Further Results on the Exploration of Combinatorial Tree in Multi-Parametric Quadratic Programming

Parisa Ahmadi-Moshkenani, Sorin Olaru, Tor Johansen

Further Results on the Exploration of Combinatorial Tree in Multi-Parametric Quadratic Programming

Parisa Ahmadi-Moshkenani[1] Sorin Olaru [2] and Tor Arne Johansen [1]

***Abstract*— A combinatorial approach has been recently proposed for multi-parametric quadratic programming and has shown to be more effective in finding the complete solution than existing geometric methods for higher-order systems. In this paper, we propose a method for exploring the combinatorial tree which exploits some of the underlying geometric properties of adjacent critical regions as the supplementary information in combinatorial approach to exclude a noticeable number of feasible candidate active sets from combinatorial tree. This method is particularly well-suited for cases where many combinations of active constraints are feasible but not optimal. Results indicate that this method can find all critical regions corresponding to non-degenerate multi-parametric programming. A post-processing algorithm can be applied to complete the proposed method in the cases in which some critical regions might not be enumerated due to degeneracies in the problem.**

I. INTRODUCTION

Using techniques based on multi-parametric quadratic programming (mp-QP) is a well known method in the area of explicit model predictive control (EMPC) [1]. By solving the optimization problem offline, this method allows the applicability of MPC to systems with relatively higher sampling rates. In the typical mp-QP algorithms, the fixed active set can be used via the Karush-Kuhn-Tucker (KKT) optimality conditions to characterize the affine local parametric optimal solution and a representation of its polyhedral critical region (CR). Most algorithms for exact mp-QP that have been proposed, iteratively build a partition of the parameter space based on geometric (polyhedral) computations that keeps track of which part of the parameter space has been explored, e.g. [2], [3] and [4]. These are referred to as geometric mp-QP algorithms. A benefit of this approach is that mostly optimal combinations of active sets are considered, avoiding unnecessary computations and storage due to the combinatorial number of possible active sets. A drawback, on the other hand, is that for problems of high dimension of the parameter space, or with many constraints, the geometric computations become complex and numerically sensitive. Moreover, the problem of keeping track of the geometric relationships between the explored and unexplored regions becomes complex due to the large number of facets of each polyhedral critical region. Hence, these algorithms tend to become slow, unreliable, or run out of memory as the problem complexity grows. In order to avoid some of the drawbacks of geometric mp-QP algorithms, a combinatorial approach has been recently proposed in [5] and [6]. By an implicit enumeration of possible combinations of active constraints in a combinatorial search tree, this method avoids geometric computations and manages the combinatorial nature of the computational complexity by using a pruning rule to simultaneously cut off branches with several infeasible active sets represented in the implicit enumeration. This allows the algorithm to deal quite effectively and efficiently with mp-QP problems having a higher number of parameters, when the geometric methods tend to fail [7]. However, it has been noted in [7] that similar to the geometric methods, the enumeration-based method does not scale well towards problems with a large number of constraints, due to exponentially increasing number of possible combinations of active constraints, i.e. the number of LPs that are needed to be solved offline. To decrease the computational complexity of combinatorial approach, [8] has proposed to use a saturation matrix pruning criterion to exclude infeasible candidate active sets a priori and hence, decreasing the number of LPs. The main drawback of this method is that computing all the vertices of the constraint polyhedron and constructing the saturation matrix may become computationally demanding or even prohibitive when the size of the mp-QP increases. Hence its applications is limited to small-scale problems [8]. The work reported in [6] suggests another method for decreasing the computational complexity based on exploiting symmetry in the problem. However, the maximal complexity reduction using this method is about 50% for systems with complete symmetry which does not hold in many cases.

The common feature of all combinatorial algorithms proposed so far is that feasible but not optimal combinations of active constraints are never pruned and they remain in the combinatorial tree while in many cases they do not contribute to any optimal set in lower levels. Since these methods ignore which combinations of active constraints can potentially be optimal in the adjacent CRs, it seems advantageous to exploit the supplementary information available about the adjacent CRs from geometric approaches. Therefore, in this paper we develop a method for exploring the combinatorial tree by only keeping the track of optimal active sets and propose a joint downward-upward exploration method to find the optimal active sets by solving a significantly smaller number of LPs. In regular cases

*The work leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement no 607957 (TEMPO).

[1]Department of Engineering Cybernetics, NTNU, O.S. Bragstads plass, 7491 Trondheim, Norway parisa.ahmadi.moshkenani@itk.ntnu.no, tor.arne.johansen@itk.ntnu.no

[2]SUPÉLEC Systems Science (E3S), Automatic Control Department, 3 rue Joliot-Curie, 91192 Gif-Sur-Yvette, France sorin.olaru@supelec.fr

with no degeneracies, this method can successfully find all possible optimal active sets and thus construct the explicit solutions and the corresponding CRs. For cases presenting degeneracies with respect to the constraints qualifications, simulation results have shown that a great percentage of CRs are found. Moreover, a post-processing algorithm is suggested to find missed CRs if there are any. Hence, this paper is structured as follows. The combinatorial approach towards solving a mp-QP is briefly explained in section II. The new algorithm together with some examples are presented in section III. In section IV the post-processing algorithm for ensuring that all optimal active sets and their corresponding CRs and control laws are found is proposed in conjunction with the simulation results and finally the paper is concluded in section V.

II. COMBINATORIAL APPROACH TOWARDS MULTI-PARAMETRIC QUADRATIC PROGRAMMING

A. Multi-Parametric Quadratic Programming

Consider the standard multi-parametric quadratic program:

$$V_N^*(x) = \min_z \ \frac{1}{2} z^T H z \tag{1a}$$

$$\text{s.t.} \quad Gz \leq Sx + W \tag{1b}$$

where $z \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$ denote the optimization variables and parameters, respectively. Assume that the problem is strictly convex, i.e. $H > 0$ and all constraints are irredundant (see [9] for more details on redundancy removing). As shown by [1], the Karush-Kuhn-Tucker (KKT) optimality conditions can be used to characterize the analytic solutions to mp-QP problem in (1):

$$Hz + G^T \lambda = 0, \qquad \lambda \in \mathbb{R}^q, \tag{2a}$$

$$\lambda^i (G^i z - W^i - S^i x) = 0, \qquad i = 1, \ldots, q, \tag{2b}$$

$$\lambda \geq 0, \quad Gz \leq Sx + W \tag{2c}$$

Defining $\mathcal{Q} = \{1, \ldots, q\}$ as the index set of all constraints in (1b), we recall that a constraint among q constraints in (1b) is said to be *active* if it holds with equality for a given z and x and *inactive* if it holds with strict inequality. Thus the active set $\mathcal{A}(z, x)$ can be described as $\mathcal{A}(z, x) := \{i \in \mathcal{Q} \mid G^i z - S^i x - W^i = 0\}$ while the corresponding inactive set $\mathcal{J}(z, x)$ is given by the set difference of $\mathcal{Q}$ and $\mathcal{A}$ i.e. $\mathcal{J}(z, x) := \mathcal{Q} \backslash \mathcal{A}(z, x)$. Furthermore, for an index set $\mathcal{A} \subseteq \mathcal{Q}$ the linear independence constraint qualification (LICQ) holds if the gradients of the corresponding constraints are linearly independent, i.e., if $G^{\mathcal{A}}$, referring to the submatrix of G that contains the constraints associated to the indices in $\mathcal{A}$, has full row rank. Assuming that we know the optimal active set $\mathcal{A}$ and that LICQ holds for this combination, we can use (2a) and (2b) to derive the parameter-dependent optimizer [1].

$$z_{\mathcal{A}}(x) = H^{-1} (G^{\mathcal{A}})^T H_{G^{\mathcal{A}}}^{-1} (W^{\mathcal{A}} + S^{\mathcal{A}} x) \tag{3}$$

where the existence of $H_{G^{\mathcal{A}}}^{-1} := (G^{\mathcal{A}} H^{-1} (G^{\mathcal{A}})^T)^{-1}$ is guaranteed due to the LICQ and positive definiteness of H. The set of inequalities in (2c) characterize the critical region for the assumed optimal active set $\mathcal{A}$. The CR is in the form of a polyhedron in the parameter space defined by the following inequalities

$$H_{G^{\mathcal{A}}}^{-1} (W^{\mathcal{A}} + S^{\mathcal{A}} x) \leq 0 \tag{4a}$$

$$G H^{-1} (G^{\mathcal{A}})^T H_{G^{\mathcal{A}}}^{-1} (W^{\mathcal{A}} + S^{\mathcal{A}} x) \leq W + Sx \tag{4b}$$

This polyhedron is the largest set of parameters for which the combination of active constraints at the optimizer remains unchanged.

B. Combinatorial Approach

In this section, we briefly summarize the main idea of the combinatorial approach which is based on the implicit enumeration of all possible combinations of active constraints. From the definition of the power set, all possible combinations of active constraints are included in power set $\mathcal{P}(\mathcal{Q})$. To consider all optimal active constraints, [5] suggests to choose the candidate active sets from $\mathcal{P}'(\mathcal{Q})$ which is a subset of $\mathcal{P}(\mathcal{Q})$ including all the sets with maximum $\tilde{m} = min\{m, q\}$ members (note that as pointed by [5], for a mp-QP with m decision variables ($z \in \mathbb{R}^m$) and q constraints, only a maximum of $\tilde{m}$ linearly independent constraints can be strongly active at the optimal solution [10]) in the order of increasing cardinality and use the following LP to check whether the candidate set is indeed optimal or not.

$$\max_{z, x, \lambda^{\mathcal{A}_i}, s^{\mathcal{J}_i}} t \tag{5a}$$

$$\text{s.t.} \quad te_1 \leq \lambda^{\mathcal{A}_i}, te_2 \leq s^{\mathcal{J}_i} \tag{5b}$$

$$t \geq 0, \lambda^{\mathcal{A}_i} \geq 0, s^{\mathcal{J}_i} \geq 0 \tag{5c}$$

$$Hz + (G^{\mathcal{A}_i})^T \lambda^{\mathcal{A}_i} = 0 \tag{5d}$$

$$G^{\mathcal{A}_i} z - S^{\mathcal{A}_i} x - W^{\mathcal{J}_i} = 0 \tag{5e}$$

$$G^{\mathcal{J}_i} z - S^{\mathcal{J}_i} x - W^{\mathcal{A}_i} + s^{\mathcal{J}_i} = 0 \tag{5f}$$

Here t is a scalar optimization variable and $e_1 = [1, \ldots, 1]^T$ and $e_2 = [1, \ldots, 1]^T$ are vectors of appropriate sizes corresponding to the vector of Lagrangian multipliers $\lambda^{\mathcal{A}_i}$ and the vector of slack variables $s^{\mathcal{J}_i}$, respectively. The pruning criterion in the combinatorial approach is based on infeasibility of the combination of active constraints and the following theorem from [5].

Theorem 1: *If an optimization problem P consisting of inequalities and equalities is infeasible, then the new optimization problem P' formed by treating some of the inequality constraints as equality constraints will also be infeasible.*

A graphical illustration of the combinatorial enumeration strategy and the involved pruning process is given in the form of a tree diagram in Fig. 1. As depicted, when a combination of active constraints is infeasible, that and all its supersets can be pruned due to Theorem 1. This pruning criterion is crucial for achieving optimal efficiency in the enumeration. Based on the above explanations, the combinatorial approach proposed by [5] and [6] can be summarized as in Algorithm 1.

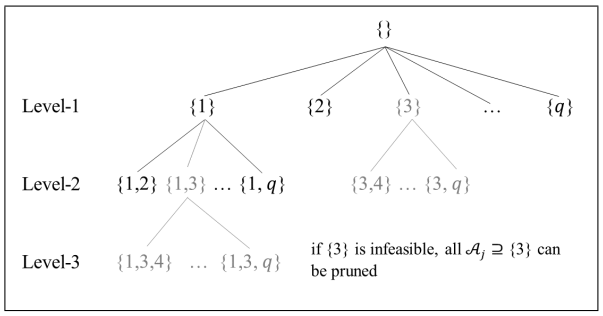

Fig. 1. Combinatorial enumeration strategy used in [5]

Algorithm 1 Combinatorial mp-QP algorithm from [5], [6]

1) choose $\mathcal{A} \in \mathcal{P}'(\mathcal{Q})$ in the order of increasing cardinality
2) if $\mathcal{A}_i$ is not pruned and $G^{\mathcal{A}_i}$ has full row rank, solve (5)
   - if feasible, use (3) and (4) to construct $z_{\mathcal{A}_i}$ and $CR_{\mathcal{A}_i}$
   - if infeasible, solve (5) without optimality constraints
     - if infeasible, add all $\mathcal{A}_j \supset \mathcal{A}_i$ to the pruned sets
3) Return to 1) until the whole set $\mathcal{P}'(\mathcal{Q})$ is explored

III. EXPLOITING SUPPLEMENTARY INFORMATION

In this section we propose a new algorithm for exploring the combinatorial tree to avoid solving a significant number of LPs by exploiting supplementary information about adjacent CRs for non-degenerate cases. While the existing combinatorial approach proceeds through the tree only in downward direction, our algorithm requires some joint downward and upward explorations. For cases with no degeneracies, identification of all optimal active combinations of constraints and hence their corresponding CRs is guaranteed. In the next section, we relax the assumption and extend the study by suggesting a post-processing algorithm to find the missed CRs that may occur in degenerate cases. To begin with, we recall the the following definitions from [3].

Definition 1. Let a polyhedron $X \in \mathbb{R}^n$ be represented by the linear inequalities $\mathcal{A}_0 \leq b$. Let the ith hyperplane $\mathcal{A}_0^i x = b^i$ be denoted by $\mathscr{H}$. If $X \cap \mathscr{H}$ is $(n-1)$-dimensional then $X \cap \mathscr{H}$ is called a *facet* of the polyhedron.

Definition 2. Two polyhedra are called *neighbouring* or *adjacent* polyhedra if they have a common facet.

The supplementary information on which this algorithm is based is the following theorem from [3].

Theorem 2: *Consider an optimal active set $\{i_1, i_2, \ldots, i_k\}$ and its corresponding minimal representation of the full-dimensional critical region CR_0. Let CR_i be a full-dimensional neighbouring critical region to CR_0 and assume LICQ holds on their common facet $\mathscr{F} = CR_0 \cap \mathscr{H}$ where $\mathscr{H}$ is the separating hyperplane between CR_0 and CR_i. Moreover, assume that there are no constraints which are weakly active at the optimizer $z^*(x)$ for all $x \in CR_0$. Then:*

Type I: If $\mathscr{H}$ is given by $G^{i_{k+1}} z_0^*(x) = W^{i_{k+1}} + S^{i_{k+1}} x$, then the optimal active set in CR_i is $\{i_1, i_2, \ldots, i_k, i_{k+1}\}$.

Type II: If $\mathscr{H}$ is given by $\lambda_0^{i_k}(x) = 0$, then the optimal active set in CR_i is $\{i_1, i_2, \ldots, i_{k-1}\}$.

Theorem 2 indicates that if there are no degeneracies, i.e. LICQ holds in the common facet between two CRs and there are no weakly active constraints at the optimizer, the optimal active sets in two adjacent CRs differ only in one constraint. Based on this, assume that we have found the candidate set $\mathcal{A}_i$ consisting of l constraints (which belongs to level-l in the combinatorial tree) to be optimal and the corresponding critical region $CR_{\mathcal{A}_i}$ is constructed. If the conditions in theorem 2 hold, all optimal active sets in the adjacent critical regions to $CR_{\mathcal{A}_i}$ can have only one constraint more or less than the constraints in set $\mathcal{A}_i$. Coming back to the combinatorial tree interpretation, this means that we can explore the combinatorial tree from top to bottom by starting from optimal active sets with minimum number of constraints and then trying to find their adjacent CRs by adding feasible constraints to those optimal active sets one by one which generates the lower levels. Algorithm 2 shows the new strategy that is based on downward exploration of the combinatorial tree in which, we do not check the feasibility of any candidate active set that is not optimal (except for the first level), and the construction and exploration of the combinatorial tree is only based on the optimal active sets. This leads to a noticeable reduction in the number of LPs needed to be solved which typically arise from the feasible but not optimal combinations of active constraints. Next, the simulation results for two systems using this new algorithm are presented.

Algorithm 2 Downward exploration of the combinatorial tree

Phase I (initialization):

1) $i = 1$, explore the entire level-1, use (5) to check for optimality of each constraint. If the constraint is not optimal, use (5) without optimality conditions to check for feasibility of that constraint. Store all optimal constraints in "optimal set" and all feasible constraints, whether they are optimal or not, in "feasible set". All constraints in "feasible set" are manipulated during the iterations in 2);
   - if no constraint is found to be optimal in 1
     - while optimal set is empty, explore the entire level-$(i+1)$, check *only* for the optimality of the combinations;
       $i := i + 1$;

Phase II (iterations):

2) while $i < \tilde{m} = min\{m, q\}$, construct level-$(i+1)$ by adding one feasible constraint from level-1 to all found optimal sets in level-i and check *only* for the optimality of new combinations;
   $i := i + 1$;

Example 1. As a first numerical test, we applied algorithm 2 to the four tank system from [11] with 4 state variables

and 2 inputs. The simulation results for this system are summarized in Table I where N, q, n_z, n_{CR} and n_{LP} represent the prediction (and control) horizon, number of constraints, number of optimization variables, number of found CRs and number of solved LPs, respectively. The last column shows the ratio of the number of solved LPs using Algorithm 2 to the number of solved LPs using Algorithm 1. It can be seen that this ratio decreases dramatically as the prediction horizon increases. This is due to the fact that by increasing the number of constraints, the number of candidate active sets increases exponentially, while most of them are only feasible but not optimal.

Example 2. As a second example, we considered the fuel cell breathing control system in [12] and [7] with 8 state variables and 1 input and prediction horizon $N = 6$ and discretized the model with sampling time $T_d = 1$ *sec.* Table II provides an insightful illustration of the particular situation that Algorithm 2 may encounter. It can be observed that 6 critical regions are not found using Algorithm 2 although conditions in Theorem 2 holds for this system. This observation is due to the fact that a critical region may be entirely surrounded by critical regions corresponding to combinations of active constraints from lower levels of the combinatorial tree. In such cases, the critical region will not be enumerated using Algorithm 2 with downward exploration. In this example, one of the CRs which is not enumerated using algorithm 2 corresponds to $A_i = [9, 10, 17]$ since none of the combinations $A_i = [9, 10]$, $A_i = [9, 17]$ and $A_i = [10, 17]$ in the upper level are found to be optimal. The black CRs in Fig. 2 are found during downward exploration via different branches while the grey CRs are missed. To cope with such particular situation, we propose a modified version of the Algorithm 2 which exploits the principle of a upward exploration of the combinatorial tree. This means that for all found optimal combinations of active constraints with k elements, we check all subsets with $k-1$ elements that have not been already enumerated to see if they are optimal or not. Any newly found optimal combination should also be considered in downward exploration recursively. Doing this, the optimal combination $A_i = [9, 10, 17]$ will be found in upward exploration as a subset of $A_i = [9, 10, 13, 17]$ and $A_i = [9, 10, 17, 19]$ will be found in downward exploration as a superset of $A_i = [9, 10, 17]$. Hence, the modified version of Algorithm 2 can be presented as in Algorithm 3.

The following theorem proves that Algorithm 3 will enumerate all critical regions for non-degenerate cases.

Theorem 3: *Algorithm 3 guarantees the enumeration of all optimal sets in the absence of degeneracy.*

Proof: *The theorem can be proved in a similar way to [2]. Feasible space of the mp-QP is a polyhedron that is partitioned in a finite number of fully connected critical regions. There is no isolated region that could not be reached starting from any region and passing through adjacent CRs. Thus we can explore the entire feasible space starting from any region (any combination of optimal active constraints from a combinatorial point of view), and find all the adjacent critical regions until the entire feasible space is explored. For non-degenerate cases, the combination of optimal active constraints in all adjacent CRs to a CR have only one constraint less or more than combination of optimal active constraints in that CR. Both cases are considered in Algorithm 3 by enumerating the subsets and supersets of each optimal active set. Hence Algorithm 3 guarantees finding all neighbouring CRs to any CR and thus, all optimal active sets will be enumerated in a recursive routine.*

The graphical representation for the exploration of the combinatorial tree in example 2 using Algorithm 3 is shown in Fig. 3. Table II implies that while all critical regions are found, the number of LPs which should be solved slightly increases in comparison to Algorithm 2, but it is still in practice considerably diminished with respect to the

TABLE I

COMPARISON BETWEEN ALGORITHM 1 AND ALGORITHM 2 FOR FOUR TANK SYSTEM FROM [11]

Method	N	q	n_z	n_{CR}	n_{LP}	$\frac{n_{LP,Alg.2}}{n_{LP,Alg.1}}$
Alg. 1	1	14	2	11	22	
Alg. 2				11	21	0.9545
Alg. 1	2	20	4	62	679	
Alg. 2				62	223	0.3284
Alg. 1	3	26	6	221	18558	
Alg. 2				221	1306	0.0703
Alg. 1	4	32	8	605	466652	
Alg 2				605	5245	0.0112
Alg. 1	5	38	10	804*	2751780*	
Alg. 2				1393	16273	0.0059

* The code execution is manually stopped after 24 hours. Only 804 CRs were found by solving 2751780 LPs while the actual number of LPs to be solved for the complete solution will be substantially larger.

TABLE II

NUMBER OF SOLVED LPS AND FOUND CRS FOR FUEL CELL BREATHING SYSTEM WITH $N = 6$, USING DIFFERENT ALGORITHMS

Method	N	q	n_z	n_{CR}	n_{LP}
Alg 1	6	30	6	230	22264
Alg 2				224	1483
Alg 3				230	2197

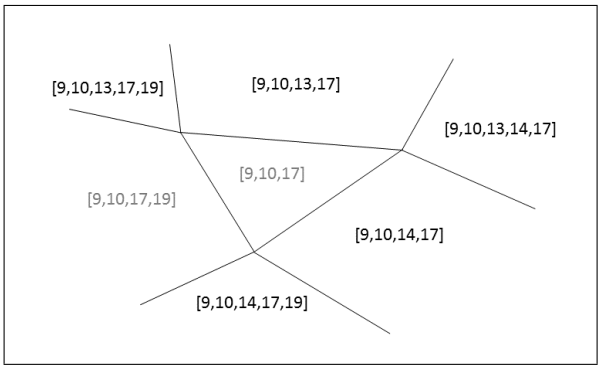

Fig. 2. Schematic representation of neighbouring CRs for example 2 in $2-D$. Regions in black are found during downward exploration in Algorithm 2 while regions in grey are not found

Algorithm 3 Downward-upward exploration of the combinatorial tree

Phase I (initialization):

1) $i = 1$, explore the entire level-1, use (5) to check for optimality of each constraint. If the constraint is not optimal use (5) without optimality conditions to check for feasibility of that constraint. Store all optimal constraints in "optimal set" and all feasible constraints, whether they are optimal or not, in "feasible set";
   - if no constraint is found to be optimal in 1)
     - while optimal set is empty explore the entire level-$(i+1)$, check *only* for optimality of the generated combinations;
       $i := i + 1$;

Phase II (recursive exploration):

2) *(downward exploration)* construct level-$(i+1)$ by adding one feasible constraint from level-1 to all found optimal sets in level-i and check *only* for the optimality of new combinations;
3) *(upward exploration)* for all found optimal active sets with k elements, check the optimality of all its subsets with $k-1$ elements that have not been enumerated yet;
4) if a new optimal set is found, go to 3), else go to 5)
5) add all optimal active sets that are found at the steps 3) and 4) to the appropriate levels in the combinatorial tree and complete the exploration of tree up to current level using downward exploration;
   $i := i + 1$;
   if $i < \tilde{m} = min\{m, q\}$ go to 2), else stop;

number of LPs in Algorithm 1. As all other combinatorial approaches, the number of LPs needed to be solved in algorithm 3 is problem dependent. An upper bound for the number of LPs can be given in terms of the number of CRs (n_{CR}), the total number of constraints (q), the number of feasible constraints (n_{feas}) and the number of levels in combinatorial tree ($\tilde{m} = min\{m, q\}$) in the following form.

$$n_{LP,max} = 2.q + n_{CR} * n_{feas} + n_{CR} * (\tilde{m} - 1) \qquad (6)$$

Here the first two terms are the maximum number of LPs that should be solved in downward exploration as in Algorithm 2. The maximum number of $2.q$ LPs are needed in initialization phase and $n_{CR} * n_{feas}$ indicates the worst case number of LPs in the iteration phase. The third term in (6) is the worst case number of LPs that are added due to upward exploration. Since the maximum number of elements in each optimal active set is $\tilde{m}$, the maximum number of subsets with only one element less than the original set is $\tilde{m}$. The number of subsets is reduced by one because at least one of these subsets is already enumerated in the downward exploration.

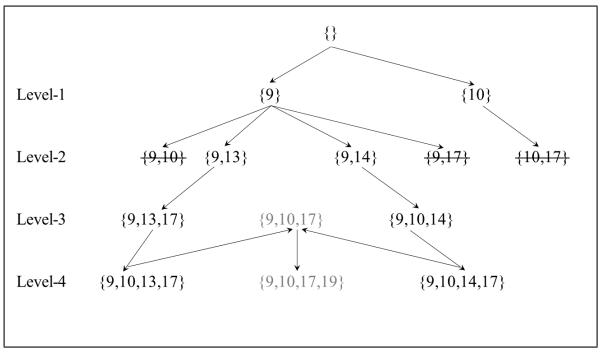

Fig. 3. Downward-upward exploration of combinatorial tree using Algorithm 3 for example 2

IV. POST-PROCESSING ALGORITHM FOR DEGENERATE CASES

The results of the previous section are achieved assuming the absence of degeneracies in the problem as underlined in Theorem 2. When degeneracy happens in the form of weakly active or weakly inactive constraints at the optimizer or failure of LICQ on the common facet between two CRs, the combinations of optimal active constraints in two adjacent CRs may differ in two or more constraints (see Theorem 5 and Lemma 1 in [3] for more details). Consequently, Algorithm 3 fails to find the optimal sets in the adjacent CRs of a CR if degeneracy occurs on all its facets. In other words, only one non-degenerate facet for each CR is enough for enumerating its corresponding optimal set. Simulation results show that the situation for which a part of feasible domain is isolated with facets such that degeneracy occurs on all these facet is not very common. However, since we do not have any a priori information about whether this situation happens or not, one way to handle such cases is to assume that the system has no degeneracies and first use Algorithm 3 to find as many CRs as possible, then to ensure that all critical regions are indeed found, and find the missed CRs if there are any, a post-processing procedure can be exploited which is based on geometric exploration of the unexplored parts. The main task of post-processing procedure is to search for any possible unexplored part within the feasible space to ensure that all CRs are found. To do this, one way is to check for every found CR whether all full-dimensional adjacent CRs along all its facets are found or not. If facet-to-facet property holds for a system [13], every facet that is not located on the boundary of feasible space belongs to exclusively two neighbouring CRs. Hence every inner facet should appear exactly two times among all the facets of all found CRs. Therefore, the first step in the post-processing algorithm is to determine whether there exists a facet, or a part of a facet in cases for which facet-to-facet property fails, among all the inner facets of found CRs that appears only once. This means that the adjacent CR which shares the corresponding facet or part of the facet with current CR to which the facet belongs is not found. As mentioned in Definition 1, a facet is the intersection of a CR with one of the corresponding hyperplanes that define the polyhedron.

Since operations on hyperplanes represented by the corresponding coefficients in its inequality are computationally less complex and quite faster than facets, it is beneficial to first check for all coinciding hyperplanes. Hence the main steps of the post-processing algorithm will be as follows. First, we consider all facets of found CRs using Algorithm 3 and check whether they are located inside the feasible set or not. To do this, we can exploit a similar method to what is done in [2] i.e., for every facet we choose an arbitrary point in the interior of the facet and generate a new parameters by moving a small step from the chosen point in the direction of the normal to the facet and check for feasibility of this new parameter. Feasibility of the new parameter indicates that the facet is located within the feasible parameter set. Next, we consider the corresponding hyperplanes for all inner facets and for each hyperplane we find all coinciding hyperplanes by some simple matrix operations. The results are then stored in a binary matrix in which both rows and columns relate to the indices of hyperplanes. By definition, the (i, j) entry is set to 1 if hyperplanes i and j coincide and 0 otherwise. Since coincidence of hyperplanes is mutual, this matrix is symmetric and it is sufficient to find just the elements of the lower or upper triangular part of the matrix. Nonzero elements in each row show a group of coinciding hyperplanes. Then for every group of coinciding hyperplanes we consider corresponding facets and for every facet we check whether the set difference of that facet with union of all the rest facets is empty or not, using polyhedral operations available in MPT3 [14]. An empty set difference means that all the adjacent CRs along that facet are found. On the other hand, a non-empty set difference means that some adjacent CRs are not found using Algorithm 3. To identify the combination of active constraints in the unexplored part, we can choose any point in the interior of the remained set from the set difference operation (for example the Chebyshev center), generate a new parameters by moving a small step from the chosen point in the direction of the normal to the facet and find the corresponding optimal active set by solving a QP as in [1]–[3]. Once we know the optimal active set, we can use that to construct the control law and CR using equations (3) and (4).The post-processing algorithm is outlined below as Algorithm 4.

Table III shows the computation time for Algorithm 1 and for Algorithm 3 in conjunction with the post-processing Algorithm 4 for example 1 on a 3.2 GHz core i5 CPU running MATLAB 2014a, using MATLAB's "linprog" with "simplex algorithm" for solving the LP (5). It should be noted that choosing appropriate LP solver has a considerable impact on the accuracy and speed of the algorithm. However, comparison between different LP solvers is beyond the scope of this paper. As it can be seen from Table III, no new CRs are found after applying post-processing algorithm since there are no degeneracies in this case. It can be seen that as the prediction horizon increases the the ratio of the computational time using the suggested method to the computational time using Algorithm 1 decreases significantly. This is due to the fact that the number of candidate active sets increases exponentially as the number of constraints increases, while the number of CRs and hence the corresponding geometric computations in post processing algorithm usually increases quite slower.

Algorithm 4 Post-processing algorithm

Phase I (initialization):

1) for every facet of all found critical regions using Algorithm 3, check whether the facet is located inside the feasible parameter space or not.
2) for all inner facets consider hyperplanes and find all coinciding hyperplanes.

Phase II (iterations):

3) for every hyperplane in a group of coinciding hyperplanes, consider the facet and check whether the union of the rest of facets in that group overlaps with the entire interior of the facet or not.
   - if part(s) of the facet is not coinciding with other facets, choose an arbitrary point on that part and generate a new parameter by stepping a a small distance from the facet in the direction of the normal to the facet. Find the combination of active constraints at the new parameter by solving a QP from [1]–[3], construct the control law and corresponding CR.

Example 3. As an example to check the efficiency of post-processing algorithm in finding missed CRs, consider again the fuel cell breathing control system with prediction horizon $N = 3$ and discretization time $T_d = 1\ sec.$ This in an example in which Algorithm 3 is not able to find all CRs. In fact 2 among the total 71 CRs are missed using Algorithm 3. One of the missed CRs in this example is related to $A_i = [11, 13, 16]$ (since none of its subsets are found to be optimal in the previous level). The neighbouring critical regions are those corresponding to the optimal active sets $A_i = [3, 11, 13]$, $A_i = [6, 11, 13]$, $A_i = [7, 11, 13]$ and $A_i = [10, 11, 13]$. It can be seen that two full-dimensional adjacent CRs differ in more than one constraints. The combination of optimal active constraints in the common facets between the adjacent CRs in this case are $A_i = [3, 11, 13, 16]$, $A_i = [6, 11, 13, 16]$, $A_i = [7, 11, 13, 16]$ and $A_i = [10, 11, 13, 16]$ which are all low-dimensional CRs due to LICQ failure ($n(A_i) > \tilde{m} = min\{m, q\}$, refer to [1] for more details).

TABLE III

COMPUTATIONAL TIME USING DIFFERENT METHODS FOR DIFFERENT PREDICTION HORIZONS FOR FOUR TANK SYSTEM

N	1	2	3	4	5
$t_{Alg.1}[s]$	0.9337	8.2482	251.5	9475.5	> 486400
$t_{Alg.3}[s]$	0.6836	3.5643	25.1	129.7	530.6
$t_{Alg.4}[s]$	10.4695	96.2353	455.9	1619.4	4723.5
$t_{Alg.3,4}[s]$	11.1531	99.7996	481	1749.1	5254.1
$\frac{t_{Alg.3,4}}{t_{Alg.1}}$	11.8184	12.0995	1.91	0.1846	< 0.0608

TABLE IV

COMPUTATIONAL TIME USING DIFFERENT METHODS FOR DIFFERENT PREDICTION HORIZONS FOR FUEL CELL SYSTEM

N	3	4	5	6	7	8
$t_{Alg.1}[s]$	5.64	35.91	131.78	457.62	981.49	3329.6
$n_{CR,Alg.1}$	71	133	186	230	261	277
$t_{Alg.3}[s]$	4.31	13.23	24.43	40.40	50.10	68.61
$n_{CR,Alg.3}$	69	131	184	230	261	277
$t_{Alg.3}[s]$	4.31	13.23	24.43	40.40	50.10	68.61
$n_{CR,Alg.3}$	69	131	184	230	261	277
$t_{Alg.3,4}[s]$	72.90	192.43	283.61	156.38	407.12	460.34
$n_{CR,Alg.3,4}$	71	133	186	230	261	277
$t_{Alg.5}[s]$	5.78	10.15	6.79	13.89	24.79	28.88
$n_{CR,Alg.5}$	23	26	4	1	1	3
$\frac{t_{Alg.3,4}}{tAlg.1}$	12.94	5.36	2.15	0.34	0.41	0.14

Alg.5: Enumeration-based approach suggested in [15]

Algorithm 3 does not find these combinations since we typically explore the tree up to level $\tilde{m}$ in combinatorial approach as mentioned before. The results of applying different algorithms on this system with different prediction horizons are presented in Table IV. It can be seen again that Algorithm 3 together with the post-processing algorithm can find all CRs while the ratio of the required time using the suggested method to the required time using Algorithm 1 decreases as the prediction horizon increases. Table IV includes also the simulation results using the enumeration-based approach suggested in [15] for solving the associated parametric linear complementarity problem which is implemented in MPT3. It can be seen that this method, although quite faster, is not able to find the complete solution while the correctness of the algorithm is usually more important than the computational time and complexity.

The computation time for different tasks in Algorithm 4 is shown in Fig. 4. It can be seen that the operations on the facets are the most time consuming tasks, while the matrix operations for polyhedra and finding the optimal active sets by solving QPs are executed relatively fast. Table IV indicates that the ratio of the computational time using the suggested method to the computational time using Algorithm 1 has a decreasing trend in general as the prediction horizon increases. This is because the number of feasible combinations of active constraints which are not optimal increases as the prediction horizon increases.

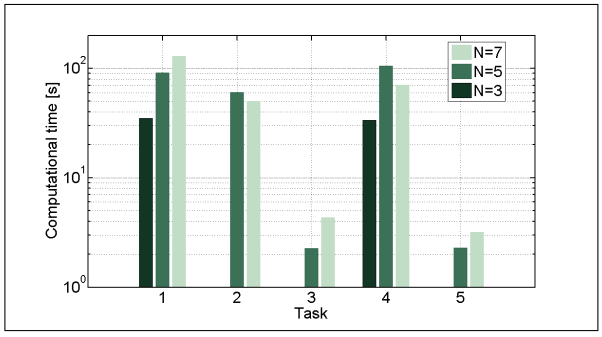

Fig. 4. Computational time for different tasks in Algorithm 4 for fuel cell system with 3 different prediction horizons (Task1: Extracting facets from polyhedra, Task2: Finding inner facets, Task3: Finding coinciding hyperplanes, Task4: Checking for each inner facet if it is entirely covered by other facets, Task5: Finding the optimal set in the adjacent CR)

V. CONCLUSIONS

In this paper, a new downward-upward strategy for exploring the combinatorial tree was suggested. This method exploits the supplementary information from geometric approaches and ignores all feasible combinations of active constraints that are not optimal. Therefore, the suggested method solves a relatively smaller number of LPs than the existing combinatorial approaches and is particularly well-suited for problems with large number of feasible but not optimal active sets. In the non-degenerate cases, identifying all critical regions is guaranteed. To ensure that all CRs are found and find any missed CR due to degeneracies in the problem, a post-processing algorithm was suggested based on the geometric exploration of unexplored parts in feasible space.

REFERENCES

[1] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[2] M. Baotić, "An efficient algorithm for multiparametric quadratic programming," *Technical Report AUT02-05, ETH Zürich, Institute für Automatik, Physikstraße 3, CH-8092, Switzerland*, 2002.

[3] P. Tøndel, T. A. Johansen, and A. Bemporad, "An algorithm for multi-parametric quadratic programming and explicit mpc solutions," *Automatica*, vol. 39, no. 3, pp. 489–497, 2003.

[4] M. M. Seron, G. C. Goodwin, and J. A. Doná, "Characterisation of receding horizon control for constrained linear systems," *Asian Journal of Control*, vol. 5, no. 2, pp. 271–286, 2003.

[5] A. Gupta, S. Bhartiya, and P. Nataraj, "A novel approach to multiparametric quadratic programming," *Automatica*, vol. 47, no. 9, pp. 2112–2117, 2011.

[6] C. Feller, T. A. Johansen, and S. Olaru, "An improved algorithm for combinatorial multi-parametric quadratic programming," *Automatica*, vol. 49, no. 5, pp. 1370–1376, 2013.

[7] C. Feller and T. A. Johansen, "Explicit mpc of higher-order linear processes via combinatorial multi-parametric quadratic programming," in *European Control Conference (ECC), 2013*, pp. 536–541.

[8] C. Feller, T. A. Johansen, and S. Olaru, "Combinatorial multi-parametric quadratic programming with saturation matrix based pruning," in *IEEE 51st Annual Conference on Decision and Control (CDC), 2012*, pp. 4562–4567.

[9] S. Olaru and D. Dumur, "Avoiding constraints redundancy in predictive control optimization routines," *IEEE Transactions on Automatic Control,*, vol. 50, no. 9, pp. 1459–1465, 2005.

[10] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[11] C. Feller, "A non-geometric approach to multi-parametric programming," Master's thesis, Diploma Thesis IST-80, University of Stuttgart, 2011.

[12] S. Hovland and J. T. Gravdahl, "Complexity reduction in explicit mpc through model reduction," in *Proceedings of the IFAC World Congress, Seoul, Korea*, 2008.

[13] J. Spjøtvold, E. C. Kerrigan, C. N. Jones, P. Tøndel, and T. A. Johansen, "On the facet-to-facet property of solutions to convex parametric quadratic programs," *Automatica*, vol. 42, no. 12, pp. 2209–2214, 2006.

[14] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. of the European Control Conference*, Zürich, Switzerland, July 17–19 2013, pp. 502–510, http://control.ee.ethz.ch/ mpt.

[15] M. Herceg, C. N. Jones, M. Kvasnica, and M. Morari, "Enumeration-based approach to solving parametric linear complementarity problems," *Automatica*, vol. 62, pp. 243–248, 2015.