Interior Point Methods applied to Quadratic Programming*

Daniel Bergmann [1]

***Abstract*— Describe in a few sentences what the paper is about and why it is interesting to read it.**

I. INTRODUCTION

TODO!

Notation. Throughout this paper, we denote the Jacobian of a function $f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$, differentiated along direction v, evaluated at point $x \in \mathbb{R}^n$ by $J_v f(x)$. The index v is omitted, if it is contextually clear. For a collection of scalar valued functions $h_i : \mathbb{R}^n \longrightarrow \mathbb{R}, i = 1, \dots, m$ with the same domain, we define

$$h(x) := \begin{pmatrix} h_1(x) \\ \vdots \\ h_m(x) \end{pmatrix}$$

for the stacked values of all h_i evaluated at $x \in \mathbb{R}^n$.

II. PROBLEM STATEMENT AND BACKGROUND

For theoretical discussions, we consider the convex constrained optimization problem

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 0,\ i = 1, \dots, m. \\ & A_{\text{eq}} x = b_{\text{eq}}. \end{aligned} \tag{1}$$

with $f_0 : \mathbb{R}^n \longrightarrow \mathbb{R}$ convex and twice differentiable, $f_i : \mathbb{R}^n \longrightarrow \mathbb{R}$ for $i = 1, \dots, m$ convex and differentiable, $A_{\text{eq}} \in \mathbb{R}^{n \times p}, b_{\text{eq}} \in \mathbb{R}^p$ with equality and inequality constraints. For such an optimization problem, we call its Lagrangian $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \longrightarrow \mathbb{R}$ with

$$L(x, \lambda, \nu) = f_0(x) + \lambda^T f(x) + \nu^T (A_{\text{eq}} x - b_{\text{eq}}).$$

Further, we denote its dual problem by

$$\begin{aligned} \underset{(\lambda,\nu)}{\text{maximize}} \quad & g(\lambda, \nu) \\ \text{subject to} \quad & \lambda \geq 0, \nu \in \mathbb{R}^p \end{aligned} \tag{2}$$

with

$$g(\lambda, \nu) = \inf_{x \in \mathbb{R}^n} L(x, \lambda, \nu).$$

Moreover, we give a MATLAB-implementation of a primal-dual interiorpoint method for convex quadratic optimization problems. Quadratic problems are a subclass of (1) and denote as

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) = \frac{1}{2} x^T Q x + c^T x \\ \text{subject to} \quad & A_{\leq} x - b_{\leq} \leq 0,\ A_{\leq} \in \mathbb{R}^{m \times n}, b_{\leq} \in \mathbb{R}^m \\ & A_{\text{eq}} x - b_{\text{eq}} = 0,\ A_{\leq} \in \mathbb{R}^{p \times n}, b_{\leq} \in \mathbb{R}^p \end{aligned} \tag{3}$$

with matrices $0 \prec Q \in \mathbb{R}^{n \times n}, c \in \mathbb{R}^n$.

III. MAIN RESULTS

TODO!make $\mathbb{R}$ prettier

A. Newton's Method

Newton's method is an iterative process to solve nonlinear equation systems

$$F(x) = 0 \tag{4}$$

for a differentiable map $F : \mathbb{R}^n \longrightarrow \mathbb{R}^m$. The algorithm works as follows: At a given point x_k, the zero of the linear approximation of F around x_k is computed. This point is chosen as the next iterate x_{k+1}. In particular, a linear approximation of F in x_k is defined as

$$L(x) := F(x_k) + JF(x_k)(x - x_k) \text{ for } x \in \mathbb{R}^n, \tag{5}$$

where $JF(x_k)$ is the Jacobian of F at the point x_k. If $JF(x_k)$ is invertible, the point $\tilde{x}$ with $L(\tilde{x}) = 0$ is exactly the solution of the linear equation $JF(x_k)x = -F(x)$. Technical conditions and proofs about convergence rates of Newton's method can be found in [1]. The procedure executing a Newton search is summarized in (1).

Algorithm 1: Newton's Method

Result: $\tilde{x}$, approximate solution of nonlinear system of equalities $F(x) = 0$, residual tolerance $\epsilon_{res} > 0$, cauchy-tolerance $\epsilon_c > 0$
Data: Function $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$, initial point x_0

while $\|x - x_{\text{last}}\| \geq \epsilon_c$ *or* $\|F(x)\| \geq \epsilon_{res}$ **do**
  compute Newton direction Δx by solving $JF(x)\Delta x = -F(x)$;
  remember last interation for checking term. crit. $x_{\text{last}} = x$;
  update current point by $x = x + \Delta x$;
end
return $\tilde{x} = x$;

Remark 1: The residual and the cauchy-criterion for termination should be combined for the newton method. Easy examples are known, where one of the criteria is satisfied

*Project within the course Convex Optimization, University of Stuttgart, July 14, 2020.

[1]Daniel Bergmann is a student of the Bachelor study program Mechatronics, University of Stuttgart, st108500@stud.uni-stuttgart.de

even though the current iteration is far from the optimal point. For details, see [1]. For theoretical reasoning, or if $\nabla^2 f(x)^{-1}$ can be used explicitely, one can also use the decreasement of $\lambda^2 = \nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x)$ (so called newton-decrement) under a certain tolerance.

For the purpose of optimizing a convex, twice differentiable objective function f_0 we want to find a zero of the gradient ∇f_0. Therefore we can apply the Newton Method to solve the non-linear equation

$$F(x) := \begin{pmatrix} \nabla f_0(x) \\ g(x) \end{pmatrix} = 0 \qquad \text{with } g(x) = \begin{pmatrix} g_1(x) \\ \vdots \\ g_p(x) \end{pmatrix}$$

. By convexity, satisfying $\nabla f_0(x^*) = 0$ is not only neccessary, but also sufficient for x^* to be a global minimum of f_0.

B. Concept of Barrier Methods

Convex optimization problems with no inequality constraints can be solved efficiently by using Newton's method. If inequality constraints are involved, Newton's method can not guarantee feasibility of a found solution. Hence it is desirable, to transform an inequality-constrained optimization problem into one, that is only equality-constrained. Therefore, we move the inequality constraints implicitly to the objective function. A simple and precise way to do this, would be to evaluate an indicator function

$$I_-(x) := \begin{cases} 0 & \text{for } u \neq 0 \\ \infty & \text{for } u > 0 \end{cases} \tag{6}$$

on the values of the inequality constraints $f_i, i = 1, \dots, m$. We obtain a problem in the following shape

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) + \sum_{i=1}^{m} I_-(f_i(x)) \\ \text{subject to} \quad & A_{\text{eq}} x - b_{\text{eq}} = 0. \end{aligned} \tag{7}$$

This problem is equivalent to (1), since it yields an objective value of $+\infty$ for every infeasible point while it is the same problem for every feasible one. We now have a formulation without inequality constraints. However, it is clearly neither convex nor continuous and hence not differentiable. Since we need these properties to solve the optimization problem computationally, we approximate the indicator function I_- by the function

$$\hat{I}_-(u) = \begin{cases} \frac{1}{t}\log(-u) & \text{for } u < 0, \\ \infty & \text{for } u \geq 0, \end{cases} \tag{8}$$

The parameter $t > 0$ sets the approximation's accuracy. A higher value for t results in a better approximation of the indicator function. By replacing the indicator function by $\hat{I}_-$, we obtain an approximation

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) - \sum_{i=1}^{m} \frac{1}{t} \log(-f_i(x)) \\ \text{subject to} \quad & A_{\text{eq}} x - b_{\text{eq}} = 0 \end{aligned} \tag{9}$$

of problem (1). Throughout this paper, we denote its Lagrangian by $L_t : \mathbb{R}^n \times \mathbb{R}^p \longrightarrow \mathbb{R}$.

Note that $\frac{1}{t}\log(-u)$ is convex, increasing in u, and differentiable on the feasible set. Hence the entire function $\sum_{i=1}^{m} \hat{I}_-(f_i(x))$ is convex and (9) is a convex Problem with differentiable objective function. These properties allow us to solve (9) computationally. We call an optimal point $x^*(t)$ of (9) with parameter t a central point and a solution to its dual problem $(\lambda^*(t), \nu^*(t))$ a dual central point. The set of (dual) solutions of (9) for all $t > 0$ we call the (dual) central path. Since for points x with $f_i(x) = 0$ for any $x \in \{1, \dots, m\}$, the objective of (9) is ∞, all central points are in the interior of the set, satisfying the inequality constraints of (1). Thus this framework is named interior point method. One can show, that solutions $(x^*(t), \lambda^*(t), \nu^*(t))$ of (9) converge to the solution (x^*, λ^*, ν^*) of (1) for $t \longrightarrow 0$. The proof can be found in [2].

C. Measure for the Approximation's quality

An immediately arising question is which conclusions about the solution (x^*, λ^*, ν^*) of (1) can be drawn from knowing a solution of (9) for a certain $t > 0$ about the value $f_0(x^*(t))$ of a central point $x^*(t)$, compared with the optimal value p^* of the original problem. For compactness, we denote the barrier term of the approximated problem as

$$\phi(x) = -\sum_{i=1}^{m} \log(-f_i(x)),$$

with its Jacobian and Hessian being

$$\nabla \phi(x) = \sum_{i=1}^{m} \frac{1}{-f_i(x)} \nabla f_i(x),$$

$$\nabla^2 \phi(x) = \sum_{i=1}^{m} \frac{1}{f_i(x)^2} \nabla f_i(x) \nabla f_i(x)^T + \sum_{i=1}^{m} \frac{1}{-f_i(x)} \nabla^2 f_i(x).$$

For the sake of simplifying notation, throughout this section we consider the problem

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & t f_0(x) + \phi(x) \\ \text{subject to} \quad & A_{\text{eq}} x = b_{\text{eq}}. \end{aligned} \tag{10}$$

that is obtained by multiplying the objective in (9) with $t > 0$. The original and the obtained problem are equivalent. Any arbitrary $x^*(t)$ a strictly feasible point of (1). Since $x^*(t)$ solves (10), there exists $\hat{\nu} \in \mathbb{R}^p$, such that
TODO!check consistency

$$\nabla L_t(x^*(t), \hat{\nu}) = t \nabla f_0(x^*(t)) + \nabla \phi(x^*(t)) + A_{\text{eq}}^T \hat{\nu} \tag{11}$$

$$= t \nabla f_0(x^*(t)) + \sum_{i=1}^{m} \frac{1}{-f_i(x^*(t))} \nabla f_i(x^*(t)) + A_{\text{eq}}^T \hat{\nu}. \tag{12}$$

$$= 0 \tag{13}$$

holds. Note that the Lagrangian only depends on $(x, \hat{\nu})$, since there are no explicit inequality constraints involved. We keep in mind, that $x^*(t)$ minimizes (9). Using this insight, we know that there exists a dual feasible point $(x^*(t), \lambda^*(t), \nu^*(t))$ of the original problem (1). In particular, we choose

$$\lambda^*(t) = -\frac{1}{tf_i(x^*(t))} \text{ for } i = 1, \dots, m, \quad \nu^*(t) = \frac{\hat{\nu}}{t}.$$

Here, $\lambda^*(t) > 0$ follows from $f_i(x^*) < 0$ for all $i = 1, \dots, m$ since x^* is strictly feasible.

Note that (13) is the derivative of the Lagrangian

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i^*(t) f_i(x) + \nu^*(t)^T (A_{\text{eq}} x^*(t) - b_{\text{eq}})$$

dividied by $t > 0$ of the original problem. The Lagrangian is convex in the first coordinate, hence we infer that $x^*(t)$ minimizes the Lagrangian of the original problem for any fixed (λ, ν). For the dual function of the original problem, we obtain

$$\begin{aligned} g(\lambda^*(t), \nu^*(t)) &= f_0(x^*(t)) + \sum_{i=1}^{m} \lambda_i^*(t) f_i(x^*(t))) \\ &\quad + \nu^*(t)^T (A_{\text{eq}} x^*(t) - b_{\text{eq}}) \\ &= f_0(x^*(t)) - \frac{m}{t}. \end{aligned} \tag{14}$$

The second of the three summands adds up to $m \cdot 1$, because of the particular choice of $\lambda^*(t)$, fractions cancel out. The last summand equals zero, since $A_{\text{eq}} x^*(t) - b_{\text{eq}} = 0$.

By weak duality, this means that the optimum $x^*(t)$ of the approximated problem (9) has an objective value $f_0(x^*(t))$ that is maximally larger by $\frac{m}{t}$ (and hence worse) than the real optimal value p^* of the original problem. Thus, one can theoretically force a desired bound on the subobtimality $\epsilon > 0$ by choosing t large enough, in particular $t := \frac{m}{\epsilon}$. However, just solving (9) with a large choice of t does not work out in general, since numerical issues can make convergence of Newton's Method dependent on the choice of the initial point x_0.

D. Algorithmic Use of the Barrier Concept

As already mentioned in section III-C, one can not solve (9) generally without a good guess of the initial value x_0. So how to make use of the barrier concept? The idea of interior methods is, to find points along the problem's central path. Two methods are introduced in the following. Emphasis of the explanations as well as the implementation in MATLAB will be on the Primal-Dual Interior Point Method.

1) Barrier Method: As mentioned before, for large t a good initial point x_0, meaning an initial point that is not far away from the actual minimum of (1), is crucial for avoiding large numerical errors. This can be achieved by starting with optimization of (9) for small $t = t_1$, which leads to a rather bad approximation of the original problem, but also to better numerical behavior. After finding $x^*(t_1)$ via Newton's method, t is increased to $t = t_2 > t_1$ by a certain rate and (9) is solved again with parameter $t = t_2$, with choice $x_0 = x^*(t_1)$ for the initial point.

We call finding the minimum $x^*(t)$ of (9) the centering step or outter iteration, while we call a single Newton step inside the centering step an inner iteration.

Algorithm 2: Barrier Method with full Newton search

Result: $x^*(t)$, approximate solution of (1) with $f_0(x^*(t)) - p^* < \frac{m}{t}$
initialization: Matrices $0 \prec Q \in \mathbb{R}^{n \times n}, c \in \mathbb{R}^n$. defining the objective function, matrices $A_{\text{eq}} \in \mathbb{R}^{m \times n} b_{\text{eq}} \in \mathbb{R}^p, A_{\leq} \mathbb{R}^{m \times n}, b_{\leq} \in \mathbb{R}^m$ defining constraints, initial point x, initial approximation parameter $t > 0$, rate for increasing appprox. param. $\mu > 1$ tolerance ϵ;
while $\frac{m}{t} \geq \epsilon$ **do**
  Compute $x^*(t)$ by solving (9) via Newton's Method, starting at x;
  Update $x := x^*(t)$;
  Increase t by $t := \mu t$
end

2) Primal-Dual Interior Point Method: Like the previously introduced algorithm, the Primal-Dual Interior Point method uses the barrier concept to handle inequality constraints. It is motivated by the following idea: Since the points generated by each outer iteration converge to the desired optimum on the central path, one does not gain much advantage by computing the central points with a high level of accuracy. So many newton-steps are computed, without improving the convergence towards the optimum value of (1). Hence, it would be useful to reduce the accuracy of each outer iteration as much as possible, without losing convergence to the optimum. Therefore, in this method only one newton step will be computed for each parameter t in the approximated problem (9). Also, the Newton step is computed differently. While in the barrier method with full newton search, the search directions are computed only considering the primal problem, in the Primal-Dual Methodwe also take the dual problem of (9) into account. In particular Newton's method is applied to a system of residual terms, that have to equal all zero by the KKT-conditons, here presented like in [2].

Theorem 1 (KKT-Conditions for convex Problems): For a convex Optimization Problem (1), the following conditions on a primal-dual point $(x, \lambda, \nu) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p$ are neccessary and sufficient for x being a solution to the primal problem and (λ, ν) being a solution to the dual problem:

$$f_i(x) \leq 0, \quad \text{for } i = 1, \dots, m \tag{15a}$$

$$A_{\text{eq}} x - b_{\text{eq}} = 0 \tag{15b}$$

$$\lambda_i \geq 0, \quad \text{for } i = 1, \dots, m \tag{15c}$$

$$\lambda_i f_i(x) = 0, \quad \text{for } i = 0, \dots, m \tag{15d}$$

$$\nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + \sum_{i=1}^{p} \nu_i \nabla h_i(x) = 0. \tag{15e}$$

Stacked in one vector, this yields the system of equalities

$$\begin{aligned} r_\mu(x,\lambda,\nu) &= \begin{pmatrix} r_{\text{dual}} \\ r_{\text{cent}} \\ r_{\text{pri}} \end{pmatrix} \\ &= \begin{pmatrix} \nabla f_0(x) + Jf(x)^T\lambda + A_{\text{eq}}^T\nu \\ -\text{diag}(\lambda)f(x) - \mu\mathbb{1} \\ A_{\text{eq}}x - b_{\text{eq}} \end{pmatrix} \stackrel{!}{=} 0. \end{aligned} \tag{16}$$

to apply Newton on. For formulation of the linear Newton equality, we also compute the jacobian

$$J_{(x,\lambda,\nu)} r_\mu(x,\lambda,\nu) \tag{17}$$

$$= \underbrace{\begin{pmatrix} \nabla^2 f_0(x) + \sum_{i=1}^m \lambda_i \nabla^2 f_i(x) & Jf(x) & A_{\text{eq}}^T \\ -\text{diag}(\lambda)Jf(x) & -\text{diag}(f(x)) & 0 \\ A_{\text{eq}} & 0 & 0 \end{pmatrix}}_{:=M_{\text{KKT}}} \tag{18}$$

of the residual and refer to it as M_{KKT}. Consequently, the Newton equality for finding the search direction $(\Delta x, \Delta\lambda, \Delta\nu)$ in each newton step is obtained by solving the linear equation

$$M_{\text{KKT}} \begin{pmatrix} \Delta x \\ \Delta\lambda \\ \Delta\nu \end{pmatrix} = b_{\text{KKT}} \tag{19}$$

with $b_{\text{KKT}} = -r_\mu(x,\lambda,\nu)$.

Unfortunately, adding the obtained step direction $(\Delta x, \Delta\lambda, \Delta\nu)$ to (x,λ,ν), does not in general yield a feasible point. Therefore we compute a suitable step-size s^* via a backtracking-linesearch, such that a certain decrease of the residual and feasibility is guaranteed for the next iteration point

$$\begin{pmatrix} x^+ \\ \lambda^+ \\ \nu^+ \end{pmatrix} = \begin{pmatrix} x \\ \lambda \\ \nu \end{pmatrix} + s^* \begin{pmatrix} \Delta x \\ \Delta\lambda \\ \Delta\nu \end{pmatrix}.$$

The detailed procedure of the backtracking linesearch is displayed in Algorithm 3.

Finally, we can present the entire algortithm of the Primal-Dual Method.

Remark 2: If a strictly feasible primal variable $x \in \mathbb{R}^n$ is known, $\lambda = -1/f_i(x) \geq 0, \nu = 0$ is always a valid choice for the initial dual variables.

E. How to find a feasible inital point

The Algorithms 2 and 4 both need a strictly feasible initial point to start. Since such a point is in general not trivial to find, one can formulate the search for the initial point as another convex optimization problem, that is easier to solve than the original one. For problem (1) one way to implement this, is solving

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & s \\ \text{subject to} \quad & f_i(x) \leq s, \quad i = 1,\ldots,m \\ & A_{\text{eq}}x - b_{\text{eq}} = 0, \end{aligned} \tag{20}$$

Algorithm 3: Backtracking linesearch

Result: Stepsize s^*, s.t. $\lambda^+ > 0, f(x^+) < 0$ and r_μ decreases by certain amount.
Data: Problem matrices, current x,λ,ν, Newton direction $\Delta x, \Delta\lambda, \Delta\nu$, barrier parameter μ, backtracking parameters $\alpha \geq 0, \beta \in (0,1)$. Initial step-size set $s_{\max} = \min\{1, \min_{i|\Delta\lambda_i<0} -\lambda_i/\Delta\lambda_i\}$

compute $r_\mu(x,\lambda,\nu)$;
$s = s_{\max}$;
$found = false$;
while $found == false$ **do**
  set $s = \beta s$;
  compute (x^+, λ^+, ν^+);
  compute $r_\mu(x^+,\lambda^+,\nu^+)$ and $f(x^+)$;
  if $f(x^+) < 0$ *and* $\|r_\mu(x^+,\lambda^+,\nu^+)\| \leq (1-\alpha s)\|r_\mu(x,\lambda,\nu)\|$ **then**
    $found = true$
  end
end

Algorithm 4: Primal-Dual Interior Point Method

Result: approximate optimizer $\hat{x}^*$, approx. opt. value $\hat{p}^*$, approx. dual optimizer $(\hat{\lambda}^*, \hat{\nu}^*)$, surrogate duality gap $\hat{\eta}^*$ as measure of optimality
Data: Problem matrices, primal-dual initial point (x,λ,ν) with $f_i(x) < 0$ for all $i = 1,\ldots,m$, $\lambda > 0, \nu \in \mathbb{R}^p$ (initial point strictly feasibile), reduction factor $\gamma \in (0,1)$, tolerances $\epsilon_{\text{feas}}, \epsilon_{\text{opt}} > 0$

Initialization;
determine problem dimensions n, m, p;
set $found = false$;
while $found == false$ **do**
  compute surrogate duality gap: $\hat{\eta} = -f(x)^T\lambda$;
  compute KKT residual vector $r_\mu(x,\lambda,\nu)$ via (16);
  compute search direction $(\Delta x, \Delta\lambda, \Delta\nu)$ by solving (19);
  determine suitable step size s via backtracking algorithm 3;
  update current primal and dual points: $(x,\lambda,\nu) = (x,\lambda,\nu) + (\Delta x, \Delta\lambda, \Delta\nu)$;
end
return $\hat{x}^* = x, \hat{p}^* = f_0(\hat{x}^*), \hat{\lambda}^* = \lambda, \hat{\nu}^* = \nu, \hat{\eta}^* = \hat{\eta}$;

via Newton's method. If a point with optimal value strictly smaller than zero for (20) is found, then this point is strictly feasible. Solving such a first, more simple problem is called a Phase I problem. More examples of such problems can be found in [2].

F. Complexity Analysis for the Barrier Method

Emphasis of this article is on implementation and idea of the algorithms, so we treat complexity analysis only by presenting results without proves. We keep this restricted to the barrier method with full newton search. We discuss the time complexity of the barrier method, meaning the total number of newton steps needed to solve (1). An upper bound of these iterations can be proven for problems with objectives that are self-concordant. While Linear and quadratic functions satisfy selfconcordance in general, any other convex optimization problem can be rewritten as a self-concordant one, so this condition is not very restrictive. The upper bound

$$\frac{f(x) - p^*}{\gamma} + c \tag{21}$$

on the maximal number of newton iterations that is needed to get a newton decrement (see remark 1) smaller than $\epsilon_{\rm nt}$, while c depends on $\epsilon_{\rm nt}$ by $\log_2 \log_2(1/\epsilon_{\rm nt})$, p^* is the primal problem's optimal value and γ depends on choice of the backtracking parameters α, β with

$$\frac{1}{\gamma} = \frac{20 - 8\alpha}{\alpha\beta(1 - 2\alpha)^2}.$$

The derivation of this bound is shown in [2], section 9.

One can show that this bound holds uniformly for any parameter t for all problems (9). Since there are exactly

$$\left\lceil \frac{\log(m/\epsilon t_0)}{\log \mu} \right\rceil$$

outer steps neccessary to solve (9) with inital parameter $t = t_0$ and tolerance ϵ, the entire barrier method needs maximally

$$N = \left\lceil \frac{\log(m/\epsilon t_0)}{\log \mu} \right\rceil \left(\frac{m(\mu - 1 - \log \mu)}{\gamma} + c \right)$$

inner newton iterations, where m denotes the number of inequality constraints on (9). iterations to yield a result with a suboptimality of ϵ or smaller. Detailed reasoning can be found in [2], section 11.5.

IV. EXAMPLES

TODO!Show and discuss simulation examples etc....

V. CONCLUSIONS

TODO!

Summarize the main points (with more details than in the preceding introduction). The paper should not be between 4 and 8 pages.

APPENDIX

Add for example your Matlab code here. (Code should be nicely formated and documented).

Appendixes should appear before the acknowledgment.

ACKNOWLEDGMENT

REFERENCES

[1] Carsten Scherer *Vorlesungsskript Einführung in die Optimierung* 2019: Lehrstuhl für Mathematische Systemtheorie, Universität Stuttgart.

[2] Stephen Boyd, Lieven Vandenberghe *Convex Optimization* 2004: Cambridge University Press.