## Introduction

The present problem under study involves sampling of high dimensional spaces with complex, non-linear constraints. The sampling algorithm used for this purpose is Latin Hypercube Sampling (LHS). LHS is a form of stratified sampling algorithm that can be extended in multiple dimensions to generate random samples from a multidimensional distribution.

LHS ensures that a range of values of variables is represented in the samples of the dimensions has all its range represented. In the context of statistical sampling, a square grid containing sample positions is a Latin square if (and only if) there is only one sample in each row and each column. A Latin hypercube is essentially the generalization of this concept to an arbitrary number of dimensions, whereby each sample is the only one in each axis-aligned hyperplane containing it.

Let us consider the problem of generating five samples from two independent uniformly distributed variables $X$ and $Y$ such that $X, Y \in [0,1]$. According to LHS, the range of $X, Y$ are both divided into five equal intervals, resulting in $5 \times 5 = 25$ cells. The requirement of LHS is that each row and each column of a list contains only one sample. This ensures that even though there are only five samples, each of the five intervals of both $X, Y$ will be sampled. For each sample $[i, j]$, the sample values of $X, Y$ are determined by

$$X = F_X^{-1}\left(\frac{i - 1 + \xi_X}{n}\right)$$

$$Y = F_Y^{-1}\left(\frac{i - 1 + \xi_Y}{n}\right)$$

Where $n$ is the sample size, $\xi_X, \xi_Y$ are random numbers ($\xi_X, \xi_Y \in [0,1]$), and $F_X$ and $F_Y$ are the cumulative probability distribution functions of $X$ and $Y$ respectively. Stratification segregates the cumulative distribution curve into equal intervals on the cumulative probability scale (0 to 1.0). This algorithm can be extended to multiple dimensions.

LHS typically requires less samples than the popular Monte Carlo Random Sampling methods. By representing each variable as its Cumulative Distribution Function (CDF) (prior distribution) and partitioning the CDF into N regions and taking a single sample from each region, increases the likelihood that the full range of the posterior distribution is sampled. Once a suitable CDF sample is made, the sample CDF value is inversely mapped back to a parameter value.

## Method

The methodology adopted here can be summarized in the following steps:

1. Generate specified number of samples using LHS.
2. For each vector generated, it is checked against all the constraints that are specified.
3. Vectors which satisfy all the constraints are the new candidates which arewritten into an output file.

In the sampling process, the minimum allowable distance between any pair of points is $h\sqrt{n}$ where $h$ is the grid size and $n$ is the number of samples. Sampling is performed such that the minimum distance between any of the sampled pairs exceeds the minimum distance (nearest-neighbor) by a specified factor. We define this factor as ratio. Hence the convergence criterion for the sampling process is given by,

$$Min.Dist > ratio \ \times h\sqrt{n}$$

If ratio is low, samples tend to be closer to each other than when the ratio is high. For example, in figure 1, for a two-dimensional sampling, it is observed that when ratio = 0.5 (right), the samples are comparatively close to each other than when the ratio = 1.0 (left). While sampling in a low-dimensional space a high value of the ratio would imply a lower chance of convergence.

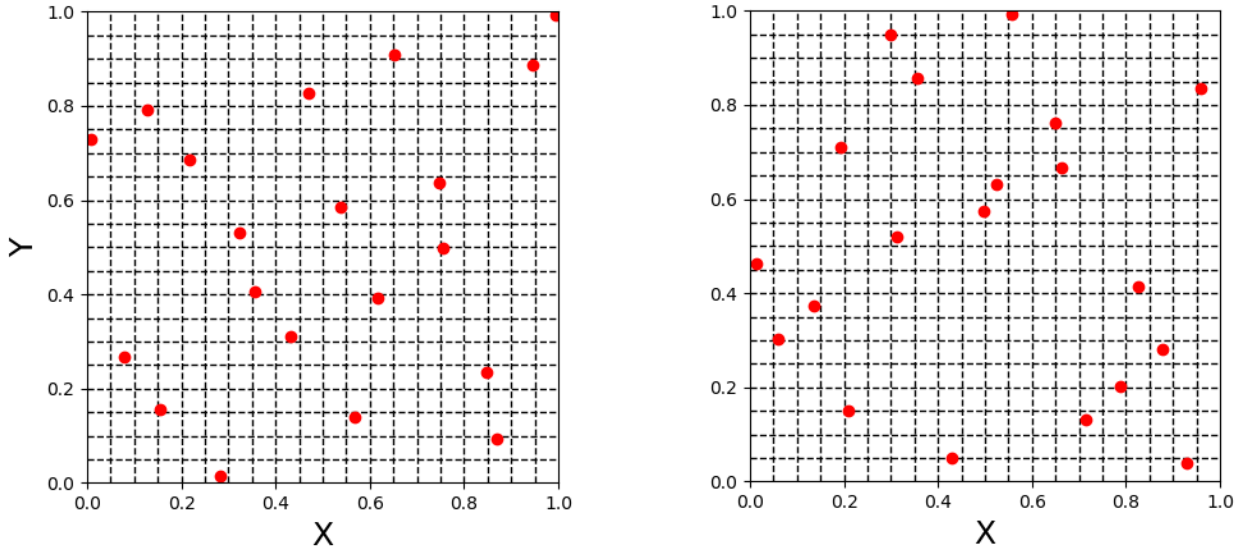

Figure 1: A two-dimensional sampling with ratio = 1.0 (left) and ratio = 0.5 (right)

The implementation of the algorithm has been adopted from Ref. 1. The samples are drawn via inverse transform sampling of a Uniform distribution in the interval [0,1].

## Results

To test the efficiency of the sampling process, pair plots have been generated for the examples provided. The histograms on the diagonals in figs. 2-5 represent the distribution of each of the feature vectors in the Euclidean space. The upper and the lower triangles depict the scatter of the feature vectors with each other. It is observed that the samples generated effectively span the Euclidean space. Figure 2 – 5 show the pair plots for each of the examples provided with 1000 samples.

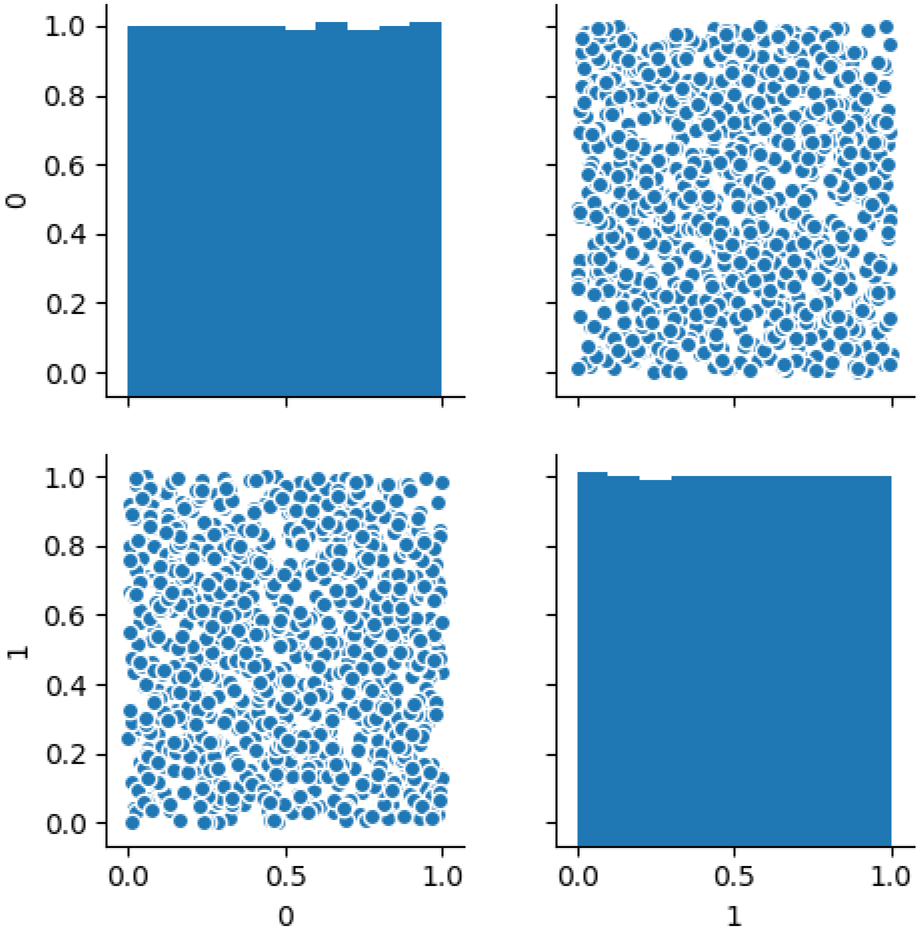

Figure 2: Mixture.txt

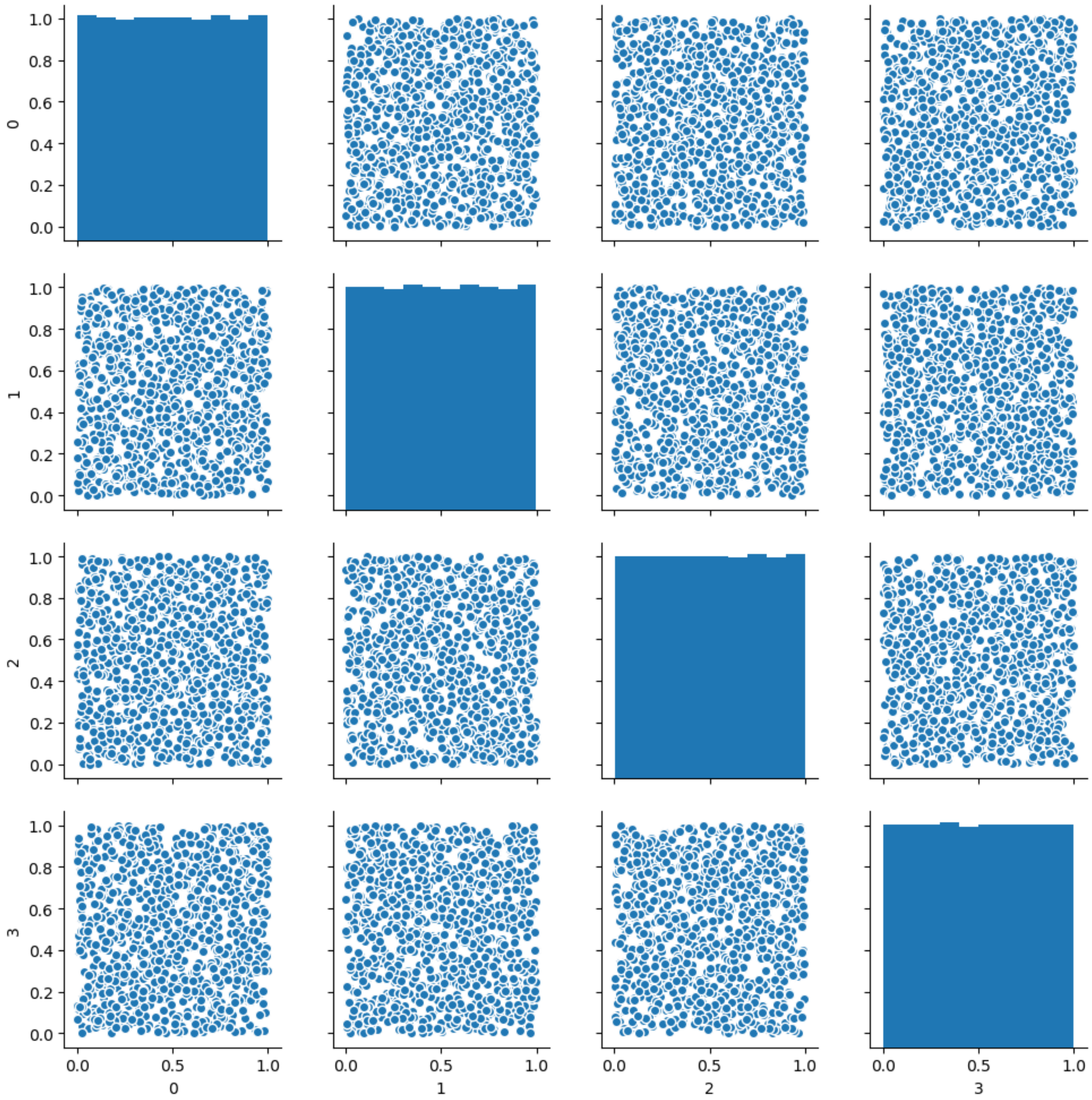

Figure 3: Formulation.txt

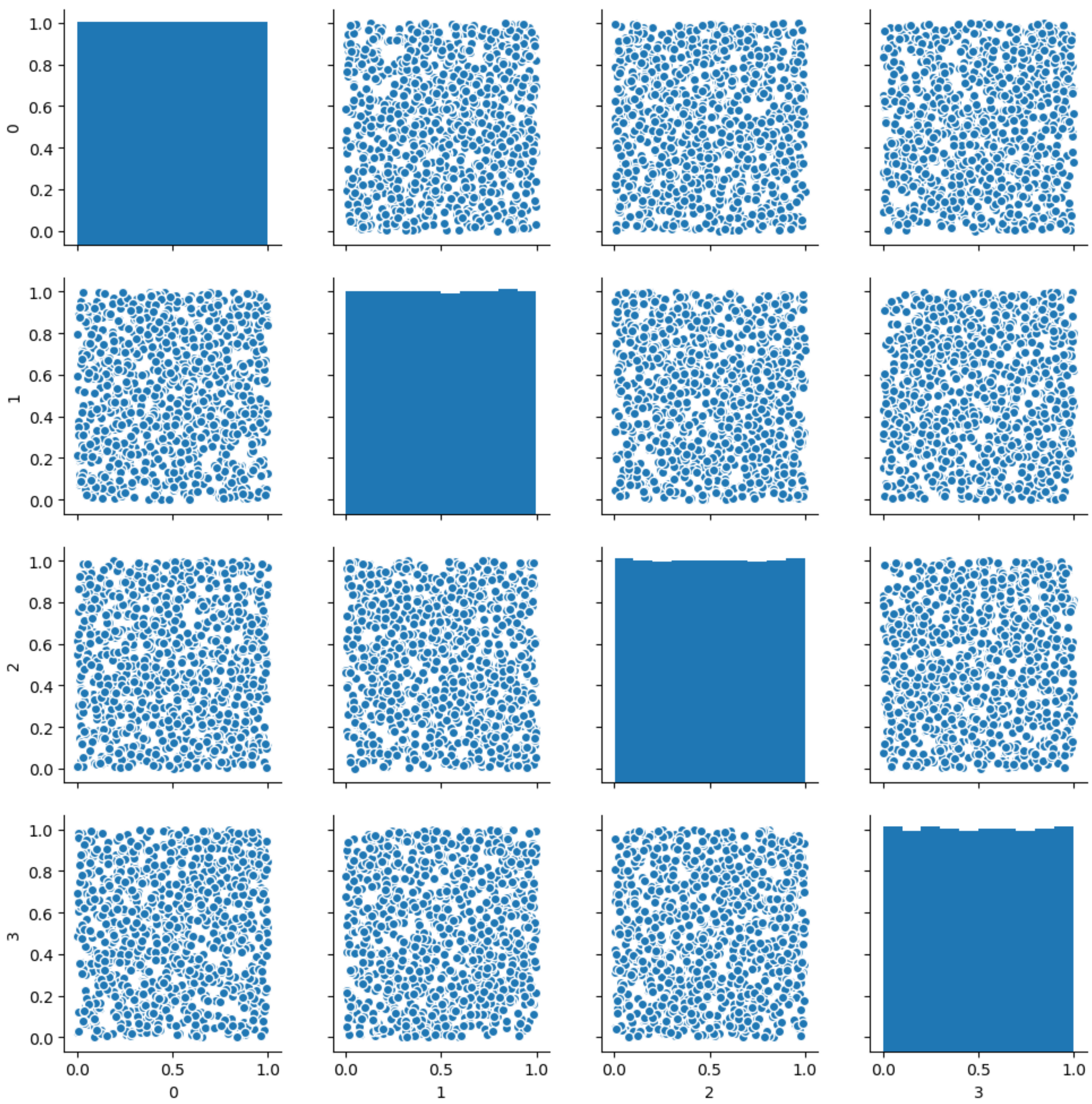

Figure 4: Example.txt

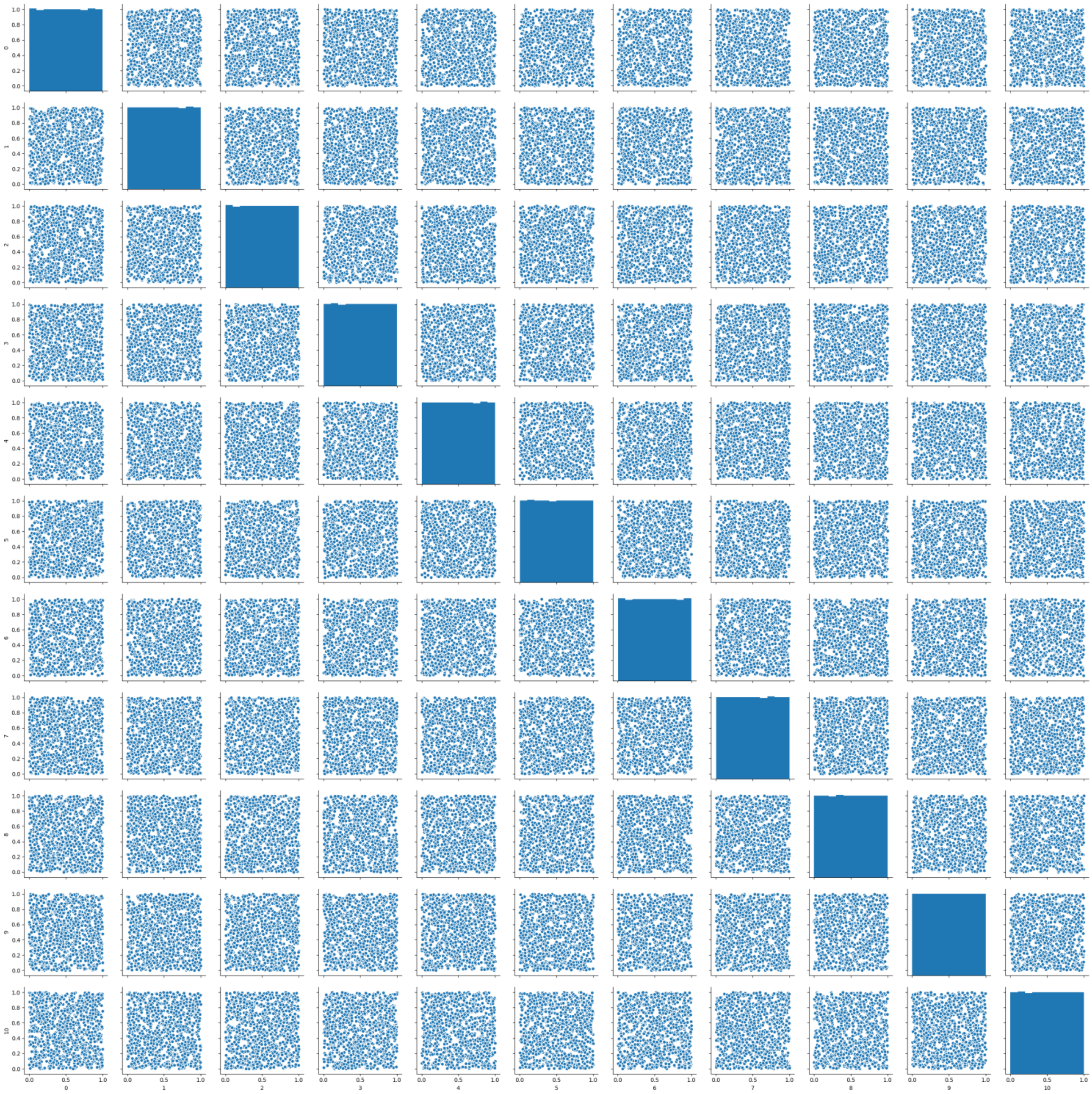

Figure 5: Alloy.txt

The table below shows realizations from one simulation. It is observed that for a simple 3 component mixture (mixture.txt) with low dimensionality and relatively number of candidates generated by the algorithm is 485. For the formulation (formulation.txt) no new candidates could be generated with 1000 samples. It could be attributed to the constraints present. For “Example.txt” file, the algorithm generates 185 samples. However, for “Alloy.txt”, it is also observed that no new candidates could be generated with 1000 samples. Although, the sampling algorithm effectively samples the space with 11 dimensions, the number of constraints limit the discovery of new candidates.

| Example File | # dimensions | No. of Constraints | No. of Samples | No. of new candidates generated |
|---|---|---|---|---|
| Mixture.txt | 2 | 1 | 1000 | 485 |
| Formulation.txt | 4 | 5 | 1000 | 0 |
| Example.txt | 4 | 3 | 1000 | 185 |
| Alloy.txt | 11 | 23 | 1000 | 0 |

## Conclusion and Future Work

The current exercise demonstrates the Latin Hypercube Sampling in high dimensions with non-linear constraints to generate new candidates. The samples have been generated using Inverse Transform Sampling of a Uniform distribution in [0,1]. It has been ensured that the samples are spread across all possible values. The algorithm has been tested with multiple example files that were provided. It appears that with high dimensions and greater number of constraints. It is worth exploring other probability distributions to generate samples in future work.