Optimization of Discrete-Continuous Dynamic Systems Based on Disjunctive Programming

Jan Oldenburg* and Wolfgang Marquardt**
Lehrstuhl für Prozesstechnik, RWTH Aachen University, D-52056 Aachen, Germany

In this contribution, a novel approach for the modeling and optimization of discrete-continuous dynamic systems based on a disjunctive problem formulation is proposed. It will be shown that a disjunctive model representation, which constitutes an alternative to mixed-integer model formulations, provides a very flexible and intuitive way to formulate discrete-continuous dynamic optimization problems. Moreover, the structure and properties of the disjunctive process models can be exploited for an efficient and robust numerical solution by applying generalized disjunctive programming techniques. The proposed modeling and optimization approach will be illustrated by means of an optimal control problem that embeds a linear discrete-continuous dynamic system.

1 Introduction

Mathematical modeling, simulation, and optimization enable the improvement of chemical process design and operation in a very systematic way. Both, the analysis and optimization-based synthesis of process systems rely on mathematical models that can predict the process behavior with a level of detail tailored to the needs of a specific application. As soon as a process is operated in a transient manner, dynamic process models are required that are capable of reflecting the process behavior including the predominant physical, chemical or biological phenomena in a wide range of operating conditions. In addition to the (time-)continuous behavior of the process state, discontinuous elements triggered by so-called discrete events are naturally present in many cases. The continuous phases, i.e. the period between two successive discrete events, of such processes can often be described by either ordinary differential equations (ODE) or differential-algebraic equations (DAE). In conjunction with an appropriate representation of the discontinuous elements a discrete-continuous dynamic system is obtained. Due to the mixed discrete and continuous behavior, these systems are also referred to as hybrid systems. Depending on the switching structure of the system, i.e. the sequence and arrangement of the discrete events, the associated optimization problems either result in purely continuous or discrete-continuous dynamic optimization problems.

In this work, a disjunctive modeling and optimization framework for the optimization-based synthesis of discrete-continuous dynamic systems is presented. In order to highlight the main ideas in this short paper, the focus is on a few selected elements of the framework and solution techniques only. The interested reader is referred to Oldenburg et al. [1, 2] for more details.

2 Disjunctive Problem Formulation

The formal representation of discrete-continuous dynamic systems by means of abstract modeling languages and mathematical models is a subject of intensive research. Considerable effort is spent on the development of modeling concepts that are capable of capturing the continuous and discrete elements of a discrete-continuous dynamic system. The ultimate goal of all these modeling efforts is to build the foundation for robust and efficient analysis, simulation and optimization of process systems. A major challenge is to cope with the combinatorial character of systems governed by both continuous and discrete dynamics.

In this contribution, a disjunctive modeling framework for the mathematical representation of discrete-continuous dynamic systems based on previous work by Oldenburg [2] and Raman and Grossmann [3] is proposed. This model representation is especially well suited for capturing the discrete and continuous elements in a highly structured and systematic way. Such a model representation constitutes an alternative to existing mixed-integer approaches proposed in the literature, e.g. [4, 5]. A major advantage of the disjunctive compared to mixed-integer approaches is that it yields a problem formulation, which can be robustly and efficiently solved with generalized disjunctive programming (GDP) techniques [3, 6].

Each continuous phase of a discrete-continuous dynamic system is represented by a stage \( k \in K = \{1, \ldots, n_s\} \), where \( n_s \) represents the maximum number of potentially existing stages. The resulting multistage mixed-logic dynamic optimization problem (MLDO) contains global and conditional (disjunctive) constraints contained in \( n_d \) disjunctions:

\[
\min_{x_k(t), z_k(t), u_k(t), p_k, \Delta t_k} \sum_{k \in K} \Phi_k(x_k(t_k), z_k(t_k), p_k, t_k)
\]

(MLDO)

* Current address: BASF Aktiengesellschaft, D-67056 Ludwigshafen, Germany
** Corresponding author. E-mail: marquardt@lpt.rwth-aachen.de, Phone: +49 241 80-96712, Fax: +49 241 80-92326

© 2005 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
The disjunctive optimization problem (MLDO) can be solved using a variety of solution approaches. The basic assumption made by all of these approaches is that the disjunctive program (MLDO) reduces to a convex disjunctive program. Nonconvex problems can also be treated with a set of heuristic extensions. This is, however, beyond the scope of this paper.

For an illustration and a comparison of the big-M and the convex hull method, we focus on a single disjunction with \( i, j \) terms as well as only one inequality end point constraint for the purpose of illustration. Furthermore, the solution of a disjunctive optimization problem is based on replacing the disjunctive constraints by binary variables \( Y^{i,j} \) (cf. Eqs. (5), (6)) (in discretized form) by a set of new constraints. The basic idea is to represent the Boolean variables selected combinations of Boolean variable values, or, in other words, combinations of activated and deactivated constraints, contained in Eq. (5). In order to avoid undesired process sequences and design configurations and to reduce the complexity, selected combinations of Boolean variable values, or, in other words, combinations of activated and deactivated constraints, are related to each other by so-called propositional logic expressions [3] as shown in Eq. (6).

### 3 Reformation of the Disjunctive Optimization Problem

The disjunctive optimization problem (MLDO) can be solved using a variety of solution approaches. The basic assumption made by all of these approaches is that the disjunctive program (MLDO) reduces to a convex disjunctive program. Nonconvex problems can also be treated with a set of heuristic extensions. This is, however, beyond the scope of this paper.

The infinite-dimensional optimization problem (MLDO) is approximated by a finite-dimensional optimization problem using a combined discretization scheme [2] based on direct single and multiple shooting. The resulting vector of discretized problems can also be treated with a set of heuristic extensions. This is, however, beyond the scope of this paper.

For an illustration and a comparison of the big-M and the convex hull method, we focus on a single disjunction with \( j \) terms as well as only one inequality end point constraint for the purpose of illustration. Furthermore, the solution of a disjunctive optimization problem is based on replacing the disjunctive constraints (cf. Eqs. (5), (6)) (in discretized form) by a set of new constraints. The basic idea is to represent the Boolean variables selected combinations of Boolean variable values, or, in other words, combinations of activated and deactivated constraints, contained in Eq. (5). In order to avoid undesired process sequences and design configurations and to reduce the complexity, selected combinations of Boolean variable values, or, in other words, combinations of activated and deactivated constraints, are related to each other by so-called propositional logic expressions [3] as shown in Eq. (6).

\[
V_{i \in D_j} \left[ \begin{array}{c} Y^{i,j} \\ q^{j_i}(x_{k_j}(t), z_{k_j}(t), u_{k_j}(t), p_{k_j}, t) = 0 \\ x_{k_j}(t_{k_j-1}) - B_{k_{j-1}} x_{k_{j-1}}(t_{k_{j-1}}) = 0 \\ r^{j_i}(x_{k_j}(t), z_{k_j}(t), u_{k_j}(t), p_{k_j}, t) \leq 0 \\ s^{j_i}(x_{k_j}(t), z_{k_j}(t), u_{k_j}(t), p_{k_j}, t_{k_j}) \leq 0 \\ \end{array} \right] \\
\forall i \in [t_{k_j-1}, t_{k_j}], \forall j \in \{1, \ldots, n_d\}, D_j = \{1, \ldots, n_j\}, \\
\Omega(Y) = True 
\]

\( Y^{i,j} \in \{True, False\} \) are Boolean variables with superscripts \( i \) and \( j \) used to indicate the \( i \)-th term of a disjunction \( j \in J \). For \( Y^{i,j} = True \) the set of differential equations and constraints associated with the disjunctive term denoted with \( i, j \) is enforced, otherwise, i.e. for \( Y^{i,j} = False \), the term marked with \( i, j \) is discarded. The \( i \)-th term contains differential equations and constraints describing the \( i \)-th mode the system residing in the continuous phase \( j \), which may – depending on the particular application – coincide with \( k \). For more information, the reader is referred to Oldenburg [2] and Barton and Lee [5].

\( x_k, z_k \) denote the vectors of differential and algebraic state variables for stage \( k \). Time-invariant parameters and control variables are represented by \( p_k \) and \( u_k \). \( f_k \) represents the set of differential-algebraic equations (DAEs) with a differential index of at most 1 (cf. Eq. (1)). It is assumed that the equation system can be transformed into semi-explicit form by simple algebraic manipulations. The initial conditions and stage transition conditions that are used to map the differential state variable values \( x_k \) across the stage boundaries for each stage are given in Eq. (2). For both intervals, \( i, j \) terms are enforced via Eqs. (3), (4). Whereas Eqs. (1)–(4) hold globally, there are conditional differential equations, initial and stage transition conditions as well as path and point constraints that are only enforced if the associated Boolean variable is \( True \) (cf. Eq. (5)). Note that the superscript \( g \) indicates that Eq. (2) is enforced globally in contrast to the stage transition conditions contained in Eq. (5). In order to avoid undesired process sequences and design configurations and to reduce the complexity, selected combinations of Boolean variable values, or, in other words, combinations of activated and deactivated constraints, are related to each other by so-called propositional logic expressions [3] as shown in Eq. (6).

\[
V_{i \in D} \left[ \begin{array}{c} \hat{\theta}^i(t) \\ \bar{\theta}^i(t) \\ \end{array} \right] \\
\forall i \in D 
\]

\( \hat{\theta}^i(t) \in \Theta, Y^i \in \{True, False\} \).
\( s^i(\theta, t_1) \) denotes the composite function \( s^i(x_k(\theta, t_1), t_1) \), where \( x_k(\theta, t_1) \) is obtained by (numerically) solving the differential model equations. \( \Theta \) is a vector collecting the upper and lower bounds defined for \( \theta \). The big-M reformulation (see e.g. Floudas [7] for more information) of the disjunction (7) is derived via

\[
(a) : \quad \tilde{s}^i(\theta, t_1) \leq M^i(1 - y^i), \quad \theta \in \Theta, \quad (b) : \quad \sum_{i \in D} y^i = 1, \quad y^i \in \{0, 1\}, \quad \forall i \in D,
\]

where \( M^i \) is a sufficiently large upper bound for the constraint \( \tilde{s}^i \). For \( y^i = 1 \), the inequality constraint (and thus the disjunctive term) is enforced since Eq. (8a) reduces to \( \tilde{s}^i(\theta, t_1) \leq 0 \). Otherwise, if \( y^i = 0 \), the constraint becomes redundant since Eq. (8a) is relaxed to \( \tilde{s}^i(\theta, t_1) \leq M^i \). The specification of a value for the big-M parameter \( M^i \) is non-trivial in general. The tightest of all valid upper bounds \( M^i \) can be calculated from \( M^i = \max \{ \tilde{s}^i(\theta, t_1) | \theta \in \Theta \}, \quad \forall i \in D \), which requires finding the maximum of the convex scalar function \( \tilde{s}^i_1(\theta_1, t_1) \). This constitutes a concave optimization problem which has to be solved globally. Otherwise, \( M^i \) will not be a valid upper bound. In practice, the parameter \( M^i \) is often determined by an ad hoc procedure, i.e. by simply assigning it to a very large positive number.

Alternatively, a reformulation of the disjunction (7) can be derived via the so-called convex hull relaxation technique. The basic idea is to disaggregate the continuous variable vector \( \theta \), which is assumed to be non-negative without loss of generality, into a set of new vectors \( \nu^i \) each of which is associated with one disjunctive term \( \theta - \sum_{i \in D} \nu^i = 0 \). With this new set of variables defined for each disjunctive term, the convex hull of a disjunction \([6, 8]\) can be stated as

\[
(a) : \quad y^i \tilde{s}^i\left(\frac{\nu^i}{y^i}, t_1\right) \leq 0, \quad (b) : \quad 0 \leq \nu^i \leq y^i \nu^i_D, \quad (c) : \quad \sum_{i \in D} y^i = 1, \quad y^i \in \{0, 1\}, \quad \forall i \in D.
\]

\( \nu^i_D \) denotes a valid upper bound for the disaggregated variables \( \nu^i \) which is typically but not necessarily identical to the upper bound \( \theta_D \) (defined via the set \( \Theta \)) of the continuous decision variables. For \( y^i = 1 \), the inequality constraint (and thus the corresponding disjunctive term) is enforced since Eq. (9a) reduces to \( \tilde{s}^i(\theta, t_1) \leq 0 \) with \( \nu^i = \theta \) using the variable disaggregation together with Eqs. (9b) and (9c). Otherwise, if \( y^i = 0 \), the disjunctive constraint becomes redundant since Eq. (9a) formally reduces to \( 0 \cdot \tilde{s}^i(\frac{\nu^i}{y^i}, t_1) \leq 0 \). Note that \( 0 \cdot \tilde{s}^i(\frac{\nu^i}{y^i}, t_1) \geq 0 \) since \( \nu^i = 0 \) for \( y^i = 0 \), which simply follows from Eq. (9b). The constraint function (9a) is continuous but not differentiable for \( y^i = 0 \). Note that in conjunction with implicit model constraints the convex hull relaxation (9) entails that the differential model equations themselves have to be modified since each \( \theta \) is to be replaced by \( \nu^i/y^i \) (cf. Eq. (9a)). This is not the case for the big-M method.

Clearly, for the purpose of the implementation of Eq. (9a) a suitable regularization strategy is required. In the work of Grossmann and Lee [3] and Oldenburg [2] it is shown that Eq. (9a) can be replaced by Eq. (10) with \( \varepsilon \) being a small positive tolerance:

\[
y^i \tilde{s}^i\left(\frac{\nu^i}{y^i + \varepsilon}, t_1\right) \leq 0, \quad \forall i \in D.
\]

A continuous relaxation\(^1\) of the disjunction (7) via a big-M or a convex hull reformulation is simply obtained by redefining the domain of the binary variable as \( 0 \leq y^i \leq 1 \). Hence, \( y^i \) is treated as a continuous variable that can adopt values ranging from 0 to 1.

There exists a tradeoff between the tightness, i.e. the relative difference between the objective function values at the relaxed and the optimal solution, of the relaxation and the number of variables and constraints involved in the big-M and convex hull reformulations. Due to the fact that no variable disaggregation is required for the big-M formulation, this reformulation clearly contains considerably fewer continuous decision variables and constraints. The quality of the reformulation can, however, not only be measured by comparing the number of variables and constraints. In fact, a very important question is how tight a disjunction is described by a reformulation with continuously relaxed binary decision variables, i.e. for \( y^i \in [0, 1] \). The tightness of a relaxation is a key performance measure for most solution techniques. Therefore, the relaxation should be as tight as possible. In this regard, the convex hull approach is superior to the big-M method due to the fact that the convex hull reformulation typically yields a much tighter relaxation. In more concrete terms, a relaxation based on convex hull can be shown to be at least as tight as a relaxation based on the big-M method [6].

### 4 Illustrative Example

In order to illustrate the proposed approach, an optimization problem based on the work of Barton and Lee [5] is treated in the following. The optimization problem can formally be stated as

\[
\min_{x} \int_{t_0}^{t_f} x^2 \, dt, \quad (P)
\]

\(^1\) Continuous relaxations are required for a numerical solution based on GDP as well as MINLP algorithms.
s.t. $t \in [t_0, t_6]$: $x(t) \in \text{Mode 1} \setminus \text{Mode 2} \setminus \text{Mode 3}$

Mode 1: $\frac{dx}{dt} = -2t x + p$, Mode 2: $\frac{dx}{dt} = \frac{x + p}{t + 10}$, Mode 3: $\frac{dx}{dt} = -4 \exp(-t) x + p$ \hspace{1cm} (11)

$x(t_0) = 1$, $p \in [-4, 4]$, $t_{k-1} = k - 1$, $k = 1, \ldots, 7$.

In each continuous phase, the system state can reside in exactly one of the three different modes leading to $3^6 = 729$ structurally different process sequences. The continuity of the state is enforced via stage transition conditions. For the solution of the optimization problem (P), a corresponding GDP problem is formulated. Each disjunction of this GDP is composed of three disjunctive terms each representing one mode. Moreover, due to the presence of six continuous phases, six disjunctions are required. Due to space limitations we omit stating the full GDP representation of problem (P) here.

Before solving the optimization problem (P), it is converted into two mixed-integer optimization problems using the big-M and the convex hull method, respectively. These optimization problems are solved with continuously relaxed binary variables. The solution to the big-M model yields an objective function value of 0.0 whereas the objective function of the convex hull model takes a value of 1.2026 at the optimal solution. When compared to the optimal mixed-integer solution with an objective function value of 1.3071 (as will be shown below), the continuous relaxation of the convex hull model is very tight with a relaxation gap $\frac{1.3071 - 1.2026}{1.3071} = 0.08$. The relaxation gap of the big-M model takes the value 1.0 with an objective function value of 0.0, which was calculated for the relaxed optimization problem.

The optimization problem (P) is solved by applying the nonlinear convex hull representation of the disjunctions and the Outer Approximation (OA) decomposition method [9]. The globally optimal solution with a parameter value $p = -0.189$ and an objective function value of 1.3071 to this optimization problem is found in major iteration eleven. Note that applying the standard OA method to solve a disjunctive optimization problem that has been reformulated using the convex hull reformulation (cf. Eq. (9)) is equivalent to applying the logic-based OA method by Türkay and Grossmann [10] to the same disjunctive optimization problem [6]. The logic-based OA method is a GDP algorithm based on decomposition.

The efficiency of the solution with the OA algorithm becomes obvious when considering that only eleven out of 729 nonlinear optimization problems have to be solved to locate the optimal solution. A computing time of 198 s was spent on a PC with a 2.4+ GHz AMD Athlon processor whereas a complete enumeration for solving the optimization problem (P) took 12.3 h on the same PC. Similar results were obtained when employing the big-M reformulation technique, and in conjunction with the OA method the performance was even slightly better. Obviously, the tightness of the convex hull reformulation could not outweigh the increased variable dimension for this particular application. In future research work, a further GDP method, the logic-based branch & bound method proposed by Grossmann and Lee [6] will be investigated in the context of discrete-continuous dynamic optimization.

5 Conclusions

A disjunctive modeling and optimization framework for the synthesis of discrete-continuous dynamic systems has been proposed. Based on two different reformulation techniques for disjunctive constraints important ideas of GDP methods and their application in the context of discrete-continuous dynamic optimization problems have been highlighted. A discrete-continuous dynamic optimization example problem has served as an illustration.

Acknowledgements  The financial support from Deutsche Forschungsgemeinschaft under the Gottfried Wilhelm Leibniz program is gratefully acknowledged.

References