Hyperspace Geography: Visualizing Fitness Landscapes beyond 4D

Janet Wiles*
School of Information Technology and Electrical Engineering
The University of Queensland
QLD 4072, Australia
j.wiles@itee.uq.edu.au

Bradley Tonkes
School of Computer Science and Engineering
University of New South Wales
btonkes@cse.unsw.edu.au

Abstract Human perception is finely tuned to extract structure about the 4D world of time and space as well as properties such as color and texture. Developing intuitions about spatial structure beyond 4D requires exploiting other perceptual and cognitive abilities. One of the most natural ways to explore complex spaces is for a user to actively navigate through them, using local explorations and global summaries to develop intuitions about structure, and then testing the developing ideas by further exploration. This article provides a brief overview of a technique for visualizing surfaces defined over moderate-dimensional binary spaces, by recursively unfolding them onto a 2D hypergraph. We briefly summarize the uses of a freely available Web-based visualization tool, Hyperspace Graph Paper (HSGP), for exploring fitness landscapes and search algorithms in evolutionary computation. HSGP provides a way for a user to actively explore a landscape, from simple tasks such as mapping the neighborhood structure of different points, to seeing global properties such as the size and distribution of basins of attraction or how different search algorithms interact with landscape structure. It has been most useful for exploring recursive and repetitive landscapes, and its strength is that it allows intuitions to be developed through active navigation by the user, and exploits the visual system's ability to detect pattern and texture. The technique is most effective when applied to continuous functions over Boolean variables using 4 to 16 dimensions.

Keywords
Hyperspace graph paper, hypergraphs, fitness landscapes, cost surfaces, royal road, hierarchical if-and-only-if

1 Navigating Hyperspace

Hyperspace Graph Paper (HSGP) is a technique for visualizing surfaces defined over moderate-dimensional binary spaces [1].[1] The key to this technique is in unfolding a hypercube using recursive steps so that the topology of a high-dimensional space is reflected in a recursive structure in a two-dimensional unfolding (called a hypergraph; see Figure 1). In this article, we describe a case study of this layout technique and its application to understanding the interaction between search algorithms and landscape structures.

Human perception can be viewed as a collection of computational primitives that exploit features such as spatial proximity or optic flow to provide immediate insight into low-dimensional spatial data (one to three dimensions). Orthographic projections are a common technique for exploring higher-dimensional spaces, but can be of limited use even for estimating distances and mapping

* Corresponding author.
1 An interactive tool is available at http://www.itee.uq.edu.au/~btonkes/hsgp.html.

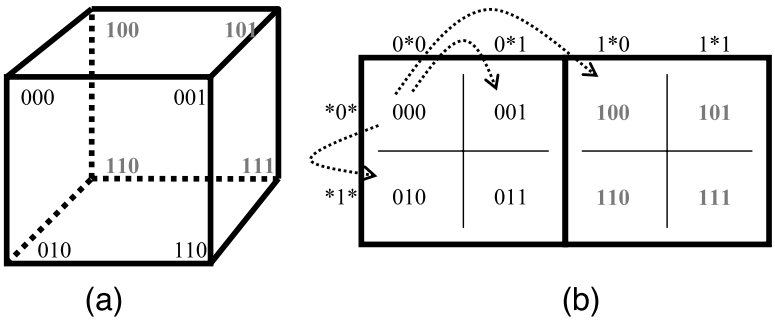

Figure 1. Hyperspace Graph Paper (HSGP) is based on unfolding all corners of a cube using a recursive translation operator. A cube has eight corners, so the corresponding unfolded graph has eight cells. Each additional dimension doubles the size of the hypergraph. For a hypercube in *n* dimensions, the corresponding unfolded hypergraph has 2^n cells.

local neighborhoods. The question arises whether humans can gain immediate insight into higher-dimensional spatial structures if suitable representations are provided. The recursive layout of the hypergraph was developed as a candidate representation for developing such a tool.

The hypergraph was originally designed as a layout technique to assist users in gaining a working knowledge of multidimensional concepts, not necessarily in search tasks or evolutionary computation. It uses hierarchical concepts familiar to many disciplines, such as Karnaugh maps in logic design [2, 3], recursive decomposition in algorithm design [4], and quadtrees in image processing [5]. These early applications were frequently used to simplify multidimensional spaces in order to focus on one part of a complex structure at a time. By contrast, in evolutionary computation, an understanding is sought for global properties of search spaces, and insights are frequently based on spatial metaphors. The landscape analogy for evolutionary search dates to Sewell Wright's work on population genetics in the 1930s, and is a generic metaphor in evolutionary computation. Many studies have used software tools for visualizing high-dimensional data; see for example [6, 7]. The use of hypergraphs in visualizing landscapes is novel in the way we tested the human ability to infer the structure of high-dimensional spaces by navigating maps constructed using recursive layout algorithms.

2 Evaluating Hypergraphs

Hypergraphs require familiarization to be an effective technique for visualizing high-dimensional landscapes. The recursive layout technique was empirically tested on computer science and mathematics graduate students using a variety of navigation tasks [8]. A one-hour training program was developed to familiarize participants with the layout of six- and eight-dimensional hypercubes, and a simple navigation task was designed (see Figure 2). Surprisingly, even among this group of highly sophisticated users, performance varied markedly, with half the participants quickly grasping the recursive layout structure and the other half not above chance in navigation accuracy after the training period. The majority of the successful participants showed a clear linear relationship between time taken and number of dimensions separating points on a path. This result was a natural consequence of the navigation heuristics suggested to the users and hence consistent with expectations. Interestingly, one unusual participant not only gave perfect performance on the tasks, with the fastest response rates of all participants, but also showed no increase in the time to estimate distances for pairs of points that were farther apart.

The study showed that the hyperspace graph paper was useful as a high dimensional visualization for some but not all participants. The accuracy of each participant was a clear indication of whether the visualization was a usable tool for that individual. The clear differences in performance showed that some participants understood and could competently use the layout representation after one hour of training, and others were unable to use it effectively at all. This difference reflects the

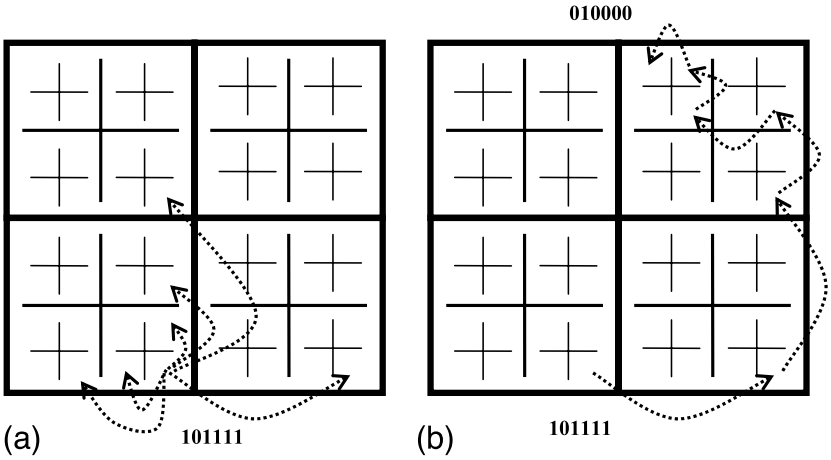

Figure 2. Neighbors and paths through hypercubes are located using translational symmetry in the hypergraph. (a) In a six-dimensional space, each point has six one-bit neighbors translated by 1, 2, and 4 cells horizontally and vertically. (b) A minimum path between two points at opposite ends of the hypercube has six steps; each step is one unit of Hamming distance closer to the target (many other equivalent paths of six steps also exist). Note that not every step that is 2^k cells horizontally or vertically translated will correspond to a one-bit neighbor—only those in the same recursive substructure.

authors' experiences with hypergraphs, that once users "get" the layout, they can use it effectively. Prior to that point (which can take some time), the layout seems to have no benefit at all. The linear relationship between distance and response rates for the majority of successful users supports the idea that a heuristic for navigation can be iteratively applied. The one outstanding participant shows that for some people, the tool allows an even more effective insight into higher-dimensional spaces.

3 Hypercube Geography

Following the visualization tests, a series of tools were developed, HSGP being the latest and most sophisticated. It allows exploration of fitness landscapes used in evolutionary computation, including a variety of schema-based and multimodal functions [1]. Each cell in a hypergraph is colored with the fitness value of its corresponding point in the fitness landscape. Repeated structures in the landscape are clearly visible as repeated patterns in medium-dimensional (8–16-dimensional) hypergraphs, and as textures in higher-dimensional ones.

4 Strengths and Limitations in Practice

Using HSGP to visualize surfaces allows the user to bridge the gap between the limits of the human visual system (three spatial dimensions) and the properties of higher-dimensional surfaces, which often cannot be effectively reasoned about with low-dimensional metaphors. Because fitness landscapes scale exponentially, there are practical limitations for any visualization technique that attempts to show an entire space. HSGP has an effective limitation of around 16 dimensions that can be usefully displayed with real-valued fitness functions, due to the constraints of screen sizes. The recursive layout can be extended to higher-dimensional spaces for coarser fitness functions, for example to gain insight into extremely sparse spaces in which the majority of fitness values are zero and single pixels suffice to represent nonzero cells. In practice, however, the insights from 16 dimensions were sufficient to generalize to most other higher-dimensional spaces. The recursive layout is appropriate for real-valued functions defined over Boolean variables. In theory the recursive

layout could also be applied to discrete variables with more than two values, but in practice the efficacy of inferring global structure from the navigation technique does diminish.

5 Applications

Understanding the properties of high-dimensional search spaces provides insight into the types of strategies that optimization algorithms could utilize. HSGP displays a variety of properties of interest, including the ruggedness of a surface, number and distribution of local and global optima, size and shape of basins of attraction, length of adaptive walks, and neutral layers [1, 8, 9].

HSGP can be used to visualize a variety of fitness landscapes, and the software includes a range of functions from the evolutionary computation literature. The simplest landscape is the Royal Road [10], which is useful for fine-tuning the user's intuitions about the hypergraph layout. More complex functions have been used to define recursive hierarchies which have been used extensively for testing genetic algorithms, such as the hierarchical if-and-only-if (HIFF; see Figure 3) [11], hierarchically defined functions (HDFs) [12], and hyperplane defined functions (hdfs) [13]. The size of the basins of attraction affects the efficacy of different evolutionary algorithms, and HSGP shows how the basin sizes of HIFF, HDF, and hdf scale with the height of local optima. Another class of functions used to develop intuition about combinatorial interactions [14] includes the tunably rugged landscapes of Kauffman's NK. Comparing landscapes with increasing ruggedness shows the corresponding increase in the number of basins of attraction and increase in search difficulties. The software also demonstrates a variety of ways of adding neutrality to landscapes, including the NKp and NKq variants of NK landscapes [15, 16]. Comparing neutral variants shows the differences between definitions of neutrality such as NKp, which projects multiple values to a

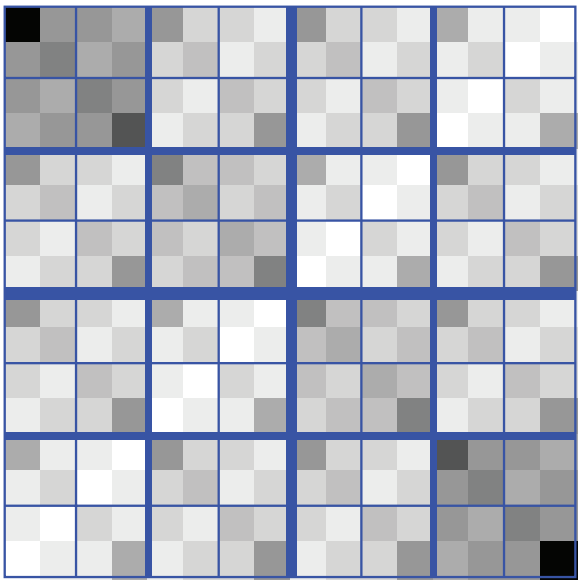

Figure 3. HIFF, the hierarchical-if-and-only-if function, shown over an eight-dimensional space. HIFF is a recursively defined function whose schemata comprise strings of either all zeros or all ones. Thus HIFF has multiple local optima, which in HSGP lie along the top left to bottom right diagonal. In the software tool, the local optima are highlighted in green and lie along the diagonal, clearly showing that there are $2^{n/2}$ of them. The global optima are at the points 0000 0000 and 1111 1111 and are at opposite corners of the space. The basins of attraction for all local optima show a similar recursive structure, and all are of size $3^{n/2}$. Such a pattern indicates that search algorithms based on hillclimbers will find low peaks just as frequently as high ones, indicating that it would not be an appropriate search strategy (in fact, HIFF was designed precisely to thwart hillclimbers, and HSGP provides a visual method to develop the intuitions that complement analytic techniques).

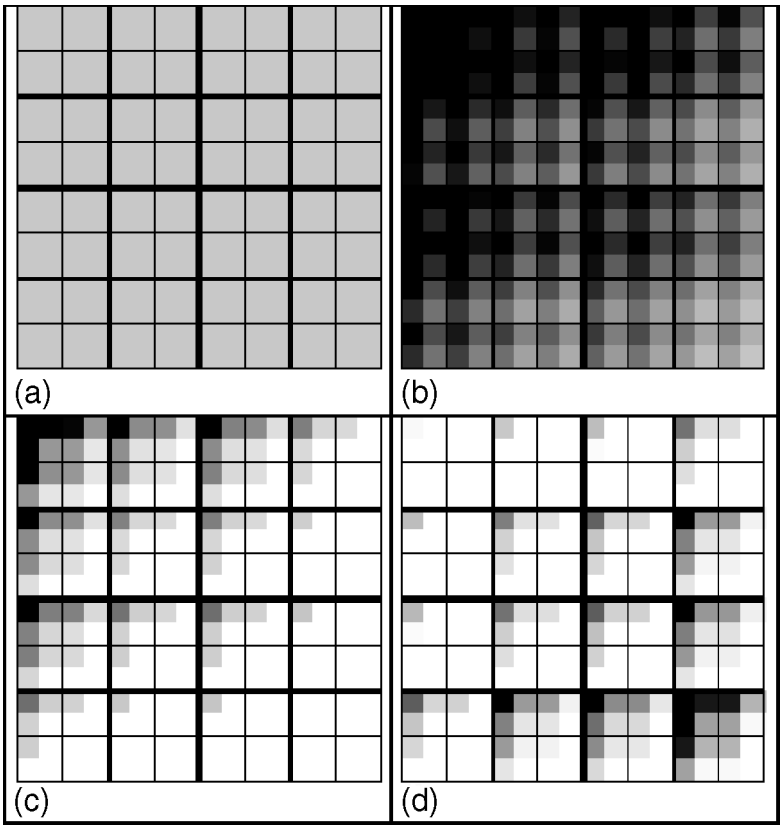

Figure 4. Performance of a search algorithm, showing PBIL's progress in optimizing HIFF. PBIL estimates the probability density function of the landscape as a linear combination of all dimensions. It iteratively adjusts its estimate towards the best in a sample of solutions. Each hypergraph shows the probability distribution function (pdf) as constructed by PBIL. (a) The initial uniform distribution. (b, c) As the algorithm progresses, structure begins to emerge, so that by the final hypergraph, one of the two global optima (0000 0000) is clearly the most likely solution predicted by PBIL. Note, however, that as the algorithm progresses, the substructure of HIFF is not reflected in the structure of the pdf. (d) An alternative run of PBIL over HIFF shows it converging on a suboptimal solution (1111 1100). As in (c), the substructure of the fitness function is not modeled by the pdf.

valley floor, and NKq, which discretizes a continuous space, and the visualizations show how increasing levels of neutrality affect each model of neutrality.

HSGP has also been used to study the interaction of particular search algorithms with landscape structure. The simplest search techniques to visualize are hill-climbing algorithms. Using HSGP to visualize an entire space shows more complex interactions. For example, the probability-density-based function PBIL [8, 17] is shown estimating the function for HIFF in Figure 4. Although HIFF has recursive substructure, any particular run of the PBIL algorithm only estimates one peak. All search algorithms have an inherent search bias, which can be explored through visualization even without a detailed understanding of the algorithm. This use of visualization allows algorithm designers to tune their intuitions about which algorithm to choose for a given application and why.

Acknowledgments

This research was supported by grants from the School of Psychology and Australian Research Council to the first author. Thanks to Cathy Mackie, who helped with collection of data, and Stephen Wong, who programmed an early prototype of the hypergraph layout. The research was begun while the first author was in the Psychology Department and the second author was in ITEE at the University of Queensland.

References

1. Wiles, J., & Tonkes, B. (2003). Mapping the Royal Road and other hierarchical functions. *Evolutionary Computation*, *11*(2), 129–149.
2. Karnaugh, M. (1953). The map method for synthesis of combinatorial logic circuits. *Transactions of the AIEE*, *72*(1), 593–598.
3. Hill, F. J., & Peterson, G. R. (1968). *Introduction to switching theory and logical design*. New York: Wiley.
4. Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1974). *The design and analysis of computer algorithms*. Reading, MA: Addison-Wesley.
5. Samet, H. (1984). The quadtree and related hierarchical data structures. *ACM Computing Surveys*, *16*(2), 187–260.
6. Collins, T. D. (1997). Using software visualization technology to help evolutionary algorithm users validate their solutions. In T. Baeck (Ed.), *The Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA'97)* (pp. 307–314). San Mateo, CA: Morgan Kaufmann.
7. Shine, W., & Eick, C. (1997). Visualizing the evolution of genetic algorithm search processes. In *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)* (pp. 367–372). Piscataway, NJ: IEEE Press.
8. Wiles, J., & Tonkes, B. (2002). Hyperspace Graph Paper: Visualising interactions between search algorithms and landscapes. In T. M. C. Smith, S. Bullock, & J. Bird (Eds.), *Beyond Fitness: Visualizing Evolution—Workshop Proceedings of the Eighth International Conference on Artificial Life* (pp. 177–182). UNSW.
9. Tonkes, B. (2002). Hyperspace graph paper Web site, http://www.itee.uq.edu.au/~btonkes/hsgp.html. The University of Queensland.
10. Van Nimwegen, E., Crutchfield, J. P., & Mitchell, M. (1999). Statistical dynamics of the Royal Road genetic algorithm. *Theoretical Computer Science*, *229*, 41–102.
11. Watson, R. A., & Pollack, J. B. (1999). Hierarchically-consistent test problems for genetic algorithms. In P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, & A. Zalzala (Eds.), *Proceedings of 1999 Congress on Evolutionary Computation (CEC 99)* (pp. 1406–1413). Piscataway, NJ: IEEE Press.
12. Pelikan, M., & Goldberg, D. E. (2000). *Hierarchical problem solving by the Bayesian optimization algorithm* (IlliGAL report no. 2000002). Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign.
13. Holland, J. H. (2000). Building blocks, cohort genetic algorithms, and hyperplane-defined functions. *Evolutionary Computation*, *8*(4), 373–391.
14. Kauffman, S. A. (1993). *The origins of order: Self-organization and selection in evolution*. Oxford, UK: Oxford University Press.
15. Newman, M., & Engelhardt, R. (1998). Effect of neutral selection on the evolution of molecular species. *Proceedings of the Royal Society of London Series B*, *256*, 1333–1338.
16. Barnett, L. (1998). Ruggedness and neutrality—The NKp family of fitness landscapes. In C. Adami, R. Belew, H. Kitano, & C. Taylor (Eds.), *Proceedings of the Sixth International Conference on Artificial Life* (pp. 17–27). Cambridge, MA: MIT Press.
17. Baluja, S. (1994). *Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning* (CMU-CS-94-163). Pittsburgh, PA: Carnegie-Mellon University.