Elektrotehniški vestnik 70(4): 208–214, 2003
Electrotechnical Review, Ljubljana, Slovenija

Hybrid Modeling and Optimal Control of an Asphalt Base Process

Boštjan Potočnik [1], Alberto Bemporad [2], Fabio Danilo Torrisi [3], Gašper Mušič [1], Borut Zupančič [1]

[1] *Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, SI-1000 Ljubljana, Slovenia*
[2] *Dip. Ingegneria dell'Informazione, Università di Siena, Via Roma 56, I-53100 Siena, Italy*
[3] *Automatic Control Laboratory, ETH - Swiss Federal Institute of Technology, CH-8092 Zürich, Switzerland*
E-mail: bostjan.potocnik@fe.uni-lj.si

Abstract. This paper addresses the issue of an optimal selection of the production and delivery plan for an asphalt base process. The process is modeled as a DHA (Discrete Hybrid Automaton) using the high level modeling language HYSDEL (HYbrid System DEscription Language) that allows converting the DHA model into an MLD (Mixed Logical Dynamical) model. The proposed solution applies to a class of optimal control problems where the goal is to minimize the total completion time. The solution algorithm, which takes into account a model of a hybrid system described as an MLD system, is based on the reachability analysis. The algorithm abstracts the behavior of the hybrid system into a "tree of evolution", where nodes of the tree represent reachable states of the system and branches connect two nodes if a transition exists between the corresponding states. To each node the cost function value is associated and, based on this value, the tree exploration is driven, searching for the optimal control profile.

Key words: hybrid systems, optimal control, reachability analysis, branch-and-bound methods

Hibridno modeliranje in optimalno vodenje procesa asfaltne baze

Povzetek. V delu obravnavamo problem optimalne izbire proizvodnje in dobave asfalta, ki je delovni proces asfaltne baze. Proces smo modelirali kot DHA (Discrete Hybrid Automaton), pri čemer smo uporabili modelirni jezik HYSDEL (HYbrid System DEscription Language), ki omogoča pretvorbo modela DHA v zapis MLD (Mixed Logical Dynamical). Predstavljena rešitev se nanaša na vrsto optimalnega vodenja, kjer je cilj minimizirati čas proizvodnje oziroma čimprej doseči zastavljen cilj. Algoritem iskanja optimalne rešitve temelji na zapisu MLD v povezavi z idejo analize dosegljivosti, ki povzame delovanje celotnega procesa v obliki "problemskega drevesa". Problemsko drevo je sestavljeno iz vozlišč, ki predstavljajo stanja sistema, in vej, ki predstavljajo povezave oziroma prehajanja med stanji procesa. Vsakemu vozlišču priredimo vrednost kriterijske funkcije, na podlagi katere poiščemo in določimo optimalno vodenje.

Ključne besede: hibridni sistemi, optimalno vodenje, analiza dosegljivosti, metode razveji in omeji

1 Introduction

The demand for increased levels of automation has given rise to the development of larger and more complex systems. New methods and advanced technologies enable automation of industrial processes to outgrow the basic low-level control functions. Higher levels usually include discrete event dynamics, i.e. *event-driven* dynamics, while the traditional control approaches are mostly dealing with continuous dynamics, i.e. *time-driven* dynamics, on lower levels. Hybrid systems combine *event-driven* and *time-driven* dynamics.

Mathematical models represent the basis to any system analysis such as simulation, control, verification, etc. In order to efficiently define the system behavior the model should not be too complicated. Also, if it were to simple, it would not be close enough to the real behavior of an observed process. We modeled a hybrid system as a *discrete hybrid automaton* (DHA) using the modeling language HYSDEL (HYbrid System DEscription Language) [15]. Employing the associated compiler, the DHA model can be translated to different modeling frameworks, such as *mixed logical dynamical* (MLD), *piecewise affine* (PWA), *linear complementarity* (LC), *extended linear complementarity* (ELC) or *max-min-plus-scaling* (MMPS) systems [12]. In this paper the MLD modeling framework presented in [6] will be adopted as it is most suitable to solve optimal control problems. Indeed, several control procedures, based on the MLD description of a process, were proposed in the literature. A model predictive control technique is presented in [6]. It is able to stabilize an MLD system on a desired ref-

Received 24 October 2002
Accepted 25 March 2003

erence trajectory, where on-line optimization procedures are solved through *mixed integer quadratic programming* (MIQP) [5]. A verification approach for hybrid systems is presented in [7].

Optimal control laws for hybrid systems have been widely investigated in recent years and many results can be found in the control science literature. Optimal control of hybrid systems in manufacturing is addressed in [1, 9], where the authors combine time-driven and event-driven methodologies to solve optimal control problems. An algorithm to optimize switching sequences for a class of switched linear problems is presented in [13], where the algorithm searches for solutions arbitrary close to the optimal ones. A similar problem is addressed in [2], where the potential for numerical optimization procedures to make optimal sequencing decisions in hybrid dynamical systems is explored. A computational approach based on ideas from dynamic programming and convex optimization is presented in [11]. Piecewise linear quadratic optimal control is addressed in [14], where the use of piecewise quadratic cost functions is extended from the stability analysis of piecewise linear systems. Optimal control based on a reachability analysis is addressed in [4] and is here extended to hybrid systems with only discrete inputs.

The paper is organized as follows. In Section 2 we address the *discrete hybrid automata* and *mixed logical dynamical* modeling frameworks. The problem formulation and proposed solution is addressed in Section 3. The proposed algorithm is applied to the Asphalt Base Process and is discussed in Section 4.3.

2 Discrete Hybrid Automata and Mixed Logical Dynamical Systems

Hybrid systems are a combination of logic, finite state machines, continuous dynamic systems and constraints. The interaction between continuous and discrete/logic dynamics is shown in Fig. 1, where both parts are connected through *A/L* (analog to logic) and *L/A* interfaces [8, 15]. The system shown in Fig. 1 can be modeled as a *discrete hybrid automaton* (DHA). The DHA model of a hybrid system can be specified by using the modeling language HYSDEL (*HYbrid System DEscription Language*). The associated HYSDEL compiler translates the DHA model into an equivalent *mixed logical dynamical* (MLD) form. The MLD form can be later used to obtain other equivalent model representations [12]. A detailed procedure for transforming the DHA model into an equivalent MLD model is described in [15].

The HYSDEL list is composed of two parts: INTERFACE declaring all the variables and parameters, and IMPLEMENTATION consisting of specialized sections in which relations between the variables are defined. These specialized sections are: AD section defining the Boolean variables from the continuous ones, LOGIC section specifying arbitrary functions of the Boolean variables, DA section defining continuous variables from the Boolean ones, CONTINUOUS section describing the linear dynamics expressed as difference equations, LINEAR section defining continuous variables as an affine function of continuous variables, AUTOMATA section specifying the state transition equations of the finite states machine as a Boolean functions, OUTPUT section specifying static linear and logic relations for the output vector, and MUST section listing all the constraints on the continuous and Boolean variables. For a more detailed description of the HYSDEL syntax the reader is referred to [15].

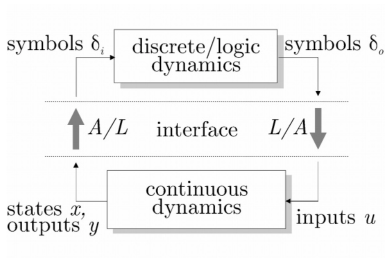

Figure 1. Hybrid control system - discrete and continuous dynamics interact through interfaces

2.1 MLD System

Once the system is modeled in HYSDEL, the companion compiler generates the equivalent MLD model [6] of the form

$$x(k+1) = Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k) \quad (1)$$
$$y(k) = Cx(k) + D_1u(k) + D_2\delta(k) + D_3z(k) \quad (2)$$
$$E_2\delta(k) + E_3z(k) \le E_1u(k) + E_4x(k) + E_5\ , \quad (3)$$

where $x = [x_c, x_l]' \in \mathbb{R}^{n_c} \times \{0,1\}^{n_l}$ is the vector of continuous and logic states, $u = [u_c, u_l]' \in \mathbb{R}^{m_c} \times \{0,1\}^{m_l}$ are the inputs, $y = [y_c, y_l]' \in \mathbb{R}^{p_c} \times \{0,1\}^{p_l}$ are the outputs, $\delta \in \{0,1\}^{r_l}$, $z \in \mathbb{R}^{r_c}$ are the auxiliary logic and continuous variables, respectively, and A, B_1, B_2, B_3, C, D_1, D_2, D_3, $E_1, \ldots, E_5$ are the matrices of suitable dimensions. Inequalities (3) can contain also additional constraints over the variables (states, inputs and auxiliary variables). This permits to include additional constraints and incorporate heuristic rules in the model.

Using the current state $x(k)$ and input $u(k)$, the time evolution of (1–3) is determined by solving $\delta(k)$ and $z(k)$ from (3), and then updating $x(k+1)$ and $y(k)$ from equations (1) and (2). The MLD system (1–3) is assumed to be completely well-posed if for a given state $x(k)$ and input $u(k)$ the inequalities (3) have a unique solution for $\delta(k)$ and $z(k)$. A simple algorithm to test well-posedness is given in [6].

3 Optimal Control of Hybrid Systems

In [4] and [6] the authors present procedures for optimal control of hybrid processes described in the MLD form. Optimal control amounts to finding the control sequence $U_0^{k_{\mathrm{fin}}-1} = \{u(0), \ldots, u(k_{\mathrm{fin}}-1)\}$ which transfers the initial state x_0 to the final state x_{fin} in the finite time $T = k_{\mathrm{fin}} \cdot T_s$ (T_s is the sampling time) while minimizing a certain performance index.

3.1 A Class of Optimal Control Problems

Because of the problem of the Asphalt Base Process at hand, the focus will be on the optimal control for hybrid systems that are described in the MLD form with only discrete (logic) inputs and where the goal is to minimize the production time. The problem will be solved by extending the existing tools based on the reachability analysis [4]. Needles to say, this kind of optimal control problems are very complex to solve and are quite frequent in industry.

3.2 Complexity of the Problem

The solution to the posed optimal control problem is the final time $T = k_{\mathrm{fin}} \cdot T_s$ and the optimal control sequence $U_0^{k_{\mathrm{fin}}-1} = \{u(0), \ldots, u(k_{\mathrm{fin}}-1)\}$, where $u(k)$ represents the input to the system at step k. If the system has m_l discrete inputs and no continuous inputs ($u(k) \in \{0,1\}^{m_l}$ and $U_0^{k_{\mathrm{fin}}-1} \in \{0,1\}^{m_l \cdot k_{\mathrm{fin}}}$), there are $2^{m_l \cdot k_{\mathrm{fin}}}$ possible combinations for $U_0^{k_{\mathrm{fin}}-1}$. Hence, the optimization problem is NP-hard and the computational time required to solve the problem grows exponentially with the problem size (a^n, for $a > 1$), so that any enumeration method would be impractical.

3.3 Optimization Based on the Reachability Analysis

In general, not all the combinations of inputs are feasible because of the constraints (3). One approach to rule out infeasible inputs is to use the reachability analysis. The idea for hybrid systems with continuous inputs presented in [4] is extended here to hybrid systems with discrete inputs.

Through the reachability analysis it is possible to extract the reachable states of the system. Enumerating all of them would not be effective as many of them will be far away from the optimal trajectory. Therefore, it is reasonable to combine the reachability analysis with procedures that can detect a reachable state not leading to the optimal solution and remove it from the exploration procedure. More precisely, reachable states detected not to lead to a better (optimal) solution are removed from the exploration of the state space. The whole procedure is a kind of the *branch and bound* strategy and involves generation of a "tree of evolution", as will be described later. By searching reachable states, we branch the evolution tree and by removing non-optimal ones we bound it.

3.4 The Reachability Analysis

Let $x(k)$ be the state at step k. The reachability analysis computes all the possible states $x^i(k+1)$ which are reachable at the next time step ($i \in \{1, 2, ...\}$ is an index marking reachable states). If the system has m_l discrete inputs, then 2^{m_l} possible next states may exist. However, because of the constraints (3), only a smaller number of states can actually be reached. The reachable states $x^i(k+1)$ are computed by applying the state $x(k)$ and all possible inputs $u_b(k)$ at step k to the MLD model (1–3) of a hybrid system.

3.5 Tree of Evolution

A tree of evolution (see Fig. 2) abstracts the possible evolution of the system over a horizon of k_{fin} steps. The nodes of the tree represent reachable states and branches connect two nodes if a transition exists between corresponding states. For a given root node $\mathcal{V}_0$, representing the initial state $x^0 = x(0)$, reachable states are computed and inserted into the tree as nodes $\mathcal{V}_i$. A cost value J_i is associated to each new node. A new node is selected based on the associated cost value J_i and new reachable states are computed [4]. More about the cost function and the node selection criteria will be presented in the following section. The construction of the tree of evolution proceeds according to the depth first strategy until one of the following conditions has been met:

- The step horizon limit $k_{\max}$ has been reached.
- The value of the cost function at the current node is greater than the current optimal one ($J_i \geq J_{\mathrm{opt}}$, where initially $J_{\mathrm{opt}} = \infty$).
- The final state has been reached ($x(k) = x_{\mathrm{fin}}$).

The node that satisfies one of the above conditions is labeled as explored. If the node satisfies the first or the third condition, the associated value of the cost function J_i becomes the current optimal one ($J_{\mathrm{opt}} = J_i$), the step instance k becomes the current optimal one ($k_{\mathrm{opt}} = k$) and the control sequence $U_0^{k_{\mathrm{fin}}-1}$ which leads from the initial node $\mathcal{V}_0$ to the current node $\mathcal{V}_i$ becomes the current optimizer. The exploration continues until there are no more unexplored nodes in the tree and the temporary control sequence $U_0^{k_{\mathrm{fin}}-1}$ becomes the optimal one.

3.6 Cost Function and Node Selection Criterion

The cost function and the node selection criterion have a great influence on the size of the tree of evolution and,

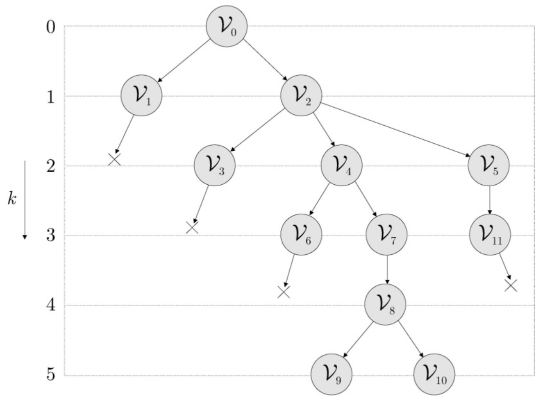

Figure 2. Tree of evolution

indirectly, on the time efficiency of the optimization algorithm. The best node selection criterion is to propagate the tree of evolution in a direction that minimizes the value of the cost function. At the same time the cost value J_i associated with a node is used to detect nodes not leading to an optimal solution thus preventing an unnecessary growth of the tree of evolution. To achieve that, the cost function must have certain properties that are described below.

As the goal is to minimize the total production time we choose the following cost function:

$$J_i(x, k) = h(x) + g(k), \tag{4}$$

where $h(x)$ presents the "distance measure" to the final state x_{fin}, with the following properties:

$$h(x_{\text{fin}}) = 0 \tag{5}$$
$$h(x(k+1)) - h(x(k)) \leq 0. \tag{6}$$

$g(k)$ is the function that gives a "measure" of elapsed time from the start, with the following property:

$$g(k+1) - g(k) > 0. \tag{7}$$

The cost value J_i decreases with the function $h(x)$ and increases with the function $g(k)$. This property can be used to detect nodes which do not lead to an optimal solution at step instance $k < k_{\text{opt}}$ (k_{opt} is time instance of the optimizer) by comparing $J_i(k)$ to $J_{\text{opt}}(k_{\text{opt}})$. When the cost value $J_i(k) \geq J_{\text{opt}}(k_{\text{opt}})$, we want to ensure that by continuing the exploration from this node no better solution than the current one can be found. To achieve that, the cost function (4) has to be monotonically increasing, i.e. in the next steps the cost value J_i can only increase. To this end, we impose

$$J(x(k+1), k+1) - J(x(k), k) \geq 0, \text{ i.e.} \tag{8}$$
$$\big(h(x(k+1)) + g(k+1)\big) - \big(h(x(k)) + g(k)\big) \geq 0. \tag{9}$$

Reaching the final state x_{fin} can be detected using cost function (4). Through equation (5) it can be easily noticed that the cost value at x_{fin} is

$$J_{\text{fin}} = g(k_{\text{fin}}). \tag{10}$$

4 A Case Study: Asphalt Base Process

The proposed approach was applied to a model of an Asphalt Base Process. The model tries to take into account all the characteristics of the plant, i.e. the process of producing and delivering asphalt. It is relatively small compared to the real-scale plants, but nonetheless can pose complex control tasks.

4.1 Description of the Plant

The asphalt base consists of one asphalt preparation reactor in which two different types of asphalt can be produced: the "rough" and the "fine" type. The reactor has one loading place able to load just one truck at a time. The loading and unloading time depends on the truck capacity. The delivery is done by four trucks of different load capacities. The traveling speeds of the truck differ and depend on their load. The service of the transport takes place on two locations with different travel distances from the base. Therefore the delivery time differs with regard to the location involved. It is also taken into consideration that the "rough" asphalt is placed before the "fine" one. The data of the process are given in Table 1.

Reactor	Capacity	100 tons
	Prep. time - "rough"	60 min.
	Prep. time - "fine"	80 min.
Truck 1	Load capacity	5 tons
	Loading time	1 min.
	Unloading time	2 min.
	Speed - empty	75 km/h
	Speed - full	60 km/h
Truck 2 and 3	Load capacity	10 tons
	Loading time	2 min.
	Unloading time	3 min.
	Speed - empty	60 km/h
	Speed - full	50 km/h
Truck 4	Load capacity	20 tons
	Loading time	3 min.
	Unloading time	4 min.
	Speed - empty	50 km/h
	Speed - full	43 km/h

Table 1. Process data

By taking into consideration that the distances to Location 1 and Location 2 are 40 and 50 kilometers, respectively, and that the delivery is done first for the "rough" and then for the "fine" asphalt, then the process of producing and delivering asphalt can be presented with Figure 3.

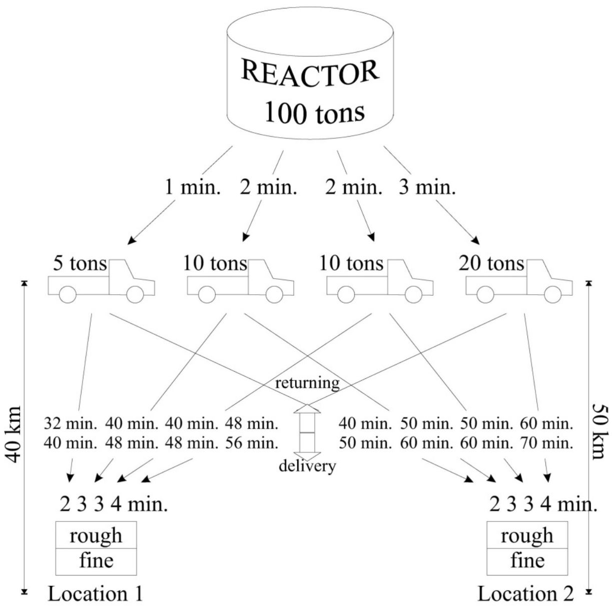

Figure 3. Process of producing and delivering asphalt

By observing Figure 3, one might come to the conclusion that the process is simple and therefore unpretentious for modeling. This is not true because the system has to take into account different orders, i.e. a quantities, types and locations. It must properly execute situations like: deliver 7 tons of asphalt to Location 1, i.e. take a truck with at least 10 tons capacity and load it with 7 tons. Taking into account all the details, the system becomes quite complex and therefore hard to model.

4.2 The DHA and MLD Models

The system is modeled as a DHA system, i.e. described in the HYSDEL language and then transformed into the MLD form using the associated HYSDEL compiler. The HYSDEL code for the Asphalt Base Process can be found on the web site `http://msc.fe.uni-lj.si/potocnik`. By taking into account sample time $T_s = 1$ minute, the HYSDEL tool generates the equivalent MLD form (1–3). The dimensions of the corresponding variables are: $x(k) \in \mathbb{R}^{20} \times \{0,1\}^{30}$, $u(k) \in \{0,1\}^{10}$, $\delta(k) \in \{0,1\}^{163}$, and $z(k) \in \mathbb{R}^{77}$. Matrices A, B_1, B_2, B_3, C, D_1, D_2, D_3 have suitable dimensions. Matrices E_1 to E_5 define 1084 inequalities. Output $y(k)$ is omitted, because all the outputs are actually the states of the system.

4.3 Control of the Asphalt Base Process

Problem formulation:

For a given initial condition, control the asphalt production and its delivery to minimize the total time.

The order for the asphalt delivery, which consists of the quantity, the type, and the location, presents the initial condition to the system:

- Location 1: 90 tons - "rough"; 40 tons - "fine".
- Location 2: 110 tons - "rough"; 60 tons - "fine".

The degrees of freedom are:

- Type of asphalt to be prepared in the reactor.
- Selection of the truck and location.

The solution to the control problem is a control sequence U_0^{T-1}. At each time 10 inputs can influence the system $u(k) \in \{0,1\}^{10}$. To estimate the minimal production time to $T_{\text{est}} = 800$ minutes, although such a time may not be feasible, then $U_0^{k_{\text{fin}}-1} \in \{0,1\}^{10\cdot 800}$ and because all the inputs are logical, 2^{8000} possible combinations of the solution vector $U_0^{k_{\text{fin}}-1}$ exist and searching the solution through all the combinations is practically impossible.

The goal is to minimize the total time for the given initial conditions (orders). According to the cost function and node selection criterion introduced earlier, we use the following cost function

$$J_i = \begin{array}{l}(R + F + 0.9Re + 0.8(R_{L1} + F_{L1} + \\ + R_{L2} + F_{L2}))f + k\end{array} , \quad (11)$$

where R and F represent the remainder of the sum of all the "rough" and "fine" orders, Re represents the quantity of an asphalt in the reactor, R_{L1} and R_{L2} represent the remainder of the "rough" type asphalt needed to deliver to Locations 1 and 2, similarly F_{L1} and F_{L2}, k is the time step and f is the factor whose properties will be explained later. The order is completed when R=F=Re=R_{L1}=F_{L1}=R_{L2}=F_{L2}= 0, hence the cost function value at the feasible solution is

$$J_i = k \; . \quad (12)$$

According to (8–9), the cost function (11) has to be monotonically increasing, i.e.

$$
\begin{aligned}
&\Big(\big(R(k{+}1){+}F(k{+}1){+}0.9Re(k{+}1){+}0.8(R_{L1}(k{+}1){+}\\
&{+}F_{L1}(k{+}1){+}R_{L2}(k{+}1){+}F_{L2}(k{+}1))\big)f{+}(k{+}1)\Big){-}\\
&{-}\Big(\big(R(k){+}F(k){+}0.9Re(k){+}0.8(R_{L1}(k){+}\\
&{+}F_{L1}(k){+}R_{L2}(){+}F_{L2}(k))\big)f{+}k\Big) =\\
&= (\Delta R + \Delta F + 0.9\Delta Re + 0.8(\Delta R_{L1} + \Delta F_{L1}+\\
&+\Delta R_{L2} + \Delta F_{L2}))f + \Delta k \geq 0,
\end{aligned} \tag{13}
$$

where $\Delta R = R(k{+}1){-}R(k)$, ΔF, ΔRe, ΔR_{L1}, ΔF_{L1}, ΔR_{L2}, ΔF_{L2} and Δk are defined accordingly. Taking into account the maximum influence of the system change to the cost function value, the parameter f must satisfy the condition

$$
\begin{aligned}
f &\leq \frac{\Delta k}{-\Big(\Delta R{+}\Delta F{+}0.9\Delta Re{+}0.8(\Delta R_{L1}{+}\Delta F_{L1}{+}\Delta R_{L2}{+}\Delta F_{L2})\Big)}\\
f &\leq \frac{1}{-\Big(0{+}0{-}0.9 20{+}0.8({-}20{+}0{+}0{+}0)\Big)} = \frac{1}{34}.
\end{aligned} \tag{14}
$$

Regarding the node selection criterion, it is reasonable to choose the node which leads to the best (current) optimal solution, i.e. the node with the smallest associated cost function value J_i at step k.

4.4 Results

Because of the complexity of the optimal control problem for the Asphalt Base Process analyzed in Section 4.3, we decided to search for a suboptimal solution for the complete optimal control problem. Searching for an optimal one would have taken too much time as the complexity of the optimization problem grows exponentially with the time horizon.

We added a constraint over the tree size of 1500 nodes to the proposed algorithm, as we expected the suboptimal solution to be within the selected interval, and set the parameter f to $1/34$. The solution is presented in Figures 4 and 5. Figure 4 shows the dynamics of completing the orders for Locations 1 and 2, while Figure 5 shows the loads of the trucks and phases of the transport (loading, delivering, unloading, returning). The first and in this case the final suboptimal solution was obtained in 9 minutes and 46 seconds. The computational times refer to the MATLAB implementation running on a Pentium III 667 MHz machine. We remark that the suboptimal solution is feasible and could also be optimal, but the algorithm would need much more time to prove optimality.

The obtained results were also compared to the results obtained in [10] (see Table 1), where the MS Project and the Preactor were used to find the solution. The MS Project is not capable, as a tool, to find a solution, but provides a general overview over the problem and hence the solution is the product of the author. On the contrary the tool Preactor is designed to solve such problems in a certain framework.

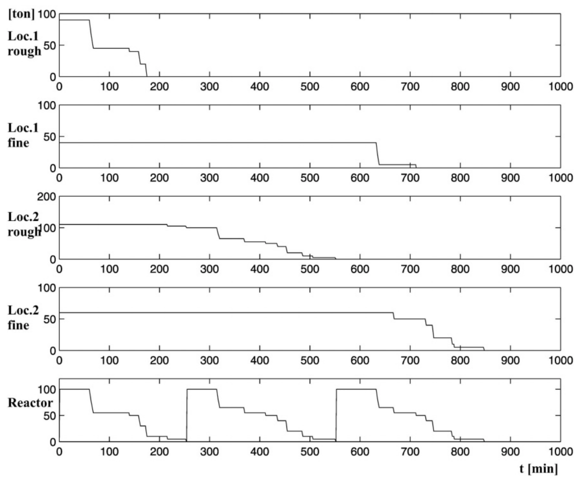

Figure 4. The dynamics of completing the orders

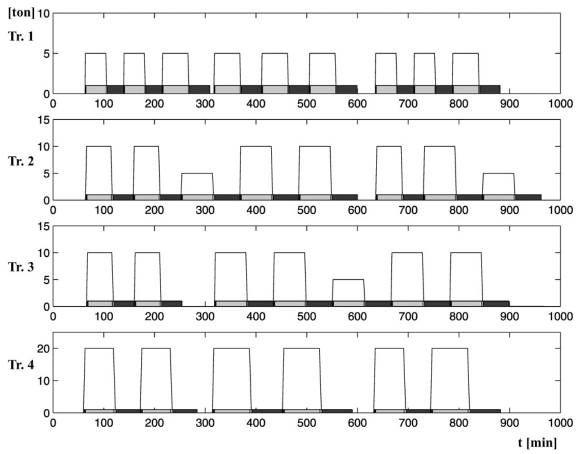

Figure 5. Time and load characteristics of the trucks

Algorithm	16h 2min
Preactor	16h 33min
MS Project	18h 26min

Table 2. Comparisons of the results

5 Conclusions

Mathematical models are the basis for simulation, control, analysis, etc. The modeling language HYSDEL allows to transform the DHA models into the MLD ones and to solve optimal control problems.

A class of optimal control problems was addressed with the goal to minimize the total production time. The problem was solved by combining the reachability analysis and the branch and bound technique. The main advantage of the approach is in cutting down the "tree of evolution" from both sides, top and bottom. Here, it has to be pointed out that the proposed algorithm is not limited to hybrid systems, where the dynamics is of an integral type. It can handle also more complex dynamics where the continuous behavior of the system is not known in advance. The suboptimal approach, based on additional knowledge of the process, was applied to the Asphalt Base Process and gives satisfactory results within acceptable time.

6 References

[1] Antsaklis P.J. (ed.) (2000). Special issue on hybrid systems: Theory and applications. *Proceedings of the IEEE*, Vol. 88, No. 7.

[2] Barton, P.I., Banga, J.R., & Galán, S., (2000). Optimization of hybrid discrete/continuous dynamic systems. *Computers and Chemical Engineering*, Vol. 24, pp. 2171–2182.

[3] Bemporad, A., Ferrari-Trecate, G., Mignone, D., Morari, M., & Torrisi, F.D., (1999). Model predictive control - Ideas for the next generation. In *Proceedings of the European Control Conference*. Karlsruhe, Germany.

[4] Bemporad, A., Giovanardi, L., & Torrisi, F.D., (2000). Performance Driven Reachability Analysis for Optimal Scheduling and Control of Hybrid Systems. In *Proc. 39th IEEE Conference on Decision and Control*, pp. 969–974, Sydney, Australia.

[5] Bemporad, A., & Mignone, D., (2000). MIQP.M: A Matlab function for solving mixed integer quadratic programs, ETH Zurich, code available at `http://control.ethz.ch/~hybrid/miqp`

[6] Bemporad, A., & Morari, M., (1999). Control of systems integrating logic, dynamic, and constraints. *Automatica*, Vol. 35, No. 3, pp. 407–427.

[7] Bemporad, A., Torrisi, F.D., & Morari, M., (2001). Discrete-time hybrid modeling and verification of the batch evaporator process benchmark. *European Journal of Control*, Vol. 7, No. 4, pp. 382–399.

[8] Branicky, M.S., Borkar V.S. & Mitter, S.K., (1998). A unified framework for hybrid control: Model and optimal control theory. *IEEE Trans. on Automatic Control*, Vol. 43, No. 1, pp. 31–45.

[9] Gokbayrak, K., & Cassandras, C.G., (1999). A hierarchical decomposition method for optimal control of hybrid systems. In *Proc. 38th IEEE Conf. on Decision and Control*, pp. 1816-1821, Phoenix, Arizona, USA.

[10] Gradišar, D., (2001). Uporaba orodij MS Project in Preactor za vodenje na višjih nivojih. *diplomsko delo*, Fakulteta za elektrotehniko, Univerza v Ljubljani, Slovenija.

[11] Hedlund, S., & Rantzer, A., (1999). Optimal control of hybrid systems. In *Proc. 38th IEEE Conf. on Decision and Control*, pp. 3972-3976, Phoenix, Arizona USA.

[12] Heemels, W.P.M.H, De Schutter, B., & Bemporad, A., (2001). Equivalence of hybrid dynamical models. *Automatica* , Vol. 37, No. 7, pp. 1085–1091.

[13] Lincoln, B., & Rantzer, A., (2001). Optimizing linear system switching. In *Proc. 40th IEEE Conf. on Decision and Control*, pp. 2063-2068, Orlando, Florida, USA.

[14] Rantzer, A., & Johansson, M., (2000). Piecewise linear quadratic optimal control. *IEEE Trans. on Automatic Control*, Vol. 45, No. 4, pp. 629–637.

[15] Torrisi, F.D., & Bemporad, A., (2002). HYSDEL - A Tool for Generating Computational Hybrid Models, *IEEE Transactions on Control Systems Technology*, in press.

Boštjan Potočnik received the B.Sc. and M.Sc. degrees in 1998 and 2001, respectively, from the Faculty of Electrical Engineering, University of Ljubljana, Slovenia. Since 2001 he has been working towards his Ph.D. degree at the same faculty. His work is focused on the field of hybrid systems.

Alberto Bemporad received the M.Sc. degree in Electrical Engineering in 1993 and the Ph.D. in Control Engineering in 1997 from the University of Florence, Italy. He spent the academic year 1996/97 at the Center for Robotics and Automation, Dept. Systems Science & Mathematics, Washington University, St. Louis, as a visiting researcher. In 1997-1999, he held a postdoctoral position at the Automatic Control Lab, ETH, Zurich, Switzerland, where he collaborated as a senior researcher in 2000-2002. Since 1999, he is assistant professor at the University of Siena, Italy. He received the IEEE Center and South Italy section "G. Barzilai" and the AEI (Italian Electrical Association) "R. Mariani" awards. He has published papers in the area of hybrid systems, model predictive control, computational geometry, and robotics. He is coauthoring the new version of the Model Predictive Control Toolbox (The Mathworks, Inc.). He is an Associate Editor of the IEEE Transactions on Automatic Control, and Chair of the IEEE Control Systems Society Technical Committee on Hybrid Systems.

Fabio Danilo Torrisi received the M.Sc. degree in Computer Engineering in 1999 from the University of Florence, Italy. Since 1999 he is a Ph.D. student at the Automatic Control Lab, ETH, Zurich, Switzerland. His research interest is in the area of hybrid systems and computational geometry.

Gašper Mušič received B. Sc., M. Sc. and Ph. D. degrees in electrical engineering from the University of Ljubljana, Slovenia in 1992, 1995 and 1998, respectively. He is holding a position of an assistant at the Faculty of Electrical Engineering, University of Ljubljana. His research interests are in discrete-event and hybrid dynamical systems, supervisory control and higher control levels, and applications in industrial process control. He has published five papers in international and national scientific journals and over twenty conference papers and is also a co-author of a student book. In 1998 he received the Vidmar award for his pedagogical work. In 1999 he received the Bedjanič award for his Ph.D. thesis.

Borut Zupančič received B.Sc., M.Sc. and Ph.D. degrees in electrical engineering from the University of Ljubljana, Slovenia, in 1977, 1979 and 1989, respectively. He is a professor at the Faculty of Electrical Engineering, University of Ljubljana. His major research interest is modeling and simulation of dynamical systems and computer-aided control system design. In 1989 he received the award of Slovenian Ministry for Research and Technology for the work in the field of computer-aided design of control systems. He was the president of Slovene Society for Simulation and Modeling. At present he is a member in the editorial board of Simulation News Europe, member of Eurosim board and vice-dean of the Faculty of Electrical Engineering, University of Ljubljana.