



Optimizing partitions of percolating graphs

Stefan Boettcher^{a,b,*}

^aCNLS MS-B258, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

^bCenter for Theoretical Studies of Physical Systems, Clark Atlanta University, Atlanta, GA 30314, USA

Abstract

The partitioning of random graphs is investigated numerically using “simulated annealing” and “extremal optimization”. While generally in an NP-hard problem, it is shown that the optimization of the graph partitions is particularly difficult for sparse graphs with average connectivities near the percolation threshold. At this threshold, the relative error of “simulated annealing” is found to diverge in the thermodynamic limit. On the other hand, “extremal optimization”, a new general purpose method based on self-organized criticality, produces near-optimal partitions with bounded error at any low connectivity at a comparable computational cost. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Optimization; Graph partitioning; Percolation; Simulated annealing; Extremal optimization; Self-organized criticality

The partitioning of graphs is generally an NP-hard optimization problem with many practical applications such as VLSI design [1] and load-balancing between parallel processors [2]. In physics, it is most closely related to finding ground states of spin glasses [3,4]. The graph (bi-)partitioning problem is easy to formulate: Take N points where N is an even number, let any pair of two points be connected by an edge with a probability p , divide the points into two sets of equal size $N/2$ such that the number of edges connecting both sets, m , is minimal: $M = m_{\text{opt}}$. The global constraint of an equal division of the points between the sets places this problem generally among the hardest problems in combinatorial optimization, requiring a computational effort that would grow faster than any power of N to determine the exact solution with certainty. Fast heuristic methods, such as Kernighan–Lin [5], are moderately successful in finding some of the vast number of near-optimal configurations. In this paper we will focus

* E-mail: stb@cnls.lanl.gov.

instead on two more generally applicable optimization methods: simulated annealing [SA] [3,4] and extremal optimization (EO) [6].

The graph partitioning problem depends on the probability p with which any two points in the system are connected. Thus, p determines the total number of edges in an instance, $L = pN(N - 1)/2$ on average, and its mean connectivity per point, $C = p(N - 1)$ on average. In fact, graph partitioning has a lot of similarities with percolation: If the mean connectivity C is very small, the graph of N points consists of many disconnected, small clusters or single points which are easily enumerated and sorted into two equal partitions with no edges between them ($m_{\text{opt}} = 0$). If C is large ($p = O(1)$), almost all points are connected into one giant cluster and $m_{\text{opt}} = O(N^2)$. But when $p = O(1/N)$, say $1 < C < 10$, the distribution of cluster sizes is broad, and thus the partitioning problem is nontrivial. Obviously, as soon as a cluster of size $> N/2$ appears, $m_{\text{opt}} > 0$. Thus, we observe a percolation-like transition at a C_{crit} with the cutsizes m_{opt} as the order parameter. We will show that SA, based on equilibrium statistical mechanics, performs poorly near this critical point. By contrast, EO, a self-organized critical [7] optimization method, approximates m_{opt} far better for any C , and is about as fast as SA at least for $C < 10$.

In SA [3,4], we try to minimize a global cost function given by $f = m + \alpha(P_1 - P_2)^2$, where P_1 and P_2 are the number of points in the respective sets. Allowing the size of the sets to fluctuate is required to improve SA's performance in outcome and computational time at the cost of an arbitrary parameter α to be determined. Then, starting at a "temperature" T_0 , the annealing schedule proceeds with lN trial Monte Carlo steps on f by tentatively moving a randomly chosen point from one set to the other (which changes m) to equilibrate the system. After that, we set $T_i = T_{i-1}(1 - \varepsilon)$, equilibrate again for lN trials, and so on, until the MC acceptance rate drops below A_{stop} for K consecutive temperature levels. At this point the optimization process can be considered "frozen" and the configuration should be near-optimal, $m \approx m_{\text{opt}}$. While SA is intuitive, controlled, and of very general applicability, its performance in practice is strongly dependent on a multitude of parameters which have to be arduously tuned. For us it is thus expedient (and most unbiased!) to rely on an extensive study of SA for graph partitioning [8] which determined $\alpha = 0.05$, $T_0 = 2.5$, $\varepsilon = 0.04$, $A_{\text{stop}} = 2\%$, and $K = 5$. Ref. [8] set $l = 16$, but performance improved noticeably for our choice, $l = 32$.

EO has recently been proposed as an alternative optimization method [6], based on insights into nonequilibrium critical phenomena. Here, each point i obtains a "fitness" $\lambda_i = g_i/(g_i + b_i)$, where g_i and b_i are the number of "good" and "bad" edges that connect that point within its set and across the partition, resp. ($\lambda_i = 1$ for isolated points.) Note that $\sum_i b_i/2 = m$. At all times, an ordered list $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$ is maintained where λ_n is the fitness of the point with the n th rank in the list. At each update we choose two points $n = n_1$ and $n = n_2$ from the two different sets at random according to their rank with probability $P(n) \sim n^{-\tau}$. These two points swap sets *no matter what* the resulting new cutsizes m may be. Then, these two points, and all points they are connected to, re-evaluate their fitness λ . Finally, the ranked list of λ 's is reordered and

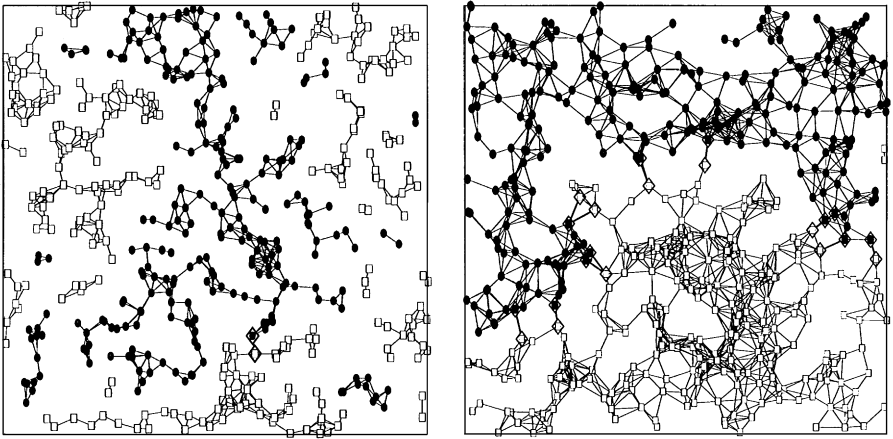


Fig. 1. Two random geometric graphs, $N = 500$, with $C = 4$ (left) and $C = 8$ (right) in an optimized configuration found by EO. At $C = 4$ the graph barely percolates, with merely one “bad” edge (thick line) connecting a set of 250 round points with a set of 250 square points, thus $m_{\text{opt}} = 1$. For the denser graph on the right, EO reduced the cutsizes to $m_{\text{opt}} = 13$ bad edges.

the process is started again. We repeat for about $400N$ update steps per run while storing the best result generated along the way. Note that no scales are introduced into the process, since – unlike in Monte Carlo – all moves are accepted. Instead of a global cost function, the rank-ordered list of fitnesses provides the information about optimal configurations. This information emerges in a self-organized manner merely by selecting with a bias *against* badly adapted points, instead of “breeding” better ones [9,10]. There is only one parameter, chosen to be $\tau = 1.5$ here, which fixes EO to run at the *edge of ergodicity* [6] and appears to be universal for graph partitioning, independent of N and C .

In our numerical simulations (for more data see Ref. [11]) we have generated “geometrical” graphs of mean connectivity C by dropping N points at random into the unit square where any two points are connected if their Euclidean distance d obeys $N\pi d^2 < C$. Two such graphs, already optimized, are shown in Fig. 1. We chose $N = 500, 1000$ and 2000 , and for each N we varied the mean connectivity between $C = 2$ and $C = 16$. Then, at each C we generated 20 different instances of graphs (identical for SA and EO) and performed 50 optimization runs for each method on that instance. Each run, we used a new random initial condition by arbitrarily dividing the points into two equal-sized sets.

Unable to determine m_{opt} for each graph exactly, we used the best m found in the 100 combined runs of EO and SA as our best cutsizes. (EO found such best m 's exclusively for $2 < C < 10$, and mostly for all other C 's.) We then average ten values from a best-of-five selection in that sequence of 50 runs and determine the relative error above the best cutsizes found for that graph. Finally, all errors from the 20 instances at a given C are averaged. In Fig. 2, we plot these average errors for EO and SA as a function of C for $N = 500$. We also plot the average best cutsizes of those 20 graphs

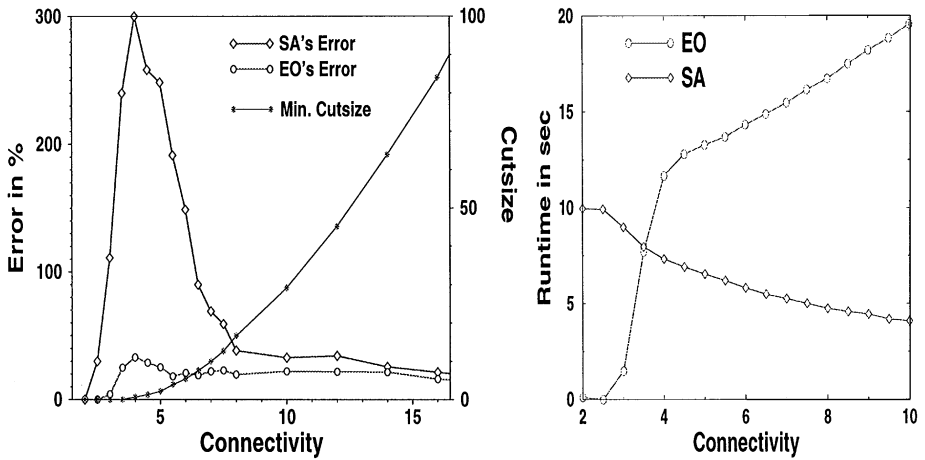


Fig. 2. Plot of the average error at $N = 1000$ for SA and EO as a function of the mean connectivity C , and of the average best cutsizes, on the left. On the right, a plot of the average runtime for SA and EO as a function of C .

for each C which rises above 0 for $C > C_{\text{crit}} \approx 4$. Exactly at C_{crit} , the error of SA is maximal, at 300%, while EO's error is less than 40% throughout. For $N = 1000$ and 2000, SA's error rises to 600% and 1100% while EO's error hardly changes. Also plotted are the averaged runtimes for SA and EO at each C , which are comparable for $C < 10$. (Due to re-sorting connected points, EO's runtime rises $\propto C$.)

I thank Allon Percus for very helpful discussions.

References

- [1] A.E. Dunlop, B.W. Kernighan, IEEE Trans. Comput. Aided Des. CAD-4 (1985) 92–98.
- [2] C.W. Ou, S. Ranka, IEEE Trans. Parallel Distributed Systems 8 (1997) 884–896.
- [3] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Science 220 (1983) 671–680.
- [4] V.J. Černý, Optim. Theory Appl. 45 (1985) 41.
- [5] B.W. Kernighan, S. Lin, Bell Systems Tech. J. 49 (1970) 291–307.
- [6] S. Boettcher, A.G. Percus, submitted.
- [7] P. Bak, C. Tang, K. Wiesenfeld, Phys. Rev. Lett. 59 (1987) 381–384.
- [8] D.S. Johnson, C.R. Aragon, L.A. McGeoch, C. Schevon, Oper. Res. 37 (1989) 865–892.
- [9] P. Bak, K. Sneppen, Phys. Rev. Lett. 71 (1993) 4083–4086.
- [10] S. Boettcher, Physica A 266 (1999) 330.
- [11] S. Boettcher, submitted.