The Steihaug Method

Vera Jackisch
verajackisch@t-online.de

Klara Burger
klara.burger@posteo.de

***Abstract*—In this project report we present the Steihaug method. This is a trust-region method that is particularly suitable for large scale optimization problems and is widely used in practice. The algorithm is based on the conjugate gradient method. We will give a short introduction to trust-region methods and the Steihaug method itself, before we will investigate the performance of the Steihaug method on several test problems.**

I. Introduction

As we know, the Newton-type method converges locally only if $||x_0 - x^*||$ is sufficiently small, whereby x_0 is the initial value and x^* the optimal solution. If $||x_0 - x^*||$ is too large, trust-region methods are one possibility to ensure global convergence. Trust-region algorithms choose a search direction and a step length in a bounded region around the current iterate, called the trust region. Here, we present a trust-region method that is widely used in practice: the Steihaug method.

II. The Steihaug Method

By $||\cdot||$ we denote the Euclidean norm.

A. The Trust-Region Method

In the following we consider the quadratic model function m_k defined by

$$m_k(x_k + p) = f(x_k) + \nabla f(x_k)^\top p + \frac{1}{2} p^\top B_k p,$$

where B_k is an approximation of the Hessian $\nabla^2 f(x_k)$.
The iteration is then given by

$$x_{k+1} = x_k + p_k$$

with

$$p_k = \arg\min_{p \in \mathbb{R}^n} m_k(x_k + p) \quad \text{s.t. } ||p|| \leq \Delta_k. \quad (1)$$

Equation (1) is called the trust-region subproblem and Δ_k is called the trust-region radius. Within this radius around the current iterate, we trust the model to be a sufficient representation of the objective function.
The trust-region radius Δ_k is an important quantitiy for the performance of the algorithm and is updated in each step. It is chosen depending on the trustworthiness of the model, which is given by the ratio of the actual and predicted reduction:

$$\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(x_k) - m_k(x_k + p_k)}.$$

The denominator is always non-negative since $m_k(x_k + p_k) \leq m_k(x_k)$ (as p_k is the solution of (1)). Therefore, if ρ_k is negative, we have $f(x_k) < f(x_k + p_k)$ and we reject this step. Is $\rho_k \approx 1$, we trust the model and increase Δ_k. For the pseudo code of the trust-region algorithm, see Algorithm 1 (taken from [2]).

B. The Steihaug Method

The Steihaug method is an iterative procedure, which improves the Cauchy Point inside the trust region. It is a conjugate gradient method with two extra stopping criteria:

1) Stop if $d_j^\top B_k d_j \leq 0$, with d_j the current search direction.
2) Stop if $||p_{j+1}|| > \Delta_k$.

In both cases, intersecting the current search direction with the trust-region boundary gives the final iterate p. See Algorithm 2 for the pseudo code of the Steihaug algorithm.

Setting $p_0 = 0$ is very important to the algorithm. First of all, with this initialization we get the Cauchy point as the first iterate:

$$p_1 = \alpha_0 d_0 = \frac{r_0^\top r_0}{d_0^\top B d_0} d_0 = -\frac{\nabla f(x_k)^\top \nabla f(x_k)}{\nabla f(x_k)^\top B_k \nabla f(x_k)} \nabla f(x_k),$$

Algorithm 1: Trust-region algorithm

input : $\Delta_{\max} > 0$, $\Delta_0 \in (0, \Delta_{\max})$, $\eta \in [0, \frac{1}{4}]$, x_0, TOL > 0
output: x^*

$k = 0$
while $||\nabla f(x_k)|| >$ TOL **do**
  Solve TR-subproblem to get p_k (e.g. with the Steihaug Method)
  Compute ρ_k
  Adapt Δ_{k+1}:
  if $\rho_k < \frac{1}{4}$ **then**
    $\Delta_{k+1} = \Delta_k \cdot \frac{1}{4}$
  else if $\rho_k > \frac{3}{4}$ and $||p_k|| = \Delta_k$ **then**
    $\Delta_{k+1} = \min(2 \cdot \Delta_k, \Delta_{\max})$
  else
    $\Delta_{k+1} = \Delta_k$
  Decide on acceptance of step:
  if $\rho_k > \eta$ **then**
    $x_{k+1} = x_k + p_k$
  else
    $x_{k+1} = x_k$
  $k = k + 1$
$x^* = x_k$

Algorithm 2: Steihaug algorithm

input : Given tolerance $\varepsilon_k > 0$, $\nabla f(x_k)$, B_k, Δ_k
output: p

Set $p_0 = 0$, $r_0 = \nabla f(x_k)$, $d_0 = -r_0$
if $||r_0|| < \varepsilon_k$ **then**
  return $p = p_0 = 0$
for $j = 0, 1, \ldots$ **do**
  if $d_j^\top B_k d_j \leq 0$ **then**
    find τ such that $p = p_j + \tau d_j$ minimizes $m_k(x_k + p)$ and satisifies $||p|| = \Delta_k$
    return p
  Set $\alpha_j = \frac{r_j^\top r_j}{d_j^\top B_k d_j}$
  Set $p_{j+1} = p_j + \alpha_j d_j$
  if $||p_{j+1}|| > \Delta_k$ **then**
    find $\tau \geq 0$ such that $p = p_j + \tau d_j$ satisfies $||p|| = \Delta_k$
    return p
  Set $r_{j+1} = r_j + \alpha_j B_k d_j$
  if $||r_{j+1}|| < \varepsilon_k$ **then**
    return $p = p_{j+1}$
  Set $\beta_{j+1} = \frac{r_{j+1}^\top r_{j+1}}{r_j^\top r_j}$
  Set $d_{j+1} = -r_{j+1} + \beta_{j+1} d_j$

Another implication is the following theorem (taken from [1]):

Theorem 1: The sequence of vectors $\{p_i\}$ generated by the Steihaug algorithm satisfies

$$0 = ||p_0|| < \ldots < ||p_j|| < \ldots < ||p|| \leq \Delta,$$

with p being the last iterate.

Proof: For proof check [1] Theorem 4.2. ■

Therefore, it is reasonable to stop iterating as soon as the trust-region boundary is reached, because any further iterate giving a lower value of m_k will be outside of the trust region.

The following convergence results concerning the Steihaug algorithm are known: In general, we have global convergence, since each iteration of the Steihaug method reduces $m_k(\cdot)$. Under some assumptions we can even prove a superlinear convergence rate. For this we have to assume

A.1. $x_k \to x^*$ as $k \to \infty$.
A.2. f is twice continuously differentiable in an open neighborhood Ω of x^*.
A.3. The Hessian matrix $H(x^*)$ of f is positive definite.
A.4. The sequences $\{p_k\}$ and $\{B_k\}$ satisfy

$$\lim_{k \to \infty} \frac{||B_k p_k + \nabla f(x_k) - \nabla f(x_k + p_k)||}{||p_k||} = 0.$$

Then the following theorem (taken from [3]) holds:

Theorem 2: If $\varepsilon_k \to \infty$, then $\{x_k\}$ converges superlinearly, i.e.

$$\lim_{k \to \infty} \frac{||x_{k+1} - x^*||}{||x_k - x^*||} = 0.$$

Proof: See [3] Theorem 4.2. ■

III. Experiments

This section discusses some numerical experiments we did with the Steihaug method using our own MATLAB implementation. We have considered

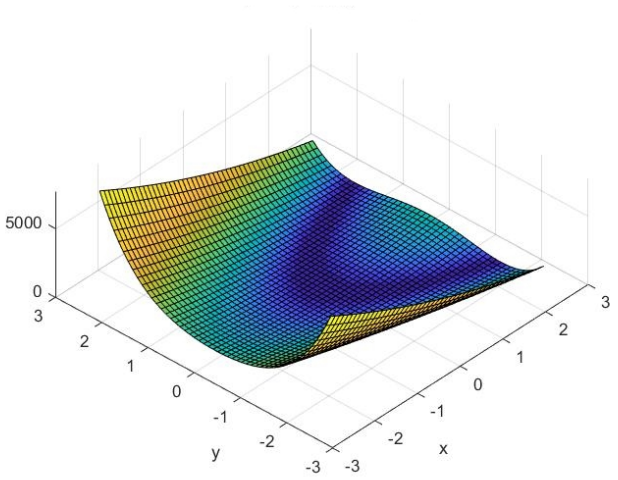

Fig. 1. Generalized Rosenbrock function in two dimensions.

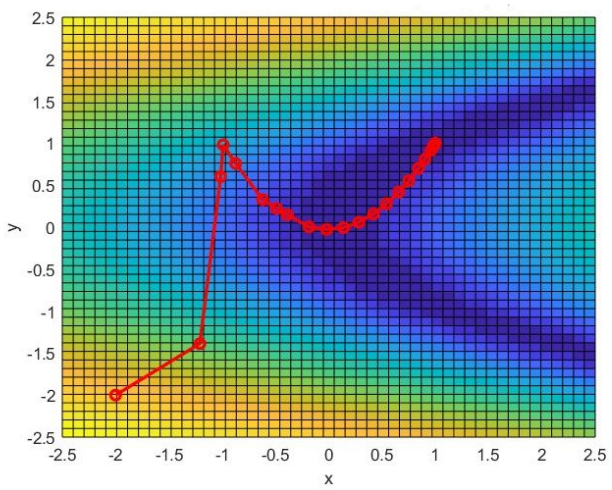

Fig. 2. Iterates of the Steihaug method while minimizing the generalized Rosenbrock function in two dimensions, starting at $x_0 = (-2, -2)$.

two different optimization problems and also compared the performance of the Steihaug method to the performance of CasADi IPOPT (see [5]) solving different problems.

A. The Generalized Rosenbrock Function:

First of all, we considered the generalized Rosenbrock function, which is a large-scale non-convex optimization problem:

$$\min_{x \in \mathbb{R}^n} \; 1 + \sum_{i=2}^{n} \left[100(x_i - x_{i-1}^2)^2 + (1 - x_{i-1})^2 \right]$$

where the optimum is reached at

$$x^* = (1, 1, 1, \ldots, 1)\,.$$

For the two-dimensional Rosenbrock function see Figure 1. This problem is a pathological problem, which is well-known for benchmarking optimization problems. It is easy to find the valley of the Rosenbrock function in which the minimum is located, but it is often much more difficult to finally converge to this minimum. This is also something we can observe when using the Steihaug method to solve this problem. In Figure 2 we have plotted the two-dimensional Rosenbrock function and the iterations of the trust-region method using Steihaug which are needed to converge to the minimum at $(1, 1)$.

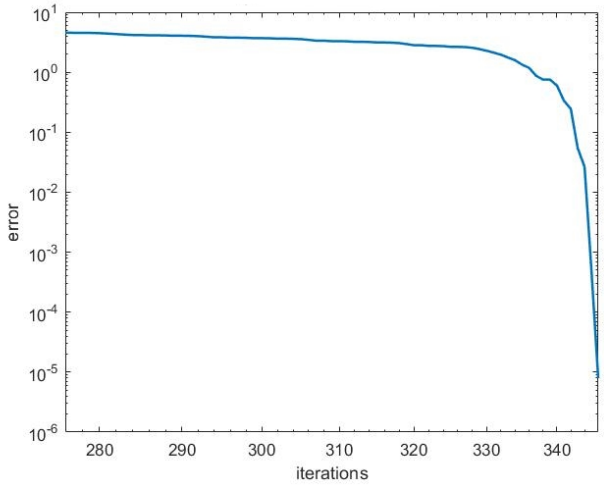

Fig. 3. Convergence rate of the error in the Euclidean norm for the Steihaug method applied to the generalized Rosenbrock function.

B. Convergence Rate

Theorem 2 states a superlinear convergence rate, which we also observed in our experiments. In Figure 3 the convergence rate of our Steihaug implementation solving the generalized Rosenbrock problem is shown. One can clearly see a superlinear convergence behaviour.

C. Behaviour of Relevant Quantities

When solving the 100-dimensional generalized Rosenbrock function, we examined the quantities

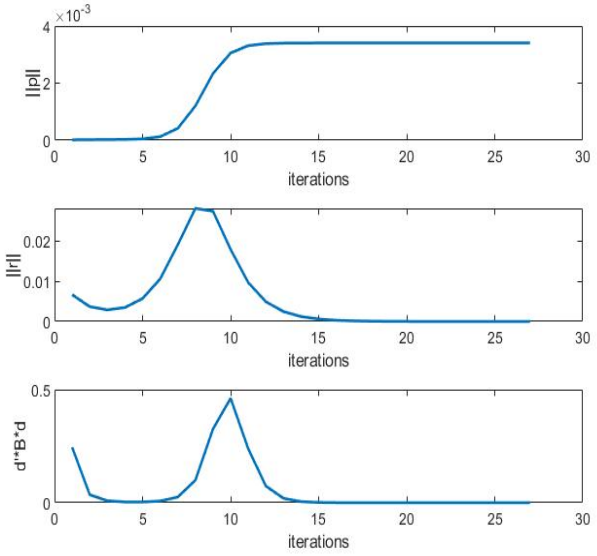

Fig. 4. Performance of Steihaug applied to the generalized Rosenbrock function.

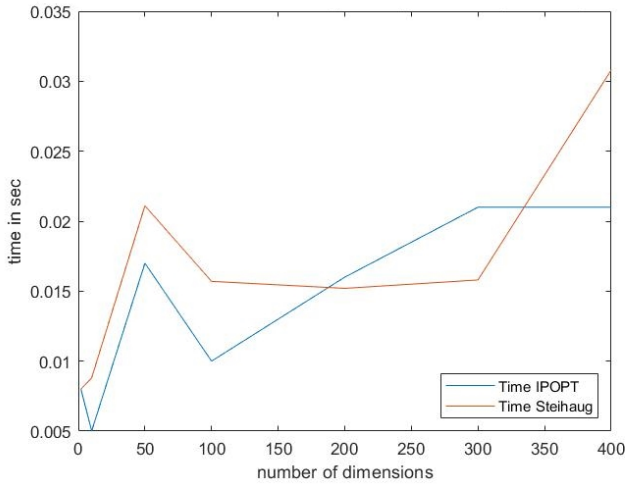

Fig. 5. A comparison of Steihaug and CasADi IPOPT solving the convex quadratic test problem.

$||p||$, the residual $||r||$ and the curvature $d^\top B_k d$. The results can be seen in Figure 4. As stated in Theorem 1, the norm of p is strictly increasing in a step of Steihaug. We also observed this in our experiments, as seen in the upper plot of Figure 4. The norm of the residual $||r||$ tends to zero as we iterate more, which is also expected. This is shown in the middle plot of Figure 4. The curvature $d^\top B_k d$ also tends to zero, which is also expected, as we examined the quantities in the last trust-region step. The change of the curvature of B_k in the direction d can be seen in the lower plot of Figure 4.

D. Second Test Problem

Another test problem we used to test the Steihaug method is the following:

$$\min_{x\in\mathbb{R}^n} \sum_{i=2}^{n}(x_i - x_{i-1})^2 + \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} x_i^2 + \sum_{i=\lfloor \frac{n}{2} \rfloor+1}^{n} (x_i - 1)^2$$

This is a convex quadratic problem. As the Steihaug method is a type of conjugate gradient method, we expect it to perform well on this type of problem and reach the minimum in a small number of iterations. When testing the Steihaug method and the CasADi IPOPT solver with this problem, we obtained some interesting results. In Figure 5, we compare the computing time of both solvers for different dimensions of the problem. In several dimensions, the Steihaug method computes the minimum even faster than the CasADi IPOPT solver. Even though we expect the Steihaug method to perform well on this problem, this result is surprising. Our Matlab implementation of the Steihaug method is not as optimized as the CasADi IPOPT solver, that is why we did not expect it to perform as well as it does on certain problems. To examine the performance of these two solvers further, we computed a performance profile comparing them.

E. Performance Profile

In this subsection we follow [8]. A performance profile is a useful tool to compare the performance of several solvers. For each problem p and solver s, we define $t_{p,s}$ as the required computing time to solve problem p with solver s. To compare the solvers, we use the *performance ratio* $r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s}:s\in\mathcal{S}\}}$, where $\mathcal{S}$ is the set of all solvers we are comparing. To obtain a performance profile, we compute the distribution function for the performance ratio:

$$\rho_s(\tau) = \frac{1}{n_p}\text{size}\{p \in \mathcal{P} : r_{p,s} \leq \tau\},$$

where n_p is the number of problems and $\mathcal{P}$ is the set of all problems. In general, solvers with large probability $\rho_s(t)$ are to be preferred.

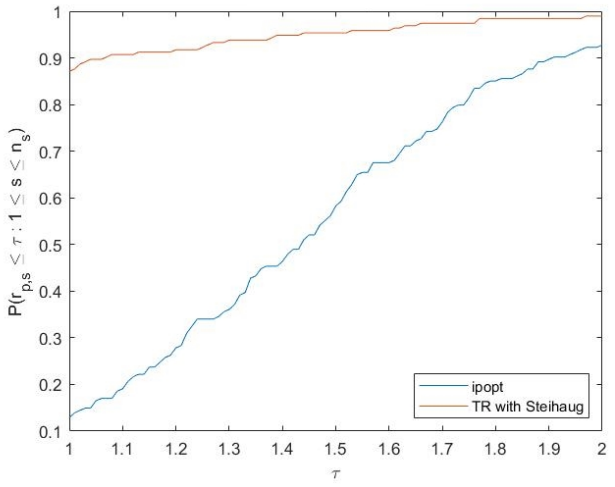

Fig. 6. Performance profile of the Steihaug method and CasADi IPOPT.

We computed a performance profile comparing the Steihaug method and the CasADi IPOPT solver. The set of problems consists of the Rosenbrock function in dimensions $d = 4, \ldots, 100$, as well as our second test problem in dimensions $d = 4, \ldots, 100$. The result can be seen in Figure 6. On this set of problems, the CasADi IPOPT solver clearly performs better than our implementation of the Steihaug method. Therefore, we can conclude that the observed behaviour in the previous subsection is a special case. As our implementation is not programmed in an optimal way, our experiments might not show the full potential of the Steihaug method.

IV. Conclusion

The Steihaug method is a simple and widely used way to solve the trust-region subproblem and is applicable to many problems. As it is a conjugate gradient type method, it works especially well on convex quadratic problems. In our experiments, we could confirm the superlinear convergence rate and the behaviour of $||p||$ which were proven in [1] and [3], respectively. In comparison with the CasADi IPOPT solver, the Steihaug method was slower in almost all cases we tested, with the exception of a convex quadratic test problem with specific dimensions. Studying this behaviour in certain dimensions would be an interesting task for the future. In addition, a more optimal implementation of the Steihaug method could improve the performance significantly and could lead to further interesting comparisons.

Acknowledgment

We would like to thank Prof. Moritz Diehl for suggesting the second test problem and for introducing us to performance profiles. We would also like to thank Florian Messerer for providing us with information about the Rosenbrock problem and suggesting it as a suitable test problem.

References

[1] J. Nocedal and S. J. Wright, "Numerical Optimization," Springer-Verlag, pp. 65–77, 1999.
[2] M. Diehl, "Numerical Optimization," Lecture Notes, Albert-Ludwigs-Universitaet Freiburg, 2020.
[3] T. Steihaug, "The Conjugate Gradient Method and Trust Regions in Large Scale Optimization," in SIAM Journal on Numerical Analysis, Vol. 20, No. 3, 1983, pp. 626–637.
[4] S. Bartels, "Numerik 3x9," Springer-Verlag, 2016.
[5] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," in Mathematical Programming Computation, Springer, Vol. 11, No. 1, 2019, pp 1–36.
[6] A. R. Conn, N. I. M. Gould and P. L. Toint, "Testing a Class of Methods for Solving Minimization Problems with Simple Bounds on the Variables," in Mathematics of Computation, Vol. 50, No. 182, 1988, pp. 399–430.
[7] A. Wächter and L. Bilger, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," in Mathematical Programming, 106, 2006, pp. 25–57.
[8] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," Math. Program., Ser. A 91: 201-213, 2002.