Phase transition for cutting-plane approach to vertex-cover problem

Timo Dewenter[1] and Alexander K. Hartmann[1, *]

[1]*Institut für Physik, Universität Oldenburg, D-26111 Oldenburg, Germany*
(Dated: January 9, 2012)

We study the vertex-cover problem which is an NP-hard optimization problem and a prototypical model exhibiting phase transitions on random graphs, e.g., Erdős-Renyi (ER) random graphs. These phase transitions coincide with changes of the solution space structure, e.g, for the ER ensemble at connectivity $c = e \approx 2.7183$ from replica symmetric to replica-symmetry broken. For the vertex-cover problem, also the typical complexity of exact branch-and-bound algorithms, which proceed by exploring the landscape of feasible configurations, change close to this phase transition from "easy" to "hard". In this work, we consider an algorithm which has a completely different strategy: The problem is mapped onto a linear programming problem augmented by a cutting-plane approach, hence the algorithm operates in a space *outside* the space of feasible configurations until the final step, where a solution is found. Here we show that this type of algorithm also exhibits an "easy–hard" transition around $c = e$, which strongly indicates that the typical hardness of a problem is fundamental to the problem and not due to a specific representation of the problem.

NP-hard combinatorial optimization problems [1, 2] are fundamental to computational complexity, because despite much effort no algorithm has been found so far, which is able to solve these problems in the *worst case* in polynomial time, leading to the famous *P-NP problem*. One way to try to understand the root of the apparent computational hardness is to analyze hard instances of problems. This has attracted much interest in statistical physics [3–5]. Phase transitions on suitably chosen ensembles of random instances were found, e.g., for the Satisfiability Problem [6], the Traveling Salesman Problem [7] or the vertex-cover problem (VC) [8]. For exact branch-and-bound algorithms [9, 10] the hardest instances are found right at these phase transitions, often related to a change from a typically polynomially ("easy") to a typically exponentially ("hard") region. Branch-and-bound algorithms systematically explore the space of feasible solutions (branching) while trying to avoid uninteresting configurations via updating efficient bounds. The behavior of these exact algorithms can be partially understood in terms of an effective dynamics inside the phase diagrams [11, 12]. In practice very efficient but not exact are stochastic algorithms, e.g. Walk-SAT [13] or ASAT [14] and message-passing algorithms [15, 16], inspired by statistical mechanics methods like the cavity approach [17]. Also these types of algorithms rely on either moving in configuration space or on calculating iteratively probabilities (weights) for different subspaces of configurations.

Here, we consider a completely different and complementary type of algorithm, *linear programming* (LP), which is a standard approach for practical optimization problems [2]. In connection with *cutting planes* (CP) [18], it is a very efficient (but apparently still worst-case exponential) approach to combinatorial optimization problems. This approach is fundamentally different from the algorithms mentioned above since it does not move inside the configuration space but instead considers non-feasible (non-combinatorial) assignments to the variables which are always more optimal than the true feasible solution. Cutting planes are constraints which are added additionally and iteratively to the problem until a feasible solution is found, which is then the optimal solution. In particular we study the vertex cover problem via LP and CP for Erdős-Renyi random graphs [19]. We show that VC with our LP/CP implementation changes from "easy" to "hard" right at the same transition point, where this change occurs for a branch-and-bound algorithm, and where the solution landscape changes from simple (replica symmetric in the spin-glass language [17]) to complex (replica-symmetry broken). Hence, our results indicate that the typical hardness of a problem seems to be quite universal since the changes from "easy" to "hard" are visible for algorithms which are based on fundamentally different notions of configuration space.

Model Let $G = (V, E)$ be an undirected graph with N vertices $i \in V$ and M edges $\{i, j\} \in E$. A vertex cover $V_{\mathrm{VC}} \subset V$ is a subset of vertices so that for all edges $\{i, j\} \in E$ at least one end i or j is contained in V_{VC}. The vertices $i \in V_{\mathrm{VC}}$ are called *covered*, *uncovered* else. We are interested in vertex covers of G of minimum cardinality $|V_{\mathrm{VC}}|$, the *minimum vertex covers*. The decision problem if a VC with fixed cardinality exists or not belongs to the class of NP-complete problems [1].

The analytical solution of VC on Erdős-Renyi graphs exhibits a phase transition at the average connectivity $c = e \approx 2.7183$: for $c < e$, the solution is replica symmetric, while for $c > e$ replica symmetry breaking was found [8]. This can be seen also numerically when clustering the minimum vertex covers [20]. Furthermore, in connection with the *leaf-removal heuristic* [21], the typical-case complexity of a branch-and-bound algorithm changes form "easy" to "hard" at $c = e$.

Linear-Programming Approach First, we translate the VC problem to an integer linear programming (ILP) problem [2], each of the N nodes of the graph is rep-

resented by a variable $x_i \in \{0, 1\}$, $i = 1, \ldots, N$. The value $x_i = 1$ denotes a covered, $x_i = 0$ indicates an uncovered node. The fact that for each edge $\{i, j\}$ i or j must be covered can be written as $x_i + x_j \geq 1$. Minimizing the cardinality of the cover means we want to minimize $\sum_i x_i$. When we relax the integer constraint to $x_i \in [0, 1]$, the set of constraints $x_i + x_j \geq 1$ describes a *polytope*. Now we obtain the following linear programming problem (LP):

Minimize $\quad x = \sum_{i=1}^{N} x_i$

Subject to $\quad 0 \leq x_i \leq 1 \; \forall \, i \in V$

$\qquad\qquad\quad x_i + x_j \geq 1 \quad \forall \, \{i, j\} \in E$

This can be solved efficiently, i.e., typically in polynomial time, by the simplex algorithm (SX) [2, 22]. We used the public available `lp_solve` [23] with Bland's *first index pivoting* [24]. Note that now the solutions are not guaranteed to be integer-valued any more, variables with $x_i \in \,]0, 1[$ we call *undecided*. Such solutions we call *incomplete*. The value of $x = \sum_i x_i$ is always a lower bound for the cardinality of a complete solution. On the other hand, if a solution computed by SX is *complete*, i.e., all variables are integer-valued, then it is immediately clear that it is a correct minimum for VC.

Cutting-Plane Approach In order to obtain more complete solutions, the CP approach [18] can be used. The basic idea is to limit the solution space by adding extra constraints, which exclude incomplete solutions. In principle, many types of extra constraints are possible. Here, we apply the following heuristics, inspired by the nature of the problem: For any graph which is a cycle, setting all variables of non-isolated nodes to 0.5 (0 else) is a solution of the LP, but incomplete. Nevertheless, for a cycle of odd length $l = 2k + 1$, the size of the minimum cover is always $k + 1 > k + 0.5$. Hence, for any cycle of length $l = 2k + 1$ in a graph, at least $k + 1$ nodes of the cycle must be covered. Thus, after an execution of SX, if the solution is incomplete, we try to detect cycles of odd length l where the condition

$$\sum_{x_i \in \text{loop}} x_i \geq \left\lceil \frac{l}{2} \right\rceil \tag{1}$$

is violated ($\lceil l/2 \rceil$ is the largest integer larger or equal to $l/2$) and add this constraint to the LP. Technically, the loops are obtained by searching a random spanning tree (ST) in the graph via a random breadth-first search and adding a randomly chosen edge, which is part of the graph but not of the ST. Our algorithm stops, if $s = 20M$ times for a randomly chosen spanning trees and for all loops emerging from these trees we did not add a constraint (because the loop was of even length or the constraint was already fulfilled by the current incomplete solution). Otherwise, SX is executed for the next time and the solution checked for completeness again.

In general, SX guarantees to obtain a solution in a corner of the polytope with minimum x. This means that still non-integer solutions can be obtained, how frequently depends also on the heuristics used in the actual SX implementation. Anyway, an incomplete solution obtained by CP+SX provides another lower bound x, usually better (but never worse) than that obtained by SX alone.

Node heuristics To complete an incomplete solution, we also applied the following "node" heuristics (NH): It randomly selects a vertex i with an undecided variable $x_i \in \,]0, 1[$ and adds $x_i = 0$ to the LP and solves it again. This forces the SX algorithm to set nodes j adjacent to i to $x_j = 1$. After each run of the SX algorithm we checked whether still undecided variables are found and if necessary the procedure is repeated. This ensures that finally a complete solution, i.e., a vertex cover is found, but it doesn't have to be a minimum one. Hence the values of x obtained in this way are upper bounds to the true minimum vertex covers. Note that we also tried a heuristics where $x_j = 1$ is added to the LP, but it provided typically higher values of x as solution, in particular for large graph connectivity $c > 5$.

Results Next, simulation results for the different types of algorithms are presented for ER random graphs of N nodes, up to $N = 280$. We used the ensemble where for each graph M edges are created randomly with uniform weight, i.e., the connectivity is $c = 2M/N$.

Fig. 1 shows the fraction p_f of graphs which exhibit a complete solution, for the SX (inset) and SX+CP algorithms, obtained from averaging over 1000 realizations of graphs and for different system sizes, respectively. Apparently, SX is able to find solutions up to about $c = 1$, where a sharp drop of p_f is visible, resembling a phase transition. Note that the percolation transition of the ER ensemble is at $c = 1$. For $c < 1$, ER random graphs consist mainly of trees, which are apparently easy to solve, even for SX. When including CP, more samples can be solved, the transition shifts to a point close to the $c = e$, where replica symmetry breaking occurs and where the exact configuration-space-based branch-and-bound algorithm (with leaf removal [21]) starts to exhibit a typically exponential running time.

An indicator for the running time of the SX+CP algorithm is the average number of extra constraints M_{extra} that were added to the LP resulting from CPs (1) to obtain complete solutions. Fig. 2 shows M_{extra}/N as a function of connectivity c. Clearly, an increase close to $c = e$ is visible. Note that M_{extra}/N increases also for all realizations (see inset), but in this case this is less informative, since the algorithm is stopped if for some time no new constraint could be added.

Finally, Fig. 3 shows the phase diagram for the VC problem. In addition to the exact minimum VC and the analytical solution [8], simulation results for the SX and the SX+CP algorithm (lower bounds) and a combination of SX/SX+CP with NH (upper bounds) are included.
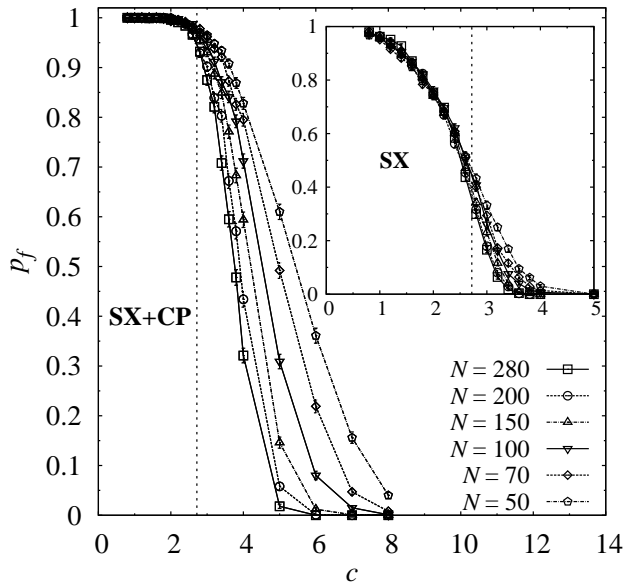
FIG. 1: Fraction p_f of complete solutions for CP approach as a function of connectivity c for ER random graphs. The inset shows p_f for the SX algorithm. The vertical line denotes $c = e$, the other lines are guides to the eyes only.
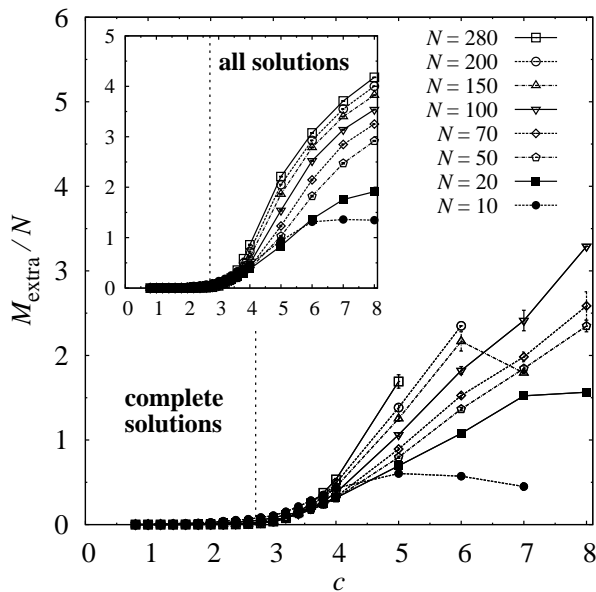

FIG. 2: Average number of extra constraints M_{extra} per node for CP algorithm as a function of c only for complete solutions and for all solutions (inset). The vertical line denotes $c = e$, the other lines are guides to the eyes only.

These bounds were obtained, respectively by averaging the "cover size" x for different system sizes N and different connectivities c, yielding $x(N, c)$. We extrapolated to infinite system sizes $x(c) = \lim_{N \to \infty} x(N, c)$ via fitting the data to functions $x(N, c) = x(c) + a \, N^{-b}$ (see inset of figure 3) or $x(N, c) = [x_c(c) + a \, N^{-b}] \cdot [1 + f \, N^{-g}]$. The latter was only used for the SX+CP/SX+CP+NH ap-

proaches for $c \geq 5$, where apparently stronger finite-size corrections occur. We found, e.g., for SX+CP at $c = 4$ a value $b = 0.88(9)$ which is compatible with the scaling of the exact value obtained by branch-and-bound [25].
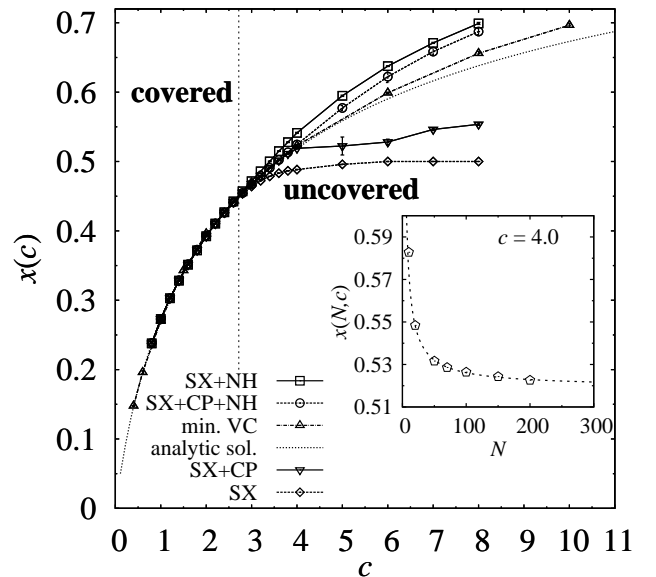

FIG. 3: Phase diagram for the fraction of covered vertices x. Minimum VC found with exact Branch-and-Bound algorithm/analytics [8] as well as upper and lower bounds obtained from the different algorithms. Inset: Finite-size scaling for SX+CP and $c = 4.0$ The vertical line denotes $c = e$, dotted line in the inset a result from the fit $(0.52 + 0.40 N^{-0.88})$ and the other lines are guides to the eyes only.

The SX algorithm alone only yields results close to the minimum VC up to $c = e$. Above this value the critical fraction of covered vertices converges towards the trivial solution $x_c = 0.5$. The SX+CP approach results in a better lower bound and deviates only for an average degree $c \geq 5$ visibly from the true minimum cover sizes. Comparing with the previous results, this means for $c \gtrsim e$, this approach typically does not yield the correct solution but comes very close to it. The upper bounds also start to deviate significantly for $c > e$, but stay rather close. In general, one sees again the importance of the critical line $c = e$: For smaller connectivities c all bounds seem to agree but beyond it they start to diverge, which is in contrast to previous analytical bounds [3], which do not match the correct result for all connectivities $c > 0$.

Conclusion/Outlook We studied the vertex-cover problem for Erdős-Renyi random graphs with a linear programming/cutting plane algorithm. The algorithm shows a clear "easy–hard" signature close to the connectivity $c = e$. This measn that this point denotes a phase transition not only for configuration-space-based quantities and algorithms [8, 21] but also for the LP/CP approach which operates outside the space of feasible solutions. Thus, the typical hardness of VC is really an

intrinsic property of the problem and not bounded to specific algorithms. This finding may be related to the fact that also in the worst-case, all algorithms known for NP-hard problems exhibit an exponential running time, i.e. it could help in order to understand better the P-NP problem.

In principle, the number of loops grows exponentially with the number of nodes. Hence, one could imagine that even within our SX+CP approach, by exhausting the set of these loop constraints, one arrives at an (exponentially slow) but *complete* algorithm. Nevertheless, there are graphs, where our constraints are clearly insufficient: A simple example is a complete graph of size $N = 4$, i.e., where each node is connected to any other node. The result of SX will be to set all $x_i = 0.5$, i.e., $x = 2$. By adding constraints for the four possible loops of odd length, the result will be $x_i = 2/3$ for all nodes, i.e., $x = 8/3$, while the correct minimum-cover size is $x = N - 1 = 3$.

One could improve the algorithm in principle by adding other types of constraints, e.g., general *Gomory-Chvátal cuts* [18]. Alternatively, one could consider small subgraphs $G' = (V', E')$, $V' \subset V$ and $E' \subset E$, solve them by an exact algorithm yielding the cardinality X' of the minimum cover. Then one could add the constraint $\sum_{i \in V'} x_i \geq X'$. We have performed some preliminary experiments with these types of cuts, but observed only marginal improvements so far, i.e., the overall behavior with the transition close to $c = e$ was preserved.

In practice often a combination of the branching, i.e. configuration-space based, and cutting, i.e., LP-based approaches, are used. Here, branching sets in when all available cutting planes are exhausted. Thus, it would be very interesting to see how this combination of approaches performs on VC for ER graphs.

Furthermore, it would be of high interest to analytically analyze the cutting plane approach, to see whether one can understand why it performs so well for $c < e$. This would lead to a better understanding of the roots of computational hardness and could also lead to refined bounds on the minimum cover sizes, providing techniques applicable to a vast range of problems.

Finally, it could be worthwhile to extend the present study to other graph ensembles. One could test whether at the same point replica symmetry breaking occurs, hierarchical clustering of solution space can be found, and the problem becomes hard for configuration-space-based branch-and-bound approaches as well as for the LP-based cutting-plane algorithms.

[1] M. R. Garey and D. S. Johnson, *Computers and intractability* (W.H. Freemann, San Francisco, 1979).
[2] C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization – Algorithms and Complexity* (Dover Publications Inc., Mineola, NY, 1998).
[3] A. K. Hartmann and M. Weigt, *Phase Transitions in Combinatorial Optimization Problems* (Wiley-VCH, Weinheim, 2005).
[4] M. Mézard and A. Montanari, *Information, Physics and Computation* (Oxford University Press, Oxford, 2009).
[5] C. Moore and S. Mertens, *The Nature of Computation* (Oxford University Press, Oxford, 2011).
[6] S. Kirkpatrick and B. Selman, Science **264**, 1297 (1994).
[7] I. P. Gent and T. Walsh, in *Proceedings of 12th European Conference on Artificial Intelligence. ECAI '96* (Wiley, Chichester, 1996), p. 170.
[8] M. Weigt and A. K. Hartmann, Phys. Rev. Lett. **84**, 6118 (2000).
[9] M. Davis, G. Logemann, and D. Loveland, Commun. ACM **5**, 394 (1962), ISSN 0001-0782, URL http://doi.acm.org/10.1145/368273.368557.
[10] R. Sedgewick, *Algorithms in C* (Addison-Wesley, Reading (MA), 1990).
[11] M. Weigt and A. K. Hartmann, Phys. Rev. Lett. **86**, 1658 (2001).
[12] S. Cocco and R. Monasson, Phys. Rev. Lett. **86**, 1654 (2001).
[13] C. H. Papadimitriou, in *Foundations of Computer Science, Annual IEEE Symposium on* (IEEE Computer Society, Los Alamitos, CA, USA, 1991), vol. 0, pp. 163–169, ISBN 0-8186-2445-0.
[14] J. Ardelius and E. Aurell, Phys. Rev. E **74**, 037702 (2006).
[15] M. Mézard, G. Parisi, and R. Zecchina, Science **297**, 812 (2002).
[16] M. Mézard and R. Zecchina, Phys. Rev. E **66**, 056126 (2002).
[17] M. Mézard, G. Parisi, and M. Virasoro, *Spin glass theory and beyond* (World Scientific, Singapore, 1987).
[18] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schriever, *Combinatorial Optimization* (Wiley, New York, 1998).
[19] P. Erdős and A. Rényi, Publ. Math. Inst. Hungar. Acad. Sci. **5**, 17 (1960).
[20] W. Barthel and A. K. Hartmann, Phys. Rev. E **70**, 066120 (2004).
[21] M. Bauer and O. Golinelli, Eur. Phys. J. B **24**, 339 (2001).
[22] G. Dantzig, Bull. Amer. Math. Soc. **54**, 1074 (1948).
[23] M. Berkelaar, K. Eikland, and P. Notebaert, *Open-source program to solve linear programming problems: lp_solve*, http://lpsolve.sourceforge.net/5.5/ (2010), online; Licence: GNU LGPL (Lesser General Public Licence).
[24] R. Bland, Mathematics of Operations Research **2** (1977).
[25] A. K. Hartmann, W. Barthel, and M. Weigt, Comp. Phys. Comm. **169**, 234 (2005).