Embedded Mixed-Integer Quadratic Optimization using Accelerated Dual Gradient Projection

Vihangkumar V. Naik * **Alberto Bemporad** *

* *IMT School for Advanced Studies Lucca, Italy*
(email: {vihangkumar.naik,alberto.bemporad}@imtlucca.it).

Abstract: The execution of a hybrid model predictive controller (MPC) on an embedded platform requires solving a Mixed-Integer Quadratic Programming (MIQP) in real time. The MIQP problem is NP-hard, which poses a major challenge in an environment where computational and memory resources are limited. To address this issue, we propose the use of accelerated dual gradient projection (GPAD) to find both the exact and an approximate solution of the MIQP problem. In particular, an existing GPAD algorithm is specialized to solve the relaxed Quadratic Programming (QP) subproblems that arise in a Branch and Bound (B&B) method for solving the MIQP to optimality. Furthermore, we present an approach to find a suboptimal integer feasible solution of a MIQP problem without using B&B. The GPAD algorithm is very simple to code and requires only basic arithmetic operations which makes it well suited for an embedded implementation. The performance of the proposed approaches is comparable with the state of the art MIQP solvers for small-scale problems.

Keywords: Mixed-integer quadratic programming, quadratic programming, Accelerated gradient projection, model predictive control, hybrid systems.

1. INTRODUCTION

Mixed Integer Quadratic Programming (MIQP) arises in various fields of applications, in particular in hybrid model predictive control (Bemporad and Morari, 1999). While there exist mature commercial software packages (IBM, Inc., 2014; Gurobi Optimization, Inc., 2014) they are not particularly tailored for embedded platforms, where computing resources are limited and simple library-free code is a requirement.

On-line evaluation of hybrid model predictive control (MPC) needs solution of a MIQP problem at each sample time (Bemporad and Morari, 1999). The only exception are small systems having few binary variables, as multi-parametric programming can be used to presolve the hybrid MPC problem off line (Bemporad et al., 2002; Bemporad, 2015a).

A Branch and Bound (B&B) (Floudas, 1995) based approach to solve MIQPs tailored for MPC is proposed by Axehill and Hansson (2006). The method uses a dual QP formulation and employs warm-starting. However this implementation does not exploit dual lower bounds on the optimal cost, that is very useful to terminate the relaxed QP solver prematurely (Fletcher and Leyffer, 1998; Axehill and Hansson, 2008).

Recently Bemporad (2015b) and Frick et al. (2015) have developed new algorithms for solving MIQPs those are tailored for embedded systems. In Frick et al. (2015) an embedded interior-point based convex programming solver is extended and is combined with a B&B setting. Bemporad (2015b) proposed an approach combining B&B with an active-set method based on nonnegative least squares (NNLS) to solve QP relaxations (Bemporad, 2016, 2017). This approach was particularly tailored to solving small-scale MIQPs such as those arise in embedded hybrid MPC applications.

Due to its combinatorial nature, solving MIQPs to optimality may be impractical in fast applications. For this reason, efforts have been made recently for solving the mixed integer problem *approximately* in a very quick manner. Heuristic methods were proposed which lead to a suboptimal solution of the MIQP while taking considerably less computation time then finding the optimal solution. Fischetti et al. (2005) proposed a heuristic method for finding an integer-feasible solution of a generic mixed integer program, known as the feasibility pump (FP). Bertacco et al. (2007) extended this approach for binary and general-integer variables and the subsequent extension is shown in Achterberg and Berthold (2007). Recently, Takapoui et al. (2016) presented a heuristic for the alternating direction method of multipliers (ADMM) to find an approximate solution of MIQP problem, within very short time.

For what concerns the solution of QP relaxations, Nesterov's fast gradient method (Nesterov, 1983) has recently received great attention by the embedded control community. Reasons for this are its simplicity, relatively good performance, and the good bounds on the worst-case number of iterations that can be obtained as compared to other

⋆ This work was partially supported by the H2020 European project DISIRE, Grant Agreement No. 636834, `http://spire2030.eu/disire/`.

methods. These characteristics make the method favorable for real-time embedded applications. FPGA implementations of fast gradient methods for embedded MPC were presented by Jerez et al. (2013). Patrinos and Bemporad (2014) presented an accelerated gradient projection method to solve the dual QP problem (accelerated dual gradient projection, GPAD). Dual gradient projection with fixed-point embedded implementation was analyzed in (Patrinos et al., 2013; Rubagotti et al., 2016).

This paper introduces two new approaches for solving embedded MIQPs using the GPAD method: (i) using B&B to find the optimal solution, (ii) without B&B to find a suboptimal integer feasible solution. The first approach exploits the fact that the same matrix structure is used in all QP relaxations, so that preconditioning is only required at the root node; subsequent QP relaxations require simply the removal of sign restrictions on some of the dual variables. In addition, the basic GPAD algorithm of Patrinos and Bemporad (2014) is extended to include restart, infeasibility detection, and early termination based on dual cost to enhance the overall performance of the proposed scheme. Regarding the second approach, we propose a heuristic to find a suboptimal integer feasible solution and avoid B&B that is a relatively simple modification to the same GPAD algorithm. As in Takapoui et al. (2016), the heuristic turns out to be quite effective in solving MIQPs approximately, in most cases very close to the optimal solution. This is demonstrated in a hybrid vehicle control example.

1.1 Notation

Let $\mathbb{R}^n$, $\mathbb{R}^{m\times n}$, and $\mathbb{N}$ denote the set of real vectors of length n, the set of real matrices of dimension m by n, and the set of natural integers, respectively. Let $\mathcal{I}\subset\mathbb{N}$ be a finite set of integers. For a vector $a\in\mathbb{R}^n$, a_i denotes the i-th entry of a, $a_{\mathcal{I}}$ the subvector obtained by collecting the entries a_i for all $i\in\mathcal{I}$, $\|a\|_\infty$ the infinite norm of a, the condition $a>0$ is equivalent to $a_i>0$, $\forall i=1,\dots,n$ (and similarly for $\geq$, $\leq$, $<$), For a matrix $A\in\mathbb{R}^{m\times n}$, its transpose is denoted by A', i-th row of A is denoted by A_i, the submatrix of A obtained by collecting the A_i rows for all $i\in\mathcal{I}$ is denoted by $A_{\mathcal{I}}$, and its Frobenius norm by $\|A\|_F$. $\lambda_{\max}(A)$ denotes the largest eigenvalue of A. A^{-1} denotes the inverse of a square matrix $A\in\mathbb{R}^{n\times n}$ (if it exists), $A\succ 0$ denotes that A is positive definite, and similarly $\succeq$, $\prec$, and $\preceq$ denote positive semidefiniteness, negative definiteness, and negative semidefiniteness, respectively. Matrix I_n denotes the identity matrix of order n, where sometimes the subscript n is dropped if the dimension is clear from the context.

2. ACCELERATED DUAL GRADIENT PROJECTION

Consider the following Mixed Integer Quadratic Programming (MIQP) problem

$$\min_z \quad V(z)\triangleq\frac{1}{2}z'Qz+c'z \tag{1a}$$
$$\text{s.t.} \quad \ell\leq Az\leq u \tag{1b}$$
$$A_{eq}z=b_{eq} \tag{1c}$$
$$\bar{A}_iz\in\{\bar{\ell}_i,\bar{u}_i\},\ i=1,\dots,p, \tag{1d}$$

where $Q\in\mathbb{R}^{n\times n}$ is the Hessian matrix, $Q=Q'\succ 0$, $z\in\mathbb{R}^n$ is the optimization vector, $c\in\mathbb{R}^n$, $A\in\mathbb{R}^{m\times n}$, $\bar{A}\in\mathbb{R}^{p\times n}$ and $A_{eq}\in\mathbb{R}^{q\times n}$, $\ell,u\in\mathbb{R}^m$, $\bar{\ell},\bar{u}\in\mathbb{R}^p$ and $b_{eq}\in\mathbb{R}^q$, $\ell\leq u$, $\bar{\ell}\leq\bar{u}$, and $p\leq m$. Binary constraints $z_i\in\{0,1\}$ are a special case of (1d) in which $\bar{A}_i$ is the i-th row of the identity matrix and the corresponding ℓ,u values are $\bar{\ell}_i=0$ and $\bar{u}_i=1$. The QP relaxation of problem (1) is obtained by replacing the integrality constraint (1d) with

$$\bar{\ell}_i\leq\bar{A}_iz\leq\bar{u}_i,\ i=1,\dots,p. \tag{1e}$$

The dual QP of (1a)–(1c), (1e) is also convex and has the form

$$\max_{y,\nu}\ \Psi(y,\nu)\triangleq-\frac{1}{2}\left[\begin{smallmatrix}y\\\nu\end{smallmatrix}\right]'\mathcal{A}Q^{-1}\mathcal{A}'\left[\begin{smallmatrix}y\\\nu\end{smallmatrix}\right]-d'\left[\begin{smallmatrix}y\\\nu\end{smallmatrix}\right]-\frac{1}{2}c'Q^{-1}c$$
$$\text{s.t.}\quad y\geq 0,\ \nu\text{ free} \tag{2}$$

where

$$\mathcal{A}=\begin{bmatrix}A\\-A\\\bar{A}\\-\bar{A}\\A_{eq}\end{bmatrix},\ b=\begin{bmatrix}u\\-\ell\\\bar{u}\\-\bar{\ell}\\b_{\rm eq}\end{bmatrix},\ d=\begin{bmatrix}d_u\\d_\ell\\d_{\bar{u}}\\d_{\bar{\ell}}\\f\end{bmatrix}=b+\mathcal{A}Q^{-1}c \tag{3}$$

and $y=\begin{bmatrix}y_u\\y_\ell\\y_{\bar{u}}\\y_{\bar{\ell}}\end{bmatrix}$, $y_u,y_\ell,d_u,d_\ell\in\mathbb{R}^m$, $y_{\bar{u}},y_{\bar{\ell}},d_{\bar{u}},d_{\bar{\ell}}\in\mathbb{R}^p$, $\nu,f\in\mathbb{R}^q$.

Algorithm 1 extends the formulation of Nesterov's fast gradient method proposed by Patrinos and Bemporad (2014) on the dual QP problem (2) to solve QP problem (1a)–(1c), (1e).

Algorithm 1 Accelerated dual gradient projection method to solve QP problem (1a)–(1c), (1e)

Input: matrices Q, c, A, ℓ, u, $\bar{A}$, $\bar{\ell}$, $\bar{u}$, A_{eq}, b_{eq};

1: $H\leftarrow\mathcal{A}Q^{-1}\mathcal{A}'$, $L\leftarrow\|H\|_F$ or $L\leftarrow\lambda_{max}(H)$, $\mathcal{A}_L\leftarrow\frac{1}{L}\mathcal{A}$, $b_L\leftarrow\frac{1}{L}b$;
2: $y_0,y_{-1}\leftarrow 0$, $\nu\leftarrow 0$;
3: $k\leftarrow 0$;
4: **repeat**
5: $\quad\beta_k\leftarrow\max\left\{\frac{k-1}{k+2},0\right\}$;
6: $\quad\left[\begin{smallmatrix}w_k\\w_{eq,k}\end{smallmatrix}\right]\leftarrow\left[\begin{smallmatrix}y_k\\\nu_k\end{smallmatrix}\right]+\beta_k\left(\left[\begin{smallmatrix}y_k\\\nu_k\end{smallmatrix}\right]-\left[\begin{smallmatrix}y_{k-1}\\\nu_{k-1}\end{smallmatrix}\right]\right)$;
7: $\quad z_k\leftarrow-Q^{-1}\mathcal{A}'\left[\begin{smallmatrix}w_k\\w_{eq,k}\end{smallmatrix}\right]-Q^{-1}c$;
8: $\quad\left[\begin{smallmatrix}s_k\\s_{eq,k}\end{smallmatrix}\right]\leftarrow(\mathcal{A}_Lz_k-b_L)$;
9: $\quad y_{k+1}\leftarrow\max\{w_k+s_k,0\}$;
10: $\quad\nu_{k+1}\leftarrow w_{eq,k}+s_{eq,k}$;
11: **until** convergence;
12: $z^*\leftarrow z_k$, $y^*\leftarrow w_k$, $\nu^*\leftarrow w_{eq,k}$;
13: $a^*\leftarrow\mathcal{A}'\left[\begin{smallmatrix}y^*\\\nu^*\end{smallmatrix}\right]$, $V^*\leftarrow-\frac{1}{2}(a^*)'Q^{-1}a^*-b'\left[\begin{smallmatrix}y^*\\\nu^*\end{smallmatrix}\right]-(Q^{-1}c)'(a^*+\frac{1}{2}c)=\Psi^*$.

Output Primal solution z^*, optimal cost V^*, dual solution (y^*,ν^*).

Note that the only difference between inequality constraints (Step 9) and equality constraints (Step 10) in Algorithm 1 is simply the sign restriction of the corresponding dual variables. This simple observation will be exploited in the B&B approach described in Section 3 when fixing binary constraints during branching.

2.1 Stopping criteria

The iterations of Algorithm 1 are stopped when the primal feasibility criterion

$$s_k^j \leq \frac{1}{L}\epsilon_G,\ \forall j = 1,\dots,2(m+p)$$
$$|s_{eq,k}^j| \leq \frac{1}{L}\epsilon_G,\ \forall j = 1,\dots,q \tag{4}$$

and the optimality criterion

$$-\left[\begin{smallmatrix} w_k \\ w_{eq,k}\end{smallmatrix}\right]' \left[\begin{smallmatrix} s_k \\ s_{eq,k}\end{smallmatrix}\right] \leq \frac{1}{L}\epsilon_V \tag{5}$$

are satisfied, where $\epsilon_G > 0$ is the feasibility tolerance and $\epsilon_V \geq 0$ the optimality tolerance (Patrinos and Bemporad, 2014). Condition (5) derives from the duality gap calculation $V(z_k) - V^* \leq V(z_k) - \Psi(\left[\begin{smallmatrix} w_k \\ w_{eq,k}\end{smallmatrix}\right]) = -\left[\begin{smallmatrix} w_k \\ w_{eq,k}\end{smallmatrix}\right]' \left[\begin{smallmatrix} s_k \\ s_{eq,k}\end{smallmatrix}\right] \leq \frac{1}{L}\epsilon_V$.

2.2 Preconditioning

It is well known that in first-order optimization methods the number of iterations strongly depends on how the problem matrices are scaled. Preconditioning the problem by appropriate scaling is usually adopted for (often largely) improving performance. In this paper we adopt the Jacobi diagonal scaling (Bertsekas, 2009) of the dual Hessian matrix $H = \mathcal{A}Q^{-1}\mathcal{A}'$:

$$\theta_j \triangleq \frac{1}{\sqrt{\mathcal{A}_j Q^{-1} \mathcal{A}_j'}} \tag{6a}$$
$$\mathcal{A}_j \leftarrow \theta_j \mathcal{A}_j,\ b_j \leftarrow \theta_j b_j \quad j = 1,\dots,2(m+p)+q. \tag{6b}$$

2.3 Restart

Accelerated gradient methods do not guarantee that the objective function decreases monotonically during iterations, and indeed ripples in the sequence of function values are often observed. Restarting the sequence of scalars β_k in Step 5 of Algorithm 1 can largely improve the convergence property of the method. We use the gradient-based adaptive restart idea of (O' Donoghue and Candès, 2015; Giselsson and Boyd, 2014) for the dual problem (2) by checking the following condition

$$-\nabla\Psi\left(\left[\begin{smallmatrix} w_k \\ w_{eq,k}\end{smallmatrix}\right]\right)' \left(\left[\begin{smallmatrix} y_{k+1} \\ \nu_{k+1}\end{smallmatrix}\right] - \left[\begin{smallmatrix} y_k \\ \nu_k\end{smallmatrix}\right]\right) > 0. \tag{7}$$

The advantage of the gradient-based restart condition (7) is that it can be immediately computed by available quantities.

2.4 Infeasibility detection

Infeasibility detection of first-order methods was investigated in (Raghunathan and Di Cairano, 2014; O' Donoghue et al., 2016). The QP problem (1a)–(1c), (1e) may be infeasible, a case that frequently occurs during a B&B procedure. In this case the dual cost $\Psi(y_k, \nu_k)$ tends to $+\infty$. The following Lemma 1 characterizes the asymptotic behavior of Algorithm 1 in case of QP infeasibility.

Lemma 1. Let the QP problem (1a)–(1c), (1e) be infeasible. Then $\lim_{k\to\infty} \mathcal{A}' \left[\begin{smallmatrix} w_k \\ w_{eq,k}\end{smallmatrix}\right] / \| \left[\begin{smallmatrix} w_k \\ w_{eq,k}\end{smallmatrix}\right] \|_\infty = 0$.

Proof. Let $\eta_k = \left[\begin{smallmatrix} w_k \\ w_{eq,k}\end{smallmatrix}\right]$. Since $H = \mathcal{A}Q^{-1}\mathcal{A}' \succeq 0$, for $\Psi(w_k, w_{eq,k}) \to +\infty$ it must occur that $d'\eta_k \to -\infty$, and therefore some components $\eta_{k,i}$ must tend to $+\infty$ for corresponding negative entries in vector d. Assume now by contradiction that the quantity $\mathcal{A}'\frac{\eta_k}{\|\eta_k\|_\infty}$ does not go to zero for $k \to \infty$. In this case there would exist a subsequence η_r such that $\|\mathcal{A}'\frac{\eta_r}{\|\eta_r\|_\infty}\|_2 \geq \epsilon$ for some $\epsilon > 0$. In this case, $\Psi(y_r, \nu_r) = -\frac{1}{2}\eta_r'\mathcal{A}Q^{-1}\mathcal{A}'\eta_r - d'\eta_r \leq -\frac{1}{2\lambda_{\max}(Q)}\|\mathcal{A}'\eta_r\|_2^2 + \|d\|_2\|\eta_r\|_2 \leq -\frac{\epsilon^2}{2\lambda_{\max}(Q)}\|\eta_r\|_\infty^2 + \|d\|_2\sqrt{2(m+p)+q}\|\eta_r\|_\infty$, where $\lambda_{\max}(Q)$ is the largest eigenvalue of Q. Since some components of η_r diverge, $\|\eta_r\|_\infty$ diverges as well, and therefore $\Psi(w_r, w_{eq,r}) \leq 0$ for r sufficiently large. This contradicts the fact that $\Psi(w_k, w_{eq,k}) \to +\infty$, and therefore $\mathcal{A}'\frac{\eta_k}{\|\eta_k\|_\infty}$ must go to zero asymptotically if the QP is infeasible. ■

Motivated by Lemma 1, we propose the infeasibility detection criterion summarized in Algorithm 2, where $\epsilon_I > 0$ is a given infeasibility detection tolerance. By letting $\mu_k = \frac{w_k}{\|\eta_k\|_\infty}$, $\pi_k = \frac{w_{eq,k}}{\|\eta_k\|_\infty}$, $\mu_k \in \mathbb{R}^{2(m+p)}$, $\pi_k \in \mathbb{R}^q$ the criterion in Step 2 of Algorithm 2 amounts to verify the following condition

$$\begin{cases} \mathcal{A}' \left[\begin{smallmatrix} \mu_k \\ \pi_k\end{smallmatrix}\right] \approx 0 \\ b' \left[\begin{smallmatrix} \mu_k \\ \pi_k\end{smallmatrix}\right] < 0 \\ \mu_k \geq 0. \end{cases} \tag{8}$$

According to Farkas Lemma (Rockafellar, 1970, p. 201), condition (8) is equivalent to an indication of the infeasibility of the QP (1a)–(1c), (1e). Moreover when $\mathcal{A}' \left[\begin{smallmatrix} \mu_k \\ \pi_k\end{smallmatrix}\right] = 0$, $b' \left[\begin{smallmatrix} \mu_k \\ \pi_k\end{smallmatrix}\right] < 0$ can be equivalently replaced by $d' \left[\begin{smallmatrix} \mu_k \\ \pi_k\end{smallmatrix}\right] < 0$.

Algorithm 2 Infeasibility detection

1: $\alpha_k \leftarrow \| \left[\begin{smallmatrix} w_k \\ w_{eq,k}\end{smallmatrix}\right] \|_\infty$;
2: **if** $\|\mathcal{A}' \left[\begin{smallmatrix} w_k \\ w_{eq,k}\end{smallmatrix}\right] \|_\infty \leq \epsilon_I\alpha_k$ **and** $d' \left[\begin{smallmatrix} w_k \\ w_{eq,k}\end{smallmatrix}\right] < -\epsilon_I\alpha_k$ **then**
3: **stop** (problem is infeasible)
4: **end if**

2.5 Early stopping criterion for the objective function

Assume that we want to stop the QP algorithm if we are not interested in getting a solution whose objective is greater than a given value V_0. In this case, since $V(z_k) \geq \Psi(y_k, \nu_k) \geq V_0$, Algorithm 1 can be stopped if

$$\Psi(y_k, \nu_k) \geq V_0. \tag{9}$$

This condition is particularly advantageous in a B&B setting to prematurely terminate the execution of the QP algorithm solving the relaxation, where V_0 is the best known cost associated with an integer-feasible solution.

3. BRANCH & BOUND MIQP ALGORITHM

Algorithm 3 describes a B&B solver for the MIQP problem (1) that is based on the QP solver described by Algorithm 1 and its extensions presented in Section 2.

The sets $I_{\bar{\ell}}$, $I_{\bar{u}}$, J denote a partition of the set of indices $i \in \{1,\dots,p\}$ ($I_{\bar{u}} \cap I_{\bar{\ell}} = \emptyset$, $J = \{1,\dots,p\} \setminus (I_{\bar{u}} \cup I_{\bar{\ell}})$) such that the integrality constraints (1d) are changed to

$$\bar{A}_{I_{\bar{u}}} z = \bar{u}_{I_{\bar{u}}} \tag{10a}$$
$$\bar{A}_{I_{\bar{\ell}}} z = \bar{\ell}_{I_{\bar{\ell}}} \tag{10b}$$
$$\bar{\ell}_J \leq \bar{A}_J z \leq \bar{u}_J \tag{10c}$$

in a given QP relaxation.

At Step 1, all integrality constraints (1d) are relaxed to (1e), that is $I_{\bar{\ell}} = I_{\bar{u}} = \emptyset$. The tuple $\mathcal{T} = (I_{\bar{\ell}}, I_{\bar{u}})$ uniquely identifies a QP relaxation. The set $\mathcal{S}$ of tuples denotes the set of pending relaxations to be solved by Algorithm 1.

At Step 2.1, the element $\mathcal{T} = \{I_{\bar{\ell}}, I_{\bar{u}}\}$ is popped from the stack $\mathcal{S}$ and the corresponding QP relaxation (1a)–(1c), (10) is solved at Step 2.2, under the additional stopping criterion (9), where V_0 cost of the best integer-feasible solution found so far.

Step 2.3.1 is only executed if the QP relaxation was feasible and did not halt because the condition $V^* > V_0$ was verified by (9). In this case, Step 2.3.1 checks whether all integrality constraints are satisfied and eventually updates the best known integer-feasible solution ζ^* and its corresponding cost V_0 to the value V^*. Otherwise, branching is executed at Steps 2.2.3.1–2.2.3.3 by picking up the index j corresponding to the quantity $t_i = \bar{A}_i z$ that is most distant from $\bar{\ell}_i, \bar{u}_i$ (Step 2.2.3.1). Such a constraint is moved from the set of inequality constraints to the set of equality constraints and two new QP relaxations $\mathcal{T}_0$, $\mathcal{T}_1$ are formed at Step 2.2.3.2. Step 2.2.3.3 push the new problems $\mathcal{T}_0$, $\mathcal{T}_1$ to the stack $\mathcal{S}$ so that the the problem with the least fractional part will be solved first.

Once no more QP relaxations are left, Step 3 checks if the value of V_0 is still $+\infty$, and in this case the MIQP problem (1) is reported infeasible.

Algorithm 3 MIQP solver based on GPAD

Input: MIQP problem matrices $Q = Q' \succ 0$, A, $\bar{A}$, A_{eq} and vectors ℓ, u, b_{eq}, $\bar{\ell}$, $\bar{u}$; feasibility tolerance $\epsilon \geq 0$.

1. **set** $V_0 \leftarrow +\infty$; $\zeta^* \leftarrow \emptyset$; $I_{\bar{\ell}} \leftarrow \emptyset$; $I_{\bar{u}} \leftarrow \emptyset$; $J \leftarrow \{1, \ldots, p\}$; $\mathcal{T} \leftarrow \{I_{\bar{\ell}}, I_{\bar{u}}\}$; $\mathcal{S} \leftarrow \{\mathcal{T}\}$;
2. **while** $\mathcal{S} \neq \emptyset$ **do**:
   - 2.1. $\{I_{\bar{\ell}}, I_{\bar{u}}\} \leftarrow$ last element $\mathcal{T}$ of $\mathcal{S}$; $\mathcal{S} \leftarrow \mathcal{S} \setminus \{\mathcal{T}\}$;
   - 2.2. **execute** Algorithm 1 to solve (1a)–(1c), (10) under condition (9);
   - 2.3. **if** the solution z^*, V^* is returned **then**
     - 2.3.1. **if** $I_{\bar{\ell}} \cup I_{\bar{u}} = \{1, \ldots, p\}$ **or** $t_i \triangleq \bar{A}_i z^* \in \{\bar{\ell}_i, \bar{u}_i\}$, $\forall i \in J$ **then** $V_0 \leftarrow V^*$, $\zeta^* \leftarrow z^*$; **otherwise**
       - 2.2.3.1. $j \leftarrow \arg\min_{i \in J} \left| t_i - \frac{\bar{\ell}_i + \bar{u}_i}{2} \right|$;
       - 2.2.3.2. $\mathcal{T}_0 \leftarrow \{I_{\bar{\ell}} \cup \{j\}, I_{\bar{u}}\}$; $\mathcal{T}_1 \leftarrow \{I_{\bar{\ell}}, I_{\bar{u}} \cup \{j\}\}$;
       - 2.2.3.3. **if** $t_j \leq \frac{\bar{\ell}_i + \bar{u}_i}{2}$ **then** append $\{\mathcal{T}_1, \mathcal{T}_0\}$ to $\mathcal{S}$ **otherwise** append $\{\mathcal{T}_0, \mathcal{T}_1\}$ to $\mathcal{S}$;
3. **if** $V_0 = +\infty$ **then** (1) infeasible **otherwise** $\mathcal{V}^* \leftarrow V_0$;
4. **end**.

Output: Solution ζ^* of the MIQP problem (1), optimal cost $\mathcal{V}^*$, or infeasibility status.

3.1 Exploiting the fixed structure of dual QP relaxations

During the execution of the B&B algorithm, from one QP relaxation to another only the relaxed constraints (10) change. These simply map to the modified constraints

$$y_{\bar{\ell}, I_{\bar{\ell}}}, y_{\bar{u}, I_{\bar{u}}} \text{ unconstrained} \tag{11a}$$

$$y_{\bar{u}, J} \geq 0, \; y_{\bar{\ell}, J} \geq 0 \tag{11b}$$

$$y_{\bar{\ell}, I_{\bar{u}}} = 0, \; y_{\bar{u}, I_{\bar{\ell}}} = 0 \tag{11c}$$

in the dual QP problem, which determines a minor change in Step 9 of Algorithm 1: the max with 0 is not taken for the components of y_{k+1} corresponding to $y_{\bar{\ell}, I_{\bar{\ell}}}$, $y_{\bar{u}, I_{\bar{u}}}$, and the components corresponding to $y_{\bar{\ell}, I_{\bar{u}}}$, $y_{\bar{u}, I_{\bar{\ell}}}$ are zeroed, therefore avoiding updating the corresponding quantities in w_k, s_k during the execution of Algorithm 1.

This is a very attractive feature due to the use of a dual QP solver, as the same QP structure (matrices and preconditioning) can be computed just once for initial relaxation at the root node and maintained unaltered during further branching.

4. HEURISTIC METHOD FOR SUBOPTIMAL BINARY-FEASIBLE MIQP SOLUTIONS

In certain embedded applications solving the B&B Algorithm 3 to optimality may not be possible. In this section we suggest a heuristics to possibly obtain suboptimal binary feasible solutions to the MIQP problem (1) without using B&B. The proposed heuristics is based on the adopted dual gradient projection framework and is similar to the one described in (Takapoui et al., 2016) for the alternating direction method of multipliers (ADMM).

Since either $\bar{A}_i z = \bar{\ell}_i$ or $\bar{A}_i z = \bar{u}_i$, the corresponding dual variables are such that either $y_{\bar{u}_i} = 0$ and $y_{\bar{\ell}_i}$ unconstrained, or vice versa. In other words, the vector $\begin{bmatrix} y_{\bar{u}_i} \\ y_{\bar{\ell}_i} \end{bmatrix}$ must belong to the nonconvex set given by the union of the orthogonal real axes. In order to impose such a nonconvex constraint, we propose the heuristics described in Algorithm 4 to define the values of $y_{\bar{\ell}}$ and $y_{\bar{u}}$ during the execution of Algorithm 1.

First, Algorithm 1 is executed to solve the QP relaxation (1a)–(1c), (1e). If this produces an optimal solution z^* that is not integer feasible, Algorithm 1 is executed again from Step 4 by applying the quantization described in Algorithm 4 after executing Steps 9 and 10. We also replace $-\begin{bmatrix} w_k \\ w_{eq,k} \end{bmatrix}' \begin{bmatrix} s_k \\ s_{eq,k} \end{bmatrix}$ with $\left| \begin{bmatrix} w_k \\ w_{eq,k} \end{bmatrix}' \begin{bmatrix} s_k \\ s_{eq,k} \end{bmatrix} \right|$ in condition (5) in Algorithm 1. Assuming that the algorithm converges, this forces the quantity $\bar{A}_i z_H^*$ to satisfy the integrality constraints (1d), where z_H^* denotes the optimal solution found using the proposed heuristic method.

Algorithm 4 Heuristics for suboptimal binary feasible MIQP solution

1: **for** all $i = 1, \ldots, p$ **do**
2: **if** $\bar{A}_i z_k \geq \frac{\bar{l}_i + \bar{u}_i}{2}$ **then**
3: $y_{\bar{\ell}_i}^{k+1} \leftarrow 0$ and $y_{\bar{u}_i}^{k+1}$ unconstrained;
4: **else**
5: $y_{\bar{u}_i}^{k+1} \leftarrow 0$ and $y_{\bar{\ell}_i}^{k+1}$ unconstrained;
6: **end if**
7: **end for**

5. NUMERICAL RESULTS

Numerical experiments were carried out on a desktop computer with Intel Core i7-4700MQ CPU with 2.40 GHz and 8 GB of RAM, running MATLAB R2015a.

First we consider the hybrid vehicle example described in (Takapoui et al., 2016), that consists of the combination

of a battery, an electric motor/generator, and an engine. For a given power demand P_t^{des} at time $t = 0, \ldots, T-1$, the objective is to plan the battery power P_t^{batt} and engine power P_t^{eng} for the given time interval, such that $P_t^{batt} + P_t^{eng} \geq P_t^{des}$. Let E_t be the energy of the battery at time t, $E_{t+1} = E_t - \tau P_t^{batt}$, where τ is the sample time and $0 \leq E_t \leq E^{max}$. The fuel cost is given by $f(P_t^{eng}, z_t)$ where $f(P, z) = \alpha P^2 + \beta P + \gamma z$ and the constraint on power is $0 \leq P_t^{eng} \leq P^{max} z_t$. The hybrid vehicle control problem is

$$
\begin{aligned}
\min \quad & \eta(E_T - E^{max})^2 + \sum_{t=0}^{T-1} f(P_t^{eng}, z_t) + \delta(z_t - z_{t-1})_+ \\
\text{s.t.} \quad & E_{t+1} = E_t - \tau P_t^{batt} \\
& P_t^{batt} + P_t^{eng} \geq P_t^{des} \\
& z_t \in \{0, 1\}, \ t = 0, \ldots, T-1
\end{aligned}
$$

where P_t^{batt}, P_t^{eng}, z_t (engine on/off) and E_t are the optimization variables. The term $\delta(z_t - z_{t-1})_+$ penalizes the engine going from the off to the on state, where $\delta \geq 0$ for all t. By choosing $T = 72$ steps, the resulting MIQP problem has $n = 862$ optimization variables, $p = 72$ binary variables, $m = 503$ inequality constraints, and $q = 575$ equality constraints.

The solver GUROBI (Gurobi Optimization, Inc., 2014) computes the optimal cost $V^* = 135.9$ in 21.05 s with default options. The cost calculated by the ADMM-based heuristic approach of Takapoui et al. (2016), denoted as miqpADMM, is 138.1 [1] and is obtained in 0.40 s for preconditioning + 3.55 s for solving the problem.

The performance of Algorithm 1+4+2 implemented in interpreted MATLAB code, denoted as miqpGPAD-H, is reported in Table 1 for different values of the feasibility tolerance ϵ_G and optimality tolerance ϵ_V. Whenever condition (7) is satisfied at a given iteration k, rather than restarting the values of β_k we just assign $w_k \leftarrow y_k$, $w_{eq,k} \leftarrow \nu_k$ in Step 6 of Algorithm 1 for that iteration. We also introduce the regularization term $10^{-3}I$ in the primal Hessian matrix Q to make it positive definite. The tolerance value used in Algorithm 2 is $\epsilon_I = 1e-2$.

ϵ_V, ϵ_G	Cost	Precond., Solving	Constr. violation
1e-2, 1e-2	131.5	1.19, 3.81 s	3.62e-1
1e-2, 1e-3	135.9	1.12, 8.62 s	8.09e-3
1e-3, 1e-3	135.9	1.07, 8.67 s	7.98e-3
1e-3, 1e-4	136.0	1.14, 13.37 s	2.59e-3
1e-4, 1e-3	136.1	1.10, 15.87 s	1.08e-3

Table 1. Performance comparison with different values of ϵ_V, ϵ_G for miqpGPAD-H

Figure 1 shows the trajectories obtained with miqpGPAD-H for $\epsilon_V = 1e-2$, $\epsilon_G = 1e-3$ and compare them with the ones obtained by GUROBI and miqpADMM. It is apparent that the proposed heuristic approach is computationally faster and very simple to implement in an embedded control platform, and keeps the quality of the solution over a sufficient level for the practical application at hand.

[1] `https://github.com/cvxgrp/miqp_admm/tree/master/matlab/vehicle.m`

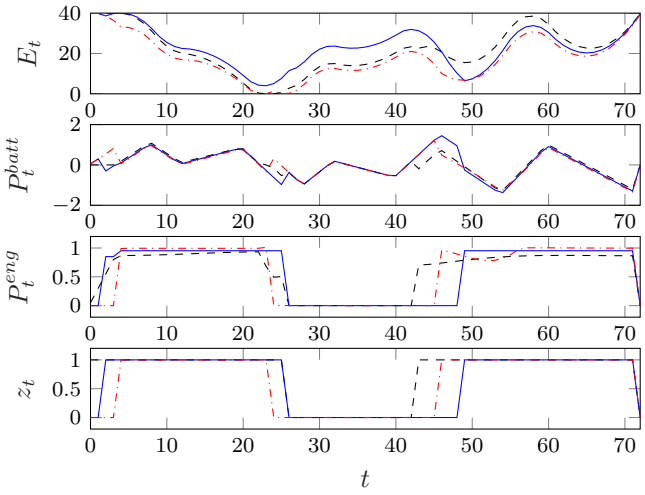

Fig. 1. Battery energy, battery power, engine power and engine on/off signals versus time: GUROBI (solid blue line), miqpGPAD-H (dash-dotted red line), and miqpADMM (dashed black line).

Next, we test the B&B method, denoted as miqpGPAD, on randomly generated MIQP problems with n variables, m inequality constraints, p binary constraints, q equality constraints, and condition number $\kappa = 10$ of the primal Hessian Q [2]. Algorithm 3 is implemented in interpreted MATLAB code and Algorithms 1, 2, 4 are implemented in Embedded MATLAB and complied. The tolerance values used in Algorithm 1 are $\epsilon_V, \epsilon_G = 1e-5$, in Algorithm 2 $\epsilon_I = 1e-2$. The CPU time reported for solving feasible MIQP problems, averaged over 50 executions is listed in Table 2. The results show that the proposed scheme performs well as compared to the commercial GUROBI solver, but has the advantage of a very simple coding, therefore making it very suitable for embedded control applications.

n	m	p	q	miqpGPAD	GUROBI
10	100	2	2	15.6	6.56
50	25	5	3	3.44	8.74
50	150	10	5	63.22	46.25
100	50	2	5	6.22	26.24
100	200	15	5	164.06	188.42
150	100	5	5	31.26	88.13
150	200	20	5	258.80	274.06
200	50	15	6	35.08	144.38

Table 2. Average CPU time (ms) on random MIQP problems over 50 instances for each combination of n, m, p, q.

6. CONCLUSION

In this paper we have presented an exact and a heuristic approach to solve MIQPs based on accelerated gradient

[2] The entries of matrix A are generated from the normal distribution $\mathcal{N}(0, 0.0025)$, ℓ, u from the uniform distribution $\mathcal{U}(0, 100)$, c from $\mathcal{N}(0, 1)$; matrix $Q = U\Sigma V'$, where U, V are orthogonal matrices generated by QR decomposition of random $n \times n$ matrices, and Σ is diagonal with nonzero entries having logarithms equally spaced between $\pm \log(\kappa)/4$ (Bierlaire et al., 1991).

projection applied on the dual QP relaxations. In spite of their simplicity of code, the proposed approaches turn out to be quite effective. In particular, the heuristic method can often provide solutions close to optimality with much reduced coding and computation efforts. Current research is devoted to combining heuristic and B&B to improve the quality of suboptimal solutions, and to tailor hybrid dynamical models to the MIQP solution technique presented in the paper, to enhance the overall performance of embedded hybrid model predictive controllers.

REFERENCES

Achterberg, T. and Berthold, T. (2007). Improving the feasibility pump. *Discrete Optimization*, 4(1), 77–86.

Axehill, D. and Hansson, A. (2006). A mixed integer dual quadratic programming algorithm tailored for MPC. In *Proc. 45th IEEE Conference on Decision and Control*, 5693–5698. San Diego, CA, USA.

Axehill, D. and Hansson, A. (2008). A dual gradient projection quadratic programming algorithm tailored for mixed integer predictive control. Technical Report LiTH-ISY-R-2833, Department of Electrical Engineering, Linköping University, Sweden.

Bemporad, A. (2015a). A multiparametric quadratic programming algorithm with polyhedral computations based on nonnegative least squares. *IEEE Trans. Automatic Control*, 60(11), 2892–2903.

Bemporad, A. (2015b). Solving mixed-integer quadratic programs via nonnegative least squares. In *Proc. 5th IFAC Conf. on Nonlinear Model Predictive Control*, 73–79. Sevilla, Spain.

Bemporad, A. (2016). A quadratic programming algorithm based on nonnegative least squares with applications to embedded model predictive control. *IEEE Trans. Automatic Control*, 61(4), 1111–1116.

Bemporad, A. (2017). A numerically stable solver for positive semi-definite quadratic programs based on nonnegative least squares. *IEEE Trans. Automatic Control*. Conditionally accepted for publication.

Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3), 407–427.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.

Bertacco, L., Fischetti, M., and Lodi, A. (2007). A feasibility pump heuristic for general mixed-integer problems. *Discrete Optimization*, 4(1), 63–76.

Bertsekas, D. (2009). *Convex Optimization Theory*. Athena Scientific.

Bierlaire, M., Toint, P., and Tuyttens, D. (1991). On iterative algorithms for linear ls problems with bound constraints. *Linear Algebra and Its Applications*, 143, 111–143.

Fischetti, M., Glover, F., and Lodi, A. (2005). The feasibility pump. *Mathematical Programming*, 104(1), 91–104.

Fletcher, R. and Leyffer, S. (1998). Numerical experience with lower bounds for MIQP branch-and-bound. *SIAM J. Optim.*, 8(2), 604–616.

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

Frick, D., Domahidi, A., and Morari, M. (2015). Embedded optimization for mixed logical dynamical systems. *Computers & Chemical Engineering*, 72, 21–33.

Giselsson, P. and Boyd, S. (2014). Monotonicity and restart in fast gradient methods. In *53rd IEEE Conference on Decision and Control*, 5058–5063.

Gurobi Optimization, Inc. (2014). *Gurobi Optimizer Reference Manual*. URL `http://www.gurobi.com`.

IBM, Inc. (2014). *IBM ILOG CPLEX Optimization Studio 12.6 – User Manual*.

Jerez, J.L., Goulart, P.J., Richter, S., Constantinides, G.A., Kerrigan, E.C., and Morari, M. (2013). Embedded predictive control on an FPGA using the fast gradient method. In *Proc. European Control Conf.*, 3614–3620.

Nesterov, Y. (1983). A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Sov. Math. Doklady*, 27(2), 372–376.

O' Donoghue, B. and Candès, E. (2015). Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, 15(3), 715–732.

O' Donoghue, B., Chu, E., Parikh, N., and Boyd, S. (2016). Conic optimization via operator splitting and homogeneous self-dual embedding. *J. Optim. Theory Appl.*, 169(3), 1042–1068.

Patrinos, P. and Bemporad, A. (2014). An accelerated dual gradient-projection algorithm for embedded linear model predictive control. *IEEE Trans. Automatic Control*, 59(1), 18–33.

Patrinos, P., Guiggiani, A., and Bemporad, A. (2013). Fixed-point dual gradient projection for embedded model predictive control. In *Proc. European Control Conf.*, 3602–3607. Zurich, Switzerland.

Raghunathan, A. and Di Cairano, S. (2014). Infeasibility detection in alternating direction method of multipliers for convex quadratic programs. In *53rd IEEE Conference on Decision and Control*, 5819–5824.

Rockafellar, R. (1970). *Convex Analysis*. Princeton University Press.

Rubagotti, M., Patrinos, P., Guiggiani, A., and Bemporad, A. (2016). Real-time model predictive control based on dual gradient projection: Theory and fixed-point FPGA implementation. *International Journal of Robust and Nonlinear Control*, 26(15), 3292–3310.

Takapoui, R., Moehle, N., Boyd, S., and Bemporad, A. (2016). A simple effective heuristic for embedded mixed-integer quadratic programming. In *Proc. American Contr. Conf.*, 5619–5625. Boston, MA, USA.