Subgraph Matching with Semidefinite Programming

C. Schellewald [a,*] C. Schnörr [a]

[a] *University of Mannheim, Dept. Math. and Comp. Science, CVGPR-Group, D–68131 Mannheim, Germany*
{cschelle, schnoerr}@uni-mannheim.de, www.cvgpr.uni-mannheim.de

Abstract

We present a convex programming approach to the problem of matching subgraphs which represent object views against larger graphs which represent scenes. Starting from a linear programming formulation for computing optimal matchings in bipartite graphs, we extend the linear objective function in order to take into account the relational constraints given by both graphs. The resulting combinatorial optimization problem is approximately solved by a semidefinite program. Preliminary results are promising with respect to view-based object recognition subject to relational constraints.

Key words: subgraph matching, combinatorial optimization, semidefinite programming, object recognition

1 Introduction

Object recognition is a key problem in computer vision. Among the various representations of objects [1], view- or appearance-based representations play an important role. This fact is also consistent with psychophysics.

A common and powerful structure for representing views of objects is to define a set of local image features V along with pairwise relations $E \subset V \times V$ (encoding spatial proximity and a (dis)similarity measure) in terms of a weight function $w : E \to \mathbb{R}_+$, that is an undirected weighted graph $G = (V, E)$. Graphs representing object views are called *model graphs* in this paper. In the same way as model graphs, *scene graphs* are computed by extracting local

* Corresponding author.

image features and spatial relationships in a preprocessing step. Thus, in order to localize and recognize objects in an image, model graphs have to be matched against scene graphs.

In this paper, we focus on the combinatorial problem to match model graphs against scene graphs. We do not discuss image preprocessing but assume the model and scene graphs to be given. Work on graph-matching includes a wide range of algorithmic approaches like neural networks [3], probabilistic relaxation [4], genetic search [5], error-correcting matching [6] or two-step iterative approaches [7], and also more specialized work like, for example, simultaneous estimation of transformation geometry [8], or matching trees in terms of the maximum clique of the association graph [9].

The primary motivation for our work is the design of algorithms for which the performance does not critically depend on the choice of tuning parameters, since the automatic choice of suitable parameter values is rather difficult in the context of computer vision systems. To this end, in previous work [11,12] we studied a semidefinite programming approach [10] to the quadratic assignment problem. In our context, the quadratic assignment problem [13] corresponds to the relational matching of two graphs with an *equal* number of vertices. A global solution to the semidefinite relaxation can be computed by interior point methods [14] which in turn leads to a good local minimum by solving an additional linear program. Thus overall approach does not involve any tuning parameter.

The competitive performance with respect to deterministic annealing approaches [15,16], the performance of which *does* depend on the choice of various parameter values, motivates to investigate a convex programming approach for the more involved *sub*graph matching problem as well. Unfortunately, the semidefinite relaxation introduced in [10] is no longer applicable in this case. Therefore, we present a novel approach to subgraph matching based on convex programming.

Starting from bipartite graph matching (Section 2.2), we complement the objective criterion with quadratic terms favouring bipartite matchings which respect the relational structure in both the model graph and the scene graph (Section 3). In Section 4, a semidefinite relaxation is developed. Numerical experiments are discussed in Section 5. We point out again that our focus is on the development of a subgraph matching algorithm on the basis of convex programming techniques in this paper. Therefore, we do not consider any issues related to image preprocessing and assume the model and scene graphs to be given like depicted in figure 1.

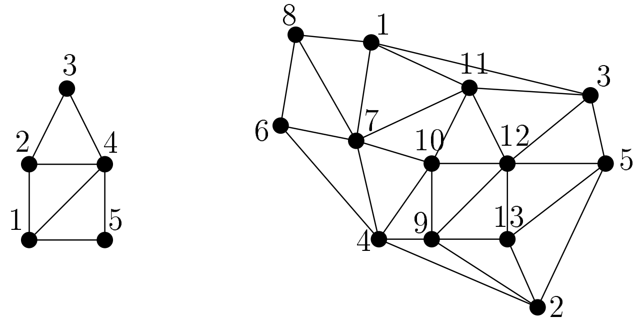

Fig. 1. Model and scene graph with $K = 5$ and $L = 13$ vertices respectively.

2 Notation and Bipartite Matching

In this section, we list some basic notation and provide the starting point of our approach, the matching problem in bipartite graphs.

2.1 Notation

We will use the following notation throughout this paper:

$x^\top$	transpose of x
I_n	$n \times n$ unit matrix
e	vector of all ones: $e_i = 1, \quad i = 1, \ldots, n$
E_{nn}	Matrix of all ones: $E_{nn} = ee^\top$
$\mathrm{Tr}[X]$	trace of the matrix X
$A \otimes B$	Kronecker product of A and B

2.2 Matching in Bipartite Graphs

In this paper, we consider undirected graphs $G = (V, E)$ with nodes $V = \{1, \ldots, n\}$ and edges $E \subset V \times V$. We denote the model graph with G_K and the scene graph with G_L. The corresponding sets V_K and V_L contain $K = |V_K|$ and $L = |V_L|$ vertices respectively. We assume $L \geq K$. Furthermore, we assume a distance function $w(i, j)$ to be given which measures the similarity of each pair of vertices $i \in V_K$ and $j \in V_L$.

If we ignore the structure in both the model graph and the scene graph, then an optimal assignment of the K vertices of the model graph can be easily found as a matching in the bipartite graph $(V_K \cup V_L, E)$, with edges $(i, j) \in E$, defined for all pairs $i \in V_K, j \in V_L$ with corresponding weights $w(i,j)$.

Let $x \in \{0,1\}^{KL}$ denote the indicator vector of the edges, with its components arranged as follows:

$$x = (x_{11}, \cdots x_{1L}, x_{21} \cdots x_{2L}, \cdots, x_{K1} \cdots x_{KL})^\top. \tag{1}$$

Thus x starts with L edges connecting some fixed vertex of V_K with all vertices in V_L followed by K edges with respect to a second vertex in V_K, etc. With a slight abuse of notation we denote the corresponding weight vector $(w(1,1), \ldots, w(K,L))^\top$ again with w. According to this order, the incidence matrix of G has the block structure $A = (A_K^\top, A_L^\top)^\top$. Then an optimal matching can be found by solving the linear program:

$$\begin{aligned} &\min_x w^\top x \ , \ x \in \{0,1\}^{KL} \\ &A_K x = e_K \ , \ A_L x \leq e_L \end{aligned} \tag{2}$$

It is well known that the incidence matrix of a bipartite graph is totally unimodular [17]. As a consequence, solving the linear program (2) for $x \in \mathbb{R}^{KL}$ gives a globally optimal integer solution $x \in \{0,1\}^{KL}$. As mentioned above, however, this particular situation has been achieved by ignoring the relational structures in both the object representation (model graph) and the image (scene graph).
In the next section, we extend the approach (2) to include relational constraints, and propose a convex programming approach which is favourable from the computational viewpoint, too.

3 A Quadratic Integer Program for Subgraph Matching

To incorporate the relational structure of both the model graph and the scene graph, we complement the linear objective function in (2) with a quadratic term:

$$\min_x w^\top x + \alpha x^\top Q x \ , x \in \{0,1\}^{KL} \tag{3}$$

The parameter $\alpha \in \mathbb{R}$ controls the influence of these additional costs.

The quadratic term in (3) involves the adjacency matrices N_K, N_L of the model graph and the scene graph, respectively, which encode the neighbourhood structure in these two graphs. For example, the adjacency matrix N_K for the model graph depicted in figure 1 is given by:

$$N_K = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

We also define the complementary adjacency matrices:

$$\begin{aligned} \bar{N}_L &= E_{LL} - N_L - I_L \\ \bar{N}_K &= E_{KK} - N_K - I_K \end{aligned}$$

With this notation and referring to the order of the set of edges defined in (1), the matrix Q in (3) is given by:

$$Q = N_K \otimes \bar{N}_L + \bar{N}_K \otimes N_L \tag{4}$$

We next explain in turn the two terms on the right handside in (4).

The first term can be written as:

$$x^\top (N_K \otimes \bar{N}_L) x = \sum_{ar}^{KL} \sum_{bs}^{KL} (N_K)_{ab} (\bar{N}_L)_{rs} x_{ar} x_{bs}$$

The interpretation of this term is that if two vertices a and b in the model graph are neighbors, $(N_K)_{ab} = 1$, then a good assignment (no costs) involves corresponding vertices r and s in the scene graph which are neighbors, too: $(\bar{N}_L)_{rs} = 0$. This is visualized in figure 2.

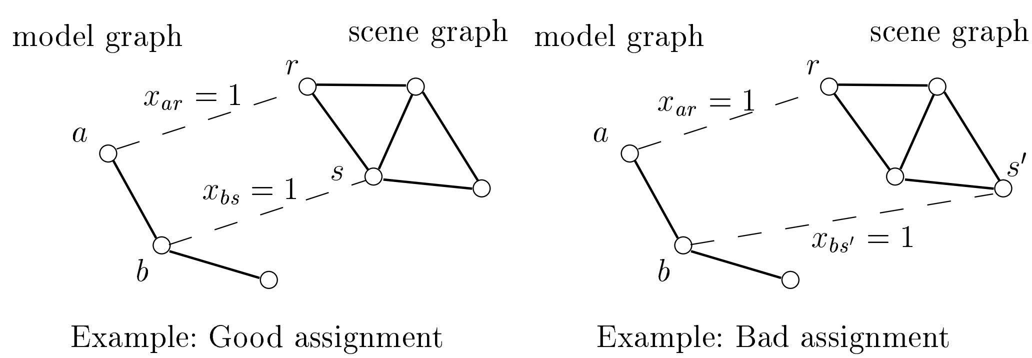

Fig. 2. **Left:** Adjacent vertices a and b in the model graph are assigned to adjacent vertices r and s in the scene graph. **Right.** Adjacent model vertices a and b are no longer adjacent in the scene graph after the assignment.

Analogously, the second part of Q gives:

$$x^\top(\bar{N}_K \otimes N_L)x = \sum_{ar}^{KL}\sum_{bs}^{KL}(\bar{N}_K)_{ab}(N_L)_{rs}x_{ar}x_{bs}$$

This term penalizes assignments where pairs of vertices become neighbors which weren't adjacent before. Figure 3 illustrates this possibilities in detail.

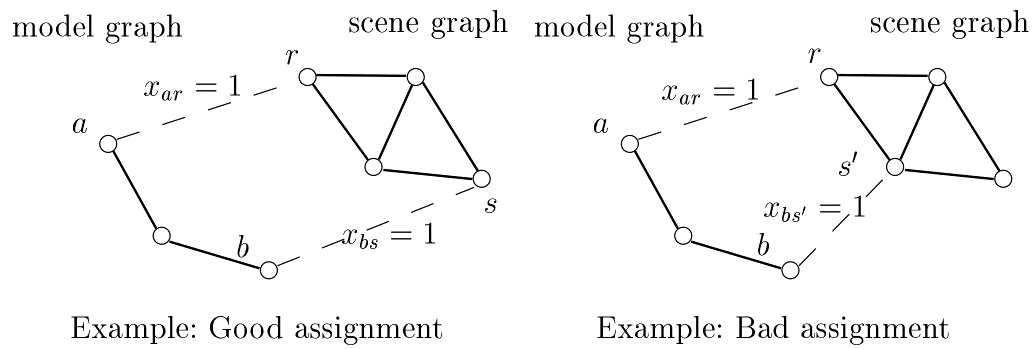

Fig. 3. **Left:** Vertices a and b which are not adjacent are assigned to vertices which are not adjacent, too. **Right:** A pair of vertices a and b become neighbors r and s' after assignment.

4 Optimization and Semidefinite Programming

In contrast to (2), computation of the global optimum of (3) is intrinsically difficult. We next derive a convex programming approach to compute a "good" local minimum. For more information on semidefinite programming we refer to [18].

First, we reformulate the objective criterion (3) as a homogeneous quadratic form:

$$w^\top x + \alpha x^\top Q x \;\rightarrow\; \mathrm{Tr}\left[\underbrace{\begin{pmatrix} 0 & \frac{1}{2}w^\top \\ \frac{1}{2}w & \alpha Q \end{pmatrix}}_{\tilde{Q}} \begin{pmatrix} 1 & x^\top \\ x & xx^\top \end{pmatrix}\right] \;\rightarrow\; \mathrm{Tr}\left[\tilde{Q}X\right] \;,$$

Thus, the unknown variables are replaced by a matrix X which is symmetric, positive definite, and has rank 1. Dropping this last constraint makes the set of feasible matrices X convex!

Furthermore, we incorporate the following linear constraints which still lead to a convex optimization problem:

- Integer constraint $x_i \in \{0,1\}, \forall i$: This constraint is *weakly* enforced by constraining the first column and row of X to be equal to its diagonal (since $x_i^2 = x_i, \forall i$).
- Sum constraint: According to the ordering introduced in (1), we translate the constraints $\sum_{j=1}^{L} x_{ij} = 1, \forall i$, into the corresponding constraints with respect to the diagonal elements of X.
- Matching constraints: The constraints in (2) are translated to constraints with respect to X by inspecting the following two terms:

$$x^\top (I_K \otimes (E_{LL} - I_L))x = \sum_{ar}^{KL} \sum_{bs}^{KL} (I_K)_{ab}(E_{LL} - I_L)_{rs} x_{ar} x_{bs}$$
$$x^\top ((E_{KK} - I_K) \otimes I_L)x = \sum_{ar}^{KL} \sum_{bs}^{KL} (E_{KK} - I_K)_{ab}(I_L)_{rs} x_{ar} x_{bs}$$

  The first of these two terms penalizes non-unique assignments of model vertices to scene vertices. Analogously, the second term penalizes assignments where model vertices are mapped to the same vertex in the scene graph. Thus, in summary, the two terms penalize all assignments which do no lead to a matching, i.e. a subset of non-incident edges. Figure 4 illustrates such configurations in detail. Conversely, all vectors x representing a matching result in the value zero of the above two quadratic forms. Accordingly, we constrain the corresponding entries in X to be zero.

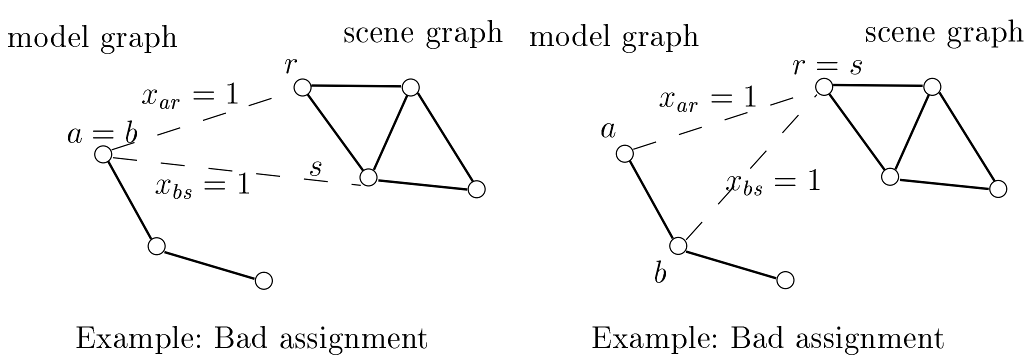

Fig. 4. Assignments which do not lead to matchings are penalized by the quadratic optimization criterion.

The linear objective function and the linear constraints described above lead to a semidefinite program of the following form:

$$\begin{aligned} \min\ & \mathrm{Tr}\left[\tilde{Q}X\right] \\ \text{s.t. } & \mathrm{Tr}[A_1X] = c_1 \\ & \mathrm{Tr}[A_2X] = c_2 \\ & \vdots \\ & \mathrm{Tr}[A_mX] = c_m \\ & X \succeq 0 \end{aligned} \tag{5}$$

The last constraint in (5) says that X has to be positive semidefinite. We wish to emphasize once more that (5) is a convex optimization problem.

Once the global optimum of (5) is computed with an interior point solver, the diagonal elements of the solution X can be interpreted as a non-integer solution x_{sol} to (3). To obtain a 0,1-integer solution representing an admissible matching we solve the following linear programming in a postprocessing step:

$$\begin{aligned} \max_x\ & x_{sol}^\top x \ , x \in \{0,1\}^{KL} \\ \text{s.t. } & A_K x = e_K \ , \\ & A_L x \leq e_L \end{aligned} \tag{6}$$

5 Preliminary Experimental Results

In this section we present a promising preliminary result of the convex subgraph matching approach. Figure 1 depicts the graph structure used in the experiment: The object consists of 5 vertices which are the corners of a simple house (see figure 1, left). Then the full scene with $L = 13$ vertices was obtained by adding 8 background vertices (figure 1, right). The structure of the graphs is the result of two Delauney triangulations of the model vertices and the scene vertices. We defined three different cost ranges for the possible assignments of the model vertices to the scene vertices:

expensive	1.0 – 1.25
cheap	0.5 – 0.75
very cheap	0.5

For the experiment all assignment costs $w(i,j)$ of the vector w are selected randomly within the cheap range or within the expensive range, depending on the fact whether the model vertex i fits to the scene vertex j or not. In our example the following assignments are chosen to be cheap: $1 \mapsto 9,\ 2 \mapsto 10,\ 3 \mapsto 11,\ 4 \mapsto 12,\ 5 \mapsto 13$. All other assignments are defined to be expensive. Additionally, some assignments from model vertices to the background are made very cheap artificially: $2 \mapsto 1,\ 3 \mapsto 8,\ 4 \mapsto 4,\ 5 \mapsto 2$.

The linear optimization for the problem (2) leads to a 0,1-solution x which assigns the model vertices to the locally best fitting scene vertices without considering the structure of the graphs. As shown in figure 5 the linear approach results in a non-desired matching.

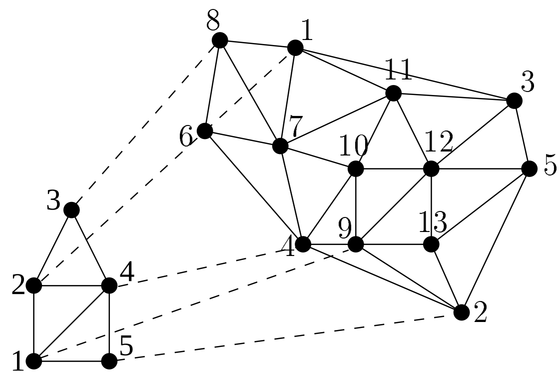

Fig. 5. Linear solution

In contrast to that the desired matching can be obtained by the semidefinite programming approach. The result is shown in figure 7. Figure 6 shows the non integer solution x_{sol} which is equal to the diagonal (without the first element) of the solution X of (5), plotted against the index. The plot is subdivided into five segments, with each segment representing all possible matchings from one vertex of the model graph to all L vertices in the scene graph. In the first segment only the assignment from model vertex 1 to scene vertex 9 has a value of nearly 1 and is therefore selected ($x_{19} = 1$) in the integer solution. The important observation in figure 6 is that for each segment, only a small number of candidates for a matching has a larger value. One can also see that each segment sums up to 1, according to the constraint $A_K x = e_K$. With (6) the integer solution x which represents a bipartite matching is calculated.

Figure 7 also shows the candidate matchings for the last three vertices of the model graph, by dotted lines.

6 Conclusion and Further Work

In this paper we presented a convex programming approach for the problem of subgraph matching. For this purpose we extended the linear programming formulation for computing optimal matchings in bipartite graphs by adding

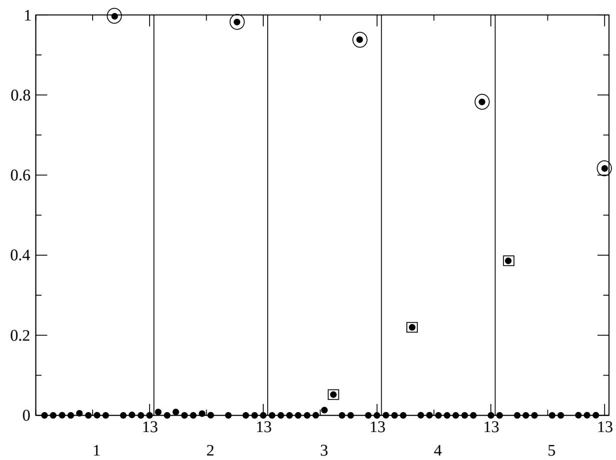

Fig. 6. Non integer solution vector x_{sol} ($\alpha = 0.10$)

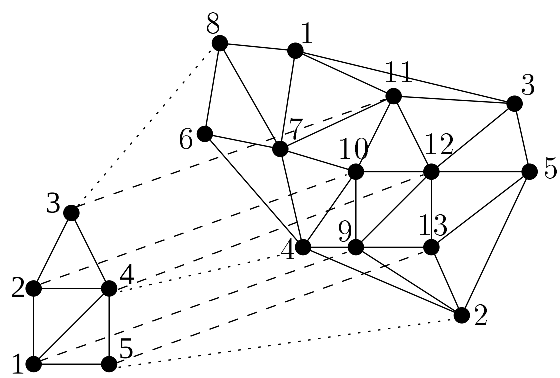

Fig. 7. solution

a quadratic term which comprises the relational constraints given by both graphs. The advantage of the convex approach is that it does not need any additional tuning parameters. As first experimental results are promising, a more detailed investigation of this approach will be beneficial.

References

[1] M. Herbert, J. Ponce, T. Boult, and A. Gross, editors. *Object Representation in Computer Vision*, volume 994 of *Lect. Not. Comp. Sci.* Springer-Verlag, 1995.

[2] H.H. Bülthoff and S. Edelman. Psychophysical support for a two-dimensional view interpolation theory of object recognition. *Proc. Nat. Acad. Science,*, 92:60–64, 1992.

[3] W.K. Konen, T. Maurer, and C. von der Malsburg. A fast dynamic link matching algorithm for invariant pattern recognition. *Neural Networks*, 7(6/7):1019–1030, 1994.

[4] W.J. Christmas, J. Kittler, and M. Petrou. Structural matching in computer

vision using probabilistic relaxation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 17(8):749–764, 1995.

[5] A.D.J. Cross, R.C. Wilson, and E.R. Hancock. Inexact graph matching using genetic search. *Pattern Recog.*, 30(6):953–970, 1997.

[6] B.T. Messmer and H. Bunke. A new algorithm for error-tolerant subgraph isomorphism detection. *IEEE Trans. Patt. Anal. Mach. Intell.*, 20(5):493–504, 1998.

[7] B. Luo and E.R. Hancock. Structural graph matching using the em algorithm and singular value decomposition. *IEEE Trans. Patt. Anal. Mach. Intell.*, 23(10):1120–1136, 2001.

[8] A.D.J. Cross and E.R. Hancock. Graph-matching with a dual-step em algorithm. *IEEE Trans. Patt. Anal. Mach. Intell.*, 20(11):1236–1253, 1998.

[9] M. Pelillo, K. Siddiqi, and S.W. Zucker. Matching hierarchical structures using association graphs. *IEEE Trans. Patt. Anal. Mach. Intell.*, 21(11):1105–1120, 1999.

[10] N. Brixius and K. Anstreicher. Solving quadratic assignment problems using convex quadratic programing relaxations. *Optimization Methods and Software*, 16(1–4):49–68, 2001.

[11] C. Schellewald, S. Roth, and C. Schnörr. Evaluation of convex optimization techniques for the weighted graph–matching problem in computer vision. In B. Radig and S. Florczyk, editors, *Mustererkennung 2001*, volume 2191 of *Lect. Notes Comp. Science*, pages 361–368, Munich, Germany, Sept. 12–14 2001. Springer.

[12] C. Schellewald, S. Roth, and C. Schnörr. Performance evaluation of a convex relaxation approach to the quadratic assignment of relational object views. Comp. Science Series, Technical Report 02/2002, Dept. Math. and Comp. Science, University of Mannheim, Germany, Feb. 2002, submitted.

[13] R.E. Burkard, E. Cela, P.M. Pardalos, and L.S Pitsoulis. The quadratic assignment problem. In P.M. Pardalos and D.-Z. Du, editors, *Handbook of Combinatorial Optimization*, pages 241–338. Kluwer Acad. Publishers, 1998.

[14] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[15] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Trans. Patt. Anal. Mach. Intell.*, 18(4):377–388, 1996.

[16] S. Ishii and M. Sato. Doubly constrained network for combinatorial optimization. *Neurocomputing*, 2001. to appear.

[17] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. J. Wiley, New York, 1988.

[18] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of Semidefinite Programming*, Boston, 2000. Kluwer Acad. Publ.