The goal of this method is to evenly sample the near-optimal feasible decision space of a linear optimization model using a Markov chain random walk algorithm.

1 Model

The model under consideration is linear and convex and can be written on the standard form:

$$
\begin{aligned}
\text{minimize} \quad & \mathbf{f}_0(\mathbf{x}) && (1)\\
\text{subject to} \quad & \mathbf{f}_i(\mathbf{x}) \leq 0 \quad i = 1...m && (2)\\
& \mathbf{h}_j(\mathbf{x}) = 0 \quad j = 1...p && (3)
\end{aligned}
$$

Besides from the constraints contained in the original optimization problem, an additional constraint is introduced, to separate the near-optimal feasible space, from the entire feasible space.

$$f_0(\mathbf{x}) \leq f_0(\hat{\mathbf{x}}) \cdot (1 + \epsilon) \tag{4}$$

Where $f_0(\hat{\mathbf{x}})$ is the objective value for the optimal solution.

As the constraints are linear they can be written as matrix products on the form:

$$\mathbf{f}_i(\mathbf{x}) \leq 0 \rightarrow A\mathbf{x} \leq \mathbf{b} \tag{5}$$

$$\mathbf{h}_j(\mathbf{x}) = 0 \rightarrow H\mathbf{x} = \mathbf{c} \tag{6}$$

The set containing all feasible near-optimal solutions then become:

$$F = \{\mathbf{x} \in \mathbf{R}^n | A\mathbf{x} \leq \mathbf{b} | H\mathbf{x} = \mathbf{c}\} \tag{7}$$

2 Presolve

In order to perform Markov chain random walk sampling a fully dimensional space is required. As several equalities are included in the definition of F and the inequalities might include bounds constraining variable ranges to zero, the fully dimensional subspace of F must be found.

The goal of the presolve process is to define the fully dimensional subspace Z of F.

Initially all linearly dependent constraints/rows in the augmented matrix $[A|\mathbf{b}]$ are identified. Linearly dependent constraints/rows, span parallel hyper-planes in F. If two such hyper-planes were to coincide, and have normal vectors pointing in opposite directions, they constrain a dimension of F

and can be represented as an equality rather than two inequalities. For all such inequality constraints constraining a dimension of F, their given row in A are moved from the matrix A to the H matrix.

From [1] (p. 523) we know that the problem can be reformulated as:

$$
\begin{aligned}
\text{minimize} \quad & \mathbf{f}_0(\hat{\mathbf{x}} + N\mathbf{z}) && (8)\\
\text{subject to} \quad & A(\hat{\mathbf{x}} + N\mathbf{z}) \leq \mathbf{b} \quad i = 1...m && (9)\\
& span(N) = null(H) && (10)
\end{aligned}
$$

Where the span of N is the null space of H and $\hat{\mathbf{x}}$ is any particular solution contained in F. Knowing this, the fully dimensional subspace of F can be defined as:

$$Z = \{\mathbf{z} \in \mathbf{R}^{n-p} | A(\hat{\mathbf{x}} + N\mathbf{z}) <= \mathbf{b}\} \tag{11}$$

Using this the optimization problem can be reformulated as:

$$
\begin{aligned}
\text{minimize} \quad & \mathbf{f}_0(\hat{\mathbf{x}} + N\mathbf{z}) && (12)\\
\text{subject to} \quad & \hat{A}\mathbf{z} \leq \hat{\mathbf{b}} && (13)\\
& \hat{A} = A \cdot N && (14)\\
& \hat{\mathbf{b}} = \mathbf{b} - A\hat{\mathbf{x}} && (15)
\end{aligned}
$$

By validating that the matrix $\hat{A}$ has full rank, we know that the subspace $\mathbf{z}$ is fully dimensional.

3 Markov chain random walk

A version of the random walk algorithm, is used to sample the subspace Z. The algorithm consist of an iterative process of drawing random directions, and taking steps of random lengths within the space Z.

Initially a point $\mathbf{z}_0$ inside the space Z must be provided. Random steps inside Z are then taken by generating a random direction θ and taking a random step t that will not violate the boundaries of Z.

$$\mathbf{z}_{i+1} = \mathbf{z}_i + \theta t \tag{16}$$

θ must be a unit vector pointing in a random direction evenly distributed on a unit hyper sphere $\mathbf{S}^{n-p}$. I will elaborate on how to do so!!!

To determine the range of t that ensures that the step t does not cross the boundary of Z, Equation 16 is substituded in to Equation 13.

$$\hat{A}(\mathbf{z}_i + \theta t) \leq \hat{b} \tag{17}$$

Isolating t will result in the upper and lower bounds:

$$t \leq \frac{\hat{b} - \hat{A}\mathbf{z}_i}{\hat{A}\theta} \text{ if } \hat{A}\theta > 0 \tag{18}$$

$$t \geq \frac{\hat{b} - \hat{A}\mathbf{z}_i}{\hat{A}\theta} \text{ if } \hat{A}\theta < 0 \tag{19}$$

Selecting t to be:

$$t = (t_{max} - t_{min})r + t_{min} \tag{20}$$

Where r is a random number drawn from a uniform distribution between 0-1.

Repeating the process of generating random directions θ and taking random step lengths t, will, if enough samples are drawn, generate a uniform sampling of the near optimal feasible set Z. Storing all samples z_i in the discrete set $Z^* = z_i \forall i = 0..d$ where d is the number of samples.

4 Decrush

Having sampled Z, these sample points must be reverted back to the F domain. This is done with a so called decrush algorithm.

$$\mathbf{x}_i = N\mathbf{z}_i + \hat{\mathbf{x}} \tag{21}$$

Repeating this for all samples a set containing d samples in F is obtained.

References

[1] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.