RESEARCH PAPERS IN MANAGEMENT STUDIES

COMBINATORIAL STRUCTURES IN NONLINEAR PROGRAMMING

Stefan Scholtes

WP 6/2002

The Judge Institute of Management Studies
Trumpington Street
Cambridge CB2 1AG

COMBINATORIAL STRUCTURES IN NONLINEAR PROGRAMMING

Stefan Scholtes

WP 6/2002

Dr Stefan Scholtes
The Judge Institute of
Management
Department of Engineering
University of Cambridge
Trumpington Street
Cambridge
CB2 1AG
UK
Tel.: +44 1223 339635
Email: s.scholtes@jims.cam.ac.uk

Please address enquiries about the series to:

Publications Secretary
Judge Institute of Management
Trumpington Street
CAMBRIDGE
CB2 1AG
Tel: 01223 339636 Fax: 01223 339701
E-mail: s.lee@jims.cam.ac.uk

Combinatorial Structures in Nonlinear Programming

Stefan Scholtes*

April 2002

Abstract

Non-smoothness and non-convexity in optimization problems often arise because a combinatorial structure is imposed on smooth or convex data. The combinatorial aspect can be explicit, e.g. through the use of "max", "min", or "if" statements in a model, or implicit as in the case of bilevel optimization where the combinatorial structure arises from the possible choices of active constraints in the lower level problem. In analyzing such problems, it is desirable to decouple the combinatorial from the nonlinear aspect and deal with them separately. This paper suggests a problem formulation which explicitly decouples the two aspects. We show that such combinatorial nonlinear programs, despite their inherent non-convexity, allow for a convex first order local optimality condition which is generic and tight. The stationarity condition can be phrased in terms of Lagrange multipliers which allows an extension of the popular sequential quadratic programming (SQP) approach to solve these problems. We show that the favorable local convergence properties of SQP are retained in this setting. The computational effectiveness of the method depends on our ability to solve the subproblems efficiently which, in turn, depends on the representation of the governing combinatorial structure. We illustrate the potential of the approach by applying it to optimization problems with max-min constraints which arise, for example, in robust optimization.

1 Introduction

Nonlinear programming is nowadays regarded as a mature field. A combination of important algorithmic developments and increased computing power over the past decades have advanced the field to a stage where the majority of practical problems can be solved efficiently by commercial software. However, this is not to say that there are no research challenges left. One considerable challenge is posed by global, i.e. non-convex, optimization problems. There are two main sources of non-convexity: An analytic and a combinatorial source. Analytic non-convexity is due to a combination of negative and positive curvature of underlying model functions,

*Judge Institute of Management & Department of Engineering, University of Cambridge, Cambridge CB2 1AG, s.scholtes@jims.cam.ac.uk

such as polynomials or trigonometric functions. In contrast, combinatorial non-convexity arises when a combinatorial selection rule is imposed on possibly quite simple, even linear, functions. Typical examples are function evaluations that make use of "max", "min", "abs", or "if" statements. Business spreadsheets are a veritable source of such combinatorial nonlinear functions. Consider for example the calculation of revenue as a function of price and production capacity. Revenue is the product of price and sales, where sales is the minimum of demand and production capacity. If capacity κ and price π are decision variables and demand d is a known function of price, then the revenue formula becomes $r(\pi, \kappa) = \pi \min\{d(\pi), \kappa\}$. Such formulas are commonplace in spreadsheet models.

A more complicated example where nested max- and min-terms occur is the resource allocation problem that arises when several projects are evaluated by decision trees and compete for limited resources. This is a timely problem area as decision tree analysis is experiencing a renaissance in the wake of the popularization of real options valuations of projects. Decision tree evaluations consist essentially of a finite sequence of nested linear combinations of max-terms, where the linear combinations correspond to taking expectations at event nodes and the max-terms correspond to choosing the best of a finite number of possible actions at decision nodes. The value of a project is likely to depend on design parameters such as marketing budgets or other limited resources and a company is faced with the problem of allocating such limited resources to a variety of projects. This resource allocation problem will involve parametric values of decision trees, i.e., sequences of max-terms. The problem becomes even more complex if, as is often the case in practice, probability estimates are subjective and several experts assign differing probability estimates to the event branches in the trees. Backward induction through the decision trees will lead to value estimates for each of these probability estimates and a conservative decision maker may well consider the minimum of the expected values as a proxy for the unknown value of the project at event nodes. With this addition, the constraints of the resource allocation problem involve sequences of max- as well as min-terms. These max- and min-terms define the combinatorial structure of the problem.

Many practical applications are a combination of both, the analytical as well as the combinatorial sources of non-convexity. A non-convex combinatorial structure may well be imposed onto smooth but already non-convex functions. Nevertheless, it is helpful to think of the two sources separately, not least because the combinatorial source introduces the additional complication of non-smoothness. In fact, for optimization problems which involve only the first, the analytic source of non-convexity, nonlinear programming provides us with a host of methods, such as sequential quadratic programming (SQP) or interior point methods. Engineers and operations researchers regularly apply these methods, albeit they cannot guarantee a global optimum, because the methods provide often substantial improvement and, at the same time, are applicable to model sizes that are orders of magnitude larger than those that can be handled by global optimization routines. It is indeed unlikely that the dominance of local optimization procedures will change in the foreseeable future. In this light, it is rather surprising that relatively little attention has been

given to the local optimization of non-convex problems arising from the second, the combinatorial source of non-convexity. This is the more surprising as some fairly general classes of such problems, such as disjunctive programs, have been studied extensively from a global optimization viewpoint. The present study attempts to fill this gap by extending the tools available for the local optimization of smooth non-convex problems to problems involving non-convex combinatorial structures.

To date, problems of the type considered here are often tackled by variants of Lemaréchal's bundle method [17], such as the bundle trust region code of Schramm and Zowe [26]. These methods were originally designed to solve non-smooth *convex* problems but they are well-defined for more general non-smooth problems if the subgradient concept is suitably broadened, see e.g. [5]. However, bundle methods have rather poor convergence properties for non-convex problems. This is not surprising since they are based on convex models, such as maxima of finitely many affine functions, of originally non-convex functions and can therefore only be guaranteed to converge to points where these convex models provide no descent directions. These methods do not exploit the specific combinatorial structure of the problem at hand, e.g. a particular combination of "max" and "min" terms, but rather replace this structure by a simpler and essentially convex combinatorial structure. An alternative approach, more akin to our own and more suitable for the exploitation of non-convex structures, has been developed for piecewise linear problems under the label of simplex-type or active-set type methods by Fourer [10, 11, 12] and others. For recent further developments along these lines we refer the reader to the comprehensive paper of Conn and Mongeau [6] which also surveys some of these methods and provides a fairly extensive bibliography. The existing combinatorial methods, however, are either restricted to piecewise affine models or to models where the combinatorial structure is inherently convex, such as the minimization of point-wise maxima of nonlinear functions. We attempt to add to this literature by presenting and analyzing a modelling framework which explicitly allows for non-convex structures as well as smooth nonlinear data.

The focus of this paper is the extension of the traditional active set and SQP approaches to combinatorial nonlinear programs. We will provide a theoretical framework for the local optimization of such problems, including appropriate stationarity conditions that are tighter than the stationarity conditions of [5] which bundle methods rely on. We then show how the SQP-idea can be extended to combinatorial nonlinear programs and that these methods have better convergence properties for these problems than bundle-type methods, both in terms of speed and in terms of identifying more appropriate stationary points.

We have already mentioned that one motivation for this study is to allow modelers to use simple nonsmooth functions such as "max", "min", "$\|.\|$", etc. as building blocks in their optimization models. There are, however, also interesting classes of optimization problems to which our approach applies directly and we will mention some of them in the next section before we proceed to the technical developments. One of these classes are optimization problems with max-min constraints which

arise in robust optimization. We will use this special case frequently to illustrate our general theoretical developments. Section 3 introduces necessary extensions of standard NLP terminology, including notions of active constraint functions, regularity and stationarity for combinatorial nonlinear programs. Section 4 explains how Lagrange multipliers can be defined and used to check stationarity. In Section 5 we discuss the use of the SQP method in the setting of combinatorial nonlinear programs. We then focus in section 6 on the subproblems that arise in the case of max-min constrained problems and show how a decomposition approach can be used to solve these problems to local optimality. We close the paper with a brief report on some preliminary numerical experiments.

2 Problem Statement and Sample Problems

We study *combinatorial nonlinear programs* of the form

$$\begin{array}{ll} \min & f(x) \\ s.t. & g(x) \in Z, \end{array} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$, and $Z \subseteq \mathbb{R}^m$. We regard the functions f, g as the data specifying an instance of the optimization problem, whilst the *structure set* Z represents the combinatorial structure of a problem class. We will assume throughout that f and g are smooth and that Z is closed. Problem formulations of the above kind are commonplace in semi-definite programming and related areas, where Z is assumed to be a convex, possibly non-polyhedral cone. In contrast to these studies, we will not assume that Z but only that it has some exploitable combinatorial characteristics. A typical set Z is a polyhedral complex, for example the boundary of a polyhedron or the set of roots of a piecewise affine function. In the latter case one may for example wish to exploit a max-min representation of the piecewise affine function, see e.g. [1]. Notice that problem (1), although of a combinatorial nature, is quite different from integer programming and does not necessarily have a discrete feasible set; whence local optimization is non-trivial.

2.1 Sample Problems

The problem formulation (1) is quite flexible and encompasses several classes of well-known optimization problems. It obviously encompasses as a special case the standard nonlinear program

$$\begin{array}{ll} \min & f(x) \\ s.t. & g(x) \leq 0 \\ & h(x) = 0, \end{array}$$

the constraints of which can be rewritten as

$$(g(x), h(x)) \in Z = \mathbb{R}^p_- \times \{0\}^q.$$

Notice that this set Z is the set of roots of the piecewise linear convex penalty-type function

$$p(u, v) = \max\{u_1, \ldots, u_p, |v_1|, \ldots, |v_q|\}.$$

Because the structure set Z is convex, the nonlinear programming problem can, in some sense, be thought of as a structurally convex problem. In particular, one obtains a convex local approximation of the problem if the data functions are replaced by their first order Taylor series. In other words, non-convexity is only introduced through nonlinearity in the data. As mentioned before, such structurally convex problems are not the focus of our study.

A first example of a structurally non-convex problem is the so-called mathematical program with complementarity constraints

$$\begin{array}{ll} \min & f(x) \\ s.t. & g(x) \leq 0 \\ & h(x) = 0 \\ & \min\{G(x), H(x)\} = 0, \end{array}$$

see e.g. [18, 23] and the references therein. The constraints are equivalent to

$$(g(x), h(x), (G_1(x), H_1(x)), \ldots, (G_r(x), H_r(x))) \in Z = \mathbb{R}^p_- \times \{0\}^q \times \mathrm{L}^r,$$

where L is the boundary of the positive orthant in $\mathbb{R}^2$. This structure set Z is non-convex and therefore the problem itself should be thought of as structurally non-convex. The set Z can again by represented as the set of roots of a piecewise linear function, e.g.

$$p(t, u, v, w) = \max\{t_1, \ldots, t_p, |u_1|, \ldots, |u_q|, |\min\{v_1, w_1\}|, \ldots, |\min\{v_r, w_r\}|\}.$$

As a final example, we mention robust variants of stochastic programs, which lead directly to optimization problems with max-min functions. A standard stochastic program with recourse over finitely many scenarios is of the form

$$\min_{x \in X} \sum_{1 \leq j \leq m} p_j \min_{y_j \in Y_j} f(x, y_j).$$

The interpretation is that a decision x needs to be taken now before a scenario j is observed with probability p_j. After the observation of the scenario a recourse action $y_j \in Y_j$ can be taken. The decision criterion is to choose x now so that the expected value of f is maximal. If it is important to ensure that the objective is small for all scenarios then the robust counterpart of this stochastic program is more appropriate. This optimization problem is of the form

$$\min_{x \in X} \max_{1 \leq j \leq m} \min_{y_j \in Y_j} f(x, y_j).$$

If there are only finitely many candidates for recourse actions in each scenario then the robust optimization problem turns into a finite max-min optimization problem

$$\min_{x \in X} \max_{1 \leq j \leq m} \min_{i \in N_j} f_i(x),$$

where $N_j \subseteq \{1, \ldots, n\}$. Such problems have been recently investigated e.g. in [13, 19] with engineering design applications in mind.

The latter optimization problem can be re-written as a constrained problem of the form

$$\begin{array}{ll} \min & \alpha \\ s.t. & \max\limits_{1\le j\le r}\min\limits_{i\in N_j} f_i(x)-\alpha\le 0. \end{array} \tag{2}$$

The constraints are obviously equivalent to

$$(f_1(x)-\alpha,\ldots,f_m(x)-\alpha)\in Z,$$

where Z is the lower zero-level set of a piecewise linear function, e.g.,

$$Z=\{z\mid p(z)\le 0\},\quad p(z)=\max_{1\le j\le r}\min_{i\in N_j} z_i.$$

The set Z is a nonconvex union of convex polyhedral cones and therefore the problem itself is structurally non-convex. More generally, max-min optimization problems are problems of the form

$$\begin{array}{ll} \min & f(x) \\ s.t. & g(x)\in Z, \end{array} \tag{3}$$

with the set Z above. We will use such max-min optimization problems to illustrate the more theoretical developments in the sequel.

3 Some Terminology

The SQP method for standard nonlinear programs is essentially an application of Newton's method to the stationarity conditions. The notions of *active constraints* and of *regularity* play an important role in the development of the stationarity conditions as well as the analysis of the SQP method. We therefore need to extend the notions of *active constraints*, *regularity*, and *stationarity* to combinatorial nonlinear programs before we can discuss a modification of SQP for such problems.

3.1 Active Constraint Functions

We regard a constraint function g_i as *inactive* at a feasible point $\bar{x}$ of problem (1) if the validity of the inclusion $g(x)\in Z$ does not depend on the value $g_i(x)$ for x close to $\bar{x}$. This intuitive notion is made precise in the following definition.

Definition 3.1 *A component z_i is called inactive for a set Z at a point $\bar{z}\in Z$ if there exists a neighborhood V of $\bar{z}$ such that for any two points $z',z''\in V$ which differ only in the i-th component (i.e. $z'_j=z''_j$ for every $j\neq i$) the inclusion $z'\in Z$ holds if and only if $z''\in Z$. All other components are called active at $\bar{z}$. A constraint function g_i is called active (inactive) at a feasible point $\bar{x}$ of problem (1) if z_i is active (inactive) for Z at $\bar{z}=g(\bar{x})$.*

Notice that activity of a constraint function depends only on the value of the constraint function at the point $\bar{x}$, not on the behavior of the function close to $\bar{x}$. The following lemma is an immediate consequence of the above definition and provides an equivalent definition of activity.

Lemma 3.1 *The component z_i is active for Z at $\bar{z} \in Z$ if and only if there exist sequences $z^k \to \bar{z}$ and $\alpha_k \to 0$ such that $z^k \in Z$ and $z^k + \alpha_k e_i \notin Z$, where e_i is the i-th unit vector.*

Inactivity of a component z_i implies that $\bar{z} + \alpha e_i \in Z$ for every sufficiently small α but the reverse implication does not hold in general. For instance, if Z is the union of the axes in $\mathbb{R}^2$ then both components satisfy the latter condition at the origin but both components are active at the origin. Another example is the set in $\mathbb{R}^2$ defined by the inequality $|z_1| \leq |z_2|$. Although $\alpha e_2 \in Z$ for every α, the second component z_2 is active at the origin.

If the components of z are rearranged so that $z = (z_{\mathcal{I}}, z_{\mathcal{A}})$ with $z_{\mathcal{I}}$ and $z_{\mathcal{A}}$ corresponding to inactive and active components, resp., at $\bar{z}$ then for every z close to $\bar{z}$

$$z \in Z \text{ if and only if } z_{\mathcal{A}} \in Z_{\mathcal{A}} = \{\zeta_{\mathcal{A}} \mid (\zeta_{\mathcal{A}}, \zeta_{\mathcal{I}}) \in Z\}. \tag{4}$$

In other words, locally around $\bar{z}$ the set Z coincides with the $(\{\bar{z}_{\mathcal{I}}\} + \mathbb{R}^{|\mathcal{I}|}) \times Z_{\mathcal{A}}$.

A natural question to ask is whether one can obtain a characterization of the active components for the union or intersection of sets from knowledge of active components of the individual sets. It is not difficult to see that an active component z_i for either $\cup_{j=0}^m Z_j$ or $\cap_{j=0}^m Z_j$ at a point $\bar{z} \in \cap_{j=0}^m Z_j$ is active for at least one of the sets Z_j at $\bar{z}$. The reverse statement, however, does not hold in general in either case. If Z_1 is the set of nonnegative reals and Z_2 is the set of nonpositive reals, then the single variable is active for both sets at the origin but inactive for the union of the two sets. Also, if Z_1 is the set of all (z_1, z_2) with $z_2 \leq 0$ and Z_2 is the union of Z_1 and the nonnegative z_2-axis then z_1 and z_2 are both active for Z_2 at the origin but only z_2 is active for $Z_1 \cap Z_2$ at the origin.

The following proposition gives an algorithmic characterization of the active components of the set Z arising in (3).

Proposition 3.1 *Let $Z = \{z \in \mathbb{R}^m \mid p(z) \leq 0\}$ with $p(z) = \max\limits_{1 \leq j \leq r} \min\limits_{i \in N_j} z_i$ and let $\bar{z} \in Z$. If $p(\bar{z}) < 0$ then all components are inactive at $\bar{z}$. If $p(\bar{z}) = 0$ then the active components can be determined in the following way:*

1. *Determine $J = \{j \mid \min\limits_{i \in N_j} \bar{z}_i = 0\}$*
2. *Determine $\bar{N}_j = \{i \in N_j \mid \bar{z}_i = 0\}$ for every $j \in J$.*
3. *Delete from J all indices k for which there exists $j \in J$ such that $\bar{N}_j$ is a proper subset of $\bar{N}_k$ and call the remaining index set $\bar{J}$.*

A component z_i is active at $\bar{z}$ if and only $i \in \cup_{j \in \bar{J}} \bar{N}_j$.

Proof. The first statement follows directly from the continuity of $p(.)$. To prove the second statement we assume that $p(\bar{z}) = 0$. Since $\bar{z} \in Z$ we have $\min_{i \in N_j} \bar{z}_i \leq 0$ for every $1 \leq j \leq r$. If $\min_{i \in N_j} \bar{z}_i < p(\bar{z})$ then this inequality remains valid in a

neighborhood of $\bar{z}$ and therefore p is unchanged if the term $\min_{i\in N_j} \bar{z}_i$ is eliminated. Hence

$$p(z) = \max_{j\in J} \min_{i\in N_j} z_i$$

for every z in a neighborhood of $\bar{z}$. Also, if $i \in N_j$ and $\bar{z}_i > \min_{k\in N_j} \bar{z}_k = 0$ then one can remove i from N_j without changing $\min_{k\in N_j} z_k$ in a neighborhood of $\bar{z}$. After these eliminations we obtain

$$p(z) = \max_{j\in J} \min_{i\in \bar{N}_j} z_i$$

for all z in a neighborhood of $\bar{z}$. Finally, one can eliminate all min-terms corresponding to an index set $\bar{N}_k$ such that there exists a proper subset $\bar{N}_j$ of $\bar{N}_k$ for some $j \in J$ since

$$\min_{i\in \bar{N}_j} z_i \geq \min_{i\in \bar{N}_k} z_i$$

for every z. Thus

$$p(z) = \max_{j\in \bar{J}} \min_{i\in \bar{N}_j} z_i$$

for every z close to $\bar{z}$. Because of this representation of p in a neighborhood of $\bar{z}$ it is obvious that $i \in \cup_{j\in\bar{J}} \bar{N}_j$ for every active component z_i at $\bar{z}$. To see the converse, let $i_0 \in \bar{J}$, $j_0 \in \bar{N}_j$ for some $j \in \bar{J}$, and define

$$z_i^k(\alpha) = \begin{cases} \frac{1}{k} & \text{if } i \in \bar{N}_{j_0}\backslash\{i_0\} \\ \alpha & \text{if } i = i_0 \\ -\frac{1}{k} & \text{otherwise.} \end{cases}$$

Notice that for every $-\frac{1}{k} \leq \alpha \leq \frac{1}{k}$

$$\begin{aligned} \min_{i\in \bar{N}_{j_0}} z_i^k(\alpha) &= \alpha \\ \min_{i\in \bar{N}_j} z_i^k(\alpha) &= -\frac{1}{k}, \quad \forall j \neq j_0. \end{aligned}$$

Therefore $p(z^k(\alpha)) = \alpha$ for every $0 \leq \alpha \leq \frac{1}{k}$. Since $z^k(\alpha) = z^k(0) + \alpha e_i$ it follows that z_{i_0} is active at $\bar{z}$. Q.E.D.

3.2 Regularity and Stationarity

We call the constraint functions *regular* at $\bar{x}$ if the gradients of all active constraint functions at $\bar{x}$ are linearly independent and call $\bar{x}$ a *stationary point* of (1) if $d = 0$ is a local minimizer of

$$\begin{aligned} \min_{d} \quad & \nabla f(\bar{x})^\top d \\ s.t. \quad & g(\bar{x}) + \nabla g(\bar{x}) d \in Z. \end{aligned} \tag{5}$$

Since activity of a constraint function only depends on the value of the constraint function at a given point, the function $g_i(\bar{x}) + \nabla g_i(\bar{x})d$ is active at $d = 0$ in (5) if and only if g_i is active at $\bar{x}$ in (1). Notice that local and global minimization for (5) are equivalent if Z is convex but not necessarily otherwise. The reader may ask why we

do not further simplify the auxiliary problem (5) by replacing Z in the constraints of (5) by an appropriate first order approximation at the point $g(\bar{x})$, e.g. by its Bouligand cone, see [3]. This would indeed simplify the development of necessary optimality conditions. However, the importance of first order optimality conditions is not that they indicate optimality - after all in the presence of nonlinearity it is unlikely that a finite computational method will produce a stationary point - but that they allow the computation of descent paths if the conditions are not satisfied. The convergence of methods with descent directions obtained from tangent cone based optimality conditions, however, often suffers from the discontinuity of the tangent cone as a function of the base point. For instance if Z is the boundary of the positive orthant in $\mathbb{R}^2$ then the only sensible tangent cone at a point on the positive x_1-axis is the x_1-axis itself. Therefore the tangent cone based subproblem that computes a descent direction is, arbitrarily close to the origin, blind for the possibility to move along the x_2-axis at the origin. Consequently, the method may converge to the origin along the x_1-axis even though the objective function allows a first order descent along the positive x_2-axis at the origin.

The first question we need to address is whether stationarity as defined above is a necessary optimality condition under appropriate assumptions. The simple example

$$\begin{array}{ll} \min & 2x_1^2 - x_2 \\ s.t. & (x_1, x_2) \in Z \end{array} \tag{6}$$

where $Z = \{(z_1, z_2) \mid z_1^2 - z_2 = 0\}$ shows that even for regular constraint functions one cannot expect a local minimizer, the origin in this case, to be stationary. We will have to make an additional assumption for this to be the case.

Definition 3.2 *A set Z is said to be locally star-shaped at a feasible point $\bar{z}$ if there exists a neighborhood V of $\bar{z}$ such that $z \in Z \cap V$ implies $\alpha z + (1-\alpha)\bar{z} \in Z \cap V$ for every $\alpha \in [0,1]$.*

In particular, convex sets are locally star-shaped. However, the set of locally star-shaped sets is much broader. Indeed, one readily verifies that the intersection *and* the union of finitely many closed locally star-shaped sets remains locally star-shaped. An important special class of locally star-shaped sets are the roots or lower level sets of piecewise affine functions which are non-convex polyhedra. Notice that the set Z employed in example (6) is not locally star-shaped anywhere. Indeed, if the structure set Z is locally star-shaped at a regular local minimizer then this local minimizer is stationary.

Proposition 3.2 *If Z is locally star-shaped at $g(x)$ and x is a regular local minimizer of (1) then x is a stationary point.*

Proof. To simplify notation we assume that all components of g are active at x^* and that $x^* = 0, g(x^*) = 0$. This can be achieved by replacing Z by the set $Z_{\mathcal{A}}$ of (4) and replacing $g(.)$ by $g(x^* + .) - g(x^*)$. By assumption, the Jacobian $\nabla g(0)$ has full row rank and thus there is an $(n-m) \times (n-m)$-matrix A such that the

$n \times n$-matrix $\begin{pmatrix} \nabla g(0) \\ A \end{pmatrix}$ is invertible. The inverse function theorem thus implies that the equation

$$\begin{aligned} g(x) &= \nabla g(0)d \\ Ax &= Ad \end{aligned}$$

has locally around the origin a unique smooth solution function $x = \xi(d)$ with $\nabla\xi(0) = \mathrm{Id}$, the $n \times n$ identity matrix. Now suppose the origin is not a stationary point. Then there exists d arbitrarily close to the origin such that $\nabla g(0)d \in Z$ and $\nabla f(0)d < 0$. Choose d close enough to zero such that $\xi(\alpha d)$ is well defined for every $\alpha \in [0,1]$ (regularity assumption) and that $\nabla g(0)\alpha d \in Z$ for every $\alpha \in [0,1]$ (local star-shapedness). With $\phi(\alpha) = f(\xi(\alpha d))$ one obtains

$$\phi'(0) = \nabla f(0)\nabla\xi(0)d = \nabla f(0)d < 0.$$

Hence $f(\xi(\alpha d)) < f(\xi(0)) = f(0)$ for all sufficiently small $\alpha > 0$ which contradicts the local optimality of $x = 0$. Q.E.D.

4 Lagrange Multipliers

The Lagrange multipliers provide the connection between Newton's method and SQP. In this section we extend the notion of Lagrange multiplier to combinatorial nonlinear programs and provide an equivalent stationarity condition in terms of these multipliers.

Definition 4.1 *The Lagrangian function associated with the data (f, g) of problem (1) is of the form*

$$L(x, \lambda) = f(x) + g(x)^\top \lambda.$$

A feasible point x is called a critical point if there exist multipliers λ such that

$$\begin{aligned} \nabla_x L(x,\lambda) := \nabla f(x) + \nabla g(x)^\top \lambda &= 0 \\ \lambda_i &= 0, \quad \textit{if } g_i \textit{ is inactive at } x. \end{aligned} \tag{7}$$

Notice that the multipliers associated with a regular critical point are unique.

Proposition 4.1 *Every stationary point of (1) is critical.*

Proof. If $d = 0$ is a local solution of (5) then it is also a solution of the linear program

$$\begin{aligned} \min_d \quad & \nabla f(\bar{x})^\top d \\ s.t. \quad & g_i(\bar{x}) + \nabla g_i(\bar{x})d = \bar{z}_i, \quad \forall i : g_i \text{ is active at } \bar{x}, \end{aligned}$$

where $\bar{z} = g(\bar{x})$. Hence linear programming duality provides the existence of the multipliers (7). Q.E.D.

Without stringent assumptions on the set Z criticality is not sufficient for stationarity at a regular feasible point of (1). Indeed, in the case of a standard nonlinear

program one needs to impose sign constraints on the multipliers of inequalities in order to completely characterize stationarity. So what is the analogue of the sign constraints for general structure sets Z? The following proposition answers this question. The key observation is that stationarity of a regular critical point is independent of the behavior of the data functions in the vicinity of this point but depends only on the value of g at the critical point and on the Lagrange multiplier λ.

Proposition 4.2 *If $\bar{x}$ is a regular critical point with multiplier λ then it is a stationary point if and only if $v = 0$ is a local solution to*

$$\begin{array}{ll} \max\limits_{v} & \lambda^\top v \\ s.t. & g(\bar{x}) + v \in Z. \end{array} \tag{8}$$

Proof. Recall that $\bar{x}$ is stationary if and only if $d = 0$ is a local solution of (5). In view of (7), $d = 0$ is a local solution of (5) if and only if it is a local solution of

$$\begin{array}{ll} \min\limits_{d} & -\lambda^\top \nabla g(\bar{x}) d \\ s.t. & g(\bar{x}) + \nabla g(\bar{x}) d \in Z. \end{array} \tag{9}$$

Since changes of inactive constraint functions have locally no impact on the validity of the constraint we may restrict our attention to the active constraints. In view of the regularity assumption there exists a solution $d(v)$ to the equations

$$\nabla g_i(\bar{x}) d = v_i, \quad i : g_i \text{ is active at } \bar{x}$$

for every v. If, on the one hand, a sequence v^k of feasible points for (8) converges to zero and $\lambda^\top v^k > 0$ then there exists a sequence $d(v^k)$ of feasible points for (5) converging to zero such that $\nabla f(\bar{x})^\top d(v^k) < 0$ and therefore $\bar{x}$ is not stationary. On the other hand, if $\bar{x}$ is not stationary then there exists a sequence d^k converging to zero such that $\nabla f(\bar{x})^\top d^k < 0$ and therefore $v^k = \nabla g(\bar{x}) d^k$ converges to zero as well and $\lambda^\top v^k > 0$, i.e. the origin is not a local solution of (8). Q.E.D.

We emphasize again that stationarity at regular points is fully characterized by the multipliers λ, the value $\bar{z} = g(\bar{x})$, and the properties of Z in a neighborhood of $\bar{z}$. There is no need for information about the function f or the constraint functions g in the vicinity of $\bar{x}$.

Notice that small inactive components v_i will neither affect the validity of the constraint $g(\bar{x}) + v \in Z$ nor, since the corresponding multipliers vanish, the value of the objective function. If we reduce the optimization problem (8) to the active components by replacing Z by its projection $Z_{\mathcal{A}}$ onto the subspace of active components, we obtain the optimization problem

$$\begin{array}{ll} \max\limits_{v} & \lambda_{\mathcal{A}}^\top v_{\mathcal{A}} \\ s.t. & g_{\mathcal{A}}(\bar{x}) + v_{\mathcal{A}} \in Z_{\mathcal{A}} \end{array} \tag{10}$$

which has a local minimizer at $v_{\mathcal{A}} = 0$ if and only if the origin is a local minimizer of the original problem (8). The latter optimization problem (10) allows us to define

strict complementarity as the requirement that there exists a neighborhood V of $\lambda_{\mathcal{A}}$ such that the origin $v_{\mathcal{A}} = 0$ remains a local minimizer of (10) if $\lambda_{\mathcal{A}}$ is replaced by any $\tilde{\lambda}_{\mathcal{A}} \in V$. In the classical case of nonlinear programming this is equivalent to the requirement that the multipliers corresponding to active inequalities are strictly positive. This definition will turn out to be useful for a convergence analysis of SQP later on.

It is illustrative to compare the concepts of critical and stationary point in the following example which, although a standard nonlinear program, is re-formulated in a non-standard way that fits our framework. Consider the problem

$$\begin{array}{ll} \min & \frac{1}{2}[(x_1-1)^2 + (x_2-1)^2] \\ s.t. & x \in Z = \{z \mid z_1 + z_2 \leq 0\}. \end{array}$$

The interesting difference of this formulation from the standard formulation $g(x) = x_1 + x_2 \leq 0$ is that our chosen structure set Z is not pointed. In our setting, the active components are z_1 and z_2 if $z_1 + z_2 = 0$ and none otherwise. Critical points can only occur where both components are active and the criticality condition

$$\begin{array}{rcl} x_1 - 1 + \lambda_1 & = & 0 \\ x_2 - 1 + \lambda_2 & = & 0 \end{array}$$

is in fact satisfied for all such points, i.e., all point x with $x_1 + x_2 = 0$ are critical with multipliers $\lambda_i = 1 - x_i$, $i = 1, 2$. The stationarity condition requires that the origin solves

$$\begin{array}{ll} \max & \lambda^\top v \\ s.t. & x + v \in Z \end{array}$$

Since $x_1 + x_2 = 0$, $x + v \in Z$ if and only if $v_1 + v_2 \leq 0$ and the origin is a maximizer if and only if λ is a non-negative multiple of the vector $e = (1, 1)$. Since $\lambda = e - x$ and $x_1 + x_2 = 0$ this holds only for $x = (0, 0)$. The minimizer of the original problem is therefore the only stationary point. Notice that, in contrast to the standard NLP formulation, the stationary point does not satisfy strict complementarity in the re-formulation with the structure set Z. This is due to the non-pointedness of the chosen structure set Z.

The following concept is useful for the further specification of Proposition 4.2 for the case of locally star-shaped structure sets Z.

Definition 4.2 *Let Z be locally star-shaped and let $z \in Z$. Then the localization of Z at z is the cone*

$$C_Z(z) = \mathrm{cl}\{v \mid \exists \alpha_v > 0 : \ z + \alpha v \in Z \ \forall \alpha \in [0, \alpha_v]\}.$$

Notice that the linear subspace spanned by the inactive components is an element of $C_Z(z)$. Readers familiar with nonsmooth analysis will realize that $C_Z(z)$ coincides with Bouligand's contingent cone of the locally star-shaped set Z at z, see [3], which contains all vectors v such that $z + \alpha_k v_k \in Z$ for some $v_k \to v$ and $\alpha_k \to 0$ from

above.

Recall that the polar cone of a cone C is defined by $C^o = \{y \mid y^\top x \leq 0,\ \forall x \in C\}$, see [22]. With this notion, the following corollary follows directly from the foregoing proposition.

Corollary 4.1 *If Z is closed and locally star-shaped at $g(\bar{x}) \in Z$ then a regular critical point $\bar{x}$ with associated multiplier λ is stationary if and only if $\lambda \in C_Z^o(z)$.*

The simple example

$$\begin{array}{ll} \max & x_2 \\ s.t. & (x_1, x_2) \in Z = \{(z_1, z_2) \mid z_2 = z_1^2\} \end{array}$$

with the origin as reference point shows that the "only if" part of this corollary does not hold if $C_Z(z)$ is replaced by the Bouligand cone of a non-star-shaped set.

If Z is closed and locally star-shaped then, in view of the latter corollary, checking stationarity for a regular feasible point is, in principle, a convex feasibility problem. Its computational tractability, however, depends on the availability of a suitable representation of $C_Z^o(z)$. If, for example $Z = \{(u, v) \mid \min\{u, v\} = 0\}$, where the minimum operation is taken componentwise over the two vectors $u, v \in \mathbb{R}^k$ then the set $C_Z(0)$ is the union of the 2^k convex cones

$$\begin{array}{rll} C_I = \{(u, v) \mid & u_i = 0, v_i \geq 0, & \text{if } i \in I \\ & u_i \geq 0, v_i = 0, & \text{if } i \notin I\}, \end{array}$$

$I \subseteq \{1, \ldots, k\}$. The convex hull of $C_Z(0)$, however, is the nonnegative orthant in R^{2k}, i.e., in this case the combinatorial difficulty of checking $\lambda^\top x \leq 0$ for every $x \in C_I$ and every $I \subseteq \{1, \ldots, k\}$ disappears. This remarkable fact lies at the heart of all the recently developed efficient methods for the local optimization of MPECs. Indeed, if verifying stationarity was a difficult combinatorial problem it would seem unlikely that an efficient method can be guaranteed to converge to a stationary point. By the same token, if the verification of stationarity for a problem of type (1) is tractable, then there is hope that an efficient local optimization method can be developed for this type of problem. The essential question is therefore, whether the convex hull of $C_Z(z)$ can be represented in a computationally tractable way. The answer to this question obviously depends on the initial representation of $C_Z(z)$. If, for example, the set is given as the union of polyhedral cones then the convex hull of $C_Z(z)$ is the sum of all these cones. If the cones are given in "primal" form $C_i = \{x \mid A_i x \leq 0\}$, $i = 1, \ldots, m$, then the stationarity problem $\lambda \in C_Z(z)^o$ is equivalent to

$$\lambda^\top (x_1 + \cdots + x_m) \leq 0, \quad \forall (x_1, \ldots, x_n): \quad A_1 x_1 \leq 0, \ldots, A_m x_m \leq 0.$$

This can be checked efficiently by a decomposition method as long as the number m of polyhedral cones in the union is not too large. Similarly, if the cones are given in dual form $C_i = \{x \mid x = A_i^\top \lambda_i, \lambda_i \geq 0\}$, $i = 1, \ldots, m$ then $C_Z(z) = \{x \mid x = A^\top \lambda, \lambda \geq 0\}$, where A contains all rows of the matrices A_i without duplication. This

generator matrix A can be quite small even if m is large. In the MPEC example above, the cones C_I are of the form

$$C_I = \{(u,v) \mid (u,v) = \sum_{i \in I} \lambda_i (e_i, 0) + \sum_{i \notin I} \mu_i (0, e_i),\ \lambda_i, \mu_i \geq 0\},$$

where e_i is the ith unit vector. The overall matrix A is therefore the unit matrix in R^{2k} even though the set $C_Z(z)$ is the union of $m = 2^k$ cones C_I.

As a further example, consider the structure set Z of the optimization problem with max-min constraints (3). After applying the local reduction used in the proof of Proposition 3.1 the structure set is locally represented as $Z = \{z \mid p(z) \leq 0\}$ with

$$p(z) = \max_{j \in \bar{J}} \min_{i \in \bar{N}_j} z_i,$$

where $\bar{z}_i = 0$ for every

$$i \in \bar{I} = \bigcup_{j \in \bar{J}} \bar{N}_j$$

and $\bar{N}_k \subseteq \bar{N}_j$ implies $\bar{N}_k = \bar{N}_j$. Obviously, $v \in C_Z(z)$ if and only if $p(v) \leq 0$ which is the case if and only if

$$v \in V = \bigcup_{r \in \times_{j \in \bar{J}} \bar{N}_j} \{w \mid w_{r_j} \leq 0\}.$$

If for some $i \in \bar{I}$ there exists a tuple $r \in \times_{j \in \bar{J}} \bar{N}_j$ with $r_j \neq i$ for every j then e_i as well as $-e_i$ are contained in V since there is a set on the right-hand side which has no constraint on w_i. On the other hand if for some $i \in \bar{I}$ every tuple $r \in \times_{j \in \bar{J}} \bar{N}_j$ has one index $r_j = i$ then $-e_i \in V$ but $e_i \notin V$ since the constraint $w_i \leq 0$ is present in every set on the right-hand side. It follows that the cone $\mathrm{conv} C_Z(z) = \mathrm{conv} V$ is the direct product of the linear subspace generated by the unit vectors e_i corresponding to inactive components v_i and those components v_i for which there exists a tuple $r \in \times_{j \in \bar{J}} \bar{N}_j$ with $r_j \neq i$ for every j and the cone generated by the negative unit vectors $-e_i$ corresponding to those i which are contained in every tuple $r \in \times_{j \in \bar{J}} \bar{N}_j$. The latter indices are precisely those for which $\bar{N}_j$ is a singleton. The following corollary is a direct consequence of this observation and the stationarity condition $\lambda \in C_Z(g(x))^o$ for regular x.

Corollary 4.2 *Suppose a regular feasible point $\bar{x}$ of (3) is critical with associated multiplier vector λ and assume the notation of Proposition 3.1 for $\bar{z} = g(\bar{x})$. Then $\bar{x}$ is stationary if and only if $\lambda_i \geq 0$ for every i such that $\bar{N}_j = \{i\}$ and $\lambda_i = 0$ for all other indices i.*

If $\lambda_i < 0$ for some i such that $\bar{N}_j = \{i\}$ then, following the arguments in the proof of Proposition 4.2, $\nabla f(\bar{x}) d < 0$ for every solution d of the equations

$$\nabla g_i(\bar{x}) d = -e_i, \quad \forall i : g_i \text{ is active at } \bar{x}.$$

To illustrate the strength of this optimality condition in comparison with the often used condition based on Clarke's subdifferential [5], we finally specify the optimality condition for the case of the unconstrained max-min problem

$$\min_x \max_{1\le j\le r} \min_{i\in N_j} f_i(x) \tag{11}$$

which can be recast as (2). In view of Proposition 3.1 it is possible to compute the set of active functions f_i at $\bar{x}$ in the following way:

1. Set $\alpha := \max_{1\le j\le r} \min_{i\in N_j} f_i(\bar{x})$
2. Determine $J = \{j \mid \min_{i\in N_j} f_i(\bar{x}) = \alpha\}$
3. Determine $\bar{N}_j = \{i \in N_j \mid f_i(\bar{x}) = \alpha\}$ for every $j \in J$.
4. Delete from J all indices j for which there exists $k \in J$ such that $\bar{N}_k$ is a proper subset of $\bar{N}_j$ and call the remaining index set $\bar{J}$.

A function f_i is active at $\bar{z}$ if and only $i \in \cup_{j\in\bar{J}} \bar{N}_j$. The following corollary is a direct consequence from the foregoing corollary in view of the reformulation (2) of (11).

Corollary 4.3 *If $\bar{x}$ is a minimizer of (11) and the gradient of all active functions f_i are affinely independent then zero is in the convex hull of the gradients of those active f_{i_0} for which there exists an index j_0 with $i_0 \in N_{j_0}$ and $f_{i_0}(\bar{x}) < f_i(\bar{x})$ for every $i \in N_{j_0}\backslash\{i_0\}$.*

Notice that a sufficient condition for the gradients of all active functions f_i to be affinely independent is that the gradients of all functions with $f_i(\bar{x}) = \alpha$ are affinely independent. One may wish to call the latter weakly active. Notice also that the above optimality condition is in general considerably stronger than the often employed condition based on the Clarke subdifferential. Indeed, if the gradients of the active selection functions of a max-min function are affinely independent, then the Clarke subdifferential of the max-min function consists of the convex hull of the gradients of all active functions[1]. It is not difficult to see that this argument extends to the stationarity condition of Clarke [4] for constrained problems if applied to (3).

[1]This can be seen as follows: It suffices to show that every gradient $\nabla f_i(\bar{x})$ is contained in the subdifferential. Suppose w.l.o.g. that $f(\bar{x}) = 0$. First reduce the max-min function to a selection of active functions: Remove all non-vanishing min-functions and in each vanishing min-function all non-vanishing selection functions f_i. This results in a max-min function which coincides locally with the original function but has only vanishing selection functions. Next delete all min-functions corresponding to sets N_j with $N_j \subseteq N_k$ for some $j \neq k$. This will not change the function since these min-functions were redundant. Now choose an index i_0 and a set N_j containing i_0. Renumber the other selection functions so that $N_j = \{1, \ldots, i_0\}$ for some j. Consider the inequality system

$$f_1(x) > \ldots > f_{i_0}(x) > \ldots > f_m(x).$$

Notice that the affine independence of the vectors $\nabla f_i(\bar{x})$ implies the linear independence of the vectors $\nabla f_i(\bar{x}) - \nabla f_{i+1}(\bar{x})$. The implicit function theorem therefore implies that the above set is nonempty with boundary point $\bar{x}$. Moreover, the max-min function coincides only with the selection function $f_{i_0}(x)$ on the above set. This proves that $\nabla f_{i_0}(\bar{x})$ is contained in the Clarke subdifferential.

5 Sequential Quadratic Programming

The Lagrange multiplier-based stationarity condition of the last section allows a straight-forward extension of the sequential quadratic programming idea to combinatorial nonlinear programs. We will discuss this extension in this section and will argue that the traditional convergence arguments for standard nonlinear programs extend, mutatis mutandis, to combinatorial programs. For the sake of simplicity and transparency and in the interest of brevity we have chosen to mimic the traditional textbook analysis, see [9, 20, 28], using the standard assumptions of regularity, strict complementarity, and a second order condition. Our aim here is merely to argue that the SQP method is a sensible procedure for combinatorial nonlinear programs rather than to provide a comprehensive study of its convergence properties under the weakest possible assumptions. In particular the strict complementarity assumption requires that the projection of the structure set Z onto the subspace of active components is "pointed" at the solution as illustrated in an example in the foregoing section. Whilst this is true for the sample problems mentioned in this paper, it may not be the case for other interesting problems which fit the modelling framework. It is an interesting question whether some of these assumptions can be relaxed, e.g. by extending the more sophisticated convergence analyzes of SQP methods, e.g. [2], or of Newton's method for nonsmooth or generalized equations, e.g. [8, 16, 21], to the present setting. However, this constitutes a research question in its own and is beyond the scope of the present study.

Recall that the problem we wish to solve is of the form

$$\begin{array}{ll} \min & f(x) \\ s.t. & g(x) \in Z. \end{array} \tag{12}$$

Given a point x and a multiplier estimate λ we can formulate the SQP subproblem

$$\begin{array}{ll} \min & \nabla f(x)d + \frac{1}{2}d^\top \nabla^2_{xx} L(x,\lambda)d \\ s.t. & g(x) + \nabla g(x)d \in Z. \end{array} \tag{13}$$

We will now argue that SQP converges locally quadratically to a stationary point x^* under the following assumptions:

- Z is locally star-shaped at $z^* = g(x^*)$,
- x^* is a regular stationary point of (12) with multiplier λ^*,
- strict complementarity holds at x^*,
- $d^\top \nabla^2_{xx} L(x^*, \lambda^*)d > 0$ for every non-vanishing direction d with $\nabla g_i(x^*)d = 0$ for all active constraint functions g_i at x^*.

The latter second order condition is the traditional second order sufficient condition for the equality constrained program

$$\begin{array}{ll} \min & f(x) \\ s.t. & g_i(x) = z^*_i, i \in \mathcal{A} \end{array} \tag{14}$$

where $z^* = g(x^*)$ and $\mathcal{A}$ is the set of active constraint indices at x^*. To simplify our exposition we assume that all constraint functions are active at x^*. This is without loss of generality since inactive constraint functions will remain inactive in a neighborhood of x^* and can therefore be discarded for a local analysis. With this assumption the criticality conditions for (12) become

$$\begin{aligned} \nabla f(x^*) + \nabla g(x^*)^\top \lambda^* &= 0 \\ g(x^*) &= z^*. \end{aligned} \tag{15}$$

The criticality conditions for the subproblem are

$$\begin{aligned} \nabla f(x) + \nabla^2 L(x,\lambda) d + \nabla g(x)^\top \mu &= 0 \\ g(x) + \nabla g(x) d &= z \\ z &\in Z. \end{aligned} \tag{16}$$

Since x^* is critical with multiplier λ^*, we know that $d = 0$ and $\mu = \lambda^*$ solves the latter system, provided $(x, \lambda, z) = (x^*, \lambda^*, z^*)$. In view of our second order condition the implicit function theorem implies that the two equations in (16) define locally unique smooth implicit functions $d(x, \lambda, z)$ and $\mu(x, \lambda, z)$. Next we employ the strict complementarity condition. Since we have assumed that all constraint functions are active, strict complementarity means that $v = 0$ is a local solution to

$$\begin{aligned} \min \quad & \mu^\top v \\ s.t. \quad & z^* + v \in Z \end{aligned} \tag{17}$$

for every μ close to λ^*. Therefore the critical points $d(x, \lambda, z^*)$ with multipliers $\mu(x, \lambda, z^*)$ are indeed stationary points of the subproblems for (x, λ) close to (x^*, λ^*). We finally show that they are the only stationary points of the subproblems close to the origin. To this end, suppose that $d(x, \lambda, z)$ is stationary for some $z \neq z^*$. We will now make use of the fact that Z is locally star-shaped at z^* and therefore the line segment $[z^*, z]$ will be a subset of Z for z close to z^*. Since $\mu = \mu(x, \lambda, z)$ is close to λ^*, strict complementarity implies that $v = 0$ is a local solution of (17). Also, stationarity of $d(x, \lambda, z)$ means that $w = 0$ is a local solution of

$$\begin{aligned} \min \quad & \mu^\top w \\ s.t. \quad & z + w \in Z. \end{aligned} \tag{18}$$

Therefore, using the fact that $[z^*, z]$ is contained in Z, μ is perpendicular to $z^* - z$ and therefore $u = 0$ is not a local solution of

$$\begin{aligned} \min \quad & [\mu + (z^* - z)]^\top u \\ s.t. \quad & z^* + u \in Z. \end{aligned} \tag{19}$$

This contradicts strict complementarity, since $\mu + (z^* - z)$ can be chosen arbitrarily close to λ^*.

We can now repeat the classical argument that under our assumptions SQP is equivalent to Newton's method applied to the optimality conditions with active set identification and converges quadratically if started at a point (x, λ) close to (x^*, λ^*).

The penalty trust-region globalization scheme of [25] and [27] applies to this setting, provided an appropriate residual function p is available. This function needs to be locally Lipschitz and an indicator for the set Z in the sense that $p \geq 0$ and $p(z) = 0$ if and only if $z \in Z$. In addition, it needs to provide an exact penalization of the constraints. This is, for example, the case if p is piecewise affine and the constraints are regular, see [25]. For a general approach to Lipschitzian penalty function we refer the reader to the recent paper [15]. For a comprehensive study of trust-region globalization we refer to the book [7].

6 A Decomposition Approach to Combinatorial Problems

In order to apply the SQP method of the last section, we need to have a method to solve the subproblems, which are combinatorial quadratic programmes. Efficient methods for the solution of such problems will most definitely have to make use of the specific structure of Z. Nevertheless, there is a simple decomposition approach which may guide the development of such methods. In this section we will briefly explain the general decomposition approach and show how it can be applied to the subproblems arising from max-min optimization. We begin again with our general problem

$$\begin{array}{ll} \min & f(x) \\ s.t. & g(x) \in Z. \end{array} \tag{20}$$

The decomposition approach for problem (20) is based on a collection of finitely many subsets $Z_i \subseteq Z$. These subsets, which we call pieces, give rise to subproblems of the form

$$\begin{array}{ll} \min & f(x) \\ s.t. & g(x) \in Z_i. \end{array} \tag{21}$$

We assume that the subproblems are simpler than the full problem (20) in the sense that a (descent) method is available to solve the subproblems (21) from a given feasible point. In order to extend the solution procedure for the subproblems to a method for the full problem (20) there has to be a link between optimality or stationarity of the subproblems and optimality or stationarity of the full problem (20). We focus on stationarity in the sequel and require

1. If (20) has a stationary point then $g(x) \in Z_i$ for at least one such stationary point x and one piece Z_i.
2. If x is a stationary point of (21) for all adjacent pieces Z_i then it is stationary for the full problem (20).

Here, a piece Z_i is called adjacent to x if $g(x) \in Z_i$. Notice that both conditions are automatically satisfied if the union of the sets Z_i coincides with Z.

Starting from a piece Z_{i_0} and a point x^0 with $g(x^0) \in Z_{i_0}$, the k-th iteration of the decomposition method proceeds as follows:

1. Given x^{k-1} with $g(x^{k-1}) \in Z_{i_{k-1}}$, use a descent method to solve (21) for $Z_i = Z_{i_{k-1}}$, starting from x^{k-1}.
2. If the descent method does not produce a stationary point of (21) then stop and report the failure of the descent method (e.g. divergence to $-\infty$ or non-convergence).
3. If the method produces a stationary point x^k of (21) and this point is stationary for every adjacent piece then stop; a stationary point of (20) has been found.
4. Otherwise, choose a piece Z_{i_k} adjacent to x^k for which x^k is not stationary and continue.

If the decomposition method stops after finitely many iterations then it either reports failure of the descent method or produces a stationary point x^k of (20). If each subproblem (21) has only finitely many stationary levels then the method stops after finitely many iterations since it is a descent method and there are only finitely many pieces Z_i. Here, a stationary level of (21) is a value $f(x)$, where x is a stationary point of (21). Obviously convex subproblems have at most one stationary level.

The design of a decomposition method is therefore broken down into three parts:

1. Decide on the partition of Z into pieces Z_i.
2. Decide on a descent method to solve the subproblems (21).
3. Develop a mechanism to decide whether the solution of a subproblem is stationary for every adjacent piece and, if not, to produce a piece adjacent to x^k for which x^k is not stationary.

It seems most desirable to make use of the multipliers corresponding to the subproblem at iteration k to decide whether x^k is stationary for every adjacent piece or, if not, to determine a descent piece.

As an example consider the decomposition method for MPECs

$$\begin{array}{ll} \min & f(z) \\ s.t. & \min\{G(z), H(z)\} = 0. \end{array}$$

One starts with an index set I and solves the standard nonlinear program

$$\begin{array}{ll} \min & f(z) \\ s.t. & G_i(z) = 0,\ i \in I \\ & G_j(z) \geq 0,\ j \notin I \\ & H_i(z) \geq 0,\ i \in I \\ & H_j(z) = 0,\ i \notin I. \end{array}$$

If $\bar{z}$ is a regular stationary point of this NLP and the multipliers corresponding to bi-active constraints $H_i(z) = G_i(z) = 0$ are both nonnegative then $\bar{z}$ is stationary for the MPEC. Otherwise one adds a bi-active index i to I if the multiplier corresponding to H_i is negative or removes i form I if the multiplier corresponding to G_i

is negative and iterates.

A similar method can be used to solve (3). The basis for this method is the following simple observation.

Lemma 6.1 $\max\limits_{1\le j\le r}\min\limits_{i\in N_j} z_i = \min\limits_{s\in \times_{j=1}^r N_j} \max\{z_{s_1},\ldots,z_{s_r}\}$.

Proof. On the one hand, if $s_j \in N_j$ with $z_{s_j} = \min_{i\in N_j} z_i$ then

$$\begin{aligned}\max_{1\le j\le r}\min_{i\in N_j} z_i &= \max_{1\le j\le r} z_{s_j}\\ &\ge \min_{s\in \times_{j=1}^r N_j} \max\{z_{s_1},\ldots,z_{s_r}\}.\end{aligned}$$

One the other hand $\min\limits_{i\in N_{j_0}} z_i \le \min\limits_{s\in \times_{j=1}^r N_j} \max\{z_{s_1},\ldots,z_{s_r}\}$ for every j_0 and therefore $\max\limits_{1\le j\le r}\min\limits_{i\in N_j} z_i \le \min\limits_{s\in \times_{j=1}^r N_j} \max\{z_{s_1},\ldots,z_{s_r}\}$. Q.E.D.

In view of the lemma the max-min optimization problem (3) can be reformulated as

$$\begin{array}{lll}\min\limits_x & f(x) & = \min\limits_{s\in\times_{j=1}^r N_j}\ \min\limits_x \quad f(x)\\ \text{s.t.} & \max\limits_{1\le j\le r}\min\limits_{i\in N_j} g_i(x)\le 0 & \text{s.t.}\quad g_{s_i}(x)\le 0, i=1,\ldots,r\end{array}$$

Given a combination $s\in\times_{j=1}^r N_j$ the decomposition method now proceeds as follows:

1. Find a stationary point $\bar{x}$ of

$$\begin{array}{ll}\min & f(x)\\ \text{s.t.} & g_i(x)\le 0, i\in\{s_1,\ldots,s_r\}.\end{array} \tag{22}$$

   We assume that constraints only appear once if $s_j = s_k$ for some $j\ne k$. If the gradients of the vanishing constraint functions at $\bar{x}$ are linearly independent then we will obtain a unique multiplier vector $\bar{\lambda}$ satisfying

$$\begin{aligned}\nabla f(\bar{x}) + \nabla g(\bar{x})^\top \bar{\lambda} &= 0\\ \bar{\lambda} &\ge 0\\ \bar{\lambda}_k &= 0,\ \text{ if } k\notin\{s_1,\ldots,s_r\} \text{ or } g_k(\bar{x})\ne 0\end{aligned}$$

2. Compute the sets $\bar{J}$ and $\bar{N}_j$ defined in Proposition 3.1 for $\bar{z} = g(\bar{x})$. According to Proposition 3.1 the functions g_i, $i\in\cup_{j\in\bar{J}}\bar{N}_j$ are the active constraint functions at $\bar{x}$.

3. We assume that the active constraint functions at $\bar{x}$ are linearly independent. Then, according to Corollary 4.2, $\bar{x}$ is stationary if and only if for every $\bar{\lambda}_{i_0} > 0$ there exists an index j_0 with $i_0\in N_{j_0}$ and $g_{i_0}(\bar{x}) < g_i(\bar{x})$ for every $i\in N_{j_0}\backslash\{i_0\}$. If this is the case for every index i_0 with $\lambda_{i_0} > 0$ then stop.

4. If this is not the case for some i_0 with $\lambda_{i_0} > 0$ then change s in the following way: For every j with $s_j = i_0$ set $s_j := i_1$, where $i_1 \in N_j$, $i_1 \neq i_0$, and $g_{i_1}(\bar{x}) \leq g_{i_0}(\bar{x})$. Go to step 1.

This method is only conceptual since the solution of the subproblems may not be obtainable in a finite number of iterations if nonlinear functions are present. This issue would have to be addressed in an implementation. We will not dwell on these details since we suggest to use this method to solve the combinatorial *quadratic programs* arising as subproblems in the SQP method, rather than using the method directly to solve max-min optimization problems as a sequence of nonlinear programs. One advantage of the SQP approach is that no new derivative information is needed during the solution of the subproblems. This makes the SQP approach more efficient than a direct composition approach.

Conceptually the method is well-defined if the active constraint functions defined in step 2 are linearly independent at every visited stationary point $\bar{x}$. We will comment on the stringency of this condition below. Let us assume it holds and let us also assume that the method to solve (22), if it converges, produces a point with a lower objective value than the starting point, provided the starting is not stationary. Then we will either observe non-convergence in the subproblem, e.g. due to an unbounded objective function, or we obtain a stationary point of the subproblem. This point is either stationary for the overall problem as tested in step 3, or the method will continue with a different subproblem. In the non-convex case we can possibly revisit the subproblem again, but only finitely often if it has only finitely many stationary values as argued above. Since there are only finitely many subproblems this will lead to finite convergence to a stationary point. Recall that convex subproblems have a single stationary level. Moreover, if the subproblems are convex then stationarity at a subproblem is equivalent to minimality and therefore stationarity for the master problem implies minimality on every adjacent piece and therefore local minimality for the master problem. Summing this up we obtain the following proposition.

Proposition 6.1 *If the functions f and g_i are smooth and convex and the gradients of the active functions at the generated solutions of all subproblems are linearly independent then the decomposition method stops after finitely many iterations and either produces a local minimizer or reports that a particular subproblem could not be solved.*

The underlying reason why this method works is that the multipliers contain sensitivity information. Indeed, the subproblem is of the form

$$\begin{array}{ll} \min & f(x) \\ s.t. & g_{s_j}(x) \leq 0, \quad j = 1, \ldots, r. \end{array}$$

We assume that constraints corresponding to indices $s_j = s_{j'}$ occur only once. Suppose $\bar{x}$ is a stationary point of the subproblem with a multiplier vector $\bar{\lambda} \in \mathbb{R}^m$ with $\bar{\lambda}_i = 0$ if $i \notin \{s_1, \ldots, s_r\}$ and

$$\nabla f(\bar{x}) + \sum_{j=1}^{r} \bar{\lambda}_{s_j} \nabla g_{s_j}(\bar{x}) \quad = \quad 0$$

$$\begin{aligned}\bar{\lambda}_{s_j} &\geq 0\\ \bar{\lambda}_{s_j} &= 0 \quad \forall j : g_{s_j}(\bar{x}) < 0.\end{aligned}$$

Suppose there exists an index $s_{j_0} = i_0$ with $\bar{\lambda}_{i_0} > 0$ and that for every j with $s_j = i_0$ there exists an $i_j \in N_j$ different from i_0 with $g_{i_j}(\bar{x}) \leq 0$. Notice that $\bar{x}$ is also a stationary point of

$$\begin{array}{ll}\min & f(x)\\ s.t. & g_{s_j}(x) \leq 0, \quad j : s_j \neq i_0\\ & g_{i_0}(x) \leq 0,\\ & g_{i_j}(x) \leq 0, \quad j : s_j = i_0,\end{array}$$

with the same multiplier vector $\bar{\lambda}$. Here we assume again that each constraint appears only once. If the active gradients are linearly independent then the multiplier vector $\bar{\lambda}_{i_0} > 0$ indicates that removing the constraint $g_{i_0}(x) \leq 0$ will improve the objective value. This can indeed be done by replacing $s_j = i_0$ by $s_j := i_j$. We do this and proceed. If $\bar{\lambda}_{i_0} > 0$ implies that there exists an index j_0 with $s_{j_0} = i_0$ and $g_i(\bar{x}) \neq 0$ for every $i \in N_{j_0} \backslash \{i_0\}$ then $\bar{x}$ is a stationary point.

Let us finally return to a discussion of the assumption that the gradients of the active functions at the generated solutions of the subproblems are linearly independent. Sard's theorem as applied in [24] suggests that this assumption is not particularly stringent, i.e. given a particular constraint function g the linear independence assumption will be satisfied for a perturbed constraint function $g(x) + a$ for almost all a in the Lebesgue sense. This result also holds for the functions f used in the reformulation (2) of the unconstrained max-min problem. We give a brief proof of this result for completeness.

Proposition 6.2 *Given a smooth mapping $f : \mathbb{R}^n \to \mathbb{R}^m$ and a vector $a \in \mathbb{R}^m$ define*

$$\begin{aligned}f^a(x) &= f(x) + a\\ I_i^a(x) &= \{j \in \{1, \ldots, m\} \mid f_j^a(x) = f_i^a(x)\}\end{aligned}$$

for $x \in \mathbb{R}^n$ and $1 \leq i \leq m$. For almost all $a \in \mathbb{R}^m$ (w.r.t. the Lebesgue measure) the gradients $\nabla f_j^a(x), j \in I_i^a(x)$ are affinely independent at every point x and for every $i \in \{1, \ldots, m\}$.

Proof. Fix an index set $I \subseteq \{1, \ldots, m\}$ and an index $i \in I$ and consider the equations

$$f_j(x) - f_i(x) = a_i - a_j, \quad j \in I \backslash \{i\}.$$

Let (C) be the condition that the vectors $\nabla f_j(x) - \nabla f_i(x)$, $j \in I \backslash \{i\}$, are linearly independent at every solution x of the equations. By Sard's theorem, for any fixed a_i the set of vectors $a \in \mathbb{R}^{I \backslash \{i\}}$ such that condition (C) is violated has Lebesgue measure zero in $\mathbb{R}^{I \backslash \{i\}}$. Thus, by Fubini's theorem, the set of vectors $a \in \mathbb{R}^I$ such that condition (C) is violated has Lebesgue measure zero in $\mathbb{R}^I$. Since the finite union of Lebesgue null sets is again a Lebesgue null set it follows that for almost all

$a \in \mathbb{R}^n$ condition (C) holds for every $I \subseteq \{1, \ldots, m\}$ and every $i \in I$. To complete the proof it suffices to show that the fact that for fixed I the vectors $\nabla f_j(x) - \nabla f_i(x)$, $i \in I \backslash \{i\}$ are linearly independent for every $i \in I$ implies that the gradients $\nabla f_j(x)$, $j \in I$, are affinely independent. Indeed, if $\nabla f_j(x)$, $j \in I$ are affinely dependent then there exists a non-vanishing vector λ such that $\sum_{j \in I} \lambda_j \nabla f_j(x) = 0$ and $\sum_{j \in I} \lambda_j = 0$. If $i \in I$ is such that $\lambda_i \neq 0$ then

$$\sum_{\substack{j \in I \\ j \neq i}} \frac{\lambda_j}{\lambda_i} (\nabla f_j(x) - \nabla f_i(x)) = 0$$

and thus $\nabla f_j(x) - \nabla f_i(x)$, $j \in I \backslash \{i\}$ are linearly dependent. Q.E.D.

6.1 Preliminary Numerical Experience

The method of the last section was coded for problems of the form

$$\begin{array}{ll} \min & \frac{1}{2} x^\top Q x + q^\top x \\ s.t. & \max\limits_{1 \leq j \leq l} \min\limits_{i \in N_j} a_i^\top x - b_i \leq 0 \end{array}$$

with positive definite matrix Q. The code was written in MATLAB and used Roger Fletcher's BQPD code to solve the QPs. We checked for replacement of active constraint functions $g_i(x)$ in the order of descending multipliers and replaced a representant g_i of the min-function $\min_{i \in N_j} g_i(x)$ by the function g_k, $k \in N_j \backslash \{i\}$, with the minimal value at the current iteration point, provided that value was non-positive.

A typical set of test problems was generated as follows: We generated an $m \times n$ matrix A using the RANDN or RAND functions in MATLAB. The use of the latter guarantees feasibility since A has positive entries and therefore $Ax - b < 0$ for sufficiently large positive x. We chose b to be the m-vector with all components equal to -1 so that the origin is infeasible for any max-min combination of the selection functions $A_i x - b_i$ since all selection functions all negative at the origin. We then solved the problem of finding a feasible solution with smallest Euclidean norm. To generate the index sets N_j, We randomly permuted the numbers $1, \ldots, 5m$ and chose N_j to be those indices among the first m components of the permutation vector which did not exceed m. The MATLAB code of the method, albeit with the MATLAB QP solver, as well as the random problem generator can be downloaded from [29].

Finding a feasible point of a max-min system is a non-trivial task in general. If an initial QP is infeasible, then one may attempt to restore feasibility by solving an appropriate phase-1 problem such as

$$\begin{array}{ll} \min\limits_{(x,s)} & e^\top s \\ s.t. & \max\limits_{1 \leq j \leq l} \min\limits_{i \in N_j} g_i(x) - s_i \leq 0 \\ & s \geq 0. \end{array}$$

starting from a feasible point $(x, s) = (\bar{x}, g(\bar{x}))$, where $\bar{x}$ could be any vector that one may wish to use as a starting point. Obviously, the phase-1 problem is only a heuristic since the method can be trapped in a local minimum of the L_1 merit function. As it turned out, a phase-1 approach was unnecessary in our tests. If the initial QP subproblem was infeasible, which occurred frequently, the BQPD code would return positive multipliers for the violated constraints which the method will swap successively for feasible constraint functions in the corresponding sets N_j. It typically took only a few iterations until a feasible point was found in this way.

The method performed credible on the random test problems. A typical problem with, e.g., $n = 100$ variables, $l = 200$ min-functions and a total of $m = 200$ selection functions was solved in 110-140 QP iterations to local optimality. We performed some tests with various randomly chosen initial QPs but in these tests the choice of initial constraint did typically not affect the number of QP iterations. Not surprisingly, the objective values achieved with different starting QPs could differ substantially. In the above mentioned test run the best of five starts outperformed the worst by up to 30%. We also performed test runs where we checked after each iteration whether the current point was Clarke stationary and occasionally found such points upon which the method could often substantially improve. It is not worth giving any more details of the numerical tests as they are very preliminary and were only performed as a sanity check and as an illustration of the practical potential of the method. More extensive numerical tests and comparisons with other methods will be necessary for a sound judgement of the numerical reliability of the method.

It is obvious that our method, being a local optimization method, can and often will be trapped in a local minimum. It is still a worthwhile method since descent is guaranteed once feasibility is detected. It is of course possible to reformulate optimization problems with max-min constraints as mixed integer NLPs, e.g. through a "big-M" approach

$$\begin{array}{ll} \min & f(x) \\ s.t. & g_i(x) + My_i \leq M \\ & \sum_{i \in N_j} y_i \geq 1, \quad j = 1, \ldots, l \\ & y \in \{0, 1\}, \end{array}$$

where M is a large constant. If the data of the original problem are linear then this results in a mixed integer linear program which is, in principle, solvable to global optimality. Even for such reformulations, which are only tractable for very moderate numbers of selection functions, the local method suggested here is useful as it can be employed to generate an initial upper bound for the objective and to improve a new upper bound in a branch and bound procedure by invoking the method from the corresponding feasible solution.

Acknowledgement. Thanks are due to my colleague Danny Ralph for many tea-room discussions on the subject and for valuable suggestions on an earlier draft. I also thank Roger Fletcher and Sven Leyffer for making the MATLAB MEX-file of

their BQPD code available to me.

References

[1] S.G. Bartels, L. Kuntz, and S. Scholtes, Continuous selections of linear functions and nonsmooth critical point theory, *Nonlinear Analysis, Theory, Methods & Applications* **24** (1995), 385-407.

[2] J.F. Bonnans and G. Launay, Implicit trust region algorithm for constrained optimization, Technical report (1992), Institute Nationale Recherche Informatique et Automation, Le Chesnay, France.

[3] G. Bouligand, Sur la semi-continuité d'inclusions et quelques sujets connexes, *Enseign. Math.* **31** (1932), 11-22.

[4] F.H. Clarke, A new approach to Lagrange multipliers, *Mathematics of Operations Research* **1** (1976), 165-174.

[5] F.H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley & Sons, New York 1983.

[6] A.R. Conn and M. Mongeau, Discontinuous piecewise linear optimization, *Mathematical Programming* **80**, 315-380.

[7] A.R. Conn, N.I.M. Gould, and Ph.L. Toint, *Trust-region methods*, SIAM Philadelphia (2000).

[8] A.L. Dontchev, Local convergence of Newton's method for generalized equations, *Comptes Rendus de l'Académie des Sciences de Paris* **332** Ser. I (1996), 327-331.

[9] R. Fletcher, *Practical Methods of Optimization,* 2nd Edition, John Wiley & Sons, Chichester 1987

[10] R. Fourer, A simplex algorithm for piecewise-linear programming I: Derivation and proof, *Mathematical Programming* **33** (1985), 204-233.

[11] R. Fourer, A simplex algorithm for piecewise-linear programming II: Finiteness, feasibility and degeneracy, *Mathematical Programming* **41** (1988), 281-315.

[12] R. Fourer, A simplex algorithm for piecewise-linear programming III: Computational analysis and applications, *Mathematical Programming* **53** (1992), 213-235.

[13] C. Kirjner-Neto and E. Polak, On the conversion of optimization problems with max-min constraints to standard optimization problems, *SIAM J. Optimization* **8** (1998), 887-915.

[14] D. Klatte and B. Kummer, *Nonsmooth Equations in Optimization*, Kluwer Academic Publishers, Boston 2002.

[15] D. Klatte and B. Kummer, Constrained minima and Lipschitzian penalities in metric spaces, *SIAM J. Opt.* 2002 (forthcoming).

[16] B. Kummer, Newton's method based on generalized derivatives for nonsmooth functions: convergence analysis, in: *Advances in Optimization* (eds. W. Oettli and D. Pallaschke), Springer Verlag, Berlin 1992, pp. 171-194.

[17] C. Lemaréchal, Bundle methods in nonsmooth optimization, in: *Proceedings of the IIASA Workshop, Nonsmooth Optimization* (eds. C. Lemaréchal and R. Mifflin), Pergamon Press, Oxford (1978), 79-102.

[18] Z.-Q. Luo, J.-S. Pang, D. Ralph, *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press 1996.

[19] D. Ralph and E. Polak, A first-order algorithm for semi-infinite min-max-min problems, Manuscript 2000.

[20] S.M. Robinson, Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear programming algorithms, *Mathematical Programming* **7** (1974), 1-16.

[21] S.M. Robinson, Newton's method for a class of nonsmooth functions, *Set-Valued Analysis*, **2** (1994), 291-305.

[22] R.T. Rockafellar, *Convex Analysis*, Princeton University Press 1970.

[23] H. Scheel and S. Scholtes, Mathematical Programs with Complementarity Constraints: Stationarity, Optimality, and Sensitivity,

[24] S. Scholtes and M. Stöhr, How stringent is the linear independence assumption for mathematical programs with stationarity constraints? *Mathematics of Operations Research* (forthcoming)

[25] S. Scholtes and M. Stöhr, Exact penalization of mathematical programs with equilibrium constraints, *SIAM Journal on Control and Optimization* **37** (1999), 617-652.

[26] H. Schramm and J. Zowe, A version of the bundle idear for minimizing a nonsmooth function: conceptual ideal, convergence analysis, numerical results, *SIAM J. Opt.* **2** (1992), 121-152.

[27] M. Stöhr, *Nonsmooth Trust Region Methods and Their Applications to Mathematical Programs with Equilibrium Constraints*, PhD Thesis (1999), published by Shaker-Verlag Aachen, Germany.

[28] R.B. Wilson, *A simplicial algorithm for concave programming*, PhD dissertation, Harvard Business School (1963).

[29] http://www.eng.cam.ac.uk/ ss248/publications