Journal of Problem Solving

Development of the PEBL Traveling Salesman Problem Computerized Testbed

Shane T. Mueller,[1] Brandon S. Perelman,[1] Yin Yin Tan,[1] and Kejkaew Thanasuan[1]

[1] Michigan Technological University

Correspondence:
Correspondence concerning this article should be addressed to Shane T. Mueller, Department of Cognitive and Learning Sciences, 1400 Townsend Drive, Michigan Technological University, Houghton, MI, 49931, or via email to shanem@mtu.edu.

Keywords:
Traveling Salesman Problem, individual differences, cognitive testing, problem solving

The traveling salesman problem (TSP) is a combinatorial optimization problem that requires finding the shortest path through a set of points ("cities") that returns to the starting point. Because humans provide heuristic near-optimal solutions to Euclidean versions of the problem, it has sometimes been used to investigate human visual problem solving ability. The TSP is also similar to a number of tasks commonly used for neuropsychological assessment (such as the trail-making test), and so its utility in assessing reliable individual differences in problem solving has sometimes been examined. Nevertheless, the task has seen little widespread use in clinical and assessment domains, in part because no standard software implementation or item set is widely available with known psychometric properties. In this paper, we describe a computerized version of TSP running in the free and open source Psychology Experiment Building Language (PEBL). The PEBL TSP task is designed to be suitable for use within a larger battery of tests, and to examine both standard and custom TSP node configurations (i.e., problems). We report the results of a series of experiments that help establish the test's reliability and validity. The first experiment examines test-retest reliability, establishes that the quality of solutions in the TSP are not impacted by mild physiological strain, and demonstrates how solution quality obtained by individuals in a physical version is highly correlated with solution quality obtained in the PEBL version. The second experiment evaluates a larger set of problems, and uses the data to identify a small subset of tests that have maximal coherence. A third experiment examines test-retest reliability of this smaller set that can be administered in about five minutes, and establishes that these problems produce composite scores with moderately high (R = .75) test-retest reliability, making it suitable for use in many assessment situations, including evaluations of individual differences, personality, and intelligence testing.

The traveling salesman problem (TSP) is a combinatorial optimization problem in which the solver attempts to find the least-cost (e.g., shortest, cheapest, fastest) route that visits each one of a fixed number of nodes in a network and returns to the starting location, with known fixed costs of travel between each pair of nodes. Like the general TSP, the Euclidean variant (E-TSP) is also NP-hard (Papadimitriou, 1977), but constrains the problem so that the locations are dispersed on a plane, and the cost of moving between nodes is the Euclidean distance between them. Along with serving as a useful exercise in optimization (e.g., Applegate, Bixby, Chvátal, & Cook, 2006), human-solved versions of the TSP (which permit approximate solutions) have become a method of growing interest for understanding visual problem solving, navigation, and planning (see MacGregor & Ormerod, 1996; MacGregor & Chu, 2011). Mueller and colleagues (Mueller, Jones, Minnery, & Hiland, 2007, Mueller, 2008; 2010) identified the problem as measuring an important aspect of biological intelligence, forming one part of a modern-day embodied Turing test, and have also connected it to naturalistic search and mental models of pathfinding (Mueller, Perelman, & Simpkins, 2013; Perelman & Mueller, 2013a; Perelman & Mueller, 2013b; Perelman, 2015; Perelman & Mueller, 2015).

In many of these contexts, the human-solved version of the task has primarily been used to understand underlying processes related to visual problem solving, and its properties in this context are becoming well understood. In addition, some have suggested that skill in solving the TSP may differ consistently among individuals, and so it may be useful in assessing differences in cognitive function across individuals. Although initial findings (e.g., MacGregor & Ormerod, 1996) found little systematic individual differences (probably because solutions were all close to optimal), subsequent studies have suggested that individuals may systematically differ in the efficiency with which the task can be solved, and that performance may be correlated with measures of fluid intelligence (Vickers, Butavicius, Lee, & Medvedev, 2001; Vickers, Mayo, Heitmann, Lee, & Hughes, 2004; Burns, Lee, & Vickers, 2006; Chronicle, MacGregor, Lee, Ormerod, & Hughes, 2008). Furthermore, because the task bears similarity to a

http://dx.doi.org/10.7771/1932-6246.1169

number of other tests related to planning, spatial reasoning, and problem solving, it may offer a complementary yet convergent measure of problem solving ability.

Its limited use in assessment contrasts with several similar tasks that have received widespread use. For example, the trail-making test (Reitan, 1955; 1958) and its computerized variants (e.g., Piper et al., 2012) are fairly similar, but the trail-making test requires following a given path rather than constructing one. Nevertheless, the path followed (especially in Form A) tends to be fairly close to the shortest possible path (see Vickers & Lee, 1998). Similarly, the Corsi block-tapping test (Corsi, 1972; Kessels, van Zandvoort, Postma, Kappelle, & de Haan, 2000) has become a widely-used measure of visual serial order memory, requiring a participant to reconstruct a given sequence (analogous to a route) presented among a set of up to ten spatial locations. However, the sequences tend not to be smooth paths, but rather haphazardly organized in the display. In a more abstract sense, two commonly used tower tests—the Tower of London (Shallice, 1982) and the Tower of Hanoi (Kotovsky, Hayes, & Simon, 1985) look at how participants find a shortest path in a problem space between a starting and goal configuration, and so have a very similar approach to the TSP. Along with problem solving in general, these tests have often been used to understand plan formation and the impact that neuropsychological disorders have on planning. Finally, a number of visual search tasks (e.g., Wolfe, 1994; Halverson & Hornoff, 2011) require a participant to efficiently search through a spatial array. Although these tasks do not require drawing a path, and the studies typically do not record eye movement trajectories associated with those paths (although, see Araujo, Kowler, & Pavel, 2001 for an eye tracking example), the most efficient eye movement path can be obtained by solving a variant of the E-TSP related to that task.

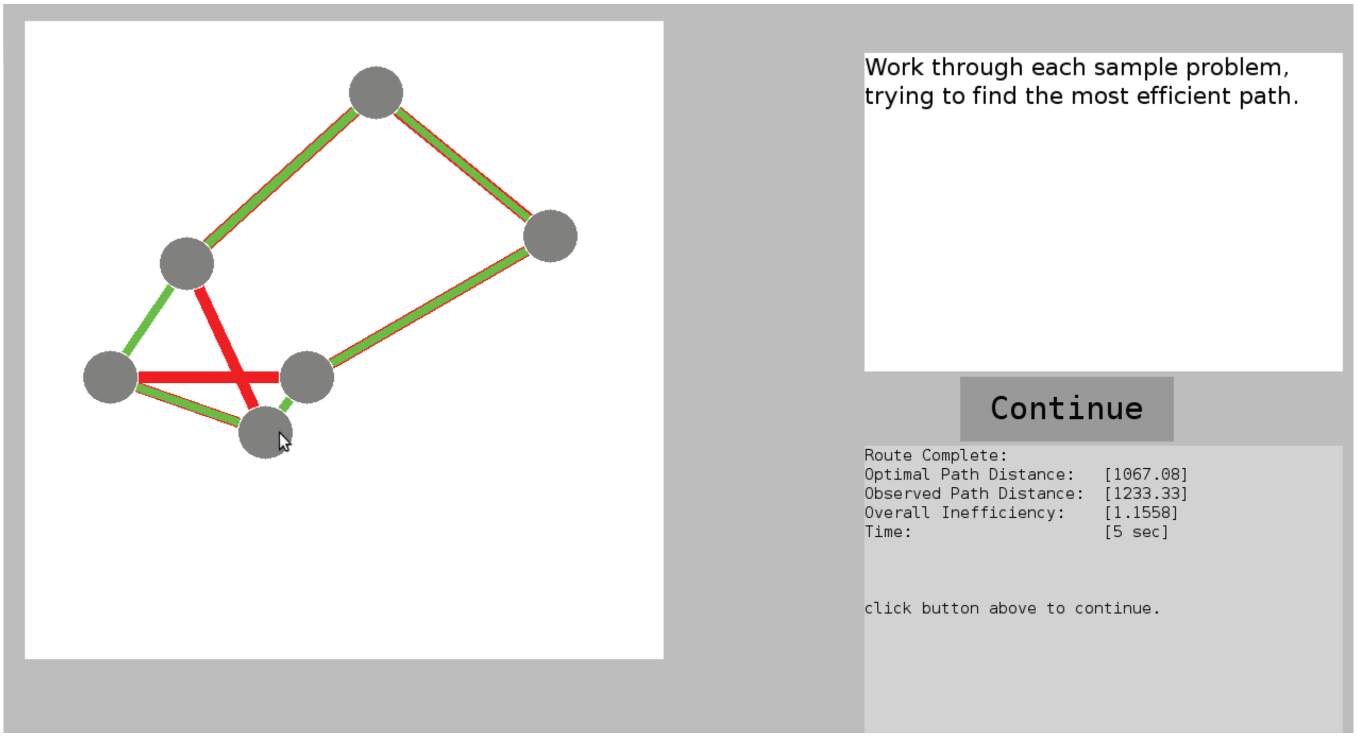
Two reasons for its limited use in assessment are (1) there have previously been no widely-available systems or standard problems for administering the test; and thus (2) there are few experiments establishing psychometric properties of such tests, such as reliability, problem difficulty, and convergent or construct validity. In response to this, we have developed a freely available E-TSP task that is included as part of the Psychology Experiment Building Language Test Battery (PEBL Version 0.14; Mueller, 2014), which has been described previously (Mueller, 2010) and has been used for data collection in several hundred publications across a broad range of topics (see Mueller & Piper, 2014). Since its release, Version 0.14 of PEBL has been downloaded more than 20,000 times, which has already made the PEBL TSP widely available to researchers and clinicians around the world. The PEBL Test Battery includes around 100 other tests, including related tasks such as trail-making, Corsi blocks, visual search, Tower of London, and Tower of Hanoi, and so it enables researchers to include the TSP alongside other measures of neuropsychological function to assess convergent validity of different measures. In this paper, we describe the basic software and present three experiments conducted on subsets of 50 test problems (five 6-node practice problems and 15 problems each of 10, 20, and 30 nodes) included in the software. These experiments help to establish the reliability and validity of the method, and basic behavioral measures of problem difficulty across the problems. This analysis will provide sufficient information to allow researchers to use the PEBL TSP test in a number of settings, to evaluate individual performance against our sample, to examine cognitive model performance against human data, to test novel TSP problems, and to help establish the validity of the measure in different contexts. First, we will provide a high-level overview of the test and describe rationale for several of its design features.

OVERVIEW OF THE PEBL TSP TASK

In the PEBL TSP task (see Figure 1), participants see the complete layout of points (i.e., cities or nodes), with unselected points colored red and a given starting location indicated in grey. Solvers complete the path by clicking the points consecutively, and the points can be selected in any order. Each subsequent click turns the chosen point from red to grey, and draws a thick red line between the previously chosen point and the newly chosen point. No backtracking is permitted. When the final unselected point is clicked, a line is drawn from that point back to the first point (i.e., the software automatically closes the solution), and a green path circuit is drawn to show the shortest (i.e., optimal) solution. After completing each problem, statistics regarding the solution are shown in the lower right, including the shortest path length, the produced path length, the ratio of these values (i.e., inefficiency), and the time taken to complete the tour. The timing and order of each click are recorded, as well as the computed inefficiency. After the feedback is reviewed, participants click a button marked "OK," to move on to the next problem. A complete description of the use and administration of the PEBL TSP task is provided in Appendix D.

All implementations of TSP-like tests have made decisions about instructions, interface, and interaction that are often irrelevant to underlying theory yet may influence task performance. In developing the PEBL test, most of the design decisions were made to facilitate faster administration, to promote solution modes that rely on intuitive and visual processes rather than highly deliberative problem solving, to use interactions that can be easily learned, to enable consistent administration conditions, and to allow for useable data traces. We review some of the details next and provide some rationale for these decisions, with comparison to previous experiments. Some aspects of tests include:

Sequential solving mode. Some studies (e.g., MacGregor & Ormerod, 1996; van Rooij, Schactman, Kadlec, Stege, 2006) have used hand-drawn administration, in which participants drew from point-to-point on paper to complete the tour.


Work through each sample problem, trying to find the most efficient path.

```
Route Complete:
Optimal Path Distance:   [1067.08]
Observed Path Distance:  [1233.33]
Overall Inefficiency:    [1.1558]
Time:                    [5 sec]


click button above to continue.
```

Figure 1.
Screenshot of PEBL TSP task after completing a practice problem. Red line indicates chosen path; green line indicates shortest path. Numeric feedback is given in the lower right window.

Typical instructions imply that participants should solve the problem sequentially, but this may be difficult to enforce. In contrast, some computerized solutions (e.g., Vickers et al., 2001; 2004; Dry, Lee, Vickers, & Hughes, 2006) have used computerized methods that permit non-sequential solutions using a click-drag-release method, while most others (Graham, Joshi, & Pizlo, 2000; Pizlo, Stefanov, Saalweachter, Li, Haxhimusa, & Kropatsch, 2006; Chronicle et al., 2008; Acuña & Parada, 2010) have required sequential solutions.

We chose to adopt sequential solutions for a number of reasons. First, it is most closely analogous to both physical navigation in general, as well as a physical TSP task we used in Experiment 1a. In these contexts, the problem must be solved by constructing the tour in sequence. Second, it permits a simpler interface that is easier to learn, which may be especially advantageous for testing children or older adults. Third, we hypothesize it encourages solvers to use intuitive visual strategies (in contrast to more deliberative cognitive ones) that are an important aspect of both empirical (e.g., van Rooij et al., 2006) and theoretical (e.g., Graham et al., 2000; MacGregor & Chu, 2011) TSP research. Finally, we suspect that even when non-sequential solutions are permitted, they are likely to be used only occasionally.

Backtracking and restarting. Although paper administration typically discourages backtracking, several computerized versions we are aware of have permitted backtracking in one form or another. Vickers et al. (2001) allowed individual links to be deleted by selecting and pressing the delete key; Graham et al. (2000) and Pizlo et al. (2006) allowed the previous move to be undone or the entire problem to be reset; and Acuña & Parada (2010) provided a clickable undo button. In contrast, Miyata, Watanabe, & Minagawa (2014) did not permit backtracking, whereas others did not mention backtracking in their method descriptions—MacGregor (2014) stated "participants completed a tour by pointing and clicking," whereas Kong & Schunn (2007) instructed participants to "indicate the path using mouseclicks." We chose to not permit backtracking for several reasons, many of which are identical to our rationale for using a sequential solving mode. First, it is more closely analogous to a physical version of the test (see Experiment 1a). Second, it is consistent with real-world movement: once you physically travel a route, you cannot un-travel it without cost. Third, in related problem solving tasks such as the Tower of London, we know of no version that permits cost-free backtracking. Although restarting is not uncommon in these tasks, it is often limited, and so also not without cost. In addition, by avoiding backtracking the interface can be simplified (no additional buttons or secondary mouse clicks for undo operations are needed), which in turn allows the test to be performed with less training and potentially among a broader population, and it is likely to produce faster overall trials. And finally, it also allows for a more easily processed data record, because each target click event is the one and only time that target is visited. Nevertheless, it remains an open empirical question whether backtracking will increase or reduce reliability and validity of TSP performance measures, and what impact it has on the overall solutions produced.

Starting location. Another interface decision relates to whether the participant is given a starting node to begin their tour. For an optimal tour, starting location is arbitrary, but this may not be true for human-generated tours (Perelman, 2015, showed evidence consistent with the suggestion that tour performance may depend on starting location). MacGregor (2012; 2014) showed that participants tended to choose starting points on the periphery of the problem more often than chance would dictate, again suggesting that starting location may play a role in solution efficiency. Although most past implementations have allowed participants to choose their own starting location, this is not universally true (cf. Miyata et al., 2014). There is some chance that the ability to choose a good starting location could account for a proportion of individual differences in the task, and so we chose to always initiate the problem at some node; either randomly-chosen or identical across all participants. In assessment settings, researchers typically favor using identical problems so as to reduce the effect of materials and highlight individual differences, and so using a single fixed starting point in each problem is consistent with good test design. Furthermore, in physical (and real-world) TSP problems, including the one used in Experiment 1a, the starting location is typically constrained to be the physical location of the traveler.

Visual interface. Several other details about the visual interface warrant discussion. Some aspects of the visual interface will tend to impact motor movement and visual search processes probably incidental to the problem solving processes usually of interest to TSP researchers, and these might be outside the cognitive functions that might be uniquely assessed with the task. For example, target size will impact aimed movement times involved in mouse targeting, and color cues (i.e., color changes when a node is visited) will reduce the need to engage in deliberate visual search to find unvisited nodes. Nevertheless, target size has rarely been reported in the past (Graham et al., 2000 noted that the targets were “small,” but few others have given any details). Several past approaches have used color-change to indicate that a node has been visited (e.g., Graham et al., 2000; Pizlo et al., 2006). This is not feasible for paper administration, but may provide a more consistent completion time profile as it helps participants prevent accidentally missing a node, and needing to search the entire field to find the missed node at the end of a solution.

Our choices were heavily influenced by another similar test: the PEBL Trail-making test (e.g., Piper et al., 2012). That task used 25-pixel radius circles that can easily be targeted with either a mouse or touchscreen, and color change to reduce contrast of visited nodes. That task has been demonstrated to be useful over the lifespan (see Piper et al., 2012), and so provides a reasonable starting point for the current interface.

Feedback. Feedback can be important for motivating and teaching participants the goal state of any task. Paper-administered tests typically do not show feedback about the best solution, but computerized solutions can, both in terms of the actual route and measures such as inefficiency (how much longer, proportionally, a solution is compared to the shortest route). Chronicle et al. (2008) reported giving a visual depiction of the best route, whereas Dry et al. (2006) gave numerical feedback about the length of the participant’s solution in comparison to the best route. Most other experiments do not report whether feedback on the path was given (e.g., neither Graham et al., 2000, nor Vickers et al., 2004 mention this). One study we are aware of (Acuña & Parada, 2010) provided extensive repetition with feedback on tour length and showed that participants were able to learn to produce more efficient tours as problems were repeated. We chose to provide feedback because it allows participants to maintain engagement and motivation in the task, and to learn whether their solution strategies are effective. Feedback was given both in the form of a green path depicting the shortest route overlaying their drawn path (i.e., the best solution), as well as a numerical score so participants could assess the length of their path versus the best solution (see Figure 1).

Problem set. Many past studies of TSP performance have examined how performance profiles change as the problem size and complexity changes. We have selected candidate problems of size 10, 20, and 30 to test, generated from two-dimensional uniform distributions constrained to have low point-to-point overlap. Larger and smaller problems have been studied previously, as well as ones with non-random contrived layouts (see MacGregor & Ormerod, 1996; Dantzig, Fulkerson, & Johnson, 1959). Our goal in this paper is to establish the psychometric properties of these particular problems, but the PEBL TSP allows other problems to be specified, and may therefore serve as a useful platform for testing other problem configurations.

Summary of software design. It is beyond the scope of this paper to explore the complete effects that alternatives to these design decisions have on performance, and some of them are likely to have measurable impacts. As an analogy, there are probably more than a dozen distinct versions of the Tower of London task in use, and many of them differ specifically in terms of the same types of properties we have just discussed—visual layout, problem set, interaction mode, feedback, timing, instructions/goals, and rules, whether backtracking and resetting is permitted, etc. Future research may provide useful guidance as to how these different properties impact solutions. Because the source code is available for inspection, modification, and redistribution, the impact of such decisions can be explored within this testing system and variations distributed to others. Furthermore, in the Discussion section, we will compare some of our results to those of previous implementations to examine the extent to which they are similar.

OVERVIEW OF EXPERIMENTS

Next, we will describe the results from a series of experiments that establish general aspects of the reliability and validity of the PEBL TSP, especially as a means of measuring the relative difficulty of different problems and systematic individual differences among people. Across all experiments, a common superset of problems was used: five six-node practice problems and 15 problems of each size 10, 20, and 30. Problems are identified with both a size and a problem code (i.e., 10-01, 20-12, 30-05, etc.), allowing performance to be linked to a particular test problem.

First, we will report the results of two related experiments (1a and 1b) that establish the test-retest reliability of the PEBL TSP, show how performance on the task relates systematically to performance on a physical TSP analog, and demonstrate that solution efficiency is insensitive to some impacts of physical stress. Then, we will report the results of a more systematic study of all 45 TSP problems (Experiment 2), and report detailed performance statistics on these problems, which could allow problem selection and model testing. Based on these results, we selected a small 15-problem subset and report the results of a test-retest study (Experiment 3) establishing the reliability of the TSP inefficiency measure.

EXPERIMENTS 1A AND 1B

In Experiment 1, we report two experiments conducted as part of a single project in which we evaluated the impact of mental stress induced by heat burden on a series of cognitive tasks whose summary values were reported by Mueller et al. (2010) and Mueller et al. (2013). The goal of these experiments was to establish how different cognitive functions were impacted by the physical and heat strain induced by wearing military protective gear (i.e., gas-impermeable clothing, gloves, goggles, and masks that enable the military to work in a potentially contaminated environment). Each of these experiments involved testing participants twice; once while wearing normal duty clothing, and once while wearing the standard Mission-Oriented Protective Posture (MOPP) gear. This permitted assessing both test-retest reliability and the impact of physical stress on performance in this task, and comparing (in Experiment 1a) performance to a physical implementation of the TSP.

METHOD

Participants. Experiment 1a involved nine volunteer participants who were off-duty members of the Wyoming National Guard. Experiment 1b involved 12 volunteer participants who were off-duty U.S. Air Force security personnel. Testing took place at a Wyoming National Guard facility in Guernsey, WY. In both groups, ages ranged from early adulthood to middle age. All participants were paid for completing the study (which included numerous other tests and travel) over a 3-day period of time. Participants signed informed consent forms prior to participating in the study, and the study was approved through an Institutional Review Board.

PEBL TSP procedure. In both experiments, the PEBL TSP task was used, with solutions entered via capacitive touch-screen monitors to avoid keyboard or mouse impairment from protective gloves. A randomly-selected starting position was chosen for each participant on each problem.

Experiment 1a tested seven problems (04 through 10) of each problem size (10, 20, and 30 node) twice, once in each of two clothing conditions. Similarly, Experiment 1b tested 12 problems (04 through 15) of each problem size (10, 20, and 30 node) twice. In addition, five small (size 6) practice problems and Problems 01, 02, and 03 of each problem size were used only once for practice on the first testing session. These problems were originally generated via the Concorde TSP software Applegate, Bixby, Chvátal, & Cook, 2003), which also provided the shortest-path solutions by which performance is evaluated.

Physical TSP procedure. In the physical TSP task, four problems each of size, 10, 15, and 20, were generated and the shortest-path solution for each was computed via the Concorde TSP software. To simplify problem layout, all nodes were located on a 10 x 10 grid system, which was laid out in a room with 12 inches between each adjacent gridline. The task was implemented by placing plastic cups at the locations designated by the problem design, with each cup containing a plastic straw. To solve the problem, participants had to simply collect a straw from each cup. While they were completing the problem, an experimenter entered their solution via a computer using an analog of the problem, clicking simultaneously with each target visit. This allowed solution order and rough approximations of solution time to be recorded. Starting and ending location for each problem was always a single 'home' location at the edge of the grid. Because a limited time window was given for performing the entire task, participants did not always complete all twelve problems in the allotted time. Consequently, mean performance efficiency across the completed problems was used as a dependent measure.

Design. The order of testing in clothing was counterbalanced across participants in both experiments; within each testing session, problem order was randomized. Participants wore their normal duty clothing (BDU or ACU) or the JSLIST ensemble over their normal duty clothing as part of the protective gear condition. Although not described here, participants took part in a series of other cognitive and physical activities, which in general produced increased physical and heat strain when wearing protective gear.

RESULTS

Experiment 1a. With the relatively small number of problems used, we will focus on performance within each problem set size. Protective gear had no impact on inefficiency on the computerized task, either within individual problem sizes or when comparing overall mean inefficiency (see Table 1). For the physical task, mean inefficiency when pooled across all problems was slightly better when wearing protective gear: an average of 1.10 versus 1.13. This difference was statistically significant, most likely because protective gear made movement more restrictive and slower, which in turn encouraged participants to produce shorter solutions. As problem size increased, mean inefficiency tended to increase, as did the test-retest correlation (consistent with the conclusions of Chronicle et al., 2008). Importantly, a single composite score of mean inefficiency under each clothing condition produced a test-retest correlation of .77 for the computerized task, and .91 for the physical task, indicating substantial systematic individual variation in task efficiency within each task. Furthermore, the correlation between mean inefficiency (across participants) for the two tasks was .795, which was statistically significant (p=.018), indicating that solution efficiency of the physical and computerized tests systematically covaried across individuals.

Experiment 1b. Experiment 1b involved more participants and more problems of each type than Experiment 1a, but without the physical analog test. As with Experiment 1a, no systematic difference in inefficiency was attributable to the clothing conditions (see Table 2), even though protective gear produced measureable physiological stress, discomfort, elevated body temperature, and impairment on other attention-based tasks. Composite scores for each problem size were all significantly correlated across individuals when comparing the two clothing conditions, with correlations around or above .8. The greater correlations in this study stem partly from the use of more problems at each problem size, which likely produced more reliable composite measures for each individual.

DISCUSSION

Experiments 1a and 1b used the PEBL TSP test and enabled measures of test-retest reliability to be assessed in a small participant pool. Important results of this study include: (1) TSP performance was fairly resilient to physiological stress that induced mental strain measurable in a number of other cognitive tasks; (2) test-retest reliability on composite inefficiency scores is reasonable for as few as 7 larger problems of 20–30 nodes each; (3) test-retest reliability for 10-node problems is sometimes poor and sometimes reasonable. The higher reliability in Experiment 1b may be attributable to the use of 12 rather than 7 problems, or it may stem from

Table 1.
Mean inefficiency in Experiment 1a in each physical condition. Standard deviations shown in parentheses.

	Standard clothing	Protective gear	*t*-test	Correlation
Computerized Task				
Size 10	1.054 (.039)	1.034 (.027)	*t*(8) = 1.4, p = .2	.213, *p* = .58
Size 20	1.074 (.041)	1.081 (.052)	*t*(8) = -.43, p = .68	.581, *p* = .1
Size 30	1.102 (.063)	1.103 (.06)	*t*(8) = -.12, p = .91	.763, *p* = .016
All trials	1.077 (.05)	1.073 (.06)	*t*(8) = .45, p = .66	.777, p = .013
Physical Task				
Size 10	1.053 (.076)	1.07 (.091)	*t*(7)=2.1, p=.078	-.19, *p*=.65
Size 15	1.12 (.089)	1.10 (.132)	*t*(7)=-.33, p=.75	.52, *p*=.19
Size 20	1.15 (.101)	1.13 (.076)	*t*(7)=1.2, p=.28	.71, *p*=.057
All trials	1.13 (.053)	1.10 (.071)	*t*(7)=2.45, p=.043	.91, *p*=.002

Note: Only eight participants completed the physical test. Each problem size set in the computerized task consisted of seven problems, whereas they consisted of four problems in the physical task.

Table 2.
Mean inefficiency in Experiment 1b. Inefficiency was not impacted by clothing condition, but performance in the two clothing conditions was highly correlated across individuals. Standard deviations shown in parentheses.

	Standard clothing	Protective clothing	*t*-test	Correlation
Size 10	1.065 (.054)	1.068 (.057)	*t*(10) = .37, p = .7	.89, *p* < .001
Size 20	1.096 (.053)	1.111 (.069)	*t*(10) = 1.13, p = .28	.78, *p* < .001
Size 30	1.107 (.050)	1.118 (.050)	*t*(10) = 1.4, p = .19	.88, *p* < .001
All trials	1.090 (.046)	1.099 (.055)	*t*(10) = 1.0, p = .34	.87, *p* < .001

Note: Each problem size set contained 12 problems.

other aspects of sampling or experimental design, but it is consistent with results reported as far back as MacGregor & Ormerod (1996). Finally, (4) high correlations between average inefficiency on a physical analog task and the PEBL TSP indicate convergent validity of the method.

Together, these results suggest that a systematic examination of a larger set of PEBL TSP problems may help identify a set of candidate problems for assessing TSP skill. Experiment 2 was designed to assess a candidate problems more systematically.

EXPERIMENT 2

Experiment 2 sought to refine and formalize the testing performed in Experiments 1a and 1b in a laboratory setting, and to establish basic performance profiles for different problems.

METHOD

Participants. Twenty-four undergraduate students from Michigan Technological University participated in the study in exchange for partial credit toward a research requirement in a psychology course.

Materials, stimuli, and design. The study was approved via the Michigan Technological University Human Subjects review board. After providing informed consent and taking part in a related navigation and planning study, participants completed a total of 50 E-TSP problems using the PEBL software described earlier. Testing was performed in one of three separated cubicles in a small testing room so that three or fewer participants were tested at one time. An experimenter was present throughout testing to answer any questions. Testing was performed using Dell Precision T1600 PCs running Windows 7, using a Planar PX2230MW 21.5” touchscreen monitor at a resolution of 1920 × 1080 (responses were made using a mouse). After five practice problems, participants completed 45 test problems (15 problems each of size 10, 20, and 30, in a randomized order). Each participant was given a randomly selected starting location for each problem.

RESULTS

In this study, we will examine several dependent variables to examine how particular forms differ in overall performance. These include:

- Inefficiency—the ratio of the distance of the produced path and the shortest known solution.
- Planning time—the time taken to make the first click.
- Execution time—the time to complete the route (including planning time).

Inefficiency. Figure 2 shows the mean inefficiency for the problems we tested in this experiment, whose mean increased from 1.01 for the small practice problems to 1.077 for the 30-node problems. Individual problems within a problem size ranged in their mean solution inefficiency as well; this may be because of systematic differences in problem difficulty or variability among participants. Because inefficiency is important for examining individual differences, we also provide histograms for each problem in Appendix B. Typically, on each problem, most participants produced solutions within 5 or 10% of optimal. For more difficult problems, a greater number of participants produced inefficient paths (i.e., greater than 10% of optimal), but the best solutions were almost always very close to optimal.

To further examine inefficiency, we computed two separate regression models for comparison using the R Statistical Computing Language Version 3.0e2 base package lm function (R core team, 2013) and the Anova function of the car package (cf. Fox & Weisberg, 2010). One model predicted inefficiency based on a linear effect of problem size and a categorical effect of participant identity (essentially computing a different intercept for each participant). A Type-II ANOVA showed that both problem size ($F(1, 1013) = 45.8$, $p < .001$) and participant identity ($F(23, 1013) = 2.90$, $p < .001$) were reliable predictors. However, the residual standard error was relatively large (0.092 units) and the multiple R^2 was relatively low (.099). To understand whether there was considerable consistent cross-test variability not accounted for by problem size, we also tested another model that replaced problem size with a categorical predictor of problem identity. A Type-II ANOVA for this model showed significant effects of both problem identity ($F(44, 970) = 2.24$, $p < .001$) and participant identity ($F(23, 970) = 2.90$, $p < .001$), but the residual standard error was nearly unchanged (0.092), and the multiple R^2 was similarly low (.14). Moreover, an ANOVA comparing these two models showed no significant differences, $F(43, 970) = 1.2$, $p = 0.16$, indicating that there were no systematic problem-specific differences that could not be accounted for by problem size alone. Across the two models, problem size accounted for only 4% of the variance, whereas replacing problem size with problem identity increased this to 8.7%, and participant identity accounted for a total of 5.8% of the variance. These relatively low proportions of variance accounted for suggest that there are fairly substantial idiosyncratic differences for individuals on tests, such that any individual test will be poor at predicting consistent individual factors related to problem solving, and combinations of scores will be necessary to create a reliable measure.

Planning time. Although in our test no specific instructions were given to create a plan before starting, these are common instructions in other problem-solving tasks such as the Tower of London. Furthermore, past studies have examined measures appropriate for assessing planning in a TSP task (e.g., Basso et al., 2001). Thus, we examined the time taken to make the first click as an index of planning time, especially to assess whether the time spent planning depends on problem size. Figure 3 shows this planning time over the four problem sizes tested. A linear regression model including problem size and participant identity as predictors (but excluding the practice problems) showed there was a small reliable positive impact of problem size on planning time (31.8 ms/city), $t(1013) = 2.2$, $p = .027$, and the model produced a multiple R^2 of .28 (with problem size accounting for only 0.3% of the total variance). A Type-II ANOVA showed that participant identity was also a reliable predictor, $F(23, 1013) = 17.6$, $p < .001$. A second regression model substituting a categorical predictor of problem identity for problem size showed that problem identity was also a significant predictor, $F(44, 970) = 1.62$, $p = .007$, and an ANOVA test showed a significant difference between these two models, $F(43, 970) = 1.50$, $p = .01$, indicating consistent problem-specific planning time differences exist that cannot be explained by problem size alone. However, the planning time effect remains relatively small, with average planning time increasing by only 600 ms as problem size increased from 10 to 30 points, in comparison to execution times on the order of tens of seconds.

Although planning time was relatively small, it might still be possible that planning time is related to overall problem solving inefficiency insofar as participants who take longer to plan may produce better solutions. We examined this both in relation to the specific problems and across individuals by aggregating planning time and inefficiency across both problem and participant, and computing their Pearson correlations. We found that solution inefficiency was not significantly correlated with planning time, either when averaged over participants ($R = -0.25$, $t(23) = -1.2$, $p = .22$) or when averaged over test ($R = 0.18$, $t(43) = 1.2$, $p = .23$). Nevertheless, the non-significant negative correlation across participants indicates people who spent more time planning produced slightly shorter paths, and the positive relationship across test indicates that more difficult problems produced slightly longer planning times. It is possible that such relationships would have been more robust had specific planning instructions been given, and so future studies may benefit from using planning instructions and measuring planning time.

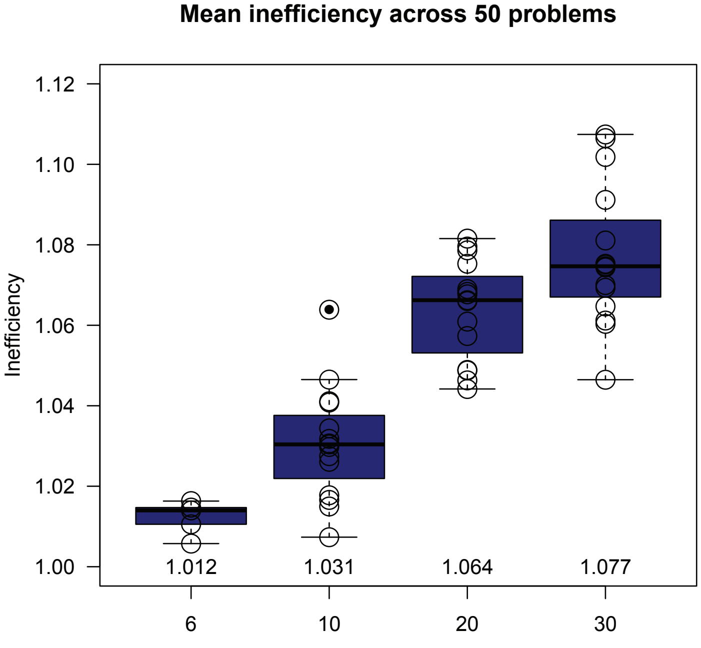


Figure 2.
Inefficiency scores (obtained/optimal path length) across problem sizes. Each circle shows the mean value produced for a single problem, and the value at the bottom of each column indicates mean inefficiency for each problem size. Boxplot shows median and inter-quartile range.

Execution time. Researchers have often examined execution time to understand the computational complexity of the underlying heuristics used to solve the TSP. Surprisingly (and unlike optimal algorithms), humans tend to produce solutions that are close to optimal in times that are linearly related to problem size (see MacGregor & Ormerod, 1996). In the present experiment, we saw large and robust effects of problem size on execution time (see Figure 4), and the best-fitting regression model obtained an equation estimating a completion time of 2377 + 638.5 ms/node. As before, a regression model using participant identity and problem size showed (via a Type-II ANOVA) significant effects of both problem size ($F(1, 1013) = 942$, $p < .001$) and participant ($F(23, 1013) = 41$, $p < .001$), with a multiple R^2 of .65 and a residual standard error of 5471 ms. A second regression substituting test for problem size was significantly better than the problem-size model according to an ANOVA test ($F(43,970) = 1.6$, $p = .009$), indicating significant problem-related aspects that were not accounted for by problem size. Overall, problem size alone accounted for 32.5% of the variance, replacing this with problem identity only improved this to 34.8%, and participant identity accounted for an additional 32.5%.

Measuring Overall TSP Performance. One natural measure indicating the overall ability to solve TSP problems is the mean inefficiency score across problems. Because inefficiency depends on problem size, one might consider estimating a slope and intercept for each individual. If these were independent, the two measures would indicate overall inefficiency (for the intercept) and how inefficiency is impacted by problem complexity (for the slope). However, in Experiment 2 we found that the slopes and intercepts estimated for each individual were highly negatively correlated (-.50), which indicates that the two estimates tend to trade off. This is likely to be a problem of estimation rather than a true relationship, and it means that intercept and slope may not be good candidates for assessing overall TSP skill. An alternative is to fit a slope constrained to have an intercept of 1.0 for problems of size 0 (which is a reasonable assumption). When we computed this for Experiment 2, these values turned out to be almost perfectly correlated with mean inefficiency (*R* = .992) and so it would appear that mean ineffi-ciency is likely to be the most robust and simplest performance measure. In our sample, mean inefficiency had a mean of 1.058 with a standard deviation of .0238, ranging between 1.02 and 1.113 across individuals. This range was roughly the same as the range of the mean inefficiencies across individual problems (which ranged from 1.00 to 1.09), which gives a range of about 4 standard deviations, and if we adjust inefficiency by subtracting 1.0, the worst performer's score was more than five times larger than the best. Inefficiency scores produce a Cronbach's α score of .8535, indicating a high degree of internal consistency (this finding is similar to the α score of .87 found by Vickers et al., 2004). To provide a means for comparing individual performance on tests to our sample, we have included summary statistics and histograms in Appendices B and C.

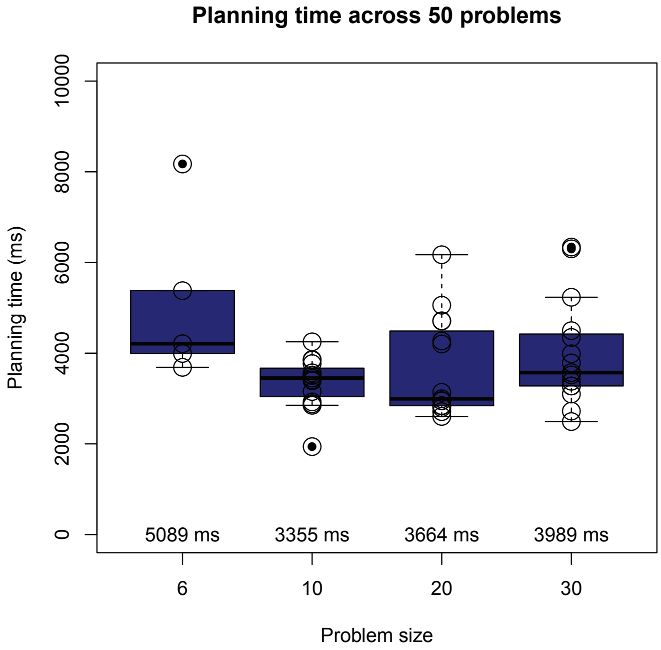


Figure 3.
Planning time (ms until first click) across problem size. Circles indicate mean planning time for each problem, and dark points indicate outliers. The relatively high planning times for the 6-node problems may be attributable to these problems being presented as practice problems.

DISCUSSION

Experiment 2 established the impact that problem size has on a number of dependent measures in the PEBL TSP Task. Solutions tended to get less efficient and to take longer as problem size increased, and this stemmed primarily from execution time rather than planning time. Both solution times and inefficiency depended on the size of the problem, and solution times also depended systematically on the problem. Together with the results of Experiment 1, the results of these experiments establish that composite scores from the PEBL TSP task differ reliably across people, are unimpaired by moderate physiological stress, and are systematically related to efficiency in a physical analog task.

However, the test set studied in Experiment 2 required solving 50 total problems, which often took close to 20 minutes to complete. It may be possible that by selecting a smaller subset of problems, we can create a test that can be administered in a brief time period (5–10 minutes) that provides similar levels of reliability and validity. This may enable future assessment of whether TSP solution efficiencies are related systematically to any other cognitive or personality measures, skills, and aptitudes, or are especially impaired in certain clinical populations or genotypes. Consequently, on the basis of the results of Experiment 2, we selected such a subset and examined the test-retest reliability of this subset directly in Experiment 3.

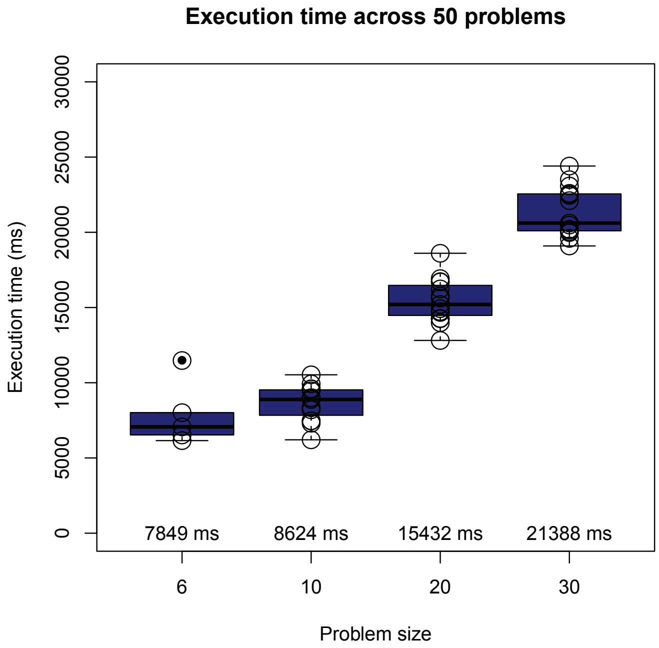


Figure 4.
Execution time as a function of problem size. Individual points represent distinct test problems. The value at the bottom of each column indicates mean execution time for each problem size.

EXPERIMENT 3

To conduct Experiment 3, we first sought to identify a subset of the 45 problems used in the previous experiments that can serve as a brief but robust measure of individual ability in TSP. To begin, we examined the results of Experiment 2, and computed the part-whole correlation of each problem to the average of the entire set (excluding the selected problem) across participants. These correlations are shown in Appendix C.

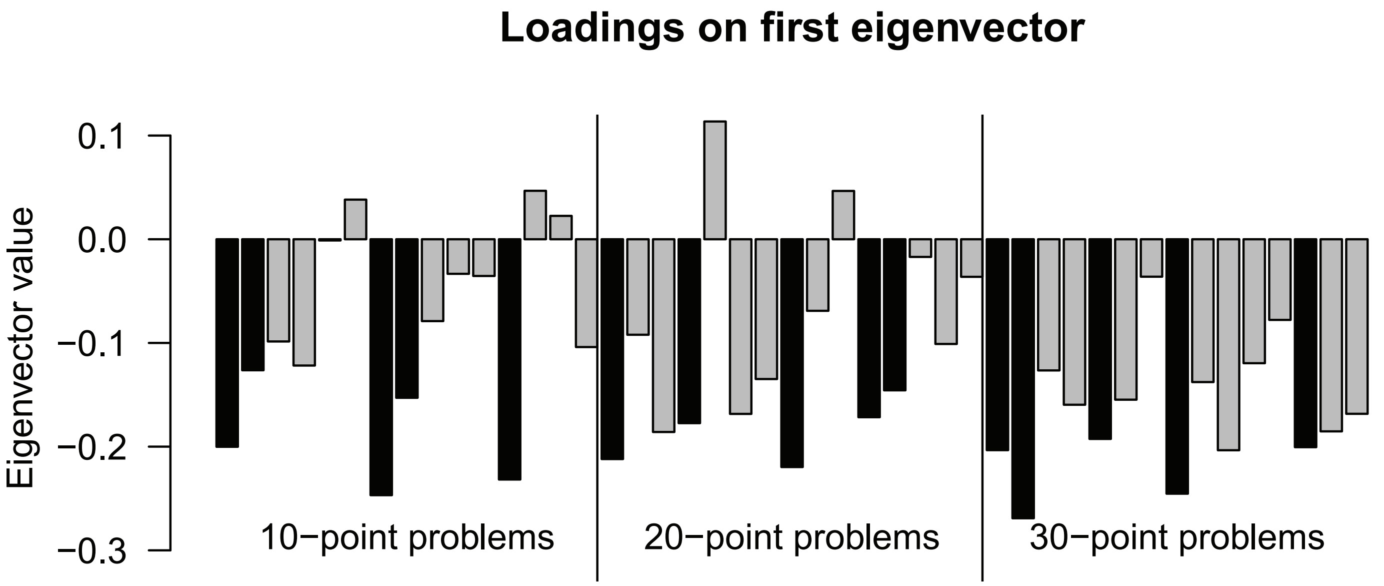


Figure 5.
Loadings on the first eigenvector showing how each test is related to the primary factor. Black bars indicate the 15 chosen problems with the highest part-whole correlations (five from each problem size).

These part-whole correlations differed across problems (ranging from -.29 to +.65), with the mean part-whole correlation increasing as problem size increased (10 nodes: +.23; 20 nodes: +.33; 30 nodes: +.43). This result is akin to the increase in test-retest reliability with problem size we saw in Experiment 1a.

Although part-whole correlation is smaller for the smallest problems (which indicates these measures are likely to be less reliable predictors of individual TSP ability), we felt it was important to include multiple problem sizes, in order to establish how efficiency and time changes parametrically with problem difficulty. Consequently, for our abbreviated problem set we selected the five problems of each set that had the largest part-whole correlations. To determine whether these form a coherent factor structure, we also examined the first eigenvector of the correlation matrix of the entire experiment to examine loadings of each problem. These are shown in Figure 5, with the selected problems displayed in black. Although the selected problems did not always correspond to the five largest values within each problem size, all had high (negative) loadings on the eigenvector. Figure 6 shows the selected set of problems. Because Study 2 indicated that little systematic variability existed in the efficiency of solutions once problem size was accounted for, the consistency of these problems may partly stem from random sampling errors rather than systematic aspects of the problems. Consequently, we conducted Experiment 3 using this subset, to establish the reliability of the 15-problem subset.

METHOD

The goal of Experiment 3 was to measure test-retest reliability of the 15 problems selected for the abbreviated set. Participants performed the TSP set twice.

Participants. A total of 32 undergraduate students at Michigan Technological University took part in the study in exchange for partial course credit. One participant only completed the first half of the study, and so was removed. A second participant produced solutions that tended to be unreasonably long, and was removed from all additional analysis.

Stimuli. In each testing block, fifteen problems were presented in a random order, following five small (6-node) practice problems. During the second administration, all problems were rotated 90 degrees and mirrored so that the problems would have identical-length optimal solutions, with somewhat different surface representations. Unlike the experiments reported previously here, all participants used the same starting node for each problem.

Procedure. After signing an informed consent document approved by the Michigan Technological University Institutional Review Board, participants completed four distinct tests. They first completed the basic 15-problem TSP set. Following this, they completed a filler task taking approximately ten minutes, and then completed the second TSP set with rotated and mirrored problems. Finally, a fourth unrelated test was completed. Following the five small practice problems, problem order was randomized within each set. The software and testing equipment was identical to that used in Experiment 2.

RESULTS

Results were similar to those found in Experiment 2. Inefficiency increased as problem size increased, with minor differences between problems of size 20 and 30. Performance did not change significantly between the two testing blocks (only for problem 30-01 was a paired-samples *t*-test significant at

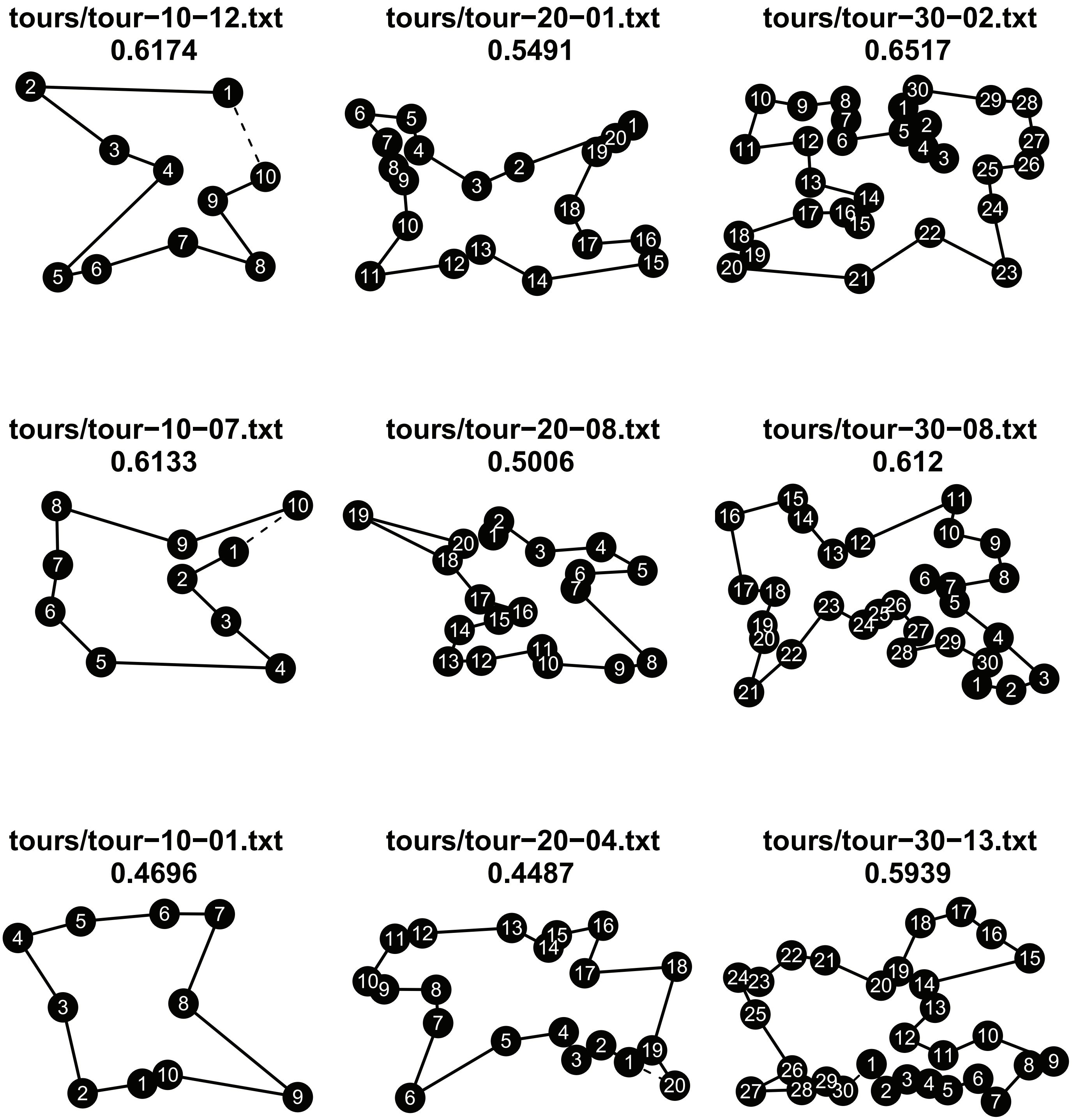


Figure 6.
The five best problems of each size (10, 20, and 30 points), as measured by the part-whole correlation of inefficiencies. Line shows the shortest solution path, and node marked "1" is by default the given starting location, which was fixed within problems for all participants in Experiment 3.

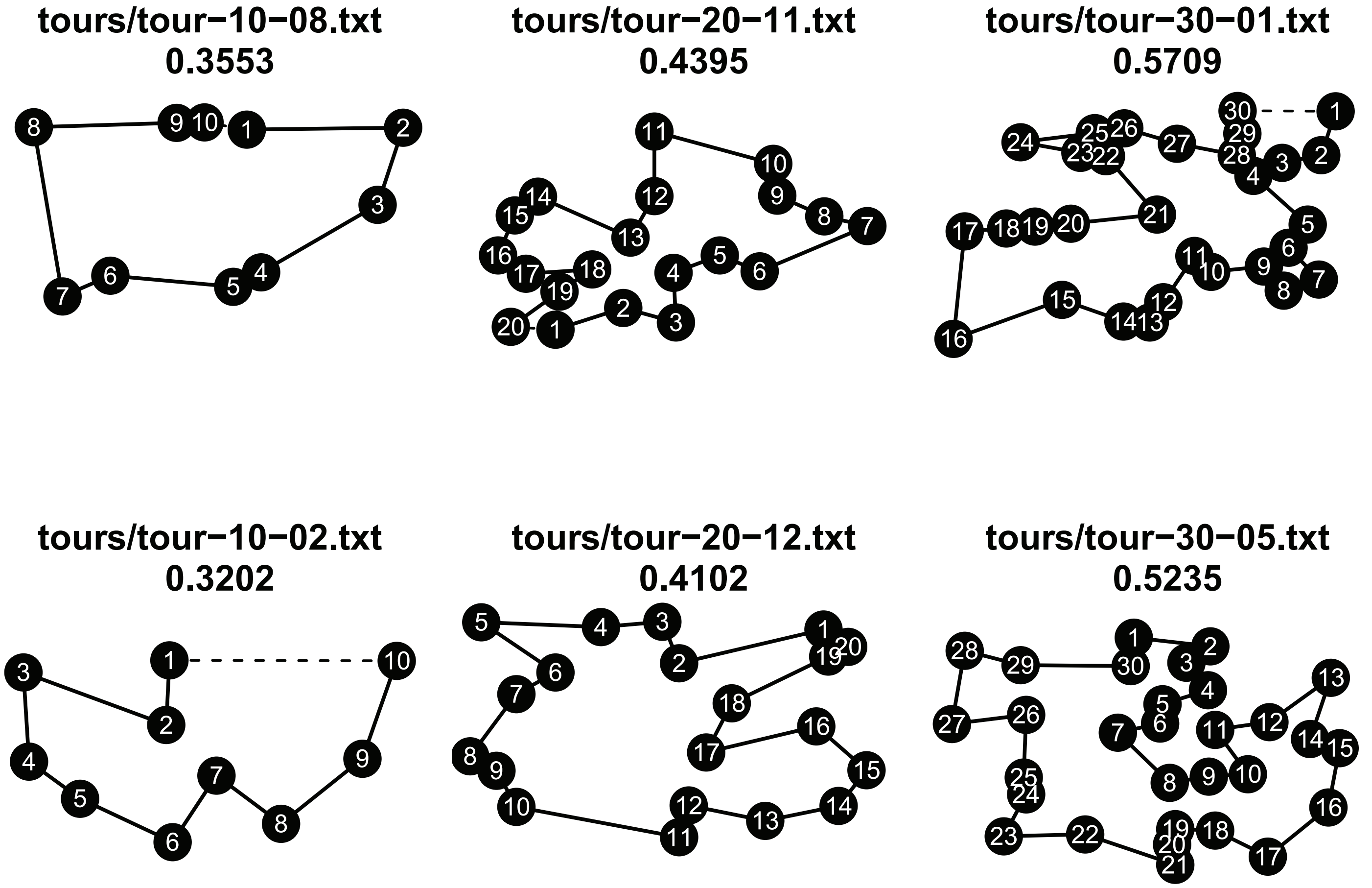


Figure 6., cont'd.
The five best problems of each size (10, 20, and 30 points), as measured by the part-whole correlation of inefficiencies. Line shows the shortest solution path, and node marked "1" is by default the given starting location, which was fixed within problems for all participants in Experiment 3.

$p < .05$, uncorrected for multiple comparisons; here mean inefficiency went from 1.1 during the pre-test to 1.14 during the post-test). Part-whole correlations for individual problems were similar in scale to the part-whole correlations of these problems in Experiment 2, although the values tended to be smaller, as might be expected based on regression to the mean of problems selected to be high on any particular value. Test-retest reliability, as assessed with both Spearman's correlation coefficient and intraclass correlation coefficients, ranged from small negative values to around 0.6 for individual problems. This indicates low and variable reliability on a per-item basis.

Although the optimal solution was shown after each trial, this apparently did not impact participants' performance when that problem was repeated, because mean inefficiency did not change significantly. It is likely that participants did not recognize the correspondence, because the problems were rotated and mirrored. Nevertheless, it is possible that participants remembered their (sub-optimal) solutions to the first administration, and used this memory during the second administration, which could inflate the test-retest reliability because of a memory effect rather than because of some underlying stable skill. Yet if a person were able to solve a problem perfectly on both administrations, they would also produce the same solution. This might not stem from memory for the solution, but rather because they were good both times.

To examine these issues, we computed the number of problems at each problem size that (1) were solved perfectly, (2) produced identical solutions for the pre-test and post-test, and (3) had non-perfect solutions that were identical. Mean values (out of five problems) are shown in Table 4, where we examined the practice problems alongside the test problems. Results indicate that identical solutions that were not perfect were rare (occurring on average less than once per participant); and that perfect solutions only happened substantially often on the small problem sizes (6 and 10 nodes). This supports the conclusion that specific memory effects did not impact the assessed reliability of the PEBL TSP task.

The inefficiency scores tended to be unreliable on an individual problem basis, but when averaged across multiple problems they were much better. Figure 7 shows scatterplots

Table 3.
Summary statistics related to inefficiency for each problem in Experiment 3, as well as averages within problem size and across the entire problem set.

Problem	Pre-test Inefficiency	Post-test Inefficiency	Difference	Part-whole Pre-test	Part-whole Post-test	Part-whole Mean	Test-retest Correlation	ICC
10-01	1.018	1.032	t(31)= 0.84, p=0.41	0.24	0.18	0.201	-0.016	-0.010
10-02	1.028	1.028	t(31)= 0.09, p=0.93	0.376	-0.138	0.259	0.029	0.044
10-07	1.035	1.053	t(31)= 1.11, p=0.28	0.234	0.209	0.186	0.034	0.024
10-08	1.014	1.015	t(31)= 0.07, p=0.95	-0.254	0.366	0.084	-0.210	-0.192
10-12	1.045	1.029	t(31)= -1.02, p=0.32	0.371	0.506	0.48	-0.250	-0.182
Average 10	**1.030**	**1.030**	**t(31)= 0.48, p=0.63**				**-0.130**	**-0.116**
20-01	1.117	1.113	t(31)= -0.25, p=0.8	0.718	0.502	0.796	0.216	0.229
20-04	1.086	1.104	t(31)= 0.93, p=0.36	0.452	0.604	0.606	0.423	0.417
20-08	1.091	1.067	t(31)= -1.61, p=0.12	0.503	0.698	0.732	0.367	0.318
20-11	1.087	1.093	t(31)= 0.36, p=0.72	0.253	0.226	0.354	-0.044	-0.030
20-12	1.113	1.086	t(31)= -1.57, p=0.13	0.081	0.018	0.191	0.020	-0.003
Average 20	**1.093**	**1.099**	**t(31)= -0.85, p=0.4**				**0.627**	**0.611**
30-01	1.102	1.140	t(31)= 2.11, p=0.04	0.573	0.629	0.685	0.368	0.299
30-02	1.083	1.099	t(31)= 1.45, p=0.16	0.469	0.751	0.68	0.372	0.354
30-05	1.121	1.107	t(31)= -1.02, p=0.31	0.673	0.551	0.686	0.606	0.584
30-08	1.102	1.105	t(31)= 0.21, p=0.84	0.559	0.542	0.715	0.369	0.378
30-13	1.065	1.072	t(31)= 0.45, p=0.66	0.392	0.317	0.48	-0.139	-0.119
Average 30	**1.104**	**1.094**	**t(31)= 1.45, p=0.16**				**0.661**	**0.651**
Grand Mean	**1.076**	**1.074**	**t(31)= 0.6, p=0.55**				**0.751**	**0.754**

of these composite scores, separately for each problem size. The low reliability for the 10-node problems is primarily a consequence of a ceiling or floor effect, as scores are compressed toward the perfect score of 1.0. Finally, Figure 8 shows the mean inefficiency scores obtained for each problem during the pre-test and the post-test. This shows that with our sample size of 32 participants, reliable estimates of problem difficulty can be obtained (R = .932 between mean pre-test and mean post-test score), with no notable decrease on the second administration. This measure is useful in that it suggests that these are reliable estimates of individual problems, making them useful for assessing performance of cognitive models.

GENERAL DISCUSSION

Across three experiments, we assessed performance on the PEBL TSP task, focusing primarily on the inefficiency scores of human-generated solutions. Experiments 1a and 1b established reasonably high test-retest reliability of composite scores, demonstrated convergent validity by showing inefficiency on the computerized TSP is related to inefficiency on a physical analog, and found that moderate physiological stress did not impair solution inefficiency. Experiment 2 examined a larger 45-problem set to establish basic psychometric properties, and Experiment 3 examined test-retest reliability on a smaller subset that can be administered in approximately 5 minutes.

COMPARISON TO RESULTS FROM PREVIOUS METHODS

The PEBL TSP test provides a computerized method for testing TSP performance using either the problems we describe here or custom problems developed by an experimenter. The version differs in a few ways from some previous versions used by other researchers, including: not permitting backtracking or restarting, requiring sequential solutions, and using relatively large target circles for easier testing. As described in the introduction, these differences are motived by the desire to create a version that can be easily learned, that is similar to first-person sequential navigation (which was demonstrated empirically in Experiment 1a), and that produces easy-to-manage data.

As a result, solution times tend to be shorter than for other implementations reported in the literature. Our regression in Experiment 2 indicated a time cost of about 640 ms per node across problem sizes. In contrast, even for well-practiced participants, Graham et al. (2000) reported solution times of 2 to 5 s per node, and Pizlo et al. (2006) estimated solution

Table 4.
Mean number of solutions produced that were perfect (inefficiencies smaller than 1.001), identical pre vs. post, and identical but not perfect pre vs. post. Mean values were the number of solutions out of 5 problems. Results show that participants rarely produced the same solution twice, except when they found a perfect solution. Standard deviation shown following ± symbol.

Problem size	Pre-test Perfect	Post-test Perfect	Pre-post Identical	Non-perfect Identical
6	3.58 ±1.03	3.64 ± .99	3.36 ± .96	0.27 ± .18
10	2.00 ± .94	1.69 ± .78	1.53 ± .84	0.48 ± .45
20	0.18 ± .39	0.094 ± .30	0.125 ± .42	0.125 ± .42
30	0 ± 0	0.06 ± .25	0.031 ± .18	0.031 ± .18

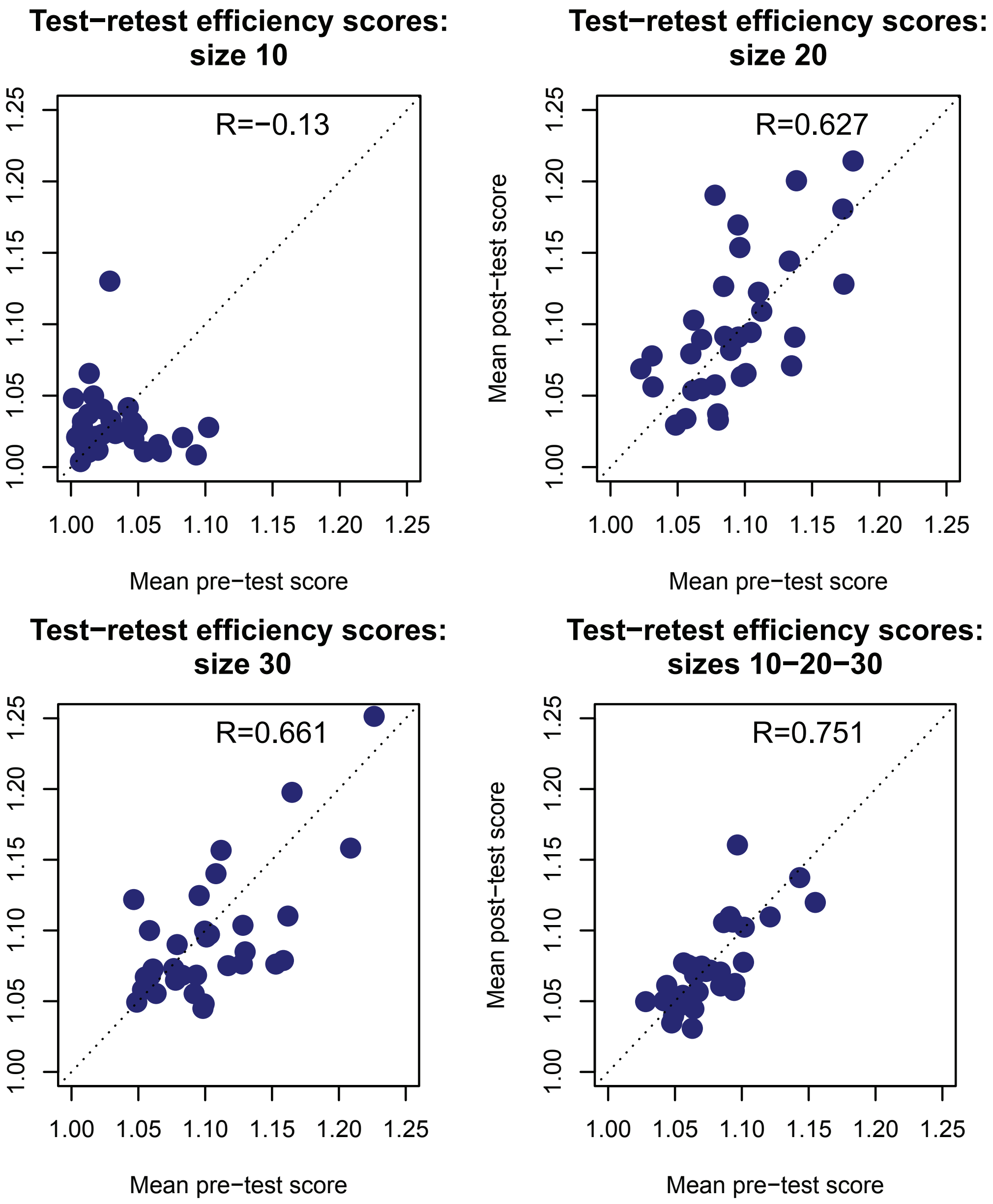


Figure 7.
Scatterplots showing mean scores for each person on problems of size 10, 20, 30, and the complete set for pre-test (horizontal axes) and post-test (vertical axes). Scores are uncorrelated for Size 10, which likely stems from the fact that scores were generally highly efficient and close to 1.0.

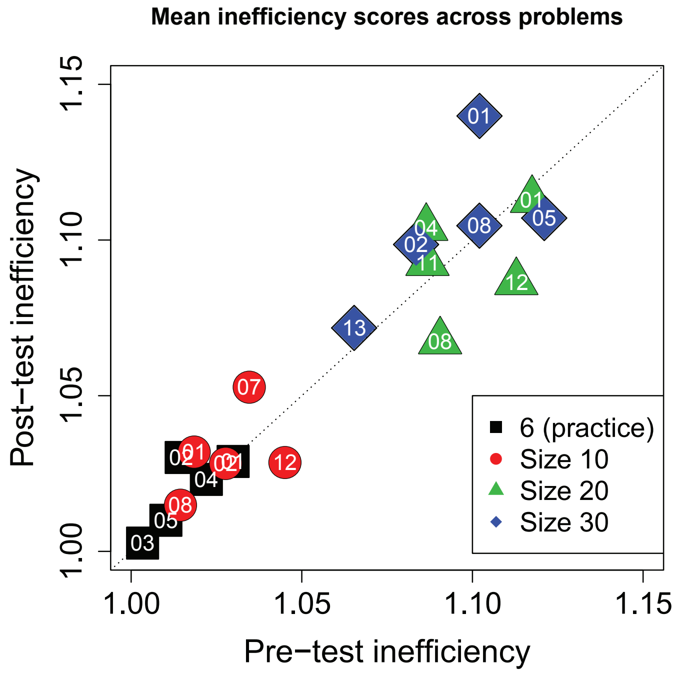


Figure 8.
Mean inefficiency scores of each problem for pre-test (horizontal axis) and post-test (vertical axis) administration. Mean efficiency of all problems was highly stable for multiple administrations, with a correlation of +.932.

times between 1 and 2 s per node. Similarly, Dry et al. (2006) estimated 1.7 s per node for unpracticed participants. Those methods allowed backtracking and editing the solution, and so some proportion of the longer solution times may have been accounted for by these additional processes.

Despite the fact that the PEBL version does not permit backtracking and must be completed sequentially, it produces path lengths (relative to optimal) similar to previous implementations. In the present study, problem sizes 10 through 30 produced solution paths that averaged 3% to 7% longer (for Experiment 2) and 3% to 11% longer (for Experiment 3) than optimal. Highly practiced participants tend to produce shorter paths: Graham et al. (2000) and Pizlo et al. (2006) reported solutions whose lengths ranged from 2% to 5% longer than optimal for problems of sizes 20 through 50, and Acuña and Parada (2010) reported that after extensive practice with the same problem, solutions tended to be within 1% of optimal for problems up to about size 20. But for naive solvers, Burns et al. (2006) reported solution lengths that increased from about 2% longer than optimal to 10% longer than optimal as problems increased from 10 to 50 nodes, by which point it had asymptoted. These results are also similar to the paper-and-pencil tests of MacGregor & Ormerod (1996), who found that solution lengths were 3.1% to 6.3% longer than optimal for 10- and 20-node problems, and MacGregor et al. (1999), whose solutions increased from around 2% to 10% longer than optimal as problem size increased from 10 to 50 nodes. Thus, although practice tends to produce substantially more efficient solutions, the other aspects of the PEBL task that make it faster than many other implementations nevertheless produce inefficiency scores in the same range.

RELIABILITY AND VALIDITY

The overall reliability of the PEBL TSP was established in several ways. Experiments 1a, 1b, and 3 each measured test-retest correlations of subsets of the 45-problem set. This is supported by a high Cronbach α (.87), and the selection of problems showing a high part-whole correlation in Experiment 2. Few past studies on TSP problems have assessed test-retest reliability as a means of assessing whether individuals differ systematically in how they solve the problems. One such study (Vickers et al., 2004, Experiment 2) established a test-retest correlation of +.68 for their 50-node TSP problem. This value is higher than any single problem in our data set, but their problem was larger and their participant population had a larger age range (17 to 50, *M* = 26, *SD* = 9.2) which might produce larger systematic variance across individuals. In comparison, in our Experiment 3, the composite scores for the abbreviated 15-problem set produced a test-retest reliability of .75, which was similar to that produced in the 21 problems in Experiment 1a (.77) but lower than that produced using 36 problems in Experiment 1b (.87). These differences in test-retest correlations are consistent with what would be predicted by the Spearman-Brown prophecy formula (Spearman, 1910; Brown, 1910), which suggests that with 36 problems, Experiment 1a and 3 would have produced test-retest correlations of .85 and .88, respectively. With 45 problems (as tested in Experiment 2), expected test-retest correlations would be .88, .89, and .90 for Experiments 1a, 1b, and 3. This indicates that our choice of problems for Experiment 3 did not have a large impact on the obtained reliability, and this in part came because we used 10-node problems that were less reliable indicators of individual differences. Although composite scores can be constructed that are reliable, individual problems tend to be less so. Researchers should be able to produce highly reliable measures with the complete 45-problem set used in Experiment 2, but with limited time may consider using the smaller set in Experiment 3, or restricting testing to just 20 and 30-city problems.

In terms of validity, Experiment 1a demonstrated that performance in the computerized test is correlated with performance in a physical version, which presumably is closely related to daily navigation tasks. Experiment 1a and 1b showed that performance in the task is not impacted by mild-to-moderate physiological stress that impacted other tasks involving sustained attention. This result is somewhat consistent with the finding of Dry et al. (2012), who showed TSP performance is only modestly (and not significantly) impacted by levels of alcohol consumption that impair a number of other cognitive tasks. This suggests that solutions to the TSP involve intuitive and highly visual processes that may in some sense be automatized. Previously, Vickers et al. (2004) established that TSP correlates well with performance on other similar problems and measures of fluid intelligence, but future studies will need to be conducted that determine the extent to which the PEBL TSP shows similar properties, and also the extent to which solutions are predicted by factors such as focal brain damage, psychiatric disorders, aging, or if it is correlated with other measures of intelligence, reasoning, problem solving, and executive function.

MODIFICATIONS AND USES

Because the PEBL TSP task is open source software, it can be freely used, modified, and exchanged. Potential modifications include methods used in other tests, such as incorporating backtrack-ing or restarting capabilities, permitting non-sequential solution methods, allowing the participant to select the starting node, and using an open-ended version of the test in which the starting location does not need to be revisited. Furthermore, test instructions can be translated by editing the .pbl file in a text editor.

In practice, the abbreviated 15-problem test with five practice problems should typically take about five minutes to complete. This makes the test feasible to incorporate within a larger testing battery, which could be used to help establish construct and criterion validity of the test.

SUMMARY AND CONCLUSIONS

The TSP has previously been used to understand aspects of visual problem solving, and research has suggested that it indexes individual differences in problem solving that are related to fluid intelligence. Despite this, the TSP has seen little use in general cognitive, clinical, or neuropsychological testing and assessment. In this paper, we described an implementation distributed via the open source PEBL test battery, including data from 45 problems and an abbreviated (15-problem) testing set that can be administered in under ten minutes. Future research that establishes the extent to which performance on the test is related to other cognitive functions, impairments, or neural pathways may help make the TSP test more informative and useful.

NOTES

Brandon S. Perelman is now at the U.S. Army Research Laboratory. Experiments 1a and 1b were conducted while Shane T. Mueller was a senior scientist at Applied Research Associates, supported by the Defense Threat Reduction Agency/ JSTO Project Number CB-08-PRO-05.

REFERENCES

Acuña, D. E. & Parada, V. (2010). People efficiently explore the solution space of the computationally intractable traveling salesman problem to find near optimal tours. *PLoS ONE, 5*(7), e11685. http://dx.doi.org/journal.pone.0011685

Applegate, D., Bixby, R., Chvátal, V., & Cook, W. (2003). *Concorde: A code for solving traveling salesman problems.* Retrieved from http://www.math.uwaterloo.ca/tsp/concorde.html

Applegate, D., Bixby, R., Chvátal, V., & Cook, W. (2006). *The traveling salesman problem.* Princeton: Princeton University Press. http://press.princeton.edu/chapters/s8451.pdf

Araujo, C., Kowler, E., & Pavel, M. (2001). Eye movement during visual search: The costs of choosing the optimal path. *Vision research, 41*(25), 3613–3625. http://dx.doi.org/10.1016/S0042-6989(01)00196-1

Basso D., Bisiacchi, P. S., Cotelli, M., & Farinello, C. (2001). Planning times during traveling salesman's problem: Differences between closed head injury and normal subjects. *Brain and Cognition 46*, 38–42. http://dx.doi.org/10.1016/S0278-2626(01)80029-4

Brown, W. (1910). Some experimental results in the correlation of mental abilities. *British Journal of Psychology, 3*,296–322. http://dx.doi.org/10.1111/j.2044-8295.1910.tb00207.x

Burns, N. R., Lee, M. D., & Vickers, D. (2006). Are individual differences in performance on perceptual and cognitive optimization problems determined by general intellgence? *Journal of Problem Solving, 1*(1), 3. http://dx.doi.org/10.7771/1932-6246.1003

Chronicle, E. P., MacGregor, J. N., Lee, M., Ormerod, T. C., & Hughes, P. (2008). Individual differences in optimization problem solving: Reconciling conflicting results. *Journal of Problem Solving, 2*, 41–49. http://dx.doi.org/10.7771/1932-6246.1030

Corsi, P. M. (1972). Human memory and the medial temporal region of the brain. *Dissertation Abstracts International, 34*, 819B.

Dantzig, G. B., Fulkerson, D. R., & Johnson, S. M. (1959). On a linear-programming, combinatorial approach to the travelling-salesman problem. *Operations Research, 7*, 58–66 http://dx.doi.org/10.1287/opre.7.1.58

Dry, M. J., Burns, N. R., Nettelbeck, T., Farquharson, A. L., & White, J. M. (2012). Dose-related effects of alcohol on cognitive functioning. *PloS ONE, 7*(11), e50977. http://dx.doi.org/10.1371/journal.pone.0050977

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 ProblemSolving,1*(1),4.http://dx.doi.org/10.7771/1932-6246.1004

Fox, J., & Weisberg, H. S. (2010). An R companion to applied regression. Thousand Oaks, CA: Sage Publications.

Graham, S. M., Joshi, A., & Pizlo Z. (2000). The traveling salesman problem: A hierarchical model. *Memory & Cognition, 28*, 1191–1204. http://dx.doi.org/10.3758/BF03211820

Halverson, T., & Hornof, A. J. (2011). A computational model of "active vision" for visual search in human computer interaction. *Human-Computer Interaction, 26*(4), 285–314. http://dx.doi.org/10.1080/07370024.2011.625237

Kessels R. P. C., van Zandvoort M. J. E., Postma A., Kappelle L. J., de Haan E. H. F. (2000). The Corsi block-tapping task: Standardization and normative data. *Applied Neuropsychology, 7*, 252–258. http://dx.doi.org/10.1207/S15324826AN0704_8

Kong, X., & Schunn, C. D. (2007). Global vs. local information processing in visual/spatial problem solving: The case of traveling salesman problem. *Cognitive Systems Research, 8*(3), 192–207. http://dx.doi.org/10.1016/j.cogsys.2007.06.002

Kotovsky K., Hayes J. R., & Simon H. A. (1985). Why are some problems hard? Evidence from Tower of Hanoi. *Cognitive Psychology, 17*, 248–294. http://dx.doi.org/10.1016/0010-0285(85)90009-X

MacGregor, J. N. (2012). Indentations and starting points in traveling sales tour problems: Implications for theory. *Journal of Problem Solving, 5*(1), 2. http://dx.doi.org/10.7771/1932-6246.1140

MacGregor, J. N. (2014). An investigation of starting point preferences in human performance on traveling salesman problems, *Journal of Problem Solving, 7*(1), Article 10. http://dx.doi.org/10.7771/1932-6246.1159

MacGregor, J. N., & Chu, Y. (2011). Human performance on the traveling salesman and related problems: A review. *Journal of Problem Solving, 3*(2), 2. http://dx.doi.org/10.7771/1932-6246.1090

MacGregor, J. N., & Ormerod, T. C. (1996). Human performance on the traveling salesman problem. *Perception & Psychophysics, 58*, 527–539. http://dx.doi.org/10.3758/BF03213088

MacGregor, J. N., Ormerod, T. C., & Chronicle, E. P. (1999). Spatial and contextual factors in human performance on the travelling salesperson problem. *Perception, 28*(11), 1417–1427. http://dx.doi.org/10.1068/p2863

Miyata, H., Watanabe, S., & Minagawa, Y. (2014). Performance of young children on "traveling salesperson" navigation tasks presented on a touch screen. *PLoS ONE, 9*(12), e115292. http:// dx.doi.org/10.1371/journal.pone.0115292

Mueller, S. T. (2008). Is the Turing test still relevant? A plan for developing the Cognitive Decathlon to test intelligent embodied behavior. In *Proceedings of the Nineteenth Midwest Artificial Intelligence and Cognitive Science Conference* (MAICS, April 2008).

Mueller, S. T. (2010). A partial implementation of the BICA Cognitive Decathlon using the Psychology Experiment Building Language (PEBL). *International Journal of Machine Consciousness, 2*, 273–288. http://dx.doi.org/10.1142/S1793843010000497

Mueller, S. T. (2014). *PEBL: The Psychology Experiment Building Language* (Version 0.14) [Computer experiment programming language]. Retrieved from http:://pebl.sourceforge.net

Mueller, S. T., Jones, M., Minnery, B. S., & Hiland, J. M. H. (2007). The BICA Cognitive Decathlon: A test suite for biologically-inspired cognitive agents. In *Proceedings of the Behavior Representation in Modeling and Simulation* (BRiMS, March 2007).

Mueller, S. T., Perelman, B. S., & Simpkins, B. G. (2013). Pathfinding in the cognitive map: Network models of mechanisms for search and planning. *Biologically Inspired Cognitive Architectures, 5*, 94–111. http://dx.doi.org/10.1016/j.bica.2013.05.002

Mueller, S. T., & Piper, B. J. (2014). The Psychology Experiment Building Language (PEBL) and PEBL test battery. *Journal of Neuroscience Methods, 222*, 250–259. http://dx.doi.org/10.1016/j.jneumeth.2013.10.024

Mueller, S. T. , Price, O. T, McClellan, G. E., Fallon, C. K., Simpkins, B., & Cox, D. (2010). *Cognitive Performance Prediction with the T3 Methodology* (DTRA Interim Technical Report, HDTRA-1-08-C-0025). Fort Belvoir, VA: Defense Threat Reduction Agency.

Papadimitriou, C. H. (1977). The Euclidean travelling salesman problem is NP-complete. *Theoretical Computer Science, 4*(3), 237–244. http://dx.doi.org/10.1016/0304-3975(77)90012-3

Perelman, B. S. (2015). *A naturalistic computational model of human behavior in navigation and search tasks* (Doctoral dissertation). Digital Commons. Michigan Technological University, Houghton, MI.

Perelman, B. S. & Mueller, S. T. (2013a). Examining memory for search using a simulated aerial search and rescue task. In *Proceedings of the 17th International Symposium on Aviation Psychology* (ISAP 2013).

Perelman, B. S. & Mueller, S. T. (2013b). Modeling human performance in simulated unmanned aerial search. In *Proceedings of the 12th International Conference on Cognitive Modeling* (ICCM 2003).

Perelman, B. S. & Mueller, S. T. (2015). Identifying mental models of search in a simulated flight task using a pat mapping approach. In *Proceedings of the 18th International Symposium on Aviation Psychology* (ISAP 2015).

Piper, B., Victoria, L., Massar, E., Kobel, Y., Benice, T., Chu, A., . . . Mueller, S. T., & Raber, J. (2012). Executive function on the psychology experiment building language tests. *Behavior Research Methods, 43*, 1–14. http://dx.doi.org/10.3758/s13428-011-0096-6

Pizlo, Z., Stefanov, E., Saalweachter, J., Li, Z., Haxhimusa, Y., & Kropatsch, W. (2006). Traveling salesman problem: A foveating pyramid model. *Journal of Problem Solving, 1*(1), 83–101. http://dx.doi.org/10.7771/1932-6246.1009

R Core Team (2013). *R: A language and environment for statistical computing.* Vienna, Austria: R Foundation for

Statistical Computing. http://www.R-protect.org/

Reitan, R. M. (1955). The relation of the Trail Making Test to organic brain damage. *Journal of Consulting Psychology, 19*, 393–394. http://dx.doi.org/10.1037/h0044509

Reitan, R. M. (1958). Validity of the Trail Making Test as an indicator of organic brain damage. *Perceptual and Motor Skills, 8*, 271–276. http://dx.doi.org/10.2466/pms.1958.8.3.271

Shallice, T. (1982). Specific impairments of planning. *Philosophical Transactions of the Royal Society B, 298*, 199–209. http://dx.doi.org/10.1098/rstb.1982.0082

Spearman, C. (1910). Correlation calculated from faulty data. *British Journal of Psychology, 3*, 271–295. http://dx.doi.org/10.1111/j.2044-8295.1910.tb00206.x

van Rooij, I., Schactman, A., Kadlec, H., & Stege, U. (2006). Perceptual or analytical processing? Evidence from children's and adult's performance on the Euclidean traveling salesperson problem. *Journal of Problem Solving, 1*, 44–73. http://dx.doi.org/10.7771/1932-6246.1006

Vickers, D., Butavicius, M., Lee, M. D., & Medvedev, A. (2001). Human performance on visually presented traveling salesman problems. *Psychological Research, 65*, 34–45. http://dx.doi.org/10.1007/s004260000031

Vickers, D., & Lee, M. D. (1998). Never cross the path of a traveling salesman: The neural network generation of Halstead-Reitan trail making tests. *Behavior Research Methods, Instruments, & Computers, 30*, 423–431. http://dx.doi.org/10.3758/BF03200675

Vickers, D., Mayo, T., Heitmann, M., Lee, M. D., & Hughes, P. (2004). Intelligence and individual differences in performance on three types of visually presented optimization problems. *Personality and Individual Differences, 36*, 1059–1071. http://dx.doi.org/10.1016/S0191-8869(03)00200-9

Wolfe, J. M. (1994). Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin & Review, 1*(2), 202–238. http://dx.doi.org/10.3758/BF03200774

APPENDIX A

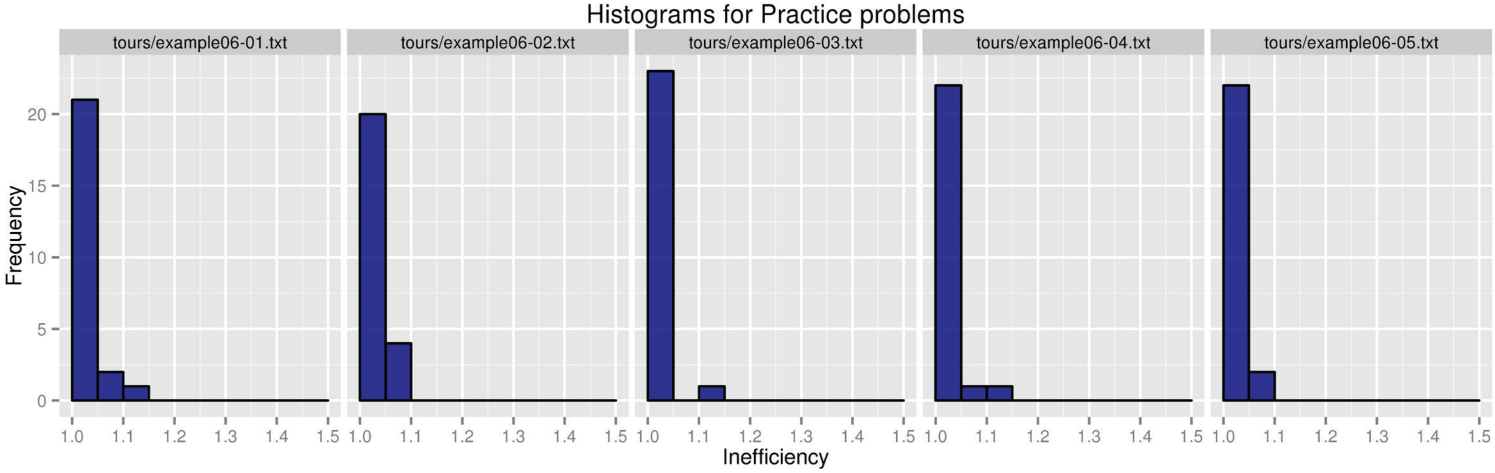
The following R function will read in a save file produced by the Concorde windows GUI, and save out a .txt file that can be used by the PEBL TSP solver. If a solved problem is saved with the name “testfile,” using `processTSPFile("testfile")` will read that file and save it as “testfile.txt.”

```
processTSPFile <- function(filebase, maxscale=500)
  {
    ##maxscale is the maximum size that the points should be scaled to,
    ##keeping the aspect ratio of x and y.
    ##Points are likely to be generated within a 100x100 grid.
    data <- scan(filebase)
    length <- data[1]
    xypoints <-cbind( data[(1+1:length*2)],
             data[(1+1:length*2)+1])
    ##scale the points, making it a bit smaller than 500 x 500 max
    xypoints <- xypoints*maxscale/max(xypoints)/1.1

    ##extract the order information:
    order <-matrix( data[((length+1)*2+1):length(data)],
            ncol=3,byrow=T)
    ord <- order[,1]+1
    newpoints <- data.frame(city=paste("City",order[ord,1]),
                xypoints[ord,],0)
    plot(xypoints[ord,],type="b")##display the path for verification
    ##save out into a new file.
    out <- file(paste(filebase,".txt",sep=""),"w")
    for(i in 1:nrow(newpoints))
      {
        cat( paste(newpoints[i,1],newpoints[i,2],
          newpoints[i,3],newpoints[i,4],
          "\n"),file=out)
      }
    close(out)
  }

> processTSPFile("testfile")
```

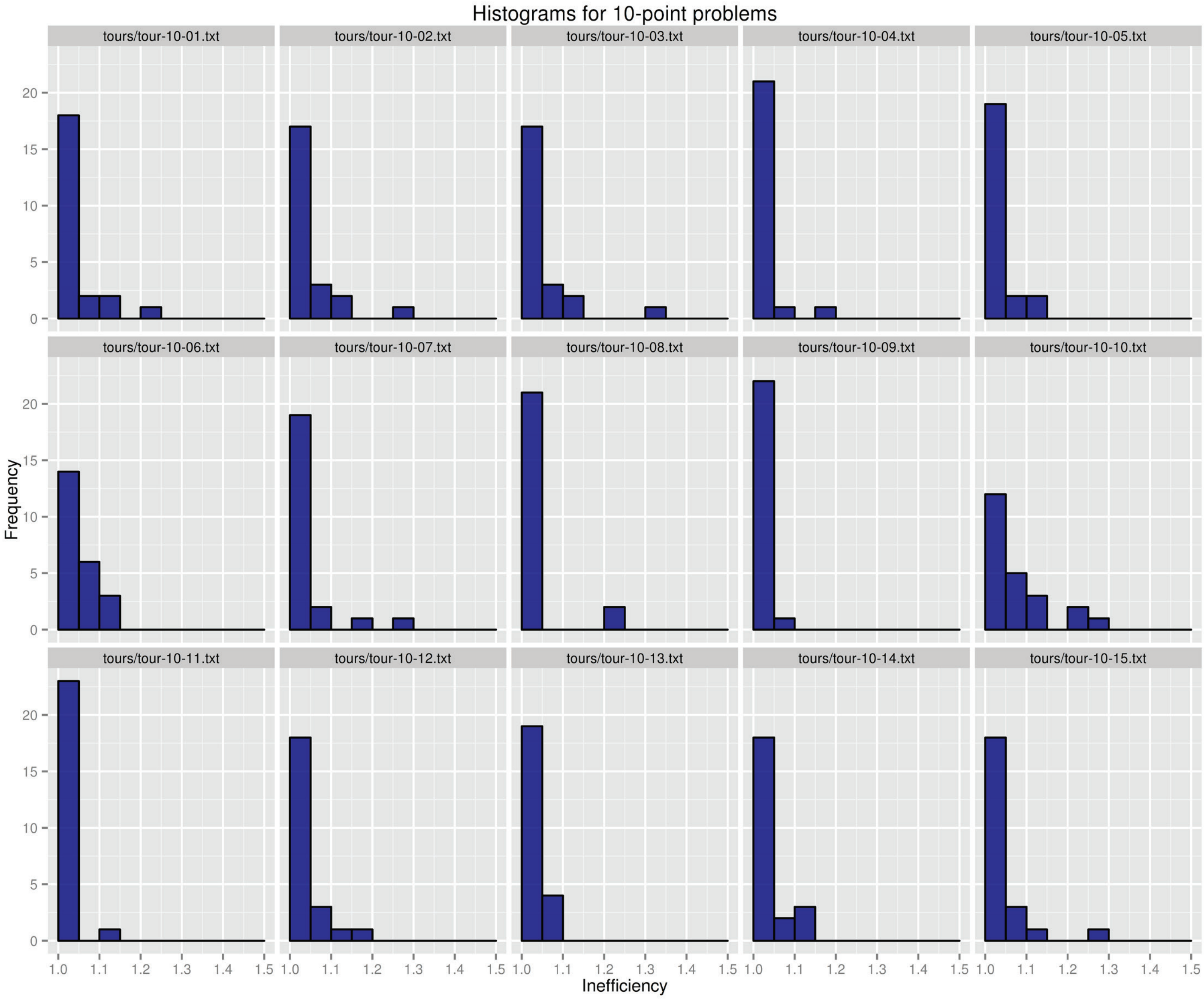
APPENDIX B

The following figures show histograms of inefficiency scores across the 50 problems tested in the present experiment.

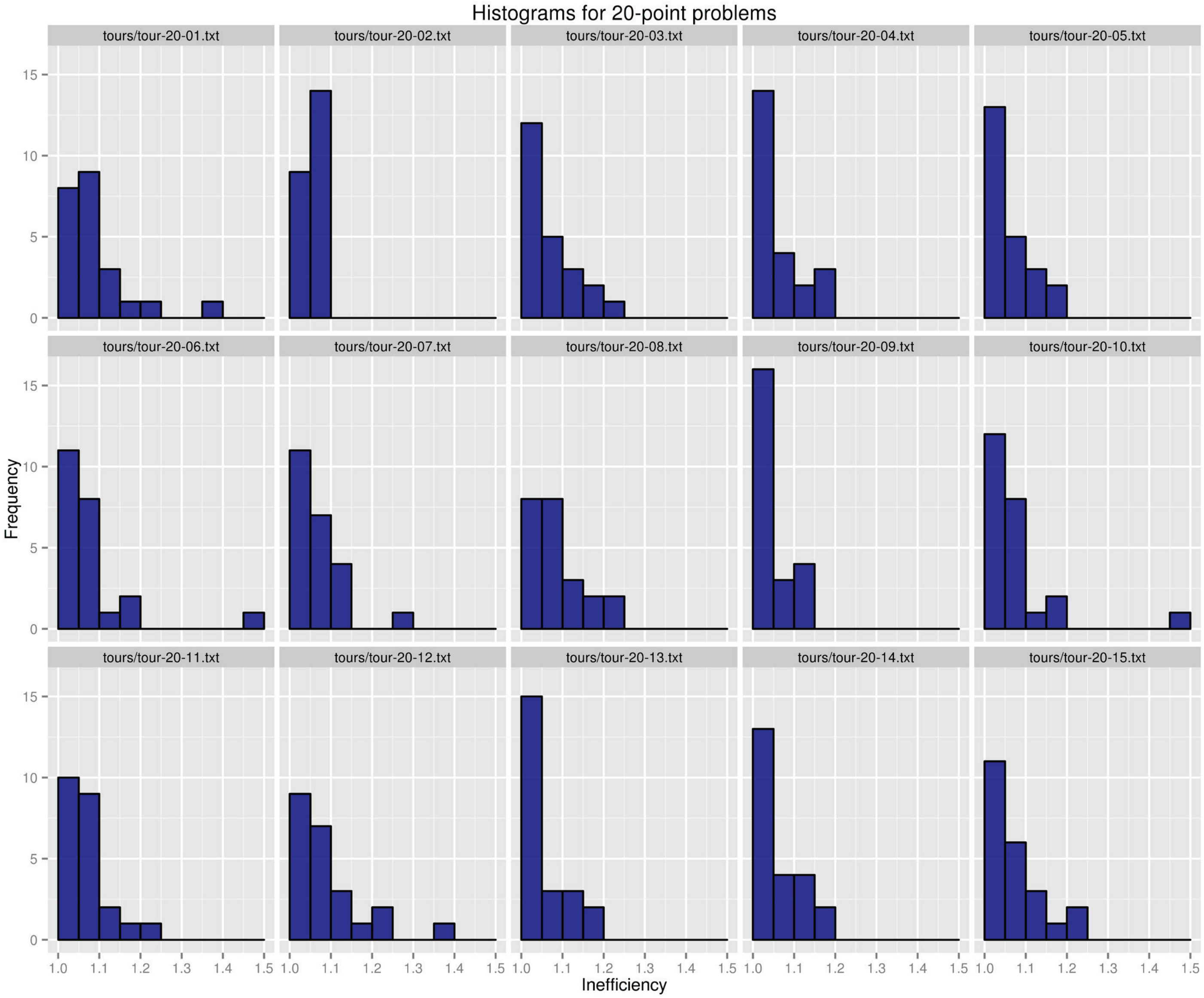
PRACTICE PROBLEMS (SIZE 6)



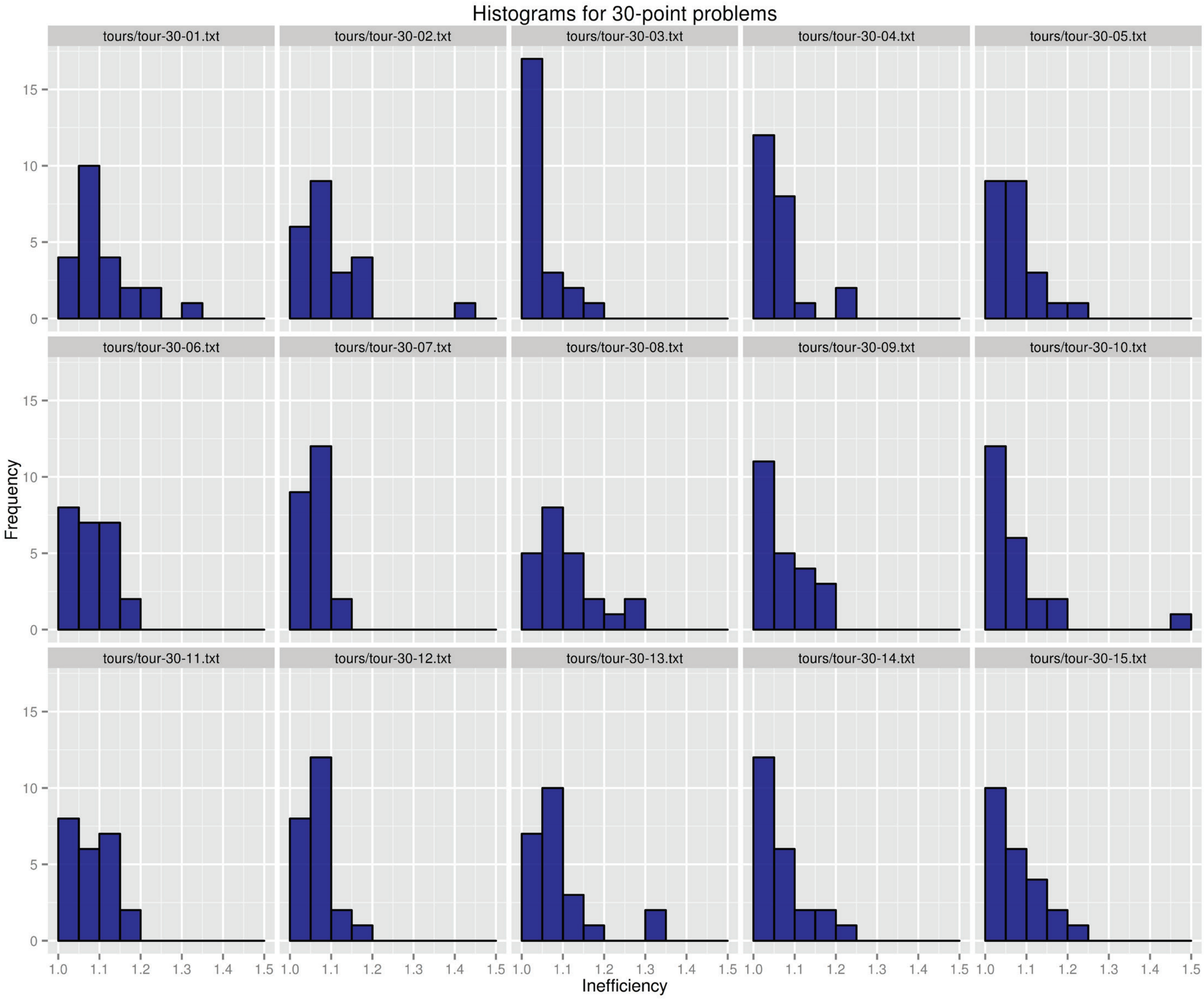
INEFFICIENCY HISTOGRAMS FOR 15 PROBLEMS OF SIZE 10.



INEFFICIENCY HISTOGRAMS FOR 15 PROBLEMS OF SIZE 20



INEFFICIENCY HISTOGRAMS FOR 15 PROBLEMS OF SIZE 30



APPENDIX C

STATISTICS FOR THE 51 TSP PROBLEMS TESTED IN EXPERIMENT 2

Problem	Points	Part-whole correlation	Planning time / efficiency correlation	Median inefficiency	95th percentile inefficiency	Median planning time	95th percentile planning time	Median solution time	95th percentile solution time
tours/example06-01.txt	6	0.0993	-0.2473	1	1.064	5312	24228	8868	26792
tours/example06-02.txt	6	-0.0979	0.2417	1	1.054	3602	10108	6128	12020
tours/example06-03.txt	6	-0.0518	0.0302	1	1	2547	9801	5256	12974
tours/example06-04.txt	6	0.4121	0.0969	1	1.081	4932	10235	7945	14498
tours/example06-05.txt	6	0.2454	0.2402	1	1.052	3317	10763	5540	14879
tours/tour-10-01.txt	10	0.4696	0.2607	1	1.138	2595	4628	7205	12934
tours/tour-10-02.txt	10	0.3202	-0.216	1.025	1.115	2458	7997	8169	21826
tours/tour-10-03.txt	10	0.1954	-0.2646	1.013	1.112	2272	12762	7150	20705
tours/tour-10-04.txt	10	0.3096	-0.1407	1.008	1.057	3283	9791	7946	15623
tours/tour-10-05.txt	10	0.0764	0.0753	1.017	1.1	2479	7436	8538	19138
tours/tour-10-06.txt	10	-0.0727	-0.2049	1	1.122	2291	11015	7326	16731
tours/tour-10-07.txt	10	0.6133	0.1542	1.009	1.17	2120	11804	7470	16642
tours/tour-10-08.txt	10	0.3553	0.1227	1	1.201	1850	6794	6636	11460
tours/tour-10-09.txt	10	0.2101	0.1703	1.018	1.035	2178	14411	8357	21323
tours/tour-10-10.txt	10	0.1051	0.1245	1	1.243	2405	7245	7778	14962
tours/tour-10-11.txt	10	0.0999	-0.2165	1	1.017	1642	6668	5920	12548
tours/tour-10-12.txt	10	0.6174	0.0196	1.019	1.123	2846	9266	8842	21036
tours/tour-10-13.txt	10	-0.1174	0.3167	1.005	1.074	2329	7405	6153	12504
tours/tour-10-14.txt	10	-0.0852	0.0804	1.048	1.11	3162	9063	7847	19874
tours/tour-10-15.txt	10	0.2747	0.0423	1	1.114	1785	4107	5969	9895
tours/tour-20-01.txt	20	0.5491	-0.0868	1.065	1.227	2551	10496	13120	33627
tours/tour-20-02.txt	20	0.2793	-0.1554	1.054	1.091	2279	25907	13112	43984
tours/tour-20-03.txt	20	0.3793	-0.0897	1.04	1.166	2655	20212	12267	35087
tours/tour-20-04.txt	20	0.4487	-0.2434	1.042	1.169	2192	11663	12936	33574
tours/tour-20-05.txt	20	-0.2975	-0.1889	1.043	1.148	2981	7431	14388	23585
tours/tour-20-06.txt	20	0.399	-0.1745	1.052	1.176	2827	20863	16636	33307
tours/tour-20-07.txt	20	0.3334	-0.5086	1.062	1.143	2993	4673	13700	22233
tours/tour-20-08.txt	20	0.5006	-0.157	1.066	1.198	1987	5450	11657	30277
tours/tour-20-09.txt	20	0.2105	-0.3595	1.035	1.116	2329	14184	13189	31559
tours/tour-20-10.txt	20	-0.1476	-0.1449	1.048	1.189	2254	6471	14122	26674
tours/tour-20-11.txt	20	0.4395	0.039	1.054	1.166	1698	7538	12533	29987
tours/tour-20-12.txt	20	0.4102	-0.0632	1.059	1.234	1810	7466	11904	28174
tours/tour-20-13.txt	20	0.0217	-0.0055	1.007	1.148	1970	4792	12901	25245
tours/tour-20-14.txt	20	0.2738	-0.2939	1	1.146	1783	7602	10428	22065
tours/tour-20-15.txt	20	0.1723	-0.1633	1.061	1.203	2517	6803	12673	23037
tours/tour-30-01.txt	30	0.5709	-0.2252	1.08	1.23	3753	10936	21814	43689
tours/tour-30-02.txt	30	0.6517	0.1211	1.08	1.19	2371	22225	18176	52532
tours/tour-30-03.txt	30	0.4398	-0.2764	1.03	1.12	1613	9343	17322	34494
tours/tour-30-04.txt	30	0.4428	-0.1906	1.04	1.22	2585	8221	19736	33391
tours/tour-30-05.txt	30	0.5235	-0.3443	1.06	1.16	2455	15429	20470	38297
tours/tour-30-06.txt	30	0.4103	-0.2166	1.08	1.15	2421	12521	18979	47784
tours/tour-30-07.txt	30	0.1588	-0.2492	1.06	1.12	1876	6344	18107	32402
tours/tour-30-08.txt	30	0.612	-0.0961	1.09	1.25	2413	6719	20526	40639
tours/tour-30-09.txt	30	0.3353	-0.1331	1.06	1.16	2900	9042	20250	55361
tours/tour-30-10.txt	30	0.4209	0.5647	1.04	1.18	2676	6018	19261	30222
tours/tour-30-11.txt	30	0.1855	-0.0095	1.06	1.17	2645	6503	17341	29580
tours/tour-30-12.txt	30	0.2173	0.2475	1.06	1.12	2732	12479	19029	34354
tours/tour-30-13.txt	30	0.5939	-0.0126	1.07	1.3	2134	11295	16972	30930
tours/tour-30-14.txt	30	0.4727	-0.2157	1.05	1.18	2461	8651	19241	31785
tours/tour-30-15.txt	30	0.3958	0.0625	1.07	1.16	2895	8548	19005	27542

APPENDIX D: OVERVIEW OF PEBL TSP TASK

The PEBL TSP task was first developed as a means to assess the impact of heat strain on one of a number of cognitive abilities (Mueller et al., 2010, Experiment 1a). The test was developed using PEBL, a cross-platform open source programming language designed for developing neuropsychological tests. The PEBL TSP task was first distributed with PEBL 0.14 (released June, 2014), and it can be found in the Documents\battery\tsp folder of the user's home directory after PEBL has been installed. The version described here[1] involves minor modifications from the one released with PEBL 0.14, but will appear in future releases of PEBL, and is available directly at https://github.com/stmueller/pebl-custom/tree/master/Mueller-TSP-JPS.

OPTIONS

A number of settings can be controlled via the PEBL launcher by clicking the "Edit" button next to the "Parameters" label. These settings are shown below, and include:

- `targsize` (default: 25). Target radius in pixels.
- `dointro` (default: 1). Whether to do the intro and practice tests. Set to 0 to skip practice tests.
- `xoffset` (default: 70). Upper left x offset of problem field.
- `yoffset` (default: 70). Upper left y offset of problem field.
- `width` (default: 500). Width of testing field in pixels, not including a target-size margin.
- `height` (default: 500). Height of testing field in pixels, not including a target-size margin.
- `usefile` (default:). Use the set of problems listed in the specified problem.
- `trialspersize` (default: 5). If usefile is set to 0, select the number of standard trials per length, up to 15.
- `chooserandomtrials` (default: 0=no). If usefile is set to 0, whether to choose random trials or the first N of each problem size.
- `inputlabel` (default: click). Whether the instructions should indicate "click" or "touch," if a touchscreen is being used.
- `doflash` (default: 0=off). Should the circle "flash" when you click it? This is useful for touchscreen administration, where the finger covers the circle and can obscure feedback.
- `dobeep` (default: 1=on). Should a click sound play when clicks made?
- `randomstart` (default: 0=no). Should each problem start at a randomly-selected starting node?

TEST PROBLEM SPECIFICATION

The PEBL TSP software can be used to test many TSP layouts, although it does not create them on its own. The problems are constrained to a 500 x 500 pixel field (although this is adjustable in the parameters). Each problem is read from a text file located in tsp\tours, and uses a five-column format (each column must be separated by a space or tab character) shown in Figure 2. These files can be created by hand, but the locations should be saved in an order consistent with the shortest path solution. The file should be a plain text file with no column headers. The first two columns are ignored, but will typically involve the label 'City' and an index indicating the original sampled order. The next two columns indicate the x and y coordinates of the city. The last column is also ignored, and will typically be 0. For historical reasons, the paths included in PEBL test battery also repeat the first point at the end, but these are disregarded by prepending the # symbol. A sample data file is shown in Figure D1. Because these problems require a solution, we provide an R function that allows the save file of the Concorde Windows GUI[2] to be transformed into this format. To use this function, a solved path in Concorde must be saved (not exported), and the resulting the file can be read and saved into the format used by the PEBL TSP via the processTSPFile() R function found in Appendix A.

OUTPUT

Within the TSP directory, data are saved in the data\subdirectory, with participant-specific data saved in separate directories named according to the participant code entered into the PEBL launcher. Data are saved into five distinct files on each run, so for a participant '99,' the files will include a click-by-click logfile ("battery\tsp\data\99\tsp-99.csv"), a problem-by-problem summary ("battery\tsp\data\99\tsp-summary-99.csv"), an overall report file ("battery\tsp\data\99\tsp-report.txt"), a simple log file recording start and

```
----------------------------
City  0    146   356   0
City  4    381   507   0
City  3    383   379   0
City  5    479   201   0
City  1    497   59    0
City  2    259   246   0
#City 0    146   356   0
----------------------------
```

Figure D1.
Sample data file used by the PEBL TSP software. Only columns 3 and 4 are used, and the final repeated row is ignored because of the # symbol. The order of the cities in the file will be used as the comparison path, even if it is not the shortest.

end time for each run (“battery\tsp\data\tsp-log.txt”), and a pooled data file that records one line per participant (“data \tsp-pooled.csv”).

A sample of the click-by-click logfile (tsp-99.csv) is shown below.

```
subnum,trial,test,type,point,targx,targy,clickX,clickY,pointid,fileorder,time,rt
99,1,tours/example06-01.txt,PRACTICE,0,479,201,0,0,1,4,3546,0
99,1,tours/example06-01.txt,PRACTICE,1,497,59,521.818,123.636,2,5,4913,1367
99,1,tours/example06-01.txt,PRACTICE,2,259,246,305.455,293.636,3,6,5555,2009
99,1,tours/example06-01.txt,PRACTICE,3,146,356,202.727,393.636,4,1,6605,3059
99,1,tours/example06-01.txt,PRACTICE,4,381,507,416.364,530.909,5,2,7373,3827
99,1,tours/example06-01.txt,PRACTICE,5,383,379,418.182,414.545,6,3,7866,4320
99,2,tours/example06-02.txt,PRACTICE,0,259,445,0,0,1,6,10301,0
99,2,tours/example06-02.txt,PRACTICE,1,329,348,369.091,386.364,6,5,11646,1345
99,2,tours/example06-02.txt,PRACTICE,2,341,47,380,112.727,5,4,12227,1926
99,2,tours/example06-02.txt,PRACTICE,3,458,164,486.364,219.091,4,3,13018,2717
```

Here, properties of each click are recorded. The first several columns record participant code, trial, problem file, and problem type, after which the next column indexes the serial order of the click on each trial (counting up from 0 indicating the first given point). The columns targx and targy record the center of each clicked target, and the columns clickX and clickY record the exact screen coordinates of the mouse click. These may not be at the exact center, and are offset slightly (70 pixels each by default) because the target field has a user-controlled gutter based on the xoffset and yoffset control parameters. The next column (pointid) indexes the selected item in order of the optimal route. Thus, for a six-city problem, an optimal solution will have values that are either 1-2-3-4-5-6 or 1-6-5-4-3-2. If the randomstart control parameter is non-zero, the pointid values will differ in their absolute values from how they appeared in the problem file (although they will be in the same relative order). Consequently, the fileorder column records the row in the file where the chosen city appears. The time column indicates the time elapsed in ms since the test began (not the particular problem), and the rt column indicates the time since the last click was made or, for the first response, since the problem began. Because the first point is given and automatically connected once the next-to-last point is selected, it is never actually clicked on. Consequently, it appears at the beginning of each trial, and we record 0,0 for the clicked x/y coordinates, and we record a response time of 0. No record is made for the implicit final-city return line that is plotted on the screen.

More useful is the trial-by-trial summary (e.g., data\99 \tsp-sum-99.csv):

```
subnum,trial,test,type,starttime,firstclick,endtime,elapsedtime,numpos,opt,obs,eff
99,1,tours/example06-01.txt,PRACTICE,3546,4926,7888,4342,6,1102.81,1102.81,1,
99,2,tours/example06-02.txt,PRACTICE,10301,11655,14411,4110,6,1010.33,1010.33,1,
99,3,tours/example06-03.txt,PRACTICE,15151,16686,19653,4502,6,1067.08,1067.08,1,
99,4,tours/example06-04.txt,PRACTICE,20464,21780,24025,3561,6,1139.48,1151.14,1.01023,
99,5,tours/example06-05.txt,PRACTICE,25389,27061,29672,4283,6,1383.97,1383.97,1,
99,6,tours/tour-20-04.txt,TEST,31484,33229,44400,12916,20,1892.66,1892.66,1,
99,7,tours/tour-10-08.txt,TEST,45380,46489,51583,6203,10,1326.61,1326.61,1,
99,8,tours/tour-10-01.txt,TEST,52300,53250,58272,5972,10,1498.68,1498.68,1,
```

Here, each row summarizes performance on a specific test. Important dependent measures include the time at which the first click was made (which may potentially indicate “planning” time), the completion time, the obtained path length, the optimal path length, and an inefficiency score (obtained path length/optimal path length).

In addition, a report file is saved (e.g., data\99\tsp -report-99.txt), summarizing performance over problem size (shown below). This same table is shown to the participant at the end of the experiment, and a sample is shown at the top of the following page.

```
-----------------------------------
PEBL TSP Task
http://pebl.sf.net
Using PEBL Version: PEBL Version 0.14
Wed Aug 20 01:20:19 2014
Participant code: 99
-----------------------------------
        Your Performance
Condition Trials Time (s) Effic. # Perfect
-----------------------------------------------
6    5    4.1596  1.00205 4
10    5    7.2976  1.00769 2
20    5    17.0638 1.0543  1
30    5    27.0496 1.0927  0
-----------------------------------------------
```

Finally, a pooled data file is saved (data\tsp-pooled.csv) that records summary data in a single row per participant. The file is saved without a header, and each row begins with the participant code, a time-stamp, and a time value indicating the number of milliseconds the entire test took. For example:

```
23,Wed Aug 20 01:20:19 2014,318987,
99,Wed Aug 20 16:52:40 2014,387523,
```

After these columns, the table in the report file is saved row-wise. This involves five columns each for each problem size used. The number of columns saved will depend on the particular test sizes used. For the standard 6-10-20-30 problem sizes the remaining columns will look like the following:

```
6,5,4.1596,1.00205,4,10,5,7.2976,1.00769,2,20,5,17.0638,1.0543,1,30,5,27.0496,1.0927,0
6,5,4.6242,1,5,10,5,14.0316,1.0109,2,20,5,18.9764,1.20484,2,30,5,26.7264,1.09872,0
```

Together, the different files enable either detailed analysis of individual solution, summaries over each participants, or easy summaries across a population.

NOTES

[1] The version described here can be downloaded at https://sourceforge.net/projects/pebl/files/special/tsp.zip. This archive can be unzipped into the pebl-exp.0.14 \battery folder to replace the existing tsp\folder.

[2] Available at http://www.math.uwaterloo.ca/tsp/concorde /downloads/downloads.htm