AN EFFICIENT BRANCH AND BOUND ALGORITHM FOR STATE ESTIMATION AND CONTROL OF HYBRID SYSTEMS

A. Bemporad, D. Mignone, M. Morari

Institut für Automatik
ETH Swiss Federal Institute of Technology
CH-8092 Zürich, Switzerland
tel.+41-1-632 7626 fax +41-1-632 1211
{bemporad,mignone,morari}@aut.ee.ethz.ch
http://www.control.ethz.ch

Keywords : Hybrid Systems, Moving Horizon Techniques, Branch and Bound

Abstract

This paper presents a new Branch and Bound tree exploring strategy for solving Mixed Integer Quadratic Programs (MIQP) involving time evolutions of linear hybrid systems. In particular, we refer to the Mixed Logical Dynamical (MLD) models introduced by Bemporad and Morari (1999), where the hybrid system is described by linear equations/inequalities involving continuous and integer variables. For the optimizations required by the controller synthesis and state estimation of MLD systems, the proposed algorithm reduces the average number of node explorations during the search of a global minimum. It also provides good local minima after a short number of steps of the Branch and Bound procedure.

1 Introduction

Bemporad and Morari (1999) and Bemporad *et al.* (1999*b*) proposed a framework for modeling, control, and state estimation/fault detection of Mixed Logical Dynamical (MLD) systems, a class of hybrid systems described by linear equations and inequalities involving integer and continuous variables. The key idea of their approach is to transform propositional logic into mixed integer linear inequalities (Bemporad and Morari 1999, Williams 1993, Cavalier *et al.* 1990, Raman and Grossmann 1992). MLD systems generalize a broad number of important classes of systems like piecewise linear systems, systems with mixed discrete/continuous inputs and states, and many more (Bemporad and Morari 1999). The general MLD form is:

$$x(t+1) = Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) \tag{1a}$$
$$y(t) = Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t) \tag{1b}$$
$$E_2\delta(t) + E_3z(t) \leq E_1u(t) + E_4x(t) + E_5 \tag{1c}$$

where x are the continuous and binary states, y the continuous and binary outputs, u the continuous and binary inputs, δ and z represent binary and continuous auxiliary variables. The latter are introduced when translating logic propositions into linear inequalities. All constraints are summarized in the inequality (1c). The description (1) only appears to be linear, as the variables δ are constrained to be binary.

For feedback control, Bemporad and Morari (1999) propose a Model Predictive Control (MPC) scheme which is able to stabilize MLD systems on desired reference trajectories while fulfilling operating constraints, and possibly take into account qualitative knowledge in the form of heuristic rules. The algorithm requires the solution of a *Mixed Integer Quadratic Programming* (MIQP) problem (Lazimy 1985, Roschchin *et al.* 1987) at each time step.

For the dual problem of state estimation and fault detection, Bemporad *et al.* (1999*b*) define a Moving Horizon Estimation problem. This consists of solving at each time step a least squares estimation problem over a finite horizon backwards from the current time. The resulting optimization problem is again an MIQP.

With the exception of particular structures, mixed-integer programming problems are classified as $\mathcal{NP}$-complete, which means that in the worst case, the solution time grows exponentially with the problem size (Raman and Grossmann 1991). Despite this combinatorial nature, several algorithmic approaches have been proposed and applied successfully to medium and large size application problems (Fletcher and Leyffer 1995), the four major ones being: *Cutting plane* methods, where new constraints ("cuts") are generated and added to reduce the feasible domain until a 0-1 optimal solution is found; *decomposition* methods, where the mathematical structure of the models is exploited via variable partitioning, duality, and relaxation methods; *Logic-based* methods, where disjunctive constraints or symbolic inference techniques are utilized which can be expressed in terms of binary variables; *branch and bound* (B&B) methods, where the 0-1 combinations are explored through a binary tree, the feasible region is partitioned into sub-domains systematically, and valid upper and lower bounds are generated at different levels of the binary tree. For MIQP problems, Fletcher and Leyffer (1995) mention Generalized Benders' Decomposition (Lazimy 1985), Outer Approximation, LP/QP based branch and bound, and B&B as the major solvers. See Roschchin *et al.* (1987) for a review of these methods. Several authors agree on the fact that B&B methods are the most successful for mixed integer quadratic programs (Fletcher and Leyffer 1995).

Devising particular B&B algorithms that exploit the problem structure can be useful to reduce the amount of computations on average. We will measure the complexity of a solution method for an MIQP by the number of relaxed QPs (Quadratic Programs) that have to be solved during B&B. This is only a rough measure of complexity since it does not take into account the complexity of the single QPs.

The approach presented here belongs to the class of B&B methods. It is suitable for optimal control/estimation problems for MLD systems where the time sequence of binary variables switches rarely over the time horizon. In cases where the computational time is limited, the approach chooses to solve those QPs first that are most promising to deliver a good suboptimal solution to the MIQP.

In Sect. 2 some standard facts about MIQPs are reviewed. In Sect. 3 the new algorithm is presented. In Sect. 4 we estimate the computational complexity of the algorithm. A few remarks about the implementation of the method are given in Sect. 5. A simulation example is shown in Sect. 6.

2 B&B Algorithms for MIQP

A Mixed Integer Quadratic Program (MIQP) has the following form

$$\begin{aligned} \min_{x} \quad & x^T Q x + b^T x && (2)\\ \text{subject to} \quad & Cx + d \leq 0 \\ & x = \left[\begin{smallmatrix} x_c \\ x_d \end{smallmatrix}\right],\ x_c \in \mathbb{R}^{n_c} \\ & x_d \in \{0,1\}^{n_d} && (3) \end{aligned}$$

and differs from a standard QP through the integrality constraint (3)[1].

[1]In a more general setup any integer value is allowed, but we restrict it to the 0-1 case here.

The idea of solving MIQPs with B&B methods relies on the *relaxation* of the integrality constraints (3), i.e. integer variables are allowed to span over the whole continuous interval $[0, 1]$. We shall refer to a relaxed problem as a *subproblem.* The optimal values of the subproblems, if they exist, represent lower bounds on the optimal value of the original MIQP (Fletcher and Leyffer 1995). A graphical representation of the concepts relaxation and separation in B&B algorithms can be drawn with the help of k-ary trees. Fig. 1 depicts a binary tree.

We recall here some standard concepts from tree data structure terminology. A *tree* consists of *nodes* and *branches.* Exactly one node of a tree is characterized as the *root.* Each node except the root has a unique *father*, i.e. a unique predecessor. Each node including the root can have none, one or more subsequent nodes, called the *children* of the node. Nodes without children are called *leaves.* A tree, where each node except the leaves has exactly k children is called a *k-ary tree.* The *depth* of a node is the number of its predecessors towards the root, i.e. the number of its 'fathers' and 'grandfathers'. The *q-th level* of a tree is the set of all nodes with depth q. The depth of the root is 0. The *length* of a tree is the maximum depth over all its nodes. The tree obtained from a node ν by deleting the branch to its father and taking ν as root of a smaller tree is the *subtree of node ν.*

Let ξ be a vector having the dimension n_d and let the symbol $\star$ mean that the corresponding entry of ξ is relaxed, i.e. free to span the interval $[0, 1]$. We associate the original MIQP without integrality constraints (3) with

$$\xi_0 = \underbrace{[\star\ ,\ \ \star\ \ ,\ \ \ldots\ \ ,\ \star]}_{n_d \quad times} \tag{4}$$

The vector ξ_0 will be assigned to the root of a k-ary tree. The separation of the original MIQP or any subproblem into relaxed QPs is done by setting selected integer variables to 0 or 1. The resulting new QP problems are assigned to the children of the node. We denote each child by a vector ξ_j, $\xi_j \in \{\star, 0, 1\}^{n_d}$. If the i-th component $\xi_j^i = 0$ (or $\xi_j^i = 1$), then the QP corresponding to that node is solved by setting the i-th binary variable to 0 (or 1). If $\xi_j^i = \star$, then the i-th binary variable of ξ_j is regarded as free within $[0, 1]$. As an example, consider an MIQP with 3 binary variables. The corresponding binary tree is shown in Fig. 1.

2.1 Strategies for B&B

The MIQP tree can be explored in several ways. The choice of the problem separation and the order in which the subproblems are considered, influences the average computational effort. A good B&B algorithm aims at quickly fathoming entire subtrees (Floudas 1995), avoiding the solution of many subproblems. Two choices determine the sequence of subproblems: (i) the *branching rule*, and (ii) the *tree exploring strategy.*

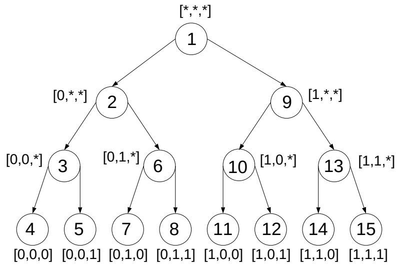

Figure 1: The binary tree for a MIQP with 3 integer variables. Each node is marked with the corresponding vector ξ_j. The numbers denote the order how the problems are solved in the depth first strategy.

The branching rule selects the next variable to branch on, and consequently the way of separating the problems. Some possible branching rules are:

First Free Variable: Among the relaxed integer variables, choose the one with the smallest index.

Maximum Fractional Part: By solving the relaxed QP of the father node, the solution for the variables that should be binary will instead usually have a fractional part. Choose the variable that has the largest distance to the nearest integer value as the next branching variable, i.e. the variable with the index j, where $j = \arg\max_i(\min\{\delta_i, 1-\delta_i\})$.

Once the variable to branch on has been selected by the branching rule, the tree exploring strategy (or node selection) determines the order in which the relaxed problems must be solved. Standard strategies are:

Depth First Strategy: The QPs are solved by a last-in first-out (LIFO) rule.

Breadth First Strategy: The problems at depth N are not solved before all problems at depths $N-1$ have been solved.

An example of depth first strategy is given in Fig. 1.

3 The Outside First Tree Exploring Strategy

Here we present an algorithm especially tailored to optimal control or estimation problems for MLD systems (1). In this case, the integer optimization vector x_d in (3) contains samples of the auxiliary variables $\delta(t)$ and/or binary inputs $u(t)$ taken at different time instants, e.g. $x_d = [\delta(t), \delta(t+1), \ldots, \delta(t+T-1)]$. The MIQPs we are considering arise from problems as described in (Bemporad and Morari 1999, Bemporad *et al.* 1999*b*).

The main motivation for the algorithm stems from observing that binary variables $\delta(t)$ seldom change their value over the time horizon T. In fact, typically binary variables $\delta(t)$ are associated with conditions on continuous states $x(t)$, for instance $[\delta(t)=1] \leftrightarrow [x(t) \geq 0]$. Because the continuous components satisfy dynamic equations, in general, their inertia will prevent frequent switches of the indicator variable $\delta(t)$, e.g. during transients. This phenomenon is even more pronounced when integer variables represent the occurrence of a fault (Bemporad *et al.* 1999*b*) which involves an irreversible physical damage, because in this case the integer variable will switch at most once its value over the horizon $[t-T, t]$. When we suspect that the system exhibits this kind of behaviour, we should try to solve first the QPs where the integer variables do indeed describe a limited number of switches.

column -1 original problem	column 0 no guaranteed switches	column 1 1 guaranteed switch	column 2 2 guaranteed switches
$[\star,\star,\star]$	$[0,0,0]$	$[0,0,1]$	$[0,1,0]$
	$[0,0,\star]$	$[0,1,1]$	$[1,0,1]$
	$[0,\star,\star]$	$[1,1,0]$	
	$[1,1,1]$	$[1,0,0]$	
	$[1,1,\star]$	$[0,1,\star]$	
	$[1,\star,\star]$	$[1,0,\star]$	

Table 1: Classification of subproblems according to guaranteed switches in the binary variables for $n_d = 3$

3.1 Guaranteed Switches

Given a subproblem marked by ξ, let I be the ordered m-tuple collecting the indices i for which $\xi^i \neq \star$,

$$I \triangleq [i_1, i_2, \ldots i_m] \quad \text{such that} \quad \xi^{i_j} \neq \star,\ \forall j = 1 \ldots m$$

In other words, I collects the indices of the variables that are already fixed. We define the number D of *guaranteed switches* as the number of indices i_q in I such that $\xi^{i_q} \neq \xi^{i_{q+1}}$. For instance, if $\xi = [0,1,1,1,\star,\star,1,0,0,\star]$ then $I = [1,2,3,4,7,8,9]$, and $D = 2$. By considering again the problem of an MIQP with 3 binary variables we can partition the 15 QPs occurring in the B&B method into four classes, according to the guaranteed switches of binary variables in each QP. This classification is given in Table 1. We use the name "guaranteed switches", because we count only the switches in the sequence of fixed integer variables. The root is assigned by definition -1 guaranteed switches. The proposed B&B algorithm solves a QP in column i of Table 1 only after all problems in the columns up to $i-1$ have been solved. In receding horizon control for MLD systems as described by Bemporad and Morari (1999) at time t the optimal solution for the optimization at time $t-1$ is known. The concept of guaranteed switches can be modified to count instead the number of differences of the fixed variables to this optimal sequence (Bemporad *et al.* 1999*a*).

According to this rule, the tree with 3 binary variables of the example above is explored in the order denoted in Fig. 2. We call the strategy *outside first*, since the MIQP tree is explored from the outside to the inside.

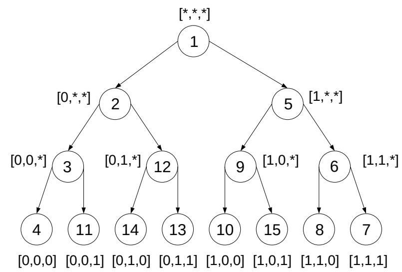

Figure 2: Order how problems are solved in the outside first strategy, assuming 'first free variable' branching rule.

3.2 The Algorithm

We adapt the general B&B algorithm structure (see e.g. (Floudas 1995)) to the outside first strategy, to obtain the following algorithm:

1. Take the original MIQP, relax all integrality constraints, mark the relaxed QP with its number of guaranteed switches, i.e. -1. Set $f_{opt} = \infty$, $k_c = -1$, $x_{opt} = [\infty, \dots, \infty]$ and initialize with the relaxed QP the list of problems to be solved.

2. If the list of problems is empty, terminate and output f_{opt}, x_{opt}.

3. If there are problems on the list marked by k_c, select one of them, remove it from the list, and solve it. If the QP is feasible, denote its cost by $f_\star$ and its solution by $x_\star$. Go to step 5. If the QP is infeasible[2], go to 2.

4. If there are no problems on the list marked by k_c, increase k_c by 1 and go to 2.

5. *Fathoming by worse cost:* If $f_\star \geq f_{opt}$, then go to 2.

6. *Integer feasibility:* If $f_\star < f_{opt}$ and $x_\star$ satisfies the integrality constraints, then set $f_{opt} = f_\star$ and $x_{opt} = x_\star$. Go to 2.

7. *Feasibility, but not integer feasibility* Separate the problem. Mark the subproblems by the number of guaranteed switches in the fixed integer variables. Add the subproblems to the list of problems. Go to 3.

Often hard restrictions on the time available to perform the online computations severely limit the chances to find a global minimizer for (2)–(3), especially for a large problem with many binary variables. In this case, the MIQP optimization should aim at providing good suboptimal solutions. With the outside first approach a k_{max} can be defined and only those problems are solved that have a number of guaranteed switches smaller than k_{max}. Alternatively the number of QPs can be limited.

[2] All the children problems will not be solved, i.e. the subtree is fathomed by infeasibility

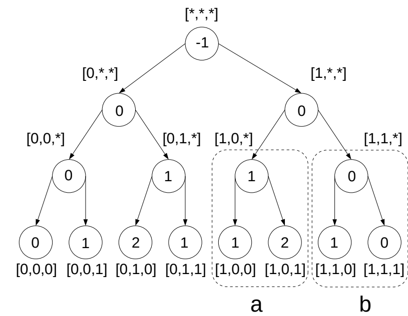

Figure 3: Number of switches for each subproblem in the outside first tree. Note the symmetry in the number of guaranteed switches of subtrees at the same depth.

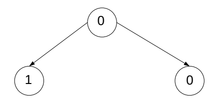

Figure 4: Tree of depth 1 with a root of 0 switches.

The outside first approach allows to select and solve those QPs first that are most promising in giving good suboptimal solutions, provided that the integer variables do not switch their value often. This is the case for the optimal control/estimation problems at hand, where local minima are also tolerable. In fact, Bemporad and Morari (1999) prove that stability of the model predictive control algorithm for MLD systems is not altered by local minima, though the convergence properties of the controller deteriorate. The same considerations hold for the estimator presented by Bemporad *et al.* (1999*b*).

4 Complexity of the Outside First B&B Algorithm

In this section we analyze the computational complexity associated with the outside first tree exploring strategy. In particular, we determine the worst case number of QPs in the outside first approach with a limited number of switches k_{max}.

4.1 Nodes with Equal Number of Switches

Given a full binary tree A_n of length n and a root with -1 switches, denote by $M_n(k)$ the number of nodes having an equal number k of guaranteed switches. The same quantity for a subtree having a root with 0 switches is denoted by $\tilde{M}_n(k)$.

In Fig. 3 we have marked the number of guaranteed switches for an MIQP tree with 3 integer variables. To simplify the nomenclature we will denote by *node with k switches* the node corresponding to the relaxed QP with k guaranteed switches in the integer variables.

Lemma 1 *Assume that the root of A_n has 0 switches. Then*

$$\tilde{M}_n(k) = \binom{n+1}{k+1}, \ \forall k = 0, \ldots, n \tag{5}$$

Proof. We prove (5) by induction on n. For $n = 1$, it is easy to verify that $\tilde{M}_1(0) = 2$, $\tilde{M}_1(1) = 1$ (see Fig. 4). Assume now, that (5) holds. By induction on the tree length n, it holds that

$$\tilde{M}_{n+1}(k) = \begin{cases} \tilde{M}_n(k) + \tilde{M}_n(k-1) & (k = 1 \ldots n+1) \\ \tilde{M}_n(0) + 1 & (k = 0) \end{cases} \tag{6}$$

Eq. (6) states that the number of nodes with k switches in a tree of length $n+1$ is the sum of the nodes with k switches in one of the main subtrees of length n and the nodes with $k-1$ switches in a subtree of length n. The latter subtree experiences an increase of the number of switches in each node, as soon as it gets adopted, i.e. it gets a new father, cfr. subtree a in Fig. 3. It is now straightforward to verify that:

$$\tilde{M}_n(k)+\tilde{M}_n(k-1) = \binom{n+1}{k+1} + \binom{n+1}{k} = \binom{n+2}{k+1}$$

which completes the proof. □

In Lemma 1 we have assumed that the root has 0 switches. The binary trees we want to consider in the MIQP context, however have by definition a root with -1 switches. By exploiting the symmetry, it is easy to prove the following theorem.

Theorem 1 *Assume that the root of A_n has -1 switches. Then:*

$$M_n(k) = 2\binom{n}{k+1} \quad \text{for} \quad k = 0 \ldots n-1 \tag{7}$$

Proof. The root has -1 switches. Therefore we have to count only the switches in the 2 subtrees of the root. Both subtrees have a root with 0 switches, and therefore by Lemma 1 $M_n(k) = 2\tilde{M}_{n-1}(k)$. □

4.2 Nodes with Less than k_{max} Switches

The number of nodes with less than k_{max} switches in a full binary tree of length n is denoted by

$$C_n(k_{max}) \ = \sum_{k=-1}^{k_{max}} M_n(k) \ = 1 + 2\sum_{k=0}^{k_{max}} \binom{n}{k+1} \tag{8}$$

Note that (8) is consistent with the total number of nodes in a full binary tree of length n, since:

$$C_n(n-1) \ = 1 + 2\sum_{k=0}^{n-1} \binom{n}{k+1} \ = 2^{n+1} - 1$$

In Table 2 we illustrate that the complexity of the outside first algorithm is polynomial in n for fixed k_{max}.

k_{max}	$C_n(k_{max})$	complexity
0	$1+2n$	$O(n)$
1	n^2+n+1	$O(n^2)$
2	$\frac{n^3}{3}+\frac{5}{3}n+1$	$O(n^3)$

Table 2: Number of nodes with less than k_{max} switches in a tree of length n

5 Implementation Scheme

The implementation of the depth first algorithm can be done with the use of a stack (last in, first out) data structure (Floudas 1995). The outside first algorithm can be implemented in a similar way using multiple stacks, rather than only one. Each of the stacks contains the subproblems having the same number of guaranteed switches. It can be verified that branching on one variable and using the 'first free variable' branching rule requires at each step of B&B at most two stacks (Bemporad *et al.* 1999*a*). Note that if δ is a vector, the switches must be counted among the same variables taken at subsequent time steps.

6 Example: The Two-Tank System

The tree exploring strategy proposed above has been used for the model predictive control of a two tank system, according to the control law presented by Bemporad and Morari (1999). We used a prediction horizon of $N_T = 5$ steps. The control synthesis requires the online solution of MIQPs with 15 binary variables. For a detailed description of the modeling of the tank system the reader is referred to Dolanc *et al.* (1997) and Bemporad *et al.* (1999*b*). We compared the outside first algorithm to the breadth first and the depth first tree exploring strategies, both used with the 'first free' node selection rule.

6.1 Finding the Global Optimum

If we search for the global optimum of the MIQP at each time step, clearly the three methods yield the same optimum. In Fig. 5 we see the trajectories of the liquid levels in the two tank system. The aim is to stabilize the levels at $h = [0.39, 0.13]$ starting from $[0.1, 0.2]$. In Fig. 6 we have plotted the number of QPs performed during the simulation in Fig. 5 for the three tree exploring strategies. In Table 3 we summarize the total number of QPs during the simulation. We see that in this particular case the outside first approach results in a smaller number of QPs than the other two approaches.

6.2 Limiting the Number of QPs

We consider again the model predictive control of the tank system. In addition we impose a restiction on the number of QPs that are allowed to be solved at

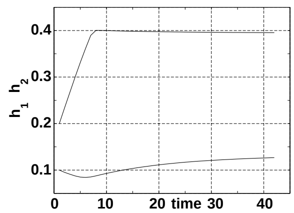

Figure 5: Trajectories of the states of the two tank system controlled by the MPC controller described by Bemporad and Morari (1999)

Approach	Total number of QPs
Depth First	19'829
Breadth First	21'007
Outside First	5181

Table 3: Data of the simulation in Fig. 5

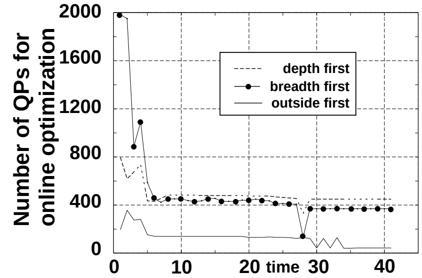

Figure 6: Number of QPs at each time step for the simulation in Fig. 5

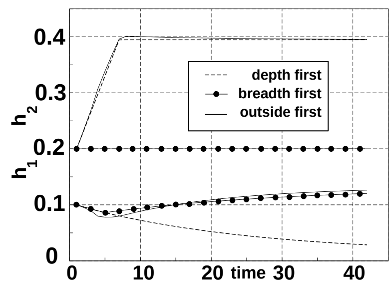

Figure 7: Trajectories of the states of the two tank system: limited number of 100 QPs at each step

each time step[3]. The computations are interrupted as soon as a maximum number of QPs has been solved and the currently best solution is returned as a possibly suboptimal value. This situation occurs if we want to meet hard time constraints on the computations. For the control problem described above the worst case number of QPs using B&B is $2^{16} - 1 = 65'535$. We limited the number of QPs to 100. In this case the results using the three tree exploring strategies differ noticeably, as seen in Fig. 7. When the number of QPs is limited, the outside first approach is the only one giving an acceptable steady state values for both states. We see that the breadth first approach is not able to guarantee set-point tracking for the first state, whereas depth first fails to control the second state.

7 Conclusions

We have presented a branch and bound method for solving Mixed Integer Quadratic Programs particularly suited for optimal control and estimation problems of mixed logical dynamical systems. The proposed tree exploring strategy chooses first subproblems which have a small number of switches in the pattern of discrete variables. When there are physical motivations for such a behaviour, the method shows a reduced average computational burden due to the fast fathoming of entire subtrees of the MIQP tree.

References

Bemporad, A. and M. Morari (1999). Control of Systems Integrating Logic, Dynamics, and Constraints. *Automatica* **35**(3), 407–427. `ftp://control.ethz.ch/pub/reports/postscript/AUT98-04.ps`.

Bemporad, A., D. Mignone and M. Morari (1999*a*). A New Branch and Bound Algorithm for Optimal Control Problems. Technical report. `ftp://control.ethz.ch/pub/reports/postscript/AUT99-03.ps` ETH Zurich.

Bemporad, A., D. Mignone and M. Morari (1999*b*). Moving Horizon Estimation for Hybrid Systems and Fault Detection. In: *Proceedings of the American Control Conference*.

Cavalier, T.M., P.M. Pardalos and A.L. Soyster (1990). Modeling and integer programming techniques applied to propositional calculus. *Computers Opns Res.* **17**(6), 561–570.

Dolanc, G., D. Juricic, A. Rakar, J. Petrovcic and D. Vrancic (1997). Three-tank Benchmark Test. Technical Report Copernicus Project Report CT94-02337. J. Stefan Institute. `ftp://ftp.utia.cas.cz/pub/staff/tesar/fault/latest/`.

Fletcher, R. and S. Leyffer (1995). Numerical experience with lower bounds for MIQP branch-and-bound. Technical report. University of Dundee, Dept. of Mathematics. Scotland, U.K. submitted to SIAM Journal on Optimization, `http://www.mcs.dundee.ac.uk:8080/~sleyffer/miqp_art.ps.Z`.

Floudas, C.A. (1995). *Nonlinear and Mixed-Integer Optimization*. Oxford University Press.

Lazimy, R. (1985). Improved algorithm for mixed-integer quadratic programs and a computational study. *Mathematical Programming* **32**, 100–113.

Raman, R. and I.E. Grossmann (1991). Relation between MILP modeling and logical inference for chemical process synthesis. *Computers and Chemical Engineering* **15**(2), 73–84.

Raman, R. and I.E. Grossmann (1992). Integration of logic and heuristic knowledge in MINLP optimization for process synthesis. *Computers and Chemical Engineering* **16**(3), 155–171.

Roschchin, V.A., O.V. Volkovich and I.V. Sergienko (1987). Models and methods of solution of quadratic integer programming problems. *Cybernetics* **23**, 289–305.

Williams, H.P. (1993). *Model Building in Mathematical Programming*. John Wiley & Sons. Third Edition.

[3]Limiting the number of QPs rather than the number of switches allows an easier comparison to the other tree exploring strategies