Journal of Problem Solving

The Role of Problem Representation in Producing Near-Optimal TSP Tours

Pierson Fleischer, Sébastien Hélie, and Zygmunt Pizlo

Purdue University

Correspondence:
Correspondence concerning this article should be addressed to Pierson Fletcher via email at pfleisch@purdue.edu.

Keywords:
TSP, shortest path, log-polar transformation

Gestalt psychologists pointed out about 100 years ago that a key to solving difficult insight problems is to change the mental representation of the problem, as is the case, for example, with solving the six matches problem in 2D vs. 3D space. In this study we ask a different question, namely what representation is used when subjects solve search, rather than insight problems. Some search problems, such as the traveling salesman problem (TSP), are defined in the Euclidean plane on the computer monitor or on a piece of paper, and it seems natural to assume that subjects who solve a Euclidean TSP do so using a Euclidean representation. It is natural to make this assumption because the TSP task is defined in that space. We provide evidence that, on the contrary, subjects may produce TSP tours in the complex-log representation of the TSP city map. The complex-log map is a reasonable assumption here, because there is evidence suggesting that the retinal image is represented in the primary visual cortex as a complex-log transformation of the retina. It follows that the subject's brain may be "solving" the TSP using complex-log maps. We conclude by pointing out that solving a Euclidean problem in a complex-log representation may be acceptable, even desirable, if the subject is looking for near-optimal, rather than optimal solutions.

INTRODUCTION

The traveling salesman problem refers to finding the order in which N cities should be visited so that the resulting closed tour is shortest (Lawler, Lenstra, Kan, & Shmoys, 1985). After the tour is produced, the result is the permutation of N numbers representing the order in which the cities were visited, and the length of the tour. If the distances are symmetrical, the number of possible tours is $(N - 1)!/2$. This number grows exponentially with N. There is no known method for finding the optimal tour without running the risk of trying all tours. This is why TSP is in the class of NP hard problems. Trying all tours is impractical even for fairly small N. Branch and bound algorithms can often produce optimal tours quickly even for reasonably large N, but one never knows whether a given problem at hand will be solved quickly (Applegate, Bixby, Chvatal, & Cook, 2006). This is why there has been large interest in formulating approximating algorithms, as well as algorithms that produce near-optimal tours. "Approximating algorithms" are algorithms for which there is a proven upper bound for error. Tour error is given as the difference between the length of the tour produced and the length of the shortest tour expressed as a percentage of the length of the shortest tour. Describing a tour as "near-optimal" means that the tour is not much longer than the shortest tour (error close to 0%), but the upper bound for an error has not been established.

There is a large body of work on approximating and near-optimal algorithms for TSP (Lawler et al., 1985). These algorithms vary with respect to computational complexity and errors. A universal assumption behind all these algorithms is that they work in the space in which the problem is defined. Usually, it is a Euclidean plane. Looking for a TSP tour in the space in which the problem is defined sounds like an obvious thing to do. Indeed, that approach would be correct, if the goal is to find the shortest tour. If the algorithm does not take the transformation into account, then using an alternate coordinate system should only hurt the chances that the algorithm will produce the optimal tour. For example: transform Cartesian coordinates of the points (cities) (x,y) into polar coordinates (r,θ) (see Figure 1a, see next page) and use the Cartesian representation with the polar coordinates: θ being the horizontal axis and r being the vertical axis (see Figure 1b, see next page). Now, we can compute the distance between points in polar space by applying the conventional distance formula to polar coordinates: $d = ((\Delta r)^2 + (\Delta\theta)^2)^{1/2}$ (note that this example is relevant to how our model works; the main and important difference is that we use a complex-log mapping, in which a natural logarithm of the radius, not the radius itself, is used). It should be obvious that the distances computed in the polar space are not only different from distances when Cartesian (x,y) coordinates are used, but they are not a linear transformation of the Euclidean distances. It follows that the

http://dx.doi.org/10.7771/1932-6246.1212

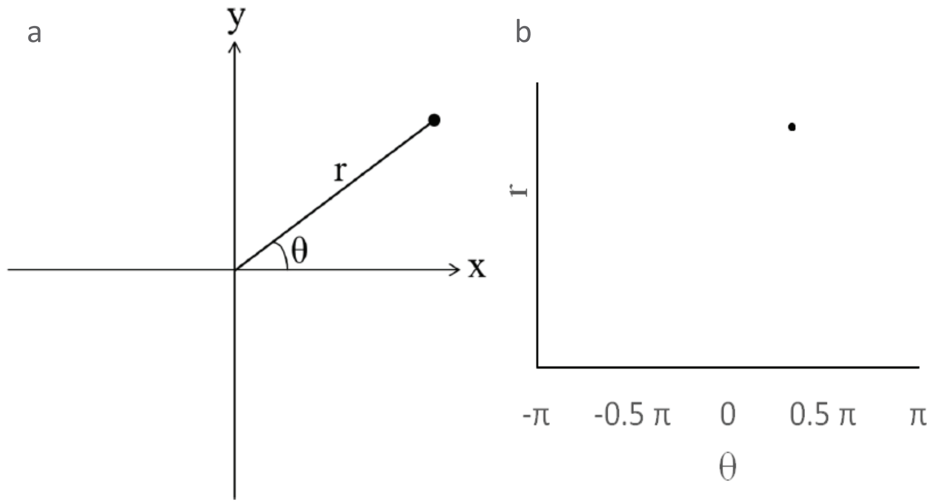

Figure 1.
(a) Cartesian (x,y) and polar (r,θ) coordinate systems. (b) Cartesian representation of polar coordinates.

shortest tour in this new representation will consist of a permutation of cities that is different from the optimal permutation in Cartesian coordinates. Using a permutation that is not optimal means that the length of the tour will not be shortest.

However, if the goal is to find a near-optimal tour, then it is not immediately obvious why the algorithm should work on the representation in which the problem was defined. Producing a near-optimal solution in a transformed representation may be easier than doing this in the original representation. By easier we mean faster, smaller error measure, or more effective with respect to cognitive/computational resources such as memory. Carruthers (2015) recently showed, using two examples, that humans can reformulate problems and work on the reformulated problem when the original problem exceeds their cognitive resources. Similarly, one of us has shown that when subjects are presented with a 15-tile puzzle, they move around one or at most two tiles at a time, putting them in the correct location and treating the remaining tiles that are still out of place as equivalent (Pizlo & Li, 2005). Subjects arrange the tiles quickly, but the number of moves tends to be highly suboptimal. An optimal, or close to optimal algorithm must consider all 15 tiles at the same time in planning the next move. This is clearly beyond the memory capacity of the human problem solver.

Finally, consider an example that has been reported recently by Kwon, Agrawal, Li, and Pizlo (2016) that provided a direct motivation for the present paper. Subjects were asked to draw a closed contour that was represented in the image by short line segments embedded in a large number of similar line segments, called distracters (Figure 2, see next page). The experimenter knew which segments were distracters and which segments belonged to the closed contour because the latter ones were produced by fragmenting a polygon. The subjects had to infer what the shape of the polygon was. They had to guess which segments belong to the closed contour representing the polygon and in which order they should be connected. Drawing a closed contour sounds like a TSP problem. But this was not a TSP problem because the subjects had to connect a small subset of contours from among a large number of distracters. There is no version of a TSP that allows the “salesperson” to ignore cities. Kwon et al. discovered that subjects treat the task as a shortest path problem (SPP). SPPs always ignore nodes and edges in order to go from the start to the end along the shortest path. But in order to produce a closed contour, the shortest path had to start and end at the same point in the image. This will not work, unless the Cartesian coordinate system in the retina is transformed into a polar coordinate system. In fact, it is more desirable to use a complex-log representation in which the radius r is transformed into a ln(r) and the polar angle is expressed in radians. This way, large and small circles on the retina have the same length, thus providing size invariance.

The choice of a log-polar map (another name for a complex-log map) was not a coincidence because it has been known for some time (since 1977; see Schwartz, 1980, for a review) that the primary projection area (area V1) in the visual cortex of the primate brain is approximately a complex-log transformation of the retina. Figure 3 (see following pages) shows how the retinal stimulus in Kwon et al.’s study looks in the complex-log representation, how the shortest (least-cost) path looks in the complex-log representation, and the corresponding closed contour on the retina (see Kwon et al., 2016, for details).

In the case of Figure 3 the origin is chosen to be the center of the image and the polar (θ) component ranges from

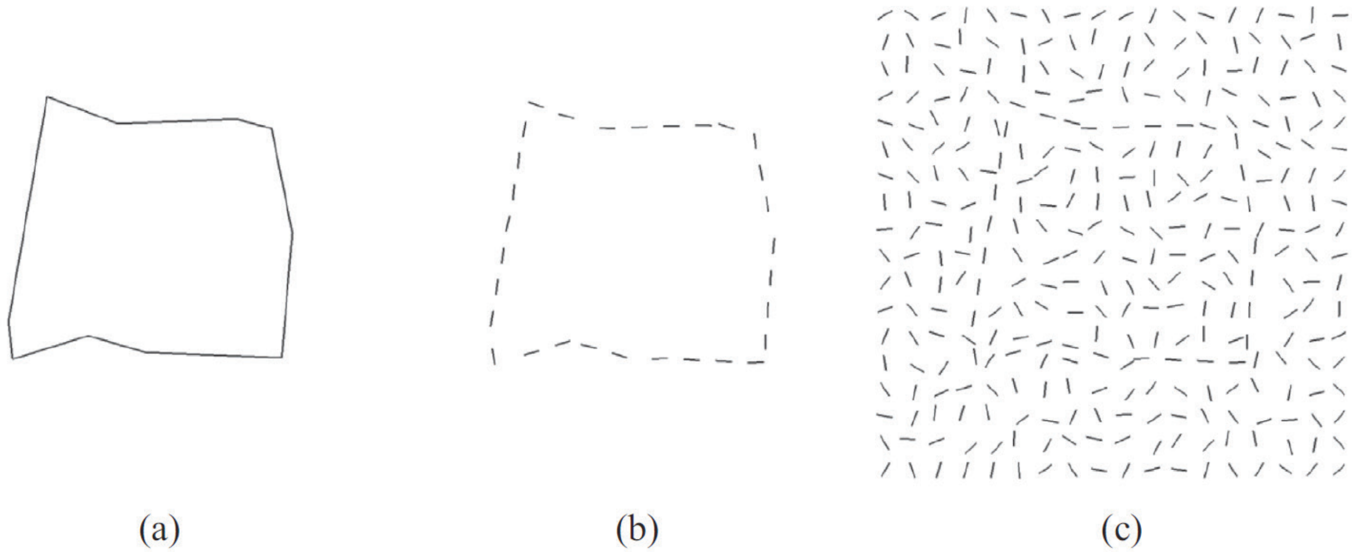

Figure 2.
Explanation of how stimuli were designed in Kwon and colleagues' (2016) study. The original closed contour (a), fragmented contour (b), and fragmented contour within a field of distractor line segments (c).

-π to π. The subject is unaware of the complex-log map in his brain because he is drawing the contour on the computer screen. Despite, or perhaps thanks to, changed representation, the subjects always produce the correct (true) polygon and they do it instantaneously. Selecting the right segments from among distracters calls for nothing else than considering all subsets of the segments of the screen. For N segments, the number of all subsets is 2^N, so evaluating all subsets is computationally intractable. In contrast, solving the SPP optimally can be done in polynomial time because the complexity of the optimal algorithm is $V\log(V + G)$, where V is the number of nodes and G is the number of edges in the graph. So, one could conclude that the visual system is well adapted: it takes what looks like an NP hard problem on the retina and solves it in polynomial time in the brain representation. Note that we cannot prove that these two problems are equivalent. Otherwise, we would have shown that P = NP, which is no small task and commonly believed not true (Garey & Johnson, 1979). But what the visual system does is very interesting. The reader may realize that a complex-log map is a member of a class of conformal maps well studied in applied mathematics (Schinzinger & Laura, 2003). Conformal maps are used in applied mathematics problems, because what is very difficult to do in the original representation may become easy in one of the conformal map representations. So, it is interesting to note that nature came up with one such map well before mathematicians did. Descartes

-π to π. The subject is unaware of the complex-log map in his brain because he is drawing the contour on the computer screen. Despite, or perhaps thanks to, changed representation, the subjects always produce the correct (true) polygon and they do it instantaneously. Selecting the right segments from among distracters calls for nothing else than considering all subsets of the segments of the screen. For N segments, the number of all subsets is 2^N, so evaluating all subsets is computationally intractable. In contrast, solving the SPP optimally can be done in polynomial time because the complexity of the optimal algorithm is $V\log(V + G)$, where V is the number of nodes and G is the number of edges in the graph. So, one could conclude that the visual system is well adapted: it takes what looks like an NP hard problem on the retina and solves it in polynomial time in the brain representation. Note that we cannot prove that these two problems are equivalent. Otherwise, we would have shown that P = NP, which is no small task and commonly believed not true (Garey & Johnson, 1979). But what the visual system does is very interesting. The reader may realize that a complex-log map is a member of a class of conformal maps well studied in applied mathematics (Schinzinger & Laura, 2003). Conformal maps are used in applied mathematics problems, because what is very difficult to do in the original representation may become easy in one of the conformal map representations. So, it is interesting to note that nature came up with one such map well before mathematicians did. Descartes

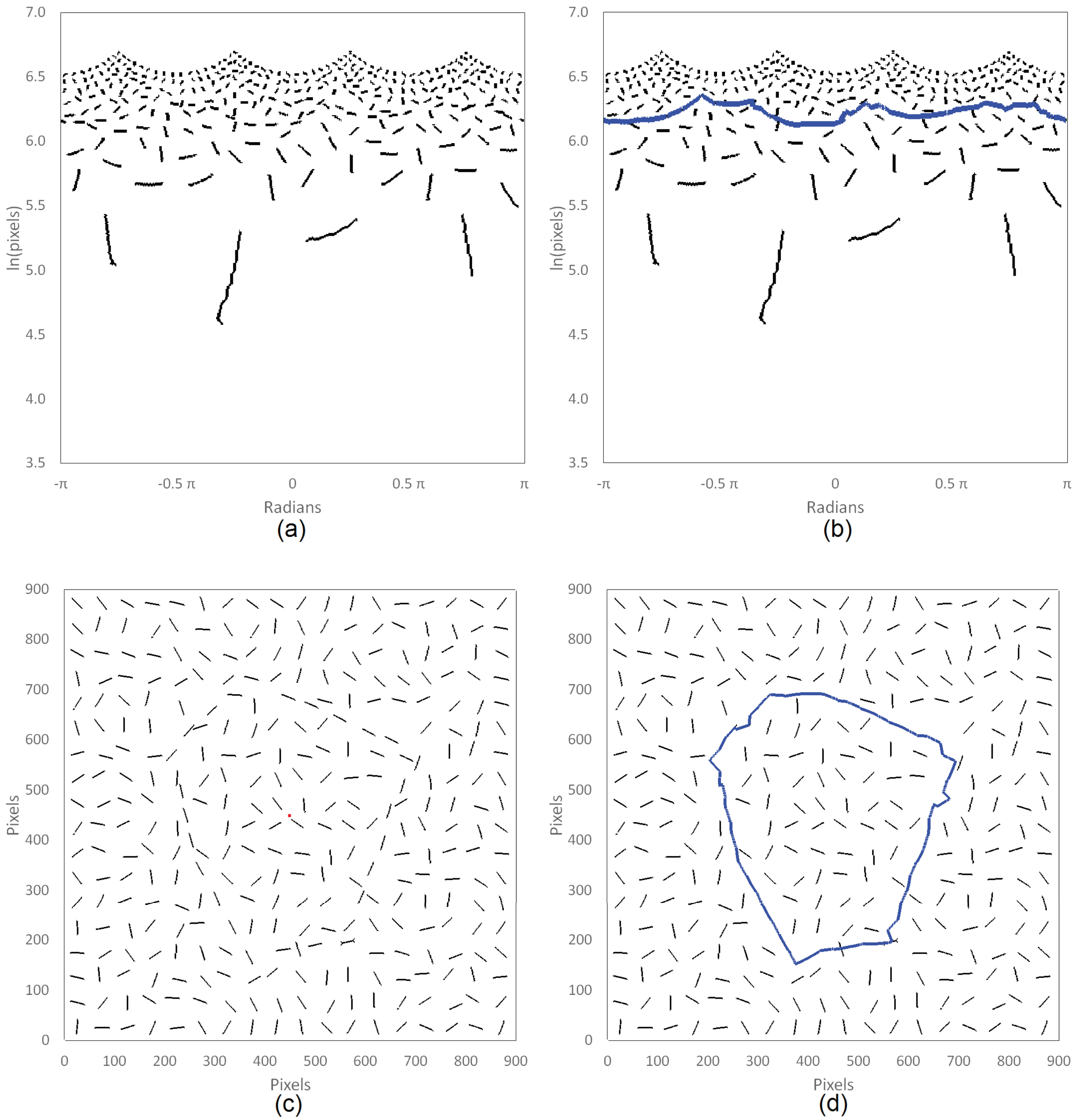

Figure 3.
The presented field of line segments as it appears on the computer screen and retina (c, d), and as it would be represented in V1 (a, b). (a) and (c) show the subject's drawn contour. The red dot at the center of (c) indicates the origin used to create the log-polar representations.

would have been pleased, because this is yet another example for his claim that the human mind is a natural geometer.

With these results in hand, we hypothesized that the TSP problem may be solved in the log-polar representation in area V1 instead of the Cartesian representation on the retina (and on the computer screen). The subjects think that they solve it as presented on the screen, but they may simply be unaware of what their brain is doing. Considering the fact that the primary visual area uses a complex-log representation, how good are the TSP tours when they are produced in complex-log, rather than on the retina or a computer screen? This paper shows in a set of simulation experiments that near-optimal algorithms applied to complex-log representations are not worse, and perhaps are even better than

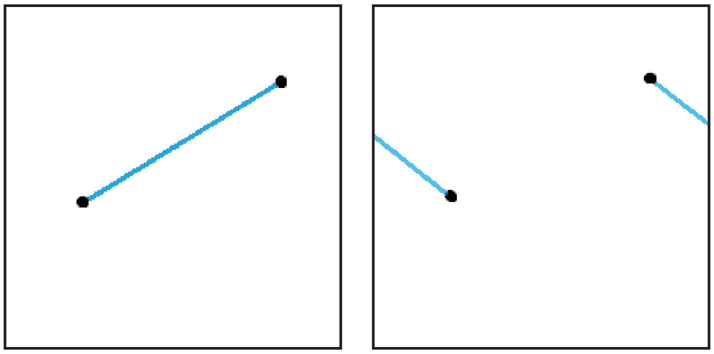

Figure 4.
The two ways that a pair of cities can be connected when the map is wrapped around a cylinder and the left edge of the map touches the right edge.

corresponding algorithms in the Cartesian representation. We also report a preliminary study in which the cities were generated on two concentric circles. Some subjects consistently produced a tour that resembled two connected concentric circles, while others produced a tour that zigzagged between the two circles. Our approximating algorithms cannot produce a two-circle tour when applied to the Cartesian representation, but they can produce both types of tours when applied to complex-log representations.

LOG-POLAR CYLINDER

In order to use a complex-log map to produce TSP tours, the transformed map must be treated as though it is wrapped around a circular cylinder. This means that a line going off the left side of the transformed map appears on the right side. The circumference of the base of the cylinder is 2π. Treating the map as the surface of a cylinder was not needed, and in fact was not desirable, when Kwon et al. (2016) solved the SPP problem in a complex-log map. A discontinuity at one of the nodes (line segments) led to a map where this segment had two representations in the complex-log map. This way, a shortest path from a point to itself in complex-log produced a closed contour on the retina. With the TSP we also need a closed contour on the retina, but unlike the SPP, in the TSP all cities must be connected. Recall that the SPP by its very nature goes through only some nodes in the graph, and so the SPP does not apply. Because the SPP is not used, a discontinuity is not needed, and so the complex-log map should be folded to form a cylinder. The cylinder interpretation of the map introduces an ambiguity in the pairwise distances because for each pair of points on the surface of the cylinder you could travel clockwise or counter-clockwise to get from one point to the other (Figure 4). In other words, there are two geodesic lines between any pair of points on the cylinder. Our algorithm always used the shorter of the two distances in the complex-log representation because this distance would have been used by any TSP algorithm when the algorithm is applied to the complex-log representation. Note that the choice of the shorter of the two distances in the complex-log map can be done by simply comparing the coordinates on the horizontal axis (angle). Next, we explain in detail how a complex-log transformation is computed.

Look at Figure 4. We start with polar coordinates in which a point is defined by its distance r from the origin and its angle θ relative to the positive x-axis. The angle is expressed in radians. The units of radius r are irrelevant in the sense that changing the unit from pixels, to centimeters, to inches will lead to a rigid translation in the complex-log map. A translation has no effect on the TSP tours. What is important is that the radius r is transformed into a natural logarithm of r. If one begins with a complex plane $z = x + iy = re^{i\theta}$, then a complex-log leads to the correct log-polar representation. If, however, one begins with two real numbers, r and θ, then one has to be sure that a natural logarithm and radians are used.

Before the complex-log transformation is applied, one must choose the origin of the polar coordinate system. In the visual system, this origin coincides with the center of the retina, which, in turn, corresponds to the fixation point. Changing the origin will change the complex-log map, including the distances between points. This means that optimal and near-optimal TSP tours will be affected by the change of the origin. This will give any model based on a complex-log map additional flexibility. For example, the model may try multiple fixation points, generate a tour for each, and then choose the shortest. Subjects could do this too. In this paper we did not control or record the positions of fixation points. We used the fixation point as a free parameter in our models.

Our models first transform the Cartesian representation of the TSP problem using the complex-log transformation and then produce a TSP tour. The resulting permutation is then reproduced in the Cartesian representation and used for the analysis of error and optimality. In order to avoid a singularity corresponding to a logarithm of zero, no city was allowed to be too close to the origin of the polar coordinate system. This was enforced by adjusting the location of the origin if it fell too close to a city.

SIMULATION EXPERIMENTS

Throughout the paper, by “optimal tour” we mean the shortest tour in the Cartesian coordinates. The errors of near-optimal algorithms are errors for the tour produced in a particular representation (Cartesian, polar or complex-log) after the permutation of cities representing the tour produced by a near-optimal algorithm has been reproduced in the Cartesian plane.

Consider the simplest TSP problem where N cities are on a circumference of a circle in Cartesian representation.

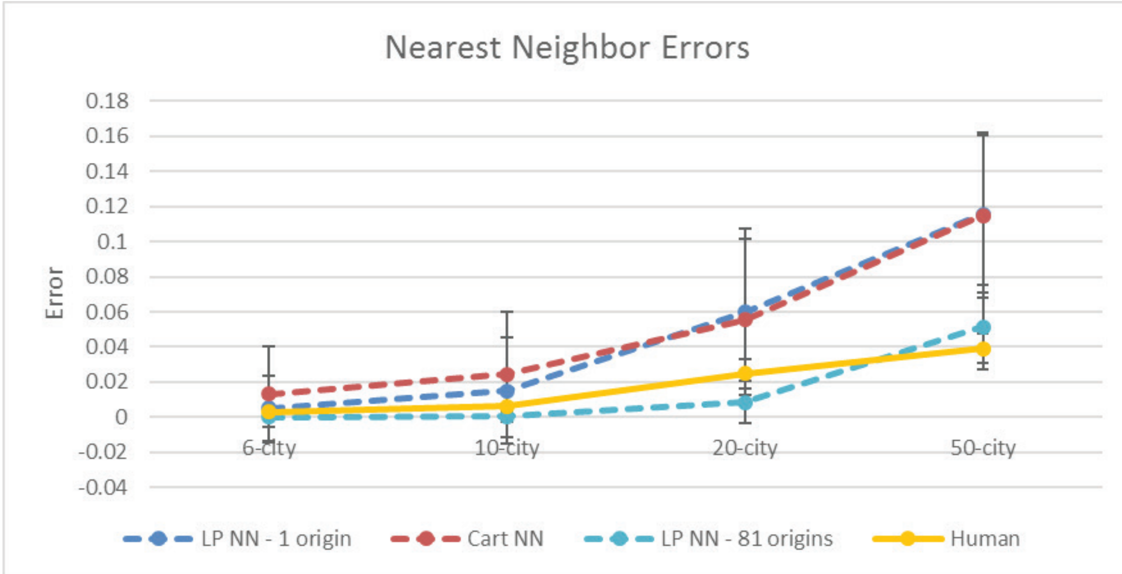

Figure 5.
Error measure of the nearest neighbor solutions in Cartesian coordinates, best of 1 nearest neighbor solutions in log-polar coordinates, best of 81 nearest neighbor solutions in log-polar coordinates, and human solutions. Error bars show standard deviation.

Humans always produce an optimal tour for such a TSP. The optimal tour is a polygon with N vertices inscribed in that circle. There are several ways to produce an optimal tour in such a case. One way is to find a convex hull of the points (MacGregor & Ormerod, 1996). An algorithm that computes a convex hull has polynomial complexity. Alternatively, one can use a nearest neighbor algorithm, which is also of polynomial complexity. If this problem is represented in a complex-log representation and if the origin of the polar coordinate system coincides with the center of the circle, the circle in Cartesian coordinates maps into a straight line in complex-log representation. More precisely, it will be a circle on the complex-log cylinder. The nearest neighbor algorithm applied in complex-log representation would produce a tour that is optimal in both representations. By optimal we mean the optimal permutation. Geodesic lines on the complex-log cylinder will map to arcs of a circle in the original Cartesian representation, not to straight-line segments representing the inscribed polygon. However, since by an optimal tour we mean an optimal permutation, we can ignore the fact that the connections between pairs of cities may or may not be straight line segments.

NEAREST NEIGHBOR ALGORITHM

Method

We started our simulation experiments with the nearest neighbor algorithm to solve randomly generated TSP problems with 6, 10, 20, and 50 cities in Cartesian coordinates and in log-polar coordinates, using up to 81 origin points in the log-polar transformations. We used 1,000 randomly generated problems for each problem size. The nearest neighbor algorithm that we used tried all N starting points and chose the shortest tour. One of our goals was to compare the solutions generated in Cartesian and log-polar spaces. One would expect that distorting the map prior to solving it would result in overall longer tours when converted back into Cartesian coordinates. We compared the solutions generated in Cartesian space to two versions of the log-polar model: (i) with only one fixation point and (ii) with 81 fixation points. In the second version, we chose the shortest tour from the 81 tours, where shortest was evaluated in Cartesian representation. The 81 fixation points represented a regular grid of points. The results of human subjects described in this section were taken from Pizlo et al. (2006).

Results

We measured the error of a tour as the ((tour length) - (shortest tour length))/(shortest tour length). Percent optimal is the proportion of optimal tours. Figures 5 and 6 (see next page) show that the nearest neighbor algorithm with one fixation in log-polar space performed as well or better compared to nearest neighbor in Cartesian space. However, both were systematically worse than the subjects in Pizlo et al.'s (2006) experiment. When 81 fixations were used and the best tour was chosen, NN algorithm in the log-polar space performed similarly to human subjects. This is a new result suggesting that one may want to reconsider NN algorithms as a good greedy method, at least with small problem sizes. In the next simulation experiment we compared pyramid algorithms when implemented in the Cartesian vs. log-polar representation.

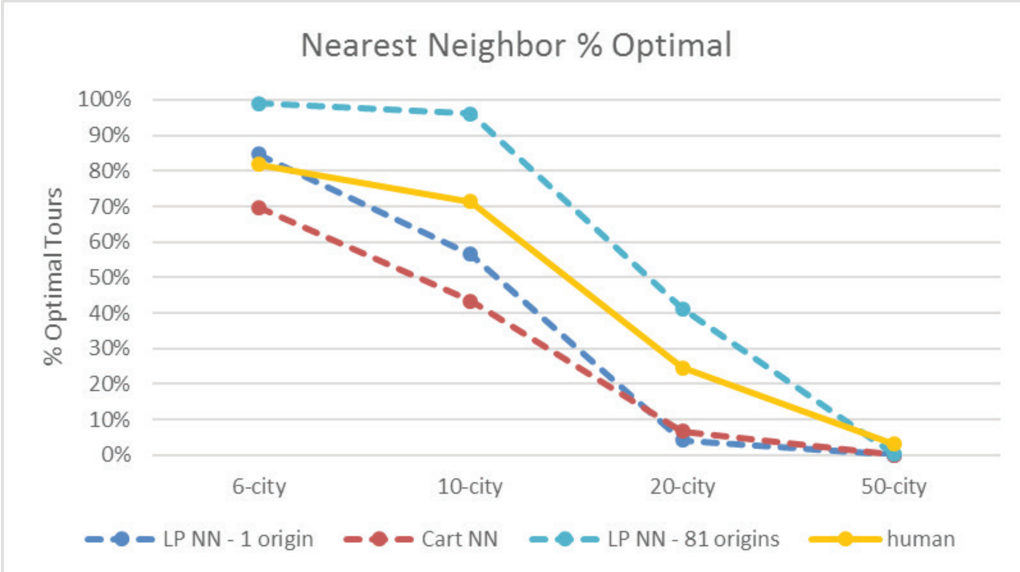

Figure 6.
The proportion of optimal tours in the nearest neighbor solutions in Cartesian coordinates, best of 1 nearest neighbor solutions in log-polar coordinates, best of 81 nearest neighbor solutions in log-polar coordinates, and human solutions.

PYRAMID ALGORITHM

Method

In our second experiment, we tested a version of a pyramid algorithm that we used in our previous studies (Haxhimusa, Kropatsch, Pizlo, & Ion, 2009). We applied the pyramid algorithm to the Cartesian representation and to the log-polar representation with multiple origins (fixation points). Before the multiresolution pyramid produces a TSP tour, a clustering algorithm is applied recursively to produce a hierarchy of clusters. The clustering merges at least two cities to form a cluster that is treated as though it were a city in the next highest layer. In our implementation, we merge the two closest cities/clusters at each merge cycle. The cluster formed this way is positioned at the midpoint between the two cities/clusters that compose the new cluster. The size or depth of a cluster does not affect this process in any way: large clusters are no more or less likely to be involved in a merge and do not affect the determination of the new cluster's location. This clustering method is simple computationally, and it resembles in some respects Boruvka's method for producing a minimum spanning tree (MST). Boruvka's algorithm was used in one of our earlier TSP studies (Haxhimusa et al., 2009). The clustering method used here leads to tours that are longer than those produced by Haxhimusa et al.'s algorithm. Note, however, that the main goal of the present study was to compare the tours produced by the same pyramid algorithm in two different representations, rather than to optimize the algorithm itself.

Clustering ceases when the top layer of the pyramid 3 has clusters. With 3 cities, there is effectively only one TSP tour since tour length is unaffected by changing the starting point or reversing the order of cities. This tour is the first tour in a sequence of tours produced by unmerging clusters one pair at a time with the rest of the tour remaining unchanged (see Figure 7, next page). Each unmerging results in two possible paths through the newly unmerged subclusters. The algorithm adopts the shorter of the two.

We used the clustering algorithm to solve problems with 6, 10, 20, and 50 cities in (1) Cartesian coordinates, (2) polar coordinates, and (3) log-polar coordinates using up to 25 origin points in the polar and log-polar transformations. The polar coordinate system has been used in the past in computer vision studies, and it is a natural candidate to shed more light on the performance in log-polar representation.

Since the distance dimension of polar coordinates is not scale invariant as it is for log-polar coordinates, we scaled that dimension by $m^*2\pi/\sqrt{((\text{map\_width})^2+(\text{map\_height})^2}$ where the scale factor m is 0.5, 1.0, or 2.0. The scale factor of 1 corresponds to equal weight being given to the length dimension and the angular dimension. A factor of 2 makes the model more biased toward making large angular changes and avoiding large radial changes. A factor of 0.5 would invert this bias.

Results

Applying the pyramid algorithm to polar coordinates with m = 1 produced substantially longer tours than applying the

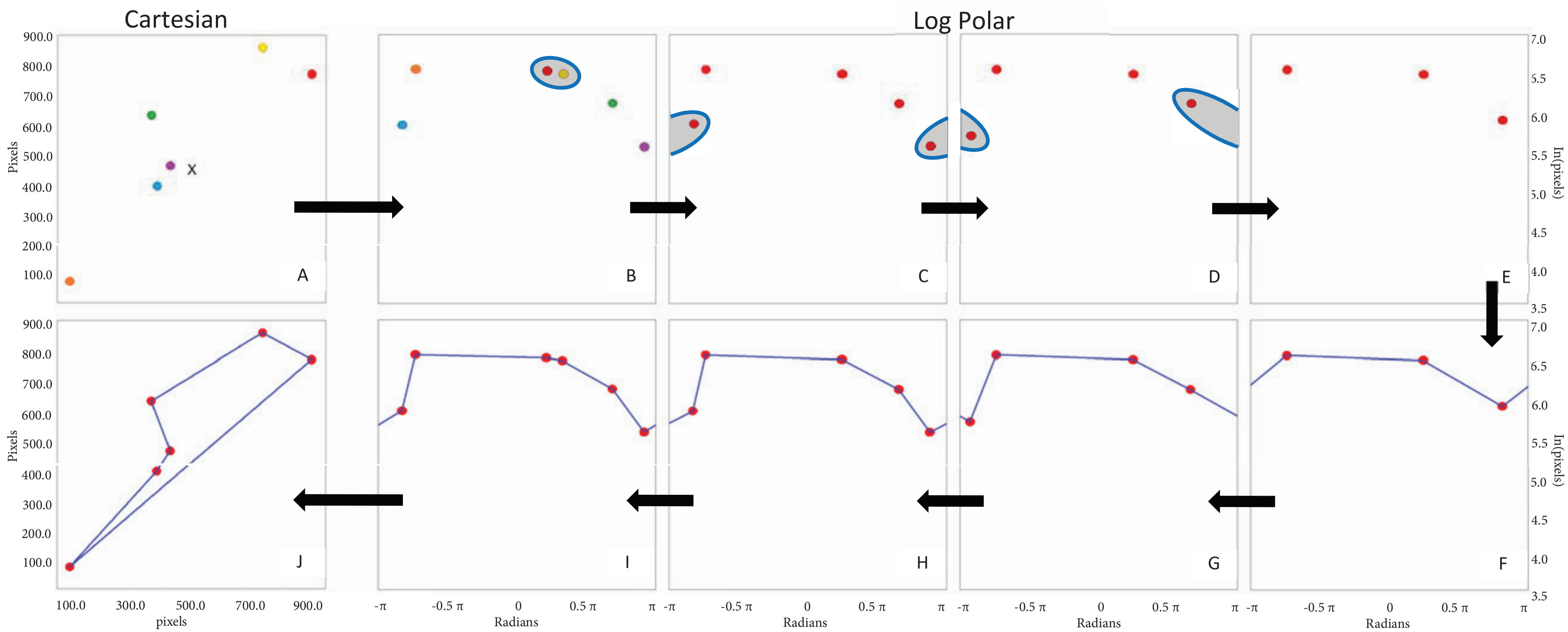

Figure 7.
A step-by-step illustration of the Pyramid algorithm solving a 6-city TSP problem. (A) The initial Cartesian map that subjects would see on the screen. (B) The Cartesian map transformed into Log-Polar space. The origin (fixation point) used for the transformation is located at coordinates (450, 450) and is marked with an X on the Cartesian map (A). The colors of the cities remain constant between the Cartesian map and the log-polar map to show how the log-polar map corresponds to the Cartesian map. (C) The log-polar map after the first cluster has been made. That new cluster is formed from and replaced the circled cities in (B). (D) The log-polar map after the second cluster has been made from the circled cities in (C) In this case the cluster is formed across the transition from $-\pi$ to $+\pi$. (E) The log-polar map after the third cluster has been made from the circled cities in (D). Note that we are now down to three cities/clusters. (F) A tour is created connecting the three cities. The starting point and order the cities are visited in is irrelevant. One of the ledges of the tour also transitions from $-\pi$ to $+\pi$. Neglecting this potential to wrap around would result in points such as the blue and purple points in (A) and (B) being treated as very far from each other in the lob-polar map (B) despite being very close to each other in the Cartesian map (A). (G) The tour after the first cluster is dissolved. (H) Dissolution of the second cluster. (I) Dissolution of the third cluster. No clusters remain and we are now back to the log-polar transformation of the original Cartesian map with the addition of a tour. (J) The Cartesian tour corresponding to the tour we found in (I).

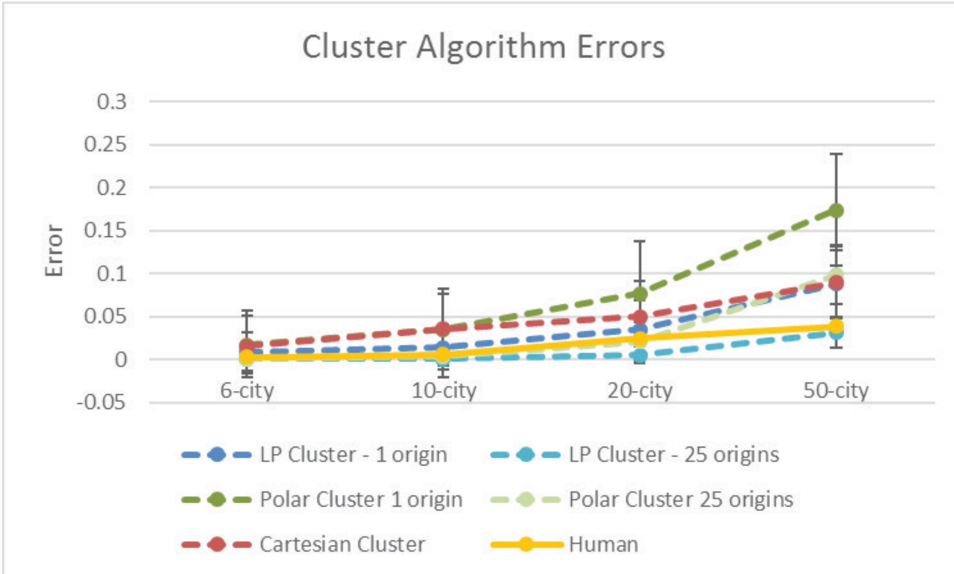

Figure 8.
Error measure of the pyramid algorithm solutions in Cartesian coordinates, 1 clustering algorithm solutions in polar and log-polar coordinates, best of 25 clustering algorithm solutions in polar and log-polar coordinates, and human solutions. Error bars show standard deviation.

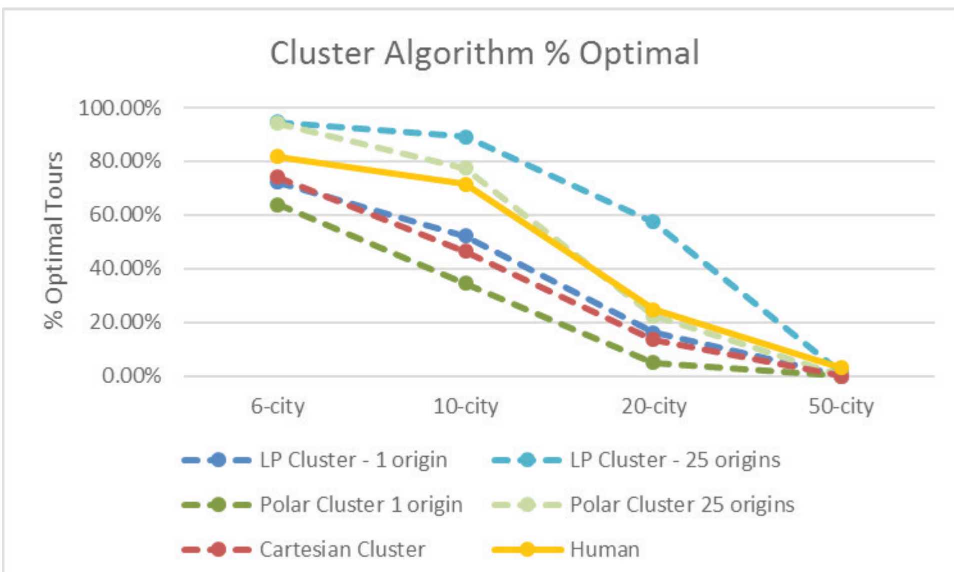

Figure 9.
The proportion of optimal tours in the clustering algorithm solutions in Cartesian coordinates, 1 clustering algorithm solution in polar and log-polar coordinates, best of 25 clustering algorithm solutions in polar and log-polar coordinates, and human solutions.

pyramid algorithm to Cartesian coordinates, as can be seen in Figure 8. In contrast, applying the pyramid algorithm to log-polar coordinates leads to better tours. The superiority of log-polar representation over polar representation holds for a single origin, as well as for 25 origins (Figures 8 and 9).

A comparison between the different scale factors for polar coordinates and log-polar, each with 25 fixation points, can be seen in Figure 10 (see next page). It is interesting to note that discounting the radius dimension relative to the angular dimension seems to produce better tours. However, the log-polar transformation outperformed all 3 versions of the polar transformation. It is possible that polar representation with the scale factor lower than 0.5 could do better, but it is unlikely that one scale factor would fit many different

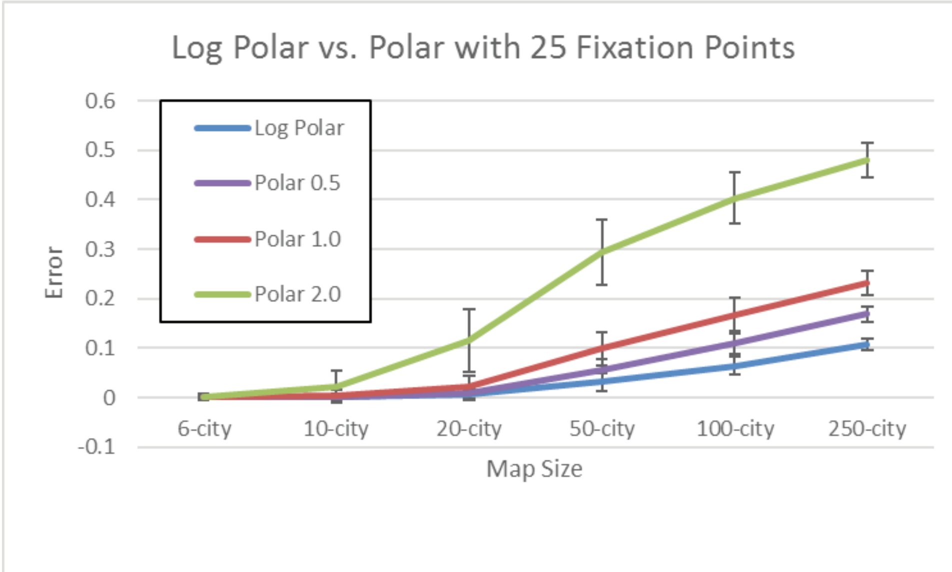

Figure 10.
The error of the tours generated under the polar transformation with 25 fixation points and three scale factors for the radius dimension compared to the error of the tours generated using the log-polar transformation with 25 fixation points. Error bars show standard deviation.

problems equally well. So, one can expect large variability from problem to problem with any particular value of the scale factor.

SPECIAL CASES OF A TSP CONFIGURATION—PSYCHOPHYSICS AND MODELS

Method

We generated 20 TSP problems where cities were arranged in two concentric circles, 8 cities randomly placed on the inner circle and 16 on the outer circle. The radii of the circles were 100 *px* and 220 *px*. These problems were solved using the pyramid algorithm in Cartesian coordinates and in log-polar coordinates using 25 origin points in the log-polar transformations. Our human data was collected from 25 undergraduate subjects. TSP problems with these arrangements of cities in concentric circles generally have optimal solutions that fall into one of two categories: zigzag tours where there are many transitions between the inner and outer circle as the tour makes its way around the center once in one direction, and "keyhole" where the tour follows one circle in one direction, transitions to the other circle once, and follow the other circle in the opposite direction before transitioning back once more to the starting point on the first circle. For all of our circle problems the optimal tour fell into the keyhole category. However, human subjects produced tours of both types—see Figure 11 (next page). A follow-up question is whether the model can also produce tours of both types by varying the origin.

Results

In Figure 12 (see next page) we again see that single tour pyramid solutions in log-polar space are as good as the Cartesian space pyramid solution and that human performance is reached using a pyramid algorithm in the log-polar space with the best of 25 fixation points.

We also found that our model, by changing the fixation point, was able to produce exactly the same tours that human subjects produced including both zigzag and keyhole tours, some examples of which can be found in Figure 11. This result suggests that changing the fixation point in the pyramid algorithm applied to the complex-log representation may account for at least some individual differences observed in solving TSP.

CONCLUSION AND DISCUSSION

The results show that two of the near-optimal algorithms (NN and pyramid) when applied to log-polar representation perform as well as, or better than they do in Cartesian space. So, even though distorting the problem space prior to solving the problem is likely to result in suboptimal solutions, the near-optimal solutions may be more likely to occur and more likely to be closer to the optimal solution when they do occur. The main reason for the improved performance in the log-polar transformation is the possibility of trying a number of fixation points (origins of the polar coordinate system) and choosing the best. The question remains whether the subjects try a number of origins and choose the best tour, or perhaps subjects have their idiosyncratic preferences for the origins. In fact, it

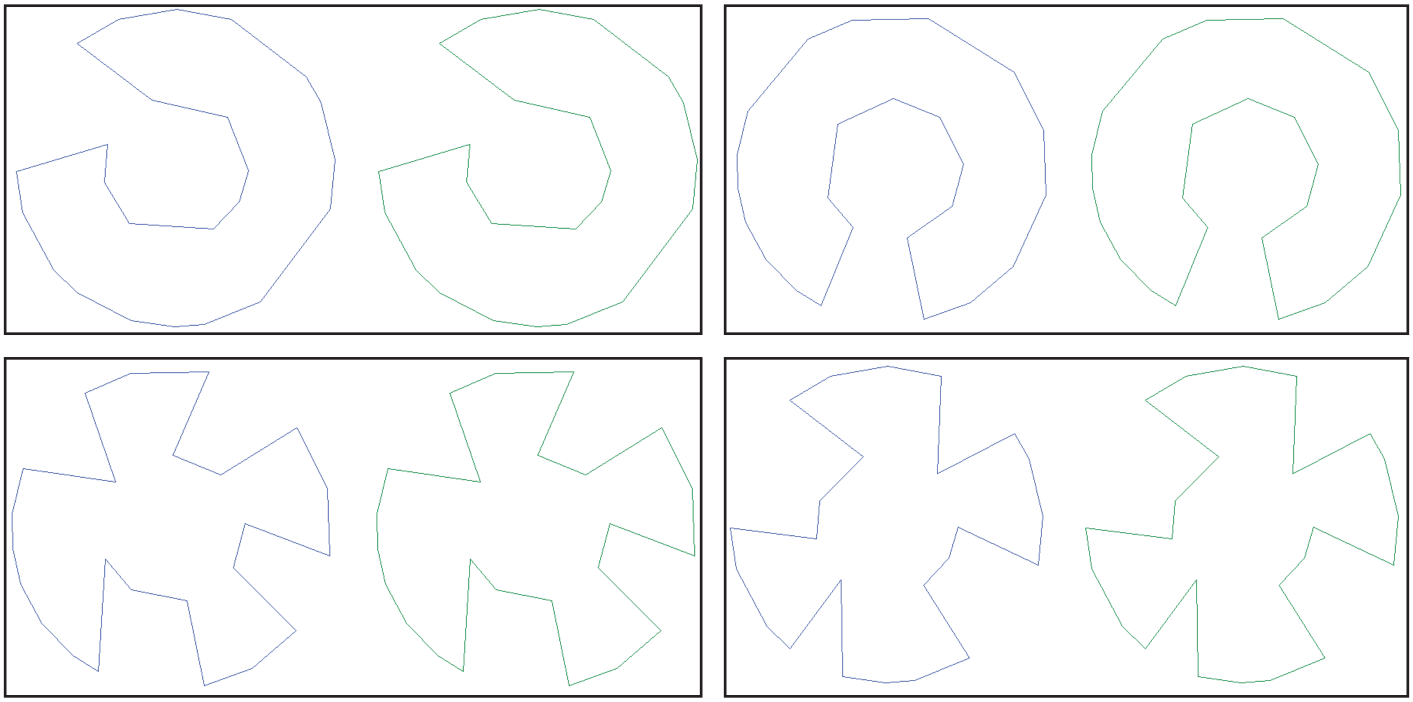

Figure 11.
Selected tours generated by humans (blue) that were replicated by the clustering algorithm (green) in log-polar space. The top two comparisons show examples of keyhole tours while the bottom two show examples of zigzag tours.

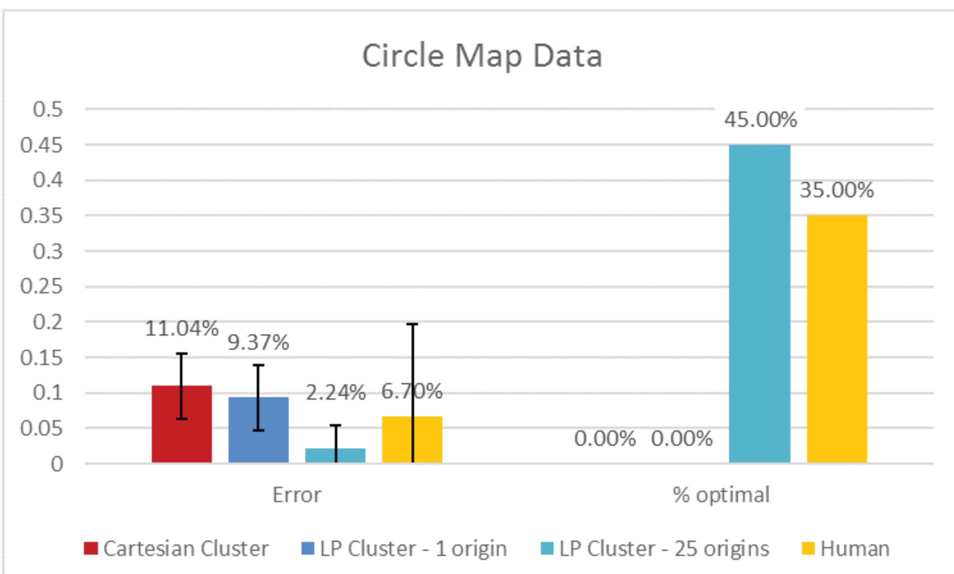

Figure 12.
The error and proportion of optimal tours in the clustering algorithm solutions in Cartesian coordinates, best of 1 clustering algorithm solutions in log-polar coordinates, best of 25 clustering algorithm solutions in log-polar coordinates, and human solutions of the concentric circle patterned maps. Error bars show standard deviation.

is possible that the subjects modify the location of the origin as they produce the tour. Recording the subject's eye fixations may shed light on this process. Alternatively, the subject may be shown a single tour for a short period of time with controlled fixation position and asked whether the tour looks optimal.

In conclusion, our study provides some evidence that the human mind may use representations that are not identical with the representations in which problems are defined or presented. This way, the subjects may, in general, be solving a problem that is different from the problem given by the experimenter.

The choice of the representation may be dictated by cognitive resources, by the efficiency of producing near-optimal solutions, by constraints of how the brain represents the incoming stimuli, and by the intrinsic aspects of the problem itself. This means it is possible that when subjects solve a TSP problem, they do so using distance approximations that were generated very early in visual processing. To use the six matches problem mentioned in the beginning of this paper, the physical constraints of a flat surface on which matches are laid and the operation of gravity, which makes it difficult to construct a 3D form from the matches, naturally suggests the 2D space as the adequate representation of the problem. So, our results are consistent with the recent suggestion put forth by Carruthers (2015). She suggests that cognitive psychologists interested in problem solving should study not only which algorithms and heuristics are used, but also in which space they operate. Hence, the role of mental representation of problems may provide a common denominator for insight problems and problems of search.

ACKNOWLEDGMENTS

Pizlo was supported by the National Eye Institute of the National Institutes of Health under award number 1R01EY024666-01. The authors thank Pranav Marla and TaeKyu Kwon for their help in the early stages of this project.

REFERENCES

Applegate, D., Bixby, R., Chvatal, V., & Cook, W. (2006). Concorde TSP solver.

Carruthers, S. (2015). *The role of the goal in problem solving hard computational problems: Do people really optimize?* (Doctoral dissertation). University of Victoria, Canada.

Garey, M. R., & Johnson, D. S. (1979). *A guide to the theory of NP-completeness* (p. 70). New York: W. H. Freemann.

Haxhimusa, Y., Kropatsch, W. G., Pizlo, Z., & Ion, A. (2009). Approximative graph pyramid solution of the E-TSP. *Image and Vision Computing*, *27*(7), 887–896.

Kwon, T., Agrawal, K., Li, Y., & Pizlo, Z. (2016). Spatially-global integration of closed, fragmented contours by finding the shortest-path in a log-polar representation. *Vision Research*, *126*, 143–163.

Lawler, E. L., Lenstra, J. K., Kan, A. R., & Shmoys, D. B. (1985). *The traveling salesman problem: A guided tour of combinatorial optimization* (Vol. 3, pp. 1–463). E. L. Lawler (Ed.). New York: Wiley.

MacGregor, J. N., & Ormerod, T. (1996). Human performance on the traveling salesman problem. *Perception & Psychophysics*, *58*(4), 527–539.

Pizlo, Z., & Li, Z. (2005). Solving combinatorial problems: The 15-puzzle. *Memory & Cognition*, *33*(6), 1069–1084.

Pizlo, Z., Stefanov, E., Saalweachter, J., Li, Z., Haxhimusa, Y., & Kropatsch, W. G. (2006). Traveling salesman problem: A foveating pyramid model. *Journal of Problem Solving*, *1*(1), 8.

Schinzinger, R., & Laura, P. A. (2003). *Conformal mapping: methods and applications*. Courier Corporation.

Schwartz, E. L. (1980). Computational anatomy and functional architecture of striate cortex: A spatial mapping approach to perceptual coding. *Vision Research*, *20*(8), 645–669.