Chapter 1
Multi-parametric Programming

Abstract. This chapter presents an overview of the approaches to solve multi-parametric programming problems. It is organized as follows. In Section 1.1, a general multi-parametric nonlinear programming (mp-NLP) problem is formulated and the Karush-Kuhn-Tucker (KKT) optimality conditions are presented. Then, the three main groups of methods to find a local minimum of a NLP problem for a given parameter vector are reviewed (Newton-type methods, penalty function methods and direct search methods). The Basic Sensitivity Theorem, which addresses the local regularity conditions for the optimal solution as function of the parameters is reviewed. Then, algorithms to find an approximate explicit solution of mp-NLP problems are described, which are based on an orthogonal (k–d tree) partition of the parameter space. Both convex and non-convex mp-NLP problems are considered. Procedures and heuristic rules for efficient splitting of a region in the parameter space and for handling the infeasible cases are formulated. In Section 1.2, a multi-parametric quadratic programming (mp-QP) problem is formulated and two approaches to find its exact explicit solution are described.

1.1 Multi-parametric Nonlinear Programming

There are two ways to address the parameter variations in mathematical programs: *sensitivity analysis*, which characterizes the change of the solution with respect to small perturbations of the parameters, and *parametric programming*, where the characterization of the solution is found for a full range of parameter values. Mathematical programs which depend only on one scalar parameter are referred to as *parametric programs*, while problems depending on a vector of parameters are referred to as *multi-parametric programs*.

The basic results within multi-parametric nonlinear programming (mp-NLP) can be found in [26]. Main topics in [26] include local regularity conditions, local sensitivity results and calculation of the parameter derivatives of the optimal solution vector.

1.1.1 Problem Formulation

Consider the nonlinear mathematical program dependent on a parameter x appearing in the objective function and in the constraints:

$$V^*(x) = \min_z f(z,x) \tag{1.1}$$

$$\text{s.t. } g(z,x) \leq 0\,, \tag{1.2}$$

where $z \in \mathbb{R}^s$ is the vector of optimization variables, $x \in \mathbb{R}^n$ is the vector of parameters, $f : \mathbb{R}^s \times \mathbb{R}^n \mapsto \mathbb{R}$ is the objective function, and $g : \mathbb{R}^s \times \mathbb{R}^n \mapsto \mathbb{R}^q$ is the constraints function. In (1.1), it is supposed that the minimum exists. It should be noted that the problem (1.1)–(1.2) includes only inequality constraints, and we remark that equality constraints can be incorporated with a straightforward modification since they are always included in the optimal active set.

Let X be a closed polytopic set of parameters, defined by $X = \{x \in \mathbb{R}^n \mid Ax \leq b\}$. In multi-parametric programming, it is of interest to characterize the solution or solutions of the mp-NLP problem (1.1)–(1.2) for the set X [26]. As described in [2], the solution of an mp-NLP problem is a triple $(V^*(x), Z^*(x), X_f)$, where:

i. the *set of feasible parameters* X_f is the set of all $x \in X$ for which the problem (1.1)–(1.2) admits a solution, i.e.:

$$X_f = \{x \in X \mid g(z,x) \leq 0 \text{ for some } z \in \mathbb{R}^s\} \ ; \tag{1.3}$$

ii. the *optimal value function* $V^* : X_f \mapsto \mathbb{R}$ associates with every $x \in X_f$ the corresponding optimal value of (1.1)–(1.2);
iii. the *optimal set* $Z^*(x)$ associates to each parameter $x \in X_f$ the corresponding set of optimizers $Z^*(x) = \{z \in \mathbb{R}^s \mid f(z,x) = V^*(x)\}$ of problem (1.1)–(1.2). If $Z^*(x)$ is a singleton for all $x \in X_f$, then $z^*(x) \triangleq Z^*(x)$ is called the *optimizer function*.

In this book we will assume that X_f is closed and $V^*(x)$ is finite for every $x \in X_f$. We denote by $g_i(z,x)$ the i-th component of the vector valued function $g(z,x)$.

Let z be a feasible point of (1.1)–(1.2) for a given parameter x. The *active constraints* are the constraints that fulfill (1.2) at equality, while the remaining constraints are called *inactive constraints*. The *active set* $\mathscr{A}(z,x)$ is the set of indices of the active constraints, i.e.:

$$\mathscr{A}(z,x) \triangleq \{i \in \{1, 2, \ldots, q\} \mid g_i(z,x) = 0\} \ . \tag{1.4}$$

The *optimal active set* $\mathscr{A}^*(x)$ is the set of indices of the constraints that are active for all $z \in Z^*(x)$, for a given $x \in X$, i.e.:

$$\mathscr{A}^*(x) \triangleq \{i \mid i \in \mathscr{A}(z,x),\ \forall z \in Z^*(x)\} \ . \tag{1.5}$$

Given an index set $\mathscr{A} \subseteq \{1, 2, \ldots, q\}$, the *critical region* $CR_{\mathscr{A}}$ is the set of parameters for which the optimal active set is equal to $\mathscr{A}$, i.e.:

$$CR_{\mathscr{A}} \triangleq \{x \in X \mid \mathscr{A}^*(x) = \mathscr{A}\} \ . \tag{1.6}$$

As it will be shown in Section 1.2, for strictly convex quadratic function f and linear constraints g, the critical regions $CR_{\mathscr{A}}$ are polyhedrons and the optimizer z^* is unique, piecewise affine, and continuous. However, for general nonlinear functions f and g, the exact solution of the multi-parametric programming problem (1.1)–(1.2) can not be found, and suboptimal methods for approximating its optimizer function $z^*(x)$ (or selection in case the optimizer function is not unique) are described in Section 1.1.5.

1.1.2 Optimality Conditions

For a given $x_0 \in X$, a local minimum z_0 of problem (1.1)–(1.2) has to satisfy the well known Karush-Kuhn-Tucker (KKT) first-order conditions [56]:

$$\nabla_z L(z_0, x_0, \lambda_0) = 0 \tag{1.7}$$

$$\mathrm{diag}(\lambda_0) g(z_0, x_0) = 0 \tag{1.8}$$

$$\lambda_0 \geq 0 \tag{1.9}$$

$$g(z_0, x_0) \leq 0 \, , \tag{1.10}$$

with associated Lagrange multiplier λ_0 and the Lagrangian defined as:

$$L(z, x, \lambda) \triangleq f(z, x) + \lambda^T g(z, x) \ . \tag{1.11}$$

Here, sufficient regularity (smoothness) is assumed, and this will be discussed later in Section 1.1.4.

Consider the optimal active set $\mathscr{A}_0$ at x_0, i.e. a set of indices to active constraints in (1.10). The above conditions are sufficient provided the following second order condition holds [56]:

$$v^T \nabla^2_{zz} L(z_0, x_0, \lambda_0) v > 0, \ \forall v \in \mathscr{F} - \{0\} \tag{1.12}$$

with $\mathscr{F}$ being the set of all directions where it is not given from the first order conditions if the objective function will increase or decrease:

$$\begin{aligned} \mathscr{F} = \ & \{v \in \mathbb{R}^s \mid \nabla_z g_{\mathscr{A}_0}(z_0, x_0) v \geq 0, \\ & \nabla_z g_i(z_0, x_0) v = 0, \ \text{for all } i \text{ with } (\lambda_0)_i > 0\} \ . \end{aligned} \tag{1.13}$$

The notation $g_{\mathscr{A}_0}$ means the rows of g with indices in $\mathscr{A}_0$.

1.1.3 *Nonlinear Programming Methods*

There exist various methods to numerically compute a local minimum z_0 of the problem (1.1)–(1.2) for a given $x_0 \in X$. The most commonly used methods can be classified in the following three groups.

1.1.3.1 Newton-Type Methods

The Newton type methods [20] appear to be the most widely used optimization methods. They try to find a point satisfying the KKT conditions (1.7)–(1.10) by using successive linearizations of the problem functions. The motivation behind this is that the linearized KKT system can be solved by using standard numeric linear algebra tools. Depending on how the conditions (1.8)–(1.10) (related to the imposed constraints) are treated, the two main groups of Newton type methods are the Sequential Quadratic Programming (SQP) methods and the Interior Point (IP) methods.

- *Sequential Quadratic Programming (SQP) methods.*
  The SQP methods iteratively solve the KKT system (1.7)–(1.10) by linearizing the nonlinear functions included in it. The resulting linearized KKT system at the $k+1$-th iteration can be considered as the KKT conditions of the following quadratic program (QP):

$$V^*_{qp}(z^k, x_0) = \min_z f_{qp}(z, z^k, x_0) \tag{1.14}$$

$$\text{s.t. } g(z^k, x_0) + \nabla_z g(z^k, x_0)(z - z^k) \le 0 \,, \tag{1.15}$$

  with the quadratic objective function given by:

$$f_{qp}(z, z^k, x_0) = \nabla_z f(z^k, x_0)^T z + \frac{1}{2}(z - z^k)^T \nabla_z^2 L(z^k, x_0, \lambda^k)(z - z^k) \,. \tag{1.16}$$

  Here, z^k and λ^k represent, respectively, the values of optimization variables and Lagrange multipliers, which solve the k-th sequential iteration of the KKT system (1.7)–(1.10). It is assumed that an initial guess z^0 is provided. In the case when the Hessian matrix $\nabla_z^2 L(z^k, x_0, \lambda^k)$ is positive semi-definite, the QP problem (1.14)–(1.16) is convex and its unique solution can be found.

  Typically, the QP sub-problem (1.14)–(1.16) is solved by using an Active Set (AS) method [56, 49], which identifies the active set of its solution z^*. The method begins with finding a feasible initial guess $\mathscr{A}_0(z, z^k, x_0) = \{i \in \{1, 2, ..., q\} \mid g_i(z^k, x_0) + \nabla_z g_i(z^k, x_0)(z - z^k) = 0\}$ of the active set by solving a linear programming problem [56]. In the next iteration, $\mathscr{A}_0(z, z^k, x_0)$ is refined by deleting a constraint from $\mathscr{A}_0(z, z^k, x_0)$ or by adding a constraint to $\mathscr{A}_0(z, z^k, x_0)$. In this way, the active set is refined iteratively until the optimal active set $\mathscr{A}^*(z^k, x_0)$ is found.

  There are several SQP methods which use approximations of the Hessian matrix $\nabla_z^2 L(z^k, x_0, \lambda^k)$ and the constraints Jacobian matrix $\nabla_z g(z^k, x_0)$, and they are referred to as quasi-Newton methods. They usually lead to slower convergence rates, but computationally less expensive iterations, in comparison to the exact SQP method. One of the quasi-Newton SQP methods is the method by Powell [61]. It uses exact constraints Jacobian matrix, but replaces the Hessian matrix $\nabla_z^2 L(z^k, x_0, \lambda^k)$ by an approximation H_k. Each new Hessian approximation H_{k+1} is obtained from the previous approximation H_k by an update formula that uses the difference of the Lagrange gradients, $\psi = \nabla_z L(z^{k+1}, x_0, \lambda^{k+1}) - \nabla_z L(z^k, x_0, \lambda^{k+1})$, and the step $\tau = z^{k+1} - z^k$ in order to obtain second order information in H_{k+1}. The most widely used update formula is the one by Broyden-Fletcher-Goldfarb-Shanno (BFGS) [56]:

$$H_{k+1} = H_k + \frac{\psi\psi^T}{\psi^T\tau} - \frac{H_k\tau\tau^T H_k}{\tau^T H_k \tau} \,. \tag{1.17}$$

Another successful quasi-Newton SQP method is the constrained Gauss-Newton method [21]. It uses approximations of the Hessian matrix, based on some Jacobian, and is applicable when the objective function is a sum of squares.

- *Interior Point (IP) methods.*
The IP methods represent an alternative way to solve the KKT system (1.7)–(1.10), which consists in replacing the nonsmooth KKT condition (1.8) by a smooth nonlinear approximation [16, 76]:

$$\nabla_z L(z_0, x_0, \lambda_0) = 0 \tag{1.18}$$

$$\lambda_{0,i} g_i(z_0, x_0) = \rho \ , \ i = 1, 2, ..., q \tag{1.19}$$

$$\lambda_0 \geq 0 \tag{1.20}$$

$$g(z_0, x_0) \leq 0 \,, \tag{1.21}$$

where $\rho > 0$ is a slack variable and $g_i(z_0, x_0)$ is the i-th constraint function. This system is then solved with a Newton-type method. The obtained solution is not a solution to the original NLP problem (1.1)–(1.2), but to the following problem [16, 76, 8]:

$$Q^*(x_0, \rho) = \inf_z [\, f(z, x_0) + \rho B(z, x_0) \,] \,. \tag{1.22}$$

Here, $B(z, x_0)$ is the so called barrier function, which is nonnegative and continuous over the region $\{z \in \mathbb{R}^s \,|\, g(z, x_0) < 0\}$ and approaches ∞ as the boundary of the feasible region $\{z \in \mathbb{R}^s \,|\, g(z, x_0) \leq 0\}$ is approached from the interior. Thus, the function $B(z, x_0)$ sets a barrier against leaving the feasible region. The solution of the barrier problem (1.22) requires for the optimization to start from a point inside the region $\{z \in \mathbb{R}^s \,|\, g(z, x_0) < 0\}$. The IP methods are also referred to as barrier function methods. They generate a sequence of feasible points whose limit is an optimal solution to the original problem (1.1)–(1.2) [8]. If the optimal solution occurs at the boundary of the feasible region, the procedure moves from the interior to the boundary. Typically, the barrier function $B(z, x_0)$ has the form [8]:

$$B(z,x_0) = \sum_{i=1}^{q} \frac{-1}{g_i(z,x_0)} \quad \text{or} \quad B(z,x_0) = -\sum_{i=1}^{q} \ln[-g_i(z,x_0)] \, . \tag{1.23}$$

The solution of problem (1.22) is closer to the true solution the smaller ρ gets. An important feature of the IP methods is that once a solution for a given ρ is found, the parameter ρ can be reduced by a constant factor and an accurate solution of the original NLP problem (1.1)–(1.2) is obtained after a limited number of Newton iterations [16, 76]. The relation between the original problem (1.1)–(1.2) and the barrier problem (1.22) is given by [8]:

$$V^*(x_0) = \lim_{\rho \to 0^+} Q^*(x_0,\rho) = \inf_{\rho > 0} Q^*(x_0,\rho) \, . \tag{1.24}$$

1.1.3.2 Penalty Function Methods

Methods using penalty functions transform a constrained problem into a single unconstrained problem or into a sequence of unconstrained problems [8]. The constraints are placed into the objective function via a penalty parameter in a way that penalizes any violation of the constraints. The penalty function methods are also referred to as the exterior penalty function methods, since they generate a sequence of infeasible points whose limit is an optimal solution to the original problem [8]. Consider the problem (1.1)–(1.2) for a given $x_0 \in X$. A penalty is desired only if the point z is not feasible, i.e., if $g(z,x_0) > 0$. A suitable unconstrained problem is therefore given by [8]:

$$J^*(x_0,\eta) = \inf_{z} [\, f(z,x_0) + \eta p(z,x_0) \,] \ \text{ s.t. } \ z \in \mathbb{R}^s \, , \tag{1.25}$$

where $p(z,x_0) = \sum_{i=1}^{q} [\max\{0,\, g_i(z,x_0)\}]^l$ is the so called penalty function, $l \geq 2$ is an integer, and $\eta > 0$ is a penalty parameter. If $g_i(z,x_0) \leq 0$, $\forall i = 1,2,...,q$ then $\max\{0,\, g_i(z,x_0)\} = 0$, $\forall i = 1,2,...,q$ and no penalty is incurred, i.e., $p(z,x_0) = 0$. On the other hand, if $g_i(z,x_0) > 0$, for some i, then $\max\{0,\, g_i(z,x_0)\} > 0$ and the penalty term $\eta p(z,x_0)$ is realized [8]. The condition $l \geq 2$ ensures that the penalty function $p(z,x_0)$ will be differentiable.

An important issue in the penalty function methods is the selection of the penalty parameter η. Consider the penalty problem [8]:

$$W^* = \sup_{\eta > 0} W(\eta) \, , \tag{1.26}$$

where $W(\eta) = J^*(x_0,\eta)$. The relation between the primal problem (1.1)–(1.2) and the penalty problem (1.26) is given by [8]:

$$V^*(x_0) = \sup_{\eta > 0} W(\eta) = \lim_{\eta \to \infty} W(\eta) \tag{1.27}$$

From this result it is clear that we can get arbitrarily close to the optimal objective value of the primal problem (1.1)–(1.2) by computing $W(\eta)$ for a sufficiently large

η. However, as pointed out in [8], there are computational difficulties associated with large penalty parameters, due to ill-conditioning. Therefore, most algorithms using penalty functions solve a sequence of problems (1.25) for an increasing sequence of penalty parameters. With each new value of the penalty parameter, an optimization technique is employed, starting with the optimal solution of problem (1.25) obtained for the parameter value chosen previously. Such an approach is sometimes referred to as *a sequential unconstrained minimization technique* [8]. More details about the penalty function methods can be found in [8].

For a given η, the optimization problem (1.25) can be solved by applying the steepest descent method [18]. Let $h(z,x_0,\eta) = f(z,x_0) + \eta p(z,x_0)$. Then, the steepest descent direction from z is $-\nabla_z h(z,x_0,\eta)$. With the method of steepest descent [18], the values of optimization variables at the $k+1$-th iteration are obtained by the formula:

$$z^{k+1} = z^k - \alpha \nabla_z h(z^k,x_0,\eta) \,, \tag{1.28}$$

where $\alpha > 0$ is the step length. In order for the steepest descent method to be successful, it is important to choose the step length α. One way to do this is to let $\alpha = \beta^m$, where $\beta \in (0,1)$ and $m \geq 0$ is the smallest nonnegative integer such that there is a sufficient decrease in $h(z,x_0,\eta)$. This means that:

$$h(z^k - \alpha \nabla_z h(z^k,x_0,\eta),x_0,\eta) - h(z^k,x_0,\eta) < -\mu \nabla_z h(z^k,x_0,\eta) \,, \tag{1.29}$$

where $\mu \in (0,1)$. This strategy, introduced in [3], is an example of a line search in which one searches on a ray from z^k in a direction in which $h(z,x_0,\eta)$ is locally decreasing. More details about the method of steepest descent can be found in [44]. Unfortunately, the methods based on steepest descent have slow local convergence, even for very simple functions [44]. This is due to the fact that the steepest descent direction scales with $h(z,x_0,\eta)$ and therefore the speed of convergence depends on conditioning and scaling. A good alternative to the steepest descent method is the conjugate gradient method [28, 1], which has improved local convergence properties. Also, the Newton-type methods can be successfully applied to solve the optimization problem (1.25).

1.1.3.3 Direct Search Methods

The direct search methods do not use or approximate the objective function's gradient, i.e. they represent derivative-free methods for optimization. These methods use values of the objective function and constraints taken from a set of sample points and use that information to continue the sampling. More precisely, the direct search methods consist in a sequential examination of trial solutions involving comparison of each trial solution with the best obtained up to that time together with a strategy for determining (as a function of earlier results) what the next trial solution will be [35]. There is a number of direct search methods for unconstrained optimization (see for example [44, 51]). However, here, the most widely used direct search methods for constrained nonlinear optimization are outlined.

- *The method of Box (the Complex method).*
The Complex method of Box [15] has been developed from the Simplex method [66, 54]. It requires for the NLP problem to be of the form:

$$V^*(x_0) = \min_z f(z, x_0) \tag{1.30}$$

$$\text{subject to :}$$

$$z_{l,i} \le z_i \le z_{u,i}\ ,\ i = 1, 2, ..., s \tag{1.31}$$

$$g_j(z, x_0) \le 0\ ,\ j = 1, 2, ..., q \tag{1.32}$$

where z_i is the i-th optimization variable, and $z_{l,i}$ and $z_{u,i}$ are the lower and upper bound on this variable.

It is assumed that an initial point z^1, which satisfies both constraints (1.31) and (1.32) is available. In this method, a set of $m \ge s+1$ points is used (referred to as complex), of which one is the given point z^1 (recall that s is the dimension of the optimization vector z). The further $(m-1)$ points required to set up the initial configuration are obtained one at a time by the use of pseudo-random numbers and ranges for each of the independent variables, i.e., $z_i = z_{l,i} + r_i(z_{u,i} - z_{l,i})$, where r_i is a pseudo-random deviate rectangularly distributed over the interval $(0, 1)$ [15]. A point so selected must satisfy the bound constraints (1.31), but need not satisfy all the functional constraints (1.32). If a functional constraint is violated, the trial point is moved halfway towards the centroid of those points already selected (where the given initial point is included) [15]. Ultimately, a satisfactory point will be found. It is assumed that the feasible region is convex. Proceeding in this way, $(m-1)$ points are found which satisfy all the constraints.

The function is evaluated at each vertex of the complex, and the vertex of the worst (maximal) function value is replaced by a point $\gamma \ge 1$ times as far from the centroid of the remaining points as the reflection of the worst point in the centroid (the new point is collinear with the rejected point and the centroid of the retained vertices) [15]. If this trial point is also the worst, it is moved halfway towards the centroid of the remaining points to give a new trial point. The above procedure is repeated until some constraint is violated. If a trial vertex does not satisfy the lower or the upper bound on some optimization variable z_i, that variable is reset to a value $z_{l,i} + \varepsilon$ or value $z_{u,i} - \varepsilon$ (depending on which bound has been violated), with ε being a small positive number. If a functional constraint $g_j(z, x_0)$ is violated, the trial point is moved halfway towards the centroid of the remaining points. Ultimately, a permissible point is found. Thus, as long as the complex has not collapsed into the centroid, progress will continue.

The idea of the Box's method is illustrated in Fig. 1.1 for the case when $s = 2$ and the number of points is $m = s+1 = 3$, i.e., for a simplex of points. The point z^2 is considered to be the worst point and c is the center of mass of the other two points (z^1 and z^3).
- *DIRECT method.*
The DIRECT algorithm (DIViding RECTangles) is a direct search method for *global* optimization, which was first introduced in [43, 42]. In [30, 27],

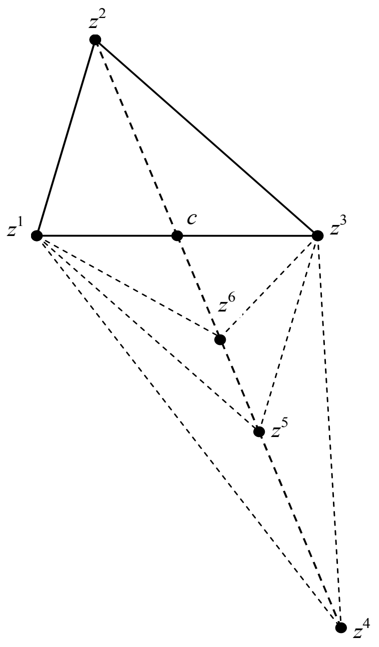

Fig. 1.1 The simplex with reflection of the point z^2 into point z^4 and two consecutive contractions (z^5, z^6) due to infeasibility.

rigorous new analysis and algorithmic improvements to the DIRECT algorithm have been presented. The DIRECT algorithm is a deterministic sampling algorithm developed in the spirit of Lipschitz optimization, and designed to overcome some of the shortcomings of traditional Lipschitzian algorithms (like the algorithm in [59]). One problem of the algorithm in [59] is its reliance on an accurate estimation of the Lipschitz constant. DIRECT solves this problem by replacing the Lipschitz constant with an adaptive internal parameter.

The DIRECT method solves the following mixed-integer nonlinear programming (MINLP) problem [42]:

$$V^*(x_0) = \min_z f(z,x_0) \tag{1.33}$$

$$\text{subject to}:$$

$$z_{l,i} \le z_i \le z_{u,i}\ ,\ i = 1, 2, ..., r \tag{1.34}$$

$$z_i \in \mathbb{Z}\ ,\ i = r+1, r+2, ..., s \tag{1.35}$$

$$g_j(z,x_0) \le 0\ ,\ j = 1, 2, ..., q \tag{1.36}$$

where $\mathbb{Z}$ is the set of integer numbers. The vector of optimization variables $z = [z_1, z_2, \dots, z_r, z_{r+1}, \dots, z_s]$ includes both real variables $(z_1, z_2, \dots, z_r)$ and integer variables $(z_{r+1}, z_{r+2}, \dots, z_s)$. The bounds on the variables limit the search to an s-dimensional hyper-rectangle. DIRECT proceeds by partitioning this rectangle into smaller rectangles, each of which has a sampled point at its center, i.e., a point where the functions have been evaluated [43, 42]. Fig. 1.2 shows the

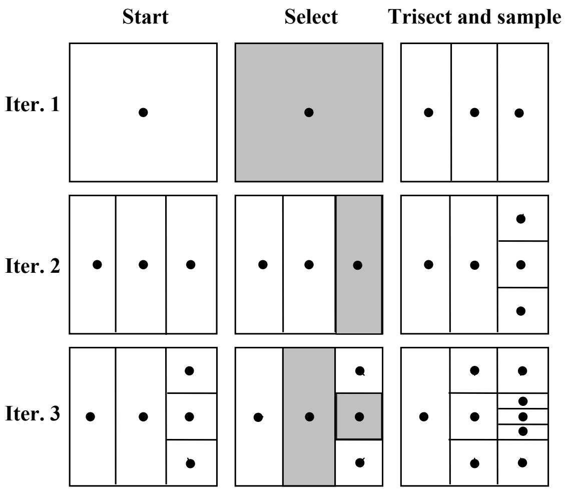

Fig. 1.2 Partitioning of the optimization variables space with DIRECT algorithm [43, 42].

first three iterations of DIRECT on a hypothetical problem with two optimization variables. At the start of each iteration, the space is partitioned into rectangles. DIRECT then selects one or more of these rectangles for further search using a technique described below. Finally, each selected rectangle is trisected along one of its long sides, after which the center points of the new rectangles are sampled. The key step in the algorithm is the selection of rectangles, since this determines how search effort is allocated across the space. The rectangles are selected using all possible relative weightings of *local* versus *global* search [43, 42]. First, it would be necessary to describe how the inequality constraints (1.36) are treated by the DIRECT method. The key to handling constraints in DIRECT is to work with an auxiliary function that combines information on the objective and constraint functions in a special manner [42]. To express this auxiliary function, an additional notation needs to be introduced. Let z_p be the center point of the p-th rectangle. Let φ_1, φ_2, ..., φ_q be positive weighting coefficients for the inequality constraints. Let the minimal value of the objective function at the current iteration be $V_{\min}(x_0)$. Let $\widetilde{V}$ be any value that satisfies $\widetilde{V} < V_{\min}(x_0) - \delta$, where $\delta > 0$. The auxiliary function, evaluated at the center of the p-th rectangle, is as follows [42]:

$$V_p^a(\widetilde{V}, x_0) = \max\{f(z_p, x_0) - \widetilde{V}, 0\} + \sum_{j=1}^{q} \varphi_j \max\{g_j(z_p, x_0), 0\} \qquad (1.37)$$

The first term of the auxiliary function represents a penalty for any deviation of the function value $f(z_p, x_0)$ above the value $\widetilde{V}$. The second term is a sum of weighted constraint violations. If $\widetilde{V}$ is the global minimum, the lowest possible value of the auxiliary function is zero and occurs only at the global minimum. At any other point, the auxiliary function is positive either due to suboptimality or infeasibility. For the global minimum to occur in the p-th rectangle, the auxiliary function must fall to zero starting from its positive value at the center point [42]. Moreover, the maximum distance over which this change can occur is the center-vertex distance d_p in the rectangle. Thus, to reach the global minimum in the p-th rectangle, the auxiliary function (1.37) must undergo a minimum rate of change, given by $e_p(\widetilde{V},x_0) = V_p^a(\widetilde{V},x_0)/d_p$ [42]. Since the point x_0 in the MINLP problem (1.33)–(1.36) is fixed, the rate of change function e_p depends only on the argument $\widetilde{V}$. The DIRECT procedure of selecting rectangles for further exploration identifies and selects all rectangles whose rate of change functions $e_p(\widetilde{V},x_0)$ participate in the lower envelope of all curves $V_p^a(\widetilde{V},x_0)/d_p$ for $\widetilde{V} < V_{\min}(x_0) - \delta$ [42]. More details about the DIRECT method can be found in [43, 42].

In this book, the DIRECT algorithm is applied to design explicit model predictive controllers for constrained nonlinear systems with quantized inputs (see Chapter 5).

1.1.4 Sensitivity Results

The solution of a mathematical program can behave in a variety of ways when perturbing the problem parameters. Depending on the problem, the solution may vary smoothly or change drastically for arbitrary small perturbations of parameter values. Let $x_0 \in X$, z_0 satisfy the KKT conditions, and $\mathscr{A}_0$ be the optimal active set at x_0. The Basic Sensitivity Theorem [26] gives local regularity conditions for the optimal solution, Lagrange multipliers and optimal objective function value as functions of x:

Theorem 1.1. *If:*

i). the functions $f(z,x)$ and $g(z,x)$ are twice continuously differentiable in z, and their gradients with respect to z and the constraints are once continuously differentiable in x in a neighborhood of (z_0,x_0),

ii). the second order sufficient condition (1.12) for a local minimum of (1.1)–(1.2) holds at z_0, with associated Lagrange multiplier λ_0,

iii) the active constraint gradients $\nabla_z g_{\mathscr{A}_0}(z_0,x_0)$ are linearly independent,

iv). $(\lambda_0)_i > 0$ when $g_i(z_0,x_0) = 0$ (strict complementary slackness),

Then:

a). z_0 is a local isolated minimizing point with unique associated Lagrange multiplier λ_0,

b). for x in the neighborhood of x_0, there exist unique, once continuously differentiable functions $z^*(x)$ and $\lambda^*(x)$ such that $z^*(x_0) = z_0$ and $z^*(x)$ is a locally unique local minimum of (1.1)–(1.2) with associated Lagrange multiplier $\lambda^*(x)$,

c). in a neighborhood of x_0, the set of active constraints is unchanged, strict complementary slackness holds, and the active constraint gradients at $z^*(x)$ remain linearly independent.

Related results for slightly different conditions, and extensions that show the existence and computation of directional derivatives of the solution with respect to x at x_0 can be found in [45, 50, 62] and others.

For the fixed active set $\mathscr{A}_0$ the KKT conditions (1.7)–(1.8) reduce to the following system of equations parameterized by x:

$$\nabla_z f(z(x),x) + \sum_{i\in\mathscr{A}_0} \lambda_i(x)\nabla_z g_i(z(x),x) = 0 \tag{1.38}$$

$$g_{\mathscr{A}_0}(z(x),x) = 0\,. \tag{1.39}$$

The functions $z(x)$ and $\lambda(x)$ implicitly defined by (1.38)–(1.39) are optimal only for those x where the active set $\mathscr{A}_0$ is optimal. Assuming z and λ are well defined on X, we characterize the critical region $CR_{\mathscr{A}_0}$ where the solution corresponding to the fixed active set $\mathscr{A}_0$ is optimal:

$$CR_{\mathscr{A}_0} \triangleq \{x \in X \mid \lambda(x) \geq 0,\ g(z(x),x) \leq 0\}\ . \tag{1.40}$$

There is a finite number of candidate active sets, so this result suggests a finite partition of X with a piecewise solution to the mp-NLP. Although explicit exact solutions cannot be found in the general nonlinear case, the above result indicates that it is meaningful to search for a continuous approximation to the optimal solution as a function of x. Continuity of the optimal solution depends on several assumptions that may be hard to verify in the general nonlinear case. However, many optimal control problems tend to lead to continuous solution functions.

1.1.5 Algorithms for Approximate Multi-parametric Nonlinear Programming

Consider the nonlinear multi-parametric program (1.1)–(1.2) dependent on the parameter x. Let X be a polytopic set of parameters, defined by $X = \{x \in \mathbb{R}^n \mid Ax \leq b\}$. In multi-parametric programming, it is of interest to characterize the solution of the mp-NLP problem (1.1)–(1.2) for the set X. The solution of an mp-NLP problem is a triple $(V^*(x), z^*(x), X_f)$ (see Section 1.1.1), where X_f is the set of feasible parameters, $V^*(x)$ is the optimal value function, and $z^*(x)$ is the optimizer function. It is assumed that X_f is closed and $V^*(x)$ is finite for every $x \in X_f$.

1.1.5.1 Approximate Solution of Convex mp-NLP

1. Convexity results.
The following assumption is made.

Assumption 1.1. *The functions f and g in the nonlinear multi-parametric program (1.1)–(1.2) are jointly convex functions of (z,x).*

The following basic result for convex multi-parametric programming was proved in [52]:

Theorem 1.2. *Consider the nonlinear multi-parametric program (1.1)–(1.2) and suppose that Assumption 1.1 holds. Then, X_f is a convex set and $V^* : X_f \mapsto \mathbb{R}$ is a convex and continuous function of x.*

Convexity of X_f and V^* is a direct consequence of the convexity of f and g, while continuity of V^* can be established under weaker conditions [26].

The main idea is to construct a feasible piecewise approximation to $z^*(x)$ on X, where the constituent functions pieces are defined on hyper-rectangles covering X. The accuracy of approximation is measured by the difference between the optimal and sub-optimal function values rather than the difference between the exact and approximation solutions. Since the optimal function value V^* cannot be assumed known, convexity is exploited to compute simple bounds to be used for constructing the approximate solution, similar to Chapter 9 in [26]. The method is applicable for piecewise linear (PWL) and piecewise nonlinear (PWNL) approximations.

Consider the vertices $\Theta = \{\theta_1, \theta_2, \ldots, \theta_{N_\theta}\}$ of any bounded polyhedron $X_0 \subseteq X_f$. Define the affine function $\bar{V}(x) = \bar{V}_0 x + \bar{l}_0$ as the solution to the following linear program (LP) [38]:

$$\min_{\bar{V}_0, \bar{l}_0} (\bar{V}_0 \theta + \bar{l}_0) \tag{1.41}$$

$$\text{subject to } \bar{V}_0 \theta_i + \bar{l}_0 \geq V^*(\theta_i), \; \forall \theta_i \in \Theta \; . \tag{1.42}$$

Likewise, define the convex PWL function [38]:

$$\underline{V}(x) = \max_{\theta_i \in \Theta} (V^*(\theta_i) + \nabla^T V^*(\theta_i)(x - \theta_i)) \tag{1.43}$$

If V^* is not differentiable at θ_i, then $\nabla V^*(\theta_i)$ is taken as any sub-gradient of V^* at θ_i [63]. $\underline{V}$ and $\bar{V}$ have the following properties [26, 39]:

Theorem 1.3. *Consider the nonlinear multi-parametric program (1.1)–(1.2) and suppose that Assumption 1.1 holds. Consider any bounded polyhedron $X_0 \subseteq X_f$. Then $\underline{V}(x) \leq V^*(x) \leq \bar{V}(x)$ for all $x \in X_0$.*

In [38], it is suggested to select a local linear approximation to the solution that minimizes the objective function approximation error subject to feasibility of the solution, similar to [9].

Lemma 1.1. *Consider the nonlinear multi-parametric program (1.1)–(1.2) and suppose that Assumption 1.1 holds. Consider any bounded polyhedron $X_0 \subseteq X_f$ with vertices $\Theta = \{\theta_1, \theta_2, \ldots, \theta_{N_\theta}\}$. If K_0 and h_0 solve the convex NLP:*

$$\min_{K_0, h_0} \sum_{i=1}^{N_\theta} (f(K_0 \theta_i + h_0, \theta_i) - V^*(\theta_i) + \mu \|K_0 \theta_i + h_0 - z^*(\theta_i)\|_2^2) \tag{1.44}$$

$$\textit{subject to } \; g(K_0 \theta_i + h_0, \theta_i) \leq 0, \; \forall \theta_i \in \Theta \; , \tag{1.45}$$

then $\widehat{z}_0(x) = K_0 x + h_0$ is feasible for the mp-NLP (1.1)–(1.2) for all $x \in X_0$.

In (1.44), $\mu > 0$ is a weighting coefficient. In general, the NLP defined in Lemma 1.1 need not have a feasible solution. As a partial remedy, the following result shows that at least for sufficiently small polyhedron X_0, feasibility can be guaranteed [38].

Lemma 1.2. *Consider the nonlinear multi-parametric program (1.1)–(1.2) and suppose that Assumption 1.1 holds. Let $X_0 \subseteq X_f$ be a sufficiently small bounded polyhedron with non-empty interior. Then there exists an affine function $\widetilde{z}(x)$ such that $g(\widetilde{z}(x),x) \leq 0$ for all $x \in X_0$.*

Proof ([38]). Since $X_0 \subseteq X_f$ is small, it follows from [26] that some unique and continuous feasible solution function $z(x)$ exists in a neighborhood that contains X_0. Since g is convex, it is straightforward to construct an affine support $\widetilde{z}(x)$. □

Since $\widehat{z}_0(x)$ defined in Lemma 1.1 is feasible in X_0, it follows that the suboptimal objective function $\widehat{V}(x) = f(\widehat{z}_0(x),x)$ is an upper bound on $V^*(x)$ in X_0 such that for all $x \in X_0$ we have:

$$0 \leq \widehat{V}(x) - V^*(x) \leq \varepsilon_0 \tag{1.46}$$

where:

$$\varepsilon_0 = -\min_{x \in X_0}(-\widehat{V}(x) + \underline{V}(x)) \; . \tag{1.47}$$

Computing ε_0 requires the solution of the NLP (1.47). If $\underline{V}$ is conservatively chosen as affine $\underline{V}(x) = V^*(\theta_i) + \nabla^T V^*(\theta_i)(x - \theta_i)$ (cf. (1.43)), this NLP is concave since $\widehat{V}$ is convex. Hence, the optimization can be done efficiently since X_0 is a polyhedron and it suffices to compare the solution at its vertices due to the concavity [36].

2. Algorithm for approximate explicit solution of convex mp-NLPs.
Consider a hyper-rectangle $X \subset \mathbb{R}^n$ where we seek to approximate the solution function $z^*(x)$ to the mp-NLP (1.1)–(1.2). In many problems of interest the approximate solution function will be evaluated in an embedded computer architecture under strict real-time requirements and with highly limited computational resources. In order to keep the computational complexity at a minimum, we require that the approximating function is PWL with a parameter space partition that is orthogonal and can be represented by a $k-d$ tree [14], [39, 31], such that the real-time search complexity is logarithmic with respect to the number of regions in the partition. The $k-d$ tree (Fig. 1.3) is a hierarchical data structure where a hyper-rectangle can be sub-divided into smaller hyper-rectangles allowing the local resolution to be adapted. When searching the tree, only one scalar comparison is required at each level. Initially the algorithm will consider the whole region $X_0 = X$. Under the convexity Assumption 1.1, the main idea of the approximate mp-NLP algorithm is to compute the solution of problem (1.1)–(1.2) at the 2^n vertices of the hypercube X_0, by solving up to 2^n NLPs. Based on these solutions, assuming they are all feasible, we compute a feasible local linear approximation function $\widehat{z}_0(x)$ to the optimal solution function $z^*(x)$, restricted to the hyper-rectangle X_0, using Lemma 1.1. If such an approximation exists, and the maximal objective function error ε_0 in X_0 is

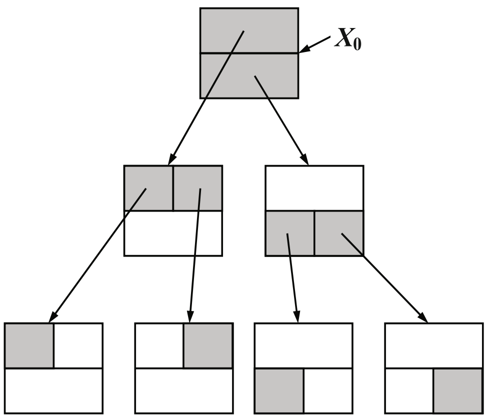

Fig. 1.3 $k-d$ tree partition of a rectangular region.

smaller than some prescribed tolerance $\bar{\varepsilon} > 0$, no further refinement of the region X_0 is needed. Otherwise, we split X_0 into two hyper-rectangles, and repeat the procedure for each of these.

Assume the tolerance $\bar{\varepsilon} > 0$ of the objective function approximation error is given. For simplicity, we consider uniform tolerance in this chapter. In later chapters, sometimes the tolerance will depend on x, which causes no problems. Denote with S_{X_0} the volume of a given hyper-rectangular region $X_0 \subset X \subset \mathbb{R}^n$, i.e. $S_{X_0} = \prod_{i=1}^{n} \Delta x_i$, where Δx_i is the size of X_0 along the axis x_i. Let $S_{\min}$ be the minimal allowed volume of the regions in the partition of X. A nonzero $S_{\min}$ is required to ensure termination of the algorithm in finite time. The following algorithm is proposed to determine an explicit approximate solution of convex mp-NLP (1.1)–(1.2) [38].

Algorithm 1.1. Explicit approximate solution of convex mp-NLP.

Input: Data to problem (1.1)–(1.2), the parameter μ (used in Lemma 1.1), the approximation tolerance $\bar{\varepsilon}$, the minimal allowed volume $S_{\min}$.
Output: Partition $\Pi = \{X_1, X_2, ..., X_{N_X}\}$ and associated PWL solution $\widehat{z}_\Pi = \{\widehat{z}_{X_1}, \widehat{z}_{X_2}, \dots, \widehat{z}_{X_{N_X}}\}$.

```
1. Initialize the partition to the whole hyper-rectangle, i.e., $\Pi = \{X\}$. Mark the
   hyper-rectangle $X$ as unexplored, and let $flag = 1$.
2. while $flag = 1$ do
3.    while $\exists$ unexplored hyper-rectangles in $\Pi$ do
4.       Select any unexplored hyper-rectangle $X_0 \in \Pi$ and compute its volume $S_{X_0}$.
5.       Solve problem (1.1)–(1.2) for $x$ fixed to each of the vertices $\theta_i$, $i = 1, \dots, N_\theta$
         of the hyper-rectangle $X_0$.
6.       if (1.1)–(1.2) has a feasible solution at all points $\theta_i$, $i = 1, \dots, N_\theta$ then
7.          Compute a linear approximation $\widehat{z}_0(x) = K_0 x + h_0$ using Lemma 1.1,
            as an approximation to be used in $X_0$.
8.          if a solution $\widehat{z}_0(x)$ was found then
9.             Compute the error bound $\varepsilon_0$, using (1.41)–(1.43), and (1.47).
```

```
10.             If $\varepsilon_0 > \bar{\varepsilon}$ and $S_{X_0} > S_{\min}$, mark the hyper-rectangle $X_0$ to be split.
                Otherwise, mark $X_0$ as explored and feasible.
11.           else
12.             If $S_{X_0} < S_{\min}$, mark $X_0$ infeasible and explored.
                Otherwise, mark $X_0$ to be split.
13.           end if
14.         else
15.           If $S_{X_0} < S_{\min}$, mark $X_0$ infeasible and explored.
              Otherwise, mark $X_0$ to be split.
16.         end if
17.     end while
18.     $flag := 0$
19.     if $\exists$ hyper-rectangles in $\Pi$ that are marked to be split then
20.         $flag := 1$
21.         while $\exists$ hyper-rectangles in $\Pi$ that are marked to be split do
22.           Select any hyper-rectangle $X_0 \in \Pi$ marked to be split.
23.           Split $X_0$ into two hyper-rectangles $X_1$ and $X_2$ by applying an heuristic
              splitting rule. Mark $X_1$ and $X_2$ unexplored, remove $X_0$ from $\Pi$,
              and add $X_1$ and $X_2$ to $\Pi$.
24.         end while
25.     end if
26. end while
```

The PWL approximation generated by Algorithm 1.1 is denoted $\widehat{z}_\Pi : \underline{X} \mapsto \mathbb{R}^s$, where $\underline{X}$ is the union of the hyper-rectangles where a feasible solution has been found. It is an inner approximation to X_f and the approximation accuracy is determined by the minimal allowed volume $S_{\min}$ of the regions. The boundary of the feasible region X_f can thus be approximated more closely by allowing smaller infeasible regions by choosing $S_{\min}$ small. We remark that $\widehat{z}_\Pi$ is generally not continuous.

Step 23 needs further specification of how a hyper-rectangle is being partitioned. A hyper-rectangle is split into two equal parts by an axis-orthogonal hyperplane that goes through its center. As in [31], the main idea is to select the hyperplane where the change of error between the solutions on each side of the hyperplane is largest (before splitting). This is implemented by comparing the solutions at the vertices of the hyper-rectangle. It is reasonable to expect that this heuristics may give a significant reduction in the error in both hyper-rectangles after splitting and its effectiveness is observed in a number of examples.

Theorem 1.4. *Consider the nonlinear multi-parametric program (1.1)–(1.2) and suppose that Assumption 1.1 holds and $S_{\min}$ is sufficiently small. Assume that the partitioning rule in step 23 guarantees that the error decreases by some minimum amount or factor at each split. Then Algorithm 1.1 terminates with an approximate solution function $\widehat{z}_\Pi$ that is feasible and satisfies $0 \le f(\widehat{z}_\Pi(x), x) - V^*(x) \le \bar{\varepsilon}$ for all $x \in \underline{X}$.*

Proof ([38]). If the algorithm terminates, the specified tolerance is met because of steps 9, 10, and 23. Since V^* is continuous, it is clear that a $k-d$ tree partition will lead to an approximation with arbitrary uniform accuracy provided the hypercubes are sufficiently small. According to Lemma 1.2, this approximation will be feasible, and since the partitioning rule ensures that the error decreases by some minimum amount or factor at each step, the algorithm will indeed terminate after a finite number of steps. $\square$

In [22], several alternative multi-parametric programming algorithms for explicit approximate solution of convex mp-NLP problems are discussed. Thus, in [24] a multi-parametric outer approximation algorithm for mp-NLP problems and multi-parametric mixed-integer nonlinear programming (mp-MINLP) problems is presented. A multi-parametric quadratic approximation algorithm is proposed in [37] and recently revisited in [23]. An approximate multi-parametric algorithm is proposed in [11], where the parameter space is divided into a set of simplices. Recently, a geometric vertex search algorithm is proposed in [53]. Some of these algorithms are extended to consider the non-convex case (see [58]).

1.1.5.2 Approximate Solution of Non-convex mp-NLP

If convexity does not hold (Assumption 1.1), then global optimization, e.g. [43, 42, 36], is generally needed in several steps of the algorithm to maintain its theoretical properties [38]:

(1) The NLP (1.1)–(1.2) must be solved using global optimization in step 5.
(2) The NLP (1.44)–(1.45) must be reformulated and solved using global optimization in step 7. It is not sufficient to impose the constraints at the vertices of the polyhedron X_0 if g is not convex. In order to resolve this problem, one may use (a conservative) convex underestimation in combination with global optimization as suggested in [26].
(3) The computation of the error bound ε_0 in step 9 assumes the knowledge of a lower bound $\underline{V}$ on the optimal objective function. The bound (1.43) does not necessarily hold if V^* is not convex. Again, convex underestimation and global optimization may be used.

On the other hand, one may argue in the favor of a computationally more efficient ad hoc approach to handle non-convex problems [38]. The reason for this is that an explicit representation of the approximate solution is available, which makes rigorous verification and validation of its properties possible. One heuristic approach is to include some interior points in addition to the set of vertices Θ when used in (1.43)–(1.45). Hence, any non-convexity related error in the computed bounds and approximation of the constraints are likely to be reduced. Moreover, based on the solutions of the associated NLPs one may locally estimate the Hessian of the optimal objective function at the points in Θ and may thus be able to detect if it is locally convex or non-convex, and adjust the number of additional interior points to be added to Θ. The introduction of such additional points does not necessarily lead to additional complexity of the PWL approximate solution, but may only serve to verify its accuracy [38].

Here, practical computational methods for explicit approximate solution of non-convex mp-NLP problems are presented. They don't necessarily lead to guaranteed properties of the explicit approximate solution, but when combined with verification and analysis methods may give a practical tool for explicit approximate solution of non-convex mp-NLPs.

1. Close-to-global solution of non-convex mp-NLPs.

In general, the objective function f can be non-convex with multiple local minima. Therefore, it would be necessary to apply an effective initialization of the mp-NLP problem (1.1)–(1.2) so to find a close-to-global solution. One possible way to obtain this is to find a close-to-global solution at a point $w_0 \in X_0$ by comparing the local minima corresponding to several initial guesses and then to use this solution as an initial guess at the neighboring points $w_i \in X_0$, $i = 1,2,...,N_1$, i.e. to propagate the solution. The following procedure is used to generate a set of points $W_0 = \{w_0, w_1, w_2, ..., w_{N_1}\} \subset X_0$ [32].

Procedure 1.1 (generation of set of points). *Consider any hyper-rectangle $X_0 \subseteq X$ with vertices $\Theta^0 = \{\theta_1^0, \theta_2^0, ..., \theta_{N_\theta}^0\}$ and center point w_0. Consider also the hyper-rectangles $X_0^j \subset X_0$, $j = 1,2,...,N_0$ with vertices respectively $\Theta^j = \{\theta_1^j, \theta_2^j, ..., \theta_{N_\theta}^j\}$, $j = 1,2,...,N_0$. Suppose $X_0^1 \subset X_0^2 \subset ... \subset X_0^{N_0}$. For each of the hyper-rectangles X_0 and $X_0^j \subset X_0$, $j = 1,2,...,N_0$, denote the set of its facets centers with $\Psi^j = \{\psi_1^j, \psi_2^j, ..., \psi_{N_\psi}^j\}$, $j = 0,1,2,...,N_0$. Define the set of all points $W_0 = \{w_0, w_1, w_2, ..., w_{N_1}\}$, where $w_i \in \left\{ \bigcup_{j=0}^{N_0} \Theta^j \right\} \cup \left\{ \bigcup_{j=0}^{N_0} \Psi^j \right\}$, $i = 1,2,...,N_1$.*

For a hyper-rectangle in the n-dimensional parameter space, the number of its vertices is $N_\theta = 2^n$ and the number of its facets centers is $N_\psi = 2n$. Therefore, the number of all points generated with Procedure 1.1 is $1 + (N_0 + 1)(N_\theta + N_\psi)$, where N_0 is the number of interior hyper-rectangles.

The following procedure is applied to search for a close-to-global solution at the points $w_i \in W_0$, $i = 0,1,2,...,N_1$ [32].

Procedure 1.2 (close-to-global solution of mp-NLP). *Consider any hyper-rectangle $X_0 \subseteq X$ with a set of points $W_0 = \{w_0, w_1, w_2, ..., w_{N_1}\}$ determined by applying Procedure 1.1. Then:*

a). Suppose local minima of the NLP (1.1)–(1.2) at the center point w_0 of X_0 have been computed. Then, determine a close-to-global solution of (1.1)–(1.2) at w_0 through the following minimization:

$$z^*(w_0) = \arg \min_{z \in \{z_1^{\text{local}}, z_2^{\text{local}}, ..., z_{N_z}^{\text{local}}\}} f(z, w_0) . \tag{1.48}$$

Here, z_i^{local}, $i = 1,2,...,N_z$ correspond to local minima of the objective function $f(z, w_0)$ obtained for a number of initial guesses z_i^0, $i = 1,2,...,N_z$.

b). Determine a close-to-global solution of the NLP (1.1)–(1.2) at the points $w_i \in W_0$, $i = 1, 2, \ldots, N_1$ in the following way:

1. *Let $z^*(w_0)$ be the close-to-global solution of the NLP (1.1)–(1.2) at the center point w_0, obtained by solving problem (1.48) in step a). Let $i = 1$.*
2. *Let $W^s = \{w_0, w_1, w_2, \ldots, w_{N_2}\} \subset W_0$ be the subset of points at which a feasible solution of the NLP (1.1)–(1.2) has been already determined.*
3. *Find the point $\widetilde{w} \in W^s$ that is most close to the point w_i, i.e., $\widetilde{w} = \arg \min_{w \in W^s} \|w - w_i\|$. Let the solution at $\widetilde{w}$ be denoted $z^*(\widetilde{w})$.*
4. *Solve the NLP (1.1)–(1.2) at the point w_i with initial guess for the optimization variables set to $z^*(\widetilde{w})$.*
5. *If a solution of the NLP (1.1)–(1.2) at the point w_i has been found, mark w_i as feasible and add it to the set W^s. Otherwise, mark w_i as infeasible.*
6. *Let $i = i + 1$. If $i \le N_1$, go to step 2. Otherwise, terminate.*

With some abuse of notation we do not distinguish between the global solution and the close-to-global solution, and denote both with $z^*(x)$. Procedure 1.2 is illustrated on Fig. 1.4. First, a close-to-global solution to the NLP (1.1)–(1.2) is determined at the center point w_0 of the hyper-rectangle X_0 (the case when no feasible solution at the center point w_0 exists is discussed later). Then, this solution is used as an initial guess when solving the NLP at the points $w_1, w_2, \ldots, w_8$ which represent the vertices and the facets centers of the smallest interior hyper-rectangle X_0^1. Then, the solutions at these points are used as initial guesses when solving the NLP at the points $w_9, w_{10}, \ldots, w_{16}$ which are the vertices and the facets centers of the interior

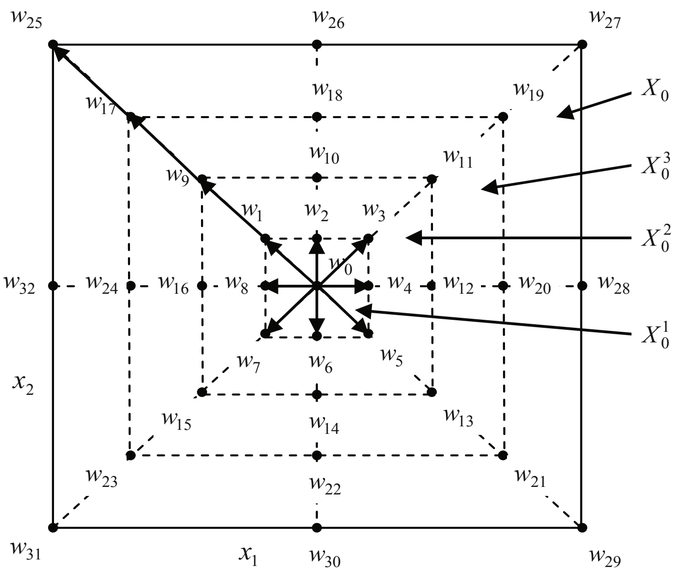

Fig. 1.4 Illustration of Procedure 1.2.

hyper-rectangle X_0^2. Next, the solutions at these points are used as initial guesses when solving the NLP at the points $w_{17}, w_{18}, \dots, w_{24}$ which represent the vertices and the facets centers of the interior hyper-rectangle X_0^3. At the end, the solutions at these points are used as initial guesses when solving the NLP at the points $w_{25}, w_{26}, ..., w_{32}$ which are the vertices and the facets centers of the hyper-rectangle X_0.

2. Computation of explicit approximate solution.
We restrict our attention to a hyper-rectangle $X \subset \mathbb{R}^n$ where we seek to approximate the solution $z^*(x)$ to the non-convex mp-NLP (1.1)–(1.2). Like in Section 1.1.5.1, we require that the parameter space partition is orthogonal and can be represented as a $k-d$ tree [14], [39, 31]. The main idea of the approach to explicit approximate solution of non-convex mp-NLPs is to construct a PWL approximation $\widehat{z}(x)$ to the close-to-global solution $z^*(x)$ on X, where the constituent affine functions are defined on hyper-rectangles covering X. It should be noted that sometimes it may be more appropriate to use a piecewise nonlinear (PWNL) approximation. In case of non-convexity, it would not be sufficient to impose the constraints only at the vertices of the hyper-rectangle X_0. This problem is resolved by including some interior points in addition to the set of vertices of X_0 [32]. These additional points represent the vertices and the facets centers of one or more hyper-rectangles contained in the interior of X_0 (see Procedure 1.1). Based on the solutions at all points, a local linear approximation $\widehat{z}_0(x) = K_0 x + h_0$ to the close-to-global solution $z^*(x)$, to be used as an approximation in the whole hyper-rectangle X_0, is determined by applying the following procedure [32].

Procedure 1.3 (computation of approximate solution). *Consider any hyper-rectangle $X_0 \subseteq X$ with a set of points $W_0 = \{w_0, w_1, w_2, \dots, w_{N_1}\}$ determined by applying Procedure 1.1. Compute K_0 and h_0 by solving the following NLP:*

$$\min_{K_0, h_0} \sum_{i=0}^{N_1} \left(f(K_0 w_i + h_0, w_i) - V^*(w_i) + \mu \|K_0 w_i + h_0 - z^*(w_i)\|_2^2\right) \tag{1.49}$$

$$\textit{subject to} \;\; g(K_0 w_i + h_0, w_i) \leq 0, \; \forall w_i \in W_0 \; . \tag{1.50}$$

In (1.49), the parameter $\mu > 0$ is a weighting coefficient. Note that the linear approximation $\widehat{z}_0(x) = K_0 x + h_0$, computed with Procedure 1.3, satisfies the constraints in the mp-NLP problem (1.1)–(1.2) only for the discrete set of points $W_0 \subset X_0$. In order to give an appropriate initialization of the NLP problem (1.49)–(1.50) for the region X_0, the already computed solutions of this problem in some of the neighboring regions can be used as initial guesses.

3. Estimation of error bounds. Suppose that a linear approximation $\widehat{z}_0(x) = K_0 x + h_0$ for the region X_0 has been computed by applying Procedure 1.3. Then it follows that the sub-optimal objective function $\widehat{V}(x) = f(\widehat{z}_0(x), x)$ is an approximate upper bound on $V^*(x)$ in X_0, such that for all $x \in X_0$, where $\widehat{z}_0(x)$ is feasible, we have:

$$0 \leq \widehat{V}(x) - V^*(x) \leq \varepsilon_0 \; . \tag{1.51}$$

As already mentioned, the objective function f can be non-convex with multiple local minima. Therefore, (1.51) is only valid if global solutions are found to all sub-problems and feasibility of $\widehat{z}_0(x)$ and $z^*(x)$ holds for all $x \in X_0$. The following procedure can be used to obtain an estimate $\widehat{\varepsilon}_0$ of the maximal approximation error ε_0 in X_0 [32].

Procedure 1.4 (computation of error bound approximation). *Consider any hyper-rectangle $X_0 \subseteq X$ with a set of points $W_0 = \{w_0, w_1, w_2, \ldots, w_{N_1}\}$ determined by applying Procedure 1.1. Compute an estimate $\widehat{\varepsilon}_0$ of the error bound ε_0 through the following maximization:*

$$\widehat{\varepsilon}_0 = \max_{i \in \{0,1,2,\ldots,N_1\}} (\widehat{V}(w_i) - V^*(w_i)) \,. \tag{1.52}$$

4. Procedure and heuristic rules for splitting a region.
The following procedure is applied to determine the split of a region X_0 for which a local linear approximation $\widehat{z}_0(x) = K_0 x + h_0$ is found, but the required accuracy $\bar{\varepsilon} > 0$ of objective function approximation is not achieved [32].

Procedure 1.5 (Determination of the split of a region). *Consider a hyper-rectangle X_0 and suppose that a local linear approximation $\widehat{z}_0(x) = K_0 x + h_0$ was found by applying Procedure 1.3. Suppose also that the required accuracy $\bar{\varepsilon}$ is not achieved. Then, determine the split of X_0 in the following way:*

1. *Let $j = 1$.*
2. *Consider splitting X_0 by a hyperplane through its center and orthogonal to the axis x_j. Denote the new hyper-rectangles with X_1^j and X_2^j.*
3. *Compute local linear approximations $\widehat{z}_1^j(x)$ and $\widehat{z}_2^j(x)$, candidates for use in X_1^j and X_2^j, respectively, by applying Procedure 1.3.*
4. *Compute estimates $\widehat{\varepsilon}_1^j$ and $\widehat{\varepsilon}_2^j$, respectively of the error bounds ε_1^j in X_1^j and ε_2^j in X_2^j, by applying Procedure 1.4. Let $\widehat{\varepsilon}^j = \widehat{\varepsilon}_1^j + \widehat{\varepsilon}_2^j$.*
5. *Let $j = j+1$. If $j \leq n$, go to step 2.*
6. *Split X_0 by a hyperplane through its center and orthogonal to the axis x_j where $\widehat{\varepsilon}^j$ is minimal.*

In step 4, the metric $\widehat{\varepsilon}^j = \widehat{\varepsilon}_1^j + \widehat{\varepsilon}_2^j$ could be replaced by other metrics such as $\widehat{\varepsilon}^j = \max(\widehat{\varepsilon}_1^j, \widehat{\varepsilon}_2^j)$.

The following rule is applied when no feasible solution to the NLP problem (1.1)–(1.2) was found at some of the points $w_i \in W_0$, $w_i \neq w_0$ [32]. Here, the set $W_0 = \{w_0, w_1, w_2, \ldots, w_{N_1}\}$ is defined in Procedure 1.1.

Heuristic splitting rule 1.1 (handling infeasibility). *Consider the following two cases:*

1. *The set of the feasible points in X_0 includes the center point w_0 and some non-empty subset of the points $w_i \in W_0$, $w_i \neq w_0$ (the set $W_0 = \{w_0, w_1, w_2, \ldots, w_{N_1}\}$ is defined in Procedure 1.1). Then, split X_0 into two types of hyper-rectangles by hyperplanes containing some of the feasible points $w_i \in W_0$:*

i. Hyper-rectangles $X_1^f, X_2^f, \dots, X_{N_f}^f$ containing only feasible points.

ii. Hyper-rectangles $X_1^{nf}, X_2^{nf}, \dots, X_{N_{nf}}^{nf}$ containing some infeasible points.

2. *The center point w_0 of X_0 is the only feasible point. Then, split X_0 on all parameter space axes by hyperplanes through w_0.*

This rule is illustrated in Fig. 1.5, where the hyper-rectangle X_0 will be split into the hyper-rectangles X_1^f with vertices $\{w_{24}, w_8, w_{17}, \alpha_1\}$, X_2^f with vertices $\{w_{23}, w_{21}, w_{24}, w_{20}\}$ and X_1^{nf} with vertices $\{w_8, w_{20}, \alpha_1, w_{19}\}$.

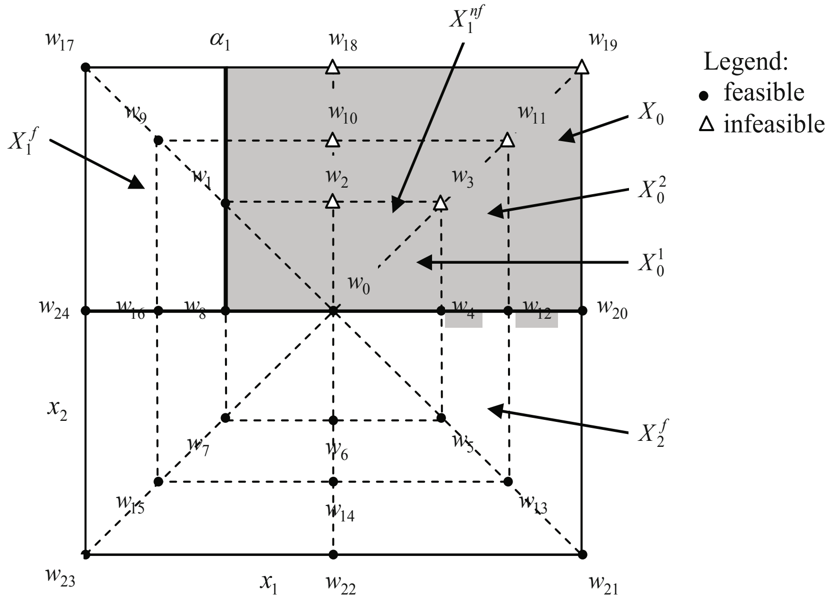

Fig. 1.5 Illustration of Heuristic splitting rule 1.1.

The following rule is applied when there is no feasible solution to the NLP problem (1.1)–(1.2) at the center point w_0 of the hyper-rectangle X_0 [32].

Heuristic splitting rule 1.2 (handling infeasibility). *If there is no feasible solution of the NLP problem (1.1)–(1.2) at the center point w_0 of X_0, split the hyper-rectangle X_0 by a hyperplane through w_0 and orthogonal to an arbitrary axis.*

The following rule is used when the NLP problem (1.49)–(1.50) in Procedure 1.3 has no solution [32].

Heuristic splitting rule 1.3 (handling infeasibility). *If the NLP problem (1.49)–(1.50) in Procedure 1.3 is infeasible, split the hyper-rectangle X_0 by a hyperplane through its center and orthogonal to an arbitrary axis.*

5. Algorithm for explicit approximate solution of non-convex mp-NLPs. Assume the tolerance $\bar{\varepsilon} > 0$ of the objective function approximation error is given. Denote with S_{X_0} the volume of a given hyper-rectangular region $X_0 \subset X \subset \mathbb{R}^n$, i.e. $S_{X_0} = \prod_{i=1}^{n} \Delta x_i$, where Δx_i is the size of X_0 along the variable x_i. Let $S_{\min} > 0$ be the minimal allowed volume of the regions in the partition of X. The following algorithm is proposed to determine an explicit approximate solution of non-convex mp-NLP (1.1)–(1.2) [32].

Algorithm 1.2. Explicit approximate solution of non-convex mp-NLP.

```
Input: Data to problem (1.1)–(1.2), the number $N_0$ of internal regions (used in
Procedure 1.1), the parameter $\mu$ (used in Procedure 1.3), the approximation
tolerance $\bar{\varepsilon}$, the minimal allowed volume $S_{\min}$.
Output: Partition $\Pi = \{X_1, X_2, ..., X_{N_X}\}$ and associated PWL solution function
$\hat{z}_\Pi = \{\hat{z}_{X_1}, \hat{z}_{X_2}, \dots, \hat{z}_{X_{N_X}}\}$.
1. Initialize the partition to the whole hyper-rectangle, i.e., $\Pi = \{X\}$. Mark the
   hyper-rectangle $X$ as unexplored, $flag := 1$.
2. while $flag = 1$ do
3.   while $\exists$ unexplored hyper-rectangles in $\Pi$ do
4.     Select any unexplored hyper-rectangle $X_0 \in \Pi$ and compute its volume $S_{X_0}$.
5.     Search for a close-to-global solution to problem (1.1)–(1.2) at the center
       point $w_0$ of $X_0$ by applying Procedure 1.2a.
6.     if a feasible solution was found to problem (1.1)–(1.2) at $w_0$ then
7.       Define a set of points $W_0 = \{w_0, w_1, w_2, \dots, w_{N_1}\}$ by applying
         Procedure 1.1.
8.       Search for a close-to-global solution to problem (1.1)–(1.2) for $x$ fixed to
         each of the points $w_i$, $i = 1, 2, \dots, N_1$ by applying Procedure 1.2b.
9.       if (1.1)–(1.2) has a feasible solution at all points $w_i$, $i = 1, \dots, N_1$ then
10.         Search for a linear approximation $\hat{z}_0(x) = K_0 x + h_0$ using
            Procedure 1.3, as an approximation to be used in $X_0$.
11.         if a solution $\hat{z}_0(x)$ was found then
12.           Compute an estimate $\hat{\varepsilon}_0$ of the error bound $\varepsilon_0$ in $X_0$ by applying
              Procedure 1.4.
13.           If $\hat{\varepsilon}_0 > \bar{\varepsilon}$ and $S_{X_0} > S_{\min}$, mark the hyper-rectangle $X_0$ to be split.
              Otherwise, mark $X_0$ as explored and feasible.
14.         else
15.           If $S_{X_0} < S_{\min}$, mark $X_0$ infeasible and explored.
              Otherwise, mark $X_0$ to be split.
16.         end if
17.       else
18.         If $S_{X_0} < S_{\min}$, mark $X_0$ infeasible and explored.
            Otherwise, mark $X_0$ to be split.
19.       end if
20.     else
```

```
21.             **If** $S_{X_0} < S_{\min}$, mark $X_0$ infeasible and explored.
                **Otherwise**, mark $X_0$ to be split.
22.          **end if**
23.       **end while**
24.       $flag := 0$
25.       **if** $\exists$ hyper-rectangles in $\Pi$ that are marked to be split **then**
26.          $flag := 1$
27.          **while** $\exists$ hyper-rectangles in $\Pi$ that are marked to be split **do**
28.             Select any hyper-rectangle $X_0 \in \Pi$ marked to be split.
29.             Split $X_0$ into hyper-rectangles $X_1, \dots, X_{N_s}$ by applying the heuristic
                splitting rules. Mark $X_1, \dots, X_{N_s}$ unexplored, remove $X_0$ from $\Pi$,
                and add $X_1, \dots, X_{N_s}$ to $\Pi$.
30.          **end while**
31.       **end if**
32. **end while**
```

1.2 Convex Multi-parametric Quadratic Programming

1.2.1 Problem Formulation

Consider the convex quadratic mathematical program dependent on a parameter x:

$$V^*(x) = \min_z \frac{1}{2} z^T H z \tag{1.53}$$

$$\text{s.t. } Gz \leq W + Sx\,, \tag{1.54}$$

where $z \in \mathbb{R}^s$ is the vector of optimization variables, $x \in \mathbb{R}^n$ is the vector of parameters, and $H \in \mathbb{R}^{s \times s}$, $G \in \mathbb{R}^{q \times s}$, $W \in \mathbb{R}^q$, and $S \in \mathbb{R}^{q \times n}$ are matrices. Here, it is supposed that $H \succ 0$, which leads to a strictly convex multi-parametric quadratic programming (mp-QP) problem (1.53)–(1.54). The case when the multi-parametric programming problem (1.53)–(1.54) is only convex, i.e. $H \succeq 0$, is considered in [73, 69].

Let X be a polytopic set of parameters, defined by $X = \{x \in \mathbb{R}^n \mid Ax \leq b\}$. In parametric programming, it is of interest to characterize the solution of the mp-QP problem (1.53)–(1.54) for the set X. The solution of an mp-QP problem is a triple $(V^*(x), z^*(x), X_f)$ (see Section 1.1.1), where X_f is the set of feasible parameters, $V^*(x)$ is the optimal value function, and $z^*(x)$ is the optimizer function. It is assumed that X_f is closed and $V^*(x)$ is finite for every $x \in X_f$.

In [12, 13], an algorithm has been developed, which expresses the solution $z^*(x)$ and the optimal value $V^*(x)$ of the mp-QP problem (1.53)–(1.54) as an explicit function of the parameters x, and the analytical properties of these functions have been characterized. In particular it has been proved that the solution $z^*(x)$ is a continuous piecewise linear function of x in the following sense [12, 13]:

Definition 1.1. A function $z(x) : X \mapsto \mathbb{R}^s$, where $X \subseteq \mathbb{R}^n$ is a polyhedral set, is piecewise linear if it is possible to partition X into convex polyhedral regions, CR_i, and $z(x) = K_i x + h_i$, $\forall x \in CR_i$.

Piecewise quadraticity is defined analogously by letting $z(x)$ be a quadratic function $x^T Q_i x + K_i x + h_i$.

1.2.2 Optimality Conditions

The solution of mp-QP problems can be approached by employing the principles of multi-parametric nonlinear programming and in particular the first-order Karush-Kuhn-Tucker (KKT) optimality conditions, which lead to the Basic Sensitivity Theorem (see Section 1.1). Instead, in [12, 13] a more direct approach has been adopted which exploits the linearity of the constraints and the fact that the function to be minimized is quadratic. The approach [12, 13] is described as follows. In order to start solving the mp-QP problem, an initial vector x_0 inside the polyhedral set X of parameters is needed, such that the QP problem (1.53)–(1.54) is feasible for $x = x_0$. Such a vector can be found for instance by solving the linear program (LP) [12, 13]:

$$\max_{x,z,\varepsilon} \varepsilon \tag{1.55}$$

$$\text{subject to :}$$

$$Gz - Sx + \varepsilon \leq W \tag{1.56}$$

$$\varepsilon \geq 0 \tag{1.57}$$

$$x \in X \; . \tag{1.58}$$

If the LP (1.55)–(1.58) is infeasible, then the QP problem (1.53)–(1.54) is infeasible for all $x \in X$. Otherwise, the QP problem (1.53)–(1.54) is solved with $x = x_0$ in order to obtain the corresponding optimal solution z_0. Such a solution is unique because $H \succ 0$ and therefore uniquely determines a set of active constraints $\tilde{G}z_0 = \tilde{S}x_0 + \tilde{W}$ among the constraints (1.54). Let $\tilde{G}$, $\tilde{S}$ and $\tilde{W}$ denote the rows of G, S and W corresponding to the active constraints. Then, the following theorem is proved [12, 13]:

Theorem 1.5. *Let $H \succ 0$. Consider a combination of active constraints $\tilde{G}$, $\tilde{S}$, $\tilde{W}$ and assume that the rows of $\tilde{G}$ are linearly independent. Let CR_0 be the set of all vectors x for which such a combination is active at the optimum (CR_0 is referred to as critical region). Then, the optimal z and the associated vector of Lagrange multipliers λ are uniquely defined linear functions of x over CR_0.*

Proof ([13]). The first-order KKT conditions for the mp-QP are given by:

$$Hz + G^T\lambda = 0 \; , \; \lambda \in \mathbb{R}^q \tag{1.59}$$

$$\lambda_i(G^i z - W^i - S^i x) = 0 \; , \; i = 1, 2, \ldots, q \tag{1.60}$$

$$\lambda \geq 0 \; , \tag{1.61}$$

where the superscript i denotes the i-th row. Equality (1.59) is solved for z:

$$z = -H^{-1}G^T\lambda \tag{1.62}$$

and the result is substituted into (1.60) to obtain the complementary slackness condition:

$$\lambda_i(-G^iH^{-1}G^{iT}\lambda_i - W^i - S^ix) = 0\ ,\ i = 1,2,\ldots,q \tag{1.63}$$

Let $\breve{\lambda}$ and $\tilde{\lambda}$ denote the Lagrange multipliers corresponding to inactive and active constraints, respectively. For inactive constraints $\breve{\lambda} = 0$. For active constraints $-\tilde{G}H^{-1}\tilde{G}^T\tilde{\lambda} - \tilde{W} - \tilde{S}x = 0$ and therefore:

$$\tilde{\lambda} = -(\tilde{G}H^{-1}\tilde{G}^T)^{-1}(\tilde{W} + \tilde{S}x)\ , \tag{1.64}$$

where $\tilde{G}$, $\tilde{W}$, $\tilde{S}$ correspond to the set of active constraints and $(\tilde{G}H^{-1}\tilde{G}^T)^{-1}$ exists because the rows of $\tilde{G}$ are linearly independent. Thus λ is a linear function of x for all $x \in CR_0$, where the active constraints set is optimal. By substituting $\tilde{\lambda}$ from (1.64) into (1.62), it is obtained:

$$z^*(x) = H^{-1}\tilde{G}^T(\tilde{G}H^{-1}\tilde{G}^T)^{-1}(\tilde{W} + \tilde{S}x) \tag{1.65}$$

and it is noted that z^* is also a linear function of x in CR_0. □

Theorem 1.5 characterizes the solution only locally in the neighborhood of a specific x_0, as it does not provide the construction of the set CR_0 where this characterization remains valid. On the other hand, this region can be characterized immediately [12, 13]. The variable z from (1.65) must satisfy the constraints (1.54):

$$GH^{-1}\tilde{G}^T(\tilde{G}H^{-1}\tilde{G}^T)^{-1}(\tilde{W} + \tilde{S}x) \le W + Sx \tag{1.66}$$

and by (1.61) the Lagrange multipliers in (1.64) must remain nonnegative:

$$-(\tilde{G}H^{-1}\tilde{G}^T)^{-1}(\tilde{W} + \tilde{S}x) \ge 0 \tag{1.67}$$

as x varies. After removing the redundant inequalities from (1.66) and (1.67), a compact representation of CR_0 is obtained. Obviously, CR_0 is a polyhedron in the x-space and represents a subset of X such that the combination of active constraints at the minimizer remains unchanged (Fig. 1.6(a)). Then, the algorithm in [12, 13] continues with the division of the rest of the parameter space $CR^{rest} = X - CR_0$ as in Fig. 1.6(b) and (c) by reversing one by one the hyperplanes defining the critical region CR_0. Iteratively each new region R_i is subdivided in a similar way as was done with X. An effective approach for partitioning the rest of the space was proposed in [25]. The following theorem justifies such a procedure to characterize the rest of the region CR^{rest} [13].

Theorem 1.6. *Let $Y \subseteq \mathbb{R}^n$ be a polyhedron, and $CR_0 = \{x \in Y \mid Ax \le b\}$ a polyhedral subset of Y, $CR_0 \neq \emptyset$. Also let:*

$$R_i = \{x \in Y \mid A^i x > b^i, A^j x \le b^j\,,\ \forall j < i\}\,,\ i = 1, 2, \ldots, m\,, \tag{1.68}$$

where $m = \dim(b)$, and let $CR^{rest} = \bigcup_{i=1}^{m} R_i$. Then:
(i) $CR^{rest} \bigcup CR_0 = Y$.
(ii) $CR_0 \bigcap R_i = \emptyset$, $R_i \bigcap R_j = \emptyset$, $\forall i \neq j$, i.e., $\{CR_0, R_1, \ldots, R_m\}$ is a partition of Y.

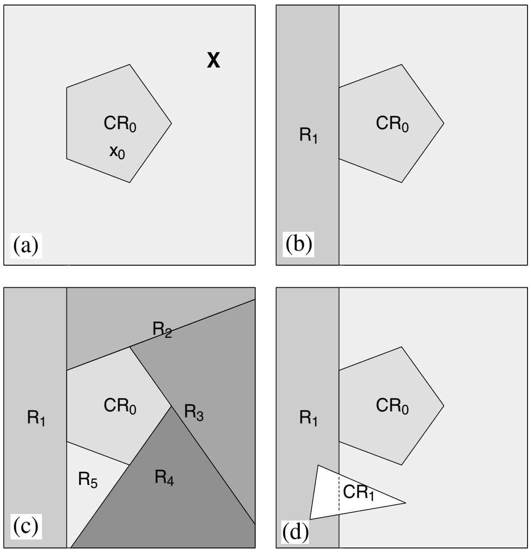

Fig. 1.6 Parameter space exploration strategy in [12, 13].

The properties of the set of feasible parameters $X_f \subseteq X$ (i.e. the set of parameters $x \in X$ such that a feasible solution $z^*(x)$ exists to the optimization problem (1.53)–(1.54), the value function $V^*(x)$ and the solution $z^*(x)$ are formulated in the following theorem [13]:

Theorem 1.7. *Consider the convex multi-parametric quadratic program (1.53)–(1.54) with $H \succ 0$, X convex. Then the set of feasible parameters $X_f \subseteq X$ is convex, the optimizer $z^*(x) : X_f \mapsto \mathbb{R}^s$ is continuous and piecewise linear and the value function $V^*(x) : X_f \mapsto \mathbb{R}$ is continuous, convex and piecewise quadratic.*

1.2.3 Algorithms for Exact Convex Multi-parametric Quadratic Programming

Based on the above results, the main steps of the off-line mp-QP solver are outlined in the following algorithm [13]:

Algorithm 1.3. Exact mp-QP.

```
Step 1. Let the current region be the whole polyhedron $X \subseteq \mathbb{R}^n$.
Step 2. Choose a vector $x_0$ in the current region by solving the linear program (1.55)–(1.58).
Step 3. For $x = x_0$, compute the corresponding optimal solution $(z_0, \lambda_0)$ by solving a QP.
Step 4. Determine the set of active constraints when $z = z_0$, $x = x_0$, and build $\tilde{G}$, $\tilde{W}$, $\tilde{S}$.
Step 5. If $r = \operatorname{rank}\tilde{G}$ is less than the number $l$ of rows of $\tilde{G}$, take a subset of $r$ linearly independent rows and redefine $\tilde{G}$, $\tilde{W}$, $\tilde{S}$ accordingly.
Step 6. Determine $\tilde{\lambda}(x)$, $z^*(x)$ from (1.64) and (1.65).
Step 7. Characterize the $CR_0$ from (1.66) and (1.67).
Step 8. Define and partition the rest of the region as illustrated in Fig. 1.6.
Step 9. For each nonempty new sub-region, go to step 2.
Step 10. When all regions have been explored, for all polyhedral regions where $z^*(x)$ is the same and whose union is a convex set, compute such a union.
```

In conclusion, Algorithm 1.3 provides the explicit solution $z^*(x)$ to the mp-QP problem (1.53)–(1.54), as the piecewise affine function:

$$z^*(x) = K_i x + h_i \ \text{ if } D_i x \leq d_i\,,\ i = 1, 2, \ldots, N_r\,, \tag{1.69}$$

where the polyhedral sets $D_i x \leq d_i$, $i = 1, 2, \ldots, N_r$ are critical regions that form a partition of the given set of states X.

1.2.3.1 Efficient Implementation of the Exact Approach to Explicit Solution of mp-QP Problems

1. Main theoretical result.

As noted in [72], the main drawback of this algorithm is that the regions R_i are not related to optimality, as they can split some of the critical regions like CR_1 in Fig. 1.6(d). A consequence is that CR_1 will be detected at least twice. The approach in [72] modifies the explicit approach in [12, 13] by analyzing several properties of the geometry of the polyhedral partition and its relation to the combination of active constraints at the optimum of the quadratic program. Based on that, they derive a new exploration strategy for sub-dividing the parameter space, which aims to:

(1) Avoid unnecessary partitioning.
(2) Avoid the solution to LP problems for determining an interior point in each new region of the parameter space.
(3) Avoid the solution to the QP problem for such an interior point.

As a consequence, there is a significant improvement of efficiency with respect to the algorithm in [12, 13]. Before describing the main idea of the approach in [72], some definitions are made [72]:

Definition 1.2. Let $z^*(x)$ be the optimal solution to (1.53)–(1.54) for a given x. We define *active constraints* the constraints with $G^i z^*(x) - W^i - S^i x = 0$ and *inactive constraints* the constraints with $G^i z^*(x) - W^i - S^i x < 0$. The *optimal active set* $\mathscr{A}^*(x)$ is the set of indices of active constraints at the optimum $\mathscr{A}^*(x) = \{i \mid G^i z^*(x) = W^i + S^i x\}$ (a superscript index is used to denote a row of a matrix). We also define as *weakly active constraint* an active constraint with an associated zero Lagrange multiplier λ_i and as *strongly active constraint* an active constraint with a positive Lagrange multiplier λ_i.

Definition 1.3. For an active set, we say that the *linear independence constraint qualification* (LICQ) holds if the set of active constraint gradients are linearly independent, i.e. $\tilde{G}$ has full row rank.

Below, the linear expression of the PWL function $z^*(x)$ over the critical region CR_k is denoted by $z_k^*(x)$.

Definition 1.4. Two polyhedra are called *neighboring* polyhedra if they have a common facet.

Definition 1.5. Let a polyhedron X be represented by $A_0 x \leq b$. We say that $A_0^i x \leq b^i$ is *redundant* if $A_0^j x \leq b^j, \forall j \neq i \Rightarrow A_0^i x \leq b^i$ (i.e. it can be removed from the description of the polyhedron). The inequality i is *redundant* with degree h if it is redundant but there exists a h-dimensional subset Y of X such that $A_0^i x = b^i$ for all $x \in Y$.

Let us consider a hyperplane defining the common facet between two polyhedra CR_0, CR_i in the optimal partition of the state space. There are two different kinds of hyperplanes [72]. The first (Type I) are those described by (1.66), which represent a non-active constraint that becomes active at the optimum as x moves from CR_0 to CR_i. This means that if a polyhedron is bounded by a hyperplane which originates from (1.66), the corresponding constraint will be activated on the other side of the facet defined by this hyperplane. In addition, the corresponding Lagrange multiplier may become positive. The other kind (Type II) of hyperplanes which bounds the polyhedra are those described by (1.67). In this case, the corresponding constraint will be non-active on the other side of the facet defined by this hyperplane. This is formulated in the following theorem [72]:

Theorem 1.8. *Consider an optimal active set $\{i_1, i_2, \ldots, i_k\}$ and its corresponding n-minimal representation of the critical region CR_0 obtained by (1.66)–(1.67) after removing redundant inequalities. Let CR_i be a full-dimensional neighboring critical region to CR_0 and assume LICQ holds on their common facet $\Phi = CR_0 \bigcap \Psi$ where Ψ 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 Ψ 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 $\{i_1, \dots, i_k, i_{k+1}\}$.

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

In degenerate cases, when the LICQ condition does not hold or there are weakly active constraints, Theorem 1.8 provides no conclusion. In particular, when moving across the facet of one critical region there may not be a single unique critical region that shares the same facet, [67]. As discussed in [72, 73, 67, 69], the method in [12, 13] is effective to handle these special cases.

2. Example.
The example represents a Model Predictive Control (MPC) problem for a double integrator [72, 40], which is transformed into the equivalent mp-QP problem (1.53)–(1.54) with H, G, W, S given by:

$$H = \begin{bmatrix} 1.079 & 0.076 \\ 0.076 & 1.073 \end{bmatrix} \tag{1.70}$$

$$G^T = \begin{bmatrix} 1 & 0 & -1 & 0 & 0.05 & 0.05 & -0.05 & -0.05 \\ 0 & 1 & 0 & -1 & 0 & 0.05 & 0 & -0.05 \end{bmatrix} \tag{1.71}$$

$$W^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 0.5 & 0.5 & 0.5 & 0.5 \end{bmatrix} \tag{1.72}$$

$$S^T = \begin{bmatrix} 1.0 & 0.9 & -1.0 & -0.9 & 0.1 & 0.1 & -0.1 & -0.1 \\ 1.4 & 1.3 & -1.4 & -1.3 & -0.9 & -0.9 & 0.9 & 0.9 \end{bmatrix} \tag{1.73}$$

The partitioning starts with finding the region where no constraints are active. As the mp-QP is created from a feasible MPC problem, the empty active set will be optimal in some full-dimensional region ($\mathscr{A}_0 = \emptyset$ and $\tilde{G}$, $\tilde{W}$ and $\tilde{S}$ are empty matrices, $z^*(x) = 0$). This critical region is then described by $0 \le W + Sx$ which contains 8 inequalities. Two of these inequalities are redundant with degree 0 (#2 and #4), the remaining 6 hyperplanes are facet inequalities of the polyhedron (see Fig. 1.7(a)). By crossing the facet given by Ψ_1, defined by inequality 1 and of Type I, as predicted by Theorem 1.8 the optimal active set across this facet is $\mathscr{A}_1 = \{1\}$, which leads to the critical region CR_1 (see Fig. 1.7(b)). After removing redundant inequalities we are left with an n-minimal representation of CR_1 containing 4 facets. The first of these is of Type II, $\lambda_1(x) = 0$. The other three are of Type I. These are inequalities #2, #6 and #7. Consider first the other side of the facet which comes from $\lambda_1(x) = 0$, see Fig. 1.7(c). The region should not have constraint 1 active, so the optimal active set is $\mathscr{A}_2 = \emptyset$. This is the same combination of active constraints as $\mathscr{A}_0$, as expected, so $\mathscr{A}_2$ is not pursued. Next, consider crossing the respective facets of inequalities #2, #6 and #7, see Fig. 1.7(d)–Fig. 1.7(f). This results in three different active sets: $\mathscr{A}_3 = \{1, 2\}$, $\mathscr{A}_4 = \{1, 6\}$ and $\mathscr{A}_5 = \{1, 7\}$. The sets $\mathscr{A}_3$ and $\mathscr{A}_4$ lead to new polyhedra as shown in the figures. The combination $\mathscr{A}_5$ leads to an interesting case of "degeneracy". The associated matrix $\tilde{G}$ has linearly dependent rows, which violates the LICQ assumption. In this case, $\mathscr{A}_5$ leads to an infeasible part of the state space.

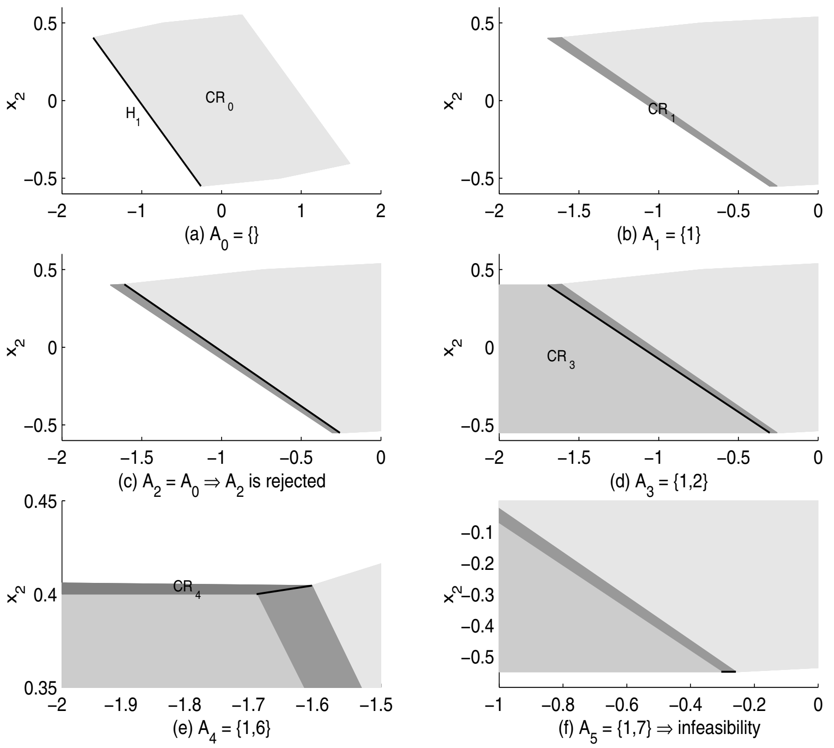

Fig. 1.7 Parameter space exploration strategy in [72].

The parameter space partition of the explicit solution to the mp-QP characterized by (1.70)–(1.73) has 14 polyhedral critical regions and it is given in Fig. 1.8.

1.2.4 Remarks on Alternative mp-QP Algorithms

This section has described in mp-QP algorithms presented in [13] and [72]. A combination of these algorithms, that uses the strategy of [72] to step over facets between neighboring regions, in combination with the QP solution of [13] in order to identify the optimal active set, is given by [4]. Like the algorithm in [72], it depends on the facet-to-facet property and the modifications described in [69] are useful. The mp-QP algorithm of [4] is the primary mp-QP algorithm of the widely used Multi-Parametric Toolbox (MPT), [46].

The combinatorial approach of [65] considers the combinations of potentially optimal active constraints. In many cases, this is not efficient since it tends to lead to many critical regions that are not full dimensional and must therefore be disregarded. An approach that exploits double representation (vertices and hyperplanes) of polyhedrons was given in [57]. It was shown in [34] that a more efficient

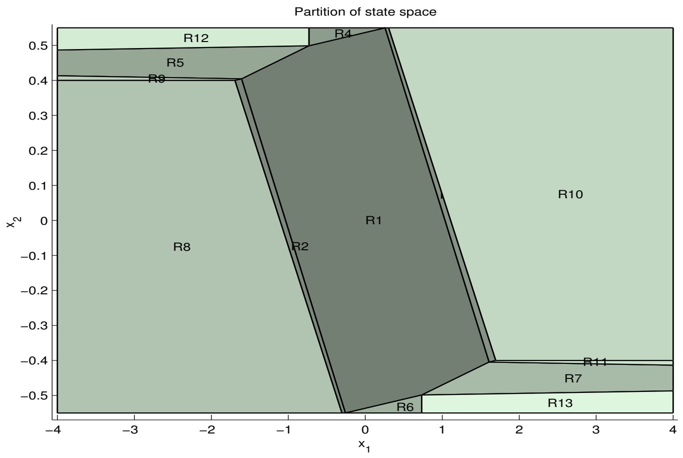

Fig. 1.8 Partition of the explicit solution to the mp-QP characterized by (1.70)–(1.73).

”non-geometric” combinatorial algorithm can be implemented by pruning infeasible candidate active sets.

Approximate and sub-optimal mp-QP algorithms have also been proposed, pursuing close approximations of lower complexity. The algorithm of [39] relies on an orthogonal partitioning of the parameter space that is built recursively to achieve an acceptable guaranteed maximum approximation error when approximating the solution with an affine function within each hyper-rectangle of the parameter space. A similar approach was taken in [9], using a simplex partition instead of hyper-rectangles, and in [64] that exploits the Delaunay tessellation. A reformulation of the MPC problem solving a sequence of simpler explicit MPC problems of horizon $N = 1$ and a nested sequence of terminal sets, a la dynamic programming, was proposed in [33]. Another sub-optimal approach based on short horizons was proposed in [40]. The use of nested invariant sets and interpolation techniques is pursued for approximations in [55]. An approximate solution for explicit MPC using set membership approximation has been introduced in [17]. While all the above approximations lead to PWA representations, [48] considers polynomial approximations.

For further review of existing algorithms, as well as multi-parametric linear programming (mp-LP) algorithms, we refer to [2].

1.3 Evaluating Piecewise Functions

Both mp-NLP and mp-QP algorithms provide the solution as some piecewise function representation, where the function pieces are defined a polyhedral partitioning into regions. All mp-QP algorithms, and most mp-NLP algorithms, return affine function pieces. Since the complexity of representation may be large even for relatively low order systems with constraints on some horizon (often thousands of regions), explicit MPC depend on efficient methods to evaluate piecewise functions.

The direct approach is to evaluate directly for each region if the current parameter (given by the current state, plus other auxiliary variables, perhaps) belongs to that region. This is clearly computationally demanding and may require more computations than solving the corresponding QP using an active-set solver, in particular if a good initialization is available for warm start. Still, it is very simple, and may even be very fast if implemented on a massively parallel computer architecture. The parallel implementation is entirely straightforward since all regions can be evaluated concurrently.

A much more computationally efficient approach was proposed in [71, 74], relying on a binary search tree representation of the polyhedral partitioning where at each level of the search tree one is able to exclude a significant fraction of the remaining candidate regions by evaluating on which side of a given hyperplane the current parameter belongs (typically a reduction of 1/3 is possible to achieve, as a rule of thumb). Hence, due to logarithmic complexity in the number of regions, the search among thousands of regions would amount of evaluating less than 20 hyperplanes. For mp-QP, the weakness of this approach is that extensive off-line computations are needed to construct a balanced binary search tree, and the piecewise function representation may still require extensive on-line computer memory. This is also true for methods that exploit optimal algorithms for selection of the hyperplanes for decisions [29]. To address this issue, the use of a truncated binary search tree in combination with direct search [6] or the lattice representation of piecewise linear functions [5] has been proposed. The realization of such piecewise affine function evaluation algorithms in dedicated hardware is investigated in [60, 41]. It should be mentioned that orthogonal partitions such as [38, 39] builds such a binary search tree as an integral part of the multi-parametric programming strategy.

Data structures other than binary search trees are also useful to support efficient evaluation. Bounding-boxes [19] and hash-tables [7] are proposed as supporting data structures to efficiently narrow down the search for the optimal polyhedral regions.

In MPC, the parameter (state) at one time instant is likely to be close to the parameter at the previous time instant, due to the continuity of trajectories of dynamic systems. Several algorithms have been proposed to build data structures that represent the topology of the polyhedral partitioning in order to quickly identify neighboring regions along the path from one parameter to the next parameter [68, 75].

Complexity of piecewise function representations can also be reduced by joining convex unions of polyhedrons, which share the same affine function piece (for the first control sample) [10]. In particular, the fact that input saturation will typically occur in a large number of regions can be directly exploited, [47]. Several methods for further compression of representation and efficient evaluation are investigated in [70].

References

1. Adams, L., Nazareth, J.L. (eds.): Linear and nonlinear conjugate gradient-related methods. Society for Industrial and Applied Mathematics, Philadelphia (1996)
2. Alessio, A., Bemporad, A.: A Survey on Explicit Model Predictive Control. In: Magni, L., Raimondo, D.M., Allgöwer, F. (eds.) Nonlinear Model Predictive Control: Towards New Challenging Applications. LNCIS, vol. 384, pp. 345–369. Springer, Heidelberg (2009)
3. Armijo, L.: Minimization of functions having Lipschitz-continuous first partial derivatives. Pacific J. Math. 16, 1–3 (1966)
4. Baotič, M.: An efficient algorithm for multi-parametric quadratic programming. Technical Report AUT02-05. Institut für Automatik, ETH Zürich (2002)
5. Bayat, F., Johansen, T.A., Jalali, A.A.: Flexible piecewise function evaluation methods with application to explicit model predictive control. In: Proceedings of the IEEE International Conference on Mechatronics, Istanbul (2011)
6. Bayat, F., Johansen, T.A., Jalali, A.A.: Combining truncated binary search tree and direct search for flexible piecewise function evaluation for explicit MPC in embedded microcontrollers. In: Proceedings of the IFAC World Congress, Milano (2011), `www.IFAC-PapersOnLine.net`
7. Bayat, F., Johansen, T.A., Jalali, A.A.: Using hash tables to manage time-storage complexity in point location problem: Application to explicit MPC. Automatica 47, 571–577 (2011)
8. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear programming: Theory and algorithms, 3rd edn. Wiley-Interscience, New Jersey (2006)
9. Bemporad, A., Filippi, C.: Suboptimal explicit MPC via approximate quadratic programming. In: Proceedings of the IEEE Conference on Decision and Control, Orlando, pp. 4851–4856 (2001)
10. Bemporad, A., Fukuda, K., Torrisi, F.D.: Convexity recognition of the union of polyhedra. Computational Geometry: Theory and Applications 18, 141–154 (2001)
11. Bemporad, A., Filippi, C.: An algorithm for approximate multiparametric convex programming. Computational Optimization and Applications 35, 87–108 (2006)
12. Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N.: The explicit solution of model predictive control via multiparametric quadratic programming. In: Proceedings of the American Control Conference, Chicago, Illinois, pp. 872–876 (2000)
13. Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N.: The explicit linear quadratic regulator for constrained systems. Automatica 38, 3–20 (2002)
14. Bentley, J.L.: Multidimensional binary search trees used for associative searching. Communications of the ACM 18, 509–517 (1975)
15. Box, M.J.: A new method of constrained optimization and a comparison with other methods. Computer J. 8, 42–52 (1965)
16. Boyd, S., Vandenberghe, L.: Convex optimization. University Press, Cambridge (2004)
17. Canale, M., Fagiano, L., Milanese, M.: Set membership approximation theory for fast implementation of model predictive control laws. Automatica 45, 45–54 (2009)
18. Cauchy, A.: Methode generale pour la resolution des systemes d'equations simultanees. Comp. Rend. Acad. Sci. Paris 536–538 (1847)
19. Christophersen, F., Kvasnica, M., Jones, C.N., Morari, M.: Efficient evaluation of piecewise control laws defined over a large number of polyhedra. In: Proceedings of the European Control Conference, pp. 2360–2367 (2007)
20. Deuflhard, P.: Newton methods for nonlinear problems. Springer, New York (2004)

21. Diehl, M., Bock, H.G., Schlöder, J.P.: Newton-type methods for the approximate solution of nonlinear programming problems in real-time. In: Di Pillo, G., Murli, A. (eds.) High Performance Algorithms and Software for Nonlinear Optimization, pp. 177–200. Kluwer Academic Publishers B.V. (2003)
22. Domínguez, L.F., Narciso, D.A., Pistikopoulos, E.N.: Recent advances in multiparametric nonlinear programming. Computers and Chemical Engineering 34, 707–716 (2010)
23. Domínguez, L.F., Pistikopoulos, E.N.: Quadratic approximation algorithm for multiparametric nonlinear programming problems. Technical report. Imperial College London (2009)
24. Dua, V., Pistikopoulos, E.N.: Algorithms for the solution of multi-parametric mixed-integer nonlinear optimization problems. Industrial & Engineering Chemistry Research 38, 3976–3987 (1999)
25. Dua, V., Pistikopoulos, E.N.: An algorithm for the solution of multiparametric mixed integer linear programming problems. Annals of Operations Research 99, 123–139 (2000)
26. Fiacco, A.V.: Introduction to sensitivity and stability analysis in nonlinear programming. Academic Press, Orlando (1983)
27. Finkel, D.E.: Global optimization with the DIRECT algorithm. Ph.D. thesis. North Carolina State University (2005)
28. Fletcher, R., Reeves, C.M.: Function minimization by conjugate gradients. Comput. J. 7, 149–154 (1964)
29. Fuchs, A.N., Jones, C.N., Morari, M.: Optimized decision trees for point location in polytopic data sets - Application to explicit MPC. In: Proceedings of the American Control Conference, Baltimore, pp. 5507–5512 (2010)
30. Gablonsky, J.M.: Modifications of the DIRECT algorithm. Ph.D. thesis. North Carolina State University (2001)
31. Grancharova, A., Johansen, T.A.: Approximate explicit model predictive control incorporating heuristics. In: Proceedings of IEEE International Symposium on Computer Aided Control System Design, Glasgow, Scotland, U.K., pp. 92–97 (2002)
32. Grancharova, A., Johansen, T.A., Tøndel, P.: Computational aspects of approximate explicit nonlinear model predictive control. In: Findeisen, R., Allgöwer, F., Biegler, L. (eds.) Assessment and Future Directions of Nonlinear Model Predictive Control. LNCIS, vol. 358, pp. 181–192. Springer, Heidelberg (2007)
33. Grieder, P., Morari, M.: Complexity reduction of receding horizon control. In: Proceedings of the 42th IEEE Conference on Decision and Control, Maui, Hawaii, USA, pp. 3179–3184 (2003)
34. Gupta, A., Bhartiua, S., Nataraj, P.S.V.: A novel approach to multiparametric quadratic programming. Automatica 47, 2112–2117 (2011)
35. Hooke, R., Jeeves, T.A.: Direct search solution of numerical and statistical problems. J. Assoc. Comput. Mach. 8, 212–229 (1961)
36. Horst, R., Tuy, H.: Global optimization. Springer, Berlin (1995)
37. Johansen, T.A.: On multi-parametric nonlinear programming and explicit nonlinear model predictive control. In: Proceedings of the IEEE Conference on Decision and Control, Las Vegas, NV, vol. 3, pp. 2768–2773 (2002)
38. Johansen, T.A.: Approximate explicit receding horizon control of constrained nonlinear systems. Automatica 40, 293–300 (2004)
39. Johansen, T.A., Grancharova, A.: Approximate explicit constrained linear model predictive control via orthogonal search tree. IEEE Transactions on Automatic Control 48, 810–815 (2003)
40. Johansen, T.A., Petersen, I., Slupphaug, O.: Explicit sub-optimal linear quadratic regulation with state and input constraints. Automatica 38, 1099–1111 (2002)

41. Johansen, T.A., Jackson, W., Schreiber, R., Tøndel, P.: Hardware synthesis of explicit model predictive controllers. IEEE Transactions Control Systems Technology 15, 191–197 (2007)
42. Jones, D.R.: The DIRECT global optimization algorithm. In: Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of Optimization, vol. 1, pp. 431–440. Kluwer, Dordrecht (2001)
43. Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the Lipschitz constant. Journal of Optimization Theory and Applications 79, 157–181 (1993)
44. Kelley, C.T.: Iterative methods for optimization. Society for Industrial and Applied Mathematics, Philadelphia (1999)
45. Kojima, M.: Strongly stable stationary solutions in nonlinear programs. In: Robinson, S.M. (ed.) Analysis and Computation of Fixed Points, pp. 93–138. Academic Press, New York (1980)
46. Kvasnica, M., Grieder, P., Baotić, M., Morari, M.: Multi-Parametric Toolbox (MPT). In: Alur, R., Pappas, G.J. (eds.) Hybrid Systems: Computation and Control, HSCC 2004. LNCS, vol. 2993, pp. 448–462. Springer, Heidelberg (2004)
47. Kvasnica, M., Fikar, M.: Performance-lossless complexity reduction in explicit MPC. In: Proceedings of the IEEE Conference on Decision and Control, pp. 5270–5275 (2010)
48. Kvasnica, M., Löfberg, J., Fikar, M.: Stabilizing polynomial approximation of explicit MPC. Automatica 47, 2292–2297 (2011)
49. Lau, M.S.K., Yue, S.P., Ling, K.V., Maciejowski, J.M.: A comparison of interior point and active set methods for FPGA implementation of model predictive control. In: Proceedings of the European Control Conference 2009, Budapest, Hungary, pp. 156–161 (2009)
50. Levitin, E.S.: Perturbation theory in mathematical programming. Wiley (1994)
51. Lewis, R.M., Torczon, V., Trosset, M.W.: Direct search methods: then and now. Journal of Computational and Applied Mathematics 124, 191–207 (2000)
52. Mangasarian, O.L., Rosen, J.B.: Inequalities for stochastic nonlinear programming problems. Operations Research 12, 143–154 (1964)
53. Narciso, D.: Developments in nonlinear multiparametric programming and control. PhD thesis, London, U.K. (2009)
54. Nelder, J.A., Mead, R.: A simplex method for function minimization. Computer J. 7, 308–313 (1965)
55. Nguyen, H.N., Gutman, P.O., Olaru, S., Hovd, M.: Explicit constrained control based on interpolation techniques for time-varying and uncertain linear discrete-time systems. In: Proceedings of the IFAC World Congress, Milano (2011), `www.IFAC-PapersOnLine.net`
56. Nocedal, J., Wright, S.J.: Numerical optimization. Springer, New York (1999)
57. Olaru, S., Dumur, D.: A parameterized polyhedra approach for explicit contrained predictive control. In: Proceedings of the IEEE Conference on Decision and Control, Bahamas, pp. 1580–1585 (2004)
58. Pistikopoulos, E.N., Georgiadis, M.C., Dua, V.: Multi-parametric programming: Theory, algorithms, and applications. Wiley-VCH (2007)
59. Piyawksii, S.A.: An algorithm for finding the absolute extremum of a function. USSR Computational Mathematics and Mathematical Physics 12, 57–67 (1972)
60. Poggi, T., Comaschi, F., Storace, M.: Digital circuit realization of piecewise affine functions with non-uniform resolution: Theory and FPGA implementation. IEEE Trans. Circuits and Systems - II: Express Briefs 57, 131–135 (2010)
61. Powell, M.J.D.: A fast algorithm for nonlinearly constrained optimization calculations. In: Watson, G.A. (ed.) Numerical Analysis, Dundee 1977. Lecture Notes in Mathematics, vol. 630. Springer, Berlin (1978)

62. Ralph, D., Dempe, S.: Directional derivatives of the solution of a parametric nonlinear program. Mathematical Programming 70, 159–172 (1995)
63. Rockafellar, R.T.: Convex analysis. Princeton University Press, New Jersey (1970)
64. Scibilia, F., Olaru, S., Hovd, M.: Approximate explicit linear MPC via Delaunay tessellation. In: Proceedings of the European Control Conference, Budapest, Hungary (2009)
65. Seron, M.M., Goodwin, G.C., De Doná, J.A.: Characterization of receding horizon control for constrained linear systems. Asian J. Control 5, 271–286 (2003)
66. Spendley, W., Hext, G.R., Himsworth, F.R.: Sequential applications of simplex designs in optimisation and evolutionary operation. Technometrics 4, 441 (1962)
67. Spjøtvold, J., Kerrigan, E.C., Jones, C.N., Tøndel, P., Johansen, T.A.: On the facet-to-facet property of solutions to convex parametric quadratic programs. Automatica 42, 2209–2214 (2006)
68. Spjøtvold, J., Rakovic, S.V., Tøndel, P., Johansen, T.A.: Utilizing reachability analysis in point location problems. In: Proceedings of the IEEE Conference on Decision and Control, San Diego (2006)
69. Spjøtvold, J., Tøndel, P., Johansen, T.A.: Continuous selection and unique polyhedral representation of solutions to convex parametric quadratic programs. Journal of Optimization Theory and Applications 134, 177–189 (2007)
70. Szücs, A., Kvasnica, M., Fikar, M.: A memory-efficient representation of explicit MPC solutions. In: Proceedings of the IEEE Conference on Decision and Control, Orlando (2011)
71. Tøndel, P., Johansen, T.A.: Complexity reduiction in explicit model predictive control. In: Proceedings of the IFAC World Congress, Barcelona (2002), `www.IFAC-PapersOnLine.net`
72. Tøndel, P., Johansen, T.A., Bemporad, A.: An algorithm for multi-parametric quadratic programming and explicit MPC solutions. Automatica 39, 489–497 (2003)
73. Tøndel, P., Johansen, T.A., Bemporad, A.: Further results on multi-parametric quadratic programming. In: Proceedings of the IEEE Conference on Decision and Control, Maui (2003)
74. Tøndel, P., Johansen, T.A., Bemporad, A.: Evaluation of piecewise affine control via binary search tree. Automatica 39, 743–749 (2003)
75. Wang, Y., Jones, C.N., Maciejowski, J.: Efficient point location via subdivision walking with application to explicit MPC. In: Proceedings of the European Control Conference, pp. 447–453 (2007)
76. Wright, S.J.: Primal-dual interior-point methods. SIAM Publications, Philadelphia (1997)