Multi-type ant colony system for solving the multiple traveling salesman problem

Abstract

The Multiple Traveling Salesman problem (mTSP) is an extension of the well-known Traveling Salesman Problem (TSP), where more than one salesman is allowed to be used in order to visit some cities just once. Furthermore, the formulation of the mTSP applies to a wide range of real-life applications, and can be extended to a wide variety of Vehicle Routing Problems (VRPs) by incorporating some additional side constraints, such as the vehicle capacity and customer demands. Although the literature for the TSP and the VRP is definitely wide, the mTSP has not received the same amount of attention yet. This paper proposes a new algorithm based on Ant Colony Optimization (ACO) for the mTSP, specifically Multi-type Ant Colony System (M-ACS), where each colony represents a set of possible global solutions. Moreover, these colonies cooperate by means of "frequent" pheromone exchanges in order to find a competitive solution for the mTSP. The algorithm performance has been compared with one of the most efficient local search algorithms for TSP, the Lin-Kernighan algorithm. Computational results confirm the competitiveness and efficiency of the strategy we propose.

Keywords: ant colony optimization (ACO); multiple traveling salesman problem (mTSP)

1. Introduction

The classical Traveling Salesman Problem (TSP) is definitely one of the most studied problems in Operation Research literature. However, the Multiple Traveling Salesman Problem (mTSP), in spite of being a generalization of TSP, has been studied much less. This combinatorial problem consists of finding a set of routes for \( m \) salesmen who all start from and turn back to a home city (depot) and each city must be visited by exactly one salesman. Compared to the classical TSP, the mTSP is harder to solve and the solution space is larger than TSP. The characteristics of mTSP seem more appropriated for real-life problems (e.g. our case study) and a wide variety of its application can be found in Bektas (2006). The Table 1 summarizes the possible application areas of the mTSP, where the first column indicates the application context and the second presents the specific type of application.

Some approaches have been formulated to solve the mTSP, such as exact algorithms. In Laporte and Nobert (1980) one of these algorithms is proposed based some constraint relaxation, also a fixed cost \( f \) is associated for each salesman. A Branch-and-Bound (BB) strategy is given by Gavish and Srikanth (1986). The mentioned BB algorithm is tested for large-scale mTSPs and the lower bound is obtained from Lagrangean problem, specifically constructed by relaxing the degree constraints used.

Another exact method is developed by Gromicho et al. (1992), where an asymmetric mTSP with a fixed number of salesmen is considered. A superior BB is applied in the mentioned research; this approach is based on a quasi-assignment (QA) relaxation obtained by relaxing the sub-tour elimination constraints (SECs). Furthermore, a symmetric instance, with 120 nodes and salesman between 2 and 12, is solved using the former approach in Gromicho (2003). The exact procedures consist of mainly in BB algorithms, which are limited to solving only problems of reasonable sizes. In that sense, the main
drawback of exact strategies is that when they need to solve large scale problem mostly have a high running time.

Table 1. Application contexts for the mTSP

<table>
<thead>
<tr>
<th>Application context</th>
<th>Type of application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print Scheduling</td>
<td>Print press scheduling (Gorenstein, 1970)</td>
</tr>
<tr>
<td></td>
<td>Pre-print advertisement scheduling (Carter and Ragsdale, 2002)</td>
</tr>
<tr>
<td>Workforce planning</td>
<td>Bank crew scheduling (Svestka and Huckfeldt, 1973)</td>
</tr>
<tr>
<td></td>
<td>Technical crew scheduling (Lenstra and Rinnooy Kan, 1975)</td>
</tr>
<tr>
<td></td>
<td>Photographer team scheduling (Zhang et al., 1999)</td>
</tr>
<tr>
<td></td>
<td>Interview scheduling (Gilbert and Hofstra, 1992)</td>
</tr>
<tr>
<td></td>
<td>Workload balancing (Okonjo-Adigwe, 1988)</td>
</tr>
<tr>
<td></td>
<td>Security service scheduling (Calvo and Cordone, 2003)</td>
</tr>
<tr>
<td>Transportation planning</td>
<td>School bus routing (Angel et al., 1972)</td>
</tr>
<tr>
<td></td>
<td>Crane scheduling (Kim and Park, 2004)</td>
</tr>
<tr>
<td></td>
<td>Local truckload pickup and delivery (Wang and Regan, 2002)</td>
</tr>
<tr>
<td>Mission planning</td>
<td>Planning of autonomous mobile robots (Yu et al., 2002)</td>
</tr>
<tr>
<td></td>
<td>Planning of unmanned air vehicles (Ryan et al., 1998)</td>
</tr>
<tr>
<td>Production planning</td>
<td>Hot rolling scheduling (Tang et al., 2000)</td>
</tr>
<tr>
<td>Satellite system</td>
<td>Designing satellite surveying system (Saleh and Chelouah, 2004)</td>
</tr>
</tbody>
</table>

Heuristic procedures have shown advantageous results for solving the mTSPs (Sze and Tiong, 2007). One of the most efficient heuristic for mTSP is reported in Lin and Kernighan (1973); the heuristic (Lin-Kernighan) was originally developed for the classical TSP. In this paper we compare the results of this efficient heuristic with the new heuristic approach proposed, using well established instances of mTSP found in literature.

Moreover, an exchange heuristic algorithm for the mTSP is given by Potvin et al. (1989). Further on Fogel (1990) proposes a parallel processing approach for solving the mTSP based on evolutionary programming. The novel idea in Fogel (1990) is an objective function, which minimize the difference between the route lengths of each salesman. Another recent heuristic algorithm is proposed in Nishanth Chandran and Ganesh (2006), where a clustering approach is tested in order to achieve a good balance of workloads among the clusters.

The Artificial Neural Network (ANN) is other heuristic approach quite used to solve the mTSPs. In Somhoom et al. (1999) and Modares et al. (1999) an ANN is presented with minmax objective, Hsu et al. (1991) also presents an ANN algorithm based on solving m standard TSPs, showing superior results to that of Wacholder et al. (1989). A self-organizing neural network approach is considered in Vakhutinsky and Golden (1994), which is based on the elastic net approach created for the TSP. In that...
sense, Torki et al. (1997) describes a self-organizing neural network for the Vehicle Routing Problem (VRP) based on an enhanced neural network model for mTSP.

Genetic Algorithms (GAs) have been widely used for combinatorial optimization, such as TSP (Zhang et al., 2010). However, its applications to the mTSP have been limited. One of the first Genetic Algorithm (GA) for solving the mTSP is reported in Zhang et al. (1999). Some applications are presented by Yu et al. (2002) and Tang et al., 2000, using GA to solve real-life problems, path planning and hot rolling scheduling respectively. Carter and Ragsdale (2006) present other successful genetic approach, including a new GA chromosome and related operators for the mTSP.

Tabu Search (TS) is another meta-heuristic approach to solve the mTSP. In Ryan et al. (1998) one solution for mTSP is reported with time windows using TS algorithm, the problem is solved using a reactive TS algorithm within a discrete event simulation framework.

In the last decade Song et al. (2003) proposed an extended Simulated Annealing (SA) heuristic for the mTSP, associating a fixed cost for each salesman. A variety of evolutionary algorithms for the mTSP are compared in Sofge et al. (2002), using a neighborhood optimization, Particle Swarm Optimization (PSO), Monte-Carlo optimization, GA and evolutionary strategies.

Finally we would like to refer to approaches which transform mTSP into standard TSP, thus any algorithm proposed for the latter can be used to the former. One of the well-know transformation for the single depot mTSP is due to Tang et al. (2000), this transformation is called the adding virtual city method. This method suggests adding a virtual city for each salesman, where infinite cost is assigned to virtual-to-virtual distances and zero cost is assigned between virtual cities and the other cities. In Gorenstein (1970) a similar method is applied using (m-1) virtual cities. Also Russell (1977) shows a solution procedure based on transforming the problem to a single TSP on an expanded graph. In Bellmore and Hong (1974) the asymmetric mTSP with m salesmen and n nodes can be converted into standard asymmetric TSP using (n+m-1) nodes, the fixed cost (C) for each salesman is also considered in this transformation. A similar transformation is described by Hong and Padberg (1977), where symmetric mTSP is transformed into a standard symmetric TSP with (n+m-1).

The algorithms based on ACO have been successfully applied to TSP (Dorigo and Gambardella, 1997a) and to several extensions of VRPs (Colorti and Roizzoli, 2007). However, these algorithms have not been explored very well for generalization of the TSP, such as mTSP. To our knowledge, the ACO meta-heuristic procedure proposed in Junjie and Dingwei (2006) and Valliavaara (2008) are the only reported studies that were applied to the mTSP (these are not related with our case study). The ACO metaheuristic was introduced by Dorigo and Gambardella (1997b) and Dorigo and Stützle (1999), such metaheuristic combines an adaptive memory with local heuristic function to repeatedly construct solutions of hard combinatorial problems.

This paper proposes a new approach (M-ACS) based on ACO, which falls under the umbrella of the meta-heuristic techniques. The algorithm uses multiple artificial ant colonies in order to solve the mTSP; each colony represents a set of possible global solutions. The colonies cooperate among themselves, sharing its experiences through “frequent” pheromone exchange. The algorithm’s performance is compared with the results of the efficient heuristic of Lin- Kernighan reported in Dazhi and Dingwei (2007), using benchmark problems form literature. Dazhi and Dingwei (2007) proposes the Lin-Kernighan heuristic based on the transformation described in Tang et al. (2000), for that reason we defined M-ACS according to this transformation as well.

The paper is structured as follows: In Section 2 the multiple travelling salesman problems is formulated, the description of the meta-heuristic and the algorithm steps are defined in Section 3. Computational results and the algorithm performance can be found in Section 4. Conclusions and future research and are outlined in Section 5.
2. Mathematical formulation of mTSP

The mTSP can be defined on a graph \( G = (V, A) \), where \( V \) is the set of \( n \) nodes (vertices) and \( A \) is the set of arcs (edges). Let \( D = (d_{ij}) \) be a cost (typically distance) matrix associated with \( A \). The matrix \( C \) is said to be symmetric when \( d_{ij} = d_{ji}, \forall (i, j) \in A \) and asymmetric otherwise. The aim of this discrete combinatorial problem is to find \( m \) routes (one for each salesman), which start and end in a same node (depot). Each salesman has to visit a node once and a node can be visited by just one salesman.

Several integer programming formulation have been proposed for the mTSP in literature, the most commonly used one is the assignment-based integer programming formulation. In this formulation the mTSP is usually formulated using an assignment-based double-index integer linear programming formulation. The decision variable can be defined as follows:

\[
x_{ij} = \begin{cases} 1 & \text{if arc } (i,j) \text{ is used on a route,} \\ 0 & \text{otherwise.} \end{cases}
\]

The general formulation of assignment-based integer programming of the mTSP can be given as follows:

\[
\text{minimize } \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} \cdot x_{ij} \quad (1)
\]

s.t.

\[
\sum_{j=2}^{n} x_{1j} = m, \quad (2)
\]

\[
\sum_{j=2}^{n} x_{j1} = m, \quad (3)
\]

\[
\sum_{i=1}^{n} x_{ij} = 1, \quad j = 2, ..., n, \quad (4)
\]

\[
\sum_{i=1}^{n} x_{ij} = 1, \quad i = 2, ..., n, \quad (5)
\]

\[
\sum_{s \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \quad \forall S \subseteq V \setminus \{1\}, S \neq \emptyset, \quad (6)
\]

\[
x_{ij} \in \{0,1\}, \forall (i, j) \in A \quad (7)
\]

Formula (1) describes the fact that the objective of the problem is the minimization of the sum of the associated costs (distance) for each arc \((i, j)\). The constraints (2) and (3) ensure that exactly \( m \) salesmen depart form and return back to node 1 (the depot). Expressions (4) and (5) represent the classical assignment constraints. Finally, constraints (6) are used to prevent subtour-s (SECs).
3. Multi-type Ant Colony System

The Multi-type Ant Colony System (M-ACS) proposed in this paper is based on the following idea: let be \( C \) a set of colonies, representing each of them a set of global solutions of the problem (mTSP). Each colony obtains a set of global solutions (each ant of the colony represents a solution to full mTSP) using an Ant Colony System (ACS) algorithm and during the route construction the different colonies cooperate, sharing experience through “frequent” pheromone exchange. However the different types of ants are also involved in a competition process, which is due to the fact that the ants are repulsed by the pheromone of ants that belong to other colony (other type of ants). Combining both mechanism (collaboration as well as competition), a set of global solutions can be reached for all colonies (better exploration process as a main advantage), selecting the best solution after the last iteration. It is important to note that the multi-type approach differs from the one proposed in Nowé et al. (2004), where each type builds a part of the solution and the different parts were disjoint. A typical application is finding a set of disjoint paths in a graph.

3.1 Construction of the salesmen routes

To solve the mTSP, the artificial ants construct solutions by successively choosing cities to visit, until each city has been visited. For the selection of a (not yet visited) city three aspects are taken into account in our algorithm: how good was the choice of the city before (\( \tau_{ij} \) pheromone trails), how promising is the choice of that city (\( \eta_{ij} \), measure of desirability), and how good was the choice of that city for the other colonies (\( \phi_{ij} \), colony pheromone trail). In the original ACS given by Dorigo and Gambardella (1997), each ant moves from present node \( i \) to the next node \( v \) using a pseudorandom rule, which only considers the first two aspects previously mentioned. The approach that we propose considers the three aspects as follows:

\[
\begin{align*}
   v &= \begin{cases} 
   \arg \max_{j \in U} (\tau_{ij})^a \cdot (\eta_{ij})^\beta \cdot (\phi_{ij})^\gamma & \text{if } q \leq q_0 \\
   \text{random} & \text{otherwise}
   \end{cases} \\
   V \cdot P_{ij}^k &= \frac{(\tau_{ij})^a \cdot (\eta_{ij})^\beta \cdot (\phi_{ij})^\gamma}{\sum_{j \in U}(\tau_{ij})^a \cdot (\eta_{ij})^\beta \cdot (\phi_{ij})^\gamma} 
\end{align*}
\]

where \( U \) represents a set of nodes which are not visited yet, \( \tau_{ij} \) pheromone trail of edge \((i, j)\), \( \eta_{ij} \) denotes the measure of desirability and can be calculated by the follows expression:

\[
\eta_{ij} = \frac{1}{d_{ij}}
\]

where \( d_{ij} \) denotes the distance between nodes \( i \) and \( j \). Moreover, \( \phi_{ij} \) indicates the average value of pheromone taken from the other colonies, after some iteration (\( F \), in the arc \((i, j)\). The parameters, \( \alpha \) and \( \beta \) determines the relative influence of pheromone versus distance (\( \alpha, \beta > 0 \)). Another parameter defined in the algorithm is \( \gamma \), which denotes the sensibility of each ant for using its own colony experience (\( \gamma = 0 \)) or also the experience of the remaining colonies (\( \gamma > 0 \)). On the other hand, \( q \) is a random number following uniform distribution in [0, 1], and \( q_0 \) is a parameter previously defined (0 \( \leq q_0 \leq 1 \)), which determines the relative importance of exploitation (formula, 8) versus exploration.
(formula, 9). If \( q \leq q_0 \) then the next node is selected according to expression (8), else, the next node is chosen according to the expression (9).

### 3.2 Pheromone update

The pheromone updating of M-ACS includes the same rules (local and global updating rules) of ACS, given by Dorigo and Gambardella (1997). According to Dorigo and Gambardella (1997), local updating rule, see expression (11), is applied to change pheromone level of edges while building a solution.

\[
\tau_{ij}^{\text{new}} = (1 - \rho)\tau_{ij}^{\text{old}} + \rho \tau_0
\]

where \( \rho \) is defined as evaporation coefficient (with \( 0 \leq \rho \leq 1 \)), thus the tail evaporation is given by \( (1 - \rho) \). The initial level of pheromone (\( \tau_0 \)) is produced by the well-know Nearest Neighbor heuristic (NN) and is defined as follows:

\[
\tau_0 = (n \cdot L_{nn})^{-1}
\]

where \( n \) is number of cities and the variable \( L_{nn} \) represents the total traveled distance applying NN heuristic. After the first algorithm iteration, the global updating rule is applied to those arcs that conform the best tour of the fist iteration. The rule is described as follows:

\[
\tau_{ij}^{\text{new}} = (1 - \rho)\tau_{ij}^{\text{old}} + \frac{\rho}{L_{\text{Best}}} \forall(i,j) \in \text{BestSol}
\]

where \( L_{\text{Best}} \) is total travel distance of the so far best solution \( \text{BestSol} \).

### 3.3 Cooperation between colonies

The proposed algorithm (M-ACS) presents significant features of swarm intelligence, contrary to the classical ACS, in the M-ACS a set of colonies cooperate in order to provide a better solution. The cooperation process, inspired by Nowé et al. (2004), consists on the exchange of pheromone trails reached by the ants that belong to each colony. Each colony deals with two matrices of pheromone trails: the first one contains the pheromone trail of its own ants, and the second matrix denotes the pheromone trails reached by the ants of remaining colonies.

The frequent pheromone exchange is performed after a number of iteration \( F \), where \( F \) is a user-defined parameter and can be established dividing the total number of iteration \( N \) in equal amount or as the user decides. Finally, the frequent pheromone exchange can be calculated as follows:

\[
\phi_s(i,j) = \frac{\sum_{c \in C \setminus s} \phi_c(i,j)}{C - 1}
\]

where index \( s \) indicates the current colony, which performs the pheromone update, taking the average pheromone values of the others colonies, excluding its own pheromone trail. Finally, in the Pseudo-code 1, the general procedure of M-ACS is presented; the procedure includes also the pheromone exchange mechanism between all colonies and is called the \textbf{new-ant-solution} procedure in the pseudo-code.
The former procedure is described in Pseudo-code 2 as follows:

**Pseudo-code 1: Procedure M-ACS**

Initialize parameters
Obtain the initial solution ($\psi^m$) by NN heuristic
\[ \psi^0 \leftarrow \psi^m \]
\[ L_{gb} \leftarrow L_{nn} \]
For each Colony $s$
   For each arc $(i, j)$
      \[ \phi_{ij} = \tau_{ij} = \tau_0 \] (Equation 12)
   EndFor
EndFor
Do Until $IT = N$
   If $IT \% N = F$ Then
      Pheromone exchange between all colonies (Equation 14)
   EndIf
   For each colony $c$
      For each ant $k$
         Build a solution $\psi^k$ using (new-ant-solution)
         If $L_k \leq L_{gb}$ Then
            \[ L_{gb} \leftarrow L_k \]
            \[ \psi_{gb} \leftarrow \psi^k \]
         EndIf
      EndFor
   EndFor
   $IT = IT + 1$
EndLoop

**Pseudo-code 2: Procedure new-ant-solution**

Initialize parameter
Locate ant $k$ in depot
Initialize traveled distance: $L^k \leftarrow 0$
While (Ant $k$ has not completed its solution)
   Select next node $j$ using expression (8) or (9)
   Update the trail level $\tau_{ij}$ (Equation 11)
   Update the tour: $\psi^k \leftarrow \psi^k + (j)$
   Update traveled distance: $L^k \leftarrow L^k + d_{ij}$
EndWhile
3.4 Complexity analysis

The time complexity of ACO algorithms is mainly based on its search strategies, where a set of m ants develop a tour construction with complexity $O(n^2)$ until a number of iterations is reached. The pheromone trails are stored in a matrix with $O(n^2)$ entries (one for each arc) as in all ACO strategies (Dorigo and Stützle, 1999). In M-ACS a set of C colonies is defined, each colony represents a subgroup of the total number of ants m. In the computational analysis this total number of ants is the important parameter and not the number of colonies. This is because the pheromone exchange between the colonies, which only is performed every 10% of the iterations, takes $O(n^2)$ as well and therefore does not increase the complexity of the standard pheromone updates within each colony. Yielding an overall time complexity of $O(n^2)$.

The Lin-Kernighan algorithm has a worst-time computational complexity of $O(n^3)$ for the TSP (or mTSP transformed) (Helsgaun, 2009). Thus, the worst-time complexity of proposed algorithm proves to be competitive in term of computational time compared with the efficient Lin-Kernighan heuristic.

4. Computational results

Some computational experiences are presented in this section in order to evaluate the performance of the new approach described in Section 3. Algorithm runs have been carried out on a personal computer equipped with an Intel Pentium processor 1.6 GHz and 1 GB of ram memory. The Multi-type ACS has been coded Java 1.6.0.

The M-ACS was tested on six benchmark problems described in TSPLIB (40). These problems have been originally solved with several approaches for the classical TSPs. Furthermore, we compare the M-ACS performance, using the mentioned datasets, with Lin-Kernighan heuristic reported in Dazhi and Dingwei (2007). The mentioned instances range from 124 to 783 cities and the number of salesman used is 3, 5 and 7 respectively. The Table 2 summarizes the benchmark problem information, where the first indicates that last three problems are asymmetric TSPs. Columns 2-3 show the problem codes and the scale respectively. The other columns show the function objective values of Lin-Kernighan heuristic and the M-ACS (average of 10 runs) for all used salesmen.

<table>
<thead>
<tr>
<th>Type</th>
<th>Codes</th>
<th>Scale</th>
<th>M-ACS M = 3</th>
<th>M-ACS M = 5</th>
<th>M-ACS M = 7</th>
<th>Lin-Kernighan heuristic M = 3</th>
<th>Lin-Kernighan heuristic M = 5</th>
<th>Lin-Kernighan heuristic M = 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symmetric</td>
<td>bier127</td>
<td>127</td>
<td>95934</td>
<td>87915</td>
<td>80345</td>
<td>95592</td>
<td>87562</td>
<td>80283</td>
</tr>
<tr>
<td></td>
<td>ts225</td>
<td>225</td>
<td>117452</td>
<td>113570</td>
<td>110551</td>
<td>117960</td>
<td>113562</td>
<td>110656</td>
</tr>
<tr>
<td></td>
<td>rat783</td>
<td>783</td>
<td>8668</td>
<td>8626</td>
<td>8533</td>
<td>8708</td>
<td>8650</td>
<td>8597</td>
</tr>
<tr>
<td>Asymmetric</td>
<td>kro124p</td>
<td>100</td>
<td>33765</td>
<td>32271</td>
<td>30907</td>
<td>33655</td>
<td>32247</td>
<td>30915</td>
</tr>
<tr>
<td></td>
<td>ftv170</td>
<td>171</td>
<td>2482</td>
<td>2341</td>
<td>2263</td>
<td>2498</td>
<td>2368</td>
<td>2272</td>
</tr>
<tr>
<td></td>
<td>rgb443</td>
<td>443</td>
<td>2604</td>
<td>2541</td>
<td>2466</td>
<td>2621</td>
<td>2555</td>
<td>2489</td>
</tr>
</tbody>
</table>

The parameter setting for M-ACS is the following one: $q_0 = 0.8$, $\alpha = \beta = \gamma = 1$, $\rho = 0.1$, a small number of ants for each colony, e.g. 10 ants. From a previous statistical study we define 3 colonies, and every 10 iterations, 10% of the total number of iteration (100), the pheromone exchange is carried out for all benchmark problems reported in Table 2.

Starting from figures of Table 2, we obtained significant differences between the results achieved by M-ACS and Lin-Kernighan heuristic results. The significant differences were ensured by mean of Wilconxon coefficient as non-parametric statistical test. Furthermore, it is important to observe that the
Lin-Kernighan heuristic provide better results when the problem scale is smaller. Moreover, the practical computational time of the approach that we propose was quite small for 100 iterations.

5. Conclusions

In this paper, the Multi-type Ant Colony System is presented as ACO strategy. The novel idea consists on the frequent pheromone exchange between the ants that belong to each colony. The new approach has been used to solve the mTSP, which has been poorly treated in literature, specifically by ACO approaches. Moreover, based on the benchmark problems, the computational results confirm that performance of our algorithm is very efficient, providing better solution quality than the well-know efficient Lin-Kernighan heuristic. In addition, the practical computational time of the M-ACS is reasonable for the benchmark problems. A case study, set up in the city of Santa Clara, confirms that our solution approach can be applied to real world instances. Thus, the proposed algorithm supports the decision making process related with the route planning to repair the electrical breakdown after natural disasters.

Future research

Future researches would focus to combine Multi-type Ant Colony System with Local Search strategy (M-ACS-LS) in order to improve the solution quality. Furthermore, analyze the performance of M-ACS-LS in others benchmark problems, such as the VRPs. Beside we have to consider a sensitivity analysis of the fixed parameters in the M-ACS based on an exhaustive statistical study.

References


[40] The symmetric and asymmetric TSPLIB, Traveling Salesman Problem Instances: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/


