1 Optimization Problem

Problem is

$$\min_{\vec{x} \in X} f_0(\vec{x}), \tag{1}$$

$$f_i(\vec{x}) \leq 0, \tag{2}$$

for $i = 1, \ldots, m$, where X is a box:

$$\{\vec{x} \in \mathbb{R}^n \mid x_j^{\min} \leq x_j \leq x_j^{\max}\}. \tag{3}$$

2 Approximate Problem

Suppose kth iterate is $\vec{x}^{(k)}$. Then, for $i = 0, \ldots, m$, replace $f_i(\vec{x})$ with

$$g_i(\vec{x}) = f_i(\vec{x}^{(k)}) + \nabla f_i(\vec{x}^{(k)}) \cdot (\vec{x} - \vec{x}^{(k)}) + \frac{\rho_i^2}{2} \left| \frac{\vec{x} - \vec{x}^{(k)}}{\vec{\sigma}} \right|^2, \tag{4}$$

where $\vec{\sigma}$ and $\vec{\rho}$ are vectors and the division by $\vec{\sigma}$ is element-wise. And make a trust region

$$T = \{\vec{x} \mid |x_j - x_j^{(k)}| \leq \sigma_j\} \cap X. \tag{5}$$

So that the new problem is

$$\min_{x \in T} g_0(\vec{x}), \tag{6}$$

$$g_i(\vec{x}) \leq 0. \tag{7}$$

3 Overall Scheme

For the kth iteration, with current candidate $\vec{x}^{(k)}$:

1. Solve approximate problem to find candidate $\vec{x}^{(k+1)}$
2. Check conservative: $g_i(\vec{x}^{(k+1)}) < f_i(\vec{x}^{(k+1)})$.
   - If no, throw away candidate, double ρ_i for each non-conservative g_i, and solve approximate problem again.
3. Halve $\vec{\rho}$ (take bigger steps) and update $\vec{\sigma}$ (decrease σ_i if x_i oscillating, increase if monotonic i.e. heading somewhere else).

4 Solving approximate problem

4.1 Evaluating dual function

Lagrangian relaxation solves a minimization problem over the space

$$\Lambda = \{\vec{\lambda} \in \mathbb{R}^{m+1} | \lambda_0 = 1, \lambda \geq 0\}, \tag{8}$$

with the objective

$$\begin{aligned}
L(x, \vec{\lambda}) &= \sum_{i=0}^{m} \lambda_i g_i(\vec{x}) &(9)\\
&= \sum_{i=0}^{m} \lambda_i f_i(\vec{x}^{(k)}) + \left(\sum_{i=0}^{m} \lambda_i \nabla f_i(\vec{x}^{(k)}) \right) \cdot (\vec{x} - \vec{x}^{(k)}) &(10)\\
&\quad + \frac{1}{2} \left(\sum_{i=0}^{m} \lambda_i \rho_i \right) \left| \frac{\vec{x} - \vec{x}^{(k)}}{\vec{\sigma}} \right|^2 &(11)\\
&= \vec{\lambda} \cdot \vec{f}(\vec{x}^{(k)}) + \sum_{j=1}^{n} h_j(x_j - x_j^{(k)}), &(12)
\end{aligned}$$

where

$$h_j(\delta_j) = \left(\vec{\lambda} \cdot \nabla f(x_0)_j \right) \delta_j + \frac{1}{2\sigma_j} (\vec{\lambda} \cdot \vec{\rho}) \delta_j^2. \tag{13}$$

Define dual function,

$$\begin{aligned}
g(\vec{\lambda}) &= \min_{x \in T} L(x, \vec{\lambda}) &(14)\\
&= \vec{\lambda} \cdot \vec{f}(\vec{x}^{(k)}) + \sum_{j=1}^{n} \left(\min_{|\delta_j| \leq \sigma_j} h_j(\delta_j) \right). &(15)
\end{aligned}$$

Define for each j

$$\begin{aligned}
a_j &= \frac{1}{2\sigma_j^2} (\vec{\lambda} \cdot \vec{\rho}) &(16)\\
b_j &= \vec{\lambda} \cdot \nabla f(x_0)_j &(17)
\end{aligned}$$

Note that we can write

$$b = \nabla f(x_0)^T \vec{\lambda}. \tag{18}$$

We now have,

$$h_j(\delta_j) = b_j \delta_j + a_j \delta_j^2. \tag{19}$$

The minimum of $h_j(\delta_j)$ is found at

$$\delta_j^* = -\frac{b_j}{2a_j} \text{ clamped to } [-\sigma_j, \sigma_j]. \tag{20}$$

And hence we can determine

$$g(\vec{\lambda}) = \vec{\lambda} \cdot \vec{f}(\vec{x}^{(k)}) + \sum_{j=1}^{n} \left(b_j \delta_j^* + a_j (\delta_j^*)^2 \right). \tag{21}$$

Now let us compute the gradient. Note that,

$$\frac{\partial a_j}{\partial \lambda_i} = \frac{\rho_i}{2\sigma_j^2}, \tag{22}$$

$$\frac{\partial b_j}{\partial \lambda_i} = \nabla f_i(\vec{x}^{(k)})_j. \tag{23}$$

Let $S \subseteq \{1, \dots, n\}$. be the indices where we don't snap to bounds. Then,

$$\frac{\partial g}{\partial \lambda_i} = f_i(x^{(k)}) + \sum_{j=1}^{n} \left(\frac{\partial b_j}{\partial \lambda_i} \delta_j^* + \frac{\partial a_j}{\partial \lambda_i} (\delta_j^*)^2 \right) \tag{24}$$

$$+ \sum_{j \in S} (b_j + 2a_j \delta_j^*) \frac{\partial \delta_j^*}{\partial \lambda_i} \tag{25}$$

$$= f_i(x^{(k)}) + \nabla f_i(\vec{x}^{(k)}) \delta^* + \rho_i \sum_{j=1}^{n} \frac{(\delta_j^*)^2}{2\sigma_j^2}. \tag{26}$$

In vectorial form,

$$\frac{\partial g}{\partial \lambda} = f_i(x^{(k)}) + \nabla f(\vec{x}^{(k)}) \delta^* + \rho \sum_{j=1}^{n} \frac{(\delta_j^*)^2}{2\sigma_j^2}. \tag{27}$$

4.2 Maximizing dual function

The dual problem is,

$$\max_{y \geq 0} g(\lambda). \tag{28}$$

We can provide g and its gradient function recursively to CCSA, which will solve it for us.

5 Main Goals

- Support sparse Jacobians
- Support affine constraints
  - Does paper handle these? (First, find where paper handles box constraints.)
  - Maybe think of this as a more complicated X, rather than a simple f_i.