

Lattice-Based Kernel Methods For Identifying Feasible Regions In Design Space

†R A Bates, †H P Wynn & ‡E Fraga

r.a.bates@lse.ac.uk

†London School of Economics and Political Science,

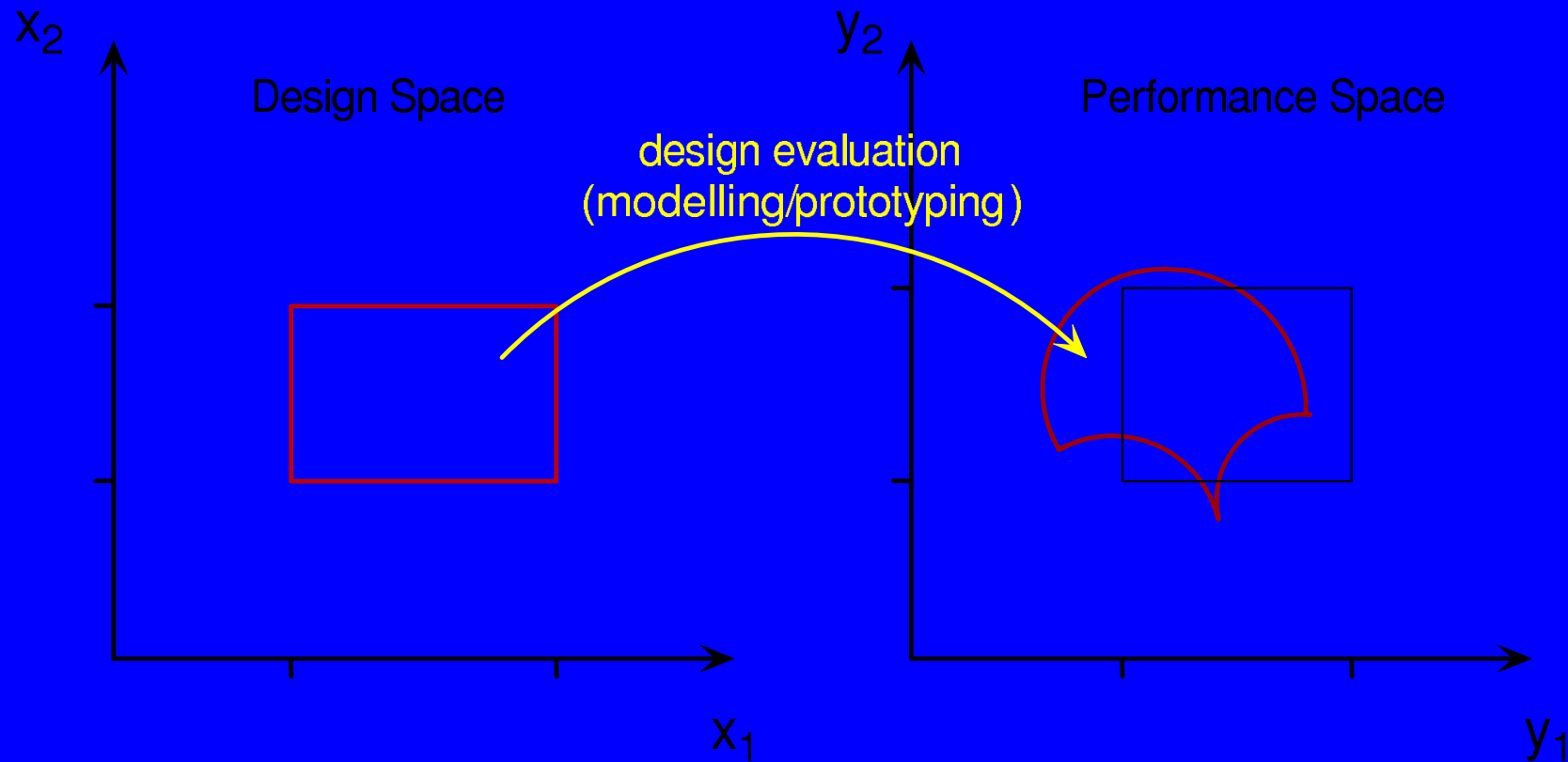
‡University College London

Introduction

- We are interested in systematic design improvement.
- This requires defining a search space of all possible design solutions: the Design Space.
- The Design Space, \mathcal{D} , is defined by a parameterisation of a system.
- There may be regions within this space that represent infeasible designs.

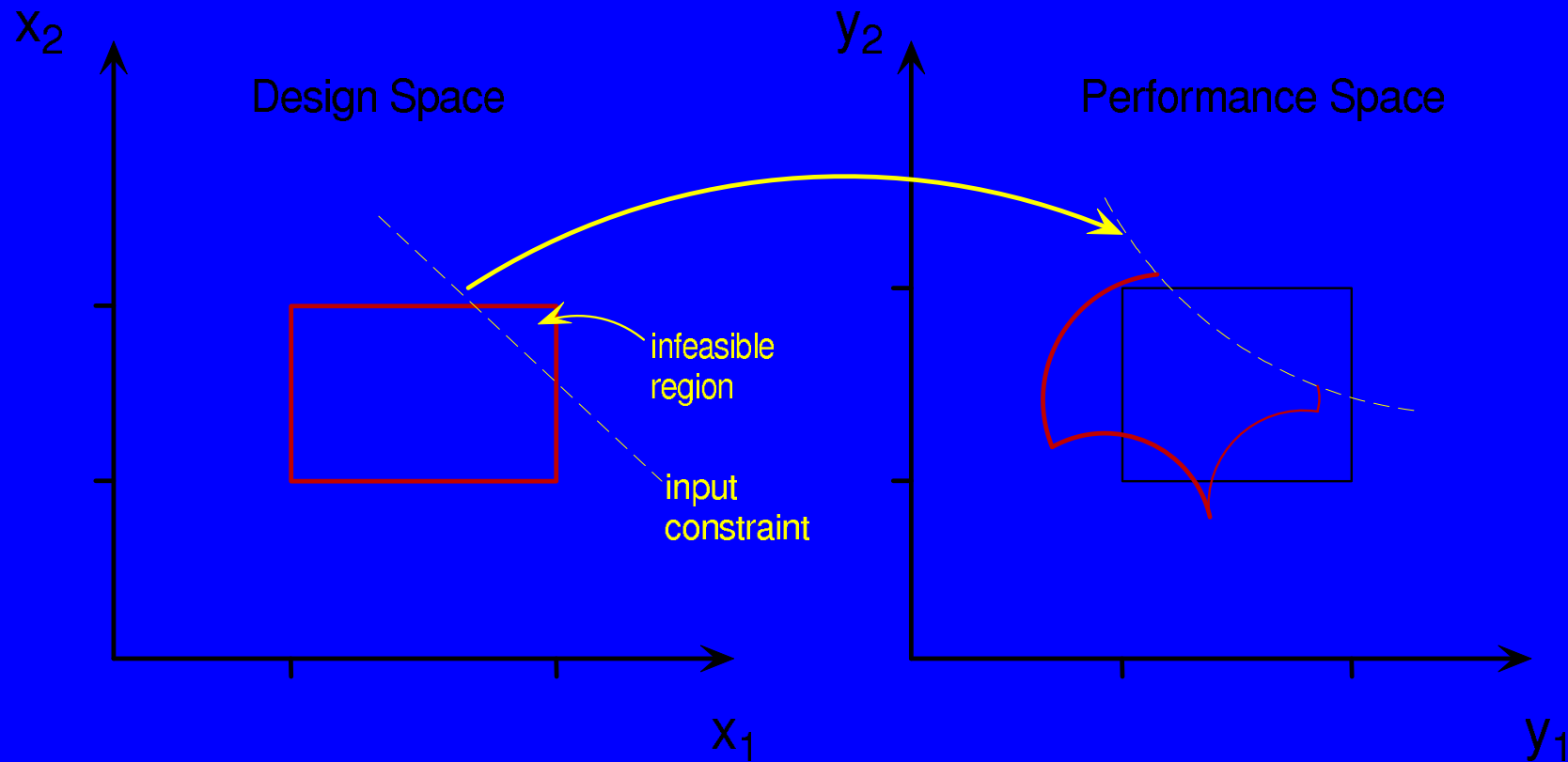
The Design Space

- Design Optimisation takes place within a predefined Design Space, \mathcal{D}



Infeasible regions within \mathcal{D}

- Certain areas of the Design Space may not produce feasible designs...

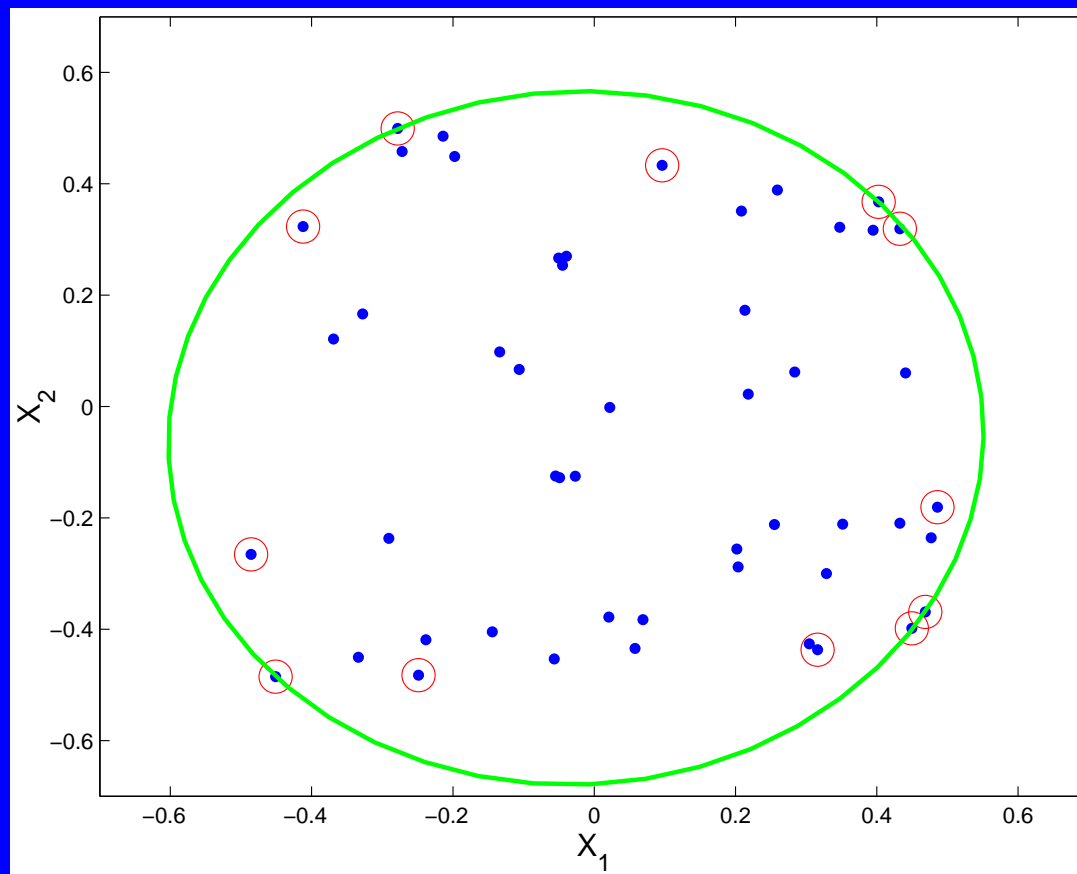


Motivation

- It is important to know the location of the feasible region(s).
 - Search algorithms need to know if a design is feasible or not
 - A better understanding of the relationship between different model parameters can be gained.
- Simple, known input constraints, e.g. $x_1 + x_2 < C$ can be used directly by a numerical optimizer, but often it is not possible to characterise a feasible region analytically.
- Trial-and-error simulation can be very costly.

Feasible Region: Convex Problems (1)

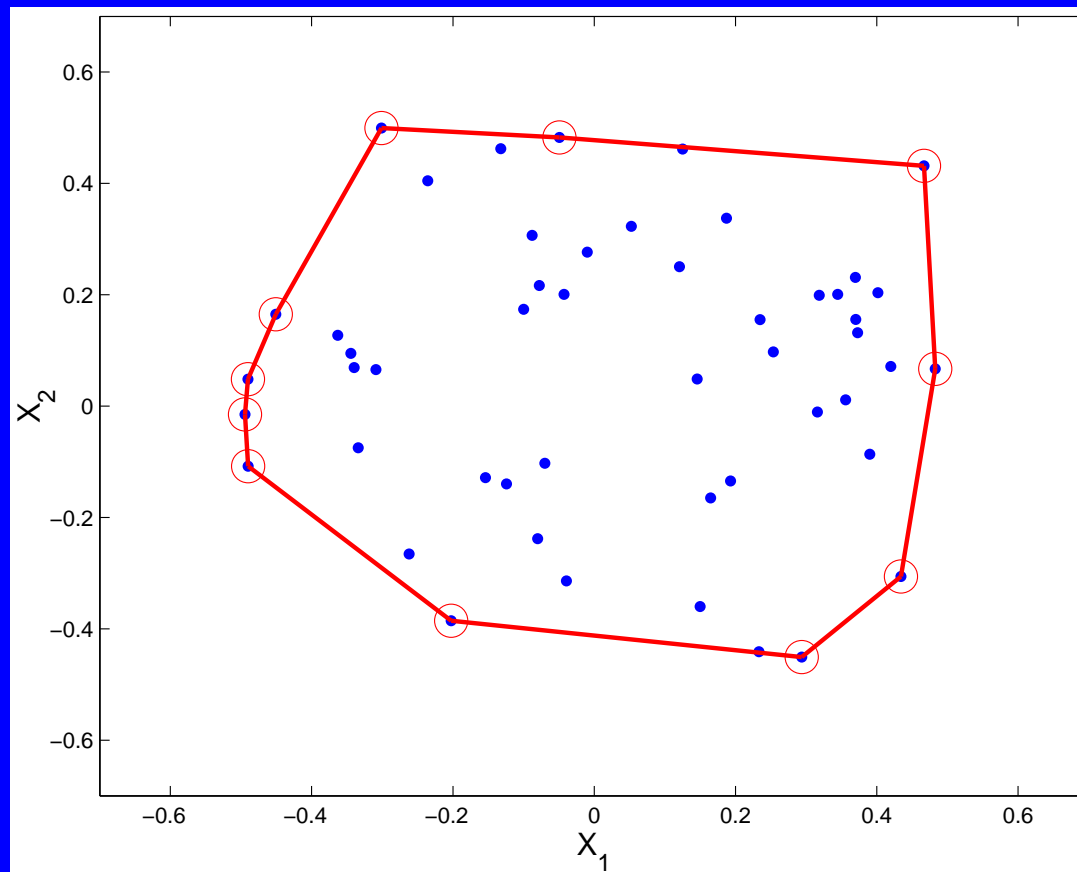
- Minimum volume outer ellipse
- No need to find boundary points.



Feasible Region: Convex Problems (2)

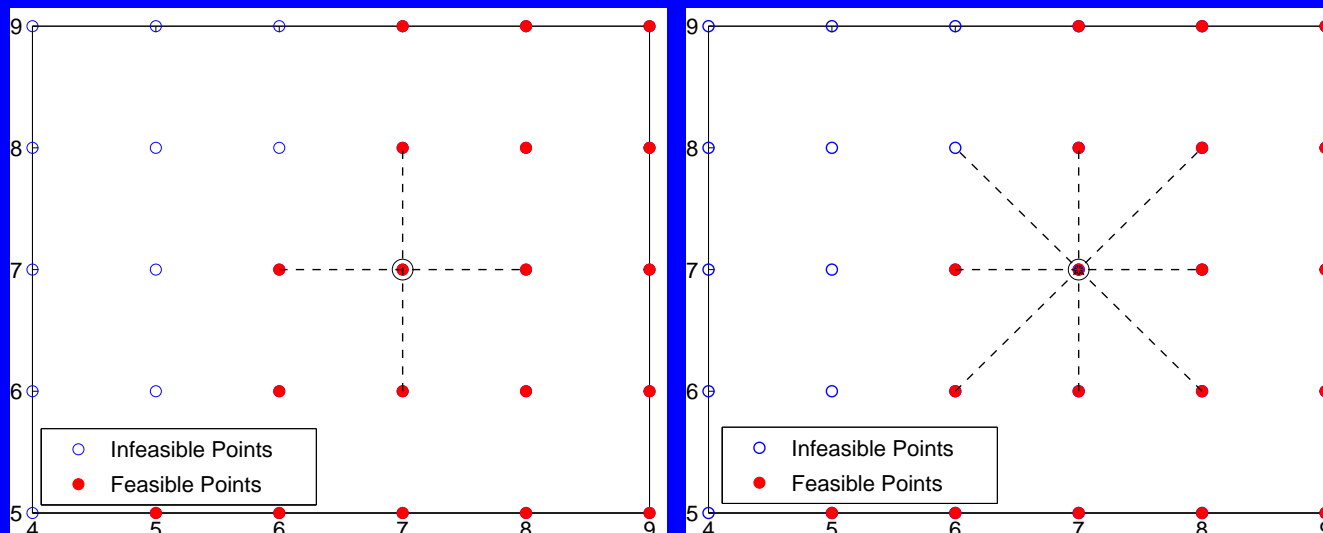
Convex Hull

- Determines boundary points, works for $d < 10$.



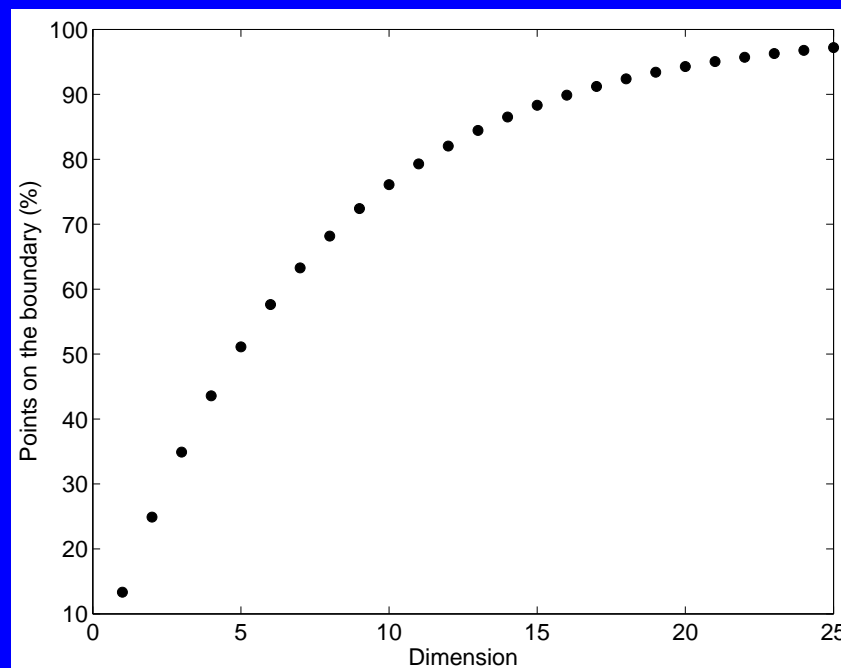
Boundary Points for Non-Convex Problems

- Need to determine boundary points.
- For a feasible region $A \in \mathcal{R}^d$, boundary points are defined as points x such that every neighborhood of x contains at least one point in A and at least one point not in A .
- Two definitions are ‘weak’ (convex) and ‘strong’ (non-convex):



Are feasible points inside the boundary?

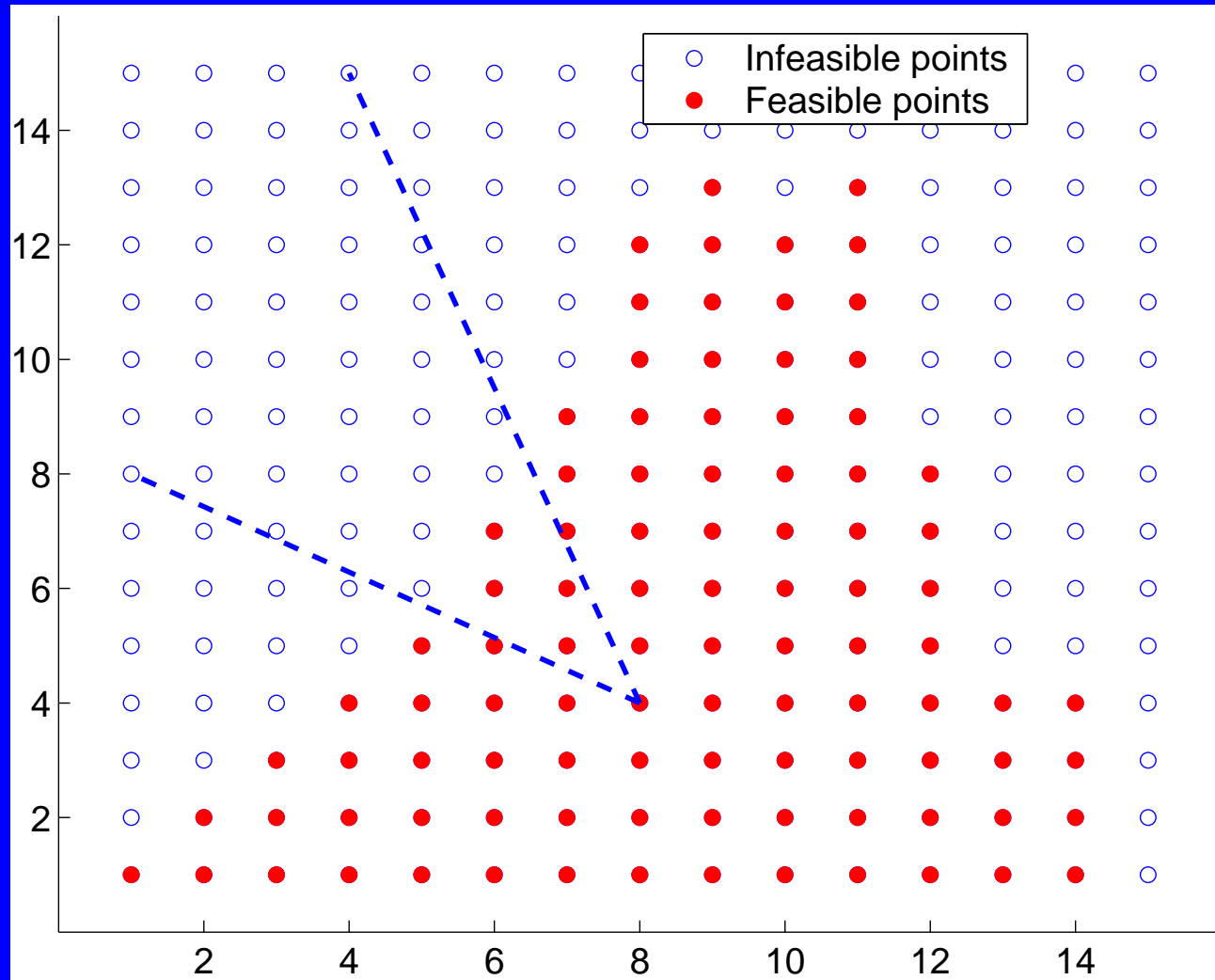
- Consider an integer grid with k levels and d dimensions
- Total pts: k^d , Internal: $(k - 2)^d$, Boundary: $k^d - (k - 2)^d$
- For $d = 5$, $k = 15$, Total points: $15^5 = 759,375$
- Assuming *all* points feasible, Internal pts = $13^5 = 371,293$ (49%), and 51% are on the boundary.



Using Search Cones (1)

- Use polyhedral cones, or *search cones*, to find boundary points.
- First step is to define an origin, usually at the centre of mass of the feasible points.
- Feasible region is then assumed to be star-shaped with respect to this origin.
- Boundary point in a cone is defined as the point inside the cone furthest from the origin.

Using Search Cones (2)



Hilbert Bases and Search Cones

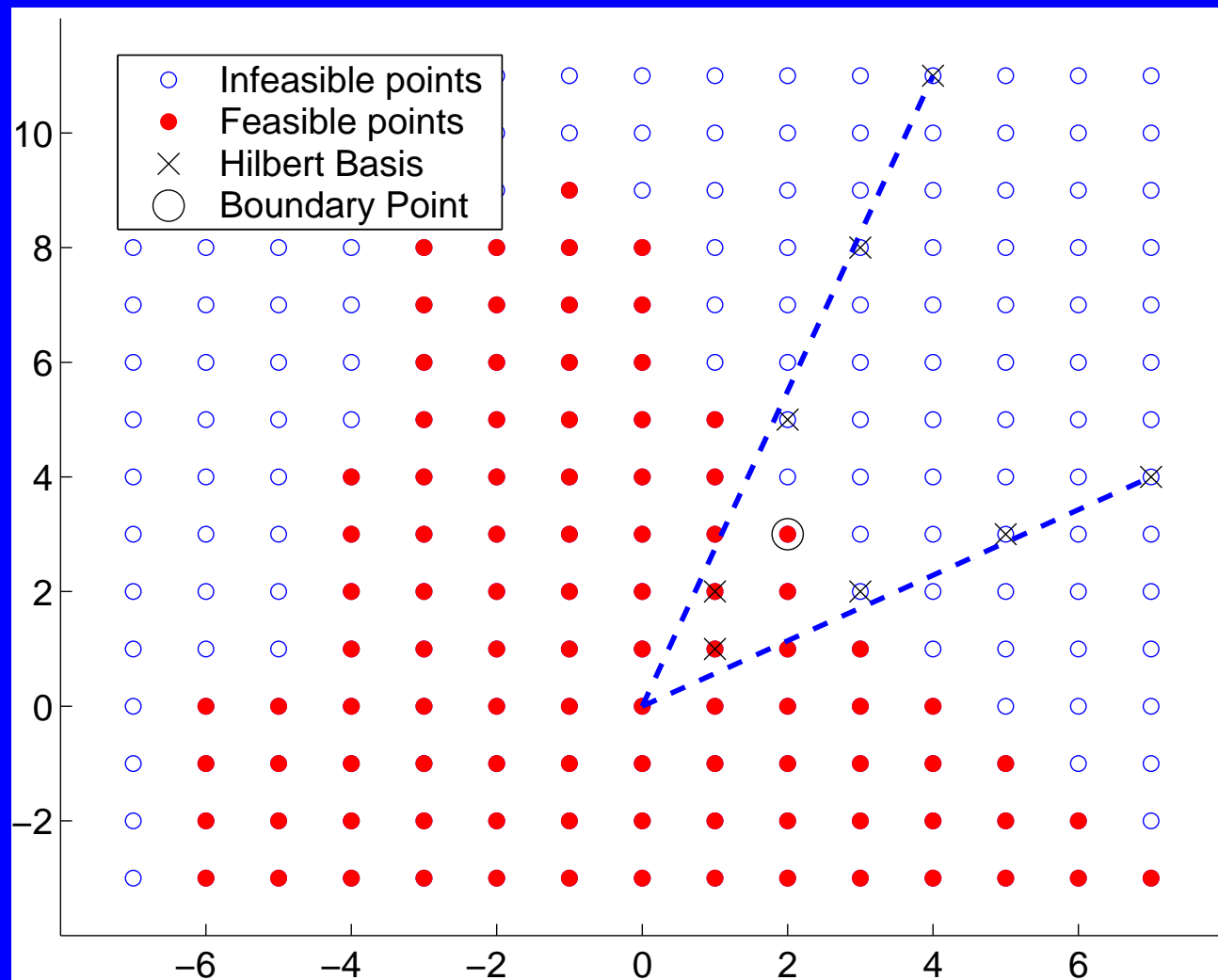
- We can use a method based on Hilbert Bases to determine which design points lie inside any given cone
- A Hilbert Basis of a cone with integer generating vectors is a special set of generators $\{g_1, \dots, g_k\}$ such that every integer grid point inside the cone can be represented as $\sum n_i g_i$ for some non-negative integer n_i .
- We can use a freely available algorithm, 4ti2: www.4ti2.de in order to compute the Hilbert Basis of points inside the cone

Algorithm for using Hilbert Bases

1. Transform the problem to the positive quadrant using simple geometric transformations and centre the data so that the data origin coincides with the origin of the spherical model.
2. Express the cone as a pair of *generating* vectors, $c_1 = [4, 11]$, $c_2 = [7, 4]$.
3. Use `4ti2` to compute the Hilbert Basis of the cone : $\{(1, 1), (1, 2), (2, 5), (3, 8), (4, 11), (3, 2), (5, 3), (7, 4)\}$.
4. Use the Hilbert Basis to generate all points on the integer grid that lie inside the cone and identify the feasible point furthest from the origin as a boundary point.

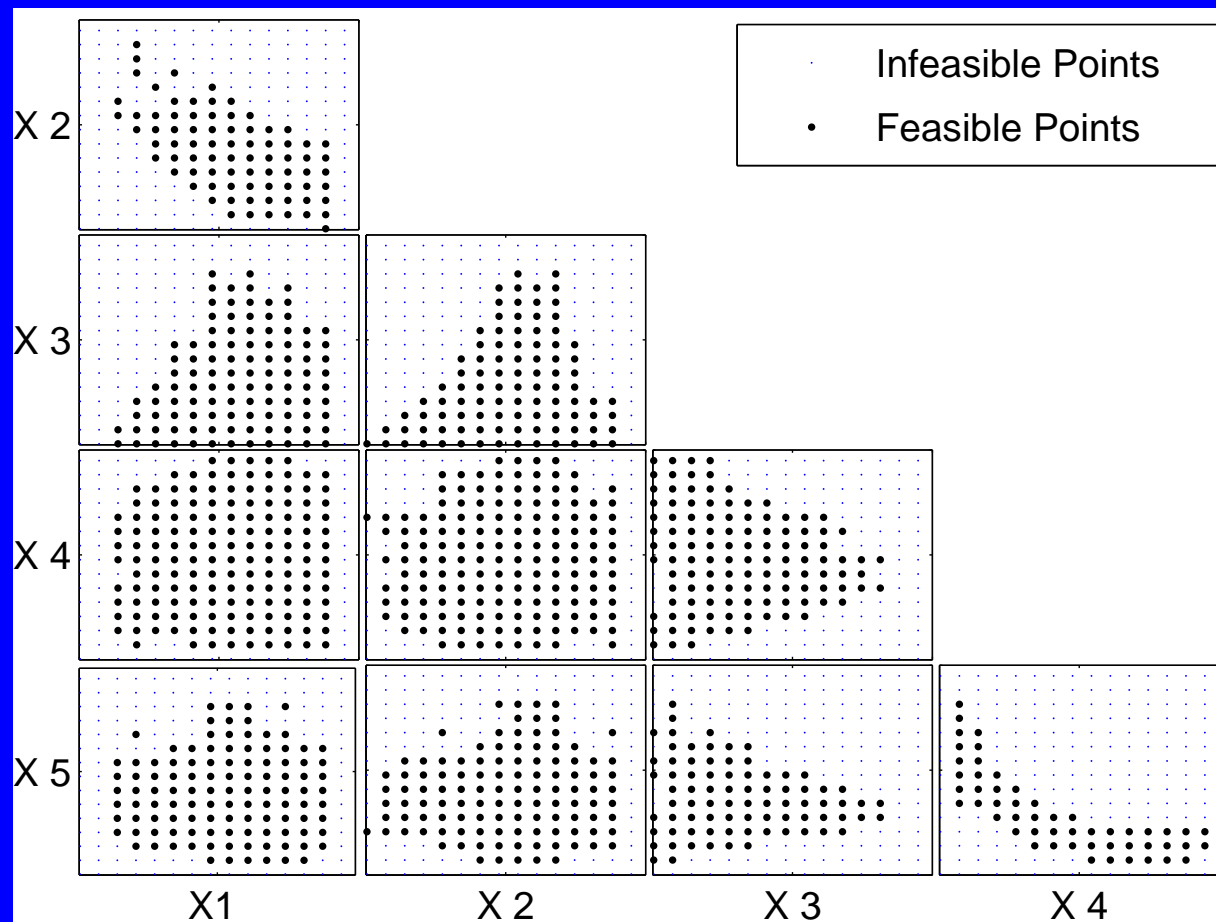
Example using Hilbert Bases

HB : $\{(1, 1), (1, 2), (2, 5), (3, 8), (4, 11), (3, 2), (5, 3), (7, 4)\}$.



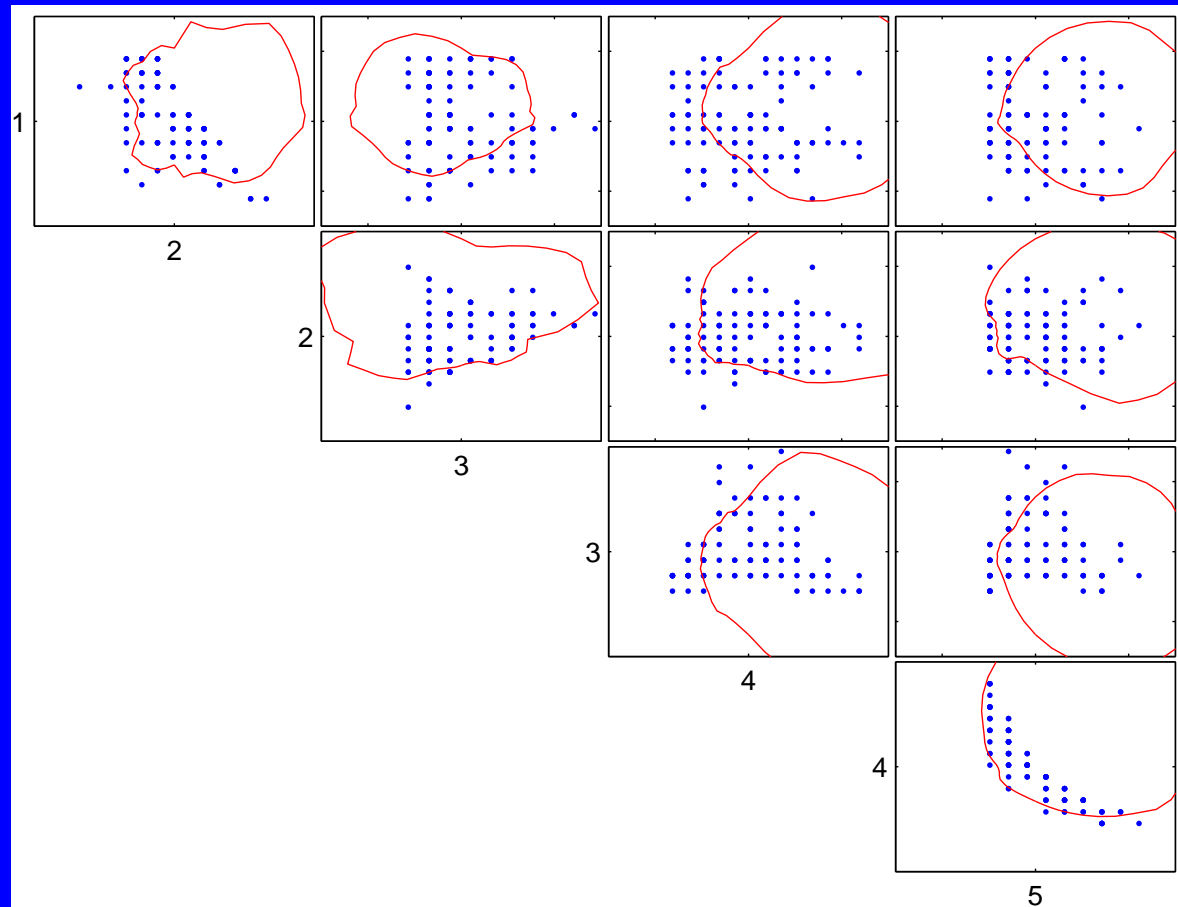
Chemical Engineering Simulation Example

- A 15 level Full Factorial design was used to search for an optimal solution. This involved $15^5 = 759,375$ simulations. There were 2955 feasible points (0.39%).



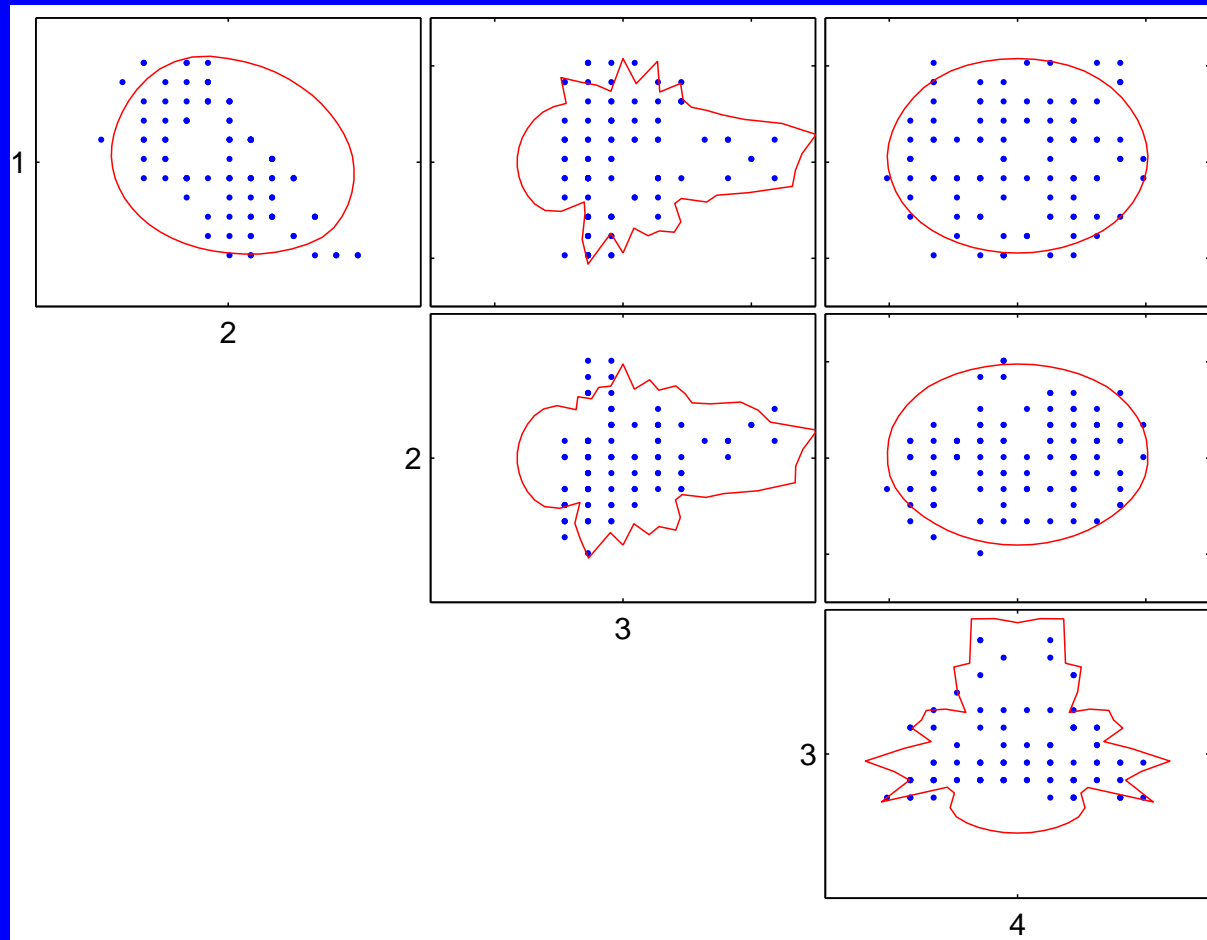
Model Fitting using Boundary Points

🔴 (a) Full Data.



Model Fitting using Boundary Points

■ (b) Subset with X_5 removed.



Conclusions

- Non-convex feasible regions can be estimated.
- Region must be single, continuous & star-shaped.
- Example uses classical Full Factorial designs, can be adapted to work with other more efficient designs.
- The choice of the origin is important.
- Need to identify boundary with limited number of points.
- Method can be used to select a subset of boundary points for modelling.
- A high level of correlation between factors (variables) can drastically reduce the volume of the feasible region. Principal Component Analysis, for example, may therefore be required.