A VERSION OF THIS PAPER HAS APPEARED AS: MULTIPLIER CONVERGENCE IN TRUST-REGION METHODS WITH APPLICATION TO CONVERGENCE OF DECOMPOSITION METHODS FOR MPECS.

G. GIALLOMBARDO AND D. RALPH. 2008.
MATHEMATICAL PROGRAMMING 112, 335-369.

Abstract. We study piecewise decomposition methods for mathematical programs with equilibrium constraints (MPECs) for which all constraint functions are linear. At each iteration of a decomposition method, one step of a nonlinear programming scheme is applied to one piece of the MPEC to obtain the next iterate. Our goal is to understand global convergence to B-stationary points of these methods when the embedded nonlinear programming solver is a trust-region scheme, and the selection of pieces is determined using multipliers generated by solving the trust-region subproblem. To this end we study global convergence of a linear trust-region scheme for linearly-constrained NLPs that we call a trust-search method. The trust-search has two features that are critical to global convergence of decomposition methods for MPECs: a robustness property with respect to switching pieces, and a multiplier convergence result that appears to be quite new for trust-region methods. These combine to clarify and strengthen global convergence of decomposition methods without resorting either to additional conditions such as eventual inactivity of the trust-region constraint, or more complex methods that require a separate subproblem for multiplier estimation.

Key words. mathematical program with equilibrium constraints, MPEC, complementarity constraints, MPCC, B-stationary, M-stationary, linear constraints, nonlinear program

AMS subject classifications. 90C30, 90C33

1. Introduction.

We consider the following mathematical program with equilibrium constraints (MPEC), all of whose constraint functions are linear:

$$
\begin{aligned}
&\min_{x\in\mathbb{R}^n} && f(x) \\
&\text{subject to} && \min_{1\le j\le \ell}\{p_{ij}(x)\} = 0 \quad i = 1,\dots,m_p \\
& && g(x) \le 0 \\
& && h(x) = 0.
\end{aligned}
\tag{1.1}
$$

Throughout the paper we will always assume that $f : \mathbb{R}^n \mapsto \mathbb{R}$ is continuously differentiable, while $g : \mathbb{R}^n \mapsto \mathbb{R}^{m_g}$, and $h : \mathbb{R}^n \mapsto \mathbb{R}^{m_h}$ are defined as

$$g(x) \stackrel{\triangle}{=} G^\top x + v \quad \text{and} \quad h(x) \stackrel{\triangle}{=} H^\top x + w$$

where $G \in \mathbb{R}^{n\times m_g}$, $H \in \mathbb{R}^{n\times m_h}$, $v \in \mathbb{R}^{m_g}$, and $w \in \mathbb{R}^{m_h}$. Moreover for every (i, j), with $1 \le i \le m_p$ and $1 \le j \le \ell$, p_{ij} is defined as

$$p_{ij}(x) \stackrel{\triangle}{=} P_{ij}^\top x + u_{ij}$$

where P_{ij} is an n-dimensional real vector and u_{ij} is a scalar. MPECs in the form of (1.1) are sometimes called mathematical programs with complementarity constraints

*Dipartimento di Elettronica, Informatica e Sistemistica, Università della Calabria, 87036 Rende (CS), Italia, and Judge Business School, University of Cambridge, Cambridge CB2 1AG, United Kingdom. E-mail: giallo@deis.unical.it, g.giallombardo@jbs.cam.ac.uk
†Judge Business School, University of Cambridge, Cambridge CB2 1AG, United Kingdom. E-mail: d.ralph@jbs.cam.ac.uk

(MPCC), because the lower-level equilibrium problem is represented via min-functions that require, for each i, at least one function $p_{ij}(x)$ to be zero while the others must take nonnegative values, i.e., complementarity between the nonnegative scalars $p_{i1}(x)$, $\ldots$, $p_{i\ell}(x)$. More generally, the lower-level constraints may consist of an equilibrium system such as a variational inequality that is parametric in upper level variables, see [32, 35].

A disjunctive approach for dealing with the MPEC (1.1) is based on a natural local piecewise decomposition, e.g., [32, 34] when $m = 2$. For any feasible point $\bar{x}$ of (1.1) we choose a set of constraint indices $I \in \mathcal{I}(\bar{x})$, where

$$\mathcal{I}(\bar{x}) \stackrel{\triangle}{=} \{I \mid I \subseteq \{(i,j) : p_{ij}(\bar{x}) = 0\} \text{ and } \forall i \; \exists j : (i,j) \in I\}\,, \tag{1.2}$$

and define the following ordinary nonlinear program NLP_I:

$$\begin{array}{lll} \min\limits_{x \in \mathbb{R}^n} & f(x) & \\ \text{subject to} & p_{ij}(x) = 0 & \forall (i,j) \in I \\ & p_{ij}(x) \geq 0 & \forall (i,j) \notin I \\ & g(x) \leq 0 & \\ & h(x) = 0. & \end{array} \tag{1.3}$$

The feasible set of the NLP (1.3) is referred to as a *piece* (or *branch* [32]) of the MPEC feasible set at, or adjacent to, $\bar{x}$. The NLP itself is called a *piece of the MPEC* at $\bar{x}$. The index set I is called a *piece index* at $\bar{x}$, and hence the set $\mathcal{I}(\bar{x})$ is the family of all piece indices at $\bar{x}$.

It is clear that locally around $\bar{x}$, the feasible set of (1.1) looks like the union of a finite but possibly huge number of ordinary NLP feasible sets, indexed by $I \in \mathcal{I}(\bar{x})$; in our situation, this would typically result in a non-convex polyhedral set. This shows the intrinsically combinatorial nature of mathematical programs with complementarity constraints. Nevertheless, as we explain next in a brief review, decomposition methods can still work efficiently.

At the kth iteration of a decomposition method we are given a feasible point x^k of the MPEC and an adjacent piece NLP_{I^k}. If x^k is not stationary for this piece then we can improve it by finding another feasible point x^{k+1} of NLP_{I^k} such that $f(x^{k+1}) < f(x^k)$. Now suppose x^k is actually stationary for this piece of the MPEC, and let π^k be the associated vector of Lagrange or Karush-Kuhn-Tucker (KKT) multipliers. The obvious question for a decomposition approach is how to choose $I^{k+1} \in \mathcal{I}(x^k)$ such that one step of a nonlinear programming method applied to $NLP_{I^{k+1}}$ at x^k will yield a new point x^{k+1} that decreases the objective function and is feasible for $NLP_{I^{k+1}}$, hence for the MPEC. Or, if such a piece index I^{k+1} does not exist, how to show this by verifying stationarity of x^k for each NLP_I adjacent to it. The latter stationarity condition is termed *piecewise* stationarity [32] which is equivalent to *B-stationarity* [40] of x^k. Under certain constraint qualifications such as the commonly assumed MPEC-LICQ (Definition 3.1) it can be seen that the signs alone of particular components of π^k allow us to choose a suitable I^{k+1} or verify B-stationarity of x^k without further work. This dual-based approach to selection of MPEC pieces appears in [43].

1.1. Development of the paper. Our goal is to understand the global convergence to B-stationary points of decomposition methods that apply a trust-region

(TR) method to each relevant piece of the MPEC, inspired by the work of Scholtes and Stöhr [43] and Stöhr [44] on penalty-based decomposition methods for MPECs, trying in particular to address some typical shortcomings arising in the global convergence analysis of such methods. To this aim our first step is to introduce, in section 2, a simple linear(-programming-based) trust-region scheme, for linearly-constrained NLP. Apart from their simplicity and robustness, linear TR methods are attractive because of their recent impact in numerical methods for large-scale nonlinear optimization [6, 7, 10]. The key feature of our linear TR algorithm, unlike the standard TR approach [11], is that we do not allow the TR radius to be arbitrarily small at the start of a serious step (cf. [10, 24, 27, 44]). We call it a trust-search method because it combines some defining features of TR and line-search methods, i.e., it is analogous to the standard Armijo line-search which takes an initial stepsize of 1 and, if a rate-of-descent condition fails, decreases the stepsize geometrically until that condition holds. This means, as in all standard linesearch methods [18] but in contrast to the standard trust region approach, that the next iterate is *sufficiently far* from the current iterate; this property yields some convergence advantages that we now describe.

We show that the trust-search has a robust global convergence property called monotone uniform descent [36] or nonstationary repulsion [15]: every feasible nonstationary point has a neighborhood that the algorithm can visit at most once, irrespective of the (feasible) starting point. We next show convergence of the trust-region subproblem multipliers: accumulation points of the trust-search iterate-multiplier pairs necessarily solve the KKT conditions of the NLP. This result appears to be new in that it holds quite generally, without resorting either to a separate subproblem to estimate multipliers, or restrictive assumptions such as strict complementarity at the eventual KKT point or asymptotic inactivity of the trust-region constraint. We stress that enforcing a lower bound on the initial TR radius at the start of each serious iteration plays a fundamental role in proving both nonstationary repulsion and convergence of multipliers of the trust-search method.

In section 3 we aim to extend the nonstationary repulsion property of the trust-search method to the decomposition scheme based on the trust-search solver.

Suppose $(\bar{x}, \bar{\pi})$ is an accumulation point of the sequence of iterate-multiplier pairs $\{(x^k, \pi^k)\}$ generated by the decomposition method, where each x^k is feasible for the MPEC (1.1). Our main assumption, needed to validate dual-based selection of MPEC pieces, is that the MPEC-LICQ holds at $\bar{x}$. Section 3 explains this along with details of other relevant stationary conditions and their relationships with one another, and also presents the decomposition method **TS-MPEC** that uses the trust-search subroutine as its NLP workhorse.

As usual, the algorithmic proof of B-stationarity of $\bar{x}$ requires that $\bar{\pi}$ be a KKT multiplier for each piece of the MPEC adjacent to $\bar{x}$. Toward this goal, decomposition methods require multiplier convergence of the embedded NLP solver. In addition the algorithm needs to identify, implicitly rather than explicitly, the so-called *multi-active* complementarity constraints — i.e., those indices i such that more than one constraint function p_{ij} takes the value zero at $\bar{x}$ — in order to avoid converging to a point that is stationary for one piece of the MPEC but not B-stationary. For the purpose of identification it is convenient to assume that a weak strict complementarity property, ULSC (Definition 3.5) holds at $\bar{x}$. We also find that convergence to zero of the step $d^k = x^{k+1} - x^k$, on an appropriate subsequence, is needed for identification. See §3.3 for details of global convergence of **TS-MPEC**, and §3.4 for some extensions.

The requirement on convergence of the step d^k to zero leads to a slight modification of basic trust-search method in section 4. This yields global convergence of the trust-search decomposition method, Theorem 4.5, without requiring either additional assumptions, such as asymptotic inactivity of the trust region (c.f. [43, Proposition 5.1]), or a separate subproblem for multiplier estimation at each iteration as in [44].

Having mentioned the key concepts, we can view the paper at an even higher level. As we said, the linear trust-search approach for linearly constrained NLP has two attractive properties: nonstationary repulsion, and convergence of multipliers which has more obvious importance in its own right, e.g., see the theoretical and algorithmic role of multipliers in [4, 21]. However neither property seems critical to global convergence of TR methods in NLP, after all the standard TR approach appears to satisfy neither property. What is more striking, then, is the critical role played by each property in global convergence of the trust-search decomposition method for MPECs with linear constraint functions. Nonstationary repulsion makes the trust-search method robust to switching from one piece to another during the decomposition algorithm, i.e., if a subsequence of iterates $\{x^k\}_{k\in K}$ is such that $I_k = I$, a fixed piece index, on this subsequence, then any limit point $\bar{x}$ is stationary for NLP_I. Yet more is needed for $\bar{x}$ to be a B-stationary point of the MPEC: a dual-based procedure for updating the piece index relies on convergence of the multipliers, in whatever way these are estimated, to a KKT multiplier for NLP_I at $\bar{x}$. The fact that the trust-search method, a very simple modification of the standard trust-region method, yields both properties recommends it strongly for MPEC decomposition methods by giving a clearer and more direct global convergence analysis than hitherto available.

Looking beyond the global convergence analysis of this paper, a goal of NLP and MPEC methods alike is fast local convergence. That is, this paper can be thought of as an intermediate step towards methods that combine fast local convergence with robust global convergence. This underlies our choice of a linear TR approach which has worthy global convergence properties, that we strengthen, and also holds the prospect of extension to hybrid NLP methods that achieve local superlinear convergence [6, 7, 10]. Unfortunately an investigation of superlinearly convergent MPEC methods is outside the scope of the paper due to the rigours of the NLP investigation. We do make some links to superlinear convergence of decomposition methods based on [32] following Condition 3.14 in §3.4. Likewise, nonlinear constraints and computational tests are important topics that are not addressed here.

1.2. Brief literature review. Trust-region methods are a well established class of algorithms for the solution of nonlinear nonconvex optimization problems. We refer the reader to the massive monograph of Conn, Gould and Toint [11], and the extensive bibliography therein, for a comprehensive guide to TR methods applied to both unconstrained and constrained problems. Our interest in such methods is particularly focussed on sequential linear programming trust-region techniques. This kind of approach has recently received some attention (see [6, 7, 10]) in the context of active-set SQP methods for large-scale nonlinear programming. In fact, the basic idea, first introduced in a paper of Fletcher and Sainz de la Maza [21], is to avoid solving an inequality constrained QP at each iteration by splitting the SQP iteration in two parts, the first one being devoted to the identification of the active set through a trust-region linear program, and the second one to the solution of an equality constrained QP to determine the new iterate. What seems to be lacking in the NLP framework is a deeper study of convergence of multipliers; see further comments after Theorem 2.11. On one hand, multiplier convergence is generally not needed to prove stationarity of accumulation points of the sequence of the iterates, even when multipliers are used in the practical implementation (see [9]), while on the other, whenever convergence of multipliers is needed for the algorithm, it is very common to estimate them by solving a supplementary least squares problem (see [6] and, for MPECs, [44]).

We mention some globally convergent methods for MPECs that are based on decomposition ideas, typically for problems where each complementarity constraint involves only two mappings ($m = 2$), starting with approaches that apply to linearly-constrained MPECs. Fukushima and Tseng [23] propose an ϵ-active set method for linearly-constrained MPECs that satisfy MPEC-LICQ globally; identification of active complementarity functions is achieved via a careful updating rule for ϵ. In each of the three methods of [28, 42, 45], one iteration requires exact solution, at least in principle, of a piece of the original, nonlinear MPEC. Jiang and Ralph [28] show global convergence of the PSQP decomposition method [32, 33] for quadratic programs with linear complementarity constraints; MPEC-LICQ is avoided by, if necessary, enumerating all pieces of the MPEC adjacent to the current iterate. Global convergence is also shown for the decomposition method of Scholtes [42], for nonlinear problems with more general constraints than complementarity constraints, and for the "extreme-point" decomposition method of Zhang and Liu [45]. Selection of MPEC pieces in [42] uses multipliers similar to the above and therefore relies on an extension of the MPEC-LICQ, whereas the latter condition is avoided in [45] by using enumeration over extreme rays in the selection process. In each of these three methods, implicit identification of active functions in complementarity constraints follows from the fact that a solution of one piece of the MPEC is found at each step.

For global convergence of nonlinearly-constrained MPECs, Scholtes and Stöhr [43] present a general trust-region framework for locally Lipschitz B-differentiable functions. For implementation, a penalty trust-region decomposition method, using dual selection criteria, is proposed and shown to be convergent, to a class of weak stationary points called C-stationary points, under MPEC-LICQ. Additional conditions, such as ULSC and asymptotic inactivity of the trust region constraint, are needed to obtain B-stationary points. A more comprehensive and penetrating analysis of penalty trust-region decomposition is given in Stöhr [44]. In particular, the TR radius may be re-set to a minimum threshold at the start of each iteration of the decomposition method, which is the essence of what we also do.

Beyond decomposition methods we only mention key words, and few related references, regarding some alternative approaches to MPECs. Smoothing [13, 22, 29, 30], regularization [31, 38, 41] and penalty methods [2, 25, 26, 38] all embed an MPEC in a family of "more regular" NLPs, indexed by a real parameter $\mu > 0$, and drive μ to zero to recover the original MPEC formulation and a solution of it. More recently, direct application of the sequential quadratic programming method, a standard NLP method, to MPECs has been justified both computationally [19] and theoretically [1, 20]. In addition there has been considerable work on theoretical and computational performance of interior-point methods that have been specially modified for solving MPECs [3, 12, 37]. Finally we mention the nonsmooth implicit programming formulation which seems natural when the MPEC is defined using lower-level equilibrium constraints that have a unique solution for each choice of the upper level variables. This formulation can be solved using bundle methods as explored in [35].

2. A trust-region method for linearly-constrained optimization.

We first present a basic trust-region method to find a stationary point of the following linearly-constrained optimization problem

$$\begin{array}{lll} & \min_{x\in\mathbb{R}^n} & f(x) \\ (2.1) & \text{subject to} & g(x) \leq 0 \\ & & h(x) = 0. \end{array}$$

As previously mentioned, we call the algorithm a trust-search method because it combines some defining features of TR and line-search methods. See further discussion to follow.

Given a feasible point $x \in \mathcal{F} \triangleq \{x \in \mathbb{R}^n \mid g(x) \leq 0, h(x) = 0\}$, we define a linear trust-region subproblem $LP(x, r)$, dependent on a trust-region radius $r > 0$, in the following way:

$$\begin{array}{lll} & \min_{d\in\mathbb{R}^n} & \nabla f(x)^\top d \\ & \text{subject to} & g(x+d) = g(x) + G^\top d \leq 0 \\ (2.2) & & h(x+d) = h(x) + H^\top d = 0 \\ & & \|d\| \leq r, \end{array}$$

where $\|\cdot\|$ is any appropriate norm. In fact a polyhedral norm turns out to be essential for computational efficiency because, after adding auxiliary variables to transform the constraint $\|d\| \leq r$ into finitely many affine inequalities, problem (2.2) is just a linear program. We make the standing assumption that holds throughout the paper:

The trust-region ball is specified with a fixed polyhedral norm.

Actually most of the results in this section hold for any norm; polyhedrality of the norm is only critical in showing convergence of multipliers, beginning with Proposition 2.10.

We denote by $d(x, r)$ an optimal solution of $LP(x, r)$, and by $v(x, r)$ the corresponding objective function value, i.e.,

$$v(x, r) \triangleq \nabla f(x)^\top d(x, r).$$

For notational simplicity we usually write $d(r)$ instead of $d(x, r)$, though we cannot take this liberty with $v(x, r)$ given the explicit role of x in the following results on the optimal value function. The importance of this kind of value function, in basic convergence analysis, has been stressed in [11, Chapter 12]. Notice also that if $x \in \mathcal{F}$ then, for any $r > 0$, $LP(x, r)$ is feasible and $v(x, r) \leq 0$, the equality being satisfied if and only if x is a stationary point of (2.1).

The common basis of trust-region methods, not unlike other descent methods, is the idea that the optimal value $v(x, r)$ of (2.2) approximates the variation of f in passing from x to $x + d(r)$. If $v(x, r)$ is, according to some criterion, a good approximation of $f(x + d(r)) - f(x)$ then we replace x by $x + d(r)$. If this is not the case, then the trust-region radius r will have to be appropriately reduced and the trust-region subproblem re-solved.

As we are about to introduce a simple trust-region scheme based on the above ideas, two main differences from typical TR methods deserve to be pointed out. The first one is quite formal. We choose not to keep track of the "null" steps during the algorithm execution; in other words we choose to update the current solution estimate only when a "serious" step $d(r) \neq 0$ is accepted. Thus, the core of our TR scheme is rather a "trust-search" subroutine, reminiscent of the standard line-search in many NLP methods [18], which includes the sequence of (null) steps where the current estimate of the solution is unchanged while the radius shrinks. The second difference is that, analogous to the Armijo line-search, at the start of each serious iteration we re-set the trust-region radius to a fixed positive value. This re-setting device is used for simplicity rather than efficiency or generality. What makes it critical to our convergence analysis is that it forces the algorithm to step sufficiently far from the current iterate to avoid convergence to a nonstationary point. Resetting the TR radius is hardly common in TR methods; although it has been used in [10, 24, 27], it does not appear to be critical to global convergence in these situations, e.g., see notes and references in [11, p.776]. By contrast, the TR method presented in [44], for the unconstrained minimization of a locally Lipschitz B-differentiable function, exploits this device to obtain strong global convergence results, improving those presented in [43].

We mention some alternatives to re-setting the TR radius to a fixed value $\rho > 0$ at the start of each serious iteration. The first is to take, at step k, the initial radius as $\rho_k = \max\{r_{k-1}, \rho\}$ where r_{k-1} is the TR radius accepted at the previous serious step; this idea has some of the flavor of the usual TR approach. The second is to take $\rho_k = \|\nabla f(x^k)\|$. This brings the trust-search method closer to the Armijo linesearch for unconstrained optimization for which the first trial iterate, with stepsize 1, is at distance $\|\nabla f(x^k)\|$ from the current iterate. In each of these cases, the initial radius ρ_k remains bounded away from 0 for all feasible iterates x^k in the neighborhood of any nonstationary point, which turns out to be key for the convergence proofs of this paper.

More generally, just as the Wolfe condition [18] is used in many line-searches to force expansion of the stepsize when appropriate, we could investigate trust-searches that are more subtle than the Armijo-style method proposed here. However we believe the current scheme captures significant, if not the main, benefits in global convergence analysis of trust-search methods, for both NLP and MPEC decomposition. Exploring other schemes, e.g., motivated by computational efficiency, is a subject for future work.

Now we introduce our trust-search scheme. Given a feasible point $x \in \mathcal{F}$, two scalars $\alpha, \beta \in (0,1)$, and a scalar $\rho > 0$, we generate a new point by taking the step $d(r)$ returned by the following trust-search subroutine $\mathbf{TS}(x)$:

[$\mathbf{TS}(x)$]

Set $r := \rho/\beta$;
Repeat
    Set $r := \beta r$;
    $d(r)$, $v(x,r) \leftarrow LP(x,r)$;
Until $f(x + d(r)) - f(x) \leq \alpha v(x,r)$.

We will refer to a direction d such that $\nabla f(x)^\top d < 0$ and $f(x+d) - f(x) \leq \alpha \nabla f(x)^\top d$ as to an *acceptable descent direction* at x.

The entire trust-search method can be described in the following way. Given a feasible starting point x^0, we generate a sequence $\{x^k\}$ of solution estimates for problem (2.1) by executing $\mathbf{TS}(x^k)$, for every $k = 0, 1, 2, \ldots$. Upon termination of $\mathbf{TS}(x^k)$, we are given $r^k = r$, $d^k = d(r^k)$, $v^k = v(x^k, r^k)$ and, unless x^k is stationary, we set $x^{k+1} := x^k + d^k$. A new execution of the trust-search subroutine is then started with respect to x^{k+1}.

ALGORITHM 2.1 (**TS-NLP**).

1. *Choose* $x^0 \in \mathcal{F}$, $\alpha, \beta \in (0,1)$, $\rho > 0$. *Set* $k := 0$.
2. v^k, $d^k \leftarrow \mathbf{TS(}x^k\mathbf{)}$.
3. *If* $v^k = 0$ *then STOP:* x^k *is stationary.*
   Else set $x^{k+1} := x^k + d^k$, $k := k+1$ *and return to 2.*

We repeat that whenever the trust-search subroutine is entered we update the initial trust-region radius to a fixed value $\rho > 0$, independently of the outcome of the previous trust-search, although the subsequent convergence analysis remains valid provided the initial trust-region radius is not less than a fixed positive tolerance. Notice also that the algorithm is monotone, since $f(x^{k+1}) \leq f(x^k)$ for every k, and maintains feasibility of iterates.

In the remainder of the section we will often deal with a (possibly arbitrary) sequence $\{x^k\} \subset \mathcal{F}$ for which we will denote by r^k, d^k, and v^k, respectively, the trust-region radius, the optimal solution, and the optimal value of problem $LP(x^k, r^k)$, returned on termination of $\mathbf{TS(}x^k\mathbf{)}$; moreover, we will denote by $(\lambda^k, \mu^k) \in \mathbb{R}^{m_g} \times \mathbb{R}^{m_h}$ the vector of KKT multipliers of $LP(x^k, r^k)$, i.e., the vectors of multipliers satisfying

$$0 \in \nabla f(x^k) + G\lambda^k + H\mu^k + N_{\mathcal{B}^k}(d^k) \tag{2.3}$$

and

$$\min\{\lambda^k, -g(x^k) - G^\top d^k\} = 0, \tag{2.4}$$

where $N_{\mathcal{B}^k}(d^k)$ is the normal cone [39] to the set $\mathcal{B}^k = \{d \in \mathbb{R}^n \mid \|d\| \leq r^k\}$ at the point d^k. To clarify the whole development we present in advance our two main convergence results, where a feasible starting x^0 is given.

- Convergence of iterates (Corollary 2.7 and Theorem 2.8)
  The linear trust-search method Algorithm 2.1 *has the nonstationary repulsion property (see Definition* 2.2*). Hence any accumulation point of the iteration sequence* $\{x^k\}$ *is stationary for the linearly-constrained NLP* (2.1)*.*

- Convergence of multipliers (Theorem 2.11)
  Let $\{x^k\}$ *be an iteration sequence as above that converges on a subsequence to the point* $\bar{x}$*. Then, there exists an associated subsequence of KKT multipliers for* $LP(x^k, r^k)$ *that is bounded, every accumulation point of which is a KKT multiplier for* (2.1) *at* $\bar{x}$*.*

The latter is especially interesting in that it appears not to have been recognized before.

2.1. Convergence of iterates. We aim to show that, under very common assumptions, any accumulation point of the sequence $\{x^k\}$ generated by Algorithm 2.1 is stationary for problem (2.1). However, we remark that convergence to a stationary point is a well known property for methods somewhat related to Algorithm 2.1, such as trust-region methods (see [11]) or projected-gradient methods (see [4, 15]).

Nevertheless we point out that the following results have wider implications. In fact the mechanism of the stationarity convergence proof, here used to show a robust convergence property, called Nonstationary Repulsion (a related idea is Monotone Uniform Descent, see [36]), of the algorithm, will be next used also to analyze convergence of multipliers.

DEFINITION 2.2 (See [15]). *An iterative feasible point algorithm for problem* (2.1) *has* nonstationary repulsion (NSR)*, if for each nonstationary point $\bar{x} \in \mathcal{F}$ there exists a neighborhood $\bar{\mathcal{U}}$ of $\bar{x}$ and $\bar{\epsilon} > 0$ such that if any iterate x^k lies in $\bar{\mathcal{U}} \cap \mathcal{F}$, then $f(x^{k+1}) < f(\bar{x}) - \bar{\epsilon}$.*

REMARK 2.3.

(i) The original version of NSR, [15, Definition 3.1], is stated without $\bar{\epsilon}$, i.e., take $\bar{\epsilon} = 0$ above, hence is slightly weaker. Assuming continuity of f, the advantage of introducing a positive $\bar{\epsilon}$ is that NSR then guarantees that any nonstationary point $\bar{x}$ has a neighborhood $\mathcal{U}' \subseteq \mathcal{U}$ such that $x^K \in \mathcal{U}$ implies $\{x^k\}_{k>K}$ does not intersect $\mathcal{U}'$. Hence, for a monotonic feasible point method with NSR, each nonstationary point has a neighborhood that can be intersected at most once by the sequence of iterates. This neighborhood is robust in the sense that it is independent of the (feasible) starting point of the algorithm.

(ii) NSR of gradient-based methods: On one hand, it appears that the standard TR method for linearly constrained NLP — which takes the TR radius at the start of a serious step as a bounded multiple of the TR radius produced by the previous step — does not satisfy NSR. On the other, many gradient-based methods satisfy NSR, for example the steepest descent method for unconstrained smooth minimization, with a linesearch or stepsize parameter that is chosen either (a) to be the largest stepsize in a geometrically decreasing sequence $1, \rho, \rho^2, \dots$ that satisfies the Armijo rule, or (b) to satisfy both the Armijo and Wolfe rules [18, Section 2.5], can easily be seen to have this property. The projected-gradient method is an extension of steepest descent to minimize a smooth function over a closed convex set. The projected-gradient method of Calamai and Moré [8], which chooses the path parameter at each iteration analogous to (b) above, is shown to satisfy NSR in [15, Proof of Theorem 3.6]. The analysis of [15] suggests that NSR also holds for the projected-gradient method that uses the stepsize analog of (a) as in [4, Chapter 2]. Moreover, we show below that the linear trust-search method satisfies NSR.

(iii) NSR of Newton-type methods: For smooth nonconvex unconstrained minimization, consider the notional algorithm in which Newton's method [18] is safeguarded by a steepest descent step as follows. If the Hessian at current iterate is positive definite with condition number bounded above by a constant, let the search direction be the Newton direction, otherwise the steepest descent direction; then choose the stepsize by procedure (a) or (b) as above. Standard theory on the Newton step [4] when the Hessian is positive definite means that NSR of steepest descent implies NSR of this safeguarded Newton method. Moreover superlinear or quadratic convergence follows if at a limit point the Hessian is positive definite with a sufficiently small condition number. A subject for future investigation is how to emulate this, in a computationally efficient way, in the case of linearly constrained NLP.

First we recall some useful properties of the value function $v(x, r)$. More details can be found in [11], which focusses on Euclidean trust regions, where a function

$$\chi(x,r) \stackrel{\triangle}{=} |v(x,r)|$$

has been used as a criticality measure applied to convex constrained optimization problems. Results in [11] show that each point on the projected-gradient path is a minimizer of the Euclidean TR subproblem with radius set to be the length of the projected-gradient, so that the criticality measure can also be interpreted as a measure of the length of the projected-gradient step. The term "criticality measure" is often referred to, in similar contexts, as a "residual measure" or an "error bound" (see [14]), all of these terms referring to measures of nonstationarity. We further note that TR methods and projected-gradient methods are distinguished, first, by the fact their relationship is only close when the TR method uses the Euclidean ball and, second, by lack of uniqueness of TR solutions — sometimes even in the Euclidean case — whereas the projected-gradient path is uniquely defined.

We start our convergence analysis by repeating an obvious fact: given $x \in \mathcal{F}$ and any $r > 0$, $v(x,r) = 0$ if and only if x is stationary for (2.1). Thus finite termination of the trust-search method implies that a stationary point has been found.

Now we state a result entailed in [11, Theorem 12.1.5.(ii)].

LEMMA 2.4. *Let $x \in \mathcal{F}$. Then for any $0 < r_1 < r_2$ it holds that*

$$\frac{v(x,r_1)}{r_1} \le \frac{v(x,r_2)}{r_2} \ .$$

Proof. Notice that $x + \frac{r_1}{r_2} d(r_2) \in \mathcal{F}$ for any $0 < r_1 < r_2$, and

$$\left\| \frac{r_1}{r_2} d(r_2) \right\| = \frac{r_1}{r_2} \|d(r_2)\| \le r_1.$$

Therefore $\frac{r_1}{r_2} d(r_2)$ is feasible in $LP(x,r_1)$, and as a consequence we have that

$$\frac{v(x,r_1)}{r_1} \le \frac{1}{r_1} \nabla f(x)^\top \left(\frac{r_1}{r_2} d(r_2) \right) = \frac{v(x,r_2)}{r_2},$$

which proves the thesis. $\square$

LEMMA 2.5. *The function $v(x,r)$ is continuous on the set $\mathcal{F} \times \mathbb{R}_+$.*

Proof. The statement is a consequence of the continuity of the solution value of convex optimization problems (see [11, Theorem 3.2.8] and [16]). $\square$

To deal with asymptotic convergence of the algorithm we first need to state the following result about finite termination of the trust-search subroutine when applied to a nonstationary point.

PROPOSITION 2.6. *For any bounded subset X of $\mathcal{F}$ and $\gamma > 0$, there exists a scalar $r_\gamma \in (0,\rho)$ with the following property. If $x \in X$ and $\hat{r} \in (0, r_\gamma]$ satisfy $\frac{v(x,\hat{r})}{\hat{r}} \le -\gamma$, then $\mathbf{TS}(x)$ terminates returning a trust-region radius $r_x > \beta\hat{r}$.*

Proof. By uniform continuity of ∇f on bounded sets, there exists $r_\gamma \in (0,\rho)$ such that

$$\sup\left\{ \|\nabla f(x+d) - \nabla f(x)\| : x \in X, \|d\| \le r_\gamma \right\} \le (1-\alpha)\gamma \ .$$

Choose any $x \in X$ and $\hat{r} \in (0, r_\gamma]$ such that $\frac{v(x,\hat{r})}{\hat{r}} \leq -\gamma$. Then for any $r \in (0, \hat{r}]$, Lemma 2.4 gives $\frac{v(x,r)}{r} \leq -\gamma$. For such r let $d(r)$ be an optimal solution of $LP(x, r)$. Then for some $s \in (0, 1)$ we have

$$\begin{aligned} f(x + d(r)) - f(x) \quad &= \nabla f(x + sd(r))^\top d(r) \\ &\leq \nabla f(x)^\top d(r) + \|\nabla f(x + sd(r)) - \nabla f(x)\| \cdot \|d(r)\| \\ &\leq \nabla f(x)^\top d(r) + (1 - \alpha)\gamma r \\ &\leq \nabla f(x)^\top d(r) - (1 - \alpha)\nabla f(x)^\top d(r) \\ &= \alpha v(x, r) \end{aligned}$$

i.e., $d(r)$ is an acceptable feasible descent direction at x. Thus, a sufficient condition for $\mathbf{TS}(x)$ to terminate is that $r \leq \hat{r}$.

Now consider the scalar sequence

$$\rho, \beta\rho, \beta^2\rho, \ldots$$

and suppose that $j \in \mathbb{N}$ is such that $\beta^j\rho > \beta\hat{r}$ and $\beta^{j+1}\rho \leq \beta\hat{r}$. From the latter inequality we obtain $\beta^j\rho \leq \hat{r}$, and hence $\beta^j\rho \in (\beta\hat{r}, \hat{r}]$. This implies that the descent direction $d(\beta^j\rho)$ satisfies the termination condition of the trust-search procedure, and therefore that $\mathbf{TS}(x)$ terminates returning a trust-region radius $r_x \geq \beta^j\rho > \beta\hat{r}$. ∎

Now consider a nonstationary point $\bar{x} \in \mathcal{F}$, hence $v(\bar{x}, \rho) < 0$. Proposition 2.6 gives finite termination of the trust-search subroutine at $\bar{x}$. In fact we can say something stronger:

Corollary 2.7. *Algorithm* 2.1*,* **TS-NLP***, is a monotonic feasible-point method with NSR.*

Proof. Let $\bar{x} \in \mathcal{F}$ be a non-stationary point for problem (2.1). Take $0 < \bar{\gamma} < -v(\bar{x}, \rho)/\rho$, and use continuity of the value function (Lemma 2.5) to obtain a bounded neighborhood $\mathcal{U}$ of $\bar{x}$ such that $\bar{\gamma} \leq -v(x, \rho)/\rho$ for $x \in \mathcal{U}$. Proposition 2.6 then gives $\hat{r} > 0$ such that, for $\bar{r} := \beta\hat{r}$ and any $x \in \mathcal{U} \cap \mathcal{F}$, $\mathbf{TS}(x)$ terminates returning a direction d_x and a trust-region radius $r_x > \bar{r}$. Notice, from Lemma 2.4, that

$$v(x, r_x) \leq r_x \frac{v(x, \rho)}{\rho} < -\bar{r}\bar{\gamma}.$$

Hence, the terminating property of the trust-search routine implies that $f(x + d_x) < f(x) - \alpha\bar{r}\bar{\gamma}$.

Let $\bar{\epsilon} = (\alpha\bar{r}\bar{\gamma})/2$ and take a subset $\bar{\mathcal{U}} \subseteq \mathcal{U}$ such that $|f(x) - f(\bar{x})| < \bar{\epsilon}$ for every $x \in \bar{\mathcal{U}}$. Nonstationary repulsion of Algorithm 2.1 easily follows since if the k-th iterate $x^k \in \mathcal{F}$ lies in $\bar{\mathcal{U}}$ then $f(x^{k+1}) < f(x^k) - 2\bar{\epsilon} < f(\bar{x}) - \bar{\epsilon}$. ∎

Given Corollary 2.7 and part (i) of the Remark 2.3, the next result is immediate, c.f. [15, Theorem 3.2].

Theorem 2.8. *Any limit point of the sequence of iterates generated by Algorithm 2.1 is stationary for* (2.1)*.*

It seems unlikely that NSR of TR methods will hold without allowing the method to re-set or otherwise expand the TR radius at the start of each serious iteration. Certainly the proof of Proposition 2.6, from which the remaining results of the section directly follow, makes explicit use of fact that the starting TR radius cannot be less than ρ. This Proposition is also a key ingredient in the proof of multiplier convergence; see comments after Theorem 2.11.

2.2. Convergence of multipliers. The main novelty in section 2 occurs here, in showing that multipliers for trust-region methods converge without additional assumptions, see Theorem 2.11. To this aim we need to introduce a preliminary result, Proposition 2.10, regarding any bounded sequence $\{x^k\}$ of feasible points for (2.1), and any bounded sequence $\{r^k\}$ of positive scalars not necessarily produced by application of $\mathbf{TS}(x^k)$. We recall that d^k, v^k and (λ^k, μ^k), respectively, denote the optimal solution, optimal value and corresponding KKT multipliers, see (2.3) and (2.4), of $LP(x^k, r^k)$. First we present the following characterization of $N_{\mathcal{B}^k}(d^k)$.

LEMMA 2.9. *Let $d^k \in \mathcal{B}^k$, with $\|d^k\| = r^k$. A vector $\tilde{d}$ is normal to the set $\mathcal{B}^k$ at the point d^k if and only if there exist a non-negative scalar η and a dual unit-norm vector $d^* \in \partial\|d^k\|$ such that $\tilde{d} = \eta d^*$ and $\langle d^k, d^*\rangle = \|d^k\|$.*

Proof. We recall from convex analysis theory that a vector $\tilde{d} \in \mathbb{R}^n$ is normal to the set $\mathcal{B}^k$ at the point d^k if and only if there exist a non-negative scalar η, and a vector $d^* \in \partial\|d^k\|$, such that $\tilde{d} = \eta d^*$ (see [39, Corollary 23.7.1]). The thesis follows by noting that, for any $d^k \neq 0$ it is possible to express $\partial\|d^k\|$ in the following way, see [5, Example 2.129, Eq. (2.250)],

$$\partial\|d^k\| = \{d^* \mid \|d^*\|_D = 1, \langle d^k, d^*\rangle = \|d^k\|\}$$

where $\|d^*\|_D$ is the dual norm of d^*. $\square$

Now we present a sufficient condition for any accumulation point of $\{(\lambda^k, \mu^k)\}$ to be a KKT multiplier vector for problem (2.1). For later convenience we state the next result over subsequences, though it equally applies if the entire sequence converges.

PROPOSITION 2.10. *Suppose that $\{x^k\} \subset \mathcal{F}$ converges on a subsequence $\mathcal{K}$ to the point $\bar{x}$. If $\{r^k\}$ is any sequence of positive scalars such that the corresponding optimal values v^k of $LP(x^k, r^k)$ satisfy*

$$\lim_{\substack{k \to \infty \\ k \in \mathcal{K}}} \frac{v^k}{r^k} = 0, \tag{2.5}$$

then $\bar{x}$ is stationary, there exists an associated multiplier subsequence $\{(\lambda^k, \mu^k)\}_{k\in\mathcal{K}}$ that is bounded, and every accumulation point of the latter is a KKT multiplier for (2.1) at $\bar{x}$.

Proof. By Lemma 2.9 there exists a scalar $\eta_k \geq 0$ and a dual unit vector d_k^* such that the condition (2.3) can be expressed in the following way

$$0 = \nabla f(x^k) + G\lambda^k + H\mu^k + \eta_k d_k^*. \tag{2.6}$$

For the moment we assume that $\|d^k\| = r^k$ on a subsequence of $\{d^k\}_{k\in\mathcal{K}}$; after giving the proof for this case we will see that the proof for the remaining case follows without further effort. By taking a subsequence if necessary, we assume without loss of generality that $\|d^k\| = r^k$ for each $k \in \mathcal{K}$ to obtain

$$0 = v^k + (\lambda^k)^\top G^\top d^k + (\mu^k)^\top H^\top d^k + \eta_k r^k.$$

Now observe, as a consequence of (2.4), that for each i such that $\lambda_i^k > 0$ there holds $g_i(x^k) + G_i^\top d^k = 0$, that is $G_i^\top d^k = -g_i(x^k) \geq 0$, where $G_i^\top$ is the i-th row of $G^\top$. Moreover, feasibility of x^k implies that $H^\top d^k = 0$. Hence we have

$$0 = v^k + (\lambda^k)^\top G^\top d^k + \eta_k r^k = v^k - \sum_{i|\lambda_i^k>0} \lambda_i^k g_i(x^k) + \eta_k r^k \ ,$$

and dividing by the positive scalar r^k

$$0 = \frac{v^k}{r^k} - \frac{\sum_{i|\lambda_i^k>0} \lambda_i^k g_i(x^k)}{r^k} + \eta_k \ .$$

Since $0 \leq -\sum_{i|\lambda_i^k>0} \lambda_i^k g_i(x^k)$, it holds that

$$0 \geq \frac{v^k}{r^k} + \eta_k \ .$$

The last inequality implies that $\{\eta_k\}_{k\in\mathcal{K}} \longrightarrow 0$, as we are assuming that $\{\frac{v^k}{r^k}\}_{k\in\mathcal{K}} \longrightarrow 0$, and therefore that $\{\eta_k d_k^*\}_{k\in\mathcal{K}} \longrightarrow 0$. But this means that

$$\{\nabla f(x^k) + \eta_k d_k^*\}_{k\in\mathcal{K}} \longrightarrow \nabla f(\bar{x}) \ .$$

Since the TR ball is constructed with the same polyhedral norm for all iterations, the set of all (sub)gradients of the constraints of $LP(x^k, r^k)$ has finite cardinality and is independent of k. By considering maximal linearly independent subsets of these gradients, it is therefore easy to show that we may choose $\{(\lambda_k, \mu_k)\}_{k\in\mathcal{K}}$ to be bounded (similar to [32, Proposition 1.3.8]). Therefore there exists an accumulation point, say $(\bar{\lambda}, \bar{\mu})$, such that

$$\{\nabla f(x^k) + G\lambda^k + H\mu^k + \eta_k d_k^*\}_{k\in\mathcal{K}'\subset\mathcal{K}} \longrightarrow \nabla f(\bar{x}) + G\bar{\lambda} + H\bar{\mu} \ .$$

Then, as a consequence of (2.6), we have that

$$\nabla f(\bar{x}) + G\bar{\lambda} + H\bar{\mu} = 0 \ . \tag{2.7}$$

Let $\bar{d}$ be an accumulation point of the bounded sequence $\{d^k\}_{k\in\mathcal{K}}$. Since x^k is a feasible point of (2.1) for any k, we have

$$g(x^k + d^k) \leq 0 \ \ \forall k \in \mathcal{K}$$

that is

$$g(x^k) + G^\top d^k \leq 0 \ \ \forall k \in \mathcal{K}.$$

Taking the limit on a subsequence of $\mathcal{K}$ we obtain the limit expression of (2.4)

$$\min\{\bar{\lambda}, -g(\bar{x} + \bar{d})\} = 0. \tag{2.8}$$

Since by (2.7) we have $G\bar{\lambda} = -\nabla f(\bar{x}) - H\bar{\mu}$, we can express the complementarity condition (2.8) in the following way:

$$0 = \bar{\lambda}^\top g(\bar{x} + \bar{d}) = \bar{\lambda}^\top g(\bar{x}) + \bar{\lambda}^\top G^\top \bar{d} = \bar{\lambda}^\top g(\bar{x}) - \nabla f(\bar{x})^\top \bar{d} - \bar{\mu}^\top H^\top \bar{d} \ ,$$

where $\bar{\lambda} \geq 0$. But from the assumption $\lim_{k\in\mathcal{K}} \frac{v^k}{r^k} = 0$ it follows that $\nabla f(\bar{x})^\top \bar{d} = 0$, and by feasibility of $\bar{x}$ it holds that $H^\top \bar{d} = 0$ and $g(\bar{x}) \leq 0$. Therefore we have that

$$\min\{\bar{\lambda}, -g(\bar{x})\} = 0 \tag{2.9}$$

which along with (2.7) proves the thesis.

It remains to show the result when $\|d^k\| < r^k$ for sufficiently large $k \in \mathcal{K}$. In this case we may take $\eta^k = 0$ and therefore obtain (2.8) and the required proof as a byproduct of the previous analysis. □

We remark that whether or not the multipliers for (2.1) are bounded, if the simplex method, or any active set method, is applied to each $LP(x^k, r^k)$, then boundedness of multipliers follows as described in the above proof. Next we come to the main result of this section.

THEOREM 2.11. *Let $\{x^k\}$ be a sequence, generated by the linear trust-search method Algorithm* 2.1, *that converges on a subsequence $\mathcal{K}$ to the point $\bar{x}$. Then, there exists an associated subsequence of KKT multipliers $\{(\lambda^k, \mu^k)\}_{k\in\mathcal{K}}$ for $LP(x^k, r^k)$ that is bounded, every accumulation point of which is a KKT multiplier for* (2.1) *at $\bar{x}$.*

Proof. First observe, since $\bar{x}$ is an accumulation point of the sequence $\{x^k\}$ generated by the algorithm, that $\bar{x}$ is also a stationary point for (2.1) from Theorem 2.8. This implies, for any subsequence $\{x^k\}_{k\in\mathcal{K}}$ converging to $\bar{x}$, that

$$\lim_{\substack{k \to \infty \\ k \in \mathcal{K}}} v^k = 0.$$

Now suppose that for every k the trust-region radius r^k is bounded away from zero. Then it follows that

$$\lim_{\substack{k \to \infty \\ k \in \mathcal{K}}} \frac{v^k}{r^k} = 0,$$

and the thesis is an immediate consequence of Proposition 2.10. We thus assume, without loss of generality, that there exists a subsequence $\{r^k\}_{k\in\mathcal{K}'\subset\mathcal{K}}$ converging to zero. Then, it is only required to prove that

$$\lim_{\substack{k \to \infty \\ k \in \mathcal{K}'}} \frac{v^k}{r^k} = 0,$$

and the thesis will follow in the same way. Suppose for a contradiction that

$$\{r^k\}_{k\in\mathcal{K}'\subset\mathcal{K}} \to 0$$

and there exist $\bar{\gamma}$ and $\bar{k}$ such that for every $k \geq \bar{k}$, with $k \in \mathcal{K}'$, we have

$$\frac{v^k}{r^k} \leq -\frac{\bar{\gamma}}{\beta} \; .$$

Now observe for all k, since $\frac{r^k}{\beta} > r^k$, that

$$v(x^k, \frac{r^k}{\beta}) \leq v^k \; .$$

Hence, dividing by $\frac{r^k}{\beta}$, for every $k \geq \bar{k}$, with $k \in \mathcal{K}'$, we have that

$$\frac{v(x^k, \dfrac{r^k}{\beta})}{\frac{r^k}{\beta}} \leq \beta \frac{v^k}{r^k} \leq -\bar{\gamma} \; .$$

Now apply Proposition 2.6 by taking $\gamma = \bar{\gamma}$ and $X = \{x \mid \|x - \bar{x}\| \leq 1\} \cap \mathcal{F}$, and obtain $r_\gamma \in (0, \rho)$ satisfying the property therein. Since $\{r^k\}_{k \in \mathcal{K}'} \to 0$ and $\{x^k\}_{k \in \mathcal{K}'} \to \bar{x}$, then for sufficiently large $k \geq \bar{k}$ with $k \in \mathcal{K}'$ we have $\frac{r^k}{\beta} \leq r_\gamma$ and $x^k \in X$. Thus by letting $\hat{r} := \frac{r^k}{\beta}$, from Proposition 2.6 it follows that $\mathbf{TS}(x^k)$ terminates returning r such that

$$r > \beta \hat{r} = r^k$$

which contradicts the choice of r^k. $\square$

It is worth noting that [21] gives a linear trust-region method relating to an ℓ_1 exact penalty function approach for NLPs with nonlinear constraints, and presents convergence results for both the iteration sequence and the associated sequence of multipliers. The extent to which we can use this analysis still needs to be explored, for example in [21] it is not clear how to deal with asymptotically vanishing subsequences of TR radii in the case when the TR constraint remains active in the limit. Note that the most delicate part of the proof of Theorem 2.11 deals with precisely this case and, in doing so, relies on Proposition 2.6 and hence on having a lower bound on the TR radius at the start of each serious iteration, as distinct from the more traditional initialization of the TR radius used in [21].

An associated multiplier convergence result for the projected-gradient method can be derived from [17]. The latter examines the KKT multipliers at the optimal solution of a quadratic program which is related to (2.2) in that a quadratic term in the objective function replaces trust-region constraints:

$$\begin{array}{ll} \displaystyle\min_{d \in \mathbb{R}^n} & \nabla f(x)^\top d \, + \, \frac{1}{2} \, d^\top d \\ \text{subject to} & g(x+d) = g(x) + G^\top d \leq 0 \\ & h(x+d) = h(x) + H^\top d = 0 \; . \end{array}$$

Theorem 3.5 of [17] bounds the distance between the KKT multipliers of this quadratic program and the multiplier set at a nearby stationary point $\bar{x}$ in terms of distance $\|x - \bar{x}\|$. As the KKT points of this problem coincide with those of the projected-gradient QP,

$$\begin{array}{ll} \displaystyle\min_{d \in \mathbb{R}^n} & \frac{1}{2} \|d + \nabla f(x)\|^2 \\ \text{subject to} & g(x+d) \leq 0 \\ & h(x+d) = 0 \; , \end{array}$$

convergence of projected-gradient multipliers follows.

3. A decomposition framework for mathematical programs with linear complementarity constraints. We now introduce a decomposition method to locate a B-stationary point of the MPEC (1.1), see §3.2. The method makes use of a suitable version of our trust-search subroutine. In fact we aim at exploiting convergence properties of our trust-search scheme when it is embedded into a local decomposition method. In §3.3 we will provide a global convergence analysis of the method. In particular, we will understand that the two main features of our trust-region scheme, NSR and convergence of multipliers, turn to be fundamental to guarantee robust convergence properties of the method. Then, in §3.4, a deeper understanding of such properties will allow us to gather a convergence proof for a more general algorithmic scheme satisfying some critical assumptions.

3.1. First-order stationarity conditions. To ease understanding of the following subjects, we briefly review some definitions and results from MPEC theory (see [40, 42]) that hold in the setting of general nonlinear constraints, while a detailed description of the method is deferred to §3.2.

Definition 3.1. *Let $\bar{x}$ be a feasible point of* (1.1). Linear independence constraint qualification (MPEC-LICQ) *is satisfied at $\bar{x}$ if the gradients*

$$\begin{array}{ll} \nabla p_{ij}(\bar{x}) & \forall (i,j) : p_{ij}(\bar{x}) = 0 \ , \\ \nabla g_i(\bar{x}) & \forall i : g_i(\bar{x}) = 0 \ , \\ \nabla h_i(\bar{x}) & \forall i \end{array}$$

are linearly independent.

Given a feasible point $\bar{x}$ of (1.1), for every $1 \leq i \leq m_p$ we denote by

$$A_i(\bar{x}) \stackrel{\triangle}{=} \{j : p_{ij}(\bar{x}) = 0\}$$

the set of active functions in the i-th complementarity constraint at $\bar{x}$. Let $|A_i(\bar{x})|$ denote the cardinality of $A_i(\bar{x})$, and let

$$M(\bar{x}) \stackrel{\triangle}{=} \{i : |A_i(\bar{x})| \geq 2\}$$

be the set of multi-active constraints, that is the set of complementarity constraints with more than one active function at $\bar{x}$.

Definition 3.2. *If $|A_i(\bar{x})| = 1$ for every $i = 1, \ldots, m_p$, or equivalently if $M(\bar{x}) = \emptyset$, then we say that* lower level strict complementarity (LLSC) *holds at $\bar{x}$.*

Definition 3.3. *If $\bar{x}$ is a feasible point of* (1.1) *and $\nabla f(\bar{x})^\top d \geq 0$ for every $d \in \mathbb{R}^n$ such that*

$$\begin{array}{rcll} \min\{\nabla p_{ij}(\bar{x})^\top d \mid j \in A_i(\bar{x})\} & = & 0 & \forall i \\ \nabla g_i(\bar{x})^\top d & \leq & 0 & \forall i : g_i(\bar{x}) = 0 \\ \nabla h_i(\bar{x})^\top d & = & 0 & \forall i, \end{array}$$

then $\bar{x}$ is a B-stationary *point of* (1.1).

An illuminating characterization of B-stationarity can be obtained in the following way, [40]. Let $\bar{x}$ and I be, respectively, a feasible point of (1.1) and a piece index for (1.1) at $\bar{x}$, i.e., $I \in \mathcal{I}(\bar{x})$, see (1.2). Consider the ordinary nonlinear program NLP_I

(1.3). Then $\bar{x}$ is B-stationary for (1.1) if and only if $\bar{x}$ is a stationary point for NLP_I, for every $I \in \mathcal{I}(\bar{x})$. This result displays the combinatorial character of an MPEC; in fact, in order to verify B-stationarity of a feasible point one may need to verify the consistency of a very large number of inequality systems, unless LLSC holds at $\bar{x}$ (in such a case $\mathcal{I}(\bar{x})$ is made up of a single set). More importantly such a characterization is the key idea for developing local decomposition methods for MPECs [32].

For a better comprehension of decomposition methods we definitely need to provide some results regarding dual stationarity conditions. First notice that $MPEC$ and NLP_I, for every $I \in \mathcal{I}(\bar{x})$, have the same Lagrangian function [40]

$$\mathcal{L}(x, \lambda, \mu, \xi) = f(x) + g(x)^\top \lambda + h(x)^\top \mu - \sum_i \sum_j p_{ij}(x)\xi_{ij} \ .$$

Definition 3.4. *If $\bar{x}$ is a feasible point of* (1.1) *and there exists a multiplier tuple $\bar{\pi} = (\bar{\lambda}, \bar{\mu}, \bar{\xi})$ satisfying*

$$\begin{array}{rcll} \nabla_x \mathcal{L}(\bar{x}, \bar{\lambda}, \bar{\mu}, \bar{\xi}) & = & 0 & \\ p_{ij}(\bar{x})\bar{\xi}_{ij} & = & 0 & \forall (i,j) \\ \bar{\lambda} & \geq & 0 & \\ g(\bar{x})^\top \bar{\lambda} & = & 0 & \end{array} \tag{3.1}$$

then $\bar{x}$ is called weakly stationary *or* critical *for* (1.1) *with multiplier $\bar{\pi}$.*

Definition 3.5. *Let $\bar{x}$ be weakly stationary with multiplier $\bar{\pi} = (\bar{\lambda}, \bar{\mu}, \bar{\xi})$. We say that* upper level strict complementarity (ULSC) *holds at $\bar{x}$ if $\bar{\xi}_{ij} \neq 0$, for every (i, j) such that $i \in M(\bar{x})$ and $j \in A_i(\bar{x})$.*

Definition 3.6. *Let $\bar{x}$ be weakly stationary with multiplier $\bar{\pi} = (\bar{\lambda}, \bar{\mu}, \bar{\xi})$.*
(i) *$\bar{x}$ is called* M-stationary *if for each $i \in M(\bar{x})$, the existence of $\hat{\jmath} \in A_i(\bar{x})$ with $\bar{\xi}_{i\hat{\jmath}} < 0$ implies $\bar{\xi}_{ij} = 0$ for the remaining indices $j \in A_i(\bar{x}) \setminus \{\hat{\jmath}\}$;*
(ii) *$\bar{x}$ is called* strongly stationary *if $\bar{\xi}$ satisfies the sign conditions*

$$\bar{\xi}_{ij} \geq 0 \quad \forall (i,j) : i \in M(\bar{x}) \text{ and } j \in A_i(\bar{x}). \tag{3.2}$$

In the context of decomposition methods, it is also worth formalizing the idea of "one-sided" stationarity [32]:

Definition 3.7. *Let $\bar{x}$ be feasible for* (1.1). *We say it is a* one-piece *or* one-sided *stationary point if there exists an adjacent piece of the MPEC for which it is stationary.*

Without any difficulty, the various stationarity conditions can be related as

$$\text{strongly stationary} \Rightarrow \left\{ \begin{array}{c} \text{M-stationary} \\ \text{B-stationary} \end{array} \right\} \Rightarrow \text{one-piece stationary} \Rightarrow \text{weakly stationary}$$

Checking strong stationarity of a weakly stationary point is trivial, simply look at the sign of multipliers corresponding to multi-active complementarity constraints. The following theorem, the idea of which appears in [33], but whose form we quote from [40], makes clear the importance of this remark.

THEOREM 3.8 (See [40, Theorem 4]). *Let $\bar{x}$ be a feasible point of* (1.1)*, where MPEC-LICQ holds. Then, $\bar{x}$ is B-stationary if and only if it is strongly stationary.*

Finally we recall from [40] that if $\bar{x}$ is a weakly but not strongly stationary point with multiplier $\bar{\pi} = (\bar{\lambda}, \bar{\mu}, \bar{\xi})$, and MPEC-LICQ holds at $\bar{x}$, then any negative multiplier corresponding to a multi-active constraint function provides a piece index $I \in \mathcal{I}(\bar{x})$ with the critical property that $\bar{x}$ is not stationary for NLP_I. To explain, suppose $\bar{\xi}_{ij} < 0$ for some $i \in M(\bar{x})$ and $j \in A_i(\bar{x})$; then, as there exists $j' \in A_i(\bar{x}) \setminus \{j\}$, there also exists $I \in \mathcal{I}(\bar{x})$ such that $(i,j) \notin I$. Since NLP_I includes the inequality constraint $p_{ij}(x) \geq 0$, the corresponding KKT multiplier at any stationary point would have to be nonnegative. Now the usual LICQ holds for NLP_I at $\bar{x}$, hence the Lagrangian conditions for NLP_I at $\bar{x}$ are uniquely satisfied by the multiplier vector $\bar{\pi}$; given negativity of $\bar{\xi}_{ij}$, the point $\bar{x}$ therefore cannot be stationary for NLP_I. Formally:

PROPOSITION 3.9. *Let $\bar{x}$ be a weakly stationary point of the MPEC, with multiplier $\bar{\pi} = (\bar{\lambda}, \bar{\mu}, \bar{\xi})$, at which the MPEC-LICQ holds. If $\bar{x}$ is not B-stationary, then there is $I \in \mathcal{I}(x)$ and $(i,j) \notin I$ such that $\bar{\xi}_{ij} < 0$; in particular, $\bar{x}$ is not stationary for NLP_I.*

This result drives the decomposition approach described below.

3.2. A trust-search decomposition method for mathematical programs with linear complementarity constraints. Now the key ideas to devise a decomposition method for MPECs can be easily understood. Bearing in mind that we can look for strongly stationary points if the common MPEC-LICQ assumption holds, we generate a sequence of solution estimates of (1.1) by exploiting the local formulation of an MPEC, as an NLP branch, coupled with strong stationarity conditions. Indeed, we recall that given a feasible point $\bar{x}$ of (1.1) and a piece index $I \in \mathcal{I}(\bar{x})$, the feasible region of NLP_I near $\bar{x}$ is a subset of the feasible region of (1.1). Therefore we can exploit a local model of NLP_I to locate a new solution estimate $\bar{x}+d$. Then, checking strong stationarity at $\bar{x}+d$, that is checking the sign of multipliers, gives us useful information to define a new model at $\bar{x}+d$, more specifically to select a new set $I \in \mathcal{I}(\bar{x}+d)$, which is expected to have promising descent properties at $\bar{x}+d$. This is essentially the viewpoint of [43, 44].

Our decomposition method for problem (1.1) is in fact based on the above observations, where, given a feasible point x of (1.1), and a piece index $I \in \mathcal{I}(x)$, we generate a new point $x+d$ by applying our trust-search scheme as if we were to solve NLP_I.

Let us extend the notation regarding the trust-search approach to decomposition methods for MPECs. Let x be a feasible point of (1.1), and let $I \in \mathcal{I}(x)$. We define a linear trust-region subproblem $LP(x, I, r)$, where $r > 0$, in the following way:

$$
\begin{array}{llll}
\min\limits_{d \in \mathbb{R}^n} & \nabla f(x)^\top d & & \\
\text{subject to} & p_{ij}(x) + P_{ij}^\top d & = 0 & \forall (i,j) \in I \\
& p_{ij}(x) + P_{ij}^\top d & \geq 0 & \forall (i,j) \notin I \\
& g(x) + G^\top d & \leq 0 & \\
& h(x) + H^\top d & = 0 & \\
& \|d\| & \leq r. &
\end{array} \tag{3.3}
$$

We denote by $d(r)$ the optimal solution of $LP(x, I, r)$, and by $v(x, I, r)$ its corresponding (optimal) value. Moreover we denote by $\lambda(r) \in \mathbb{R}^{m_g}$, $\mu(r) \in \mathbb{R}^{m_h}$, and $\xi(r) \in \mathbb{R}^{m_p} \times \mathbb{R}^\ell$ the vectors of KKT multipliers at the solution of $LP(x, I, r)$, i.e., the vectors of multipliers satisfying

$$0 \in \nabla f(x) + G\lambda(r) + H\mu(r) - \sum_i \sum_j P_{ij}\xi_{ij}(r) + N_{\mathcal{B}(r)}(d(r)) \tag{3.4}$$

and

$$\min\{\lambda(r), -g(x + d(r))\} = 0, \tag{3.5}$$

and

$$\min\{\xi_{ij}(r), p_{ij}(x + d(r))\} = 0 \quad \forall (i,j) \notin I, \tag{3.6}$$

where $N_{\mathcal{B}(r)}(d(r))$ is the normal cone to the set $\mathcal{B}(r) = \{d \in \mathbb{R}^n \mid \|d\| \leq r\}$ at the point $d(r)$.

Finally we can introduce our trust-search decomposition scheme. Given a feasible point x, a piece index $I \in \mathcal{I}(x)$, two scalars $\alpha, \beta \in (0, 1)$, and a scalar $\rho > 0$, we generate a new point by taking the step $d(r)$ returned from the following trust-search subroutine $\mathbf{TS}(x, I)$:

[$\mathbf{TS}(x, I)$]
    Set $r := \rho/\beta$;

    Repeat
        Set $r := \beta r$;
        $d(r)$, $v(x, I, r) \leftarrow LP(x, I, r)$;
    Until $f(x + d(r)) - f(x) \leq \alpha v(x, I, r)$.

Hence, given a feasible starting point x^0, and a starting piece index $I^0 \in \mathcal{I}(x^0)$ we generate a sequence $\{x^k\}$ of solution estimates for problem (1.1) by executing $\mathbf{TS}(x^k, I^k)$ for every $k = 0, 1, 2, \ldots$. Upon termination of $\mathbf{TS}(x^k, I^k)$ we are given r^k, $d^k := d(r^k)$ and $\pi^k = (\lambda^k, \mu^k, \xi^k) := (\lambda(r^k), \mu(r^k), \xi(r^k))$; and, unless x^k is stationary, we set $x^{k+1} := x^k + d^k$. Furthermore we use ξ^k in selecting a new piece index $I^{k+1} \in \mathcal{I}(x^{k+1})$, and a new execution of the trust-search subroutine is started with respect to x^{k+1} and I^{k+1}. The entire decomposition trust-search method can be described in the following way:

Algorithm 3.10 (**TS-MPEC**).

1. *Choose* $x^0 \in \mathcal{F}$, $I^0 \in \mathcal{I}(x^0)$, $\alpha, \beta \in (0,1)$, $\rho > 0$. *Set* $k := 0$.

2. d^k, $\pi^k = (\lambda^k, \mu^k, \xi^k) \leftarrow$ **TS(**x^k, I^k**)**

3. *If* π^k *gives strong stationarity of* x^k *then STOP,* x^k *is B-stationary.*
Else set $x^{k+1} := x^k + d^k$, $I^{k+1} \leftarrow$ **Select**$[\mathcal{I}(x^{k+1}), \xi^k]$, $k := k+1$ *and return to 2.*

It only remains to give a formal description of a selection rule for the piece index $I^{k+1} \in \mathcal{I}(x^{k+1})$, in which we use families of indices $A_i(x)$ and $M(x)$ described in §3.1:

Select $[\mathcal{I}(x^{k+1}), \xi^k]$

If $M(x^{k+1}) \neq \emptyset$ and $\min\{\xi^k_{ij} \mid (i,j) \in I^k,\, i \in M(x^{k+1})\} < 0$ then

Let $(i', j') \in \arg\min\{\xi^k_{ij} \mid (i,j) \in I^k, i \in M(x^{k+1})\}$;

Select $\tilde{\jmath} \in A_{i'}(x^{k+1}) \setminus \{j'\}$ and set

$$I^{k+1} := \{(i', \tilde{\jmath})\} \cup I^k \setminus \{(i', j')\} \ .$$

Else

Set $I^{k+1} := I^k$.

Motivation for this kind of selection rule goes back to standard active set methods [18] for nonlinear programs, and, for MPECs, to [43, 44]. We remove from the piece index I^k the pair (i', j') corresponding to the constraint function which seems provide the most promising reduction chances for the objective function, in view of the negative value of $\xi^k_{i'j'}$. Notice that the rule is based on the modification of at most one constraint at a time, and that, since no particular assumption has been made regarding the structure of I^0, the insertion of $(i', \tilde{\jmath})$ might well be not necessary if $I^k \setminus \{(i', j')\}$ belongs to $\mathcal{I}(x^{k+1})$. We remark that such a rule is only one possible way of generating a new piece index; see Condition 3.15 in the next section for a more general selection framework.

3.3. Global convergence analysis. We aim to analyze convergence properties of the sequence $\{x^k\}$ generated by the trust-search decomposition method Algorithm 3.10 (**TS-MPEC**). We mention that the role of our later assumptions, namely ULSC (see Definition 3.5) and convergence of d^k to zero, can be inferred from the statement and proof of part (ii) of Theorem 3.11 which describes 2-step nonstationary repulsion of **TS-MPEC**. Furthermore, implicit identification of multi-active complementarity constraints is shown in the proof. A tacit assumption throughout is that the initial point x^0 is feasible for the MPEC and the initial piece index I^0 lies in $\mathcal{I}(x^0)$.

THEOREM 3.11. *Let* $\{x^k\}$ *be generated by Algorithm* 3.10 *applied to the MPEC* (1.1) *and* $\bar{x} \in \mathbb{R}^n$.

(i) *If* $\bar{x}$ *is not one-piece stationary then it has a neighborhood* U *that can intersect the sequence of iterates at most once.*

(ii) *Suppose that* $\bar{x}$ *is a weakly stationary but non-M-stationary point at which MPEC-LICQ holds. Then it has a neighborhood* U *such that if consecutive iterates* x^K *and* x^{K+1} *lie in* U *then* $U \cap \{x^k\}_{k>K+1} = \emptyset$.

The neighborhood U *in each case is independent of the (feasible) starting point* x^0.

Proof. (i) The result is clear for non-feasible points, Algorithm 3.10 being of the feasible-point type. For any feasible point $\bar{x}$ that is not one-piece stationary, see part (i) of Remark 2.3 and take U to be the intersection of the neighborhoods corresponding to $\mathcal{U}'$ for each adjacent piece of the MPEC.

(ii) Let $\bar{x}$ be weakly stationary but not M-stationary for the MPEC. Since there are only finitely many pieces adjacent to $\bar{x}$, we can use NSR of the trust-search method applied to each NLP_I for which $\bar{x}$ is feasible but nonstationary, as follows. Let $\bar{\mathcal{J}}$ be the family of such piece indices I. Given NSR of the trust-search subroutine and monotonicity of the decomposition method, there is a neighborhood U of $\bar{x}$ such that for $x^K \in U$ and $I^K \in \bar{\mathcal{J}}$ (where the algorithm ensures x^K is feasible for NLP_{I^K}) we have $\{x^k\}_{k>K} \cap U = \emptyset$. So, without loss of generality, let U be a ball of arbitrarily small radius $r > 0$ about $\bar{x}$, and x^K be a feasible point of some piece NLP_{I^K} for which $I^K \notin \bar{\mathcal{J}}$, i.e., $\bar{x}$ is stationary for NLP_{I^K}, and x^K, $x^{K+1} \in U$. We only need to show that the **Select** procedure chooses $I^{K+1} \in \bar{\mathcal{J}}$. For such I^{K+1}, the previous argument using NSR and monotonicity gives the result: $U \cap \{x^k\}_{k>K+1} = \emptyset$.

Let $\pi^K = (\lambda^K, \mu^K, \xi^K)$ be the multiplier generated by $\mathbf{TS}(x^K, I^K)$ in calculating the step d^K that yields $x^{K+1} = x^K + d^K$, and $\bar{\pi} = (\bar{\lambda}, \bar{\mu}, \bar{\xi})$ be the KKT multiplier associated with weak stationarity of $\bar{x}$. Since $\bar{x}$ is stationary for NLP_{I^K} and MPEC-LICQ at $\bar{x}$ implies LICQ for NLP_{I^K} at $\bar{x}$, then $\bar{\pi}$ is also the (unique) KKT multiplier for NLP_{I^K} at $\bar{x}$. Furthermore, since $\bar{x}$ is not M-stationary, there exist $\bar{\imath} \in M(\bar{x})$ and distinct indices $\bar{\jmath}, \hat{\jmath} \in A_{\bar{\imath}}(\bar{x})$ such that $\bar{\xi}_{\bar{\imath}\bar{\jmath}} < 0$ and $\bar{\xi}_{\bar{\imath}\hat{\jmath}} \neq 0$. Notice that it is only possible to have a negative KKT multiplier when the corresponding constraint of NLP_{I^K} is an equality, that is $(\bar{\imath}, \bar{\jmath}) \in I^K$, c.f. Proposition 3.9.

Now observe that, depending on the size of the radius r of U, convergence of TR multipliers (Theorem 2.11) implies that π^K is arbitrarily close to $\bar{\pi}$. So let

$$\begin{aligned} \epsilon_1 &= \min\{|\bar{\xi}_{ij}| : (i,j) \text{ is such that } \bar{\xi}_{ij} \neq 0\}, \\ \epsilon_2 &= \min\{|p_{ij}(\bar{x})| : (i,j) \text{ is such that } p_{ij}(\bar{x}) \neq 0\}, \end{aligned}$$

where the latter takes the value $+\infty$ if every $p_{ij}(\bar{x}) = 0$, and define $\epsilon = \min\{\epsilon_1, \epsilon_2\}$.

Then choose $r > 0$ such that, for all (i,j), since $x^K \in U$ then $|\xi^K_{ij} - \bar{\xi}_{ij}| < \frac{\epsilon}{2}$ and if $x \in U$ then $|p_{ij}(x) - p_{ij}(\bar{x})| < \frac{\epsilon}{2}$. In particular, since $\xi^K_{\bar{\imath}\bar{\jmath}} \leq -\epsilon$ and $|\xi^K_{\bar{\imath}\hat{\jmath}}| \geq \epsilon$, we have $\xi^K_{\bar{\imath}\bar{\jmath}} < -\frac{\epsilon}{2}$ and $|\xi^K_{\bar{\imath}\hat{\jmath}}| > \frac{\epsilon}{2}$. Therefore $\bar{\imath} \in M(x^{K+1})$, i.e., $M(x^{K+1}) \neq \emptyset$, and, since $(\bar{\imath}, \bar{\jmath}) \in I^K$,

$$\min\{\xi^K_{ij} \mid (i,j) \in I^K,\, i \in M(x^{K+1})\} \;\leq\; \xi^K_{\bar{\imath}\bar{\jmath}} \;<\; -\frac{\epsilon}{2}.$$

Thus, given x^{K+1}, the **Select** procedure will first choose $(i', j') \in I^K$, such that $i' \in M(x^{K+1})$ and $\xi^K_{i'j'} \leq \xi^K_{\bar{\imath}\bar{\jmath}} \;<\; -\frac{\epsilon}{2}$, and then $\tilde{\jmath} \in A_{i'}(x^{K+1}) \setminus \{j'\}$ to define

$$I^{K+1} := \{(i', \tilde{\jmath})\} \cup I^K \setminus \{(i', j')\}.$$

Notice, since $p_{i'\tilde{\jmath}}(x^{K+1}) = 0$ and $x^{K+1} \in U$, that

$$\frac{\epsilon}{2} > |p_{i'\tilde{\jmath}}(x^{K+1}) - p_{i'\tilde{\jmath}}(\bar{x})| = |p_{i'\tilde{\jmath}}(\bar{x})|,$$

which implies $p_{i'\tilde{\jmath}}(\bar{x}) = 0$ and we see that $\bar{x}$, which is feasible for NLP_{I^K}, must be feasible for $NLP_{I^{K+1}}$ too. Moreover, since $\xi^K_{i'j'} < -\frac{\epsilon}{2}$, we have $\bar{\xi}_{i'j'} < 0$ and, as $(i', j') \notin I^{K+1}$, of course it follows that $\bar{x}$ is not stationary for $NLP_{I^{K+1}}$ (similar to

Proposition 3.9). Thus $I^{K+1} \in \bar{\mathcal{J}}$ as promised earlier. ∎

COROLLARY 3.12. *Let* $\{x^k\}$ *be a sequence generated by Algorithm* 3.10 *applied to the MPEC* (1.1).

(i) *Any accumulation point* x^* *of* $\{x^k\}$ *is one-piece, hence weakly stationary for* (1.1).

(ii) *If* $(x^*, 0)$ *is an accumulation point of* $\{(x^k, d^k)\}$ *such that MPEC-LICQ holds at* x^**, then* x^* *is M-stationary.*

(iii) *If* $(x^*, 0)$ *is an accumulation point of* $\{(x^k, d^k)\}$ *such that both MPEC-LICQ and ULSC hold at* x^**, then* x^* *is B-stationary.*

Proof. (i) The statement follows from Part (i) of Theorem 3.11, because any point that is not one-piece stationary cannot be an accumulation point of $\{x_k\}$.

(ii) Assume that $(x^*, 0) = \lim_{k \in K}(x^k, d^k)$ where x^* is weakly stationary but not M-stationary. Part (ii) of Theorem 3.11 provides a neighborhood of x^* that contains at most finitely many iterates in the subsequence $\{x^k\}_{k \in K}$, contradicting convergence of this subsequence to x^*.

(iii) The statement follows from Part (ii) since an M-stationary point that satisfies ULSC is B-stationary. ∎

The proofs of parts (ii) and (iii) make use of a new condition, convergence of $\{d^k\}$ to zero on an appropriate subsequence, that is required in addition to MPEC-LICQ and ULSC to guarantee convergence to a B-stationary point. In fact, such a condition is not needed at all for convergence of some NLP methods including the trust-region scheme proposed in the first part of the paper.

3.4. Properties of a general NLP solver for global convergence of decomposition methods. We observe that the last two convergence results are based on nonstationary repulsion and convergence of multipliers of our trust-search scheme, and on an appropriate dual-based scheme for selecting the new piece index. The point of this subsection is to identify properties of NLP methods that are critical to our previous convergence analysis of decomposition methods. This will have immediate application to a modified TR method in section 4 and also invites research beyond TR methods.

By **Decomp** we denote a general decomposition algorithm for the MPEC (1.1). Given a feasible point x^k of (1.1), and a piece index $I^k \in \mathcal{I}(x^k)$, the k-th iteration of **Decomp** is aimed at determining a new feasible point x^{k+1}, and a new piece index $I^{k+1} \in \mathcal{I}(x^{k+1})$. In particular, one step of an algorithm for linearly constrained NLP is applied to NLP_{I^k} at x^k: the subroutine $\mathbf{\Gamma}(x^k, I^k)$ returns a step d^k, so that $x^{k+1} := x^k + d^k$ is feasible for NLP_{I^k}, and an estimate $\pi^k = (\lambda^k, \mu^k, \xi^k)$ of the KKT multipliers of NLP_{I^k}. Upon termination of $\mathbf{\Gamma}(x^k, I^k)$, a selection procedure $\mathbf{\Delta}(x^{k+1}, \xi^k)$ is executed to generate $I^{k+1} \in \mathcal{I}(x^{k+1})$ before starting next iteration. A scheme of such a general algorithm is described below:

ALGORITHM 3.13 (**Decomp**).

1. *Initialization. Set* $k := 0$.

2. d^k, $\pi^k = (\lambda^k, \mu^k, \xi^k) \leftarrow \mathbf{\Gamma}(x^k, I^k)$.

3. *If* π^k *gives strong stationarity of* x^k *then STOP,* x^k *is B-stationary.*

Else set $x^{k+1} := x^k + d^k$, $I^{k+1} \leftarrow \mathbf{\Delta}(x^{k+1}, \xi^k)$, $k := k+1$ *and go to 2.*

For simplicity of notation we have only shown the inputs of $\mathbf{\Gamma}$ as x^k and I^k though other inputs could include additional information such as the previous multiplier π^{k-1}.

Now we state a list of conditions to be satisfied by **Decomp** in order to guarantee significant convergence properties of the sequence $\{x^k\}$. This list clarifies what is essential to our analysis rather than being a recommendation of properties that "all good methods" for linearly constrained NLP should satisfy, after all the standard TR approach does not necessarily satisfy NSR.

CONDITION 3.14. *For each piece index* I *such that* NLP_I *is feasible:*

(i) $\mathbf{\Gamma}(\cdot, I)$ *is a subroutine that, given a point* x*, returns a step* d *and multiplier estimate* π*. It defines a monotonic, feasible point algorithm with nonstationary repulsion for solving* NLP_I*.*

(ii) *If* $\{x^k\}_{k\in\mathcal{K}}$ *is a feasible sequence converging to a stationary point* $\bar{x}$ *of* NLP_I*, then the sequence* $\{\pi^k\}_{k\in\mathcal{K}}$ *obtained by executing* $\mathbf{\Gamma}(x^k, I)$ *is bounded, and each of its accumulation points* π *is a KKT multiplier for* NLP_I *at* $\bar{x}$*.*

(iii) *Given* $\mathcal{K}$ *as above, the sequence of steps* $\{d^k\}_{k\in\mathcal{K}}$ *returned by* $\mathbf{\Gamma}(x^k, I)$ *converges to zero.*

We remark that the projected-gradient method [8] satisfies Condition 3.14: As pointed out in Remark 2.3(i), NSR is given by [15]. Part (ii) follows from [17, Theorem 3.5] as discussed after Theorem 2.11 above. Part (iii) is implicit in the method itself which measures stationarity by the length of the projected-gradient.

Returning to **Decomp**, here is the condition for piece selection.

CONDITION 3.15. *The selection procedure* $\mathbf{\Delta}(x, \xi)$ *to find a piece index* $I_x \in \mathcal{I}(x)$*, see* (1.2)*, is based on the search for at least one "sufficiently negative" multiplier, say* $\xi_{i'j'}$*, corresponding to a multi-active constraint function at* x*. In particular, if there exist* $i' \in M(x)$ *and* j' *with*

$$\xi_{i'j'} \le \delta \min\{\xi_{ij} \mid i \in M(x)\} < 0 \tag{3.7}$$

for some fixed $\delta \in (0, 1)$*, then* $\mathbf{\Delta}(x, \xi)$ *returns* $I_x \in \mathcal{I}(x)$ *such that for some such* (i', j') *we have* $(i', j') \in I \setminus I_x$*.*

Notice that the selection rule adopted in Algorithm 3.10 satisfies Condition 3.15. Indeed, I^{k+1} is derived from I^k by inserting $(i', \tilde{\jmath})$ in place of the pair (i', j') satisfying (3.7). More generally a selection rule to derive I^{k+1} should be based on the elimination from I^k of an element satisfying (3.7), if any, and on the possible "restoration" of the piece index so obtained, in case $I^k \setminus \{(i', j')\} \notin \mathcal{I}(x^{k+1})$.

Here is the counterpart of Theorem 3.11 with a general NLP routine replacing the trust-search method **TS-NLP**.

THEOREM 3.16. *Let* $\{x^k\}$ *be generated by Algorithm* 3.13 (**Decomp**) *applied to the MPEC* (1.1)*, and* $\bar{x} \in \mathbb{R}^n$*. Assume that* **Decomp** *satisfies conditions* 3.14(i)-(ii) *and* 3.15.

(i) *If* $\bar{x}$ *is not one-piece stationary then it has a neighborhood* U *that can intersect the sequence of iterates at most once.*

(ii) *Suppose that $\bar{x}$ is a weakly stationary but non-M-stationary point at which MPEC-LICQ holds. Then it has a neighborhood U such that if consecutive iterates x^K and x^{K+1} lie in U then $U \cap \{x^k\}_{k>K+1} = \emptyset$.*

The neighborhood U in each case is independent of the (feasible) starting point x^0.

Proof. The proof of Theorem 3.11 applies to the general case with one small amendment to account for the constant δ used in the general selection rule $\mathbf{\Delta}$: define $\epsilon = \delta \min\{\epsilon_1, \epsilon_2\}$. ∎

The next result is proved exactly as Corollary 3.12 but with Theorem 3.16 replacing Theorem 3.11.

COROLLARY 3.17. *Let x^* be any accumulation point of the sequence $\{x^k\}$ generated by Algorithm* 3.13 (**Decomp**) *applied to the MPEC* (1.1)*, and assume that* **Decomp** *satisfies conditions* 3.14 *and* 3.15.

(i) *x^* is one-piece, hence weakly stationary for* (1.1).
(ii) *If MPEC-LICQ holds at x^*, then x^* is M-stationary for* (1.1).
(iii) *If both MPEC-LICQ and ULSC hold at x^*, then x^* is B-stationary for* (1.1).

It would be tempting to conjecture that Corollary 3.17 holds under Condition 3.14 with NSR of each $\Gamma(\cdot, I)$ weakened to only require that limit points of the method are stationary for NLP_I. However, this may be insufficient for the Corollary because the algorithm $\Gamma(\cdot, I)$ should be robust in the sense that its iteration sequence can be interrupted, by switching to another piece and then returning to NLP_I several iterations later, without risking stationarity of limit points. On the other hand, a method satisfying NSR is robust to switching, as is obvious from the trivial proof of Theorem 3.11(i), or being used in a hybrid with another method. A formal statement on hybrid schemes that include a method with NSR appears in [15, Theorem 3.2].

We conclude by recalling that local superlinear convergence of the PSQP decomposition method [32, Chapter 6], which solves a quadratic program based on a Lagrangian function at each iteration, occurs near any B-stationary point at which MPEC-LICQ and a second-order sufficient condition hold. Furthermore superlinear convergence of PSQP entails convergence of d^k to zero and convergence of multipliers. It would therefore be worth investigating a hybrid decomposition method that relies on LP-based trust-region subproblems for global convergence, but attempts to switch to QP-based subproblems to asymptotically achieve superlinear convergence.

4. Improving convergence of the trust-search decomposition method.

It is clear from the analysis presented in section 3 that the convergence properties of a decomposition algorithm for MPECs depend on the features of the NLP method that is employed. In particular, convergence properties of the trust-search decomposition method Algorithm 3.10 would be more attractive should the NLP solver, **TS-NLP** (Algorithm 2.1), satisfy part (iii) of Condition 3.14. In order to address this shortcoming, in §4.1 we modify the trust-search subroutine $\mathbf{TS}(x)$ of **TS-NLP** to ensure $d^k \to 0$, while preserving nonstationary repulsion and convergence of multipliers, by refining the developments of section 2. The modified trust-search subroutine is called $\mathbf{TS}^+(x)$. Next, in §4.2, we will merely apply the general framework of Theorem 3.16 and Corollary 3.17, and observe that the decomposition method for MPEC that uses $\mathbf{TS}^+(x)$, called $\mathbf{TS}^+$**-MPEC**, has good global convergence properties.

4.1. A modified trust-search method for linearly-constrained NLP.

Notice that Proposition 2.10 gives a sufficient condition for convergence of multipliers, independently of the particular method used to generate the sequence $\{x^k\}$: $\{v^k/r^k\} \to 0$ on an appropriate subsequence $\mathcal{K}$. Since we also want $\{d^k\}_{k\in\mathcal{K}} \to 0$, it seems reasonable to add a new condition defining an acceptable descent direction $d(r)$ by using $|v(x,r)|/r$ as a forcing function on $\|d(r)\|$:

$$f(x+d(r)) - f(x) \le \alpha v(x,r) \quad \text{and} \quad \|d(r)\| \le \frac{|v(x,r)|}{r} \,. \tag{4.1}$$

The modified trust-search scheme becomes:

[$\mathbf{TS}^+(x)$]
Set $r := \rho/\beta$;

Repeat
Set $r := \beta r$;
$d(r)$, $v(x,r) \leftarrow LP(x,r)$;
Until (4.1) holds.

The remaining part of the section will be devoted to reconsidering convergence results of section 2, in view of the new trust-search scheme.

Recall that $\mathcal{F}$ denotes the feasible region of the linearly-constrained NLP (2.1). Recall also our standing assumption that a fixed polyhedral norm is used in forming TR subproblems. The following two results correspond to Proposition 2.6 and Corollary 2.7, respectively. To improve readability the proofs are given in details only for those parts where the modified trust-search scheme may affect the results.

Proposition 4.1. *For any bounded set X of $\mathcal{F}$ and $\gamma > 0$, there exists a scalar $r_\gamma \in (0,\rho)$ with the following property. If $x \in X$ and $\hat{r} \in (0, \min\{r_\gamma, \gamma\}]$ satisfy $\frac{v(x,\hat{r})}{\hat{r}} \le -\gamma$, then $\mathbf{TS}^+(x)$ terminates, returning a trust-region radius $r_x > \beta\hat{r}$.*

Proof. We use the proof of Proposition 2.6 as a guide. In particular, let r_γ be given by that proof and choose $\hat{r} \in (0, \min\{r_\gamma, \gamma\}]$. Let $r \in (0, \hat{r}]$ and, as usual, let $d(r)$ denote an optimal solution of $LP(x,r)$. We have, as in the proof of Proposition 2.6, that

$$f(x+d(r)) - f(x) \le \alpha v(x,r).$$

Also

$$\|d(r)\| \le r \le \hat{r} \le \gamma \le \frac{|v(x,\hat{r})|}{\hat{r}} \le \frac{|v(x,r)|}{r} \,,$$

where the last inequality follows from Lemma 2.4. That is, $d(r)$ is an acceptable feasible descent direction at x. Thus, a sufficient condition for $\mathbf{TS}^+(x)$ to terminate is that $r \le \hat{r}$, and the proof can be completed by the same argument as the one used in proving Proposition 2.6. □

Corollary 4.2. *The modified trust-search algorithm, Algorithm 2.1 with $\mathbf{TS}^+$ replacing $\mathbf{TS}$, has nonstationary repulsion.*

Proof. The statement is a straightforward consequence of Corollary 2.7, whose validity for $\mathbf{TS}^+(x)$ easily follows from Proposition 4.1. □

The modified trust-search subroutine gives the essence of Condition 3.14:

THEOREM 4.3. *Let $\{x^k\}$ be a feasible sequence for* (2.1) *on which the objective function f is non-increasing, and $\{x^k\}_{k\in\mathcal{K}}$ be a convergent subsequence with limit $\bar{x}$. If*

$$\mathcal{K} \subset \{k \mid x^{k+1} \textit{ is generated by } \mathbf{TS}^+(x^k)\}$$

then $\bar{x}$ is stationary for (2.1) *such that*

(i) *both $\{|v^k|/r^k\}_{k\in\mathcal{K}}$ and $\{d^k\}_{k\in\mathcal{K}}$ converge to zero, and*

(ii) *there exists a bounded subsequence of multipliers $\{(\lambda^k, \mu^k)\}_{k\in\mathcal{K}}$ corresponding to $\{x^k\}_{k\in\mathcal{K}}$, each accumulation point of which is a KKT multiplier for* (2.1) *at $\bar{x}$.*

Proof. (i) This proof follows the proof of Theorem 2.11 in almost every detail, with the modified trust-search subroutine replacing the original. We therefore only give details where the effect of the modification to the trust-search subroutine can be seen, which is for the case when there exists a subsequence $\{r^k\}_{k\in\mathcal{K}'\subset\mathcal{K}}$ converging to zero and, for a contradiction, we assume that there exist $\bar{\gamma}$ and $\bar{k}$ such that for every $k \geq \bar{k}$ with $k \in \mathcal{K}'$ it holds that

$$\frac{v^k}{r^k} \leq -\frac{\bar{\gamma}}{\beta} \ .$$

As in the proof of Theorem 2.11, we obtain

$$\frac{v(x^k, \frac{r^k}{\beta})}{\frac{r^k}{\beta}} \leq -\bar{\gamma} \ .$$

Now apply Proposition 4.1 (previously Proposition 2.6) by taking $\gamma = \bar{\gamma}$, $X = \{x \mid \|x - \bar{x}\| \leq 1\} \cap \mathcal{F}$, and obtain $r_{\bar{\gamma}} \in (0, \rho)$ satisfying the property therein. Since $\{r^k\}_{k\in\mathcal{K}'} \to 0$ and $\{x^k\}_{k\in\mathcal{K}'} \to \bar{x}$, then for sufficiently large $k \geq \bar{k}$ with $k \in \mathcal{K}'$ we have

$$\frac{r^k}{\beta} \ \leq \ \min\{r_{\bar{\gamma}}, \bar{\gamma}\}$$

and $x^k \in X$. As a consequence, by letting $\hat{r} := \frac{r^k}{\beta}$, we have that $\mathbf{TS}^+(x)$ returns r such that

$$r > \beta\hat{r} = r^k$$

which contradicts the choice of r^k.

Therefore $\{|v^k|/r^k\}_{k\in\mathcal{K}}$ converges to zero. Moreover, since $\frac{|v^k|}{r^k}$ is a forcing function on $\|d^k\|$ by construction of the algorithm, see (4.1), then $\{d^k\}_{k\in\mathcal{K}}$ converges to zero as well.

(ii) The statement follows from part (i) in view of Proposition 2.10. ∎

4.2. A modified trust-search decomposition method. Here we embed the new trust-search scheme into our original decomposition approach for linearly-constrained MPECs. We consider the algorithm

ALGORITHM 4.4 ($\mathbf{TS}^+$**-MPEC**).

1. *Choose* $x^0 \in \mathcal{F}$, $I^0 \in \mathcal{I}(x^0)$, $\alpha, \beta \in (0,1)$, $\rho > 0$. *Set* $k := 0$.

2. d^k, $\pi^k = (\lambda^k, \mu^k, \xi^k) \leftarrow \mathbf{TS}^+(x, I)$.

3. *If* π^k *gives strong stationarity of* x^k *then STOP,* x^k *is B-stationary.* *Else set* $x^{k+1} := x^k + d^k$, $I^{k+1} \leftarrow \mathbf{Select}[\mathcal{I}(x^{k+1}), \xi^k]$, $k := k+1$ *and return to 2.*

where $\mathbf{TS}^+(x, I)$ denotes $\mathbf{TS}^+(x)$ applied to NLP_I, and we use the same selection rule as in section 3. The following result is a straightforward consequence of the properties of the modified trust-search subroutine given in the previous subsection. As usual we assume x^0 is a feasible point for the linearly-constrained MPEC (1.1) and take $I^0 \in \mathcal{I}(x^0)$.

THEOREM 4.5. *Let* $\{x^k\}$ *be a sequence generated by* **TS$^+$-MPEC**, *Algorithm* 4.4, *applied to the MPEC* (1.1), *and* x^* *be an accumulation point of this sequence.*

(i) *The accumulation point* x^* *is one-piece hence weakly stationary for* (1.1).
(ii) *If MPEC-LICQ holds at* x^* *then it is M-stationary for* (1.1).
(iii) *If both MPEC-LICQ and ULSC hold at* x^*, *then it is B-stationary for* (1.1).

Proof. In view of Corollary 4.2 and Theorem 4.3, we have that Condition 3.14 is satisfied by $\mathbf{TS}^+(x, I)$. Furthermore, recall that the selection rule proposed in §3.2 satisfies Condition 3.15. Therefore, part (i) follows from part (i) of Theorem 3.16, because any point that is not one-piece stationary cannot be an accumulation point of $\{x_k\}$, while part (ii) follows from Part (ii) of Theorem 3.16. ∎

We notice that comparable convergence results in the field typically require some stronger assumptions than ours. In particular in [43] convergence to a strong stationary point where MPEC-LICQ holds is guaranteed provided that the trust-region radii remain bounded away from zero, the trust region is asymptotically inactive, and the active complementarity component are correctly identified on the limit (see [43, Proposition 5.1]). Similar but stronger results are presented in [44] where boundedness of the trust-region radii away from zero is no longer needed. Nevertheless, both asymptotic inactivity of the trust region (see [44, Proposition 3.4]) and correct identification of active complementarity components (see [44, Proposition 3.1-3.6, Corollary 3.1]) are still required. Moreover convergence of multipliers is only obtained, under ULSC, by using an auxiliary least squares problem to get multiplier estimates (see [44, Proposition 3.4-3.6, Corollary 3.1]). However, these comparisons may not be entirely fair in that [43, 44] deal with a broader class of problems than we do, namely MPECs with nonlinear constraint functions.

It is also worth mentioning (again) that the ϵ-active set method of [23] yields B-stationary limit points under similar conditions to those above but without the ULSC. This is achieved at the expense of considerably complicating the algorithm however.

5. Conclusion. At this point the relevant contributions of the paper are apparent from both the NLP and MPEC points of view. First, in section 2, regarding linearly-constrained NLP, we present a linear trust-search method **TS-NLP** that strengthens, via nonstationary repulsion, the global convergence properties that might be expected of trust-region methods; and give a new global convergence result on convergence of multipliers. The latter is convenient and general in that it does not require a supplementary least squares problem to be solved at each iteration, or any additional conditions to be satisfied by the problem or iteration sequence.

Second, in section 3, we exploit these results to both improve and simplify global convergence to B-stationary points of decomposition methods for linearly-constrained MPECs. Part of this work is to identify the convergence properties that would be required of a general NLP solver to ensure good global convergence properties of an MPEC decomposition scheme. This leads to the modified trust-search subroutine **TS$^+$** for NLP, which is embedded in decomposition scheme **TS$^+$-MPEC** that is analyzed in section 4. Viewed from above, the meld of NSR and multiplier convergence provided by the trust-search method is rather striking in the context of decomposition schemes for MPECs.

We readily acknowledge the natural questions of how to obtain superlinear convergence, possibly via a hybrid of modified trust-search subroutine with a quadratic programming step, at each iteration of **TS$^+$-MPEC**; and how to deal with nonlinear constraints. These questions are issues for future work.

Acknowledgements. We would like to thank Francisco Facchinei and Stefan Scholtes for their respective insights on NLP trust-region and MPEC decomposition methods. We also thank two anonymous referees for their detailed comments that helped clarify the paper, and for drawing our attention to references [24, 27].

REFERENCES

[1] M. Anitescu, *On using the elastic mode in nonlinear programming approaches to mathematical programs with complementarity constraints*, SIAM Journal on Optimization, 15 (2005), pp. 1203–1236.

[2] M. Anitescu, P. Tseng, and S. J. Wright, *Elastic-mode algorithms for mathematical programs with equilibrium constraints: global convergence and stationarity properties*, Tech. Report ANL/MCS P1242-0405, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, USA, May 2005.

[3] H. Y. Benson, D. F. Shanno, and R. J. Vanderbei, *Interior-point methods for nonconvex nonlinear programming: Complementarity constraints*, Tech. Report ORFE 02-02, Department of Operations Research and Financial Engineering, Princeton University, July 2002, to appear in Mathematical Programming, Series A.

[4] D. P. Bertsekas, *Nonlinear programming*, Athena Scientific, Belmont, Massachussets, 1995.

[5] J. F. Bonnans and A. Shapiro, *Perturbation analysis of optimization problems*, Springer-Verlag, New York, 2000.

[6] R. H. Byrd, N. I. M. Gould, J. Nocedal, and R. H. Waltz, *An algorithm for nonlinear optimization using linear programming and equality constrained subproblems*, Mathematical Programming, 100 (2004), pp. 27–48.

[7] ———, *On the convergence of successive linear-quadratic programming algorithms*, SIAM Journal on Optimization, 16 (2005), pp. 471–489.

[8] P. H. Calamai and J. J. Moré, *Projected gradient methods for linearly constrained problems*, Mathematical Programming, 39 (1987), pp. 93–116.

[9] C. M. Chin and R. Fletcher, *Numerical performance of an SLP-filter algorithm that takes EQP steps*, Tech. Report NA/202, Department of Mathematics, University of Dundee, Dundee, UK, 2001.

[10] ———, *On the global convergence of an SLP-filter algorithm that takes EQP steps*, Mathematical Programming, 96 (2003), pp. 161–177.

[11] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust-region methods*, MPS-SIAM, 2000.

[12] A. V. de Miguel, M. Friedlander, F. Nogales, and S. Scholtes, *An interior-point method for MPECS based on strictly feasible relaxations*, SIAM Journal on Optimization, 16 (2005), pp. 587–609.

[13] F. Facchinei, H. Jiang, and L. Qi, *A smoothing method for mathematical programs with equilibrium constraints*, Mathematical Programming, 85 (1999), pp. 81–106.

[14] F. Facchinei and J. S. Pang, *Finite-dimensional variational inequalities and complementarity problems, Volume I*, Springer series in operations research, Springer, New York, 2003.

[15] M. C. Ferris and D. Ralph, *Projected gradient methods for nonlinear complementarity prob-*

lems via normal maps, in Recent advances in nonsmooth optimization, D. Z. Du, L. Qi, and R. S. Womersley, eds., World Scientific Publishing, 1995.

[16] A. V. Fiacco, *Introduction to sensitivity and stability analysis in nonlinear programming*, Mathematics in Science and Engineering 165, Academic Press, London, 1983.

[17] A. Fischer, *Modified Wilson method for nonlinear programs with nonunique multipliers*, Mathematics of Operations Research, 24 (1999), pp. 699–727.

[18] R. Fletcher, *Practical methods of optimization*, John Wiley and Sons, 1987.

[19] R. Fletcher and S. Leyffer, *Solving mathematical programs with complementarity constraints as nonlinear programs*, Optimization Methods and Software, 19 (2004), pp. 15–40.

[20] R. Fletcher, S. Leyffer, D. Ralph, and S. Scholtes, *Local convergence of SQP methods for mathematical programs with equilibrium constraints*, SIAM Journal on Optimization, (2006).

[21] R. Fletcher and E. Sainz de la Maza, *Nonlinear programming and nonsmooth optimization by successive linear programming*, Mathematical Programming, 43 (1989), pp. 235–256.

[22] M. Fukushima and J. S. Pang, *Convergence of a smoothing continuation method for mathematical programs with complementarity constraints*, in Ill-posed Variational Problems and Regularization Techniques, M. Théra and R. Tichatschke, eds., vol. 447 of Lecture Notes in Economics and Mathematical Systems, Berlin/Heidelberg, 1999, pp. 99–110.

[23] M. Fukushima and P. Tseng, *An implementable active-set algorithm for computing a B-stationary point of a mathematical program with linear complementarity constraints*, SIAM Journal on Optimization, 12 (2002), pp. 724–739.

[24] S. A. Gabriel and J. S. Pang, *A trust region method for constrained nonsmooth equations*, in Large scale optimization: State of the art, W. W. Hager, D. W. Hearn, and P. M. Pardalos, eds., Kluwer Academic Publishers, 1994.

[25] X. Hu and D. Ralph, *Convergence of a penalty method for mathematical programming with complementarity constraints*, Journal of Optimization Theory and Applications, 123 (2004), pp. 365–390.

[26] X. X. Huang, X. Q. Yang, and D. L. Zhu, *A sequential smooth penalization approach to mathematical programs with complementarity constraints*, Numerical Functional Analysis and Optimization, 27 (2006), pp. 71–98.

[27] H. Jiang, M. Fukushima, L. Qi, and D. Sun, *A trust region method for solving generalized complementarity problems*, SIAM Journal on Optimization, 8 (1998), pp. 140–157.

[28] H. Jiang and D. Ralph, *QPECgen, a MATLAB generator for mathematical programs with quadratic objectives and affine variational inequality constraints*, Computational Optimization and Applications, 13 (1999), pp. 25–59.

[29] ———, *Smooth SQP methods for mathematical programs with nonlinear complementarity constraints*, SIAM Journal on Optimization, 10 (2000), pp. 779–808.

[30] G. Lin and M. Fukushima, *Hybrid algorithms with active set identification for mathematical programs with complementarity constraints.* preprint, Department of Applied Mathematics and Physics, Kyoto University, November 2002, to appear in Optimization Methods and Software.

[31] X. Liu, G. Perakis, and J. Sun, *A robust SQP methods for mathematical programs with complementarity constraints*, 34 (2006), pp. 5–33.

[32] Z. Q. Luo, J. S. Pang, and D. Ralph, *Mathematical programs with equilibrium constraints*, Cambridge University Press, 1996.

[33] ———, *Piecewise sequential quadratic programming for mathematical programs with nonlinear complementarity constraints*, in Multilevel Optimization: Algorithms, Complexity and Applications, A. Migdalas, P. M. Pardalos, and P. Värbrand, eds., Kluwer Academic Publishers, Norwell, MA, 1998, pp. 209–229.

[34] G. Maier, F. Giannessi, and A. Nappi, *Indirect identification of yield limits by mathematical programming*, Engineering Structures, 4 (1982), pp. 86–98.

[35] J. V. Outrata, M. Kocvara, and J. Zowe, *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints*, Kluwer, 1998.

[36] E. Polak, *Computational methods in optimization: a unified approach*, Academic Press, New York, 1971.

[37] A. U. Raghunathan and L. T. Biegler, *Interior point methods for mathematical programs with complementarity constraints*, SIAM Journal on Optimization, 15 (2005), pp. 720–750.

[38] D. Ralph and S. J. Wright, *Some properties of regularization and penalization schemes for MPECs*, Optimization Methods and Software, 19 (2004), pp. 527–556.

[39] R. T. Rockafellar, *Convex analysis*, Princeton University Press, Princeton, New Jersey, 1970.

[40] H. Scheel and S. Scholtes, *Mathematical programs with complementarity constraints: sta-*

tionarity, optimality, and sensitivity, Mathematics of Operations Research, 25 (2000), pp. 1–22.

[41] S. Scholtes, *Convergence properties of a regularization scheme for mathematical programs with complementarity constraints*, SIAM Journal on Optimization, 11 (2001), pp. 918–936.

[42] ———, *Nonconvex structures in nonlinear programming*, Operations Research, 52 (2004), pp. 368–383.

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

[44] M. Stöhr, *Nonsmooth trust region methods and their applications to mathematical programs with equilibrium constraints*, Shaker-Verlag, Aachen, Germany, 1999.

[45] J. Zhang and G. Liu, *A new extreme point algorithm and its application in PSQP algorithms for solving mathematical programs with linear complementarity constraints*, Journal of Global Optimization, 19 (2001), pp. 335–361.