Neural Network, Kernighan-Lin and Multilevel Heuristics for the Graph Bisection Problem on Geometrically Connected Graphs

Gonzalo Hernandez
Center for Mathematical Modeling
University of Chile
Santiago, Chile
Email: ghernandez@dim.uchile.cl

Jorge Cornejo, Roberto Leon, Luis Salinas
Department of Informatics
Federico Santa Maria Technical University
Valparaiso, Chile
Email: {jcornejo,rleon,lsalinas}@inf.utfsm.cl

Abstract

A neural network heuristic for the Graph Bisection Problem is studied numerically on geometrically connected graphs and its performance is compared with the Kernighan - Lin (KL) and Multilevel (ML) heuristics. For medium-scale sparse graphs with $n = 2000$ to $n = 12000$ nodes it was obtained that the NN heuristic applied to the Graph Bisection Problem present a greedy behaviour in comparison to other local improvements heuristics: Kernighan-Lin, Multilevel. The experimental results for large graphs recommend to use KL as partitioning heuristic for sparse geometrically connected graphs.

I. Introduction

The graph partitioning (GPP) is a well known NP-Complete combinatorial optimization problem, see refs. [4], [10], that has been applied in different areas of computer science, for example: processes load balancing (see ref. [6]) and circuit layout.

Let $G = (V, E)$ be a finite, undirected and connected graph, where V is the set of vertices and $E \subseteq V \times V$ is the set of edges. The GPP consists then in finding p subsets of vertices $V_1, V_2, ..., V_p$, the partition of set V, verifying:

$$\bigcup_{k=1}^{p} V_k = V, \qquad \sum_{k=1}^{p} n_k = n \text{ where } n_k = |V_k|\,, n = |V| \tag{1}$$

and such that the cardinality of the cut set:

$$\bigcup_{\substack{k,l=1 \\ k \neq l}} \{(i,j) \in E / i \in V_k, j \in V_l, \forall i,j = 1,...,n, i \neq j\} \tag{2}$$

is minimal. Several heuristics have been proposed to solve GPP, see for instance refs. [2], [3], [9], [11]. Different computational studies, see refs. [1], [7], [8], [12], have shown that the best heuristic is Multilevel in terms of the distance to the optimal solution.

In this work is studied numerically a neural network (NN) heuristic for the Graph Bisection Problem (a special case of the Graph Partitioning Problem).

II. Neural Network Heuristic for the GBP

A neural network with parallel dynamics is defined by, see ref. [5]: (a) a connectivity matrix $W = (w_{ij})$ $i, j = 1, ..., n$, where w_{ij} represents the interaction weight between neurons i, j; (b) a threshold vector $b = (b_i)$ $i = 1, ..., n$, where b_i is the threshold of neuron i; (c) a local transition function f_H :

$$\begin{array}{ll} x(0) \in \{0,1\}^n & x(t+1) = (x_1(t+1), ..., x_n(t+1)) \\ x_i(t+1) = f_H\left(\sum_{j=1}^{n} w_{ji} x_j(t) - b_i\right) & i = 1, ..., n \end{array} \tag{3}$$

where f_H is the Heaviside function.
In ref. [5] was proved that if the connectivity matrix is symmetric with non negative diagonal, the parallel dynamics (3) converges to fixed points or cycles of length 2. Therefore, the parallel dynamics (3) defines an optimization heuristic.
A combinatorial optimization problem equivalent to the GBP must be obtained in order to solve it with an optimization heuristic of the kind (3) and defined by a quadratic objective function. First, we have to redefine the variables: $x_i \in \{0,1\} \leftrightarrow y_i \in \{-1,1\}, y_i = 2x_i - 1$. Then: $y_i = -1$ iff $i \in V_1$, $y_i = 1$ iff $i \in V_2$. In addition:

$$\begin{aligned} \left|\{(i,j) \in E / i \in V_1, j \in V_2\}\right| &= -\sum_{i\in V_1}^{n}\sum_{j\in V_2}^{n} w_{ij} y_i y_j \\ \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} y_i y_j &= -4\left|\{(i,j) \in E / i \in V_1, j \in V_2\}\right| + |E| \end{aligned} \tag{4}$$

Hence the following combinatorial optimization problem is equivalent to the GBP:

$$\min_{y\in\{-1,1\}^n} -\tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} y_i y_j \qquad \text{s.t.} \quad \sum_{i=1}^{n} y_i = 0 \tag{5}$$

If the constraint is penalized:

$$\min_{y\in\{-1,1\}^n} G_\alpha(y) = -\tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} r_{ij}(\alpha)\, y_i y_j \tag{6}$$

where:

$$r_{ij}(\alpha) = \begin{cases} 0 & \text{If } i = j \\ 1-\alpha & \text{If } i \neq j \wedge (i,j) \in E \\ -\alpha & \text{If } i \neq j \wedge (i,j) \notin E \end{cases} \tag{7}$$

and α is the penalization parameter. Therefore, a neural network can be associated to the GBP with connectivity matrix $R(\alpha) = (r_{ij}(\alpha))\ i,j = 1,...,n$, zero threshold vector $b = (0)_{i=1}^{n}$, Lyapunov functional $G_\alpha(y)$ and parallel dynamics defined by:

$$y(0) \in \{-1,1\}^n, \quad y(t+1) = (y_1(t+1),...,y_n(t+1)), \quad y_i(t+1) = sgn\left(\sum_{j=1}^{n} r_{ji}(\alpha)\, y_j(t)\right)\ i = 1,...,n \tag{8}$$

where $sgn(u)$ is the sign function.
The parallel dynamics (8) converges only to fixed points or cycles of length 2, which also are local minima of G_α. Hence, it defines a local search optimization heuristic for G_α.

III. Performance of the NN, KL and ML Heuristics

The numerical results were obtained using medium-scale sparse geometrically connected graphs, which are useful to measure the performance of heuristics that produce improvements in the solution by local changes. A geometrically connected graph of size n and connectivity radius r can be constructed by n random points (x_i, y_i) $i = 1,...,n$ in the unit square $S = [0,1] \times [0,1]$. These points will represent the location of the nodes and it computes the graph connectivity matrix $W = (w_{ij})$. The nodes i,j are connected and then $w_{ij} = w_{ji} = 1$ if and only if:

$$dist(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq r \tag{9}$$

In figure 1 is shown an example of a geometrically connected graph with $n = 4000$ and $r = 0.028$.

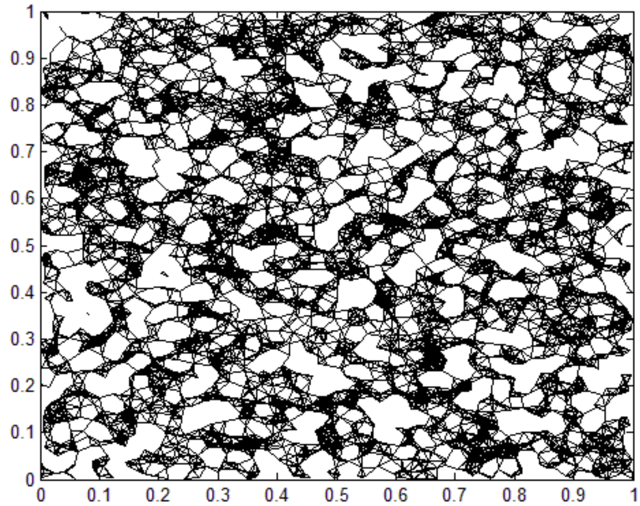

Figure 1. Example of a geometrically connected graph with $n = 4000$ and $r = 0.028$.

The performance of the Neural Network (NN), Kernighan-Lin (KL) and Multilevel (ML) sequential heuristics were computed using the graphs of table 1 and applying the methodology that follows:

- For each graph of table 1, 50 random initial conditions were generated: $y^0 \in \{-1, 1\}^n$.
- The NN, KL, and ML heuristics were applied starting from each initial condition y^0.
- The minimum cut was computed $MinCut = \min_{y^0} \left| \left\{ (i,j) \in E / y_i^\infty \neq y_j^\infty \right\} \right|$, where y^∞ is the configuration obtained from y^0 applying the dinamic (8).
- For the Multilevel heuristic were chosen a threshold equals to 500 and 1000, the Heavy Edge Matching for the coarsening phase and the Kernighan-Lin heuristic for the partitioning and uncoarsening with refinement phases, according to the results of refs. [1], [7].

Name	#Nodes (n)	#Edges $\|E\|$	Radius r	Name	#Nodes (n)	#Edges $\|E\|$	Radius r
G_{2000A}	2000	21307	0.06	G_{4000D}	4000	30144	0.035
G_{2000B}	2000	57213	0.1	G_{6000}	6000	31671	0.024
G_{4000A}	4000	11997	0.022	G_{8000}	8000	39788	0.02
G_{4000B}	4000	13054	0.023	G_{10000}	10000	49817	0.018
G_{4000C}	4000	19283	0.028	G_{12000}	12000	61366	0.0165

Table 1. Geometrically connected graphs used for the numerical study.

The numerical results that were obtained for the NN, KL and ML heuristics are summarized in tables 2 and 3.

Heuristic	NN		KL		ML (500)		ML (1000)	
Graph	$MinCut$	$Time\ [s]$	$MinCut$	$Time\ [s]$	$MinCut$	$Time\ [s]$	$MinCut$	$Time\ [s]$
G_{2000A}	4373	101	432	239	411	701	418	697
G_{2000B}	5936	124	1817	133	1880	894	1841	845
G_{4000C}	1166	786	548	1301	638	2053	704	1992
G_{4000D}	2013	788	561	1958	935	5179	954	5101
G_{6000}	2353	2856	1151	4473	1083	9741	1124	9702
G_{8000}	2970	6328	1730	8785	1601	20844	1587	20093
G_{10000}	3589	10036	2391	20989	1969	39449	2314	38389
G_{12000}	4453	22776	2790	49255	2852	81860	2788	80803

Table 2. Comparison of the performance of the NN, KL and ML sequential heuristics.

From table 2 it is clear that the solutions computed by NN have the larger distance to the best known solution for the graphs used. In table 3 are shown the distance of the best solution computed by the NN heuristic and the KL heuristic with respect to the best solution computed by the other heuristics: ML(500) and ML(1000).

Graph	$d_{NN,KL}$	$d_{NN,ML(500)}$	$d_{NN,ML(1000)}$	$d_{KL,ML(500)}$	$d_{KL,ML(1000)}$
G_{2000A}	0.90	0.91	0.90	0.05	0.03
G_{2000B}	0.69	0.68	0.69	-0.03	-0.01
G_{4000C}	0.53	0.45	0.40	-0.16	-0.28
G_{4000D}	0.72	0.54	0.53	-0.67	-0.70
G_{6000}	0.51	0.54	0.52	0.06	0.02
G_{8000}	0.42	0.46	0.47	0.07	0.08
G_{10000}	0.33	0.45	0.36	0.18	0.03
G_{12000}	0.37	0.36	0.37	0.02	0.00

Table 3. Distance of the best solution computed by the NN heuristic with respect to the others heuristics.

The NN heuristic presents a behaviour similar to a greedy heuristic; in fact, the quality of the solutions computed by NN tend to increase as n increases. The greedy quality of the NN heuristic can be explained due to the sparsity and local connectivity of the geometric connected graphs used in this study. Finally the performance of KL is better that the ML(500) and ML(1000) heuristics in small size graphs but similar in large graphs.

Acknowledgement. This work was partially supported by FONDECYT (Research Project 1050808 and 1070269) and UTFSM DGIP (Research Project 24.07.21).

IV. Conclusions

The numerical experiments developed in this work allow to affirm that the neural network heuristic applied to the Graph Bisection Problem present a greedy behaviour: It obtains good solutions in shorter times in comparison to the other local improvements heuristics that were studied: Kernighan-Lin, Multilevel. The NN heuristic must be applied as a sub-heuristic of a local improvements heuristic. For this reason, in order to improve the convergence times of ML we propose that the partitioning and uncoarsening with refinement phases of this heuristic must be performed by the NN heuristic in the case of sparse locally connected graphs, such geometrically connected graphs. For large graphs, the results for best known solution and convergence times suggest that for sparse geometrically connected graphs is recommendable to use KL as partitioning heuristic.

References

[1] Baños, R.,C. Gil, J. Ortega and F.G. Montoya, Parallel Heuristic Search in Multilevel Graph Partitioning, Proceedings of the 12th Euromicro Conference on Parallel, Distributed and Network-Based Processing (EUROMICRO-PDP'04), 2004.
[2] Battiti, R., and A. Bertosi, Greedy, Prohibition and Reactive Heuristic for Graph Partitioning, IEEE Transactions on Computers, Vol. 48, No 4, April 1999.
[3] Bui, T.N., and B. R. Moon, Genetic Algorithm and Graph Partitioning, IEEE Transactions on Computers, Vol. 45, No 7, pp. 841 - 855, July 1996.
[4] Garey, M.R., and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, New York: W. H. Freeman and Company, 1979.
[5] Goles, E., and S. Martinez, Neural and Automata Networks, Maths. and Its Applications Series, V. 58, Kluwer 1991.
[6] Hendrickson, B., Graph partitioning models for parallel computing, P. Computing, V. 26, N. 12, pp. 1519-1534, 2000.
[7] Karypis, G., V. Kumar, A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs, SIAM J. Scientific Computing, Vol. 20, No 1, pp. 359-392, 1998.
[8] Karypis, G., V. Kumar, Multilevel k-way Partitioning Scheme for Irregular Graphs, Journal of Parallel and Distributed Computing, Vol. 48, Iss. 1, pp. 96-129, 1998.
[9] Kernighan, B.W., and S. Lin, An Efficient Heuristic Procedure for Partitioning Graphs, The Bell System Tech. J., Vol. 49, pp. 291 - 307, 1970.
[10] Papadimitriou, Ch.H. and S. Steiglitz, Combinatorial Optimization, Prentice-Hall, Englewood Cliffs, HJ, 1982.
[11] Rolland, E., H. Pirkul, and F. Glover, A Tabu Search for Graph Partitioning, Annals of Operations Research, Metaheuristics in Combinatorial Optimization, Vol. 63, 1996.
[12] Walshaw, C., Multilevel Refinement for Combinatorial Optimisation Problems, Annals of Operations Research, Vol. 131, pp. 325 - 372, 2004.