Local Parametric Optimisation Algorithm Summary

Fabio Greenwood

06/08/2018

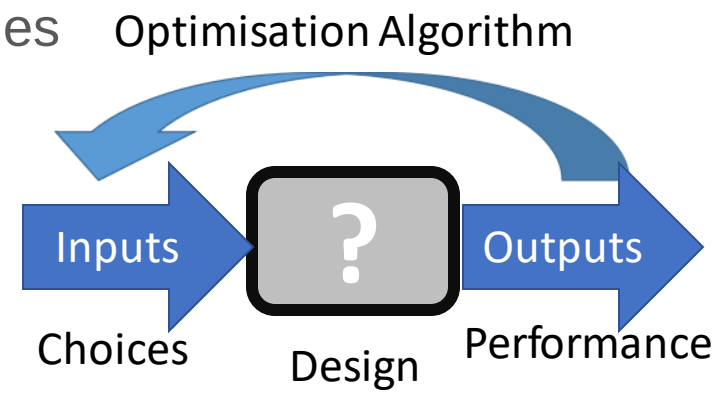
Parametric Optimisation Introduction

The accompanying script is an parameteric optimisation algorithm. These algorithms effectively are able to tune design parameters (I.e. length of section, width of beam), analyse the adjusted design and based on the result make a choice on a new set of inputs (referred to as an iteration) to improve the design

This process is repeated automatically many times to try and get the "best" possible design with minimal input from the engineer. This type of process has many other benefits such as data analysis and design exploration

This document explores how this method works and also briefly discusses information such as: strengths, weaknesses, potential improvements and application however the main propose of this document is to allow a non-engineer to better understand the attached code

In the adjoining folder there is a .exe file that will allow you to run the algorithm for yourself

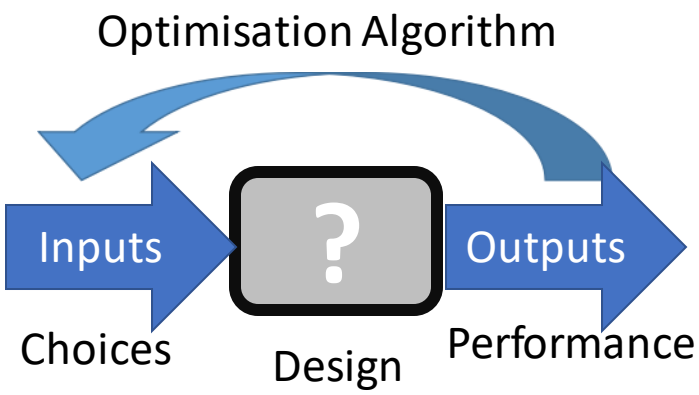


Parametric Optimisation Project

The accompanying script is an parameteric optimisation algorithm. Within the script "function1" represents the external CAE simulation software. The arguments passed into it represent different the parameters of different attempted designs and the output of the function a single performance parameter to be optimised

In a real application these arguments will have to be fed into simulation input files, which can then be called into a simulation software with command line orders.

The important part of this script can't see inside this function, only the inputs/outputs and must attempt to optimise this imaginary design within the simulation software



Algorithm Specifics

This is a local optimisation algorithm working by only considering a finite localised group of designs in the design space at time

The function will select a new design based on the current group

The function will discard the lowest design in the group when it evaluates a new design

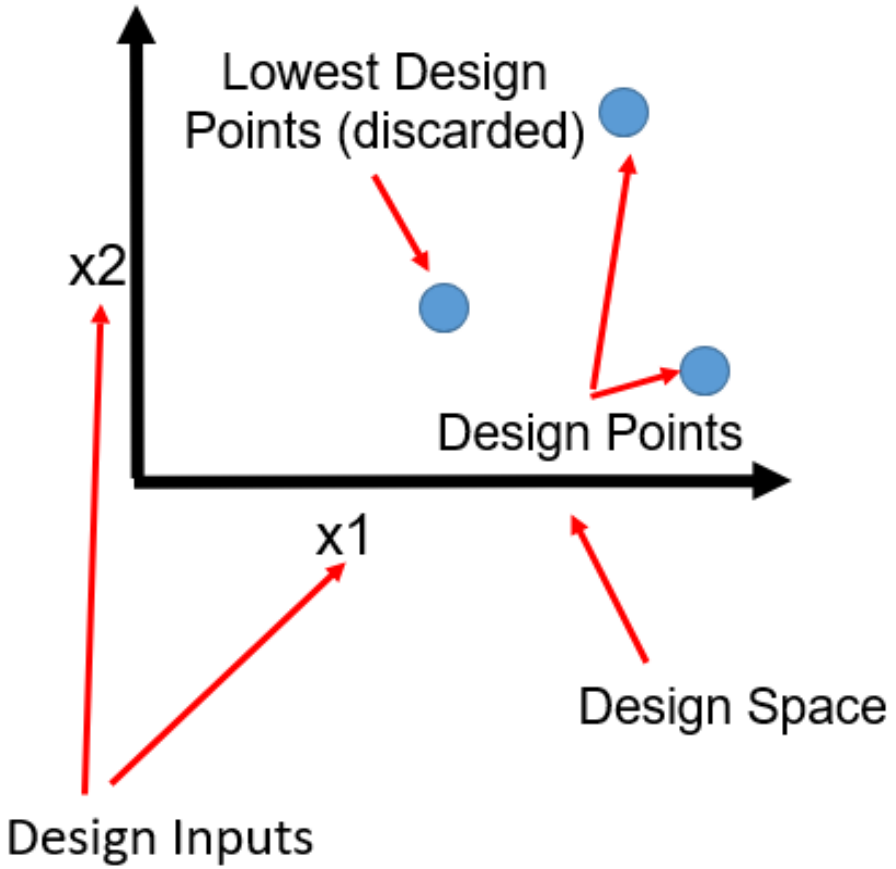
May reach a false/local maxima I.e. not the global maximum across the design space

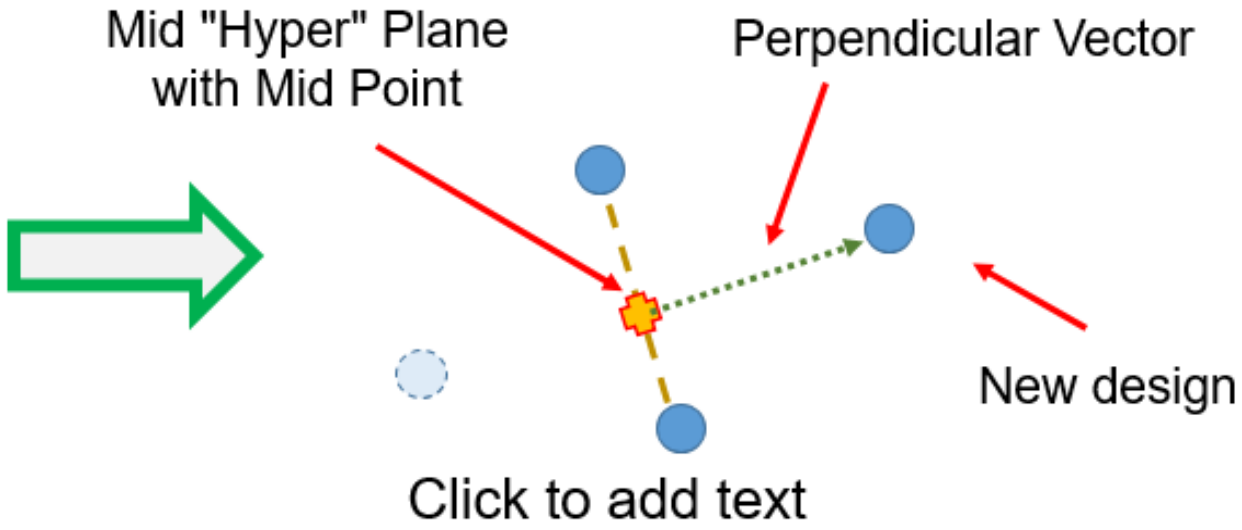
The group of design considered is also referred to as "the hand"

The process is explained for 2 input parameters but can be applied to problems with more inputs

The hand will contain one more design then the number of input parameters

Process Diagram





PTO for accompanying description

Detailed Process

1. The lowest design in the hand/group is moved from the hand/group to a separate container
2. The mid-point in the design space of the remaining designs/points is found
3. The vector normal to a plane (or "hyperplane") co-incident (within the design space) with the all remaining points is found, referred to here as the “Progress Vector”
   - The direction of this vector may have to be reversed, to ensure that it is pointing away for the lowest design from the mid-point. This is checked and done if needed
   - This vector is normalised (length set to 1)
4. The new progress vector is compared to the old progress vector, if the directional difference is more than 90 deg, then the “Step Length” variable will be reduced, if not it is increased
5. The new point is generated by starting from the “mid-point” and progressing the distance of the step length along the progress vector

Additional Notes

Please note some of the below terminology is industry specific

1. Local optimisation, not a global solution
2. Mono-objective
3. Currently not self terminating, user must click to advance algorithm
4. Deterministic (depending on DoE)
5. Change the value "numOParas" to change the number of input parameters, please note that UI will crash if less then 4 are set and will only ever display the first four inputs
6. Has not been heavy tested performance-wise for problems with a large number of input parameters (8+)
7. If used in a real world application please change the algorithm to begin in the centre of the design space (inputs=0) for best performance
8. Algorithm works when the design space bounds are between –1 to 1, any parameters passed to and from it should be are scaled so the algorithm continues to work in these bounds
   - e.g. if your real input parameter can range between 1 and 2 apply the function f(x)=(x+3)/2 to any design input requested by the algorithm
9. Optimal point for optimised function is when all inputs are at 0

Potential for improvement

1. Potential to optimise/improve step length adjustment technique and parameters and also to adjust the initial step length itself
2. Algorithm to create .txt or .csv log of designs/debugging
3. Better charts: parallel axis, multi-history, design tables
4. Intergration to external software

Author Details

Fabio Greenwood

Fabio.greenwood@gmail.com

0787 040 1486

BEng Aerospace Engineering (Hons)

Brunel University