Multilevel + Neural Network Heuristic for the Graph Bisection Problem on Geometrically Connected Graphs*

G. Hernandez[1], F. Bravo[1], P. Montealegre[1], F. Nuñez[1], L. Salinas[2]

[1] Center for Mathematical Modeling, University of Chile,
Blanco Encalada 2120, 7th Floor, Santiago, Chile (ghernandez@dim.uchile.cl).

[2] Department of Informatics, Santa María University,
Av. España 1680, Valparaíso, Chile.

Abstract. The Multilevel algorithm (ML) has been applied successfully as a metaheuristic for different combinatorial optimization problems: Graph Partitioning, Traveling Salesman, Graph Coloring, see refs. [6,7,18]. The main difficulty of ML are the convergence times needed to obtain solutions at a distance of 7% - 5% to the best known solution in large scale problems. In order to reduce these convergence times we studied numerically a Parallel Multilevel heuristic with Neural Network partitioning and uncoarsening + refinement phases (PML+PNN) for the Graph Bisection Problem on geometrically connected graphs. Our main result establish that for graphs with $n \in [4000,12000]$ vertices, the performance of the parallel ML+NN heuristic increases linearly as n increases with respect to the parallel ML heuristic. For $n \in \{10000,12000\}$ the distance to the best solution found is 0.32,0.25 respectively that is obtained with a quadratic computing time. This suggests improving the performance of the PML+PNN heuristic by means of a hill climbing improvement heuristic.

Keywords: Graph Bisection Problem, Multilevel + Neural Network Heuristic.

1 Introduction

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

$$\bigcup_{k=1}^{p} V_k = V, \sum_{k=1}^{p} n_k = n, \text{ where } n_k = |V_k|, n = |V| \quad (1)$$

and such that the cardinality of the cut set:

$$|c(V_1, \ldots, V_p)| = \bigcup_{k,l=1,k\neq l} \{(i,j) \in E / i \in V_k, j \in V_l, \forall i,j = 1, \ldots, n, i \neq j\} \quad (2)$$

* This work was partially supported by: FONDECYT 1100805, BASAL Financing Program for Scientific and Technological Centers of Excellence, UTFSM DGIP 24.09.54.

is minimal. If $p = 2$ and $n_1=n_2=(n/2)$ the graph partitioning problem reduces to the graph bisection problem. The GPP is a well known NP hard combinatorial optimization problem, see refs. [4,16], that has been applied in several areas of computer science, for example: processes load balancing (see ref. [6]) and circuit layout (see refs. [14,19]). Different heuristics and metaheuristics have been proposed to solve GPP, see for instance refs. [2,3,10,17]. In 1970 Kernighan and Lin, see ref. [14], proposed an effective local search heuristic that decreases the cardinality of the cut set by changing vertices from various partitions (swaps). The Multilevel algorithm, see ref. [6,7] computes a partition in 3 phases: coarsening, partitioning, uncoarsening, where the partitioning phase is performed by the KL heuristic or a greedy heuristic. The main idea of ML is to recursively reduce (coarse) the graph until obtain a size in which the partitioning phase can be applied more efficiently. Once an optimal partition is obtained the reduced graph is expanded to its original size (uncoarse) and the optimal partition is refined at each level. Several computational studies, see for instance refs. [1,7,11,12,13,18], have shown that ML is the best heuristic with respect to the distance to the optimal solution, and this fact has suggested its application to other combinatorial problems. The main pitfall of ML is the convergence times needed to obtain solutions in large scale problems at a 5% distance to the best known solution. In order to reduce these convergence times we studied numerically a multilevel heuristic with neural network (NN) partitioning phase (ML+NN).

2 Parallel Heuristics for the Graph Bisection Problem

A neural network is a dynamical system defined by, see refs. [5,8]:

- A connectivity matrix $W=(w_{ij})$ $i,j=1,...,n$, where w_{ij} represents the interaction weight between neurons i,j.
- A threshold vector $b=(b_i)$ $i=1,...,n$, where b_i is the threshold of neuron i.
- A local transition function f_H: At each time step all neurons change its state according to:

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

In the Hopfield neuronal model f_H is the Heaviside function. In refs. [5,8] it was proved that if the connectivity matrix W is symmetric with non negative diagonal, the dynamics (3) converges to fixed points or cycles of length 2, which also are local minima of the quadratic Lyapunov functional E_P defined by:

$$E_P(x(t)) \ = -\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ji}x_i(t)x_j(t-1) + \sum_{i=1}^{n} b_i(x_i(t) + x_i(t-1)) \tag{4}$$

Therefore, the dynamics (3) defines an optimization heuristic of the functional E_P. On the other hand, the GBP can be modeled by a combinatorial optimization problem with quadratic objective function and linear constraint. First, we have to redefine the variables:

$$x_i \in \{0,1\} \leftrightarrow y_i \in \{-1,1\} \tag{5}$$

Then: y_i=-1 iff $i \in V_1$ and y_i=1 iff $i \in V_2$. In addition:

$$\begin{gathered} |\{(i,j) \in E / i \in V_1, j \in V_2, \forall i,j = 1,\ldots,n, i \neq j\}| = -\sum_{i \in V_1}\sum_{j \in V_2} w_{ij}y_iy_j \\ \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}y_iy_j = -4|\{(i,j) \in E / i \in V_1, j \in V_2\}| + |E| \\ \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}y_iy_j = 4\sum_{i \in V_1}\sum_{j \in V_2} w_{ij}y_iy_j + |E| \end{gathered} \tag{6}$$

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

$$\begin{gathered} \min_{y \in \{-1,1\}^n} \ -\tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}y_iy_j \\ \text{such that} \quad \sum_{i=1}^{n} y_i = 0 \end{gathered} \tag{7}$$

If the constraint is quadratically penalized is obtained an unconstrained non linear integer problem:

$$\min_{y \in \{-1,1\}^n} \ G_\alpha(y) = -\tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} r_{ij}(\alpha)y_iy_j \tag{8}$$

where:

$$r_{ij}(\alpha) = \begin{cases} 0 & \text{If } i = j \\ 1-\alpha & \text{If } i \neq j \wedge (i,j) \in E \\ -\alpha & \text{If } i \neq j \wedge (i,j) \notin E \end{cases} \tag{9}$$

Therefore, the following optimization heuristic based on the neural network dynamics can be applied to find a local minimum of the GBP:

$$
\begin{aligned}
&y(0) \in \{-1,1\}^n \\
&y(t+1) = (y_1(t+1),\ldots,y_n(t+1)) \\
&y_i(t+1) = sgn\left(\sum_{j=1}^{n} r_{ji}(\alpha) y_j(t)\right) \; i = 1,\ldots,n
\end{aligned}
\tag{10}
$$

where $sgn(u)$ is the sign function. The dynamics (10) converges only to fixed points or cycles of length 2, which also are local minima of G_α. The NN dynamics can be parallelized in a distributed memory cluster in two ways: Distribute the update of the dynamics in different processors and run sequentially several copies of the dynamics in different processors. The second way is more efficient than the first one because it is not necessary to send in each iteration the updated state to all processors.

The sequential version of the Multilevel algorithm, see ref. [6,7,11,13] have 3 phases:

Phase 1. Coarsening: In the first phase the graph is reduced recursively conserving its structure until obtain a graph with size in which the partitioning phase can be applied more efficiently. In this way, it is obtained a sequence of graphs $G_0=G,G_1,G_2\ldots,G_p$ such that G_{i+1} is constructed from G_i by applying one of the following matching process: Random Matching, Heavy Edge Matching, Light Edge Matching, Heavy Clique Matching. The number of nodes of G_p is called the threshold and is usually chosen less than 1000.

Phase 2. Partitioning: In this phase, an optimal partition is computed for the reduced graph G_p by applying a fast greedy algorithm or the Kernighan and Lin heuristic.

Phase 3. Uncoarsening and Refinement: In the last phase, the reduced graph G_p is expanded to its original size and the optimal partition is refined at each level. In order to obtain a good solution it is not sufficient to apply a greedy heuristic and in general is applied the Kernighan and Lin heuristic.

In the Multilevel Algorithm the coarsening phase is the more time consuming and therefore must be parallelized in an efficient way. For this reason we implemented the coarsening phase described in ref. [12]. In the second and third phase of the parallel Multilevel algorithm we applied a simple parallel version of the Kernighan and Lin heuristic that runs several copies of the heuristic in different processors and chooses the best solution found.

In this work we propose a modified version of the parallel Multilevel heuristic that considers a Neural Network heuristic for the partitioning and coarsening + refinement phases (PML+PNN). The modified phases are parallelized by running several copies of the NN dynamics in different processors. In the next section the performance of the proposed parallel PML+PNN heuristic will be compared with the standard parallel Multilevel heuristics PML on several instances of the GBP Problem defined on sparse geometrically connected graphs.

3 Performance of the ML+NN Heuristic

A geometrically connected graph of size n and connectivity radius r can be constructed as follows:

i) Generate n random points (x_{i},y_{i}) i=1,...,n in the unit square S=[0,1]×[0,1]. These points will represent the location of the nodes.

ii) Compute the graph connectivity matrix W=(w_{ij}) i,j=1,...,n. The nodes i≠j are connected if and only if:

$$dist(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq r \Rightarrow w_{ij} = w_{ji} = 1 \qquad (11)$$

In figure 1 is shown an example of a geometrically connected graph with n=4000 and r=0.030.

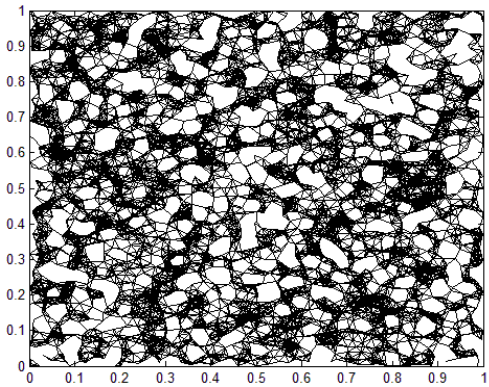

Fig .1. Example of geometrically connected graphs with n=4000 and r=0.030.

A geometrically connected graph can be used to model local connectivity while maintaining some level of irregularity. This characteristic can be appreciated in the degree of each vertex. In figure 2 is shown the histogram of degrees of the geometrically connected graph of figure 1.

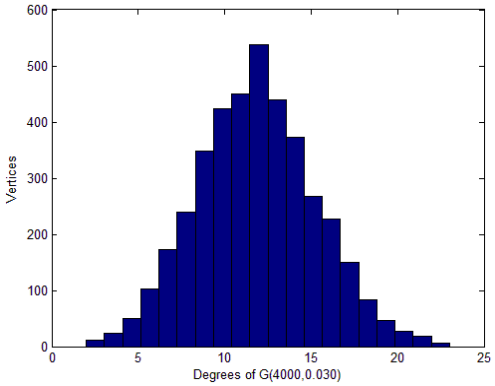

Fig. 2. Degree histogram of the geometrically connected graph of figure 1.

The Parallel Multilevel (PML) and Parallel Multilevel + Neural Networks (PML + PNN) heuristics were studied using the following geometrically connected graphs:

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

Name	Number of edges n	Number of arcs $\|E\|$	Radius r
G_{4000}	4000	13054	0.023
G_{6000}	6000	31671	0.024
G_{8000}	8000	39788	0.02
G_{10000}	10000	49817	0.018
G_{12000}	12000	61366	0.0165

The performance of the PML and PML+PNN heuristics were computed using the graphs of table 1 and applying the methodology that follows:

Step 0. For each graph of table 1, 50 random and balanced initial conditions were generated: y^0. The heuristic described below were executed using 5 processors.

Step 1. From each initial condition y^0, the PML heuristic was applied considering:

A threshold equals to 500 and 1000.
The Heavy Edge Matching for the coarsening phase.
The Kernighan-Lin heuristic for the phases of partitioning and uncoarsening with refinement, according to the results of refs. [1,11,12].
The best minimum cut *MinCut* was computed over all the solutions obtained from the initial conditions y^0.

Step 2. From each initial condition y^0, the PML+PNN heuristic was applied considering:

A threshold equals to 1000.
The Heavy Edge Matching for the coarsening phase.
The parallel Neural Network heuristic, see equation (10), for the phases of partitioning and uncoarsening with refinement.
The best minimum cut *MinCut* was computed over all the solutions obtained from the initial conditions y^0.

The PML+PNN heuristic differ from the standard PML in the phases of partitioning and uncoarsening with refinement. This new version of the PML is proposed in order to reduce the convergence times of the standard version applying a greedy type of heuristic. The numerical results that were obtained are summarized in following tables 2 and 3.

Table 2. Performance of the Parallel ML(500), Parallel ML(1000), Parallel ML+NN heuristics.

Heuristic	PML (500)		PML (1000)		PML + PNN	
Graph	*MinCut*	*Time* [*s*]	*MinCut*	*Time* [*s*]	*MinCut*	*Time* [*s*]
G_{4000}	935	5179	1021	4913	1925	4012
G_{6000}	1083	9741	1204	8085	2099	6975
G_{8000}	1601	20844	1682	17580	2430	16128
G_{10000}	1969	39449	2314	36430	2881	32003
G_{12000}	2852	81860	2891	77611	3791	74117

Table 3. Relative performance of the Parallel ML with respect to the Parallel ML+NN heuristics.

Graph	$d_{PML(500),PML+PNN}$	$T_{PML(500),PML+PNN}$	$d_{PML(1000),PML+PNN}$	$T_{PML(1000),PML+PNN}$
G_{4000}	0.51	0.29	0.47	0.22
G_{6000}	0.48	0.40	0.43	0.16
G_{8000}	0.34	0.29	0.31	0.09
G_{10000}	0.32	0.23	0.20	0.14
G_{12000}	0.25	0.10	0.24	0.05

where:

$$d_{PML(U),PML+PNN} = 1 - \frac{MinCut\ PML(U)}{MinCut\ PML + PNN}$$
$$T_{PML(U),PML+PNN} = \frac{Time\ PML(U)}{Time\ PML + PNN} - 1 \quad (12)$$

And U represents the threshold of PML.

In tables 2 and 3 can be observed that for medium-sized graphs with $n \in [4000,12000]$ vertices, the relative distance of the PML and PML+PNN heuristics MinCut has a decreasing linear tendency as the number of nodes is increased, with a quadratic computing time for both heuristics.

The best result of the PML+PNN heuristic is obtained for $n \in \{10000,12000\}$ where a performance comparable to PML(1000) is achieved with run time reduced at 14% and 5%, respectively.

Conclusions

The parallel ML+NN heuristic was studied for the Graph Bisection Problem defined in sparse graphs with local connectivity. The main result establishes that the performance of this heuristics increases linearly as n increases with respect to the parallel ML heuristic with a quadratic computing time. For $n \in \{10000,12000\}$ is obtained a performance comparable to PML(1000) with run time reduced at 14% and 5%, respectively.

As future work is proposed to improve the PML+PNN by implementing a hill climbing algorithm that allow to escape the high energy local minimum reached by the NN 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. Hendrickson, B., and R. Leland, A multilevel algorithm for partitioning graphs, Proc. Supercomputing 95, 1995.
8. Hopfield, J., Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences, V. 79, 2554--8, 1982.
9. Hopfield, J.J. and D.W. Tank, Neural computation of decisions in optimization problems, Biological Cybernetics V. 55, pp. 141-146, 1985.
10. Johnson, D.S., C. R. Aragon, L. A. Geoch and C. Schevon, Optimization by Simulated Annealing: An Experimental Evaluation; Part I, Graph Partitioning, Operations Research, Vol. 37, pp. 865 - 892, 1989.
11. 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.
12. Karypis, G., V. Kumar, A parallel algorithm for multilevel graph partitioning and sparse matrix ordering, Journal of Parallel and Distributed Computing, Vol. 48, Iss. 1, pp. 71-95, 1998.
13. 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.
14. Kernighan, B.W., and S. Lin, An Efficient Heuristic Procedure for Partitioning Graphs, The Bell System Tech. J., Vol. 49, pp. 291 - 307, 1970.

15. Lengauer, T., Combinatorial algorithms for integrated circuit layout, Wiley, New York, USA, 1990.
16. Papadimitriou, Ch.H. and S. Steiglitz, Combinatorial Optimization, Prentice-Hall, Englewood Cliffs, HJ, 1982.
17. Rolland, E., H. Pirkul, and F. Glover, A Tabu Search for Graph Partitioning, Annals of Operations Research, Metaheuristics in Combinatorial Optimization, Vol. 63, 1996.
18. Walshaw, C., Multilevel Refinement for Combinatorial Optimisation Problems, Annals of Operations Research, Vol. 131, pp. 325 - 372, 2004.
19. Wichlund, S., On multilevel circuit partitioning, Proceedings of the 1998 IEEE/ACM international conference on Computer-aided, United States, pp. 505 - 511, 1998.