2005 International Symposium on Nonlinear
Theory and its Applications (NOLTA2005)
Bruges, Belgium, October 18-21, 2005

On Verified Computation in Combinatorial Optimization

Christian Jansson†

†Inst. of Computer Science III, TU Hamburg–Harburg
Schwarzenbergstraße 95, 21071 Hamburg, Germany
Email: jansson@tu-harburg.de

Abstract—Many current deterministic solvers for NP-hard combinatorial optimization problems are based on nonlinear relaxation techniques that use floating point arithmetic. Occasionally, due to solving these relaxations, rounding errors may produce erroneous results, although the deterministic algorithm should compute the exact solution in a finite number of steps. This may occur especially if the relaxations are ill-conditioned or ill-posed, and if Slater's constraint qualifications fail. We show how verified results can be obtained by rigorously bounding the optimal value of nonlinear semidefinite relaxations, even in the ill-posed case. All rounding errors due to floating point arithmetic are taken into account.

1. Introduction

Many algorithms in optimization require that appropriate rank conditions are fulfilled and that strictly feasible solutions of the primal and the dual problem exist; that is, it is assumed that *Slater's constraint qualifications* hold. The algorithms terminate if residuals that measure approximately the primal feasibility, the dual feasibility, and the duality gap are sufficiently small (see for example Mittelmann [10]). Small residuals often provide a small backward error, i.e. the computed solution is the exact solution of a slightly perturbed problem.

Nevertheless, there are many applications where backward error analysis may not be suitable. The first class consists of ill-conditioned problems with dependencies in the input data. The second class are ill-posed problems for which Slater's constraint qualifications are not fulfilled (see Gruber and Rendl [5], and Gruber et al. [4]). For such problems the solution does not depend continuously on the input data, and small perturbations can result in infeasibility and/or erroneous approximations.

Ill-conditioned and ill-posed problems are not rare in practice. In a recent paper, Ordóñez and Freund [13] stated that 71% of the lp-instances in the NETLIB Linear Programming Library are ill-posed. This library contains many industrial problems. Several problems become ill-posed due to the modelling (for example problems with redundant constraints, identically zero variables, and free variables transformed to variables bounded on one side), others appear as ill-posed relaxations in combinatorial optimization. Relaxations are widely used for solving difficult combinatorial problems efficiently with branch-bound-and-cut methods (see for example Goemans and Rendl [3]). We want to mention that backward error analysis is not generally applicable to this class of optimization problems. This is pointed out by Neumaier and Shcherbina [12]:

> *However, backward error analysis has no relevance for integer linear programs with integer coefficients, since slightly perturbed coefficients no longer produce problems of the same class.*

Moreover, they present an innocent-looking linear integer problem where the commercial, high quality state-of-the-art solvers CPLEX, BONSAIG, GLPK, XPRESS, XPRESS-MP/INTEGER, and MINLP failed. The reason is that the relaxations are not solved rigorously.

The major goal of this paper is to show how branch-and-bound algorithms for combinatorial optimization problems can be made safe, even if ill-posed semidefinite relaxations or cuts are used. We discuss this in the example case of Graph partitioning. Then, more general, we investigate *semidefinite programming problems* in *block diagonal form*:

$$f_p^* := \min \sum_{j=1}^{n} \langle C_j, X_j \rangle \quad \text{s.t.} \quad \sum_{j=1}^{n} \langle A_{ij}, X_j \rangle = b_i, \ i = 1, \ldots, m \quad X_j \geq 0, \ j = 1, \ldots, n. \tag{1}$$

where $C_j, A_{ij}, X_j \in S^{s_j}$, the linear space of real symmetric $s_j \times s_j$ matrices, and $b \in \mathbf{R}^m$. By $\langle ., . \rangle$ we denote the usual inner product on the linear space of symmetric matrices, which is defined as the trace of the product of two matrices. $X \succeq 0$ means that X is positive semidefinite. Hence, $\succeq$ denotes the *Löwner partial order* on this linear space. We assign $f_p^* := +\infty$ if the set of feasible solutions is empty.

If $s_j = 1$ for $j = 1, \ldots, n$ (i.e. C_j, A_{ij}, and X_j are real numbers), then (1) defines the standard linear programming problem. Therefore, semidefinite programming is a nonlinear extension of linear programming.

The *Lagrangian dual* of (1) is

$$f_d^* := \max b^T y \quad \text{s.t.} \ \sum_{i=1}^{m} y_i A_{ij} \preceq C_j \quad \text{for } j = 1, \ldots, n, \tag{2}$$

where $y \in \mathbf{R}^m$. We assign $f_d^* := -\infty$ if the set of dual feasible solutions is empty. The constraints $\sum_{i=1}^{m} y_i A_{ij} \preceq C_j$ are called *linear matrix inequalities (LMI's)*.

The problems satisfy the *weak duality* condition

$$f_d^* \le f_p^*. \tag{3}$$

Strong duality (i.e. $f_d^* = f_p^*$) requires additional conditions. It may happen that both optimal values are finite, but there is a nonzero duality gap and an optimal solution does not exist. Also it may happen that one optimal value is finite and the other one is infinite. If Slater's constrained qualifications are fulfilled, then both optimal values are finite and strong duality is fulfilled.

We present for semidefinite programming problems a rigorous lower bound of the primal optimal value and a rigorous upper bound of the dual optimal value. In most cases the required computational effort is small compared to the effort for computing approximate solutions. All rounding errors due to floating point arithmetic are rigorously estimated. It is of particular importance that these rigorous bounds can be used outside the code of any imaginable semidefinite solver as a reliable postprocessing routine.

Several presented results can be regarded as an extension of methods for linear programming (Jansson [7] and Neumaier and Shcherbina [12]) and convex programming [6] to the ill-conditioned and ill-posed case.

2. Graph Partitioning Problems

In this section we consider Graph Partitioning. These combinatorial problems are known to be NP-hard, and finding an optimal solution is difficult. Graph Partitioning has many applications, among those is VLSI design. There are varying branch-and-bound methods known using diverse relaxation techniques for solving this problem.

In a branch-and-bound framework the problem is recursively divided into subproblems, and each subproblem is relaxed to an optimization problem that is easier to solve and provides a lower bound of the optimal value for the original subproblem. Subproblems with a lower bound larger than the objective value of an already known feasible solution cannot contain a global optimal solution and are eliminated. Verified results in branch-and-bound are obtained if these bounds are computed rigorously, that is all rounding errors are regarded. The efficiency of branch-and-bound is essentially determined by the quality of the lower bounds. Because of the nonlinearity introduced by the positive semidefinite cone, semidefinite relaxations provide tighter bounds for many combinatorial problems than linear programming relaxations.

To simplify matters, we discuss here the special case of the Equicut Problem and the semidefinite relaxations proposed by Gruber and Rendl [5]. These have turned out to deliver tight lower bounds. General Graph Partitioning Problems can be treated similarly.

Given an edge-weighted graph G with an even number n of vertices, the problem is to find a partitioning of the vertices into two sets of equal cardinality which minimizes the sum of weights a_{ij} of the edges joining the two sets. The algebraic formulation is obtained by representing the partitioning as an integer vector $x \in \{-1,1\}^n$ satisfying the parity condition $\sum_i x_i = 0$. Then the Equicut Problem is equivalent to

$$\min \sum_{i<j} a_{ij}(1 - x_i x_j)/2 \quad \text{s.t.} \quad x \in \{-1,1\}^n,\ \sum_{i=1}^n x_i = 0,$$

where $A = (a_{ij})$ is the symmetric matrix of edge weights. This follows immediately, since $1 - x_i x_j = 0$ iff the vertices i and j are in the same set. The objective can be written as

$$\sum_{i<j} a_{ij}(1 - x_i x_j)/2 = (x^T L x)/4,$$

where $L := \mathrm{Diag}(Ae) - A$ is the *Laplace matrix* of G, and e is the vector of ones. Using $x^T L x = \mathrm{trace}(L(xx^T))$ and $X = xx^T$, it can be shown that this problem is equivalent to

$$\begin{aligned} &f_p^* = \min\langle L, X\rangle/4 \\ &\text{s.t.} \quad \mathrm{diag}(X) = e,\ e^T X e = 0,\ X \succeq 0,\ \mathrm{rank}(X) = 1. \end{aligned}$$

Since $X \succeq 0$ and $e^T X e = 0$ implies X to be singular, the problem is ill-posed, and for arbitrarily small perturbations of the right hand side it becomes infeasible.

By definition, the Equicut Problem has a finite optimal value f_p^*, and a rigorous upper bound of f_p^* is simply obtained by evaluating the objective function for a given partitioning integer vector x. In order to compute a rigorous lower bound, the nonlinear rank one constraint is left out yielding an ill-posed semidefinite relaxation, where the Slater's condition does not hold. Assume that $X = xx^T$ is an optimal solution, and let $\tilde{y} \in \mathbf{R}^{n+1}$. Then some computations yield

$$f_p^* = \langle L, X\rangle/4 \ge \sum_{i=1}^n \tilde{y}_i + \sum_{i=1}^n \lambda_i(D) q_i^T X q_i,$$

where the real symmetric matrix D is defined by

$$D := (1/4)L - \mathrm{Diag}(\tilde{y}_{1:n}) - \tilde{y}_{n+1}(ee^T),$$

and $\lambda_i(D)$ are the real eigenvalues with orthonormal eigenvectors q_i for $i = 1, \cdots, n$.

Since $X = xx^T$ with $x \in \{-1,1\}^n$ satisfies $\lambda_{\max}(X) = n$, and X is positive semidefinite it follows that the primal boundedness qualifications $0 \le q_i^T X q_i \le n$ for $i = 1, \cdots, n$ are fulfilled. Hence, the second sum can be bounded from below by

$$\sum_{i=1}^n \lambda_i(D) q_i^T X q_i \ge \sum_{i=1}^n \lambda_i(D)^- n.$$

where $\lambda_i(D)^- := \min(0, \lambda_i(D))$. Thus, we obtain

Theorem 1 *Let $\tilde{y} \in \mathbf{R}^{n+1}$, and assume that the matrix D has at most l negative eigenvalues, and let $\underline{d} \le \lambda_{\min}(D)$. Then*

$$f_p^* \ge \sum_{i=1}^n \tilde{y}_i + l \cdot n \cdot \underline{d}^- =: \underline{f}_p^*.$$

n	t	t_1	$\mu(\tilde{f}_d^*, \underline{f}_p^*)$
200	8.81	0.19	6.86788e-008
400	41.27	0.89	3.82904e-007
600	131.47	2.69	1.05772e-006

Table 1: Results for Graph Partitioning

Moreover, it can be shown that $\underline{f}_p^*$ is equal to the optimal value of the semidefinite relaxation, provided $\tilde{y}$ is the correct optimal Lagrange parameter vector (dual optimal solution). Semidefinite solvers usually compute approximate Lagrange parameter vector, and thus approximations $\tilde{y}$ close to these parameters produce a rigorous lower bound close to the optimal value, and the overestimation is negligible.

On a computer rounding errors occur, and the lower bound $\underline{f}_p^*$ must be computed rigorously. Therefore, a rigorous lower bound of $\underline{d}^-$ and an upper bound of l must be computed, and then the sum must be evaluated in the downward rounding mode. One possibility for computing $\underline{d}^-$ and l, is to calculate approximate orthonormal eigenvectors which are stored in a matrix Q. Then, with interval arithmetic the matrix QXQ^T is evaluated, and last with Gershgorin's Theorem the required bounds can be obtained. Another possibility (which we have implemented) is an approach due to Rump [14, 15], which was used for solving rigorously sparse linear systems. Further references for computing rigorous bounds of some or all eigenvalues and for interval arithmetic are Alefeld and Herzberger [1], Floudas [2], Neumaier [11].

In Table 1 we display some numerical results for problems which are given by Gruber and Rendl [5]. Matlab m-files can be found at *http://uni-klu.ac.at/groups/math/optimization/*. For this suite of ill-posed problems with up to 600 constraints and 180000 variables SDPT3 computes approximate lower bounds $\tilde{f}_d^*$ of the optimal value. The small relative errors $\mu(\tilde{f}_d^*, \underline{f}_p^*)$ show that the overestimation of the rigorous lower bound $\underline{f}_p^*$ can be neglected. SDPT3 terminates with $tc = 0$ (normal termination) for the first two examples. Only in the last case $n = 600$ the warning $tc = -5$*: Progress too slow* is returned, but a close rigorous lower bound is computed. Comparing the times t for computing the approximations with SDPT3 and t_1 for computing $\underline{f}_p^*$ with Theorem 1 one can see that the additional time t_1 for the rigorous lower bound is small compared to the time required for the approximations.

Summarizing, Theorem 1 facilitates cheap and rigorous lower bounds for the optimal value of graph partitioning problems. Similar results can be obtained for Quadratic Assignment Problems and Max Cut Problems.

3. Semidefinite Programming

In this section we describe for the semidefinite programming problem (1) two basic theorems which bound rigorously the optimal value. For further results about semidefinite programs and proofs we refer to [8].

Theorem 2 *Let $\tilde{X}_j \in S^{s_j}$ for $j = 1, \ldots, n$, and assume that each $\tilde{X}_j$ has at most k_j negative eigenvalues. Suppose that the following dual boundedness qualifications hold valid:*

(i) Either the dual semidefinite problem is infeasible,

(ii) or f_d^* is finite, and there are simple bounds $\overline{y} \in (\mathbf{R}_+)^m$, such that for every $\varepsilon > 0$ there exists a dual feasible solution $y(\varepsilon)$ satisfying

$$-\overline{y} \;\le\; y(\varepsilon) \le\; \overline{y}, \qquad \text{and} \qquad f_d^* - b^T y(\varepsilon) \;\le\; \varepsilon. \quad (4)$$

Let

$$r_i = b_i - \sum_{j=1}^{n} \langle A_{ij}, \tilde{X}_j \rangle \quad \text{for } i = 1, \ldots, m, \quad (5)$$

$$\underline{\lambda}_j \le \lambda_{\min}(\tilde{X}_j) \quad \text{for } j = 1, \ldots, n, \quad \text{and} \quad (6)$$

$$\begin{aligned} \varrho_j \ge \sup\{\, \lambda_{\max}(C_j - \sum_{i=1}^{m} y_i A_{ij}) : \\ -\overline{y} \le y \le \overline{y},\; C_j - \sum_{i=1}^{m} y_i A_{ij} \ge 0 \,\} \end{aligned} \quad (7)$$

for $j = 1, \ldots, n$. Then

$$f_d^* \le \sum_{j=1}^{n} \langle C_j, \tilde{X}_j \rangle - \sum_{j=1}^{n} k_j \underline{\lambda}_j^- \varrho_j + \sum_{i=1}^{m} |r_i| \overline{y}_i =: \overline{f}_d^*, \quad (8)$$

where $\underline{\lambda}_j^- := \min(0, \underline{\lambda}_j)$.

Theorem 3 *Let $\tilde{y} \in \mathbf{R}^m$ and assume that the following primal boundedness qualifications hold valid:*

(i) Either the primal semidefinite problem is infeasible,

(ii) or f_p^* is finite, and there are simple bounds $\overline{x} \in (\mathbf{R}_+)^n$ such that for every $\varepsilon > 0$ there exists a primal feasible solution $(X_j(\varepsilon))$ satisfying

$$\lambda_{\max}(X_j(\varepsilon)) \le \overline{x}_j \text{ for } j = 1, \ldots, n, \quad (9)$$

$$\text{and } \sum_{j=1}^{n} \langle C_j, X_j(\varepsilon) \rangle - f_p^* \le \varepsilon. \quad (10)$$

Let

$$D_j = C_j - \sum_{i=1}^{m} \tilde{y}_i A_{ij}, \text{ and } \underline{d}_j \le \lambda_{\min}(D_j) \text{ for } j = 1, \ldots, n. \quad (11)$$

Assume that D_j has at most l_j negative eigenvalues. Then

$$f_p^* \ge b^T \tilde{y} + \sum_{j=1}^{n} l_j \underline{d}_j^- \overline{x}_j =: \underline{f}_p^*. \quad (12)$$

In Theorem 2 upper bounds of the residuals $|r_i|$ are required. One possibility to calculate the residuals is to use interval arithmetic, and taking the supremum of the computed interval quantities. Perron-Frobenius theory can be used for computing an upper bound of ϱ_j. It follows that an appropriate upper bound is

$$\varrho_j = \varrho(|C_j| + \sum_{i=1}^{m} \overline{y}_i |A_{ij}|),$$

where ϱ denotes the spectral radius, which can be rigorously estimated by some norm. We want to mention that the previous theory also allows to consider problems with interval input data. Corresponding corollaries can be formulated in a canonical way by using the inclusion isotonicity principle of interval arithmetic.

In practice, there are frequently situations where details of modelling a problem or the generation of input data may not be known precisely, and may cause ill-posed problems. For example because of redundant constraints, identically zero variables, describing free variables as the difference of nonnegative variables, or replacing a vector by its outer product as in Section 4, the constraints do not satisfy Slater's constraint qualifications, but the boundedness of optimal solutions is not affected. Therefore, the previous theory may be used if either the user has a rough idea about the order of magnitude of the optimal solutions, or if he accepts that the absolute value of the optimal solutions is not much larger than the absolute value of the computed approximations multiplied by some positive factor, i.e., he trusts the order of magnitude:

$$\begin{array}{c} \overline{x}_j = \mu \cdot \lambda_{\max}(\tilde{X}_j) \text{ for } j = 1, \ldots, n, \\ \text{and } \overline{y}_i = \mu \cdot |\tilde{y}_i| \text{ for } i = 1, \ldots, m. \end{array}$$

These bounds can be viewed as a form of a-posteriori regularization for judging the computed approximate solution of an ill-posed optimization problem. Because this boundedness assumption is not (completely) verified, the results are not fully rigorous. Nevertheless, this stage of rigor is with rounding error control and we may speak of a *rounding error controlled weak verification*.

Further verified results for the problems in the test suites of optimization problems NETLIB LP and SDPLIB can be found in [8] and [9].

References

[1] G. Alefeld and J. Herzberger. *Introduction to Interval Computations*. Academic Press, New York, 1983.

[2] C.A. Floudas. *Deterministic Global Optimization - Theory, Methods and Applications*, volume 37 of *Nonconvex Optimization and Its Applications*. Kluwer Academic Publishers, Dordrecht, Boston, London, 2000.

[3] M. Goemans and F. Rendl. Combinatorial Optimization. In H. Wolkowicz et al., editor, *Handbook of Semidefinite Programming*. Kluwer Academic Publishers, 2000.

[4] G. Gruber, S. Kruk, F. Rendl, and H. Wolkowicz. Presolving for Semidefinite Programs Without Constraint Qualifications. In G. Gruber et al., editor, *Proceedings of HPOPT97, Second Workshop on High Performance Optimization Techniques, Rotterdam, Netherlands*, 1997.

[5] G. Gruber and F. Rendl. Computational experience with ill-posed problems in semidefinite programming. *Computational Optimization and Applications*, 21(2):201–212, 2002.

[6] C. Jansson. A rigorous lower bound for the optimal value of convex optimization problems. *J. Global Optimization*, 28:121–137, 2004.

[7] C. Jansson. Rigorous Lower and Upper Bounds in Linear Programming. *SIAM J. Optim.*, 14(3):914–935, 2004.

[8] C. Jansson. Termination and Verification for Ill-posed Semidefinite Programming Problems, 2005. `http://optimization-online.org/DB_HTML/2005/06/1150.html`.

[9] C. Keil and C. Jansson. Computational Experience with Rigorous Error Bounds for the Netlib Linear Programming Library. *Reliable Computing*, 2005 (to appear).

[10] H.D. Mittelmann. An independent benchmarking of SDP and SOCP solvers. *Math. Programming Ser. B*, 95:407–430, 2003.

[11] A. Neumaier. *Interval Methods for Systems of Equations*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1990.

[12] A. Neumaier and O. Shcherbina. Safe bounds in linear and mixed-integer programming. *Mathematical Programming, Ser. A*, 99:283–296, 2004.

[13] F. Ordóñez and R.M. Freund. Computational experience and the explanatory value of condition measures for linear optimization. *SIAM J. Optimization*, 14(2):307–333, 2003.

[14] S.M. Rump. Validated Solution of Large Linear Systems. In R. Albrecht, G. Alefeld, and H.J. Stetter, editors, *Validation numerics: theory and applications*, volume 9 of *Computing Supplementum*, pages 191–212. Springer, 1993.

[15] S.M. Rump. Verification Methods for Dense and Sparse Systems of Equations. In J. Herzberger, editor, *Topics in Validated Computations — Studies in Computational Mathematics*, pages 63–136, Elsevier, Amsterdam, 1994.