Solving the Homogeneous Probabilistic Traveling Salesman Problem by the ACO Metaheuristic

Leonora Bianchi[1], Luca Maria Gambardella[1], and Marco Dorigo[2]

[1] IDSIA, Strada Cantonale Galleria 2, CH-6928 Manno, Switzerland
{leonora,luca}@idsia.ch
http://www.idsia.ch
[2] IRIDIA, Université Libre de Bruxelles
CP 194/6, Avenue Franklin Roosevelt 50, 1050 Brussels, Belgium
mdorigo@ulb.ac.be
http://iridia.ulb.ac.be/~mdorigo/

Abstract. The Probabilistic Traveling Salesman Problem (PTSP) is a TSP problem in which each customer has a given probability of requiring a visit. The goal is to find an a priori tour of minimal expected length over all customers, with the strategy of visiting a random subset of customers in the same order as they appear in the a priori tour.
We propose an ant based a priori tour construction heuristic, the probabilistic Ant Colony System (pACS), which is derived from ACS, a similar heuristic previously designed for the TSP problem. We show that pACS finds better solutions than other tour construction heuristics for a wide range of homogeneous customer probabilities. We also show that for high customers probabilities ACS solutions are better than pACS solutions.

1 Introduction

Consider a routing problem through a set V of n customers. On any given instance of the problem each customer i has a known position and a probability p_i of actually requiring a visit, independently of the other customers. Finding a solution for this problem implies having a strategy to determine a tour for each random subset $S \subseteq V$, in such a way as to minimize the expected tour length. The most studied strategy is the a priori one. An a priori strategy has two components: the a priori tour and the updating method. The a priori tour is a tour visiting the complete set V of n customers; the updating method modifies the a priori tour in order to have a particular tour for each subset of customers $S \subseteq V$. A very simple example of updating method is the following: for every subset of customers, visit them *in the same order* as they appear in the a priori tour, skipping the customers that do not belong to the subset. The strategy related to this method is called the 'skipping strategy'. The problem of finding an a priori tour of minimum expected length under the skipping strategy is defined as the Probabilistic Traveling Salesman Problem (PTSP). This is an NP-hard problem [3,1], and was introduced in Jaillet's PhD thesis [8].

The PTSP approach models applications in a delivery context where a set of customers has to be visited on a regular (e.g., daily) basis, but all customers do

M. Dorigo et al. (Eds.): ANTS 2002, LNCS 2463, pp. 176–187, 2002.

not always require a visit, and where re-optimizing vehicle routes from scratch every day is infeasible. In this context the delivery man would follow a standard route (i.e., an a priori tour), leaving out customers that on that day do not require a visit. The standard route of least expected length corresponds to the optimal PTSP solution.

In the literature there are a number of algorithmic and heuristic approaches used to find suboptimal solutions for the PTSP. Heuristics using a nearest neighbor criterion or savings criterion were implemented and tested by Jézéquel [9] and by Rossi-Gavioli [11]. Later, Bertsimas-Jaillet-Odoni [3] and Bertsimas-Howell [2] have further investigated some of the properties of the PTSP and have proposed some heuristics for the PTSP. These include tour construction heuristics (space filling curves and radial sort), and tour improvement heuristics (probabilistic 2-opt edge interchange local search and probabilistic 1-shift local search). More recently, Laporte-Louveaux-Mercure [10] have applied an integer L-shaped method to the PTSP and have solved to optimality instances involving up to 50 vertices.

Most of the heuristics proposed are an adaptation of a TSP heuristic to the PTSP, or even the original TSP heuristic, which in some cases gives good PTSP solutions. No application to the PTSP of nature-inspired algorithms such as ant colony optimization (ACO) [4] or genetic algorithms can be found in the literature. This paper investigates the potentialities of ACO algorithms as tour construction heuristics for the homogeneous PTSP, that is, for the PTSP where customers have the same probability of requiring a visit.

In the remainder of the paper we first introduce the PTSP objective function (section 2), then we describe the ACO algorithms which we tested (section 3), and in section 4 we report the experimental results obtained. The concluding section 5 summarizes the results obtained and indicates future directions for research on the PTSP.

2 The PTSP Objective Function

Let us consider an instance of the PTSP. We have a completely connected graph whose nodes form a set $V = \{i = 1, 2, ..., n\}$ of customers. Each customer has a probability p_i of requiring a visit, independent of the others. A solution for this instance is a tour λ over all nodes in V (an 'a priori tour'), to which is associated the expected length objective function

$$E[L_\lambda] = \sum_{S \subseteq V} p(S) L_\lambda(S) \ . \tag{1}$$

In the above expression, S is a subset of the set of nodes V, $L_\lambda(S)$ is the distance required to visit the customers in S (in the same order as they appear in the a priori tour), and $p(S)$ is the probability that all the customers in S require a visit:

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

Jaillet [8] showed that the evaluation of the PTSP objective function (eq.(1)) can be done in $O(n^2)$. In fact, let us consider (without loss of generality) an a priori tour $\lambda = (1, 2, \ldots, n)$; then its expected length is

$$E[L_\lambda] = \sum_{i=1}^{n} \sum_{j=i+1}^{n} d_{ij} p_i p_j \prod_{k=i+1}^{j-1} (1 - p_k) + \sum_{i=1}^{n} \sum_{j=1}^{i-1} d_{ij} p_i p_j \prod_{k=i+1}^{n} (1 - p_k) \prod_{l=1}^{j-1} (1 - p_l) . \quad (3)$$

This expression is derived by looking at the probability for each arc of the complete graph to be used, that is, when the a priori tour is adapted by skipping a set of customers which do not require a visit. For instance, an arc (i, j) is actually used only when customers i and j do require a visit , while customers $i+1, i+2, ..., j$ do not require a visit. This event occurs with probability $p_i p_j \prod_{k=i+1}^{j-1}(1 - p_k)$ (when $j \leq n$). In the special class of PTSP instances where $p_i = p$ for all customers $i \in V$ (the homogeneous PTSP), equation (3) becomes

$$E[L_\lambda] = p^2 \sum_{r=0}^{n-2} (1 - p)^r L_\lambda^{(r)} \quad (4)$$

where $L_\lambda^{(r)} \equiv \sum_{j=1}^{n} d(j, (j + 1 + r) \bmod n)$. The $L_\lambda^{(r)}$'s have the combinatorial interpretation of being the lengths of a collection of gcd(n,$r + 1$) sub-tours[1] $\lambda_p^{(r)}$, obtained from tour λ by visiting one customer and skipping the next r customers. As an example, Fig. 1 shows $\lambda_p^{(0)}$ (i.e., the a priori tour), $\lambda_p^{(1)}$ and $\lambda_p^{(2)}$ for a PTSP with 8 customers.

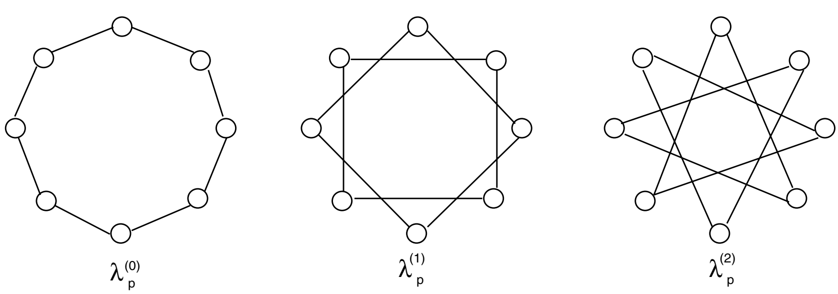

Fig. 1. The lengths of the (sub)tours $\lambda_p^{(0)}$, $\lambda_p^{(1)}$ and $\lambda_p^{(2)}$, constitute the first three terms of the expected length for the homogeneous PTSP. From left to right, the total length of each set of (sub)tours gives the terms $L_\lambda^{(0)}$, $L_\lambda^{(1)}$ and $L_\lambda^{(2)}$ of equation (4).

[1] The term 'gcd' stays for 'greatest common divisor'.

3 Ant Colony Optimization

In ACO algorithms a colony of artificial ants iteratively and stochastically constructs solutions for the problem under consideration using artificial pheromone trails and heuristic information. The pheromone trails are modified by ants during the algorithm execution in order to store information about 'good' solutions. Most ACO algorithms follow the algorithmic scheme given in Fig. 2.

ACO are stochastic solution construction algorithms, which, in contrast to local search algorithms, may not find a locally optimal solution. Many of the best performing ACO algorithms improve their solutions by applying a local search algorithm after the solution construction phase. Our primary goal in this work is to analyze the PTSP tour construction capabilities of ACO, hence in this first investigation we do not use local search.

We consider a particular ACO algorithm, the probabilistic Ant Colony System, or pACS. This is an adaptation to the PTSP of the ACS algorithm [7,5], which was successfully applied to the TSP. In the following, we describe how pACS (and ACS) builds a solution and how it updates pheromone trails.

```
procedure ACO metaheuristic for combinatorial optimization problems
    Set parameters, initialize pheromone trails
    while (termination condition not met)
        ConstructSolutions
        ApplyLocalSearch        % optional
        UpdateTrails
    end while
```

Fig. 2. High level pseudocode for the ACO metaheuristic.

3.1 Solution Construction

A feasible solution for an n-city PTSP is an a priori tour which visits all customers. Initially m ants are positioned on their starting cities chosen according to some initialization rule (e.g., randomly). Then, the solution construction phase starts (procedure *ConstructSolutions* in Fig. 2). Each ant probabilistically builds a tour by choosing the next customer to move to on the basis of two types of information, the pheromone τ and the heuristic information η. To each arc joining two customers i, j it is associated a varying quantity of pheromone τ_{ij}, and a heuristic value $\eta_{ij} = 1/d_{ij}$, which is the inverse of the distance between i and j. When an ant k is on city i, the next city is chosen as follows.

- With probability q_0, a city j that maximizes $\tau_{ij} \cdot \eta_{ij}^{\beta}$ is chosen in the set $J_k(i)$ of the cities not yet visited by ant k. Here, β is a parameter which determines the relative influence of the heuristic information.

– With probability $1 - q_0$, a city j is chosen randomly with a probability given by

$$p_k(i,j) = \begin{cases} \frac{\tau_{ij} \cdot \eta_{ij}^{\beta}}{\sum_{r \in J_k(i)} \tau_{ir} \cdot \eta_{ir}^{\beta}}, & \text{if } j \in J_k(i) \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Hence, with probability q_0 the ant chooses the best city according to the pheromone trail and to the distance between cities, while with probability $1 - q_0$ it explores the search space in a biased way.

3.2 Pheromone Trails Update

Pheromone trails are updated in two stages. In the first stage, each ant, after it has chosen the next city to move to, applies the following local update rule:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \rho \cdot \tau_0, \tag{6}$$

where ρ, $0 < \rho \leq 1$, and τ_0, are two parameters. The effect of the local updating rule is to make less desirable an arc which has already been chosen by an ant, so that the exploration of different tours is favored during one iteration of the algorithm.

The second stage of pheromone update occurs when all ants have terminated their tour. Pheromone is modified on those arcs belonging to the best tour since the beginning of the trial (best-so-far tour) by the following global updating rule

$$\tau_{ij} \leftarrow (1 - \alpha) \cdot \tau_{ij} + \alpha \cdot \Delta\tau_{ij}, \tag{7}$$

where

$$\Delta\tau_{ij} = ObjectiveFunc_{best}^{-1} \tag{8}$$

with $0 < \alpha \leq 1$ being the pheromone decay parameter, and $ObjectiveFunc_{best}$ is the value of the objective function of the best-so-far tour. In pACS the objective function is the PTSP expected length of the a priori tour, while in ACS the objective function is the a priori tour length.

4 Experimental Tests

4.1 Homogeneous PTSP Instances

Homogeneous PTSP instances were generated starting from TSP instances and assigning to each customer a probability p of requiring a visit, with p ranging from 0.1 to 0.9 with a 0.1 interval. We considered 21 TSP instances taken from two benchmarks. The first benchmark is the TSPLIB at http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95. From this benchmark we considered 7 symmetric instances[2] with a number of city between 30 and 200.

[2] The TSPLIB symmetric instances considered are oliver30, eil51, eil76, kroA100, lin105, ch150, d198.

The second benchmark is a group of instances where customers are randomly distributed on the square $[0, 10^6]$. For generating random instances we used the Instance Generator Code of the 8^{th} DIMACS Implementation Challenge at http://www.research.att.com/~dsj/chtsp/download.html. We considered 7 uniform distributed instances and 7 clustered distributed instances from this benchmark, with a number of cities respectively equal to 50, 100, 150,..., 350.

4.2 Comparison between pACS and Other Tour Construction Heuristics

We compared pACS with two simple tour construction heuristics, the radial sort and the random best heuristic. The random best heuristic generates random tours and selects the one with the shortest expected length. Random best and pACS were run on the same machine (a Pentium Xeon, 1GB of RAM, 1.7 GHz processor) for the same CPU time ($stoptime = k \cdot n^2$ CPU seconds, with $k = 0.01$). For pACS we chose the same settings which yielded good performance in earlier studies with ACS on the TSP [5]: $m = 10$, $\beta = 2$, $q_0 = 0.98$, $\alpha = \rho = 0.1$ and $\tau_0 = 1/(n \cdot Obj)$, where n is the number of customers and Obj is the value of the objective function evaluated with the nearest neighbor heuristic [5]. For each experiment, we run 5 independent trials of pACS.

Radial sort builds a tour by sorting customers by angle with respect to the 'center of mass' of the customer spatial distribution. The 'center of mass' coordinates have been computed here by averaging over the customers coordinates. The a priori tour which radial sort builds does not depend on the customer probabilities, and a unique tour is thus used as a priori tour for all probabilities of the PTSP. Even if very simple, this heuristic is interesting for the PTSP, because of the conjecture [2] that the tour generated by radial sort is near optimal for small customer probabilities. Moreover, the combination of radial sort and the 1-shift local search have shown to be the best combination of tour construction and tour improvement heuristics in [2]. A disadvantage of radial sort is that it is only applicable to those PTSP instances where the coordinates of customers are known. In general, this is not the case for asymmetric PTSP instances, where the arc weights may have, for instance, the meaning of travel times.

The average relative performance of pACS with respect to radial sort and random best heuristics is shown in Fig. 3. The first observation is that pACS always performs better than radial sort and random best, for each probability and for each type of instance, while random best is always very poor both with respect to pACS and to radial sort. Secondly, radial sort and pACS are equivalent for small probabilities (prob = 0.1). This results supports the conjecture of near-optimality of radial sorted tours for small probability, and it is interesting that this also applies to non uniform instances, such as TSPLIB and random clustered instances.

4.3 Absolute Performance

For the PTSP instances we tested, the optimal solution is not known. Therefore, an absolute performance evaluation of a PTSP heuristic can only be done against

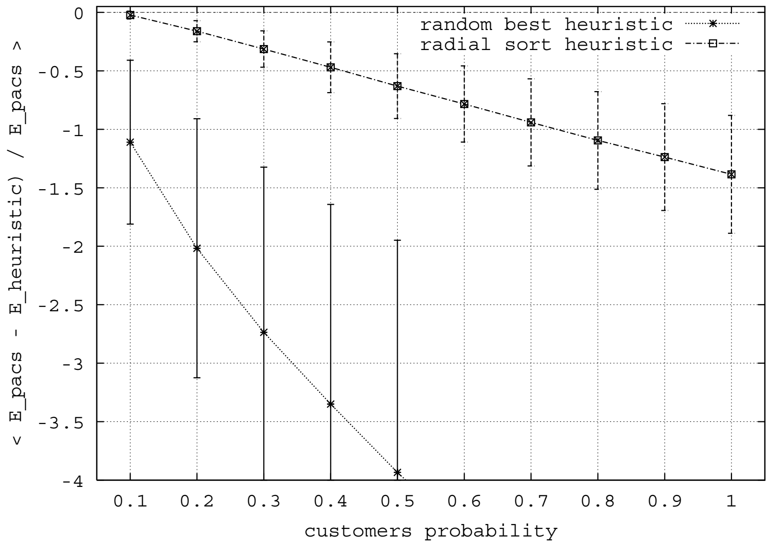

Fig. 3. Relative performance of pACS with respect to radial sort and random best heuristics. On the horizontal axis there is the customers probability. Each point is an average over 21 symmetric PTSP instances. Error bars represent average deviation, defined as $\sum_{i=1}^{n} |x_i - <x> |/n$, with $n = 21$.

some lower bound of the optimal PTSP solution, when this is available and tight enough. A lower bound to the optimal solution would give us an upper bound to the error performed by the pACS heuristic with respect to the PTSP optimum. In fact, if LB is the lower bound and $E[L_{\lambda^*}]$ is the optimal solution value, then by definition we have

$$E[L_{\lambda^*}] \geq LB \ . \tag{9}$$

If the solution value of pACS is $E[L_\lambda]$, then the following inequality holds for the relative error

$$\frac{E[L_\lambda] - E[L_{\lambda^*}]}{E[L_{\lambda^*}]} \leq \frac{E[L_\lambda] - LB}{LB} \ . \tag{10}$$

In the following we apply two different techniques for evaluating a lower bound to the optimal PTSP solution (and thus for evaluating the absolute performance of pACS). In the first case a theoretical lower bound is used while in the second case the lower bound is estimated by using Monte Carlo sampling.

Theoretical lower bound to the PTSP optimum. For the homogeneous PTSP and for instances where the optimal length L_{TSP} of the corresponding TSP is known, it is possible to use the following lower bound to the optimal expected length, as was proved in [2]

$$LB = pL_{TSP}(1 - (1 - p)^{n-1}) \ . \tag{11}$$

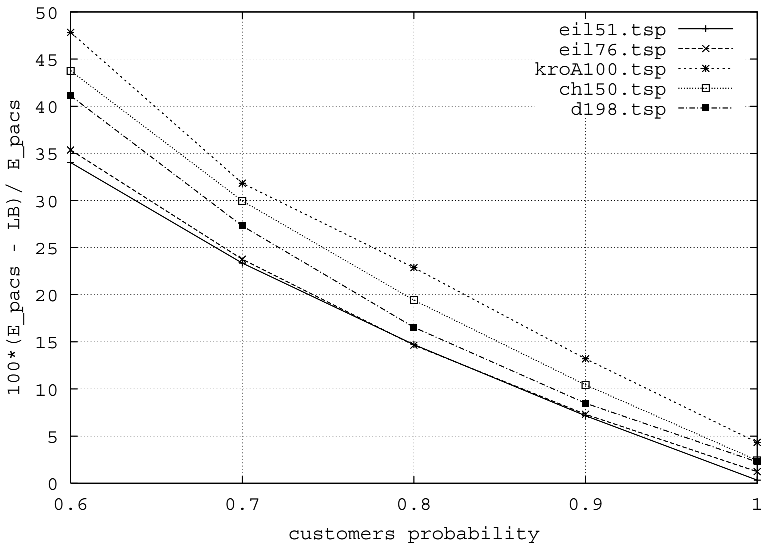

Fig. 4. Upper bound of relative percent error of pACS for 5 TSPLIB instances. Note that, for decreasing customers probability, the upper bound to the relative error becomes bigger at least partially because the lower bound to the optimum becomes less tight.

If we put this lower bound into the right side of equation (10), we obtain an upper bound of the relative error of pACS. Fig. 4 shows the absolute performance of pACS, evaluated with this method, for a few TSPLIB instances. From the figure we see that, for example, pACS finds a solution within 15% of the optimum for a homogeneous PTSP with customers probability 0.9. This technique for evaluating the absolute performance of a PTSP heuristic is rigorous, but has the limitation that the lower bound for small probabilities is not tight, so that it produces big overestimates of the error. The following technique is more flexible and gives better estimates of the error, even if, as we will see, it also has some limitations.

Estimated a posteriori optimum. The expected tour length under re optimization is defined as the average of the lengths of the optimal TSP solution to each subset of customers, and it is also called a posteriori optimum, since it is the value obtained by solving a TSP problem once the set of customers requiring a visit on a certain day is known. The a posteriori optimum is a lower bound on the optimal PTSP solution, because the length induced by the PTSP a priori tour on a subset of customers cannot be smaller than the optimal TSP solutions for that subset of customers.

The exact evaluation of the a posteriori optimum is impractical, because it requires the solution of 2^n instances of the TSP to optimality. The technique proposed in [2], consists in making two approximations. First, only a random sample of the 2^n subsets of customers is selected, by means of a stratified Monte Carlo sampling (see [2] for a detailed description). Second, each random sample of customers S is solved to near optimality as a TSP by choosing the best of $|S|/\gamma$ random tours (γ is a parameter) and applying to it the 3-opt local search.

This technique can be applied easily only to small instances (say, up to 100 customers). Otherwise, care must be taken in order to avoid overflow when generating the stratified samples from a set of 2^n subsets of customers. We report average results for 10 random uniform and clustered instances in Fig. 5. In our tests we used $\gamma = 0.5$ and about 400 samples. From the figure we see that pACS is within 8% of the optimum when applied to uniform random instances, while it is within 14% of the optimum if the instances are clustered.

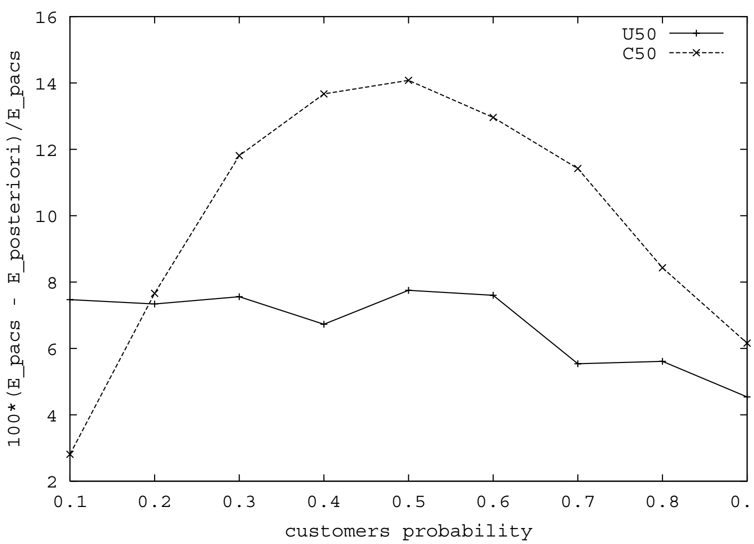

Fig. 5. Relative percent error with respect to the estimated a posteriori optimum for random uniform instances (U50) and for random clustered instances (C50) of 50 customers. Each point is an average over 10 random instances.

4.4 Comparison between pACS and ACS

In some cases an a priori tour found by a TSP heuristic can also be a good solution for the PTSP. An example of this is the a priori tour found by radial sort for small probabilities, as discussed in section 4.2. In this section we address the question of whether an a priori tour found by the ACS heuristic is also good for the PTSP, or at least as good a the solution found by pACS. In order to assess the relative performance of ACS versus pACS independently of the details of the settings, the two algorithms were run with the same parameters. The choice of

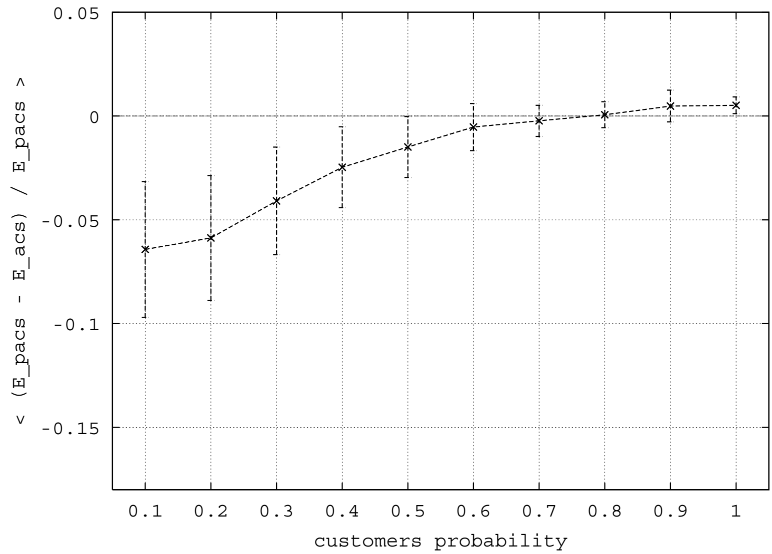

Fig. 6. Relative performance of pACS versus ACS for the homogeneous PTSP. The vertical axis represents $(E[L_\lambda(pACS)] - E[L_\lambda(ACS)])/E[L_\lambda(pACS)]$. On the horizontal axis there is the customer probability p. Each point of the graph is an average over 21 symmetric homogeneous PTSP instances. Note that for $p = 1$ ACS outperforms pACS, since for a fixed CPU stopping time ACS makes more iterations.

this parameter setting is the simplest among many other possibilities for comparing ACS and pACS. In fact, it would also be useful to tune pACS parameters, not only to achieve a better performance, but also to see how much they differ from ACS parameters (which are tuned on the TSP). Fig. 6 summarizes the results obtained. The figure shows the relative performance of pACS versus ACS, averaged over the 21 tested symmetric PTSP instances. From the figure we see that for small enough probabilities pACS outperforms ACS. Nevertheless, for all the problems we tested there is a range of probabilities $[p_0, 1]$ for which ACS outperforms pACS. The critical probability p_0 at which this happens depends on the problem.

The reason why pACS does not always perform better than ACS is clear if we consider two aspects. The first is the time complexity (speed) of ACS versus pACS. In both algorithms one iteration (i.e., one cycle through the *while* condition of Fig. 2) is $O(n^2)$ [6], but the constant of proportionality is bigger in pACS than in ACS. To see this one should consider the procedure *UpdateTrail* of Fig. 2, where the best-so-far tour must be evaluated in order to choose the arcs on which pheromone is to be updated. The evaluation of the best-so-far tour requires $O(n)$ time in ACS and $O(n^2)$ time in pACS. ACS is thus faster and always performs more iterations than pACS for a fixed CPU time.

The second reason why ACS performs better than pACS for high probabilities is that the length of an a priori tour (ACS objective function) may be considered as an $O(n)$ approximation to the $O(n^2)$ expected length (pACS objective function). In general, the worse the approximation, the worse will be the solution quality of ACS versus pACS. The quality of the approximation depends on the set of customer probabilities p_i. In the homogeneous PTSP, where customer probability is p for all customers, it is easy to see the relation between the two objective functions. For a given a priori tour λ of length L_λ we have

$$\Delta = L_\lambda - E[L_\lambda] = (1-p^2)L_\lambda - \sum_{r=1}^{n-2}(1-p)^r L_\lambda^{(r)}, \tag{12}$$

which implies

$$\Delta \sim O(q) \tag{13}$$

for $q \to 0$, with $q = 1-p$. Therefore, the higher the probability, the better is the a priori tour length L_λ as an approximation for the expected tour length $E[L_\lambda]$.

5 Conclusions and Future Work

In this paper we investigated the potentialities of pACS, a particular ACO algorithm, for the homogeneous PTSP. We showed that the pACS algorithm is a promising tour construction heuristic for the PTSP. We compared pACS with other tour construction heuristics and we provided an estimation of the absolute error with respect to the optimal PTSP solution for some instances. We also compared pACS to ACS, and we showed that for customers probability close to 1, the ACS heuristic is a better alternative than pACS.

In this paper the ACO metaheuristic was applied without a local search for improving the a priori tour. The study of an efficient local search for the PTSP, which should greatly improve the solution quality of pACS and of any tour construction heuristic in general, is an important direction of research. At present we are investigating the heterogeneous PTSP, for different probability configurations of customers. This is an interesting issue, since it is closer to a real-world problem than the homogeneous PTSP.

Acknowledgments. This research has been partially supported by the Swiss National Science Foundation project titled "On-line fleet management", grant 16R10FM, and by the "Metaheuristics Network", a Research Training Network funded by the Improving Human Potential programme of the CEC, grant HPRN-CT-1999-00106. The information provided in this paper is the sole responsibility of the authors and does not reflect the Community's opinion. The Community is not responsible for any use that might be made of data appearing in this publication. Marco Dorigo acknowledges support from the Belgian FNRS, of which he is a Senior Research Associate.

References

1. D. J. Bertsimas. *Probabilistic Combinatorial Optimization Problems.* PhD thesis, MIT, Cambridge, MA, 1988.
2. D. J. Bertsimas and L. Howell. Further results on the probabilistic traveling salesman problem. *European Journal of Operational Research*, 65:68–95, 1993.
3. D. J. Bertsimas, P. Jaillet, and A. Odoni. A priori optimization. *Operations Research*, 38:1019–1033, 1990.
4. M. Dorigo, G. Di Caro, and L. M. Gambardella. Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137–172, 1999.
5. M. Dorigo and L. M. Gambardella. Ant Colony System: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.
6. M. Dorigo, V. Maniezzo, and A. Colorni. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 26(1):29–41, 1996.
7. L. M. Gambardella and M. Dorigo. Solving symmetric and asymmetric TSPs by ant colonies. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC'96)*, pages 622–627. IEEE Press, Piscataway, NJ, 1996.
8. P. Jaillet. *Probabilistic Traveling Salesman Problems.* PhD thesis, MIT, Cambridge, MA, 1985.
9. A. Jézéquel. *Probabilistic Vehicle Routing Problems.* Master's thesis, MIT, Cambridge, MA, 1985.
10. G. Laporte, F. Louveaux, and H. Mercure. An exact solution for the a priori optimization of the probabilistic traveling salesman problem. *Operations Research*, 42:543–549, 1994.
11. F. A. Rossi and I. Gavioli. *Aspects of Heuristic Methods in the Probabilistic Traveling Salesman Problem*, pages 214–227. World Scientific, Singapore, 1987.