Cognitive Approaches to the Traveling Salesperson Problem: Perceptual Complexity that Produces Computational Simplicity

Bradley J. Best

Micro Analysis & Design
4949 Pearl E Circle, Suite 300
Boulder, CO 80301
bbest@maad.com

Abstract

The Traveling Salesperson Problem (TSP) is a classical problem in complexity theory that has provided significant motivation for the development of that theory. The TSP has also shown that exact solutions come at a high price: exponential time complexity. As a result, heuristic solutions to the TSP are often the only practical means for dealing with this intractable problem. Heuristic solutions are a hallmark of human cognition, although most heuristic solutions to the TSP share little inspiration with human cognition. This is unfortunate, since human solutions to the TSP have proven to be both accurate and rapid. Understanding human approaches to the TSP holds the promise to inform the development of heuristic approaches to computationally hard problems. This report explores and compares cognitive mechanisms implicated with human solutions to the TSP and their impact on the complexity and accuracy of the solution. In particular, the impact of Euclidean clustering is compared to a distorted clustering distance measure, and the efficacy of solving TSPs in a breadth-first manner is compared with the efficacy of solving TSPs in a depth-first manner. The empirical evidence indicates that distorted clustering and breadth-first solution development are superior to the tested alternatives.

Computational Complexity and the TSP

The Traveling Salesperson Problem (TSP) consists of attempting to find the shortest complete tour through a series of points (cities), starting and ending with the same point (see Figure 1). This problem is a member of the set of computationally hard, or NP-complete, problems, that can only be solved in polynomial time by a nondeterministic Turing machine. A naïve approach to solving the TSP, enumeration of all of the possible tours and selection of the best of those, results in time complexity proportional to $n!$ (there are $\frac{1}{2}(n-1)!$ total tours to consider). Although dynamic programming methods improve on this somewhat, producing a solution in time complexity that is exponential, there are no known polynomial-time algorithms for the TSP (Papadimitriou, 1995).

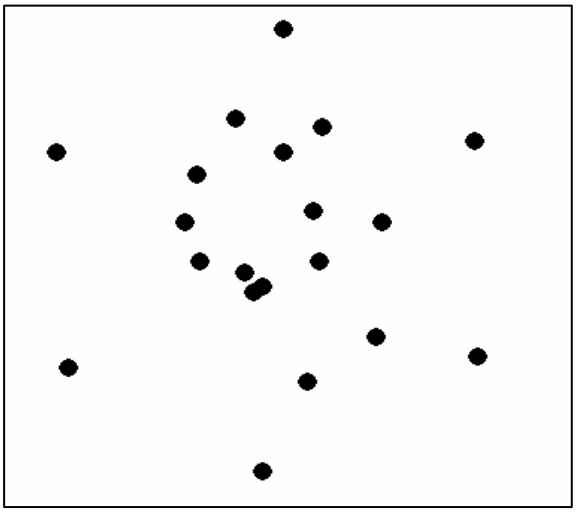

Figure 1: A Traveling Salesperson Problem

Parallel computation is often touted as the answer to computationally hard problems. However, since the actual computational work achieved by a parallel system – which must be exponential for a computationally hard problem – is a product of the number of processors multiplied by the time taken in processing, the time complexity of a parallel processing solution must come through either exponential time complexity (which represents no gain), or through the employment of a number of processors that is exponentially related to the size of the input. Even in the massively parallel human visual system, the number of processing elements involved in processing stimuli is proportional to the area of that input due to retinotopic mapping of visual cortical areas, and therefore scales in polynomial terms (Hubel, 1988).

The employment of parallel computation in conjunction with fast processors, clever exponential algorithms, and elegant programming techniques may represent the best we can hope for. However, this paper advances an alternative: It may be more profitable to spend greater efforts in understanding clever *heuristic* algorithms, particularly those based on human cognition and applied to computationally hard problems, where the relative lack of

processing speed and the intractability of the problems produces a forcing function that drives problem decomposition and simplification in ways that traditional AI methods rarely emulate.

This echoes a position expressed by Newell (1990, p.107), that the scientific objective of AI is to understand the relationship of knowledge and search in problem solving, and further that this need can only be fulfilled by studying approaches in domains that fall on both ends of the spectrum: brute-force search, and knowledge guided problem solving. The position expressed here, however, goes beyond that by encouraging AI scientists to think of cognitive mechanisms themselves as knowledge: they are a form of (heuristic) processing which is used advantageously to reduce search as much as possible, or even to eliminate it entirely. That is, there is no search for the proper knowledge to apply – the knowledge is built into cognitive mechanisms that are applied automatically. This type of analysis may be particularly beneficial in domains that are known to be resistant to search techniques (i.e., computationally hard problems).

Heuristic Solutions to the TSP

The TSP is both extremely easy to understand and frustratingly hard to solve efficiently. As a result, this problem has been at the core of research programs in many different disciplines including Computer Science, Psychology, and Operations Research (OR). This tradition provides a wealth of potential heuristic approaches to choose from and a ready store of data about the merits of those approaches. OR algorithms, in particular, have been well-cataloged in terms of complexity and algorithmic description. The following three algorithms, as documented by Golden, Bodin, Doyle, & Stewart (1980), will be useful for comparison purposes:

- Nearest Neighbor algorithm: starting with a point picked at random, connect it to its nearest unattached neighbor until no points remain. The overall complexity is $O(N^3)$.
- Cheapest Insertion: starting with a city and its nearest neighbor, iteratively insert the next city that produces the lowest increase in length. The overall complexity is $O(N^3 logN)$.
- Convex Hull with cheapest insertion (CCI): starting with the convex hull as an initial tour, remaining cities are inserted according to a least cost heuristic. The overall complexity is $O(N^2 logN)$.

The key question will be whether these mechanisms appear in human solutions, and how the human (cognitive) solutions compare in terms of complexity. The convex hull, in particular, has been implicated in human solutions by MacGregor & Ormerod (1996), Ormerod & Chronicle (1999), and Best (2004).

Human Solutions to the TSP

Best (2004) presented a range of TSP problem types to human solvers, including the 10-point and 20-point problems used by MacGregor & Ormerod (1996), sets of randomly constructed 10-point, 20-point, and 30-point problems, and a set of shaped problems with specific contours (intended to be 'atypical' in shape to expose specific perceptual processing). The performance of human solvers on these problem sets, in percentage deviation from the optimal solution, is presented in Table 1. In general, the human solvers produced solutions that were of consistently high quality.

Table 1: Quality of Human Solution by Problem Type (% above optimal ± SD).

Problem type	Human Performance
MacGregor & Ormerod (1996) 10 Point	2.7%±1.7%
MacGregor & Ormerod (1996) 20 Point	8.2%±2.9%
Random 10 Point	1.7%±1.2%
Random 20 Point	4.1%±3.0%
Random 30 Point	5.0%±1.1%
Shaped	3.7%±2.5%

Another, potentially more significant, aspect of the human solutions to TSPs presented in Best (2004) was the consistently rapid solution of the problems which was approximately *linearly* proportional to the number of points in the problem. Figure 3 shows the amount of time

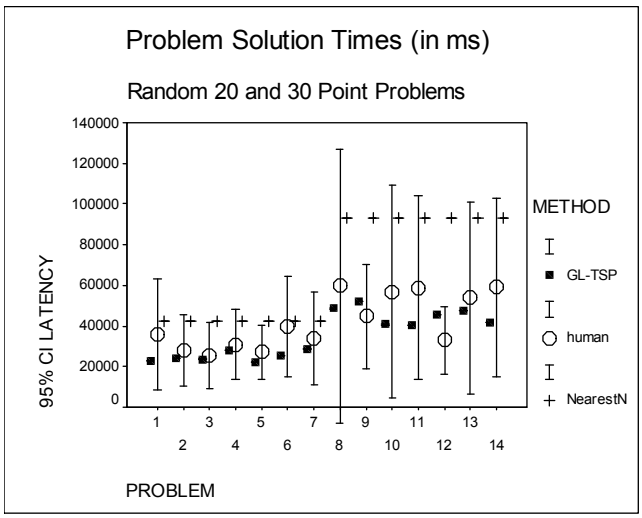

Figure 2: Human TSP Solution Times Compared to Heuristic Algorithms

taken, in milliseconds, for human participants to complete problems in the Best (2004) study that were either 20-point problems (numbers 1-7) or 30-point problems (numbers 8-14). This key finding – that human solutions scale approximately linearly in time – also replicates similar findings by Graham, Joshi, & Pizlo (2000). In addition,

the figure displays latency predictions made by a cognitive model of human performance on the task which scales similarly, and the nearest neighbor algorithm, which does not. The next section will explore cognitive models of TSP solution in more depth.

Cognitive Models of Human Solutions

A latent assumption in this work is that the human mind approximates a limited capacity computational device, and that if we were to determine the actual algorithm and data used, a program written on modest contemporary hardware would be able to attain the same level of performance. The following algorithms, though by no means an exhaustive listing of work in the field, overlap in significant ways and as a result will be examined in more detail:

Global-Local TSP (Best, 2004). This algorithm employs the convex hull as an initial plan for solution, and works from coarse to fine by leveraging a clustering algorithm based on the CODE theory of human perceptual clustering (Compton & Logan, 1993).

Pyramidal Algorithm (Graham, Joshi, & Pizlo, 2000). This algorithm employs a hierarchical structure based on Gaussian filters that also leverages the clustering structure of a TSP. Problem solution proceeds from the top down, decomposing the problem and inserting newly decomposed sections into the already complete initial tour.

Commonalities Extracted from Cognitive Models

The combination of cognitive models, heuristics, and solution algorithms discussed here employ a range of mechanisms that aid with efficient solution of TSPs. The following points are particularly salient:

- Bottom-up organization (perception) of the problem into a more meaningful whole.
- Top-down solution (coarse to fine processing) that employs problem decomposition.
- Use of outside contours (e.g., the convex hull) in guiding rough solutions.
- Insertions in a partial tour.
- Search of a solution space that is highly pruned relative to the complete solution space.
- Perceptual grouping that does not depend on a strict Euclidean metric and is instead distorted.

Most of these items are commonplace and are broadly accepted as a means of organizing solutions. The last item, however, is particularly 'cognitive' in its emphasis and stands out from the others. Though many cognitive mechanisms are initially assumed to be an impoverished form of processing, many of the apparent limitations of cognition turn out to have key functional benefits. This possibility will now be examined in greater detail.

An Empirical Comparison of Clustering Methods as the Basis for TSP Algorithms

Clustering methods are of particular interest in the case of applying cognitive methods to solving the TSP. First, there is good evidence that the human visual system can accomplish clustering through feed-forward, parallel visual processing, and that this processing involves grouping that takes place between local elements (e.g., Han, Humphreys, & Lin, 1999). Second, clustering methods are computationally well-understood and conceptually simple. Third, exploring clustering allows us to study the bottom-up influences on successful top-down problem-solving. Fourth, and finally, similar clustering mechanisms have been employed in various models of the TSP, presenting a unique affordance to compare and contrast these mechanisms on a task for which they are highly relevant.

The following section will examine the interaction of clustering methods with problem solutions that either progress in a breadth fashion (globally working at progressively deeper levels of detail) or a depth fashion (locally working deeper before working deeper on non-local parts of the problem. Each of these methods can thus be viewed as a tree traversal of the hierarchical cluster tree where visited nodes are placed into a tour of clusters, culminating in a tour where the nodes corresponding to the individual points in the problem.

Euclidean Clustering vs. Warped Space Clustering

The clustering methods used here are strictly agglomerative clustering (Hastie, Tibshirani, & Friedman, 2001), a method of hierarchical clustering. In conventional agglomerative clustering, items are initially assigned to clusters consisting only of themselves, after which the clusters are combined pairwise in order of closest geometric proximity, using a Euclidean distance metric to determine the next pair to be clustered, until a tree is assembled that culminates in a single root cluster containing all of the other clusters. As clusters are combined, the parent cluster location is determined by weighting the child clusters according to the number of items they represent (parent cluster locations represent a center of mass).

Several theories of human perceptual grouping, on the other hand, employ a more flexible conception of grouping where the grouping is subject not just to strict Euclidean measures, but also to non-local effects and weighting that takes other groups into account (e.g., Compton & Logan, 1993; Best & Gunzelmann, 2005). In the current work, the distortion in clustering space is accomplished by dividing distances from one cluster to another by the total distance from the first cluster to all of the other clusters (i.e., by dividing by a measure of the dispersion of the whole display).

Figure 3 shows the hierarchical grouping structure that was induced from the TSP presented in Figure 1 using Euclidean agglomerative grouping. In this figure, the solid circles represent groups with the radii of the circles representing the number of individual points accounted for, while the straight lines connect siblings that are collected together at the next higher level.

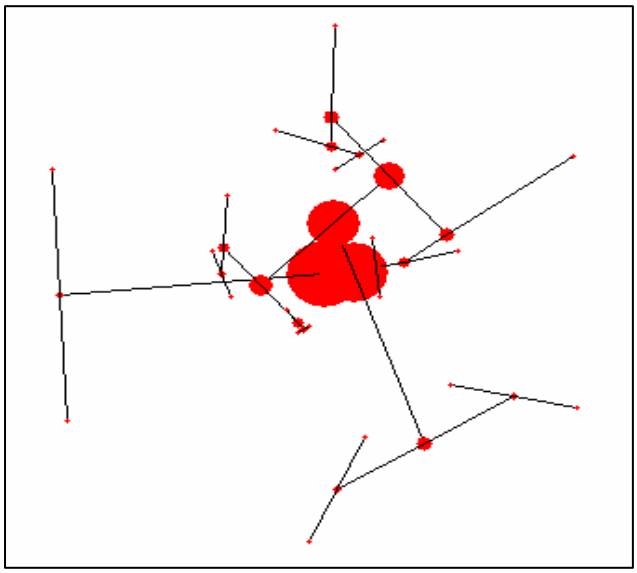

Figure 3: Hierarchical Cluster Structure for a TSP

The first comparison of interest is a comparison of the efficacy of solution given clusters that are Euclidean versus given a distorted (and arguably more cognitively plausible) perceptual space. Figure 4 shows a comparison of the coarse-to-fine cluster traversal algorithm when using normal Euclidean space against the same algorithm employing a weighted distance metric for cluster agglomeration. The algorithm was run on the four data sets reported above regarding human performance on the TSP. In general, the cluster-based representation

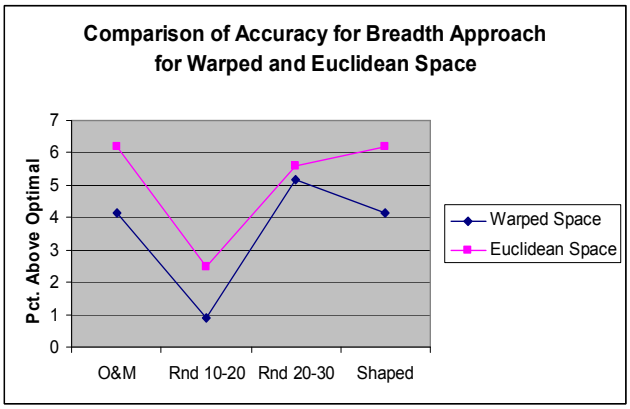

Figure 4: Effects of Euclidean and Warped Space

supported solutions that were similar in quality to human solutions. The warped space representation results in superior performance on every one of these data sets when compared to a standard Euclidean distance measure (note that the algorithm is not stochastic, so these are point predictions).

Depth-First vs. Breadth-First Problem Solution

The goal of the second comparison is to determine the utility of the hierarchical clustering structure as a representation for solving the TSP. As a tree structure, there are two immediately obvious methods of traversing the structure: depth-first and breadth-first. A depth-first use of the tree in solving the TSP can be accomplished by greedily exploiting the larger clusters, leaving stragglers for the end of the problem-solving episode. Exploration in depth is typical of human problem-solving (particularly methods like progressive deepening; e.g., Gobet, 1997), though it is not immediately obvious if that method is applicable to organizing a solution to a visual problem such as the TSP while using a hierarchical cluster tree. Breadth-first use, on the other hand, involves starting with the tree nodes closest to the root, and planning routes through those clusters before moving to finer levels of detail (this approach appears to be extremely similar to the Pyramidal algorithm of Graham, Joshi, and Pizlo, 2000).

Given the advantage displayed for warped space, and the interest in achieving good performance, the depth versus breadth comparison was made only for the warped distance clustering measure. The results for the warped space are

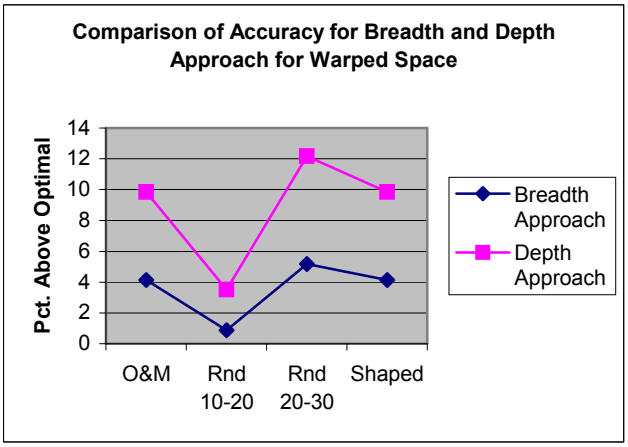

Figure 5: Impact of Breadth of Solution

displayed in Figure 5. The breadth approach had an advantage over the depth approach across problem sets (and again, since the models are not stochastic these are exact point predictions). In general, the magnitude of error displayed by the depth approach was consistently more than twice that of the breadth approach.

Complexity of Cluster-Based Solutions. These solutions have two major components: (1) the cost of generating the hierarchical structure, and (2) the cost of traversing the tree to produce the solution. The second of these is clearly the efficient aspect of a cluster-based solution – insertions are

executed in a very limited problem space. The algorithm described here operates on a tree where the only choices made are between a single pair of permutations at each local decision. For example, in the tour A-B-C, if node B is selected for expansion, it will consist of at most two sub-nodes (which can, of course, have their own sub-nodes). Therefore the choice is between expanding to A-D-E-C or A-E-D-C, where the choices to be made are constrained by the hierarchical cluster data structure. Thus, given an existing hierarchical cluster data structure (i.e., assuming (1) above is already complete), there are two decisions to be made per node, and since the structure is a binary tree, for n input items, there will be $2^{\log n} - 1$ decisions to make.

Although this analysis provides no relief from the computationally hard nature of the problem, it does suggest how people may be accomplishing the task in *apparent* linear time. In particular, given that the visual system is massively parallel across the visual field, and that clustering is a primitive operation (i.e., distributed across those processing elements), the time cost of human solutions may potentially only be on the order of $2^{\log n}$ (for problems that do not exceed the limitations of that system).

Clustering Using Warped Space

Less formally, illustrative examples of the problems faced by algorithms in this domain may also give the reader an intuitive sense why clustering using warped space provides an advantage. It appears that clustering in Euclidean space provides more variable results, though in the same range. Upon closer inspection, interesting patterns started to appear among the inferior solutions. In particular, inferior solutions based on Euclidean space clustering tended to have a crossed arc form early in the solution, and due to the very local exploration of the space later in the solution, the algorithm was unable to backtrack and undo the early crossing. Figure 5 shows the solution development process for a particular problem for the warped space (left column) and Euclidean space (right column). The initial decomposition is flawed for the Euclidean clustering space in a way that the algorithm never sufficiently recovers from. Despite making good local decisions after that, the best solutions in that part of the solution space are still hugely inferior.

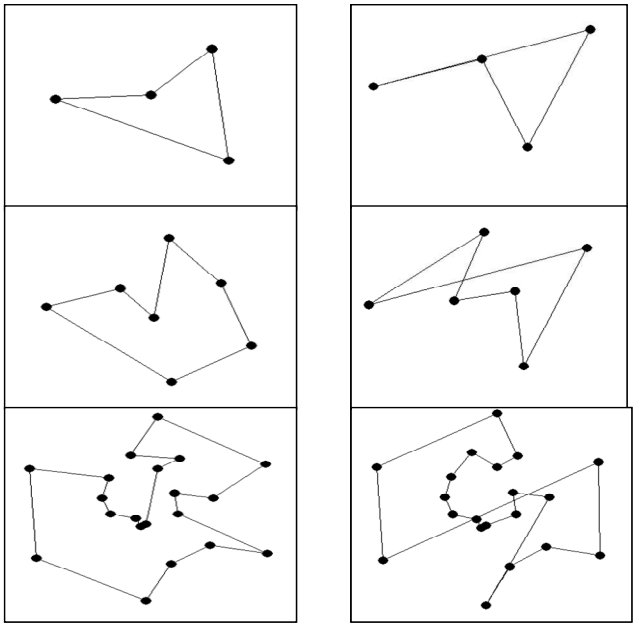

Figure 6: Warped Space Solution (L) and Euclidean Space Solution (R)

Sufficiently Accurate Global Information

One takeaway from the prior discussion is that bad early commitments may be difficult or impossible to undo. The worse the 'global' information, the better the backtracking capability needs to be, while the better the global information, the more the algorithm can rely on (fast) local search mechanisms. That is, the way a problem is partitioned at the top of the tree appears to be more critical than the way it is partitioned closer to the bottom – by the time the solution process gets that far, choices are limited by the constraints of the previous solution, so they can neither be that bad nor be that good. The gain in quality of TSP solution, if it is to be made, is to be made early in the solution process.

On the other hand, the better global information comes from a process that is no more complex than the inferior global information (i.e., the two clustering algorithms barely differ by a single line of code – while the output is different, their structure is nearly identical). The information that comes from the warped clustering algorithm is, in fact, a distortion of the information present, yet it is superior in problem solving because it allows a more advantageous global structure to emerge.

Conclusions

In Best (2004) and Graham, Joshi, and Pizlo (2000), the time complexity of human TSP solution was roughly linearly proportional to the number of points. This surprising fact demands explanation. Part of the explanation is clearly that human solutions are heuristic solutions, not perfect solutions. However, people appear to make a more profitable tradeoff between speed and accuracy than many of the heuristic algorithms that attempt to do the same. People appear to be leveraging general-purpose perceptual mechanisms that give them access to a cluster-based representation augmented by strongly suggestive contours (e.g., Han, Humpreys, & Chen, 1999).

Very Specific Solutions to Very General Problems

Computer scientists are often encouraged to avoid coming up with solutions that are so tailored to a specific problem that they are good for little else. The case of the TSP, however, provides a compelling counterpoint. From a complexity theory perspective, approaches that yield tractable solutions for one NP-complete problem yield

tractable solutions for all of them. From a cognitive science perspective, it is unlikely that people have special-purpose TSP solution knowledge or neural apparatuses, so studying the TSP affords an opportunity to expose general purpose cognitive capabilities which are nonetheless extremely well suited for this domain. From a more practical (applied) perspective, the TSP is itself a very abstract task, and though solutions to it may be 'special purpose', as specific solutions to a general problem they may be much more readily applied to other problems than typical special purpose algorithms. As a case in point, several of the algorithms discussed here have been applied in practical domains to problems such as robotic route planning (Best, Lebiere, & Gacy, 2005).

Mechanisms Must Have Constraints

Limited processing is the rule in studies of human problem solving. Work on modeling the TSP can be viewed as a search for limited mechanisms that still provide accurate and rapid problem solution, with a particular emphasis on avoiding overkill. As Newell (1990, p. 253) pointed out in regards to the potential for the Soar cognitive architecture production matching mechanism to solve TSPs:

> Thus, if there are M elements in the working memory, a production with K conditions can be forced to consider of the order of Mk possibilities for instantiation, satisfying a fixed but arbitrarily large set of equality tests among the possibilities. Indeed, this matching scheme is sufficiently powerful to be NP-complete (Garey & Johnson, 1979), meaning (for instance) that one can encode into the matching of a single (horrendous) production the requirement to solve a traveling salesman problem.

Clearly, Newell is not arguing for Soar production matching as a reasonable or viable algorithm for solving TSPs, but rather is cautioning us against seeking out overly powerful mechanisms. The goal of the current work is not to find a single, monolithic, algorithm that solves a TSP. Rather, it is to decompose algorithmic solutions in the hope of understanding the structure of approximate solutions to the TSP, and to isolate functional aspects of those algorithms through a process of comparison and combination of mechanisms. Hopefully, this work is the first of many steps in that direction.

Future Work

There are many unanswered questions posed by the current work, several of which can be addressed within the framework constructed for this report. First and foremost is the issue of scalability: How do approaches based on hierarchical clustering fare as problem size increases? Second is the issue of the perceptual reality and validity of distorted grouping and the various clustering constructs. Third, and final, is the continuation of a program to deploy algorithms capable of handling computationally intractable problems into real, applied settings.

References

Best, B. J., 2004, *Modeling Human Performance on the Traveling Salesperson Problem: Empirical Studies and Computational Simulations.* Doctoral dissertation, Department of Psychology, Carnegie Mellon University, Pittsburgh, PA.

Best, B. J., Lebiere, C., and Gacy, M. (2005). Leveraging a Cognitive Architecture to Help a Robot Walk and Chew Gum at the Same Time. In *Proceedings of the 12th Conference on Behavior Representation In Modeling and Simulation*.

Best, B. J., and Simon, H. A., 2000, Simulating Performance on the Traveling Salesman Problem. *Proceedings of the 2000 International Conference on Cognitive Modeling*, pp. 42-49.

Compton, B. J., and Logan, G. D., 1993, Evaluating a computational model of perceptual grouping by proximity. *Perception & Psychophysics*, 53(4), pp. 403–421.

Gobet, F., 1997, A pattern-recognition theory of search in expert problem solving. *Thinking and Reasoning*, 3, pp. 291-313.

Graham, S. M., Joshi, A., and Pizlo, Z., 2000, The traveling salesman problem: A hierarchical model. *Memory & Cognition*, 28, pp. 1191-1204.

Han, S., Humphreys, G. W., and Lin, C., 1999, Parallel and Competetive Processes in Hierarchical Analysis: Perceptual Grouping and Encoding of Closure. *Journal of Experimental Psychology: Human Perception and Performance*, 25(5), 1411-1432.

Hastie, T., Tibshirani, R., Friedman, J., 2001, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag.

Hubel, D., S., 1988, *Eye, Brain, and Vision*. New York: W. H. Freeman and Company.

Kubovy, M., and Wagemans, J., 1995, Grouping by proximity and multistability in dot lattices: a quantitative gestalt theory. *Psychological Science*, 6(4), pp. 225–234.

MacGregor, J. N., and Ormerod, T., 1996, Human performance on the traveling salesman problem. *Perception and Psychophysics*, 58, pp. 527-539.

MacGregor, J. N., Ormerod, T., and Chronicle, E. P., 2000, A model of human performance on the traveling salesperson problem. *Memory & Cognition*, 28(7), pp. 1183-1190.

Michie, D., Fleming, J. G., and Oldfield, J. V., 1968, A comparison of heuristic, interactive and unaided methods of solving a shortest-route problem. *Machine Intelligence*, 3, pp. 245-255.

Newell, A., & Simon, H. A., 1972, *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

Ormerod, T. C., and Chronicle, E. P., 1999, Global perceptual processing in problem solving: The case of the traveling salesperson. *Perception and Psychophysics*, 61, 1227-1238.

Papadimitriou, C., H., 1995, *Computational Complexity*. Reading, MA: Addison-Wesley.