Project plan: Approximation of NP-hard optimization problems
European Research Council Advanced Grant

This document consists of parts of the description of work of the above grant. It is not a complete document and is intended to give an introduction to the planned research.

1 General research area

The goal of complexity theory is to study the amount of computational resources that are needed to solve a particular computational problem. The most studied resource is computer time and here "time" is used to mean the number of elementary operations getting a theory that is independent of technology.

Clearly it is more difficult to solve large instances of a problem and thus one studies the running time as a function of the input length. A definition that has turned out to be useful is to say that a problem can be solved efficiently if the running time increases polynomially in the size of the input. This class of problems is denoted by P and another central complexity class is NP; problems where a found solution can be verified in polynomial time. The question whether these two complexity classes are equal is the most famous open problem in complexity theory. It is almost universally believed that the two classes are not equal but it seems like our understanding of computation is far from the point where this can be proved. The common approach, also adopted in this proposal, to get interesting results is to assume that NP $\neq$ P and derive consequences of this assumption.

A family of hard problems is given by the NP-complete problems defined originally by Cook [11]. Many basic problems fall into this class, some famous examples being graph colorability, the traveling salesman problem and satisfiability of Boolean formulas.

In graph colorability we are given a graph and we are asked, using a minimal number of distinct colors, to color the nodes in such a way that no two connected nodes have the same color. Classical NP-completeness tells us that it is hard to find the optimal number of colors and it is even hard to determine whether it is possible to use exactly three colors. Research in later years have focused on more fine-tuned questions given by various forms of approximation. A problem being NP-complete means that we cannot find the optimal answer but maybe we can find a reasonably good answer? One common interpretation of "reasonably good" would be here to look at the quotient of the objective function value of the obtained solution to that of the optimal solution.

In this quotient measure the approximability of colorability is rather well understood. If n is the number of nodes in the graph then it is possible to approximate the minimal number of colors within a factor of $O(n(\log\log n)^2/(logn)^3)$ [18], but hard to approximate this number within $O(n^{1-\varepsilon})$ for any constant $\varepsilon > 0$ [15], and if we are willing to make a slightly non-standard assumption ε can be made to tend to zero with n [26]. The finer question of how many colors are needed to color a graph that is guaranteed to be 3-colorable is more open. The best upper bound on the number of colors that is needed is $O(n^c)$ for $c \approx .2072$ [9] while the best hardness result, if you are only willing to assume that NP $\neq$ P is 5 [25] and if you are willing to make stronger, slightly nonstandard, assumptions this bound can be increased to any constant [12].

Graph-colorability is just one basic computational problem and there are many other problems for which similar questions can be asked. The general question is that given an instance of an NP-hard optimization problem and a guarantee that the optimal solution is of a given quality, what is the best quality of a solution that can be found by an efficient algorithm?

An important subclass of optimization problems is given by constraint satisfaction problems, or simply CSPs, where we are given a number of constraints over variables from a given domain and the task is to satisfy as many constraints as possible. Some CSPs are very resistant to approximation and even in the case when we can satisfy all constraints for a given instance it is impossible to efficiently find an assignment that does significantly better than a random assignment. Satisfiability of clauses of length at least three belongs to this category [21].

Some simpler CSPs show a much richer behavior and a prime example here would be Max-Cut where essentially the picture is now complete [33]. Max-Cut, formulated as a CSP, is an extremely simple problem in that it only specifies equalities between two bits. To what extent we can more fully understand more complicated CSPs remains to be seen.

Studying approximability of NP-hard optimization problems is naturally divided into two, rather disjoint but still closely connected, types of result.

On the one hand we want to derive positive results, showing that a certain problem does allow an efficient approximation algorithm producing outputs of a certain quality. This is done by designing and analyzing an efficient algorithm. Such results are also called "upper bounds on approximability".

On the other hand one wants to derive negative results in the form of showing that a certain problem does not allow an approximation of a given quality. As essentially all studied problem belong to the class NP they can be solved perfectly if NP=P and thus some complexity assumption is needed to prove a negative result, the most standard (and minimal) being NP≠P. Such *lower bound* results are usually proved through reductions as elaborated below.

1.1 Techniques for upper bounds

The key technique used for obtaining good approximation algorithms is semidefinite programming. It is true that in some cases linear programming is sufficient and purely combinatorial algorithms can sometimes give good bounds but as a single tool semidefinite programming has no rival for the title as king of approximation algorithms.

The basic primitive of semidefinite programming is to optimize a linear function of a matrix, subject to linear constraints and the constraint that the matrix is positive semidefinite. The entries of this matrix can many times be thought of as inner products of pairs of vectors. These vectors are in their turn usually of unit length and should be seen as generalizations of Boolean variables which can be seen as one-dimensional vectors of unit length. This basic technique was introduced to give the famous algorithm for Max-Cut by Goemans and Williamson [17].

Let us point out that unit length vectors can model elements from sets larger than two in a natural way and a standard way to model d values is to use the corners of a regular simplex in $d-1$ dimensions or some closely related construction. This has proved useful both in graph-colorability [24, 1] and constraint satisfaction problems over domain sizes larger than 2 [22].

1.2 Techniques for lower bounds

In principle one can start with any NP-complete problem and make a reduction to a problem of interest. If this reduction has the property that positive instances of the original problem yield new instances with optimal objective value at least c and negative instances yield new instances with optimal objective value at most s then we can conclude that it is NP-hard to approximate our problem within a factor c/s.

This can be achieved in a purely combinatorial way but the key tool used is that of a Probabilistically Checkable Proof (or more simply a PCP). In a PCP a probabilistic verifier is given a proof of an NP-statement and wants to verify the truth of this statement by doing probabilistic spot-checks of the proof.

In a standard PCP the verifier reads a constant number of bits in the proof, i.e. independently of the size of the NP-statement or its proof, always accepts a correct proof for a correct statement, and rejects any proof of an incorrect statement with probability at least one half. The amazing PCP-theorem of Arora et al. [2] states that any NP-statement admits such a PCP.

From the PCP-theorem one can derive inapproximability results by methods similar to those used to prove NP-completeness of various problems during the last 35 years. To get tight results, significant care is needed to both design a special purpose PCP and to very carefully analyze the connection between the PCP and the targeted optimization problem.

Looking abstractly a proof is given by a sequence of bits. A PCP is no different and a key component in the analysis of PCPs has been to analyze properties of Boolean valued functions on the hypercube. One essential tool to use is harmonic analysis. Some results, typically the early ones, [21] is a good example, depend on very simple properties of the Fourier transform such as Parseval's identity. Later results use harmonic analysis of increasing sophistication, one key tool being the Bonami-Beckner [8, 6] hyper-contractive inequality.

1.2.1 The unique games conjecture

By using the PCP-theorem one has obtained very strong NP-completeness and NP-hardness results, proving many variants of basic problems to be difficult. One basic problem is the label cover problem where we have a graph and each node should be given a label from a finite set. For each edge in the graph there is a constraint on the pair of labels given to the two end point of this edge. Khot [27] proposed to study instances when these constraint are in the form a one-to-one constraint, i.e. given that one of the end points is given a specific label there is a unique label that need to be given to the other endpoint.

The hardness of approximating the optimization problem of satisfying as many constraints as possible in such a unique label cover problem turns out to be of unexpected importance. The key question is whether it is difficult to decide, for an arbitrarily small $\varepsilon > 0$, whether it is possible to satisfy a fraction $1-\varepsilon$ of the constraints or whether the best label assignment satisfies only a fraction ε of the constraints.

The conjecture that this problem is NP-hard for any $\varepsilon > 0$, once the labels are from a sufficiently large but constant size domain, has become known as the “unique games conjecture” (UGC). A number of hardness results, one prime example being the hardness for vertex cover [30] but there are many more, have been proven based on this conjecture. Thus establishing this conjecture would bring many consequences and would be a great step forward. On the other hand falsifying the conjecture might, depending on exactly how this is done, bring even greater rewards in the form of new understanding.

2 Proposed research

We will continue our efforts in exploring the approximability of NP-hard optimization problems. This is a very active area of research world wide and the plan is that Stockholm will be a center of activity that can play a leading role in these exciting developments.

This is mathematical research and it is difficult to guess what results might be obtained as it is far from certain which conjectures are true, but let us discuss a few possibilities.

In the past we have seen upper bounds based on semi-definite programming matched by lower bounds which assume the unique games conjecture. One possible conclusion of this is that the unique games conjecture is true and we have found the best performance of a polynomial time algorithm. A more optimistic view is that the unique games conjecture captures the power of semi-definite programming[1] and that there is a new algorithmic technique around the corner which will wipe out both the UGC and the derived lower bounds. We feel that this is unlikely but the rewards of this being true are such that we feel this possibility should be explored.

Naturally working on the other side, proving the unique games conjecture is another top priority.

We consider many questions within approximability as possible to attack, both of the fine-tuned questions studying the quality of the efficiently obtainable solution as a function of the quality of the optimal solution but also basic questions of the simple approximation factor.

A specific problem that will be addressed is the asymmetric Traveling Salesman Problem where the distance from u to v is different from the distance from v to u, but where we do have the triangle inequality. Here it is unknown whether there exist an efficient algorithm that always finds the optimal tour within a factor which does not depend on n, the number of cities to visit as the smallest ratio obtainable

[1] It seems clear that semi-definite programming is not strong enough to disprove the unique games conjecture.

by efficient approximation algorithm is currently $\Omega(\log n)$. For this problem there seems to be no general consensus in the research community of the true answer.

Section 2, Research Proposal

i. State-of-the-art and objectives:

1 Specific Goals

The proposed project aims to create a center of excellence that aims at understanding the approximability of NP-hard optimization problems. In particular, for central problems like vertex cover, coloring of graphs, and various constraint satisfaction problems we want to study upper and lower bounds on how well they can be approximated in polynomial time.

Many existing strong results are based on what is known as the Unique Games Conjecture (UGC) and a significant part of the project will be devoted to studying this conjecture.

We expect that a major step needed to be taken in this process is to further develop the understanding of Boolean functions on the Boolean hypercube. We anticipate that the tools needed for this will come in the form of harmonic analysis which in its turn will rely on the corresponding results in the analysis of functions over the domain of real numbers.

2 Field overview, high level

The main question in complexity theory is to determine how hard it is to solve certain given problems. The definition of “hard” in this context varies from application to application, but a common definition is to estimate the amount of computation time needed to solve the problem. It has been widely accepted that a running time that can be bounded by a function that is a polynomial in the input length gives a robust definition of “reasonable running time” and the class of all problems that can be solved in polynomial time is denoted by P. There also exists a large class of decision problems with the property that an affirmative answer can be *verified* in polynomial time with the aid of a proof; this class is denoted by NP.

Consider the following problem: Given a Boolean formula on n variables, determine whether it is satisfiable or not, i.e., is there a truth assignment to the variables such that the formula evaluates to true. Clearly, a satisfying assignment to the variables is a proof of the fact that the formula is satisfiable. Such a proof can be verified in polynomial time by direct substitution in the formula. It is, however, not known how to *find* a proof in polynomial time. We can solve the above problem by *trying all possible proofs* but that takes time exponential in n. The above problem is in fact contained in a subset of NP, consisting of the so called

NP-*complete* problems. These problems are equally hard in the sense that if one of them can be solved in polynomial time, then all of them can and furthermore so can any other problem in NP. It would be extremely surprising if the NP-complete problems turned out to be solvable in polynomial time for at least two reasons. On the structural level it would be strange that for any problem where a solution can be verified quickly this solution can also be found quickly. On a more down to earth level it seems reasonable to expect that if polynomial time algorithms existed for all NP-complete problems, someone should have discovered one such algorithm for one of the many hundred well-studied NP-complete problems. In view of this we make the common assumption that solving NP-complete problems is difficult and in particular that NP$\neq$P.

While optimization problems strictly speaking cannot belong to NP since they are not decision problems, they can often be shown to be at least as hard as some NP-complete problem. In that case they are called NP-*hard* and cannot be solved optimally in polynomial time if P $\neq$ NP. This leads to investigations how well the optimal value of such difficult approximation problem can be approximated in polynomial time.

In 1990 Feige et al. [14] found a fundamental connection between the area of probabilistic checkable proofs (PCP) and efficient approximability of these NP-hard optimization problems. The constructions were improved leading to the famous PCP theorem by Arora et al. [2, 3] which showed that the class NP can be characterized using a very limited probabilistic interactive proof system. This characterization gives approximation hardness results for the important optimization problems Maximum 3-Satisfiability and Maximum Clique. The techniques used to construct such probabilistic interactive proof system have been refined over the years and we have a very long sequence of results with some highlights given in [7, 20, 21, 13]. A main tool, introduced in this area by Håstad in [20] is harmonic analysis of Boolean functions. While the first paper only used very simple properties from harmonic analysis, some later papers use known tools from real analysis such as the Bonami-Beckner [8, 6] hypercontractive inequalities and some other use an known translation from the Boolean domain to the domain of real numbers [32, 4, 5].

For the upper bounds of approximation the development has been almost equally fast. Goemans and Williamson showed that semidefinite programming, a generalization of linear programming that is efficiently solvable, can be used in approximation algorithms for problems such as Maximum Satisfiability and Maximum Cut [17]. This method improved the upper bounds for these problems considerably, and has since been used for approximating many other problems such as coloring [24, 9] and constrain satisfaction problems [22, 11, 35].

The two directions of research—improved approximation hardness results and improved approximation algorithms—are very dependent on each other. Together they have given us a much more complete picture of the approximability of NP problems than anyone could imagine was possible fifteen years ago. It is also clear that there is much to do before we get a complete understanding of approximability.

3 Some previous results in the area

Let us discuss some previous results in the area. For natural reason the discussion is focused on areas close to our interests.

3.1 Approximation resistance

One main branch of research on approximability of NP-hard optimization problems concerns approximate solutions to constraint satisfaction problems more succinctly called CSPs. An instance of such a problem is given as a collection of constraints, i.e., functions from some domain to $\{0,1\}$, and the objective is to satisfy as many constraints as possible. An approximate solution of a constraint satisfaction program is simply an assignment that satisfies roughly as many constraints as possible. For each such CSP, there exists a very naive algorithm that approximates the optimum within a constant factor: The algorithm that just guesses a solution at random. Håstad [21] proved the very surprising fact that this algorithm is essentially the best possible efficient algorithm for several constraint satisfaction problems, unless $\mathrm{P} = \mathrm{NP}$.

We call predicates for which no efficient algorithm can do substantially better than picking a random assignment "approximation resistant". To be approximation resistant is a much stronger property of a predicate than the corresponding decision problem being NP-complete as here efficient computation does not seem to be able to do *anything* useful. A natural and profound question to ask is: What is it that makes a CSP approximation resistant?

The situation for constraints that depend on only two variables is now resolved. When these variables are Boolean, the celebrated Goemans-Williamson algorithm [17] imply that every Boolean 2-CSP has a non-trivial approximation algorithm. This was extended to more and more classes of constraints and finally Håstad [22] proved that any constraint satisfaction problem over any fixed size domain where each constraint involves at most two variables does allow a non-trivial efficient approximation algorithm.

In the case of constraints that act over more than two variables, the most interesting results deal with Boolean variables. Zwick [37] classified all such constraints acting over three variables and here the results are strikingly simple and a constraint is approximation resistant iff it is implied by a parity constraint on its inputs.

The situation with constraints that depend on four Boolean variables has been studied extensively by Hast [19] and about 90 % of these have been classified as either approximation resistant or non-trivially approximable. Here the situation seems to be less structured and no clear pattern has emerged. For instance from the results on three variables one could have hoped that if a predicate P implies a predicate Q and P is approximation resistant then so is Q. Hast showed in his thesis that this statement is false.

For constraints that depend on more variables the results are sporadic and it is quite possible that the set of approximation resistant predicates form a very complicated set, that cannot be easily described. It could also be the case that there is a beautiful characterization to be found and only more research in the area can tell us which is the case.

3.2 Optimal algorithms and the unique games conjecture

The ultimate goal in approximability of NP-hard problems is to know, for each basic optimization problem, exactly which is the best approximation ratio that can be obtained in polynomial time.

As discussed above this ratio is known for some problems and essentially in all early cases where we know its exact value, either the upper bound or the lower bound is straightforward. This situation seems to be changing.

The algorithm of Goemans and Williamson [17] gave an approximation ratio for Max-Cut which is roughly .878 but it is exactly

$$\min_{\theta} \frac{2\theta}{\pi(1-\cos\theta)}$$

which is probably a transcendental number. In a surprising result Khot et. al [28] showed, modulo the truth of two conjectures, that this is indeed the correct constant. One of these conjectures has later been established [32] by a very elegant method using harmonic analysis to move the question from the Boolean domain to the real domain. The remaining conjecture to be established is the "Unique Games Conjecture"(UGC) of Khot [27]. This is a conjecture that has played a central role in the research area in the last couple of years, and let us discuss this conjecture in more detail. We formulate the underlying problem of UGC as a label cover question.

We have a graph G and the goal is to assign a label from $[L]$ to each node of the graph. For each edge $e = (u, v)$ there is a permutation π_e and the edge is satisfied if $\pi_e(\ell_u) = \ell_v$ where ℓ_u and ℓ_v are the labels of the vertices u and v. The conjecture is now that for any constants $\varepsilon, \delta > 0$ there is a size L for the label set such that it is NP-hard to distinguish label cover instances where the best assignment satisfies an $(1-\varepsilon)$-fraction of the edges from instances where the best assignment satisfies only a δ-fraction of the edges.

The term "unique games" comes from the fact that when G is bipartite the problem can be formulated as a very efficient two prover game where, for each answer from one prover, there is a unique answer from the other prover that makes the verifier accept.

Put differently, the UGC says that the optimization problem to determine the maximal number of simultaneously satisfied edges in a unique label cover instance is very difficult to approximate. This turns out to be a very strong assumption which can be used to derive many other inapproximability results. One good example is the result by Khot and Regev [30] proving that for any $\varepsilon > 0$, Vertex Cover cannot be approximated better than $2-\varepsilon$ in polynomial time, which matches the easy upper bound.

Austrin [4, 5], extending previous work [28, 32], has used the UGC to take a closer look at constraint satisfaction problems with two variables in each constraint and he has discovered a very close connection between approximability results using semidefinite programming and inapproximability results. In a semidefinite programming approach to CSPs a vector valued solution is first found which is modified to a Boolean solution by "rounding". The approximation factor is then decided by how well the worst vector configuration is rounded. This rounding has been studied systematically [31] and the best approximation ratio for Max-2-Sat is approximately[2] .94017 and is obtained by a very non-obvious rounding procedure.

Austrin has shown [4] that, assuming the UGC, the constant obtained by Lewin et al. is in fact optimal. To do this he takes the vector configurations that are hard to round and uses these to define a probabilistically checkable proof. The method turns out to be quite general and in a follow up paper Austrin [5] shows that for any CSP on two variables it is possible to take (a probability space of) vector configurations that are (simultaneously) hard to round and construct a PCP. If the vector configurations fulfill a certain technical condition this gives matching upper and lower bounds for the approximability of the problem at hand. The technical condition is non-trivial but in some sense natural and the bad configurations of Max-Cut and Max-2-Sat do satisfy it. An interesting corollary to his results is that, assuming the UGC, the hardest to approximate instances of Max-2-Sat can be very unbalanced with each variable appearing about twice as often in the positive form as negated. Similar methods have also been used by O'Donnell and Wu [33] to give us a very detailed understanding of the approximability of Max-Cut, again

[2] This has only been proven numerically, but there is little doubt that it is correct.

assuming the UGC.

The UGC can also be used to study CSPs of greater width and in particular Håstad [23] has used it show that for sufficiently large k, a random predicate that depends on k Boolean variables is, with high probability, approximation resistant. This result sheds some light on the structure of approximation resistant predicates discussed in the previous section. Håstad's result builds on a result by Samorodnitsky and Trevisan [36] that shows (assuming UGC) that if k is on the form $2^t - 1$ there are predicates of width k accepting $k+1$ inputs which are approximation resistant.

As discussed above there has been a close connection between approximation algorithms based on semidefinite programming and matching hardness results based on UGC. Very recently this has been strengthened further by Raghavendra [35] showing that integrality gaps for semidefinite programming can be translated to UGC-based hardness results in great generality. This very close connection can be interpreted in different ways. One could say that the UGC is true and semidefinite programming is the ultimate technique for getting efficient algorithms with good approximation ratios and all that remains is to prove the UGC. Being more optimistic, if one believes in the power of algorithms, one could argue that semidefinite programming is the best we know and it is not very efficient against the instances for label cover on which UGC is based and hence should not be applicable to problems obtained from UGC by reduction. There might be more powerful methods around the corner wiping out the UGC and violating our believed inapproximability results. This line of reasoning is, in my ears, more speculative but as the consequences of this being the correct state of affairs are so great that this line of research cannot be disregarded.

Another possibility is that we have finally found a natural decision problem which is neither solvable in polynomial time nor NP-complete. In other words the UGC would be false but any result derived from it remains true if we reduce "NP-hard" to "not solvable in polynomial time". We do think, however, that it would be premature to conjecture that this is the case, but it is a possibility to be kept in mind.

In any case, resolving the status of the UGC is extremely important for the research area, and we will work towards this end keeping our eyes open for all possibilities.

3.3 Traveling Salesman Problems

The traveling salesman problem is possibly the most famous combinatorial optimization problems. It can be formalized as follows: Given a complete directed graph with n nodes $c_1, \ldots, c_n$ and the edges marked with nonnegative integers

$d(c_i, c_j)$, find a permutation π of the nodes that minimizes $d(c_{\pi(n)}, c_{\pi(1)}) + \sum_{i=1}^{n-1} d(c_{\pi(i)}, c_{\pi(i+1)})$. Note that the distance function d may, in general, be asymmetric, i.e., it may hold that $d(c_i, c_j) \neq d(c_j, c_i)$, but we assume that d it obeys the triangle inequality as this is the interesting case from an approximability perspective.

There is an approximation algorithm due to Frieze et al. [16] that always delivers a solution with cost within a factor $\log_2 n$ from the optimum and this remains, up to a constant factor the best known bound. We remark that the reader may be more familiar with the *symmetric* TSP with triangle inequality, for which Christofides' classical approximation algorithm [10] gives a solution with cost within a factor $3/2$ of the optimum. As for approximation hardness, the currently strongest result is that it is, in the asymmetric case, NP-hard to compute a solution that is within factor $117/116$ of the optimum [34]. Closing this gap is a major open question in the field.

4 Possible areas of focus

It is very difficult to predict where research will take us during a period as long as 5 years. It is impossible to say which unproven theorems are true, which are within reach and which require genuinely new ideas and to which extent these can be found within our group.

Let us however, give at least and indication of the problems we will attack at the beginning of this period.

The first direction of research is to resolve whether the asymmetric TSP with triangle inequality can be approximated within a factor significantly better than $\log n$, a question that has been open for more than twenty years. The ultimate result here is of course to get a tight answer but any progress narrowing the gap between the upper ($O(\log n)$) and lower bounds (117/116) would be good progress.

The second direction is to obtain a better understanding of approximability of constraint satisfaction problems. We will here both investigate which predicates are approximation resistant and to further investigate the approximability of Boolean constraints on two and three variables. In the future a complete approximability result might not be given by a number but rather by a function $AP(x)$. The significance of this function would be that if the optimal solution satisfies a fraction x of the constrains then the approximation algorithm is guaranteed to find a solution that satisfies a fraction $AP(x)$ of the constraints. Most papers so far have only studied functions of the form $AP(x) = Cx$ but there are some exceptions, i.e., [29, 33] and we agree that this is the correct way to go.

The third direction is to study UGC, the unique games conjecture. The ultimate goal would of course be to prove or disprove it.

A fourth direction is to study classical problems such as Vertex Cover or coloring problems. In particular proving that there is a constant δ such that coloring a 3-coloring graph with n^{δ} colors is hard would be a fantastic result. There has been a chain of improved upper bounds of this form with the strongest obtained by Chlamtal [9]. Lower bounds for this problem are weak and it is only known that it is, assuming strengthened variants of the UGC, hard to color a 3-colorable graph with any constant number of colors [12].

A fifth direction is to study Boolean functions on Boolean hypercube using harmonic analysis. We expect that most concrete questions will here come from the analysis of concrete PCPs, but for the long term success it is important to study this area also with the approach of classical pure mathematics. We need to explore this subject for its own sake to understand it more fully.

References

[1] S. Arora, E. Chlamtac, and M. Charikar. New approximation guarantee for chromatic number. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computation*, pages 215–224, Seattle, 2006. ACM.

[2] S. Arora, C. Lund, R. Motwani, M. Sudan, and M.Szegedy. Proof verification and intractability of approximation problems. *Journal of the ACM*, 45:501–555, 1998.

[3] S. Arora and S. Safra. Probabilistic checking of proofs: a new characterization of NP. *Journal of the ACM*, 45:70–122, 1998.

[4] P. Austrin. Balanced Max-2-Sat might not be the hardest. In *Proceedings of 39th ACM Symposium on Theory of Computating*, pages 189–197, 2007.

[5] P. Austrin. Towards sharp inapproximability for any 2-CSP. In *Proceedings of 48th Annual IEEE Symposium of Foundations of Computer Science*, pages 307–317, 2007.

[6] W. Beckner. Inequalities in Fourier analysis. *Annals of Mathematics*, 102:159–182, 1975.

[7] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs and non-approximability—towards tight results. *SIAM Journal on Computing*, 27:804–915, 1998.

[8] A. Bonami. Etude des coefficients de Fourier des fonctiones de $L^p(G)$. *Ann. Inst. Fourier*, pages 335–402, 1970.

[9] E. Chlamtac. Approximation algorithms using hierarchies of semidefinite programming relaxations. In *48th Annual IEEE Symposium on Foundations of Computer Science*, pages 691–701, 2007.

[10] N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. Technical report, Graduate School of Industrial Administration, Carnegie-Mellon University, 1976.

[11] S. Cook. The complexity of theorem proving procedures. In *3rd Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[12] I. Dinur, E. Mossel, and O. Regev. Conditional hardness for approximate coloring. In *Proceedings of 38th Annual ACM symposium on Theory of Computing*, pages 344–353, 2006.

[13] I. Dinur and S. Safra. On the hardness of approximating minimum vertex-cover. *Annals of Mathematics*, 162:439–485, 2005.

[14] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43:268–292, 1996.

[15] U. Feige and J. Kilian. Zero-knowledge and the chromatic number. *Journal of Computer and System Sciences*, 57:187–200, 1998.

[16] A. Frieze, G. Galbiati, and F. Maffioli. On the worst-case performance of some algorithms for the asymmetric traveling salesman problem. *Networks*, 12:23–39, 1982.

[17] M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.

[18] M. Halldorsson. A still better performance guarantee for approximate graph coloring. *Information Processing Letters*, 45:19–23, 1993.

[19] G. Hast. *Beating a random assignment*. KTH, Stockholm, 2005. Ph.D Thesis.

[20] J. Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182:105–142, 1999.

[21] J. Håstad. Some optimal inapproximability results. *Journal of ACM*, 48:798–859, 2001.

[22] J. Håstad. Every 2-csp allows nontrivial approximation. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computation*, pages 740–746, 2005.

[23] J. Håstad. On the approximation resistance of a random predicate. In *Proceedings of RANDOM 2007 and APPROX 2007, LNCS 4627*, pages 149–163, 2007.

[24] D. Karger, R. Motwani, and M. Sudan. Approximate graph coloring by semidefinite programming. *Journal of the ACM*, 45:246–265, 1998.

[25] S. Khanna, M. Linial, and S. Safra. On the hardness of approximating the chromatic number. *Combinatorica*, 20:393–415, 2000.

[26] S. Khot. Improved inapproximability results for maxclique and chromatic number. In *Proceedings of 42nd Annual IEEE Symposium of Foundations of Computer Science*, pages 600–609, 2001.

[27] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings of 34th ACM Symposium on Theory of Computating*, pages 767–775, 2002.

[28] S. Khot, E. M. G. Kindler, and R. O'Donnell. Optimal inapproximability results for max-cut and other 2-variable CSPs? In *Proceedings of 45th Annual IEEE Symposium of Foundations of Computer Science*, pages 146–154, 2004.

[29] S. Khot and R. O'Donnell. SDP gaps and UGC-hardness for maxcutgain. In *Proceedings of 47th Annual IEEE Symposium of Foundations of Computer Science*, pages 217–226, 2006.

[30] S. Khot and O. Regev. Vertex cover might be hard to approximate to within $2-\varepsilon$. In *Proc. of 18th IEEE Annual Conference on Computational Complexity (CCC)*, pages 379–386, 2003.

[31] D. Livnat, M. Lewin, and U. Zwick. Improved rounding techniques for the Max 2-Sat and Max Di-Cut problems. In *Proc. of 9th IPCO*, pages 67–82, 2002.

[32] E. Mossel, R. O'Donnell, and K. Oleszkiewicz. Noise stability of functions with low influences: invariance. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 21–30, 2005.

[33] R. O'Donnell and Y. Wu. An optimal sdp algorithms for max-cut and equally optimal long code tests. Proceedings of 40th ACM Symposium on Theory of Computating, to appear, 2008.

[34] C. H. Papadimitriou and S. Vempala. On the approximability of the traveling salesman problem. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 126–133, 2000. revised versions available.

[35] P. Rahavendra. From integrality gaps to conditional hardness results. Proceedings of 40th ACM Symposium on Theory of Computating, to appear, 2008.

[36] A. Samorodnitsky and L. Trevisan. Gowers uniformity, influence of variables and PCPs. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 11–20, 2006.

[37] U. Zwick. Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In *Proceedings 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 201–210. ACM, 1998.