

The goal of this method is to evenly sample the near-optimal feasible decision space of a linear optimization model using a version of Markov Chain Monte Carlo (MCMC) sampling named the hit and run algorithm [1, 2].

## 1 Model

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

$$\text{minimize } \mathbf{f}_0(\mathbf{x}) \quad (1)$$

$$\text{subject to } \mathbf{f}_i(\mathbf{x}) \leq 0 \quad i = 1 \dots m \quad (2)$$

$$\mathbf{h}_j(\mathbf{x}) = 0 \quad j = 1 \dots p \quad (3)$$

With  $\mathbf{x} \in \mathbf{R}^n$ . 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.

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

Where  $\hat{\mathbf{x}}$  is an optimal solution to the original problem and  $f_0(\hat{\mathbf{x}})$  is the objective value for the optimal solution. The slack on optimality is given by the slack variable  $\theta$ .

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

$$\mathbf{f}_i(\mathbf{x}) \leq 0 \Rightarrow A\mathbf{x} \leq \mathbf{b} \quad (5)$$

$$\mathbf{h}_j(\mathbf{x}) = 0 \Rightarrow H\mathbf{x} = \mathbf{c} \quad (6)$$

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

$$F = \{\mathbf{x} \in \mathbf{R}^n | A\mathbf{x} \leq \mathbf{b} \wedge H\mathbf{x} = \mathbf{c}\} \quad (7)$$

## 2 Presolve

In order to apply the hit and run sampling algorithm, a fully dimensional space is required. As several equalities are included in the definition of  $F$  and hidden equalities might be found among the inequality constraints, the fully dimensional subspace of  $F$  must be found. Hidden equalities occur when two inequalities limit the range of a variable to zero.

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

## 2.1 Step 1

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.

## 2.2 Step 2

From [3] (p. 523) we know that the problem can be reformulated as:

$$\text{minimize } \mathbf{f}_0(\hat{\mathbf{x}} + N\mathbf{z}) \quad (8)$$

$$\text{subject to } A(\hat{\mathbf{x}} + N\mathbf{z}) \leq \mathbf{b} \quad i = 1 \dots m \quad (9)$$

$$\text{span}(N) = \mathcal{N}(H) \quad (10)$$

Where the span of  $N$  is the null space of  $H$  and  $\hat{\mathbf{x}}$  is any particular solution contained in  $F$ . As all equalities have been eliminated from the problem, the fully dimensional subspace of  $F$  can be defined as:

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

Using this the optimization problem can be reformulated as:

$$\text{minimize } \mathbf{f}_0(\hat{\mathbf{x}} + N\mathbf{z}) \quad (12)$$

$$\text{subject to } \hat{A}\mathbf{z} \leq \hat{\mathbf{b}} \quad (13)$$

With  $\hat{A}$  and  $\hat{\mathbf{b}}$ :

$$\hat{A} = A \cdot N \quad (14)$$

$$\hat{\mathbf{b}} = \mathbf{b} - A\hat{\mathbf{x}} \quad (15)$$

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

## 2.3 Calculating the Null space $N$

Using QR decomposition the Null space of a matrix can be calculated.

Given the  $(m \times n)$  matrix  $H$ , it can be decomposed to the orthogonal  $(m \times m)$  matrix  $Q$ , and  $R$  which is a  $(m \times n)$  upper triangular matrix.

$$H = QR = [Q_1 Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \quad (16)$$

Where  $Q_1$  is  $(m \times r)$  and  $Q_2$   $(m \times m - r)$ , with  $r$  being the rank of  $H$ .  
The null space of  $H$  is can then by found as  $Q_2 = N(H^T)$   
For proof see [4]

### 3 Hit and run sampling

A version of the hit and run algorithm [5], 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  $\theta$  and taking a random step  $t$  that will not violate the boundaries of  $Z$ .

$$\mathbf{z}_{i+1} = \mathbf{z}_i + \theta t \quad (17)$$

$\theta$  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!!! But this is easy.

To determine the range of  $t$  that ensures that the step  $t$  does not cross the boundary of  $Z$ , Equation 17 is substituted in to Equation 13.

$$\hat{A}(\mathbf{z}_i + \theta t) \leq \hat{b} \quad (18)$$

Isolating  $t$  will result in the upper and lower bounds:

$$t_{max} = \inf \frac{\hat{b} - \hat{A}\mathbf{z}_i}{\hat{A}\theta} \text{ given } \hat{A}\theta > 0 \quad (19)$$

$$t_{min} = \sup \frac{\hat{b} - \hat{A}\mathbf{z}_i}{\hat{A}\theta} \text{ given } \hat{A}\theta < 0 \quad (20)$$

Selecting  $t$  to be:

$$t = (t_{max} - t_{min})r + t_{min} \quad (21)$$

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

Repeating the process of generating random directions  $\theta$  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  $\mathbf{z}_i$  in the discrete set  $Z^* = \mathbf{z}_i \forall i = 0..d$  where  $d$  is the number of samples.

”The reason that the Markov chain corresponding to the iterates  $Z_0, Z_1, \dots, Z_d$  converges in distribution to a uniform distribution over  $Z$  is easily seen from the fact that 1) it is possible to go from any point in  $Z$  to a neighborhood of any other point in one step, and 2 ) the uniform distribution is a stationary distribution of the chain.” [6]

The hit and run algorithm is  $md$  hard [7]. Where  $m$  is the number of constraints and  $d$  is the number of variables.

## 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.

$$\mathbf{x}_i = N\mathbf{z}_i + \hat{\mathbf{x}} \quad (22)$$

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

## 5 Application

Currently the method has been implemented, with a working presolve and samling method. The model works on Pypsa models and has been run on the Prime server. All code available at : <https://github.com/TimToernes/MAA2>

## 6 Issues with this method

The hit and run sampler must operate on the entire problem. Meaning that all variables included in the model is part of the problem. The network models considered consist of  $\approx 100$  investment variable and  $\approx 1e9$  time dependent variables. As it is only the investment variables that is of true interest, the problem becomes extremely large due to the time dependent variables. Currently I have not determined a method of separating the investment and time dependent variables into two separate problems. Methods such as benders-decomposition has been considered.

As the mixing time (number of samples needed) has complexity  $O(n^3)$  for the hit and run algorithm, it is simply not feasible to draw enough samples, as it would require years of computing. Therefore this method has been discarded for now. The method has however been found feasible for problems up to  $d = 1000$ .

## 7 Ideas for further work

Figure out if sampling  $Z$  evenly, generates evenly distributed samples in  $F$  when samples are mapped from  $Z$  to  $F$ .

As of now, i have had sucess with applying the method on problems with  $n \approx 10.000$ , and i believe that  $n \approx 100.000$  is doable on a better pc. This is however one to two order of magnitude from the 1.000.000+ variables included in most modern energy system optimization models. Therefore i have considered two options. 1) Using decomposition methods, such as Benders decomposition to decompose the problem in to manageable sub-problems. Specifically it is only the non-timedependable variables that are truly of interest, and as these only make up  $\approx 1000$  of the variables in the energy system model. By decomposing the timedependable variables from the non-timedependable variables, and sampling the non-timedependable subspace, could be a solution. I have, however, not managed to figure out if this is a viable solution, as decomposition methods often require several iterations between the two sub-problems. 2) The largest contributor to the large number of variables in energy system models, is the time-series data used. By generating augmented time-series, using techniques such as self organizing maps [8], one can generate shorter time-series, thereby reducing the problem. This is however, a less favorable option than solution (1) as using augmented time-series

## References

- [1] Yuansi Chen et al. “Fast MCMC Sampling Algorithms on Polytopes”. In: (), p. 86.
- [2] Seksan Kiatsupaibul, Robert L. Smith, and Zelda B. Zabinsky. “An analysis of a variation of hit-and-run for uniform sampling from general regions”. In: *ACM Transactions on Modeling and Computer Simulation* 21.3 (Mar. 2011), pp. 1–11. ISSN: 1049-3301, 1558-1195. DOI: 10.1145/1921598.1921600. URL: <https://dl.acm.org/doi/10.1145/1921598.1921600> (visited on 10/01/2020).
- [3] Stephen Boyd and Lieven Vandenbergh. *Convex Optimization*. Cambridge University Press, 2004.
- [4] Scott Hyde. “Orthonormal Basis for the Column Space and the Null Space using the QR Decomposition”. In: (), p. 2. URL: <https://jekyll.math.byuh.edu/courses/m343/handouts/qrbasis.pdf>.
- [5] Robert L. Smith. “Efficient Monte Carlo Procedures for Generating Points Uniformly Distributed over Bounded Regions”. In: *Operations Research* 32.6 (1984), pp. 1296–1308. DOI: 10.1287/opre.32.6.1296. eprint: <https://doi.org/10.1287/opre.32.6.1296>. URL: <https://doi.org/10.1287/opre.32.6.1296>.
- [6] R. L. Smith. “The hit-and-run sampler: a globally reaching markov chain sampler for generating arbitrary multivariate distributions”. In: *Proceedings Winter Simulation Conference*. 1996, pp. 260–264.
- [7] Claude Belisle, Arnon Boneh, and Richard Caron. “Convergence properties of Hit-and-Run samplers”. In: *Communications in Statistics. Stochastic Models* 14 (Jan. 1998). DOI: 10.1080/15326349808807500.
- [8] Hasan Umitcan Yilmaz et al. “Reducing energy time series for energy system models via self-organizing maps”. In: *it - Information Technology* 61.2-3 (1Apr. 2019), pp. 125 –133. DOI: <https://doi.org/10.1515/itit-2019-0025>. URL: <https://www.degruyter.com/view/journals/itit/61/2-3/article-p125.xml>.