Paper 76

©Civil-Comp Press, 2012
Proceedings of the Eighth International Conference
on Engineering Computational Technology,
B.H.V. Topping, (Editor),
Civil-Comp Press, Stirlingshire, Scotland

A Method for Maximin Constrained Design of Experiments

E. Myšáková, M. Lepš and A. Kučerová
Department of Mechanics, Faculty of Civil Engineering
Czech Technical University in Prague, Czech Republic

Abstract

This paper presents a new approach for generating a design of experiments in constrained and non-regular spaces. The methodology is based on the triangulation of the admissible space by Delaunay triangulation method. Then, a heuristic smoothing method for generating uniform finite element meshes within the triangulated space is applied to obtain uniformly spaced designs. Although not 100% reliable, the proposed method can produce superior designs compared with already known optimal solutions.

Keywords: design of experiments, mixture experiment, constrained design spaces, non-regular design spaces, space-filling, Delaunay triangulation.

1 Introduction

Space-filling design strategies known as a design of experiments (DoE) constitute an essential part of any experimentation. Our contribution is aimed at one particular domain of constrained design spaces. The most frequent example is the case of a mixture experiment, where individual inputs form a unity volume or unity weight [1, Chapter 11-5]. This only condition leads to the simplex space; further limits of individual inputs then form a polytope, still convex but generally irregular space. Therefore, all traditional DoEs [1] that are constructed for hypercube spaces cannot be applied here.

Although the problem is known for decades, the progress of methods for DoEs does not follow current development within the area of computer experiments [2]. The main difference between classical and modern DoEs is the number of samples where, for the latter, the hundreds of samples is a usual scenario. Then, the classical approaches based on fixed small-sample templates [3, 4] cannot be used. Up-to-date, the authors have found only few references on DoEs in constrained design spaces.

References [5] and [6] apply traditional latin hypercube (LH) designs to a bounding box followed by a genetic algorithm (GA) and hill-climbing optimization algorithms, respectively, to fulfil original constraints. Here, the LH methodology is merely used for minimization of the searched space than for nice properties of LH designs. Another approach is presented in [7], where interesting points are found by a GA and then, the final solution is located by sequential linear programming. The solution is in this case general, however, the computational demands are enormous.

In this paper a different approach based on Delaunay triangulation (DT) of an admissible domain and an utilization of nice properties of the `Distmesh` tool (DM) [8] is presented. Our results will be compared to seven constrained examples in two dimensions and one three dimensional example presented in [7], namely a placing of design points in a triangle, parallelogram, pentagon, hexagon, heptagon, octagon, irregular hexagon and prism, see Fig. 1 for optima presented in [7].

The paper is organized as follows. Section 2 describes three frequent objective functions that are used for comparison of space-filling designs and that will be used hereafter. For other measures and their comparison, see reference [9]. Next section is devoted to the short presentation of the methodology used in referencing paper [7] followed by our approach in Section 4. Finally, the fifth section presents comparison and analysis of obtained results.

2 Objective functions

Since we are interested in space-filling properties, three most common objective functions are examined. The first is Euclidean maximin metric (EMM) [10, 11] for its simplicity and easiness in visualization. The EMM is the minimal distance out of all distances between any two design points and is to be maximized:

$$E^{\mathrm{EMM}} = \min\{..., L_{ij}, ...\}, \quad i = 1, ..., n; \quad j = (i+1), ..., n \quad , \tag{1}$$

where n is the number of design points and L_{ij} is the Euclidean distance between points i and j. From the experiments point of view EMM expresses the worst case scenario of the closeness of two experiments. Even for computer experiments the assumption that an evaluation is costly is still valid. Therefore, the possible duplicity of two closed points remains a crucial task.

The second measure is the Audze-Eglais objective function (AE) proposed in [12]. It is based on an analogy with a potential energy of the set of points. The points are distributed uniformly when the potential energy E^{AE} proportional to the inverse of the squared distances among points is minimized, i.e.

$$E^{\mathrm{AE}} = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{1}{L_{ij}^2} \quad . \tag{2}$$

Since the objective is a sum of distances, it is not heavily disturbed by outliers from

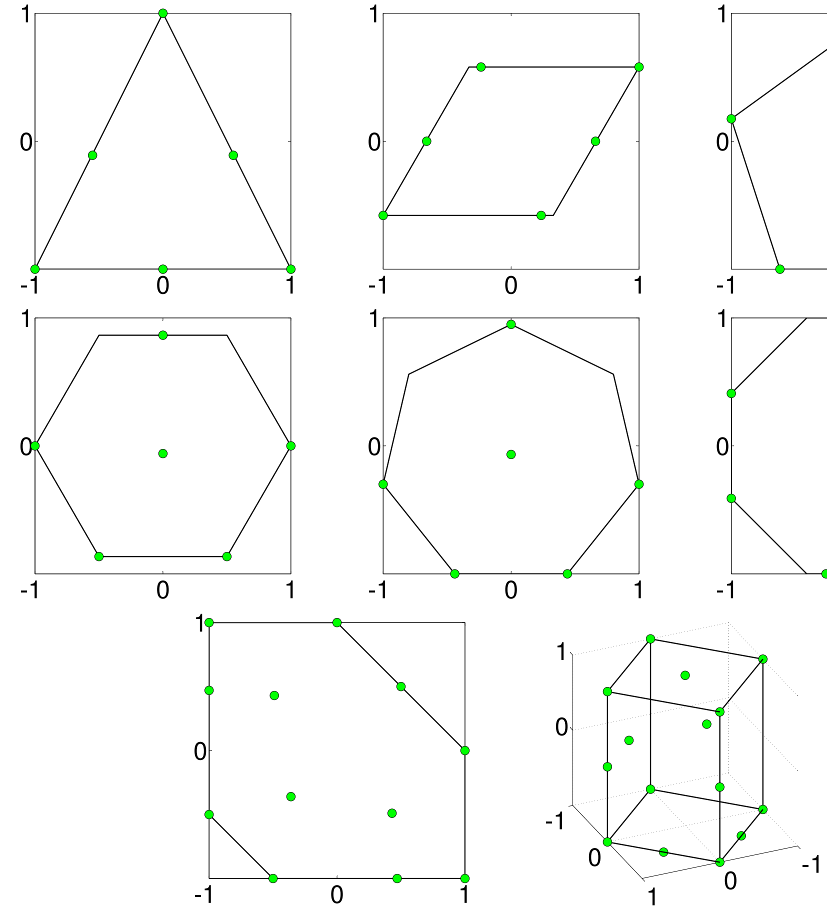


Figure 1: Reference designs [7]. Note that designs are created in a unitless domain $[-1,1]^{Dimension}$; the real designs are created by linear transformation to user-specific bounds.

the potential energy point of view. Therefore, such measure represents an average property of the set of points.

The third objective function is *D-optimality* (Dopt) proposed by Kirsten Smith in [13]. We minimize a negative value of a determinant of a linear information matrix $\mathbf{Z}$, i.e.

$$E^{\text{Dopt}} = -\det(\mathbf{Z}^{\mathbf{T}}\mathbf{Z})\,,\text{where} \tag{3}$$

$$\mathbf{Z} = \begin{bmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} \end{bmatrix}. \tag{4}$$

3 Genetic algorithm (GA) based method

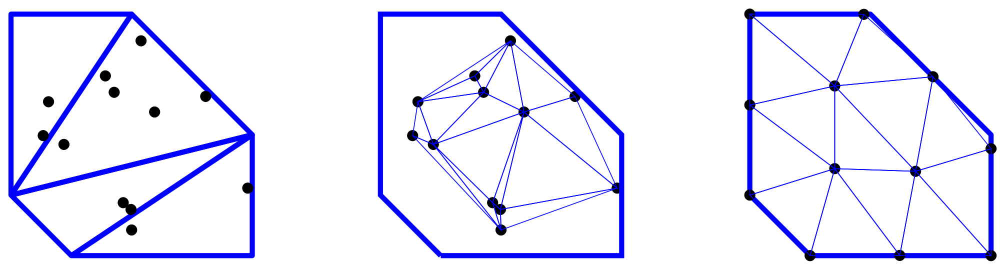
A hybrid optimization method has been proposed in [7] to solve irregular DoE problems. It is based on the efficient combination of a genetic algorithm (GA) and a sequential linear programming (SLP) methodology. Firstly, the GA is used to locate crude positions of individual points and then, the SLP is applied to find nearest local optima. Particularly, a binary version of a GA and an interior point method from Matlab is utilized. The primary objective function is the presented D-optimality in its linear form. However, the global optima of such specified problem have duplicities, i.e. few points share same positions. Although duplicities can be sometimes welcomed, here they are assumed to be deficiencies of the particular designs. Therefore, the authors in [7] have applied Bayesian modification of an information matrix which is based on adding higher order terms into the matrix $\mathbf{Z}$. Particularly, examples from the referenced paper have been solved with added quadratic terms. Note that some additional constant must be added to diagonal elements of $(\mathbf{Z}^{\mathbf{T}}\mathbf{Z})$ to solve the singularity of the resulting matrix, see e.g. [14] or [9] for more details. Although this methodology is able to find optimal solutions, not always they are global ones and also the computational demands are not low. Hence, a new method is presented in the next section.

4 Method using Delaunay triangulation (DT) and `Distmesh` tool (DM)

A triangulation is a term suitable for 2D, generally it means the partition of the domain by simplexes. Delaunay triangulation (DT) is the most popular triangulation method [15]. It is based on a convex hull of given points V describing the admissible domain, where the convex hull is the smallest convex set containing all points in V. Then, DT triangulates the convex hull such that there is no point of V inside the circumsphere of any simplex in the triangulation.

Because it is relatively simple to create DT and then compute a volume and other properties of simplexes, see Appendix A, we have a rough estimation, how is the admissible region formed. An example of utilizing such methodology has been firstly presented in [16] for regular design spaces. We extended this idea for constrained design spaces by incorporating the `Distmesh` tool (DM) [8].

In our method the domain described by corner vertices is triangulated by DT and the desired number of random points is generated inside, see Fig. 2a). Each triangle will contain a portion of the required number of samples based on a ratio of its volume to the total volume of the admissible space. Since the `floor` command is used, the missing points to the total number of points are added to the biggest simplex, see e.g. illustrative example in Fig. 2a), where smaller triangles get only two points, whereas the biggest triangle three plus three remaining. And again, since the computation of the simplexes' volumes is simple, see Appendix A, the procedure is very fast.



(a) Triangulation of the domain with randomly generated points.

(b) Triangulation of random points forming a truss-like structure.

(c) The final design after the application of the `Distmesh` tool.

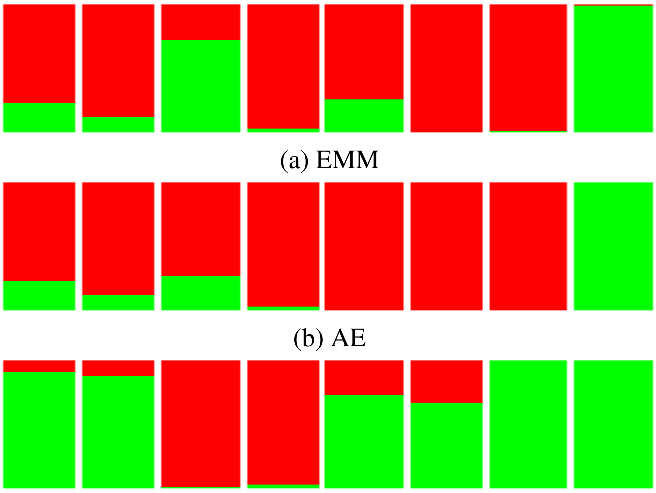
Figure 2: The generation of a uniform mesh from randomly generated points.

Then the DM tool is applied. The `Distmesh` tool is a heuristic smoothing algorithm for generating uniform meshes [15]. It is well-known that the most uniform meshes for the Finite Element Method (FEM) are characterized with uniformly spaced nodes (but not vice-versa!). Therefore, we have tried utilized this nice property of the DM tool. The DM is based on a simple dynamical system of expanding pin-jointed structure, here characterized by the second mesh, see Fig. 2b). Those trusses that are too short are causing repulsive forces that move the too close nodes apart, see Fig. 2c) for the final solution. The main disadvantage apart from high computational demands is the need to return nodes that leave the prescribed admissible domain. The DM offers solutions for basic entities, polygon used in our computations is one of them, see the original paper [8] for more details. For a 3D domain, a heuristic Monte Carlo procedure is used that moves the outlier to the nearest random point inside the domain.

5 Results

The proposed procedure has been run one hundred times for the sake of statistics. However, the referenced paper [7] has only one value from one run, therefore the comparison of these two algorithms will not be precise. The detailed results for each solved example (triangle, parallelogram, pentagon, hexagon, heptagon, octagon, irregular hexagon and prism) are presented in Tabs. 1–8 in Appendix B in the barchart form along with the visualization of the best and the worst designs of our procedure. The best designs for referenced procedure are already presented in Fig. 1. Figure 3 shows the relative winning score (RWS)[1] [17] for our method and the algorithm presented in [7]. The RWS is a statistic of 100 runs divided into ones with better results than reference values and ones with worse. We can see that our method clearly wins

[1]Note that RWS graphs are plotted by Merlin Statistical Software for Microsoft Excel `http://www.heckgrammar.kirklees.sch.uk/index.php?p=10310`, particularly the *Mosaic Plot* has been used.



(a) EMM

(b) AE

(c) Dopt

Figure 3: The comparison of the DM tool and the referenced algorithm [7] for three objectives; the vertical axis is the percentage of success (the bigger area, the better method), the horizontal axis stems for 8 individual examples (triangle, parallelogram, pentagon, hexagon, heptagon, octagon, irregular hexagon, prism). Key: Red color = DM tool, Green color = referenced algorithm.

in EMM and AE objectives, i.e. has attained a bigger area. The RWS comparison is used to save the space, for more detailed analysis see Figs. 6–7 in Appendix C, where the boxplot results of all hundred runs are shown. Last but not least, Fig. 4 shows the resulting design points from all hundred runs of our method on solved examples. Note that in several examples the local optima are created by rotating the optimal position of points around the center point.

Since the reference algorithm was optimized for Dopt objective function, it is not surprising that it wins in this objective, but not predominantly. In all but two last examples our methodology was able to find a superior solution even for the Dopt objective. Such situation is depicted in Fig. 5, where our solution (on right) attains a better Dopt value. The reason is probably in the added terms of Bayesian updating that does not allow the reference procedure to find the global optimum.

Since the codes have not been deeply optimized from implementation point of view, the analysis of computational demands cannot be rigorously done. However, we can state general requirements of the proposed method. The random generator used for the creation of the random points before applying the DM tool is very fast with no optimization cycle. The DM tool is the most demanding one. There is several Delaunay triangulations inside the loop of the Distmesh tool that are needed to preserve the inner structure to be physically consistent. And still, as is visible from the EMM performance and the 3D example, the Distmesh has problems with the quality of the boundary surface mesh, see also the discussion e.g. in [15].

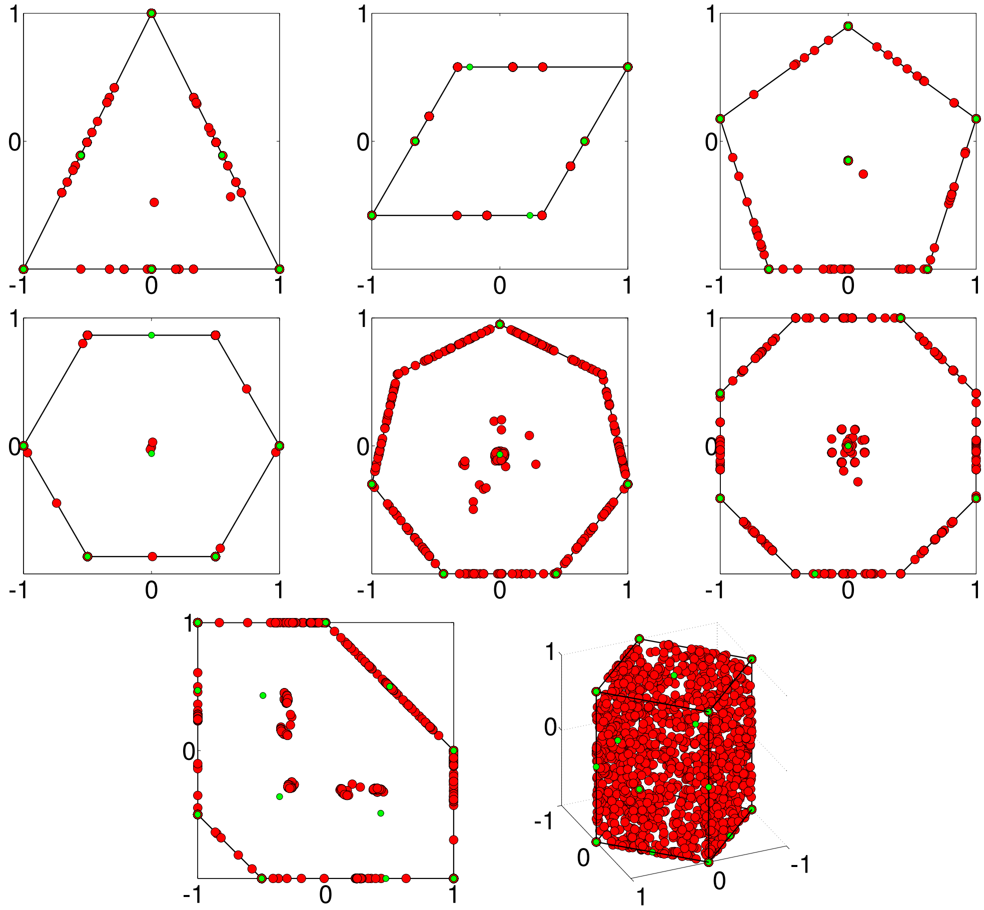


Figure 4: Red points stems for the resulting design points from all 100 runs of the DM tool, green points are reference designs [7].

6 Conclusions

The design of experiments for constrained spaces and computer experiments is relatively new and unexplored area. The constraints complicates the application of all contemporary DoE algorithms for regular design spaces. The presented paper is a pioneering work that brings a new methods and unpublished results. It is important to note that the method presented is independent on the number of dimensions as long as the procedure of returning points lying outside the prescribed domain in higher dimensions is provided. Using DT we are able to apply the DM tool on any irregular domain in N-dimensional space. Only the computational demands can limit the application in higher dimensions.

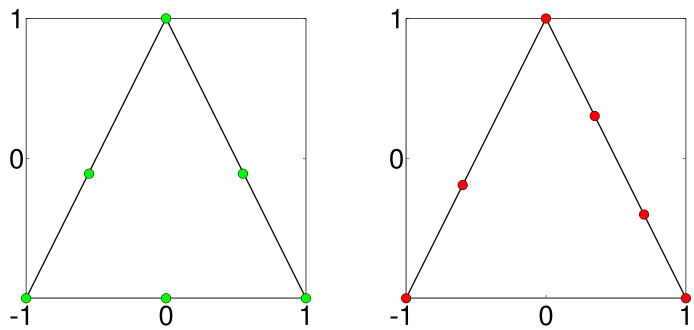


Figure 5: Comparison of designs for Example 1 (triangle and 6 design points). Left picture shows the reference design (*Dopt* = -50.0598), right picture shows the design generated by the new method with the best result in the *Dopt* objective function (*Dopt* = -53.0001). Lower value is better.

Acknowledgments

This work was supported by the Grant Agency of the Czech Republic - Project No. P105/12/1146. This financial assistance is gratefully acknowledged.

References

[1] Douglas C. Montgomery. *Design and Analysis of Experiments, 5th Edition*. Wiley, 5 edition, June 2000.

[2] K-T. Fang, R. Li, and A. Sudjianto. *Design and modeling for computer experiments*. Chapman & Hall/CRC, 2006.

[3] John A. Cornell. Experiments with Mixtures: A Review. *Technometrics*, 15(3):437–455, 1973.

[4] John A. Cornell. Experiments with Mixtures: An Update and Bibliography. *Technometrics*, 21(1):95–106, 1979.

[5] Matthieu Petelet, Bertrand Iooss, Olivier Asserin, and Alexandre Loredo. Latin hypercube sampling with inequality constraints. *AStA Advances in Statistical Analysis*, 94:325–339, 2010.

[6] Fabian Fuerle and Johann Sienz. Formulation of the Audze–Eglais uniform Latin hypercube design of experiments for constrained design spaces. *Advances in Engineering Software*, 42(9):680 – 689, 2011.

[7] M. Hofwing and N. Strömberg. D-optimality of non-regular design spaces by using a Bayesian modification and a hybrid method. *Structural and Multidisciplinary Optimization*, 42:73–88, 2010.

[8] P-O. Persson and G. Strang. A simple mesh generator in MATLAB. *SIAM Review*, 46(2):329–345, June 2004.

[9] E. Janouchová and A. Kučerová. Competitive comparison of optimal designs of experiments for sampling-based sensitivity analysis. *Computers & Structures*,

Sent for publication, 2011.

[10] Edwin R. van Dam, Gijs Rennen, and Bart Husslage. Bounds for maximin latin hypercube designs. *Operations Research*, 57:595–608, May 2009.

[11] B. Husslage. *Maximin Designs for Computer Experiments*. PhD thesis, Tilburg University, 2006.

[12] P. Audze and V. Eglais. New approach for planning out of experiments. *Problems of Dynamics and Strengths*, 35:104–107, 1977. Zinatne Publishing House.

[13] K. Smith. On the standard deviations and interpolated values of an observed polynomial function and its constants and the guidance they give towards a proper choice of the distribution of observations. *Biometrika*, 1/2:1–85, 1918.

[14] W. DuMouchel and B. Jones. A simple Bayesian modification of D-optimal designs to reduce dependence on an assumed model. *Technometrics*, 36(1):37–47, 1994.

[15] Long Chen and Michael Holst. Efficient mesh optimization schemes based on Optimal Delaunay Triangulations. *Computer Methods in Applied Mechanics and Engineering*, 200(9-12):967–984, February 2011.

[16] K. Crombecq, I. Couckuyt, D. Gorissen, and T. Dhaene. Space-filling sequential design strategies for adaptive surrogate modelling. In B. H. V. Topping and Y. Tsompanakis, editors, *Proceedings of the First International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*. Civil-Comp Press, Stirlingshire, UK, 2009.

[17] J. Nosek and M. Lepš. More practical graphical methodology for comparison of single-objective optimization algorithms. *Sent for publication*, 2011.

[18] MathPages. Simplex volumes and the Cayley-Menger determinant. www pages: http://www.mathpages.com/home/kmath664/kmath664.htm, 2011.

A Computation of simplex volume

Because we know the coordinates of simplex vertices, we use the formula which requires these (and only these) inputs [18].

The computation of a volume of a simplex in 2D (3 vertices):

$$V_2 = \frac{1}{2!} \begin{vmatrix} 1 & x_{1(1)} & x_{2(1)} \\ 1 & x_{1(2)} & x_{2(2)} \\ 1 & x_{1(3)} & x_{2(3)} \end{vmatrix}$$

The computation of a volume of a simplex in 3D (4 vertices):

$$V_3 = \frac{1}{3!} \begin{vmatrix} 1 & x_{1(1)} & x_{2(1)} & x_{3(1)} \\ 1 & x_{1(2)} & x_{2(2)} & x_{3(2)} \\ 1 & x_{1(3)} & x_{2(3)} & x_{3(3)} \\ 1 & x_{1(4)} & x_{2(4)} & x_{3(4)} \end{vmatrix}$$

The computation of a volume of a simplex in nD ($n + 1$ vertices):

$$V_n = \frac{1}{n!} \begin{vmatrix} 1 & x_{1(1)} & x_{2(1)} & \cdots & \cdots & x_{n(1)} \\ 1 & x_{1(2)} & x_{2(2)} & \cdots & \cdots & x_{n(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1(n+1)} & x_{2(n+1)} & \cdots & \cdots & x_{n(n+1)} \end{vmatrix}$$

In the notation $x_{a(b)}$ a is a variable (dimension) and b is a design point.

B Results for seven individual examples

Legend for Tables 1-8: The results of 100 runs of the DM tool for 8 examples. The first row shows the results of the EMM metric (higher is better), the second row shows the results of the AE metric (lower is better) and the third row shows the results of the $Dopt$ objective (lower is better). The first column presents the barcharts of results of the selected objective over those 100 runs. The second column shows the best and the third column shows the worst designs according to the selected objective function, respectively. Green dash line is a reference value taken from [7], green points show reference designs.

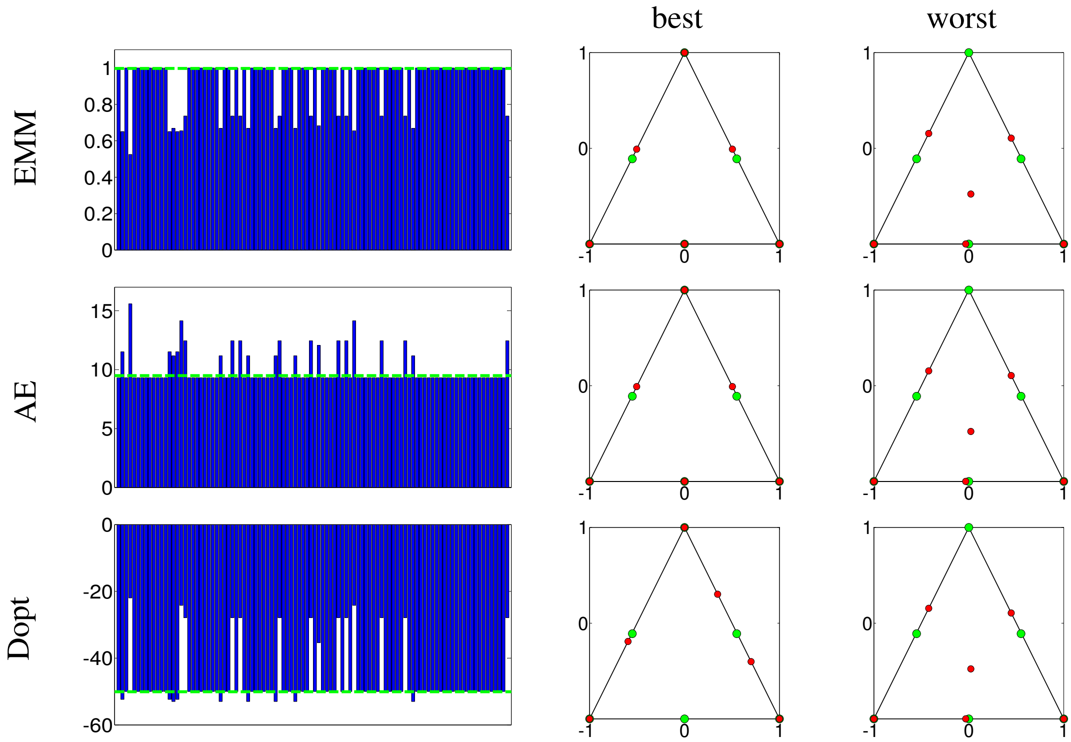


Table 1: Example 1 (triangle and 6 design points).

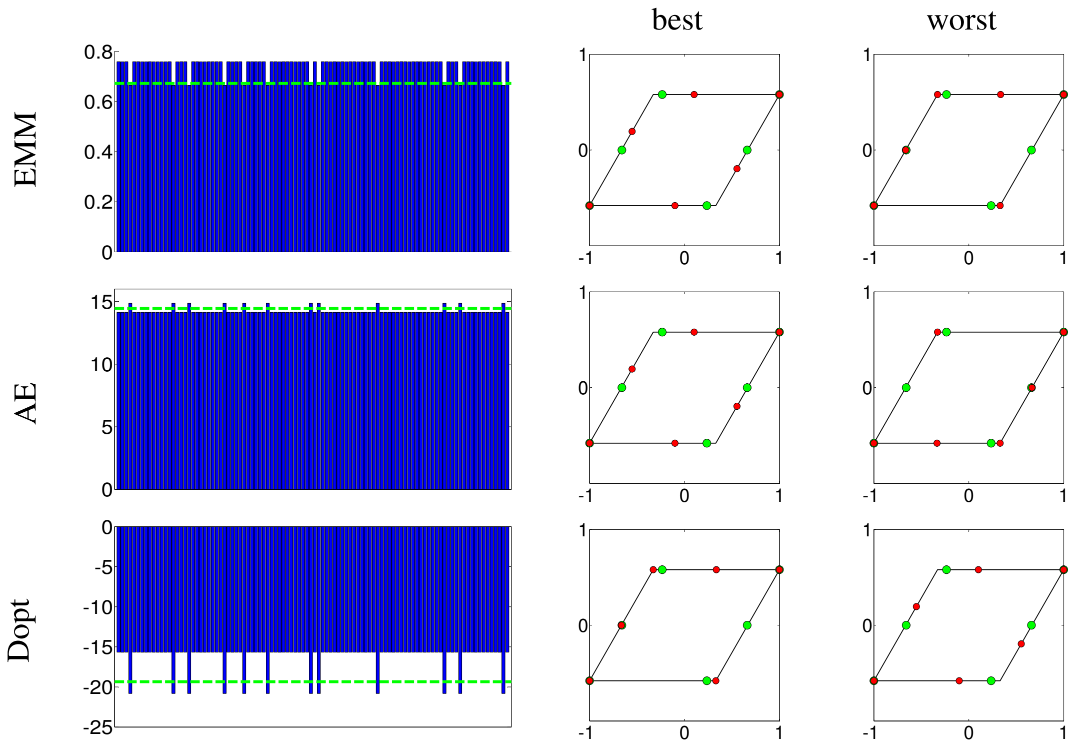


Table 2: Example 2 (parallelogram and 6 design points).

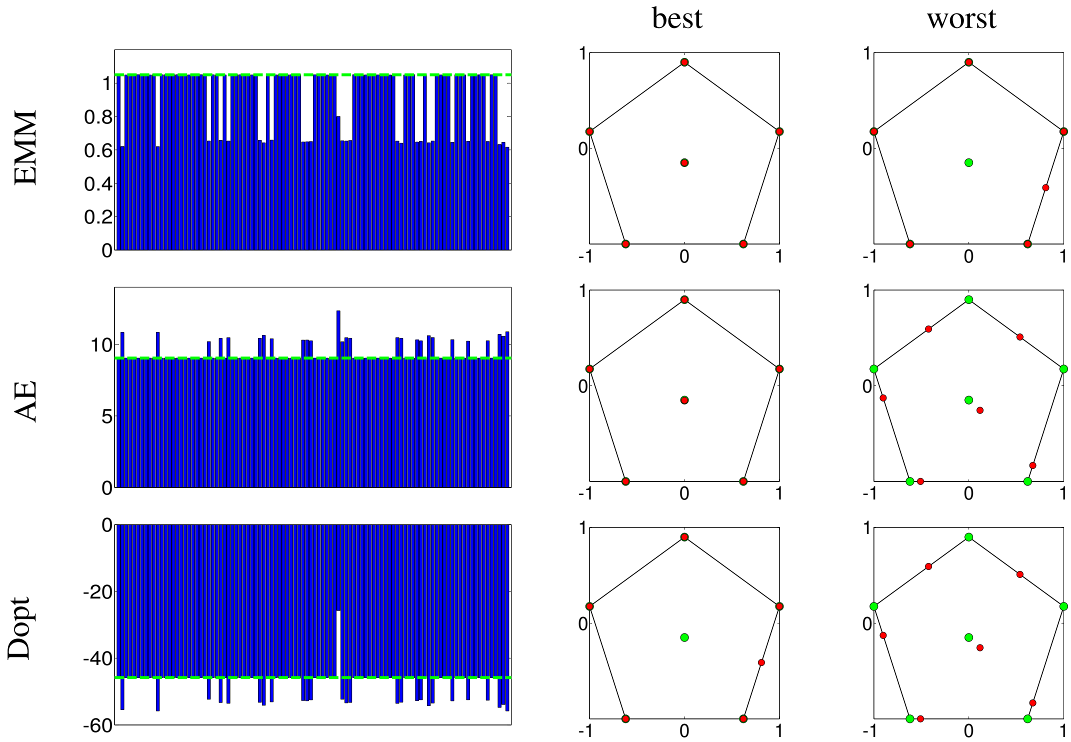


Table 3: Example 3 (pentagon and 6 design points).

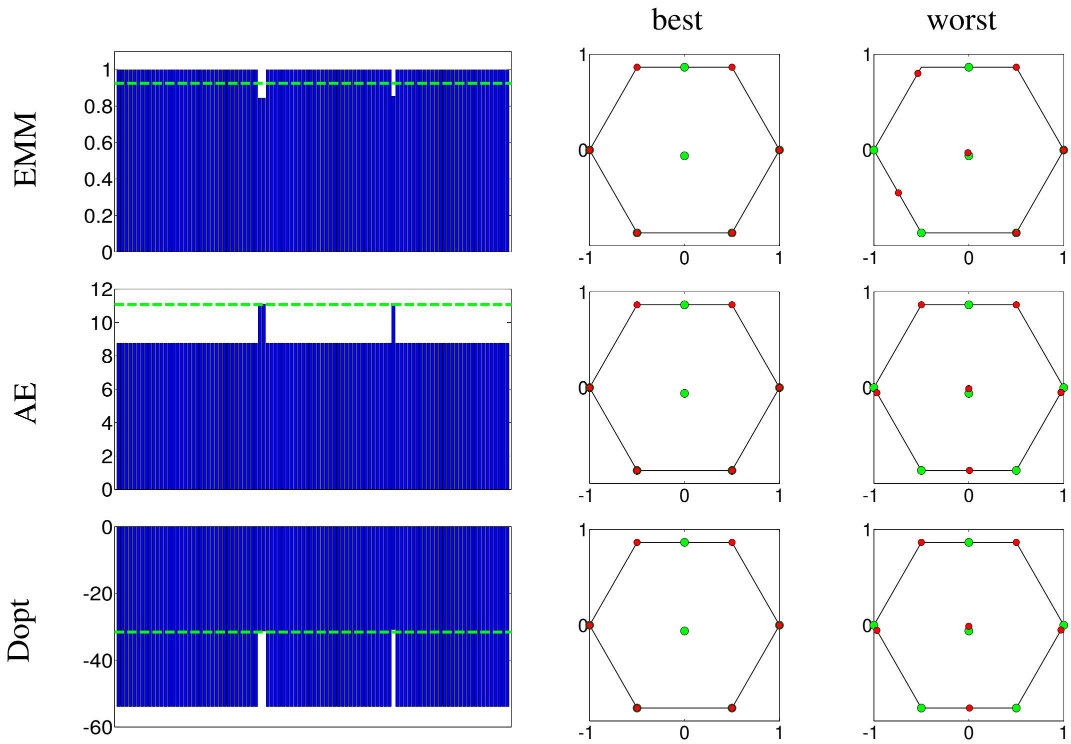


Table 4: Example 4 (hexagon and 6 design points).

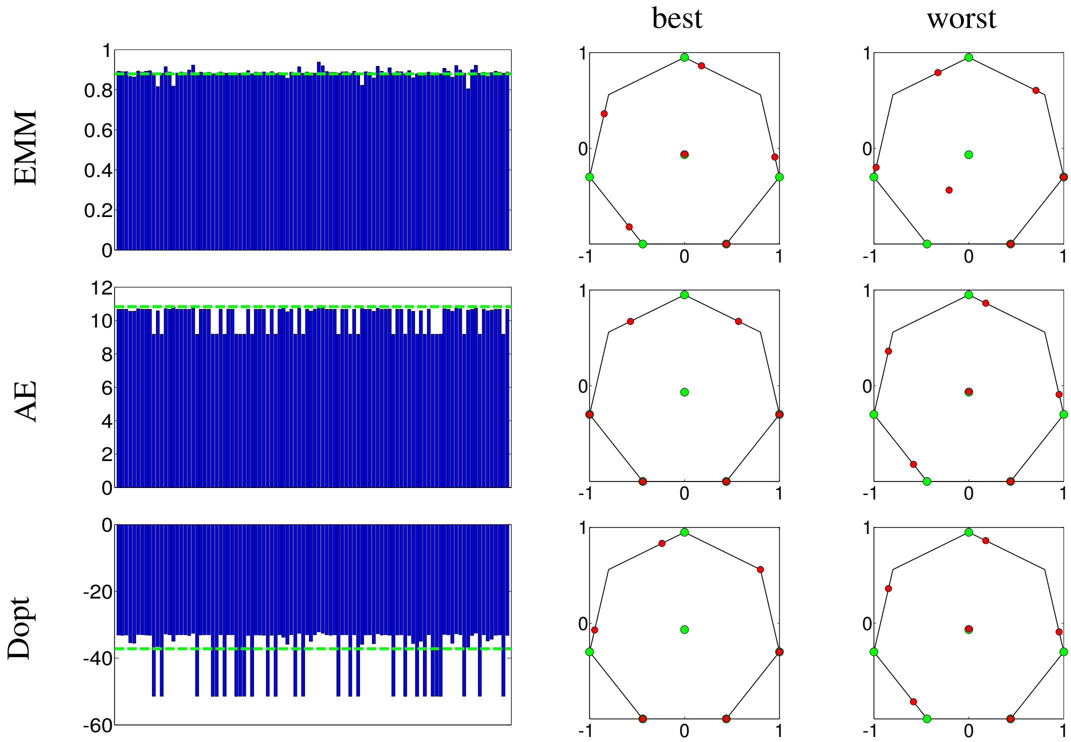


Table 5: Example 5 (heptagon and 6 design points).

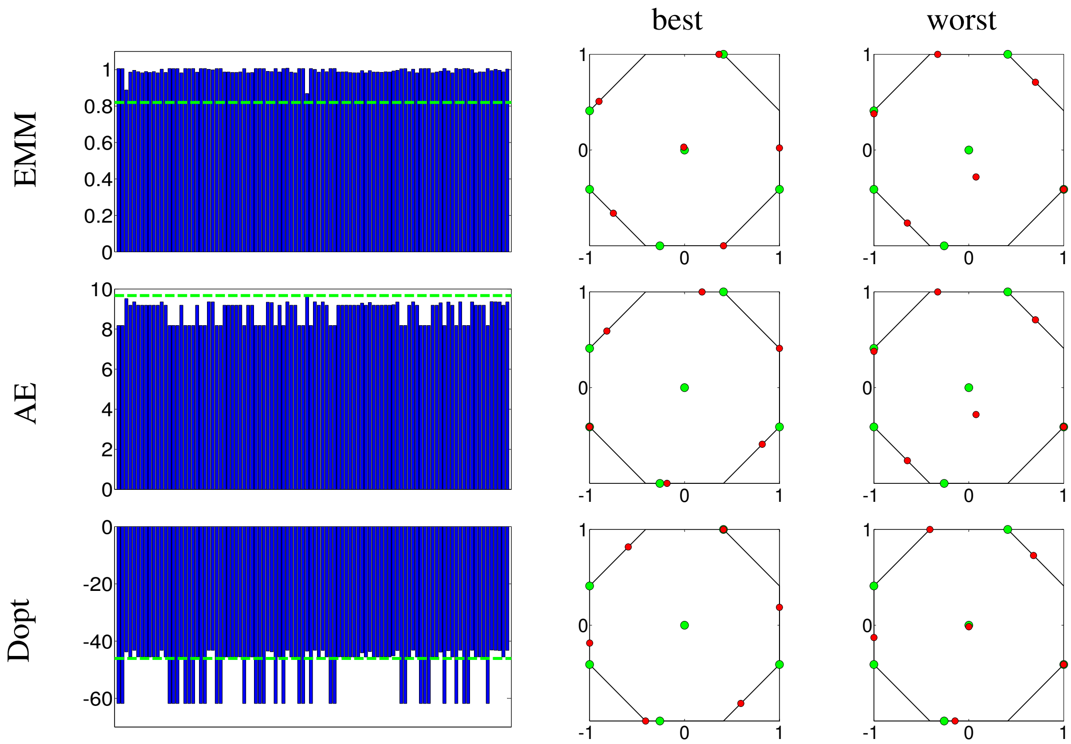


Table 6: Example 6 (octagon and 6 design points).

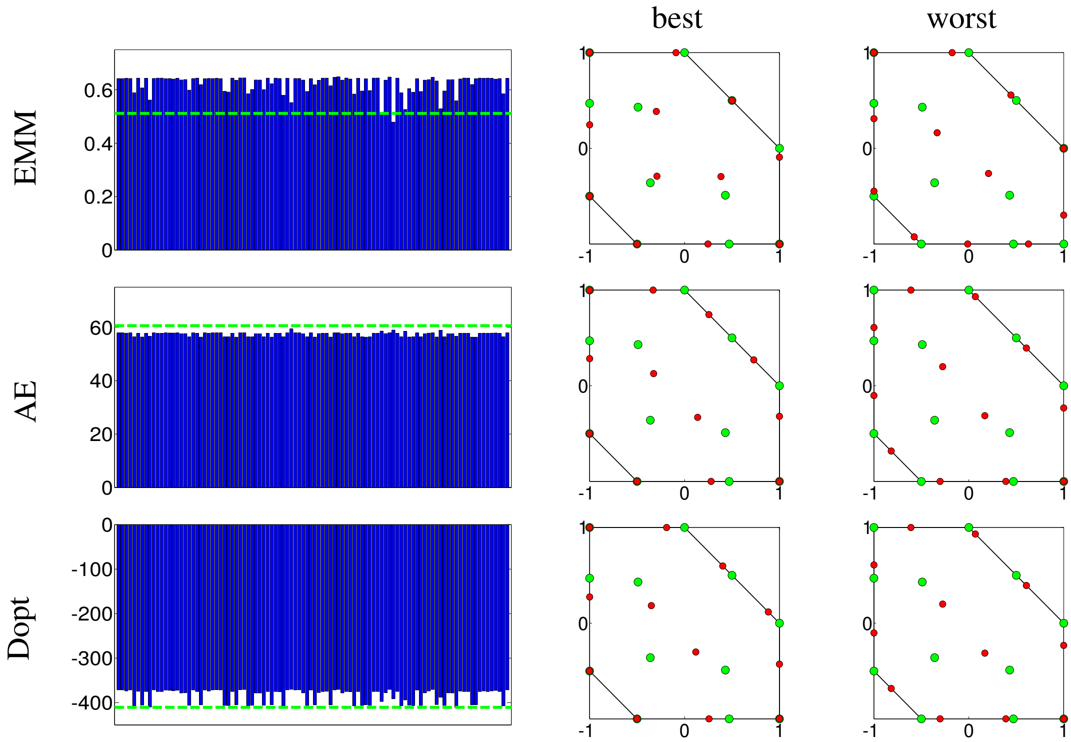


Table 7: Example 7 (irregular hexagon and 12 design points).

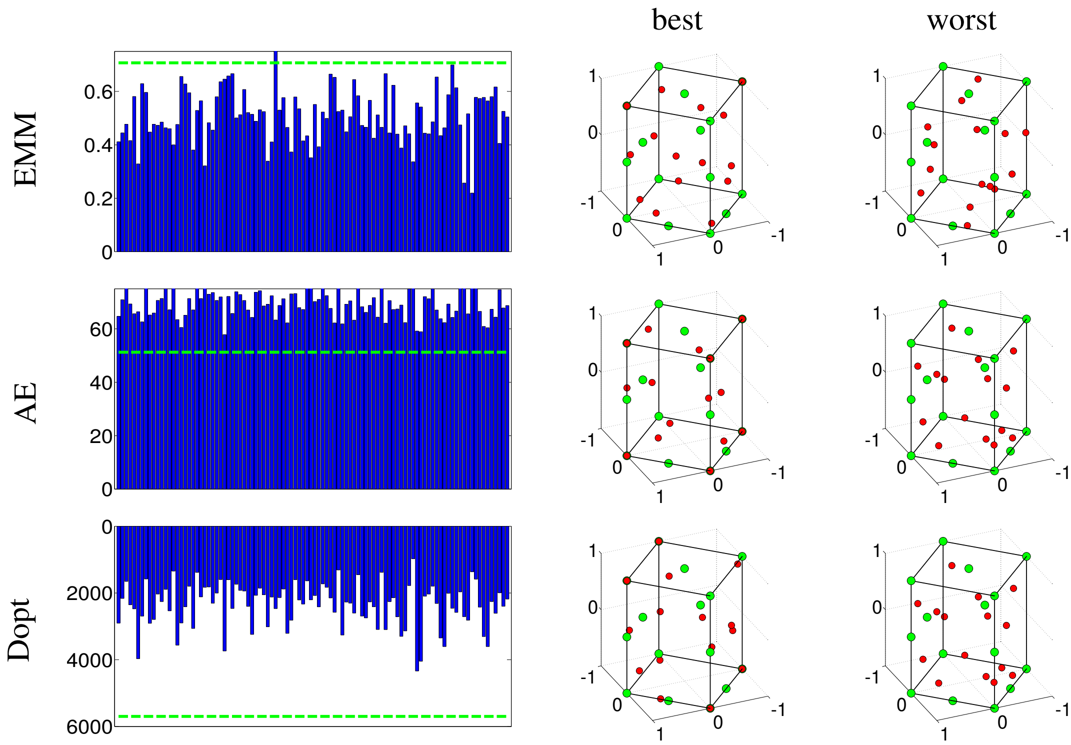
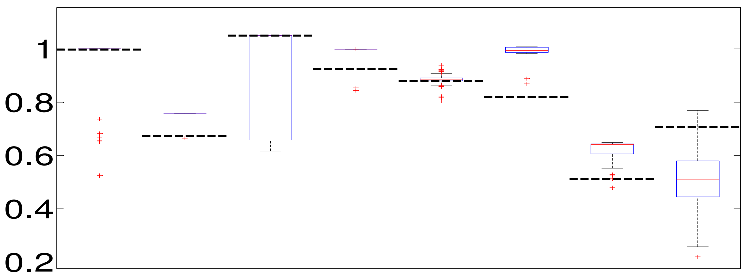
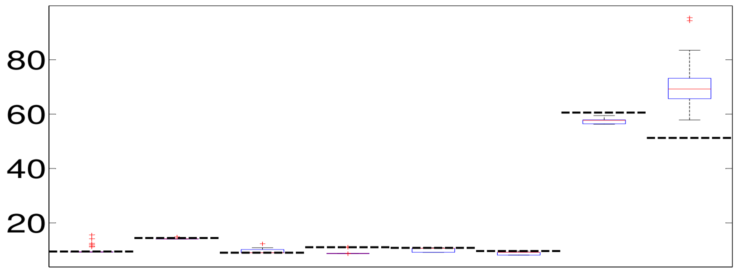


Table 8: Example 8 (prism and 15 design points).

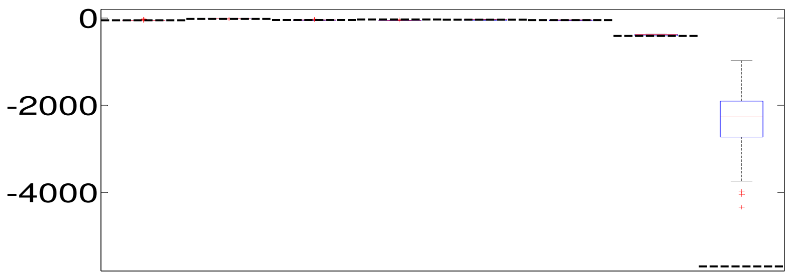
C Boxplot results



(a) EMM (higher is better)

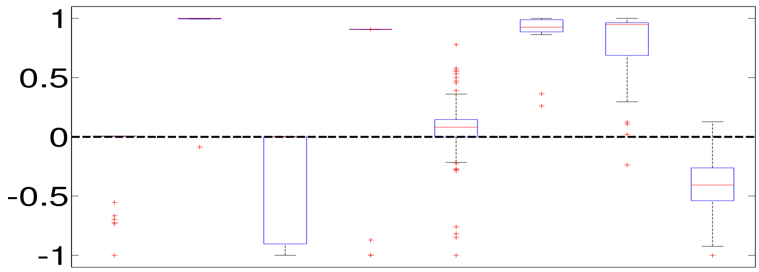


(b) AE (lower is better)

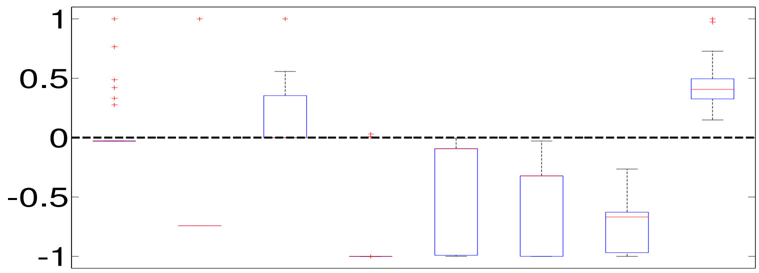


(c) Dopt (lower is better)

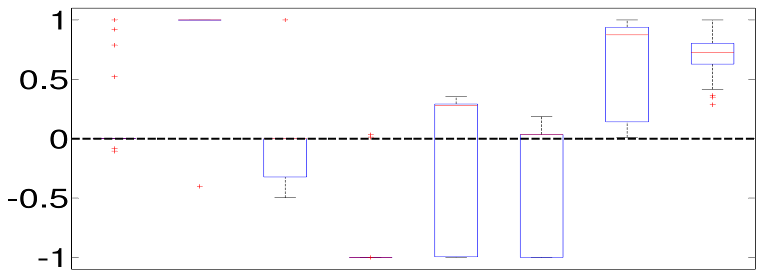
Figure 6: The boxplots of results for 8 individual examples (triangle, parallelogram, pentagon, hexagon, heptagon, octagon, irregular hexagon, prism) and three objective functions. Black dash lines are reference values taken from [7].



(a) EMM (higher is better)



(b) AE (lower is better)



(c) Dopt (lower is better)

Figure 7: The boxplots of results for 8 individual examples (triangle, parallelogram, pentagon, hexagon, heptagon, octagon, irregular hexagon, prism) and three objective functions. Values are normalized. Black dash line is a reference value taken from [7].