ELSEVIER

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers & Operations Research ■■■ (■■■) ■■■–■■■

computers & operations research

www.elsevier.com/locate/dsw

Aggregation for the probabilistic traveling salesman problem

Ann Melissa Campbell

Department of Management Sciences, Tippie College of Business, University of Iowa, Iowa City, Iowa 52252-1000, USA

Abstract

In the probabilistic traveling salesman problem (PTSP), customers require a visit with a given probability, and the best solution is the tour through all customers with the lowest expected final tour cost. The PTSP is an important problem, both operationally and strategically, but is quite difficult to solve with realistically sized problem instances. One alternative is to aggregate customers into regions and solve the PTSP on the reduced problem. This approach raises questions such as how to best divide customers into regions and what scale is necessary to represent the full objective. This paper addresses these questions and presents computational results from experiments with both uniformly distributed and clustered data sets. The focus is on large problem instances where customers have a low probability of requiring a visit and the CPU time available is quite limited. For this class of instances, aggregation can yield very tight estimates of the full objective very quickly, and solving an aggregated form of the problem first can often lead to full solutions with lower expected costs.

Keywords: ■; ■; ■

1. Introduction

For many companies, only a subset of customers require a pickup or delivery each day. Information may not be available far enough in advance to create optimal schedules each day for those customers that require a visit or the cost to acquire sufficient computational power to find such solutions may be prohibitive. For these reasons, it is not unusual to design a distance minimizing tour containing all customers, and each day follow the ordering of this a priori tour to visit only the realized customers. Such a practice has been documented, for example, for the delivery of Meals on Wheels in Atlanta by

E-mail address: ann-campbell@uiowa.edu.

0305-0548/$ - see front matter © 2005 Published by Elsevier Ltd.
doi:10.1016/j.cor.2005.02.024

Bartholdi et al. [1]. These a priori tours also create a regularity of service that can be beneficial for both the customers and the drivers. Customers will be served at roughly the same time each day they require service, and the drivers can become very familiar with their routes. Starting from such a tour can be useful, too, as a starting point for reoptimization if there is time available on the day of service. The expected value of these tours can be used strategically as an estimate of the resources required to serve a set of customers. For example, in the motivating problem for [2], a market survey identified the probability that customers in the service area will request a delivery, and the expected tour cost served as an estimate of the time and resources required to serve the customers. Thus, these a priori tours can be very useful both operationally and strategically, but finding optimal, or even good, tours can be quite challenging if the number of potential customers is large. Due to the probabilistic nature of the problem, evaluating a proposed schedule can be quite expensive, so large problems are hard to solve even a priori when time is usually less of an issue.

We became interested in probabilistic routing problems as a result of our study of home delivery problems (HDP) [3,4]. In this study, which was inspired by e-grocers, issues addressed include how to charge a customer for a delivery based on the estimated cost of serving the customer. Because costs are primarily determined by the routing of the delivery trucks, the marginal cost of serving a particular customer is highly influenced by whether or not its neighbors receive a delivery as well. If an order is received early in the ordering window, estimating the cost to serve a particular customer can reasonably be based on projections of later orders if we have some knowledge of the probability of other customers placing an order. In evaluating algorithms for solving this variation of the probabilistic traveling salesman problem (PTSP), we found that the few existing algorithms for the PTSP can be prohibitively expensive and time consuming for realistically sized problem instances. For on-line grocery services, for example, a particular depot may be responsible for serving many square miles representing at least hundreds, but probably thousands, of customers. Also, with new orders arriving and information about the customers continually being updated, these PTSPs need to be re-solved fairly often. Many other applications of the PTSP, beyond just grocery delivery, involve large numbers of customers, such as the routing for package delivery services.

One approach for all of these applications is to aggregate customers into delivery regions, but how to properly aggregate customers and integrate this into a solution methodology is not clear. Aggregation refers to grouping customers together and then representing them by one point spatially and, in this context, with a single probability value. Pre-existing aggregation is readily available in the form of city blocks, postal zip codes, tracts, census blocks, etc. and is used widely in marketing and facility location. There has been little analysis, theoretical or computational, though, on how aggregation can be used in solving routing problems or how aggregation can impact solution quality, especially in a probabilistic context. This paper takes a step in this direction.

One clear benefit of aggregation is the reduction in the a priori problem size. We are particularly interested in applications where there is a limited amount of time available to solve the PTSP, so it will be interesting to evaluate the tradeoff between considering the full problem vs. solving a smaller, aggregated form when time is constrained. This will be one focus of our study. In aggregating customers, there are many choices to make in terms of how to divide customers into regions and how many regions to create. A second focus of our study will be to evaluate the relationship between the level and type of aggregation with the PTSP objective value.

This paper is organized as follows. In the next section, we review the relevant literature in this area. Section 3 discusses how to reformulate the PTSP to account for aggregation, Section 4 defines the aggregation schemes to be evaluated, and Section 5 describes a related error result. Section 6 introduces the design of the computational experiments and presents results for uniformly distributed and clustered data sets. Finally, conclusions and insights are offered in Section 7.

2. Literature review

Patrick Jaillet's dissertation [5] introduces the PTSP and demonstrates some interesting properties of optimal tours including the fact that such a tour may intersect itself. He summarizes key results in [6], where he provides a formulation for the expected value of a tour and bounds the relationship between optimal PTSP and TSP solutions.

Berman and Simchi-Levi [7] focus on instances of the PTSP with heterogeneous probabilities, where most of Jaillet's results involve homogeneous probabilities. They establish a lower bound for such instances and explain how to combine this bound with a branch-and-bound algorithm to find an optimal a priori tour. They do not provide any computational results, but it is unlikely such an approach would work well with large problem instances.

Rossi and Gavioli [8] discuss how to modify construction heuristics for the TSP specifically to solve the PTSP. Their heuristics are based on Clarke and Wright and nearest neighbor techniques and do not include any local improvement. The expected costs of the resulting solutions are compared with those found using basic TSP heuristics. Based on their computational experiments, the authors conclude that it is important to use solution techniques specifically developed for the PTSP if the number of customers is greater than 50 and the probability of each customer requiring a visit is less than 60%.

In [9], Bertsimas and Howell explore the use of TSP heuristics for solving the PTSP and propose an algorithm for the PTSP based on constructing an initial solution using the spacefilling curve heuristic [10] followed by local search. Variations of the 2-OPT and 1-Shift techniques developed for the TSP in [11] are introduced that compute the change in objective in an expected value sense. The equations presented in [9] to efficiently compute these improvements have been shown to have small errors [12], but even with these small errors, the authors are able to show improvement based on expected value becomes more important as n becomes large. They also find that expected value based local improvement is particularly important when probability values are significantly less than 1. This confirms the results established by Rossi and Gavioli [8].

There have been recent efforts to speed up local search procedures for the PTSP and closely related problems. In [13], Tang and Miller-Hooks introduce approximate expressions for the PTSP and explain how these can be incorporated in algorithmic approaches. Beraldi et al. [14] have developed efficient neighborhood search techniques for the PTSP with pickup and deliveries which apply to the PTSP. Algorithms are also emerging based on sophisticated metaheuristics. These include an evolutionary algorithm [15], a stochastic annealing approach [16], and ant colony metaheuristics [17,18]. An exact approach was introduced in [19], but computational tests indicate success only with instances of 50 customers or less. The authors found that among the instances studied, it was much harder to solve instances with low individual probability values.

Bertsimas generalizes some of Jaillet's results and discusses a series of other probabilistic combinatorial optimization problems (PCOPs), such as the probabilistic minimum spanning tree and vehicle routing problems, in his thesis [20] and related papers [9,21,22]. In this class of problems, each node i is present with probability p_i where $i = 1, \ldots, n$. Significant contributions include extending performance results for heuristic algorithms for the deterministic problems to the probabilistic context, including the spacefilling curve heuristic for the TSP to the PTSP.

In the probabilistic vehicle routing problem (PVRP), both customer realization and total customer demand are random variables. The expected cost of such a solution must account for the fact that the truck will have to return to the depot if capacity is reached. An exact approach was introduced for this problem by Gendreau et al. in [23]. The problem is formulated as a stochastic integer program and solved by means of an integer L-shaped method. The largest instances solved involve 70 customers, and solutions could not be found for all of the instances within the time restrictions. It is interesting to note the authors' observation that the presence of stochastic customers (like in the PTSP) makes the problems more difficult than the stochastic demand (not in the PTSP). Many papers focus purely on the stochasticity of the demand (known as the SVRP).

As indicated in the introduction, there has been an extensive study of aggregation in solving location problems. In most of the location literature, customers are deterministic, and the distances considered are direct from customers to the depots or stores without any routing between customers. In this context, aggregation refers to grouping a set of customers together and assigning them as a group to be served by the same facility. Many papers introduce and evaluate aggregation schemes for specific problems, such as [24] for maximum covering problems. Francis and Lowe [25] were the first to look at bounding the error associated with a particular aggregation scheme and introduced the idea of using such a bound to drive the choice of aggregation scheme. Ref. [26] is an example of a paper that follows this approach to design the method of aggregation for a specific location problem. The notion of specific types of error that can be created by aggregation has also been discussed, such as by Current and Schilling [27], and has led to the development of alternative aggregation schemes, such as the one in [28] for the p-median problem.

The closest work appears to be [29], where the authors aggregate deliveries by postal codes due to the large number of home deliveries required from a mail order company. The authors argue that the company should have sufficient historical delivery information to estimate the number of requests for a particular post code, but they do not consider any other options or levels to this aggregation scheme. Each postal code is represented by a single (x, y) coordinate that is the weighted center of homes in the postal district, similar to what we will do here, and the demand is based on the peak number of deliveries to each region on a single day. Based on these estimates, the authors solve a fairly traditional vehicle routing problem to link the postal codes and propose using these as fixed routes. There is no notion of expected cost, nor is there any discussion of how the postal codes are converted into the final routes that visit individual homes. The paper primarily focuses on how to handle the added constraint that restricts each truck to travel only between adjacent postal codes.

3. The model with aggregation

In a solution to the PTSP, all of the customers are sequenced on one tour. On the day of service, when all demands are known, the customers that have been realized can be visited in the sequence defined by this a priori tour. Solution methods for the PTSP focus on minimizing the expected cost of these final tours. If aggregation is used, each customer is assigned to a particular region, and the a priori tour becomes a tour through regions rather than customers. Thus, we need a way to evaluate the expected cost of a sequence of regions.

The expected cost associated with a particular sequence of customers $1, \ldots, n$ can be evaluated by Eq. (1) [20]:

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} p_i p_j d_{i,j} \prod_{k=i+1}^{j-1} (1 - p_k) + \sum_{i=2}^{n} \sum_{j=1}^{i-1} p_i p_j d_{i,j} \prod_{k=i+1}^{n} (1 - p_k) \prod_{l=1}^{j-1} (1 - p_l), \tag{1}$$

where

- p_i: the probability of customer i requiring a visit,
- $d_{i,j}$: distance between customers i and j.

The first part of the equation represents the expected cost associated with using an arc (i, j) in a *forward* direction while the second part is the expected cost associated with using an arc in the *reverse* direction to complete the tour. The expected cost of an arc is based on the probability that the customers at both end points of the arc are realized, the probability that the none of the customers in between these on the tour are realized, and the length of the arc.

A key reason for doing aggregation is to reduce the number of terms in the expected cost expression, but we need to be careful to make sure the new cost expression properly reflects the full problem. If the n customers are each assigned to one of r regions, we must determine a new probability value for each region as well as compute new distance costs. The probability associated with each region S will need to reflect all of the customers assigned to this region. Since a tour will travel to each region only once, we can define p_S as the probability that region S will require a visit. We can compute p_S with Eq. (2).

$$p_S = 1 - \prod_{i \in S} (1 - p_i). \tag{2}$$

This is the probability that region S will have at least one realized demand given that customer orders are independent events. It should be clear that as customers are added to a region, p_S will increase.

Next, we will compute distances between regions. To compute Euclidean distances, we need a spatial location to represent each region. There are many references in the location literature that discuss the virtues of using the centroid of a region rather than the median, such as [28]. Based on this, we propose a weighted variation of the centroid calculation in Eqs. (3) and (4), where the weights are based on the individual customer probabilities.

$$x_S = \frac{\sum_{i \in S} p_i x_i}{\sum_{i \in S} p_i}, \tag{3}$$

$$y_S = \frac{\sum_{i \in S} p_i y_i}{\sum_{i \in S} p_i}. \tag{4}$$

Note that we must divide the numerator in each of these equations by the sum of the customer probabilities in that region. This is because the sum of probabilities for the customers in a particular region will not necessarily equal one.

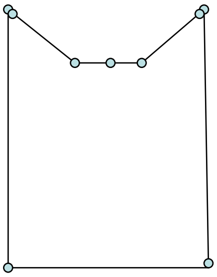

Fig. 1. Nine customer TSP.

We can use these coordinates to compute the Euclidean distance between each pair of regions. Eq. (1) can then be replaced by the aggregated a priori expected value equation found in Eq. (5):

$$\sum_{S=1}^{r-1}\sum_{T=S+1}^{r} p_S p_T d_{S,T} \prod_{U=S+1}^{T-1}(1-p_U) + \sum_{S=2}^{r}\sum_{T=1}^{S-1} p_S p_T d_{S,T} \prod_{U=S+1}^{r}(1-p_U)\prod_{V=1}^{T-1}(1-p_V), \quad (5)$$

where

- $d_{S,T}$: distance between regions S and T.

Eq. (5) greatly resembles Eq. (1), but the number of terms here can be several orders of magnitude smaller than Eq. (1) because of aggregation. By keeping Eq. (5) similar to Eq. (1), we can use the solution methods established for the PTSP to evaluate the impact of aggregation.

3.1. Bounding issues

In the location literature, the solution found using centroids to represent each region provides a lower bound for many of the problems studied [25]. For the PTSP, though, using centroids and their associated probabilities provides neither an upper nor a lower bound for the full PTSP. Examples where aggregation, via Eq. (5), can underestimate the total expected cost can be found easily since this equation does not include the distance traveled between customers in a region. Aggregation can sometimes, though, create an overestimate of the full PTSP objective. This can occur because, unlike in location problems, we have to consider the cost to travel between regions. The centroid of the points in a region may be more expensive to include in a route than the component customers due to the proximity to other customers or regions. The following example demonstrates this point. For simplicity, we will consider aggregating just two customers where all customers have $p = 1$. The original TSP tour is given in Fig. 1. If the two customers highlighted and circled in Fig. 2 are combined and represented with their centroid, we get Fig. 3. If Fig. 1 is the optimal TSP tour, removing the two customers and replacing them with their centroid clearly increases the expected cost of the tour.

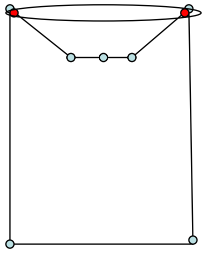

Fig. 2. The customers to be aggregated.

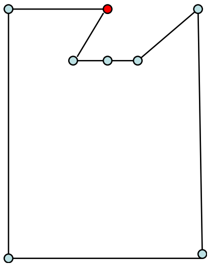

Fig. 3. Tour through the centroid.

3.2. Extensions

The strength of this model is its simplicity and comparability to the original equation, but its potential weaknesses lie in its inherent assumptions. It implies that the distance traveled within a region (if the number of realized demands is greater than 1) does not impact the solution. These costs, in fact, do not impact the ordering of regions, but ignoring them may impact how closely the expected cost of the aggregated problem resembles the expected cost of the full problem. We propose two modifications to the objective function to reflect customer interaction.

3.2.1. Roundtrip approximation

The first involves a new parameter E_S defined in Eq. (6):

$$E_S = \sum_{i \in S} 2 p_i d_{i,S}, \tag{6}$$

where

- $d_{i,S}$ is the Euclidean distance between a customer and the centroid of its assigned region.

E_S represents the expected cost of serving each customer in a region S via a roundtrip from the centroid and serves as a rough estimate of the cost to serve a region. We can add the E_S terms to Eq. (5) to create Eq. (7).

$$\sum_{S=1}^{r-1}\sum_{T=S+1}^{r} p_S p_T d_{S,T} \prod_{U=S+1}^{T-1}(1-p_U) + \sum_{S=2}^{r}\sum_{T=1}^{S-1} p_S p_T d_{S,T} \prod_{U=S+1}^{r}(1-p_U)\prod_{V=1}^{T-1}(1-p_V) + \sum_{S=1}^{r} E_S. \tag{7}$$

3.2.2. Spacefilling curve approximation

A second approach is to compute E_S in a slightly more rigorous manner. Since the regions are identified prior to solving the aggregated PTSP, we can create a tour through each region and use it to compute an E_S value. For consistency, we propose using a spacefilling curve construction heuristic [10] to order the customers in each region. We can then evaluate its expected cost using Eq. (1) and use this cost as the region's E_S value. The objective function then remains the same as in Eq. (7).

Note that regardless of how E_S values are computed, we can use such an approach to expand the final tour through regions. Once the tour between regions is optimized, given any definition of E_S, we can create tours among all of the customers in each region using spacefilling curve and simply link these subtours together to create a full a priori tour. We will examine this idea in our computational experiments.

4. Aggregation schemes

As stated earlier, the primary literature on aggregation comes from location theory, and there are a variety of ways suggested to group customers together based on distance. Here probability is another dimension to the problem that we need to consider. Grouping customers strictly based on location may lead to regions with very different probabilities of requiring a visit. Likewise grouping customers so that each region is equally likely to require a visit may create regions of very different size and shape. Thus, for this initial study, we will look at two simple aggregation approaches that capture these different ideas. These are clearly not comprehensive but should create insight into what creates a successful aggregation scheme for PTSP problems.

4.1. By distance

The first aggregation scheme will be based on distance and is a grid-based approach, such as those discussed in many location papers [26]. For a given parameter g, the customer service area will be divided evenly into g segments along the x-axis and g segments along the y-axis to create a total of g^2 regions of equal area. If customers are uniformly distributed, each region should have roughly the same number of customers.

The potential downfall of such an approach lies in the fact that not all customers are evenly distributed over the service area in real world applications. If some grids have many more potential customers than others, such as if one grid contains an apartment complex where another contains a developing subdivision, it is not clear that the true problem will be represented very well.

4.2. *By probability*

The second form of aggregation is to divide the customer service area into regions of roughly equal probability. This may help remedy some of the possible negative issues with grids since customers should be fairly evenly distributed among the regions, but now the regions will clearly not be of the same size.

To divide potential customers into regions based on probability, we define a parameter *maxp* that represents the maximum likelihood of requiring a visit in a region. Starting from an initial region that includes all of the customers, we proceed as follows. We compute the likelihood that the current region will require a visit using Eq. (2). If it is greater than *maxp*, we determine if the current region is wider horizontally or vertically. We create a new region and start removing customers from the current region and assigning them to the new region. If the current region is wider horizontally, we start by assigning the customer in the current region with the smallest x-coordinate to the new region and keep assigning to the new region based on the lowest x-coordinate until the probability is evenly distributed between both regions. This is similar to Voronoi Diagram/Delaunay Triangulation ideas [30]. The process is the same if the current region is larger in the y-direction, where we re-assign customers to the new region based on y-coordinates. This procedure repeats until all regions have probability less than or equal to *maxp*. By looking at the shape of the current region and using this to guide our division process, we preserve some of the advantages of the grid approach and keep the regions from being extremely tall or wide which would distort distance calculations.

5. Related error result

It is difficult to make many statements about the error associated with combining aggregation with heuristics for solving the PTSP. We can make some statements, though, about a deterministic version of the problem, which allow us to make some claims about the probabilistic version due to a published result.

We start by defining a new deterministic variation of our problem, *TSP-REG*, where the customers are each assigned to one of r $(r \geqslant 2)$ regions. The problem is to sequence the customers within each region as well as the order of the regions themselves to minimize the total distance traveled to visit all of the customers. The optimal solution to this version of the TSP is represented by $OPT_{TSP\text{-}REG}$ where OPT_{TSP} is the value of the optimal solution to the TSP without regions. The two values are related in the following way:

Theorem 1. *$OPT_{TSP\text{-}REG} \leqslant (2 + 2r)OPT_{TSP}$ when distances between customers obey the triangle inequality.*

Proof. We can demonstrate this using a similar argument to the one by Christofides for the TSP in [31]. One simple heuristic for solving *TSP-REG* starts by finding the minimum spanning tree T that connects all of the customers. If we take this minimum spanning tree T and remove the arcs between customers in the same region, we are left with a region spanning tree T^{R}. For each region i, $i = 1, \ldots, r$, we can also start from a copy of T and remove the arcs that are not necessary for the nodes in region i to remain connected. The remaining graph for each region will be labeled T_i.

Choose an arbitrary starting region i and find a customer assigned to that region that is included in T^{R} (there must be at least one). From here, the tour proceeds to each node in region i using the arcs of T_i, traversing each arc at most twice to complete the tour. When the tour through the region is complete, the tour uses arcs in T^{R} to travel to another region that has not yet been toured. This process repeats until all regions and customers have been visited, and then the tour returns to its starting point.

The tour between all of the regions has a cost of no more than twice the cost of the arcs of T^{R} (which a subtree of T), and each region i can be visited at no more than twice the cost of T_i (which is also a subtree of T). Note that without any specific restrictions on distance, the cost of the minimum spanning tree for the customers within a region is not necessarily less than the cost of T_i. If we restrict distances between customers to obey the triangle inequality, as we do, this relationship will hold. Thus, the upper bound for the cost of this heuristic for *TSP-REG* is $2MST + r(2MST)$ where *MST* is the cost of the minimum spanning tree T. It is a well known fact that $MST \leqslant OPT_{TSP}$, so we can expand this result to:

$$OPT_{TSP\text{-}REG} \leqslant 2MST + r(2MST) \leqslant 2OPT_{TSP} + r(2OPT_{TSP}). \tag{8}$$

Since $OPT_{TSP} \leqslant OPT_{TSP\text{-}REG}$, this means we have a $2+2r$ approximation algorithm for $OPT_{TSP\text{-}REG}$. □

We can use this result to make a claim about the probabilistic version of *TSP-REG* given the following theorem.

Theorem 2 (*Bertsimas et al.* [22]). *Let* L_{D} *be the length of the optimal solution to the deterministic PCOP and let* L_{H} *be the length of the heuristic solution to the same problem. Let* p *be the coverage probability and* $E[L_p]$ *be the expected length of the a priori solution to the corresponding PCOP. If the heuristic has the property*

$$\frac{L_{\mathrm{H}}}{L_{\mathrm{D}}} \leqslant c \tag{9}$$

then

$$E\left[\frac{L_{\mathrm{H}}}{E[L_p]}\right] \leqslant \frac{c}{p}. \tag{10}$$

Theorem 3. *Let* L_{H} *be the length of the heuristic solution to TSP-REG. Let* p *be the coverage probability and* $E[L_p]$ *be the expected length of the a priori solution to PTSP-REG. The two values are related according to*:

$$E\left[\frac{L_{\mathrm{H}}}{E[L_p]}\right] \leqslant \frac{2+2r}{p}. \tag{11}$$

This result is based on a simple substitution from Theorem 1 into Theorem 2. Note that as p gets smaller, the gap widens. This supports the idea that it is less effective to solve deterministic versions of the problem when p values are low.

6. Computational experiments

The next step is to evaluate how these choices concerning aggregation impact the solutions computationally. We present experiments with many different data sets and options to get a good picture of the relationship between various factors. Following the structure of Bertsimas and Howell's computational experiments in [9], we consider randomly generated data sets where customers are uniformly distributed about a square and report the average objective value from 10 data sets. While their experiments focused on 100 customer data sets, we are interested in evaluating how the impact of aggregation changes with the number of customers. Thus, we created 10 data sets of 100 customers, 250 customers, 500 customers, and 1000 customers each. Additionally, we consider a selection of well known TSPLIB data sets to see how the results change when customers are clustered.

For both uniform and clustered data sets, we examine variations where all customers have a 1% chance of being realized ($p = .01$) and where all customers have a 10% chance of being realized ($p = .10$). As discussed earlier, we were primarily motivated by applications where this probability value is low. These two values should help us understand how probability impacts the success of aggregation within this lower range.

We examine the use of grid based aggregation (*grid*) as well as dividing customers into regions based on probability. We experiment with g values of 2, 5, 7, and 10, in addition to solving a PTSP without aggregation. For consistency, we choose maximum probability values, *maxp*, such that the number of regions created when aggregation is based on probability is roughly the same as when grids are used. For example, if we want to find a *maxp* value that induces approximately 4 grids when there are 100 customers with 1% probability, we compute *maxp* as follows. First we determine the number of customers that would be in each region if customers are evenly divided. In this example, this would be $100/4 = 25$. We then compute the probability that a region of this size would require a visit using Eq. (2) and use this as the *maxp* value. Here this would be $maxp = 1 - (.99)^{25}$ since this is 1 minus the probability that none of the 25 are realized. For simplicity, the level of aggregation in all of our experiments will be represented by the associated g value.

For each data set and choice of aggregation scheme, we solve the resulting PTSP five ways. This is such that the results are not biased by a poor choice of heuristic. We allow each heuristic the same amount of CPU time (120 s). With this time limit, many aggregated versions converge where the full versions would not unless the limit was increased by several magnitudes. In [9], the authors recommend using a spacefilling curve heuristic to construct an initial solution. They recommend improving the solution using a TSP improvement method, such as 2-OPT, if the probability values are high or an expected value based improvement method, such as a modified version of 1-Shift, if probability values are low. A 2-OPT, or 2-p-OPT, improvement method evaluates the change in objective that results from reversing the tour between each pair of nodes. A 1-Shift, or 1-p-Shift, improvement method looks at removing each node from its current position in the tour and inserting it at all other points in the tour. Based on their recommendations, the five methods tested are:

1. SFC: spacefilling curve construction heuristic only.
2. 2-p-OPT: spacefilling curve followed by an expected value version of 2-OPT.
3. 1-p-Shift: spacefilling curve followed by an expected value version of 1-Shift.
4. 2-OPT: spacefilling curve followed by 2-OPT.

Table 1
Average *REG* values for 100 customers with 10% probability

	None		2-p-OPT		1-p-Shift		2-OPT		1-Shift	
	grid	*maxp*	*grid*	*maxp*	*grid*	*maxp*	*grid*	*maxp*	*grid*	*maxp*
2	383.34	380.80	383.34	380.80	383.34	380.80	383.34	658.76	383.34	380.80
5	575.60	580.38	563.07	568.80	562.20	567.59	573.63	612.42	579.55	578.31
7	592.69	597.38	579.14	584.34	578.00	583.32	590.86	607.47	594.18	597.42
10	600.72	606.90	586.39	591.80	585.14	598.53	600.74	606.96	600.30	606.53
ALL	606.47	606.47	591.41	591.41	598.30	598.30	606.22	606.22	605.97	605.97

5. 1-Shift: spacefilling curve followed by 1-Shift.

As in [9], each improving swap that is encountered in the 2-OPT or 2-p-OPT routines is made, where the 1-Shift procedures choose the best swap among all of the options in a given iteration. All of these methods converge when there are no more improving swaps to be made. For ease of comparison, we will often show the results from just one of these solution approaches.

In these experiments, we examine a series of solution values, representing different ways aggregation can be used. These include the value of the region based objective function (*REG*), as described by Eq. (5) and the extensions in Section 3.2. By comparing these results to the solution values without aggregation, we can get a good idea of how well an aggregated objective approximates the full problem. By looking at the CPU times for these experiments, we can evaluate the savings in CPU time. We can also expand the aggregated PTSP solution to include all of the customers. This involves using a spacefilling curve heuristic to quickly order the customers within each region, linking the regions together to create a full a priori tour, and then evaluating its cost (*FULL*). This requires essentially no extra time, but allows for a direct comparison of approaches to solving the full problem. To take full advantage of the time limit, we can make additional improvements if time permits. Thus, one set of experiments considers taking the subtours created for each region and using any remaining time to improve these subtours on a customer level. These final objective values will be labeled *REGIMP*.

All of these experiments were coded in C and carried out on a 2.66 GHz Pentium IV processor. The uniformly distributed data sets can be found at *myweb.uiowa.edu/acampbll*.

6.1. Uniformly distributed data sets

The results for the uniformly distributed problems confirm the results in [9] that expected value based improvement methods create significantly better final solution values for PTSP's. For example, consider Table 1 that presents the results of the five heuristics averaged from the 100 customer data sets with $p = .10$ and $E_S = 0$. Each row in these tables represents a different g value with the last row representing the PTSP solution without aggregation. The labels *grid* and *maxp* denote the results when grid based and probability based aggregation are used, respectively. The table shows that 2-p-OPT and 1-p-Shift approaches create significant improvements in the solution values over the solutions established by the construction heuristic, where the traditional 2-OPT and 1-Shift routines do not. The same behavior appears to hold for the aggregated versions of the problems and is consistent throughout all of the experiments with

Table 2
Average CPU time for 100 customers with 10% probability

	None		2-p-OPT		1-p-Shift		2-OPT		1-Shift	
	grid	*maxp*	*grid*	*maxp*	*grid*	*maxp*	*grid*	*maxp*	*grid*	*maxp*
2	0.01	0.02	0.02	0.03	0.02	0.02	12.02	24.02	0.02	0.02
5	0.02	0.02	0.05	0.13	0.14	0.62	96.01	120.01	0.03	0.03
7	0.03	0.02	0.58	3.57	4.53	44.45	120.01	120.01	0.04	0.09
10	0.03	0.03	3.82	35.05	44.15	120.00	120.01	120.01	0.08	0.36
ALL	0.07	0.07	35.88	35.88	120.01	120.01	120.01	120.01	0.34	0.34

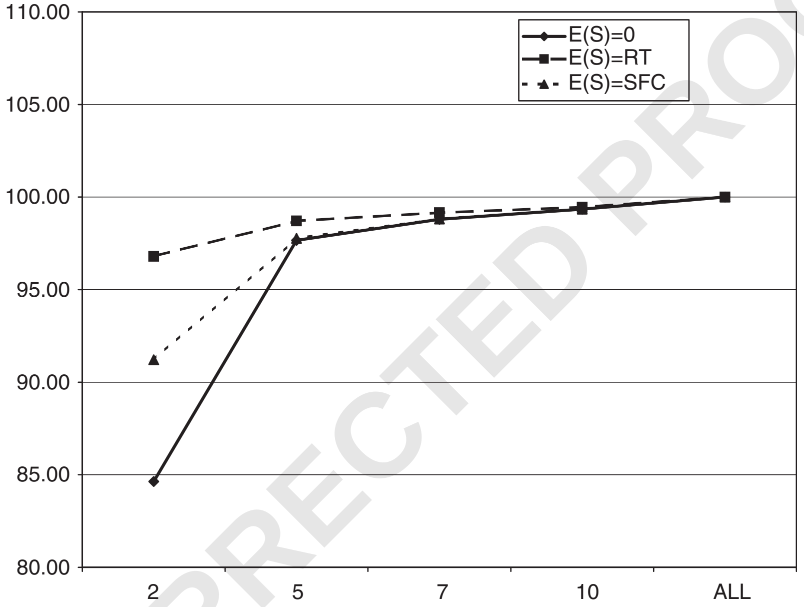

Fig. 4. Average *REG* values for 250 customers with 1% probability with probability based aggregation.

the uniformly distributed data sets. Thus, in the tables we will focus on the results from using 2-p-OPT, unless otherwise indicated. Another interesting point concerning the results from the five approaches is the amount of CPU time used. Table 2 demonstrates that even though 2-OPT leads to little improvement in expected cost for the full problem, it often uses the full CPU time available. Like the objective values, the CPU times reported will be those obtained from using 2-p-OPT.

Given this, we will now look at the performance of the different objective functions. Earlier in the paper, we presented three different objective functions to approximate the expected value of an aggregated solution. The three differ in terms of how the cost to travel within a region is modelled. The first ($E_S = 0$) includes no cost for travel within a region, the second ($E_S = RT$) includes an expected roundtrip to each customer from the region's centroid, and the third ($E_S = SFC$) uses the spacefilling curve algorithm to order the customers in a region and computes the expected cost of this mini-tour. Fig. 4 illustrates these

Table 3
Average *REG* values for 1% probability for uniformly distributed data sets

	100		250		500		1000	
	grid	*maxp*	*grid*	*maxp*	*grid*	*maxp*	*grid*	*maxp*
2	81.82	90.73	80.22	84.64	74.94	84.40	62.55	62.53
5	98.11	98.37	97.36	97.66	96.09	96.09	92.87	92.89
7	99.24	99.42	98.69	98.80	97.59	97.87	95.57	96.02
10	99.68	99.95	99.28	99.35	98.35	98.42	96.82	96.92
ALL	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

different objective functions in terms of the average values they create as the level of aggregation (g) changes. This figure is based on the 250 customer data sets with $p = .01$, and the shape of this graph is typical of our experiments. For ease of comparison, the y-axis represents the percentage of the solution found using 2-p-OPT without aggregation, and the x-axis corresponds to the different g values. In Fig. 4, it is clear that using $g = 2$ leads to serious underestimates of the full problem when $E_S = 0$. This is not surprising since this is where the largest number of customers are included in each region. When $g = 2$, though, the alternate objective functions help close the gap considerably. These graphs demonstrate that representing a 250 customer set with only 25 regions ($g = 5$) can yield a very tight estimate of the full objective value even without considering the travel between customers within a region. This indicates that travel within regions makes up a small part of the objective function if regions are selected well, and for reasonable choices of g, the choice of how to compute E_S is not critical in achieving a good estimate of the full objective.

Next we want to consider how the *REG* estimates respond to changing n and p values. For these experiments, we will simply consider the form of the objective where $E_S = 0$. The results for the data sets with customer probabilities of 1% are summarized in Table 3. As with the previous graph, the results provided are percentages of the 2-p-OPT solutions found without aggregation. As we would expect, Table 3 indicates that as g increases, the *REG* values get closer to the objective values found without aggregation. Not surprisingly, the initial gaps grow wider as the number of customers (n) increases. What is surprising, though, is how quickly the results from a very coarse approximation of the problem closely resemble the objective without aggregation (even with $E_S = 0$). In an application such as the one that motivated this study, it is helpful to know that, on average, a 7 × 7 grid can yield in under 3 s an approximation within 5% of the solution found without aggregation for 1000 customers. The CPU times associated with these results are included in Table 4. Even with $p = .10$, we can still get a good estimate of the full objective function with fairly low g values. Based on the set of experiments, one possible rule of thumb to achieve an objective function within 90% of the full objective is to choose a level of aggregation such that the expected demand in each region is less than or equal to .5. This rule holds fairly consistently across the different choices of p and n for the uniformly distributed data sets. In these experiments, we found that all aggregate methods converge in less than the time limit, but grid based approaches converge much faster. Grid based approaches also often lead to slightly lower objective values.

Given the time limit and the heuristic nature of the solution techniques, the solutions found without aggregation are clearly not optimal and are thus overestimates of the best solution of the associated PTSP. Thus, the gaps between the *REG* values and the values obtained without aggregation can result from both

Table 4
Average CPU time for 1% probability for uniformly distributed data sets

	100		250		500		1000	
	grid	*maxp*	*grid*	*maxp*	*grid*	*maxp*	*grid*	*maxp*
2	0.02	0.03	0.06	0.06	0.24	0.26	1.67	1.66
5	0.06	0.13	0.09	0.15	0.28	0.34	1.73	1.85
7	0.60	3.82	0.99	3.58	1.21	3.87	2.70	5.48
10	3.63	35.44	23.75	112.53	36.61	116.52	45.90	118.39
ALL	39.68	39.68	120.02	120.02	120.08	120.08	120.94	120.94

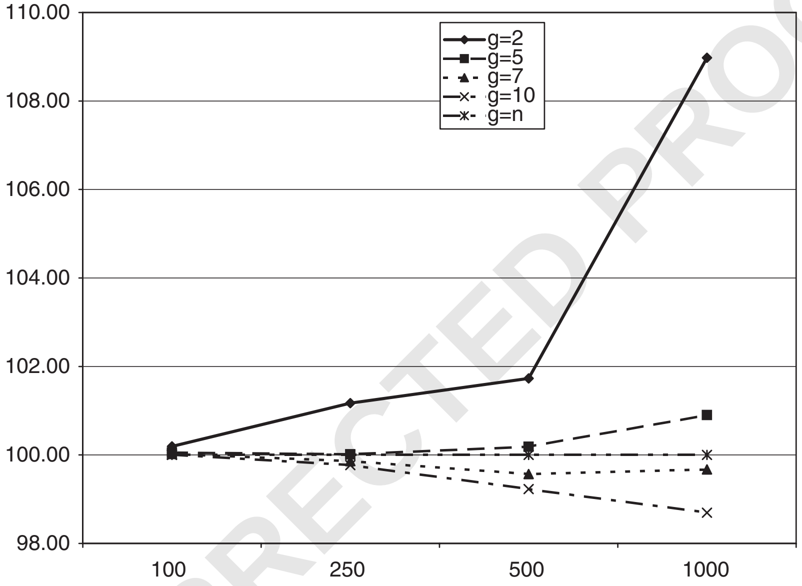

Fig. 5. Average *FULL* values for 1% probability with grid based aggregation.

the fact that aggregation may lead to an underestimate of the PTSP solution and the solutions involving all customers may be overestimates of this value. In the next set of experiments, we expand the aggregate a priori tour into a tour involving all customers and evaluate its expected value. These values, *FULL*, allow a fair evaluation of the tradeoff between optimizing over regions instead of solving the whole problem on a customer level and require no real additional CPU time.

Solution values for 1% probability are presented graphically in Fig. 5. In this figure, the x-axis represents the number of customers, the y-axis represents the percentage of the solution found without aggregation, and the graphed lines correspond to different g values. The results in Fig. 5 indicate that even though coarse aggregation may lead to an underestimating *REG* value, the sequence it proposes can be expanded to create a solution with an expected value very similar to the one found without aggregation. For example, the a priori solution with $g = 2$ and $n = 500$ only exceeds the solution found without aggregation, on average, by less than 2%. For most data set sizes, the 7×7 and 10×10 grids actually lead to better solutions

Table 5
Average *FULL* vs. *REGIMP* values for 1% probability for 100 and 250 customers

	100				250			
	FULL		*REGIMP*		*FULL*		*REGIMP*	
	grid	*maxp*	*grid*	*maxp*	*grid*	*maxp*	*grid*	*maxp*
2	100.19	100.09	81.82	90.73	101.17	100.87	100.47	100.20
5	100.05	100.04	98.11	98.37	100.01	99.99	99.90	99.88
7	100.02	100.02	99.24	99.42	99.86	99.84	99.79	99.80
10	100.01	100.00	99.68	99.95	99.77	99.75	99.75	99.75
ALL	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

Table 6
Average *FULL* vs. *REGIMP* values for 1% probability for 500 and 1000 customers

	500				1000			
	FULL		*REGIMP*		*FULL*		*REGIMP*	
	grid	*maxp*	*grid*	*maxp*	*grid*	*maxp*	*grid*	*maxp*
2	101.73	101.12	101.31	100.86	108.98	107.18	108.85	106.98
5	100.19	99.99	100.15	99.97	100.90	100.49	100.88	100.47
7	99.57	99.50	99.55	99.48	99.67	99.26	99.66	99.25
10	99.23	99.22	99.22	99.22	98.70	98.60	98.69	98.60
ALL	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

than the ones found without aggregation. This indicates that it is more important to make improvements on the level of regions than on the customer level when time is restrictive.

When the p values increase to 10%, the different levels of aggregation lead to predictable steadily improving results except when $g = 2$. There are not as many examples where the average *FULL* values are less than the objectives found without aggregation, but these very tight estimates can be created in considerably less CPU time.

We next consider allowing the remaining CPU time to further improve the aggregated solutions. Since the aggregated problems tend to converge much faster, the leftover CPU time may be used to create improvements at the customer level. The results when customer probability is 1% are presented in Tables 5 and 6. These tables provide the average solution values both before (*FULL*) and after regional improvement (*REGIMP*) to allow for comparison. The *REGIMP* values improve over the *FULL* values by varying degrees. On average, we find aggregation yields better final solution values than without aggregation when roughly 20 customers per region or fewer are used. These improvements are also often accomplished still without using the full CPU time available. From the full set of experiments, we find that starting from an aggregated version of the problem is particularly helpful when time is limited and p values are quite low, but the method of aggregation (grid or probability based) makes little difference in the final objective values.

Table 7
Average *REG* values for *GR*666 with 1% probability

	None		2-p-OPT		1-p-Shift		2-OPT		1-Shift	
	grid	*maxp*	*grid*	*maxp*	*grid*	*maxp*	*grid*	*maxp*	*grid*	*maxp*
2	261.45	343.75	257.67	327.42	257.67	327.42	257.67	327.42	257.67	327.42
5	398.63	417.10	383.21	392.62	383.21	391.57	389.32	394.30	384.38	394.42
7	424.08	424.43	394.37	401.73	391.21	397.88	415.12	410.21	412.95	407.71
10	442.71	430.71	405.14	400.90	398.88	428.83	430.02	412.51	446.63	421.93
ALL	454.33	454.33	454.33	454.33	454.33	454.33	430.14	430.14	454.57	454.57

Table 8
Average CPU times for 1% probability for *GR*666

	None		2-p-OPT		1-p-Shift		2-OPT		1-Shift	
	grid	*maxp*	*grid*	*maxp*	*grid*	*maxp*	*grid*	*maxp*	*grid*	*maxp*
2	0.50	0.52	0.52	0.52	0.50	0.52	120.49	120.49	0.52	0.52
5	0.52	0.53	0.53	0.69	0.58	1.48	120.49	120.50	0.53	0.53
7	0.52	0.55	0.80	5.00	2.84	78.05	120.49	120.51	0.52	0.66
10	0.52	0.52	7.44	120.50	62.46	120.50	120.49	120.50	0.61	3.50
ALL	0.78	0.78	120.13	120.13	120.41	120.41	120.44	120.44	120.45	120.45

6.2. Clustered data sets

Next we explore what happens when data sets are no longer uniformly distributed but are clustered. This should give a better idea of the impact of the choice of aggregation scheme (grid or probability based) and help establish which results hold across a variety of data sets. The key difference here, in terms of the tables, is that we are examining three specific clustered data sets rather than looking at the average over 10 different ones. These include *ALI*535, *DSJ*1000, and *GR*666 of 535, 1000, and 666 customers, respectively.

One surprising result is how poorly 2-p-OPT and 1-p-Shift improvement routines work for the full, unaggregated problem when data is clustered, even if probability values are low. For example, consider Table 7 that presents the results of the five heuristics on the GR666 data set with $p = .01$. 2-p-OPT and 1-p-Shift make no improvement over the initial solution where 2-OPT yields significant differences. In Table 8, we see that the 2-OPT experiments require the full time limit where many of the other methods do not. This creates some challenges in terms of presenting the results, since the aggregated solutions based on expected value based improvement approaches are often still the best and are found much faster. The values used in the remaining tables will be the percentage the 2-p-OPT values are of the unaggregated solution found using 2-OPT. One likely explanation for the difference in performance between methods is that once customers are aggregated into regions, the distribution of the resulting regions may not be as clustered as in the original data set. The CPU times presented will be also be based on 2-p-OPT.

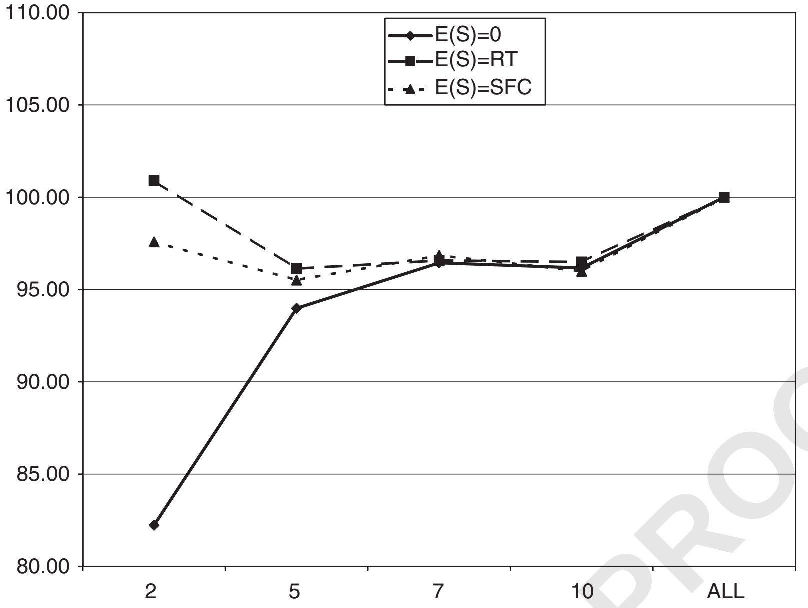

Fig. 6. Average *REG* values for *ALI*535 customers with 1% probability with probability based aggregation.

Table 9
Average *REG* values for 1% probability for clustered data sets

	*ALI*535		*DSJ*1000		*GR*666	
	grid	*maxp*	*grid*	*maxp*	*grid*	*maxp*
2	63.23	82.23	65.76	65.84	59.90	76.12
5	91.70	94.21	90.43	93.69	90.51	91.67
7	96.74	97.14	97.37	103.59	96.51	95.37
10	95.35	97.26	103.18	101.83	99.97	95.90

Fig. 6 compares the three different objectives and ways of defining E_S as the level of aggregation changes when $p = .01$ for *ALI*535. Unlike with uniformly distributed data sets, two of these objectives overestimate the full objective when $g = 2$. Like with the uniformly distributed data sets, there is a big improvement in the quality of the *REG* estimates when $g = 5$. It is also interesting to note that estimates from the probability based solution approach are much tighter than grid based approaches (Tables 9 and 10).

We next examine how the objective with $E_S = 0$ performs with customer probability values of 1% in Table 11. We can see that g values of 7 and 10 again create very tight approximations of the objective. In the majority of these results, the *maxp* average solution value is larger than the associated *grid* value. When probability increases to 10% in Table 12, grid and probability based aggregation schemes again create very different values. As with uniformly distributed data sets, increasing the probability widens the gap between the aggregated and full solution values. The rule of thumb based on expected demand

Table 10
Average CPU times for 1% probability for clustered data sets

	*ALI*535		*DSJ*1000		*GR*666	
	grid	*maxp*	*grid*	*maxp*	*grid*	*maxp*
2	0.30	0.30	1.64	1.70	0.52	0.52
5	0.31	0.45	1.67	1.86	0.53	0.69
7	1.14	4.84	2.12	7.14	0.80	5.00
10	5.28	120.27	11.44	121.59	7.44	120.50

Table 11
Average *REG* values for 1% probability for clustered data sets

	*ALI*535		*DSJ*1000		*GR*666	
	grid	*maxp*	*grid*	*maxp*	*grid*	*maxp*
2	63.23	82.23	65.76	65.84	59.90	76.12
5	91.70	94.21	90.43	93.69	90.51	91.67
7	96.74	97.14	97.37	103.59	96.51	95.37
10	95.35	97.26	103.18	101.83	99.97	95.90

Table 12
Average *REG* values for 10% probability for clustered data sets

	*ALI*535		*DSJ*1000		*GR*666	
	grid	*maxp*	*grid*	*maxp*	*grid*	*maxp*
2	29.26	39.34	26.55	25.87	27.76	32.94
5	64.11	67.56	48.00	56.91	61.70	59.14
7	79.50	78.25	61.11	70.13	70.90	71.98
10	74.27	86.03	77.68	81.67	85.14	77.60

of .5 appears to generally hold, but the large differences between *grid* and *maxp* solutions make it a little harder to verify.

If we expand the aggregated solutions to create tours involving all customers, we obtain the results in Fig. 7 when $p = .01$. In this graph, the x-axis represents the data set, the y-axis represents the percentage of the 2-OPT solution found without aggregation, and the graphed lines represent the different g values. When customers of probability 1% are considered, we see that aggregation can create significant advantages for $g > 2$. This is also accomplished with much less CPU time, since using 2-p-OPT leads to convergence before the time limit in many cases. For both p values, probability based aggregation is clearly preferable.

If the remaining CPU time is used to make improvements at a customer level, we attain the results in Table 13 when customer probabilities equal 1% and Table 14 when customer probabilities equal 10%. We find that the two stage approach of aggregating first and then improving on a customer level can lead

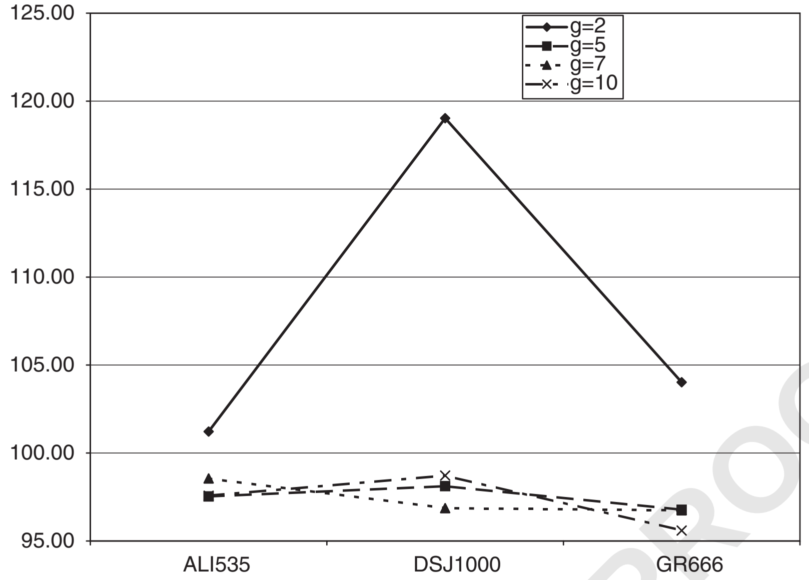

Fig. 7. Average *FULL* values for 1% probability with probability based aggregation for clustered data sets.

Table 13
Average *FULL* vs. *REGIMP* values for 1% probability for clustered data sets

	*ALI*535				*DSJ*1000				*GR*666			
	FULL		*REGIMP*		*FULL*		*REGIMP*		*FULL*		*REGIMP*	
	grid	*maxp*	*grid*	*maxp*	*grid*	*maxp*	*grid*	*maxp*	*grid*	*maxp*	*grid*	*maxp*
2	110.30	101.22	110.12	101.18	123.73	119.03	123.66	118.85	105.15	104.02	105.08	103.92
5	100.19	97.53	100.18	97.50	99.71	98.12	99.69	98.11	100.61	96.77	100.60	96.77
7	99.61	98.55	99.58	98.53	98.27	96.87	98.27	96.86	99.03	96.75	98.97	96.75
10	98.01	97.57	97.98	97.57	99.42	98.72	99.41	98.72	97.81	95.59	97.81	95.59

Table 14
Average *FULL* vs. *REGIMP* values for 10% probability for clustered data sets

	*ALI*535				*DSJ*1000				*GR*666			
	FULL		*REGIMP*		*FULL*		*REGIMP*		*FULL*		*REGIMP*	
	grid	*maxp*	*grid*	*maxp*	*grid*	*maxp*	*grid*	*maxp*	*grid*	*maxp*	*grid*	*maxp*
2	184.66	118.24	182.11	117.76	321.81	301.38	320.97	300.26	158.82	139.17	157.17	138.25
5	115.66	97.71	114.92	96.95	139.09	127.39	138.77	127.24	121.50	103.62	121.36	102.53
7	105.79	97.45	105.01	97.07	126.95	110.28	126.83	110.17	108.96	101.03	108.85	101.00
10	101.92	98.09	101.35	98.09	108.77	102.48	108.65	102.48	103.90	98.25	103.77	98.25

to significant improvements in the final objective values. For g values of 5 or higher, probability based aggregation with improvement yields better solutions than without aggregation on all three data sets when $p = .01$. When $p = .10$, the same is true for *ALI*535 and some improvement occurs for *GR*666. Probability based *REGIMP* values are lower than grid based *REGIMP* values in all of these experiments.

7. Conclusions

The test results indicate that, as expected, using grid vs. probability based aggregation makes little impact on uniform data sets, but can have significant impact when data sets are clustered. We have found that quite coarse levels of aggregation can lead to good objective value estimates, but aggregation needs to become finer as the customer probabilities increase. One general rule we propose is to divide customers into regions such that the total expected demand in a region is no more than .5 in order to achieve an estimate within 90% of the full objective value. These experiments show that even aggregated objective functions that widely underestimate the full objective, but correctly model the distances between regions, can work well in creating a good a priori tour and can do so quickly. We also found that starting from an aggregated solution and using it is a basis for improvement leads to better final solution values than not using aggregation, especially for very low values of p. Our experiments also indicate that many research opportunities remain in solving the PTSP, especially when data is not uniformly distributed.

Acknowledgements

This material is based on work supported by the National Science Foundation, through Grant Number 0237726. The author would like to acknowledge the helpful suggestions made by the two anonymous referees.

References

[1] Bartholdi JJ, Platzman LK, Lee Collins R, Warden WH. A minimal technology routing system for meals on wheels. Interfaces 1983;13:1–8.

[2] Jaillet P, Odoni A. The probabilistic vehicle routing problem. In: Vehicle routing: methods and studies. Amsterdam: North-Holland; 1988. p. 293–318.

[3] Campbell A, Savelsbergh M. Decision support for consumer direct grocery initiatives. Transportation Science; 2004, forthcoming.

[4] Campbell A, Savelsbergh M. Incentive schemes for attended home delivery services. Transportation Science; 2004, under review.

[5] Jaillet P. The probabilistic traveling salesman problems. PhD thesis. Cambridge, MA: Massachusetts Institute of Technology; 1985.

[6] Jaillet P. A priori solution of a traveling salesman problem in which a random subset of the customers are visited. Operations Research 1988;36(6):929–36.

[7] Berman O, Simchi-Levi D. Finding optimal a priori tour and location of traveling salesman with nonhomogenous customers. Transportation Science 1988;22:148–54.

[8] Rossi F, Gavioli I. Aspects of heuristic methods in the probabilistic traveling salesman problem. In: Advanced school on stochastics in combinatorial optimization. Singapore: World Scientific; 1987.

[9] Bertsimas D, Howell L. Further results on the probabilistic traveling salesman problem. European Journal of Operational Research 1993;65:68–95.

[10] Bartholdi JJ, Platzman LK. An $o(n \log n)$ planar traveling salesman heuristic based on spacefilling curves. Operations Research Letters 1982;1:121–5.

[11] Lin S. Computer solution of the traveling salesman problem. Bell System Technical Journal 1965;44:2245–69.

[12] Bianchi L, Knowles J, Bowler N. Local search for the probabilistic traveling salesman problem: correction to the 2-p-opt and 1-shift algorithms. European Journal of Operational Research 2005;162:206–19.

[13] Tang H, Miller-Hooks E. Approximate procedures for the probabilistic traveling salesperson problem. Transportation Research Record; 2004, forthcoming.

[14] Beraldi P, Ghiani G, Laporte G, Musmanno R. Efficient neighborhood search for the probabilistic pickup and delivery travelling salesman problem. Networks; 2004, forthcoming.

[15] S. Rosenow, A heuristic for the probabilistic TSP. In: Schwarze H, et al., editors. Operations research proceedings 1996. Berlin: Springer; 1997.

[16] Bowler NE, Fink TM, Ball RC. Characterization of the probabilistic traveling salesman problem. Physical Review E 2003; 68.

[17] Bianchi L, Gambardella LM, Dorigo M. Solving the homogeneous probabilistic traveling salesman problem by the ACO metaheuristic. Proceedings of ANTS 2002—from ant colonies to artificial ants: third international workshop on ant algorithms, Lecture notes in computer science, vol. 2463. Berlin: Springer; 2002. p. 176–87.

[18] Branke J, Guntsch M. New ideas for applying ant colony optimization to the probabilistic tsp. Proceedings of EvoCOP 2003—third European workshop on evolutionary computation in combinatorial optimization, Lecture notes in computer science, vol. 2611. Berlin: Springer; 2003. p. 166–75.

[19] Laporte G, Louveaux F, Mercure H. A priori optimization of the probabilistic traveling salesman problem. Operations Research 1994;42(3):543–9.

[20] Bertsimas D. Probabilistic combinatorial optimization problems. PhD thesis. Cambridge, MA: Massachusetts Institute of Technology.

[21] Bertsimas D. A vehicle-routing problem with stochastic demand. Operations Research 1992;40(3):574–85.

[22] Bertsimas D, Jaillet P, Odoni A. A priori optimization. Operations Research 1990;38(6):1019–33.

[23] Gendreau M, Laporte G, Seguin R. An exact algorithm for the vehicle routing problem with stochastic demands and customers. Transportation Science 1995;29(2):143–55.

[24] Daskin MS, Haghani A, Khanal M, Malandraki C. Aggregation effects in maximum covering models. Annals of Operations Research 1989;18:115–39.

[25] Francis RL, Lowe TJ. On worst-case aggregation analysis for network location problems. Annals of Operations Research 1992;40:229–46.

[26] Rayco MB, Francis RL, Tamir A. A p-center grid-positioning aggregation procedure. Computers and Operations Research 1999;26:1113–24.

[27] Current JR, Schilling DA. Elimination of source a and b errors in p-median location problems. Geographical Analysis 1987;19(2):95–110.

[28] Zhao P, Batta R. Analysis of centroid aggregation for euclidean distance p-median problem. European Journal of Operational Research 1999;113:147–68.

[29] Beasley JE, Christofides N. Vehicle routing with a sparse feasibility graph. European Journal of Operational Research 1997;98:499–511.

[30] de Berg M, Schwarzkopf O, van Kreveld M, Overmars M. Computational geometry: Algorithms and applications. Berlin: Springer; 2000.

[31] Christofides N. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Carnegie Mellon University, Graduate School of Industrial Administration; 1976.