Hybrid ant models with a transition policy for solving a complex problem

CAMELIA-M. PINTEA, George Coșbuc N. College, 400083 Cluj-Napoca, Romania.
E-mail: cmpintea@yahoo.com

GLORIA CERASELA CRIŞAN, Department of Mathematics and Computer Science, Vasile Alecsandri University, 600115 Bacău, Romania.
E-mail: ceraselacrisan@ub.ro

CAMELIA CHIRA, Department of Computer Science, Babeș-Bolyai University, 400084 Cluj-Napoca, Romania.
E-mail: cchira@cs.ubbcluj.ro

Abstract

Combinatorial problems arising in diverse application domains and the growing complexity of many real-world situations emphasize the need for efficient solving methods. One such problem is the bandwidth minimization problem having broad applications in engineering, science, logistics or information recovery. This well-known \( \text{NP} \)-complete problem refers to the minimization of the bandwidth for non-zero entries of a sparse symmetric matrix by permuting its rows and columns. A new ant-based approach to matrix bandwidth minimization is proposed. The introduced model is based on the hybridization of the Ant Colony System technique with new problem-tailored local search mechanisms. Computational experiments show a good behaviour of the proposed method for the considered set of bandwidth minimization problem instances.

Keywords: Matrix bandwidth minimization problem, ant colony optimization, local search, hybrid models.

1 Introduction

Intensively studied for more than 50 years [10], Combinatorial Optimization Problems (COPs) develop in many and diverse areas including network design, storage and retrieval, sequencing and scheduling, algebra theory, automata and language theory, program optimization and game theory. Many of these COPs are \( \text{NP} \)-hard and cannot be solved within polynomial computation times. Therefore, metaheuristics are often proposed as powerful strategies that can efficiently detect high-quality (near optimal) solutions to these complex problems using reasonable resources.

The formalization of the current real-life problems raises many difficulties in designing solving methods: parameter connections, large number of constraints, high dimension of the solution space, high density of local optimal solutions and data uncertainty or dynamicity. The continuous growing complexity of these problems leads to a permanent search for new frameworks able to address them and to efficiently provide high-quality solutions.

The problem of bandwidth minimization is a well-known \( \text{NP} \)-complete optimization problem [22] raising a high interest in applied mathematics and having broad applications that involve matrix manipulations in fields such as engineering, physics, computer science and economics. When solving extremely large systems of linear equations, a good reordering of rows and columns can tremendously improve the solving time. The Matrix Bandwidth Minimization Problem (MBMP) considers
Hybrid ant models

A symmetric matrix and seeks for a permutation of its rows and columns such that the non-zero elements are located as close as possible to the main diagonal. The bandwidth of a square symmetric matrix $A$ of order $n$ is the value $\beta = \max_{a_{ij} \neq 0} |i - j|$ representing the distance from the main diagonal beyond which all elements are zero. To solve the MBMP for $A$ means to find a permutation $\pi \in S_n$ (where $S_n$ is the group of all permutations of order $n$) of its rows and columns that minimizes the bandwidth of the transformed matrix.

As many other COPs, the MBMP can be naturally described in terms of graph theory. The square symmetric matrix $A$ of order $n$ can be seen as the adjacency matrix of a weighted undirected graph $G_A = (V, E)$ with $V = \{1, 2, \ldots, n\}$ and $E = \{(i, j) \mid a_{ij} \neq 0\}$. This leads to the definition of the bandwidth for the corresponding graph $G_A$: $\beta(G_A) = \max_{(i, j) \in E} |i - j|$. To solve the transformed problem means to find a permutation $\pi$ of $V$ that minimizes the graph bandwidth. It has been shown that the graph-equivalent form of the MBMP is $NP$-complete [22].

This article presents a new approach to heuristically address the MBMP based on the hybridization of the Ant Colony System (ACS) technique [7] with new local search (problem-tailored) mechanisms. As practical experience suggests and many relevant studies show [1, 4, 5, 14, 26], the hybridization of several computational methods exhibits emergent, catalytic behaviour: the compound method provides better results than each component does.

ACS is an ant colony optimization model [7] replicating the behaviour of social insects to the search space, commonly described using a graph representation. Each edge has an associated cost as well as a pheromone value corresponding to a desirability measure. In the ACS model, each ant generates a complete tour (associated to a problem solution) by probabilistically choosing the next node at each path intersection based on the cost and the amount of pheromone on the connecting edge (according to the state transition rule). Stronger pheromone trails are preferred by ants and the most promising tours build up higher amounts of pheromone in time. The amount of pheromone is modified by each ant while building the tour using a local pheromone update rule. Furthermore, a global update rule is applied to the edges belonging to the best tour determined after all ants have completed their tours [7]. The ACS technique has been successfully engaged for solving various COPs such as travelling salesman, quadratic assignment, scheduling, vehicle routing and protein folding.

In the proposed hybrid ant-based model for solving the MBMP, two procedures aimed at reducing the bandwidth are designed and used for solution improvement during a local search stage. The skeleton of the resulting hybrid model is used to develop further hybrid ant algorithms based on the T-ACS model presented in [25]. All introduced algorithms are engaged in a set of numerical experiments for solving several MBMP instances. Comparisons with the performance of the basic ACS technique are discussed (based on a detailed statistical analysis) and the obtained results are highly encouraging.

The structure of the article is as follows: Section 2 presents a brief overview of the related work; the proposed algorithms are described in Section 3; computational experiments and numerical results are discussed in Section 4, and the last section presents some conclusions and describes the future development of our work.

2 Related work

The MBMP opens several promising and interesting theoretic investigations. One direction refers to the investigation of the bandwidth recognition: what is the structure of a graph that enables the usage of fast algorithms in order to decide if its bandwidth has a specific value. For example,
Hybrid ant models

the bandwidth 2 is investigated in [3]. Another interesting research topic is the two-dimensional bandwidth problem: how to embed a graph into a planar grid such that the maximum distance between adjacent vertices is as small as possible [18].

Some specific structures for a graph enable polynomial time algorithms for bandwidth finding [13]. For general graphs, approximation algorithms run in polylogarithmic time [2]. For some particular graph structures, there are different approximation algorithms. For example, a 3-approximation one for dense graphs is presented in [12], or an \(O(\log n)\)-approximation algorithm for caterpillars is given in [11]. A very efficient combined approach, starting from GRASP and based on Branch and Bound is presented in [19].

Most heuristic algorithms designed for MBMP are level based. The quest is to find a level partition with the width as small as possible and the depth as large as possible. A level structure for a graph is a partition on its vertices into some equivalence classes \(L_1, L_2, \ldots, L_k\), such that:

- all vertices adjacent to vertices in level \(L_1\) are in either level \(L_1\) or \(L_2\),
- all vertices adjacent to vertices in level \(L_k\) are in either level \(L_{k-1}\) or \(L_k\),
- for \(1 < i < k\), all vertices adjacent to vertices in level \(L_i\) are in either level \(L_{i-1}\), \(L_i\) or \(L_{i+1}\).

One of the first and simplest heuristic methods for MBMP is the Cuthill–McKee algorithm [6] which builds a permutation of the vertices by making a list that starts from a vertex with minimum degree and appends all the vertices that are adjacent to those already in list, in ascending degree order. The procedure stops when the list is completed, meaning that all \(n\) positions are filled.

Some important approaches to tackle the MBMP include simulated annealing [24], GRASP [23], Tabu Search [20] and a node-shift heuristic [17]. Starting from the MBMP, new problems such as the anti-bandwidth problem [15] have been defined.

In [16], an ant colony optimization (ACO) approach to MBMP hybridized with hill climbing is presented. Artificial ants are activated and coordinated by a queen that also manages the common memory. The proposed algorithm (ACO-HC) is hybridized with a hill-climbing local-search procedure, added to the end of each ant process, just before the ant’s solution is sent to the queen process. The idea of the hill-climbing procedure—to swap \(\pi(v)\) with \(\pi(u)\) only if the bandwidth can be reduced—is inspired by the Tabu Search method from [20]. At the end of any iteration, the queen updates the memory trail with the current global best solution, or the iteration best solution.

As opposed to the ACO-HC model, the current article proposes a natural ant-based approach to MBMP based on the highly effective Cuthill–McKee algorithm [6]. Furthermore, the local search procedures engaged in the proposed models aim to offer a less computationally expensive search when compared with hill climbing.

3 Proposed hybrid ant-based models for solving the MBMP

The proposed approaches to address the MBMP are hybridizations of the ACS technique [7] with two local search procedures (resulting in two different hybrid ant algorithms) using the Cuthill–McKee algorithm [6] as a starting point. Furthermore, an ACS variant called T-ACS [25] is studied in conjunction with the same local search procedures. The T-ACS model enhances ACS with features of reinforcement learning by considering a state transition policy based on the count of the visiting times for the available transition nodes. By counting from the beginning of the algorithm, for each edge, the visiting times and introducing these values into the formula (2) of the probability distribution used by each ant to construct its path, the T-ACS not only accelerates the rate of exploration, but also
Hybrid ant models

improves the accuracy of exploration gradually [25]. A solution is an array storing the permutation of order \( n \).

The ACS framework used for the MBMP is based on the level structure described in [8] for the Cuthill–McKee algorithm [6] and consists of the following steps:

(I) the initial phase computes the current matrix bandwidth and sets the parameter values;

(II) the construction phase starts by placing all the ants in nodes from the first level, and repeatedly making pseudo-randomly choices from the available neighbours. After each step, the local update rule (3) is applied. This second phase ends with the global pheromone update rule (4);

(III) the final phase consists of writing the best solution.

These three phases are iteratively executed within a given number of iterations \( (N_C) \). The computational complexity of all ACS variants presented here is \( O(n^4 \cdot m \cdot N_C) \), with \( m \) being the number of artificial ants.

The pseudocode of the proposed hybrid ant-based model is given below:

**Hybrid ACS model for solving MBMP**

begin

I. Initialization: computes the current bandwidth; initialize pheromone trails; sets the parameters values;

II. while (maximum number of iterations not reached) do

Swap Procedure

while (maximum number of ants not reached) do

build a partial solution using ACS based on (1), (2)

apply local pheromone update rule (3)

Swap Procedure

apply global pheromone update rule (4)

end while

end while

III. write the best solution

End

The integration of a local search phase within this ACS approach to MBMP is expected to facilitate the refinement of ants’ solutions. A local search mechanism (called Swap procedure) is engaged twice within the proposed model: first, at the beginning of any iteration and secondly, after each partial solution is built. These insertion points of the local search procedure are different from the ACO-HC model [16]. As our Swap module needs only to compute the min and the max of the nodes degrees, and as our problem has a very specific characteristic—generally there are multiple permutations that lead to the same bandwidth [16], we decided to continually use it during the construction phase. Even if we expect that some ants find bad solutions, we are focused on the good solutions found by others—with the help of repeatedly applied Swap procedures. This opportunistic approach can be compared with that exploited in [20]: as generally the elite solution set from Tabu Search do not dramatically change from one iterations to another, it is not necessary to update it after every move.

All trails are initialized with the same pheromone quantity denoted by \( \tau_0 \). Each ant \( k \) moves from a node \( i \) to a node \( j \) selected as follows:

\[
 j = \begin{cases} 
 \arg\max_{u \in I(k)} \left\{ \left( \tau(i,u)^{\alpha} \cdot [\delta(i,u)]^{\beta} \right) \right\} & \text{if } q \leq q_0 \\
 \text{randomly chosen using } S & \text{otherwise}
\end{cases}
\]  

(1)
where $J_k(i)$ is the set of available moves for ant $k$ from node $i$, $q$ is a random value from the $(0, 1)$ interval, $\delta(i, u)$ is the distance between $i$ and $u$, $\alpha$, $\beta$, and $q_0$ are parameters and $S$ is the probability distribution expressing the random-proportional rule [7]:

$$p_k(i, s) = \begin{cases} \frac{\tau(i, s)^\alpha \delta(i, s)^{-\beta}}{\sum_{u \in J_k(i)} \tau(i, u)^\alpha \delta(i, u)^{-\beta}} & \text{if } s \in J_k(i) \\ 0 & \text{otherwise} \end{cases}$$ (2)

After each transition from vertex $i$ to vertex $j$, the trails $\tau_{ij}$ are modified using the ACS local update rule:

$$\tau_{ij} = (1 - \rho) \tau_{ij} + \rho \tau_0$$ (3)

The ACS global update pheromone rule is as follows:

$$\tau_{ij} = (1 - \rho) \tau_{ij} + \rho \Delta \tau_{ij}$$ (4)

The evaporation rate is denoted by $\rho$ (with a value between 0 and 1) and $\Delta \tau_{ij}$ is the inverse of the cost-value of the current best-known all-time solution [7].

As a local search mechanism, we tested two swapping procedures called $PSwap$ and $MPSwap$ (both given below). The first procedure $PSwap$ swaps all the vertices with the highest degree with randomly selected vertices with lowest degree. The $MPSwap$ local search mechanism modifies the $PSwap$ procedure in order to avoid stagnation by selecting the swapping nodes with lowest degree such that the bandwidth is smaller.

**PSwap Procedure**

find the maximum and minimum degrees

for all indices $x$ with the maximum degree

randomly select $y$, an unvisited node with a minimum degree

$SWAP(x, y)$

end for

**MPSwap Procedure**

find the maximum and minimum degrees

for all indices $x$ with the maximum degree

select $y$, an unvisited node with the smallest degree such that the matrix bandwidth decreases

$SWAP(x, y)$

end for

The resulting algorithm based on ACS hybridized with the $PSwap$ procedure is called $hACS$ in this article, while the hybridization with the second proposed local search procedure $MPSwap$ is called $hMACS$.

Furthermore, we propose to investigate the performance of two other hybrid ant models based on the $T$-ACS algorithm [25] combined with the same local search procedures described above ($PSwap$ and $MPSwap$). The resulting algorithms are called $hT$-$ACS$ and $hT$-$MACS$, respectively. In order to enforce the exploration capabilities of artificial ants, the formula (2) is modified by using $\lambda(i, j)$—
number of times the available edge \((i, j)\) is visited since the start of the algorithm—instead of \(\alpha\):

\[
p_A(i, s) = \begin{cases} 
\frac{\tau(i, s)^\alpha \delta(i, s)^\beta}{\sum_{u \in J_k(i)} \tau(i, u)^\alpha \delta(i, u)^\beta} & \text{if } s \in J_k(i) \\
0 & \text{otherwise}
\end{cases}
\]  

(5)

During the early stages of the search process, acceptable solutions are quickly found; when the search is more advanced, (5) makes the ants choose less visited edges favouring the rapid discovery of new optimal solutions.

Basic ACS, a random approach Rand that always randomly selects the next move, and the four proposed approaches hACS, hMACS, hT-ACS and hT-MACS to the MBMP are further investigated in terms of solution quality and computation time.

4 Computational experiments and statistical analysis

All the algorithms were implemented in Java and ran on an AMD 2600 computer with 1024 MB memory and 1.9 GHz CPU clock. Nine symmetric Euclidean instances from National Institute of Standards and Technology, Matrix Market, Harwell-Boeing sparse matrix collection (MatrixMarket matrix coordinate pattern symmetric) [21] are used as benchmarks.

The parameter values for all ACS implementations are: 10 ants, 10 iterations, \(q_0 = 0.95\), \(\alpha = 1\), \(\beta = 2\), \(\rho = 0.0001\), \(\tau_0 = 0.1\). The first two values are chosen to meet the hardware restrictions, whereas the last five parameters have been given the values most often used for ACS implementations, proven to provide the best results.

Numerical results are presented in Table 1. The best value from 20 trials is reported for each of the compared algorithms: the random variant Rand, the standard ACS technique and the four proposed hybrid ant models (hACS, hMACS, hT-ACS and hT-MACS). As expected, all hybrid models obtain better results than both Rand and ACS, bearing no local search mechanism. If we compare the two local search procedures proposed (hACS versus hMACS and also hT-ACS versus hT-MACS) an overall better performance is observed for the MPSwap procedure (integrated in hMACS and hT-MACS algorithms). Furthermore, the hybrid models based on T-ACS (hT-ACS and hT-MACS) seem to produce better results compared with those based on ACS (hACS and hMACS) emphasizing the benefits of a balanced search process.

<table>
<thead>
<tr>
<th>No.</th>
<th>Instance</th>
<th>Rand</th>
<th>ACS</th>
<th>hACS</th>
<th>hT-ACS</th>
<th>hMACS</th>
<th>hT-MACS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>can__24</td>
<td>19</td>
<td>17</td>
<td>14</td>
<td>12</td>
<td>11</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>can__61</td>
<td>50</td>
<td>47</td>
<td>43</td>
<td>42</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>3</td>
<td>can__62</td>
<td>46</td>
<td>39</td>
<td>20</td>
<td>12</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>can__73</td>
<td>39</td>
<td>37</td>
<td>28</td>
<td>23</td>
<td>22</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>can__96</td>
<td>31</td>
<td>31</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>can__187</td>
<td>63</td>
<td>63</td>
<td>63</td>
<td>33</td>
<td>33</td>
<td>33</td>
</tr>
<tr>
<td>7</td>
<td>can__229</td>
<td>172</td>
<td>163</td>
<td>120</td>
<td>126</td>
<td>120</td>
<td>109</td>
</tr>
<tr>
<td>8</td>
<td>can__256</td>
<td>237</td>
<td>241</td>
<td>148</td>
<td>197</td>
<td>189</td>
<td>178</td>
</tr>
<tr>
<td>9</td>
<td>can__268</td>
<td>249</td>
<td>234</td>
<td>165</td>
<td>211</td>
<td>210</td>
<td>202</td>
</tr>
</tbody>
</table>

Best values found are given in bold face type.
### Hybrid ant models

**Table 2. Paired Student’s t-test results for \( hT-ACS \) and \( hT-MACS \)**

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Means (95% CI)</th>
<th>Standard deviation</th>
<th>Median</th>
<th>Avg. absolute deviation from median</th>
</tr>
</thead>
<tbody>
<tr>
<td>( hT-ACS )</td>
<td>74.8 [12.31, 137.2]</td>
<td>81.3</td>
<td>33</td>
<td>56.9</td>
</tr>
<tr>
<td>( hT-MACS )</td>
<td>68.6 [10.43, 126.7]</td>
<td>75.6</td>
<td>33</td>
<td>53.1</td>
</tr>
<tr>
<td>( hT-ACS ) vs ( hT-MACS )</td>
<td>6.22 [0.5766, 11.87]</td>
<td>7.34</td>
<td>3</td>
<td>5.44</td>
</tr>
</tbody>
</table>

**Table 3. Statistical analysis results for \( Rand \), \( hACS \), \( hT-ACS \), \( hMACS \) and \( hT-MACS \) on the nine problem instances considered**

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>( \bar{x} )</th>
<th>( s^2 )</th>
<th>( \bar{b} )</th>
<th>( \tau )</th>
<th>( \gamma - \beta(1 - \beta t)^{-\tau} )</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Rand )</td>
<td>1.1476</td>
<td>1.5537</td>
<td>1.3539</td>
<td>0.8476</td>
<td>393.8786</td>
<td>5</td>
</tr>
<tr>
<td>( hACS )</td>
<td>0.4186</td>
<td>0.2442</td>
<td>0.5835</td>
<td>0.7173</td>
<td>397.8533</td>
<td>4</td>
</tr>
<tr>
<td>( hT-ACS )</td>
<td>0.2351</td>
<td>0.1200</td>
<td>0.5103</td>
<td>0.4607</td>
<td>398.8022</td>
<td>3</td>
</tr>
<tr>
<td>( hMACS )</td>
<td>0.1982</td>
<td>0.1099</td>
<td>0.5544</td>
<td>0.3575</td>
<td>398.9898</td>
<td>2</td>
</tr>
<tr>
<td>( hT-MACS )</td>
<td>0.0474</td>
<td>0.1456</td>
<td>3.0701</td>
<td>0.0155</td>
<td>399.7422</td>
<td>1</td>
</tr>
</tbody>
</table>

The best solution obtained by each heuristic is compared with the overall minimal value obtained for each problem instance.

A paired t-test [27] is engaged to compare the \( hT-ACS \) and \( hT-MACS \) algorithms (based on \( T-ACS \)) which seem to have a better performance than those based on \( ACS \). The results of this statistical test are presented in Table 2.

The result of the paired t-test (Table 2) is \( t = 2.54 \). The probability of this result, assuming the null hypothesis, is for average value 0.035. The result is \(< 0.05\) indicating a significant statistical difference between the two considered algorithms.

A further statistical analysis is performed using the Expected Utility Approach [9] to analyse the comparative behaviour for all algorithms. Let \( x \) be the percentage difference of the heuristic solution from the best known solution of a particular heuristic (the minimum solution found by any method) on a given problem instance:

\[
x = \frac{\text{heuristic solution} - \text{best known solution}}{\text{best known solution}} \times 100\%
\]

The expected utility function used is \( \gamma - \beta(1 - \beta t)^{-\tau} \), where \( \gamma = 500, \ \beta = 100, \ t = 0.05 \) and \( \bar{b}, \tau \) are the estimated parameters of the Gamma function. The following notations are used for calculating the value of the expected utility function for each heuristic (based on the number of problem instances engaged in numerical experiments—which is nine):

\[
\bar{x} = \frac{1}{9} \sum_{j=1}^{9} x_j, \ s^2 = \frac{1}{9} \sum_{j=1}^{9} (x_j - \bar{x})^2, \ \bar{b} = \frac{s^2}{\bar{x}}, \ \hat{\tau} = \left( \frac{\bar{x}}{\bar{b}} \right)^2
\]

Table 3 presents the results of the statistical analysis based on the experiments given in Table 1. The statistical test confirms \( hT-MACS \) as the most accurate heuristic from the proposed models obtaining Rank 1 (see the last column in Table 3).

Furthermore, we compare \( Rand \) and the four hybrid ant-based algorithms in terms of execution time. The average running time of each algorithm is computed based on the 20 runs performed and the results are presented in Table 4.
Table 4. Average execution time (in seconds) for Rand, hACS, hT-ACS, hMACS and hT-MACS models

<table>
<thead>
<tr>
<th>No</th>
<th>Instance</th>
<th>Rand</th>
<th>hACS</th>
<th>hT-ACS</th>
<th>hMACS</th>
<th>hT-MACS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>can_24</td>
<td>0.62</td>
<td>0.65</td>
<td>0.55</td>
<td>0.48</td>
<td>0.60</td>
</tr>
<tr>
<td>2</td>
<td>can_61</td>
<td>1.23</td>
<td>1.17</td>
<td>1.22</td>
<td>1.13</td>
<td>1.18</td>
</tr>
<tr>
<td>3</td>
<td>can_62</td>
<td>0.79</td>
<td>0.90</td>
<td>0.74</td>
<td>0.82</td>
<td>0.90</td>
</tr>
<tr>
<td>4</td>
<td>can_73</td>
<td>1.00</td>
<td>1.13</td>
<td>1.15</td>
<td>1.13</td>
<td>1.30</td>
</tr>
<tr>
<td>5</td>
<td>can_96</td>
<td>1.25</td>
<td>1.33</td>
<td>1.56</td>
<td>1.59</td>
<td>1.68</td>
</tr>
<tr>
<td>6</td>
<td>can_187</td>
<td>1.65</td>
<td>3.22</td>
<td>4.52</td>
<td>6.87</td>
<td>5.20</td>
</tr>
<tr>
<td>7</td>
<td>can_229</td>
<td>2.68</td>
<td>9.67</td>
<td>11.21</td>
<td>14.73</td>
<td>9.90</td>
</tr>
<tr>
<td>8</td>
<td>can_256</td>
<td>5.68</td>
<td>22.45</td>
<td>34.48</td>
<td>40.33</td>
<td>26.05</td>
</tr>
<tr>
<td>9</td>
<td>can_268</td>
<td>4.88</td>
<td>22.31</td>
<td>30.21</td>
<td>34.28</td>
<td>21.90</td>
</tr>
</tbody>
</table>

The better performing MPSwap local search procedure causes a longer running time for the corresponding algorithms (hMACS and hT-MACS) compared with the hACS and hT-ACS which use the simpler PSwap mechanism. A similar performance in terms of execution time can be observed for hACS and hT-ACS, both of them using the same local search mechanism. Rand constantly manifests rapid executions but the corresponding solution quality is constantly low.

5 Conclusions and future work

Several hybrid ant colony algorithms based on the well-known Cuthill–McKee algorithm are developed to address the MBMP. Two main local search mechanisms are tested in conjunction with the ACS and T-ACS models. Numerical results for several benchmark instances are promising and indicate a better performance in terms of solution quality of the T-ACS technique hybridized with the MPSwap local search mechanism (able to avoid stagnation by selecting those swapping nodes with lowest degree that decrease the bandwidth value). However, the same local search mechanism requires longer running time compared with the PSwap procedure emphasizing the trade-off to be made between solution quality and execution time.

Further investigations should be made to study these algorithms on other available problem instances from [21] and to highlight any patterns in their behaviour or a quality bias. Moreover, new local search procedures such as those inspired by the node-shift heuristic [17] as well as the potential use of a concurrent implementation on a coarse-grained parallel architecture will be investigated.

Funding

This research is supported by Grant PN_II_TE 320, Emergence, auto-organization and evolution: New computational models in the study of complex systems, funded by CNCSIS, Romania.

References

Hybrid ant models


Received 25 October 2010