Mining Markov Network Surrogates for Value-Added Optimisation

Alexander Brownlee

www.cs.stir.ac.uk/~sbr
sbr@cs.stir.ac.uk

Outline

- Value-added optimisation
- Markov network fitness model
- Mining the model
- Examples with benchmarks
- Case study: cellular windows
- Discussion / conclusions

Value-added Optimisation

- A philosophy whereby we provide more than simply optimal solutions
- Information gained during optimisation can highlight sensitivities and linkage
- This can be useful to the decision maker:
  - Confidence in the optimality of results
  - Aids decision making
  - Insights into the problem
    - Help solve similar problems
    - Highlight problems / misconceptions in definition

Value-added Optimisation

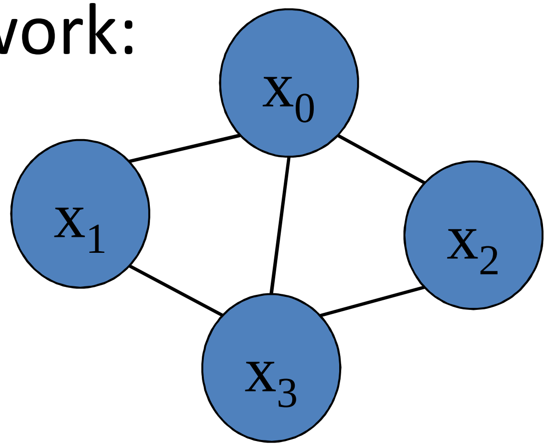
- This information can come from
  - the trajectory followed by the algorithm
  - models built during the run
- If we are constructing a model as part of the optimisation process, anything we can learn from it comes "for free"
- Some examples from MBEAs / EDAs
  - M. Hauschild, M. Pelikan, K. Sastry, and C. Lima. Analyzing probabilistic models in hierarchical BOA. IEEE TEC 13(6):1199-1217, December 2009
  - R. Santana, C. Bielza, J. A. Lozano, and Pedro Larranaga. Mining probabilistic models learned by EDAs in the optimization of multi-objective problems. In Proc. GECCO 2009, pp 445-452

Markov network fitness model (MFM)

- Suited to bit string encoded problems
- Originally developed as part of DEUM EDA
  - A probabilistic model of fitness, directly sampled to generate solutions, replacing crossover and mutation operators
- Markov network is undirected probabilistic graphical model
  - energy U(x) of a solution x equates to a sum of clique potentials, in turn equates to a mass distribution of fitness
  - energy has negative log relationship to probability, so minimise U to maximise f
- MFM can be used as a surrogate

FM with Markov Networks

- Two aspects to building a Markov network:
  - Structure
  - Parameters (α)

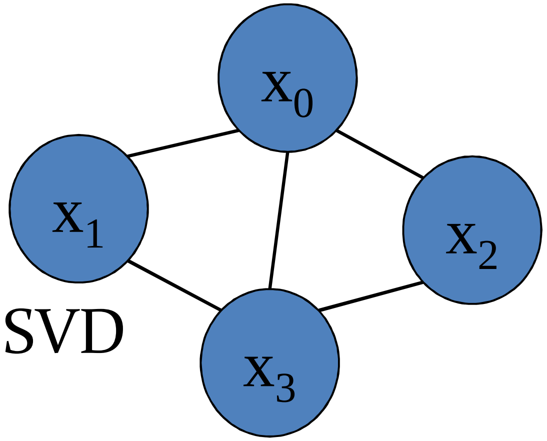


- Model can be represented by:

$$\begin{aligned} &\alpha_0 x_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 \\ &+ \alpha_{01} x_0 x_1 + \alpha_{02} x_0 x_2 + \alpha_{03} x_0 x_3 + \alpha_{13} x_1 x_3 + \alpha_{23} x_2 x_3 = -\ln(f(x)) \\ &+ \alpha_{013} x_0 x_1 x_3 + \alpha_{023} x_0 x_2 x_3 + c \end{aligned}$$

  - Compute parameters using sample of population
  - Variables are -1 and +1 instead of 0 and 1
- The terms in the MFM correspond to Walsh functions (can represent any bit string encoded problem)

Building a Model



Calc Markov network parameters using SVD

1011 f=1

$$(1)\alpha_0 + (-1)\alpha_1 + (1)\alpha_2 + (1)\alpha_3 + (1)(-1)\alpha_{01} + (1)(1)\alpha_{02} + (1)(1)\alpha_{03} + (-1)(1)\alpha_{13} + (1)(1)\alpha_{23} + (1)(-1)(1)\alpha_{013} + (1)(1)(1)\alpha_{023} + c = -\ln(1)$$

1111 f=4

$$(1)\alpha_0 + (1)\alpha_1 + (1)\alpha_2 + (1)\alpha_3 + (1)(1)\alpha_{01} + (1)(1)\alpha_{02} + (1)(1)\alpha_{03} + (1)(1)\alpha_{13} + (1)(1)\alpha_{23} + (1)(1)(1)\alpha_{013} + (1)(1)(1)\alpha_{023} + c = -\ln(4)$$

1001 f=1

$$(1)\alpha_0 + (-1)\alpha_1 + (-1)\alpha_2 + (1)\alpha_3 + (1)(-1)\alpha_{01} + (1)(-1)\alpha_{02} + (1)(1)\alpha_{03} + (-1)(1)\alpha_{13} + (-1)(1)\alpha_{23} + (1)(-1)(1)\alpha_{013} + (1)(-1)(1)\alpha_{023} + c = -\ln(1)$$

1000 f=3

$$(1)\alpha_0 + (-1)\alpha_1 + (-1)\alpha_2 + (-1)\alpha_3 + (1)(-1)\alpha_{01} + (1)(-1)\alpha_{02} + (1)(-1)\alpha_{03} + (-1)(-1)\alpha_{13} + (-1)(-1)\alpha_{23} + (1)(-1)(-1)\alpha_{013} + (1)(-1)(-1)\alpha_{023} + c = -\ln(3)$$

0011 f=2

$$(-1)\alpha_0 + (-1)\alpha_1 + (1)\alpha_2 + (1)\alpha_3 + (-1)(-1)\alpha_{01} + (-1)(1)\alpha_{02} + (-1)(1)\alpha_{03} + (-1)(1)\alpha_{13} + (1)(1)\alpha_{23} + (-1)(-1)(1)\alpha_{013} + (-1)(1)(1)\alpha_{023} + c = -\ln(2)$$

α_0=-0.38	α_1=0.16	α_2=0.02	α_3=-0.34
α_{01}=-0.07	α_{02}=0.25	α_{03}=-0.11	α_{13}=-0.11
α_{23}=-0.25	α_{013}=-0.34	α_{023}=-0.02	c=-0.61

MFM Predicts Fitness

- Example; for individual X={1011}
- Substitute variable values into energy function and solve:

$$U(x) = \alpha_0 - \alpha_1 + \alpha_2 + \alpha_3 - \alpha_{01} + \alpha_{02} + \alpha_{03} - \alpha_{13} + \alpha_{23} - \alpha_{013} + \alpha_{023} + c$$

$$f(x) = e^{-U(x)}$$

- This can then be used to predict fitness as a surrogate

MFM as a surrogate

- Can either
  - completely replace fitness function (GA essentially samples the MFM)
  - take a mixed approach, where MFM is retrained occasionally, and used to filter candidate solutions
- e.g. Speeding up benchmark FFs
  - A. Brownlee, O. Regnier-Coudert, J. McCall, and S. Massie. Using a Markov network as a surrogate fitness function in a genetic algorithm. Proc. IEEE CEC 2010, pp. 4525-4532
- e.g. Speeding up feature selection
  - A. Brownlee, O. Regnier-Coudert, J. McCall, S. Massie, and S. Stulajter. An application of a GA with Markov network surrogate to feature selection. International Journal of Systems Science, 44(11):2039-2056, 2013.
- Now we consider how the model might be mined

Mining the model (1)

$$-\ln(f(x)) = U(x)/T$$

- As we minimise energy, we maximise fitness. So to minimise energy:

$$\alpha_i x_i$$

- If the value taken by x_i is 1 (+1) in high-fitness solutions, then a_i will be negative
- If the value taken by x_i is 0 (-1) in the high-fitness solutions, then a_i will be positive
- If no particular value is taken by x_i optimal solutions, then a_i will be near zero

Mining the model (2)

$$-\ln(f(x)) = U(x)/T$$

- As we minimise energy, we maximise fitness. So to minimise energy:

$$\alpha_{ij} x_i x_j$$

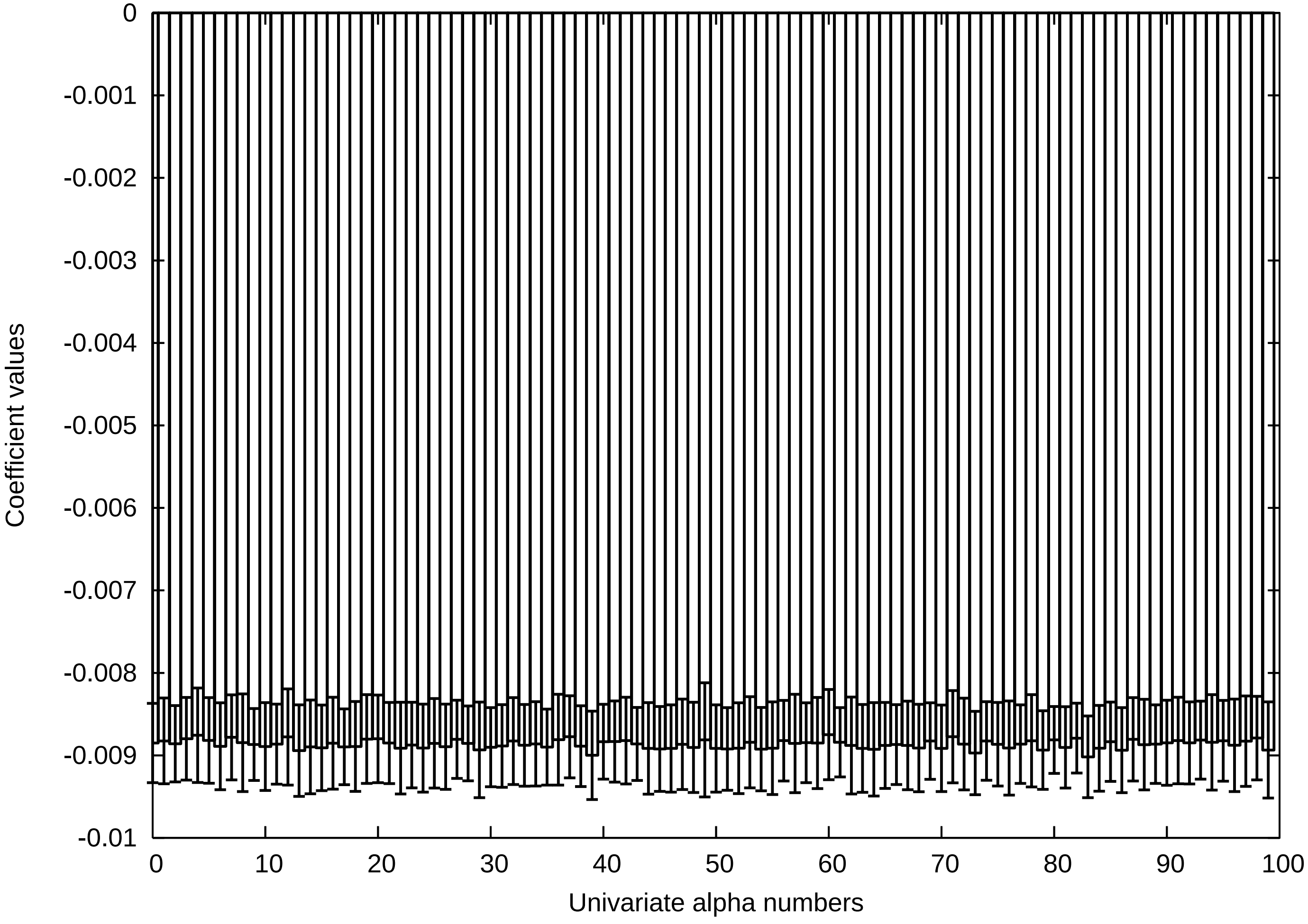
- If the values taken by x_i and x_j are equal (+1) in the optimal solutions, then a_i will be negative
- If the values taken by x_i and x_j are opposite (-1) in the optimal solutions, then a_{ij} will be positive
- Higher order interactions follow this pattern

Examples with Benchmarks

- A few well-known benchmarks to get the idea
- In these experiments, the MFM replaces FF
- Solutions generated at random and used to train model parameters

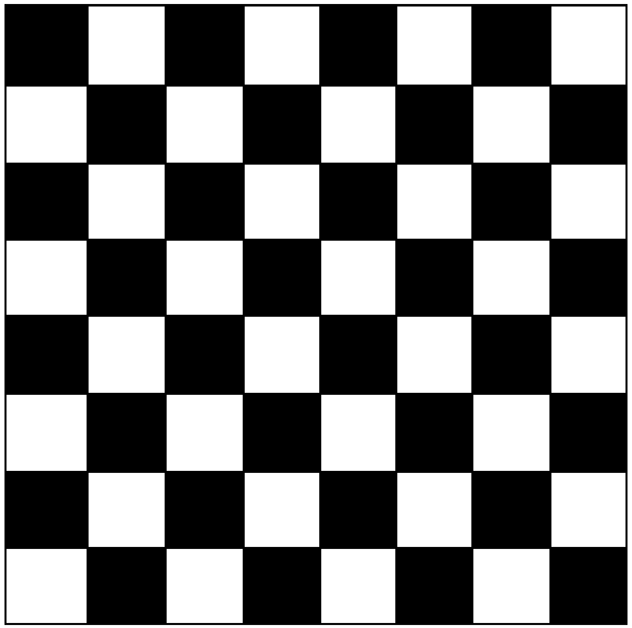
Onemax

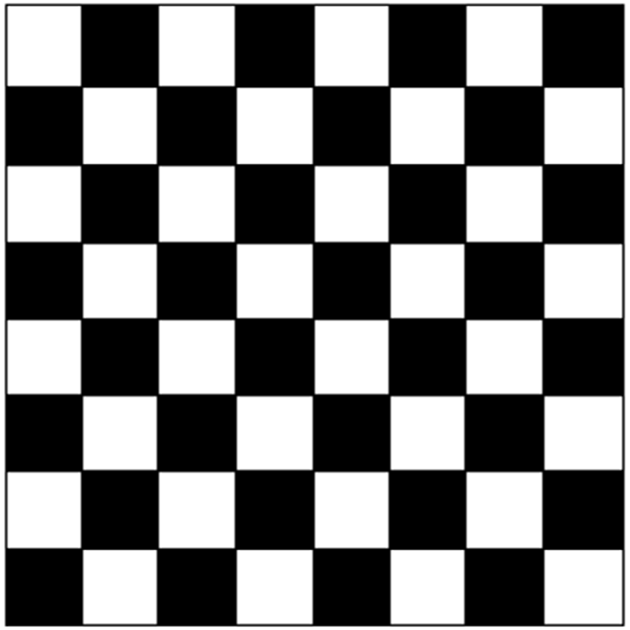
- Fitness is the sum of x_i set to 1



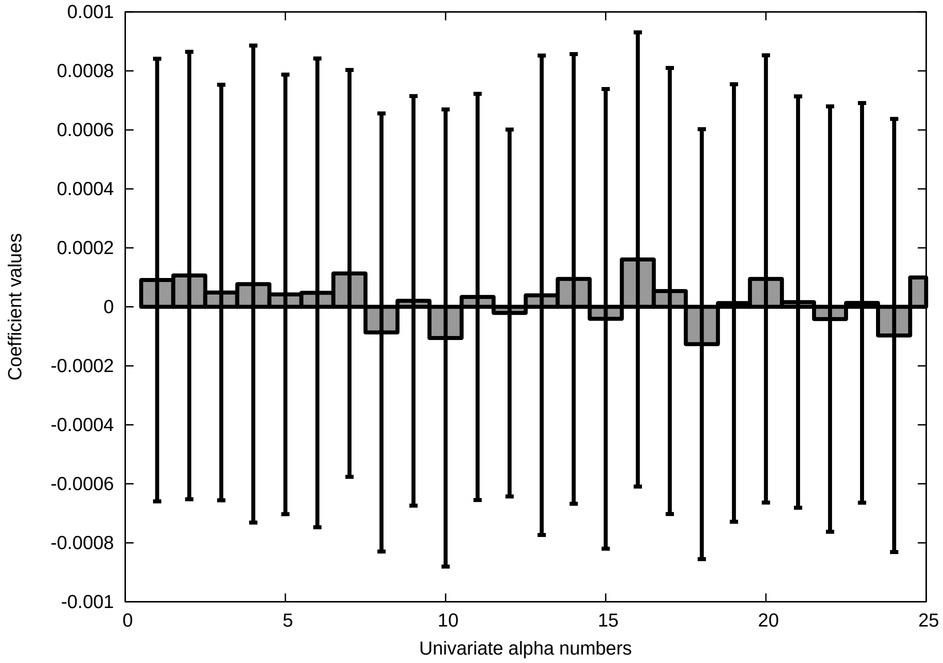
Checkerboard 2D

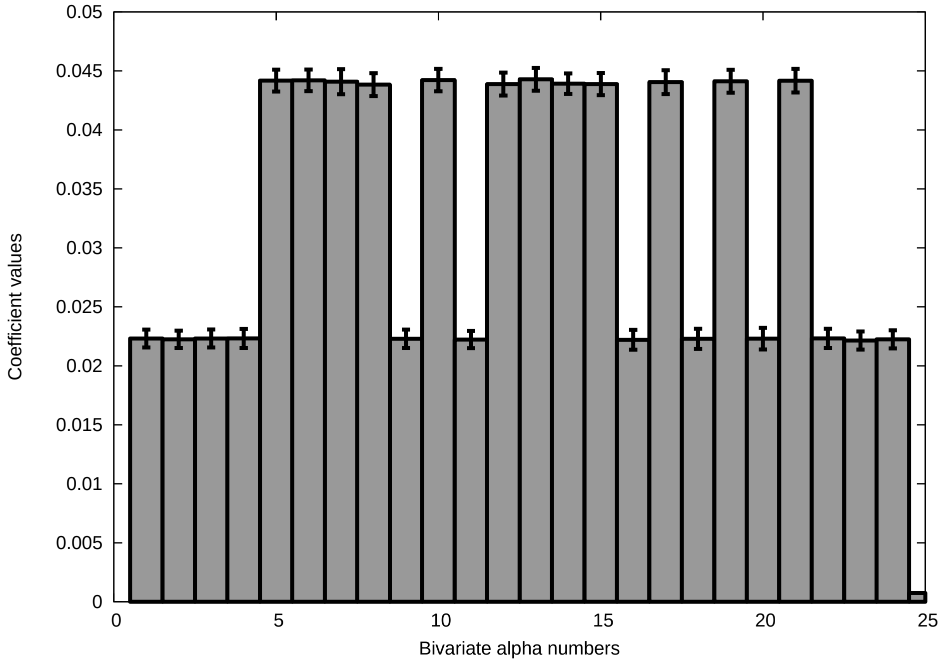
- Form an s x s grid of the x_i: fitness is the count of neighbouring x_i taking opposite values



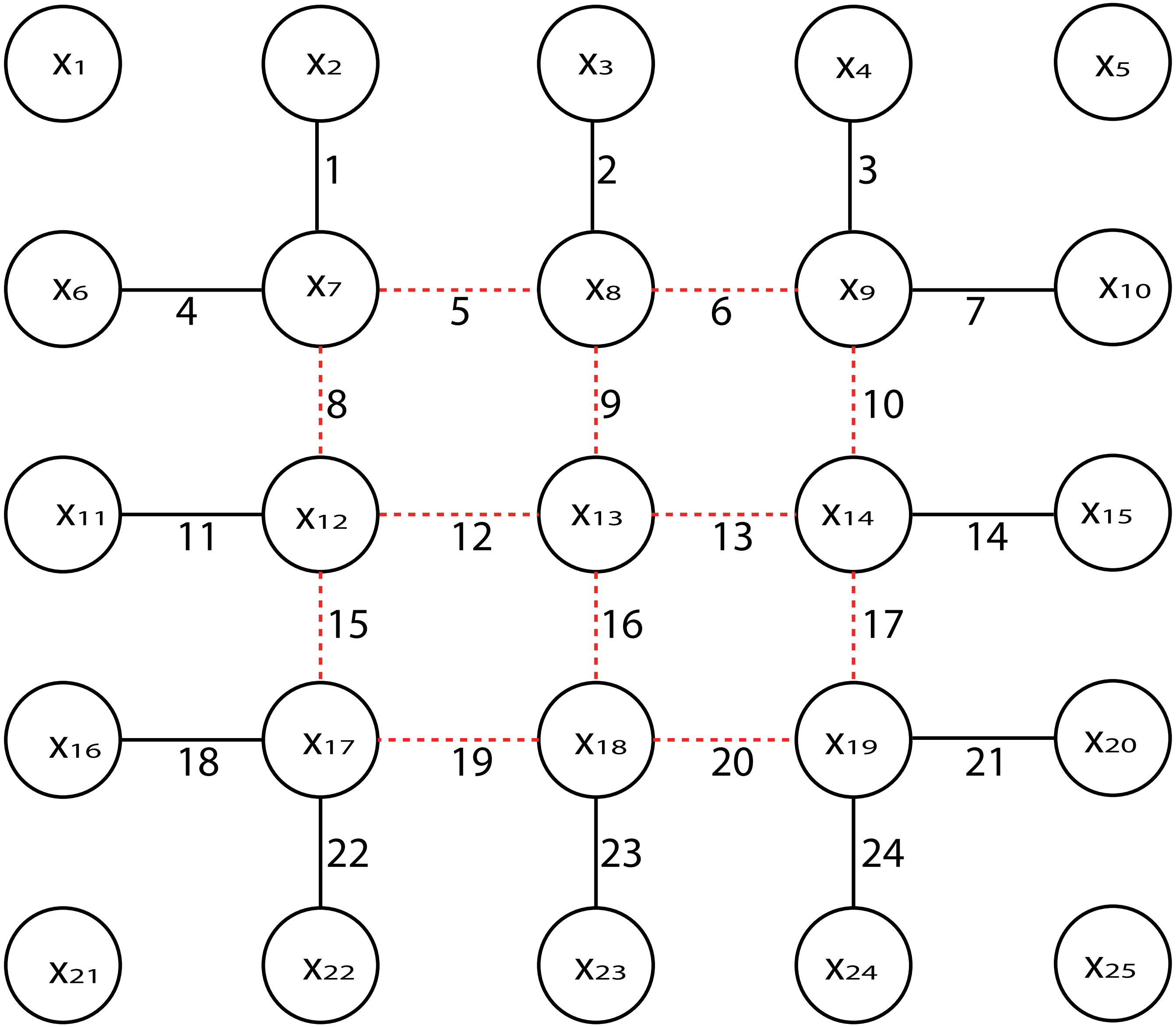


Checkerboard 2D



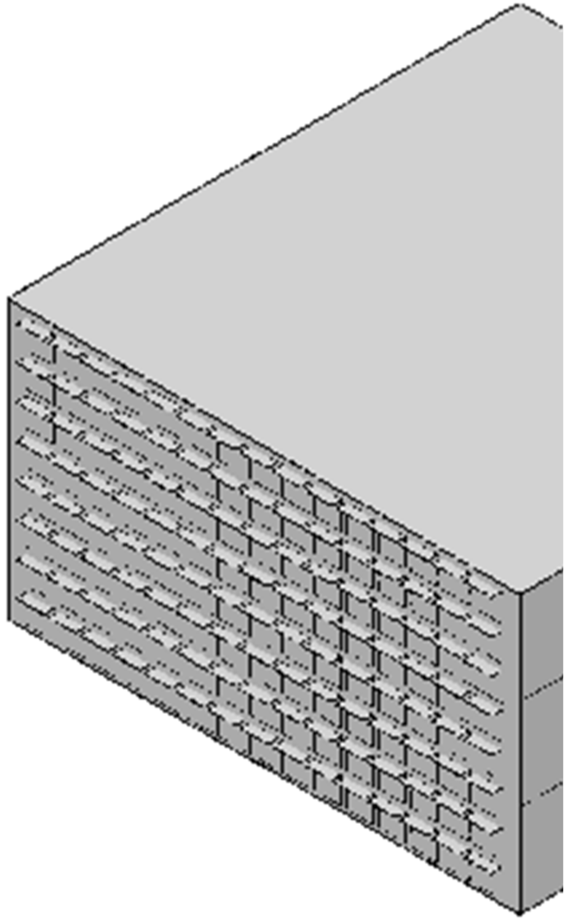


Checkerboard 2D



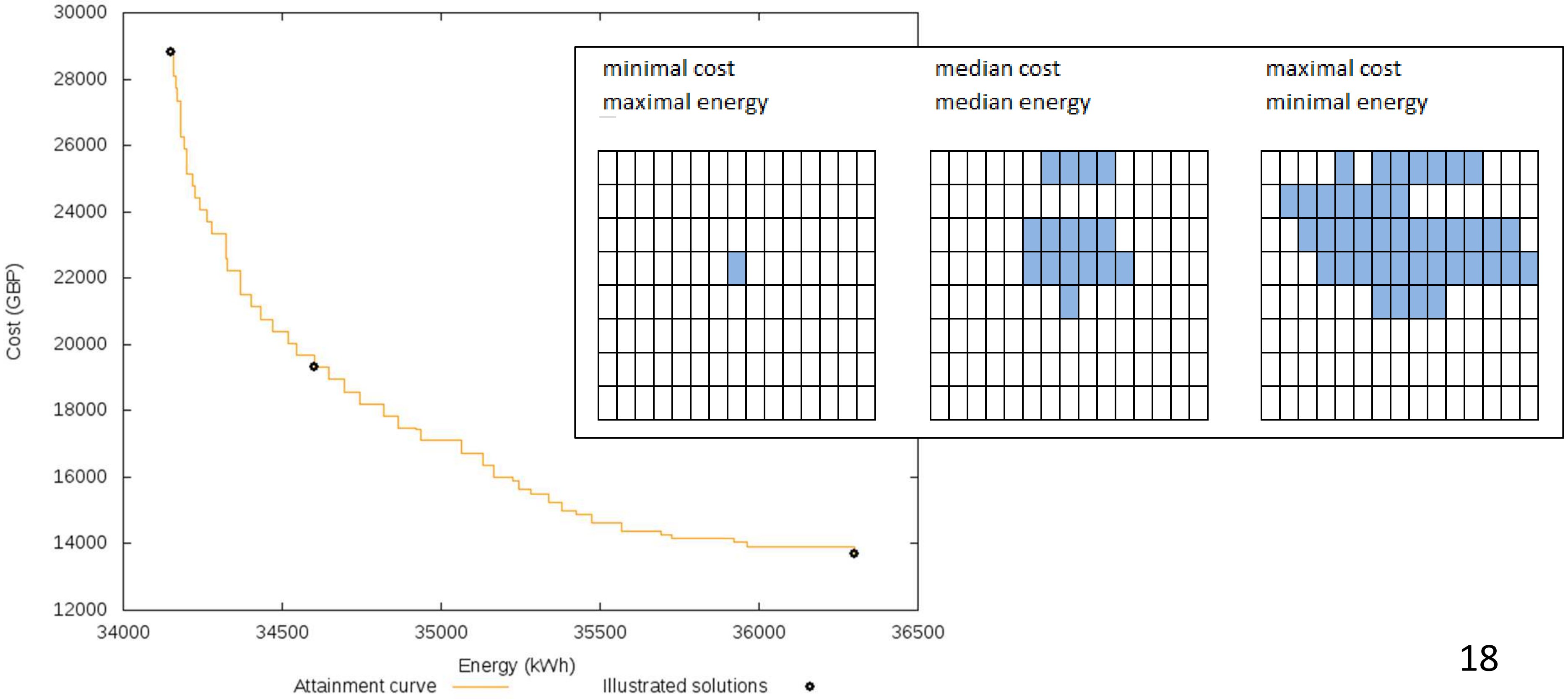
RW Example: Cellular Windows

- Optimise glazing for an atrium in a building
- Switch on glazing in 120 cells
  - 120 bits encoding
- Minimise energy use and construction cost
  - Energy for lighting, heating and cooling
  - Costly to compute: motivating use of surrogate



Optimisation run

- Optimisation run used NSGA-II to find approximated Pareto-optimal solutions



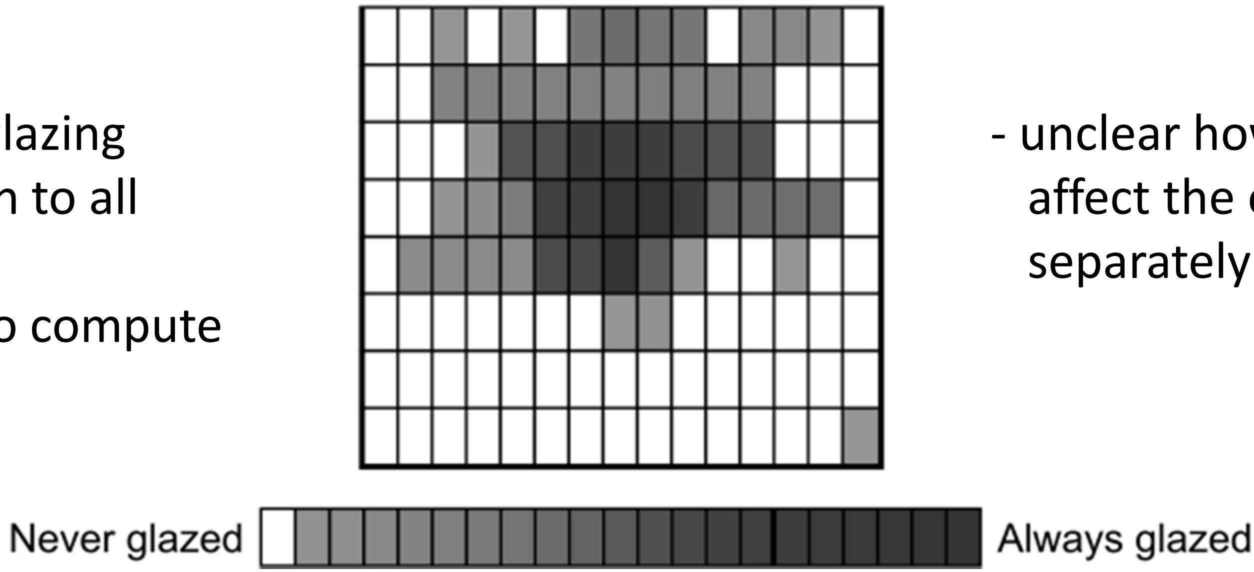
Optimisation run

- Trade-off and the specific designs in it are already helpful for a decision maker
- But:
  - Lowest cost solution missing due to randomness
  - Slightly odd window shapes
- What might be the impact of aesthetic changes to these solutions?

Adding value

- Earlier paper tried two approaches
- Frequency that cells are glazed in the approximated Pareto optimal sets

+ shows glazing common to all optima
+ cheap to compute



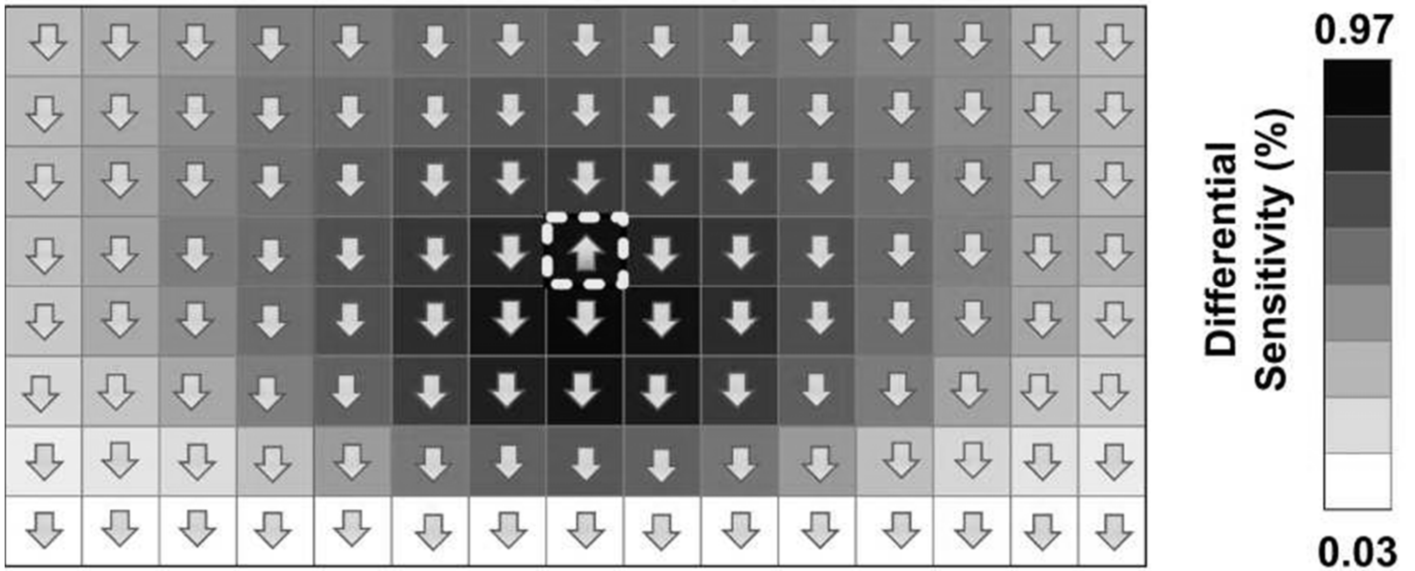
- unclear how cells affect the objectives separately

Adding value

- Local sensitivity – Hamming-1 neighbourhood of approx. Pareto optimal solutions

+ shows possible local improvements
+ shows impact on objectives separately

- needs further fitness evaluations

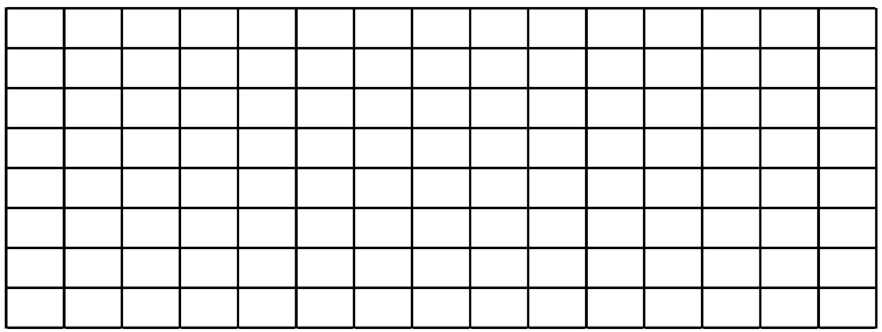


Adding value

- Both of these approaches are useful, but could be supplemented…
- A surrogate could be mined to discover similar or additional insights into the problem
- Here, as a proof of concept, we train the MFM using solutions from the NSGA-II run, allowing for direct comparisons with the existing work
- Applies to energy and cost objectives for demonstration, though cost is cheap and probably doesn't need a surrogate in practice
- (no solutions passed back to algorithm at present)

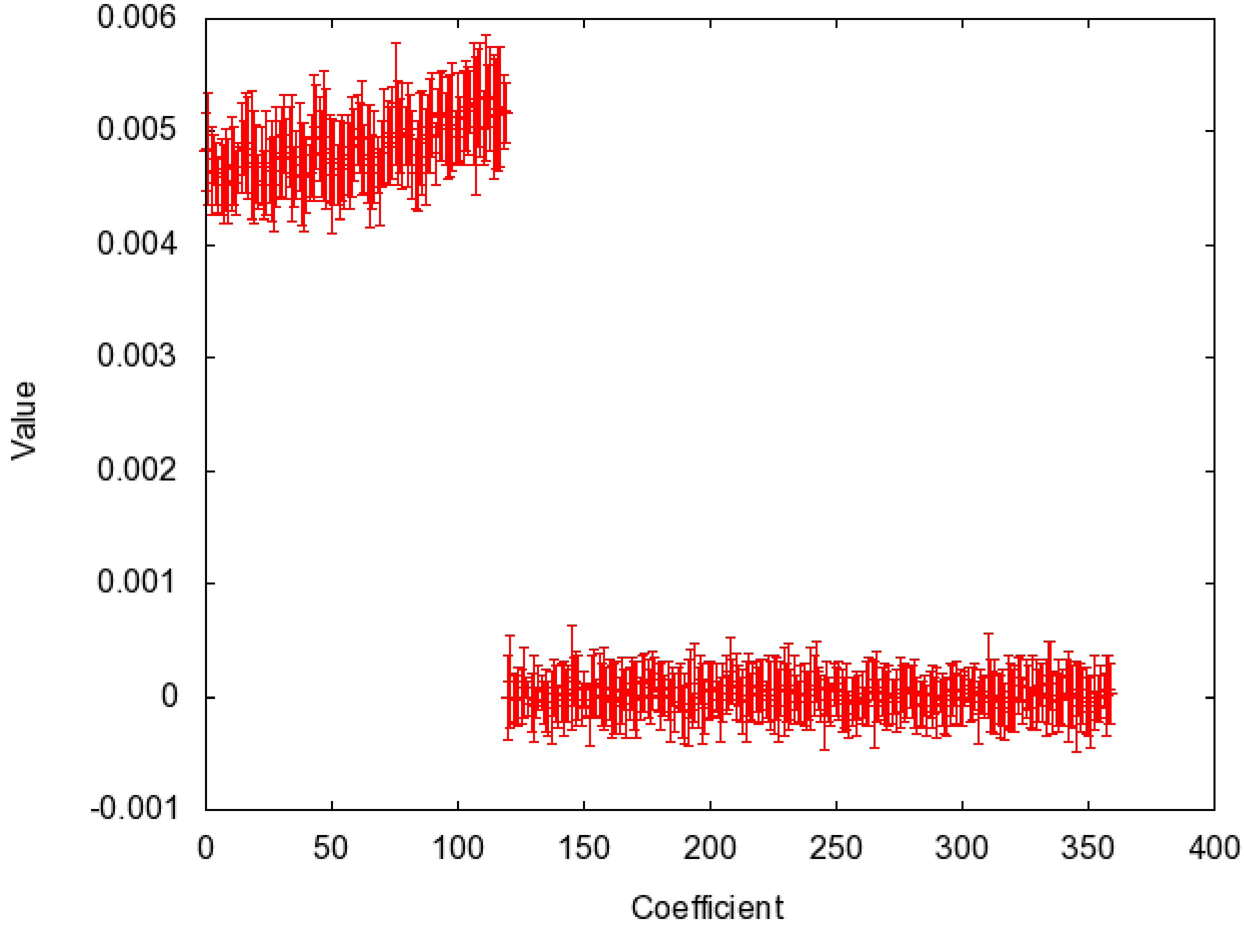
Lattice model structure

- Initial experiments used MFM with a lattice structure
  - One $a_i x_i$ term for each cell
  - One $a_{ij} x_i x_j$ for each pair of neighbouring cells in grid
- 400 highest fitness solutions from first 1000 used to train model



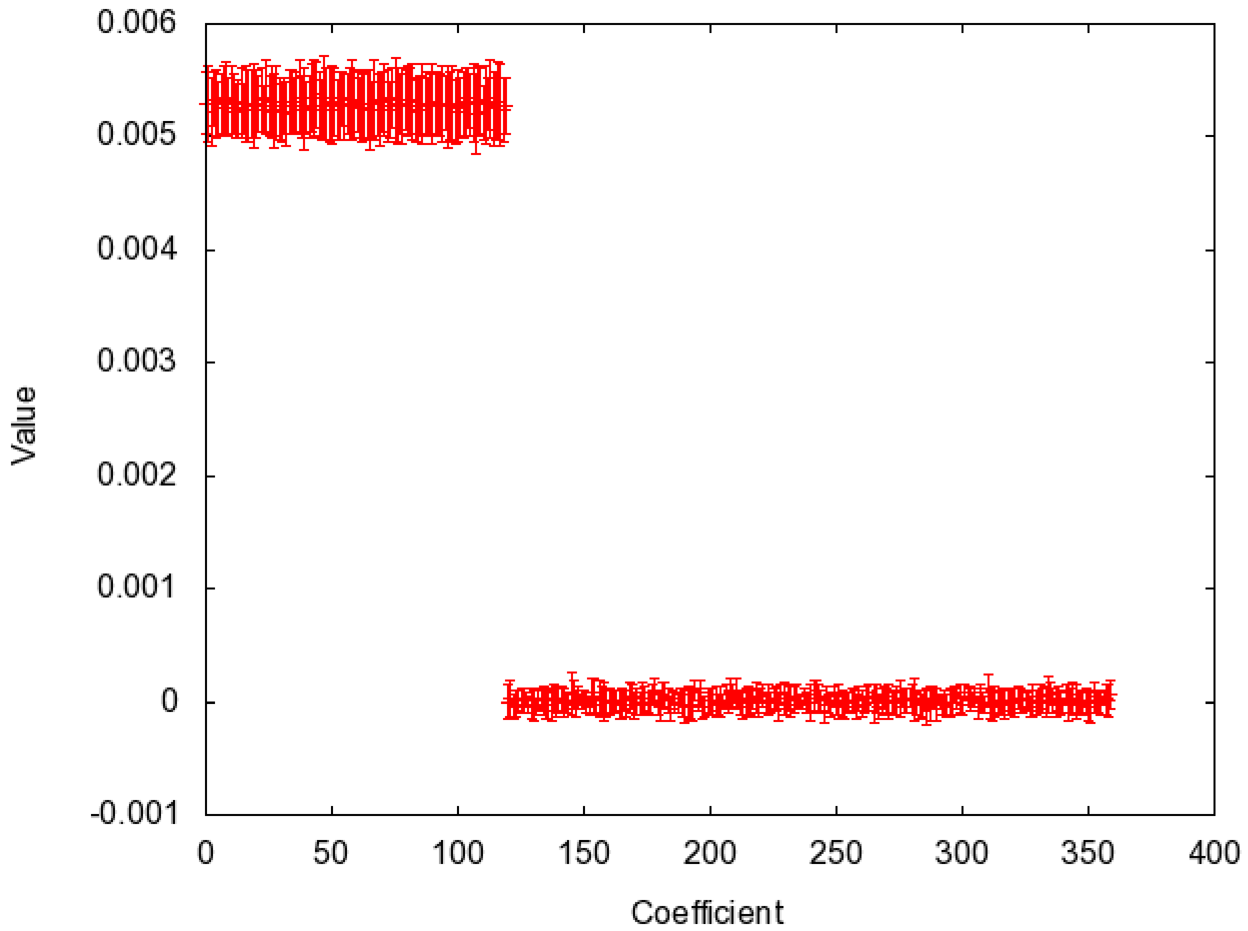
Lattice model structure

- Energy



Lattice model structure

- Cost



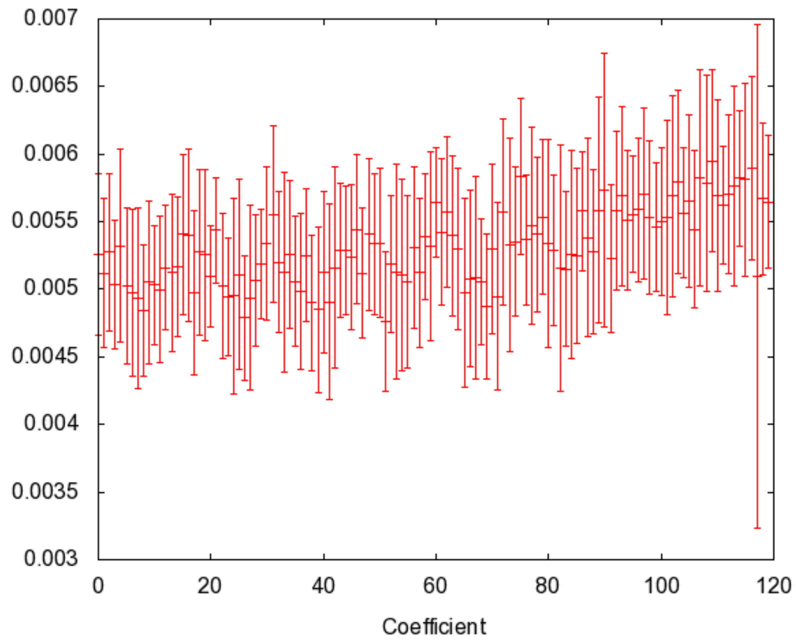
Univariate structure

- Bivariate terms have no impact on objectives (no linkage) so tried univariate structure
  - One $a_i x_i$ term for each cell
- 140 highest fitness solutions from first 400 used to train model

Univariate model structure

- Energy

0.0053	0.0051	0.0053	0.005	0.0053	0.005	0.005	0.0049	0.0048	0.0051	0.005	0.005	0.0052	0.0051
0.0054	0.0054	0.005	0.0053	0.0053	0.0051	0.0054	0.005	0.0049	0.0049	0.0051	0.0048	0.0049	0.0051
0.0053	0.0056	0.0052	0.0051	0.0053	0.0051	0.005	0.0053	0.0049	0.0048	0.0051	0.0049	0.0052	0.0053
0.0052	0.0054	0.0051	0.0054	0.0053	0.0053	0.0048	0.0052	0.0051	0.0051	0.0051	0.0053	0.0051	0.0054
0.0056	0.0054	0.0056	0.0054	0.0053	0.005	0.0051	0.0051	0.0051	0.0049	0.0053	0.0049	0.0056	0.0053
0.0058	0.0054	0.0055	0.0054	0.0055	0.0053	0.0053	0.0052	0.0052	0.0053	0.0052	0.0056	0.0054	0.0053
0.0057	0.0052	0.0056	0.0057	0.0055	0.0056	0.0056	0.0057	0.0055	0.0055	0.0055	0.0055	0.0057	0.0058
0.0057	0.0054	0.0058	0.0058	0.0059	0.0057	0.0056	0.0057	0.0058	0.0058	0.0058	0.0059	0.0057	0.0057

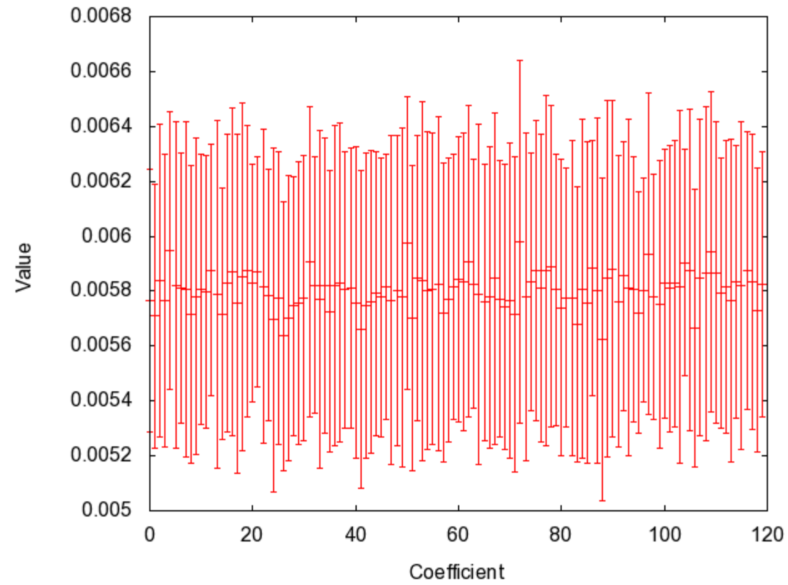


- Bias towards the lower and outer edges
- Cells in these regions shouldn't be glazed
- Matches patterns seen in PF and local sensitivity analysis

Univariate model structure

- Cost

0.0058	0.0057	0.0058	0.0058	0.0059	0.0058	0.0058	0.0058	0.0057	0.0058	0.0058	0.0058	0.0059	0.0058
0.0058	0.0059	0.0058	0.0059	0.0059	0.0058	0.0059	0.0058	0.0058	0.0057	0.0058	0.0056	0.0057	0.0057
0.0058	0.0059	0.0058	0.0058	0.0058	0.0057	0.0058	0.0058	0.0058	0.0058	0.0058	0.0057	0.0057	0.0058
0.0058	0.0058	0.0058	0.0058	0.0058	0.0060	0.0057	0.0058	0.0058	0.0058	0.0058	0.0058	0.0057	0.0058
0.0058	0.0058	0.0059	0.0058	0.0058	0.0058	0.0058	0.0058	0.0058	0.0057	0.0058	0.0057	0.0060	0.0058
0.0059	0.0058	0.0059	0.0059	0.0058	0.0057	0.0058	0.0058	0.0057	0.0058	0.0058	0.0059	0.0058	0.0056
0.0059	0.0058	0.0059	0.0058	0.0058	0.0057	0.0058	0.0059	0.0058	0.0058	0.0058	0.0058	0.0058	0.0058
0.0059	0.0057	0.0058	0.0059	0.0059	0.0059	0.0058	0.0058	0.0058	0.0058	0.0058	0.0059	0.0058	0.0057



- Values similar: cells have equal impact
- All positive: minimum cost solution is all unglazed

Benefits

- Information comes without running additional fitness evaluations (in fact with a time saving, if use of surrogate speeds up run)
- Sensitivities linked explicitly to objectives (compared to analysis of PF)
- Analysis rooted in multiple generations of run, not just final one

Value Added

- Could visualise the model as optimisation proceeds, as extra feedback, or as part of the final results
- Knowing the sensitive variables, we can adjust the solutions for factors not considered by the optimisation (e.g. aesthetics), aware of likely impact on optimality
  - e.g fixing odd window shapes
- model may indicate where a metaheuristic has not fully converged on the global optimum

Value Added

- If solutions match the model's suggestions, we can be more confident that they are optimal
- Counter-intuitive results can highlight errors in the model (perhaps the lack of linkage means that the model doesn't consider neighbouring glazing properly?)
- Model may suggest good solutions long before the EA has found them

Conclusions

- If we have a model, it can be worth seeing if it contains useful information
- MFM used as a surrogate fitness function
- Mined the model for additional information about the problem to "add value" to the optimisation run
- How might MFM be extended to other representations?
- Can we adopt the mining approach for other model types?