Links between complex networks and combinatorial optimization

Reti complesse e ottimizzazione combinatoria

Andrea Roli

SOMMARIO/*ABSTRACT*

Recenti risultati nell'ambito dei problemi di soddisfacimento di vincoli e di ottimizzazione combinatoria mostrano che i grafi possono essere utilmente applicati come modelli per la struttura dei problemi. In primo luogo, risultati e tecniche derivanti dalle ricerche sui grafi (in particolare gli studi sulle cosiddette "reti complesse") permettono di studiare e comprendere più approfonditamente il comportamento degli algoritmi. Inoltre, alcune caratteristiche dei grafi utilizzate come modelli per la struttura di istanze possono essere utilizzate per progettare euristiche più efficaci e per definire criteri per la selezione dell'algoritmo più opportuno. In questo lavoro saranno presentati alcuni dei lavori più rappresentativi in questo settore.

Recent results in constraint satisfaction and combinatorial optimization have shown that complex networks (such as scale-free networks, characterized by a frequency node degree following a power law) can be fruitfully used to modeling problem structure. By conveying results and tools from complex networks to combinatorial optimization, it is possible to achieve a deeper understanding of algorithm behavior. Moreover, some features of the network that models an instance can guide the design of specific heuristics and enable to choose the best solver among a portfolio of algorithms. This work describes two representative cases in this interdisciplinary research field.

Keywords: Combinatorial optimization problems, constraint graph, problem structure, complex networks

1 Introduction

Research on complex graphs (or, equivalently, networks) is nowadays spreading to diverse fields, including biology and genetics, information technology and economics [3, 4, 42]. Important results have been achieved on the foundations of complex networks [1] and relevant are the applications in which these results have impact, such as in computer networks, like the Internet and the World Wide Web [2].

Concepts and results stemming from complex networks have also been recently introduced in the field of combinatorial optimization, in which graphs enable researchers to capture and model some aspects of the structure of a combinatorial optimization problem instance. By bridging together complex networks and combinatorial optimization, this approach has produced a break-through in the field, leading to a deeper understanding of the behavior of algorithms and the design of more effective solvers.

This contribution aims at providing a brief description of two examples of the cross fertilization between complex networks and combinatorial optimization, besides a succinct overview of the main topics of the field. In Section 2, we introduce the main notions and results concerning complex networks. Section 3 provides a description of the use of complex networks to modeling problem structure and in Section 4 we describe two applications which enlighten the usefulness of modeling combinatorial problems with graphs.

2 Complex networks

Graphs are the basic structure for a large number of phenomena and problems, from physics to economics, from computer science to social sciences. Many problems can be formulated in terms of networks which describe the relations among components and this abstraction enables us to characterize some general properties of the problem or the system.

Among the most famous examples, we mention *scale-free* networks [4, 3], in which few nodes (the *hubs*) have a huge number of links to other nodes,

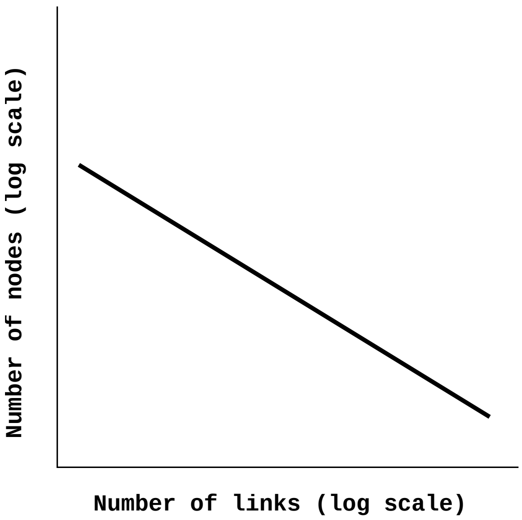

Figure 1: Power law distribution of node links.

while many nodes have just a few links. Such networks are characterized by a node degree distribution that follows a power law: The probability $p(k)$ of finding a node with exactly k links is $p(k) \sim k^{-\alpha}$ (see Fig.1). Important systems such as power networks and the World Wide Web have such a property. The topology of scale-free networks makes them robust against accidental failures, but extremely sensitive to coordinated attacks.

Besides scale-free networks, we also mention *small-world* networks [43, 42], that reached the popularity with the 1990's movie "Six degrees of separation". Beyond the popularity, small-world is a very important phenomenon emerging in very different and apparently distant fields. Small-world graphs, such as friendship networks, have low characteristic path length and high clustering. Given a graph $G = (V, E)$, where V is the set of nodes and E the set of edges, the characteristic path length can be informally defined as the average path length between any pair of nodes. The clustering coefficient γ of a graph G quantifies the probability that, given node v_1 connected to v_2 and v_3, there is an edge between v_2 and v_3. For instance, friendship relations are characterized by a high value of γ. To a certain extent, the clustering quantifies the cliquishness of a graph. A typical example of a small-world graph is the one related to film actors, constructed by connecting two nodes if the corresponding actors played in a same movie.

3 Networks to model structure in COPs

Combinatorial Optimization Problems (hereinafter referred to as COPs) can be defined by:

- a set of variables, $X = \{x_1, \ldots, x_n\}$;
- variable domains, $D_i \ \forall i = 1, \ldots, n$, i.e., values that variables can assume;
- constraints among variables, $c_j(Y_j)$, $Y_j \subseteq X$, $\forall j = 1, \ldots, m$ limiting the combinations of feasible assignments;
- an *objective function* to be optimized (minimized or maximized).

The set of all possible feasible assignments is $\mathcal{S} = \{s = \{(x_1, v_1), \ldots, (x_n, v_n)\} \mid v_i \in D_i, s \text{ satisfies all the constraints}\}$. $\mathcal{S}$ is usually called a *search* (or *solution*) *space*, since each element of the set can be seen as a candidate solution. To solve a COP one has to find a solution $s^* \in \mathcal{S}$ with minimal objective function value[1], that is, $f(s^*) \leq f(s) \, \forall s \in S$. An important class of combinatorial problems, strictly related to COPs, is the one of Constraint Satisfaction Problems (CSPs). They can be seen as a special case of COPs with constant objective function, as they are solved by any feasible solution.

Due to the practical importance of COPs and CSPs, many algorithms to tackle them have been developed. These algorithms can be divided in two classes: *complete* and *approximate* algorithms. Complete algorithms are guaranteed to find an optimal solution in bounded time for any finite size instance [24]. In the worst case, complete methods need exponential computation time to solve COPs that are $\mathcal{NP}$-hard [9]. (At least, as far as we suppose that $\mathcal{P} \neq \mathcal{NP}$.) These algorithms can also require a too long runtime for practical purposes. Therefore, for many real-world problems, researchers resort to approximate methods, that are usually very efficient in finding (near-)optimal solutions while sacrificing the guarantee of proving optimality. Among the most important complete algorithms to tackle COPs and CSPs, we mention branch and bound and constraint programming [24, 38]. On the other side, metaheuristics [7] are a class of approximate search strategies very commonly used. The current research trend in this field goes into the direction of integration of techniques [22] and of the exploitation of problem structural properties for both designing effective heuristics and choosing the most suitable solver for a given instance.

The definition of structure emerging from the literature on CSPs and COPs is usually based on the informal notion of a property enjoyed by non-random problems. Thus, the term *structured* should characterize instances derived from real-world problems. Commonly, we say structured for a problem that shows, under some abstraction, regularities such as well defined

[1] In this work we deal, without loss of generality, with minimization problems.

subproblems, patterns or correlations among problem variables.

The impact of problem structure on search performance has been studied from different perspectives. Studies on the impact of problem structure on heuristic search can be found, for example, in [5, 44, 41, 19]. Important results and observations on structure and problem hardness are reported in [13, 12, 11] and the effects of problem encoding are discussed in [15, 6]. The estimation of algorithm performance by analyzing instance structural properties can also be effectively exploited to choose the best available solver among a portfolio. Notable examples are the works by Leyton-Brown et al. [18] and Guerri and Milano [14].

Some problems suggest a natural structural description, since they have a representation suitable for structure analysis. A classical example are problems defined on graphs, such as the graph coloring problem and the k-cardinality tree problem. In general, for COPs and CSPs a constraint graph can be defined, in which nodes correspond to variables and edges connect two variables if there exists a constraint involving them. In the case of non binary constraints, it is also common to define a hypergraph, in which hyper-edges can connect more than two nodes.[2]

The definition of a graph that captures some features of problem structure is not unique. For example, let us consider instances of the satisfiability problem (SAT). SAT can be stated as follows: Given a set of clauses, each of which is the logical disjunction of $k > 2$ *literals* (a literal is a boolean variable or its negation), we ask whether a truth assignment to the variables exists that satisfies all the clauses. For SAT instances the *interaction graph* [25] can be defined, in which nodes correspond to variables and an edge connects two nodes if the corresponding variables appear in the same clause (see Fig. 2). Also a bipartite graph can be defined that represents both literals and clauses as nodes and edges connect literals to the clauses involving them.

The shift from a problem model to its representation via graphs enables to relate graph properties and characteristics with structural properties of problem instances. Hence, the connection between complex networks and COPs becomes a preferential way for capturing and exploiting problem structure. In the following section, we will discuss two examples among the applications in this area.

4 Applications

The analysis of complex networks derived by combinatorial problems is a quite recent research area. A fundamental issue to address for effectively attacking real-world COPs is the identification of the characteristics of an instance that make it difficult to be solved and the exploitation of this information to design effective heuristics tailored to a particular structure. Furthermore, results in this direction can also lead to the construction of well-defined benchmarks to test and compare algorithms. These issues can be considered as branches of the empirical and theoretical analysis of algorithm behavior.

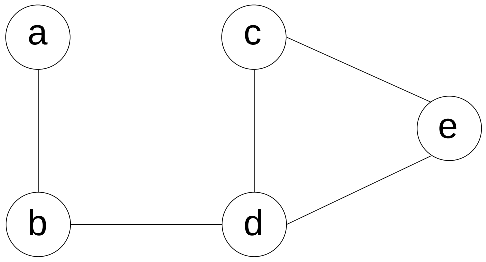

Figure 2: Interaction graph associated to the SAT instance $(a \vee \neg b) \wedge (b \vee d) \wedge (c \vee \neg d \vee \neg e)$. Note that one interaction graph corresponds to a set of formulas, therefore it just represents essential interactions among variables.

In the following, we will describe two representative cases of this research area. The former is a study on the relation between node degree distribution and multi-move local search behavior. The latter concerns the impact of small-world topology on complete and approximate algorithms efficiency.

In the next sections, we will refer to as problem *hardness* with the meaning of execution time or search cost (e.g., number of iterations) required to return a solution to a problem.[3]

4.1 Node degree distribution and multi-move local search

The impact of node degree on local search performance is one of the facets of the studies that try to convey results from complex networks to combinatorial optimization. This research line has its roots in a phenomenon called *criticality and parallelism*, first observed in [21] and further studied [26, 34, 29, 27, 31].

The criticality and parallelism phenomenon has been observed in the context of local search algorithms applied to combinatorial optimization problems [21, 17, 16]. Local search strategies start from a candidate solution and iteratively perform small changes to it with the goal to eventually find a (near-)optimal solution. In [21, 17, 16], local search is mod-

[2] In this work we do not focus on relationships between constraint graph properties and special cases of algorithm complexity, as discussed for instance in [8, 35].

[3] The concept of hardness will be related to the typical-case complexity and not to the worst-case.

ified by applying some local moves in parallel.[4]

It has been shown that the effectiveness of these algorithms depends on the parallelism degree τ (number of simultaneous moves): if τ increases, the solution quality also increases up to a maximal point (corresponding to τ_{opt}) at which it starts to decrease. It has also been shown that τ_{opt} is negatively correlated with the average node degree: the higher the degree, the lower τ_{opt}. This result can be explained by observing that the node degree gives a rough estimation of the level of constraint tightness in an instance: The higher the average node degree is, the more tightly the variables are connected. In [21, 17, 16] it is also shown that the optimal parallelism value is associated to a phase transition, hence the term criticality.

A phenomenon with analogous characteristics has been discovered in parallel local search for SAT [26, 34, 29, 27, 31], in which a parallel version of local search has been applied to both random and structured instances of SAT and its optimization version MAXSAT. In the MAXSAT problem, the goal is to minimize the number of unsatisfied clauses, or a weighted sum of them.

The algorithm used in those studies is a variant of GSAT [36], a greedy local search designed for attacking SAT.[5] GSAT starts from a random assignment and looks for a better one by moving from one state to another one in its neighborhood (defined as the set of states at Hamming distance equal to 1). Given a current state, the next state is chosen by *flipping* the variable that, if flipped, leads to the greatest gradient in the number of satisfied clauses. This algorithm can be easily extended to a multi-move local search by dividing the variables in τ subsets and flipping one variable of each subset according to the GSAT flip criterion. The effect of a variable flip on the objective function value is evaluated as if it were the only one to change, so flips are taken as if they were performed in parallel.

For random instances, it has been experimentally shown that the best performance is achieved with an optimal parallelism degree τ_{opt} that is monotonically non increasing with the average node degree of the graph associated to the instance attacked. A typical example is depicted in Fig.3. The plot shows the average best solution (evaluated as the number of unsatisfied clauses) returned by Parallel GSAT (PGSAT) run with varying number of parallel flips (from 1 to 20 variable flips per move). We can observe that the average best solution curve has a minimum corresponding to a

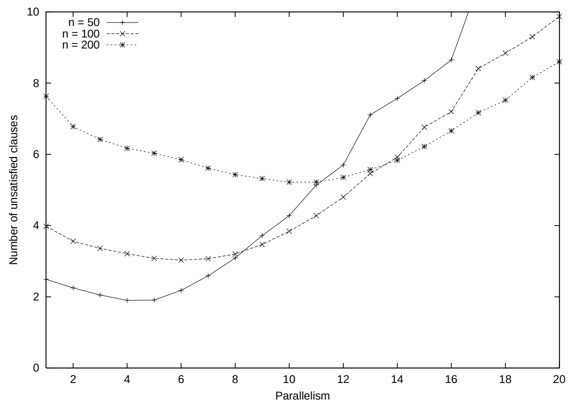

Figure 3: Average error (number of unsatisfied clauses) vs. τ. Random MAXSAT instances with 50,100,200 variables and 218,430,860 clauses respectively. Results are averaged over 10000 runs.

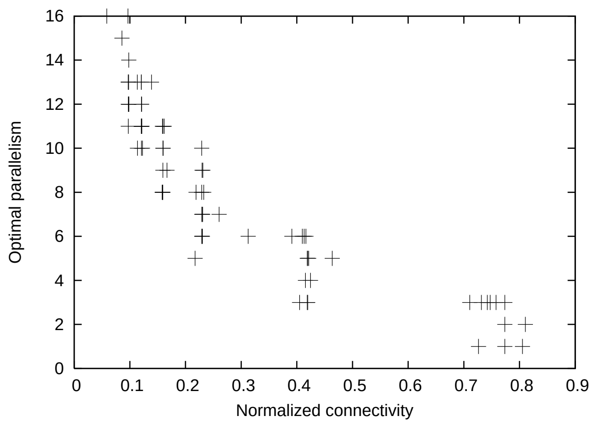

Figure 4: τ_{opt} vs. normalized average node degree.

particular value of τ, that is instance dependent. The instance property that affects τ_{opt} is the average node degree of the graph associated to the boolean formula. Fig.4 reports the plot capturing the empirical relation between average node degree and optimal parallelism. As we can note, τ_{opt} is monotonically non-increasing with the average "connectivity" among nodes.

The analysis of the node degree frequency of the graphs associated with SAT/MAXSAT instances is also particularly informative. The node degree frequency of the random SAT instances is regular (it approximately fits the Gaussian distribution) and the highest peaks are distributed around the mean (see Fig.5). Therefore, the average node degree can be used as a key parameter for relating instance and τ_{opt}.

Conversely, structured instances, deriving from real-world problems such as circuit testing or planning, have in general a more spread and non-uniform degree frequency. For instance, Fig.6 shows the node

[4] The considered algorithms have been implemented sequentially. "Parallel moves" are used with the meaning of "synchronous moves". Anyhow, these results could be beneficial also for implementations on parallel architectures.

[5] GSAT has been chosen since the previous results on criticality and parallelism were obtained by applying a (quite simple) local search characterized by a strong hill climbing tendency.

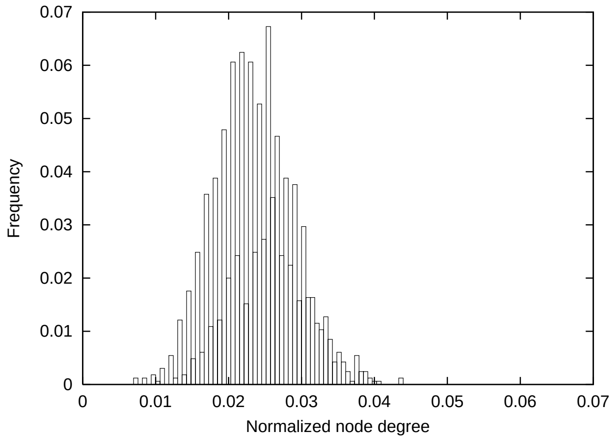

Figure 5: Frequency vs. normalized node degree in a random 3SAT instance with 1650 variables. The normalized average node degree is 0.0239.

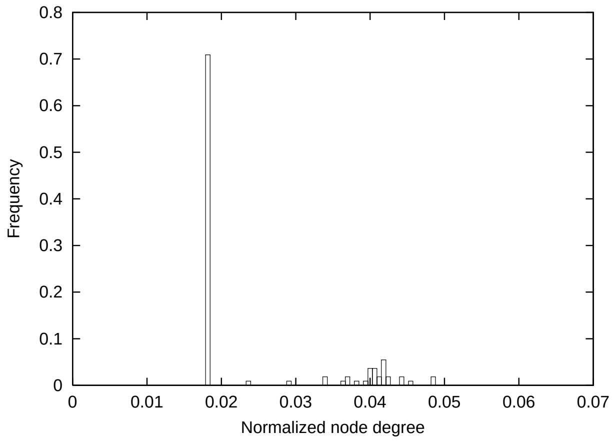

Figure 6: Frequency vs. normalized node degree in the instance *ii16a1* that encodes an inductive inference problem. The instance has 1650 variables and. The normalized average node degree is 0.0239.

degree frequency of a SAT instance encoding an inductive inference problem.[6] This asymmetric and non-Gaussian distribution strongly affects the algorithm performance as a function of τ. In particular, the location of the highest peaks in the frequency turns out to be the most relevant characteristic influencing the optimal parallelism τ_{opt}, i.e., the degree corresponding to the highest peaks is the parameter that mainly affects the actual value of τ_{opt}. For example, we report the case of the behavior of PGSAT on the aforementioned instances, which have the same average normalized node degree $\overline{q} = 0.0239$, but different node degree frequency. Fig.7 and Fig.8 show the average and median error (number of unsatisfied clauses) respectively on the structured instance (*ii16a1*) and the

[6] Instance retrieved from www.satlib.org

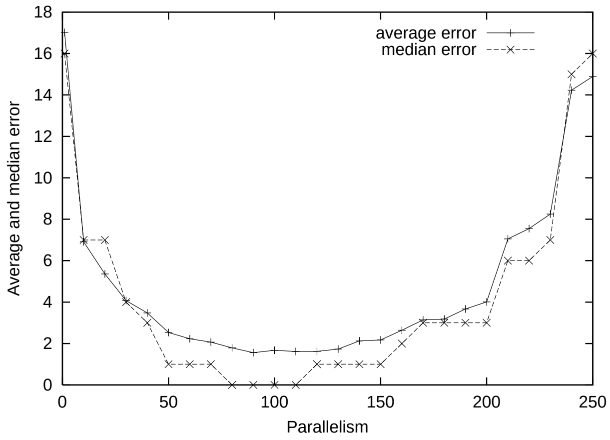

Figure 7: Average and median error against τ for PGSAT on the instance *ii16a1*.

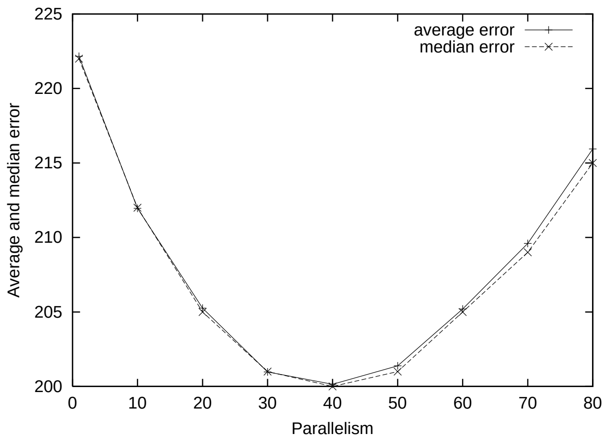

Figure 8: Average and median error against τ for PGSAT on the instance *3sat1650*.

random one (*3sat1650_q-const*) for PGSAT with different values of τ. Results are averaged over 500 trials. We observe that also for the structured instance there exists an optimal value of τ. Nevertheless, despite the fact that the two instances have the same average connectivity, the optimal parallelism is higher for *ii16a1* than for *3sat1650_q-const*. *ii16a1* has, indeed, a high peak close to $0.018 < \overline{q}$.

The main practical contribution of these studies is that the node degree of the interaction graph can be used as a heuristic criterion to optimally tune the number of parallel moves. For example, this criterion has been successfully applied in [28, 30] to improve the performance of a local search method for MAXSAT. There are still some open issues in this research; for instance, a formal relation between degree frequency and τ_{opt} that generalizes the criticality and parallelism phenomenon is still missing.

4.2 Small-world topology vs. problem hardness

One of the most representative studies that connect network topology and problem hardness is due to Walsh in [39], in which the author studies the possible correlation between small-world topology and problem hardness. Small-world graphs [42, 43] are characterized by low characteristic path length and high clustering. Walsh observes that different CSPs and COPs, such as instances of graph coloring, have a small-world topology. To solve the graph coloring problem, one has to find a color assignment to the nodes such that no adjacent nodes has the same color and the total number of colors used has to be minimized. The instances analyzed in [39], derived from register allocation problems, reveal a small-world structure. This characteristic makes them particularly hard to solve when using a complete search strategy driven by a heuristic that tries to exploit local information to estimate global properties. Indeed, Walsh conjectures that local properties, such as the clustering, can be bad predictors for global properties, such as the path length and thus mislead the search. This conjecture is supported by empirical evidence and it has also been found that graph coloring instances with small-world topology show heavy-tail behavior [12], characterized by a runtime distribution with a power-low decay of the tail. This study is further investigated and refined in [40], where it is shown that also the node degree has to be taken into account, in order to observe a correlation between hardness and small-world topology. Indeed, instances with small-world topology with scale-free node degree distribution appear to be very easily solved by a complete algorithm. This could be explained by observing that scale-free graph instances have small *backbones*, i.e., components of the solution which are present in all the solutions [23]. A small backbone reduces the probability of wrong choices at the beginning of the search, therefore the heavy-tail behavior —typical of early wrong choices difficult to recover from— does not arise.

A study on the correlation between SAT instance hardness and small-world topology has been undertaken in [30, 31]. The case examined considers instances of SAT that are characterized by an interaction graph with small-world topology. In [30, 32] a model is proposed to generate small-world SAT instances, by interpolating between regular lattice instances and random ones. This procedure is also quite similar to the one used in [43] to generate small-world graphs by interpolating between lattice and random graphs (see Fig.9). Starting from a lattice graph, links are randomly removed and rewired. Small-world graphs can be obtained by randomly rewiring just a few links between nodes. Each instance is obtained by taking from a lattice SAT instance all the clauses except for a prefixed number which are randomly chosen from a random SAT instance with the same number of variables and clauses. Small-world instances occur approximately when the number of random clauses inserted is the $15 \div 20\%$ of the whole number of clauses. For those values the proximity ratio, that measures how close a graph is to the small-world topology, is maximal.

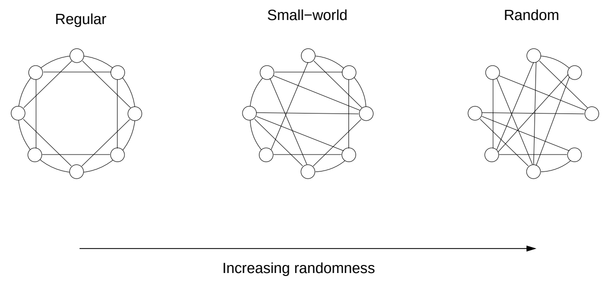

Figure 9: Morphing between a lattice graph and a random graph. Small-world graphs can be obtained by randomly rewiring just a few links between nodes.

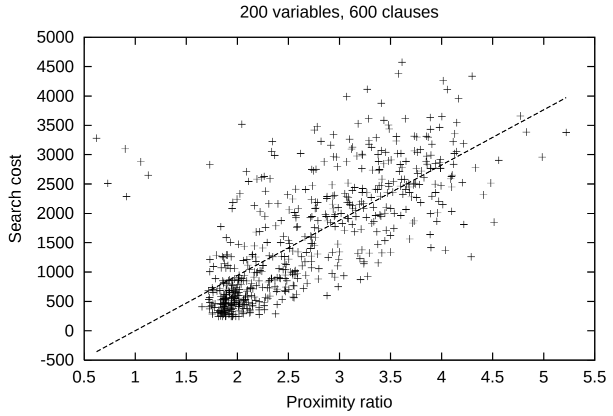

Figure 10: Correlation between search cost and proximity ratio.

Experimental results clearly show that, when the benchmark is tackled with a systematic technique such as BerkMin [10], small-world SAT instances are the hardest. In Fig.10, the search cost is plotted against the proximity ratio. The correlation between search cost and proximity ratio is apparent: The higher the proximity ratio, the higher the search cost.

The correlation between small-world topology and problem hardness has also been investigated in the case of approximate algorithms, namely local search, in [30, 32]. The emerging picture is quite complex, since the correlation between small-world topology and problem hardness is not as striking as in the case of complete search. However, the tendency of small-

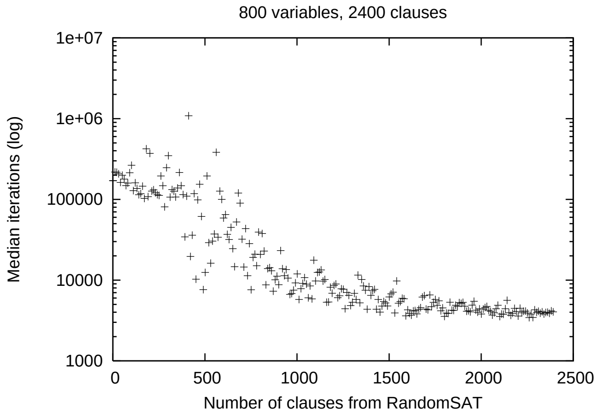

Figure 11: Search cost of ILS across the instances, from lattice to random structure. Points represent median iterations over 1000 runs. Log-scale on the y-axis has been used.

world SAT instances of being harder if tackled by a greedy local search is still observable. An illustrative example is depicted in Fig.11, in which the average search cost is plotted against the whole instance spectrum, from highly structured (left of x-axis) to random (right of x-axis). The instances are tackled with iterated local search (ILS, [20, 28]), a metaheuristic that incorporates a local search into a high level adaptive exploration strategy. The average search cost for solving the instances is the highest for small-world SAT instances.

The studies on the relation between small-world topology and problem hardness are still at an early stage and further investigation is required to assert under which hypotheses small-world instances are harder than others. Nevertheless, it is very important to observe that the research in the field of small-world and scale-free networks has given new impulse to an apparently not strictly related area. For instance, graph visualization tools have started to be applied to analyze the evolution of the interaction graph of SAT instances during their resolution [37]. As soon as a variable is instantiated, its corresponding node is deleted from the graph. Therefore, at each solving step, a subgraph of the initial one can be visualized. The graph structure evolution shows some specific and important characteristics of the solver used. For instance, some heuristics visually reveal their effectiveness by showing that the initial graph can be drastically reduced by assigning first some crucial variables, corresponding to hubs in the graph.[7]

[7]These variables are called *backdoors* [45].

5 Conclusion and outlook to the future

In this paper, we have outlined the main results achieved on the analysis of algorithm behavior w.r.t. topological features of the graph associated to the instance tackled. In particular, we have described recent studies on the relation between node degree and algorithm behavior and the impact of small-world topology on the search cost.

On the one hand, these results can explain, in some cases, why an instance is particularly easy/hard to solve and can be applied to the construction of well-defined benchmarks. On the other hand, they can guide the design of effective search heuristics exploiting structural properties of the instance to be solved. Moreover, graph features can be used to optimally set algorithm parameters and to choose among a portfolio of algorithms.

Whilst the research in this interdisciplinary field is quite recent, it is very promising and we believe that it could lead to important theoretical and practical results. The main research direction in this area is concerned with the exploitation of network properties to improve the solvers and understand their behavior, especially when applied to real-world problems. Besides the classification of instances based upon their network topology, some technical tools commonly used in graph analysis, such as spectral properties of adjacency matrices or models for network formation, could be applied as well. Furthermore, not only instance structure can be modeled as a graph, but also the search space explored by local search strategies.[8] Thus, some results and algorithms on evolution and navigation of complex networks could also be transferred to the realm of solving algorithm for combinatorial optimization.

6 Acknowledgments

I thank Marco Dorigo and the anonymous reviewers for useful comments on this work.

REFERENCES

[1] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(47), 2002.

[2] R. Albert, H. Jeong, and A.-L. Barabási. Diameter of the World Wide Web. *Nature*, 401:130–131, 1999.

[3] A.-L. Barabási. *Linked: The New Science of Networks*. Perseus Books Group, 2002.

[8]A preliminary description of this topic can be found in [33].

[4] A.-L. Barabási and E. Bonabeau. Scale-free networks. *Scientific American*, 288:60–69, 2003.

[5] J.C. Beck and M.S. Fox. Dynamic problem structure analysis as a basis for constrained - directed scheduling heuristics. *Artificial Intelligence*, 117:31–81, 2000.

[6] R. Béjar, A. Cabiscol, and C.P. Gomes C. Fernàndez, F. Manyà. Capturing structure with satisfiability. In *Proc. of CP 2001 – 7th International Conference on Principles and Practice of Constraint Programming*, Lecture Notes in Computer Science, pages 137–152. Springer, 2001.

[7] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, September 2003.

[8] E.G. Freuder. A sufficient condition for backtrack-free search. *Journal of the ACM*, 29(1):24–32, 1982.

[9] M.R. Garey and D.S. Johnson. *Computers and intractability; a guide to the theory of NP-completeness.* W.H. Freeman, 1979.

[10] E. Goldberg and Y. Novikov. BerkMin: A fast and robust SAT solver. In *Design Automation and Test in Europe (DATE)*, pages 142–149, 2002.

[11] C.P. Gomes and B. Selman. Problem structure in the presence of perturbations. In *Proc. of AAAI-97 – 14th National Conference on Artificial Intelligence*, 1997.

[12] C.P. Gomes, B. Selman, N. Crato, and H. Kautz. Heavy-Tayled phenomena in Satisfiability and Constraint Satisfaction Prpblems. *Journal of Automated Reasoning*, 24:67–100, 2000.

[13] C.P. Gomes and D. Shmoys. Completing quasigroups or latin squares: A structured graph coloring problem. In *Proc. of the Computational Symposium on Graph Coloring and Extensions*, 2002.

[14] A. Guerri and M. Milano. Learning techniques for automatic algorithm portfolio selection. In *Proc. of ECAI-04 – 16th European Conference on Artificial Intelligence*, Valencia, Spain, August 2004.

[15] H.H. Hoos. SAT-encodings, search space structure, and local search performance. In *Proc. of IJCAI-99 – 16th International Joint Conference on Artificial Intelligence*, pages 988–993, 1999.

[16] S.A. Kauffman. *At home in the universe.* Oxford University Press, New York, 1995.

[17] S.A. Kauffman and W. Macready. Technological evolution and adaptive organizations. *Complexity*, 26(2):26–43, March 1995.

[18] K. Leyton-Brown, E. Nudelman, G. Andrew, J. McFadden, and Y. Shoham. Boosting as a metaphor for algorithm design. In *Proc. of CP03 – 9th International Conference on Principles and Practice of Constraint Programming*, 2003.

[19] K. Leyton-Brown, E. Nudelman, and Y. Shoham. Learning the empirical hardness of optimization problems. In *Proc. of CP02 – 8th International Conference on Principles and Practice of Constraint Programming*, volume 2470 of *Lecture Notes in Computer Science.* Springer, 2002.

[20] H. Ramalhino Lourenco, O. Martin, and T. Stützle. Iterated Local Search. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics.* Springer-Verlag, 2003.

[21] W.G. Macready, A.G. Siapas, and S.A. Kauffman. Criticality and parallelism in combinatorial optimization. *Science*, 271:56–59, January 1996.

[22] M. Milano, editor. *Constraint and Integer Programming: Toward a Unified Methodology.* Kluwer Academic Publishers, 2004.

[23] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky. Determining computational complexity from characteristic 'phase transition'. *Nature*, 400:133–137, 8 July 1999.

[24] G.L. Nemhauser and A.L. Wolsey. *Integer and Combinatorial Optimization.* John Wiley & Sons, New York, 1988.

[25] I. Rish and R. Dechter. Resolution versus search: Two strategies for SAT. *J. Automated Reasoning*, 24:225–275, 2000.

[26] A. Roli. Criticality and parallelism in GSAT. *Electronic Notes in Discrete Mathematics*, 9, 2001.

[27] A. Roli. Criticality and parallelism in structured SAT instances. In *Proc. of CP02 – 8th International Conference on Principles and Practice of Constraint Programming*, volume 2470 of *Lecture Notes in Computer Science*, pages 714–719. Springer, 2002.

[28] A. Roli. Design of a new metaheuristic for MAXSAT problems. In *Proc. of CP02 – 8th International Conference on Principles and Practice of Constraint Programming*, volume 2470 of

Lecture Notes in Computer Science, page 767. Springer, 2002.

[29] A. Roli. Impact of structure in parallel local search for SAT. In *Proc. of SAT 2002 – 5th International Symposium on the Theory and Applications of Satisfiability Testing*, Cincinnati, Ohio, USA, 2002.

[30] A. Roli. Metaheuristics and structure in satisfiability problems. Technical Report DEIS-LIA-03-005, University of Bologna (Italy), May 2003. PhD Thesis - LIA Series no. 66.

[31] A. Roli. Problem structure and search: Empirical results and open questions. In *Proc. of CPAIOR'03 – 5th Int. Workshop on Integration of AI and OR techniques in Constraint Programming for Combinatorial Optimization Problems, Montreal, Canada*, 2003.

[32] A. Roli. On the impact of small-word constraint graphs on local search. In *Proc. of LSCS-04 – 1st International Workshop on Local Search Techniques in Constraint Satisfaction*. 2004.

[33] A. Roli. Symmetry-breaking and local search: A case study. In *Proc. of SymCon'04 – 4th International Workshop on Symmetry and Constraint Satisfaction Problems*. 2004.

[34] A. Roli and C. Blum. Critical Parallelization of Local Search for MAX–SAT. In *AI*IA2001: Advances in Artificial Intelligence*, volume 2175 of *Lecture Notes in Artificial Intelligence*, pages 147–158. Springer, 2001.

[35] T. Schiex. A note on CSP graph parameters. Technical report, INRA, Toulouse (France), 1999.

[36] B. Selman, H.J. Levesque, and D. Mitchell. A new method for solving hard satisfiability problems. In *Proceedings of AAAI92*, pages 440–446. AAAI Press, 1992.

[37] C. Sinz. Visualizing the internal structure of SAT instances. In *Proc. of SAT 2004 – 7th International Conference on Theory and Applications of Satisfiability Testing, Vancouver, BC, Canada*, May 2004.

[38] E. Tsang. *Foundation of Constraint Satisfaction*. Acedemic Press, 1993.

[39] T. Walsh. Search in a small world. In *Proc. of IJCAI-99 – 16th International Joint Conference on Artificial Intelligence*, pages 1172–1177, 1999.

[40] T. Walsh. Search on high degree graphs. In *Proc. of IJCAI-01 – 17th International Joint Conference on Artificial Intelligence*, 2001.

[41] J.P. Watson, L. Barbulescu, A.E. Howe, and L.D. Whitley. Algorithm Performance and Problem Structure for Flow-Shop Scheduling. In *Proc. of AAAI-99 – 16th National Conference on Artificial Intelligence*, 1999.

[42] D.J. Watts. *Small Worlds: The Dynamics of Networks between Order and Randomness*. Princeton University Press, 1999.

[43] D.J. Watts and S.H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[44] C. Williams and T. Hogg. Exploiting the deep structure of constraint problems. *Artificial Intelligence*, 70:72–117, 1994.

[45] R. Williams, C.P. Gomes, and B. Selman. Backdoors to typical case complexity. In *Proc. of IJCAI-03 – 18th International Joint Conference on Artificial Intelligence*, 2003.