A Rank 1 SDP Approach for the Sensor Network Localization Problem

Inaugural-Dissertation

zur

Erlangung des Doktorgrades der

Mathematisch-Naturwissenschaftlichen Fakultät

der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Ania López

aus Parchim

Dezember 2008

Aus dem Mathematischen Institut
der Heinrich-Heine-Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Referent: Prof. Dr. Florian Jarre

Korreferent: Prof. Dr. Helmut Ratschek

Tag der mündlichen Prüfung: 12.12.2008

Lina,
te dedico esta tesis,
por ser tú el resultado más bonito
de estos años de estudios de doctorado.

Abstract

This thesis analyzes the problem of localizing sensors in wireless ad hoc networks. Wireless ad hoc networks can be found in several practical applications, among which the most known is the climatic monitoring. Every sensor in such a network has the ability to register specific factors. These can be forwarded (if necessary) to other sensors within the network until reaching a so-called anchorpoint. Due to this special form, these networks are applied for example to predict forest fires, earthquakes or climatic changes.

The Sensor Network Localization Problem (SNLP) is a NP-hard problem and is solved by the approximate solution of a non-linear least squares problem in practice.

Leaning on the work of Ye et al. [5] a semidefinite program (SDP) formulation is proposed:

$$\min\{C \bullet X \mid \mathcal{A}(X) = b,\ X \in \mathcal{S}^n_+\},$$

where the variable is $X \in \mathcal{S}^n$, the space of real symmetric matrices of dimension n. The vector $b \in \mathbb{R}^m$, the linear mapping $\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^m$ and the matrix $C \in \mathcal{S}^n$ are given. The cone $\mathcal{S}^n_+$ is the set of all real symmetric positive semidefinite matrices of dimension n, i.e. $\mathcal{S}^n_+ := \{M \in \mathcal{S}^n \mid S \succeq 0\}$. The notation $C \bullet X$ denotes the inner product of the symmetric matrices C and X and is given by

$$C \bullet X = tr(C\,X) = \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij}\,X_{ij}.$$

This SDP formulation is a relaxation of the original problem (in the model proposed in this thesis, the exact solution of the original problem has to fulfil additionally a rank 1 condition). The great advantage of employing SDPs is their good numerical treatment via interior point methods.

For special cases (so-called "uniquely" localizable networks) is shown, that the presented SDP provides the exact solution for the SNLP.

The interesting problem variant, which is common in practice, incorporates measurement errors due to interferences between the sensors in a network. This yields a disturbed localization problem. In this context a local error bound for the disturbed solution is presented.

To improve the solution of the relaxed problem, several heuristics are presented and compared numerically. The so-called *Curvature Descent* method achieves global and local quadratic convergence.

Zusammenfassung

In dieser Arbeit wird das Problem der Lokalisierung von Sensoren in drahtlosen (Ad Hoc) Netzwerken untersucht.
Drahtlose Ad Hoc Netzwerke finden sich in sehr vielen Anwendungen. Die bekanntesten sind die klimatischen Beobachtungen. Dabei hat jeder Sensor die Möglichkeit bestimmte Bedingungen zu registrieren und diese bei Bedarf an weitere Sensoren (bis zum Erreichen eines sog. Ankerpunktes) weiterzugeben. So werden solche Netzwerke z.B. zur Vorhersagung von Waldbränden, Erdbeben oder zur allgemeinen Registrierung klimatischer Veränderungen eingesetzt.

Das SNLP (Sensor Network Localization Problem) ist NP-vollständig und wird in der Praxis über die Annäherung an die Lösung des zugehörigen nicht-linearen least squares Problems gelöst.

Anlehnend an die Arbeit von Ye [5] wird eine Modellierung als Semidefinites Programm (SDP) der Form

$$\min\{C \bullet X \mid \mathcal{A}(X) = b,\ X \in \mathcal{S}^n_+\},$$

beschrieben. Hierbei ist die Variable $X \in \mathcal{S}^n$, dem Raum der reellen symmetrischen Matrizen der Dimension n. Der Vektor $b \in \mathbb{R}^m$, die lineare Abbildung $\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^m$ und die Matrix $C \in \mathcal{S}^n$ sind gegebene Parameter. Der Kegel $\mathcal{S}^n_+$ ist der Raum aller reellen symmetrischen positiv semidefiniten Matrizen der Dimension n, d.h. $\mathcal{S}^n_+ := \{M \in \mathcal{S}^n \mid S \succeq 0\}$. Die Notation $C \bullet X$ bezieht sich auf das innere Produkt von den symmetrischen Matrizen C und X und ist gegeben durch

$$C \bullet X = tr(C\,X) = \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij}\,X_{ij}.$$

Diese SDP Formulierung ist eine Relaxierung des ursprünglichen Problems, dessen exakte Lösung (bei der in dieser Arbeit vorgeschlagenene Modellierung) noch zusätzlich eine Rang 1 Bedingung erfüllen muß. Der Vorteil von Semidefiniten Programmen ist ihre gute Lösbarkeit mit Innere-Punkte-Verfahren.

Für spezielle Fälle (sog. "uniquely" lokalisierbare Netzwerke) wird gezeigt, daß das vorgeschlagene SDP die exakte Lösung des SNLP liefert.

Die interessantere Problemvariante resultiert durch die Miteinbeziehung von Meßfehlern, was zu einem gestörten Lokalisierungsproblem führt und dem Fall entspricht, der in der Praxis normalerweise eintritt (z.B. durch Störungen des Signals zwischen den Sensoren durch evtl. topographische Gegebenheiten). In diesem Zusammenhang wird eine lokale Fehlerschranke für die gestörte Lösung

hergeleitet.

Um die Lösung des relaxierten Problems zu verbessern, werden mehrere Heuristiken vorgestellt und numerisch verglichen. Für die sog. *Curvature Descent* Methode wird globale und lokal quadratische Konvergenz erreicht.

Contents

Introduction

Nowadays, wireless communication is one of the most important aspects in modern life.
Microstructure and semiconductor technology has become every day life, and thus a tiny device is able to intercommunicate over hundreds of kilometers with other receivers even without the need of constant electricity nor cables.
Mobile phones are just the eye-catching example for the extraordinary development in communication technologies during the past decades. The field of research of wireless communication is huge.

The purpose of this thesis is to analyze the problem of localizing sensors inside a (wireless ad hoc) network from a mathematical point of view.

Ad hoc wireless networks are used e.g. for environmental observation. They consist of several very small devices which have the ability to communicate among each other and register specific factors (e.g. temperature, humidity, gas concentration of atmosphere, seismic activity, etc.). Normally, these sensors are randomly distributed over a certain area (e.g. by an airplane, or they move with sea currents). The goal is therefore to find the (exact) position of each sensor within the network, using the distance information (say for example sensor $x_{(i)}$ is $2,5\ m$ away from sensor $x_{(j)}$, etc.)
Of course, with the use of GPS technology, the localization problem would disappear. Still, using GPS would imply that the technology of each sensor is more sophisticated and the energy consumption is higher, which would increase the entire costs of the network.
Therefore, the employment of anchor points is very common in practice. This means, that only a few sensors of the network are equipped with GPS (or simply located a priori).

Mathematically, the Sensor Network Localization Problem results in a sustainability problem, namely find vectors (sensors) $x_{(i)}$ to fulfil the given distance information.
It can be modeled as an unconstrained non-convex optimization problem (least squares formulation for error minimization, cp. (2.3)).

In general, unconstrained, non-convex optimization problems are very difficult to solve as they may have several critical points and any solving algorithm may get stuck in a neighbourhood of such a stationary point. Finding a solver that converges to the global minimum (if existent) has not yet been successful. To overcome this gap, the use of heuristics developed for special problem classes is very common.

For the solution of the Sensor Network Localization Problem we propose a rank 1 semidefinite relaxation of the non-convex problem formulation.
We prove that for uniquely localizable networks this relaxation, which can be solved efficiently by interior point algorithms, provides the exact solution of the localization problem.

In practice, the given distance information is full of measurement errors (due to technical details, e.g. topographical situation, etc.). The localization problem can therefore be solved only approximately.

We obtain a good initial approximation to the SNLP solution by solving the rank 1 SDP relaxation, and present different heuristics to solve the general unconstrained non-convex problem starting from this initial point.
As only descent methods are proposed, global convergence is easily shown. For the proposed *Curvature Descent* approach also local quadratic convergence is obtained.

Completing the work, numerical examples are presented and the solution of the different approaches are compared.

I thank Prof. Dr. Florian Jarre for the at all times constructive supervision of this thesis. Especially I thank his goodwill to make it possible to find an equilibrium between working and having a little child at home.

Chapter 1

Theory of Semidefinite Programs

As Semidefinite Programs are studied, this chapter provides a short summary of basic properties. The following general notation is used

- $\mathcal{S}^n$ denotes the space of real symmetric matrices of dimension n,
  i.e. $\mathcal{S}^n := \{M \in \mathbb{R}^{n \times n} \mid M = M^T\}$.
- $\mathcal{S}^n_+$ denotes the cone of positive semidefinite matrices,
  i.e. $\mathcal{S}^n_+ := \{M \in \mathcal{S}^n \mid x^T M x \geq 0 \quad \forall x \in \mathbb{R}^n\} = \{M \in \mathcal{S}^n \mid M \succeq 0\}$.
- $A \bullet B$ denotes the inner trace product between two symmetric matrices A and B of same dimension,
  i.e. $A \bullet B = tr\,(A^T B) = tr\,(A\,B) = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij}\, B_{ij}$.

1.1 Basic Properties and Notation

Semidefinite problems can be seen as a special case of conic problems, namely minimization over the self-dual cone of semidefinite matrices.
The primal problem formulation is given by

$$\begin{aligned} \min \quad & C \bullet X \\ \text{s.t.} \quad & A^{(i)} \bullet X = b_i \quad \text{for } i = 1, ..., m \\ & X \in \mathcal{S}^n_+ \end{aligned} \tag{PSDP}$$

for some real n-dimensional symmetric matrices $A^{(i)}$ and C and a given vector $b = (b_1, ..., b_m)^T \in \mathbb{R}^m$.

The corresponding dual formulation is

$$\begin{aligned} \max \quad & b^T y \\ \text{s.t.} \quad & C - \sum_{i=1}^{m} y_i A^{(i)} \succeq 0. \end{aligned} \tag{DSDP}$$

As can be seen from the latter, there is a strong analogy between linear and semidefinite programs. In fact, most of the theoretical behaviour of linear problems can be transferred to semidefinite programs. For example, it is easily shown that weak duality holds, i.e.

$$C \bullet X \geq b^T y$$

for any feasible $X \in \mathcal{S}^n$ and $y \in \mathbb{R}^m$.

Furthermore, if (PSDP) and (DSDP) satisfy Slater's condition, i.e. there exists a strictly feasible matrix $X \succ 0$, $A^{(i)} \bullet X = b_i$ for $i = 1, .., m$, resp. $C \succ \sum_{i=1}^{m} y_i A^{(i)}$, then a matrix X and a vector y are optimal for (PSDP) and (DSDP) if and only if there exists a matrix $S \in \mathcal{S}^n$ and the following optimality conditions are satisfied

$$\sum_{i=1}^{m} y_i \; A^{(i)} + S = C, \tag{OC1}$$

$$\mathcal{A}(X) = b \quad X \succeq 0, \; S \succeq 0, \tag{OC2}$$

$$XS = 0, \tag{OC3}$$

where $\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^m$ is a linear mapping defined as

$$\mathcal{A}(X) := \begin{pmatrix} A^{(1)} \bullet X \\ \vdots \\ A^{(m)} \bullet X \end{pmatrix},$$

see e.g. [11].

The last optimality condition (OC3) is stronger than the "natural" one $X \bullet S = 0$ and leads to a different formulation of the complementary slackness theorem for SDP (see [1], Corollary 2.11):

Proposition 1.1
Let X be a feasible matrix for (PSDP) *with eigenvalues $\lambda_1, ..., \lambda_n$ and $S = C - \sum_{i=1}^{m} Y_i \; A^{(i)}$ be feasible for the dual problem* (DSDP) *with eigenvalues $\omega_1, ..., \omega_n$.*

Then X and S are primal and dual optimal solutions, respectively, if and only if they commute and there is a permutation π of eigenvalues of S such that

$$\lambda_i \omega_{\pi_i} = 0 \quad for \quad i = 1, ..., n.$$

From this proposition follows that at least one of λ_i or ω_{π_i} (corresponding to the same eigenvector u^i) must be zero for each $i = 1, ..., n$. Therefore it holds that

$$rank(X) + rank(S) \leq n. \tag{1.1}$$

At this point the analogy to linear programs ends, in the sense that it is not clear how to predict rank(X) or rank(S) in general. In linear programs one can say that in absence of non-degeneracy, exactly m components of the optimal solution are zero. For SDPs it can be stated that

$$rank(X) < n$$

as the optimum of the primal (PSDP) is attained on the boundary of the semidefinite cone. Pataki [21] studies facial structure of feasible sets of SDPs and shows the stronger result

$$\frac{rank(X)(rank(X)+1)}{2} \leq m + r$$

where r is the dimension of the minimal face of the feasible set of (PSDP) containing X.

In the seminal work of Nesterov and Nemirovski [19], it was shown that the SDP problem (PSDP) can be solved in polynomial time by using the selfconcordant barrier function

$$\phi(X) = -\ln(\det(X^{-1})).$$

There are several interior-point methods for solving SDPs in practice, a listing can be found in [9].

For the numerical implementation of our test problems we used Sedumi [23], which is a Matlab Toolbox that implements the self-dual embedding technique as proposed in [17].

Chapter 2

The Sensor Network Localization Problem

This chapter gives a formulation of the Sensor Network Localization Problem and outlines an overview of the most known approaches.

2.1 Problem Description

Consider a (wireless) ad hoc network that consists of n sensors and m anchors. The network can be seen as a graph with $n+m$ nodes. The positions of the n nodes that correspond to the sensors are not known. The positions of the nodes that correspond to the m anchorpoints are known (e.g. via GPS). The (unknown) sensors will be notated with the variables $x_{(1)}, ..., x_{(n)} \in \mathbb{R}^d$ (in this thesis we will restrict our observations to the case $d = 2$), and the anchors with the variables $a_{(n+1)}, ..., a_{(n+m)} \in \mathbb{R}^d$.

In real world problems it is unusual that all sensors/anchors are able to intercommunicate. Therefore, a so-called radio range is defined. This is a magnitude that provides information of the furthest distance where two sensors/anchors are still able to communicate with each other. In other words, if the given radiorange (which normally is the same for all nodes in the graph) is $r \in \mathbb{R}^+$, sensor $x_{(i)}$ can communicate with sensor $x_{(j)}$ if and only if $\|x_{(i)} - x_{(j)}\|_2 \leq \mathrm{r}$. With this assumption a (typically incomplete) subgraph is obtained.

The radio range defines subsets $N_x \subsetneqq \{1, ..., n\} \times \{1, ..., n\}$ and $N_a \subsetneqq \{1, ..., n\} \times \{1, ..., m\}$ in the following way:

$$(i,j) \in N_x \Longleftrightarrow \|x_{(i)} - x_{(j)}\|_2 \leq r \quad \forall\, i,j = 1,...,n \quad \text{and}$$
$$(l,k) \in N_a \Longleftrightarrow \|x_{(l)} - a_{(n+k)}\|_2 \leq r \quad \forall\, l = 1,...,n,\ k = 1,...,m.$$

With this, only certain distance information is given, namely the values d_{ij} and $d_{lk} \in \mathbb{R}$ for $(i,j) \in N_x$ and $(l,k) \in N_a$ which represent $\|x_{(i)} - x_{(j)}\|_2$ and $\|x_{(l)} - a_{(n+k)}\|$, respectively. In chapter 4 the problem will be discussed when the given distance measurements are not exact.

The aim is to find the (eventually) exact position of the n unknown sensors using the distance information given. A mathematical formulation of the Sensor Network Localization Problem (in the following abbreviated as (SNLP)) is given as follows:

Given m anchor locations $a_{(n+k)} \in \mathbb{R}^d$ for $k = 1,2,...,m$ and some distance measurements d_{ij}, d_{lk} for $(i,j) \in N_x \subset \{1,...,n\}^2$, and $(l,k) \in N_a \subset \{1,...,n\} \times \{1,...,m\}$, find $x_{(i)} \in \mathbb{R}^d$, $i = 1,2,...,n$, the locations of n sensors, such that

$$\begin{aligned} \|x_{(i)} - x_{(j)}\|_2^2 &= d_{ij}^2, \quad \forall (i,j) \in N_x \quad \text{and} \\ \|x_{(l)} - a_{(n+k)}\|_2^2 &= d_{lk}^2, \quad \forall (l,k) \in N_a. \end{aligned} \tag{2.1}$$

This problem can be seen as an optimization or more precisely as a satisfiability problem in the following sense:

$$\begin{aligned} \min \quad & 0 \\ s.t. \quad & x_{(1)},...,x_{(n)} \in \mathbb{R}^d \\ & \|x_{(i)} - x_{(j)}\|_2^2 = d_{ij}^2 \quad \text{for } (i,j) \in N_x \\ & \|x_{(l)} - a_{(n+k)}\|_2^2 = d_{lk}^2 \quad \text{for } (l,k) \in N_a. \end{aligned} \tag{2.2}$$

This problem is a non-convex optimization problem and moreover, it is NP-hard (for details see appendix A).

2.2 Known Approaches

There have been several approaches to solve (2.2). The most common technique in practice is the so called *multilateration* which arises from the least squares solution of the given nonlinear equations of (2.1):

$$\min_{x \in \mathbb{R}^{dn}} \sum_{(i,j) \in N_x} \| \|x_{(i)} - x_{(j)}\|_2^2 - d_{ij}^2 \|_2^2 + \sum_{(l,k) \in N_a} \| \|x_{(l)} - a_{(n+k)}\|_2^2 - d_{lk}^2 \|_2^2. \tag{2.3}$$

It is clear, that the latter is an unconstrained non-convex optimization problem. It therefore can have several minima and usually its solution is computed by heuristics.

2.2.1 SDP relaxation by Ye et al.

A more elegant interpretation of the SNLP was proposed by Biswas & Ye [5], which has been driven much attention during the last couple of years. In their work, a rank 2 SDP model was proposed in the following way. Let

$$X = [x_{(1)}, ..., x_{(n)}] \in \mathbb{R}^{2\times n}$$

be the matrix containing the desired sensors listed as column vectors. Let $e_{ij} \in \mathbb{R}^n$ be the vector with 1 at the i-th, -1 at the j-th position and zero elsewhere and $e_l \in \mathbb{R}^n$ the vector of all zeros except an -1 at the l-th position. The conditions (2.1) can then be stated as

$$\begin{aligned} \|x_{(i)} - x_{(j)}\|_2^2 &= e_{ij}^T X^T X e_{ij} \quad \text{for} \quad (i,j) \in N_x \quad \text{and} \\ \|x_{(l)} - a_{(n+k)}\|_2^2 &= (a_{(n+k)}; e_l)^T [I_d; X]^T [I_d; X](a_{(n+k)}; e_l) \quad \text{for} \quad (l,k) \in N_a. \end{aligned} \tag{2.4}$$

Thus, problem (2.1) evolves into finding a symmetric matrix $Y \in \mathbb{R}^{n\times n}$ and a matrix $X \in \mathbb{R}^{2\times n}$ such that

$$\begin{aligned} & e_{ij}^T Y e_{ij} = d_{ij}^2 \quad \text{for} \quad (i,j) \in N_x \\ & (a_{(n+k)}; e_l)^T \begin{pmatrix} I_d & X \\ X^T & Y \end{pmatrix} (a_{(n+k)}; e_l) = d_{lk}^2 \quad \text{for} \quad (l,k) \in N_a \\ & Y = X^T X. \end{aligned} \tag{2.5}$$

Relaxing the last condition to $Y \succeq X^T X$, which is equivalent to require

$$Z = \begin{pmatrix} I_d & X \\ X^T & Y \end{pmatrix} \succeq 0$$

(by Shur's complement), transforms the SNLP into a standard SDP problem, namely

$$
\begin{aligned}
\min\ & 0 \\
\text{s.t. } & Z_{1:d,1:d} = I_d \\
& (\mathbf{0}; e_{ij})(\mathbf{0}; e_{ij})^T \bullet Z = d_{ij}^2 \quad \forall \quad (i,j) \in N_x \\
& (a_{(n+k)}; e_l)(a_{(n+k)}; e_l)^T \bullet Z = d_{lk}^2 \quad \forall \quad (l,k) \in N_a \\
& Z \succeq 0,
\end{aligned}
\tag{2.6}
$$

where $Z_{1:d,1:d}$ denotes the block of the matrix Z consisting of the first d rows and columns.

So & Ye showed in [25] that for uniquely localizable networks the solution of (2.6) is equivalent to the solution of the original problem.

2.2.2 Euclidean Distance Matrix Formulation

In [6] Wolkowicz et al. generalize the SDP relaxation of Ye et al. (2004) by connecting it to the theory of Euclidean Distance Matrices.
A $n \times n$ symmetric matrix $D = (d_{ij})$ with nonnegative elements and zero diagonal is called *pre-distance matrix* (or dissimilarity matrix).
In addition, if there exist points $x_{(1)}, ..., x_{(n)}$ in $\mathbb{R}^d$ such that

$$d_{ij}^2 = \|x_{(i)} - x_{(j)}\|_2^2 \quad i,j = 1, ..., n,$$

then D is called an *Euclidean Distance Matrix* (EDM). Note, that in an EDM the entire distance information between all sensors $x_{(i)}$ is given.
For a given matrix $B \in \mathcal{S}^n$ the linear operators $\mathcal{D}_e$ and $\mathcal{K}$ are then introduced by

$$\mathcal{D}_e := diag(B)\ e^T + e\ diag(B)^T, \quad \mathcal{K}(B) := \mathcal{D}_e(B) - 2B,$$

where e is the vector of ones and $diag(B)$ is the vector containing the diagonal elements of the matrix B. The adjoint operators are

$$\mathcal{D}_e^*(D) = 2Diag(De), \quad \mathcal{K}^*(D) = 2(Diag(De) - D),$$

where $Diag(De)$ is the diagonal matrix whose (diagonal) entries are the components of the vector De.
It can be shown that $\mathcal{K}$ maps the cone of positive semidefinite matrices onto the cone of Euclidean distance matrices. This fact establishes a relation between the Euclidean distance matrix formulation and the SDP formulation of the problem.

The anchors are treated as variables, setting

$$\bar{Y} := PP^T = \begin{bmatrix} XX^T & XA^T \\ AX^T & AA^T \end{bmatrix}$$

with $A = [a_{(n+1)}, ..., a_{(n+m)}]^T \in \mathbb{R}^{m \times d}$, $P^T = [X^T \; A^T]^T$.
An EDM problem equivalent to the SNLP is obtained with

$$\begin{aligned} \min \; & f(Y) := \frac{1}{2}\|W \circ (\mathcal{K}(Y) - E)\|_F^2 \\ \text{s.t.} \; & g_u(Y) \leq 0 \\ & g_l(Y) \geq 0 \\ & Y - PP^T = 0 \end{aligned} \qquad (SNL_{EDM})$$

where $E \in R^{(n+m)\times(n+m)}$ denotes the matrix of the given distances $e_{ij} = d_{ij}$ for $(i,j) \in N_x$ and $e_{lk} = d_{lk}$ for $(l,k) \in N_a$, the matrix $W \in \mathbb{R}^{(n+m)\times(n+m)}$ is a weight matrix (with higher weight entries when anchor points or exact known data are considered) and the functions g_u and g_l fulfil upper and lower bounds for the sensors/anchors which are not connected.
The last constraint is then relaxed to $Y \succeq PP^T$ which is equivalent to the simpler condition $Y \succeq 0$.

The problem (SNL_{EDM}) can be reduced to a smaller dimensional one by using the fact that the block of Y which corresponds to the anchorpoints information is a priori known and therefore stands for a clique in the underlying graph. Any other set of nodes which are highly connected to the network or where it is known, that there exists exact distance information can be used to reduce the problem even more (for details see [6]).

Further in [6], a primal-dual interior/exterior-point algorithm based on the Gauss-Newton search direction is proposed for solving (SNL_{EDM}). Particularly, this search direction (described in [13]) does not restrict the iterates to the interior of the feasible set, and thus usually lower rank optimal solutions are obtained. These solutions are a better approximation to the exact solution than solutions with higher rank. In [6] different approaches for solving the SNLP are compared numerically and the SDP relaxation of the EDM problem performs very well with the proposed interior/exterior-point algorithm.

2.2.3 SOC relaxation

In [27] and [7] a Second Order Cone (SOC) relaxation of the SNLP is proposed. The SNLP is equivalent to the problem:

$$\begin{aligned} \min \quad & \sum_{(i,j)\in N_x} \mid y_{ij} - d_{ij}^2 \mid + \sum_{(l,k)\in N_a} \mid y_{lk} - d_{lk}^2 \mid \\ \text{s.t.} \quad & y_{ij} = \|x_{(i)} - x_{(j)}\|_2^2 \quad \forall\, (i,j) \in N_x \\ & y_{lk} = \|x_{(l)} - a_{(n+k)}\|_2^2 \quad \forall\, (l,k) \in N_a. \end{aligned}$$

This can be relaxed into a SOC problem in the following way:

$$\begin{aligned} \min \quad & \sum_{(i,j)\in N_x} \mid y_{ij} - d_{ij}^2 \mid + \sum_{(l,k)\in N_a} \mid y_{lk} - d_{lk}^2 \mid \\ \text{s.t.} \quad & y_{ij} \geq \|x_{(i)} - x_{(j)}\|_2^2 \quad \forall\, (i,j) \in N_x \\ & y_{lk} \geq \|x_{(l)} - a_{(n+k)}\|_2^2 \quad \forall\, (l,k) \in N_a. \end{aligned} \qquad (SNL_{SOC})$$

It is shown in [27] that the SOC relaxation contains the SDP relaxation (proposed in [5]) and is therefore weaker. It has the great advantage that the problem dimension is much smaller than the SDP relaxation and a solution via interior point methods is faster. A smoothing and coordinate gradient descent method is proposed in [27] which has the feature to easily distribute over many processors in parallel.

The main disadvantage of the SOC relaxation of the SNLP is, that it is only possible to find good approximations of the localization of the sensors inside the convex hull of the anchors. This means, that only problems with anchorpoints at the boundary of the network make sense for this problem formulation. Unfortunately, this fact restricts the formulation to a very specific range of SNLPs. Nevertheless, the SOC relaxation can be seen as a pre-location method when handling large scale problems.

2.3 Our Approach

For networks with exact distance data d_{ij} and d_{lk} for $(i,j) \in N_x$ and $(l,k) \in N_a$ the known approaches from literature offer satisfactory results. Furthermore, networks that are uniquely localizable can be solved exactly via the SDP relaxation as proposed in [5].

SNLPs with disturbed distance data are still difficult to solve. We therefore see the necessity to develop other approaches that perform well at disturbed problems but still provide exact solutions for uniquely localizable problems.

We propose a SDP relaxation of the SNLP by modelling it with a rank 1 matrix. This formulation has the advantage that the resulting matrix of the relaxed problem provides a good initial starting point for the general least squares formulation of the original problem (2.2). This means, the solving procedure splits into two parts.
Firstly, the SDP problem is solved and an initial approximation vector $\tilde{x}$ to the problem solution is obtained.
Secondly, a solution refinement is made using descent heuristics for the unconstrained non-convex least squares problem formulation (2.3). These heuristics are presented in chapter 5 and can be applied to any unconstrained optimization problem.

We show that our proposed relaxation provides exact solution to the original problem if the underlying network is uniquely localizable.

From a numerical point of view, we obtain good accuracy of the approximate solution for disturbed problems. Our results are presented in chapter 7 and are compared to those of Ye et al.

Chapter 3

A Rank 1 SDP Approach

In this chapter a rank 1 SDP approach is presented, which gives an approximate solution to the SNLP. Moreover, its theoretical behaviour is analyzed.

3.1 Construction of a Rank 1 Model

In order to have a better overview, the problem formulation of (2.2) is repeated for the case $d = 2$:

$$\begin{aligned} \min \quad & 0 \\ s.t. \quad & x_{(1)}, ..., x_{(n)} \in \mathbb{R}^2 \\ & \|x_{(i)} - x_{(j)}\|_2^2 = d_{ij}^2 \quad \text{for } (i,j) \in N_x \\ & \|x_{(l)} - a_{(n+k)}\|_2^2 = d_{lk}^2 \quad \text{for } (l,k) \in N_a. \end{aligned} \tag{3.1}$$

This problem can be reformulated in the following way.

Let $x = (x_{(1)}^T, ..., x_{(n)}^T)^T \in \mathbb{R}^{2n}$ be the vector containing all the sensors listed one below the other. By setting

$$Y := \begin{bmatrix} 1 & x^T \\ x & xx^T \end{bmatrix} \in \mathbb{R}^{(2n+1)^2}$$

it holds that

$$\begin{aligned} \|x_{(i)} - x_{(j)}\|_2^2 &= Y \bullet B_{ij} \quad \text{for all} \quad (i,j) \in N_x \quad \text{and} \\ \|x_{(l)} - a_{(n+k)}\|_2^2 &= Y \bullet B_{lk}^a \quad \text{for all} \quad (l,k) \in N_a, \end{aligned} \tag{3.2}$$

where B_{ij} denotes the matrix containing I_2 at the $(2i : 2i+1)$-th and the $(2j : 2j+1)$-th block and containing $-I_2$ at the $(2i : 2i+1, 2j : 2j+1)$-th and

the $(2j : 2j+1, 2i : 2i+1)$-th block, elsewhere zeros; i.e.

$$B_{ij} := \begin{pmatrix} \mathbf{0} & \dots & \dots & \dots & \mathbf{0} \\ \vdots & I_2 & & -I_2 & \vdots \\ \vdots & & & & \vdots \\ \vdots & -I_2 & & I_2 & \vdots \\ \mathbf{0} & \dots & \dots & \dots & \mathbf{0} \end{pmatrix}$$

and B^a_{kl} denotes the matrix containing $\|a_{(n+k)}\|_2^2$ at the $(1,1)$-th position, $-a_{(n+k)}$ at the $(2l : 2l+1)$-position of the first column and row (or transposed, respectively), and I_2 at the $(2l : 2l+1)$-th block, elsewhere zeros, i.e.

$$B^a_{lk} := \begin{pmatrix} \|a_{(n+k)}\|_2^2 & 0 & \dots & 0 & -a^T_{(n+k)} & 0 & \dots & 0 \\ 0 & & & & & & & \\ \vdots & & \mathbf{0} & & & & \mathbf{0} & \\ 0 & & & & & & & \\ -a_{(n+k)} & & & & I_2 & & & \\ 0 & & & & & & & \\ \vdots & & \mathbf{0} & & & & \mathbf{0} & \\ 0 & & & & & & & \end{pmatrix}.$$

Problem (3.1) results in:

$$\begin{aligned} \min \quad & 0 \\ s.t. \quad & Y \bullet B_{ij} = d_{ij}^2 \quad \text{for all} \quad (i,j) \in N_x \\ & Y \bullet B^a_{lk} = d_{lk}^2 \quad \text{for all} \quad (l,k) \in N_a \\ & Y(1,1) = 1 \\ & Y \in \mathcal{S}_+^{2n+1} \\ & \operatorname{rank}(Y) = 1, \end{aligned} \tag{3.3}$$

where $\mathcal{S}_+^{2n+1}$ denotes the space of symmetric positive definite matrices of dimension $(2n+1)$.
By dropping the rank 1 constraint a relaxation of the original problem (3.1) is obtained. This leads to the standard SDP formulation:

$$\begin{aligned} \min \quad & 0 \\ s.t. \quad & Y \bullet B_{ij} = d_{ij}^2 \quad \text{for all} \quad (i,j) \in N_x \\ & Y \bullet B_{lk}^a = d_{lk}^2 \quad \text{for all} \quad (l,k) \in N_a \\ & Y(1,1) = 1 \\ & Y \in \mathcal{S}_+^{2n+1}. \end{aligned} \tag{3.4}$$

Any feasible matrix Y for (3.4) has the shape

$$Y = \begin{pmatrix} 1 & y^T \\ y & Z \end{pmatrix} \tag{3.5}$$

with $Z \succeq yy^T$.

As there is no objective function, it can be formulated as a maximize-problem. By introducing the$(2n+1)$-dimensional slack matrix/variable S the dual SDP has the shape:

$$\begin{aligned} \min \quad & v + \sum_{(i,j)\in N_x} d_{ij}^2 \, y_{ij} + \sum_{(l,k)\in N_a} d_{lk}^2 \, y_{lk} \\ s.t. \quad & S = \begin{pmatrix} v & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} + \sum_{(i,j)\in N_x} y_{ij} B_{ij} + \sum_{(l,k)\in N_a} y_{lk} B_{lk}^a \\ & S \in \mathcal{S}_{(2n+1)}^+, \; v \in \mathbb{R}, \; y_{ij} \in \mathbb{R}, \; y_{lk} \in \mathbb{R}, \end{aligned} \tag{3.6}$$

for given $(i,j) \in N_x$ and $(l,k) \in N_a$.

From the theoretical point of view the dual problem has always a feasible solution (sc. $v = y_{ij} = y_{lk} = 0$ for all $(i,j) \in N_x$ and $(l,k) \in N_a$). If the primal problem (3.4) has a feasible solution, the optimal value of it (and of the dual) must be zero. The following relation holds

Lemma 3.1
Let $\hat{Y}$ be a solution of (3.4) $\hat{S}$ a solution of (3.6). It holds

1. $\hat{Y}\hat{S} = 0$,
2. $rank(\hat{Y}) + rank(\hat{S}) \leq 2n+1$,
3. $rank(\hat{Y}) \geq 1$,
4. $rank(\hat{S}) \leq 2n$.

Proof:
See the duality theorem for SDP problems [1]. □

This means that if an optimal solution matrix from the dual has rank $2n$, then every solution of the primal (3.4) has rank 1 and both problems are equivalent. Therefore, the SNLP can be solved in polynomial time.

3.2 Solution of the relaxed problem vs. exact solution of the SNLP

It follows from theorem (**??**) below that if exact (and enough) distance information is given, the dual problem has a solution of rank $2n$. Therefore, using a standard interior point algorithm (in our case we worked with Sedumi [23]) it is easy to find a rank 1 solution of (3.4).
For a better understanding when "enough" distance information is given, we will take over the terminus uniquely localizable as stated for the first time in [25]. We transfer the notation and the important statements and adapt the concept to our rank 1 formulation.

Definition 3.2
Problem (2.1) *is called uniquely localizable if there is an unique localization* $\hat{x} = (\hat{x}_{(1)}, ..., \hat{x}_{(n)})^T \in \mathbb{R}^{2n}$ *and there are no* $x_{(i)} \in \mathbb{R}^h$ *for* $i = 1, ..., n$, *where* $h > 2$, *such that*

$$\|x_{(i)} - x_{(j)}\|_2^2 = d_{ij}^2 \quad \textit{for } (i,j) \in N_x$$

$$\|x_{(l)} - (a_{(n+k)}, \mathbf{0})^T\|_2^2 = d_{lk}^2 \quad \textit{for } (l,k) \in N_a$$

$$x_{(j)} \neq \begin{pmatrix} \hat{x}_{(j)} \\ \mathbf{0} \end{pmatrix} \quad \textit{for some} \quad j \in \{1, ..., n\}.$$

This implies that when a network is uniquely localizable, the problem can not have a non-trivial localization in some higher dimension (i.e. a localization different from the one obtained by setting $x_{(i)} = (\hat{x}_{(i)}, \mathbf{0})^T \in \mathbb{R}^h$ for $i = 1, ..., n$), where anchor points are augmented to $(a_{(n+k)}, \mathbf{0}) \in \mathbb{R}^h$ for $k = 1, ..., m$.

The following proposition is interesting in its own right.

Proposition 3.3
If every sensor point is connected, directly or indirectly to an anchor point in (3.1), *then any solution of* (3.4) *is bounded, i.e.* Z_{pq} *in 3.5 is bounded for all* $p, q = 1, ..., 2n$.

In this case, the network is called connected.

Proof:
From positive definiteness follows, that the determinant of every minor of Z must be positive, i.e.

$$Z_{pp}Z_{qq} - Z_{pq}^2 > 0 \tag{3.7}$$

for every $p, q = 1, ..., 2n$. It is therefore sufficient to show, that every diagonal element of Z is bounded. It is clear, that the diagonal elements are bounded below by 0, because of positive definiteness.
Assume now that a sensor $x_{(l)}$ is connected to an anchor $a_{(n+k)}$, for any $l \in \{1, ..., n\}$ and $k \in \{1, ..., m\}$; it holds

$$\begin{aligned}\|x_{(l)} - a_{(n+k)}\|_2^2 &= \|x_{(l)}\|_2^2 - 2x_{(l)}^T a_{(n+k)} + \|a_{(n+k)}\|_2^2 \\ &\leq Z_{2l-1,2l-1} + Z_{2l,2l} - 2x_{(l)}^T a_{(n+k)} + \|a_{(n+k)}\|_2^2 \\ &= d_{lk}^2.\end{aligned}$$

This implies that $\|x_{(l)}\|$ in (3.5) is bounded. Using Cauchy-Schwarz, we get

$$Z_{2l-1,2l-1} + Z_{2l,2l} \leq d_{lk}^2 + 2\|x_{(l)}\|\|a_{n+k}\| - \|a_{(n+k)}\|_2^2.$$

Again, because the diagonal elements of Z are positive, each of the former summands is bounded.
Furthermore, if a sensor $x_{(i)}$ is in turn connected to a sensor $x_{(l)}$ (which is a sensor connected directly to an anchor point), using (3.7) we have

$$\begin{aligned}(\sqrt{Z_{2i-1,2i-1} + Z_{2i,2i}} - \sqrt{Z_{2l-1,2l-1} + Z_{2l,2l}})^2 \\ \leq Z_{2i-1,2i-1} + Z_{2i,2i} - 2(Z_{2i-1,2l-1} + Z_{2i,2l}) + Z_{2l-1,2l-1} + Z_{2l,2l} \\ = d_{i,l}^2\end{aligned}$$

and therefore $Z_{2i-1,2i-1}$ and $Z_{2i,2i}$ are bounded.

□

Theorem 3.4
The following statements are equivalent:

1. *Problem* (2.1) *is uniquely localizable.*
2. *The max-rank solution matrix* $\hat{Y}$ *of* (3.4) *has rank 1.*
3. *The solution matrix* $\hat{Y}$ *of* (3.4) *satisfies* $\hat{Z} = yy^T$ *(cp.* (3.5)*).*

Remark 3.5
A primal (dual) max-rank solution is a solution that has the highest rank among all solutions for the primal problem (3.4) (dual (3.6)).

Proof:
The equivalence between 2. and 3. is straightforward. We proof the equivalence between 1. and 2.
For the direction from 2. to 1. we will only have to prove, that if the max-rank solution has rank 1 then it is unique.
Suppose therefore, that there are two different feasible rank 1 solutions

$$\tilde{Y} = \begin{pmatrix} 1 & \tilde{x}^T \\ \tilde{x} & \tilde{x}\tilde{x}^T \end{pmatrix} \quad \text{and} \quad \hat{Y} = \begin{pmatrix} 1 & \hat{x}^T \\ \hat{x} & \hat{x}\hat{x}^T \end{pmatrix}$$

of (3.4) with $\tilde{x} \neq \hat{x}$. From convexity it arises that the matrix $Y = \alpha\tilde{Y} + \beta\hat{Y}$ for $\alpha + \beta = 1$ and $\alpha, \beta > 0$ is a feasible solution and its rank must be 1 (since the max-rank is assumed to be 1). The matrix Y is given by

$$Y = \begin{pmatrix} 1 & \alpha\tilde{x}^T + \beta\hat{x}^T \\ \alpha\tilde{x} + \beta\hat{x} & \alpha\tilde{x}\tilde{x}^T + \beta\hat{x}\hat{x}^T \end{pmatrix}.$$

As Y has rank 1

$$\alpha\tilde{x}\tilde{x}^T + \beta\hat{x}\hat{x}^T = (\alpha\tilde{x}^T + \beta\hat{x}^T)(\alpha\tilde{x} + \beta\hat{x})$$

must hold. This implies that $\|\tilde{x} - \hat{x}\| = 0$, which is a contradiction.
For the direction from 1. to 2. we suppose that there is a feasible solution Y of (3.4) with rank $r > 1$. It follows that $Z \succeq xx^T$ and $Z \neq xx^T$. Thus, there exists a matrix $X' \in \mathbb{R}^{r \times 2n}$, such that

$$Z - xx^T = X'^T X'.$$

Let x'_j be the j-th column vector of X' of dimension r. Consider the point

$$p_{(i)} = \begin{pmatrix} x_{2i-1} \\ x_{2i} \\ x'_{2i-1} \\ x'_{2i} \end{pmatrix} \in \mathbb{R}^{2+2r} \quad \text{for} \quad i = 1, ..., n.$$

It holds, that:

$$\|p_{(i)}\|_2^2 = Z_{2i-1,2i-1} + Z_{2i,2i} \quad \text{and} \quad p_{(i)}^T p_{(j)} = Z_{2i-1,2j-1} + Z_{2i,2j}$$

for all $i, j = 1, ..., n$ and there exists at least one $p_{(i)}$ such that $\|p_{(i)}\| \neq \|x_{(i)}\|$. Hence, we have that

$$\begin{aligned}\|p_{(i)} - p_{(j)}\|_2^2 &= Z_{2i-1,2i-1} + Z_{2i,2i} - 2(Z_{2i-1,2j-1} + Z_{2i,2j}) + Z_{2j-1,2j-1} + Z_{2j,2j} \\ &= d_{ij}^2\end{aligned}$$

and

$$\begin{aligned}\|p_{(l)} - (a_{(n+k)}; \mathbf{0})^T\|_2^2 &= \|a_{(n+k)}\|_2^2 - 2a_{(n+k)}^T x_{(l)} + Z_{2l-1,2l-1} + Z_{2l,2l} \\ &= d_{lk}^2\end{aligned}$$

for all $(l,k) \in N_a$ and $(i,j) \in N_x$.

In other words, $p = (p_{(1)}, ..., p_{(n)})^T$ is a (non trivial) localization of problem (3.1) in $\mathbb{R}^{2+r}$, which is a contradiction.

□

Summing up, the SDP method in addition to a standard interior point algorithm guarantees to find the solution to the SNLP if the input graph (network) is uniquely localizable.

This is a great achievement for problems with exact distance data. Unfortunately, in real applications the distance information is often linked to error measurements, which arise e.g. from topographical occurrence or general interference. We will therefore concentrate our efforts on non uniquely localizable problems and on the disturbed SNLP (which will be described in the next section).

Chapter 4

Error Measurements and Noisy Data

In real world problems it is usual to get noisy data due to errors in distance measurements. This chapter describes the SNLP with noisy data.

4.1 Problem Description with Noisy Data

In practice it is very common to deal with measurement errors due to topographical (or other) factors.
We therefore introduce to our model the variables $\tilde{d}_{ij}$ and $\tilde{d}_{lk}$ for $(i,j) \in N_x$ and $(l,k) \in N_a$. The measurement errors δ_{ij} and δ_{lk} are assumed to be small real numbers:

$$\tilde{d}_{ij} := d_{ij} + \delta_{ij} \quad \text{with} \quad \delta_{ij} \in \mathbb{R} \quad \text{and}$$

$$\tilde{d}_{lk} := d_{lk} + \delta_{lk} \quad \text{with} \quad \delta_{lk} \in \mathbb{R}.$$

The disturbed SNLP can be relaxed to:

$$\begin{aligned} & \min 0 \\ \text{s.t.} \quad & B_{ij} \bullet Y = \tilde{d}_{ij} \quad \text{for} \quad (i,j) \in N_x \\ & B_{lk} \bullet Y = \tilde{d}_{lk} \quad \text{for} \quad (l,k) \in N_a \\ & Y(1,1) = 1 \\ & Y \succeq 0. \end{aligned} \tag{4.1}$$

Any feasible solution of (4.1) will have the shape

$$\tilde{Y} = \begin{pmatrix} 1 & \tilde{y}^T \\ \tilde{y} & \tilde{Z} \end{pmatrix} \tag{4.2}$$

with $\tilde{Z} \succeq \tilde{y}\tilde{y}^T$.

To evade notation difficulties we will call $x \in \mathbb{R}^{2n}$ the vector containing the exact positions of the original network, i.e. x fulfils (2.1). With $\bar{x} \in \mathbb{R}^{2n}$ we will denote a solution of the disturbed SNLP. Note, that $\bar{x}$ does not have to exist. Finally, $\tilde{x} \in \mathbb{R}^{2n}$ will be a vector obtained from the solution matrix $\tilde{Y}$ of (4.1).

Suppose, that from a solution matrix $\tilde{Y}$ a vector $\tilde{x}$ is formed (e.g. by setting $\tilde{x} = \tilde{y}$). We assume, that $\tilde{x}$ is a "good" approximation to $\bar{x}$. We cannot state that $\tilde{x}$ is a good approximation to x, as the relation between $\bar{x}$ and x is not clear nor easy to understand (a local error bound for $\|x - \bar{x}\|$ is given in the next section).

We then turn to the least squares problem formulation of (2.1):

$$\min_{x \in \mathbb{R}^{2n}} \sum_{(i,j) \in N_x} (\|x_{(i)} - x_{(j)}\|_2^2 - \tilde{d}_{ij}^2)^2 + \sum_{(l,k) \in N_a} (\|x_{(l)} - a_{(n+k)}\|_2^2 - \tilde{d}_{lk}^2)^2 \,. \tag{4.3}$$

This problem is an unconstrained, non-convex problem which therefore may have many different local minimizers.

No algorithm is known which guarantees convergence to a global solution of problems of this type. By the second order optimality conditions it is clear, that there exists a neighbourhood of $\bar{x}$ where the problem is convex. Within this neighbourhood the Hessian of the function to be minimized is positive definite. Of course, this region of positive definiteness is impossible to find in practice. Starting at an arbitrary point, any algorithm could "stuck in" a local minimizer which may be very far away from the global, and could have a much bigger function value.
We developed several heuristic approaches which are presented in detail in the next chapter. This methods can be applied to any kind of unconstrained, non-convex problems. We will call the procedure of solving the least squares problem of (4.3) *rounding technique*.

4.2 A Local Error Bound

Trying to obtain an error bound between the exact solution x of the SNLP and a solution $\bar{x}$ of the disturbed SNLP, a local error bound for a single sensor is developed.

We start with a reduced problem. Let $x^* = (0,0)^T$ be the exact sensor position which shall be found. There are three anchors $a_{(1)}, a_{(2)}, a_{(3)} \in \mathbb{R}^2$ given with (exact) distances $d_i = \|a_{(i)}\|_2$, respectively. The aim is to give an upper bound for the error that occurs in localizing the position of x^* if the distances are disturbed to $\tilde{d}_1, \tilde{d}_2, \tilde{d}_3$. The following assumptions are needed:

1. Assume that the given values $d = (d_1, d_2, d_3)^T$ are consistent, that means, that the mapping $\varsigma : d \mapsto x$ is well defined.
2. We disturb the distances at most $\varepsilon \in \mathbb{R}$, that means $\|\tilde{d} - d\| \leq \varepsilon$.
3. Assume that $\varepsilon \neq \|a_{(i)}\|$ for $i = 1, 2, 3$.

The problem can be expressed by the solution of

$$\max_{x \in \mathbb{R}^2} \{\|x\|_2 \mid \|\tilde{d} - d\| \leq \varepsilon\}. \tag{4.4}$$

Using the infinity norm, condition $\|\tilde{d} - d\|_\infty \leq \varepsilon$ yields

$$\mid \tilde{d} - \|a_{(i)}\|_2 \mid \leq \varepsilon$$

for all $i = 1, 2, 3$ and therefore

$$-\varepsilon \leq \tilde{d} - \|a_{(i)}\|_2 \leq \varepsilon$$

that corresponds to

$$-\varepsilon \leq \|a_{(i)} - x\|_2 - \|a_{(i)}\|_2 \leq \varepsilon.$$

In order to obtain differentiable solutions, the inequalities are squared and the feasible set of problem (4.4) results in

$$M := \{x \in \mathbb{R}^2 \mid \|x_{(i)} - x\|_2^2 \leq (\|x_{(i)}\|_2 + \varepsilon)^2, \|x_{(i)} - x\|_2^2 \geq (\|x_{(i)}\|_2 - \varepsilon)^2 \text{ for } i = 1, 2, 3\}.$$

Problem (4.4) turns into finding the solution of the following problem:

$$\max\{\|x\| \mid x \in M\}. \tag{4.5}$$

Problem (4.5) is an optimization problem with convex and concave restrictions. It is easily shown that its solution $\bar{x}$ can only be an extreme point of M.
Proof:
We show that always more than one restriction of problem (4.5) is active. For this purpose we show a contradiction if we assume that exactly one restriction is active.
1. For the case that the only active restriction is concave, the contradiction is evident, as the Hessian of the corresponding Lagrange function is negative semidefinite along this restriction.

2. For the case that the only active restriction is convex, assume without loss of generality that

$$\|a_{(1)} - x\|_2^2 = (\|a_{(1)}\|_2 + \varepsilon)^2.$$

Regularity holds, so the KKT-conditions must be fulfilled. Let

$$\begin{aligned} f(x) &= x_1^2 + x_2^2 \quad \text{and} \\ g_1(x) &= \|a_{(1)} - x\|_2^2 - (\|a_{(1)}\|^2 + \varepsilon)^2 = 0 \end{aligned}$$

be the function and active restriction. It follows that

$$\begin{aligned} \nabla f(x) &= 2x \quad \text{and} \\ \nabla g_1(x) &= -2(a_{(1)} - x). \end{aligned} \tag{4.6}$$

The KKT-conditions imply

$$\begin{aligned} & 2x = 2u(a_{(1)} - x) \\ \overset{u \neq -1}{\Rightarrow} \; & x = \frac{u}{1+u} a_{(1)} \\ \overset{\text{in (4.6)}}{\Rightarrow} \; & \frac{1}{(1+u)^2} \|a_{(1)}\|_2^2 = (\|a_{(1)}\|_2 + \varepsilon)^2 \end{aligned}$$

It must therefore hold that

$$\begin{aligned} & \frac{1}{(1+u)^2} > 0 \\ \Leftrightarrow \; & u(u+2) < 0 \end{aligned}$$

which is a contradiction to the KKT-condition for the Lagrange multipliers, i.e. $u > 0$. □

A way to solve problem (4.5) is to compare the norm of all extreme points of the feasible set M. These are at most six. We calculated them analytically by comparing the intersections of the respective circumferences. At this point we introduce the possibility that the disturbed problem may have different measurement errors in each distance measure. This means that $\|\tilde{d}_i - d_i\|_2 \leq \varepsilon_i$ for $i = 1, 2, 3$. As the formulas are quite long, one restriction (for example the one corresponding to $a_{(3)}$) of M is omitted. This causes only an enlargement of M. For a single unknown sensor x^* the local error bound to every pair of anchors within the radio range can then be calculated. Of those values, the smallest will be taken.

For a better understanding, consider the following example. Let

$$a_{(1)} = (1, \frac{1}{2})^T, \quad a_{(2)} = (-1, 1)^T, \quad d_1^2 = \frac{5}{4}, \quad d_2^2 = 2.$$

For $\varepsilon_1 = 0.1$ and $\varepsilon_2 = 0.15$ the feasible set M of (4.4) is plotted in figure 4.1 (dashed area). Any point x in M fulfils the disturbed restriction

$$\|a_{(i)} - x\|_2^2 = \tilde{d}_i, \qquad \text{for} \quad i = 1, 2.$$

The (extreme) point of M that is most far away from the original true sensor position ($x^* = (0,0)^T$) is the point I_3.

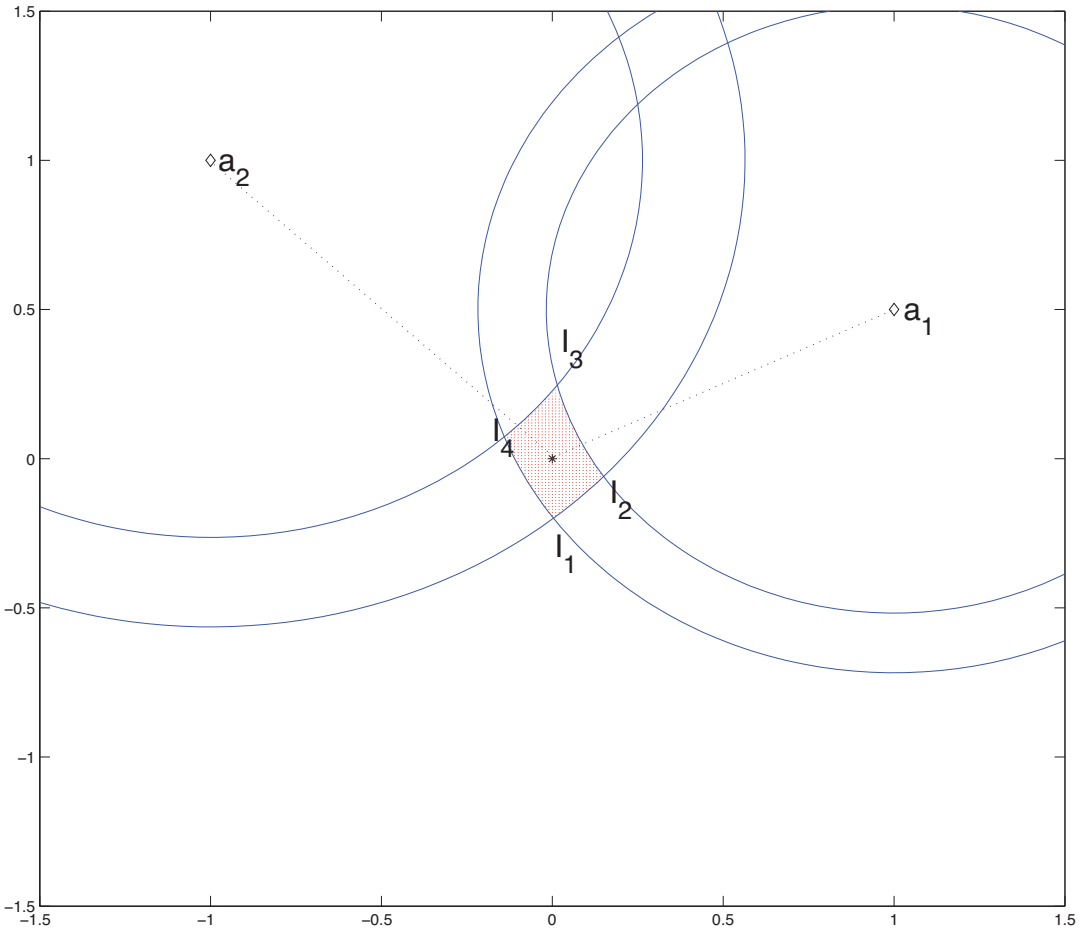


Figure 4.1: Local error bound for the point $(0,0)$ for two adjacent anchors a_1 and a_2.

The procedure is explained next.

Suppose that the point x^* is connected to several anchors. For every pair of anchors $p = (p_1, p_2)^T$ and $q = (q_1, q_2)^T$ to which the sensor x^* is connected, follow the next steps.

1. The exact distances $d_1 = \|p\|_2$ and $d_2 = \|q\|_2$ and positive values ε_1 and ε_2 are given. The following circumferences are of interest:
   (a) GL_1: $(p_1 - x)^2 + (p_2 - y)^2 = (d_1 + \varepsilon_1)^2$
   (b) GL_2: $(q_1 - x)^2 + (q_2 - y)^2 = (d_2 + \varepsilon_2)^2$
   (c) GL_3: $(p_1 - x)^2 + (p_2 - y)^2 = (d_1 - \varepsilon_1)^2$
   (d) GL_4: $(q_1 - x)^2 + (q_2 - y)^2 = (d_2 - \varepsilon_2)^2$
2. These circumferences are intersected and the following points of intersection are obtained:
   (a) $(i_{(1,2)}, i_{(2,1)})$ result from $GL_2 \cap GL_1$,
   (b) $(i_{(2,3)}, i_{(3,2)})$ result from $GL_2 \cap GL_3$.
   (c) $(i_{(3,4)}, i_{(4,3)})$ result from $GL_4 \cap GL_3$.
   (d) $(i_{(1,4)}, i_{(4,1)})$ result from $GL_4 \cap GL_1$.
3. As the values ε_i represent small disturbation in measurements we find the extreme points of M by
   (a) $I_1 = \min\{\|i_{(1,2)}\|_2^2, \|i_{(2,1)}\|_2^2\}$.
   (b) $I_2 = \min\{\|i_{(2,3)}\|_2^2, \|i_{(3,2)}\|_2^2\}$.
   (c) $I_3 = \min\{\|i_{(3,4)}\|_2^2, \|i_{(4,3)}\|_2^2\}$.
   (d) $I_4 = \min\{\|i_{(1,4)}\|_2^2, \|i_{(4,1)}\|_2^2\}$.
4. Then, the solution of (4.5) is obtained by comparing the edges :

$$x_{pq} = \max\{\sqrt{I_1}, \sqrt{I_2}, \sqrt{I_3}, \sqrt{I_4}\}. \tag{4.7}$$

After having found an error bound concerning each pair of anchors p and q, all error bounds are compared and the smallest value gives the local error bound for the solution x of (4.5):

$$x = \min\{x_{pq} \mid p,\ q \in \mathbb{R}^2\ :\ \|p\| < r, \|q\| < r\}\}, \tag{4.8}$$

where r denotes the given radio range.

The single formulas of the step 3. are given next:

$$I_1 = C(A_1 + 2s(p,q)\|p\|\|q\|\sin(p,q)\sqrt{B_1}),$$

$$I_2 = C(A_2 + 2s(p,q)\|p\|\|q\|\sin(p,q)\sqrt{B_2}),$$

where $s(p,q) = sign(\sin(p,q))$ denotes the signum of the sinus of the angle between the vectors p and q and

$$C := \frac{1}{\|p\|^2 + \|q\|^2 - 2\|p\|\|q\|\cos(p,q)},$$

$$A_1 := \varepsilon_2\|p\|(\varepsilon_2 + 2\|q\|)(\|p\| - \|q\|\cos(p,q)) + \varepsilon_1\|q\|(\varepsilon_1 + 2\|p\|)(\|q\| - \|p\|\cos(p,q)) + 2\|p\|^2\|q\|^2\sin^2(p,q),$$

$$B_1 := \varepsilon_2\|p\|(\varepsilon_2 + 2\|q\|)(\|p\| - \|q\|\cos(p,q)) + \varepsilon_1\|q\|(\varepsilon_1 + 2\|p\|)(\|q\| - \|p\|\cos(p,q)) - (\varepsilon_1\|p\| - \varepsilon_2\|q\|)^2 + \|p\|^2\|q\|^2\sin^2(p,q) - (\varepsilon_1^2 - \varepsilon_2^2)(\frac{1}{4}(\varepsilon_1^2 - \varepsilon_2^2) + \varepsilon_1\|p\| - \varepsilon_2\|q\|),$$

$$A_2 := \varepsilon_2\|p\|(\varepsilon_2 + 2\|q\|)(\|p\| - \|q\|\cos(p,q)) + \varepsilon_1\|q\|(\varepsilon_1 - 2\|p\|)(\|q\| - \|p\|\cos(p,q)) + 2\|p\|^2\|q\|^2\sin^2(p,q),$$

$$B_2 := \varepsilon_2\|p\|(\varepsilon_2 + 2\|q\|)(\|p\| - \|q\|\cos(p,q)) + \varepsilon_1\|q\|(\varepsilon_1 - 2\|p\|)(\|q\| - \|p\|\cos(p,q)) - (\varepsilon_1\|p\| + \varepsilon_2\|q\|)^2 + \|p\|^2\|q\|^2\sin^2(p,q) - (\varepsilon_1^2 - \varepsilon_2^2)(\frac{1}{4}(\varepsilon_1^2 - \varepsilon_2^2) + \varepsilon_1\|p\| + \varepsilon_2\|q\|),$$

Similar expressions are given for I_3 and I_4.
For the above example, the values of $I_1, ..., I_4$ are given as follows:

$$I_1 = 0.04$$

$$I_2 = 0.0261$$

$$I_3 = 0.0604$$

$$I_4 = 0.0251.$$

This results in a local error bound value of **0.2458**.

Unfortunately, this local error bound is a very pessimistic upper bound. In practice, the extreme points of the feasible set M seem to be never attained. Moreover, this local error bound is very restrictive. It handles the adjacent vectors as fixed anchors, neglecting the very presumably case that these vectors may be sensors whose positions are as well disturbed.

Another approach to detect in the obtained approximation to the SNLP of those sensors that may be far away from the original true sensors is to analyze how big the trace of the obtained matrix $\tilde{Y}$ of (4.2) is. This was first pointed out in [5]. The basic idea behind this is that if the matrix $\tilde{Y}$ is a rank 1 matrix

$$\tilde{Z} = \tilde{y}\tilde{y}^T$$

holds (cp. (4.2)). In the other cases

$$\tilde{Z} \succeq \tilde{y}\tilde{y}^T$$

holds. This inequality constitutes error analysis for position estimation. For example

$$tr(\tilde{Z} - \tilde{y}\tilde{y}^T) = \sum_{j=1}^{n} (\tilde{Z}_{2j-1,2j-1} + \tilde{Z}_{2j,2j} - \|\tilde{y}_{(j)}\|^2) \tag{4.9}$$

the total trace of the difference matrix measures the efficiency and quality of distance data $\tilde{d}_{ij}$ and $\tilde{d}_{kl}$. In particular, the individual trace $\tilde{Z}_{2j-1,2j-1} + \tilde{Z}_{2j,2j} - \|\tilde{y}_{(j)}\|^2$ helps to estimate the position estimation $\tilde{y}_{(j)}$ for sensor j. For smaller values of the trace, a higher accuracy of the estimation can be osberved.

4.3 Finding a Starting Point for the Rounding Technique

It is interesting how to choose a good starting point for the rounding technique using the information given from the matrix $\tilde{Y}$ resulting from (4.1). Until now, the literature has not paid much attention to this question (only Wolcowicz et al. in [6] compared numerically different choices of starting points derived from the relaxation of (SNL_{EDM})). The most common choice in practice is to set the first column (starting from the second element) from $\tilde{Y}$, i.e. $\tilde{x} = \tilde{y}$. But of course, there is much more information in the matrix which is neglected with this choice.

The question that arises is, how to interpret the given solution matrix $\tilde{Y}$. In the exact case and when the network is uniquely localizable, $Z = yy^T$ holds. Here, it is clear that a solution (vector) to the SNLP is for example either given by the first row (starting from the 2nd entry) of Y or by the square root entries of the diagonal of Z. In the not uniquely localizable case or in the inexact case, it is not clear which entries of the matrix $\tilde{Y}$ represent the searched vector x. Normally it holds $\sqrt{diag(\tilde{Z})} \neq \tilde{y}$. It is not clear which choice for the solution vector is better.

We derived an approximation vector to x from the given matrix $\tilde{Y}$ by using its eigenvalue decomposition in the following way:

Choose the starting point $\tilde{x}$ as the solution of the minimization problem:

$$\begin{aligned} \min \quad & \|\mathrm{ss^T} - \tilde{\mathrm{Y}}\|_{\mathrm{F}} \\ s.t. \quad & s \in \mathbb{R}^2. \end{aligned} \tag{4.10}$$

Therefore $\tilde{Y}$ is projected onto the set of rank 1 matrices. As $\tilde{Y}$ is symmetric (and real) the objective function can be reduced to

$$f(s) = \|s\|_2^4 - 2s^T\tilde{Y}s + \text{trace}(\tilde{Y})$$

and the gradient to

$$g(s) = 4(\|s\|_2^2 s - \tilde{Y}s).$$

The stationary points are subsequently given by the equality

$$\tilde{Y}s = \|s\|_2^2 s. \tag{4.11}$$

This means, the vector s is an eigenvector of $\tilde{Y}$ to the corresponding eigenvalue $\|s\|_2^2$.
As the Hessian is given by

$$H(s) = 4(2ss^T + \|s\|_2^2 I - \tilde{Y})$$

it is easily shown that the solution vector s is constructed by

$$s_{\text{opt}} = v_{\max}\lambda_{\max} \tag{4.12}$$

where $\lambda_{\max}$ denotes the largest eigenvalue of $\tilde{Y}$ and $v_{\max}$ the corresponding normalized eigenvector.

As we propose to improve the obtained approximation from the relaxed problem by a rounding technique, we compared numerically the choice of (4.12) with the standard choice of the corresponding entries in the matrix Y in (4.1) and respectively in the matrix Z in the disturbed (2.6). The choice of (4.12) provided the most accurate solutions.

However, finding an exact solution matrix Y (whose first column contains the exact solution vector x) is equivalent to finding the minimal rank matrix in the intersection of the semidefinite cone and an affine space. This problem is still to be adressed. Recently, [16] [26] proposed randomization methods for SDP rank reduction. These methods can generate a low rank positive semidefinite matrix in an approximate affine space.

Chapter 5

Descent Heuristics for Unconstrained Minimization

In this chapter we present three different heuristics for solving general unconstrained non-convex minimization problems. These results were presented in [10].
The *increased gradient descent* method is a first-order method which tries to minimize a function f and contemporarily maximize the norm of the gradient of f.
The *curvature descent* is a second-order method which can be seen as a mixture of Newton's and Trust Region Methods. Its general idea is to emphasize the attention on to the space of the negative eigenvalues of the Hessian at a given iterate, and not only on the smallest (as it would be in Trust Region Methods). This approach produced the best numerical results for the SNLP as shown in chapter 7. Therefore, we will describe this approach more accurately in the next chapter.
Finally, the *decreasing curvature* tries to use third-order information. It anticipates points with more negative curvature in the Hessian, in order to provide more descent at the next iteration step.

5.1 General Assumptions

Let $f : \mathbb{R}^n \to \mathbb{R}$ be three times continuously differentiable. The problem under consideration is to find – among several local minimizers of f – a local minimizer x^* with a small value $f(x^*)$. We compare several heuristics that do not guarantee to find a global minimizer but that aim to find a local minimizer with "rather small value of f".

The goal is to find "good" local minimizers within a certain domain of attraction given by the initial point. We emphasize that this can only be a heuristics. Nev-

ertheless, numerical examples indicate that the new approaches proposed here often lead to better local solutions than the steepest descent path.

For smooth strictly convex functions f and under standard conditions (regarding the descent) all descent paths eventually lead to the same optimal solution, and Newton's method generally performs very well. Our aim is not to compete with Newton's method but to provide a cheap first order method that is "efficient" *far away* from a local solution when the Hessian of f is not positive definite. Close to a solution, first order methods – such as the ones proposed here – are poor and generally need to be replaced with other approaches.
The following principles for minimizing f starting at a given point $x^0 \in \mathbb{R}^n$ motivate the new approaches:

1. In order to remain within a domain of attraction given by x^0, only descent steps are considered.

2. As the global behaviour – far from a local minimizer – is of interest, only short steps derived from first order approximations are considered.

3. The first order predictions used in this paper are based on the following naive motivation for *non-convex* minimization:

   (a) Aim towards points where the gradient gets larger (and f gets smaller) because large gradients give a hint that a large descent may be possible from there.

   (b) Aim towards points where the negative curvature is amplified (and f gets smaller) because negative curvature might result in long descent steps from there.

   Again, we emphasize that both motivations make little sense for convex minimization.

The following notation is used:

- We will call $g = \nabla f(x)$ the gradient and $H = \nabla^2 f(x)$ the Hessian of f at the point x.

- The orthogonal eigenvalue decomposition of H is given by $H = UDU^T$, i.e. U is a unitary matrix whose columns u^i are eigenvectors to the eigenvalues $D_{ii} = \lambda_i(H) = \lambda_i$. Without loss of generality we assume that the eigenvalues are sorted in decreasing order and the first t are positive (i.e. $\lambda_1 \geq ... \geq \lambda_t \geq 0 > \lambda_{t+1} \geq ... \geq \lambda_n$, $t = 0$ and $t = n$ is, of course, possible).

5.2 Increasing gradient descent

Given an iterate x^k, the progress of a short step along the steepest descent direction is proportional to the norm of $\nabla f(x^k)$. To be able to have a large descent in the next step, it may therefore be of advantage not only to minimize f but also to maximize $\|\nabla f\|$ at the same time. This idea leads to a descent step for the function

$$\varphi(x, \alpha) := f(x) - \alpha \|\nabla f(x)\|^2 \tag{5.1}$$

where $\alpha > 0$ is a parameter that balances the reduction of f versus the increase of $\|\nabla f\|$.
The steepest descent direction for φ is given by

$$\Delta x = -\nabla f(x^k) + 2\alpha \nabla^2 f(x^k) \nabla f(x^k). \tag{5.2}$$

We point out that matrix vector products such as $\nabla^2 f(x^k)v$ can typically be computed (for example by automatic differentiation) cheaply without forming $\nabla^2 f(x^k)$.

The proposed search step is given by the partial derivative $\Delta x = -\nabla_x \varphi(x, \alpha)$ in (5.2) where the choice of $\alpha = \alpha(x)$ depends on x.
Let

$$d^1 := -\frac{\nabla f(x^k)}{\|\nabla f(x^k)\|} \qquad \text{and} \tag{5.3}$$

$$d^2 := \frac{\nabla(\|\nabla f(x^k)\|^2)}{\|\nabla(\|\nabla f(x^k)\|^2)\|} = \frac{\nabla^2 f(x^k)\nabla f(x^k)}{\|\nabla^2 f(x^k)\nabla f(x^k)\|} \tag{5.4}$$

be the normalized steepest descent directions for f and for $-\|\nabla f\|_2^2$.

Any Δx with $\Delta x^T d^1 < 0$ and $\Delta x^T d^2 < 0$ is a descent direction for both, f and $-\|\nabla f\|_2^2$.

The normalized descent direction $\widetilde{\Delta x} := \Delta x / \|\Delta x\|$ is chosen as a nonnegative combination of the vectors d^1 and d^2, i.e.

$$\widetilde{\Delta x} = \beta d^1 + \gamma d^2 \tag{5.5}$$

where $\beta \geq 0$ and $\gamma \geq 0$ depend on α.

A normalized measure of descent for f is given by $\widetilde{\Delta x}^T d^1$ and ranges from 1 (for $\beta = 1,\ \gamma = 0$) to $(d^1)^T d^2$ (for $\beta = 0,\ \gamma = 1$), a negative value of $\widetilde{\Delta x}^T d^1$ indicating

ascent.

To find a meaningful choice of α, or, equivalently of β and γ subject to the condition

$$1 = \|\widetilde{\Delta x}\|^2 = \beta^2 + 2\beta\gamma(d^1)^T d^2 + \gamma^2 \tag{5.6}$$

we require that

$$\widetilde{\Delta x}^T d^1 = \rho + (1-\rho)(d^1)^T d^2 \tag{5.7}$$

for some $\rho \in (0, 1]$.
The value ρ indicates the amount of descent of f compared to the steepest descent.

Lemma 5.1
Depending on the choice of ρ, the search direction $\widetilde{\Delta x}$ is

- *the steepest descent direction for f if $\rho = 1$,*
- *the steepest descent direction for $\|\nabla f\|$ if $\rho = 0$. In the worst case, $\widetilde{\Delta x}$ could be opposite to $-\nabla f$.*

If $\rho > 0.5$ then $\widetilde{\Delta x}$ is a descent direction for the minimization of f.

Proof:
Straightforward from (5.7). □

From the equations (5.6) and (5.7) we can set the values of β and γ in dependence of ρ.
Let

$$c := (d^1)^T d^2 = -\frac{\nabla f(x^k)^T D^2 f(x^k) \nabla f(x^k)}{\|\nabla f(x^k)\| \|D^2 f(x^k) \nabla f(x^k)\|}$$

be the cosine of the angle $\sphericalangle(d^1, d^2)$. The numbers β, γ then solve the system of equations

$$\beta^2 + 2\beta\gamma c + \gamma^2 = 1 \tag{5.8}$$

$$\beta + \gamma c = r := \rho + (1-\rho)c. \tag{5.9}$$

The solution is given by

$$\begin{aligned} \gamma &= \sqrt{\frac{1-r^2}{1-c^2}} = \sqrt{\frac{(1-\rho)(1+\rho+c-\rho c)}{1+c}} \quad \text{and} \\ \beta &= \rho + c(1-\rho-\gamma). \end{aligned} \tag{5.10}$$

For the interesting case that the increase of $\|\nabla f\|$ is given high priority and the descent property is barely guaranteed, i.e. for $\rho = 0.5$ the expression for γ simplifies to

$$\gamma = \sqrt{\frac{3+c}{4+4c}}.$$

The choice of β, γ can be translated back to the choice of α: Multiplying the equation

$$\beta d^1 + \gamma d^2 = \frac{\Delta x}{\|\Delta x\|} = \frac{-\nabla f(x^k) + 2\alpha D^2 f(x^k)\nabla f(x^k)}{\| - \nabla f(x^k) + 2\alpha D^2 f(x^k)\nabla f(x^k)\|}$$

from left with $(\beta d^1 + \gamma d^2)^T$, using the definitions of d^1, d^2 from (5.3) and (5.4), and squaring the result we obtain

$$1 = \frac{(\|\nabla f(x^k)\|(\beta + \gamma c) + 2\alpha\|D^2 f(x^k)\nabla f(x^k)\|(\gamma + \beta c))^2}{\| - \nabla f(x^k) + 2\alpha D^2 f(x^k)\nabla f(x^k)\|^2}.$$

Solving this quadratic equation for α yields

$$\alpha = \alpha(x^k) = \frac{\gamma}{2\beta}\frac{\|\nabla f(x^k)\|}{\|D^2 f(x^k)\nabla f(x^k)\|}. \tag{5.11}$$

Remark 5.2
A descent method for φ based on this choice of α for a given parameter $\rho \in (0, 1]$ is called "increasing gradient descent method with parameter ρ" (briefly "$igd(\rho)$").

For further notation we will denote the given search direction for the iterate x^k by $\Delta x^k_{igd(\rho)} (= \beta d^1 + \gamma d^2)$ as it depends only on ρ (and x^k).

Algorithm 5.3 (Increasing Gradient Descent)
Initialization : Let $x^0 \in \mathbb{R}^n$ and $\rho \in (0.5, 1]$ be given.

For k=0,1,2,... repeat until a stopping criterion is reached

(i) $d^1 = -\frac{\nabla f(x^k)}{\|\nabla f(x^k)\|}$ and $d^2 = \frac{\nabla^2 f(x^k)\nabla f(x^k)}{\|\nabla^2 f(x^k)\nabla f(x^k)\|}$.

(ii) $c = {d^1}^T d^2, \quad \gamma = \sqrt{\frac{(1-\rho)(1+\rho+c-\rho c)}{1+c}}, \quad \beta = \rho + c(1-\rho-\gamma)$.

(iii) $\Delta x^k_{igd(\rho)} = \beta d^1 + \gamma d^2$

(iv) $x^{k+1} = x_k + \lambda_k \Delta x^k_{igd(\rho)}$ where $\lambda_k \approx \underset{\lambda>0}{\operatorname{argmin}} f(x^k + \lambda \Delta x^k_{igd(\rho)})$.

Remark 5.4
The finding of λ_k in step (iv) of algorithm 5.3 results in a global minimization of

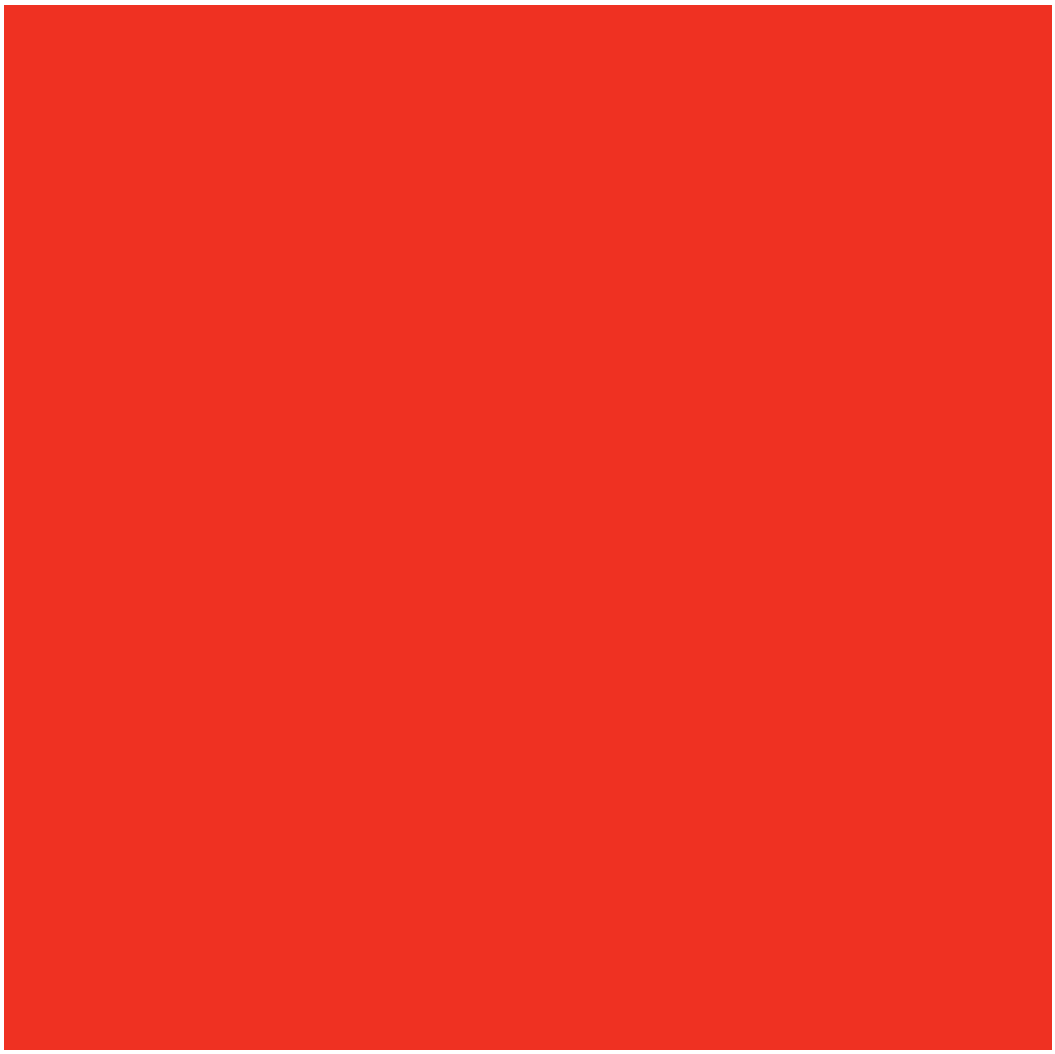
a 1-dimensional unconstrained minimization problem. This problem, also known as *linesearch* is discussed in detail for the Curvature Descent approach in the last section. For the numerical comparison of all methods presented in this chapter we used only constant short steps λ.

5.2.1 First Example

The behaviour of the descent path $igd(\rho)$ with $\rho = 1$ (steepest descent) and $\rho = 0.6$ is illustrated with a simple example: For $x \in I\!R^2$ let

$$f(x) := 10x_1 + x_2 + x_1^2(4 + \frac{x_2^2}{4}).$$

The function f is unbounded below (it tends to $-\infty$ along the line $x_1 = 0,\ x_2 < 0$) and has a local minimizer at $x_1 = -1,\ x_2 = -2$. Figures 5.1 and 5.2 below indicate the starting points for which the steepest descent method ($igd(1)$) and $igd(0.6)$ converge to the local minimizer. For all other starting points the methods converge to the infimum of f (i.e. to $-\infty$).



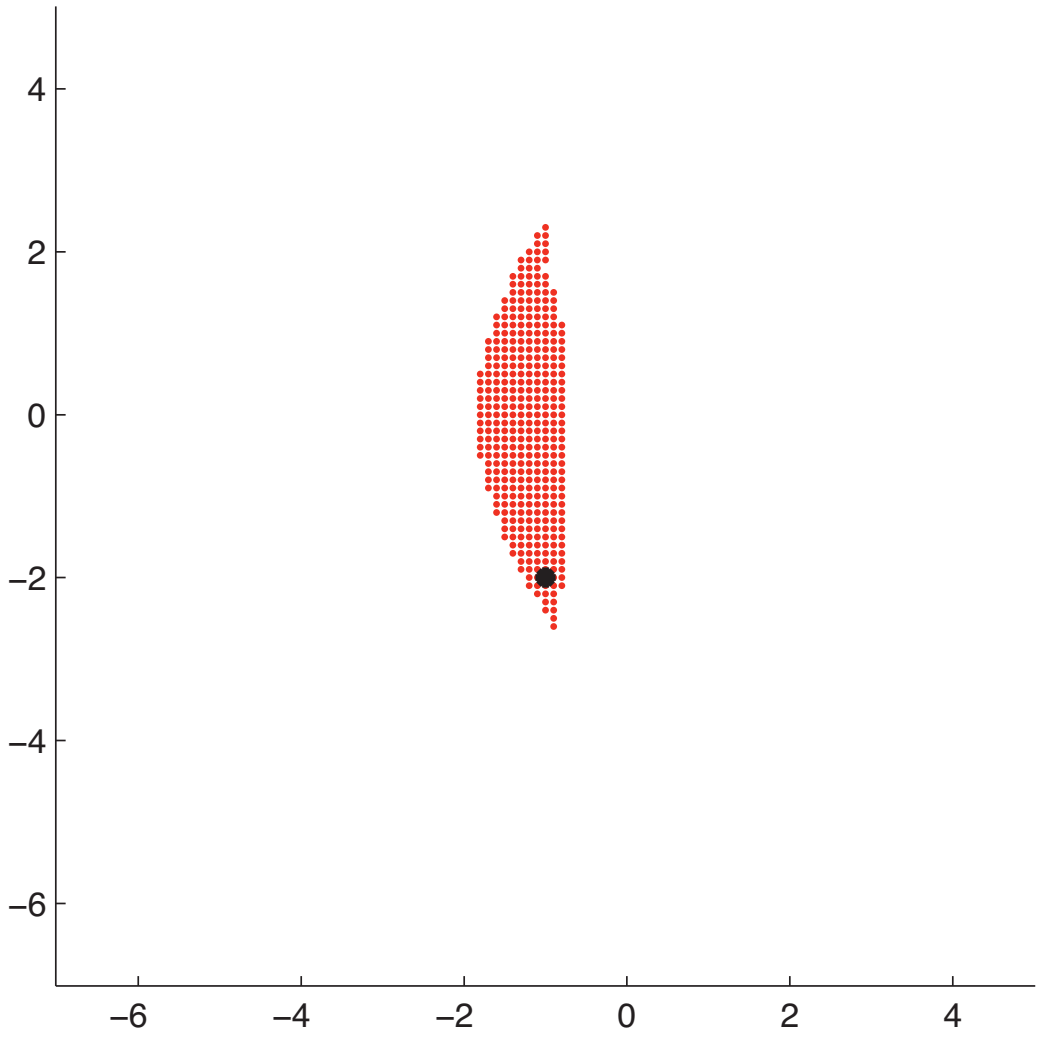


Figure 5.2: Area indicating starting points that converge to the local minimizer for $igd(0.6)$.

5.2.2 Second Example

For illustration we also display the plot of a steepest descent path and the $igd(0.6)$-path for a convex quadratic function f, $f(x) = \frac{1}{2}x_1^2 + x_2^2$.

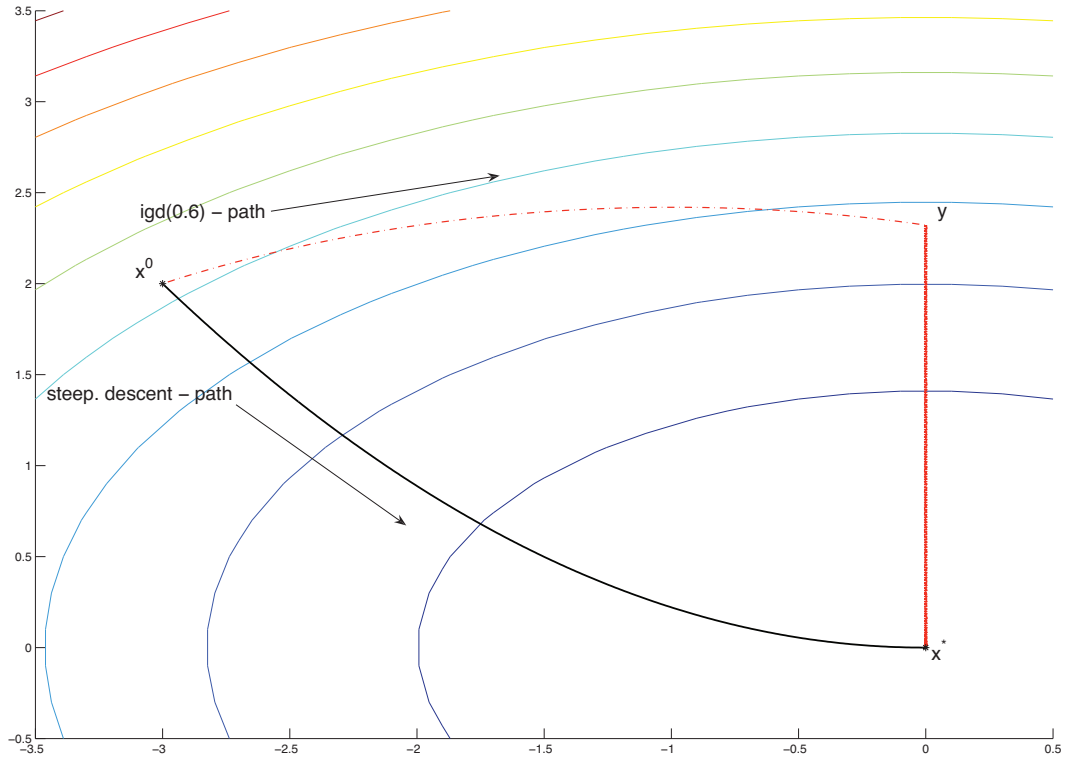


Figure 5.3: Red, dashed: path of $igd(0.6)$, black: path of steepest descent

In this example, the norm of the gradient is increased at first while f is reduced. For a function f that is bounded below, the decrease of both, $-\|\nabla f\|^2$ and f must stop at some point y. When reaching the point y, the direction of increasing gradient points opposite to the descent direction and is thus ignored.

5.3 Curvature descent

In the following two section the descent path is modified based on the use of second and third order derivatives.

Proposition 5.5
Given an iterate x, let $\tilde{g} := U^T g$ (given by the gradient g of f at x and the eigenvalue decomposition of the Hessian $H = UDU^T$) .
For small $\varepsilon > 0$ it holds that

$$f(x + \varepsilon u^i) \approx f(x) + \varepsilon g^T u^i = f(x) + \varepsilon \tilde{g}^T U^T u^i = f(x) + \varepsilon \tilde{g}_i,$$

and

$$\|\nabla f(x+\varepsilon u^i)\|^2 \approx \|\nabla f(x)\|^2 + 2\varepsilon g^T H u^i = \|\nabla f(x)\|^2 + 2\varepsilon \tilde{g}_i D_{ii}.$$

Proof:
Without loss of generality we assume that the components $\tilde{g}_i$ of $\tilde{g}$ satisfy

$$\tilde{g}_i \leq 0.$$

(Else multiply the corresponding column u^i of U by -1). Next, apply Taylor's Theorem. □

Remark 5.6
From proposition 5.5 follows, that when $D_{ii} < 0$ (and $\tilde{g}_i \neq 0$) a selection of $\omega_i > 0$ effects that both, f is locally reduced along $x + \omega_i u^i$, and $\|\nabla f\|^2$ is increased. On the other hand, for $D_{ii} > 0$, an increase of $\|\nabla f\|^2$ along $\omega_i u^i$ is only possible at the expense of increasing f at the same time.

The concept below avoids such search directions u^i as far as possible.

Any search direction $\Delta x \in I\!R^n$ can be written as $\Delta x = \sum_{i=1}^n \omega_i u^i$ with $\omega_i \in I\!R$.

We set

$$\tilde{D}_{ii} := \max\{\varepsilon, D_{ii}\} \tag{5.12}$$

for a small value $\varepsilon > 0$ so that the obtained positive definite matrix

$$\tilde{H} = U\tilde{D}U^T \tag{5.13}$$

defines a descent direction

$$\Delta x_{cd} = -\tilde{H}^{-1} g \tag{5.14}$$

for f.

The search step Δx_{cd} depends on the choice of $\varepsilon > 0$ as long as H is not positive definite. If H is positive definite Δx_{cd} is the Newton step and converges locally quadratically.

Lemma 5.7
Let H be not positive definite. As $\varepsilon \to 0$ it follows that $\Delta x_{cd}/\|\Delta x_{cd}\|$ converges to the space

$$Q := span\{u^i \mid D_{ii} < 0\}.$$

Proof:
As the u^i are an orthogonal basis of eigenvectors, we find values $\tau_i \in \mathbb{R}$ such that

$$g = \sum_{i=1}^{n} \tau_i u^i = U\tau,$$

with $\tau = (\tau_1, .., \tau_n)^T$.
It then holds

$$\lim_{\varepsilon \to 0} \frac{\Delta x_{cd}}{\|\Delta x_{cd}\|} = \lim_{\varepsilon \to 0} \frac{-U\tilde{D}^{-1}U^T g}{\|\Delta x_{cd}\|} = \lim_{\varepsilon \to 0} \frac{-U\tilde{D}^{-1}\tau}{\|\Delta x_{cd}\|}.$$

Without loss of generality, let the first t eigenvalues of D be positive. It follows that

$$\begin{aligned}
\Delta x_{cd} &= \sum_{k=1}^{t} \frac{\tau_k}{\lambda_k} u^k + \sum_{l=t+1}^{n} \frac{\tau_l}{\varepsilon} u^l \\
&= \frac{1}{\varepsilon}\{\varepsilon \sum_{k=1}^{t} \frac{\tau_k}{\lambda_k} u^k + \sum_{l=t+1}^{n} \tau_l u^l\} \qquad \text{and} \\
\|\Delta x_{cd}\| &= \sqrt{\sum_{i=1}^{n} [\sum_{k=1}^{t} \frac{\tau_k}{\lambda_k} u^k(i) + \sum_{l=t+1}^{n} \frac{\tau_l}{\varepsilon} u^l(i)]^2} \\
&= \frac{1}{\varepsilon} \sqrt{\sum_{i=1}^{n} \{\varepsilon^2 [\sum_{k=1}^{t} \frac{\tau_k}{\lambda_k} u^k(i)]^2 + 2\varepsilon [(\sum_{k=1}^{t} \frac{\tau_k}{\lambda_k} u^k(i))(\sum_{l=t+1}^{n} \tau_l u^l(i))] + [\sum_{l=t+1}^{n} \tau_l u^l(i)]^2\}}.
\end{aligned}$$

Using the latter and taking the limes results in

$$\lim_{\varepsilon \to 0} \frac{\Delta x_{cd}}{\|\Delta x_{cd}\|} = \frac{\sum_{l=t+1}^{n} \tau_l u^l}{\sqrt{[\sum_{l=t+1}^{n} \tau_l u^l]^2}},$$

which corresponds to a vector in the space Q. □

For small $\varepsilon > 0$ the descent step Δx_{cd} lies in some sense near the space Q of negative curvature. Therefore, it is called Curvature Descent step.
In numerical examples small positive values of ε produced slightly better local minimizers than the limiting direction $\Delta x_{cd}/\|\Delta x_{cd}\|$ obtained for $\varepsilon \to 0$.

Algorithm 5.8 (Curvature Descent)
Initialization : Let $x^0 \in \mathbb{R}^n$ and $\varepsilon > 0$ be given.

For k=0,1,2,... repeat until a stopping criterion is reached

(*i*) $g = \nabla f(x^k)$, $H = \nabla^2 f(x^k) = UDU^T$.
(*ii*) $\tilde{D}_{ii} = \max\{\varepsilon, D_{ii}\}$, $\tilde{H} = U\tilde{D}U^T$.
(*iii*) $\Delta x^k_{cd} = -\tilde{H}^{-1}g$.
(*iv*) $x^{k+1} = x_k + \lambda_k \Delta x^k_{cd}$ where $\lambda_k \approx \underset{\lambda>0}{\operatorname{argmin}} f(x^k + \lambda \Delta x^k_{cd})$.

5.4 Decreasing Curvature Descent

The approach in sections 5.2 and 5.3 was motivated by aiming towards points from which a large descent can be anticipated since the norm of the gradient is increasing. This approach can be generalized aiming at areas with "more negative curvature". At points with negative curvature another big descent to the next minimizer can be anticipated. Both anticipations are purely heuristically, but nevertheless numerical experiments indicate a good performance.
Let

$$\tilde{\varphi}(x, \kappa) := f(x) - \kappa\chi(\nabla^2 f(x))$$

where

$$\chi(H) := \sum_{i=1}^{n} \min\{0, \lambda_i(H)\}^2$$

and $\lambda_i(H)$ denotes the i-th eigenvalue of the symmetric matrix H. Hence, χ is the sum of squares of the negative eigenvalues of H.
Let $\sigma : I\!R^n \to I\!R$ be the symmetric (in the sense of permutation) function

$$\sigma(x) := \sum_{i:x_i<0} x_i^2.$$

Denoting $\lambda : \mathbb{S}_n \to \mathbb{R}^n$ the function which assigns to every symmetric matrix H the vector of its eigenvalues, i.e. $\lambda(H) = (\lambda_1(H), ..., \lambda_n(H))^T$, we can set the symmetric spectral function corresponding to χ to

$$\sigma(\lambda(H)) = \chi(H) = \sum_{i:\lambda_i(H)<0} \lambda_i(H)^2.$$

By Theorem 1.1. in [14] the function $\chi(H)$ is differentiable, and its derivative is given by

$$\frac{\partial d}{\partial H}\chi(H) = \frac{\partial d}{\partial H}(\sigma \circ \lambda)(H) = U(Diag(\sigma'(\lambda(H))))U^T = U(Diag(2\lambda_-(H)))U^T,$$

where $\lambda_-(H) = (0, ...0, \lambda_{t+1}(H), ..., \lambda_n(H))^T$ (denoting the vector containing only the negative eigenvalues of H, elsewhere 0). We also write shortly

$$H_- = U(Diag(\lambda_-(H)))U^T.$$

In consequence, the i-th component of the derivative of $\tilde{\varphi}(x,\kappa)$ is given by

$$(Df(x))_i - 2\kappa(\nabla^2 f(x))_- \bullet \Delta H_i.$$

where $\Delta H_i := \frac{\partial}{\partial x_i}\nabla^2 f(x)$.
Let $\hat{d}^3$ be the vector with components

$$\hat{d}^3_i = (\nabla^2 f(x))_- \bullet \Delta H_i$$

and $d^3 = \hat{d}^3/\|\hat{d}^3\|$. As in Section 5.2 we construct a search direction Δx having positive scalar products with d^1 and d^3. In contrast to Section 5.2 we do not choose a nonnegative combination of d^1 and d^3 but restrict the search direction to the space Q.

Therefore, we obtain a search direction $\Delta x_{cdc} = s$ as a solution of

$$\min_{(s,t)\in\mathbb{R}^{n+1}} \{-t \mid s^T d^3 \geq t, s^T d^1 \geq t, s^T s \leq 1, s \in Q\} \tag{5.15}$$

maximizing the cosine of the angles $\sphericalangle(\Delta x, d^1)$ and $\sphericalangle(\Delta x, d^3)$ over Q.

Note that this problem is a convex problem, and Slater's condition is satisfied for $(s,t) = 0 \in \mathbb{R}^{n+1}$. Thus, any point satisfying the KKT-conditions is a local (and global) solution. In our numerical tests, we simply used Sedumi ([23]) for solving this subproblem. By exploiting the structure of the subproblem, there certainly exist more efficient solutions.

Algorithm 5.9 (Decreasing Curvature Descent)
Initialization : Let $x^0 \in \mathbb{R}^n$ and $c > 0$ be given.

For k=0,1,2,... repeat until a stopping criterion is reached

(*i*) $d^1 = -\nabla f(x^k)$, $\hat{d}^3_i = (\nabla^2 f(x^k))_- \bullet (\frac{\partial}{\partial x_i}\nabla^2 f(x^k))$, $d^3 = \frac{\hat{d}^3}{\|d^3\|}$.

(*ii*) Set (s,t) by solving (5.15).

(*iii*) If $t < c$, set $\Delta x^k_{dcd} = \Delta x^k_{cd}$ from algorithm 5.8.
Else set $\Delta x^k_{dcd} = s$.

(*iv*) $x^{k+1} = x_k + \lambda_k \Delta x^k_{dcd}$ where $\lambda_k \approx \underset{\lambda>0}{\operatorname{argmin}} f(x^k + \lambda \Delta x^k_{dcd})$.

5.4.1 Example plot

Figure 5.4 refers to the same example as Figures 5.1 and 5.2. The dark shaded area in Figure 5.4 is the set of starting points from which the Curvature Descent method converges to the local minimizer $(-1,-2)^T$ while missing the points with

lower objective value along the negative x_2-axis. Comparing this plot with Figure 5.2 seems to indicate that the Curvature Descent approach is inferior to $igd(0.6)$. However, more difficult test problems indicate otherwise.
We point out that the dcd-method does not make sense for two-dimensional examples, and it is therefore not plotted here.

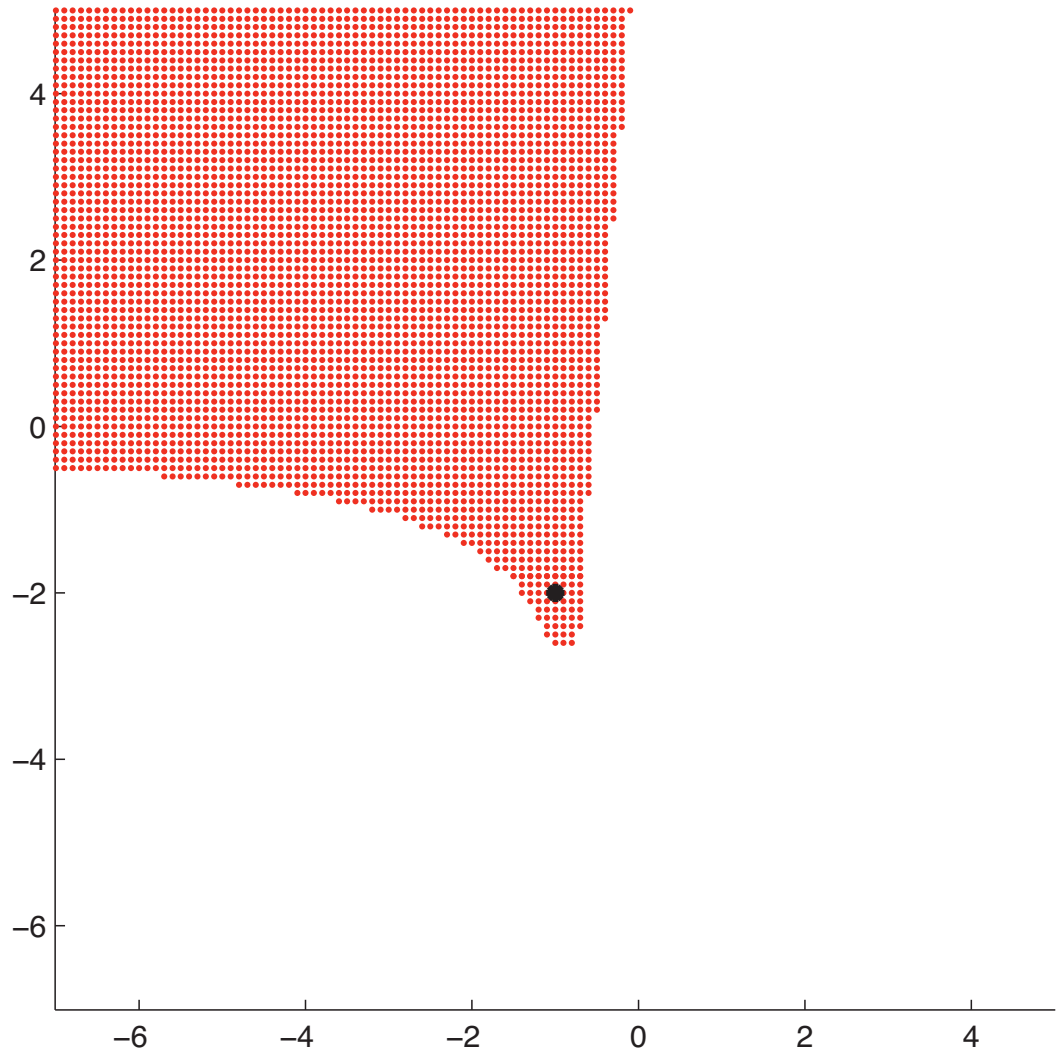


Figure 5.4: Area indicating starting points that convergence to the local minimizer for cd.

Remark 5.10
We implemented the three methods for the rounding technique for the SNLP, choosing the starting point as stated in section 4.3. Details of the numerical

results are shown in the next chapter. As mentioned before, the search direction Δx_{cd} leads in general to the most accurate results. We will therefore explain some more details regarding the application to the SNLP in the next section.

Chapter 6

Curvature Descent for the SNLP

This chapter describes in more details how the Curvature Descent method proposed in chapter 5 was employed for solving the SNLP. The steps of algorithm 5.8 are discussed more extensively and some details to the implementation are given.
It is important to repeat, that the Curvature Descent is a second order method, where at each iterate an eigenvalue decomposition of the Hessian is inevitable. Although, the great advantage for the SNLP is, that using the fact that the Hessian is sparse, much computation time can be saved.

6.1 Exploiting Sparsity of the Hessian

The numerical results showed that the Curvature Descent in section 5.3 was the most effective method when looking for precision in the solution. In this section we will focus on the fact, that the sparsity of the Hessian in (4.3) can be used to save computation time. We will therefore describe in detail the function, the gradient and the Hessian.
If we rename some variables, the function to minimize would be the following:

$$\begin{aligned} f(x) &= \sum_{i=1}^{p} \|A^{(i)} \bullet xx^T - b^{(i)}\|_2^2 \\ &= \sum_{i=1}^{p} \|x^T A^{(i)} x - b^{(i)}\|_2^2, \end{aligned} \tag{6.1}$$

where p denotes the number of given distance information, meanly $\mid N_x \mid + \mid N_a \mid$, $A^{(i)}$ the corresponding matrix for the linear constraint and $b^{(i)}$ the corresponding (squared) disturbed distance information. Note that the matrices $A^{(i)}$ are $2n$-dimensional, because they are derived from the matrices B_{ij} and B^a_{lk} neglecting

the first row and column (cp. (4.1) and (4.3)).

The number of summands p depends on the radio range, and therefore on the given distance information, it always holds

$$p \leq n^2 + nm,$$

where n is the number of (unknown) sensors and m the number of anchors.

The gradient and the Hessian are then:

$$\begin{aligned} g(x) &= 4\sum_{i=1}^{p}(x^T A^{(i)} x - b^{(i)})(A^{(i)}x) \qquad \text{and} \\ H(x) &= 4\sum_{i=1}^{p}\{2A^{(i)}x(A^{(i)}x)^T + (x^T A^{(i)} x - b^{(i)})A^{(i)}\}. \end{aligned} \tag{6.2}$$

To exploit sparsity of the matrices $A^{(i)}$, we introduce 2 vectors

$$\begin{aligned} a_1 &= (1, 0, -1, 0)^T \\ a_2 &= (0, 1, 0, -1)^T \end{aligned} \tag{6.3}$$

When a matrix $A^{(i)}$ corresponds to a condition $\|x_{(p)} - x_{(q)}\|_2^2$ for some $(p, q) \in N_x$, the term $x^T A^{(i)} x$ can be rewritten as

$$x^T A^{(i)} x = (a_1^T x_{pq})^2 + (a_2^T x_{pq})^2 \tag{6.4}$$

where $x_{pq} = (x_{2p-1}, x_{2p}, x_{2q-1}, x_{2q})^T$.

6.2 Line Search

At step (iv) in algorithm 5.8 we did not determine the step size λ_k exactly, as the 1-dimensional function $f(x^k + \lambda \Delta x_{cd}^k)$ is not convex (for simplicity we will set $d^k = \Delta x_{cd}^k$ throughout this section). Instead we approximated the value

$$\lambda_k \approx \operatorname*{argmin}_{\lambda > 0} f(x^k + \lambda d^k),$$

claiming that the Wolfe conditions hold:

$$f(x^k + \lambda_k d^k) \leq f(x^k) + c_1 \lambda_k {d^k}^T \nabla f(x^k) \tag{W1}$$

$${d^k}^T \nabla f(x^k + \lambda_k d^k) \geq c_2 {d^k}^T \nabla f(x^k) \tag{W2}$$

for fixed values $0 < c_1 < c_2 < 1$.

Condition W1 is known as Armijo rule and ensures that λ_k decreases f "sufficiently". Condition (W2) is the so-called curvature condition, which forces that the slope of the function $\phi(\lambda) = f(x^k + \lambda d^k)$ at λ_k is c2 times greater than at $\lambda = 0$.
In practice, one can choose the value λ_k with a backtracking procedure to guarantee the Wolfe conditions. This means, a start value $\bar{\lambda}$ is set and a bisection method is executed until the desired conditions are fulfilled.

Using the Wolfe conditions in the line search, the Curvature Descent Method converges to a stationary point of f. For details see e.g. proposition 6.2.4 in [11].

Getting close to a local minimum, Curvature Descent is exactly Newton's Method and converges therefore quadratically.

Chapter 7

Numerical Examples

To complete the analysis of the SNLP with the proposed Rank 1 Approach we present some numerical examples.

For our test problems we randomly generated sensors and anchors in $[-0.5, 0.5]^2$. We then computed the distance information according to a given radio range and added a measurement error in the following way:

$$\tilde{d}_{ij} = d_{ij}(1 + randn(1) * nf) \quad \forall \quad (i,j) \in N_x$$

$$\tilde{d}_{lk} = d_{lk}(1 + randn(1) * nf) \quad \forall \quad (l,k) \in N_a$$

where nf denotes the noise factor (in most problems $nf = 0.1$, which represents up to 10% measurement error) and randn is a Matlab function, whose output is a pseudo-random value drawn from a normal distribution with mean 0 and standard deviation.

We then treated the sensors as unknown, solved the resulting SDP problem given by (4.1) with Sedumi and derived from the resulting matrix a good initial approximation to the $2n$ dimensional vector of unknown sensors $x_{(1)}, ..., x_{(n)}$.

The following notation will be used through this chapter:

- $\tilde{x}$ denotes the vector obtained from $\tilde{Y}$ in (4.2) by (4.12).
- $\hat{x}$ is exactly the vector $\tilde{y}$ in (4.2).
- x_{SDP2} denotes the vector obtained from X in the matrix Z from (2.6) used in [5].
- n denotes the number of unknown sensors.
- m denotes the number of anchors.

- rr denotes the radio range.
- nf denotes the noise factor.

7.1 Example of the whole procedure

To give an overview of the procedure of solving the SNLP we will discuss a random problem in each single step. Consider the following network consisting of 50 sensors, 3 anchors and a radio range of 0.45.

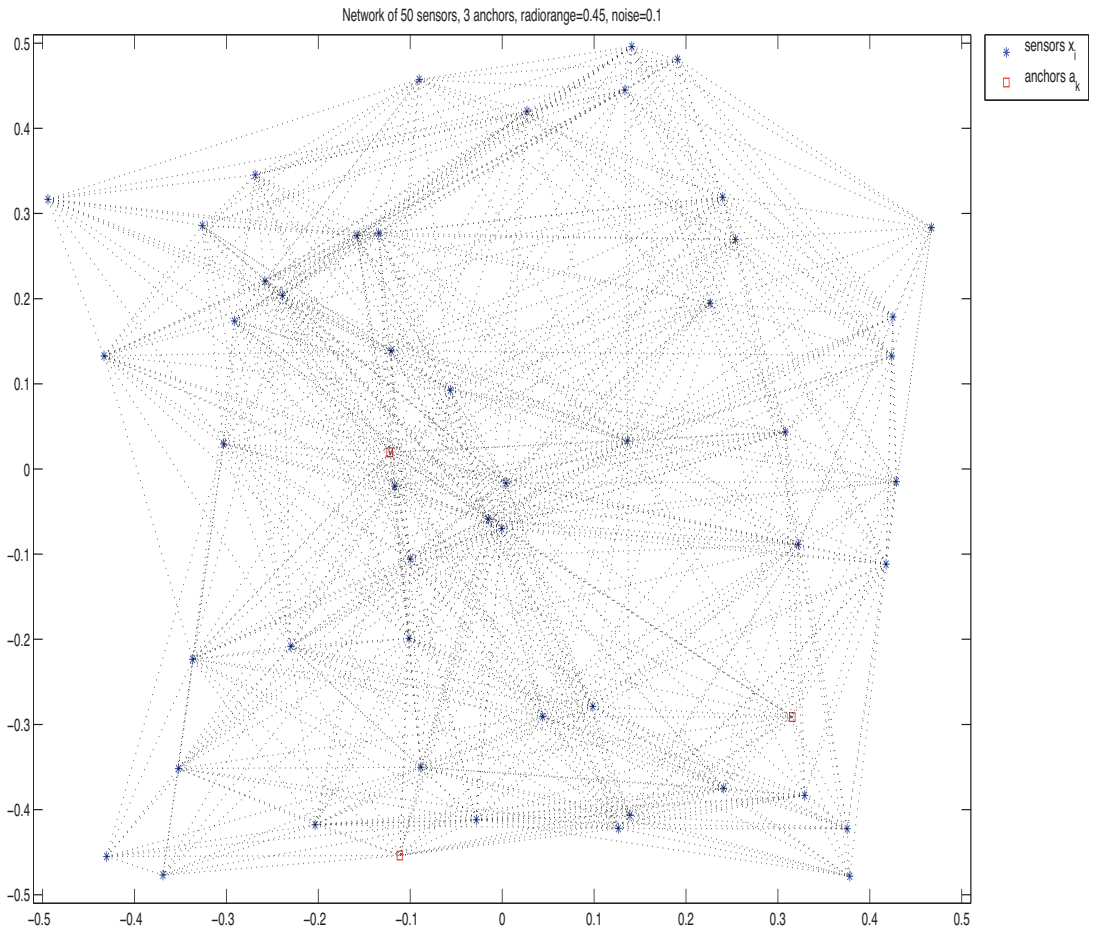


Figure 7.1: Example network.

Up to 10% of measurement errors were added. The solution of the SDP of (4.1)

leads to a matrix $\tilde{Y} \in \mathbb{R}^{101\times 101}$.
Let $\tilde{x}$ be an approximation to the exact solution as described in (4.12). Figure 7.2 shows the approximated sensors $\tilde{x}_{(i)}$ of $\tilde{x}$ marked with a green circle.
They are connected with a blue line to the respective true sensor $x_{(i)}$ which is marked with a blue star.

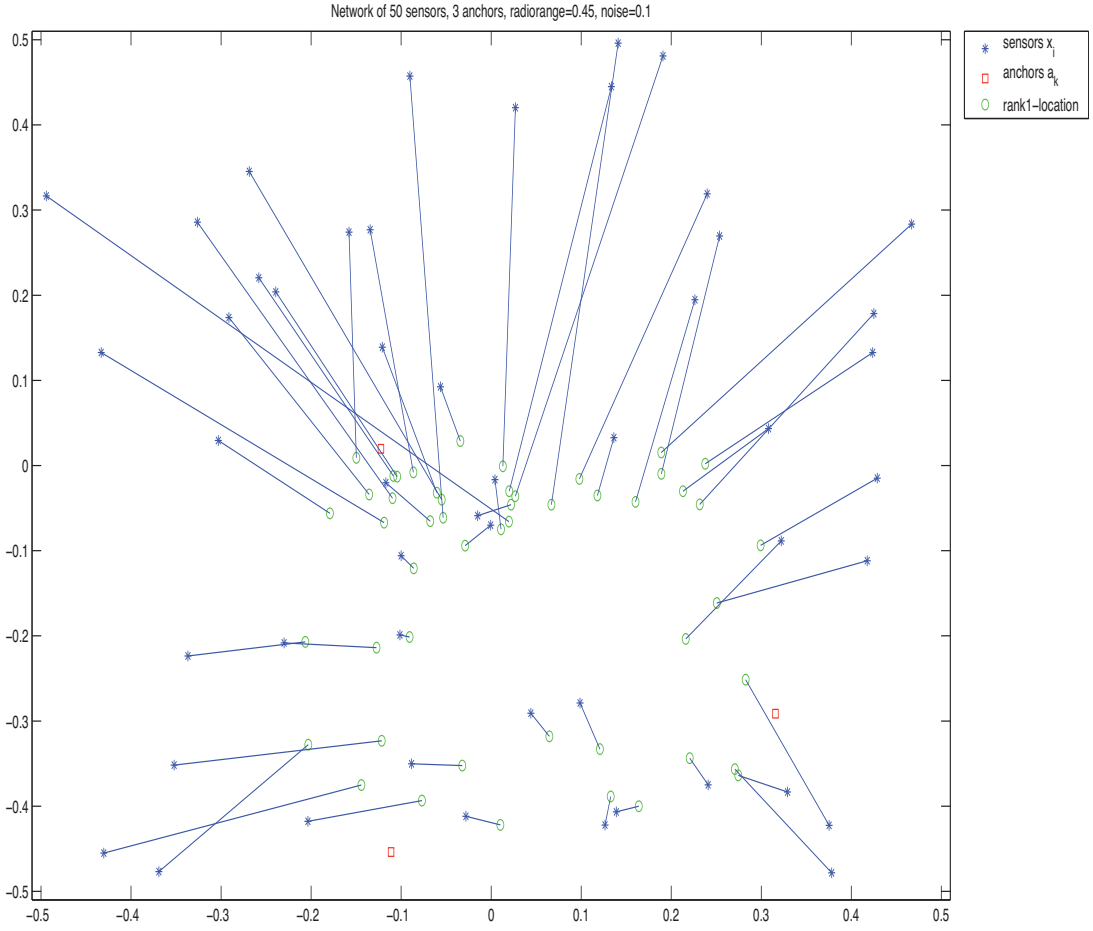


Figure 7.2: Solution of the rank 1 relaxation.

The vector $\tilde{x}$ is then taken as the starting point for the unconstrained least squares problem (4.3). Using the procedure Curvature Descent leads to a solution vector $\hat{x}$, whose (2 dimensional) components are plotted in figure 7.3 and marked again by a green circle. To see the improvement of the approximation every vector $\hat{x}_{(i)}$ is connected by a blue line to the respective true solution $x_{(i)}$.

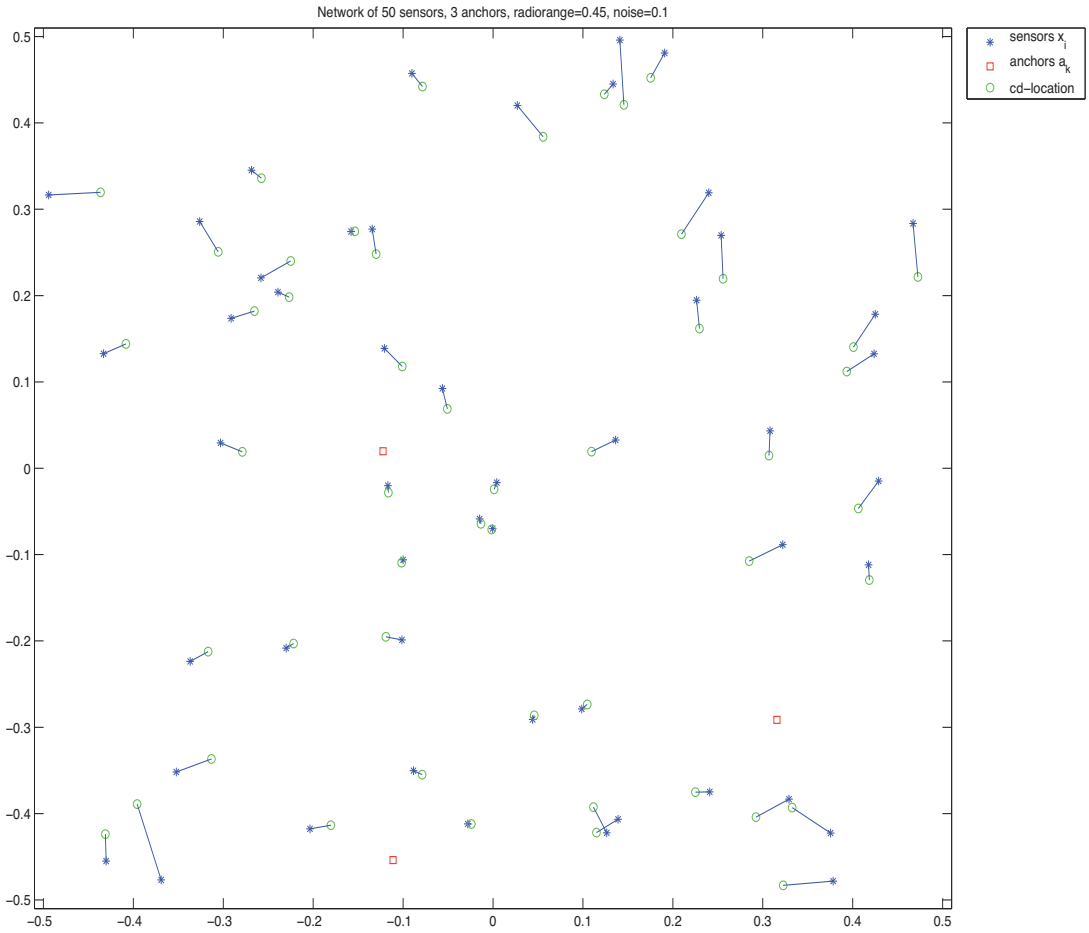


Figure 7.3: Solution refinement using curvature descent.

We repeat that it makes no sense to expect to find the true solution, as the problem is disturbed. In fact, as a measure of accuracy of our numerical tests we took a "normalized final accuracy". This is computed as follows: First, the data of the exact problem is used as a starting point for the steepest descent method for minimizing the sum of squares of the (perturbed) residuals (4.3). The result

yields an (typically very good) upper bound for the global minimizer of (4.3). The normalized final accuracy then lists how far the final accuracy deviates from this upper bound, negative values indicating that a solution was found which is better than the upper bound.

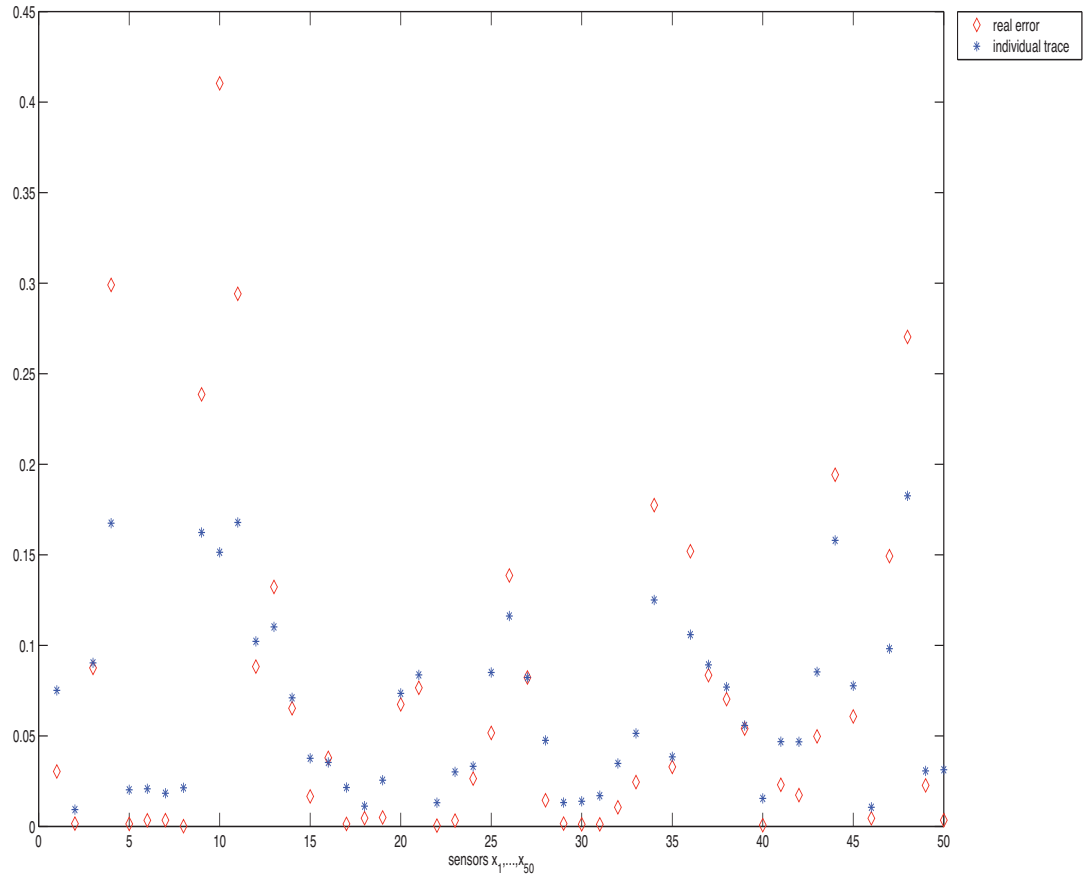


Figure 7.4: Individual trace compared to real error.

For this example, the normalized final accuracy was 0.3156×10^{-5}. To complete the analysis we show the individual trace test, as mentioned in section 4.2. Figure 7.4 shows at the horizontal axe the number of the sensors. The blue stars represent the individual trace of (4.9). The red diamonds represent the true error of the respective sensor approximation (the first column/row of the matrix $\tilde{Y}$ obtained by the relaxation) to the true sensor.

7.2 Comparing the different Descent Heuristics

In this section the approaches described in the last chapter on random disturbed SNLP test problems are compared.

Normally, the problem has many local minimizers. We present our computational results comparing the steepest descent method ($\Delta x_{igd(1)}$), with the other approaches: $\Delta x_{igd(\rho)}$ ($\rho = 0.6$), Δx_{cd} and Δx_{dcd}. We used a linesearch with short steps and compared the quality of the solution to the rounding technique presented in [15] (abbreviated here as SDP2). The table lists the "normalized final accuracy" obtained for each of the five approaches.

All approaches were stopped when a local minimizer was reached. The final error therefore does not depend on the local behaviour of the method but on its global performance.

For the search direction Δx_{dcd} we used the fact that the Hessian is sparse. The computation of the third derivative was very cheap in this case. The Δx_{dcd}-step was changed to Δx_{cd} when the optimal value t in (5.15) was less than 0.1.

Table 7.1 shows the results for 20 random problems in different dimensions starting from the point x_{SDP2}. To allow a better representation, all showed values were first multiplied with 10^2.

n, m, rr, nf	SDP2	Δx_{cd}	Δx_{dcd}	$\Delta x_{igd(1)}$	$\Delta x_{igd(0.6)}$
50, 5, 0.3, 0.15	1.2048	**0.7212**	**0.7212**	0.9399	0.7409
20, 3, 0.35, 0.1	0.1031	**-1.36e-05**	0.0585	0.0883	0.0593
30, 3, 0.2, 0.1	0.0964	**0.0799**	0.3884	0.0931	0.0888
30, 3, 0.25, 0.1	0.1066	0.0839	0.0930	0.1079	**0.0657**
30, 3, 0.3, 0.1	0.0140	0.7261	0.0575	0.0096	**0.0087**
30, 3, 0.3, 0.15	0.6029	0.4813	**0.0093**	0.5070	0.4813
45, 4, 0.35, 0.1	**-1.59e-05**	**-1.59e-05**	**-1.59e-05**	**-1.59e-05**	**-1.59e-05**
45, 5, 0.3, 0.1	0.0598	0.0598	**0.0563**	0.0781	0.0598
45, 5, 0.4, 0.15	**-1.13e-07**	**-1.13e-07**	**-1.13e-07**	**-1.13e-07**	**-1.13e-07**
50, 3, 0.2, 0.1	0.0663	**0.0379**	0.1412	0.0737	0.0380
50, 5, 0.25, 0.15	1.1010	**0.4588**	0.0510	0.6390	0.4987
50, 5, 0.3, 0.1	**-5.12e-05**	**-5.12e-05**	**-5.12e-05**	**-5.12e-05**	**-5.12e-05**
50, 5, 0.3, 0.15	0.8397	**-0.0018**	0.1710	0.3444	1.0803
60, 6, 0.2, 0.15	0.0075	**-0.0029**	0.0147	0.0800	0.0117
60, 6, 0.3, 0.1	**-1.08e-07**	**-1.08e-07**	0.4051	**-1.08e-07**	**-1.08e-07**
80, 5, 0.25, 0.1	0.0374	**0.0059**	0.0085	0.2181	0.0330
80, 8, 0.25, 0.1	**0.0843**	**0.0843**	0.3648	**0.0843**	**0.0843**
120, 10, 0.3, 0.1	**-4.10e-05**	**-4.10e-05**	**-4.10e-05**	**-4.10e-05**	**-4.10e-05**
120, 10, 0.1, 0.2	**0.0045**	0.0120	0.0667	0.0133	0.0085

Table 7.1: Normalized final accuracy multiplied with 10^2. Starting point: x_{SDP2}

Table 7.1 is representative for most of the problems we tested with other combinations of noise factor, radio range and dimension. As our approaches are just heuristically it is clear, that we cannot expect that cd will always yield better (in the sense of a smaller residuum) solutions. Nevertheless we can see a trend to more accurate results.

Next, we show the results for the same problems starting from the point $\tilde{x}$.

n, k, rr, nf	SDP2	Δx_{cd}	Δx_{dcd}	$\Delta x_{igd(1)}$	$\Delta x_{igd(0.6)}$
50, 5, 0.3, 0.15	0.9473	0.9417	**0.7230**	1.733235	**0.7230**
20, 3, 0.35, 0.1	0.1031	**-1.35e-05**	0.0585	**-1.35e-05**	**-1.35e-05**
30, 3, 0.2, 0.1	0.0964	**0.0789**	0.1618	0.0876	0.0815
30, 3, 0.25, 0.1	0.1066	**0.0647**	0.0907	**0.0647**	**0.0647**
30, 3, 0.3, 0.1	**0.0140**	0.0848	0.0399	0.0848	0.0848
30, 3, 0.3, 0.15	**0.4813**	**0.4813**	**0.4813**	**0.4813**	**0.4813**
45, 4, 0.35, 0.1	**-1.59e-05**	**-1.59e-05**	**-1.59e-05**	**-1.59e-05**	**-1.59e-05**
45, 5, 0.3, 0.1	0.0598	0.0598	**0.0563**	0.0598	0.0598
45, 5, 0.4, 0.15	**-1.13e-07**	**-1.13e-07**	**-1.13e-07**	**-1.13e-07**	**-1.13e-07**
50, 3, 0.2, 0.1	0.0663	**0.0372**	0.0502	**0.0372**	**0.0372**
50, 5, 0.25, 0.15	1.1009	**0.1575**	**0.1575**	**0.1575**	**0.1575**
50, 5, 0.3, 0.1	**-6.74e-06**	**-6.74e-06**	**-6.74e-06**	**-6.74e-06**	**-6.74e-06**
50, 5, 0.3, 0.15	0.83977	**- 0,0018**	0.1709	0.1709	0.1709
60, 6, 0.2, 0.15	0.0075	**-0.0029**	0.0220	0.0003	**-0.0029**
60, 6, 0.3, 0.1	**-1.08e-07**	**-1.08e-07**	0.4051	**-1.08e-07**	**-1.08e-07**
80, 5, 0.25, 0.1	0.0374	**0.0059**	0.0075	0.0298	**0.0059**
80, 8, 0.25, 0.1	**0.0843**	**0.0843**	0.3637	**0.0843**	**0.0843**
120, 10, 0.3, 0.1	**-4.10e-05**	**-4.10e-05**	**-4.10e-05**	**-4.10e-05**	**-4.10e-05**
120, 10, 0.1, 0.2	0.0045	**0.0021**	0.0408	0.0026	**0.0021**

Table 7.2: Normalized final accuracy multiplied with 10^2. Starting point for the heuristics: $\tilde{x}$

The latter tables 7.1 and 7.2 are representative for most of the problems we tested. It can be seen a trend towards more accurate solutions by taking the starting point $\tilde{x}$ derived from (4.12).

In many cases, we find not only a solution with a slightly better residuum but also with very different local minimizers. We show the graphical interpretation of the first example in the table above. Figures 7.5 and 7.6 show the position of the true sensors with a star, the anchors with a square.
Sensors within radio range of each other are connected by a dashed line. The true solutions are marked by a star, the computed solutions by a circle that is connected to the corresponding star by a blue line.

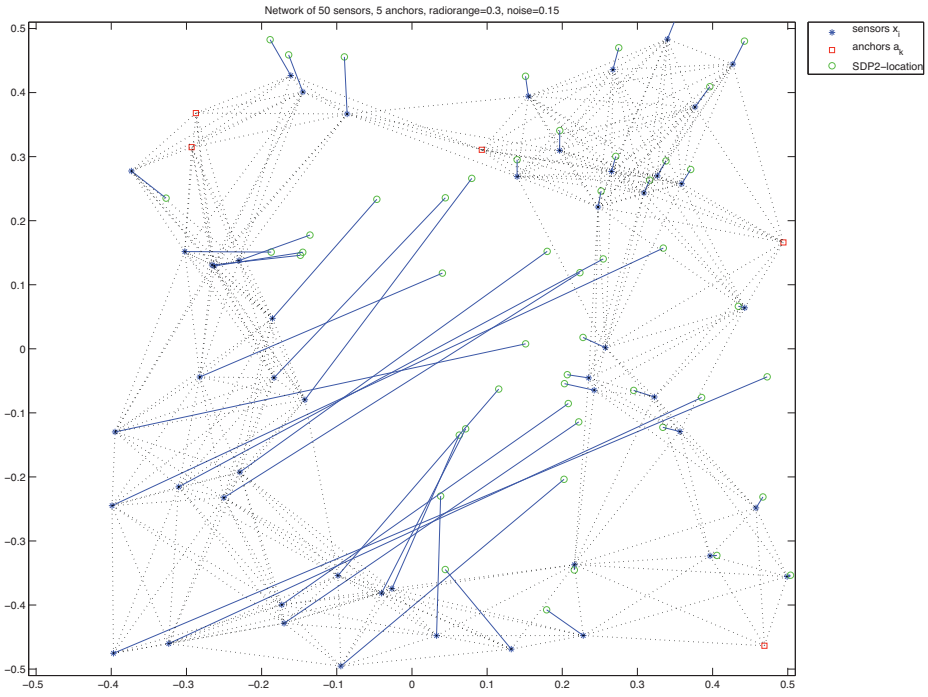


Figure 7.5: SDP2 approach. Approx. time of computation : 3 sec.

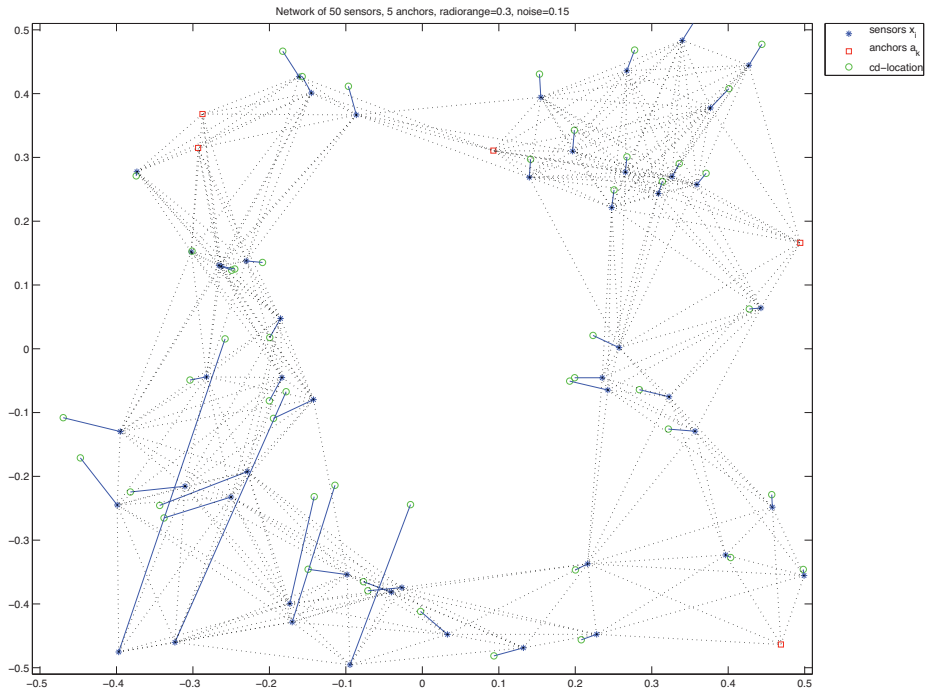


Figure 7.6: Curvature Descent Δx_{cd}. Approx. time of computation: 9 sec.

7.3 Results for Curvature Descent

For completeness we will summarize the results of the Curvature Descent method. The next table shows in every row the average result for 100 test problems for certain combination of number of unknown sensors, number of anchors, radio range and noise factor. An average normalized final accuracy is given as well as an average computation time. These results are again compared to the rank 2 relaxation of [5].

	SDP2		Δx_{cd}	
n, m, rr, nf	average acc.	average *c.t.*	average acc.	average *c.t.*
30, 3, 0.25, 0.1	0.0599	1.9	0.0269	24.3
30, 3, 0.35, 0.1	0.2844	3	0.2370	13.5
40, 10, 0.2, 0.1	0.0158	1.9	0.0065	25.9
45, 4, 0.4, 0.15	0.4383	2	0.1422	7.2
45, 5, 0.3, 0.1	0.1644	1.6	0.09e-2	10.2
50, 3, 0.3, 0.2	1.4023	2.4	0.9606	38.4
50, 5, 0.4, 0.1	0.1008	6	3.47e-4	8.2
50, 8, 0.25, 0.15	0.1207	2.59	0.0611	18.3
55, 5, 0.25, 0.05	0.0169	2	0.0085	23.8
60, 6, 0.35, 0.1	0.11e-2	4.7	0.04e-2	11.7
60, 6, 0.4, 0.15	0.34e-2	7.5	0.05e-2	16.5
70, 7, 0.25, 0.05	0.0141	3.4	0.0068	23.9
70, 7, 0.3, 0.1	0.0640	5.8	0.0120	15
75, 8, 0.45, 0.2	1.2611	44.5	1.76e-5	67.7
80, 8, 0.3, 0.1	0.07e-2	16.4	0.01e-2	21.8
80, 15, 0.35, 0.2	0.7063	22.3	0.0069	49.1
100, 10, 0.25, 0.1	0.0566	12.2	0.0212	32.9
100, 10, 0.35, 0.15	0.3514	54.7	1.04e-5	82.7
120, 12, 0.3, 0.1	0.1058	77.8	3.22e-4	114.5

Table 7.3: Average accuracy (acc.) and computation time (c.t.) of whole procedure compared to literature.

Summary and Outlook

In this work a new SDP relaxation for the SNLP was presented which provides a good initial starting point for solution refinement techniques.
Its theoretical strength for uniquely localizable problems was shown. The beauty of the semidefinite problem formulation of the SNLP lies in its "convexification" of a non-convex problem. However, this ansatz still works only in theory.

For practical aspects, a descent heuristic was proposed which despite using second order information exploits sparsity in the problem formulation. Hence, it is competitive with known first order methods when weighting accuracy to computational time.

We restricted our numerical tests to smaller problems (~ 100 sensors) as we assume that larger networks can be subdivided into smaller ones. Our main interest was focused on finding more accurate solutions and not on handling large-scale SDPs.

Further investigation could be performed in order to find improvements to the proposed descent heuristics by applying different line search techniques.

Although the presented Decreasing Curvature Descent did not achieve best results for the SNLP we believe that this third order approach is completely new in the field of unconstrained non-convex optimization. Therefore, it could be a good starting point for further research and it might also be possible to adapt this approach to other classes of problems.

Appendix A

The SNLP is a NP-hard problem

The SNLP can be reduced from the partition problem, as sketched out by Tseng in [28].

The Partition problem is known as:

Given positive integers $t_1, t_2, .., t_p$ and $P = \{1, ..., p\}$, the question is if there exists a partition $I \subset P$ with

$$\sum_{i \in I} t_i = \sum_{j \in P \setminus I} t_j.$$

This problem is NP-complete (see for example [12]).

For dimension $d = 1$ the partition problem can be reduced to a SNLP with the following assumptions.
Let exactly one anchor point $x_{(0)} = 0$ be given, further let there be "cyclic" distance information given, i.e.

$$\begin{aligned} d_i &= \mid x_{(i-1)} - x_{(i)} \mid \quad \text{for} \quad i = 1, ..., n \quad \text{and} \\ d_{n+1} &= \mid x_{(n)} - x_{(0)} \mid = \mid x_{(n)} \mid . \end{aligned} \tag{A.1}$$

At least one unknown sensor is positive and at least one unknown sensor is negative (else the problem would be trivial).

The partition problem of the set $d_1, ..., d_{n+1}$ results in finding two disjunct sets I_1 and I_2 such that

$$\sum_{i \in I_1} d_i - \sum_{j \in I_2} d_j = 0. \tag{A.2}$$

The partitions I_1 and I_2 can then be interpreted as follows:

$$\begin{aligned} i \in I_1 &\Longleftrightarrow x_{(i-1)} - x_{(i)} > 0 \quad \text{and} \\ i \in I_2 &\Longleftrightarrow x_{(i-1)} - x_{(i)} < 0 \end{aligned} \tag{A.3}$$

and specially $n+1 \in I_1 \Longleftrightarrow x_{(n)} > 0$ or vice versa.

It can be shown that: *It is possible to find a solution $x_{(1)}, ..., x_{(n)}$ of the SNLP given by* (A.1) *if and only if there exists a partition problem of $d_1, ..., d_{n+1}$.*

Proof:
Let $x_{(1)}, ..., x_{(n)}$ be a solution to the SNLP problem (A.1) with $x_n > 0$, then it holds

$$\begin{aligned} &\sum_{i:x_{(i-1)}>x_{(i)}} \mid x_{(i-1)} - x_{(i)} \mid - \sum_{i:x_{(i-1)}<x_{(i)}} \mid x_{(i-1)} - x_{(i)} \mid + \mid x_{(n)} - x_{(0)} \mid \\ &= \sum_{i:x_{(i-1)}>x_{(i)}} x_{(i-1)} - x_{(i)} + \sum_{i:x_{(i-1)}<x_{(i)}} x_{(i)} - x_{(i-1)} + x_{(n)} \\ &= 0. \end{aligned}$$

If the indices $1, ..., n+1$ are the divided in the subsets I_1 and I_2 (cp. (A.3)), it is found a partition of the positive numbers d_i such that

$$\sum_{i \in I_1} d_i - \sum_{j \in I_2} d_j = 0.$$

Otherwise, if there exists a partition I_1 and I_2 holding (A.2), then a solution to (A.1) can be constructed as follows:

Without loss of generality, let $n+1 \in I_1$. Start by setting

$$x_n = d_{n+1}.$$

If next the index $n \in I_1$, set

$$x_{n-1} = x_n + d_n$$

(else set $x_{n-1} = x_n - d_n$) and so on. □

Notations

In the following are listed and explained all the symbols and abbreviations used in this thesis.

$\mathbb{R}$	Set of real numbers.
$\mathbb{R}^+$	Set of real numbers ≥ 0
$\mathbb{R}^d$	Set of all real vectors of dimension d.
x_i	i-th entry of the vector x.
$x_{(i)}$	Vector name (in most cases $x_{(i)} \in \mathbb{R}^d$, $d = 2$).
$A_{ij},\ A_{i,j}$	Entry of the matrix A corresponding to the i-th row and the j-th columnn.
$Z_{1:d,1:d}$	Matrix containing the first d rows and columns of the matrix Z.
$\mathcal{S}_+^n$	space of symmetric positive definite matrices of dimension $n \times n$.
$A \succeq 0$	A is positive semidefinite and symmetric, $A \in \mathcal{S}_+^n$.
$diag(Z)$	The vector containing the diagonal elements of the matrix Z.
$Diag(z)$	The matrix containing at the diagonal the elements of the vector z, elsewhere 0.
$\|\|\|_2$	Euclidean norm.
$\|\|\|_F$	Frobenius Norm.
$tr(A\ B)$	Trace between matrix A and B, $tr(A^T B) = \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij} B_{ij}$.
$A \circ B$	Hadamard Product of two matrices, $A \circ B = A_{ij} B_{ij}$).
$A \bullet B$	Inner product of two matrices of same dimension, $A \bullet B = tr(A^T B) = \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij} B_{ij}$.
$(a; b)$	The vector containing the vectors a and b listed b below a.
e	Vector of all ones.
$\mathbf{0}$	Vector or matrix containing only zero entries.

acc.	accuracy.
cp.	compare.
c.t.	computation time.
e.g.	*exempli gratia.*
et al.	*et alii.*
i.e.	*id est*
sc.	*scilicet.*
s.t.	subject to.

Bibliography

[1] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal of Optimization*, 5:13-51, 1995.

[2] R. Bhatia. *Matrix Analysis*, Springer, New York, 1996.

[3] S. J. Benson, Y. Ye and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal of Optimization*, 10(2):443-461, 2000.

[4] P. Biswas and Y.Ye. A distributed method for solving semidefinite programming arising from ad hoc wireless sensor network localization. Technical report, Deptartment of Management Science and Engineering, Stanford University, 2004.

[5] P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. *Proceedings 3rd IPSN*: 4-54, 2004.

[6] Y. Ding, N. Krislock, J. Qian and H. Wolkowicz. Sensor Network Localization, Euclidean Distance Matrix Completions, and Graph Realization.

[7] L. Doherty, L. El Ghaoui and K. S. J. Pister Convex position estimation in wireless sensor networks. In *Proceedings 20th INFOCOM, IEEE Computer Society*, 2673-2684, Los Alamitos, USA, 2004.

[8] R. Fletcher . *Practical Methods of Optimization*, Wiley & Sons, Chichester, 1987.

[9] C. Helmberg. Semidefinite Programming. *http://www-user.tu-chemnitz.de/ helmberg/semidef.html*.

[10] F. Jarre and A. López. Descent Heuristics for Unconstrained Minimization. Available at *http://www.optimization-online.org/DB_HTML/2008/09/2079.html*.

[11] F. Jarre and J. Stoer. *Optimierung*, Springer, Berlin, 2004.

[12] R.M. Karp. Reducibility Among Combinatorial Problems. In R.E. Miller and J.W. Thatcher (editors). *Complexity of Computer Computations*: 85-103. Plenum, New York, 1972.

[13] S. Krunk, M. Muramatsu, F. Rendl, R. J. Vanderbei and H. Wolkowicz. The Gauss-Newton direction in semidefinite programming. *Optimization Methods and Software*. 15(1):1-28, 2001.

[14] A. S. Lewis. Derivatives of Spectral Functions. *Mathematics of Operations Research*, 21, No.3: 576-588 , 1996.

[15] T. C. Liang, T. C. Wang and Y. Ye. A gradient search method to round the semidefinite programming relaxation solution for ad hoc wireless sensor network locations. Technical Report, Deptartment of Management Science and Engineering, Stanford University, 2004.

[16] Z-Q. Luo, S. Zhang, N.D. Sidiropoulos and P. Tseng. Approximation bounds for quadratic optimization with homogeneous quadratic constraints. *SIAM Journal of Optimization*, 2006.

[17] Mizuno, Todd and Y. Ye. A $O(\sqrt(n)L)$-iteration homogeneous and self-dual linear programming algorithm. *Mathematics of Operations Research*, 19:53-67, 1994.

[18] H. D. Mittelmann. Several SDP-codes on problems for SDPLIB. *http://plato.la.asu.edu/ftp/sdplib.html*, 2008.

[19] Y. E. Nesterov and A. S. Nemirovski. *Interior Point Polynomial Algorithms in Convex Programming*. SIAM Publications, SIAM, Philadelphia, USA, 1994.

[20] J. Nocedal and S. J. Wright. *Numerical Optimization*, Springer Series in Operation Research, New York, 1999.

[21] G. Pataki. The Geometry of Semidefinite Programming. In H. Wolkowicz, R. Saigal and L. Vandenberghe. *Handbook of Semidefinite Programming*:29-65, Kluwer Academic Publishers, 2000.

[22] R. Saigal, L. Vanderberghe and H. Wolkowicz. *Handbook of Semidefinite Programming-Theory, Algorithms and Applications*, Kluwer Academic Publishers, 2000.

[23] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods Software*, 11& 12: 625-653, 1999.

[24] D. Sun and J. Sun. Semismooth matrix-valued functions. *Mathematics of Operations Research*, 27: 150-169, 2002.

[25] A. M.-C. So and Y. Ye. Theory of semidefinite programming for sensor network localization. *Mathematical Programming*, 109, No. 2 - 3 (B): 367-384, 2007.

[26] A. So, Y. Ye and J. Zhang. A unified theorem on SDP rank reduction. Technical Report, Operation Research, Stanford University, Stanford, USA, 2006.

[27] P. Tseng. Second-order cone programming relaxation of sensor network localization. *SIAM Jorunal of Optimization*, 18:156-185, 2007.

[28] P. Tseng. SOCP relaxation of sensor network localization, Talk at the Univ. of Vienna, available at http://www.math.washington.edu/ tseng/personal.html.

Aknowledgements

There are many people that deserve a special nomination for having contributed to the success of this thesis, to whom I want to thank in their respective language.

Mein allergrößter Dank geht an Marcus, mein ständiger Begleiter, dessen Geduld und Ausgeglichenheit mich immer wieder inspirieren.

Meinen Eltern danke ich für die bedingungslose Unterstützung die ich seit meiner Geburt erfahren durfte. Muchas gracias por la interminable confianza y el exceso de amor.

Sabine danke ich für die hingebungsvolle Rolle als Patentante, die beste Freundschaft, Dein immer offenes Ohr.

Wiebke, danke für das Korrekturlesen. Es ist schön wieder gemeinsam ein gleiches Ziel zu verfolgen.

Ganz besonders will ich unseren Freunden Nadine und Bö danken. Die Aufenthalte in der Mensa bleiben unvergesslich. Für die ständigen Aufmerksamkeiten in Bezug auf Lina, vor allem den Kontakt zu Familie Siek bin ich Euch ewig dankbar.

Besonders hervorzuheben bleibt die herzliche Betreuung unserer Tochter Lina Amalia durch viele der genannten. Inbesondere die liebevolle Betreuung durch Hilde und Gerd hat viel zur Durchführbarkeit dieser Arbeit geholfen.

Danke.

Die hier vorgelegte Dissertation habe ich eigenständig und ohne unerlaubte Hilfe angefertigt. Die Dissertation wurde in der vorgelegten oder in ähnlicher Form noch bei keiner anderen Institution eingereicht. Ich habe bisher keine erfolglosen Promotionsversuche unternommen.

I do hereby declare, that the material contained in this dissertation is an original work perfomed by myself without illegitimate help. The material in this thesis has not been previously submitted for a PhD degree at any university.

Düsseldorf, den 12. Dezember 2008

Ania López