CONCLUSIONS

In this paper we have considered the problem of efficiently computing routes for destination coded vehicle (DCV) that transport bags in an airport on a “mini” railway network. This is a nonlinear, nonconvex, mixed integer optimization problem, and very expensive to solve in terms of computational effort. Therefore, we have used an alternative approach for reducing the complexity of the computations by simplifying and approximating the nonlinear optimization problem by a mixed integer linear programming (MILP) problem. The advantage is that for MILP optimization problems the global optimal solution can be efficiently computed with available solvers. These two formulations of the optimization problem have been used to compute the route of DCVs using model predictive control (MPC) for a benchmark case study.

Simulation results confirm that computing the route choice using the original nonlinear formulation for the MPC optimization problem gives usually better performance than using the MILP formulation, but at the cost of significantly higher computational efforts. To reduce the computation time while obtaining good results, one can solve the original MPC optimization problem, but using at each step the local solution of the corresponding MILP formulation as initial guess.

In future work we will apply this method to more complex case studies where we will also consider controlling the switch into junctions.

## ACKNOWLEDGEMENTS

Research supported by the STW-VIDI project “Multi-Agent Control of Large-Scale Hybrid Systems”, the BSIK project “Next Generation Infrastructures”, the Transport Research Centre Delft, the Delft Research Center Next Generation Infrastructures, and the European 7th framework STREP project “Hierarchical and Distributed Model Predictive Control”.

## REFERENCES

- Atamtürk, A. and Savelsbergh, M. (2005). Integer-programming software systems. *Annals of Operations Research*, 140(1), 67–124.
- Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35, 407–427.
- de Neufville, R. (1994). The baggage system at Denver: Prospects and lessons. *Journal of Air Transport Management*, 1(4), 229–236.
- De Schutter, B. (2002). Optimizing acyclic traffic signal switching sequences through an extended linear complementarity problem formulation. *European Journal of Operational Research*, 139(2), 400–415.
- Dowsland, K. (1993). Simulated annealing. In C. Reeves (ed.), *Modern heuristic techniques for combinatorial problems*, chapter 2, 20–69. John Wiley & Sons, Inc., New York, USA.
- Fay, A. (2005). Decentralized control strategies for transportation systems. In *Proceedings of the International Conference on Control and Automation*, 898–903. Budapest, Hungary.
- Gang, H., Shang, J., and Vargas, L. (1996). A neural network model for the free-ranging AGV route-planning problem. *Journal of Intelligent Manufacturing*, 7(3), 217–227.
- Glover, F. and Laguna, F. (1997). *Tabu Search*. Kluwer Academic Publishers, Norwell, Massachusetts, USA.
- Hallenborg, K. and Demazeau, Y. (2006). Dynamical control in large-scale material handling systems through agent technology. In *Proceedings of the International Conference on Intelligent Agent Technology*, 637–645. Hong Kong, China.
- Kaufman, D., Nonis, J., and Smith, R. (1998). A mixed integer linear programming model for dynamic route guidance. *Transportation Research Part B: Methodological*, 32(6), 431–440.
- Lewis, F. (1986). *Optimal Control*. John Wiley & Sons, Inc., New York, USA.
- Maciejowski, J. (2002). *Predictive Control with Constraints*. Prentice Hall, Harlow, UK.
- Reeves, C. and Rowe, J. (2002). *Genetic Algorithms - Principles and Perspectives: A Guide to GA Theory*. Kluwer Academic Publishers, Norwell, Massachusetts, USA.
- Taráu, A., De Schutter, B., and Hellendoorn, J. (2008). Travel time control of destination coded vehicles in baggage handling systems. In *In the IEEE International Conference on Control Applications*, 293–298. San Antonio, Texas, USA.
- Taráu, A., De Schutter, B., and Hellendoorn, J. (2009). Route choice control of automated baggage handling systems. In *Proceedings of the 88th Annual Meeting of the Transportation Research Board*. Washington DC, USA.
- van den Berg, M., De Schutter, B., Hellendoorn, J., and Hegyi, A. (2008). Influencing route choice in traffic networks: A model predictive control approach based on mixed-integer linear programming. In *Proceedings of the 17th IEEE International Conference on Control Applications*, 299–304. San Antonio, Texas.