Traveling Salesman Problem: A Foveating Pyramid Model

Zygmunt Pizlo,[1] Emil Stefanov,[1] John Saalweachter,[1] Zheng Li,[1]
Yll Haxhimusa,[2] and Walter G. Kropatsch[2]

Abstract

We tested human performance on the Euclidean Traveling Salesman Problem using problems with 6–50 cities. Results confirmed our earlier findings that: (a) the time of solving a problem is proportional to the number of cities, and (b) the solution error grows very slowly with the number of cities. We formulated a new version of a pyramid model. The new model has an adaptive spatial structure, and it simulates visual acuity and visual attention. Specifically, the model solves the E-TSP problem sequentially by moving attention from city to city, the same way human subjects do. The model includes a parameter representing the magnitude of local search. This parameter allows modeling individual differences among the subjects. The computational complexity of the current implementation of the model is $O(n^2)$, but this can most likely be improved to $O[n\log(n)]$. Simulation experiments demonstrated psychological plausibility of the new model.

Introduction

In this paper we discuss the mental mechanisms involved in solving the Traveling Salesman Problem (TSP). TSP refers to the task of finding the shortest tour of *n* cities given the intercity distances (costs). When the distances between cities are Euclidean, the problem is called E-TSP (Graham, Joshi, & Pizlo, 2000). A simple way to present E-TSP to a subject is to show *n* points on a computer screen and ask the subject to produce a tour by clicking on the points. TSP (including E-TSP) belongs to the class of difficult problems called NP-hard.[1] Difficult means that finding an optimal solution (i.e., the shortest tour) may require, in the worst case, performing an exhaustive search through all tours. Because the number of tours for an *n*-city problem is equal to $(n - 1)!/2$ and, thus, grows very quickly with the number of cities *n*, determining lengths of all tours is usually impossible. For this reason, TSP is called computationally intractable. Because of the intractability of TSP, researchers interested in problem solving concentrated their efforts

[1]Department of Psychological Sciences, Purdue University, West Lafayette, IN 47907-2081, USA
[2]Institute of Computer Aided Automation, Vienna University of Technology, Favoritenstraße 9/183-2, A-1040 Vienna, Austria

Address correspondence to: Zygmunt Pizlo, Telephone: (765) 494-6930; Facsimile: (765) 496-1264; Email: pizlo@psych.purdue.edu

http://dx.doi.org/10.7771/1932-6246.1009

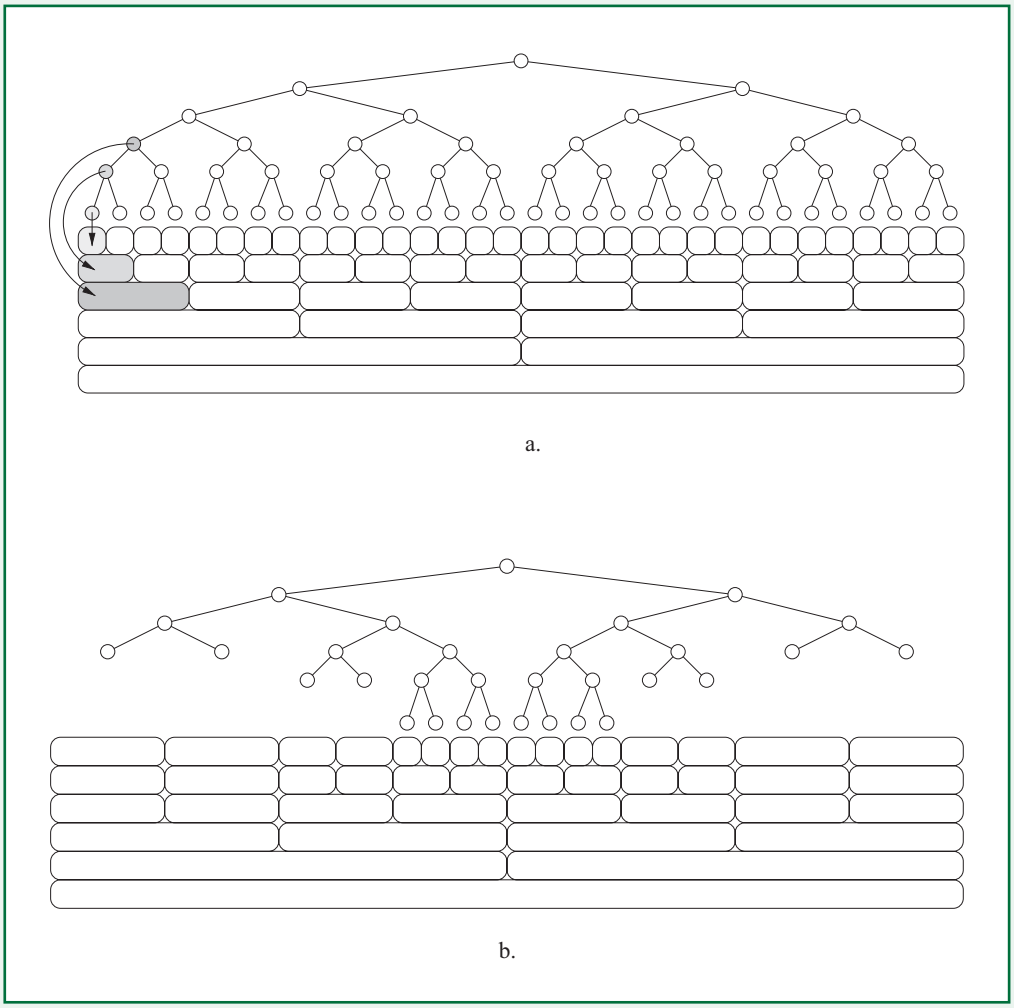
on finding approximating algorithms—that is, algorithms that can produce near-optimal solutions fairly quickly. Good approximating algorithms can produce solutions that are only a few percent longer than an optimal solution, and the time of solving the problem is a low-order polynomial function of the number of cities (Christofides, 1976; Lawler, Lenstra, Rinnooy Kan & Shmoys, 1985; Gutin & Punnen, 2002).

Interestingly, humans are known to produce close-to-optimal solutions to E-TSP problems in time that is (on average) proportional to the number of cities (Graham et al., 2000; MacGregor & Ormerod, 1996; MacGregor, Ormerod, & Chronicle, 2000; Pizlo & Li, 2004; Pizlo, Stefanov, Saalweachter, Haxhimusa and Kropatsch, 2005; Vickers, Butavicius, Lee, & Medvedev, 2001). That is, the tours produced by the subjects are, on average, only a few percent longer than the shortest tours, and the solution time is a linear function of the number of cities. The two main attempts to emulate human performance by a computational model were undertaken by Graham et al. (2000) and by MacGregor et al. (2000). The present study directly derives from that of Graham et al. (2000). In particular, the model presented in this paper is an elaboration of that of Graham et al.

Graham et al.'s attempt to formulate a new approximating algorithm for E-TSP was motivated by the failure to identify an existing algorithm that could provide a good fit to the subjects' data. The main aspect of Graham et al.'s (2000) model was its (a) (multiresolution) pyramid architecture and (b) coarse-to-fine process of successive tour approximations. They showed that performance of this model (proportion of optimal solutions and average solution error) is statistically equivalent to human performance. Pyramid algorithms have been used extensively in both computer and human vision literature (e.g., Jolion & Rosenfeld, 1994; Pizlo et al., 1995) but not in problem solving. The work of Graham et al. was the first attempt to use pyramid algorithms to solve TSP (Pizlo, Joshi, & Graham, 1994; Graham, Pizlo, & Joshi, 1995, 1996). One of the most attractive aspects of pyramid algorithms, which makes them suitable for problems such as early vision or E-TSP, is that they allow the solving (approximately) of global optimization tasks without performing a global search. Shortly after Graham et al. formulated their model, Arora (1998) described a pyramid algorithm for producing approximate E-TSP solutions. Arora was not interested in modeling human performance but, rather, in formulating an algorithm that allows for trading computational complexity (speed) for error of the solution (accuracy; see also Gutin & Punnen, 2002, for a description of Arora's results).

The next (second) section presents an overview of the new model, called the Foveating Pyramid (FP) model. A psychophysical experiment on E-TSP is described in the third section. Fitting FP to the results of the five subjects is described in the fourth section. The fifth section concludes the paper. The appendix has a pseudo-code, as well as estimated worst case complexity of the new model.

Figure 1.
(a) One-dimensional pyramid. Receptive fields of the individual nodes are indicated in the bottom by the oval shapes. (b) Pyramid with a fovea. Sizes of the smallest receptive fields are not uniform across the image. This is similar to the nonuniform distribution of the visual acuity in the human visual system.

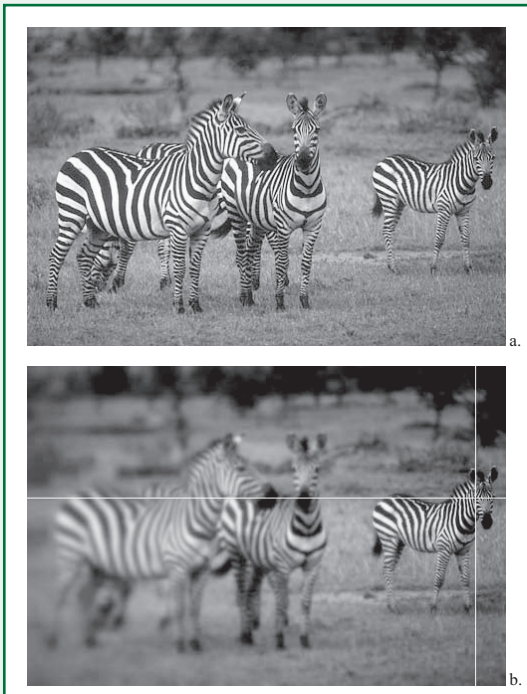


Foveating Pyramid (FP) Model

Figure 1 illustrates two versions of a one-dimensional pyramid model. The pyramid shown in Figure 1a is conventional in the sense that the first (bottom) layer has representation of the entire input (e.g., an image). The pyramid shown in Figure 1b is called foveating because it simulates a nonuniform distribution of visual acuity. In the human visual system, the visual acuity is inversely proportional to the distance from the center of the retina, fovea (Pizlo, 1988). This dependence of visual acuity on the distance from the fovea is related to the nonuniform density of receptors on the retina. The nonuniform density of receptors allows the visual system to avoid handling too much visual

information at one time but to still have a high resolution within the fovea and a quite large field of view in the periphery. The foveating pyramid model has similar properties. It is a conventional pyramid in the sense of having multiple representations of the image, the representations being different in the sizes of the receptive fields and their resolution. Unlike conventional pyramids, where every layer has information about the entire image, in the new pyramid the highest resolution representation (at the first layer) has information about only a small part of the image around its fixation point. The representation on the second layer has information about a larger part of the image, but the information is characterized by a lower resolution than the first layer and so on. This kind of foveating pyramid for computer vision applications was presented by Burt (1988). In the implementation described by Burt, the lower resolution representations were produced by Gaussian blurring. Burt's model is illustrated in Figure 2, where the fixation point is marked by the center of a large cross. The regions that are farther from the fixation point are blurred proportionally more. Note that the image in Figure 2 is not intended to accurately characterize the distribution of visual acuity. For a quantitative relationship, see Pizlo (1988).

Figure 2.
Illustration of the effect of nonuniform distribution of visual acuity. The fixation point in (b) is set on the eye of the baby zebra on the right (the center of the white cross). Notice the blurring effect compared to the original image (a). (The image in [a] was taken from the Berkeley Image Database [Martin et al., 2001] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/.)

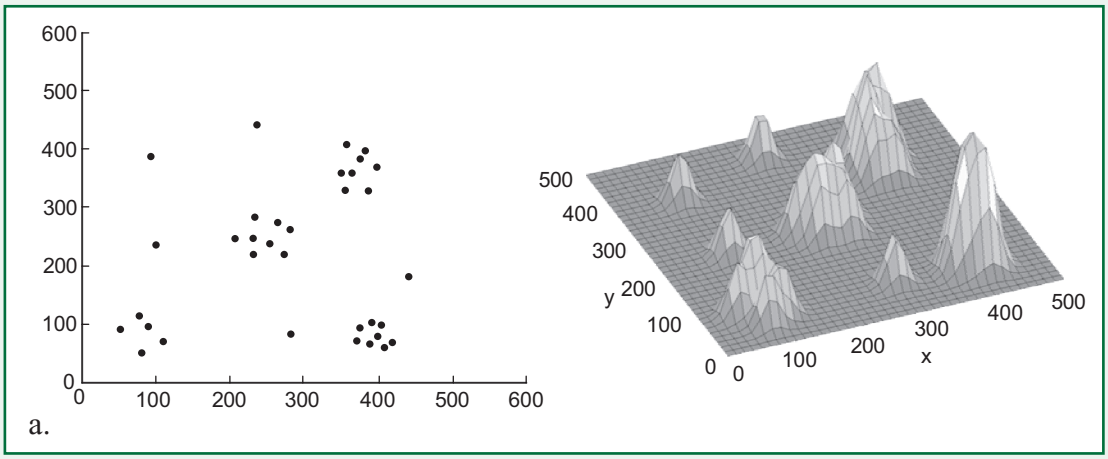


When a human subject produces a solution to an E-TSP problem by working successively on individual parts of the problem, the size of the attentional window is not necessarily fixed. It is known that a human observer may attend to a small part of the visual field when the details are important. Alternatively, she may attend to a large part of the visual field when the global aspects are of interest (Pizlo et al., 1995). Our pyramid model has similar properties. The choice about which part of the E-TSP problem to analyze and how large this part should be is established in a top-down process in which the global aspects are used to guide decisions on more local representations.

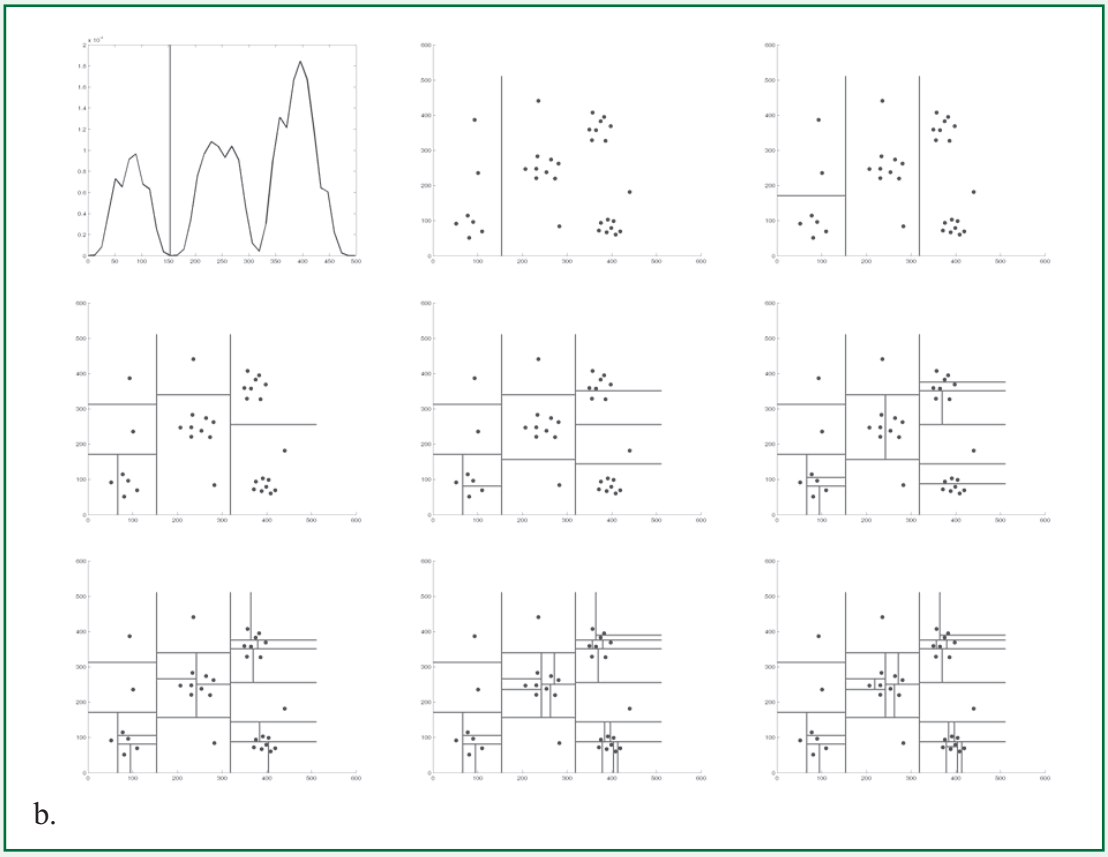
Before the solution process begins, the FP forms a hierarchical representation of the problem. The goal of building this representation is to identify clusters of cities on many levels of resolution, as well as to establish spatial relations among the clusters. This is done in a top-down process called bisection, which is illustrated in Figure 3 . Figure 3a shows an E-TSP problem where several clearly defined clusters are present. Figure 3b shows the intensity distribution after the problem was blurred by a Gaussian filter (the size of the Gaussian filter is not critical). Then, for each position on the X-axis, the maximum intensity $I_{maxY}(X)$ along the Y direction is found. Similarly, for each position on the Y-axis, the maximum intensity $I_{maxX}(Y)$ along the X direction is found. Then, a minimum of each of these two distributions is determined, min $I_{maxY}(X)$, min $I_{maxX}(Y)$, and the smaller of the two corresponds to the boundary between two clusters on the coarsest level (see Figure 3b, top left and top center panels).[2] Then, the process of blurring (with proportionally smaller filters) and bisecting is repeated as shown on the remaining panels of Figure 3b. The bisection process ends when each city is in its own cluster. After the pyramid representation is built, the solution process begins. (See Appendix for a pseudo-code and complexity of the algorithm).

The solution process starts at the top layer, *h,* in which the entire problem is represented by two clusters. The initial tour involves just these two clusters. More exactly, the tour involves centers of gravity of the cities within clusters. The starting city is chosen randomly. Then, the tour is refined recursively within the receptive fields that contain the starting city. As a result, the tour is represented on many layers of the pyramid; different parts of the tour "residing" on layers having different spatial resolution. The cities in the fovea are represented with the highest resolution, and the cities that are farther away from the fovea are represented with resolution that falls off gradually. This is analogous to the blurring effect illustrated in Figure 2. After the starting city and its immediate neighborhood is described at the highest resolution, the simulated fovea is moved to the next city (clockwise or counterclockwise) on the existing tour, and the part of the tour "ahead" of the current city is projected down to layers with higher resolution so that a gradual transition between resolutions is maintained (the direction of the solution, clockwise vs. counterclockwise, is determined randomly). When a part of the tour

Figure 3.
(a) On the left is a 34-city problem with four clusters. On the right is an intensity distribution produced by Gaussian blurring. (b) The top-down (coarse-to-fine) process of determining cluster boundaries.



a.



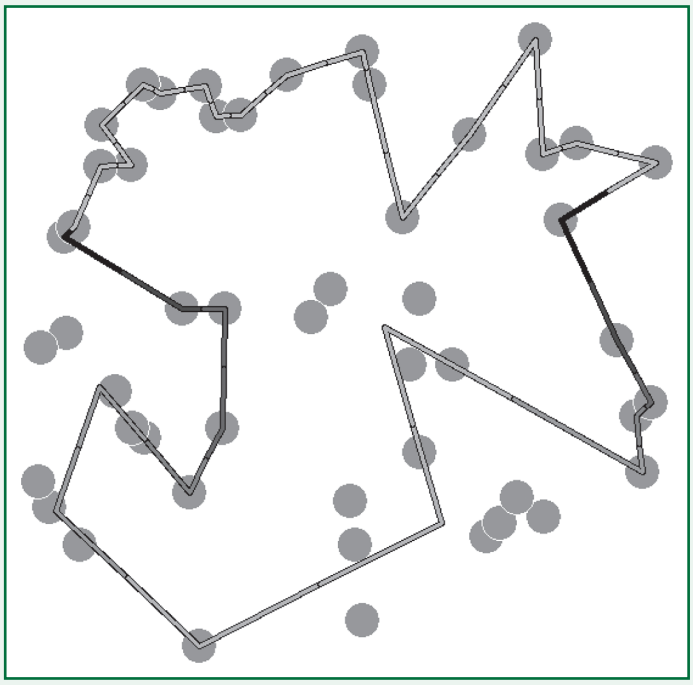
b.

on a layer *j* is projected to layer *j*–1, new clusters emerge. These clusters are inserted into the existing tour by using cheapest insertion (Lawler et al., 1985). Specifically, the new cluster (which is treated as a city) is inserted between pairs of consecutive cities in the existing tour, but only those pairs that are within *k* cities of the current city in the existing tour are tried. Thus, the parameter *k* represents the magnitude of local search involved in solving TSP. The values of *k* in the range between 0 and 8 were used. Larger values of *k* correspond to greater amounts of local search, which is likely to lead to better solutions.

At this point, the reader is encouraged to view the video illustrating the solution process (psych.purdue.edu/tsp/files/animation/Sample_BisecionPyr_50City.htm). The video shows the process of solving a 50-city E-TSP problem by the foveating bisection pyramid model. In this case the model produced an optimal tour. In the demo, the parts of the tour that are shown in green correspond to the highest spatial resolution. So the green appears first around the point from which the model starts solving the problem. The other parts of the tour have different shades of gray: The lighter the gray, the coarser the resolution. In the regions where the spatial resolution of the problem representation is coarse, the tour connects centers of gravity of cities rather than individual cities. As the fovea gets closer to a given cluster, the cluster gets subdivided into smaller clusters, and the tour is modified so as to go through the centers of gravity of the new clusters.

A snapshot of the solution process is shown in Figure 4. A small part of the problem is already solved (the part of the tour on the top), and the rest is not. Only a coarse approximation is available at this stage for large parts of the problem.

Figure 4.
A snapshot from the solution process by the bisection foveating pyramid.



To summarize the main aspects of the FP model: The FP solves an E-TSP problem in a top-down process of successive approximations, as the Gaussian pyramid model did (Graham et al., 2000). The main difference is that FP uses much less information about the problem at any given time because only a small part of the problem is described with the highest resolution at any given time. In order to produce a solution to an E-TSP problem, the FP puts its fovea at one city and then simulates the movement of the fovea around the problem so that it can successively "see" all parts of the problem. The movements of the FP's fovea are decided by a lower resolution representation of the E-TSP problem. The tour modification on the higher resolution layers is performed by a cheapest insertion method, which is restricted to a local region of the tour. Despite the fact that the "visual attention" of the foveating model is spatially limited, the solution tours are equally short as those produced by the Gaussian pyramid model of Graham et al. (2000). The computational complexity of the FP is, in the current implementation, $O(n^2)$ (see Appendix). This means that the time required to solve any given problem is, in the worst case, a quadratic function of the number of cities. The average time cannot be longer than that, although it may be shorter. The reader should keep this fact in mind when psychophysical results are discussed. The experimenter cannot evaluate the worst-case computational complexity of the mental algorithm. The experimenter can only measure an average complexity.

Psychophysical Experiment: Human Performance on E-TSP

In order to test the psychological plausibility of the foveating pyramid model, we performed an experiment in which subjects solved E-TSP problems. These psychophysical results were then used to fit the model to the subjects' data.

Subjects

Five subjects, including three authors were tested. The other two subjects (OSK and BSL) were naive about the hypotheses being tested. ZP and ZL received extensive practice in solving E-TSP problems (ZP served as a subject in Graham et al.'s study, as well). OSK and BSL solved only a few problems before they were tested. Ethical approval was obtained from the Institutional Review Board at Purdue University.

Stimuli

Problems with 6, 10, 20, and 50 cities were used (25 randomly generated configurations of points were used for each number of cities). The problems were presented on a computer screen. The cities were represented by dots in a window of size 512×512 pixels.[3]

Procedure

Each subject solved the same problems in a random order. The subject was instructed to produce a tour by clicking on the cities. Specifically, he was asked to produce as short a tour as possible. The subject knew that the time of solutions was recorded, but the speed of solution was not emphasized. The subject was free to start at any city. He could undo the last move (recursively) or start over. The time and position of each click, as well as the solution tour, were recorded. The subject's performance was evaluated by the average time per city, the proportion of optimal solutions, and the average error of solution. The error was computed as the difference between the length of the tour produced by the subject and the length of the shortest tour, normalized to the latter. Thus, an optimal solution had an error of zero.

Results and Discussion

Results are shown in Figure 5. Overall, the results are very similar to those reported by Graham et al. (2000). It can be seen in Figure 5a that there is no systematic effect of the number of cities on the average time per city. This means that the average solution time is proportional to the number of cities. This fact suggests that the computational complexity of the mental mechanisms is quite low, perhaps even linear—O(n). Note, however, that the complexity of the mental mechanisms might actually be higher for two reasons. First, we cannot evaluate the worst case complexity in the case of human subjects; we can only measure average solution time. Because average time is not greater than the maximal time, it follows that the results shown in Figure 5a provide a lower bound for the worst-case complexity of the mental mechanisms.[4] Second, one has to include the contribution of the hand movements. Specifically, it is known that the movement time is proportional to the logarithm of the movement size (when the target size is constant). This is referred to as Fitts' law (Fitts, 1954). When the number of cities is larger and the area of the image within which they are generated stays the same, the intercity distances are smaller, and so it takes less time to move the computer mouse from one city to another. It follows that the time of executing n movements is likely to grow slower than linearly with n. As a result, when the overall time of solving an n-city E-TSP grows linearly with n, planning the E-TSP solution itself may actually grow faster than linearly with n. The contribution of motor control to the time it takes to solve E-TSP has been recently evaluated by Dry, Lee, Vickers & Hughes (2006). They showed that the contribution of motor control does not change the effect of the number of cities on the time it takes to solve the problem.

Note that OSK spent more time than other subjects in solving the problems. This resulted in slightly better performance, which is shown on the next two graphs.

Figure 5.
(a) Time per city in E-TSP; (b) the effect of the number of cities on the proportion of optimal solutions; (c) the effect of the number of cities on the solution error.

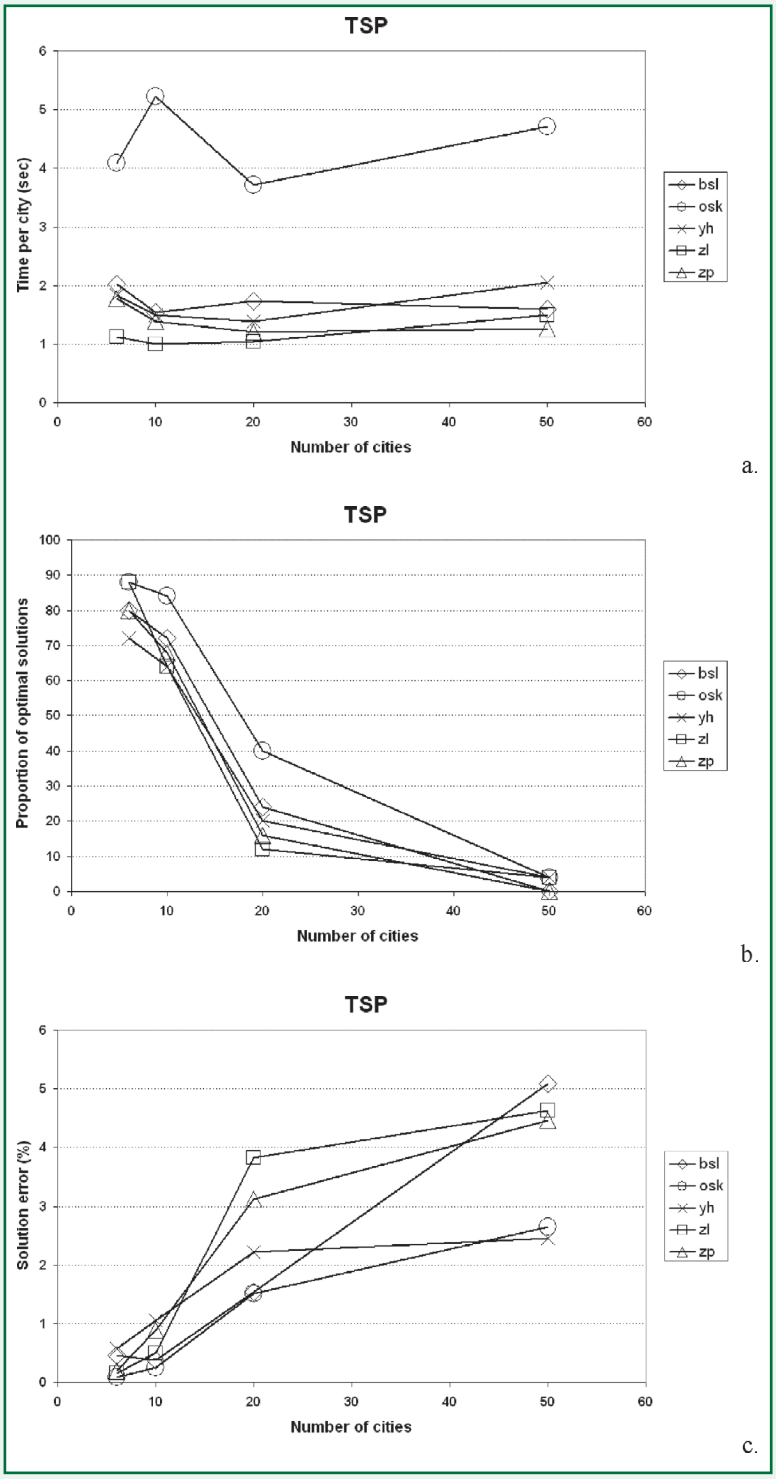


Figure 5b shows the proportion of optimal solutions as a function of the number of cities. In the case of 6-city problems, the optimal solutions were produced 80–90% of the time. In the case of 10-city problems, optimal solutions were produced 60–85% of the time. This performance closely matches that in Graham et al.'s (2000) study. In the case of 50-city problems, three subjects produced one optimal solution each (out of 25 problems). Each of the three subjects found the shortest path for a different problem. It is seen that OSK's performance tended to be better than that of other subjects. Recall that OSK had minimal experience in solving E-TSP problems. This suggests that experience had no beneficial effect on performance. It is more likely that the better performance was related to the fact that he took more time solving the problem.

Is it possible that the effect of the number of cities on the proportion of optimal solutions is simply related to the fact that problems with more cities have more tours? For example, a 3-city E-TSP has only one tour and it is the shortest tour. A 4-city E-TSP has two different tours, but only one does not have a self intersection, and this non-self-intersecting tour is the shortest tour. It is known that subjects almost never produce self-intersecting tours, so it is reasonable to expect that they "select" their tour from the set of non-self-intersecting tours (van Rooij, Stege, & Schactman, 2003). The number of self-intersecting tours depends on the distribution of cities. If all cities are on the convex hull, then there is only one non-self-intersecting tour (regardless of the number of cities), and this is the shortest tour. For a random distribution of cities, the restriction that the tour is non-self-intersecting is not very rigorous, however; there are still a large number of non-self-intersecting tours. Consider a 10-city E-TSP with a random distribution of cities. Our simulation tests showed that the average non-self-intersecting tour for a randomly generated 10-city E-TSP is 35% longer than the shortest tour, whereas the average from all tours is about 85% longer than the shortest tour (see MacGregor, Chronicle & Ormerod, 2004 for a similar argument). Next, the longest non-self-intersecting tour is 80% longer than the shortest tour, whereas the longest 10-city tour is on average 180% longer than the shortest tour. Clearly, imposing a constraint that the tour must be non-self-intersecting restricts the set of tours. However, it is not likely that subjects choose randomly a non-self-intersecting tour because the 10-city tour produced by subjects is on average less than 1% longer than the shortest tour, which is two orders of magnitude better than an average from non-self-intersecting tours.

Figure 5c shows the effect of the number of cities on the average error of the solution. Solution error is defined as a difference between the length of the subject's tour and the length of the shortest tour, normalized to the latter. It is seen that the errors are small and do not depend strongly on the number of cities. Again, OSK's performance was overall better than that of the other four subjects, although the differences among the subjects were not very large.

Fitting the Foveating Bisection Pyramid Model

This model has only one free parameter, and it is the magnitude *k* of the local search in the cheapest insertion method. The model was applied to the 100 E-TSP problems used in the psychophysical experiment with the value of *k* in the range between 0 and 8. For each problem and each subject, the solution whose error was closest to that produced by a given subject was selected. Then, the average error, the proportion of optimal solutions, and the average estimated *k* were computed for each subject.

Figure 6 shows the effect of the number of cities on the average estimated *k*. First, it can be seen that the individual variability in the estimated *k* is not large. This is related to the fact that the individual variability in the length of the solution tours was small as well. Depending on the number of cities, *k* varies from close to zero to about 5. The subjects who produced better solutions (OSK, YH) tended to have higher values of *k*. Figure 7 shows the fit of the FP to the individual subjects' data. Figure 7a shows the average error and Figure 7b shows the proportion of optimal solutions. The FP's performance in Figure 7b is based on the same solutions that optimized *k* values in the case of results in Figure 7a. In other words, *k* was optimized in the case of results shown in Figure 7a but not 7b; the fit in Figure 7b involved no free parameter. Overall, the fits are quite good, which suggests that the FP provides a plausible explanation of the underlying mental mechanisms.

Figure 6.
Estimated values of k for individual subjects and problems with different numbers of cities.

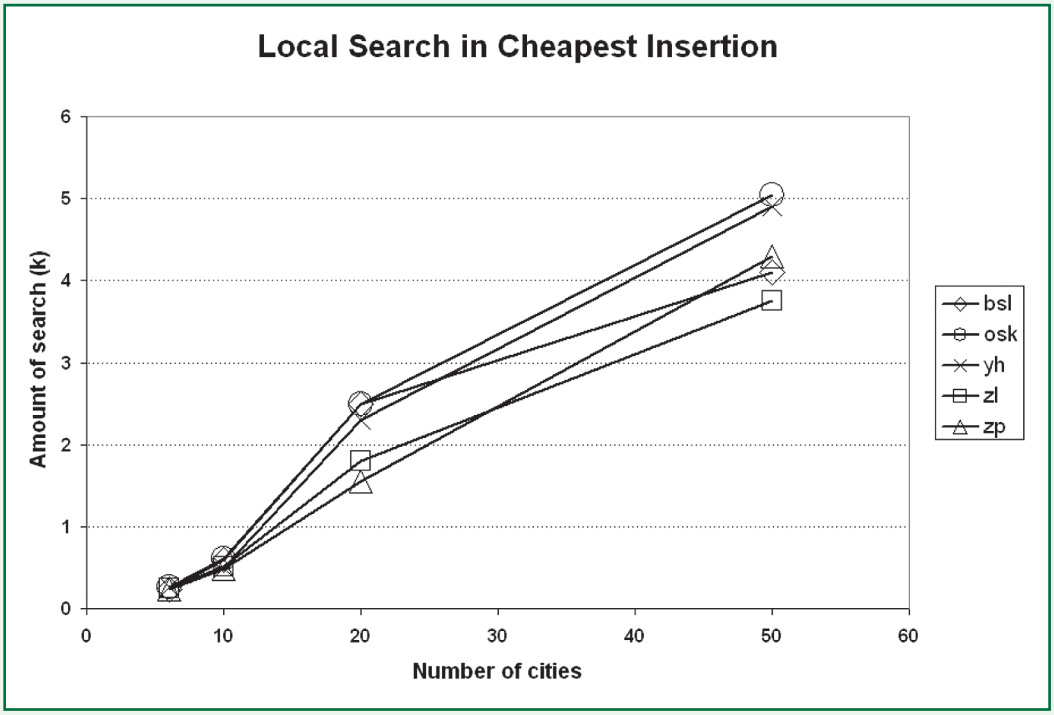
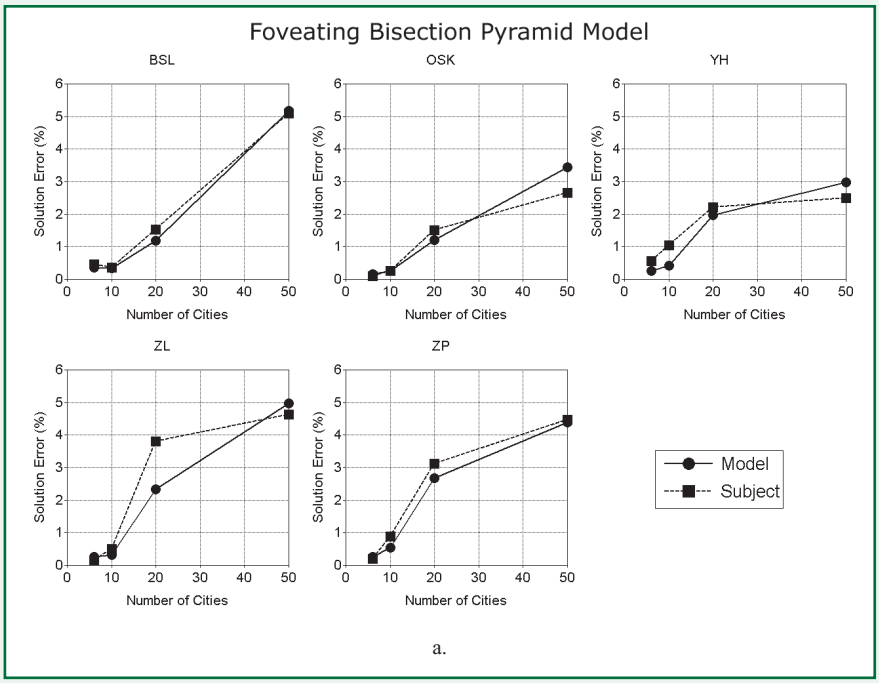
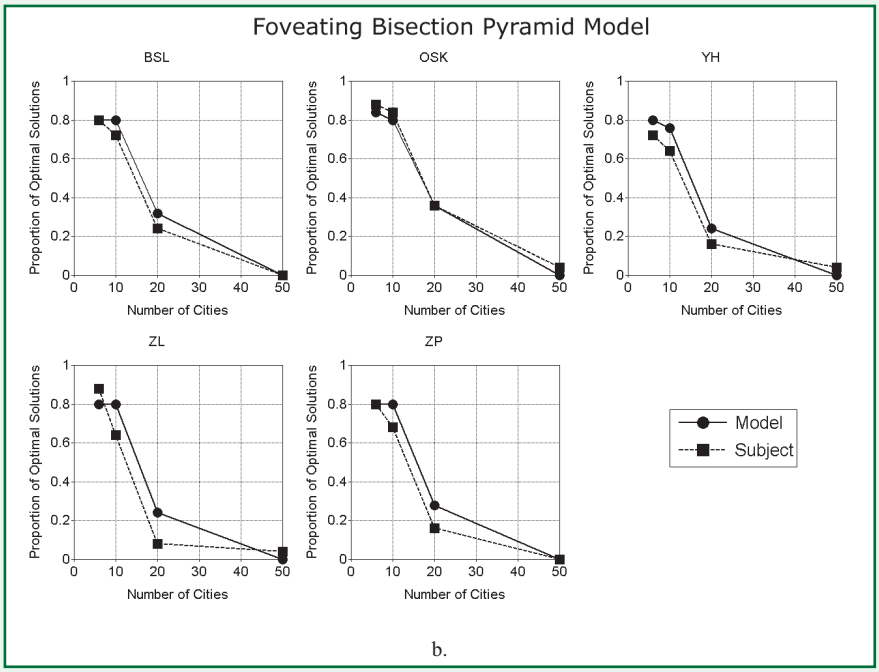


Figure 7.
(a) Model fits to the performance of individual subjects. Average error. (b) Model fits to the performance of individual subjects. Proportion of optimal solutions. The model data were produced by using k values estimated from fitting the error.



a.



b.

Summary

We formulated a new pyramid model for solving E-TSP problems. This new model (FP) approximately solves E-TSP problems in time that is a quadratic function of the number of cities. The FP simulates the nonuniform distribution of visual acuity, as well as the movements of visual attention. By comparing performance of FP with that of five subjects, we found that the model produces solutions whose quality (length of the tour and proportion of optimal tours) is very close to that produced by the subjects. The FP analyzes and uses both global and local properties of E-TSP problems. Global properties have been emphasized by MacGregor and colleagues (MacGregor & Ormerod; Chronicle, 1999). Local properties have been emphasized by Vickers and his colleagues (Vickers, Lee, Dry, & Hughes, 2003). By combining both local and global properties, the FP can find close to optimal E-TSP tours by performing local search only. As a result, the computational complexity of FP is very low and comparable to the complexity of mental mechanisms.

Pyramid algorithms have been used to model human problem solving in the case of other problems, as well. In a recent study, Pizlo & Li (2005) tested humans on a combinatorial problem called the 15-puzzle and three other variants of different sizes (5, 8, and 35 puzzle). Similarly to E-TSP, this family of puzzles is computationally intractable, but humans find a path to the goal state in time that is only a quadratic function of the problem size. For the 15-puzzle, the number of states is 16!/2, which is approximately equal to 10^{13}. It is obvious that a human solver cannot represent in her memory all states of this puzzle. Instead, she has to focus on only a small part of the problem space at a time. In order to model the mental mechanisms, Pizlo & Li (2005) used a pyramid which at any time analyzes only a small number of states. In order to solve the puzzle, the model shifts its attention across different but related parts of the graph representing the puzzle. The model's performance was very similar to human performance. It is tempting to speculate that pyramid algorithms capture something important about the architecture of cognitive functions, such as problem solving, decision making, visual perception, and motor control (see the recent work on motor control by Kwon, Pizlo, Zelaznik, & Chiu, 2005). All of these cognitive functions can be formulated as optimization problems. The search spaces are large, but humans provide approximate solutions quickly. Pyramids seem to be the right way to represent the optimization problem and then to find the global optimum without performing a global search.

Notes

1. Nondeterministic polynomial time—hard.
2. Current model uses horizontal and vertical bisections only and, thus, is rotationally variant (Bister, Cornelis, & Rosenfeld, 1990). In order to make the model orientation invariant, bisections with other orientations will have to be added. The use of only two

directions of bisection (vertical and horizontal) implies that this model may not be able to produce good solutions when long strings of cities are present if their direction is different from the direction of the bisection. This problem is likely to be removed when such strings of cities are explicitly detected and represented in the pyramid.

3. The programs for generating problems, collecting data, analyzing the data, and testing our model can be downloaded from: psych.purdue.edu/tsp/workshop/downloads.html.
4. The authors are grateful to Dr. van Rooij for raising this issue.

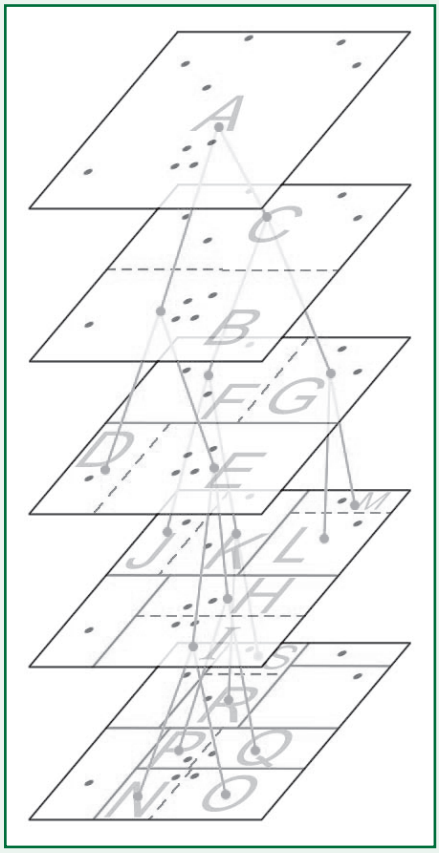
Appendix

Below is the pseudo-code of the FP model. The pseudo-code is followed by a table specifying computational complexity of the individual functions. Figure 8 provides a schematic illustration of the pyramid representation.

```
BUILD-PYRAMID(map)
  apex ← CREATE-NODE(map*)
  SUBDIVIDE(apex)

SUBDIVIDE(parentNode)
  if CITY-COUNT(parentNode) = 1
        done
  cut ← CHOOSE-CUT(region[parentNode])
  regions ← DIVIDE-REGION(region[parentNode>], cut)
  for each r in regions
        n ← CREATE-NODE(r)
        ADD-CHILD(parentNode,n)
  for each child c of parentNode
        SUBDIVIDE(c)

SOLVE-PYRAMID(apex,k)
  tour ← apex
  while HAS-INTERNAL-NODES(tour)
        REFINE-TOUR(tour,k)

REFINE-TOUR tour,k)
  n ← SELECT-NEXT-NODE(tour)
  REMOVE-FROM-TOUR(n,tour)
  for each child c of n
        position ← LOCAL-CHEAPEST-INSERTION (c,tour,k)
        INSERT-INTO-TOUR(c,tour,position)

CITY-COUNT(node)
```

Counts the cities in the region represented by a node.

Figure 8.
Schematic illustration of the pyramid representation. Each of the five large squares represents a layer of the bisection pyramid. The dots represent the cities. The top layer has only one region called A. The bottom layer has as many regions as there are cities. A dashed line within a given layer is the cut representing the boundary between regions (clusters) detected at that layer (see Figure 3b). The letters represent the regions in the order they are detected. The lines between layers represent the connections between parent and child nodes. Dots at the end of each line represent the center of a region. B is an internal node: It does have children. D is a leaf node: It has no children.



`CHOOSE-CUT(`*`region`*`)`
Determines the boundary between regions as described in Section 2.

`DIVIDE-REGION(`*`region`*`,`*`cut`*`)`
Divides a region into two at the specified cut.

`CREATE-NODE(region)`
Creates a node representing a region of the map.

`ADD-CHILD(`*`parent`*`,`*`child`*`)`
Links `child` *as a child node of* `parent`*.*

`HAS-INTERNAL-NODE(`*`tour`*`)`

The tour has no internal nodes when all nodes in the tour are leaf nodes. An internal node is a node with at least one child node.

`SELECT-NEXT-NODE(tour)`
This function picks the next node to refine while simulating the fovea phenomenon.

`REMOVE-FROM-TOUR(node,tour)`
Removes a node from the tour.

`LOCAL-CHEAPEST-INSERTION (node,tour,k)`
Find the locally optimal position to insert the node into the tour using local cheapest insertion.

`INSERT-INTO-TOUR(node,tour,position)`
Add the node to the tour at the specified position.

*A map is a layout of cities represented as a list of coordinates.

In the table, n is the number of cities in the problem, k is the local search parameter, and h is the height of the pyramid representation. In most cases, $h = \Theta(\log(n))$. In the worst case, $h = \Theta(n)$. Thus, the complexity of building and solving is, in most cases, $O(n \cdot \log(n) + n \cdot k) = O(n \cdot \log(n))$—in the worst case $O(n^2)$—and the complexity of our implementation is $O(n^2 + n \cdot k) = O(n^2)$.

Function Name	***Complexity***	***Implementation***
BUILD-PYRAMID	$\mathbf{O(n \cdot h)}$	$\mathbf{O(n^2)}$
SUBDIVIDE	$O(n \cdot h)$	$O(n^2)$
CITY-COUNT	$O(1)$	$O(1)$
CHOOSE-CUT	$O(n)$	$O(n)$
DIVIDE-REGION	$O(n)$	$O(n)$
CREATE-NODE	$O(1)$	$O(1)$
ADD-CHILD	$O(1)$	$O(1)$
SOLVE-PYRAMID	$\mathbf{O(n \cdot h + n \cdot k)}$	$\mathbf{O(n^2 + n \cdot h + n \cdot k)}$
HAS-INTERNAL-NODE	$O(1)$	$O(1)$
REFINE-TOUR	$O(h + k)$	$O(n + h + k)$
SELECT-NEXT-NODE	$O(h)$	$O(n + h)$
REMOVE-FROM-TOUR	$O(1)$	$O(n)$
LOCAL-CHEAPEST-INSERTION	$O(k)$	$O(k)$
INSERT-INTO-TOUR	$O(1)$	$O(n)$

References

Arora, S. (1998). Polynomial-time approximation schemes for Euclidean TSP and other geometric problems. *Journal of the Association for Computing Machinery, 45,* 753–782.

Bister, M., Cornelis, J., & Rosenfeld, A. (1990). A critical view of pyramid segmentation algorithms. *Pattern Recognition Letters, 11,* 605–617.

Burt, P. J. (1988). Smart sensing within a pyramid machine. *IEEE Proceedings, 76,* 1006–1015.

Christofides, N. (1976). Worst-case analysis of a new heuristic for the Traveling Salesman Problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA.

Dry, M., Lee, M.D., Vickers, D., & Hughes, P. (2006). Human performance on visually presented Traveling Salesperson Problems with varying numbers of nodes. *Journal of Problem Solving* (this issue).

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology, 47,* 381–391.

Graham, S. M, Joshi, A., & Pizlo Z. (2000). The Traveling Salesman Problem: A hierarchical model. *Memory & Cognition, 28,* 1191–1204.

Graham, S. M., Pizlo Z., & Joshi A. (1995). An exponential pyramid model of solving the Traveling Salesman Problem. Paper presented at the Annual Meeting of the Society for Mathematical Psychology, Irvine, CA.

Graham, S. M., Pizlo, Z., & Joshi, A. (1996). An exponential pyramid model of solving the Traveling Salesman Problem. *Journal of Mathematical Psychology, 40,* p. 356 (abstract).

Gutin, G. & Punnen, A. P. (2002). *The Traveling Salesman Problem and its variations*. Boston: Kluwer.

Hackbusch, W. (2003). *Multi-grid methods and applications*. Springer.

Jolion, J. M. & Rosenfeld, A., (1994). *A pyramid framework for early vision.* Dordrecht: Kluwer.

Kwon, O. S., Pizlo, Z., Zelaznik, H., & Chiu, G. (2005). Multiresolution model of human motor control. Paper presented at the Annual Meeting of the Psychonomic Society, Toronto, CA.

Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., & Shmoys, D. B. (1985). *The Traveling Salesman Problem*. New York: Wiley.

MacGregor, J. N., Chronicle, E. P., & Ormerod, T. C. (2004). Convex hull or crossing avoidance? Solution heuristics in the Travelling Salesperson Problem. *Memory and Cognition, 32,* 260–270.

MacGregor, J. N., & Ormerod, T. C. (1996). Human performance on the Traveling Salesman Problem. *Perception & Psychophysics, 58,* 527–539.

MacGregor, J. N., Ormerod, T. C., & Chronicle, E. P. (1999). Spatial and contextual factors in human performance in the Traveling Salesman Problem. *Perception, 28,* 1417–1427.

MacGregor, J. N., Ormerod, T. C., & Chronicle, E. P. (2000). A model of human performance on the Traveling Salesperson Problem. *Memory & Cognition, 28,* 1183–1190.

Martin, D., Fowlkes, C., Tal, D. & Malik, J. (2001) A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. Proc. 8th Int'l Conf. Computer Vision, 2 416—423.

Pizlo, Z. (1988) Physiology based simulation model of triangle shape recognition. *Biological Cybernetics, 58,* 51–62.

Pizlo, Z., Joshi, A., & Graham, S. M. (1994). Problem solving in human beings and computers. Technical Report, CSD TR #94-075, Department of Computer Sciences, Purdue University, November.

Pizlo, Z. & Li, Z. (2004) Graph pyramid algorithms as models of human problem solving. *Proceedings of IS&T/SPIE Conference on Computational Imaging, 5299,* 205–215.

Pizlo, Z. & Li, Z. (2005) Solving combinatorial problems: The 15-puzzle. *Memory & Cognition, 33,* 1069–1084.

Pizlo, Z., Rosenfeld, A., & Epelboim, J. (1995). An exponential pyramid model of the time-course of size processing. *Vision Research, 35,* 1089–1107.

Pizlo, Z., Stefanov, E., Saalweachter, J., Haxhimusa, Y. & Kropatsch, W.G. (2005) Adaptive pyramid model for the traveling salesman problem. Workshop on Human Problem Solving, Purdue University (http://psych.purdue.edu/tsp/workshop/files/presentations/pizlo.pdf).

van Rooij, I., Stege, U., & Schactman, A. (2003). Convex hull and tour crossings in the Euclidean Traveling Salesperson Problem: Implications for human performance studies. *Memory & Cognition, 31,* 215–220.

Vickers, D., Butavicius, M., Lee, M., & Medvedev, A. (2001). Human performance on visually presented Traveling Salesman Problems. *Psychological Research, 65,* 34–45.

Vickers, D., Lee, M. D., Dry, M., & Hughes, P. (2003). The roles of the convex hull and the number of potential intersections in performance on visually presented Traveling Salesperson Problems. *Memory & Cognition, 31,* 1094–1104.

Acknowledgments

Pizlo, Stefanov, Saalweachter, Li, and Haxhimusa were supported by a grant from the Air Force Office of Scientific Research. Haxhimusa and Kropatsch were supported by Austrian Science Found FSP-S9103-N04 and FWF-P18716-N13.