A Model of Fast Human Performance on a Computationally Hard Problem

Bradley J. Best (bbest@maad.com)
Micro Analysis & Design, 4949 Pearl E. Circle, Suite 300
Boulder, CO 80301 USA

Abstract

Human performance on the Traveling Salesperson Problem (TSP) is of consistently high quality and scales approximately linearly in time with problem size. A model leveraging parallel processing of perceptual grouping and a local serial search achieves both a comparable quality of performance and comparable time complexity.

Human Performance on the Traveling Salesperson Problem

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. This problem is a member of the set of computationally hard, or NP-complete, problems, for which the best solutions known are obtained in exponential time relative to the problem size.

Michie, Oldfield, and Fleming (1968) performed one of the earliest studies of human performance on the Traveling Salesperson Problem (TSP). They found that human performance approached, and in the case of one individual, exceeded, that of a specialized graph traversal algorithm designed for solving search problems.

MacGregor and Ormerod (1996) described a set of experiments designed to test the hypothesis that the difficulty of a TSP is due to the number of points falling on the interior of the problem as opposed to those falling on the convex hull, or outer contour of the problem. In general they found that human performance was, in fact, less good on those problems with more points in the interior.

Ormerod and Chronicle (1999) conducted experiments to determine whether human solvers of the TSP were sensitive to the global contour of individual problems, and found evidence confirming this hypothesis.

Best and Simon (2000) described a solution method based on a spatially local search of alternative paths that attained the quality of human solutions with low computational expense. Graham, Joshi, and Pizlo (2000) developed a fully parallel solution algorithm for the TSP based on a hierarchical processing approach and proposed it as a candidate model of human performance on the task. MacGregor, Ormerod, and Chronicle (2000) advanced a different model of human performance on the task, suggesting that the human solution process starts with the convex hull, and iteratively adds points to the solution by comparing costs among the remaining (unused) points, and selecting the lowest cost insertion.

Figure 1 displays a TSP with the convex connected by arcs and with one insertion of an interior point (point 3) completed. Optimal solutions must connect points on the convex hull in order, so for problems such as this one where the majority of the points fall on the convex hull, the convex hull provides a good basis for problem solution.

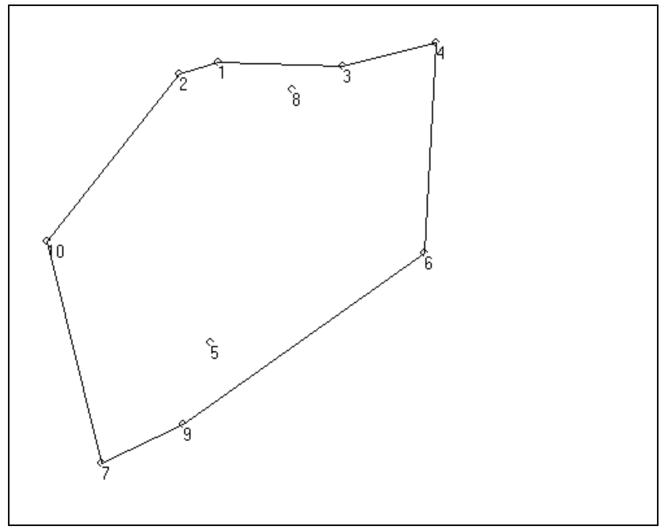

Figure 1: Traveling Salesperson Problem with convex hull

To allow for discrimination between the various alternative theories of human performance on the TSP, Best (2004) conducted a set of experiments collecting fine-grained performance data describing human performance on the TSP. In addition to recording the quality of individual solutions, detailed latency data were also collected. These data encompassed all interactions with the task interface at the level of individual mouse movements recorded at the time resolution of the operating system in a computer version of the TSP task. Significant findings included effects of problem size on accuracy and latency, individual differences on accuracy and latency, and a distinct pattern of latency of movement within problems that provided insight into the process used by solvers. The remainder of this paper presents a summary of the Best (2004) studies.

Human solvers were presented with blocks of TSP problems of the following types: 1) the problem set from MacGregor and Ormerod (1996) consisting of 10 and 20 point problems, 2) 10 and 20 point problems constructed from a uniform random distribution, 3) 20 and 30 point problems constructed from a uniform random distribution, and 4) problems with definite contours (e.g., points selected along intersecting lines). Solvers connected points in the order of their choosing but were not permitted to backtrack. The performance of human solvers in percentage deviation from the optimal solution is presented in Table 1.

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%

In addition to variation of problem type, solvers were also presented with two other manipulations: 1) a repeated block of 20 point problems to determine the impact of learning (the improvement was non-significant; see Table 2), and 2) an interface manipulation where the problem was blurred except for the area immediately around the mouse pointer. The intention of this manipulation was to examine the detrimental effects of obscuring the global display information. Surprisingly, performance was actually better in this condition (though non-significantly; see Table 3).

Table 2: Quality of human solution on 20 point problems in blocks 1 and 2 (% above optimal ± SD).

Random 20 Point Block 1	Random 20 Point Block 2
4.1%±3.0%	3.1%±1.3%

Table 3: Quality of human solution for random 10 and 20 point problems using the normal and obscured interface (% above optimal ± SD).

Problem Set	Normal	Obscured
Random 10 Point	1.7%±1.2%	0.7%±0.6%
Random 20 Point	4.1%±3.0%	2.7%±2.0%

Human solutions were also characterized in terms of latency to complete solutions (Table 4).

Table 4: Latency of human solution by problem type (time in seconds ± SD).

Problem type	Human Performance
MacGregor & Ormerod (1996) 10 pt	23.8s±16.2s
MacGregor & Ormerod (1996) 20 pt	39.6s±19.4s
Random 10 Point	15.2s±5.6s
Random 20 Point	29.3s±10.3s
Random 30 Point	52.3s±33.6s
Shaped	16.8s±6.7s

Besides accuracy for complete solutions, the resolution of the data permits examining latency for moves within a particular solution (i.e., for each move from the first move to the last move). Figure 2 shows a graph of the human latency performance for individual moves for random 10 point problems (similar results are obtained for other problem types but are omitted here due to space limitations).

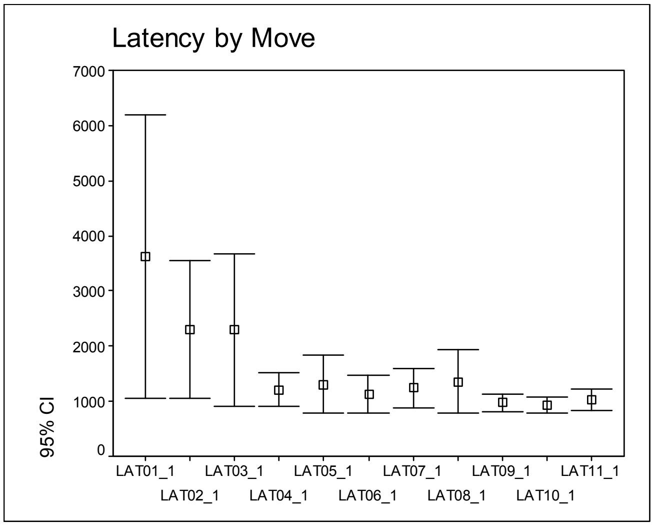

Figure 2: Latency in ms for moves 1-11 during 10 point random problem solution (with standard error bars).

The mouse movement data from the task was analyzed to provide a more complete picture of the TSP solution process. The initial moves during problem solution were characterized by a large number of individual mouse strokes, representing exploratory movements around the problem (Figure 3 presents results for random 20 point problems; other problem types produced consistent results). Although it is normal to covertly shift attention without moving the mouse, the converse is unlikely, and these individual mouse movements can be interpreted as shifts of attention. This indicates that there is significant scanning of the problem during the initial stages of solution, and rules out a purely "pop out" explanation of human performance.

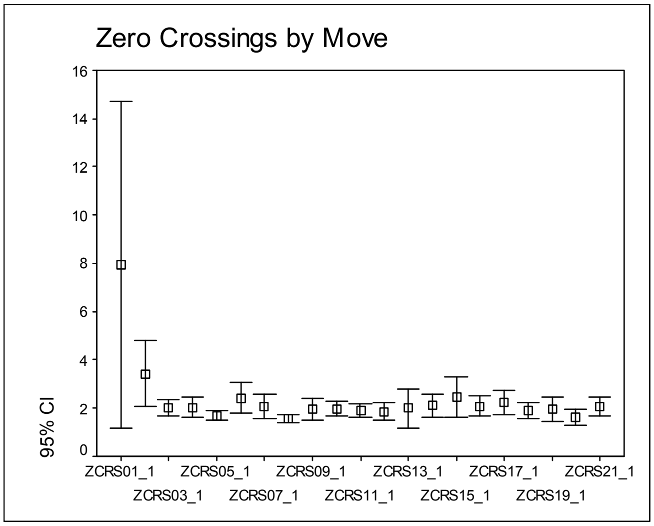

Figure 3: Individual mouse movements taken for moves 1-21 during problem solution for 20 point random problems (with standard error bars).

Summarizing the results of Best (2004), human performance during the solution process is characterized by the following findings: 1) Problem solution times are approximately linearly proportional to the number of problem points, with individual moves taking ~1.5 seconds; 2) Solution accuracy is generally within ~5% of optimal path length for the majority of solvers and problems; 3) Accuracy is lower on more complex problems where complexity is determined by factors including the number of points in a problem, problem shape, and the number of interior (non-convex hull) points; 4) Problem solving is preceded by 2-3 seconds during which the problem is scanned and a rough solution is developed; 5) Planning requires only low-frequency spatial information provided by a blurred display; 6) The pattern of mouse movement involved initial travel that was indirect, consisting of multiple exploratory movements, while beyond the first two moves, mouse movements become more direct and there is a reduction of distance traveled and time and strokes taken; 7) Evidence for learning is minimal indicating that the strategies and processes used by human solvers are innate or well-practiced, and therefore general (e.g., perceptually based, weak methods, etc.).

Implications for Modeling Human Performance

The experimental data presented above demonstrates that the human performance scales roughly linearly with problem size. This implies that the human solution is likely to rely on either parallel processing, or a locally constrained serial search that only considers a constant sized subset of the problem as solutions progress, or some combination of these. In fact, the data is sufficiently detailed to rule out all of the aforementioned computational accounts of human performance on the problem.

The model of Best and Simon (2000) does not consider the contour of the entire problem (the convex hull) to formulate a rough solution. The data collected by Best (2004) indicates that human solvers do, in fact, start with a rough solution, while Ormerod and Chronicle (1999) demonstrated that human solvers were both sensitive to and used the convex hull in judgments of TSP solution quality.

The model of Graham, Joshi, and Pizlo (2000) does not proceed with a serial solution process the way human solvers do. The similarity between human solutions using the obscured interface in Best (2004) with the normal interface indicates that human solvers are not simply playing out a complete solution that was developed prior to working on the problem, but rather are working interactively with the problem in a serial fashion.

The model of MacGregor, Ormerod, and Chronicle (2000) does not consider just the (locally) relevant points at a particular point in the solution process. However, mouse-tracking results from Best (2004) show only local movements later in problem solving, thus ruling out this model

Using Constraints from Human Performance to Develop a Computational Model

The constraints from empirical studies of human performance on the TSP suggest that a computational model of human performance should incorporate the following elements: 1) a global, parallel process that produces the convex hull of the problem, 2) a hierarchical clustering process that decomposes the problem into sub-problems for local solution, 3) a serial method of local search that considers a roughly constant-sized set of candidate points, and 4) a perceptual method for avoiding premature closure of paths (which would produce crossed arcs). The remainder of this paper describes the construction and evaluation of such an algorithm, designated the GL-TSP (Global-Local TSP solver).

Mechanisms for Clustering and Contour Perception

Compton and Logan (1993) described an extension to the CODE theory of clustering which was developed to account for human behavior in grouping (clustering) of dot diagrams. Although related to theoretical approaches such as Kubovy and Wagemans (1995), and computational approaches based on hierarchical clustering methods, this theory provides an added advantage by describing a 'strength of grouping' surface which allows the direct calculation of goodness of Figure (Best, 2004).

The CODE theory specifies the construction of a 'strength of grouping' surface where the value of the surface at any point is given by the sum of exponential intensity distributions centered about the problem points. Each of these distributions has a standard deviation proportional to the distance to the nearest point. A two-dimensional cross section of an example surface produced by three points is shown in Figure 4 while a three dimensional depiction of a CODE surface is shown in Figure 5.

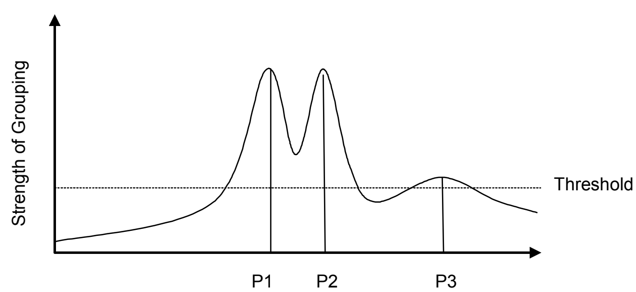

Figure 4: Cross section of strength of grouping surface corresponding to three points.

Goodness of Figure Calculations The calculation of goodness of Figure can be accomplished by using the 'strength of grouping' surface produced by the algorithm the CODE theory is based on. By inspecting the shape of the surface between any two points, a determination of goodness can be produced through calculating the number of distinct bumps (i.e., zero crossings of the derivative). For the purposes of this algorithm, any two points that can be reached across the surface without crossing any additional bumps are defined as a 'good' path and is perceptually valid. An example of a TSP displaying the good paths obtained using this method is presented in Figure 5 (the paths are superimposed over the three dimensional CODE surface which is displayed in relief).

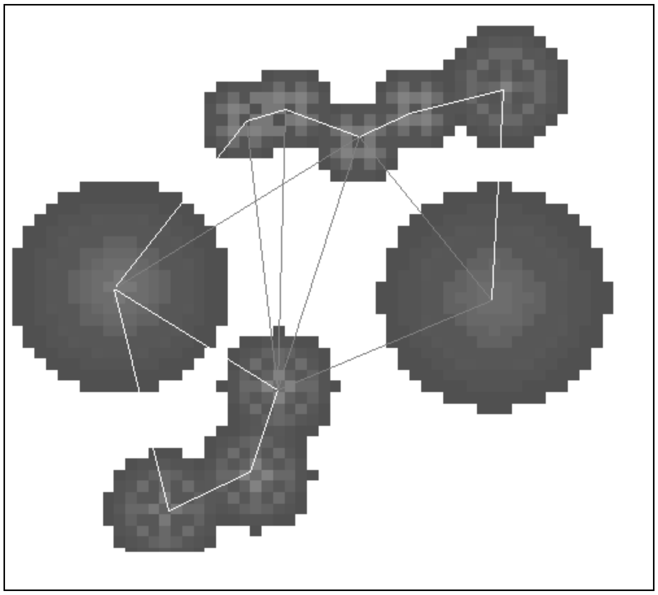

Figure 5: Good paths (perceptually valid paths) extracted from strength of grouping surface.

Mechanisms for Local (Serial) Search

Recasting problem solving as search through a problem space has a long history within problem solving research (e.g., Newell and Simon, 1972). More recent efforts, such as Gobet (1997) and Gobet and Simon (1996), have carefully specified the constraints and limitations of search as a human problem solving process. One of the primary findings of this research is that pruning and selective exploration play a significant role in reducing the number of computations performed by human solvers. That is, human solvers do not perform exhaustive search, but instead perform a more limited search of the better options in the search space.

Human solvers of the TSP appear to have two main methods for reducing the computational complexity of the search space. One of these methods is the hierarchical decomposition of the overall problem into smaller problems, which is achieved by clustering and forming a rough overall plan without considering details. The other method is the dependence on good paths in the representation of the problem which reduces the number of branches at each point in the search space. Although the total number of TSP solutions possible is equal to the factorial of the number of points, the number of perceptually valid paths is substantially less than that, and allows search to proceed in a remarkably smaller subspace.

Integrating Local and Global Processes into a TSP Solver

The GL-TSP algorithm integrates a global level that produces a rough solution to the problem, and a local level that searches through the local section of the rough global plan.

The global plan is initially constructed from the convex hull of the clusters produced by the CODE algorithm. However, this may leave out some number of clusters from inclusion in the plan, and, if so, these clusters must be integrated prior to the implementation of the local solution (that is, deciding whether a cluster will be included in the portion of the local solution being worked on must be completed prior to working on that section). These orphan clusters are included by inserting the cluster between a pair already included in the plan that minimally increases the overall path length (subject to noise in distance estimation). These insertions are rarely necessary in problems of 10 nodes, since most points fall within a cluster that makes up the hull of clusters, but become an important determiner of overall solution shape for problems of 20 and more nodes. Thus, the global plan can be considered a convex hull cheapest insertion algorithm performed on the clusters instead of on individual points. The result of the global planning process is an ordering of the clusters, while ordering of individual points is left up to the local stage of the algorithm.

At each local stage of the algorithm, a path must be planned through a set of potential nodes. This set of nodes is created by adding nodes from the current cluster in the global plan. If the set then contains less than six nodes, nodes are added from the next cluster in order from the global plan. If the set still contains less than six nodes, nodes are added from the third cluster in order. If, at this point, there are only three nodes in the set (i.e., each of the clusters was a single point), nodes from a fourth cluster are added. This results in a set of, at most, six nodes, and at least, four nodes (except when there are no nodes remaining in the problem). The GL-TSP algorithm establishes a goal to reach the last node in this set (the farthest along in the global plan), and establishes subgoals to visit all of the intervening nodes. It then uses the operation of traversing individual perceptually valid paths to reduce the difference from the goal state, and finds a solution path that reaches the target node. Multiple solution paths are evaluated in terms of their complete distance (subject to noise in estimation), and the first node along the chosen path is selected by connecting that edge. The local stage of the algorithm then repeats (without saving its previous result) and replans from the new current node.

The local search is conducted according to the following pseudocode:

1) Set the cluster index to the current cluster (cluster 1)
2) Add nodes from the indexed cluster until either:
    a) The planning set contains six nodes
    b) The planning set contains all remaining nodes

3) If (the planning set contains less than six nodes and the cluster index is < 3)
4) or (the planning set contains three nodes) then:
    a) Increment the cluster index
    b) Go to step 2

The selection of an endpoint for the local search differentiates this method from hill-climbing. The path selected is the shortest path, given the endpoint. Shorter paths are possible among the set of points. Further, the individual step taken need not be the shortest: it is simply the first step along the shortest path to the chosen endpoint and may be longer than alternative choices. This local search can be characterized overall as an optimal shortest-path solver, made stochastic by perceptual noise, and limited to searching the representation provided by the clustering algorithm.

Model Evaluation

The GL-TSP algorithm described here makes an explicit accuracy prediction based on the solutions obtained for individual problems, and an explicit latency prediction for each individual problem based on the time taken to compute a global plan, the local hierarchical decomposition, and the available perceptually good paths within the local part of the problem. The algorithm uses noise in distance estimation to produce stochastic behavior, so algorithm results are produced by aggregating multiple runs.

A latency prediction for solving each problem was constructed by adding an estimate for constructing the global plan at 300ms per node of the problem (approximately 3 seconds for a 10-point problem), plus 900ms for every untraversed perceptually 'good' path emanating from the current node (approximately 3 eye fixations to measure the path), plus 300ms to actually make the move. These estimates allow direct comparison between the latency pattern that arises from the basic operations of the GL-TSP algorithm and the pattern of latencies demonstrated by human solvers. Since all variation of predictions of latency for problems of the same size is due solely to the number of available perceptually good paths (for a given problem size, the latency associated with constructing a global plan and making moves is constant), the latency prediction of this model is that more available options will require serial consideration and thereby slow the decision time.

For the purpose of comparison, the Nearest Neighbor algorithm also generates latency predictions. In this case, the prediction is a multiple of the number of algorithmic comparisons made allowing comparison to the GL-TSP algorithm in terms of computational steps taken.

Figure 6 shows latency predictions for problems consisting of 10 and 20 randomly distributed points while Figure 7 shows latency predictions for problems consisting of 20 and 30 randomly distributed points. The latency predictions are presented in the Figures with actual human performance on the task, showing standard error bars for the human data. In general, the latency results for the GL-TSP algorithm are consistent with human performance, with the GL-TSP latency estimate falling within the confidence interval for human performance on almost every individual problem. It is especially noTable that the GL-TSP latency predictions scale almost exactly with the human performance, while the Nearest Neighbor method does not.

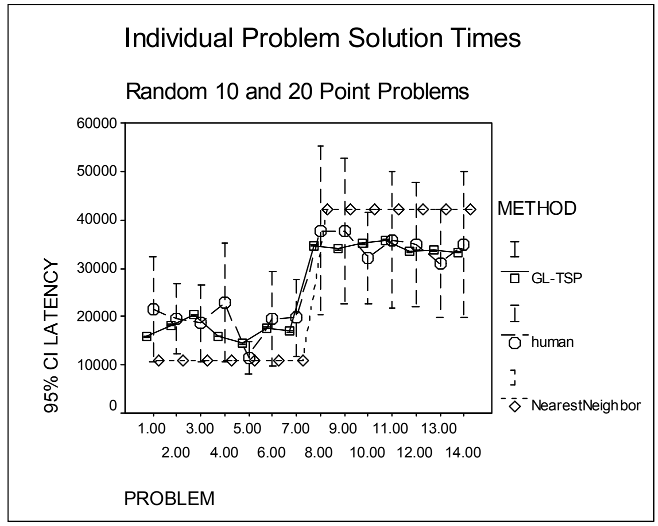

Figure 6: Latency of human performance compared to the Nearest Neighbor algorithm and the GL-TSP algorithm for 10 and 20 point problems with standard error bars.

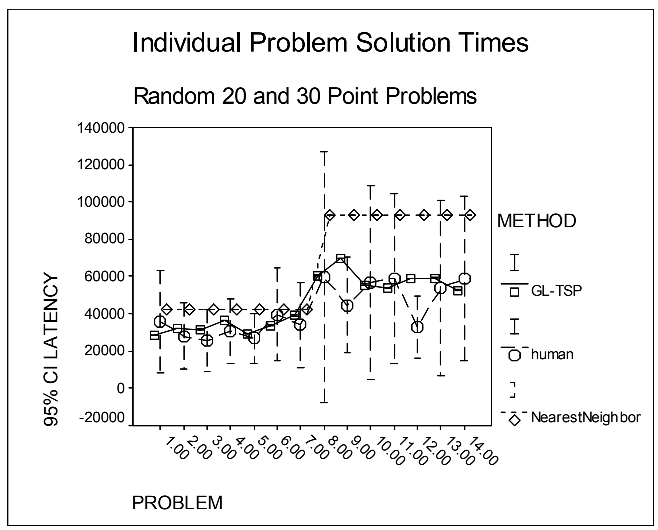

Figure 7: Latency of human performance compared to the Nearest Neighbor algorithm and the GL-TSP algorithm for 20 and 30 point problems with standard error bars.

Since these estimates are based on the actual number of comparisons made by the serial level of the model, this is a demonstration of the feasibility of achieving a roughly linear time complexity performance through leveraging a parallel system that performs clustering (as well as 'goodness of path' evaluations). The Nearest Neighbor algorithm, and in fact, any algorithm that considers all available remaining candidate points at each step, does not scale in the same way as either the human performance or the GL-TSP algorithm.

Although the latency results are consistent with human performance, this would be meaningless unless the quality of the performance also corresponded to human results. The representative model performance is presented in Table 5.

Table 5: Quality of human solution compared to quality of algorithm solution (% above optimal ± SD).

Problem type	Human	GL-TSP
MacGregor & Ormerod (1996) 10	2.7%±1.7%	1.1%±1.8%
MacGregor & Ormerod (1996) 20	8.2%±2.9%	3.9%±5.3%
Random 10 Point	1.7%±1.2%	0.2%±1.0%
Random 20 Point	4.1%±3.0%	5.2%±7.6%
Random 30 Point	5.0%±1.1%	7.4%±7.7%
Shaped	3.7%±2.5%	2.6%±3.8%

Looking at accuracy in the aggregate, the GL-TSP algorithm slightly outperforms the mean human performance, but falls within the demonstrated range of human performance.

Conclusion

This report summarizes the work presented by Best (2004) describing the development of a computationally instantiated theory that describes human performance on the Traveling Salesperson Problem. This theory is supported by a wide variety of empirical constraints and evidence that simultaneously argue against existing models of human performance on the task. It is composed of a parallel process that performs hierarchical clustering and contour detection, and a serial problem-space search process that performs local search along the rough solution plan. The serial portion of the algorithm is subject to substantial processing constraints that allow only a limited amount of processing. These limitations are overcome by leveraging a parallel perceptual implementation that allows the serial portion of the model to focus on comparing relatively good options, rather than exhaustively comparing all options. This theory explains both the quality of human performance on the TSP task and the latency performance of human performance on the TSP task. In particular, it provides a compelling account of the roughly linear scaling of human solution times when comparing solutions of varying sizes.

Although this theory is presented as a specific theory for solving TSPs, it is composed of two independent portions that are significantly more general and have been validated in other domains: a perceptual front end that produces a hierarchically clustered representation of dot patterns and good paths connecting the dots, and a general search mechanism that works on a problem space representation and traverses the problem state graph in pursuit of a solution.

Acknowledgments

The work described here was completed at Carnegie Mellon University as a doctoral thesis (Best, 2004) under the advising of John Anderson and Herb Simon. The complete data and models for this project are publicly available and are archived at http://act.psy.cmu.edu/ftp/models/tsp/.

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., 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.

Gobet, F., and Simon, H. A. (1996). The roles of recognition processes and look-ahead search in time-constrained expert problem solving: Evidence from grandmaster level chess. *Psychological Science*, 7, pp. 52-55.

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

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.