The Lagrangian Relaxation for the Combinatorial Integral Approximation Problem

Michael Jung[a]*, Gerhard Reinelt[a]† and Sebastian Sager[a]‡

[a] *Interdisciplinary Center for Scientific Computation, University of Heidelberg*

13 February 2012

Abstract

We are interested in methods to solve *mixed-integer nonlinear optimal control problems* (MIOCPs) constrained by ordinary differential equations and combinatorial constraints on some of the control functions. To solve these problems we use a *first discretize, then optimize* approach to get a specially structured *mixed-integer nonlinear program* (MINLP). We decompose this MINLP into an NLP and an MILP, which is called the *combinatorial integral approximation problem* (CIAP). Previous results guarantee an integer gap for the MINLP depending on the objective function value of the CIAP. The focus of this study is the analysis of the CIAP and of a tailored branch and bound method. We link the huge computational gains compared to commercial MILP solvers to an analysis of subproblems on the branching tree.

To this end we study properties of the Lagrangian of the CIAP. Special focus is given to special ordered set constraints that are present due to an outer convexification of the control problem. Also subproblems that arise by application of branch and bound schemes are of interest. We prove polynomial runtime of the algorithm for special cases and give numerical evidence for efficiency by means of a numerical benchmark problem.

Keywords: optimal control, MIOCP, integer programming, MILP, Lagrangian relaxation

1 Introduction

Mixed-integer nonlinear optimal control problems (MIOCPs) are interesting from an academic, but also from a practical point of view. Many control applications can be described by nonlinear differential equations. Often, integrality requirements apply. Such requirements stem from modeling, e.g., a control function that takes the value 1 if a communication between two robots takes place and 0, else. This allows to formulate practically relevant constraints, such as a minimum degree of communication per robot, (1). The other source of integrality requirements are practical restrictions, such as engines that need to operate at a minimum output rate or are coupled with on/off valves or the choice of one of several gears in a vehicle. An online benchmark library of MIOCPs with further references can be found at (13).

Many different approaches from control theory, hybrid systems, nonlinear optimization or mixed-integer linear programming have been proposed for this important problem class. In the

*Corresponding author.
Im Neuenheimer Feld 368, D-69120 Heidelberg,
Tel.: 0049-6221-548230,
Email: michael.jung@iwr.uni-heidelberg.de
†Email: gerhard.reinelt@informatik.uni-heidelberg.de
‡Email: sebastian.sager@iwr.uni-heidelberg.de

interest of avoiding redundancy we refer to (17; 16; 15) for recent literature surveys. In this work we focus on one particular approach that can be described as follows. First, one applies a *partial outer convexification*, which leads to a control function with linearly entering binary control functions. Second, one *discretizes*, i.e., one approximates the control functions by basis functions with finite support and replaces the path constraints by constraints on a finite time grid. Third, a relaxation of the resulting MINLP is solved. Fourth, based on the solution of this NLP binary controls are determined.

For the last step, in (17; 15) a *Sum Up Rounding* strategy has been proposed. It yields a trajectory with a priori given bound on the integer gap in linear time, (15). However, this solution may violate combinatorial constraints. Therefore in (16) a different approach has been proposed, based on the solution of an MILP. This MILP is called the *combinatorial integral approximation problem* (CIAP). The results in (15) guarantee an integer gap for the MINLP depending on the objective function value of the CIAP. Our focus is the analysis of a tailored branch and bound method for the CIAP. We explain the huge computational gains compared to commercial MILP solvers by showing the equivalence of subproblems on the branching tree with the Lagrangian of the CIAP. We also prove polynomial runtime of the algorithm for special cases and give numerical evidence for efficiency by means of a numerical benchmark problem.

Often important insight on a problem is gained by investigating the feasibility polytope. An investigation of min down/up polytopes, for example, can be found in (8). However, this approach was not very helpful for the CIAP, as the number of facets depends heavily on the solution of the relaxed MINLP, (16). Instead, we study the Lagrangian relaxation for the CIAP. The Lagrangian relaxation is one of the most used relaxations for MILPs. Its first application was the one-tree relaxation of the traveling salesman problem in the famous Held-Karp algorithm in (5; 6). The traditional application fields are variants of the knapsack problem like, e.g., facility location and capacity planning (12), general assignment, network flow and the unit commitment problem (11). The general approach is thoroughly explained in (4) and in (10). The first only illustrates the theory, while the latter also contains very helpful algorithmic insights. We directly solve the Lagrangian relaxation of the unconstrained CIAP and use this solution in our branch and bound algorithm.

The paper is organized as follows. In Section 2 we sum up the original MIOC problem formulation, the theoretical result that justifies the CIAP, and the tailored branch and bound algorithm. In Section 3 we obtain the main results of this paper. We analyze and show equivalence between Lagrangian relaxations and the problems solved in the proposed branch and bound scheme of Section 2.3. Also the polynomial runtime behavior is proven. Numerical evidence is provided by means of a benchmark problem in Section 8. Section 9 summarizes the main results of the paper.

2 Problem formulation

For convenience, we sum up the original and the outer convexification MIOC problem formulations in Section 2.1, the MILP and the theoretical result that justifies the CIAP in Section 2.2, and finally the tailored branch and bound algorithm in Section 2.3, as described in (16).

2.1 The mixed–integer optimal control problem

We are interested in finding optimal controls $u(\cdot)$ and $v(\cdot)$ for a system of ordinary differential equations on a given time horizon,

$$\dot{x}(t) = f(t, x(t), u(t), v(t)), \quad t \in [0, t_{\mathrm{f}}]. \tag{1}$$

The continuous controls $u(\cdot)$ are restricted to values from a connected set $\mathcal{U}$, whereas the *integer control functions* $v(\cdot)$ may only take values from a finite set,

$$v(t) \in \Omega := \{v^1, v^2, \ldots, v^{n_\omega}\}, \quad t \in [0, t_\mathrm{f}], \tag{2}$$

with $v^i \in \mathbb{R}^{n_v}$. Later on, we want to relax this problem for intermediate calculations by replacing the discrete set Ω by a convex set. It has been shown that taking the convex hull of Ω is not a good relaxation. It may result in ill-defined functions $f(\cdot)$ and artificial singular arcs, compare (17; 7). Superior is a *partial outer convexification*, an equivalent reformulation. We reformulate the ODE (1) to

$$\dot{x}(t) = \sum_{i=1}^{n_\omega} f(t, x(t), u(t), v^i)\, \omega_i(t), \quad t \in [0, t_\mathrm{f}] \tag{3}$$

with binary control functions

$$\omega(t) \in \{0,1\}^{n_\omega}, \quad t \in [0, t_\mathrm{f}], \tag{4a}$$

having the equivalence

$$\omega_i(t) \quad = \quad \begin{cases} 1, & \text{if } v(t) = v^i, \\ 0, & \text{if } v(t) \neq v^i \end{cases} \tag{4b}$$

in mind. To get a bijection between the original formulation and this new formulation, a *special ordered set of type 1* (SOS1) condition is imposed,

$$\sum_{i=1}^{n_\omega} \omega_i(t) = 1, \quad t \in [0, t_\mathrm{f}]. \tag{4c}$$

We are particularly interested in additional *combinatorial constraints* on the integer control functions. These constraints may have a form that cannot be represented by pointwise constraints like in (4a), but may restrict the control function in a more general way. An example is an upper bound on the total number of switches on the time horizon. To avoid the cumbersome notational work, we hide these constraints behind the generic restriction

$$\omega(\cdot) \in \Omega_c := \{\omega(\cdot) \text{ fulfills (4a), (4c) and combinatorial constraints}\}, \tag{4d}$$

where Ω_c is an eligible subset of $L^\infty([0, t_\mathrm{f}])$. The subscript c denotes the continuous version (almost everywhere on $[0, t_\mathrm{f}]$) and is later on replaced by a d, once we discretize and work on a finite time grid.

We are ready to formulate the MIOCP. We want to minimize a Mayer term

$$\min_{x,u,\omega} \Phi(x(t_\mathrm{f})) \tag{5a}$$

over the differential states $x(\cdot)$, the continuous controls $u(\cdot)$ and the binary controls $\omega(\cdot)$ subject to the ODE system (3) with fixed initial values

$$x(0) = x_0, \tag{5b}$$

nonlinear path and control constraints of the form

$$0 \leq c(x(t), u(t)), \quad t \in [0, t_\mathrm{f}], \tag{5c}$$

a feasible domain for the continuous controls

$$u(t) \in \mathcal{U}, \quad t \in [0, t_\mathrm{f}], \tag{5d}$$

and binary control functions $\omega(\cdot) \in \Omega_c$ that fulfill

Note that problemis linear in the binary control functions. Generalizations of this problem class are discussed in (14).

2.2 Combinatorial integral approximation

The general algorithm we propose can be described as follows.

1. We relax the integrality constraint on $\omega(\cdot)$ and obtain an OCP problem.
   We solve it with any suitable algorithm for (continuous) optimal control, e.g., with Bock's direct multiple shooting method (2; 9).
   We obtain a lower bound on the optimal solution of the original problem (15) and a relaxed solution for the binary controls denoted by $\alpha(\cdot)$.

2. We approximate the relaxed solution $\alpha(\cdot)$ of the OCP to obtain feasible binary controls $\omega(\cdot)$ by solving the *combinatorial integral approximation problem*, see below.

3. A simulation of the ODE with the feasible controls $\omega(\cdot)$ provides an upper bound for MIOCP (5). If the gap between lower and upper bound is too large, the control grid is adapted, compare (17).

The *combinatorial integral approximation problem* (CIAP) arises as a consequence of the following theorem from (15).

Theorem 1 (Integral Approximation Theorem)**.** *Let $x(\cdot)$ and $y(\cdot)$ be solutions of the initial value problems*

$$\dot{x}(t) = A(t, x(t)) \cdot \alpha(t), \quad x(0) = x_0, \tag{6a}$$
$$\dot{y}(t) = A(t, y(t)) \cdot \omega(t), \quad y(0) = y_0 \tag{6b}$$

with $t \in [0, t_\mathrm{f}]$, for given measurable functions $\alpha, \omega : [0, t_\mathrm{f}] \to [0,1]^{n_\omega}$ and a differentiable $A : \mathbb{R}^{n_x+1} \to \mathbb{R}^{n_x \times n_\omega}$. If positive numbers $C, L \in \mathbb{R}^+$ exist such that for $t \in [0, t_\mathrm{f}]$ almost everywhere it holds that

$$\left\| \frac{\mathrm{d}}{\mathrm{d}t} A(t, x(t)) \right\| \leq C, \tag{6c}$$
$$\| A(t, y(t)) - A(t, x(t)) \| \leq L \| y(t) - x(t) \|, \tag{6d}$$

and $A(\cdot, x(\cdot))$ is essentially bounded by $M \in \mathbb{R}^+$ on $[0, t_\mathrm{f}]$, and it exists $\epsilon \in \mathbb{R}^+$ such that for all $t \in [0, t_\mathrm{f}]$

$$\left\| \int_0^t \alpha(\tau) - \omega(\tau) \, \mathrm{d}\tau \right\| \leq \epsilon, \tag{6e}$$

then it also holds

$$\| y(t) - x(t) \| \leq ((M + Ct)\epsilon + \| y_0 - x_0 \|) \, e^{Lt} \tag{6f}$$

for all $t \in [0, t_\mathrm{f}]$.

The theorem states that if two controls $\alpha(\cdot)$, $\omega(\cdot)$ are close in the sense of (6e), then the corresponding states $x(\cdot)$ and $y(\cdot)$ are close in the sense of (6f). As objective and constraint functions are continuous functions of the differential states, this motivates to choose the binary control function $\omega(\cdot)$ such that for a given relaxed solution $\alpha(\cdot)$ the expression (6e) is minimized.

Let the binary controls $\omega(\cdot)$ be discretized on a time grid

$$t_j < t_{j+1}, t_0 = 0, t_{n_t} = t_\mathrm{f}, \quad \Delta t_j := t_{j+1} - t_j \tag{7}$$

with n_t time intervals. It makes sense to discretize the binary control by a piecewise constant function, described by $p \in \{0,1\}^{n_\omega \cdot n_t}$ such that $\omega_k(t) = p_{k,j}$ for all $t \in [t_j, t_{j+1})$. Let $q_{k,j}$ be the average of the relaxed control $\alpha_k(\cdot)$ on $[t_j, t_{j+1}]$, i.e.,

$$q_{k,j} := \frac{1}{\Delta t_j} \int_{t_j}^{t_{j+1}} \alpha_k(\tau) \, \mathrm{d}\tau.$$

Then minimizing (6e) with infinity norm is identical to the min-max problem

$$\min_{p} \max_{k,j} \left| \sum_{j=0}^{i} (p_{k,j} - q_{k,j}) \Delta t_j \right|.$$

This min-max formulation can be reformulated as an MILP,

$$\begin{aligned}
\min_{p,\eta} \quad & \eta \\
\text{s.t.} \quad & \eta \geq +\sum_{j=0}^{i} (p_{k,j} - q_{k,j}) \Delta t_j, \quad k = 1 \dots n_\omega, j = 0 \dots n_t - 1, \\
& \eta \geq -\sum_{j=0}^{i} (p_{k,j} - q_{k,j}) \Delta t_j, \quad k = 1 \dots n_\omega, j = 0 \dots n_t - 1, \qquad \text{(CIAP)} \\
& 1 = \sum_{k=1}^{n_\omega} p_{k,j}, \qquad\qquad\qquad\quad j = 0 \dots n_t - 1, \\
& p \in \Omega_d
\end{aligned}$$

with $\Omega_d \subseteq \{0,1\}^{n_\omega \cdot n_t}$ being the equivalent set to Ω_c from (4d) on the time grid (7). Note that it is the constraint

$$p \in \Omega_d \tag{8}$$

that is usually violated by solutions that are obtained with the *Sum Up Rounding* strategy (15) and the reason, why an MILP needs to be solved. As an example, the set

$$\Omega_d^\sigma := \left\{ p \in \{0,1\}^{n_\omega \cdot n_t} : \sum_{j=1}^{n_t - 1} |p_{k,j+1} - p_{k,j}| \leq \sigma_{k,\max}, \quad k = 1 \dots n_\omega \right\} \tag{9}$$

would set upper bounds $\sigma_{k,\max}$ for the number of switches on the time horizon. We look at different sets Ω_d in Section 3.

2.3 A tailored branch and bound algorithm

To solve problem CIAP, we propose a structure exploiting pure branch and bound algorithm. It uses the structure of the approximation inequalities from CIAP that model the min-max formulation. We branch on controls p in increasing order of the time index j in $p_{k,j}$ using SOS1-branching. This way, for a node of depth d, overall $n_\omega d$ variables are fixed, and all $2n_\omega d$ values

$$\eta_{k,i}^{\pm} := \pm \sum_{j=0}^{i} (p_{k,j} - q_{k,j}) \Delta t_j$$

for $i \leq d$ have been calculated. The natural lower bound

$$\eta \geq \max_{k, i \leq d} \{\eta_{k,i}^{\pm}\}$$

is a fast alternative to the solution of an LP relaxation. Often also the evaluation of the combinatorial constraints (8) can be performed efficiently. For example, for the set Ω_d^σ from (9) the number of switches up to node d can be updated and checked by n_ω simple additions.

As bound evaluation and feasibility check are usually very fast in comparison to solving an LP, we use *full strong branching*. The branching tree is then searched with standard techniques. As an alternative to a memory-intensive *best-first search* we consider a *dive and best* strategy to

prune large parts of the branching tree. It combines best-first search with the temporary usage of heuristics to find good solutions.

On each node of the branching tree we store the depth d of the node, the fixed control variables $p_{k,j}$ for all $j < d$, and the corresponding lower bound η of the objective function. Algorithm 1 gives an outline of the branch and bound scheme.

Algorithm 1 Combinatorial branch and bound (see (16))

Require: Time grid $\{t_j\}$, relaxed controls q, additional constrainted set $\Omega_d \in \{0,1\}^{n_\omega}$, e.g., SOS1-constraints.

Ensure: Optimal solution (η^*, p^*) of CIAP.

Initialize the B&B tree Q with an empty node (0, {}, 0.0).
Set best solution: $p^*_{k,j} \leftarrow 0$, $\eta^* \leftarrow \max_{k=1}^{n_\omega}\{\sum_{j=0}^{n_t-1} q_{k,j}\Delta t_j\}$.
while $Q \neq \emptyset$ **do**
  $a \leftarrow$ next node of Q given by search strategy.
  if $a.\eta > \eta^*$ **then** {Pruning step.}
    Prune node.
  else if $a.d = n_t$ **then**
    Mark a as currently optimal node.
    Set new best solution: $\eta^* \leftarrow a.\eta$, $p^* \leftarrow a.p$.
  else {Create child nodes, use strong branching.}
    for all possible permutations ϕ of $\{0,1\}^{n_\omega}$ **do**
      Create new node n with $n.d \leftarrow a.d + 1$, $n.p \leftarrow a.p$.
      Set $n.p_{k,d+1} \leftarrow \phi_k$.
      if $n.p$ fulfills the combinatorial constraints until time step $d+1$ **then**
        $n.\eta \leftarrow \max\left\{a.\eta, \max_{k=1}^{n_\omega}\{\pm\sum_{j=0}^{d}(p_{k,j} - q_{k,j})\Delta t_j\}\right\}$
        Add n into Q.
      end if
    end for
  end if
end while
return optimal solution (η^*, p^*).

3 Analysis of the Lagrangian of the CIAP

Algorithm 1 performs very well on numerical examples, compare (16) and Section 8. The way that feasibility and a lower bound are calculated is usually much faster than the solution of a LP relaxation. In this section, we are interested in the quality of the provided bound. We start with a short general introduction to Lagrangian relaxation, before we use it to analyze the quality of the provided bounds in the branching tree for different sets Ω_d in the following subsections. In Section 4.1 we consider the case $n_\omega = 1$ with one control function, in Section 7.1 the more important case in which the SOS1 constraint (4c) applies.

The Lagrangian relaxation is a popular relaxation for MILPs, e.g., (4; 10). The principle is to relax an MILP by dropping constraints and penalizing their violation in the objective function. Consider an integer program (IP)

$$\begin{aligned} \min_x \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & Dx \leq d \\ & x \text{ integer.} \end{aligned} \tag{10}$$

A Lagrangian relaxation of (10) is given by the IP

$$\begin{array}{ll} \min\limits_{x} & c^T x + \lambda^T (Ax - b) \\ \text{s.t.} & Dx \leq d \\ & x \text{ integer} \end{array} \tag{11}$$

for given non-negative Lagrangian multipliers $\lambda \in \mathbb{R}_0^+$.

Definition 4. *The Lagrangian function is defined as*

$$z_{LR}(\lambda) := \min_x \{c^T x + \lambda^T (Ax - b) \mid Dx \leq d,\ x \text{ integer}\}$$

and the Lagrangian problem is defined as the problem of finding the maximum value of the Lagrangian function for all feasible multipliers:

$$z_{LR} := \max_{\lambda} \{z_{LR}(\lambda) \mid \lambda \geq 0\}.$$

Lemma 2. *For the Lagrangian problem's solution z_{LR} and the value z_{LP} of the canonical LP-relaxation of* (10) *we have*

$$z_{LP} \leq z_{LR}.$$

These are well known properties and descriptions of the Lagrangian relaxation and are proved in standard MILP textbooks as mentioned above.

4.1 One-dimensional case

We start with the case $n_\omega = 1$ for which apparently no SOS1 constraint (4c) applies. We decide to penalize all constraints in (CIAP) that belong to the min-max-reformulation, leading to the Lagrangian function

$$z_{LR}(\lambda, \mu) = \min_{\eta \in \mathbb{R},\ p \in \Omega_d} L(\lambda, \mu, \eta, p)$$

with

$$\begin{aligned} L(\lambda, \mu, \eta, p) &= \eta + \sum_{i=0}^{n_t-1} \lambda_i \left(-\eta + \sum_{j=0}^{i} (p_j - q_j)\Delta t_j \right) \\ &\quad + \sum_{i=0}^{n_t-1} \mu_i \left(-\eta - \sum_{j=0}^{i} (p_j - q_j)\Delta t_j \right) \\ &= \eta \left(1 - \sum_{i=0}^{n_t-1} (\lambda_i + \mu_i) \right) + \sum_{i=0}^{n_t-1} (\lambda_i - \mu_i) \cdot \left(\sum_{j=0}^{i} (p_j - q_j)\Delta t_j \right). \end{aligned}$$

However, if $\sum_{i=0}^{n_t-1} \lambda_i + \mu_i \neq 1$, the function $L(\cdot)$ of $z_{LR}(\lambda, \mu)$ is unbounded due to the free variable η. Therefore, any solution of z_{LR} satisfies this condition – and thereby η can be eliminated from the problem:

$$\bar{z}_{LR}(\lambda, \mu) = \min_{p \in \Omega_d} \sum_{i=0}^{n_t} (\lambda_i - \mu_i) \cdot \left(\sum_{j=0}^{i} (p_j - q_j)\Delta t_j \right)$$

$$\begin{array}{rll} z_{LR} = & \max\limits_{\lambda, \mu \in \mathbb{R}_+^{n_t}} & \bar{z}_{LR}(\lambda, \mu) \\ & \text{s.t.} & \sum\limits_{i=0}^{n_t-1} \lambda_i + \mu_i = 1 \end{array}$$

Another way to interpret this, is as a relaxation of the original min-max-problem in which the maximum is relaxed to be the convex combination of its arguments instead. This convex combination is maximized with respect to the weights, hence its value is the same as the maximum individual value.

4.1.1 Root node of branching tree

We are interested in the behavior of the Lagrangian function when added as a lower bound in a branch and bound framework. We start by looking at the unrestricted case in the root node of the tree, reflected by

$$\Omega_d = \{0,1\}^{n_t}.$$

A trivial lower bound for the CIAP is 0. Unfortunately, no better bound is provided by Lagrangian relaxation.

Proposition 3. *If $\Omega_d = \{0,1\}^{n_t}$, then the value of the Lagrangian relaxation is*

$$z_{LR} = 0.$$

Proof. Reordering the terms in the Lagrangian function results in

$$\bar{z}_{LR}(\lambda,\mu) = \min_{p\in\{0,1\}^{n_t}} \sum_{i=0}^{n_t-1} (p_i - q_i)\Delta t_i \left(\sum_{j=i}^{n_t-1} \lambda_j - \mu_j \right)$$

If λ and μ are fixed to any feasible values, then the optimal values for p directly emerge, since there is no coupling between the different points in time:

$$p_i^* = \begin{cases} 0, & \text{if } \sum_{j=i}^{n_t-1} \lambda_j - \mu_j \geq 0, \\ 1, & \text{else.} \end{cases}$$

Dividing the index set $I = \{0,\ldots,n_t-1\}$ into two subsets accordingly, i.e., $G = \{i \in I \mid \sum_{j=i}^{n_t-1} \lambda_j - \mu_j \geq 0\}$ and $I\backslash G$, and using the optimal solution p^*, the Lagrangian function can be rewritten:

$$\bar{z}_{LR}(\lambda,\mu) = \sum_{i\in I\backslash G} \underbrace{(1-q_i)\Delta t_i \left(\sum_{j=i}^{n_t-1} \lambda_j - \mu_j \right)}_{\leq 0} - \sum_{i\in G} \underbrace{q_i \Delta t_i \left(\sum_{j=i}^{n_t-1} \lambda_j - \mu_j \right)}_{\geq 0} \leq 0.$$

The maximum value $z_{LR} = 0$ is taken, e.g., with the following feasible multipliers

$$\bar{\lambda}_0 = \bar{\mu}_0 = 0.5 \text{ and } \bar{\lambda}_i = \bar{\mu}_i = 0 \text{ for } i \neq 0.$$

□

4.1.2 Branching constraints

Deeper in the branching tree, the controls are fixed for some variables p_i, $i \in F \subseteq I$ due to branching. We show that either exactly one constraint of the type

$$\eta \geq \pm \sum_{j=0}^{i} (p_j - q_j)\Delta t_j$$

or exactly two of these constraints with different signs determine the value of the Lagrangian.

To identify the optimal solution of the Lagrangian relaxation of this node, some definitions are needed:

Definition 5. *We separate the complete time horizon in different time intervals* $[i_1, i_2]$ *and we define for each of these intervals the terms*

$$v_{i_1,i_2} := \sum_{i\in[i_1,i_2]\cap F}(p_i - q_i)\Delta t_i - \sum_{i\in[i_1,i_2]\setminus F} q_i\Delta t_i \tag{12}$$

$$\bar{v}_{i_1,i_2} := \sum_{i\in[i_1,i_2]\cap F}(p_i - q_i)\Delta t_i + \sum_{i\in[i_1,i_2]\setminus F} (1-q_i)\Delta t_i \tag{13}$$

The terms v *and* $\bar{v}$ *give the value of the integral approximation of a time interval* $[i_1, i_2]$ *during which all free controls are either* 0 *(in the case of* v*) or* 1 *(in the case of* $\bar{v}$*). We also need the following indices to state our results:*

- $i^* \in I$ *such that* v_{0,i^*} *is maximal,*
- $i_1^*, i_2^* \in I$ *such that* $v_{i_1^*,i_2^*}$ *is maximal,*
- $j^* \in I$ *such that* $-\bar{v}_{0,j^*}$ *is maximal,*
- $j_1^*, j_2^* \in I$ *such that* $-\bar{v}_{j_1^*,j_2^*}$ *is maximal.*

With these terms, we can directly give the value of the Lagrangian relaxation.

Proposition 4. *The value of the Lagrangian relaxation is*

$$z_{LR} = \max\left\{v_{0,i^*},\ -\bar{v}_{0,j^*},\ \frac{1}{2}v_{i_1^*,i_2^*},\ -\frac{1}{2}\bar{v}_{j_1^*,j_2^*}\right\}.$$

In the optimal solution, depending on which of the terms is the maximum, either $\lambda_{i^*} = 1$ *or* $\mu_{j^*} = 1$ *or* $\mu_{i_1^*} = \lambda_{i_2^*} = 0.5$ *or* $\lambda_{j_1^*} = \mu_{j_2^*} = 0.5$.

Proof. The outline of the proof is as follows. We interpret the problem as similar to a knapsack problem. We introduce variables to enhance readability of the procedure. Next, we explain the concept of phases that helps to understand the objective function of interest. As a result, four dominant phases naturally emerge as the phases which maximize this special objective in different circumstances.

For the Lagrangian relaxation we have the following min-max problem:

$$\begin{aligned} z_{LR} = \max_{\lambda,\mu\in\mathbb{R}_+^{n_t}} \quad & \min_{p\in\{0,1\}^{n_t}} \quad \sum_{i=0}^{n_t-1}(\lambda_i - \mu_i)\cdot\left(\sum_{j=0}^{i}(p_j - q_j)\Delta t_j\right) \\ \text{s.t.} \quad & p_i \text{ fixed for } i \in F, \\ & \sum_{i=0}^{n_t-1}\lambda_i + \mu_i = 1. \end{aligned}$$

We reformulate the Lagrangian relaxation using the solution of the inner minimization problem. This solution remains of the same structure as observed in the unrestricted case 4.1.1. The only difference is that we have to distinguish three cases for each variable p_j instead of only two. The additional case to be considered emerges when the variable p_j is fixed and not free. With the set

$$G = \{i \in I \mid \sum_{j=i}^{n_t-1}\lambda_j - \mu_j \geq 0\}$$

as in the Section 4.1.1 above, the problem takes the form

$$\begin{aligned} z_{LR} = \max_{\lambda,\mu\in\mathbb{R}_+^{n_t}} \quad & \sum_{i=0}^{n_t-1}\left(\sum_{j=i}^{n_t-1}\lambda_j - \mu_j\right)\Delta t_i \begin{Bmatrix} (p_i - q_i), & \text{if } i \in F \\ (-q_i), & \text{if } i \in G \\ (1-q_i), & \text{if } i \notin F\cup G \end{Bmatrix} \\ \text{s.t.} \quad & \sum_{i=0}^{n_t-1}\lambda_i + \mu_i = 1 \end{aligned}$$

Here, we can see the clear structure of a budget as the constraint and a special objective function, which uses the solution of the inner minimization problem.

Definition 6. *We introduce variables α and β to enhance readability:*

$$\begin{aligned}\alpha_i &:= \sum_{j=i}^{n_t-1} \lambda_j, \quad \alpha_{n_t} := 0 \\ \beta_i &:= \sum_{j=i}^{n_t-1} \mu_j, \quad \beta_{n_t} := 0.\end{aligned}$$

The old variables λ and μ mean a change in the new variables α and β. The restrictions that were posed on the multipliers λ and μ are easily translated into

$$\begin{aligned} 0 \leq \lambda_i &= \alpha_i - \alpha_{i+1}, \\ 0 \leq \mu_i &= \beta_i - \beta_{i+1}, \\ 1 = \sum_{i=0}^{n_t-1} \lambda_i + \mu_i &= \alpha_0 + \beta_0. \end{aligned} \tag{14}$$

With these variables, the objective of the Lagrangian function becomes

$$\tilde{z}_{LR}(\alpha,\beta) = \sum_{i=0}^{n_t-1} (\alpha_i - \beta_i)\Delta t_i \left\{ \begin{array}{ll} (p_i - q_i), & \text{if } i \in F, \\ (-q_i), & \text{if } i \notin F \text{ and } \alpha_i \geq \beta_i, \\ (1-q_i), & \text{if } i \notin F \text{ and } \alpha_i < \beta_i. \end{array} \right\} \tag{15}$$

Taking a closer look at these terms, the only possible positive summands are those with $i \in F$ and $sgn(\alpha_i - \beta_i) = sgn(p_i - q_i)$. The Lagrangian relaxation maximizes this function under the above stated restrictions on α and β. These restrictions can be described as a restriction on the initial sum of the factors α and β and both are non-increasing. As a direct result, the total change in α and β together is limited by 1. Therefore, one can interpret the knapsack constraint as a budget for the change in factors α and β. The aim is now to optimally use these changes from the initial factors whose sum is 1 to evade negative parts and to optimally use positive summands in the objective.

Since only the change in α and β is limited, we introduce a phase as an interval in time at whose initial border $\alpha_i - \beta_i$ increases by $\delta \in [-1, 1]$ and at whose ending border this change is reversed by adding $-\delta$. The value $v_{i,j}$ regarding the objective of a phase from time step i to j is

$$v_{i,j}(\alpha,\beta,\delta) = \delta \sum_{k=i}^{j} \Delta t_k \left\{ \begin{array}{ll} (p_k - q_k), & \text{if } k \in F, \\ (-q_k), & \text{if } k \notin F \text{ and } \alpha_k \geq \beta_k, \\ (1-q_k), & \text{if } k \notin F \text{ and } \alpha_k < \beta_k. \end{array} \right\}.$$

Each solution to the problem is composed of an overlapping of such phases. For starting phases, i.e., phases that begin at time step 0, the cost is only contributed by the change at the end, because for the beginning factors $\alpha_0 + \beta_0 = 1$ must hold, i.e., all of the budget is given there. For intermediate phases, the costs of the phase are contributed at both its start and its end.

We now consider a solution as an overlapping of phases with some special properties. We choose the phases with some properties such that we get a unique set of phases to each set of multipliers.

1. The most important part is that no phases with different signs of δ may overlap. This implies that by knowing the sign of a phase, we directly also know the sign of $\alpha - \beta$ during that phase, it is the same. We can hence determine the value of $v_{i,j}(\alpha,\beta,\delta)$ of a phase without knowing α and β and always the same choice is made in the inner brackets.

2. The second choice is such that we can directly sum the budget consumption as the sum over phase heights. Therefore, a phase has to be as wide as possible regarding time without changing its sign (e.g. δ_1 on the left-hand side of Figure 1 is correct, whereas δ_4 should be chosen wider on the right-hand side).

3. To get a unique set of phases we have to add the constraint that each phase is chosen as high as possible regarding $|\delta|$.

In Figure 1 these choices are made correctly on the left-hand side, whereas there are some mistakes on the right-hand side.

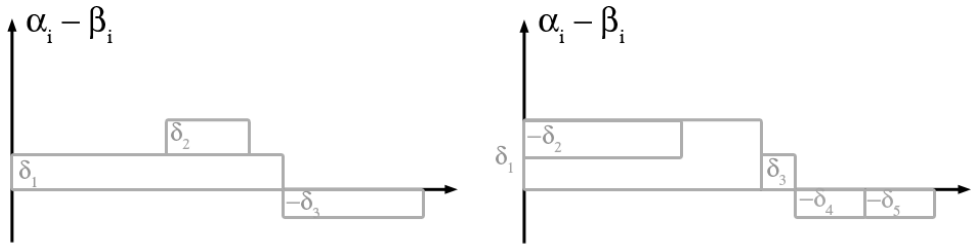

Figure 1: On the left-hand side the correct choices are made. On the right-hand side, e.g., the phase with value δ_1 must be made wider according to rule 2 and lower according to rule 3.

With these choices for the phases, let P_0 be the set of phases starting at time step 0 and P be the set of phases starting later, then

$$\sum_{p \in P_0} |\delta_p| + \sum_{p \in P} 2|\delta_p| = \alpha_0 - \beta_0 \leq 1, \tag{16}$$

because it is the sum over all changes in $\alpha - \beta$ and already the sum over all changes in α and β taken separately was at most 1.

A set of multipliers thus produces a unique set of phases $P(\alpha, \beta)$. Let for each phase p its beginning be specified by b_p and its end by e_p. With the definitions of v and $\bar{v}$, see (12) and (13), the objective value of such a solution (α, β) can directly be split into the values of the single phases, i.e.,

$$\tilde{z}_{LR}(\alpha, \beta) = \sum_{p \in P(\alpha,\beta), \delta_p > 0} \delta_p v_{b_p, e_p} + \sum_{p \in P(\alpha,\beta), \delta_p < 0} \delta_p \bar{v}_{b_p, e_p}.$$

With the indices i^*, j^*, etc. defined as above,

$$\begin{aligned}
\tilde{z}_{LR}(\alpha, \beta) &= \sum_{\substack{p \in P(\alpha,\beta) \\ \delta_p > 0}} \delta_p v_{b_p, e_p} + \sum_{\substack{p \in P(\alpha,\beta), \\ \delta_p < 0}} \delta_p \bar{v}_{b_p, e_p} \\
&= \sum_{\substack{p \in P(\alpha,\beta), \\ \delta_p > 0}} |\delta_p| \cdot v_{b_p, e_p} - \sum_{\substack{p \in P(\alpha,\beta), \\ \delta_p < 0}} |\delta_p| \bar{v}_{b_p, e_p} \\
&\underset{\text{def } i^*, \text{ etc.}}{\leq} \sum_{\substack{p \in P(\alpha,\beta), \\ \delta_p > 0, i_p = 0}} |\delta_p| \cdot v_{0,i^*} - \sum_{\substack{p \in P(\alpha,\beta), \\ \delta_p < 0, i_p = 0}} |\delta_p| \cdot \bar{v}_{0,j^*} \\
&\quad + \sum_{\substack{p \in P(\alpha,\beta), \\ \delta_p > 0, i_p \neq 0}} |\delta_p| \cdot v_{i_1^*, i_2^*} - \sum_{\substack{p \in P(\alpha,\beta), \\ \delta_p < 0, i_p \neq 0}} |\delta_p| \cdot \bar{v}_{j_1^*, j_2^*} \\
&\leq \sum_{\substack{p \in P(\alpha,\beta), \\ i_p = 0}} |\delta_p| \cdot \max\{v_{0,i^*}, -\bar{v}_{0,j^*}\} \\
&\quad + \sum_{\substack{p \in P(\alpha,\beta), \\ i_p \neq 0}} 2|\delta_p| \cdot \max\{\tfrac{1}{2} v_{i_1^*, i_2^*}, -\tfrac{1}{2} \bar{v}_{j_1^*, j_2^*}\} \\
&\underset{\text{eq (16)}}{\leq} \max\left\{v_{0,i^*}, -\bar{v}_{0,j^*}, \tfrac{1}{2} v_{i_1^*, i_2^*}, -\tfrac{1}{2} \bar{v}_{j_1^*, j_2^*}\right\}.
\end{aligned}$$

This directly states how the optimal solution looks, and one can easily determine the corresponding multipliers, which are set as described in the formulation of the proposition 4. □

Remark 7. *The first two solutions can be interpreted as taking into account the most binding inequality of the type* $\eta \geq \pm \sum_{i=0}^{i^*/j^*} (p_i - q_i)\Delta t_i$. *The free controls are set considering the worst-case scenario regarding the bound (either all* 0 *in the "+"-case or all* 1 *in the "−"-case). Hence, the bound takes a stronger value, if there are less degrees of freedom left in* $[0, i^*/j^*]$.

In the last two cases the solution only considers a time interval $[i_1, i_2]$ *and the largest term*

$$\pm \sum_{i=i_1}^{i_2} (p_i - q_i)\Delta t_i$$

However, since there is no valid inequality containing this term and η, *we have to split a valid inequality to explain the behavior:*

$$\begin{aligned} \eta &\geq \pm \sum_{i=0}^{i_2} (p_i - q_i)\Delta t_i \\ &= \pm \sum_{i=i_1}^{i_2} (p_i - q_i)\Delta t_i \underbrace{\pm \sum_{i=0}^{i_1-1} (p_i - q_i)\Delta t_i}_{\geq -\eta} \\ \Rightarrow \quad \eta &\geq \pm \frac{1}{2} \sum_{i=i_1}^{i_2} (p_i - q_i)\Delta t_i \end{aligned}$$

We observe that the factor of $\frac{1}{2}$ *is added to cancel control decisions in* $[0, i_1]$.

The quality of the two different types of solutions differs because of the prefactor of $\frac{1}{2}$. Therefore, the branching strategy should lead to bounds of the first type, i.e., either v_{0,i^*} or $|\bar{v}_{0,j^*}|$ takes the maximal value. In this case it should be easier to enlarge the lower bounds. As stated in Lemma 5 this is the case if $F = \{0, \ldots, |F|-1\}$. The branching strategy of Algorithm 1 presented in Section 2.3 does exactly this by fixing the controls ordered by time.

Lemma 5. *If the first* $|F|$ *controls are the fixed ones, i.e.,* $F = \{0, \ldots, |F| - 1\}$, *then it holds for all time indices in* F*:*

$$\begin{aligned} \max\{v_{0,i^*}, -\bar{v}_{0,j^*}\} &\geq \tfrac{1}{2} v_{i_1,i_2} \quad \forall\, i_1 < i_2 \\ \max\{v_{0,i^*}, -\bar{v}_{0,j^*}\} &\geq -\tfrac{1}{2} \bar{v}_{j_1,j_2} \quad \forall\, j_1 < j_2 \end{aligned}$$

and therefore, the value of the Lagrangian relaxation becomes

$$z_{LR} = \max\{v_{0,i^*}, -\bar{v}_{0,j^*}\}$$

Proof. Both inequalities can be proven directly and we only show the first case, since the second one can be shown analogously. It holds that for $i, j \leq |F| - 1$: $v_{i,j} = \bar{v}_{i,j}$.

$$\begin{aligned} \frac{1}{2} v_{i_1,i_2} &\underset{(\text{def } v)}{=} \frac{1}{2}\left(v_{0,i_2} - v_{0,i_1-1}\right) \\ &\underset{(\text{def } i^*, j^*, v = \bar{v})}{\leq} \frac{1}{2}\left(v_{0,i^*} - \bar{v}_{0,j^*}\right) \\ &\leq \frac{1}{2}\left(\max\{v_{0,i^*}, -\bar{v}_{0,j^*}\} + \max\{v_{0,i^*}, -\bar{v}_{0,j^*}\}\right) \\ &= \max\{v_{0,i^*}, -\bar{v}_{0,j^*}\} \end{aligned}$$

□

We conclude with the statement that the Lagrangian relaxation can be computed quite fast. The values v_{0,i^*} and $\bar{v}_{0,j^*}$ can be computed in linear and the values $v_{i_1^*,i_2^*}$ and $\bar{v}_{j_1^*,j_2^*}$ in quadratic time with respect to n_t. Therefore, the total computing time to calculate the Lagrangian relaxation is $\mathcal{O}({n_t}^2)$. The bound dominates the bound obtained from an LP-relaxation, compare Lemma 2.

For Algorithm 1, we can already justify the major algorithmic choices:

- The branching scheme is chosen such that the expected bound from proposition 4 gets as large as possible. This means we want to avoid solutions with the prefactor of $\frac{1}{2}$ in their bound. Therefore, we branch forward in time, since in this case we never have to take them into account as shown in Lemma 5.
- The algorithm's bound is the value of the Lagrangian relaxation in the one-dimensional case without SOS1 constraints.

7.1 Multiple controls with SOS1-constraint

In this section we investigate the case $n_\omega > 1$ with more integer controls that are coupled by the SOS1 constraint (4c). We focus on the time-index based branching strategy of Algorithm 1 to prove that the algorithm also solves Lagrangian relaxations. The strategy branches time step by time step, i.e., $F = \{0, \dots, |F| - 1\}$, since this strategy emerged as being superior in the one-dimensional case.

The Lagrangian function is given by

$$\begin{aligned} \bar{z}_{LR}(\lambda, \mu) \quad = \quad & \min_{p \in \{0,1\}^{n_\omega}} \quad \sum_{k=1}^{n_\omega} \sum_{j=0}^{n_t-1} (p_{k,j} - q_{k,j}) \Delta t_j \left(\sum_{i=j}^{n_t-1} \lambda_{k,i} - \mu_{k,i} \right) \\ & \text{s.t.} \qquad p_{k,j} \text{ fixed for } j \in F, \\ & \qquad \sum_{k=1}^{n_\omega} p_{k,j} = 1, \quad j \notin F. \end{aligned}$$

Analogously to the one-dimensional case this function is maximized such that

$$\sum_{k=1}^{n_\omega} \sum_{j=0}^{n_t-1} \lambda_{k,j} + \mu_{k,j} = 1.$$

Proposition 6. *The value of the Lagrangian function is*

$$z_{LR} = \max_{i \in F, k \in \{1,\dots,n_\omega\}} \left\| \sum_{j=0}^{i} (p_{k,j}^{fixed} - q_{k,j}) \Delta t_j \right\|$$

Proof. Since the special branching strategy is used, the inner problem becomes

$$\begin{aligned} \bar{z}_{LR}(\lambda, \mu) \quad = \quad & \sum_{k=1}^{n_\omega} \sum_{j \in F} (p_{k,j} - q_{k,j}) \Delta t_j \left(\sum_{i=j}^{n_t-1} \lambda_{k,i} - \mu_{k,i} \right) + \\ & + \min_{p \in \{0,1\}^{n_\omega}} \quad \sum_{k=1}^{n_\omega} \sum_{j \notin F} (p_{k,j} - q_{k,j}) \Delta t_j \left(\sum_{i=j}^{n_t-1} \lambda_{k,i} - \mu_{k,i} \right) \\ & \qquad \text{s.t.} \qquad \sum_{k=1}^{n_\omega} p_{k,j} = 1, \quad j \notin F. \end{aligned}$$

Let $c_{k,j} := \Delta t_j \sum_{i=j}^{n_t-1} \lambda_{k,i} - \mu_{k,i}$ denote the coefficient of $p_{k,j}$ in the objective. The minimization problem can directly be solved for fixed λ, μ:

$$p^*_{k,j} = \begin{cases} p^{fixed}_{k,j}, & \text{if } j \in F, \\ 1, & \text{if } j \notin F \wedge c_{k,j} \leq c_{l,j} \; \forall l \neq k \\ 0, & \text{else.} \end{cases}$$

For each time interval only one control is activated. And since there is no interconnection between the different time intervals, the problem can be solved independently for each time interval j. There, the $p_{k,j}$ with the smallest objective coefficient $c^*_j = c_{k,j}$ is activated.

If the relaxed controls $q_{i,j}$ also fulfill the SOS1-constraint, which they should due to the consistency of the relaxed model, the value of the minimization part in the Lagrangian function is still always non-positive, as it has been in the unrestricted, one-dimensional case. This can be seen in the following:

$$\begin{aligned} \sum_{k=1}^{n_\omega} \sum_{j \notin F} c_{k,j}(p^*_{k,j} - q_{k,j}) &\underset{(\text{def } p^*)}{=} \sum_{j \notin F} \left(c^*_j - \sum_{k=1}^{n_\omega} c_{k,j} q_{k,j} \right) \\ &\underset{(\text{def } c^*_j)}{\leq} \sum_{j \notin F} \left(c^*_j - \sum_{k=1}^{n_\omega} c^*_j q_{k,j} \right) \underset{(\text{SOS1 on } q_{\cdot,j})}{=} \sum_{j \notin F} \left(c^*_j - c^*_j \right) = 0. \end{aligned}$$

Therefore, the outer maximization problem does not waste budget by setting multipliers $\lambda_{k,j}, \mu_{k,j} \neq 0$ for $j \notin F$. However, for the fixed part, the Lagrangian's solution is a convex combination of the terms $\pm \sum_{j=0}^{i} (p_{k,j} - q_{k,j}) \Delta t$. Since there is no degree of freedom left for the p regardless of the values of λ and μ, the maximum of the terms in the convex combination is the solution. □

This result justifies that Algorithm 1 ignores the part of the problem which is not yet fixed in the multidimensional case with only SOS1-constraints. The algorithm's bound is the true value of the Lagrangian relaxation in this case as in the one-dimensional unconstrained case.

7.2 Additional combinatorial constraints

By adding additional constraints to the feasible controls Ω, the inner minimization problem $\bar{z}_{LR}(\lambda, \mu)$ becomes less flexible. The multipliers need not take the clean values as previously described in proposition 4, but can take any arbitrary values. We give a small example with switching constraints: It consists of one control over two time intervals and no branching has been done, the values of the relaxed controls are $q_0 = 0$, $q_1 = 1$, no switches are allowed. The multiplier settings from proposition 4 would all lead to a bound of 0. However, with $\lambda_0 = \frac{2}{3}$ and $\mu_1 = \frac{1}{3}$, both possible settings of the controls $p_0 = p_1 = 0$ and $p_0 = p_1 = 1$ lead to the same value of $\frac{1}{3}$ for the bound. However, in this case, both are still far from the true optimal solution's value, which is 1. This shows that the optimal multipliers cannot be generated in the same way as above anymore.

If the algorithm branches on the variables ordered by time, for the fixed part F, the Lagrangian relaxation of the unrestricted model and the restricted one coincide. Only in the free part at the end, the solutions of the true Lagrangian relaxation can differ from the one obtained as in the propositions. However, as long as the combinatorial constraints are not too restrictive, the difference between the two is not too big. Especially, since for a difference of the multiplicators in the end part, one has to pay by lowering the budget of the multiplications λ and μ for the first, fixed part F. Therefore, we assume that this bound is still strong after some branching has taken place. These observations are consolidated by the numerical results in Section 8. However, as the restrictions get more and more severe, the problem becomes harder to solve. Yet, there

is a point at which this process stops and gets inverted, due to the branching tree becoming smaller when the restrictions become tighter.

The special case of a restriction on the total amount of switches between different controls by a small number s_m is of practical interest.

Corollary 7. *In this case, with fixed problem data n_ω and s_m, the number of solutions is polynomial in n_t, or more precisely it is $\mathrm{O}({n_t}^{s_m})$.*

Proof. The solutions are the leafs of the branching tree. The maximum size of the i-th level of the branching tree is bounded by

$$n_\omega \sum_{j=0}^{\min\{i,s_m\}} \binom{i}{j} (n_\omega - 1)^j = \mathrm{O}(i^{s_m} {n_\omega}^{s_m+1}) = \mathrm{O}({n_t}^{s_m} {n_\omega}^{s_m+1}) = \mathrm{O}({n_t}^{s_m}).$$

This sum is derived by taking all possible solutions over i time steps which have up to s_m switches. First, we fix the time steps at which the switches are done. The index j runs through the different possible numbers of switches realized in any solution and the binomial distributes the switches over different settings in time. After each of the j switches one of the $n_\omega - 1$ controls which are not active has to be activated. This happens j times and hence there are $(n_\omega - 1)^j$ permutations of controls that have j switches. The additional n_ω at the beginning counts the different initial control settings for the first time step. □

For small values of s_m and n_ω the size of the branching tree's size decreases remarkably. This is important, because if we want a better solution, we employ a finer time grid and solve the problem again on this finer grid. In this situation, only n_t in the formula changes, and we can observe that the size of the tree does not increase too much. And therefore, also the computational times do not increase too much, since they are coupled to the size of the branching tree.

8 Computational results

We present numerical results for a benchmark MIOCP from a previous study, (15). After applying the *partial outer convexification* technique it reads

$$\begin{aligned}
\min_{x,v} \quad & x_2(t_\mathrm{f}) \\
\text{s.t.} \quad & \dot{x}_0(t) = -x_0(t)\omega_1(t) + (x_0(t) + x_1(t))\omega_2(t) + (x0(t) - x1(t))\omega_3(t), \\
& \dot{x}_1(t) = (x_0(t) + 2x_1(t))\omega_1(t) + (x_0(t) - 2x_1(t))\omega_2(t) + (x_0(t) + x_1(t))\omega_3(t), \\
& \dot{x}_2(t) = x_0^2(t) + x_1^2(t), \\
& x(0) = (0.5, 0.5, 0)^T, \\
& x_1(t) \geq 0.4, \\
& \omega_k(t) \in \{0,1\},\ k = 1,2,3, \quad \sum_{k=1}^{3} \omega_k(t) = 1
\end{aligned}$$

with $t \in [t_0, t_\mathrm{f}] = [0, 1]$. Note that this problem is almost identical to the one investigated in (18) and originally in (3). We added the additional path constraint $x_1(t) \geq 0.4 \quad t \in [t_0, t_\mathrm{f}]$ to generate a singular arc in the relaxed problem, which makes the CIAP nontrivial. After a discretization on an equidistant control grid size $\Delta t = \frac{1}{n_t}$ and a relaxation of the integrality

constraint we obtain the NLP

$$
\begin{aligned}
\min_{x,q} \quad & x_2(t_\mathrm{f}) \\
\text{s.t.} \quad & \dot{x}_0(t) = -x_0(t)q_{1,j} + (x_0(t) + x_1(t))q_{2,j} + (x0(t) - x1(t))q_{3,j}, \\
& \dot{x}_1(t) = (x_0(t) + 2x_1(t))q_{1,j} + (x_0(t) - 2x_1(t))q_{2,j} + (x_0(t) + x_1(t))q_{3,j}, \\
& \dot{x}_2(t) = x_0^2(t) + x_1^2(t), \\
& x(0) = (0.5, 0.5, 0)^T, \\
& x_1(t_j) \geq 0.4, \\
& q_{k,j} \in [0,1],\ k = 1,2,3, \quad \sum_{k=1}^{3} q_{i,j} = 1,
\end{aligned}
\tag{17}
$$

with $t \in [t_j, t_{j+1}]$ and $j = 0 \ldots n_t - 1$. We are interested in solutions $p \in \{0,1\}^{n_\omega \cdot n_t}$ that minimize (CIAP) and for which the number of switches is bounded,

$$
\frac{1}{2} \sum_{k=1}^{3} \sum_{j=1}^{n_t - 1} |p_{k,j+1} - p_{k,j}| \leq \sigma_{\max}. \tag{18}
$$

All results are obtained on a machine with an Intel dual core CPU with 2.66GHz and 8GB RAM. For different discretizations n_t the relaxed optimal control problems were solved a priori with Bock's direct multiple shooting method, (2), using a Runge-Kutta 4/5 method for the ODE systems. For problem (CIAP) on page 5 with different values of $\sigma_{\max}$ the calculated optimal relaxed controls $q \in [0,1]^{n_\omega \cdot n_t}$ are of relevance.

We solve the MILP (CIAP) with Algorithm 1 that implicitly uses Lagrangian relaxation and *Cplex 12.1* with default options, which uses LP relaxations. As in (16) for a MIOCP without SOS1 constraint, the computational times of our implementation are orders of magnitude faster than *Cplex* for the CIAP related to problem (17).

To understand how much of the computational speedup is due a) to improved bounds of the Lagrangian vs. the LP relaxation and b) to faster solution of the relaxation, we plot a) the number of processed nodes within the branch and bound tree and b) the average computation time per node in Figures 2 and 3 on a logarithmic scale.

In Figure 2 the upper bound on allowed switches $\sigma_{\max}$ is fixed to 30, and the number of control discretization intervals is varied in the range between 100 and 185. The numerical results allow a couple of interpretations.

1. The numbers of branch and bound nodes qualitatively behave alike for the two algorithms. Problems that are difficult for one algorithm to solve are generally also difficult for the other algorithm to solve.

2. The average computational times per node are almost constant.

3. *Cplex* is almost always dominated by Algorithm 1 in regard to the number of processed nodes and the average computational time per node. This is what we expected. The number of visited nodes is smaller since the bound of the combinatorial branch and bound is the Lagrangian relaxation which is superior to the bound obtained by an LP relaxation of the same problem. The computational time per node is about two orders of magnitude faster since instead of solving an LP, the algorithm separately checks feasibility and computes the bound both in $\mathcal{O}(n_t)$.

4. The effort is not monotone in the problem size. This behavior is connected to the objective values of the approximation. They do not necessarily get better for finer grids if the new grid is not a subdivision of the old grid, which is unlikely for equidistant grids. As a rule of thumb, the lower the optimal objective, the smaller the tree that has to be searched.

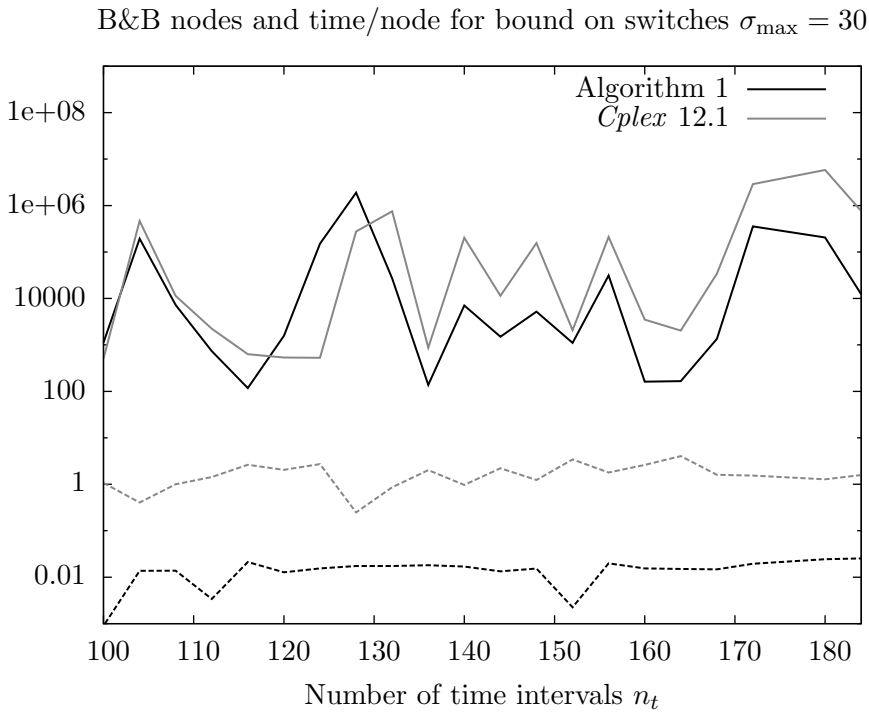

Figure 2: The number of processed branch and bound nodes (solid) as an indication for the quality of lower bounds that allow to cut off subtrees and the average computational time per node in ms (dashed) as a function of the number of control discretization intervals n_t. Note that the Lagrangian bound of Algorithm 1 dominates the LP bound of *Cplex* for $n_t > 130$ and that the calculation of the relaxation is by two orders of magnitude faster.

In Figure 3 four different control discretization grid sizes are kept constant, while $\sigma_{\max}$ is varied. Again, we make some observations.

1. *Cplex* solves problems with $\sigma_{\max} = 0$ in a preprocessing step.

2. Again, qualitatively the curves of branch and bound nodes that need to be solved are alike, and the average computational costs are constant.

3. The dominance of the Lagrangian relaxation bound vs. the LP relaxation bound is not visible. It must not hold anymore due to the addition of combinatorial constraints as discussed in section 7.2. However, we assume that for larger instances the Lagrangian bound becomes better as already seen for the unconstrained problems.

4. As a function of $\sigma_{\max}$, the number of processed B&B nodes has its maximum in the middle between 0 and the number of switches the Sum Up Rounding strategy would take, which marks the right hand side end of the plots (as it can be calculated in linear time and can be proven to solve (CIAP), compare (16)). As already shortly discussed in Section 7.2, combinatorial constraints are most difficult if they are neither too severe nor too light. Too severe restrictions allow the branch and bound algorithm to cut off a major part of its tree. And for very light restrictions, the bound of the Lagrangian relaxation is very good.

5. The size of the B&B tree varies strongly in some cases, when $\sigma_{\max}$ is increased. It seems to be the case that this is related to changes of the optimal objective value η of (CIAP). If η has a value slightly above Δt and is reduced to $\Delta t - \epsilon$ for a different value of $\sigma_{\max}$, this often leads to a significant decrease in the number of nodes which have to be processed. This is because there are many nodes with bound 1. They are created when one "wrong

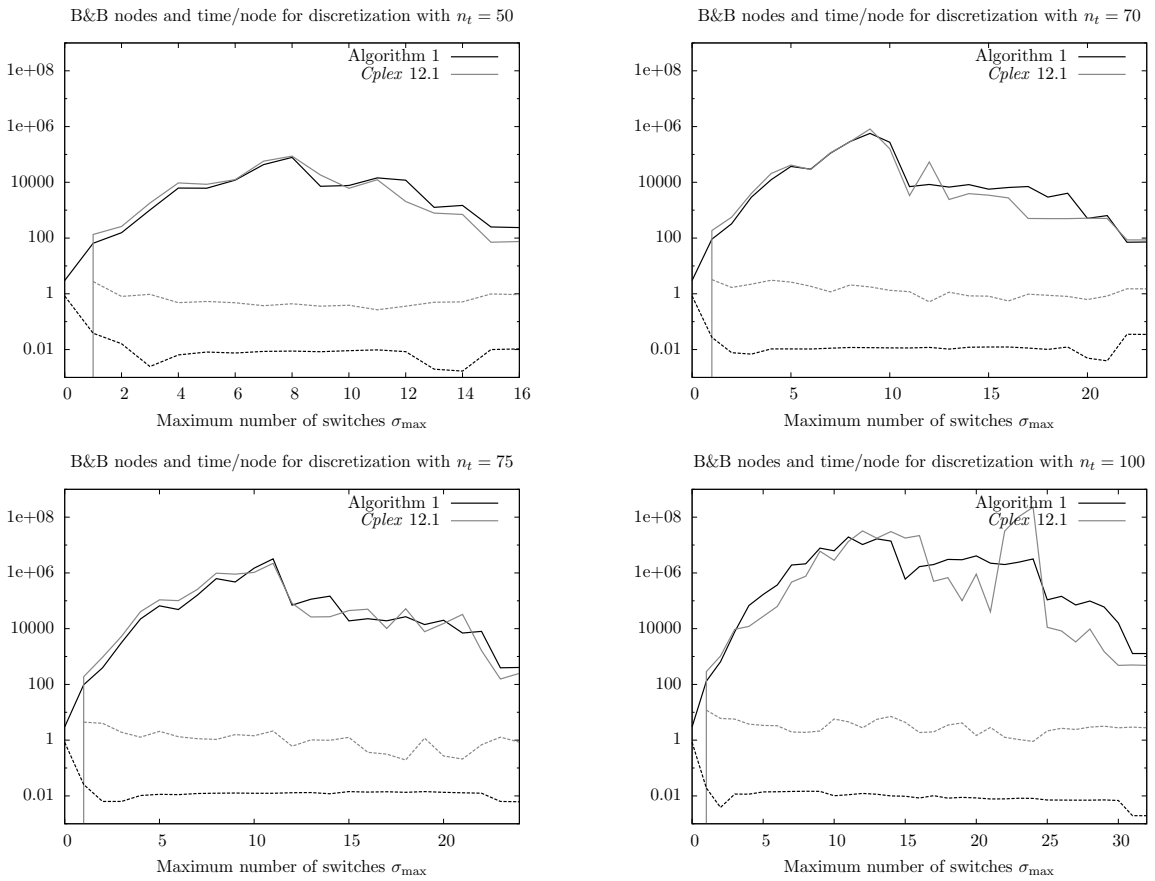

Figure 3: As in Figure 2, but now for fixed control discretization grids $n_t = 50, 70, 75, 100$ and varying upper bounds $\sigma_{\max}$ on the number of switches. For the small numbers of intervals the effect of the better relaxation bounds cannot be observed. The most difficult problems occur for intermediate values of $\sigma_{\max}$.

decision" is made higher up in the tree, and afterwards all decisions for the controls are made such that the min max term is dominated by the early deviation. Apparently, there is a large number of nodes that does not violate this bound and cannot be pruned. For the initial parts, until the hardest problem is reached, the two algorithms behave quite alike. After the hardest problem, there is usually one sharp decrease as explained above and then a steady decrease until the solution stays the same for all values of $\sigma_{\max}$. At this point the switching constraint is no true constraint anymore, because the optimal solution already fulfills the switching constraint. However, the steady decrease has some deviations in it and we can observe that the deviations get bigger with the problem size.

Summing up, the improvement of the combinatorial branch and bound approach is due to an almost constant factor of two orders of magnitude going back to a faster evaluation of the relaxations. For small numbers of intervals the bounds from Lagrangian and LP relaxation have similar effects. For larger problem dimensions the beneficial effect of the Lagrangian relaxation kicks in, yielding up to another two orders of magnitude. Although the Lagrangian relaxation can be proven to be better than the LP relaxation, we use the relaxation of the unconstrained problem. Hence, this is valuable numerical evidence for the appropriateness of these relaxations.

9 Conclusions

We analyzed the Combinatorial Integral Approximation problem and a tailored branch and bound algorithm to solve this MILP. We looked at subproblems that occur as subproblems in a branching scheme from a Lagrangian relaxation point of view.

The Lagrangian relaxation technique is usually used in one of two settings. First, it is often applied to decompose difficult optimization problems. The coupling constraints are penalized in the objective function, resulting in an overall speedup. Second, it is used because the resulting subproblems can be solved efficiently with tailored algorithms. A prominent example is the 1-tree subproblem that occurs in the traveling salesman problem. We showed that this approach is also helpful for the CIAP, although it is unconventional in the sense that it is used inside a branch and bound scheme, and not as a reformulation of the whole problem.

We showed that the branching strategy yields good results because it leads to Lagrangian relaxation bounds without a prefactor $\frac{1}{2}$, compare Theorem 4 and the special case of Lemma 5. The bounds used within our algorithm use the exact values of the Lagrangian relaxation when no additional constraints are present.

Furthermore, we showed that for the proposed branching strategy solving the Lagrangian relaxation on the whole time horizon is identical to solving the CIAP on a smaller time horizon, taking previous fixing decisions into account. This allows for a fast linear time-algorithm.

These results have been highlighted by means of a numerical benchmark example. It can be observed that the behavior of Algorithm 1 is very competitive even after adding further combinatorial constraints. A detailed study of classes of such additional constraints is interesting future work.

Acknowledgments

Financial support of the Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences and of the EU project EMBOCON under grant FP7-ICT-2009-4 248940 is gratefully acknowledged.

References

[1] P. Abichandani, H. Benson, and M. Kam, *Multi-vehicle path coordination under communication constraints*, in *American Control Conference*, IEEE, 2008, pp. 650–656.

[2] H. Bock and K. Plitt, *A Multiple Shooting algorithm for direct solution of optimal control problems*, in *Proceedings of the 9th IFAC World Congress*, Pergamon Press, Budapest, 1984, pp. 242–247.

[3] M. Egerstedt, Y. Wardi, and H. Axelsson, *Transition-time optimization for switched-mode dynamical systems*, IEEE Transactions on Automatic Control 51 (2006), pp. 110–115.

[4] A.M. Geoffrion, *Lagrangean relaxation for integer programming*, in *Approaches to Integer Programming*, Mathematical Programming Studies, Vol. 2, Springer Berlin Heidelberg, 1974, pp. 82–114.

[5] M. Held and R.M. Karp, *The traveling-salesman and minimum cost spanning trees*, Operations Research 18 (1970), pp. 1138–1162.

[6] M. Held and R.M. Karp, *The traveling-salesman problem and minimum spanning trees: Part ii*, Mathematical Programming 1 (1970), pp. 6–25.

[7] C. Kirches, S. Sager, H. Bock, and J. Schlöder, *Time-optimal control of automobile test drives with gear shifts*, Optimal Control Applications and Methods 31 (2010), pp. 137–153.

[8] J. Lee, J. Leung, and F. Margot, *Min-up/min-down polytopes*, Discrete Optimization 1 (2004), pp. 77–85.

[9] D. Leineweber, I. Bauer, H. Bock, and J. Schlöder, *An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part I: Theoretical aspects*, Computers and Chemical Engineering 27 (2003), pp. 157–166.

[10] C. Lemarechal, *Lagrangian relaxation*, in *Computational Combinatorial Optimization*, M. Jünger and D. Naddef, eds., Lecture Notes in Computer Science, Vol. 2241, chap. 4, Springer, 2001, pp. 112–156.

[11] J.A. Muckstadt and S.A. Koenig, *An application of lagrangian relaxation to scheduling in power-generation systems*, Operations Research 25 (1977), pp. 387–403.

[12] H. Pirkul and V. Jayaraman, *A multi-commodity, multi-plant, capacitated facility location problem: formulation and efficient heuristic solution*, Computers & Operations Research 25 (1998), pp. 869–878.

[13] S. Sager, *MIOCP benchmark site*, http://mintoc.de, URL `http://mintoc.de`.

[14] S. Sager, *Reformulations and algorithms for the optimization of switching decisions in nonlinear optimal control*, Journal of Process Control 19 (2009), pp. 1238–1247.

[15] S. Sager, H. Bock, and M. Diehl, *The integer approximation error in mixed-integer optimal control*, Mathematical Programming (2011).

[16] S. Sager, M. Jung, and C. Kirches, *Combinatorial integral approximation*, Mathematical Methods of Operations Research (2011).

[17] S. Sager, G. Reinelt, and H. Bock, *Direct methods with maximal lower bound for mixed-integer optimal control problems*, Mathematical Programming 118 (2009), pp. 109–149.

[18] M. Szymkat and A. Korytowski, *The Method of Monotone Structural Evolution for Dynamic Optimization of Switched Systems*, in *IEEE CDC08 Proceedings*, IEEE, 2008.