A Multivalued Neural Network for the Degree-Constrained Minimum Spanning Tree Problem*

Domingo López-Rodríguez[1], Enrique Mérida-Casermeiro[1], and Juan M. Ortíz-de-Lazcano-Lobato[2]

[1] Department of Applied Mathematics, University of Málaga, Málaga, Spain; {dlopez,merida}@ctima.uma.es

[2] Department of Computer Science and Artificial Intelligence, University of Málaga, Málaga, Spain; jmortiz@lcc.uma.es

Abstract The Degree Constrained Minimum Spanning Tree (DCMST) on a graph is the problem of generating a minimum spanning tree with constraints on the number of arcs that can be incident to vertices of the graph. In this paper, a new neural heuristic for the DCMST problem has been developed, making use of the multivalued recurrent model MREM, that has obtained very good results in other combinatorial optimization problems. The computational performance of our approach is compared against the performance of some algorithms from specialized literature. All these approaches are tested using standard problems taken from the literature.

1 Introduction

Consider an undirected complete graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{n_1, \ldots, n_N\}$ is the set of N nodes (or vertices) and $\mathcal{E} = \{e_1, \ldots, e_m\}$ is the set of m arcs (or edges), with given costs c_ℓ for each $\ell \in \{1, \ldots, m\}$, and numbers $b_i \in \mathbb{Z}^+$.

A spanning tree of the connected graph $\mathcal{G}$ can be defined as a maximal subset of $\mathcal{E}$ (edges of $\mathcal{G}$) that contains no cycle. Equivalently, a spanning tree is a minimal set of edges that connect all vertices in the graph.

The degree constrained minimum spanning tree (DCMST) problem on $\mathcal{G}$ is to find a spanning tree $\mathcal{T} = (\mathcal{V}, \mathcal{E}')$, with $\mathcal{E}' \subset \mathcal{E}$, such that the expression

$$C(\mathcal{T}) = \sum_{\ell \in \mathcal{E}'} c_\ell \tag{1}$$

is minimum, subject to $d_i \leq b_i$ for all $i \in \mathcal{V}$, where d_i is the number of arcs incident at node i, that is, the degree of each node in the tree $\mathcal{T}$ is bounded by a positive constant b_i.

* This work has been partially supported by Junta de Andalucía project number P06-TIC-01615.

While the unconstrained minimum spanning tree (MST) problem can be solved easily in polynomial time, the DCMST problem is NP-hard [1]. It is well-known [2,3] that even approximating optimal DCMST solutions within a constant factor is NP-hard. Therefore, heuristics are often used to find good solutions in a reasonable amount of time.

Direct and frequent applications of DCMST problem arise in the design of telecommunication and energy networks, as well as in the design of networks for computer communication, transportation, sewage and plumbing.

- The most common example of application is where N terminals (or nodes) need to be connected by making use of a minimum length of wiring or canals or pipes. However, the handling capacity of each of the terminals imposes a constraint on the number of wires (or arcs) that can be connected to any terminal (see Narula and Ho [4] for additional examples).
- The DCMST is used by Gavish [5] as a subproblem in the design of a centralized computer network.
- The DCMST may be used in the design of a road system which has to serve a collection of cities and has an additional restriction that no more than b_i roads may meet at any crossing (node i), see [6].
- It is also possible to impose a degree constraint on nodes in communication networks, in order to limit the vulnerability of the network in case of node failure.

Gavish [7] provides other examples of practical instances of the DCMST, and in [8] several examples of the types of optimisation problems that are faced in the process of designing computer communication networks are presented.

The rest of this work is organized as follows: In Sec. 2, some previous works on this problem are highlighted. Sec. 3 is focused in the neural model MREM, used in this paper, which is formally described. Then, in Sec. 4, the actual implementation of this model in order to solve DCMST instances is developed. Results of the experimental simulations are presented in Sec. 5. The last Section, Sec. 6, is devoted to give some conclusions and to indicate some future research lines.

2 Previous Works

In [4], Narula and Ho describe two greedy heuristics, and an exact, branch and bound approach (based on the lagrangean relaxation approach for the TSP that was employed by Held and Karp [9,10]). Gabow [11] presents an algorithm based on edge-exchanges on a reduced graph for the DCMST. Gavish [7] derives lower bounds based on a lagrangean relaxation by making use of a subgradient optimization approach. Savelsbergh and Volgenant [6] develop a branch and bound method based on a lagrangean relaxation, used in conjunction with a heuristic approach and an edge elimination idea that was first applied to the Traveling Salesman Problem (see [12]). Volgenant [13] described a branch and bound procedure, based on lagrangean relaxation and edge exchanges. Some heuristics,

including a neural network approach, for solving the DCMST, were developed by Craig, Krishnamoorthy, and Palaniswami [14].

Zhou and Gen [15,16] present an approach to solve DCMSTs using a genetic algorithm. Their method uses the concept of Prüfer numbers [17]. Unfortunately their papers do not provide enough detail to completely duplicate their algorithm, nor are there sufficient computational results to get a good understanding of the performance of their algorithm.

Boldon, Deo, and Kumar [2] develop four heuristics for solving the DCMST and implement them on a massively parallel SIMD machine. Deo and Kumar [18] study 29 constrained spanning trees including the degree constrained spanning tree and obtain results using a massively parallel computer. Their algorithm is based on repeatedly solving Minimum Spanning Tree (MST) problems with increasing penalties for arcs involved in degree violations.

Recently, Sequeiro [19] presented an ant colony model for solving multiobjective combinatorial optimization problems, whose main application was the DCMST problem.

3 The Neural Model MREM

It consists in a series of multivalued neurons, where the state of i-th neuron is characterized by its output (v_i), that can take any value in any finite set $\mathcal{M}$. This set can be a non numerical one, but, in this paper, the neuron outputs only take value in $\mathcal{M} \subset \mathbb{N}$.

The state vector $\boldsymbol{V} = (v_1, v_2, \ldots, v_N) \in \mathcal{M}^N$ describes the network state at any time, where N is the number of neurons in the net. Associated with any state vector, there is an energy function $E : \mathcal{M}^N \to \mathbb{R}$, defined by the expression:

$$E(\boldsymbol{V}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{i,j} f(v_i, v_j) + \sum_{i=1}^{N} \theta_i(v_i) \tag{2}$$

where $W = (w_{i,j})$ is the synaptic weight matrix, $f : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ is usually a similarity function since it measures the similarity between the outputs of neurons i and j. Note that MREM with only two states ($\{-1, 1\}$ or $\{0, 1\}$) and $f(x, y) = xy$ is precisely Hopfield's (bipolar or binary) model. For each i, $\theta_i \colon \mathcal{M} \to \mathbb{R}$ represents the bias or threshold function associated to neuron i.

The purpose of the net is to minimize the energy function described before. To this end, a random initial state $\boldsymbol{V}_0$ is introduced into the net and at time t, state vector $\boldsymbol{V}(t)$ will be changed into another state vector $\boldsymbol{V}(t+1)$ (defined by the computational dynamics) if $E(\boldsymbol{V}(t+1)) < E(\boldsymbol{V}(t))$. If there is not any $\boldsymbol{V}'$, in the neighborhood of $\boldsymbol{V}(t)$, such that $E(\boldsymbol{V}') < E(\boldsymbol{V}(t))$, then the net stops iterating.

In this case, $\boldsymbol{V}(t)$ is a local minimum of the energy function. Intuitively, the set of local minima will depend on the dynamics considered for this model.

Several dynamics (both sequential [20] and parallel [21]) are allowed in this model, due to its generality, although the most adequate dynamics will be determined by the problem to be tackled.

This model has been successfully applied to other combinatorial optimization problems, see [22,20,23,24,25].

4 Implementation for DCMST Problem

In order to get DCMST problem solved by MREM, a dynamics based on the edge-exchange technique [13] has been implemented.

Our proposed implementation consists in two MREM networks performing jointly. One of them, which will be referred to as H_1, will be the main network, trying to solve the problem by means of edge-exchange techniques. The other network, H_2, is an auxiliary network performing in the background the node clustering needed by H_1, as will be made clearer soon.

In order to represent a solution tree, the net H_1 will have m neurons, one per edge of the graph. The output or state of H_1 will be a vector $\boldsymbol{V} = (v_1, \ldots, v_m) \in \{0,1\}^m$, where $v_j = 1$ indicates that j-th edge in the initial graph $\mathcal{G}$ is present in the solution tree $\mathcal{T}$, and $v_j = 0$ otherwise.

For this network H_1, only state vectors $\boldsymbol{V}$ representing feasible trees are allowed, what gives the condition $\sum_{j=1}^{m} v_j = N - 1$.

From a simple identification of the cost function, given by Eq. (1), and the energy function, given by Eq. (2), the following can be deduced:

- The synaptic weight matrix, W, is equal to 0.
- The similarity function, in this case, is also equal to the zero function in $\mathcal{M} \times \mathcal{M}$.
- The threshold function $\theta_i : \mathcal{M} \to \mathbb{R}$ is

$$\theta_i(v) = \begin{cases} c_i, \text{ if } v = 1 \\ 0, \text{ if } v = 0 \end{cases}$$

That is, the energy function only adds the costs or weights corresponding to edges k such that $v_k = 1$, i.e., edges present in the solution tree.

It must be noted that, in this combinatorial optimization problem, the only nonzero element in the description of the network is the bias or threshold functions.

First, the net is initialized with a random feasible solution. To this end, the algorithm presented in Table 1 is used.

Once the feasible initial state $\boldsymbol{V}$ for H_1 is obtained, this net iterates in order to reduce the value of its energy function, which is equivalent to the cost function given by Eq. (1).

In every iteration, H_1 selects (sequentially) two edges (i_1 and i_2) present in the tree (that is, $v_{i_1} = v_{i_2} = 1$), and delete them, by considering $\boldsymbol{V}'$, a modification of the current state $\boldsymbol{V} = \boldsymbol{V}(t)$, such that $v'_{i_1} = v'_{i_2} = 0$. It must be noted that the subgraph induced by $\boldsymbol{V}'$ has 2 or 3 connected components.

Now, the net H_2 performs a node clustering in order to identify the connected components of the given subgraph. To this end, this net H_2 will have N

Table 1. Algorithm to build the initial state $\boldsymbol{V} = (v_1, \ldots, v_m)$ of H_1.

1. Select one edge $\ell = (i, j)$ at random. $\mathcal{L} = \{i, j\}$. $v_\ell = 1$.
2. Repeat until $\mathcal{L} = \mathcal{V}$:
(a) Study the set of candidate edges $\mathcal{E}_\mathrm{c}$, connecting nodes in $\mathcal{L}$ to nodes outside $\mathcal{L}$, and not violating the degree constraint.
(b) If $\mathcal{E}_\mathrm{c} = \varnothing$, select randomly a node i with $\deg(i) \geq b_i$. Then select an edge $e_k = (i, j)$ for some random j, and make $v_k = 0$.
(c) If $\mathcal{E}_\mathrm{c} \neq \varnothing$, select one edge $e_k = (i_k, j_k) \in \mathcal{E}_\mathrm{c}$ at random. $\mathcal{L} = \mathcal{L} \cup \{i_k, j_k\}$. $v_k = 1$.

neurons, and the output of each neuron, denoted by $s_i \in \{1, 2, 3\}$, will indicate the connected component the i-th node is assigned to.

This enables the first net, H_1, to locate new candidate edges, whose end-points are in different connected components of the considered subgraph, and not violating the degree constraint. That is, the set of candidates $\mathcal{E}_\mathrm{c} = \{\ell_1, \ldots, \ell_K\}$, where ℓ_j is the index of the arc with end-points (x_j, y_j), which verify $s_{x_j} \neq s_{y_j}$, $\deg(x_j) \leq b_{x_j}$ and $\deg(y_j) \leq b_{y_j}$, for all $j = 1, \ldots, K$. With this definition of the set of candidates, we ensure that the output of the net, in each iteration, represents a feasible tree satisfying all the restrictions.

Then, H_1 computes the increment of energy ΔE corresponding to exchange edges i_1, i_2 with all possible candidate edges ℓ_{j_1}, ℓ_{j_2}:

$$\Delta E(\ell_{j_1}, \ell_{j_2}) = c_{\ell_{j_1}} + c_{\ell_{j_2}} - (c_{i_1} + c_{i_2})$$

It must be noted that the term $c_{i_1} + c_{i_2}$ is constant since edges i_1 and i_2 are *a priori* fixed. So, the net actually computes the reduced energy increment, given by $\Delta E^*(\ell_{j_1}, \ell_{j_2}) = c_{\ell_{j_1}} + c_{\ell_{j_2}}$.

The scheduling of the network selects a pair of distinct edges ℓ_{j_1}, ℓ_{j_2} which minimize the increase of energy:

$$\Delta E^*(\ell_{j_1}, \ell_{j_2}) = \min_{j,k:j\neq k} \Delta E^*(\ell_j, \ell_k)$$

Then, the following state $\boldsymbol{V}(t+1) = (v_1(t+1), \ldots, v_m(t+1))$ of the net is defined from $\boldsymbol{V}'$ as follows:

$$v_j(t+1) = \begin{cases} 1, & \text{if } j = \ell_{j_1}, \ell_{j_2} \\ v'_j, & \text{otherwise} \end{cases}$$

With this dynamics, the net H_1 always decreases its energy value, achieving, in the limit, a local minimum which represents a spanning tree satisfying all degree constraints.

An example of the proposed dynamics is represented in Fig. 1.

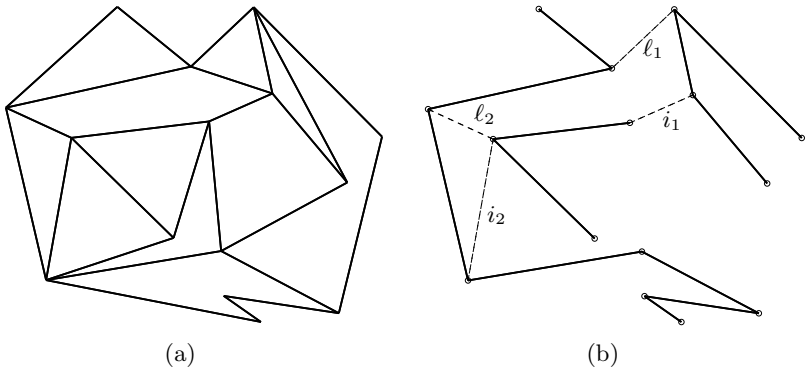

Figure 1. Example of the proposed dynamics: (a) original graph, (b) edges i_1 and i_2 are removed from the tree and ℓ_1 and ℓ_2 are inserted, without violating any degree constraint ($b_i = 3$ for all i).

5 Simulation Results

In this Section, we compare the efficiency of our method to some well-known algorithms designed to solve the DCMST problem.

The test graphs considered in our experiments belong to a family of structured hard graphs which are built by using non-Euclidean distances between nodes. Following the convention in earlier papers in the literature, all coordinates and distances are set to integer units. In addition, the bound b_i in the degree of each node has been taken constant for all nodes, $b_i = b$.

SHRD Graphs: The first node is connected to all other nodes by an edge of length L, the second node is connected to all nodes but the first by an edge of length $2L$ and so on. As usual, the value for L is 20.

In all cases, we have made 20 independent executions for each graph size and value of b.

Our approach has been compared against a series of algorithms, which constitute the state-of-the-art for solving DCMSTs, most of them taken from the review made in [26], such as:

- Evolutionary Algorithms like F-EA, and P-EA [26], and W-EA [27].
- Problem space search (PSS) [26].
- Simulated Annealing (SA) [26].
- Branch and Bound (B&B) [26].

The main difference between the distinct evolutionary approaches mentioned above is the encoding used to represent a valid tree. While F-EA and P-EA use the Prüfer number [17] of the tree to encode the individuals, the W-EA approach is weight-coded. See their references for more details.

Up to date, the best results were obtained by the weight-coded EA (W-EA).

To compare the relative efficiency of these algorithms, we have used the d-Prim algorithm [4] as a reference algorithm to compute relative quality improvements for the other approaches. This d-Prim algorithm is a variant of the

Table 2. Comparative results of the algorithms mentioned in the text against the proposed algorithm. Best results per row are indicated in bold.

			F-EA	P-EA	PSS	SA	B&B	W-EA	Prop.	
Problem	N	b	Avg.	Avg.	Avg.	Avg.	Avg.	Avg.	Avg.	t
SHRD150	15	3	13.66	15.07	16.62	14.93	18.03	14.20	**20.00**	1.03
		4	10.83	0.39	12.99	11.61	**14.76**	11.42	11.11	1.00
		5	4.00	-1.07	**9.60**	9.07	**9.60**	3.53	6.90	0.88
SHRD200	20	3	11.32	5.38	10.91	10.43	10.91	12.29	**23.07**	3.17
		4	6.82	0.80	7.05	5.57	7.05	8.50	**14.06**	3.46
		5	6.28	1.46	7.30	7.74	7.30	7.96	**9.80**	3.12
SHRD250	25	3	13.07	13.41	15.40	14.73	15.40	16.51	**25.45**	7.94
		4	4.84	1.59	6.79	5.56	6.79	6.83	**16.83**	9.12
		5	5.37	5.92	6.74	5.19	8.29	9.01	**11.39**	8.49
SHRD300	30	3	6.51	6.51	11.27	9.53	11.27	12.50	**26.40**	17.53
		4	7.30	3.79	10.58	8.45	10.58	11.76	**17.81**	19.76
		5	2.18	0.19	5.74	2.50	4.74	5.77	**12.39**	19.20
TOTAL			7.68	4.45	10.00	8.78	10.39	10.02	**16.27**	7.89

well-known Prim algorithm to compute the minimum spanning tree of a graph, in which the degree constraint is imposed.

Values appearing in Table 2 are computed as follows:

$$\frac{C_{d-\text{Prim}} - C_{\text{alg}}}{C_{d-\text{Prim}}} \cdot 100$$

where $C_{d-\text{Prim}}$ and C_{alg} are the cost of the optimal solutions found by d-Prim and a given algorithm, respectively. Larger values indicate better results.

It must be noted that our model is able to outperform the other algorithms in most cases. It obtains much better results on average than the other approaches, and specially as the number of nodes increases. So, for larger problem instances, it is expected that our model achieve very good solutions.

In the last column, time values of our model are presented. It can be observed that our technique is not very time-consuming, being able to solve a large problem instance ($N = 30$), in less than 20 seconds.

6 Conclusions and Future Work

In this work we have presented the application of the multivalued neural model MREM to the solution of the well-known degree-constrained minimum spanning tree problem.

With the use of two networks from this model, we are able to build a feasible solution to this problem, satisfying the degree constraints in every iteration, as well as to compute the connected components of the solution.

By means of computational experiments, our method has proved to outperform other algorithms from the literature.

It must be observed that, if a parallel dynamics for this model is used, computational times may be drastically reduced. This will be an issue of study in future works.

Future research will cover aspects such as to develop a method for solution improvement, based on stochastic dynamics of the network, or to incorporate techniques to escape from local minima of the energy function of deterministic nature.

References

1. Garey, M.R., Johnson, D.S.: Computers and Intractability. A guide to the theory of NP-Completeness. W. H. Freeman and Company (1979)
2. Boldon, B., Deo, N., Kumar, N.: Minimum-weight degree-constrained spanning tree problem: Heuristics and implementation on an simd parallel machine. Parallel Computing **22** (1996) 269 – 382
3. Knowles, J., Corne, D.: A new evolutionary approach to the degree-constrained minimum spanning tree problem. IEEE Trans. On Evolutionary Computation **4** (2000) 125 – 134
4. Narula, S., Ho, C.A.: Degree constrained minimum spanning tree. Computers and Operations Research **7** (1980) 239 – 249
5. Gavish, B.: Augmented lagrangean based algorithms for centralized network design. IEEE Transactions on Communications **33** (1985) 1247 – 1257
6. Savelsbergh, M., Volgenant, T.: Edge exchanges in the degree-constrained minimum spanning tree problem. Computers and Operations Research **12** (1985)
7. Gavish, B.: Topological design of centralized computer networks – formulations and algorithms. Networks **12** (1982) 355 – 377
8. Gavish, B.: Topological design of centralized computer networks – the overall design problem. European Journal of Operations Research **58** (1992) 149 – 172
9. Held, M., Karp, R.M.: The travelling-salesman problem and minimum spanning trees. Operations Research **18** (1970) 1138 – 1162
10. Held, M., Karp, R.M.: The travelling-salesman problem and minimum spanning trees: Part ii. Mathematical Programming **1** (1971) 6 – 25
11. Gabow, H.N.: Good algorithm for smallest spanning trees with a degree constraint. Networks **8** (1978) 201 – 208
12. Volgenant, A., Jonker, R.: The symmetric travelling salesman problem and edge exchanges in minimum 1-trees. European Journal of Operational Research **12** (1983) 394 – 403
13. Volgenant, A.: A langrangean approach to the degree constrained minimum spanning tree problem. European Journal of Operational Research **39** (1989) 325 – 331
14. G., C., Krishnamoorthy, M., Palaniswami, M.: Comparison of Heuristic Algorithms for the Degree Constrained Minimum Spanning Tree. In: Metaheuristics: Theory and Applications. Kluwer (1996)
15. Zhou, G., Gen, M.: Approach to the degree-constrained minimum spanning tree problem using genetic algorithms. Engineering Design and Automation **3** (1997) 156 – 165
16. Zhou, G., Gen, M.: A note on genetic algorithms for degree constrained minimum spanning tree problems. Networks **30** (1997) 105 – 109

17. Moon, J.W.: Various Proofs of Cayley's Formula for Counting Trees. In: A Seminar on Graph Theory. New York: Holt, Rinehart and Winston (1967)
18. Deo, N., Kumar, N.: Computation of constrained spanning trees: A unified approach. Network Optimization **450** (1997) 194 – 220
19. Sequeira Cardoso, P.J.: Ant Colony Algorithms for Multiple Objective Combinatorial Optimization. PhD thesis, University of Seville (2006)
20. Mérida-Casermeiro, E., Galán-Marín, G., Muñoz Pérez, J.: An efficient multivalued hopfield network for the travelling salesman problem. Neural Processing Letters **14** (2001) 203 – 216
21. Mérida-Casermeiro, E., Muñoz Pérez, J., Domínguez-Merino, E.: An n-parallel multivalued network: Applications to the travelling salesman problem. **LNCS 2686** (2003) 406 – 413
22. Mérida-Casermeiro, E., Muñoz-Pérez, J., Benítez Rochel, R.: A recurrent multivalued neural network for the n-queens problem. Lecture Notes in Computer Science **2084** (2001) 522 – 529
23. Mérida-Casermeiro, E., López-Rodríguez, D.: Multivalued neural network for graph maxcut problem. In: Lecture Series on Computer and Computational Sciences 2004. Volume 1. (2004) 375 – 378
24. Mérida-Casermeiro, E., López-Rodríguez, D.: Graph partitioning via recurrent multivalued neural networks. Lecture Notes in Computer Science **3512** (2005) 1149 – 1156
25. López-Rodríguez, D., Mérida-Casermeiro, E., Ortiz-de-Lazcano-Lobato, J.M., López-Rubio, E.: Image compression by vector quantization with recurrent discrete networks. Lecture Notes in Computer Science **4132** (2006) 595 – 605
26. Krishnamoorthy, M., Ernst, A.T., Sharaiha, Y.M.: Comparison of algorithms for the degree constrained minimum spanning tree. Journal of Heuristics **7** (2001) 587 – 611
27. Raidl, G.R., Julstrom, B.A.: A weighted coding in a genetic algorithm for the degree-constrained minimum spanning tree problem. In: Proceedings of the 2000 ACM Symposium on Applied Computing, ACM Press (2000)