Numerical Behavior of a Stabilized SQP Method for Degenerate NLP Problems

El–Sayed M. E. Mostafa[1] *, Luís N. Vicente[2] *, and Stephen J. Wright[3] **

[1] Centro de Matemática, Universidade de Coimbra, 3001-454 Coimbra, Portugal. On leave from the Department of Mathematics, Faculty of Science, Alexandria University, Alexandria, Egypt (emostafa@mat.uc.pt).
[2] (Correspondent author) Departamento de Matemática, Universidade de Coimbra, 3001-454 Coimbra, Portugal (lnv@mat.uc.pt).
[3] Department of Computer Sciences, University of Wisconsin, Madison, WI 53706, USA (swright@cs.wisc.edu).

Abstract. In this paper we discuss the application of the stabilized SQP method with constraint identification (sSQPa) recently proposed by S. J. Wright [12] for nonlinear programming problems at which strict complementarity and/or linear independence of the gradients of the active constraints may fail to hold at the solution. We have collected a number of degenerate problems from different sources. Our numerical experiments have shown that the sSQPa is efficient and robust even without the incorporation of a classical globalization technique. One of our goals is therefore to handle NLPs that arise as subproblems in global optimization where degeneracy and infeasibility are important issues. We also discuss and present our work along this direction.

Key words. nonlinear programming, successive quadratic programming, degeneracy, identification of active constraints, infeasibility

1 Introduction

We consider the general nonlinear programming (NLP) problem written in the form

$$\min_{z} \quad \phi(z) \qquad \text{subject to} \quad g_I(z) \leq 0, \quad g_E(z) = 0, \tag{1}$$

where $\phi : \mathbb{R}^n \to \mathbb{R}$, $g_I : \mathbb{R}^n \to \mathbb{R}^{m_I}$, $g_E : \mathbb{R}^n \to \mathbb{R}^{m_E}$ ($m_E = m - m_I$; m the total number of constraints) are assumed to be twice continuously differentiable

* Support for the first two authors was provided by Centro de Matemática da Universidade de Coimbra, by FCT under grant POCTI/35059/MAT/2000, and by the European Union under grant IST-2000-26063.

** Support for this author was provided by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing, U.S. Department of Energy under contract W-31-109-Eng-38, and the National Science Foundation under grants CDA-9726385 and ACI-0082065.

functions. Let us write the two subsets of indices for the inequality and equality constraints, respectively, as

$$I = \{i = 1, \dots, m_I\}, \qquad E = \{i = m_I + 1, \dots, m\}.$$

Throughout this article the subsets I and E refer to the elements in the subsets of inequality and equality constraints. The Lagrangian function associated with this problem is

$$\mathcal{L}(z, \lambda) = \phi(z) + \lambda_I^T g_I(z) + \lambda_E^T g_E(z),$$

where $\lambda = (\lambda_I, \lambda_E) \in \mathrm{I\!R}^m$ are the Lagrange multipliers associated with the inequality and equality constraints in (1). For simplicity we write the vector of the multipliers as (λ_I, λ_E), while the accurate form would be $(\lambda_I^T, \lambda_E^T)^T$. The default norm in this paper is the ℓ_2.

2 Assumptions, notations, and basic results

The local convergence theory of the stabilized SQP method with constraint identification is based on the following assumptions [12]:

Assumption 1 *Let ϕ and g be twice Lipschitz continuously differentiable in a neighborhood of a point z^*. Let the Mangasarian-Fromovitz constraint qualification, the first-order necessary optimality conditions, and a form of second-order sufficient optimality conditions hold at z^*.*

Note that there is no assumption made about the linear independence of the gradients of the active constraints. Since the vector of optimal Lagrange multipliers is not unique if the gradients of the active constraints are linearly dependent, we need to consider the set of optimal Lagrange multipliers, denoted by $\mathcal{S}_\lambda$:

$$\mathcal{S}_\lambda = \{\lambda \mid \nabla_z \mathcal{L}(z^*, \lambda) = 0,\ \lambda_I^T g_I(z^*) = 0,\ \lambda_I \geq 0\}.$$

The optimal primal-dual set consists of the pairs (z, λ) in

$$\mathcal{S} = \{z^*\} \times \mathcal{S}_\lambda.$$

We also remark that there is no assumption about strict complementarity between z^* and the elements in $\mathcal{S}_\lambda$.

We need now several definitions to describe the stabilized SQP method with constraint identification. The set of active inequality constraints at z^* is defined as

$$\mathcal{B} = \{i = 1, \dots, m_I \mid g_{I_i}(z^*) = 0\}.$$

For any optimal multipliers $\lambda^* \in \mathcal{S}_\lambda$ we define the set

$$\mathcal{B}_+(\lambda^*) = \{i \in \mathcal{B} \mid \lambda^*_{I_i} > 0\}.$$

The set of strong active constraints and the set of weak active constraints are defined as:

$$\mathcal{B}_+ = \bigcup_{\lambda^* \in \mathcal{S}_\lambda} \mathcal{B}_+(\lambda^*), \qquad \mathcal{B}_0 = \mathcal{B} \backslash \mathcal{B}_+.$$

The distance of a pair (z, λ) to the optimal primal-dual set $\mathcal{S}$ is denoted by $\delta(z, \lambda)$:

$$\delta(z, \lambda) = \mathrm{dist}((z, \lambda), \mathcal{S}),$$

where

$$\mathrm{dist}((z, \lambda), \mathcal{S}) = \inf_{(z^*, \lambda^*) \in \mathcal{S}} \|(z^*, \lambda^*) - (z, \lambda)\|.$$

3 Stabilized SQP with constraint identification

SQP methods have shown to be quite successful in solving NLP problems. For degenerate NLP problems, where at the solution the linear independence of the gradients of the active constraints and/or the strict complementarity condition may not hold, Wright [12] has designed a stabilized SQP method, *algorithm sSQPa*, to handle such type of problems. For that purpose, a constraint identification procedure, *procedure ID0*, has been developed to identify the active inequality constraints and, furthermore, to classify them as strong or weak active constraints. The method also considers the solution of an LP subproblem to provide an *interior* multipliers estimate and a stabilization of the traditional QP subproblem. Both the constraint identification procedure and the interior multipliers estimate are called when no sufficient reduction is obtained by the solution of the stabilized QP subproblem. Wright [12] proved a superlinear rate of local convergence for algorithm sSQPa with procedure ID0.

In the next subsections, we give a complete description of the overall algorithm, extending the presentation of Wright [12] to the general case of inequality and equality constraints.

3.1 SQP algorithm

Standard SQP methods for the NLP problem (1) typically solve a sequence of QP subproblems of the following form

$$\begin{aligned} &\min_{\Delta z} \ \nabla\phi(z)^T \Delta z + \frac{1}{2} \Delta z^T \nabla_{zz} \mathcal{L}(z, \lambda) \Delta z \\ &\text{subject to } \ g_I(z) + \nabla g_I(z)^T \Delta z \leq 0, \qquad g_E(z) + \nabla g_E(z)^T \Delta z = 0, \end{aligned} \tag{2}$$

where (z, λ) is the current iterate. The stabilized SQP method [12] considers instead the following minimax subproblem

$$\begin{aligned} \min_{\Delta z} \max_{\lambda_I^+ \geq 0, \lambda_E^+} \ & \nabla\phi(z)^T \Delta z + \frac{1}{2} \Delta z^T \nabla_{zz} \mathcal{L}(z, \lambda) \Delta z + (\lambda_I^+)^T [g_I(z) + \nabla g_I(z)^T \Delta z] \\ & + (\lambda_E^+)^T [g_E(z) + \nabla g_E(z)^T \Delta z] - \frac{\mu}{2} \|\lambda^+ - \lambda\|^2, \end{aligned}$$

where $\mu \geq 0$ is a given parameter, and the solution $\lambda^+ = (\lambda_I^+, \lambda_E^+)$ provides a new update for the Lagrange multipliers associated with the inequality and equality constraints. This minimax subproblem is in turn equivalent to the following QP subproblem

$$\begin{aligned} \min_{(\Delta z, \lambda^+)} \quad & \nabla\phi(z)^T \Delta z + \frac{1}{2}\Delta z^T \nabla_{zz}\mathcal{L}(z,\lambda)\Delta z + \frac{\mu}{2}\|\lambda^+\|^2 \\ \text{subject to} \quad & g_E(z) + \nabla g_E(z)^T \Delta z - \mu(\lambda_E^+ - \lambda_E) = 0, \\ & g_I(z) + \nabla g_I(z)^T \Delta z - \mu(\lambda_I^+ - \lambda_I) \leq 0, \end{aligned} \tag{3}$$

with $\lambda^+ = (\lambda_I^+, \lambda_E^+)$; see [6], [12]. One can easily see that the QP subproblem (3) is posed in both z and λ^+ and has therefore m variables more than the traditional QP subproblem (2).

The stabilizing parameter μ introduced in the above subproblems is chosen as $\mu = \eta(z,\lambda)^\sigma$ with $\sigma \in (0,1)$, where $\eta(z,\lambda)$ is the size of the residual of the first-order necessary conditions given by

$$\eta(z,\lambda) = \left\| \begin{bmatrix} \nabla_z \mathcal{L}(z,\lambda) \\ \min(\lambda_I, -g_I(z)) \\ g_E(z) \end{bmatrix} \right\|,$$

with the min operator applied component-wise. In fact, $\eta(z,\lambda)$ represents a practical way of measuring the distance to the primal-dual set $\mathcal{S}$; see e.g., [12, Theorem 2].

The quantity $\eta(z,\lambda)$ also provides an estimate for the set of active constraints $\mathcal{B}$:

$$\mathcal{A}(z,\lambda) = \{i = 1,2,\ldots,m_I \mid g_{I_i}(z) \geq -\eta(z,\lambda)^\tau\}, \quad \tau \in (0,1), \tag{4}$$

see [12]. It is clear that when (z,λ) approaches a primal-dual solution, then the distance $\delta(z,\lambda)$ decreases and the interval of feasibility measured by the lower bound $-\eta(z,\lambda)^\tau$ reduces too, improving the quality of the estimation provided by $\mathcal{A}(z,\lambda)$. In addition, the estimated set $\mathcal{A}(z,\lambda)$ is partitioned into a subset $\mathcal{A}_+$ of estimated strong active constraints and a subset $\mathcal{A}_0$ of estimated weak active constraints. Depending on the decrease on $\eta(z,\lambda)$ provided by the solution of the QP subproblem (3), an LP subproblem is solved in order to maximize the multipliers corresponding to the inequality constraints in the subset $\mathcal{A}_+$, keeping the remaining multipliers corresponding to inequality constraints at zero. The identification procedure and the interior multipliers estimation will be introduced in subsections 3.2 and 3.3, respectively. Now, we restate the algorithm sSQPa [12].

Algorithm sSQPa
Choose parameters $\tau, \sigma \in (0,1)$, a tolerance $\texttt{tol} > 0$, and an initial starting point (z^0, λ^0) with $\lambda_I^0 \geq 0$. Compute $\mathcal{A}(z^0,\lambda^0)$ using (4), call procedure ID0 to compute the subsets $\mathcal{A}_+$ and $\mathcal{A}_0$, and solve the LP subproblem (6) to obtain $\hat{\lambda}^0$. Set $k \leftarrow 0$ and $\lambda^0 \leftarrow \hat{\lambda}^0$.
While $\eta(z^k, \lambda^k) > \texttt{tol}$ do

Solve (3) for $(\Delta z, \lambda^+)$, and set $\mu^k = \eta(z^k, \lambda^k)^\sigma$.
If $\eta(z^k + \Delta z, \lambda^+) \le \left(\eta(z^k, \lambda^k)\right)^{1+\sigma/2}$
  set $(z^{k+1}, \lambda^{k+1}) \leftarrow (z^k + \Delta z, \lambda^+)$; set $k \leftarrow k+1$;
else
  compute $\mathcal{A}(z^k, \lambda^k)$, and then apply ID0 to obtain $\mathcal{A}_+$ and $\mathcal{A}_0$;
  solve the LP subproblem (6) to obtain $\hat{\lambda}^k$, and set $\lambda^k \leftarrow \hat{\lambda}^k$;
end(if)

end(while)

For each iterate (z^k, λ^k) one solves the QP subproblem (3). If the computed step $(\Delta z, \lambda^+)$ yields a sufficient decrease in $\eta(z, \lambda)$, then $(\Delta z, \lambda^+)$ is accepted, otherwise $(\Delta z, \lambda^+)$ is rejected and the sSQPa algorithm switches to its `else` condition. In such a case, the set $\mathcal{A}(z, \lambda)$ is updated and the procedure ID0 is called to partition the set $\mathcal{A}(z, \lambda)$ into the subsets $\mathcal{A}_+$ and $\mathcal{A}_0$. A new multipliers estimate $\hat{\lambda}^k$ is computed by the solution of an LP subproblem.

The next result shows that the rate of local convergence of algorithm sSQPa is superlinear for degenerate problems [12, Theorem 7]. It is also shown that when (z^0, λ^0) is close to the optimal set $\mathcal{S}$, the initial call of procedure ID0 is the only one that is needed. The numerical experiments presented in this paper confirm these statements.

Theorem 1. *Suppose that assumption 1 holds. Then there exists a constant $\bar{\delta} > 0$ such that for any (z^0, λ^0) with $\delta(z^0, \lambda^0) \le \bar{\delta}$, the* `if` *condition in algorithm sSQPa is always satisfied and the sequence $\{\delta(z^k, \lambda^k)\}$ converges superlinearly to zero with q-order $1 + \sigma$.*

3.2 Constraint identification

The set $\mathcal{A}(z, \lambda)$ defined in (4) has been used to estimate the active inequality constraints in a neighborhood of a solution, see [2], [12]. In this estimation all inequality constraints with function values greater than or equal to $-\eta(z, \lambda)^\tau$ are considered in $\mathcal{A}(z, \lambda)$. Under the standing assumptions it can be shown [12, Theorem 3] that in a sufficiently small neighborhood of the solution the set $\mathcal{A}(z, \lambda)$ successfully estimates the active set $\mathcal{B}$.

Lemma 1. *Let assumption 1 holds. Then, there exists $\delta_1 > 0$ such that for all (z, λ) with $\delta(z, \lambda) \le \delta_1$, it holds $\mathcal{A}(z, \lambda) = \mathcal{B}$.*

As we have said before, it is also desirable to partition the set $\mathcal{A}(z, \lambda)$ in two sets: one corresponding to constraints that are candidate to be strong and the other containing the constraints that are candidates to be weak. To achieve this purpose it is convenient to solve the following LP subproblem [12] for a given subset $\hat{\mathcal{A}} \subset \mathcal{A}(z, \lambda)$ containing the candidates for weak active constraints.

$$\max_{\tilde{\lambda}_I, \tilde{\lambda}_E} \sum_{i \in \hat{\mathcal{A}}} \tilde{\lambda}_i$$

$$\text{subject to } \left\| \nabla\phi(z) + \sum_{i\in\mathcal{A}(z,\lambda)} \tilde{\lambda}_{I_i} \nabla g_{I_i}(z) + \sum_{i\in E} \tilde{\lambda}_{E_i} \nabla g_{E_i}(z) \right\|_\infty \leq \chi(z,\lambda,\tau),$$
$$\tilde{\lambda}_{I_i} \geq 0 \quad \text{for all} \quad i \in \mathcal{A}(z,\lambda), \quad \tilde{\lambda}_{I_i} = 0 \quad \text{for all} \quad i \in I \backslash \mathcal{A}(z,\lambda),$$

where $\chi(z,\lambda,\tau)$ is given by

$$\chi(z,\lambda,\tau) =$$
$$\max\left(\eta(z,\lambda)^\tau, \left\| \nabla\phi(z) + \sum_{i\in\mathcal{A}(z,\lambda)} \lambda_{I_i} \nabla g_{I_i}(z) + \sum_{i\in E} \lambda_{E_i} \nabla g_{E_i}(z) \right\|_\infty \right). \quad (5)$$

The multipliers $\hat{\lambda}_E$ corresponding to the equality constraints have no sign restriction in this LP subproblem.

In the following lines we restate the constraint identification procedure ID0 proposed in [12] based on the solution of LP subproblems of this type. The output of procedure ID0 is a partition of $\mathcal{A}(z,\lambda)$ into two sets $\mathcal{A}_+$ and $\mathcal{A}_0$: $\mathcal{A}_+$ contains the candidates for strong active constraints and $\mathcal{A}_0$ the candidates for weak active constraints.

Procedure ID0
Given τ, $\hat{\tau}$ with $0 < \hat{\tau} < \tau < 1$ and a point (z,λ), compute $\chi(z,\lambda,\tau)$ from (5), $\xi(z,\lambda,\tau,\hat{\tau}) = \max\big(\eta(z,\lambda)^{\hat{\tau}}, \chi(z,\lambda,\tau)\big)$, and $\mathcal{A}(z,\lambda)$ from (4). Define $\hat{\mathcal{A}}_{\text{init}} = \mathcal{A}(z,\lambda)\backslash\{i \mid \lambda_{I_i} \geq \xi(z,\lambda,\tau,\hat{\tau})\}$ and set $\hat{\mathcal{A}} \leftarrow \hat{\mathcal{A}}_{\text{init}}$.
Repeat

If $\hat{\mathcal{A}} = \emptyset$, stop with $\mathcal{A}_0 = \emptyset$ and $\mathcal{A}_+ = \mathcal{A}(z,\lambda)$.
Solve the LP (5) for $\tilde{\lambda}$ and set $\mathcal{C} = \{i \in \hat{\mathcal{A}} \mid \tilde{\lambda}_{I_i} \geq \xi(z,\lambda,\tau,\hat{\tau})\}$.
If $\mathcal{C} = \emptyset$

stop with $\mathcal{A}_0 = \hat{\mathcal{A}}$ and $\mathcal{A}_+ = \mathcal{A}(z,\lambda)\backslash\hat{\mathcal{A}}$;

else

set $\hat{\mathcal{A}} \leftarrow \hat{\mathcal{A}} \setminus \mathcal{C}$;

end(if)

end(repeat)

One can see that procedure ID0 will not be exited unless the set $\mathcal{C}$ is empty. The idea is to start with a superset of $\mathcal{A}_0$ given by $\hat{\mathcal{A}} = \hat{\mathcal{A}}_{\text{init}}$ and to remove iteratively from $\hat{\mathcal{A}}$ the constraints, stored in $\mathcal{C}$, that have been estimated to be strong by the LP subproblem (5).

It is shown in [12, Theorem 4] that the two subsets, $\mathcal{A}_+$ and $\mathcal{A}_0$, produced by procedure ID0 successfully estimate $\mathcal{B}_+$ and $\mathcal{B}_0$ in the vicinity of z^*.

Lemma 2. *Let assumption 1 holds. Then, there exists $\delta_2 > 0$ such that whenever $\delta(z,\lambda) \leq \delta_2$, procedure ID0 terminates with $\mathcal{A}_+ = \mathcal{B}_+$ and $\mathcal{A}_0 = \mathcal{B}_0$.*

3.3 Interior multipliers estimate

After the application of the constraint identification procedure ID0, the partition of $\mathcal{A}(z,\lambda)$ into the two subsets $\mathcal{A}_+$ and $\mathcal{A}_0$ is available. It is therefore possible to try to make the multipliers corresponding to the estimated strong active constraints in $\mathcal{A}_+$ as far from zero as possible. This is particularly desirable when solving NLP problems arising as subproblems in global optimization. Such interior multipliers estimate can be obtained by solving an LP subproblem of the following form (see [12]), adapted here to include the equality constraints:

$$
\begin{aligned}
\max_{\hat{t},\hat{\lambda}_I,\hat{\lambda}_E} \quad & \hat{t} \\
\text{subject to} \quad & \hat{t} \ \le \hat{\lambda}_{I_i} \quad \text{for all} \quad i \in \mathcal{A}_+, \\
& -\mu\, \mathsf{e} \le \nabla\phi(z) + \sum_{i\in\mathcal{A}_+} \hat{\lambda}_{I_i} \nabla g_{I_i}(z) + \sum_{i\in E} \hat{\lambda}_{E_i} \nabla g_{E_i}(z) \le \mu\, \mathsf{e}, \\
& \hat{\lambda}_{I_i} \ge 0 \quad \text{for all} \quad i \in \mathcal{A}_+, \qquad \hat{\lambda}_{I_i} = 0 \quad \text{for all} \quad i \in I \backslash A_+, \qquad (6)
\end{aligned}
$$

where e is a vector whose entries are all ones and the variables $\hat{\lambda}_E$ are unrestricted in sign.

Under the standing assumptions, it is shown in [12, Theorem 5] that the LP subproblem (6) is feasible and bounded in a sufficiently small neighborhood of the solution. Furthermore, the distance $\delta(z,\hat{\lambda})$ is bounded above by a multiple of $\delta(z,\lambda)^\tau$.

Lemma 3. *Let assumption 1 holds. Then, there exists $\delta_3 > 0$ such that for all (z,λ) with $\delta(z,\lambda) \le \delta_3$ the LP subproblem* (6) *is feasible, bounded, and its optimal objective is greater than or equal to*

$$\epsilon_{\lambda^*} = \max_{\lambda^*\in\mathcal{S}_\lambda} \ \min_{i\in\mathcal{B}_+} \lambda_i^*.$$

Furthermore, there exists $\beta > 0$ such that $\delta(z,\hat{\lambda}) \le \beta\delta(z,\lambda)^\tau$.

If there exists linear dependency of the gradients of the active constraints, then the vector of optimal Lagrange multipliers is not unique, and one can think of computing the multipliers with the largest possible size. This goal would be particularly relevant when we consider NLP problems arising as subproblems of an enumeration scheme applied to a global optimization problem. With this purpose in mind we have studied a few strategies. The one that seems most relevant consists of solving a second LP subproblem once (6) has been solved. The idea is to maximize the size of the multipliers in $\mathcal{A}_+$ while keeping the lower bound $\hat{t} > 0$ in the infinity norm that has been achieved by solving (6). So, after solving (6), one could solve the following LP subproblem:

$$\max_{\hat{\lambda}_I,\hat{\lambda}_E} \ \sum_{i\in\mathcal{A}_+} \hat{\lambda}_{I_i}$$

$$\begin{aligned}
\text{subject to} \quad & \hat{t} \leq \hat{\lambda}_{I_i} \quad \text{for all} \quad i \in \mathcal{A}_+, \\
& -\mu\, \mathsf{e} \leq \nabla\phi(z) + \sum_{i \in \mathcal{A}_+} \hat{\lambda}_{I_i} \nabla g_{I_i}(z) + \sum_{i \in E} \hat{\lambda}_{E_i} \nabla g_{E_i}(z) \leq \mu\, \mathsf{e}, \\
& \hat{\lambda}_{I_i} \geq 0 \quad \text{for all} \quad i \in \mathcal{A}_+, \qquad \hat{\lambda}_{I_i} = 0 \quad \text{for all} \quad i \in I \backslash \mathcal{A}_+.
\end{aligned}$$

The numerical experiments have shown, however, that there is not too much gain in solving this second LP subproblem. In fact, the LP subproblem (6) has produced in most instances multipliers whose size was quite close to the largest one.

4 Numerical experiments

We have developed a Matlab implementation of the stabilized SQP method with constraint identification (algorithm sSQPa) and tested it for a variety of degenerate problems. We used Matlab to solve the LPs and QPs that are needed by the sSQPa method.

We divide the numerical results into three major subsections. In subsection 4.1 we are concerned with the speed of local convergence of the method as well as with its global behavior without any globalization strategy. In subsection 3.2 we describe the numerical performance of procedure ID0 within algorithm sSQPa. Subsection 4.3 describes the use of the sSQPa method to find feasible points for feasible degenerate problems and least infeasible points for infeasible degenerate problems.

4.1 Problems with objective function

In this subsection, we consider 12 degenerate NLP test problems. For every problem we tested three different starting points with increasing distance to z_*, and for each case we plot $\log_{10} \|z - z_*\|$ *vs* iteration number. The performance of sSQPa for each problem is also shown by plotting $\log_{10} \eta(z, \lambda)$ *vs* iteration number for the farthest away starting point. When possible we compare the performance of sSQPa with other solvers for NLP.

The numerical results are obtained without any globalization strategy. The stopping criterion is $\eta(z, \lambda) \leq 10^{-8}$. We have computed the initial Lagrange multipliers λ^0 by solving the least-squares problem

$$\min_{\lambda_{\mathcal{A}_{\mathrm{ls}}}, \lambda_E} \left\| \nabla\phi(z^0) + \nabla g(z^0)^T_{\mathcal{A}_{\mathrm{ls}}} \lambda_{\mathcal{A}_{\mathrm{ls}}} + \nabla g(z^0)^T_E \lambda_E \right\|^2 \quad \text{s.t.} \quad \lambda_{\mathcal{A}_{\mathrm{ls}}} \geq 0, \tag{7}$$

where $\mathcal{A}_{\mathrm{ls}}$ is given by $\{i \in I \mid g_{I_i}(z^0) \geq -\epsilon_{\mathrm{ls}}\}$ with $\epsilon_{\mathrm{ls}} > 0$. In the implementation we used $\epsilon_{\mathrm{ls}} = 2.0$. The algorithm sSQPa has been designed to set $\lambda^0 \leftarrow \hat{\lambda}^0$. Our numerical experiments have however shown that the solution of (7) is a better choice for λ^0.

Other parameters have been set as follows: $\sigma = 0.95$, $\tau = 0.95$, and $\hat{\tau} = 0.85$. However we have used a different value for σ in the decrease condition

$\eta(z^k + \Delta z, \lambda^+) \le \left(\eta(z^k, \lambda^k)\right)^{1+\sigma/2}$ that appears in the sSQPa algorithm. We have tried several possibilities and conclude that a robust choice for σ in this condition is 0.05.

Test problem 1: The first test problem is HS113 [5], where degeneracy is due to lack of strict complementarity. This problem has 10 variables and 8 inequality constraints. The performance of sSQPa for three different starting points is given in figure 1. The numerical results show convergence from remote points and fast rate of local convergence. In figure 7 one can see the decrease in $\eta(z, \lambda)$. Table 1 shows that sSQPa is quite competitive with other solvers on this problem in number of iterations (starting from the standard point for this problem).

Fig. 1. Convergence from three different starting points for problems 1 and 2.

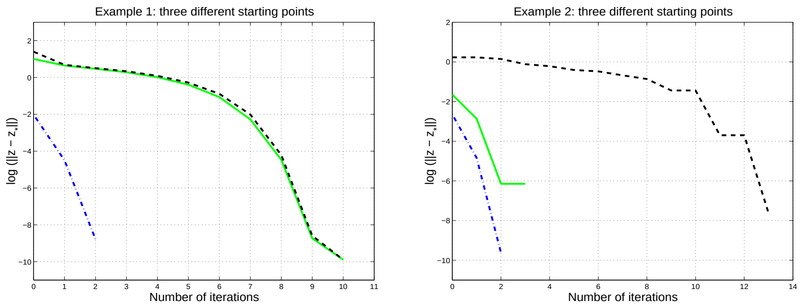


Table 1. Performance of different solvers on example 1.

Solver	NPSOL	SNOPT	NITRO	LOQO	sSQPa
Iterations	14	32	15	16	9
Objective	24.306	24.306	24.306	24.306	24.306

Test problem 2: The second test problem is from [2] and it is a modified version of HS46 [5]. Degeneracy in this problem is due to lack of strict complementarity. The problem has 5 variables and 3 inequality constraints. Figures 1 and 7 show convergence from remote points and fast rate of local convergence.

In this problem the effect of updating the multiplier using the interior multiplier $\hat{\lambda}$ can be nicely observed. In fact, we can see from table 2 that at the first, tenth, and twelfth iterations the decrease in $\eta(z, \lambda)$ has been poor. In this case sSQPa selects the `else` condition and updates λ^k by $\lambda^k \leftarrow \hat{\lambda}^k$, speeding up the rate of local convergence.

Test problem 3: The third test problem is also from [2] and it is rank deficient. This problem is a modified version of HS43 [5] and has 4 variables and 4 inequality constraints. We see that sSQPa has converged from the three starting points and has exhibited a fast rate of local convergence (see figures 2 and 7).

Fig. 2. Convergence from three different starting points for problems 3 and 4.

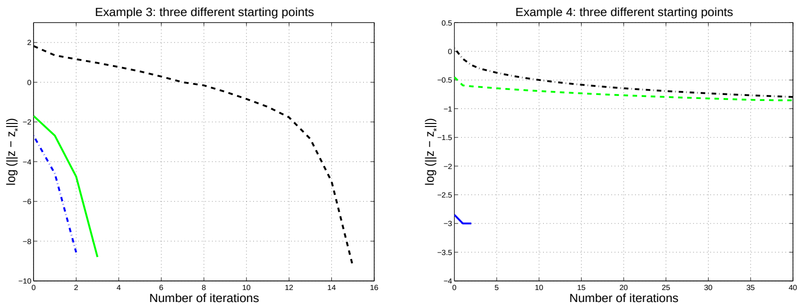


Table 2. Performance of sSQPa on example 2. cond(H^k) is the condition number of the Hessian of the QP, i_{id} is the number of iterations needed by procedure ID0.

k	$\phi(z^k)$	$\eta(z^k, \lambda^k)$	i_{id}	$\|\lambda^k\|$	$\|z^k - z^*\|$	cond(H^k)
0	3.33763e+00	7.87595e+00	0	2.14476e-01	1.70244e+00	$-$
1	3.33763e+00	7.87595e+00	2	5.01797e+00	1.70244e+00	6.82398e+15
2	4.68754e-01	2.43281e+00	0	8.36948e-02	1.34192e+00	$+\infty$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
9	2.74649e-07	1.03603e-03	0	9.99099e-04	3.81399e-02	3.11708e+03
10	2.74649e-07	1.03603e-03	1	0	3.81399e-02	1.20856e+05
11	2.81219e-10	2.02709e-05	0	1.04462e-05	2.18212e-04	$+\infty$
12	2.81219e-10	2.02709e-05	1	0	2.18212e-04	4.09199e+10
13	2.05082e-23	5.54501e-12	0	6.09971e-18	2.93470e-08	$+\infty$

Test problem 4: The fourth test problem is HS13 [5]. It is rank deficient and, furthermore, it does not satisfy the Karush-Kuhn-Tucker and Mangasarian-Fromovitz constraint qualifications. It has 2 variables, 1 inequality constraint, and 2 bound constraints.

In figure 2 one can see the convergence behavior from the three starting points and observe that sSQPa approaches a point different from z^* in all of them. Convergence from the first starting point is achieved in two iterations, while convergence for the other two starting points is very slow and the lowest value of $\eta(z, \lambda)$ is 10^{-3} in 40 iterations. Other solvers, among them NPSOL, SNOPT, and NITRO, exhibit a similar behavior for this problem by not converging to the solution.

Test problem 5: The fifth problem is a modified version of HS100 [5] and it is rank deficient. The problem has 7 variables and 5 inequality constraints and

Fig. 3. Convergence from three different starting points for examples 5 and 6.

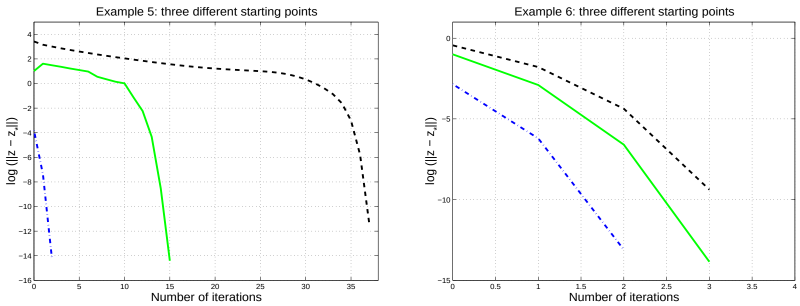
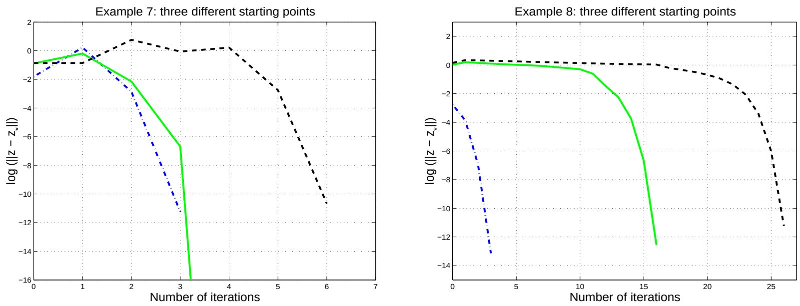


Fig. 4. Convergence from three different starting points for examples 7 and 8.



has the following form:

$$
\begin{array}{lll}
\min\ \phi(z) & = (z_1-10)^2+5(z_2-12)^2+z_3^4+3(z_4-11)^2 & \\
 & \quad +10z_5^6+7z_6^2+z_7^4-4z_6z_7-10z_6-8z_7 & \\
\text{s.t.}\ \ g_1(z) & = 2z_1^2+3z_2^4+z_3+4z_4^2+5z_5-127 & \leq 0, \\
g_2(z) & = 7z_1+3z_2+10z_3^2+z_4-z_5-282 & \leq 0, \\
g_3(z) & = 23z_1+z_2^2+6z_6^2-8z_7-196 & \leq 0, \\
g_4(z) & = 4z_1^2+z_2^2-3z_1z_2+2z_3^2+5z_6-11z_7 & \leq 0, \\
g_5(z) & = z_1^2+1.5z_2^4+0.5z_3+2z_4^2+2.5z_5-63.5 & \leq 0.
\end{array}
$$

Figures 3 and 7 show that the global and local performance of sSQPa is good for this problem.

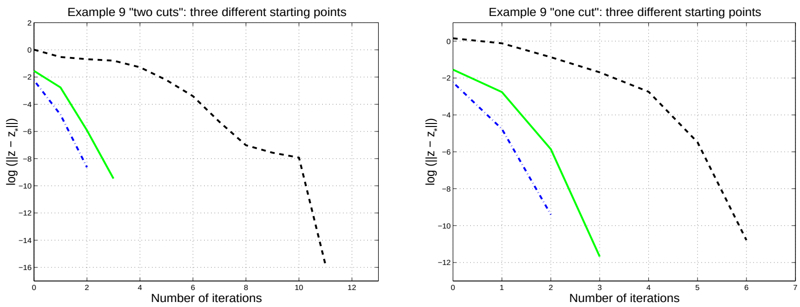
Test problem 6: This test problem is considered in [11] and it is rank deficient. It has 2 variables and 2 inequality constraints. There is nothing special to report; the global and local behavior of sSQPa for this problem are fine (see figures 3 and 7).

Test problem 7: The seventh problem is a quadratic problem with quadratic constraints introduced in [7]. The problem has 3 variables and 6 inequality constraints. The degeneracy is due to lack of strict complementarity. The performance of sSQPa for this problem is shown in figures 4 and 7 and was good both globally and locally.

Table 3. Performance of different solvers on example 8.

Solver	NPSOL	SNOPT	NITRO	LOQO	sSQPa
Iterations	2	4	17	23	16
Objective	1.0	1.0	1.000002	1.0	1.0

Fig. 5. Convergence from three different starting points for example 9.



Test problem 8: This test problem is HS32 [5] in which strict complementarity does not hold. It has 3 variables, 1 inequality constraint, 1 equality constraint, and 3 bound constraints. Global convergence and fast local convergence for this example can be confirmed from figures 4 and 7. In addition, table 3 gives a comparison between sSQPa and other solvers on this problem for the standard starting point associated with this problem.

Test problem 9: In this example we have modified problem HS40 by adding (one and two) cuts to the set of constraints. The cuts have made the problem rank deficient. In the case of two cuts, the problem has 4 variables, 2 inequality constraints, and 3 equality constraints:

$$\begin{array}{lll} \min \phi(z) & = -z_1 z_2 z_3 z_4 & \\ \text{s.t. } g_1(z) & = -z_1 z_2 z_3 z_4 + 0.25 & \leq 0, \\ g_2(z) & = -0.5 z_1 z_2 z_3 z_4 + 0.124999 & \leq 0, \\ g_3(z) & = z_1^3 + z_2^2 - 1 & = 0, \\ g_4(z) & = z_1^2 z_4 - z_3 & = 0, \\ g_5(z) & = -z_2 + z_4^2 & = 0. \end{array}$$

The one-cut case is generated by omitting the second constraint. The numerical behavior for this example is shown in figures 5 and 8.

Test problem 10: This problem is taken from [1] and it is rank deficient. It has 3 variables and 4 inequality constraints. We observe that sSQPa exhibits a

Fig. 6. Convergence from three different starting points for examples 10 and 11.

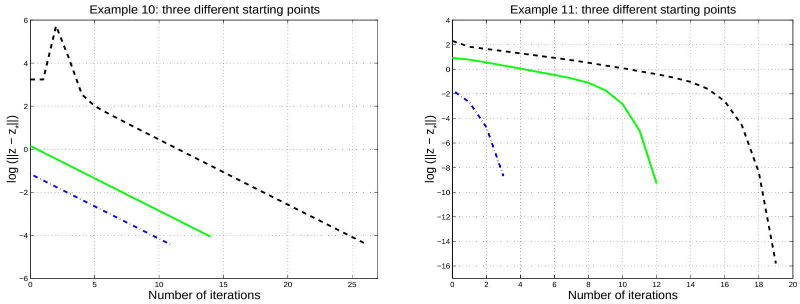


Table 4. Performance of different solvers on example 10.

Solver	DONLP2	FilterSQP	LANCELOT	LINF	LOQO	MINOS	SNOPT	sSQPa
$\|z_f - z^*\|$	1.5e-16	5.3e-09	8.7e-07	1.1e-08	1.6e-07	4.8e-06	3.4e-07	1.1e-08
Iters.	4	28	336	28	200	27	3	27

linear rate of local convergence for this example (see figure 8). In table 4 we have restated the comparison made in [1], listing the distance $\|z_f - z^*\|$ of the final point z_f to the optimal solution z^* and the number of iterations. It is shown in [1] that methods that are based on augmented Lagrangian functions do not perform well on this problem. To some extent the stabilized SQP method has the flavor of augmented Lagrangian methods since the quadratic model of the Lagrangian is augmented by another term involving the stabilization parameter, see the QP problem (3), and this might explain the not so good performance of the sSQPa algorithm on this problem.

Test problem 11: This problem is a modified version of problem HS43 and example 3, where an equality is included to the set of constraints. It is rank deficient and has the following form:

$$\begin{array}{lll}
\min \phi(z) & = z_1^2 + z_2^2 + 2z_3^2 + z_4^2 - 5(z_1 + z_2) - 21z_3 + 7z_4 & \\
\text{s.t. } g_1(z) & = z_1^2 + z_2^2 + z_3^2 + z_4^2 + z_1 - z_2 + z_3 - z_4 - 8 & \le 0, \\
g_2(z) & = z_1^2 + 2z_2^2 + z_3^2 + 2z_4^2 - z_1 - z_4 - 10 & \le 0, \\
g_3(z) & = 2z_1^2 + z_2^2 + z_3^2 + 2z_1 - z_2 - z_4 - 5 & \le 0, \\
g_4(z) & = -z_2^3 - 2z_1^2 - z_4^2 - z_1 + 3z_2 + z_3 - 4z_4 - 7 & = 0.
\end{array}$$

The numerical behavior of sSQPa for this problem can be seen from figures 6 and 8 and it is characterized by fast local convergence for the three different starting points.

Test problem 12: This test problem is a modified version of problem 6. It has an unique minimizer at the origin. It is also rank deficient and has the

Fig. 7. Performance of sSQPa on examples 1 to 8.

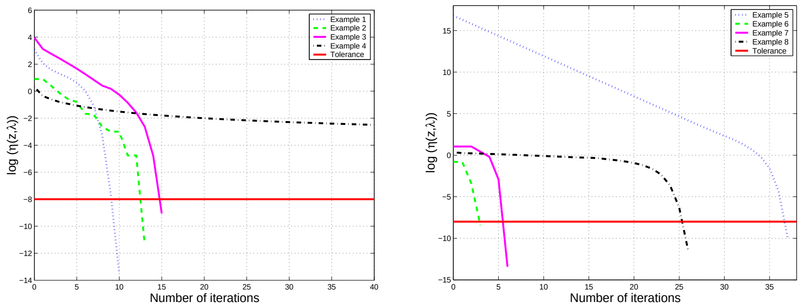
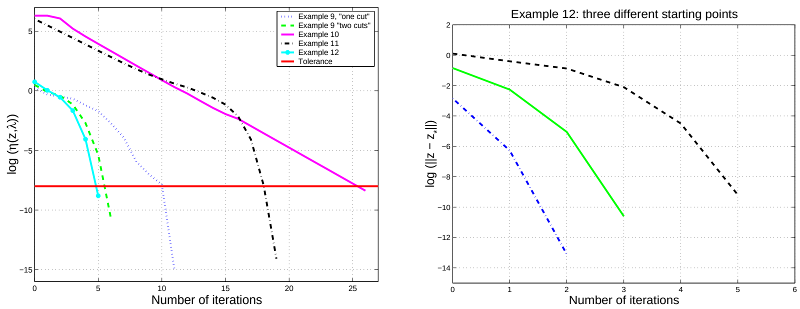


Fig. 8. Performance of sSQPa on examples 9 to 12; convergence from three different starting points for example 12.



following form:

$$\begin{aligned} \min\ \phi(z) &= z_1 \\ \text{s.t.}\ \ g_1(z) &= (z_1-2)^2 + z_2^2 - 4 \quad\ \ \le 0, \\ g_2(z) &= -(z_1-4)^2 - z_2^2 + 16 \le 0. \end{aligned}$$

The optimal set of the Lagrange multipliers is

$$\mathcal{S}_\lambda = \{(\alpha, \alpha/2) \mid \alpha \ge 0\},$$

which is clearly unbounded, implying that the MFCQ does not hold at the solution. The algorithm sSQPa exhibits a fast local rate of convergence on this example. Figure 8 shows the performance of sSQPa on this example.

4.2 Performance of procedure ID0

The aim of this subsection is to give some insight on the numerical behavior of procedure ID0 within algorithm sSQPa on the test problems introduced in subsection 4.1. The goal is to see how close to z^* the procedure ID0 is able to correctly identify the active constraints and its partition into strong and weak active constraints. We report in table 5 the value of $\|z - z^*\|$ from which these identifications is always correct.

With the exception of example 2, we used the results of the farthest away starting point. In fact, procedure ID0 does not detect the active set correctly in example 2 for the second and third starting points. In addition, procedure ID0 does not detect the active set correctly for any of the three starting points of example 7. In general, we can say that procedure ID0 does a good job identifying the active constraints and its partition into strong and weak.

Table 5. Detection of the correct active constraints and its partition into strong and weak.

Test Problem	$\mathcal{A}(z,\lambda)$	$\|z - z^*\|$	$\mathcal{A}_+$	$\mathcal{A}_0$	$\|z - z^*\|$
1	{1,2,3,4,5,7}	5.3e-01	$\mathcal{A}_+ = \mathcal{A}(z,\lambda)$	$\mathcal{A}_0 = \emptyset$	6.2e-05
2	{1,2,3}	1.4e-05	$\mathcal{A}_+ = \emptyset$	$\mathcal{A}_0 = \mathcal{A}(z,\lambda)$	2.4e-10
3	{1,3,4}	1.4e-01	$\mathcal{A}_+ = \{1,3\} \neq \mathcal{B}_+$	$\mathcal{A}_0 = \{4\} \neq \mathcal{B}_0$ [4]	1.4e-01
4	{1,3}	8.7e-01	$\mathcal{A}_+ = \mathcal{A}(z,\lambda)$	$\mathcal{A}_0 = \emptyset$	6.3e-01
5	{1,4,5}	1.1e+00	$\mathcal{A}_+ = \mathcal{A}(z,\lambda)$	$\mathcal{A}_0 = \emptyset$	9.7e-04
6	{1,2}	3.6e-01	$\mathcal{A}_+ = \mathcal{A}(z,\lambda)$	$\mathcal{A}_0 = \emptyset$	4.3e-05
7					
8	{2,3}	6.3e-01	$\mathcal{A}_+ = \mathcal{A}(z,\lambda)$	$\mathcal{A}_0 = \emptyset$	6.3e-01
9 "two cuts" [5]	{1}	1.2e-08	$\mathcal{A}_+ = \mathcal{A}(z,\lambda)$	$\mathcal{A}_0 = \emptyset$	1.2e-08
9 "one cut"	{1}	7.6e-01	$\mathcal{A}_+ = \mathcal{A}(z,\lambda)$	$\mathcal{A}_0 = \emptyset$	7.6e-01
10	{1,2,3,4}	4.3e-02	$\mathcal{A}_+ = \mathcal{A}(z,\lambda)$	$\mathcal{A}_0 = \emptyset$	2.1e-02
11	{1,3}	2.1e-01	$\mathcal{A}_+ = \mathcal{A}(z,\lambda)$	$\mathcal{A}_0 = \emptyset$	2.1e-01
12	{1,2}	6.4e-10	$\mathcal{A}_+ = \mathcal{A}(z,\lambda)$	$\mathcal{A}_0 = \emptyset$	6.4e-10 [6]

4.3 Problems without objective function

We have tested the ability of the sSQPa method to find feasible points for feasible degenerate problems and to find least infeasible points for infeasible degenerate problems. For this purpose the objective function and its derivatives were set to zero in the algorithm. No globalization scheme was used. We have tried four

[4] For the first and second starting points it is obtained a correct partitioning of $\mathcal{A}(z,\lambda)$ into $\mathcal{A}_+$ and $\mathcal{A}_0$, while for the third starting point g_4 is incorrectly in A_0 from the tenth iteration until the end.

[5] The second cut is perturbed a little so that its value at the solution is 10^{-7} (see problem 9). Procedure ID0 identifies the two constraints in $\mathcal{A}(z,\lambda)$ from the initial starting point until $\|z - z^*\| = 1.2e - 08$, and then excludes g_2 incorrectly from $\mathcal{A}(z,\lambda)$.

[6] At the first iteration $\mathcal{A}_+ = \mathcal{A}(z,\lambda) = \{1,2\}$, then in the following 4 iterations g_2 is excluded from $\mathcal{A}(z,\lambda)$ until the last iteration (the fifth iteration) at which g_2 returns to $\mathcal{A}(z,\lambda)$ with a value of $-3.04e-09$.

possibilities for the function $\eta(z,\lambda)$ used in the stopping criterion:

$$\eta_1(z,\lambda_I) = \left\| \begin{bmatrix} \min(\lambda_I, -g_I(z)) \\ g_E(z) \end{bmatrix} \right\|, \quad \eta_2(z,\lambda) = \left\| \begin{bmatrix} \nabla_z \mathcal{L}(z,\lambda) \\ \min(\lambda_I, -g_I(z)) \\ g_E(z) \end{bmatrix} \right\|,$$

$$\eta_3(z,\lambda) = \left\| \begin{bmatrix} g_{\mathcal{A}(z,\lambda)}(z) \\ g_E(z) \end{bmatrix} \right\|, \qquad \eta_4(z) = \left\| \begin{bmatrix} \max(g_I(z), 0) \\ g_E(z) \end{bmatrix} \right\|.$$

Note that $\eta_2(z,\lambda)$ is the one that is used in computing stationary points and that $\eta_4(z)$ represents a measure of the true feasibility.

We ran the 12 feasible problems introduced in section 4.1. In addition, we have designed 2 more infeasible problems that will be described later on in this section. For each of these problems we considered three different starting points. The stopping criterion was also $\eta(z,\lambda) \le 10^{-8}$. The initial multipliers have been obtained by

$$\lambda^0 = \begin{bmatrix} \max(0, g_I(z^0)) \\ g_E(z^0) \end{bmatrix}.$$

The overall results are given in table 6, where we report the average number of iterations and the number of wins out of 57 trials for each of the four types of $\eta(z,\lambda)$ given above. We observe that $\eta_1(z,\lambda)$ and $\eta_4(z)$ seem to be most efficient choices. With the exception of the second and third starting points of example 8, and of the three starting points of example 11, sSQPa with η_4 has successfully converged to feasible points.

Table 6. Performance of sSQPa for finding feasible or infeasible points.

	$\eta_1(z,\lambda_I)$	$\eta_2(z,\lambda)$	$\eta_3(z,\lambda)$	$\eta_4(z)$
Iteration average	10.18	10.63	12.95	9.23
Number of wins	50	48	25	52

In general, the feasible point computed changes with the starting point z^0 and the choice of $\eta(z,\lambda)$. The sSQPa method also worked well for the two cases of problem 9 and problem 12, with the particularity that the feasible point computed was found to be also stationary.

Next, we introduce two infeasible test problems, where at the least infeasible points the gradients of the nearby active constraints are linearly dependent.

Test problem 13: This test problem is a modification of both of problems 6 and 12, and has the following form:

$$\begin{aligned} g_1(z) &= (z_1 - (2+\epsilon))^2 + z_2^2 - 4 &\le 0, \\ g_2(z) &= -(z_1 - 4)^2 - z_2^2 + (4+\epsilon)^2 &\le 0. \end{aligned}$$

If ϵ is set to zero, the problem becomes feasible and rank deficient at the solution. The problem is infeasible for small positive values of ϵ.

Fig. 9. Convergence for example 13: $\epsilon = 10^{-4}$.

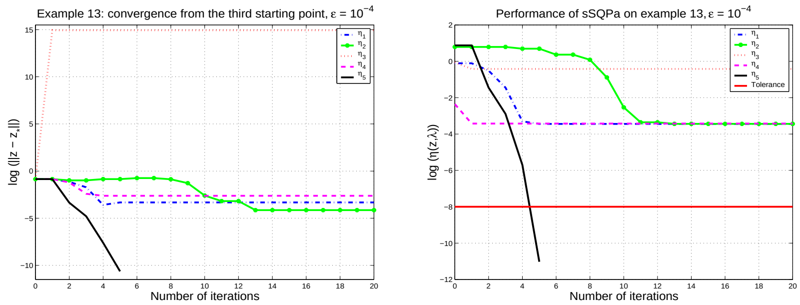
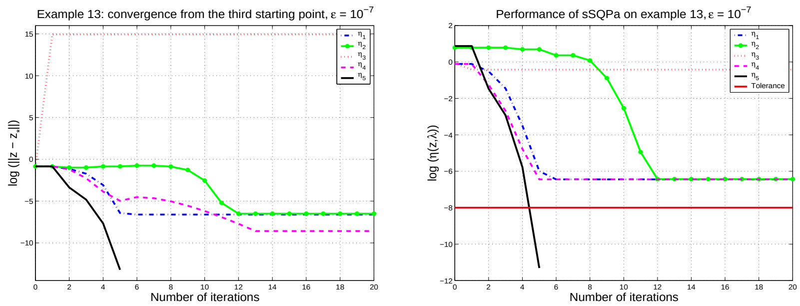


Fig. 10. Convergence for example 13: $\epsilon = 10^{-7}$.



In the computations, we have tested this example for $\epsilon = 10^{-4}$ and $\epsilon = 10^{-7}$. Again, the sSQPa method has been applied to these two instances with the objective function and its derivatives set to zero and without any globalization scheme.

The results are given in figures 9 and 10 for the two instances ($\epsilon = 10^{-4}$ and $\epsilon = 10^{-7}$) and for the four choices of $\eta(z, \lambda)$, in terms of the distance to the least-squares minimizer z^* of the constraints obtained by Matlab and also in terms of $\eta(z, \lambda)$. Here we tested another function $\eta(z, \lambda)$ for the stopping criterion:

$$\eta_5(z, \lambda) \;=\; \left\| \sum_{i \in \mathcal{A}(z,\lambda) \cup E} g_i(z) \nabla g_i(z) \right\| .$$

Test problem 14: Test problem 14 is also infeasible, defined by an hyperplane and a circle:

$$\begin{aligned} g_1(z) &= z_1 + \epsilon && \leq 0, \\ g_2(z) &= (z_1 - (1+\epsilon))^2 + z_2^2 - 1 && \leq 0. \end{aligned}$$

The results are given in figures 11 and 12. The main conclusion that we can draw from these two test problems is that the stabilized SQP with constraint

Fig. 11. Convergence for example 14: $\epsilon = 10^{-4}$.

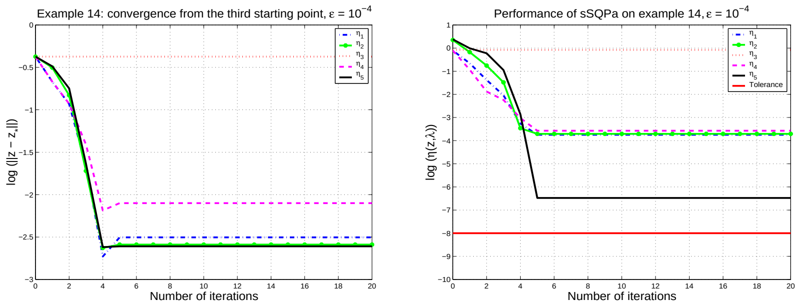
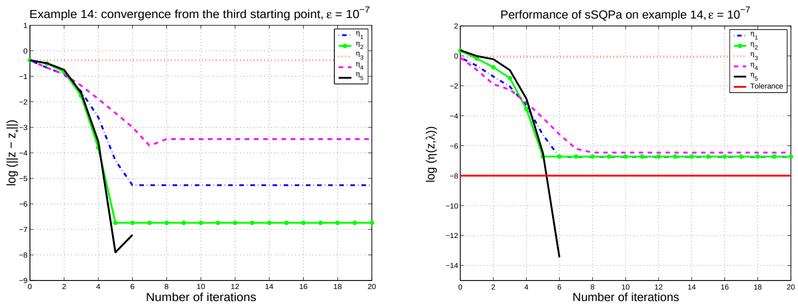


Fig. 12. Convergence for example 14: $\epsilon = 10^{-7}$.



identification (algorithm sSQPa) was quite effective to determine least infeasible points with nearby rank deficiency. Among the five measures of least infeasibility, $\eta_5(z, \lambda)$ seems to be the most efficient.

References

1. M. Anitescu, *Degenerate nonlinear programming with a quadratic growth condition*, SIAM J. Optim., 10 (2000) 1116–1135.
2. F. Facchinei, A. Fischer, and C. Kanzow, *On the accurate identification of active constraints*, SIAM J. Optim., 9 (1998) 14–32.
3. W. W. Hager, *Stabilized sequential quadratic programming*, Comput. Optim. and Appl., 12 (1999) 253–273.
4. M. Heinkenschloss, M. Ulbrich, and S. Ulbrich, *Superlinear and quadratic convergence of affine-scaling interior-point Newton methods for problems with simple bounds without strict complementarity assumption*, Math. Programming, 86 (1999) 615–635.
5. W. Hock and K. Schittkowski, *Test examples for nonlinear programming codes*, Lecture Notes in Economics and Math. Systems, 187, Springer Verlag, Berlin, 1981.
6. D.-H. Li and L. Qi, *A stabilized SQP method via linear equations*, Technical report, Mathematics Department, University of New South Wales, 2000.

7. A. V. de Miguel and W. Murray, *Generating optimization problems with global variables*, Technical Report TR/SOL 01–3, Management Science & Eng., Stanford Univ., 2001.
8. D. Ralph and S. J. Wright, *Superlinear convergence of an interior-point method despite dependent constraints*, Math. Oper. Res., 25 (2000) 179–194.
9. L. N. Vicente and S. J. Wright, *Local convergence of a primal–dual method for degenerate nonlinear programming*, Preprint 00–05, Department of Mathematics, University of Coimbra, 2000 (Revised May 2001).
10. S. J. Wright, *Modifying SQP for degenerate problems*, Preprint ANL/MCS–P699-1097, Argonne National Laboratory, 1998, (Revised June, 2000).
11. S. J. Wright, *Superlinear convergence of a stabilized SQP method to a degenerate solution*, Comput. Optim. and Appl., 11 (1998) 253–275.
12. S. J. Wright, *Constraint identification and algorithm stabilization for degenerate nonlinear programs*, Preprint ANL/MCS–P865-1200, Argonne National Laboratory, 2000 (Revised Nov. 2001).