

Noticing Expert Patterns of Behaviour From Functional Optimizations

Submitted by - Kushlendra Mishra, Y8111018

October 26, 2009

1 Problem Definition

Many real-world design or decision making problems involve simultaneous optimization of multiple objectives. For multiple objectives there may not exist one solution which is best (global minimum or maximum) with respect to all objectives. The solution set of multi-objective optimization may consist of solutions none of which can be said to be better than the other. The set of non-dominated solutions, can be considered to be the “good designs” for the particular design task. Often, this set of “good design” instances is low dimensional manifold embedded in high dimensional design space.

Practical design tasks may be very complex; often involving the optimization of many objectives with a large number of parameters. A complex design task can be broken into smaller design tasks, involving fewer number of parameters and objectives. More often than not, these smaller design tasks are recurring, they may emerge in many different design problems. Any expert designer should have a repository of good designs for these smaller design problems. While these patterns of “good designs” for small design problems can be made available to the automated expert designer as a knowledge base, but these patterns exist only as inanimate information, without any interaction with other patterns. On the other hand if an automated designer could learn the recurring patterns of good designs as it performs the design tasks, then it could also establish the interrelation between different patterns, and build more complex patterns consisting of interrelations between previously learnt patterns. It can thus enhance its knowledge base with experience, much like a human designer.

Our aim is to explore how to learn patterns of good designs from the results of practical multi-objective optimization problems. We also want to extend and generalize such a learning process and knowledge structuring to areas other than design.

2 Introduction

2.1 What makes an Expert

Research has shown that it is not only the general abilities that differentiate experts from novices. Instead, it's the extensive knowledge that they have acquired that leads to expertise. This knowledge base affects the way they notice, organize, represent and interpret information in their environment. This is responsible for their above average abilities in remembering, reasoning and solving problems.

In [Bransford: 2003], Bransford has listed the following features of experts knowledge revealed in past researches:

- Experts notice features and meaningful patterns of information that are not noticed by novices.
- Experts have acquired a great deal of content knowledge that is organised in ways that reflect a deep understanding of their subject matter.

- Experts’ knowledge cannot be reduced to a set of isolated facts or propositions but, instead reflects contexts of applicability: that is, the knowledge is conditionalized on a set of circumstances.
- Experts are able to flexibly retrieve important aspects of their knowledge with little attention or effort.
- Though experts know their disciplines thoroughly, this does not guarantee that they are able to teach it to others effectively.

All the above characteristics of expertise are a result of how the experts “chunk” various elements of knowledge by an underlying function or strategy. Chess masters are able to chunk together several chess pieces in a configuration that is governed by some specific component of the game [DeGroot: 2008]. Expert circuit technicians chunk several individual circuit components that perform the function of a typical amplifier. Latter they are able to recognize the arrangement of many individual circuit elements as “amplifier chunks”. These chunks are further “conditionalized”– they include the context in which they are applicable. Knowledge that doesn’t have context information often remains inert, even though it may be relevant in practical situations.

The more useful and frequent of these chunks give rise to ‘symbols’. The term symbol means different things in different contexts [Mukerjee-Dabbeeru: 2009]. In cognitive sciences, it stands for the tight binding of the psychological impression of the sound (the phonological pole) with the mental image of the meaning . In [DeLoache: 2004], DeLoache gives a very general definition of a symbol as something that someone intends to represent something other than itself. On the other hand, in formal usage, a symbol is a token constituting a finite alphabet, from which higher level abstractions are derived.

An expert system also works using the knowledge-base and the set of rules and heuristics that it is programmed to work with. It may have a bigger knowledge base than a human expert, but a human expert still is more reliable than an expert system. The reason is the flexibility that humans have in using acquired symbols. A human expert learns her symbols from exposure of the world. The chunks of knowledge, or symbols are “grounded” on experience. Symbols acquired latter are defined in terms of these grounded symbols [Mukerjee-Dabbeeru: 2009]. Automated expert systems work with knowledge that remains static, the symbols here don’t evolve with experience to include newer context knowledge.

2.1.1 Chunking as a form of dimensionality reduction

Chunking can be thought of to be a process of dimensionality reduction. For example, in chess, out of the huge number of board positions only few are feasible, and out of these feasible board positions only a small fraction of moves are likely to lead to favourable result for the player. In course of her training, a chess player keeps picking up these “good” board positions and builds a collection. If she comes across a novel board position, she will search for a move that leads to one of these good configuration. On the other hand, a novice player will search and evaluate a much larger number of next board configurations. The player has effectively “learnt” a small number of good board positions out of huge number of possible configurations. This collection of good moves also helps her to recognize good moves faster as she can readily compare a move with her collection and decide whether it is favorable or not.

Likewise, a human design expert constantly explores the design space e.g. through sketching. The designer focuses mostly on the designs are good in some functional sense. Eventually some common characteristics of these good designs emerge and are abstracted away. These patterns of common characteristics of good designs represent constraints in the ways various design variables may be combined. By this process the designer comes to know which of the large number of parameters are really the important ones in a routine design task.

In [Mukerjee-Dabbeeru: 2009], it has been illustrated how an “infant designer” could learn symbols by learning usage patterns or image schemas grounded on experience. An infant designer first starts

trying different possible designs for a particular design task and evaluating them against a goodness functions. Gradually her knowledge of “good” and “bad” designs becomes more concrete. The method of learning good designs that is followed in the referred work resembles quite closely to the learning process of the human designer.

To learn a pattern of good design in a particular design task we first obtain a sample of good designs using multi-objective optimization. We then learn the manifold in which these good design points are embedded using a suitable manifold learning algorithm.

3 Literature Review

3.1 Dimensionality reduction

Dimensionality reduction is the mapping of high-dimensional data to a lower dimension without any significant loss of information. Reducing dimensionality of data is important for feature extraction, compact coding and computational efficiency. The data can be transformed into “good” representations for further processing, constraints among feature variables may be identified, and redundancy eliminated.

Ideally, the reduced representation has a dimensionality that corresponds to the intrinsic dimensionality of the data. The intrinsic dimensionality of data is the minimum number of parameters needed to account for the observed properties of the data.

3.1.1 Linear dimensionality reduction

Linear techniques assume that the data lie on or near a linear subspace of the high-dimensional space.

Principal Component Analysis (PCA) One of the important techniques of dimensionality reduction is the Principal Component Analysis [Shlens: 2003]. Principal Components Analysis constructs a low-dimensional representation of the data that describes as much of the variance in the data as possible. This is done by finding a linear basis of reduced dimensionality for the data, in which the amount of variance in the data is maximal. PCA has been successfully applied in a large number of domains such as face recognition and coin classification. The main drawback of PCA is that the size of the covariance matrix is proportional to the dimensionality of the data points. As a result, the computation of the eigen vectors might be infeasible for very high-dimensional data.

Linear Discriminant Analysis Another technique of importance is the Linear Discriminant Analysis or LDA. Linear Discriminant Analysis attempts to maximize the linear separability between data points belonging to different classes. In contrast to PCA, LDA is a supervised technique. LDA finds a linear mapping M that maximizes the linear class separability in the low-dimensional representation of the data. The above two techniques of belong to the class of linear dimensionality reduction techniques.

3.1.2 Non-linear dimensionality reduction

Linear techniques of dimensionality reduction fail to map if the data set to be mapped is embedded in a non-linear manifold. Ideally, an algorithm should be able to map the data set to a subspace of the same dimension as the manifold. But the linear methods map smallest convex subspace encapsulating the manifold, often of much higher dimension than the manifold itself. Non linear algorithms are more relevant for our work as non-linear relations are extremely common between design variables in practical design problems.

The linear techniques of dimensionality reduction are well established and understood. In contrast, most non-linear dimensionality reduction techniques have been proposed more recently and therefore are less well studied. There are two subclasses of non-linear dimensionality reduction techniques which

differ in their approach to the problem. Global techniques of non-linear dimensionality reduction are the techniques that attempt to preserve the global properties of data

Multi-dimensional scaling Multidimensional scaling represents a collection of nonlinear techniques that maps the high-dimensional data representation to a low-dimensional representation while retaining the pairwise distances between the data points as much as possible. The quality of the mapping is expressed in the stress function, a measure of the error between the pairwise distances in the low-dimensional and high-dimensional representation of the data. MDS is widely used for the visualization of data, e.g., in fMRI analysis and in molecular modelling.

Multidimensional scaling has been proved to be successful in many applications, but it suffers from the fact that it is based on Euclidean distances, and does not take into account the distribution of the neighboring data points. If the high-dimensional data lies on or near a curved manifold, such as in the Swiss roll data set, MDS might consider two data points as near points, whereas their distance over the manifold is much larger than the typical inter-point distance. Isomap is a technique that resolves this problem by attempting to preserve pairwise geodesic (or curvilinear) distances between data points.

Isomap In Isomap, the geodesic distances between the data points x_i are computed by constructing a neighborhood graph G , in which every data point x_i is connected with its k nearest neighbors x_{i_j} in the data set X [Tenenbaum-Silva-Langford: 2000]. The shortest path between two points in the graph forms a good (over)estimate of the geodesic distance between these two points, and can easily be computed using Dijkstra's shortest-path algorithm. The geodesic distances between all data points in X are computed, thereby forming a pairwise geodesic distance matrix. The low-dimensional representations y_i of the data points x_i in the low-dimensional space Y are computed by applying multidimensional scaling on the resulting distance matrix.

An important weakness of the Isomap algorithm is its topological instability. Isomap may construct erroneous connections in the neighborhood graph G . Such short-circuiting can severely impair the performance of Isomap. Several approaches have been proposed to overcome the problem of short-circuiting [Saxena-Gupta-Mukerjee: 2004], by removing data points with large total flows in the shortest path-algorithm or by removing nearest neighbors that violate local linearity of the neighborhood graph.

Locally linear embedding Isomap is a technique that attempts to retain the global properties of data. Local Linear Embedding (LLE) [Saul-Roweis: 2003] is a local technique for dimensionality reduction that is similar to Isomap in that it constructs a neighborhood graph representation of the data points. In contrast to Isomap, it attempts to preserve solely local properties of the data, making LLE less sensitive to short-circuiting than Isomap. Furthermore, the preservation of local properties allows for successful embedding of non-convex manifolds. In LLE, the local properties of the data manifold are constructed by expressing the data points as a linear combination of their nearest neighbors. In the low-dimensional representation of the data, LLE attempts to retain the reconstruction weights in the linear combinations as well as possible.

LLE describes the local properties of the manifold around a data point x_i by writing the data point as a linear combination W_i (the so-called reconstruction weights) of its k nearest neighbors x_{i_j} . Hence, LLE fits a hyperplane through the data point x_i and its nearest neighbors, thereby assuming that the manifold is locally linear. The local linearity assumption implies that the reconstruction weights W_i of the data points x_i are invariant to translation, rotation, and rescaling. Because of the invariance to these transformations, any linear mapping of the hyperplane to a space of lower dimensionality preserves the reconstruction weights in the space of lower dimensionality.

LLE has been successfully applied for, e.g., super-resolution [Chang-Yeung-Xiong: 2004] and sound source localization [Duraiswami-Raykar: 2005]. However, there also exist experimental studies that report weak performance of LLE. In [Lim-Ciechomski-Sarni-Thalman: 2003], LLE is reported to fail in the visualization of even simple synthetic biomedical datasets. In [39], it is claimed that LLE performs

worse than Isomap in the derivation of perceptual-motor actions. A possible explanation lies in the difficulties that LLE has when confronted with manifolds that contain holes.

3.2 Multi-Objective Optimization Algorithms

3.2.1 Optimization based on aggregation

In classical techniques of multi-objective optimization the objectives are combined to form one single objective by using some knowledge of the problem being solved. The optimization of the single objective results in a single point pareto-optimal solution.

Objective weighting One of the classical methods of multi-objective optimization is the Method of objective weighting. Multiple objective functions are combined into one overall objective function by assigning fractional weights to the individual objective functions which sum to one. In this method the optimal method the optimal solution is controlled by the weight vector W . Preference of an objective can be changed by modifying the corresponding weight. Therefore depending on the importance of objectives a suitable weight vector is chosen and the single objective problem is used to find the desired solution. The only advantage of this method over others is that the emphasis of one objective over the other can be controlled and the obtained solution is a usually Pareto-optimum solution.

Distance functions In the method of Distance Functions the scalarization is achieved by using a demand-level vector which has to be specified by decision maker. The single objective function is obtained by taking the sum of (usually) euclidean distances. The solution obtained by optimizing the above function depends on the demand level vector. The solution to this optimization problem is not guaranteed, hence the decision maker must have a thorough knowledge of individual optima of each objective prior to the selection of the demand level. This problem is similar to objective weighting. The only difference is that in this method the goal for each objective is required to be known whereas in the previous method the relative importance of each objective is required.

Min-Max formulation The Min-Max Formulation method is different in principle than the above two methods. This method attempts to minimize the relative deviations of the single objective functions from individual optimum. That is it tries to minimize the objective conflict. This method can yield the best possible compromise solution when objectives of equal priority are to be optimized.

The most profound drawback of these methods is their sensitivity towards weights or demand vectors. The solutions obtained largely depend on the underlying weight-vector or demand level. Thus for situations, different weight-vectors need to be used and the same problem needs to be solved a number of times.

3.2.2 Evolutionary algorithms for multi-objective optimization

In real world situations decision makers often need different alternatives in decision making. Moreover if some of the objectives are noisy or have discontinuous variable space the aggregation methods may not work effectively.

Multi-Objective evolutionary algorithms can find multiple Pareto-optimal solutions simultaneously. The first practical evolutionary multi-objective optimization algorithm, called Vector Evaluated Genetic Algorithm (VEGA), was developed by Schaffer in 1984 [Schaffer: 1984@]. One of the problems with VEGA, as realised by Schaffer himself, is its bias towards some Pareto-optimal solutions. Later, Goldberg suggested a non-dominated sorting procedure to overcome this weakness of VEGA.

The idea behind the non-dominated sorting procedure is that a ranking selection method is used to emphasize good points and a niche method is used to maintain stable sub-populations of good points. The Non-Dominated Sorting Genetic Algorithm (NSGA) was developed by Deb and Srinivas [Srinivas-Deb: 1994].

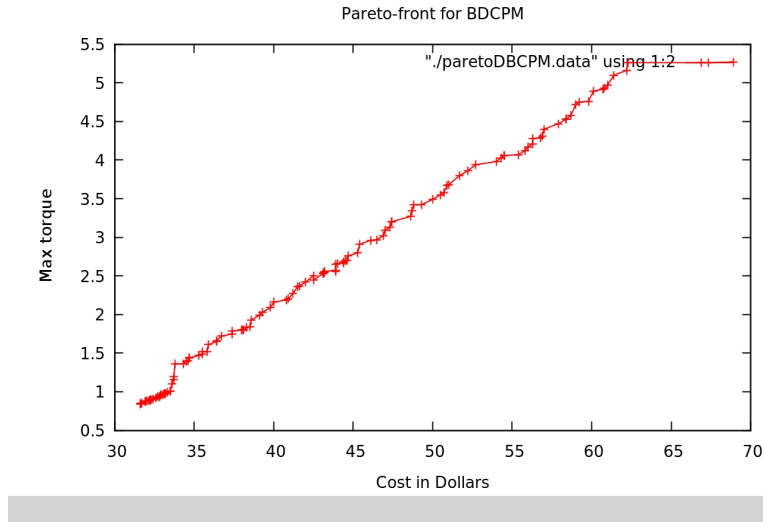


Figure 1: paretofront for the BDCPM design problem

Any evolutionary algorithm consists of a number of iterations of the selection, crossover and mutation operators over a set of possible solutions (population). NSGA varies from the simple evolutionary algorithms in the way selection operator works. Before the selection is performed, the population is ranked on the basis of an individual’s non-domination. The non-dominated individuals present in the population are first identified from the current population. Then all these individuals are assumed to constitute the first non-dominated front in the population and assigned a large dummy fitness value. The population is then reproduced according to the dummy fitness values.

4 The work so far

We tried to learn the manifold of good designs of brushless d. c. permanent magnet motor design problem. The problem has been earlier modeled in [Deb-Sindhya: 2008]. We were able to obtain a pareto front similar to that in the referred work. After running the Isomap algorithm on the obtained paretofront, the plot of residual error vs mapping dimensionality shows the biggest drop from 2 to 3.

Figure 3, 4 and 5 show the three dimensional mapping obtained by Isomap for this problem.

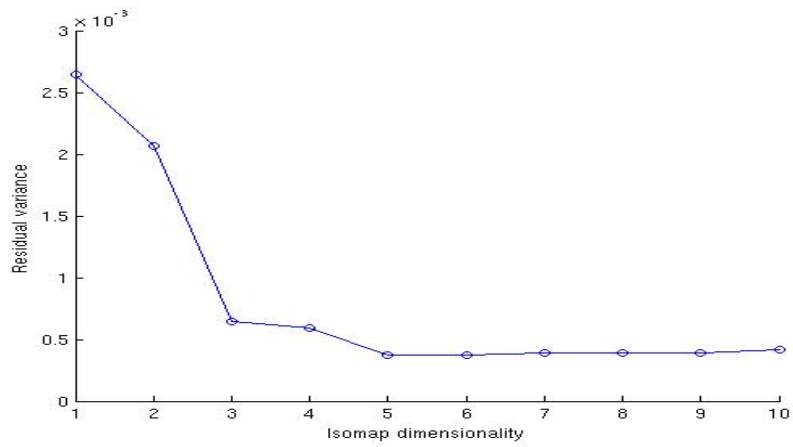


Figure 2: Plot of residual variance for BDCPM problem Isomap mappings

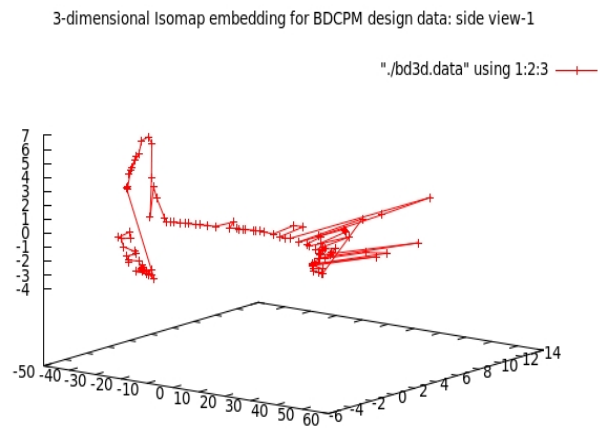


Figure 3: 3-d Isomap mapping: side view 1

3-dimensional Isomap embedding for BDCPM design data: side view-2

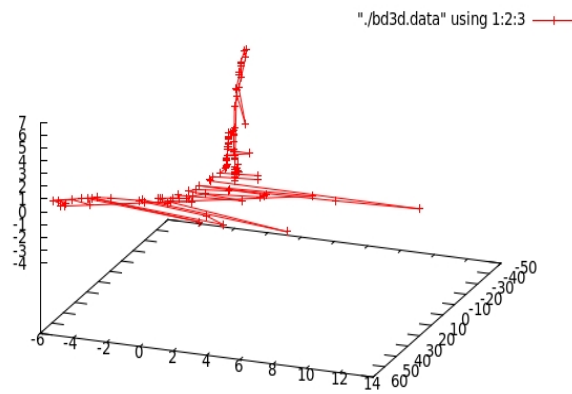


Figure 4: 3-d Isomap mapping: side view 2

3-dimensional Isomap embedding for BDCPM design data: top view

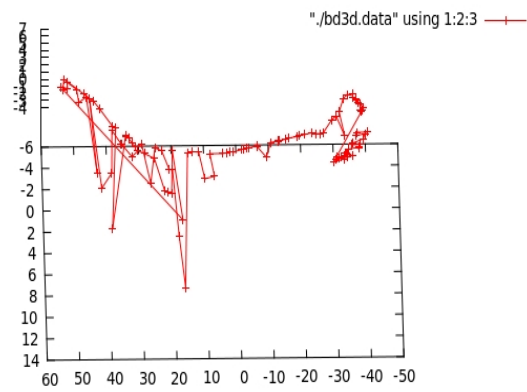


Figure 5: 3-d Isomap mapping: top view

References

- M. Belkin and P. Niyogi, *Laplacian Eigenmaps and Spectral techniques for Embedding and clustering*, ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS
- Amitabha Mukerjee and M. M. Dabbeeru, *The Birth of Symbols in Design*, ASME 2009 International Design Engineering Technical Conferences and Computers and Information Engineering Conference August-September 2009.
- A. Saxena, A. Gupta, A. Mukerjee, *Non-Linear Dimensionality Reduction by Locally Linear Isomaps*, LECTURE NOTES IN COMPUTER SCIENCE, Springer
- J. B. Tenenbaum, V. Silva, and J. C. Langford, *A Global Geometric Framework for Nonlinear Dimensionality Reduction*, Science
- S. Martin and A. Backer, *Estimating Manifold Dimension by Inversion Error*, SAC '05: Proceedings of the 2005 symposium on Applied Computing.
- M. A. Carreira-Perpinan and R. S. Zemel, *Proximity graphs for Clustering and Manifold Learning*, Advances in Neural Information Processing Systems 17: Proceedings of the 2004 Conference.
- L. K. Saul and S. T. Roweis *Think globally, fit locally: unsupervised learning of low dimensional manifolds*, The Journal of Machine Learning Research, MIT Press Cambridge, MA, USA.
- N. Srinivas and K. Deb, *Multiobjective Optimization Using Nondominated sorting in Genetic Algorithms*, Evolutionary Computation.
- K. Deb, A. Pratap, S. Agrawal and T. Meriyan, *A Fast and Elitist multiobjective Genetic Algorithm*, IEEE transactions on evolutionary computation.
- J. Shlens *A tutorial on Principal Component Analysis*, Citeseer.
- M. Zeleny *Multi criteria decision making: eight concepts of optimality*, Journal of Human Systems Management.
- J. Bransford, *How people learn: Brain, mind, experience, and school*, National Academy Press.
- J. S. DeLoache, *Becoming symbol-minded*, Trends in Cognitive Sciences, 2004.
- K. Deb and K. Sindhya, *Deciphering innovative principles for optimal electric, brushless dc permanent magnet motor design*, IEEE World Congress on Computational Intelligence
- H. Chang, D. Y. Yeung and Y. Xiong, *Super-resolution through neighbor embedding*, In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, pages 275-282.
- R. Duraiswamy and V. C. Raykar, *The manifolds of spatial hearing*, in Proceedings of International Conference on Acoustics Speech and Signal Processing, volume 3, pages 285-288, 2005.
- J. D. Schaffer, *Some experiments in machine learning using vector evaluated genetic algorithm.*, Doctoral dissertation, Vanderbilt University, Electriccal Engineering, Tennessee.
- I. S. Lim, P. H. Ciecchowski, S. Sarni and D. Thalmann, *Planar arrangement of high-dimensional biomedical data sets by Isomap coordinates*, in Proc. of the 16th IEEE Symposium on Computer-Based Medical Systems.
- A D De Groot, *Thought and choice in chess*, Amsterdam University Press.
- C. Hegde, M. B. Wakin and R. G. Baraniuk, *Random Projections for Manifold Learning*, Neural Information Processing Systems.