Solving and Approximating Combinatorial Optimization Problems (towards MAX CUT and TSP)

Jaroslav Nešetřil[1] and Daniel Turzík[2] *

[1] Charles University
Department of Applied Mathematics
Malostránské nám. 25
118 00 Praha 1, Czech Republic

[2] Institute of Chemistry and Technology
Department of Mathematics
Technická 5
166 28 Praha 6, Czech Republic

Abstract. We present a brief outline of recent development of combinatorial optimization. We concentrate on relaxation methods, on polynomial approximate results and on mutual relationship of various combinatorial optimization problems. We believe that this complex web of results and methods is typical for the modern combinatorial optimization. This paper is an introduction to our full paper [53].

1 Introduction

1.1

Combinatorial optimization deals with a special type of optimization problems with the property that set of all (feasible) solutions forms a finite set. 15 years ago, on the eve of the Khachiyan's ellipsoid method, we wrote a survey [52] for SOFSEM 1980. The present paper cannot be an update of that paper. The development has been too rapid and too extensive to be able to cover even its highlights. The present paper is also essentially shorter than [52]. But with all the differences the main theme is the same and we want to begin along the same lines.

1.2

The basic problem of Combinatorial Optimization can be described as follows:

Given a finite set S of objects and a weight function $w : S \longrightarrow \mathbf{R}$ find

$$\max\{w(s)\ ;\ s \in S\}$$

* Partially suported by GAČR 0194 and GAUK 194 grants.

.

In most cases the set S is highly structured and typically it can be described as a certain collection of subsets of an (underlying) set X, i.e. $S \subseteq \mathcal{P}(X)$. Then in turn we may view S as a finite subset of $\mathbf{R}^V$ of $|V|$-dimensional vector space whose coordinates are indexed by elements of V. Narrowing our focus still further, the most important case is the following:

There exists a weight function $w : V \longrightarrow \mathbf{R}$ such that the weight $w(s)$ of an object $s = (s_v; v \in V) \in \mathbf{R}^V$ is given by

$$w(s) = \sum_{v \in V} w(s_v) \ .$$

Thus we assume that the weight is a linear function of its coordinates and we speak about Linear Objective Combinatorial Optimization problem (shortly LOCO problem). All problems considered in this paper are LOCO problems.

In many cases the size of S is huge (exponential) yet the problem can be solved in time bounded by the polynomial function on $|V| = n$.

1.3

Combinatorial optimization shares one important aspect with combinatorics and discrete mathematics:
It is not a cohesive theory which is recognized and studied per se but rather a collection (or a ZOO) of results, algorithms and even trics which are best described by a series of examples. The following are some of the best known and traditional examples of LOCO problems. We describe them first informally, LOCO formulation will follow.

1.4 Examples of LOCO problems

SP: In a given configuration find a shortest path between two given points.

MST: In a given configuration find a minimum scheme which guarantees mutual communication.

CHP: In a given city find the shotest route of a (hypothetical, tireless) postman to visit each street at least once.

MF: Find a maximum flow through a network.

TSP: In a given region find a shortest tour of a travelling salesman (visiting all the palces).

MM: For every pair (x, y) of SOFSEM '97 participants there is a known measure $e(x, y) \geq 0$ of their empathy. Schedule an accomodation in two bed rooms to maximize the success of this meeting.

MC: Given a set of animosities find a best partition (say by a fence) in 2 classes minimizing the animosities within the classes.

MS: Given a set of senders find maximal set among them which do not interfere.

1.5

All these problems can be formulated as LOCO problems exactly and concisely by means of graphs as follows. This reformulation also explains the above acronyms:

SP= Shortest Path
In an edge-weighted graph G find a shortest path between two vertices.

MST= Minimum Spanning Tree
Given a graph $G = (V, E)$ together with a weight function $w : E \longrightarrow \mathbf{R}$ find a tree (V, E') for which

$$\sum_{e \in E'} w(e)$$

is minimal.

CHP= Chinese Postman Problem
Given a weighted graph $G = (V, E)$, $w : E \longrightarrow \mathbf{R}$ find a walk

$$v_0, e_1, v_1, \ldots, e_t, v_t = v_0$$

which contains all edges of G.

MF= Maximal Flow in a Network
(This needs a few formal definitions.) A network is a directed graph $G = (V, E)$ (i.e. $E \subseteq V \times V$) with edge capacities $c : E \longrightarrow \mathbf{R}^+$ and two specified vertices z (the source) and s (the sink). A flow in a network (G, c, z, s) is a function $f : E \longrightarrow \mathbf{R}^+$ which obeys capacities (i.e. $0 \leq f(e) \leq c(e)$ for every edge $e \in E$) and which preserves conservation law for its inner vertices (i.e. vertices different from z and s). By this we mean that

$$\sum_{(x,y) \in E} f(x, y) = \sum_{(y,x) \in E} f(y, x)$$

for every $x \neq z, s$. The size $|f|$ of a flow f is defined by

$$|f| = \sum_{(z,y) \in E} f(z, y) - \sum_{(x,z) \in E} f(x, z) \ .$$

The problem of maximal flow can be formulated as follows: Given a network (G, c, z, s) find the maximal size of a flow.

TSP= Travelling Salesman Problem
Given a graph $G = (V, E)$ with edge weights w find an ordering of (all) its vertices $v_0, v_1, \ldots, v_n = v_0$ such that

$$\sum_{i=1}^{n} w(v_{i-1}, v_i)$$

is minimal. If the weight function satisfies the triangle inequality (or if w is given by the euclidean distance) than we speak about metric (or euclidean) TSP.

MM= Maximal Matching

Given a graph $G = (V, E)$ with edge weights w find a set M of disjoint edges such that

$$\sum_{e \in M} w(e)$$

is maximal.

MC= Max Cut = Maximal Cut

Given a graph $G = (V, E)$ with edge weights find a bipartite subgraph (V, E') for which

$$\sum_{e \in E'} w(e)$$

is maximal. This is a formulation of Max Cut problem for positive edge weights.

MS= Maximal Stable Set

Given a graph G find a maximal set $A \subseteq V$ which does not contain any edge (such a set is called a stable set or independent set; its maximal size is denoted by $\alpha(G)$).

1.6

These problems are comming from different areas (both theory and applications) and their history is to a great degree disjoint. Perhaps more importantly, the computational complexity of these problems have a very different behaviour. This was studied recently in great detail and one can say that this research is one of the driving forces of the whole theoretical computer science. One of the aims of this paper is to provide a reader with at least few glimpses of this fascinating development.

On the other hand side combinatorial optimization is very close to applied problems and one should say perhaps surprisingly successfull in solving large projects. Here one should mention very successfull design of chips (see [37]; chips designed by University of Bonn were reportedly used in machines in a recent match with Kasparov) and a recent business-success of CPLEX company (visit *http:/www.ilog.com/html/press_cplex.html*).

1.7

But despite of all these differences the above problems form a compact body of results and techniques and one can say that together form a key part of modern Combinatorial Optimization. This fact has double meaning for our presentation:

First, as the above problems reflect most of the development of combinatorial optimization, the subject is vast and we have so restrict. So even in the full version [53] we decided to concentrate on two problems: Travelling Salesman Problem and Max Cut Problem.

Secondly, as the above problems form a closely knight group we can illustrate the main trends of the development by considering only few typical examples.

One of our main goals is to demonstrate mutual relationship and similarities between above problems and techniques devised for their solution. This will be done here. This paper provides then a framework for the more delailed study of particular cases of TSP and Max Cut in the full version of our paper [53].

2 A FEW CONNECTIONS AND IMPLICATIONS

2.1

The Max Cut and TSP were not chosen randomly. Quite to the contrary. They both present traditional problems of combinatorial optimization. It is often said that these problems are important as they are applied and appear in many practical situations. This is of course a logical and safe claim in todays world but in reality problems appear quite rarely in a purity demanded by TSP and Max Cut formulation. Of course there are exceptions and they are recorded e.g. in [1], [7], [63]. But perhaps one shouldn't claim that this is the most important motivation for studiyng such problems. What is perhaps more to the point is that both these problems became sort of a testing ground and cornerstones of modern combinatorial optimization [2], [27], [45], [6], [22], [55], [61], [59], [58]. These problems were studied for a long time and despite all the efforts they still appear hard.

2.2

There is a theoretical justification for this hardness. In fact there is a manifold evidence for it and this has been one of the main driving forces of theoretical computer science. The following table gives the basic information about the (worst case) complexity of our problems.

SP	MST	CHP	MF	TSP	MM	MC	MS
P	P	P	P	H	P	H	H

Here P stands for the existence of polynomial time algorithm and H for NP-hardness; as all the corresponding decision problems belong to NP these decision problems actually belong to NP-complete problems.

2.3

But while our problems appear to be hard yet because of their formulation simplicity (and, yes, "application appeal") they attracted considerable attention and from the early days efforts have been made to solve large (or modestly large) problems. Because of the inherent complexity of the problems no direct approach worked and no simple master algorithm has been devised. Rather a complex web of trics, mutual interplays, partial and approximate solutions, and heuristical arguments have been found. To a certain degree this in fact established Combinatorial Optimization as we know it today. One can say that the main goal of this short survey is to provide an interested reader with a description of a few facets of this (what we believe) fascinating development.

2.4

Thus we concentrate mainly on the theoretical aspects (with an eye on potential and recorded experiments; as we know the theory is here very close to applied work [7], [32] , [37], [45]). We introduce all the key notions and try to display main interconnections between various problems.

2.5

The above complexity table does not exhaust the complicated 50 years or so history of these problems. Many interesting results were obtained on both ends of our spectrum. Perhaps more rapid development was on the polynomial side of the spectrum.

2.6

The interesting history of polynomial solutions of MST starts with Borůvka [8] and Jarník [31]. The manifold uprovements of the MST algorithms would fill a rather long paper by itself (and this has been done e.g. in [25],[42] and [51]). Let us just note that all efforts so far (narrowly) missed the main goal: to find a linear deterministic algorthm for MST problem. (The only problem in our list with known linear algorithm is SP Problem).

However the ideas behind the speedup of existing algorithms and their refinements led to isolation important structures (Set Union problem, Fibonacci heaps, verification algorithm), see [66], [34], [40] and thus provided a background for much of the recent development.

2.7

Conceptually one of the easiest algorithms is so called a greedy algorithm due to Kruskal [43]. This algorithm can be used as an heuristic for most (if not all) combinatorial problems which have a hereditary set of feasible partial solutions: We start with an an initial (say trivial) solution and extend this solution step by step so that each step the extension is optimal. This greedy heuristic leads to the notion of matroid and greedoid, see [69], [41], which can be seen as the combinatorial structures supporting the optimality of greedy algorithm. The greedy algorithm is a useful heuristic even in the case when it does not leads to an optimum. Examples of this include also TSP.

2.8

Network flow algorithms play a central role ([66] is one of the best references for this). The same is true for the seemingly isolated problem of maximal weighted matching (Problem MM). Mathing theory and matching algorithms present perhaps the best developed field of graph theory and [49] is still best source.

2.9

These problems are important as they until recently influence other problems. For example the best heuristic for the metric TSP (due to Christofides) was based on the MM problem .

It is important that the MM problem can be sometimes regarded as a flow problem. Also Max Cut problem can be viewed as a Maximal Flow problem but we have to accept negative capacities. It is a bit surprising that this extension of flow problems leads to such drastical change of the complexity of the problem. It is worth to note that this is not the case in the MST problem.

On the other hand side even the shortest path problem includes Travelling salesman problem if we accept negative weights.

Continuing with this introductory overview we mention that the polynomiality of (all pairs) shortest path problem (i.e. SP), together with Maximal Matching Problem implies that Chinese Postman problem is polynomial. On the other hand side we may view Chinese Postman Problem as a flow (or circulation) problem of minimal cost.

2.10

This does not exhaust all the implications and translations between the above problems. In fact one of the essential features of Combinatorial Optimization (and in a sense of complexity theory too) is an abundance of reductions and translations.

3 A CASE ANALYSIS

3.1

In a certain sense the Maximal Stable Set problem (i.e. MS) is universal as it captures e.g. any subgraph type problem. It is also a prototype of an NP-complete problem and one of the problems which were most thoroughly investigated. The reader may ask what is the connection of this problem to our two problems. Well for example we shall see bellow that the best known method for approximate solutions follow the same line.

3.2

Imagine that we want to solve a maximal stable set problem in a (large) graph $G = (V, E)$ with n vertices. After trying several heuristics (such as greedy algorithm) we would like to get something more sophisticated which would give us some performance quarantee (if only in some cases). The standard approach which goes back to the beginning of fifties is to consider geometric form and a linear relaxation of our (combinatorial) problem:
We denote by $\mathbf{R}^V$ the n-dimensional vector space with coordinates indexed by elements of V. Thus any independent subset $A \subseteq V$ may be thought as vector (i.e. incidence 0-1 vector) of $\mathbf{R}^V$. The convex hull of the set of all independent sets generates a polyhedron, *a stable set polyhedron* of G, which we denote by $MS(G)$. Obviously all the vertices of $MS(G)$ have integral coordinates and they correspond to independent sets. Thus it suffices to find the vertex x of $MS(G)$ with largest weight $\sum_{v\in V} x_v$. However this is just a reformulation of MS problem. (It is also an integer LP problem - so no hope to solve it - at least presently). However Linear Programming provided an important methodology which constitutes the key part of polyhedral combinatorics.

3.3

We restate the problem by writing a set of linear constraints in the following way:

$$\max \sum_{v\in V} x_v$$

subject to constraints

$$\left.\begin{array}{rl} x_v + x_{v'} \leq 1 & \{v, v'\} \in E \\ x_v \geq 0 & v \in V\ . \end{array}\right\} \tag{1}$$

Now these inequalities determine a polyhedron $MS_L(G)$ which obviously contains polyhedron $MS(G)$. Not only that all the integral vectors in $MS_L(G)$ belong to $MS(G)$ and thus the linear programming problem (1) present a *linear relaxation* of MS problem.

3.4

As one sees easily the polyhedron $MS_L(G)$ properly contains $MS(G)$ and thus an optimal solution to (1) does not necessarily gives a solution of MS. What are the obstacles? E.g. for odd cycle C_5 the polyhedron $MS_L(C_5)$ has the vector $(1/2, 1/2, 1/2, 1/2, 1/2)$ as its vertex. However, in any case two important facts are valid:

i. The optimal solution to problem (1), i.e. optimization over polyhedron MS_L provides an upper bound for the propblem MS.

ii. We can add the "obstacle" inequalities to the system (1) in a hope that we shall locate the position of polyhedron MS within MS_L more accurately.

Thus the new system of inequalities induces again a linear relaxation of the original problem. Gomory [23] and Chvátal [10] showed a systematic way of generating new obstacle inequalities (by taking linear combinations and rounding), such that every (integer) problem may be transformed to a linear programming problem. This method - cutting plane method - proved to be extremaly usefull in many instances. However recently it has been shown [60] that there are problems which need exponentially many cutting planes to be added. Also these problems are related to MS problem.

3.5

Returning to our main scheme, for Maximal Stable Set problem three more accurate relaxations were isolated which lead to convex sets $MS_{ODD}(G)$, $MS_{CL}(G)$, $MS_{TH}(G)$ as follows:
$MS_{ODD}(G)$ is the set of all vectors in $\mathbf{R}^V$ satisfying (1) and conditions

$$\sum_{v \in C} x_v \leq \frac{|C|-1}{2} \ , \ C \text{ an odd cycle in } G. \tag{2}$$

$MS_{CL}(G)$ is the set of all vectors in $\mathbf{R}^V$ satisfying (1) and conditions

$$\sum_{v \in K} x_v \leq 1 \ , \ K \text{ a clique (i.e. a complete subgraph) in } G. \tag{3}$$

Both $MS_{ODD}(G)$ and $MS_{CL}(G)$ are polyhedra in $\mathbf{R}^V$.

3.6 $MS_{TH}(G)$

This is based on "Lovász theta function" first defined in the seminal paper by Lovász [46], see [48] for the following formulation:
$MS_{TH}(G)$ is defined as the set of all vectors $d = (d_v \ ; \ v \in V) \in \mathbf{R}^V = \mathbf{R}^n$ for which there exists a positive semidefinite $(n+1) \times (n+1)$ matrix $Z = (z_{ij})$ satisfying

$$\left.\begin{array}{rl} z_{00} = 1 & \\ z_{i0} = z_{0i} = z_{ii} = d_i & i = 1, \ldots, n \\ z_{ij} = 0 & \{i,j\} \in E \ . \end{array}\right\} \tag{4}$$

(If d is the incidence vector of an independent set in G, then putting $z_{ij} = d_i d_j$, $z_{00} = 1$ and $z_{0i} = z_{i0} = d_i$ we get a matrix Z which satisfies the above properties. As $Z = (1,d) \cdot (1,d)^T$ the matrix Z is also positive semidefinite. This shows that $MS_{TH}(G)$ is a relaxation of the stable set polyhedron $MS(G)$.

3.7

The convex set $MS_{TH}(G)$ fails to be a polytope. However as one can decide polynomially whether a given matrix is positive semidefinite, one can solve a separation problem for $MS_{TH}(G)$ polynomially and thus by one of the main result of [27] one can optimize over $MS_{TH}(G)$. This has important consequences, which we outline now.

3.8

All these convex sets lead to important classes of graphs and their important properties:

Graphs with $MS_{ODD}(G) = MS(G)$ are called t-perfect graphs (and their study was proposed by Chvátal [11]). From all the classes this is perhaps the most esoteric class (which has not yet been characterized by other means and, particularly, no P membership algorithm for deciding whether a graph is t-perfect is known).

3.9

Still the MS problem for t-perfect graphs can be solved in polynomial time. That may seem to be surprising on the first glance but according to the general theory [27] it suffices to solve a corresponding separation problem:

Given $y \in \mathbf{Q}^V$ we want to decide whether $y \in MS_{ODD}(G)$ or otherwise to find a hyperplane of type (1) or (2) which is violated by y. Checking (1) is easy. Thus let us assume $y_v \geq 0$ and $y_v + y_{v'} \leq 1$ for every edge. We want to check whether (2) holds. This is possible to solve by the following trick:

In our situation define a vector $z = (z_e \ ; \ e \in E) \in \mathbf{R}^E$ by

$$z_e = 1 - y_v - y_{v'} \geq 0$$

for an edge $e = \{v, v'\}$. Then

$$\sum_{e \in C} z_e = |C| - 2 \sum_{v \in C} y_v$$

and thus (2) is equivalent to the validition of the condition

$$\sum_{e \in C} z_e \geq 1$$

for every odd circuit C of G. However we can think of z_e as nonnegative edge weights of E and thus it suffices to check whether (with the weights z_e) the shortest odd circuit has length ≥ 1.

3.10

Thus the MS problem for t-perfect graphs was finally reduced to the shortest odd circuit in an edge weighted (undirected) graph. This problem is possible to solve by another neat trick:

Given a graph $G = (V, E)$ we define a new graph $G' = (V', E')$ by

$$V' = V \times \{0, 1\} \ , \qquad E' = \{\{(v, i), (v', j)\} \ ; \ \{v, v'\} \in E \ , \ i \neq j\} \ .$$

The shortest path from $(v, 0)$ to $(v, 1)$ is necessarily of an odd length and thus we may solve our problem by $|V|$ iterations of the shortest path algorithm.

3.11

The shortest odd cycle problem is an interesting variation of SP problem with a long history and important connections. The reader can try to convict himself that to detect and to find shortest odd cycle (in directed) and shortest even cycle (in undirected graphs) are easy problems. However the problem of deciding whether a given directed graph contains an even cycle is much harder and also more important. This problem has been shown to be related to coloring of hypergraphs [65], to problems in statistical physics [36] and to the problem when the permanent of a given 0-1 matrix A (a problem believed to be not easily algoritmicaly solvable [68]) can be reduced to the easy computation of the determinant of a matrix A' which arises from A by changing some 1 to -1 (this problem goes back to Polya), see also [39], [19]. Very recently all these problems were solved by surprising characterization theorem and a polynomial algorithm [13]. This algorithm involves among others MM polynomial algorithm. The complexity of the algorithm is $O(n^3)$. Interestingly and unexpectedely the present known complexity of most naturally defined problems seems to be bounded by n^3. We do not know any reason for it.

3.12

The above hierarchy of convex sets defines furher classes of graphs. It is easy to prove that $MS(G) = MS_L(G)$ iff the graph G is bipartite (i.e. iff G does not contain any odd cycle). However the graphs defined by $MS(G) = MS_{TH}(G)$ coincide with graphs satisfying $MS(G) = MS_{CL}(G)$ and these graphs are called perfect graphs. It follows from the above theory that there is a polynomial time algorithm to solve MS problem for perfect graphs. This is a non-trivial result for which no other proof is known (and which thus relies on the ellipsoid method [27]).

3.13

In general, MS problem is difficult and NP-complete even when restricted to very special classes of graphs (such as planar graphs). The most polynomial cases are covered by the above general polyhedral approach (there are some exceptions see e.g. [5]). Starting with [24] it became a promissing line of research to look for approximate algorithms. In the case of MS problem this mean the following:

Denote by $\alpha(G)$ the maximal size of an independent set in the graph G. Let A be an algorithm which finds a (not necessarily largest) independent set of size $A(G)$ in G. We say that A is an *c-approximation algorithm* if

$$\alpha(G) \leq c \cdot A(G) \ .$$

We also say that MS problem (and quite analogously any CO problem) admits *a polynomial approximation scheme* iff for every $c > 1$ there exists a polynomial algorithm A which is c-optimal. It is wellknown that there are NP-complete problems which admit a polynomial approximation scheme (such as bin packing problem [17]). However for many problems the existence of a polynomial approximation scheme remained until recently an open and fundamental problem.

3.14

For the maximal stable set problem it had been observed in [20] that an existence of $c > 1$ polynomial approximation algorithm yeilds the existence of a polynomial approximation scheme. However the existence of such approximation scheme was disproved in a dramatic way in [18], [4] and finally [3] where it has been shown that for an $\varepsilon > 0$ the maximal independent set in a graph $G = (V, E)$ cannot be approximated within a factor $|V|^{\varepsilon}$.

These results were the first deep results about non-approximability of combinatorial optimization problems and they extend (using [56]) to most interesting combinatorial problems.

3.15

Both for the Max Cut problem and TSP this development was especially interesting. This will be discussed in more details in our talks. Let us just state here that the best known c-approximation algorithm for the Max Cut problem has ratio $c = 1.1383$ [21] and it is based on a semidefinite relaxation of the Max Cut polytope due to Delorme and Poljak [15]. This is a great progress from an obvious 2-approximation algorithm (see [57]). It has been shown recently [44] that this relaxation is closely related to Lovász theta function. Currently the best non-approximability result is for $c = 1.0624$ [28]. For the TSP problem, like for the Maximal Stable Set problem, there is no polynomial approximation algorithm [3]. However for metric TSP problem there is an easy $3/2$ approximation algorithm [9] which is still the best. Again for matric TSP no polynomial approximation scheme (under $P \neq NP$), exists [3]. To a great surprise the euclidean TSP behaves differently. Recently Arora [2] produced polynomial approximation scheme not only for Euclidean TSP, but also polynomial approximation schemes for other euclidean optimization problems among them euclidean Steiner Tree Problem (see also [35]). These results hold in any fixed dimension d (and the dependence on d in approximation algorithms is double exponential). This dependence on d is necessary as was recently shown in an important paper by Trevisan [67].

3.16

The problem to determine the approximation status of a particular optimization problem received recently lot of attention and in several cases the full solution was achieved [28], [38], [67] (such as 3-SAT problem [28]). However for metric TSP and the Max Cut problem the situation is far from beeing solved. This will be discussed in great detail in the full version of this paper [53].

3.17 Conclusion

We tried to outline some of the main directions of contemporary Combinatorial Optimization. We concentrate mainly on theoretical questions where the progress has been remarkable. Still we had to omit several important issues such as classification problem (see [16], [29], [30], [38], [64]) or randomized algorithms (see [33], [47], [50]). We also only mentioned few aspects of TSP and Max Cut problems; the present paper should be seen as an introduction to the manysided research related to these problems. However all this theoretical work found its way to large scale computing and indeed some practical problems. Significant and unexpectedly complex (and indeed very complex) problems have been solved. This is documented e.g. in [1], [7], [12], [14], [26], [32], [54], [62], [61]. The paper [53] will describe some of these methods in a greater detail.

References

1. D.Applegate, R.Bixby, V.Chvátal, W.Cook: Finding cuts in the TSP. DIMACS Tech. Report, 95–105.
2. S.Arora: Polynomial time approximation schemes for Euclidean TSP and other geometric problems. In: Proc. of the 37th IEEE FOCS, 1966.
3. S.Arora, C.Lund, R.Motwani, M.Sudan, M.Szegedy: Proof verification and intractability of approximation problems. 33rd FOCS, 1992, 14–23.
4. S.Arora, S.Safra: Probabilistic checking of proofs: a new characterization of NP. 33rd FOCS, 1992, 2–13.
5. E.Balas, V.Chvátal, J.Nešetřil: On the maximum weight clique problem. J. of Oper. Research (1985).
6. F.Barahona: On cuts and matchings in planar graphs. Mathematical Programming 60(1993), 53–68.
7. R.E.Bland, D.F.Shallcross: Large travelling salesman problems arising from experiments in X-ray crystalography: a preliminary report on computation. Oper.Res.Lett. 8(1989), 125–128.
8. O.Borůvka: O jistém problému minimálním (about a certain minimal problem). Práce mor. přírodověd. spol. v Brně III, 3(1926), 37–58.
9. N.Christofides: Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Carnegie-Mellon University, 1976.
10. V.Chvátal: Edmonds polytopes and hierarchy of combinatorial problems. Discrete Math. 4(1973), 305–337.
11. V.Chvátal: On certain polytopes associated with graphs, J.Comb.Th. B, 18(1975), 138–154.
12. H.Crowder, M.W.Padberg: Solving large-scale symmetric travelling salesman problems to optimality. Management Sci 26(1980), 495–509.
13. W.McCuaig, N.Robertson, P.D.Seymour, R.Thomas: Permanents, Pffafian orientations and even directed circuits. (to appear; see also a preliminary version in STOCK (1997)).
14. G.B.Dantzig, D.R.Fulkerson, S.M.Johnson: Solution of large-scale travelling salesman problem, Oper.Res. 2(1954), 393–410.
15. C.Delorme, S.Poljak: Laplacian eigenvalues and the max-cut problem. Mathematical Programming 63(1993), 557–574.

16. T.Feder, M.Vardi: Monotonne monadic SNP and constraint satisfaction. Proceedings of the 25th ACM STOC, ACM, 1993.
17. W.Fernandez de la Vega, G.S.Lueker: Bin packing can be solved within $1+\varepsilon$ in linear time. Combinatorica 1(1981), 349–355.
18. U.Fiege, S.Goldwasser, L.Lovász, S.Safra, M.Szegedy: Interactive proofs and the hardness of approximating cliques. Journal of the ACM 43(1966), 268–292.
19. A.Galluccio, M.Loebl: Even cycles and H-homomorphismus. (to appear)
20. M.R.Garey, D.S.Johnson: The complexity of near optimal coloring. J. Assoc. Comput. Math. 23(1976), 43–49.
21. M.Goemans, D.Williamson: .878-approximation algorithm for Max-Cut and Max-2-SAT. 25th STOC, 1994, 422–431.
22. M.Goemans, D.Williamson: Improved approximation algorithm for maximum cut and satisfiability problems using semidefinite programming. Journal of the ACM 42(1995), 1115–1145.
23. R.E.Gomory: Outline of an algorithm for integer solutions to linear programs. Bull. Amer. Math. Soc. 64(1958), 275–278.
24. R.L.Graham: Bounds for certain multiprocessing anomalities. Bell System Tech. J. 45(1969), 1563–1581.
25. R.L.Graham, P.Hell: On the history of the minimum spanning tree problem. Annals of the History of Computing 7(1985), 43–57.
26. M.Grötschel, O.Holland: Solution of large-scale symmetric travelling salesman problems. Math.Programing 51(1991),141–202.
27. M.Grötschel, L.Lovász, A.Schrijver: Geometric algorithms and combinatorial optimization. Springer Verlag, 1988.
28. J.Hastad: Some optimal inapproximability results. In: Proceedings 29th STOC (1997), 1–10.
29. P.Hell, J.Nešetřil: On the complexity of H-coloring. J.Comb.Th. B, 48(1990), 92–110.
30. P.Hell, J.Nešetřil, X.Zhu: Duality and polynomial testing of graph homomorphism. Trans.Amer.Math.Soc. 348, 4(1996), 1281–1297.
31. V.Jarník: O jistém problému minimálním (about a certain minimal problem). Práce mor. přírodověd. spol. v Brně IV, 4(1930), 57–63.
32. M.Jünger, G.Reinelt, S.Thienel: Practical problems solving with cutting plane algorithms. In: Combinatorial Optimization (W.Cook, L.Lovász, P.Seymour, ed.), DIMACS Series vol.20, AMS(1995), 111–152.
33. G.Kalai: A subexponential randomized simplex algorithm. Proc. 24th STOCK, 1992, 475–482.
34. D.Karger, P.N.Klein, R.E.Tarjan: A randomized linear-time algorithm to find minimum spanning trees. J.Assoc.Comp.Mach. 42(1995), 321–328.
35. M.Karpinski, A.Zelikovsky: New approximation algorithms for the Steiner tree problems. Technical Report ECCC TR95-030, 1995.
36. P.W.Kasteleyn: Graph theory and crystal physics. In: Graph Theory and Theoretical Physics (F.Harary, ed.), Academic Press, New York, 1967, 43–110.
37. A.Kehrbaum, B.Korte: Calculi. West Deutcher Verlag, 1995.
38. S.Khanna, M.Sudan, D.P.Williamson: A complete classification of the approximability of maximization problems derived from Boolean constraint satisfaction. In: Proceedings 29th STOC (1997), 11–20.
39. V.Klee, R.Ladner, R.Mauber: Sign-solvability revisited. Linear Algebra Appl. 59(1984), 131–158.
40. J.Komlos: Linear verification for spanning trees. Combinatorica 5(1985), 57–65.

41. B.Korte, L.Lovász, R.Schrader: Greedoids. Springer 1990.
42. B.Korte, J.Nešetřil: Vojtěch Jarník's work in combinatorial optimization. KAM Series 95, 315.
43. J.B.Kruskal: On the shortest spanning subtree of a graph and the travelling salesman problem. Proc.Amer.Math.Soc. 7(1856), 48–50.
44. M.Laurent, S.Poljak, F.Rendl: Connections between semidefinite relaxation of the Max Cut and Stable Set Problems. (to appear)
45. E.L.Lawler, J.K.Lenstra, A.H.G.Rinnooy Kan, D.B.Shmoys: The travelling salesman problem. John Wiley, 1985.
46. L.Lovász: On the Shannon capacity of a graph. IEEE Transactions on Information Theory 25(1979), 1–7.
47. L.Lovász: Randomized algorithms in combinatorial optimization. In: Combinatorial optimization (ed. W.Cook, L.Lovász, P.Seymour), DIMACS Series vol. 20, AMS 1995, 153–180.
48. L.Lovász, A.Schrijver: Cones of matrices and set-functions and 0-1 optimization. Siam Journal of Optimization 1(1991), 166–190.
49. L.Lovász, M.Plummer: Matching Theory. North Holland 1986.
50. J.Matoušek, M.Sharir, E.Welzl: A subexponential bound for linear programming, Proc. 8th Ann.Symp. on Comput. Geom. (1992),1–8.
51. J.Nešetřil: A few remarks on the history of MST-problem. Archivum Mathematicum 33(1997), 16–22.
52. J.Nešetřil, S.Poljak: Geomatrical and algebraical connections of combinatorial optimization. SOFSEM '80, Bratislava 1980, 35–78.
53. J.Nešetřil, D.Turzík: Solving and approximating Combinatorial Optimization Problems. (to appear).
54. M.W.Padberg, G.Rinaldi: Facet identification for the symmetric travelling salesman polytope. Math.Programming 47(1990), 219–257.
55. C.H.Papadimitriou: Euclidean TSP is NP-complete. Theoretical Computer Science 4(1977), 237–244.
56. C.H.Papadimitriou, M.Yannakakis: Optimization, approximation and complexity classes. Journal of Computer and System Sciences 43(1991), 425–440.
57. S.Poljak, D.Turzík: A polynomial algorithm for constructing a large bipartite subgraph with an application to satisfiability problem. Canadian Math. J. 24(1982), 519–524.
58. S.Poljak, D.Turzík: Maximum cut and circulant graphs. Discrete Math. 108(1992), 379–392.
59. S.Poljak, Zs.Tuza: Maximum cuts and largest bipartite subgraphs. In: Combinatorial Optimization (W.Cook, L.Lovász, P.Seymour, ed.), DIMACS Series vol.20, AMS(1995), 181–244.
60. P.Pudlák: Lower bounds for resolution and cutting plane proofs and monotonne computations. J. Symbol. Logic (to appear).
61. G.Reinelt: TSPLIB - A travelling salesman problem library. ORSA J.Computing 3(1991), 376–384.
62. G.Reinelt: Fast heuristics for large geometric travelling salesman problems. ORSA J.Comput. 4(1992), 206–217.
63. D.B.Shmoys: Computing near-optimal solution to combimatorial optimization problems. In: Combinatorial Optimization (W.Cook, L.Lovász, P.Seymour, ed.), DIMACS Series vol.20, AMS(1995), 355–398.
64. T.Schaefer: The complexity of satisfiability problems. Proceedings of the 10th ACM STOC, ACM, 1978.

65. P.Seymour: On the two coloring of hypergraphs. Quart.J.Math. Oxford 25(1975), 303–312.
66. R.E.Tarjan: Data structures and network algorithms. CBMS-NSF Regional Conf. Series in Applied Math., SIAM 44, 1983.
67. L.Trevisan: When Hamming meets Euclid: The approximability of geometric TSP and MST. In: Proceedings 29th STOC (1997), 21–29.
68. L.G.Valiant: The complexity of computing the permanent. Theoret.Comp.Sci. 8(1979), 189–201.
69. D.Welsh: Matroid theory. Academic Press 1976.