"Vasile Alecsandri" University of Bacău
Faculty of Sciences
Scientific Studies and Research
Series Mathematics and Informatics
Vol. 21 (2011), No. 1, 113-120

A HYBRID TECHNIQUE FOR A MATRIX BANDWIDTH PROBLEM

GLORIA CERASELA CRIŞAN and CAMELIA-M. PINTEA

Abstract. The Combinatorial Optimization Problems have today many complex real-life instances; even using extensive computing resources, their large dimensions and difficult constraints make the exact solving methods to be inappropriate. This is why heuristic methods are used in order to quickly obtain very good solutions. Here we propose a hybrid heuristic method for the Matrix Bandwidth Minimization Problem, based on an Ant Colony Optimization method and several local-search mechanisms. This well-known NP-complete problem refers to finding a permutation of the rows and columns of a sparse symmetric matrix in order to minimize its bandwidth for the nonzero entries. The Matrix Bandwidth Minimization Problem has broad applications in science, logistics or engineering.

1. INTRODUCTION

The exact solving methods for the complex real-world problems we are faced today become inappropriate, as they often need exhaustive search and therefore intensive use of computing resources. In industrial high-risk process management, for example, the temporal decisional restrictions often override the need of the optimum solution – so, a "close enough" but quickly obtained solution is by far better than the optimum but useless one. Heuristic methods, offering rapid suboptimum solutions, are under researchers' attention for more than fifty years, since the general concept of *satisficing solution* was for the first time defined in the Artificial Intelligence domain [12].

Keywords and phrases: Ant Colony Optimization (ACO), metaheuristics, algorithm performance
(2010) Mathematics Subject Classification: 68Q85, 68W10.

The Matrix Bandwidth Minimization Problem (MBMP) considers a symmetric matrix and seeks for a permutation of its rows and columns such that the non-zero elements lay as close as possible to the main diagonal. This is a well-known NP-complete Combinatorial Optimization Problem (COP), with high-interest applications in engineering, physics, computer science and economics.

This paper presents an Ant Colony Optimization (ACO) approach to solve the MBMP, hybridized with new local-search mechanisms. The ACO model is inspired by the real ant colony ability in finding short paths between the nest and the food location [3]. Today, many ACO variants are used in solving various static or dynamic COPs.

The structure of this paper is as follows: Section 2 presents the investigated problem, Section 3 describes the hybrid algorithm for solving it, the results are presented in Section 4 and we finally conclude in Section 5.

2 The Matrix Bandwidth Minimization Problem (MBMP)

The *bandwidth* of a square symmetric matrix *A* of order *n* is $\beta = \max_{a_{ij} \neq 0} |i - j|$.

To solve the *MBMP* for *A* means to find a permutation π of the rows (and columns) of the matrix *A*, that minimizes the bandwidth of the resulted matrix. As many COPs, the *MBMP* can be re-cast as a graph-theory problem. Starting from *A*, one can define the graph $G_A = (V, E)$ with $V = \{1, 2, ..., n\}$, and $E = \{(i, j) \ iff \ a_{ij} \neq 0\}$. The bandwidth of the graph G_A now can be stated as $\beta = \max_{(i,j) \in E} |i - j|$. In its graph-equivalent form, the *MBMP* is $\mathcal{NP}$-complete [9].

There are several *MBM* sub-problems that can be solved in polynomial time; an example of this rare situation is the chain graphs category [4]. For the general case, approximation algorithms run in polylogarithmic time [1].

Almost all heuristic approaches for *MBMP* exploits an appropriate level structure (a partition $L_1, L_2, ..., L_k$ of the vertex set satisfying some desirable conditions). The algorithm designer chooses the condition satisfied by the vertices from the same level. For example, the Cuthill-McKee method [2] uses the sorted list of vertices degree in ascending order for constructing a level structure. A variant of this old, effective and very simple method is used today for the MATLAB *symrcm* command [13].

Almost all metaheuristic approaches were used for MBMP; simulated annealing [11], *GRASP* [10], Tabu Search [7], or an innovative node-shift heuristic [6] constantly develop and expand the set of available algorithms. Ant Colony Optimization is a metaheuristic inspired by the behavior of the real ants: a set of artificial ants cooperate using an indirect communication mediated by pheromone they deposit while traversing a graph and building solutions. Ants prefer strong pheromone trails and the most promising tours accumulate higher amounts of pheromone.

Ant Colony System is an *ACO* heuristic that uses two pheromone updating rules, in order to favor exploration. At the beginning, *m* ants are positioned on graph vertices chosen according to some initialization rule (e.g., randomly). Each ant constructs a feasible solution by repeatedly applying a stochastic greedy rule (the state transition rule). While building its tour, the ant also modifies the amount of pheromone on the traversed edges by applying the local updating rule. When all ants have completed their tour, the amount of pheromone on edges is modified again (by applying the global updating rule). The local rule is applied by each ant during the solution construction; only the ant that finds the best tour is allowed to apply the global rule [3].

One hybrid *ACO* metaheuristic (with hill-climbing) approach for *MBMP* is presented in [5]. The ants are started and managed by a queen that also coordinates the common memory. The hill climbing method is added to the end of each ant job, before the ant's solution is sent to the queen process. After any iteration, the queen updates the memory trail with the current global best solution, or the iteration best solution.

In the following, it is presented an *Ant Colony System* (ACS) solving method for *MBMP*, hybridized with new local procedures, based on the efficient Cuthill-McKee method [2].

3 The *ACS-Swap* Solving Method

The hybridized *ACS-Swap* solving method for *MBMP* consists of the following three steps:

- computing the current matrix bandwidth and setting the parameters values;
- putting all the ants in the node from the first level, and repeatedly making pseudo-randomly choices from the available neighbors. After each step, the *Swap* procedure is applied, followed by the local update rule [3]. This second phase ends by the global pheromone update rule [3] and is iterated for *NC* times (*NC* is an algorithm parameter);
- writing the best solution.

The *Swap* procedure is a local search method based on the same idea as the Cuthill-McKee algorithm [2]: it swaps all the vertices with the highest degree with randomly selected ones with minimum degrees that really decrease the bandwidth (thus avoiding algorithm stagnation). The pseudocode for the Swap procedure follows.

```
Swap Procedure
find the maximum and minimum vertex degrees
for all indices x having the maximum degree
    select y, an unvisited node with a minimum degree
    such as the matrix bandwidth decreases
    interchange (x,y)
end for
```

Every single iteration from the phase two of the *ACS-Swap* presented here uses the *Swap* procedure. The reason is the specificity or our investigated problem: generally, there are many permutations that lead to the same bandwidth [5], so we expect many ants find not-so-good solutions, but we rely on the few ants that are "lucky enough" – as a result of repeatedly applied *Swap* method.

4 Results

The *ACS-Swap* was implemented in Java and executed on an AMD 2600 processor with 1024 MB memory and 1.9 GHz CPU clock. The parameter values for all ACS implementations are: 10 ants, 10 iterations, $\alpha = 1$, $\beta = 2$, q_0=0.5, $\rho = 0.0001$, $\tau_0 = 0.1$.

Nine symmetric Euclidean instances from Harwell-Boeing sparse matrix collection [8] were used as benchmarks (see Table 1). The results are presented in Table 2. The best value from 20 trials is presented both for the classic *ACS* and for the new *ACS-Swap* method. The average value is given in column *AVG* and the average execution time is depicted in the *AVGT* column. The best solution is reported in column *MIN*, and the number of times the best solution was reached is given in the column #.

Table 1. The benchmark instances from National Institute of Standards and Technology, Matrix Market, Harwell-Boeing sparse matrix collection *(MatrixMarket matrix coordinate pattern symmetric)*[8]

No	Instance	Euclidean Characteristics
1.	**can___24**	**24 24 92**
2.	**can___61**	**61 61 309**
3.	**can___62**	**62 62 140**
4.	**can___73**	**73 73 225**
5.	**can___96**	**96 96 432**
6.	**can__187**	**187 187 839**
7.	**can__229**	**229 229 1003**
8.	**can__256**	**256 256 1586**
9.	**can__268**	**268 268 1675**

As expected, the new approach performed better than the classic ACS method, using the same computing time on small instances and more resources on large instances.

Table 2. Experimental results with *ACS* and *ACS-Swap* on nine instances from Harwell-Boeing sparse matrix collection [8]

No	Instance	*ACS*				*ACS-Swap*			
		MIN	#	AVG	AVGT	MIN	#	AVG	AVGT
1.	**can___24**	17	2	18.85	1.78	**14**	1	17.1	0.56
2.	**can___61**	48	1	49.8	2.61	**47**	1	49.85	1.36
3.	**can___62**	39	2	43.8	3.73	**23**	2	30.8	0.82
4.	**can___73**	37	5	38.15	2.65	**35**	4	36.60	1.41
5.	**can___96**	**31**	20	31	6.75	**31**	20	31	2.33
6.	**can__187**	63	20	20	11.12	**62**	5	62.75	6.93
7.	**can__229**	165	1	169.70	10.04	**153**	1	263.20	22.85
8.	**can__256**	226	9	238.75	17.23	**231**	1	233.85	49.14
9.	**can__268**	234	1	243.45	15.58	**231**	1	242.65	38.25

5 Conclusions and Future Work

This paper presents a hybrid heuristic method for solving the Matrix Bandwidth Minimization Problem, based on Ant Colony Optimization and a problem-specific local search method. The numerical results obtained for nine benchmarks from [8] are encouraging, showing a better performance than the classic ACS method.

Future investigations will explore the more instances from [8] and will also add more problem-specific knowledge during the local-search phase. The benefits of the multi-threaded features of Java will be used for designing a concurrent application.

Bibliography

1. Blum, A., Konjevod, G., Ravi, R., Vempala, S.: **Semi-definite relaxations for minimum bandwith and other vertex-ordering problems**. In 30th ACM Symposium on Theory of Computing, 284–293 (1998)
2. Cuthill, E, McKee, J.: **Reducing the bandwidth of sparse symmetric matrices** In Proceedings of the 24th National Conference ACM, 157–172 (1969)
3. Dorigo, M., Stützle, T. - ***Ant Colony Optimization***, MIT Press (2004)
4. Kloks, T., Kratsch, D., Müller, H.: **Bandwidth of chain graphs**. Information Processing Letters, 68, 6, 313–315 (1998)
5. Lim, A., Lin, J., Rodrigues, B., Xiao, F.: **Ant Colony Optimization with hill climbing for the bandwidth minimization problem**. Applied Soft Computing 6, 2, 180–188 (2006)
6. Lim, A., Rodrigues, B., Xiao, F.: **Heuristics for matrix bandwidth reduction**. European Journal of Operational Research 174, 1, 69–91 (2006)
7. Martí, R., Laguna, M., Glover, F. Campos, V.: **Reducing the bandwidth of a sparse matrix with Tabu Search**, European Journal of Operational Research 135, 2, 211–220 (2001)
8. National Institute of Standards and Technology, Matrix Market, Harwell-Boeing sparse matrix collection, http://math.nist.gov/MatrixMarket/data/Harwell-Boeing/ (last accessed April 2011)
9. Papadimitriou, C.H.: **The NP-completeness of the bandwidth minimization problem**, Computing 16, 3, 263–270 (1976)
10. Piñana, E., Plana, I., Campos, V., Martí, R.: **GRASP and Path Relinking for the matrix bandwidth minimization**, European Journal of Operational Research 153, 1, 200–210 (2004)
11. Rodriguez-Tello, E., Jin-Kao, H., Torres-Jimenez, J.: **An improved Simulated Annealing Algorithm for the matrix bandwidth minimization**. European Journal of Operational Research 185, 3, 1319–1335 (2008)
12. Simon, H. A.: **Rational choice and the structure of the environment.** Psychological Review, 63, 2, 129–138 (1956)

13. http://www.mathworks.co.uk/products/matlab/demos.html?file=/products/demos/shipping/matlab/sparsity.html (last accessed April 2011)

Gloria Cerasela Crişan,
Department of Mathematics and Informatics
Faculty of Sciences, "Vasile Alecsandri" University of Bacău
ceraselacrisan@ub.ro

Camelia-M. Pintea,
G. Coşbuc N.College, Cluj-Napoca
cmpintea@yahoo.com