STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LV**, Number 4, 2010

A REINFORCEMENT LEARNING APPROACH FOR SOLVING THE MATRIX BANDWIDTH MINIMIZATION PROBLEM

GABRIELA CZIBULA[1], ISTVAN-GERGELY CZIBULA[1] AND CAMELIA-MIHAELA PINTEA[2]

Abstract. In this paper we aim at investigating and experimentally evaluating the reinforcement learning based model that we have previously introduced to solve the well-known matrix bandwidth minimization problem (*MBMP*). The *MBMP* is an *NP*-complete problem, which is to permute rows and columns of a matrix in order to keep its nonzero elements in a band lying as close as possible to the main diagonal. The *MBMP* has been found to be relevant to a wide range of applications including circuit design, network survivability, data storage and information retrieval. The potential of the reinforcement learning model proposed for solving the *MBMP* was confirmed by the computational experiment, which has provided encouraging results.

1. Introduction

The *MBMP* is an *NP*-complete problem, which has been found to be relevant to a wide range of applications including circuit design, network survivability, data storage and information retrieval.

Given a square matrix $\mathcal{A} = (a_{ij})_{n\times n}$, the matrix bandwidth minimization problem consists in finding a permutation of the rows and columns of $\mathcal{A}$ that keeps the nonzero elements in a band lying as close as possible to the main diagonal of $\mathcal{A}$. The problem can be easily stated in terms of graph optimization problem, considering a vertex for each row and a vertex for each column and connecting vertex i to vertex j through an edge either if $a_{ij} \neq 0$ or $a_{ji} \neq 0$. This way, a graph $G_{\mathcal{A}}$ is associated to the problem of minimizing the bandwidth of matrix $\mathcal{A}$ and the original problem is equivalent to the problem

Received by the editors: November 5, 2010.

2000 *Mathematics Subject Classification.* 65K10, 68T05.

1998 *CR Categories and Descriptors.* I.2.6[**Computing Methodologies**]: Artificial Intelligence – *Learning*; I.2.8[**Computing Methodologies**]: Problem Solving, Control Methods, and Search – *Heuristic methods* .

Key words and phrases. Combinatorial optimization, Matrix Bandwidth Minimization Problem, Reinforcement Learning.

of finding a labeling f of the vertices that minimizes the maximum difference between labels of adjacent vertices in $G_{\mathcal{A}}$. The matrix bandwidth minimization problem has been shown to be NP-complete [10].

The bandwidth minimization problem is relevant to a wide range of optimization applications. In solving large linear systems, Gaussian elimination can be performed much faster on matrices with a reduced bandwidth. Bandwidth minimization has also found applications in circuit design and saving large hypertext media [13]. Other practical problems are found in data storage, VLSI design, network survivability,industrial electromagnetics [5], finite element methods for approximating solutions of partial differential equations, large-scale power transmission systems, circuit design, chemical kinetics and numerical geophysics [6, 7].

Reinforcement Learning (RL) [4] is an approach to machine intelligence in which an agent can learn to behave in a certain way by receiving punishments or rewards on its chosen actions.

We have previously introduced in [8] a theoretical reinforcement learning based model for solving the *MBMP* problem. In this paper we detail our previous approach, also providing an experimental evaluation of it. The obtained results are good enough, indicating the potential of using RL techniques for solving the *MBMP*.

To our knowledge, excepting our proposal, a RL model for solving the $MBMP$ problem hasn't been reported in the literature, so far.

The rest of the paper is organized as follows. Section 2 briefly presents existing approaches for solving the Matrix Bandwidth Minimization Problem. The reinforcement learning model that we propose for solving the *MBMP* is introduced in Section 3. Section 4 provides an experimental evaluation of the RL approach and Section 5 contains some conclusions of the paper and future development of our work.

2. Related Work

Because of the importance of the bandwidth minimization problem, much research has been carried out in developing algorithms for it.

Cuthill and McKee propose in 1969 in [1] the first, best-known, stable and simple heuristic method for *MBMP*. It is the well-known CM algorithm, which used Breadth-First Search to construct a level structure of the graph. By labeling the vertex in the graph according to a level structure, good bandwidth minimization results are achieved in a short time. CM starts from a vertex with minimum degree and constructs a list of vertices that is the new vertices ordering. At each step, all the vertices that are adjacent to those already in list are appended, in ascending degree order. As the graph is connected, the

procedure stops when the list has $|V|$ positions filled, V being the number of vertices. The popular software package MatLab uses the command SYMRCM to find a permutation of vertices with good bandwidth, based on a variation of the Cuthill and McKee method.

The GPS algorithm proposed Gibbs, Poole and Stockmeyer in 1976 [14], is also based on level structure. Computational results show that the GPS algorithm is comparable to the CM algorithm in solution quality while being several times faster.

Marti et al. have used in [6] Tabu Search for solving the the *MBMP* problem. They used a candidate list strategy to accelerate the selection of moves in the neighborhood of the current solution. Extensive experimentation show that their Tabu Search outperforms the best-known algorithms in terms of solution quality in reasonable time.

A $GRASP$ with *Path Relinking* method given by Pinana et al. in [7] has been shown to achieve better results than the Tabu Search procedure but with longer running times.

Lim et al. propose in [13] a Genetic Algorithm integrated with Hill Climbing to solve the bandwidth minimization problem. Computational experiments show that the proposed approach achieves the best solution quality when compared with the GPS algorithm, *Tabu Search*, and the $GRASP$ with *Path Relinking* methods, being faster than the latter two heuristics.

A simulated annealing algorithm is presented in [15] for the matrix bandwidth minimization problem. The algorithm proposed by Tello et al. is based on three distinguished features including an original internal representation of solutions, a highly discriminating evaluation function and an effective neighborhood.

More recently, the Ant Colony Optimization (ACO) metaheuristic has been used in [17, 9] in order to solve the *MBMP*. The ant colony system was also hybridized in [9] with two local procedures which improve the system's performance.

3. A Reinforcement Learning Model for Solving *MBMP*

In this section we introduce the reinforcement learning model proposal for solving the *MBMP* problem. The theoretical model was previously introduced in [8] and will be detailed in this section and experimentally evaluated in Section 4.

3.1. **Reinforcement learning.**

Reinforcement learning is a synonym of learning by interaction [19]. During learning, the adaptive system tries some actions (i.e., output values) on its environment, then, it is reinforced by receiving a

scalar evaluation (the reward) of its actions. The reinforcement learning algorithms selectively retain the outputs that maximize the received reward over time. Reinforcement learning tasks are generally treated in discrete time steps.

At each time step, t, the learning system receives some representation of the environment's state s, it takes an action a, and one step later it receives a scalar reward r, and finds itself in a new state s'. The two basic concepts behind reinforcement learning are trial and error search and delayed reward [4].

One key aspect of reinforcement learning is a trade-off between *exploitation* and *exploration* [20]. To accumulate a lot of reward, the learning system must prefer the best experienced actions, however, it has to try (to experience) new actions in order to discover better action selections for the future.

3.2. **Our RL model.** Let us consider, in the following, that $\mathcal{A}$ is the symmetric matrix of order n whose bandwidth has to be minimized. The assumption that $\mathcal{A}$ is symmetric does not reduce the generality of the model.

We extend the set of vertices $\mathcal{V}$ from the graph $G_{\mathcal{A}}$ associated with the *MBMP* problem with a vertex denoted by s_0 and connected to all other vertices, i.e $\mathcal{V} = \{1, 2, ..., n\} \bigcup s_0$.

A general RL task is characterized by four components: a state space $\mathcal{S}$ that specifies all possible configurations of the system; the action space $\mathcal{A}$ that lists all available actions for the learning agent to perform; the transition function that specifies the possibly stochastic outcomes of taking each action in any state; and a reward function that defines the possible reward of taking each of the actions.

The RL task associated to the *MBMP* is defined as follows:

- The state space $\mathcal{S}$ (the agent's environment) consists of the extended set of vertices $\mathcal{V}$, i.e. $\mathcal{S} = \mathcal{V}$. The initial state si of the agent in the environment is s_0. A state $sf \in \mathcal{S}$ reached by the agent at a given moment after it has visited states $si, s_1, s_2, ...s_k$ is a *terminal* (final) state if the number of states visited by the agent in the current sequence is $n + 1$, i.e. $k = n$.
- The action space $\mathcal{A}$ is implicitly defined by a transition function between the states from $\mathcal{S}$. The transition function between the states is defined as $h : \mathcal{S} \to \mathcal{P}(\mathcal{S})$, where $h(s) = \{1, 2, ..., n\} \setminus \{s\}$, $\forall s \in \mathcal{S}$. This means that, at a given moment, from the state s the agent can move in any state from $\mathcal{S}$, excepting state s. We say that a state s' that is accessible from state s, i.e $s' \in h(s)$, is the *neighbor* (*successor*) state of s.

- The transitions between the states are equiprobable, the transition probability $P(s, s')$ between a state s and each neighbor state s' of s is equal to $\frac{1}{n}$.
- The reward function will be defined below (Formula (1)).

The *MBMP* formulated as a RL problem consists in training the agent to find a path $si, \pi_1, \pi_2, ..., \pi_n$ from the initial to a final state, i.e a permutation π of $\mathcal{V}$ that minimizes the matrix bandwidth. We denote by $\mathcal{A}^\pi = (a_{ij}^\pi)_{n \cdot n}$ the matrix obtained from the initial matrix $\mathcal{A}$ by permuting its lines and columns in the order $\pi_1, \pi_2, ..., \pi_n$.

It is known that the estimated utility of a state [3] in a reinforcement learning process is the estimated *reward-to-go* of the state (the sum of rewards received from the given state to a final state). So, after a reinforcement learning process, the agent learns to execute those transitions that maximize the sum of rewards received on a path from the initial to a final state.

As we aim at obtaining a permutation π of $\mathcal{V} = \{1, 2, ..., n\}$ that minimizes the matrix bandwidth, we define the reinforcement function as follows (Formula (1)):

- the reward received after a transition to a non terminal state is 0;
- the reward received after a transition to a final state π_n after states $si, \pi_1, \pi_2, ...\pi_{n-1}$ were visited is minus the bandwidth of matrix $\mathcal{A}^\pi$.

$$(1) \qquad r(\pi_k | si, \pi_1, \pi_2, ...\pi_{k-1}) = \begin{cases} 0 & if \quad k <> n \\ -\max_{a_{ij}^\pi \neq 0} |i - j| & otherwise \end{cases},$$

where by $r(\pi_k | \pi_1, \pi_2, ...\pi_{k-1})$ we denote the reward received by the agent in state π_k, after it has visited states $\pi_1, \pi_2, ...\pi_{k-1}$.

Considering the reward defined in Formula (1), as the learning goal is to maximize the total amount of rewards received on a path from the initial to a final state, it can be easily proved that the agent is trained to find a permutation π of $\mathcal{V} = \{1, 2, ..., n\}$ that minimizes the bandwidth of matrix $\mathcal{A}$.

During the training step of the learning process, the agent will determine its *optimal policy* in the environment, i.e the *policy* that maximizes the sum of the received rewards.

3.2.1. *The training process of the $MBMP$ agent.* For training the $MBMP$ agent, the $TD(\lambda)$ [4] algorithm is used. It is a temporal-difference (TD) method [18] combined with eligibility traces to obtain a more general and efficient learning method. λ refers to the use of an *eligibility trace* [21].

The *eligibility trace* is one of the basic mechanisms used in reinforcement learning to handle delayed reward. An eligibility trace is a record of the occurrence of an event such as the visiting of a state or the taking of an action

[4]. By associating one of such traces to every possible action in every state, the following temporal credit assignment is implemented: "Earlier states/actions are given less credit for the current TD error".

We have used the backward view of the $TD(\lambda)$ algorithm and *accumulating eligibility traces*. The basic idea is that on each step, the eligibility traces for all states decay, and the eligibility trace for the one state visited on the step is incremented by 1.

The idea of the training process is the following:

- The agent starts with some initial estimates of the states' utilities.
- During some training episodes, the agent will experiment (using the ϵ-Greedy action selection mechanism) some (possible optimal) paths from the initial to a final state, updating the states' utilities estimations according to the $TD(\lambda)$ algorithm.
- During the training process, the states' utilities estimations converge to their exact values, thus, at the end of the training process, the estimations will be in the vicinity of the exact values.

It is proven that $TD(\lambda)$ converges with probability 1 to an optimal policy and utility function as long as all state-action pairs are visited an infinite number of times and the policy converges in the limit to the Greedy policy [22].

After the training step of the agent has been completed, the solution learned by the agent is constructed by starting from the initial state and following the *Greedy* mechanism until a solution is reached. From a given state i, using the *Greedy* policy, the agent transitions to an unvisited neighbor j of i having the maximum utility value.

Consequently, the solution of $MBMP$ reported by the RL agent is a permutation π of $\{1, 2, ..., n\}$ such that $U(\pi_1) \geq U(\pi_2) \geq ... \geq U(\pi_n)$. By U we have denoted the utility function whose values were learned during the training step.

4. Computational Experiment

An experiment for evaluating the RL model proposed in Section 3 will be presented in the following. The proposed system was implemented in Java, using as benchmark *can__24* instance from Harwell-Boeing sparse matrix collection [11].

The parameter values for our implementation are as follows.

- The number of training episodes is set to 12000.
- The learning rate $\alpha \in [0, 1]$ is set to 0.01.
- The parameter indicating the use of eligibility traces $\lambda \in [0, 1]$ is set to 1.

- The discount factor $\gamma \in [0, 1]$ used for decreasing the eligibility traces is set to 0.9.
- Regarding the ϵ parameter used for the *epsilon*-Greedy action selection mechanism during the training step, the following strategy was used: we have started with $\epsilon = 0.85$ in order to favor exploration, then after the training progresses ϵ is decreased until it reaches a value near 0, 10^{-5}, which means that at the end of the training exploitation is favorized.

Figure 1 depicts the training process of the $MBMP$ agent, illustrating how, at the end of the training, the bandwidth of the learned solution becomes stable.

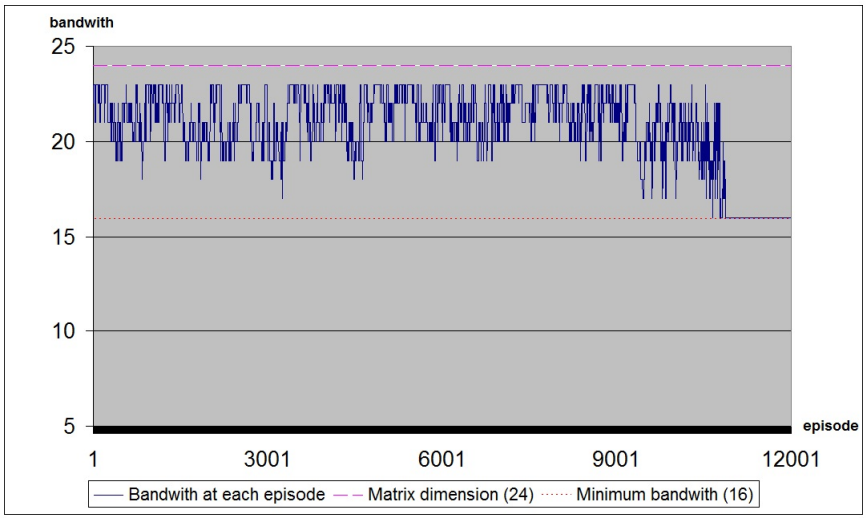

FIGURE 1. Results

We will compare in the following our approach with an approach that uses ant colony systems [16, 23]. The reason for this choice is that the ant systems can be related to reinforcement learning, as the pheromone by which the ants communicate can be viewed as a kind of reinforcement.

An ant colony system, ACS, was proposed in [9] for solving $MBMP$. The standard ACS was also hybridized with two local search mechanisms. For the benchmark *can_24*, ACS algorithm has reported a bandwidth 17. Compared with the standard ACS system, the solution reported our RL agent is better, but is worst than the solution reported by the hybrid ACS systems.

The results obtained by our approach are promising, but may be improved by further extensions of the proposed RL model.

5. Conclusions and Further Work

We have investigated in this paper a reinforcement learning based model for solving the matrix bandwidth minimization problem. The performed computational experiment has provided encouraging results, indicating the potential of using reinforcement learning techniques for the solving *MBMP*.

Further work will be made in order to improve the proposed RL model by investigating different reinforcement functions and by adding different local search mechanisms in order to increase the performance. We also aim at extending the proposed model to a distributed RL approach and at further evaluating it.

ACKNOWLEDGEMENT

This work was supported by CNCSIS - UEFISCSU, project number PNII - IDEI 2286/2008.

References

[1] Cuthill, E., McKee, J., *Reducing the bandwidth of sparse symmetric matrices*, Proceedings of the 1969 24th national conference, ACM Press, New York, NY, USA, 1969, pp. 157–172.

[2] Mitchell, T. M.: Machine Learning. New York: McGraw-Hill, 1997.

[3] Russell, S.J., Norvig, P., *Artificial intelligence. A modern approach*, Prentice-Hall International, 1995.

[4] Sutton, R., Barto, A., *Reinforcement Learning*, The MIT Press, Cambridge, London, 1998.

[5] Esposito, A., Catalano, M., S., Malucelli F., Tarricone, L., *Sparse Matrix Bandwidth Reduction: Algorithms, applications and real industrial cases in electromagnetics*, Advances in the theory of Computation and Computational Mathematics, Vol. 2, 1998, pp. 27–45.

[6] Marti, R., Laguna, M., Glover, F. and Campos, V., *Reducing the Bandwidth of a Sparse Matrix with Tabu Search*, European Journal of Operational Research, Vol. 135, No. 2, 2001, pp. 211–220.

[7] Pinana, E., Plana, I., Campos, V. Marti, R., *GRASP and Path Relinking for the Matrix Bandwidth Minimization*, European Journal of Operational Research, Vol. 153, Issue 1, 2004, pp. 200–210.

[8] Czibula, G., Crisan, G.C., Pintea, C.M., Czibula, I.G., Soft computing approaches on the Bandwith Problem, Proceedings of International Conference on Applied Mathematics (ICAM7), 2010, to be published.

[9] Pintea, C-M.,Crisan G-C., Chira C., *A Hybrid ACO Approach to the Matrix Bandwidth Minimization Problem*, HAIS **1** 2010, Springer LNCS (LNAI) 6076, 2010, pp. 405–412.

[10] Papadimitriou, C.H., *The NP-completeness of the bandwidth minimization problem*, Computing 16, 3, 1976, pp. 263-270.

[11] National Institute of Standards and Technology, Matrix Market, *Harwell-Boeing sparse matrix collection*, http://math.nist.gov/MatrixMarket/data/Harwell-Boeing/.

[12] Berry, M., Hendrickson, B., Raghavan, P., *Sparse matrix reordering schemes for browsing hypertext*, Lectures in Appl. Math. 32: The Mathematics of Numberical Analysis, 1996, pp. 99–123.
[13] Lim, A., Rodrigues, B., and Xiao, F., *Integrated genetic algorithm with hill climbing for bandwidth minimization problem*, Proceedings of the 2003 international Conference on Genetic and Evolutionary Computation, Lecture Notes In Computer Science., Springer-Verlag, Berlin, Heidelberg, 2003, pp. 1594–1595.
[14] Gibbs, N.E., Poole, W.G., Stockmeyer, P.K., *An algorithm for reducing the bandwidth and profile of sparse matrix*, SIAM Journal on Numerical Analysis **13(2)**, 1976, pp. 236–250.
[15] Rodriguez-Tello, E., Jin-Kao, H., Torres-Jimenez, J., *An improved Simulated Annealing Algorithm for the matrix bandwidth minimization*, European J. of Oper. Res., **185(3)**, 2008, pp. 1319–1335.
[16] Dorigo, M., Gambardella, L.M., *Ant Colony System: a cooperative learning approach to the traveling salesman problem*, IEEE Trans. on Evolutionary Computation, **1(1)**, 1997, pp. 53–66.
[17] Lim, A., Lin, J., Rodrigues, B., Xiao, F., *Ant Colony Optimization with hill climbing for the bandwidth minimization problem*, Appl. Soft Comput. , **6(2)**, 2006, pp. 180–188.
[18] Sutton, R.S., *Learning to predict by the methods of temporal differences*, Machine Learning **3**,1998, pp. 9–44.
[19] Perez-Uribe, A., *Introduction to Reinforcement learning*, 1998, http://lslwww.epfl.ch/∼anperez/RL/RL.html.
[20] Thrun, S.B., *The role of exploration in learning control*, *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*, Van Nostrand Reinhold, New York, NY, 1992.
[21] Singh, S.P., Sutton, R.S., *Reinforcement learning with replacing eligibility traces*, Machine Learning **22**, 1996, pp. 123–158.
[22] Dayan P., Sejnowski, T.J., *TD(lambda) Converges with Probability 1*, Machine Learning, Volume 14, Number 1, 1994, pp. 295–301.
[23] Lupea, M., *Default reasoning by ant clony optimization*, Studia Univ. Babes-Bolyai, Informatica, **LIV**, Number 2, 2009, pp. 71–82.

[1]Babeş-Bolyai University, 400084 Cluj-Napoca, Romania; [2];George Coşbuc N. College, 400083 Cluj-Napoca, Romania;
E-mail address: gabis@cs.ubbcluj.ro, istvanc@cs.ubbcluj.ro, cmpintea@yahoo.com