A Trust-Funnel Algorithm for Nonlinear Programming

Daniel P. Robinson
Johns Hopkins University
Department of Applied Mathematics and Statistics

Collaborators:
Frank E. Curtis (Lehigh University)
Nick I. M. Gould (Rutherford Appleton Laboratory)
Philippe Toint (University of Namur)

Southern California Optimization Day
May 23, 2014

Outline

1. A Trust-Funnel Algorithm
   - Overview
   - The normal step
   - The projected gradient step
   - The tangential step
   - Which steps to compute?
   - Step classification and update strategy
   - Summary

Outline

Outline

The aim of this talk

- Present an algorithm based on the trust-funnel concept for

$$\min_{x} \; f(x) \quad \text{s.t.} \quad c(x) \le 0$$

- By introducing slacks, we have the problem

$$\min_{x,s} \; f(x) \quad \text{s.t.} \quad c(x,s) = 0, \; s \ge 0$$

  where $c(x,s) := c(x) + s$

- We consider solving a sequence of barrier subproblems

$$\min_{x,s} \; f(x) - \mu \sum \ln([s]_i) =: f(x,s) \quad \text{s.t.} \quad c(x,s) = 0$$

- The naive approach of applying the equality constrained algorithm will not work because of the implicit constraint $s > 0$
- In particular, we require fraction-to-the boundary constraints, a slack reset procedure, and variable scaling

The barrier subproblem

$$\min_{x,s}\; f(x) - \mu \sum \ln([s]_i) =: f(x,s) \quad \text{s.t.} \quad c(x,s) = 0$$

Basic subproblem:

$$\min_{d=(d^x,d^s)} \; m_k^f(d) = f(x_k,s_k) + \nabla f(x_k,s_k)^T d + \tfrac{1}{2} d^T H_k d$$

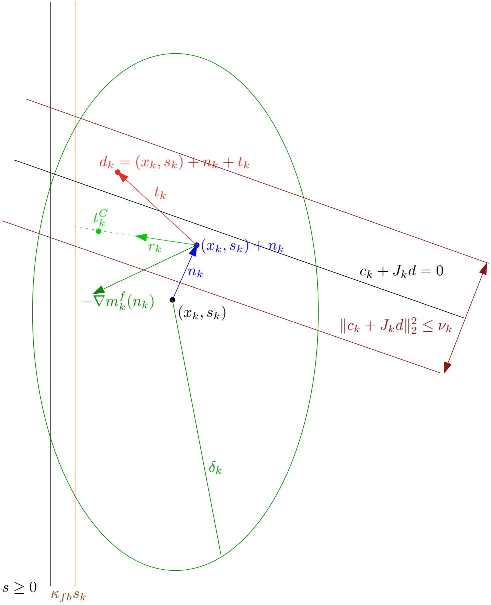
subject to

$$\begin{aligned} c(x_k,s_k) + J(x_k,s_k)d &= 0 \\ \|P_k^{-1} d\|_2 &\leq \delta_k \\ s_k + d^s &\geq \kappa_{\text{fb}} s_k \end{aligned}$$

where $\kappa_{\text{fb}} \in (0,1)$, $J(x,s) := \nabla c(x,s)$,

$$P_k = \begin{pmatrix} I & 0 \\ 0 & S_k \end{pmatrix} \quad \text{and} \quad H_k := \begin{pmatrix} \nabla^2_{xx}\mathcal{L}(x_k,y_k) & 0 \\ 0 & Y_k S_k^{-1} \end{pmatrix}$$

- Normal step computation
- Projected gradient step computation
- Tangential step computation
- y-iterations
- f-iterations
- v-iterations



Outline

1. A Trust-Funnel Algorithm
   - Overview
   - The normal step
   - The projected gradient step
   - The tangential step
   - Which steps to compute?
   - Step classification and update strategy
   - Summary

The normal step n_k

The idea

Aim to reduce $v(x,s) = \|c(x,s)\|_2$

How do we do this?

By computing an approximate solution to

$$\min_{n} \; m_k^v(n) \quad \text{s.t.} \quad \|P_k^{-1} n\| \leq \delta_k^v, \quad s_k + n^s \geq \kappa_{\text{fb}} s_k$$

- $m_k^v(n) = \|c(x_k, s_k) + J(x_k, s_k)n\|_2$
- $\kappa \in (0,1)$
- $\delta_k^v > 0$ is a trust-region radius

Note: Assume that we know a value $v_k^{\max}$ such that $v(x_k, s_k) \leq v_k^{\max}$

The normal step n_k

What do we mean by an approximate solution?

We require that n_k satisfies

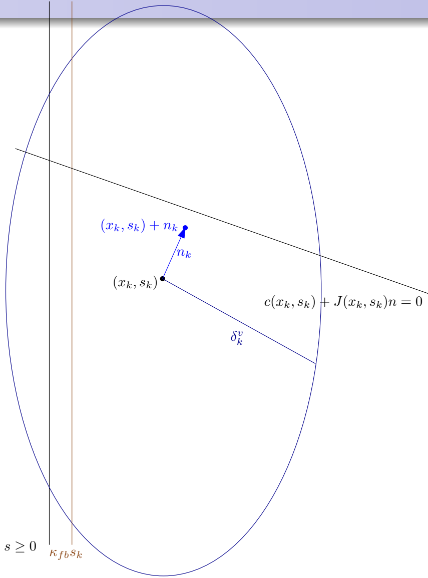
- $\|P_k^{-1} n_k\| \leq \delta_k^v$
- $s_k + n_k^s \geq \kappa_{\text{fb}} s_k$
- $m_k^v(n_k) \leq m_k^v(n_k^{\text{C}})$
- $P_k^{-1} n_k \in \textbf{range}\left(P_k J(x_k, s_k)^T\right)$

How can we guarantee this?

- truncated preconditioned CG
- dogleg approach

- $\pi_k^v = \|P_k J(x_k, y_k)^T c(x_k, s_k)\|$

Main diagram



Outline

An approximate projected gradient r_k

Define the approximate projected gradient as

$$r_k = -P_k^2\left[\nabla m_k^f(n_k) + J(x_k, s_k)^T y_k\right]$$

where y_k is an approximate solution to

$$\min_{y \in \mathbb{R}^m} \ \tfrac{1}{2}\left\|P_k\left[\nabla m_k^f(n_k) + J(x_k, s_k)^T y\right]\right\|^2$$

that satisfies at least one of

1. $\pi_k^f \le \epsilon$ and $v_k \le \epsilon$ (approximate KKT)
2. $\pi_k^f \le \tfrac{1}{2}\pi_k^v$ (do not compute a tangential step)
3. $\chi_k^f \ge \tfrac{1}{2}\pi_k^f$ (descent direction)

where

$$\pi_k^f = \left\|P_k\left[\nabla m_k^f(n_k) + J(x_k, s_k)^T y_k\right]\right\| \quad \text{and} \quad \chi_k^f = -\frac{\nabla m_k^f(n_k)^T r_k}{\pi_k^f}$$

Main diagram

Outline

1. A Trust-Funnel Algorithm
   - Overview
   - The normal step
   - The projected gradient step
   - The tangential step
   - Which steps to compute?
   - Step classification and update strategy
   - Summary

Relaxed SQP tangential step t_k

Define the Cauchy point

$$t_k^{\mathrm{C}} := t_k^{\mathrm{C}}(\alpha_{\mathrm{T}}^{\mathrm{C}}), \quad \text{where} \quad t_k^{\mathrm{C}}(\alpha) := \begin{pmatrix} t_k^{\mathrm{C}x}(\alpha) \\ t_k^{\mathrm{C}s}(\alpha) \end{pmatrix} := -\alpha \begin{pmatrix} r_k^x \\ r_k^s \end{pmatrix} = -\alpha r_k$$

and $\alpha_{\mathrm{T}}^{\mathrm{C}}$ is the minimizer of

$$\begin{aligned} \min_{\alpha \geq 0} \quad & m_k^f(n_k + t_k^{\mathrm{C}}(\alpha)) \\ \text{s.t.} \quad & \|P_k^{-1}(n_k + t_k^{\mathrm{C}}(\alpha))\| \leq \min\{\delta_k^v, \delta_k^f\} \\ & s_k + n_k^s + t_k^{\mathrm{C}s}(\alpha) \geq \kappa_{\mathrm{fb}}(s_k + n_k^s) \end{aligned}$$

Then, t_k is a relaxed SQP tangential step if

$$m_k^f(n_k + t_k) \leq m_k^f(n_k + t_k^{\mathrm{C}}) \tag{1a}$$

$$s_k + n_k^s + t_k^s \geq \kappa_{\mathrm{fb}}(s_k + n_k^s) \tag{1b}$$

$$\|P_k^{-1}(n_k + t_k)\|_2 \leq \min\{\delta_k^v, \delta_k^f\} \tag{1c}$$

$$m_k^v(n_k + t_k) \leq \kappa_{\mathrm{tg}} m_k^v(0) + (1 - \kappa_{\mathrm{tg}}) m_k^v(n_k) \tag{1d}$$

▸ Main diagram

Very Relaxed SQP tangential step t_k

Define the Cauchy point

$$t_k^{\mathsf{C}} = t_k^{\mathsf{C}}(\alpha_{\mathsf{T}}^{\mathsf{C}}), \quad \text{where} \quad t_k^{\mathsf{C}}(\alpha) := \begin{pmatrix} t_k^{\mathsf{C}x}(\alpha) \\ t_k^{\mathsf{C}s}(\alpha) \end{pmatrix} := -\alpha \begin{pmatrix} r_k^x \\ r_k^s \end{pmatrix} = -\alpha r_k$$

and $\alpha_{\mathsf{T}}^{\mathsf{C}}$ is the minimizer of

$$\begin{array}{ll} \min\limits_{\alpha \geq 0} & m_k^f(n_k + t_k^{\mathsf{C}}(\alpha)) \\ \text{s.t.} & \|P_k^{-1}(n_k + t_k^{\mathsf{C}}(\alpha))\| \leq \min\{\delta_k^v, \delta_k^f, \kappa_{\mathsf{v}} v_k^{\max}\} \\ & s_k + n_k^s + t_k^{\mathsf{C}s}(\alpha) \geq \kappa_{\mathsf{fb}}(s_k + n_k^s) \end{array}$$

Then, t_k is a very relaxed SQP tangential step if

$$m_k^f(n_k) - m_k^f(n_k + t_k) \geq m_k^f(n_k) - m_k^f(n_k + t_k^{\mathsf{C}}) \tag{2a}$$

$$s_k + n_k^s + t_k^s \geq \kappa_{\mathsf{fb}}(s_k + n_k^s) \tag{2b}$$

$$\|P_k^{-1}(n_k + t_k)\| \leq \min\{\delta_k^v, \delta_k^f, \kappa_{\mathsf{v}} v_k^{\max}\} \tag{2c}$$

$$m_k^v(n_k + t_k) \leq \kappa_{\mathsf{tt}} v_k^{\max} \tag{2d}$$

Main diagram

Outline

1. A Trust-Funnel Algorithm
   - Overview
   - The normal step
   - The projected gradient step
   - The tangential step
   - Which steps to compute?
   - Step classification and update strategy
   - Summary

Normal step

- Have to compute it: $\pi_k^v > \frac{1}{2}\pi_{k-1}^f$ or $v_k \geq 0.9 v_k^{\max}$
- Option to compute: $\pi_k^v > 0$
- Do not compute: $\pi_k^v = 0$

Projected gradient step

- Compute it: $\|P_k^{-1} n_k\| \leq 0.9 \min\{\delta_k^v, \delta_k^f\}$
- Option to compute it otherwise.

Tangential step

- Compute iff a projected gradient was computed and $\pi_k^f > \frac{1}{2}\pi_k^v$

Outline

1. A Trust-Funnel Algorithm
   - Overview
   - The normal step
   - The projected gradient step
   - The tangential step
   - Which steps to compute?
   - Step classification and update strategy
   - Summary

Three types of steps:

- y-iterations focus on better multiplier estimates
- f-iterations focus on reducing the barrier function f
- v-iterations focus on reducing infeasibility as measured by v

▸ Step computation diagram

Definition of a y-iteration

We classify the kth iteration as a y-iteration if $n_k = t_k = 0$.

Updates for a y-iteration

- $w_{k+1} \leftarrow w_k$
- $\delta^f_{k+1} \leftarrow \delta^f_k, \quad \delta^v_{k+1} \leftarrow \delta^v_k$
- $v^{\max}_{k+1} \leftarrow v^{\max}_k$

Notation: $w_k = (x_k, s_k)$ and $w_{k+1} = (x_{k+1}, s_{k+1})$

▸ Step computation diagram

Definition of an f-iteration

We classify the kth iteration as an f-iteration if $t_k \neq \mathbf{0}$

$$v(w_k + d_k) \leq v_k^{\max} \quad \text{(recall } v(w_k) \leq v_k^{\max}\text{)}$$

$$m_k^f(w_k) - m_k^f(w_k + d_k) \geq \tfrac{1}{2}\left[m_k^f(w_k + n_k) - m_k^f(w_k + d_k)\right]$$

Updates for an f-iteration

$\delta_{k+1}^v \leftarrow \delta_k^v, \quad v_{k+1}^{\max} \leftarrow v_k^{\max}$

If $f(w_k) - f(w_k + d_k) \geq \frac{1}{2}\left[m_k^f(w_k) - m_k^f(w_k + d_k)\right]$ **then**

- $w_{k+1} \leftarrow w_k + d_k$
- perform a slack reset
- possibly increase δ_k^f

else

- $w_{k+1} \leftarrow w_k$
- decrease δ_k^f

▸ Step computation diagram

Definition of a v-iteration

The kth iteration as a v-iteration if it is not a y- or an f-iteration.

Updates for a v-iteration

$\delta^f_{k+1} \leftarrow \delta^f_k$

If $n_k \neq 0$, $m^v_k(w_k) - m^v_k(w_k + d_k) \geq \frac{1}{2}\left[m^v_k(w_k) - m^v_k(w_k + n_k)\right]$, and
$v(w_k) - v(w_k + d_k) \geq \frac{1}{2}\left[m^v_k(w_k) - m^v_k(w_k + d_k)\right]$ **then**

- $w_{k+1} \leftarrow w_k + d_k$
- perform a slack reset
- possibly increase δ^v_k
- decrease $v^{\max}_k$

else

- $w_{k+1} \leftarrow w_k$, $v^{\max}_{k+1} \leftarrow v^{\max}_k$, decrease δ^v_k

Outline

Summary

- Presented an inexact barrier-SQP algorithm for solving general nonlinear optimization problems based on a trust-funnel approach.
- Trial steps are composite steps formed from a normal step (designed to improve feasibility) and a tangential step (designed to decrease the barrier objective function).
- The method is matrix free, i.e., all conditions may be obtained via iterative methods.
- Subsets of core calculations are performed during each iteration based on appropriate criticality measures.
- Effective preconditioning is a challenge.
- Numerical results are in progress (part of GALAHAD)

References

Nicholas I. M. Gould and Ph. L. Toint,
Nonlinear programming without a penalty function or a filter.
Mathematical Programming, 2009.

Nicholas I. M. Gould and Daniel P. Robinson and Ph. L. Toint,
Corrigendum: Nonlinear programming without a penalty function or a filter.
Mathematical Programming, 2011.