Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers and Chemical Engineering 28 (2004) 1951–1966

Computers & Chemical Engineering

www.elsevier.com/locate/compchemeng

Continuous reformulations of discrete–continuous optimization problems

Oliver Stein [a], Jan Oldenburg [b], Wolfgang Marquardt [b,*]

[a] *Lehrstuhl C für Mathematik, RWTH Aachen University, Templergraben 55, D-52056 Aachen, Germany*
[b] *Lehrstuhl für Prozesstechnik, RWTH Aachen University, Templergraben 55, D-52056 Aachen, Germany*

Received 24 July 2003; received in revised form 22 December 2003; accepted 26 March 2004

Available online 6 May 2004

Abstract

This paper treats the solution of nonlinear optimization problems involving discrete decision variables, also known as generalized disjunctive programming (GDP) or mixed-integer nonlinear programming (MINLP) problems, that arise in process engineering. The key idea is to eliminate the discrete decision variables by adding a set of continuous variables and constraints that represent the discrete decision space of the optimization problem. With such a reformulation, we are able to apply solution algorithms for purely continuous nonlinear optimization problems to efficiently calculate local minima of GDP or MINLP problems. In this contribution, we propose different alternatives to reformulate GDP/MINLP problems as continuous optimization problems. We furthermore investigate theoretical properties of the different reformulations with regard to their numerical solution. The proposed formulations are illustrated and analyzed on the basis of optimization problems dealing with process engineering applications involving stationary as well as dynamic process models.

Keywords: Discrete–continuous nonlinear optimization; Disjunctive programming; Complementarity; Process engineering; Mixed-integer dynamic optimization

1. Introduction

Mathematical programming techniques have been successfully applied for solving a large variety of optimization problems that arise from process engineering applications. In many cases, these optimization problems involve continuous as well as discrete decision variables. Examples include, e.g. the synthesis of heat exchanger or reactor networks, the optimization of separation processes, such as sequencing and tray optimization problems of distillation columns and the optimization of entire process flowsheets (Floudas, 1995). Problems with dynamic process models, such as the optimization of hybrid discrete–continuous dynamic systems, have also been treated recently (Avraam, Shah, & Pantelides, 1998). Optimization for integrated process design and control (Bansal, Perkins, & Pistikopoulos, 2002) as well as for the design of transient processes (Oldenburg, Marquardt, Heinz, & Leineweber, 2003; Sharif, Shah, & Pantelides, 1998) are further examples which rely on dynamic models. The discrete decisions in all these problems are usually related to the structure of the process whereas typical continuous variables are process states such as temperatures, concentrations or flows.

Extensive work has been addressed to problems with linear objective function and constraints, known as mixed-integer linear programming (MILP) problems. In fact, a number of powerful algorithms have been developed which are ready to solve practically relevant, large-scale problems of this type. As soon as the objective function and the constraints comprise nonlinear terms in the continuous variables, as is usually the case for problems in process engineering, the optimization problem is referred to as mixed-integer nonlinear programming (MINLP) problem. Algorithms for MINLP problems are either based on branch & bound with nonlinear programming (NLP) subproblems or on decomposition methods that alternately solve NLP and MILP subproblems. These algorithms are guaranteed to locate the global optimum if the nonlinear

* Corresponding author. Tel.: +49-241-8096712; fax: +49-241-8092326.
E-mail address: marquardt@lpt.rwth-aachen.de (W. Marquardt).

0098-1354/$ – see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2004.03.011

objective function as well as all constraints are convex. An alternative way to represent optimization problems with discrete and continuous decision variables is to formulate a generalized disjunctive program (GDP) where Boolean variables are used to activate or deactivate subsets of the constraints (Raman & Grossmann, 1994). The disjunctive representation supports the modeling of a design problem since the qualitative (logical) and quantitative (equations) parts of the problem are captured in a natural and direct way (Grossmann & Hooker, 2000). Moreover, a number of tailored solution techniques have been developed that exploit the specific structure of the disjunctive optimization problem (Lee & Grossmann, 2000; Türkay & Grossmann, 1996).

Optimization problems involving nonconvex objective function and constraints, as considered in this contribution, are by far more difficult to solve. Global MINLP algorithms for nonconvex problems have been developed in recent years and have been applied successfully to certain classes of problems. However, due to the high computational effort required by these methods, they are not yet capable to solve large-scale process engineering applications. Consequently, heuristic extensions have been proposed to enable the use of standard MINLP methods for nonconvex problems. These methods are rather efficient and thus applicable to large-scale problems while no guarantee for global optimality can be given. It is possible to apply NLP based branch & bound or decomposition methods, depending on the algorithm chosen with slight modifications to account for the nonconvexities.

Recently, Raghunathan and Biegler (2003) have proposed to reformulate discrete–continuous optimization problems as purely continuous optimization problems with complementarity constraints. In this approach, the discrete variable set of an MINLP problem is replaced by continuous variables which are restricted to take discrete values by enforcing special types of continuous constraints. These nonconvex and partially degenerate constraints are referred to as complementary or equilibrium constraints.

Similarly, the aim of this work is to obtain a purely continuous reformulation of MINLP problems to be able to apply algorithms designed for continuous optimization problems. It will be shown that all continuous reformulation approaches inevitably lead to nonconvex optimization problems which will only be tackled with local solution techniques in this contribution. Thus, our aim is to propose efficient ways to locally solve nonconvex MINLP problems on the basis of NLP solution methods. In Section 2, we start our investigations with a GDP problem formulation comprising disjunctive constraints and logical expressions. Subsequently, Section 3 shortly discusses the so-called big-M approach which can be used to reformulate the disjunctive program as an MINLP problem.

Starting from Section 4, we concentrate on the idea of eliminating the discrete decision variables of the reformulated problems by adding a set of continuous constraints. One possibility to describe the discrete set of decision variables is to use complementarity constraints as proposed by Raghunathan and Biegler (2003). As these constraints are degenerate, they have to be tackled by tailored NLP solvers, or they can be regularized. The latter automatically yields a relaxation of the discrete set to a continuous, one-dimensional set, where the variables from the relaxed set are only *approximate* for the original decision variables. We also show that there is an alternative approach to approximate continuous decision variables, when we replace the complementarity constraints by a circle condition.

In Section 5, we introduce a completely different reformulation for the discrete decision set where it can actually be replaced *exactly* by a one-dimensional set, i.e. without relaxing the original problem to an approximation. Our reformulation leads to non-degenerate continuous constraints, so that not only tailored but *any* NLP solver may be used to treat the problem numerically. Moreover, the new one-dimensional decision set does not become discrete in the presence of disjoint equality constraints in contrast to the sets from Section 4.

The proposed reformulation and solution methods are illustrated in Section 6 with two example problems involving a stationary and a dynamic process model, respectively.

2. Optimization problem formulation

We consider a generalized disjunctive representation (Raman & Grossmann, 1994) of nonlinear optimization problems, where an objective function is minimized subject to two different types of constraints, i.e. global constraints that hold irrespectively of any discrete decision and constraints contained in disjunctions that are only enforced if a corresponding Boolean variable $Y_{i,k}$ is True. Hence, the optimization problem is formulated as follows:

$$\min_{x,Y} \; \Phi(x) + \sum_{k\in K} b_k \tag{P1}$$

$$\text{s.t.} \quad f(x) = 0, \tag{1}$$

$$g(x) \le 0, \tag{2}$$

$$\bigvee_{i\in D_k} \begin{bmatrix} Y_{i,k} \\ h_{i,k}(x) = 0, \\ r_{i,k}(x) \le 0, \\ b_k = \gamma_{i,k}, \end{bmatrix}, \quad k \in K, \tag{3}$$

$$D_k = \{1, 2, \ldots, n_k\}, \qquad K = \{1, 2, \ldots, m\},$$
$$\Omega(Y) = \text{True}, \qquad Y_{i,k} \in \{\text{True}, \text{False}\}. \tag{4}$$

In the GDP formulation (P1), x represents a vector of continuous process variables and $Y_{i,k}$ are Boolean variables. b_k is a continuous scalar and $\gamma_{i,k}$ represents a fixed charge. The objective function comprises the sum of all fixed charges and a nonlinear term $\Phi(x)$. Whereas the model Eq. (1) and inequality constraints (2) hold irrespective of discrete choices, there are further (model) equations and inequality constraints that are contained in m disjunctions. Each disjunction $k \in K$ may consist of several terms $i \in D_k$, where n_k defines the number of terms for each disjunction. Note that exactly one term $i \in D_k$ holds per disjunction, i.e. $\bigvee_{i\in D_k}$ is understood as an 'exclusive or' operator. The disjunctive constraints contained in the i-th term of disjunction k are only enforced if the Boolean variable value $Y_{i,k}$ is True. Otherwise, if $Y_{i,k}$ is False, the corresponding constraints are removed from the optimization problem. The Boolean variables themselves are related to each other by so called propositional logic constraints (4). These logic constraints are used to model interrelationships between disjunctive constraints. For example, assume that the first disjunctive term from disjunction $k = 1$ has to be selected ($Y_{1,1} =$ True) if the first term from disjunction $k = 2$ is removed from the constraint set ($Y_{1,2} =$ False). This situation can be expressed by the implication

$$\neg Y_{1,2} \Rightarrow Y_{1,1}, \tag{5}$$

which can be transformed into a constraint of type (4) (Williams, 1999):

$$Y_{1,2} \vee Y_{1,1} = \text{True}. \tag{6}$$

In many applications, especially in process synthesis problems, each disjunction (3) contains exactly two terms of the following form:

$$\begin{bmatrix} Y_k \\ h_k(x) = 0, \\ r_k(x) \le 0, \\ b_k = \gamma_k, \end{bmatrix} \vee \begin{bmatrix} \neg Y_k \\ B^k x = 0, \\ \\ b_k = 0, \end{bmatrix}, \quad k \in K. \tag{7}$$

Hence, only one Boolean variable Y_k is used in each disjunction. If Y_k is True, the corresponding constraints are enforced and fixed charges b_k may be assigned with a nonzero value. Otherwise, if Y_k is False, a subset of the process variables x or as well as the fixed charges of disjunction k are set to zero to account for the fact that the corresponding process unit does not exist.

In order to illustrate the disjunctive problem formulation, we consider a simple example taken from Lee and Grossmann (2000), which has only one disjunction of the generic type (3).

Example 2.1 (Lee & Grossmann, 2000).

$$\min_{x,Y}\ (x_1-3)^2 + (x_2-2)^2 + b \tag{8}$$

$$\text{s.t.}\quad 0 \le x_i \le 8, \quad i = 1, 2, \tag{9}$$

$$\begin{bmatrix} Y_1 \\ (x_1)^2 + (x_2)^2 - 1 \le 0 \\ b = 2 \end{bmatrix}, \tag{10}$$

$$\vee \begin{bmatrix} Y_2 \\ (x_1-4)^2 + (x_2-1)^2 - 1 \le 0 \\ b = 1 \end{bmatrix}, \tag{11}$$

$$\vee \begin{bmatrix} Y_3 \\ (x_1-2)^2 + (x_2-4)^2 - 1 \le 0 \\ b = 3 \end{bmatrix}, \tag{12}$$

$$Y_i \in \{\text{True}, \text{False}\}, \quad i = 1, 2, 3. \tag{13}$$

We will revisit this simple example problem throughout the paper to illustrate reformulation and solution techniques.

3. Reformulation of the disjunctive optimization problem

The fact that any optimization problem in disjunctive form (P1) can be posed as an equivalent MINLP problem (Grossmann & Hooker, 2000) allows to straightforwardly solve problem (P1) with any available MINLP solver. The optimization problem (P1) is reformulated as an MINLP problem by transforming the disjunctive constraints into big-M constraints and by replacing the Boolean variables $Y_{i,k}$ by binary variables $y_{i,k} \in \{0, 1\}$. Hence, the GDP (P1) can be transformed into the following MINLP problem with positive big-M constants $M_{i,k}$, which are used to represent sufficiently large bounds (Williams, 1999):

$$\min_{x,y}\ \Phi(x) + \sum_{k\in K} b_k \tag{P2}$$

$$\text{s.t.}\quad f(x) = 0, \tag{14}$$

$$g(x) \le 0, \tag{15}$$

$$-M_{i,k}(1-y_{i,k}) \le h_{i,k}(x) \le M_{i,k}(1-y_{i,k}), \tag{16}$$

$$r_{i,k}(x) \le M_{i,k}(1-y_{i,k}) \tag{17}$$

$$b_k = \sum_{i\in D_k} \gamma_{i,k} y_{i,k}, \tag{18}$$

$$Ay \le a, \tag{19}$$

$$\sum_{i\in D_k} y_{i,k} = 1, \tag{20}$$

$$y_{i,k} \in \{0, 1\}, \quad i \in D_k,\ k \in K. \tag{21}$$

A constraint from Eqs. (16)–(18) is enforced if $y_{i,k} = 1$. Otherwise, if $y_{i,k} = 0$, the corresponding constraint becomes redundant, given the absolute values of the big-M constants $M_{i,k}$ are large enough. The propositional logic constraints (4) can be modeled by linear constraints (19) on the binary variables. For the arguments here and in the sequel of the article it is important to note that with these inequalities not only exclusive but also inclusive 'or'-relations can be modeled, although a binary variable itself takes only *exclusively* the values 0 or 1. In fact, for two binary variables y_1 and y_2 the inclusive relation $y_1 + y_2 \geq 1$ becomes exclusive under the additional relation $y_1 + y_2 \leq 1$.

Note that there is no increase in the problem size in terms of variables or constraints due to a reformulation on the basis of big-M constraints. However, it is known that a continuous relaxation of the constraints (16)–(18) in many cases yields a weak lower bound to the solution of the optimization problem (Grossmann, 2002). This is a profound drawback for all solution methods relying on tight lower bounds. Moreover, the specification of the big-M constants is not trivial and problem dependent.

Example 2.1 (continued). A big-M reformulation of the illustrative example problem 2.1 is stated as (cf. Lee & Grossmann, 2000):

$$\min_{x,y} \; (x_1-3)^2 + (x_2-2)^2 + b \tag{22}$$

$$\text{s.t.} \quad 0 \leq x_i \leq 8, \; i = 1, 2, \tag{23}$$

$$(x_1)^2 + (x_2)^2 - 1 \leq M(1-y_1), \tag{24}$$

$$(x_1-4)^2 + (x_2-1)^2 - 1 \leq M(1-y_2), \tag{25}$$

$$(x_1-2)^2 + (x_2-4)^2 - 1 \leq M(1-y_3), \tag{26}$$

$$b = 2y_1 + y_2 + 3y_3, \tag{27}$$

$$\sum_{i=1}^{3} y_i = 1, \tag{28}$$

$$y_i \in \{0, 1\}, \quad i = 1, 2, 3, \tag{29}$$

where the big-M parameter M is assigned with $M = 30$.

A convex hull reformulation of the disjunctions proposed by Lee and Grossmann (2000) avoids the aforementioned drawbacks. Using the convex hull reformulation, the number of variables is, however, increased significantly when compared to the disjunctive problem formulation (P1). Throughout this paper, we will not make use of the convex hull reformulation technique but such a reformulation of the illustrative example problem can be found in Lee and Grossmann (2000). On the basis of either a big-M or a convex hull reformulation we can apply various solution algorithms for MINLP or GDP problems. For an excellent overview of standard as well as logic-based solution methods including very recent developments we refer to Grossmann (2002).

4. Representing the discrete decisions by *approximate* continuous variables

Instead of applying an MINLP algorithm for solving the discrete–continuous optimization problems introduced in the previous section directly, we intend to reformulate problem (P2) such that no discrete variables are present anymore.

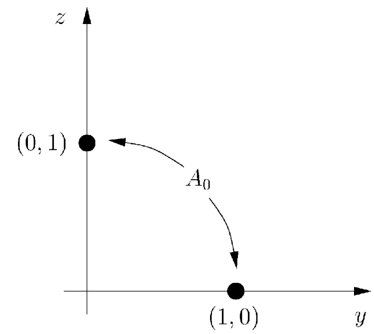

Fig. 1. Values of the discrete variable (y, z).

In particular, the discrete set defined in (20) and (21) will be replaced by a set of restrictions involving continuous variables only, which can be used as constraints to form a purely continuous NLP. Since NLP solvers are usually designed to work with continuous variables, i.e. variables from at least one-dimensional sets, the basic idea here is to increase the dimension of the constraint sets for $y_{i,k}$. Note that the discrete variables $y_{i,k}$ as defined in (21) are contained in a set of dimension zero. We call a set one-dimensional if it can be regularly parameterized by means of a single parameter.

For the presentation of our main ideas, we consider a single disjunction with $n_k = 2$ as it appears in (3), and we use Example 2.1 to illustrate how the introduced techniques extend to the case of a single disjunction with $n_k = 3$. We put $y_k := y_{1,k}$ as well as $z_k := y_{2,k}$ and drop the fixed index k. This leads to a single binary decision variable $y \in \{0, 1\}$ and its negation z, where the pair (y, z) can then attain exactly one of the values $(1, 0)$ and $(0, 1)$, i.e.

$$(y, z) \in A_0 = \{(1, 0), \; (0, 1)\}, \tag{30}$$

(cf. Fig. 1). Hence, in this case the conditions (20) and (21) are replaced equivalently by (30).

In the case $n_k > 2$ there are several ways to use the set A_0 to reformulate (20) and (21) equivalently. A first possibility is to introduce additional variables $z_i = 1 - y_i$ and replace only (21) by the conditions $(y_i, z_i) \in A_0$, $i \in D$. For Example 2.1 this approach leads to a reformulation of (29) by

$$(y_1, z_1) \in A_0, \tag{31}$$

$$(y_2, z_2) \in A_0, \tag{32}$$

$$(y_3, z_3) \in A_0. \tag{33}$$

Note that the constraint (28) guarantees that exactly one of the variables y_1, y_2, y_3 is equal to 1, since these variables *can only take* the values 0 and 1. This restriction can be relaxed in conjunction with an alternative approach for modeling binary decision variables proposed in Section 5. Having these later developments in mind, already at this point we suggest an alternative reformulation of (20) and (21) using A_0, but without the constraint (28).

In fact, it is not hard to see that the constraints (20) and (21) can be equivalently replaced by the conditions in the general case where we have $k \in K$ disjunctions with $n_k > 2$

$$\left(y_{i,k}, \sum_{j\in D_k\setminus\{i\}} y_{j,k}\right) \in A_0, \quad i \in D_k, \quad k \in K. \tag{34}$$

For Example 2.1 this approach leads to a reformulation of (28) and (29) by

$$(y_1, y_2 + y_3) \in A_0, \tag{35}$$

$$(y_2, y_1 + y_3) \in A_0, \tag{36}$$

$$(y_3, y_1 + y_2) \in A_0. \tag{37}$$

An advantage of the latter reformulation is that it does not increase the problem dimension by auxiliary variables $z_{i,k}$.

4.1. The reformulation by a complementarity condition

There are a number of ways to describe A_0 with continuous constraints. Let us first look at the suggestion of Raghunathan and Biegler (2003) to replace (30) with the equivalent set of constraints

$$y \cdot z = 0, \tag{38}$$

$$y \geq 0, \quad z \geq 0, \tag{39}$$

$$y + z = 1. \tag{40}$$

In fact, the constraints (38) and (39) are known as a *complementarity* condition. They model a piecewise linear set with one kink at the origin in $\mathbb{R}^2$, as depicted in Fig. 2. Together with the constraint (40) one obtains exactly the set A_0 (cf. Fig. 3).

It is well-known that sets whose description contains complementarity conditions are not easy to treat numerically. In fact, the so-called Mangasarian–Fromovitz constraint qualification is violated everywhere in the feasible set as soon as a complementarity condition appears (cf. Scheel & Scholtes, 2000). This constraint qualification, however, is known to

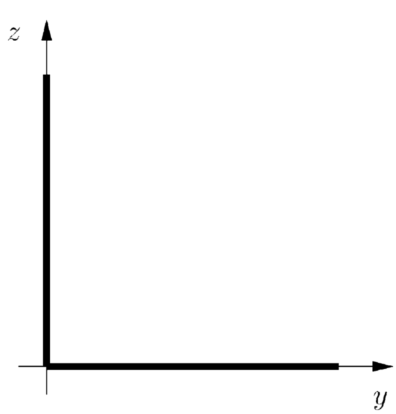

Fig. 2. The points which satisfy the complementarity conditions: (38) and (39).

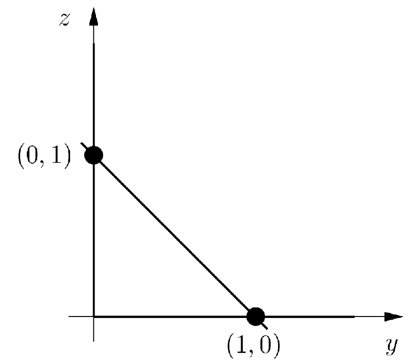

Fig. 3. Modeling discrete variables with a complementarity condition: (38)–(40).

characterize the (numerical) stability of the described set (cf. Jongen & Weber, 1991; Robinson, 1976).

There are many suggestions on how a complementarity condition can be treated numerically, in particular in the literature on so-called mathematical programs with equilibrium constraints (MPECs) which are optimization problems with complementarity conditions in the constraints (cf., e.g. Luo, Pang, & Ralph, 1996). In spite of the discouraging result of the violated Mangasarian–Fromovitz constraint qualification, recently it turned out that SQP and interior point methods with minor modifications can still successfully solve MPECs (cf. Leyffer, 2003; Raghunathan & Biegler, 2003). Another approach is to use regularization techniques.

One well-known regularization is to replace the condition (38) in (38)–(40) by its relaxation

$$y \cdot z \leq \mu \tag{41}$$

with some positive parameter μ. The idea is to trace the solutions of the corresponding auxiliary problems to a solution of the original problem while driving μ to zero.

For the reformulation of binary variables this approach means that the discrete set A_0 is replaced by the one-dimensional set

$$A_\mu = \Big\{ (0, 1) + t(1, -1) | t \in [0, 0.5 - \sqrt{0.25 - \mu}] \cup [0.5 + \sqrt{0.25 - \mu}, 1] \Big\} \tag{42}$$

which is disconnected for $\mu < 0.25$ as illustrated in Fig. 4. Hence, we have replaced discrete by continuous variables, at least via an approximation. In the following we will refer to the variables from the set A_μ as *approximate* continuous.

Example 2.1 (continued). For the numerical solution of the problem from Example 2.1 we now employ the big-*M* reformulation (22)–(28) of the disjunctive optimization problem (8)–(13). Here, the approximate constraints are defined by

$$(y_1, y_2 + y_3) \in A_\mu, \tag{43}$$

$$(y_2, y_1 + y_3) \in A_\mu, \tag{44}$$

$$(y_3, y_1 + y_2) \in A_\mu. \tag{45}$$

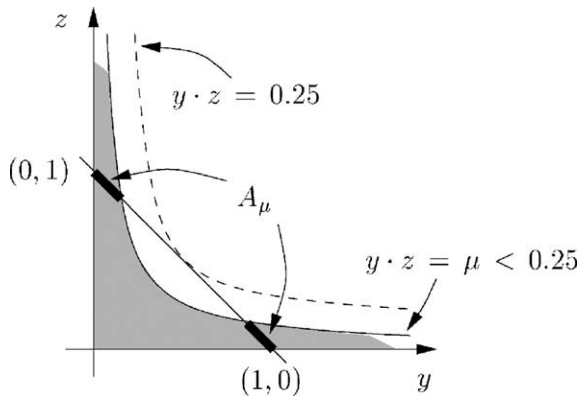

Fig. 4. Continuous variables for the relaxed complementarity condition.

or, equivalently,

$$y_1(y_2 + y_3) \le \mu, \tag{46}$$

$$y_2(y_1 + y_3) \le \mu, \tag{47}$$

$$y_3(y_1 + y_2) \le \mu, \tag{48}$$

$$y_1 + y_2 + y_3 = 1. \tag{49}$$

$$y_i \ge 0, \quad i = 1, \dots, 3, \tag{50}$$

They replace the integrality constraints (28) and (29) for the binary variables y (cf. Fig. 4).

We choose $\mu = 0.3$ as initial value and solve a sequence of NLP problems where μ is steadily decreased to the limit $\mu = 0$. Note, that the set $A_{\mu=0.3}$ leads to a fully continuous relaxation of all binary variables according to $0 \le y_i \le 1, i = 1, 2, 3$. Consequently, the NLP solver finds the relaxed optimal value 1.013 with $y_\mu = [0.029, 0.971, 0]$. The relaxation of y is restricted to continuous values for which $y_1 + y_2 + y_3 = 1$ and $y_i \sum_{j \ne i} y_j \le \mu, i = 1, 2, 3,$ do hold. As soon as $\mu \le 0.25$, the line segment is divided into two pieces as shown in Fig. 4. The optimal objective function value 1.172 with integral values $y = [0, 1, 0]$ of problem (22)–(29) is finally found when μ approaches 0. The problem was solved using GAMS (Brooke, Kendrick, Meeraus, & Raman, 1998) as modeling system and SNOPT (Gill, Murray, & Saunders, 1998) as well as CONOPT (Drud, 1995) as NLP solvers in less than one CPU second.

4.2. The reformulation by a circle condition

We point out that there are two serious drawbacks of the reformulation by a complementarity condition. First, a look at Fig. 3 shows that the kink at the origin is irrelevant for our problem because of the additional constraint (40). Thus, there is no need to use the numerically demanding complementarity condition (38) and (39) together with (40), but any function with a smooth zero set and the correct intersection points would do. For example, one can use the constraint

$$(y - \tfrac{1}{2})^2 + (z - \tfrac{1}{2})^2 = \tfrac{1}{2} \tag{51}$$

which is illustrated in Fig. 5.

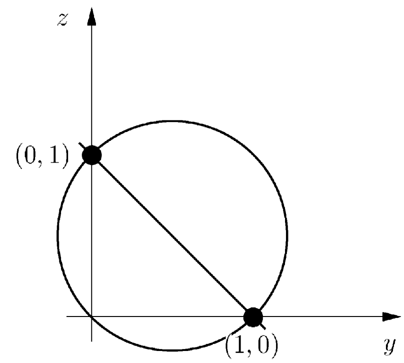

Fig. 5. The circle condition.

Here, the Mangasarian–Fromovitz constraint qualification and even the stronger linear independence constraint qualification are satisfied everywhere in the set A_0 (for background information on constraint qualifications we refer to Bazaraa, Sherali, & Shetty, 1993). This can be seen as an important advantage when compared to the properties of A_0 represented by the complementarity condition. A second drawback which the circle condition (51) shares with the a complementarity condition (38) and (39) is that the variables are still contained in the discrete set A_0 which is described by continuous constraints. In order to obtain a one-dimensional set we can again relax the conditions that describe A_0 to obtain a set A_ν corresponding to A_μ (cf. Eq. (42)) introduced in Section 4.1. In fact, the constraints

$$(y - \tfrac{1}{2})^2 + (z - \tfrac{1}{2})^2 \le \tfrac{1}{2}, \tag{52}$$

$$\left(y - \frac{1}{2}\right)^2 + \left(z - \frac{1}{2}\right)^2 \ge \left(\frac{1}{\sqrt{2}} - \nu\right)^2, \tag{53}$$

$$y + z = 1 \tag{54}$$

with $\nu > 0$ describe sets A_ν (cf. Fig. 6), which correspond to the sets A_μ ($\mu > 0$) via a reparametrization, i.e. we arrive at the same set of approximate continuous variables.

Unfortunately, for the limiting case $\nu = 0$ the circle is not described by one equality constraint but by two inequalities with gradients pointing in opposite directions, so that

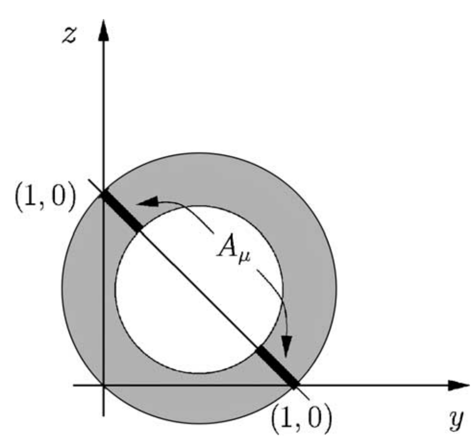

Fig. 6. The circle relaxation.

the Mangasarian–Fromovitz constraint qualification is then again violated in A_0.

Example 2.1 (continued). For the reformulation of Example 2.1 the relaxed circle condition means to replace the integrality constraint (29) by

$$\left(\frac{1}{\sqrt{2}}-\nu\right)^2 \leq \left(y_1-\frac{1}{2}\right)^2+\left(y_2+y_3-\frac{1}{2}\right)^2 \leq \frac{1}{2} \tag{55}$$

$$\left(\frac{1}{\sqrt{2}}-\nu\right)^2 \leq \left(y_2-\frac{1}{2}\right)^2+\left(y_1+y_3-\frac{1}{2}\right)^2 \leq \frac{1}{2} \tag{56}$$

$$\left(\frac{1}{\sqrt{2}}-\nu\right)^2 \leq \left(y_3-\frac{1}{2}\right)^2+\left(y_1+y_2-\frac{1}{2}\right)^2 \leq \frac{1}{2} \tag{57}$$

As in the previous section, we employ the big-*M* reformulation in order to solve problem (22)–(29). The initial set A_ν is defined with $\nu = 1/\sqrt{2}$, a value that corresponds to $\mu = 0.25$ in conjunction with the complementarity condition (cf. Figs. 4 and 6). While ν is driven to 0, a set of NLP problems is solved until an optimal solution is found as reported in the previous section. For this example, there was no qualitative difference between the numerical properties of the two sets A_μ and A_ν defined by either the regularized complementarity or the relaxed circle condition. With both NLP solvers SNOPT and CONOPT we are able to obtain the solution in less than 1 s on a 600 MHz SUN Ultra Workstation.

5. Representing the discrete decisions by *exact* continuous variables

Although both the reformulation by a complementarity condition and the reformulation by a circle condition from Section 4 lead to well-performing numerical methods for Example 2.1, they share two intrinsic drawbacks:

- the replacement for A_0 is one-dimensional, but only approximate,
- in the (limiting) case of an exact description for A_0, the Mangasarian–Fromovitz constraint qualification is violated.

Since these properties may affect the numerical solution of larger and more complex problems than the one of Example 2.1, we propose a different continuous reformulation of the integrality constraints with better theoretical features. Our subsequent considerations are based on two alternative model reformulations, that allow us to replace the discrete decision variables defined in (21) by variables $y_{i,k}$, which are *not* defined on a discrete set as, e.g. A_0. In fact, these two model reformulations have the property of being equivalent to the corresponding disjunctive optimization problem in conjunction with one-dimensional rather than discrete variables $y_{i,k}$. Before presenting the two model reformulations in greater detail, we focus on the variables

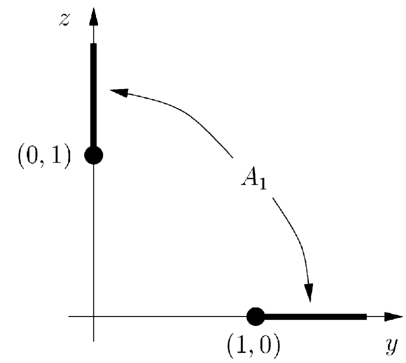

Fig. 7. A one-dimensional feasible set for (y, z).

$y_{i,k}$ and show how a continuous, one-dimensional set A_1 can be defined using appropriate constraints.

5.1. *A continuous representation of the discrete decision variables*

We define $y_{i,k}$ as continuous variables of dimension one according to:

$$y_{i,k} \in \{0\} \cup [1, \infty).$$

Of course, now the negation of $y_{i,k}$ in general does not coincide with $1 - y_{i,k}$, as the value of $y_{i,k}$ might exceed 1. On the other hand, for the case $n_k = 2$ as above we obtain

$$(y, z) \in A_1 = ([1, \infty) \times \{0\}) \cup (\{0\} \times [1, \infty)) \tag{58}$$

(cf. Fig. 7). Hence, the negation of $y_{i,k}$ is coded by the variable $z_{i,k}$.

The set A_1 is obviously one-dimensional and is an *exact* rather than approximate model of a discrete decision. Therefore, the variables defined by the set A_1 are referred to as *exact* continuous.

To be able to apply an NLP solution algorithm, we have to describe A_1 by continuous constraints. One possibility, of course, is to use the complementarity conditions (38) and (39) with the additional constraint

$$y + z \geq 1. \tag{59}$$

However, it is also possible to choose a function with an appropriate zero set, such that the linear independence constraint qualification holds everywhere in the feasible set. A function with these properties is the so-called *Fischer–Burmeister* function:

$$\varphi_{\mathrm{FB}}(y, z) = y + z - \sqrt{y^2 + z^2} \tag{60}$$

(cf. Fig. 8 for a plot of its graph and some level lines).

This means that we can write

$$A_1 = \{(y, z) \in \mathbb{R}^2 | \varphi_{\mathrm{FB}}(y, z) = 0,\ y + z \geq 1\}. \tag{61}$$

Equivalently, we could use the *natural residual* function

$$\varphi_{\mathrm{NR}}(y, z) = \tfrac{1}{2}(y + z - \sqrt{(y - z)^2}) \tag{62}$$

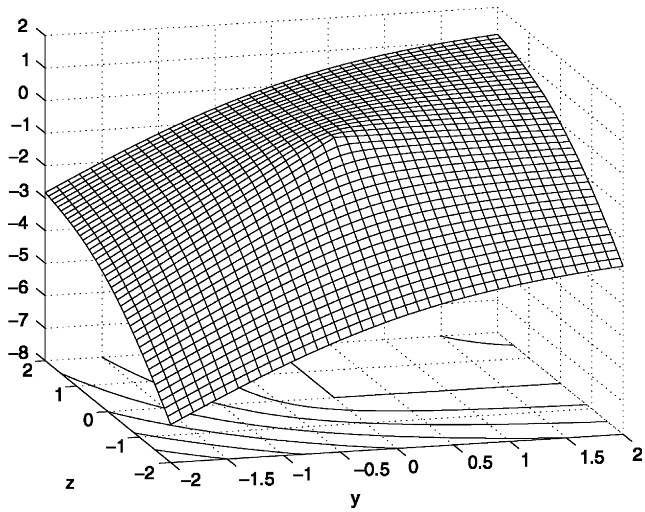

Fig. 8. Graph and level lines of φ_{FB}.

instead of φ_{FB}, as well as a multitude of other so-called *NCP*-functions (for a survey, see, e.g. Chen, Chen, & Kanzow, 2000). *NCP*-functions are used for the description of nonlinear complementarity problems. They are designed such that their zero set coincides with the set defined by (38) and (39) (cf. Fig. 2). A description like (61) reveals better numerical properties than the original description via (38) and (39). For example, whereas the Mangasarian–Fromovitz constraint qualification is violated everywhere in the set under a description via (38) and (39), the description as the zero set of the Fischer–Burmeister function even leads to the validity of the linear independence constraint qualification everywhere in the set, except for the origin.

Note that in the case $n_k > 2$ we can now describe the binary decisions via

$$\left(y_{i,k}, \sum_{j \in D_k \setminus \{i\}} y_{j,k} \right) \in A_1, \quad i \in D_k, k \in K. \tag{63}$$

In terms of the Fischer–Burmeister function, and using (61), this condition is equivalent to

$$\varphi_{\mathrm{FB}} \left(y_{i,k}, \sum_{j \in D_k \setminus \{i\}} y_{j,k} \right) = 0, \tag{64}$$

$$\sum_{i \in D_k} y_{i,k} \geq 1, \quad i \in D_k, k \in K. \tag{65}$$

To obtain an optimization problem which is equivalent to the original disjunctive problem (P1) we can either use a problem formulation based on bilinear products or on specially tailored big-*M*constraints as we will see below.

5.2. Modeling propositional logic constraints with exact continuous variables

Of course we still have to answer the question of how logical conditions on two logical variables Y_1 and Y_2 should be modeled when (y_1, z_1) and (y_2, z_2) are not discrete but continuous as proposed in (58). This can easily be done by adding inequality constraints. In fact, $Y_1 \wedge Y_2$ is true if and only if $y_1 \geq 1$ and $y_2 \geq 1$. Moreover, $Y_1 \vee Y_2$ is true if and only if $y_1 + y_2 \geq 1$. For the negation of Y_1 we can *not* use $1 - y_1$, as y_1 might take a value strictly larger than one. On the other hand, for $n_k = 2$ the negation of Y_1 is already coded in the variable z_1. Moreover, for $n_k > 2$ the negation of $y_{i,k}$ is coded in $\sum_{j \in D_k \setminus \{i\}} y_{j,k}$, and one can proceed as above. We point out that, just like in the discrete case, inclusive as well as exclusive 'or'-relations can be modeled with exact continuous variables.

For example, to express the implication $Y_2 \Rightarrow Y_1$, we would first write the equivalent propositional logic relation: $\neg Y_2 \vee Y_1 = \text{True}$. A transformation using variables as defined in (63) is then given as $z_2 + y_1 \geq 1$, where z_2 represents the negation of y_2. Note that if y_1, y_2 were defined on A_0, we could state $y_1 - y_2 \geq 0$ exploiting that $z_2 := 1 - y_2$ holds in this case.

5.3. A problem reformulation based on binary multiplication

A problem reformulation based on binary multiplication reads as:

$$\min_{x,y} \ \Phi(x) + \sum_{k \in K} b_k \tag{P3}$$

$$\text{s.t.} \quad f(x) = 0, \tag{66}$$

$$g(x) \leq 0, \tag{67}$$

$$y_{i,k} h_{i,k}(x) = 0, \tag{68}$$

$$y_{i,k} r_{i,k}(x) \leq 0, \tag{69}$$

$$y_{i,k}(b_k - \gamma_{i,k}) = 0, \tag{70}$$

$$\bar{A} y \leq \bar{a}, \tag{71}$$

$$\sum_{i \in D_k} y_{i,k} \geq 1, \tag{72}$$

$$\varphi_{\mathrm{FB}} \left(y_{i,k}, \sum_{j \in D_k \setminus \{i\}} y_{j,k} \right) = 0, \quad i \in D_k, k \in K, \tag{73}$$

where each disjunctive constraint is multiplied by a variable $y_{i,k}$. If $y_{i,k} = 0$, the corresponding constraint becomes redundant. On the other hand, a constraint contained in a disjunction is enforced with $y_{i,k} = 1$. The fact that any disjunctive constraint could equivalently be enforced by $y_{i,k} \geq 1$ motivated the idea to model the binary variables by continuous variables which are defined on A_1 instead of A_0. To account for the fact that the reformulation of propositional logic constraints into linear inequalities follows slightly different rules than those holding in conjunction with problem formulation (P2), the matrix A and the vector a are modified to $\bar{A}$ and $\bar{a}$, respectively (cf. Eq. (71)).

It is important to note that the problem formulation (P3) has the drawback of being nonconvex even if the nonlinear, disjunctive constraints of the original optimization problem are convex, as in the case of the Example 2.1. Thus, a problem reformulation based on binary multiplication would be employed only if the disjunctive optimization problem was nonconvex itself. In fact, a large portion of process engineering applications lead to nonconvex problems. Hence, this drawback should not be regarded as a strong limitation. Also note that the nonconvex expressions in (69) can be convexified if $r_{i,k}$ is a convex function (Stubbs & Mehrotra, 1999). However, in the following we will not necessarily make the latter assumption. Let us finally point out that (68) and (69) do *not* lead to complementarity conditions, due to the absence of sign restrictions on the factors.

Example 2.1 (continued). For an illustration, we apply the problem formulation (P3) to the GDP problem (8)–(13). This leads to the following optimization problem:

$$\min_{x,y,z} (x_1-3)^2+(x_2-2)^2+b \tag{74}$$

$$\text{s.t.} \quad 0 \le x_1,\ x_2 \le 8, \tag{75}$$

$$y_1((x_1)^2+(x_2)^2-1) \le 0, \tag{76}$$

$$y_2((x_1-4)^2+(x_2-1)^2-1) \le 0, \tag{77}$$

$$y_3((x_1-2)^2+(x_2-4)^2-1) \le 0, \tag{78}$$

$$y_1(b-2)=0, \tag{79}$$

$$y_2(b-1)=0, \tag{80}$$

$$y_3(b-3)=0, \tag{81}$$

$$y_1+y_2+y_3 \ge 1, \tag{82}$$

$$\varphi_{\mathrm{FB}}(y_1,\ y_2+y_3)=0, \tag{83}$$

$$\varphi_{\mathrm{FB}}(y_2,\ y_1+y_3)=0, \tag{84}$$

$$\varphi_{\mathrm{FB}}(y_3,\ y_1+y_2)=0. \tag{85}$$

Instead of φ_{FB}, we could also use φ_{NR} or other *NCP*-functions here.

To circumvent the introduction of nonconvexity into the model by binary multiplication, we present an alternative, convex reformulation approach on the basis of tailored big-M constraints in Section 5.4, which can also be used in conjunction with exact continuous variables as defined in Eq. (58). Before the alternative model reformulation on the basis of big-M constraints will be shown, we shortly elaborate another distinct property of the binary multiplication-based model formulation.

5.3.1. The case of inconsistent equalities

In many applications, the constraints (68)–(71) in (P3) lead to implicit restrictions on the exact continuous variables. In particular, the Eq. (68) as well as (70) have to hold

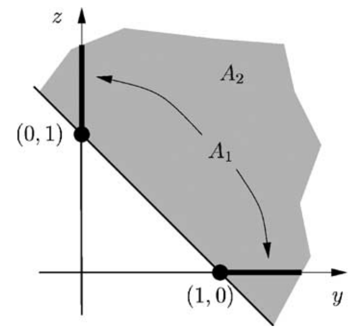

Fig. 9. A two-dimensional feasible set for (y, z).

simultaneously for $i \in D_k$, $k \in K$. In process engineering applications, the underlying equations (i.e. $h_{i,k}(x)=0$, $i \in D_k$ as well as $b_k-\gamma_{i,k}=0$, $i \in D_k$) are often inconsistent for fixed $k \in K$, i.e. they do not admit a common solution or, put geometrically, the sets described by these equations are disjoint. Note that this is an inherent property of a GDP problem with so-called disjoint disjunctions which have non-empty intersecting feasible regions (Vecchietti, Lee, & Grossmann, 2003). This is particularly the case, if for fixed $k \in K$ the values $\gamma_{i,k}$ are pairwise distinct for $i \in D_k$. This implies that *at most* one of the variables $y_{i,k}, i \in D_k$, is non-vanishing. For example, in (79)–(81) from the reformulation of Example 2.1, at most one of the variables y_1, y_2, y_3 can be non-zero. Consequently, the Eqs. (83)–(85) become redundant.

In the case $n_k=2$ with $y=y_1$ and $z=y_2$ this means that the equation $yz=0$ holds automatically. As a consequence, the only constraint needed for the description of A_1 (cf. Fig. 7) is $y+z \ge 1$, i.e. the set

$$A_2=\{(y,z) \in \mathbb{R}^2 | y+z \ge 1\}$$

coincides with A_1 in the case of inconsistent equalities (cf. Fig. 9).

Although the pair (y, z) does not vary in the complete two-dimensional set A_2 from Fig. 9, in the restrictions one does not code the same information twice. This can be expected to lead to better numerical performance when NLP solvers are applied.

Example 2.1 (continued). Since in Example 2.1 the equations for the investment cost (79)–(81) are inconsistent, the constraints (83)–(85) become redundant. Thus, these constraints do not have to be enforced in this case. In fact, the numerical results with the binary multiplication reformulation of the GDP problem (8)–(13) show that *no* extra constraints are required that define a set A_1 due to the inconsistency of the constraints (79)–(81), which are related to the investment cost of the process network. The optimization problem (74)–(82) was solved with the NLP solver SNOPT in again less than a CPU second on the same SUN computer used before. However, the nonconvex reformulation (74)–(82) introduces various locally optimal points into the optimization problem, which originally had a unique optimal solution.

Consequently, the usage of local NLP solvers, such as, e.g. SNOPT, will in general be highly sensitive to the optimization problem initialization. This fact is confirmed by the numerical results we obtained for the example problem, i.e. the objective function values 8.789 and 4.528 were found as local optima in addition to the globally optimal point 1.172, depending on how the problem was initialized. The required computing time to find these local minima was shorter than the time used to solve the same problem on the basis of the big-M model and the set A_μ. This is explained by the fact that the NLP problem is solved only once due to the inconsistency property of the disjunctive constraints, which inherently form a set A_1 for a reformulation according to (74)–(82).

5.4. A problem reformulation based on tailored big-M constraints

In this section, we present the reformulation of a disjunctive optimization problem using tailored big-M constraints, which also allow the use of exact continuous variables. With this reformulation approach, we avoid the introduction of nonconvexities into the model since no binary multiplication is applied to transform the disjunctions.

The big-M approach used to model problem (P2) will not work in the case in which the discrete variables are to be replaced by exact continuous variables described by the set A_1. This is due to the negation of $y_{i,k}$, which cannot be expressed by $1 - y_{i,k}$, but instead by the sum $\sum_{j\in D_k\setminus\{i\}} y_{j,k}$. Moreover, the value of b_k cannot be determined as simply as in (18). Hence, we suggest the following problem reformulation:

$$\min_{x,y}\ \Phi(x) + \sum_{k\in K} b_k \tag{P4}$$

$$\text{s.t.}\quad f(x) = 0, \tag{86}$$

$$g(x) \le 0, \tag{87}$$

$$-M_{i,k}\left(\sum_{j\in D_k\setminus\{i\}} y_{j,k}\right) \le h_{i,k}(x) \le M_{i,k}\left(\sum_{j\in D_k\setminus\{i\}} y_{j,k}\right), \tag{88}$$

$$r_{i,k}(x) \le M_{i,k}\left(\sum_{j\in D_k\setminus\{i\}} y_{j,k}\right), \tag{89}$$

$$-M_{i,k}\left(\sum_{j\in D_k\setminus\{i\}} y_{j,k}\right) \le b_k - \gamma_{i,k} \le M_{i,k}\left(\sum_{j\in D_k\setminus\{i\}} y_{j,k}\right), \tag{90}$$

$$\bar{A}y \le \bar{a}, \tag{91}$$

$$\sum_{i\in D_k} y_{i,k} \ge 1,\ k \in K, \tag{92}$$

$$\varphi_{\mathrm{FB}}\left(y_{i,k}, \sum_{j\in D_k\setminus\{i\}} y_{j,k}\right) = 0, \quad i \in D_k, k \in K, \tag{93}$$

In comparison to (P3), a major advantage of (P4) can be seen in the transformation of the disjunctive constraints (cf. (88)–(90)). In fact, Eq. (89) is convex, if $r_{i,k}$ is a convex function in the process variables x. On the other hand, the problem formulation (P4) has two drawbacks, which have to be considered as well. Firstly, the big-M constants only ensure that the right hand side of Eq. (89) has a bounded value, if the values $y_{i,k}$ are restricted by an appropriate upper bound. Secondly, disjunctive optimization problems comprising disjoint sets of equations cannot be exploited as easily as is the case in conjunction with model (P3). Thus, the constraints (92) and (93) that form the set A_1 cannot be simplified as shown in the previous section, where the condition (92) has been sufficient.

Example 2.1 (continued). The big-M reformulation according to (P4) of Example 2.1 takes the following form:

$$\min_{x,y}\ (x_1 - 3)^2 + (x_2 - 2)^2 + b \tag{94}$$

$$\text{s.t.}\quad 0 \le x_1,\ x_2 \le 8, \tag{95}$$

$$(x_1)^2 + (x_2)^2 - 1 \le M(y_2 + y_3), \tag{96}$$

$$(x_1 - 4)^2 + (x_2 - 1)^2 - 1 \le M(y_1 + y_3), \tag{97}$$

$$(x_1 - 2)^2 + (x_2 - 4)^2 - 1 \le M(y_1 + y_2), \tag{98}$$

$$-M(y_2 + y_3) \le b - 2 \le M(y_2 + y_3), \tag{99}$$

$$-M(y_1 + y_3) \le b - 1 \le M(y_1 + y_3), \tag{100}$$

$$-M(y_1 + y_2) \le b - 3 \le M(y_1 + y_2), \tag{101}$$

$$\varphi_{\mathrm{FB}}(y_1, y_2 + y_3) = 0, \tag{102}$$

$$\varphi_{\mathrm{FB}}(y_2, y_1 + y_3) = 0, \tag{103}$$

$$\varphi_{\mathrm{FB}}(y_3, y_1 + y_2) = 0, \tag{104}$$

$$y_1 + y_2 + y_3 \ge 1. \tag{105}$$

Note that, although the disjunctive constraints of this problem comprise inconsistent equalities, we have to explicitly state the Fischer–Burmeister constraints (102)–(104) when using the tailored big-M model (94)–(101). The numerical results obtained with the NLP solvers SNOPT and CONOPT show that the tailored big-M model performs well. From a set of arbitrarily chosen starting points we are able to find at least a locally optimal solution. Although the model formulation (95)–(101) is now convex, we still suffer from the fact that the constraints (102)–(104) are nonconvex.

6. Illustrative example problems

In this section, the continuous reformulation of disjunctive optimization problems and the corresponding solution techniques are illustrated by means of two example problems. The example of a process network optimization is considered first to show how the proposed methods can be applied to problems with propositional logic constraints. Moreover, we present an integrated design and control problem for a distillation process involving a nonconvex dynamic process model to show that the proposed algorithms are also capable of solving disjunctive or mixed-integer dynamic optimization problems.

6.1. Optimization of a process network

This example problem taken from Kocis and Grossmann (1987) treats the optimization of a process network. The model formulations presented here are based on a disjunctive problem formulation proposed by Türkay and Grossmann (1996). The disjunctive process model is not stated here explicitly due to space limitations. It is important to note that this problem can be solved globally with any of the solution methods reviewed by Grossmann (2002) since the objective function is linear and the equality constraints relax as convex inequalities. As already indicated, the guarantee to locate a global solution cannot be given when applying any of the proposed continuous reformulation techniques in conjunction with a local NLP method.

In the following, we present how the process network optimization problem can be formulated as a continuous optimization problem with discrete decision variables replaced by either *approximate* or *exact* continuous variables. At first, we take a closer look at the reformulation using *approximate* continuous variables. The optimization problems discussed in the following were implemented in GAMS and solved with the NLP solver SNOPT.

6.1.1. Reformulation by approximate continuous variables and solution

We represent the discrete decision variables by the complementarity condition introduced in Section 4.1 as well as by the circle condition proposed in Section 4.2. The disjunctive process model is transformed into a mixed-integer problem using big-M constraints according to (P2):

$$\min_{x,y} \Phi := c_1 + c_2 + c_3 + x_4 + 1.8\,x_1 + 1.2\,x_5 + 7\,x_6 - 11\,x_8 \tag{106}$$

$$\text{s.t.} \quad x_1 - x_2 - x_3 = 0, \tag{107}$$

$$x_7 - x_4 - x_5 - x_6 \leq 0, \tag{108}$$

$$x_5 \leq 5, x_8 \leq 1, \tag{109}$$

$$\begin{aligned} &-M(1-y_1) \leq x_8 - 0.9\,x_7 \leq M(1-y_1),\\ &x_7 \leq M\,y_1,\\ &x_8 \leq M\,y_1,\\ &c_1 = 3.5\,y_1, \end{aligned} \tag{110}$$

$$\begin{aligned} &-M(1-y_2) \leq x_4 - \ln(1+x_2) \leq M(1-y_2),\\ &x_2 \leq M\,y_2,\\ &x_4 \leq M\,y_2,\\ &c_2 = 1\,y_2, \end{aligned} \tag{111}$$

$$\begin{aligned} &-M(1-y_3) \leq x_5 - 1.2\,\ln(1+x_3) \leq M(1-y_3),\\ &x_3 \leq M\,y_3,\\ &x_5 \leq M\,y_3,\\ &c_3 = 1.5\,y_3, \end{aligned} \tag{112}$$

$$\begin{aligned} &y_1 - y_2 \geq 0,\\ &y_1 - y_3 \geq 0,\\ &y_2 + y_3 \leq 1,\\ &y_1 - y_2 \leq 1,\\ &y_1 - y_3 \leq 1. \end{aligned} \tag{113}$$

$$y_i \in \{0,1\}, \quad i = 1,2,3, \tag{114}$$

The big-M constant M was set to 10 for all calculations. The optimal solution of this problem has an objective function value $\Phi = -1.9231$ with binary variables values $y = [1,0,1]$ as reported by Türkay and Grossmann (1996). This result is confirmed by solving the optimization problem with a branch & bound (SBB) and an outer approximation (DICOPT++) method, where y_1, y_2 and y_3 are treated as discrete variables.

The discrete variables are defined as continuous variables $y_i \in [0,1]$, $i = 1,2,3$, for the reformulation approaches based on the complementarity as well as the circle constraints. Consequently, the constraints (114) are replaced by the complementarity constraints (38)–(40) or the constraints related to the circle condition (52)–(54), respectively. The complementarity condition is regularized with a parameter μ as stated in Eq. (41). We choose $\mu = 0.3$ as initial value and solve a sequence of NLP problems where μ is steadily decreased with μ approaching 0. Note, that the set $A_{\mu=0.3}$, as any other set $A_{\mu>0.25}$, leads to a fully continuous relaxation of all binary variables according to $0 \leq y_i \leq 1, i = 1,2,3$. Consequently, the NLP solver finds the relaxed optimal value $\Phi_\mu = -10.65$ with $y_\mu = [0.1,0,0]$. As soon as $\mu \leq 0.25$ the relaxation of y_i is restricted to continuous values for which $y_i + z_i = 1$ *and* $y_i z_i \leq \mu, i = 1,2,3$ do hold, i.e. the line segment is divided into two disconnected pieces as shown in Fig. 4. The optimal objective function value $\Phi_\mu = 0$ with integral values $y = [0,0,0]$ of problem (106)–(114), which is finally found when μ approaches 0, is a local minimizer only. The reason for this behavior is explained by the fact that the values of y at the relaxed optimal value are very close to those belonging to the local minimum found here. In fact, various locally optimal points are introduced into the optimization problem due to the nonconvexity of the complementarity conditions.

The same suboptimal point is obtained when the circle condition is applied to represent a set A_ν for the variables y. In this case, we start the first optimization with $\nu =$

$1/\sqrt{2}$, which yields the same relaxed solution as above: $\Phi_\nu = -10.65$ and $y_\nu = [0.1, 0, 0]$. Subsequently, ν is driven to 0 until $y_0 = [0, 0, 0]$. Again, the nonconvex constraints representing the set A_ν are responsible for the failure of the algorithm to locate the global solution. The computing time required to solve the problem to local optimality was less than 1 CPU second.

6.1.2. Reformulation by exact continuous variables and solution

Exact continuous variables can be used to replace the discrete decision variables y of the network optimization problem in conjunction with a modification of the big-M problem formulation (106)–(114). In particular, we employ the tailored big-M approach proposed in Section 5.4. This model is equivalent to the disjunctive formulation for variables y defined on a one-dimensional set A_1. Hence, the model reads as

$$\min_{x,y} \Phi := c_1 + c_2 + c_3 + x_4 + 1.8\,x_1 + 1.2\,x_5 + 7\,x_6 - 11\,x_8 \tag{115}$$

$$\text{s.t.}\quad \text{Eqs.(107)–(109)}, \tag{116}$$

$$\begin{aligned} &-M\,z_1 \le x_8 - 0.9\,x_7 \le M\,z_1,\\ &x_7 \le M\,y_1,\\ &x_8 \le M\,y_1,\\ &-M\,z_1 \le c_1 - 3.5 \le M\,z_1,\\ &-M\,y_1 \le c_1 \le M\,y_1, \end{aligned} \tag{117}$$

$$\begin{aligned} &-M\,z_2 \le x_4 - \ln(1+x_2) \le M\,z_2,\\ &x_2 \le M\,y_2,\\ &x_4 \le M\,y_2,\\ &-M\,z_2 \le c_2 - 1 \le M\,z_2,\\ &-M\,y_2 \le c_2 \le M\,y_2, \end{aligned} \tag{118}$$

$$\begin{aligned} &-M\,z_3 \le x_5 - 1.2\,\ln(1+x_3) \le M\,z_3,\\ &x_3 \le M\,y_3,\\ &x_5 \le M\,y_3,\\ &-M\,z_3 \le c_3 - 1.5 \le M\,z_3,\\ &-M\,y_3 \le c_3 \le M\,y_3, \end{aligned} \tag{119}$$

$$\begin{aligned} &z_2 + y_1 \ge 1,\\ &z_3 + y_1 \ge 1,\\ &z_2 + z_3 \ge 1,\\ &z_1 + y_2 + z_2 + y_3 \ge 1,\\ &z_1 + y_3 + z_3 + y_2 \ge 1, \end{aligned} \tag{120}$$

$$\varphi_{\text{FB}}(y_i, z_i) = 0, \quad i = 1, 2, 3. \tag{121}$$

$$y_i + z_i \ge 1, \quad i = 1, 2, 3. \tag{122}$$

where z_i, $i = 1, 2, 3$, represent the negation of y_i, $i = 1, 2, 3$, in the three two-term disjunctions. Note that the relation $z_i = 1 - y_i$ does not hold here.

The optimization problem formulation (115)–(122) is again implemented in GAMS and solved using SNOPT. We do not focus on extensive studies and comparisons of the computing time in conjunction with this example problem. However, since we do not have to solve a series of NLP problems for which a parameter is driven to zero, the time to solve the problem ((115)–(122) is significantly shorter than for the problem based on approximate continuous variables. More importantly, the nonconvex optimization problem formulation (115)–(122) is again highly sensitive to the problem initialization. For this reason, we carried out various optimization runs with 10 different sets of initial values for the variables y and z. The MATLAB function *rand* was used to generate random numbers for y between 0 and 1. In order to make sure that the constraint (122) is satisfied the initial values for z are assigned with $z = 1 - y$. In 4 out of 10 cases SNOPT finds the global optimal solution $\Phi = -1.9231$, whereas in 6 cases the local solution $\Phi = -1.7210$ is found.

Alternatively it is possible to employ the binary multiplication model in conjunction with exact continuous variables. According to (P3) the model can be stated as:

$$\min_{x,y} \Phi := c_1 + c_2 + c_3 + x_4 + 1.8\,x_1 + 1.2\,x_5 + 7\,x_6 - 11\,x_8 \tag{123}$$

$$\text{s.t.}\quad \text{Eqs. (107)–(109)}, \tag{124}$$

$$\begin{aligned} &y_1(x_8 - 0.9\,x_7) = 0,\\ &z_1 x_7 = 0,\\ &z_1 x_8 = 0,\\ &y_1(c_1 - 3.5) = 0,\\ &z_1 c_1 = 0, \end{aligned} \tag{125}$$

$$\begin{aligned} &y_2(x_4 - \ln(1+x_2)) = 0,\\ &z_2 x_2 = 0,\\ &z_2 x_4 = 0,\\ &y_2(c_2 - 1) = 0,\\ &z_2 c_2 = 0, \end{aligned} \tag{126}$$

$$\begin{aligned} &y_3\,(x_5 - 1.2\ln(1+x_3)) = 0,\\ &z_3\,x_3 = 0,\\ &z_3\,x_5 = 0,\\ &y_3\,(c_3 - 1.5) = 0,\\ &z_3\,c_3 = 0, \end{aligned} \tag{127}$$

$$\begin{aligned} &z_2 + y_1 \ge 1,\\ &z_3 + y_1 \ge 1,\\ &z_2 + z_3 \ge 1,\\ &z_1 + y_2 + z_2 + y_3 \ge 1,\\ &z_1 + y_3 + z_3 + y_2 \ge 1, \end{aligned} \tag{128}$$

$$y_i + z_i \ge 1, \quad i = 1, 2, 3. \tag{129}$$

Note that for the binary multiplication reformulation of this example problem there is no need to employ an NCP function (cf. (60)). This is different for the tailored big-M reformulation (cf. (121)). The corresponding equations are redundant due to the inconsistency of the reformulated equalities associated with the cost terms c_i of the process units. Furthermore, no big-M constants have to be specified using the problem formulation (123)–(129).

The optimization problem (123)–(129) was implemented in GAMS and solved using the local NLP solver SNOPT. We again carried out various optimization runs with 10 different sets of initial values for the variables y and z. The same random numbers that were already generated for the preceding problem formulation were used here. Similar to the results obtained for the reformulation based on tailored big-M constraints, SNOPT finds in 4 out of 10 cases the globally optimal solution $\Phi := -1.9231$, whereas in 6 cases the local solution $\Phi := -1.7210$ is found.

6.2. *Integrated design and control of a binary distillation column*

The second example problem treats the integrated design and control of a binary distillation column that is subject to a disturbance in the feed composition. The process model is described by a set of differential-algebraic equations (DAEs). A similar example problem has already been proposed before (Bansal et al., 2002; Schweiger & Floudas, 1997). Here, a binary mixture is separated using a distillation column with $N = 30$ trays where the feed tray location, the vapor stream and the reflux ratio are the decision variables.

The objective of the optimization problem is to minimize the integral squared error (ISE) of the top and bottom composition deviations from their respective set points after the disturbance occurred. At $t_\mathrm{s} = 400\,\mathrm{min}$ of operation, the feed composition jumps from $x_\mathrm{F} = 0.45$ to $x_\mathrm{F} = 0.54$. The decision variables vapor stream V and reflux ratio R are treated as time-invariant parameters whereas the feed tray location is modeled as a time-invariant discrete decision via multi-term disjunctions (cf. Eq. (143)) and the associated Boolean variables Y_k. We consider a total process duration of 800 min. The disjunctive optimization problem is formulated according to:

$$\min_{V,R,Y_k,k=1,\dots,N} \Phi := \mathrm{ISE}(t_\mathrm{f}) \tag{130}$$

$$\frac{\mathrm{d\,ISE}}{\mathrm{d}t} = 100[(x_0 - 0.02)^2 + (x_{N+1} - 0.98)^2] \tag{131}$$

$$\frac{\mathrm{d\,ISE}}{\mathrm{d}t} = 100\left(\frac{1}{\pi}\arctan\left[(t - t_\mathrm{s}) \times 10^3\right] + \frac{1}{2}\right), \tag{132}$$

$$10n\frac{\mathrm{d}x_0}{\mathrm{d}t} = L_1x_1 - Vy_0 - Bx_0, \tag{133}$$

$$n\frac{\mathrm{d}x_k}{\mathrm{d}t} = L_{k+1}x_{k+1} - L_kx_k + V(y_{k-1} - y_k) + F_kz_\mathrm{F}, \quad k = 1, \dots, N, \tag{134}$$

$$10n\frac{\mathrm{d}x_{N+1}}{\mathrm{d}t} = V(y_N - x_{N+1}), \tag{135}$$

$$0 = L_1 - V - B, \tag{136}$$

$$0 = L_{k+1} - L_k + F_k, \quad k = 1, \dots, N, \tag{137}$$

$$0 = V - D - R, \tag{138}$$

$$y_k = \frac{\alpha x_k}{1 + (\alpha - 1)x_k}, \quad k = 1, \dots, N, \tag{139}$$

$$z_\mathrm{F} = 0.54 + 0.09\left(\frac{1}{\pi}\arctan\left[(t - t_\mathrm{s})\right] + \frac{1}{2}\right), \tag{140}$$

$$x_{N+1}(t_\mathrm{f}) \geq 0.98, \tag{141}$$

$$x_0(t_\mathrm{f}) \leq 0.02, \tag{142}$$

$$\bigvee \begin{bmatrix} Y_k \\ F_k = 1\,\mathrm{kmol} \\ F_i = 0 \\ i \in D \setminus \{k\},\ D = 1, \dots, N \end{bmatrix}, \tag{143}$$

$$Y_k = \mathrm{False}, \quad k = 1, \dots, 10, \tag{144}$$

$$Y_k \in \{\mathrm{True}, \mathrm{False}\}, \quad k = 1, \dots, N, \tag{145}$$

$$t \in [0, t_\mathrm{s} + 400\,\mathrm{min}].$$

The time-invariant tray holdup n is assigned to 0.175 kmol, the bottom and top holdups are specified as $10n$. The constant relative volatility is set to $\alpha = 2.5$. Furthermore, the initial values for the differential state variables are specified as $x_k(t_0) = 0.45, k = 1, \dots, N$, and $\mathrm{ISE}(t_0) = 0.0$. The constraint defined in Eq. (144) is used to enforce a feed tray position that is above tray 10.

The dynamic model was implemented in gPROMS (2002). The optimization with the different model reformulations presented in the following was carried out using DyOS (2002), a software tool for dynamic optimization that interfaces modeling environments compliant to a CAPE-OPEN interface specification by Keeping and Pantelides (1999) to solution algorithms for numerical optimization. DyOS implements a control-vector parameterization approach (Kraft, 1985), which converts the infinite-dimensional optimization problem into an NLP problem. The arising small and dense NLP problems are solved with NPSOL (Gill, Murray, Saunders, & Wright, 1986), a solver which is well suited for this problem class. Alternatively, SNOPT can be used as NLP solver in the DyOS framework.

6.2.1. *Reformulation by approximate continuous variables and solution*

The disjunctions (143) can be represented by the following set of big-M constraints:

$$0 \leq F_k \leq 1y_k, \tag{146}$$

$$\sum_{k=11}^{30} y_k = 1, \quad (147)$$

$$y_k = 0, \quad k = 1, \ldots, 10, \quad (148)$$

where binary variables $y_k \in \{0, 1\}$ are used instead of Boolean variables $Y_k \in \{\text{True}, \text{False}\}$.

As in the example problems treated before, the discrete decision variables y_k are relaxed continuously and enforced to take discrete values 0 or 1 by either imposing complementarity constraints or the circle condition.

The formulation based on complementarity constraints is analyzed first. The complementarity condition is regularized with a parameter μ as stated in Eq. (41). We choose $\mu = 0.3$ as initial value and solve a sequence of NLP problems where μ is steadily decreased until $\mu = 0$. The first solution with fully relaxed variables y_k ($\mu = 0.3$) yields a lower bound to the objective function value of $\Phi_\mu = 17.4804$ and 3 active feed trays, i.e. $y_{11} = 0.0738$, $y_{24} = 0.3926$ and $y_{25} = 0.5336$. The corresponding optimal values for the vapor stream and the reflux ratio are $V = 1.4868$ kmol/min and $R = 0.9468$, respectively. Integral values for the variables y_k together with the constraint enforcing a single feed tray, is found for μ approaching 0. The optimal value for the objective is $\Phi_0 = 17.4986$ with $y_{25} = 1$, $V = 1.5425$ kmol/min and $R = 1.0023$. The corresponding liquid composition profiles on various trays are shown in Fig. 10. By means of explicit enumeration of all possible feed trays, it was confirmed that the solution found with the formulation based on complementarity constraints is globally optimal under the assumption that each dynamic optimization problem with a fixed and single feed tray location was solved to global optimality. Altogether, 9 optimization runs were carried out with decreasing values for μ.

For the first 5 runs, the relaxation parameter μ was decreased with steps $\Delta\mu = 0.05$. In the vicinity of $\mu = 0$, smaller steps had to be chosen to ensure convergence of the NLP solver NPSOL, i.e. $\mu = 0.01, 0.001, 0.0$. A total number of 64 NLP iterations and an accumulated CPU-time of 29 min was required to solve the integrated design and control problem on a 1.5 GHz PC. Interestingly, when replacing the complementarity constraints by an NCP-function (see Jiang & Ralph, 2000 for more information), like, e.g. the Fischer–Burmeister function φ_{FB}, no numerical problems were observed for small values of the regularization parameter μ. In fact, the numerical results were even more favorable in terms of the total computing time (22 min) and the number of NLP iterations (58) using the NCP-function replacing the complementarity constraints.

The same optimal result is obtained using the circle condition instead of the complementarity formulation. We again start with the discrete decision variables being continuously relaxed. A lower bound for the integral squared error is found, which is identical to the one obtained with the fully relaxed complementarity formulation. The optimal solution is obtained after 5 successive optimization runs, for which ν was set to $1/\sqrt{2} - r_i$ with values of 0.1, 0.2, 0.3, 0.4 and 0.5 for r_i^2, respectively. The optimization took an accumulated number of 71 iterations and a CPU time of 29 min on a 1.5 GHz PC. When compared to the solution approach using complementarity constraints, no numerical problems were observed for the circle approach with r_i being close to r_a. In fact, the difference $\Delta r_i = \sqrt{0.1}$ between two subsequent iterations could be kept constant without facing any numerical difficulties. The behavior of the variables y_k indicating the feed tray location is depicted in Fig. 11. A continuous relaxation of all binary decision variables ($r_i^2 = 0.0$) leads to three active feed trays 11, 24 and 25. With an increasing value of r_i^2, tray 25 becomes more and more dominant. For $r_i^2 = 0.5$, we eventually have tray 25 as the single feed tray of the distillation column.

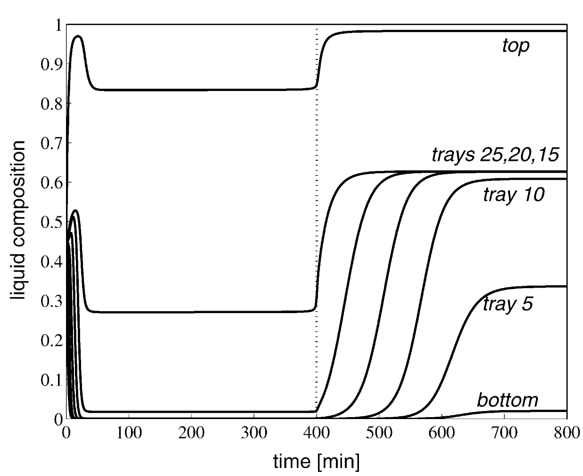

Fig. 10. Top, bottom and selected tray compositions.

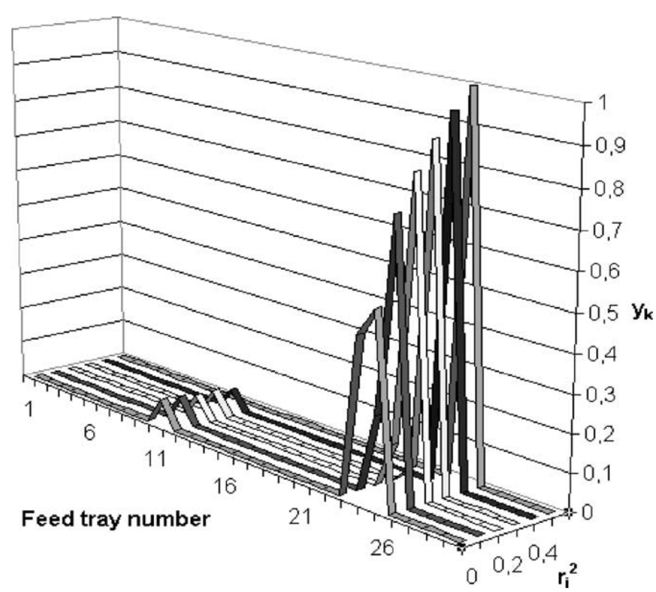

Fig. 11. From a continuous relaxed to binary solution for the feed tray location variable y_k.

6.2.2. Reformulation by exact continuous variables and solution

Alternatively, the approach using exact continuous variables can also be used to solve the integrated design and control problem. The optimization problem formulation has to be modified according to (P4). In particular, tailored big-M constraints are employed to model the discrete decision where to locate the feed using variables y_k:

$$-M\left(\sum_{j\in D\setminus\{k\}} y_j\right) \le F_k - 1 \le M\left(\sum_{j\in D\setminus\{k\}} y_j\right),$$
$$k = 11, \ldots, 30, \tag{149}$$

$$-M\, y_k \le F_k \le M\, y_k, \quad k = 11, \ldots, 30, \tag{150}$$

$$y_k = 0, \quad k = 1, \ldots, 10. \tag{151}$$

The set on which the variables $y_k \in \{0\} \cup [1, \infty)$ are defined is expressed by the Fischer–Burmeister function φ_{FB} as explained in the previous section.

The optimization problem was solved using the NLP solver NPSOL. The solution with continuously relaxed variables $y_k \in [0, 1]$, as used for the first iteration of the approximate approaches, is taken as initialization. This point was obtained in conjunction with the tailored big-M model (132)–(142) and (149)–(151) together with the constraint (147). The optimal solution, i.e. feed tray 25 is obtained in eight NLP iterations and 24.8 min of CPU-time.

In order to analyze the impact of the initialization on the solution obtained with the exact reformulation approach, a set of 10 different, randomly generated values between 0 and 1 are taken as initialization for the variables y_k. The result of these optimization runs yields that the initialization has again a large impact on the solution that will be obtained. In particular, feed trays 12–16 and 18 are found as locally optimal solutions with objective function values ranging from 19.3 to 21.1. The globally optimal feed tray location $y_{25} = 1$ is not found in any of these optimization runs.

7. Conclusions

In this contribution, we have shown how disjunctive and mixed-integer nonlinear optimization problems can be solved to local optimality using algorithms designed for purely continuous optimization problems. The discrete decisions, which are usually expressed in terms of binary variables, are here modeled by continuous variables together with an appropriate set of constraints representing the discrete nature of the problem.

The idea to represent discrete decisions by approximate continuous variables has already been proposed by Raghunathan and Biegler (2003). The complementarity constraints employed by these authors lead to a violation of the so-called Mangasarian–Fromovitz constraint qualification, a fact that can cause numerical problems in conjunction with NLP solution techniques. Although tailored SQP and IP methods are able to solve these problems successfully (cf. Leyffer, 2003), we find it more convenient to reformulate discrete decisions in a *non-degenerate* way in the first place, so that any off-the-shelf NLP solver may be applied to the reformulated problem.

As one such possibility, the discrete decision variables can be modeled using circle constraints with better numerical properties since the linear independence constraint qualification is satisfied everywhere in the feasible set. It is shown that a circle condition leads to a continuous approximation of discrete decision variables which is equivalent to the description by the complementarity constraints. Both approaches based on approximate continuous variables can be applied to standard mixed-integer model representations of disjunctive optimization problems, such as models based on big-M constraints or on a convex hull representation of the disjunctions.

Exact continuous variables can, however, also be used to represent discrete decisions if the disjunctive process model is transformed appropriately. Two different model reformulations, a tailored big-M reformulation and a model based on binary multiplication, are proposed which are equivalent to the disjunctive problem in conjunction with exact continuous variables. The resulting nonlinear optimization problem is favorable due to the fact that the constraint qualifications are satisfied and the discrete decisions are represented by exact rather than approximate continuous variables.

The example problems presented in this paper illustrate how the different solution methods can be applied to a broad class of process engineering problems involving discrete and continuous decision variables. Although it is possible to efficiently solve disjunctive optimization problems using the methods proposed in this paper, we can only expect to find local solutions. This is explained by the fact that any continuous reformulation of a disjunctive optimization problem leads to a nonconvex optimization problem. Consequently, global optimization algorithms should be applied whenever the problem size admits to do so.

Acknowledgements

The financial support from Deutsche Forschungsgemeinschaft under the Gottfried Wilhelm Leibniz program is gratefully acknowledged. Additionally, we would like to thank Achim Küpper for his contribution to the example problem of Section 6.2 within the scope of his diploma thesis.

References

Avraam, M. P., Shah, N., & Pantelides, C. C. (1998). Modelling and optimisation of general hybrid systems in the continuous time domain. *Computers and Chemical Engineering, 22*, S221–S228.

Bansal, V., Perkins, J. D., & Pistikopoulos, E. N. (2002). A case study in simultaneous design and control using rigorous, mixed-integer dynamic optimization models. *Industrial and Engineering Chemistry Research, 41*(4), 760–778.

Bazaraa, M., Sherali, H., & Shetty, C. (1993). *Nonlinear programming*. New York: Wiley.

Brooke, A., Kendrick, D., Meeraus, A., & Raman, R. (1998). GAMS—A user's guide. Technical report. GAMS Development Corporation, Washington, USA.

Chen, B., Chen, X., & Kanzow, C. (2000). A penalized Fischer–Burmeister NCP-function. *Mathematical Programming, 88*, 211–216.

Drud, A. (1995). A system for large scale nonlinear optimization, tutorial for CONOPT subroutine library. Technical report, ARKI Consulting and Development A/S, Bagsvaerd, Denmark.

DyOS (2002). DyOS user manual, Release 2.1. Lehrstuhl für Prozesstechnik, RWTH Aachen, Aachen, Germany.

Floudas, C. A. (1995). *Nonlinear and mixed-integer optimization: Fundamentals and applications*. New York: Oxford University Press.

Gill, P. E., Murray, W., & Saunders, M. A. (1998). SNOPT: An SQP algorithm for large-scale constrained optimization. Technical report, Stanford University, Stanford, USA.

Gill, P. E., Murray, W., Saunders, M. A., & Wright, M. H. (1986). User's guide for NPSOL, Version 4.0. Systems Optimization Laboratory, Stanford University, Stanford, USA.

gPROMS (2002). gPROMS user guide, Release 2.1.1. Process Systems Enterprise Ltd., London, UK.

Grossmann, I. E. (2002). Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and Engineering, 3*, 227–252.

Grossmann, I. E., & Hooker, J. (2000). Logic based approaches for mixed integer programming models and their application in process synthesis. In M. Malone, J. Trainham, & B. Carnahan (Eds.), *Foundations of computer-aided process design* (Vol. 323, pp. 70–83). AIChE Symp. Series. CACHE Publications.

Jiang, H., & Ralph, D. (2000). Smooth SQP methods for mathematical programs with nonlinear complementary constraints. *SIAM Journal on Optimization, 10*(3), 779–808.

Jongen, H. T., & Weber, G.-W. (1991). Nonlinear optimization: Characterization of structural stability. *Journal of Global Optimization, 1*, 47–64.

Keeping, B., & Pantelides, C. C. (1999). Cape Open WP4.1 numerical solvers open interface specification draft (pp. 357–369). CO-NUMR-EL-03.

Kocis, G. R., & Grossmann, I. E. (1987). Relaxation strategy for the structural optimization of process flow sheets. *Industrial Engineering and Chemical Research, 26*, 1869–1880.

Kraft, D. (1985). On converting optimal control problems into nonlinear programming problems. *Computational Mathematical Programming*, NATO ASI Series, Vol. F15, Springer-Verlag, Berlin, Heidelberg, pp. 261–280.

Lee, S., & Grossmann, I. E. (2000). New algorithms for nonlinear generalized disjunctive programming. *Computers and Chemical Engineering, 24*, 2125–2141.

Leyffer, S. (2003). Complementarity constraints as nonlinear equations: Theory and numerical experiences. Argonne National Laboratory Preprint ANL/MCS-P1054-0603.

Luo, Z., Pang, J. & Ralph, D. (1996). *Mathematical programs with equilibrium constraints*. Cambridge: Cambridge University Press.

Oldenburg, J., Marquardt, W., Heinz, D., & Leineweber, D. B. (2003). Mixed-logic dynamic optimization applied to batch distillation process design. *AIChE Journal, 49*(11), 2900–2917.

Raghunathan, A., & Biegler, L. (2003). Mathematical programs with equilibrium constraints (MPEC) in process engineering. *Computers and Chemical Engineering, 27*(10), 1381–1392.

Raman, R., & Grossmann, I. E. (1994). Modelling and computational techniques for logic based integer programming. *Computers and Chemical Engineering, 18*(7), 563–578.

Robinson, S. (1976). Stability theory for systems of inequalities. Part II. differentiable nonlinear systems. *SIAM Journal on Numerical Analysis, 13*, 497–513.

Scheel, H., & Scholtes, S. (2000). Mathematical programs with complementarity constraints: stationarity, optimality, and sensitivity. *Mathematics of Operations Research, 25*, 1–22.

Schweiger, C. A., & Floudas, C. A. (1997). Interaction of design and control: Optimization with dynamic models. In W. Hager, & P. Paradolos (Eds.), *Optimal control: Theory, algorithms, and applications* (pp. 388–435). Dordrecht: Kluwer Academic Publishers.

Sharif, M., Shah, N., & Pantelides, C. C. (1998). On the design of multicomponent batch distillation columns. *Computers and Chemical Engineering S22*, 69–76, suppl.

Stubbs, R., & Mehrotra, S. (1999). A branch-and-cut method for 0-1 mixed convex programming. *Mathematical Programming, 86*, 515–532.

Türkay, M., & Grossmann, I. E. (1996). Logic-based MINLP algorithms for the optimal synthesis of process networks. *Computers and Chemical Engineering, 20*(8), 959–978.

Vecchietti, A., Lee, S., & Grossmann, I. E. (2003). Modeling of discrete/continuous optimization problems: characterization and formulation of disjunctions and their relaxations. *Computers and Chemical Engineering, 27*(3), 433–448.

Williams, H. P. (1999). *Model building in mathematical programming*. Chichester: Wiley.