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!Some general introducing sentences about the topic, motivation and relevance of problem/algorithm.

In this paper we give an introduction to the results presented in paper(s) [**?**].

We present the problem statement (optimization problem) the main results/algorithms, discuss the underlying ideas and illustrate the results by numerical simulations.

Notation. Define notation.

II. PROBLEM STATEMENT AND BACKGROUND

For theoretical discussions, we consider the convex constrained optimization problem

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

with $f_0 : \mathbb{R}^n \longrightarrow \mathbb{R}$ convex and twice differentiable, $f_i : \mathbb{R}^n \longrightarrow \mathbb{R}$ for $i = 1, \ldots, m$ convex and twice TODO!differentiable, $A_{\text{eq}} \in \mathbb{R}^{n \times p}, b_{\text{eq}} \in \mathbb{R}^p$ with equality and inequality constraints. Moreover, we give a MATLAB-implementation of a primal-dual interiorpoint method for a convex quadratic optimization problem. Quadratic problems are a subclass of (1) and denote as

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

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

III. MAIN RESULTS

TODO!define dual problem

A. 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 feasibiliy of a solution. It is hence desirable, to transform an inequality-constrained optimization problem into a only equality-constrained one. Therefore, we move the inequality constraints implicitley to the objective function. TODO!improve sentence TODO!A simple and also precise way to do this, evaluate an indicator function

$$I_-(x) := \begin{cases} 0 & \text{for } u \neq 0 \\ \infty & \text{for } u > 0 \end{cases} \tag{3}$$

on the values of the inequality constraints $f_i, i = 1, \ldots, m$. Then, the optimization Problem has the shape

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) + \sum_{i=1}^{m} I_-(f_i(x)) \\ \text{subject to} \quad & A_{\text{eq}} x - b_{\text{eq}} = 0,\ i = 1, \ldots, p. \end{aligned} \tag{4}$$

This problem is an equivalent to (1) and has no 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

$$\hat{I}_-(u) = \begin{cases} \frac{1}{t} \log(-u) & \text{for } u < 0, \\ \infty & \text{for } u \geq 0, \end{cases} \tag{5}$$

The parameter $t > 0$ sets the approximation's accuracy. The higher t is, the better the indicator function is approximated. By replacing the Indicator functions by $\hat{I}_-$, we obtain an approximation

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) - \sum_{i=1}^{m} \frac{1}{t} \log(-f_i(x)) \\ \text{subject to} \quad & A_{\text{eq}} x - b_{\text{eq}} = 0 \end{aligned} \tag{6}$$

of problem (1).

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 (6) is a convex Problem with differentiable objective function. These properties allow us to solve (6) computationally. We call an optimal point $x^*(t)$ of (6) 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 (6) for all $t > 0$ we call the (dual) central path. One can show, that solutions $(x^*(t), \lambda^*(t), \nu^*(t))$ of (6) converge to the solution (x^*, λ^*, ν^*) of (1) for $t \longrightarrow 0$. The proof is shown in [2].

B. Measure for the Approximation's quality

An immediately arising question is, what conclusions about the solution (x^*, λ^*, ν^*) of (1) can be drawn from a knowing a solution of (6) for a certain $t > 0$. By TODO!show arguments?

*Project within the course Convex Optimization, University of Stuttgart, July 8, 2020.

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

For compactness, we denote the barrier term of a the approximated problem as

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

and keep in mind, that its derivatives are

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

C. Algorithmic Use of the Barrier Concept

As already mentioned in section **??**, one can not in general solve (6) without a good guess at 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) Interior Point Method with Full Newton Search: 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 (6) for small $t = t_1$, which leads to a 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 (6) is solved again with parameter $t = t_2$, with choice $x_0 = x^*(t_1)$ for the initial point. For step n of the algorithm call finding $x^*(t)$ the centering, and updating and updating t and $x^*(t)$ an outer iteration or centering point and a iteration of the newton algorithm within the centering step an inner iteration. The whole procedure is written in Algorithm 1.

Algorithm 1: Interior Point 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 $A_{\text{eq}}, b_{\text{eq}}, A_{\leq}, b_{\leq}$ defining constraints, initial point x, initial approximation parameter $t > 0$, rate for increasing appprox. param. $\mu > 1$ tolerance ϵ

TODO!dimensions of constr. matrices

while $\frac{m}{t} \geq \epsilon$ **do**

  Compute $x^*(t)$ by solving (6) 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 centralpoints 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 (6). Additionally, the Newton step is computed differently. While in the the search directions are computing only considering the primal problem, in the Primal-Dual Methodwe also take the dual problem of

TODO!write dual problem

problem (6) 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:

$$f_i(x) \leq 0, \quad \text{for } i = 1, \dots, m \tag{7a}$$
$$A_{\text{eq}} x - b_{\text{eq}} = 0 \tag{7b}$$
$$\lambda_i \geq 0, \quad \text{for } i = 1, \dots, m \tag{7c}$$
$$\lambda_i f_i(x) = 0, \quad \text{for } i = 0, \dots, m \tag{7d}$$
$$\nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + \sum_{i=1}^{p} \nu_i \nabla h_i(x) = 0. \tag{7e}$$

Stacked in one vector, this yields

$$\begin{aligned} F(x, \lambda, \nu) :&= r_\mu(x, \lambda, \nu) = \begin{pmatrix} r_{\text{dual}} \\ r_{\text{cent}} \\ r_{\text{pri}} \end{pmatrix} \\ &= \begin{pmatrix} \nabla f_0(x) + Jf(x)^T \lambda + A_{\text{eq}}^T \nu \\ -\text{diag}(\lambda) f(x) - \mu \mathbb{1} \\ A_{\text{eq}} x - b_{\text{eq}} \end{pmatrix} = 0. \end{aligned} \tag{8}$$

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

$$\frac{\mathrm{d}\,(r_\mu(x, \lambda, \nu))}{\mathrm{d}(x, \lambda, \nu)^T} = \begin{pmatrix} \nabla^2 f_0(x) + \sum_{i=1}^m \lambda_i \nabla^2 f_i(x) & Jf(x) & A_{\text{eq}}^T \\ -\text{diag}(\lambda) Jf(x) & -\text{diag}(f(x)) & 0 \\ A_{\text{eq}} & 0 & 0 \end{pmatrix} \tag{9}$$

of the residual. Consequently, the Newton equality for finding the search direction $(\Delta x, \Delta \lambda, \Delta \nu)$ in each newton step is exactly

$$\begin{pmatrix} \nabla^2 f_0(x) + \sum_{i=1}^m \lambda_i \nabla^2 f_i(x) & Jf(x) & A_{\text{eq}}^T \\ -\operatorname{diag}(\lambda) Jf(x) & -\operatorname{diag}(f(x)) & 0 \\ A_{\text{eq}} & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \lambda \\ \Delta \nu \end{pmatrix} = - \begin{pmatrix} r_{\text{dual}} \\ r_{\text{cent}} \\ r_{\text{pri}} \end{pmatrix} \tag{10}$$

(TODO!equality should fit in col.!

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 2.

Algorithm 2: 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 $s_{\max} := \min\{1, \min_{i|\Delta\lambda_i < 0} -\lambda_i/\Delta\lambda_i\}$

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

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

Remark 1: 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.

TODO!explain notation of f without index

D. How to find a feasible inital point

The Algorithms 1 and 3 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

Algorithm 3: 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 (8);
  compute search direction $(\Delta x, \Delta\lambda, \Delta\nu)$ by solving (10);
  determine suitable step size s via backtracking algorithm 2;
  update current primal and dual points: $(x, \lambda, \nu) = (x, \lambda, \nu) + (\Delta x, \Delta\lambda, \Delta\nu)$;
end
return $\hat{x}^* = x, \hat{p}^* = f_0(\hat{x}^*), \hat{\lambda}^* = \lambda, \hat{\nu}^* = \nu, \hat{\eta}^* = \hat{\eta}$;

than the original one. For problem (1) one way to implement this, is solving

$$\begin{array}{ll} \underset{x}{\text{minimize}} & s \\ \text{subject to} & f_i(x) \leq s, \quad i = 1, \ldots, m \\ & A_{\text{eq}} x - b_{\text{eq}} = 0, \end{array} \tag{11}$$

via Newton's method. If a point with optimal value strictly smaller than zero for (11) 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].

E. Complexity Analysis Barrier Method

In this section 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

$$\frac{f(x) - p^*}{\gamma} + c \tag{12}$$

on the maximal number of newton iterations that is needed to obtain a certain accuracy, determined by the initial point x, the optimal value p^*, γ a constant dependent on the backtracking parameters and c dependent on the accuracy tolerance. One can show via reasoning with the Lagrangian, TODO!detailed?

that this bound holds uniformly for any parameter t for all problems (6). Since there are exactly

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

outer steps neccessary to solve (6) with inital parameter $t = t_0$ and accuracy tolerance ϵ, the entire barrier method needs maximally

$$N = \ldots$$

TODO!
iterations to yield a result with a suboptimality of ϵ or smaller.

F. Newton's Method

TODO!move to beginning Newton's method is an iterative process to solve nonlinear equality systems

$$F(x) = 0 \tag{13}$$

for a differentiable map $F : \mathbb{R}^n \longrightarrow \mathbb{R}^m$. The idea of this algorithm is 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

$$L(x) := F(x_k) + JF(x_k)(x - x_k) \text{ for } x \in \mathbb{R}^n, \tag{14}$$

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

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.

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.