Schedae Informaticae Vol. 21 (2012): 127–139
doi: 10.4467/20838476SI.12.008.0818

The Finite Termination Property of an Algorithm for Solving the Minimum Circumscribed Ball Problem

Andreas Gläser[1,2], Christian Grossmann[1], Ulrich Lunze[2]

[1]Institut für Numerische Mathematik,
Technische Universität Dresden,
D-01062 Dresden, Germany

[2]Institut für Produktionstechnik,
Westsächsische Hochschule Zwickau,
D-08056 Zwickau, Germany

Abstract. In this paper basic mathematical tasks of coordinate measurement are briefly described and a modified optimization algorithm is proposed. Coordinate measurement devices generate huge data set and require adapted methods to solve related mathematical problems in real time. The proposed algorithm possesses a simplified step size rule and finds the solution of the minimum circumscribed ball fitting after only a finite number The iteration is of the steepest descent type applied to the related distance function. But, in contrast to standard algorithms it uses a modified step size rule that takes into account the specific properties of the occurring objective function. This small difference in the code improves the performance of the algorithm and it enables real time use of the proposed method in coordinate measurement machines. The efficiency of the prosed algorithm will be illustrated by some typical examples.

Keywords: minimum circumscribed, maximum inscribed, minimum zone, geometrical form elements, coordinate measuring, mini–max problem, steepest descent.

1. Introduction

The monitoring of production processes requires to compare the manufactured work piece with its original design. This design, as a rule, bases upon a technical drawing or a CAD-model which uses simple geometrical elements like circles, balls, lines, planes and cylinders. Coordinate measuring machine (CMM), described e.g. in [5, 14], scan essential parts of the work piece and generate a huge number of m data points $p^i \in \mathbb{R}^n$, $n = 2$ (planar case) or $n = 3$ (spatial case), $i = 1, ..., m$ (m very large). The data points obtained from the measurements have to be converted into some simplified model that could be finally compared with the design parameters. The mathematical task is to find the parameters of these geometrical objects that fit the data points in an appropriate sense – Gauss fitting, minimum zone (mz) fitting, minimum circumscribed (mc) or maximum inscribed (mi). In [12] one can find a detailed description of the variety of such models used in coordinate measurement.

The Gauss fitting e.g. consists of the minimization of the quadratic distance from the data points to the wanted geometrical object. This is a standard smooth nonlinear optimization problem which can be solved sufficiently accurate within acceptably short time even for large numbers m of data.

However, often industrial standards of the coordinate measurement, like [2, 11], require to solve non-smooth problems. One of these tasks e.g. is the minimum zone fitting which requires to minimize the maximum of the absolute distance between the data points and the geometrical object. Further tasks of the coordinate measurement are the maximum inscribed and the minimum circumscribed ball problem.

These briefly sketched main tasks of the coordinate measurement lead to classical mini–max problems, but with huge data sets and with a highly specific structure of the objective function.

In the present paper we concentrate upon the minimum circumscribed ball problem, which is a convex mini–max problem.

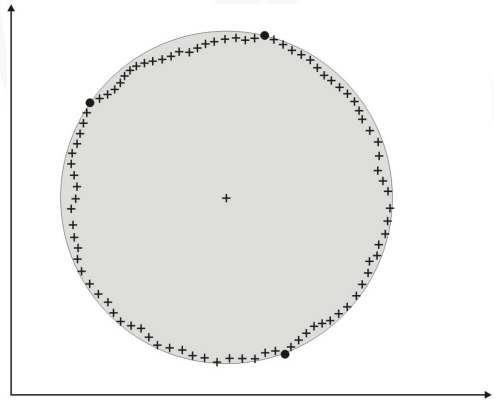

Figure 1. Example of a two dimensional bolt

A two dimensional example of data points of a bolt obtained by a measuring

machine is shown in Fig. 1.

The main requirement for data management in coordinate measurement machinery is to select or develop such optimization algorithms that perform the given tasks in real time. As already said, the simplest task is Gauss fitting. However, national and international standards, as described e.g. in Geometrical Dimensioning and Tolerancing ISO 1101 [11], BS 7172 [4] and in ASME Y14.5 [2], set rules that require also the mz-element, mi-element and mc-element. While for the Gauss fitting fast and sufficiently exact algorithms exist, this is not so for the mc, mi and mz problems. In [9, 15, 13] one can find some specific methods to tackle these tasks. But the given algorithms work often do not work fast and exact enough.

Unlike the ball already the most simple geometrical objects (cylinder, line, plane) lead to non-convex distance functions. In these cases a relaxed goal consists in finding at least a local solution of the mini–max problems (see [8, 6]). In most of the computed practical cases we observed that the obtained solutions were also global ones. But, so far no clear further characterization, e.g. by geometrical position of the data points, is known to ensure this in the general case.

There exist different possibilities to related mini-max problems to appropriate smooth approximate problems. In [9] e.g. the L_p-norm with $p >> 1$ is used instead of the maximum norm. However, the rounding errors in the computer numbers often lead to instabilities of the related techniques.

Another approach applies penalty-barrier techniques (see e.g. [7]) to transform the original mini–max problem into some smooth nonlinear optimization problem. In the considered cases in coordinate measurement applications the Hessian matrix often turned out to be non-definite because of the non-convexity. In such cases, some additional regularization techniques could be applied to overcome this difficulty. Such procedures were examined by the Physikalisch-Technische Bundesanstalt [1]. These algorithms often failed in case of non-convex distance functions. In certain applications these codes find global solutions of the mini–max problem. However, they require many iteration steps and additional adjustments which makes them rather slowly and inaccurately working.

We have to acknowledge that the mz, mi and mc problems still require to develop new algorithms which can handle the millions of data points produced by modern coordinate measuring machines. Such algorithms have to make advantage of the specific structure of the functions. The method we propose here is an attempt to improve the efficiency of solvers for one of the mathematical problems that arise in coordinate measurement.

2. Problem definition

Let $\langle \cdot, \cdot \rangle : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ denote some scalar product in $\mathbb{R}^n$ and let $|\cdot|$ denote the related norm. For given data points $p^i \in \mathbb{R}^n$, $i \in I := \{1...m\}$, $m \in \mathbb{N}$ the minimum

circumscribed ball fitting requires to determine the center $x \in \mathbb{R}^n$ of the ball

$$B_r(x) := \{y \in R^m : \langle x-y, x-y\rangle \le r^2\},\ \},\quad r \ge 0$$

with the minimal radius r that contains all data points p^i, $i \in I$. This yields the optimization problem

$$r^2 \to \min\,! \qquad \text{subject to} \quad p^i \in B_r(x),\ i \in I, \quad x \in \mathbb{R}^n, \quad r \ge 0. \tag{1}$$

Let us abbreviate

$$F(x) := \max_{i\in I} f_i(x) \quad \text{with} \quad f_i(x) := \frac{1}{2}\left\langle p^i - x, p^i - x\right\rangle,\ i \in I. \tag{2}$$

Then (1) is equivalent to

$$F(x) \to \min\,! \qquad \text{subject to} \quad x \in \mathbb{R}^n. \tag{3}$$

This is a classical mini–max problem (compare [6]), but with a specific structure of the involved functions f_i. This specific structure will be exploited to construct an algorithm that solves the problem in a finite number of steps. We notice that (2) possesses a unique optimal solution. This follows from the boundedness and the closedness of the level sets $W(x) := \{y \in \mathbb{R}^n : F(y) \le F(x)\}$ and of the strict convexity of F.

As usual in mini–max optimization the problem (2) can be equivalently described by

$$\mu \to \min! \quad \text{s.t.} \quad f_i(x) - \mu \le 0,\ i \in I. \tag{4}$$

Let $I_0(x)$ denote the set of active constraints at the point x, i.e.

$$I_0(x) := \{i \in I : f_i(x) = F(x)\}.$$

Taking (4) into account we obtain the following optimality criterion.

Lemma 1. *Problem (2) possesses a unique solution $\bar{x} \in \mathbb{R}^n$ and $\bar{x}$ is a solution of (2) if and only if*

$$0 \in \operatorname{conv}\{\nabla f_i(\bar{x})\}_{i\in I_0(\bar{x})} \tag{5}$$

and equivalently

$$\bar{x} \in \operatorname{conv}\left\{p^i\right\}_{i\in I_0(\bar{x})}. \tag{6}$$

Proof. Trivially some $\tilde{x} \in \mathbb{R}^n$ and $\tilde{\mu}$ exist such that

$$f_i(\tilde{x}) - \tilde{\mu} < 0,$$

i.e. $(\tilde{x}, \tilde{\mu})$ forms a Slater point and the KKT-conditions are necessary and sufficient for optimality. Taking the structure of the problem (4) into account then the related KKT-conditions have the form. With

$$\nabla f_i(x) = x - p^i$$

condition (5) is equivalent to (6). □

A simple property of $I_0(\bar{x})$ at the solution $\bar{x} \in \mathbb{R}^n$ is described by

Lemma 2. *If* $\{p^i\}$ *contains at least two different points then* $\text{card}\,(I_0(\bar{x})) > 1$.

Proof. At least one constraint is active at the optimal $\bar{x}$ solution of (1). Assume that $\text{card}\,(I_0(\bar{x})) = 1$, i.e. $I_0(\bar{x}) = \{j\}$ with some $j \in I$. The KKT-conditions (6) imply

$$\bar{x} - p^j = 0.$$

This yields

$$F(\bar{x}) = 0 = \max_{i \in I} f_i(\bar{x}) \geq F(x) \quad \forall x \in \mathbb{R}^n.$$

Since $F(x) \geq 0\ \forall x \in R^n$ this implies $F(x) \equiv 0$ which cannot hold if at least two different data points exist. □

Next, we study the local behavior of F. As the maximum of a finite number of smooth function at any $x \in \mathbb{R}^n$ we obtain the **steepest descent direction** (compare [6]) via the convex problem

$$\max_{i \in I_0(x)} \langle \nabla f_i(x), d \rangle \to \min! \quad \text{s.t.} \quad \langle d, d \rangle \leq 1. \tag{7}$$

To guarantee the finite termination property of our later described algorithm we assume that the direction finding problem (7) can solved exactly by some finite method. An appropriate technique is described later.

Remark 1. *There exist various algorithms (cf. [6, 8]) to solve the optimization problem (7). We notice that the Lagrange dual problem of (7) is just to find the shortest distance to a closed convex polyhedron. We refer e.g. to [3] for algorithms to solve problems of this type.*

In the case $x \neq \bar{x}$, i.e. if x is not the solution of (2), we have $|\bar{d}| = 1$ and

$$F(x + \alpha \bar{d}) < F(x) \quad \text{for} \quad \alpha > 0, \quad \text{sufficiently small.}$$

Now we study the behavior of the function F at the point $\hat{x} = x + \alpha \bar{d}$. Let denote

$$\delta := \max_{i \in I_0(x)} \left\langle \nabla f_i(x), \bar{d} \right\rangle < 0 \tag{8}$$

and let $\alpha > 0$ be sufficient small. Further denote

$$\hat{\delta} := \max_{i \in I_0(x)} \left\langle \nabla f_i(\hat{x}), \bar{d} \right\rangle = \max_{i \in I_0(x)} \left\langle \nabla f_i(x + \alpha \bar{d}), \bar{d} \right\rangle. \tag{9}$$

The property $\nabla f_i(x + \alpha\, d) = \nabla f_i(x) + \alpha\, d$ implies

$$\hat{\delta} = \max_{i \in I_0(x)} \left\langle \nabla f_i(x), \bar{d} \right\rangle + \alpha \left\langle \bar{d}, \bar{d} \right\rangle \overset{\langle \bar{d}, \bar{d} \rangle = 1}{=} \delta + \alpha \quad > \quad \delta. \tag{10}$$

For sufficiently small step sizes $\alpha > 0$ holds

$$I_0(x + \alpha \bar{d}) \subseteq I_0(x)$$

and we obtain

$$\max_{i \in I_0(x + \alpha \bar{d})} \left\langle \nabla f_i(x + \alpha \bar{d}), \bar{d} \right\rangle = \delta + \alpha \tag{11}$$

(compare (9),(10)).

Lemma 3. *Let $x \neq \bar{x}$, let $\bar{d}$ be the related solution of (7) and chose the step size*

$$\alpha = -\delta,$$

where δ is defined by (7). Further let

$$I_0(x + \alpha\bar{d}) \subseteq I_0(x), \tag{12}$$

i.e. no new constraints become active, then is $\bar{x} := x + \alpha\bar{d}$ the solution of (2).

Proof. With (12) we have

$$f_j(x + \alpha\bar{d}) < \max_{i \in I_0(x)} f_i(x + \alpha\bar{d}) \quad \forall j \in I \setminus I_0(x).$$

We use now (10) and obtain

$$\hat{\delta} := \max_{i \in I_0(x)} \left\langle \nabla f_i(x + \alpha\bar{d}), \bar{d} \right\rangle = \delta + \alpha = 0$$

and (11) implies

$$0 \in \operatorname{conv}\{\nabla f_i(x + \alpha\bar{d})\}_{i \in I_0(x + \alpha\bar{d})}.$$

□

We notice that

$$\tilde{y} := x - \delta\bar{d} \tag{13}$$

is the unique solution of the optimization problem

$$F_{I_0(x)}(y) := \max_{i \in I_0(x)} f_i(y) \to \min!$$

Next, we study the behavior of the functions f_i for $i \in I \setminus I_0(x)$. In particular we are interested which of the indices of $I \setminus I_0(x)$ cannot be contained the new index set $I_0(x + \alpha\bar{d})$.

Lemma 4. *Let $x \in \mathbb{R}^n$, let $\bar{d}$ be the related solution of (7) and consider indices which satisfy*

$$i \in I \setminus I_0(x) \quad \textit{and} \quad \left\langle \bar{d}, \nabla f_i(x) \right\rangle \leq \delta.$$

Then we have

$$i \notin I_0(x + \alpha\bar{d})$$

and

$$\left\langle \nabla f_i(x + \alpha\bar{d}), \bar{d} \right\rangle \leq \delta + \alpha. \tag{14}$$

Proof. We obtain (14) directly from (10). Let $i \in I \setminus I_0(x) : \left\langle \nabla f_i(x), \bar{d} \right\rangle \leq \delta$ then the structure of F implies

$$F(x + \alpha\bar{d}) = F(x) + \alpha \max_{i \in I_0(x)} \left\langle \nabla f_i(x), \bar{d} \right\rangle + \frac{1}{2} \left\langle \bar{d}, \bar{d} \right\rangle.$$

Now, we have

$$\begin{aligned} F(x + \alpha\bar{d}) &= F(x) + \alpha\delta + \tfrac{1}{2} \left\langle \bar{d}, \bar{d} \right\rangle \\ &> f_i(x) + \alpha \left\langle \nabla f_i(x), \bar{d} \right\rangle + \tfrac{1}{2} \left\langle \bar{d}, \bar{d} \right\rangle \\ &= f_i(x + \alpha\bar{d}). \end{aligned} \tag{15}$$

□

3. Algorithm

Algorithm 1 Minimum Circumscribed Ball

1: SET $x^0 := \frac{1}{m}\sum_{i=1}^{m} p^i$
2: $k := 0$;
3: **while** TRUE **do**
4: $\mu^k := \max_{i\in I} f_i(x^k)$;
5: $I_0^k := \left\{ i \in I : f_i(x^k) = \mu^k \right\}$;
6: SOLVE the optimization problem
$\max_{i\in I_0} \left\langle \nabla f_i(x^k), d \right\rangle \to \min!$ s.t. $\langle d, d\rangle \leq 1$
such that d^k is unique solution and
SET $\delta_k := \max_{i\in I_0} \left\langle \nabla f_i(x^k), d^k \right\rangle$
7: **if** $d^k == 0$ **then**
8: **return** $\bar{x} := x^k$;
9: **end if**
10: $J^k := \left\{ i \in I : \left\langle \nabla f_i(x^k), d^k \right\rangle > \delta^k \right\}$
11: **if** $J^k == \emptyset$ **then**
12: **return** $\bar{x} := x^k - \delta_k d^k$;
13: **end if**
14: FIND $\alpha_k > 0$ such that
$\max_{i\in I_0^k} f_i(x^k + \alpha_k d^k) = \max_{i\in J^k} f_i(x^k + \alpha_k d^k)$
15: SET $x^{k+1} := x^k + \alpha_k d^k$;
16: SET $k := k + 1$;
17: **end while**

The difference to the standard steepest descent algorithm lays in the ste size rule in line 14. Since the considered optimization problem has the discussed simple structure the instruction of line 14 of the algorithm is to find $\alpha_k > 0$ such that

$$F(x^k) + \alpha_k \delta_k = \max_{i\in J^k} f_i(x^k) + \alpha_k \left\langle \nabla f_i(x^k), d^k \right\rangle. \tag{16}$$

This is just a simple linear equation. We underline that in each iteration step a new active constraint is found. So the index set I_0^k will be changed in each sweep of the algorithm.

Lemma 5. *The **Algorithm 1** is well defined and generates a sequence* $\{x^k\}$ *with* $F(x^k) > F(x^{k+1})$. *The method terminates with the optimal solution* $\bar{x}$ *after finite number of steps according to the criteria on line 8 and line 12, respectively.*

Proof. The property $F(x^k) > F(x^{k+1})$ is guaranteed by line 15 with d^k as steepest descent direction and $\alpha_k > 0$ according to line 14.

If x^k is the solution of (2) then the algorithm terminates at line 8. Now, we turn to the finite termination property.

Only indices $i \in J^k \subset I \setminus I_0^k$ may satisfy the condition

$$\max_{i \in I_0^k} f_i(x^k + \alpha_k d^k) = \max_{i \in I \setminus I_0^k} f_i(x^k + \alpha_k d^k)$$

(compare Lemma 4) which implies

$$f_i(x^k + \alpha_k d^k) \leq F(x^k + \alpha_k d^k) \quad \forall i \in I \setminus J^k.$$

In the case $J^k = \emptyset$ we can apply Lemma 3 and obtain $-\delta_k$ as the optimal step length and $\bar{x} := x^k - \delta_k d^k$ (line 12) as solution of (2).

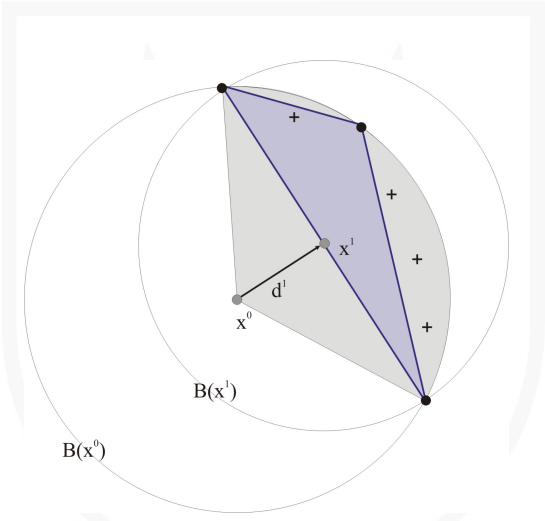

Figure 2. Illustration of the case $J^k = \emptyset$

Otherwise, according to line 14 we construct new additional active constraints with indices i which belong to J^k. This implies that some $i \in J^k$ exists with

$$f_i(x^k + \alpha_k d^k) = F(x^k + \alpha_k d^k).$$

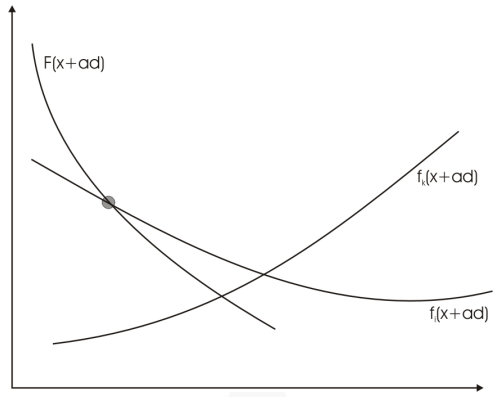

Figure 3. Illustration of the case $f_i(x^k + \alpha_k d^k) = F(x^k + \alpha_k d^k)$

Further, for some $i \in I_0^k$ we have

$$f_i(x^k + \alpha_k d^k) = F(x^k + \alpha_k d^k).$$

This yields

$$I_0^{k+1} \neq I_0^k \quad \wedge \quad I_0^{k+1} \subset \quad I_0^k \cup J^k.$$

Let denote

$$\hat{x}^k := x^k - \delta^k d^k$$

as unique solution of the optimization problem

$$F_{I_0^k}(x) \to \min!$$

(compare (13)) and

$$\mu^k := F_{I_0^k}(\hat{x}^k) < F(\hat{x}^k).$$

$J^k \neq \emptyset$ implies $d^{k+1} \neq d^k$ and $\hat{x}^k \neq \hat{x}^{k+1}$ where $\hat{x}^{k+1}$ is the unique solution of $F_{I_0^{k+1}}$. This leads to

$$F_{I_0^k}(\hat{x}^k) < F_{I_0^k}(\hat{x}^{k+1})$$

because $\hat{x}^k$ is unique solution of $F_{I_0^k}$ and the active constraints from I_0^k are contained in I_0^{k+1}. The set $\{\mu^k\}$ has the property $\mu^k < \mu^{k+1}$ for all iteration steps k.

Then we obtain

$$I_0^k \neq I_0^{k+j} \quad \forall j \in \mathbb{N}$$

directly from the property of the set $\{\mu^k\}$. The index set I consists of a finite number of indices. Hence, there exists only a finite number of different index sets I_0^k.

□

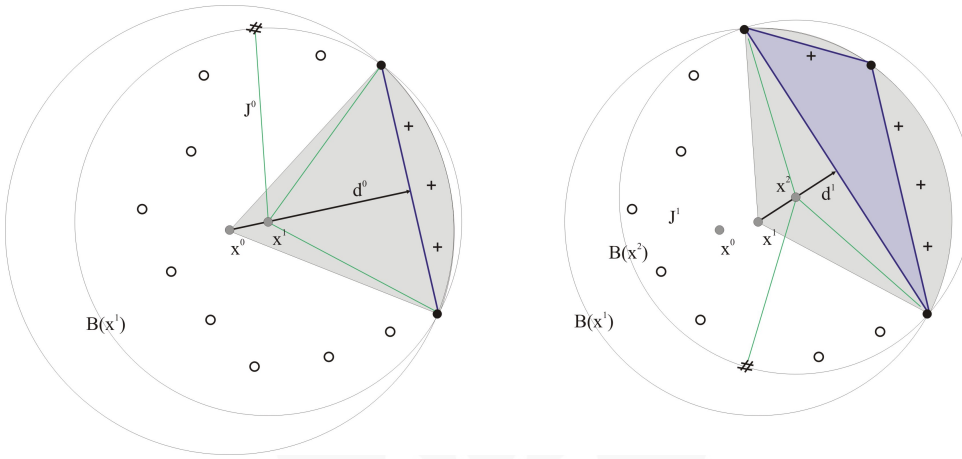

Figure 4. Geometrical interpretation of one step of Algorithm 1

The point marked by # in Fig. 4 is the additional data point p^i, $i \in J^k$ with $i \in I_0^{k+1}$. The data points related to the index set J^k are marked by o, while the black dots mark the data points with indices in the set I_0^k.

The dots with + denote the data points p^i with $\left\langle d^k, \nabla f_i(x^k) \right\rangle \geq \delta^k$. From Step 1 to Step 2 we can see the change from J^0 to J^1 and the property $J^1 \subset J^0$.

4. Computational experiments

The proposed algorithm for the minimum circumscribed ball problem has been tested at different academic examples as well as in real time applications in coordinate measurement machines. In particular, the finite termination property of the algorithm guaranteed an excellent performance of the implemented code. Further, comparisons were done with different classical optimization codes, e.g. the MATLAB sqp-algorithm.

Here we report results obtained for the following about three test problems:

Examples IDEAL. The data points are randomly distributed on the surface of the ball in $\mathbb{R}^3$ with the given center and radius. In the reported two cases we used 1000 data points and 100000 data points, respectively.

Examples RANDOM. In these examples in $\mathbb{R}^3$ the data data points simulate some typical production errors, i.e. the points are randomly distributed with a distance from the center in a rather narrow interval. In the experiments we used $[9.75, 10.25]$

and $[19.75, 20.25]$, respectively.

Examples CIRCLE. These data points simulate some typical production errors from a measured circle (planar example). As intervals for radius we have taken $[9.75, 10.25]$.

In Tab. 1 are listed the results obtained for these examples. As computational methods methods we included Algorithm 1, the steepest descent algorithm with the Cauchy step size (see e.g. [8]) and an SQP method (MATLAB code sqp) applied to 4. The integer k indicates the maximal number of complete sweeps of the considered algorithms. The column labeled F indicates the total number of simultaneous calculations of **all** m functions f_i.

Table 1. Computational results for the considered examples (I = Ideal, R = Random, C = Circle)

Example	Algorithm 1			Cauchy step size			MATLAB sqp		
	k	F calls	time	k	F calls	time	k	F calls	time
I1000	5	13	0.016	9	281	0.047	3	33	17.76
I100000	6	15	0.389	6	182	3.479	3	33	412.36
R1000	10	27	0.019	8	421	0.063	5	51	6.08
R100000	11	39	0.842	7	346	6.257	4	42	15271.00
C500	7	19	0.008	5	284	0.095	9	93	4.16
C5000	5	13	0.156	3	166	0.139	12	121	8141.5

5. Conclusions

The finite termination property proved for Algorithm 1 and the simplified step size rule therein makes the method as a good candidate for real time calculations as required in modern coordinate measurement machines. The implemented code is already included in firm ware in coordinate measurement machines.

In the considered case of the minimal circumscribed ball only points p^i, $i \in I$ relate to active constraints if there lay on the boundary of the optimal ball or circle, respectively. Since our Algorithm 1 is of active set type this implies that the number of iteration steps depends upon the number of data points on the boundary.

The algorithm with Cauchy step size required in case of the example CIRCLE5000 only 3 iteration sweeps, but the line search in this algorithm resulted in 50 additional function evaluations. As indicated above, we have included the MATLAB code sqp into the given comparison. The related algorithm turned out to be rather slow in case of the examples RAND100000 and CIRCLE5000. This is caused by the required approximation of all dual variables and inclusion of all constraints into the generated quadratic subproblems. A direct comparison of the

cpu time is difficult since Algorithm 1 was implemented in C++. If one considers the increase of computer time for different model sizes then the sqp code showed a non-linear increase. The proposed Algorithm 1 does not show such an increase of cpu time if the model size is increased.

Acknowledgements. We thank the referees for their serious critics of an earlier draft of the paper. The research of first author thanks the Dr.-Ing. Siegfried Werth Foundation for funding of his research by a scholarship.

6. References

[1] Anthony G.T., Bittner B., Butler B.P., Cox M.G., Drieschner R., Elligsen R., Forbes A.B., Gross H., Hannaby S.A., Harris P.M., Kok J.; *Chebyshev reference software for the evaluation of coordinate measureing machine data*, NPL Teddington – UK, PTB Braunschweig – Germany, CWI Amsterdam – Netherlands, 1993.

[2] ASME Y14.5-2009; *Dimensioning and Tolerancing*, American Society and Mechanical Engineers, 2009.

[3] Bobrow J.E.; *A Direct Minimization Approach for Obtaining the Distance between Convex Polyhedra*, Int. J. of Robotics Res. 11, 1992, p. 5.

[4] BS 7172; *British Standard guide to assessment of position, size and departure from nominal form of geometric feature.*

[5] Christoph R., Neumann H.J.; *Multisensor-Koordinatenmesstechnik*, sv corporate media, 2006.

[6] Dem'janov W.F., Malozemov W.N.; *Introduction to Minimax*, Dover Publication, New York 1990.

[7] Grossmann C., Kaplan A.A.; *Strafmethoden und modifizierte Lagrange Funktionen in der nichtlinearen Optimierung*, Teubner, Leipzig 1979.

[8] Grossmann C., Kleinmichel H., Vetters K.; *Minimaxprobleme und nichtlineare Optimierung*, Math. Meth. OR 20, 1976, pp. 23–38.

[9] Goch G., Lübke K.; *Tschebyscheff approximation for the calculation of maximum inscribed/minimum circumscribed geometry elements and form deviations*, CIRP Annals – Manufacturing Technology 57 I. 1, 2008, pp. 517–520.

[10] Hadrich M.; *Ein Verfahren zur Tschebyscheff-Approximation von Formelementen der Koordinatenmesstechnik*, Measurement, vol. 12, 2004, pp. 337–344.

[11] ISO 1101; *Geometrical product specification (GPS) - geometrical tolerating - tolerating of form, direction, position und run*, Beuth Verlag, Berlin 2006.

[12] Koch W., Lotze W., Lunze U.; *Paarungslehrung nach dem Taylorschen Grundsatz durch nichtlineare Optimierung – Praxiseinsatz in der modernen Fertigungstechnik*, QZ Qualität und Zuverlässigkeit 36(4), 1991, pp. 219-224.

[13] Lotze W.; *General solution for Tsebyshev approximation of forn elements in coordinate measurement*, Measurement 12, 1994, pp. 337–344.

[14] The MultiSensor; The International Newsletter of Werth Messtechnik. http://www.werth.de/en/navigation/werth-newsletter.html

[15] Späth H., Watson G.A.; *Smallest circlumscribed, largest inscribed and minimum zone circle or sphere via sequential linear programming*, Math. Comm. 6, 2001, pp. 29–38.