

Minconvex Factors of Prescribed Size in Graphs*

Nicola Apollonio [†], András Sebó [‡]

January 24, 2006

Abstract

We provide a polynomial algorithm that determines for any given undirected graph, positive integer k and a separable convex function on the degree sequences, k edges that minimize the function. The motivation and at the same time the main application of the results is the problem of finding a subset of k vertices in a line graph, that cover as many edges as possible (we called it maxfix cover even though k is part of the input), generalizing the vertex cover problem for line graphs, equivalent to the maximum matching problem in graphs.

The usual improving paths or walks for factorization problems turn into edge-disjoint unions of pairs of such paths for this problem. We also show several alternatives for reducing the problem to b -matching problems – leading to generalizations and variations.

The algorithms we suggest also work if for any subset of vertices, upper, lower bound constraints or parity constraints are given. In particular maximum (or minimum) weight b -matchings of given size can be determined in polynomial time, combinatorially, in more than one way, for arbitrary b . Furthermore, we provide an alternative "gadget reduction" method to solve the problems which leads to further generalizations, provides ideas to alter the

*A preliminary shortened version with part of the results (and a small part of the proofs) has appeared in the IPCO 2004 volume: LNCS 3064 (D. Bienstock and G. Nemhauser, eds.).

[†]University "La Sapienza" di Roma, supported by the European project ADONET while the author visited Laboratoire Leibniz (IMAG), Grenoble, Nicola.Apollonio@uniroma1.it

[‡]CNRS, Leibniz-IMAG, Grenoble, France, Andras.Sebo@imag.fr

usual optimization problems themselves, and solves the arising problems in polynomial time. We also show the limits of the methods by proving the NP-completeness of some direct extensions, in particular of the generalization to all convex functions.

1 Introduction

Let $G = (V, E)$ be a graph that may contain loops and parallel edges, and let $k > 0$ be an integer. The main result of this work is to provide a polynomial algorithm for finding a subgraph of cardinality k that minimizes some pre-given objective function on the edges or the degree sequences of the graph. The main example will be the sum of the squares of the degrees (minsquare problem) showing how the algorithm works for the sum of any one dimensional convex function of the degrees (Section 2.3), including also linear functions (Section 2.5). This is the *minconvex* factor problem. The sum of squares function is general enough to exhibit the method in full generality, and at the same time concrete enough to facilitate understanding, moreover this was originally the concrete problem we wanted to solve.

It also arises in a natural way in the context of vertex-covers of graphs, and this was our starting point:

given a graph and an integer t (as part of the input) find a subset of vertices of cardinality t that cover the most number of edges in a graph, that is find a *maxfix cover*. This problem, introduced by Petrank in [16] under the name of *max vertex cover*, obviously contains the vertex cover problem, so it is NP-hard in general. However, VERTEX COVER is polynomially solvable for line graphs (it is straightforwardly equivalent to the maximum matching problem). What about the maxfix cover problem in line graphs ?

The following figure shows such a maxfix cover: The vertices of G are represented with small balls; nine edges of $L(G)$ — that cover altogether 60 edges— have a big ball in their middle. (In Maxfix Lineland candles of birthday cakes are put in the middle of edges of a graph drawn on the cake. A candle in the middle of an edge e selects all the incidences of that edge to have a (fictitious) candle (that is, the total number of candles is equal to the number of covered edges of the line graph). The figure shows how to celebrate 60th birthdays with only 9 candles.

A maxfix cover for $L(G)$ is a set $T \subseteq E(G)$ minimizing the number of

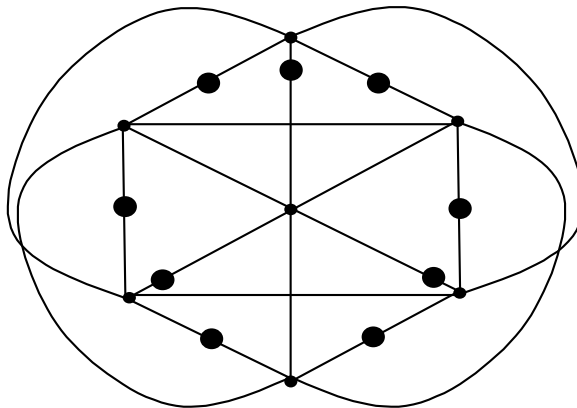


Figure 1: A sixtieth birthday cake with nine candles

incident pairs of edges in the remaining edge-set $F = E(G) \setminus T$, $|F| = k := n - t$. Clearly, the number of such pairs is $\sum_{v \in V} \binom{d_F(v)}{2}$. Since the sum of the degrees is constant, this is equivalent to minimizing $\sum_{v \in V} d_F^2(v)$. This sum will be called the value of F . A subgraph of k edges will be called *optimal* if it minimizes this value, and the problem of finding an optimal subgraph of k edges will be called the *minsquare* problem. The main result of this work is to provide a polynomial algorithm for solving this problem.

Let us introduce some notation and terminology used throughout the paper. Let G be a graph. Then $n := n(G) := |V(G)|$; $E(X)$ ($X \subseteq V(G)$) is the set of edges induced by X , that is, with both endpoints in X ; $\delta(X)$ denotes the set of edges with exactly one endpoint in X . For $X \subseteq V(G)$ let $d(X) := |\delta(X)|$. We will not distinguish subgraphs from subsets of edges. For a subgraph $F \subseteq E(G)$ let $d_F(v)$ ($v \in V$) be the *degree* of v in F , that is, the number of edges of F incident to v . The maximum degree of G will be denoted by Δ_G . The line graph of G will be denoted by $L(G)$. The *Euclidean norm* of a vector $a \in \mathbb{R}^n$, denoted by $\|a\|$, is the number $\sqrt{\sum_{i=1}^n a_i^2}$ (thus $\|a\|^2 = \sum_{i=1}^n a_i^2$). The l_1 *norm* of a , denoted by $|a|$, is the number $|a| := \sum_{i=1}^n |a_i|$.

Given $b : V(G) \rightarrow \mathbb{N}$, a *b-matching* (also known as simple *b-matching* or *b-factor*) is a subset of edges $F \subseteq E(G)$ such that $d_F(v) = b(v)$ for every $v \in V(G)$; b is a *degree sequence* (in G) if there exists a *b-matching* in G .

More generally, an (f, g) -matching, where $f, g : V(G) \rightarrow \mathbb{N}$, is $F \subseteq E(G)$ with $f(v) \geq d_F(v) \geq g(v)$ for all $v \in V(G)$.

In the same way as minimum vertex covers are exactly the complementary sets of maximum stable sets, maxfix covers are the complementary sets of ‘minfix induced subgraphs’, that is, of sets of vertices of pre-given cardinality that induce the less possible edges. (Ad extrema 0 edges, when the decision version of the minfix induced subgraph problem with input k specializes to answering the question ‘ $\alpha \geq k$?’.) Similarly, minfix covers are the complements of maxfix induced subgraphs.

As we will see in 4, of all these variants the only problem that can be solved in relatively general cases is maxfix cover. The others are NP-hard already in quite special cases.

The maxfix cover problem has been even more generally studied, for hypergraphs: find a set of vertices of given size $t \in \mathbb{N}$ that hits a maximum cardinality (highest weight) set of hyperedges. For *Edge-Path* hypergraphs, that is hypergraphs whose vertices are the edges of a given underlying graph G and whose set of hyperedges is a given family of weighted paths in G , several results have been achieved:

In [1] and [2] polynomial algorithms have been worked out for special underlying graphs (caterpillars, rooted arborescences, rectangular grids with a fixed number of rows, etc.) and for special shaped collection of paths (staple-free, rooted directed paths, L -shaped paths, etc.), and it has been shown that the problem is NP-complete for a fairly large set of special Edge-Path hypergraphs. When the Edge path hypergraph has the form (G, \mathcal{P}) , \mathcal{P} being the family of all paths of length two in G , the problem is a maxfix cover problem in the line graph $L(G)$ of G .

Until the last section we will state the results in terms of the minsquare problem, because of the above mentioned application, and the general usability and intuitive value of the arguments.

A support for the polynomial solvability of the minsquare problem is general convex optimization. It is well-known [10] that convex function can be minimized in polynomial time over convex polytopes under natural and general conditions. Hence convex functions can be optimized over ‘ b -matching polytopes’ and intersections of such polytopes with hyperplanes (or any other solvable polyhedron in the sense of [10]). However, the optima are not necessarily integer, neither when minimizing over the polytope itself, nor for the intersection.

Minimizing a convex function over b -matchings, that is the integer points of the b -matching polytope, is still easy with standard tools: single improving paths suffice, and the classical algorithms for finding such paths [9], [17] do the job. However, for our problem, where the set of b -matchings is intersected with a hyperplane, single paths do not longer suffice (see at the end of this Introduction); yet we will show that pairs of paths along which a non-optimal solution can be improved do always exist and yield a polynomial algorithm.

In this way the integer optimum of a quite general convex function can still be determined in polynomial time over the intersection of (f, g) -matching polyhedra with hyperplanes. This is less surprising in view of the following considerations.

Intuitively, the best solution is the ‘less extremal’ one. Clearly, if $r := 2k/n$ is an integer and G has an r -regular subgraph, then it is an optimal solution of the minsquare problem. This is the ‘absolute minimum’ in terms of k and n . The existence of an r -regular subgraph is polynomially decidable (with the above mentioned methods) which makes the problem look already hopeful: it can be decided in polynomial time whether this absolute minimum can be attained or not.

If $2k/n$ is not integer, it is also clear to be uniquely determined how many edges must have degree $\lceil 2k/n \rceil$ and how many $\lfloor 2k/n \rfloor$ in a subgraph, so as the sum of the degrees of the subgraph is $2k$. However, now it is less straightforward to decide whether this absolute minimum can be attained or not, since the number of all cases to check may be exponential. On the example of Figure 1 we have $k = 9$, $n = 7$ so the absolute optimum consists of four vertices of degree 3 and three vertices of degree 2. It is attained by the nine selected edges (with the candles) showing that they form an optimal cover. At first sight the general problem may appear hopeless.

Yet the main result of the paper states that a subgraph F is optimal *if and only if there is no vector $t : V(G) \rightarrow \mathbb{N}$ such that:*

- t is a degree sequence in G .
- $\sum_{v \in V} d_F(v) = \sum_{v \in V} t(v)$, that is each t -matching has the same size as F ,
- $\sum_{v \in V} |d_F(v) - t(v)| \leq 4$, that is t differs from F by at most 4 in l_1 -norm), and

– $\sum_{v \in V} t^2(v) < \sum_{v \in V} d_F^2(v)$, that is, t has better objective value than F .

Since the number of vectors v that satisfy the last three conditions is smaller than n^4 , and it can be decided for each whether it satisfies the first condition by classical results of Tutte, Edmonds-Johnson, and various other methods (see accounts in [9], [17], [13]), the result implies a polynomial algorithm for the minsquare problem.

If t satisfies the above four conditions, the function (vector) $\kappa := t - d_F$ will be called an *improving* vector with respect to F . We have just noticed the following.

- (1) *If an improving vector exists it can be found in polynomial time.*

The graph G consisting of two vertex-disjoint triangles shows that one cannot replace 4 by 2 in the second condition, unlike in most of the other factorization problems. Indeed, choose $k = 4$, and let F contain the three edges of one of the triangles and one edge from the other. The value of this solution is 14, the optimum is 12 and one has to change the degree of at least four vertices to improve (Figure 2). Optimizing linear functions over the degree sequences of subgraphs of requested cardinality k (part of the input) is not algorithmically simpler than the results we prove (even if the statements that make this possible can be made easier in this case see Section 2.5), and the same results hold for a quite general set of objective functions (see Section 2.3). If we have put in the center minsquare factors it is because the origins of the problem and because we still find them to be a representative example of the new problems we can solve.

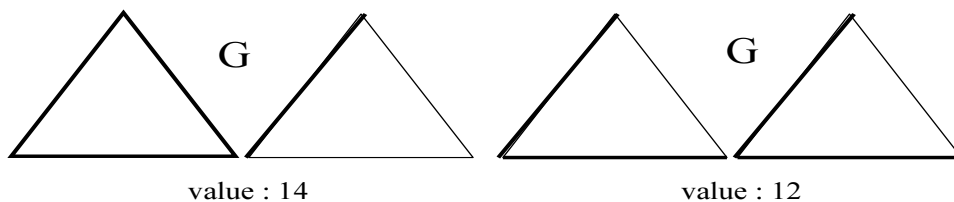


Figure 2: To improve, the degrees must change in four vertices

The paper is organized as follows: in Section 2 we develop the key lemmas (Section 2.1) that are behind the main result and make the algorithm work. Then we prove the main result and state a polynomial algorithm that solves the minsquare problem (Section 2.2). In Sections 2.3, 2.4, 2.5 we characterize the functions for which the procedure is valid, exhibit some additional conditions for which the method works, and state some connections to other problems. In Section 3 we provide an alternative method for solving the minconvex factor problem that leads to further generalizations. Finally in Section 4 we show the NP-completeness of some natural variations of the problem.

2 Main Results

The following result is a variant of theorems about improving alternating walks concerning b -matchings (f -factors). In this paper we avoid speaking about refined details of these walks. We adopt a viewpoint that is better suited for our purposes, and focuses on *degree sequences*. (In Section 2.5 we mention some ideas concerning efficient implementation.)

Let G be a graph, and $F, F' \subseteq E(G)$. Then $P \subseteq E(G)$ will be called an $F - F'$ *alternating walk*, if $P \subseteq F \cup F'$ and $\sum_{v \in V} |d_{P \cap F}(v) - d_{P \cap F'}(v)| \leq 2$; *even* if $|\sum_{v \in V} d_{P \cap F}(v) - d_{P \cap F'}(v)| = 0$, *odd* if $|\sum_{v \in V} d_{P \cap F}(v) - d_{P \cap F'}(v)| = 2$. Clearly, an even walk contains the same number of edges of F and F' , and in an odd walk one of them has one more edge than the other.

An $F - E(G) \setminus F$ -alternating walk that has at least as many edges in F as in $E(G) \setminus F$ will be simply called an F -walk. For an F -walk P (where F is fixed) define $\kappa_P : V(G) \rightarrow \mathbb{Z}$ by $\kappa_P(v) := d_{P \setminus F}(v) - d_{P \cap F}(v)$, $v \in V$; clearly, $|\kappa_P| = 2$ or 0 ; κ_P will be called the *change* (of F along P).

2.1 The key-facts

We state here three simple but crucial facts:

- (2) *If $F \subseteq E(G)$ and P is an F -alternating walk, then $d_F + \kappa_P$ is a degree sequence.*

Indeed, $d_F + \kappa_P$ is the degree sequence of $F \Delta P$, where Δ denotes the symmetric difference, $F \Delta P := (F \setminus P) \cup (P \setminus F)$.

In other words, an alternating walk is a subgraph P of G that has the property that $d_{P \cap F} = d_{P \cap F'}$ in all but at most two vertices of G . The degree of F and F' can differ in two vertices (by 1) or in one vertex (by 2). We call these vertices the *endpoints* of the alternating walk, and if the two endpoints coincide we say that the vertex is an endpoint of the path with *multiplicity* 2 (twice).

Note that we will not use any fact or intuition about how these paths ‘go’, the only thing that matters is the change vector $\kappa_P := (d_{P \cap F'}(v) - d_{P \cap F}(v))_{v \in V}$, and the fact (2) about it: adding this vector to the degree sequences of F , we get a feasible degree sequence again.

If $|d_{P \cap F}(v) - d_{P \cap F'}(v)| = 0$ for all $v \in V$, that is in every node of P there is the same number of incident edges in F and F' , then we say it is an *alternating cycle*.

The following statement is a variant of folklore statements about improving paths concerning graph factors, generalizing Berge’s improving paths for matchings:

(3) *Let $F, F' \subseteq E$. Then $F \Delta F'$ is the disjoint union of alternating walks, so that for all $v \in V$, v is the endpoint of $|d_F(v) - d_{F'}(v)|$ of them (with multiplicity).*

Equivalently, for every $v \in V$, the alternating walks in (3) starting at v either all start with an F -edge or all start with an F' -edge.

Indeed, to prove (3) note that $F \Delta F'$, like any set of edges, can be decomposed into edge-disjoint alternating walks: edges, as one element sets are alternating walks, and they are edge-disjoint. Take a decomposition that consists of a minimum number of walks. Suppose for a contradiction that for some $u \in V(G)$ there exist two walks, P_1 and P_2 such that $P_1 \cap F$ has more edges in u than $P_1 \cap F'$, $P_2 \cap F$ has less edges in u than $P_2 \cap F'$, and let $P := P_1 \cup P_2$. It follows that u is an endpoint of both P_1 and P_2 , moreover with different signs, and we get: $\sum_{v \in V} |d_{P \cap F}(v) - d_{P \cap F'}(v)| \leq$

$$\sum_{v \in V} |d_{P_1 \cap F}(v) - d_{P_1 \cap F'}(v)| + \sum_{v \in V} |d_{P_2 \cap F}(v) - d_{P_2 \cap F'}(v)| - 2 \leq 2 + 2 - 2 = 2.$$

Therefore, P is also an alternating walk, hence it can replace $\{P_1, P_2\}$ in contradiction with the minimum choice of our decomposition, and (3) is proved. \square

We see that in case $|F| = |F'|$, the number of alternating walks in (3) with one more edge in F is the same as the number of those with one more edge in F' . It follows that the symmetric difference of two subgraphs of the same size can be partitioned into edge-sets each of which consists either of an alternating path (also allowing circuits) or of the union of two (odd) alternating paths.

The statement (3) will be useful, since walks will turn out to be algorithmically tractable. For their use we need to decompose improving steps into improving steps on walks.

Let a and λ be given positive integers, and let $\delta(a, \lambda) := (a + \lambda)^2 - a^2 = 2\lambda a + \lambda^2$. Then we have:

If λ_1 and λ_2 have the same sign, then $\delta(a, \lambda_1 + \lambda_2) \geq \delta(a, \lambda_1) + \delta(a, \lambda_2)$.

Indeed, $2(\lambda_1 + \lambda_2)a + (\lambda_1 + \lambda_2)^2 \geq 2\lambda_1 a + 2\lambda_2 a + \lambda_1^2 + \lambda_2^2$, since the left hand side minus the right hand side is $2\lambda_1 \lambda_2 \geq 0$, for λ_1 and λ_2 have the same sign.

For each given factor F and each given $\lambda \in \mathbb{Z}^{V(G)}$ define $\delta(F, \lambda)$ as $\|d_F + \lambda\|^2 - \|d_F\|^2$, that is,

$$\delta(F, \lambda) = \sum_{v \in V(G)} \delta(d_F(v), \lambda(v)).$$

(4) *If $\lambda_1, \dots, \lambda_t$ are vectors such that for every $v \in V(G)$, $\lambda_1(v), \dots, \lambda_t(v)$ have the same sign (this sign may be different for different v) and $\lambda = \lambda_1 + \dots + \lambda_t$, then $\delta(F, \lambda) \geq \delta(F, \lambda_1) + \dots + \delta(F, \lambda_t)$.*

Indeed, apply the inequality stated above to every $v \in V$, and then sum up the n inequalities we got. \square

Now if F is not optimal, then by (3) and (4) one can also improve along pairs of walks. The details are worked out in the next section.

2.2 Solving the Minsquare problem

Recall that for given $F \subseteq E(G)$ an improving vector is a vector $\kappa : V(G) \rightarrow \mathbb{Z}$ such that $b := d_F + \kappa$ is a degree sequence, $\sum_{v \in V(G)} |\kappa(v)| \leq 4$, and $\sum_{v \in V} b(v)^2 < \sum_{v \in V} d_F(v)^2$, while $\sum_{v \in V} d_F(v) = \sum_{v \in V} b(v)$.

Theorem 2.1. *Let G be a graph. If a factor F is not optimal, then there exists an improving vector.*

Proof. Let F_0 be optimal. As F is not optimal one has

$$0 > \|d_{F_0}\|^2 - \|d_F\|^2 = \|d_F + d_{F_0} - d_F\|^2 - \|d_F\|^2 = \delta(F, d_{F_0} - d_F)$$

By (3) $F \Delta F_0$ is the disjoint union of $m \in \mathbb{N}$ F -alternating paths P_1, \dots, P_m . In other words, $F_0 = F \Delta P_1 \Delta \dots \Delta P_m$, and using the simplification $\kappa_i := \kappa_{P_i}$ we have:

$$d_{F_0} = d_F + \sum_{i=1}^m \kappa_i,$$

where we know that the sum of the absolute values of coordinates of each κ_i ($i = 1, \dots, m$) is ± 2 or 0. Since F and F_0 have the same sum of coordinates

$$|\{i \in \{1, \dots, m\} : \sum_{v \in V(G)} \kappa_i(v) = 2\}| = |\{i \in \{1, \dots, m\} : \sum_{v \in V(G)} \kappa_i(v) = -2\}|,$$

and denote this cardinality by p .

Therefore those $i \in \{1, \dots, m\}$ for which the coordinate sum of κ_i is 2 can be perfectly coupled with those whose coordinate sum is -2 ; do this coupling arbitrarily, and let the sum of the two members of the couples be $\kappa'_1, \dots, \kappa'_p$. Clearly, for each κ'_i ($i = 1, \dots, p$) the coordinate-sum is 0, $\sum_{v \in V(G)} |\kappa'_i(v)| \leq 4$, and

$$d_{F_0} = d_F + \sum_{i=1}^p \kappa'_i.$$

Now by (2) each of $d_F + \kappa'_i$ ($i = 1, \dots, p$) is a degree sequence and by (3) $\kappa'_i(v)$ and $\kappa'_j(v)$ have the same sign, $v \in V(G)$, $i', j' \in \{1, \dots, p\}$. To finish the proof we need that at least one of these is an improving vector, which follows from (4):

$$0 > \delta(F, d_{F_0} - d_F) = \delta(F, \sum_{i=1}^p \kappa'_i) \geq \sum_{i=1}^p \delta(F, \kappa'_i)$$

It follows that there exists an index i , $1 \leq i \leq p$ such that $\delta(F, \kappa'_i) < 0$.

□

Corollary 2.1. *The minsquare and the maxfix-cover problem can be solved in polynomial time.*

Indeed, the maxfix cover problem has already been reduced (see beginning of the introduction) to the minsquare problem. Since the value of any solution, including the starting value of the algorithm, is at most n^3 , and an $O(n^3)$ algorithm applied n^4 times decreases it at least by 1, the optimum can be found in at most $O(n^{10})$ time. \square

It can be easily shown that the improving vectors provided by the theorem are in fact alternating walks - similarly to other factorization problems - or edge disjoint unions of such alternating walks. If someone really wants to solve such problems these paths can be found more easily (by growing trees and shrinking blossoms) than running a complete algorithm that finds a b -matching. By adding an extra vertex, instead of trying out all the n^4 possibilities, one weighted matching-equivalent algorithm is sufficient for improving by one. (Adding the nonedges with higher weights one can reformulate the existence of pairs of edge-disjoint improving paths as one matching problem.) However, the goal of this paper is merely to prove polynomial solvability. Some remarks on more refined methods can be found in 2.5.

Various polynomial algorithms are known for testing whether a given function $b : V \rightarrow \mathbb{N}$ is a degree sequence of the given graph $G = (V, E)$. Such algorithms are variants, extensions of Edmonds' algorithm for 1-matchings [7], and have been announced in [8], or can be reduced to matchings altogether with a variety of methods for handling these problems, for surveys see [13], [9], [17]. The complexity of the matching algorithm is bounded by $O(n^{2.5})$, and can be used for making an improving step.

Then n^3 calls of this matching subroutine are sufficient; the resulting somewhat more careful algorithm uses $O(n^{5.5})$ operations for finding a min-square factor.

2.3 Characterizing when it works

We will see here that the proof of Theorem 2.1 works without change for the *minconvex* factor problem, that is, if we replace squares by any set of functions $f_v : \mathbb{N} \rightarrow \mathbb{R}$ ($v \in V$) for which $\delta(F, \lambda) := \sum_{v \in V(G)} f_v(d_F(v) + \lambda) - f_v(d_F(v))$ satisfies (4). This is just a question of checking. However,

a real new difficulty arises for proving Corollary 2.1: the difference between the initial function value and the optimum is no more necessarily bounded by a polynomial of the input, it is therefore no more sufficient to improve the objective value by 1 in polynomial time.

The problem already arises for linear functions: suppose we are given rational numbers p_v ($v \in V(G)$) on the vertices, and $f_v(x) := p_v x$. The length of input is $O(n \log \max\{|p_v| : v \in V(G)\})$, but if we cannot make sure a bigger improvement than by a constant, then we may need $O(n \max\{|p_v| : v \in V(G)\})$ steps.

However, this is a standard problem and has a standard solution, since a slight sharpening of (1) is true: the improving vector κ with the highest $|\delta(F, \kappa)|$ value can also be found in polynomial time. Indeed, one has to take the optimum of a polynomial number of values. Together with the following standard trick the polynomial bound for the length of an algorithm minimizing $\sum_{v \in V(G)} f_v(d_F(v))$ among subgraphs $F \subseteq E(G)$ can be achieved:

(5) *Let F_t , ($t \in \mathbb{N}$) denote the current factor and F^* the optimal one. Starting with an arbitrary factor and choosing repeatedly an improving vector κ with maximum $|\delta(F_t, \kappa)|$ (that is, minimum $\delta(F_t, \kappa) < 0$) value, there are at most $O(n \log \max\{|p_v| : v \in V(G)\})$ improving steps.*

To see this observe first that, by (3) the number l of alternating paths factorizing the symmetric difference of two factors of the same size is at most twice this common size, hence $l = O(m(G))$. Let μ_t denote the change vector where $\delta(F_t, \kappa)$ attains its minimum value over all change vectors κ . We have from the last inequality of the proof of Theorem 2.1:

$$0 > f(d_{F^*}) - f(d_{F_t}) = \delta(F_t, d_{F^*} - d_{F_t}) = \delta(F_t, \sum_{i=1}^s \kappa'_i) \geq l \delta(F_t, \mu_t),$$

that is, $f(d_{F_{t+1}}) - f(d_{F_t}) = \delta(F_t, \mu_t) \leq 1/l(f(d_{F^*}) - f(d_{F_t}))$. It follows that the difference of the current solution from the optimum is multiplied in each step by $1 - 1/l$; after l steps it is multiplied by approximately $1/e$, and after hl steps by $1/e^h$. Since the input is rational, we can suppose it is integer, and with $h = O(n \log \max\{|p_v| : v \in V(G)\})$, after hl steps the difference of the current and optimal solution is less than 1, that is, equal to 0. \square

Note that the case of linear functions that we use for an example can be solved very easily independently of our results. It is a special case of

problems minimizing a linear function on the edges, that is of the following problem: given $w : E(G) \rightarrow \mathbb{Z}$ and $k \in \mathbb{N}$ minimize the sum of the edge-weights among subgraphs of cardinality k . (The node-weighted problem can be reduced to edge-weights defined with $w(ab) := p_a + p_b$ ($a, b \in V$); indeed, then $w(F) = \sum_{v \in V(G)} p_v d_F(v)$.) Add now an extra vertex x_0 to the graph and join it with every $v \in V(G)$ by $d_G(v)$ parallel edges. A minimum weight subgraph with degrees equal to $2(|E(G)| - k)$ in x_0 and $d_G(v)$ for all $v \in V$ intersects $E(G)$ in a minimum weight k -cardinality subgraph. (The same can be achieved under more constraints see Section 2.5.)

Let us make clear now the relation of inequality (4) with some well-known notions.

A function $f : D \rightarrow \mathbb{R}$ ($D \subseteq \mathbb{R}^n$) is said to be convex if for any $x, y \in D$ and $\alpha, \beta \in \mathbb{R}$, $\alpha + \beta = 1$ such that $\alpha x + \beta y \in D$ we have $f(\alpha x + \beta y) \leq \alpha f(x) + \beta f(y)$. Note that we do not require D to be convex, that is for instance D can also be any subset of integers.

A particular case of the defining inequality that will actually turn out to be equivalent is

$$(f(x_1) + f(x_2))/2 \geq f((x_1 + x_2)/2),$$

that is, say for $x_1 = a$, $x_2 = a + 2$:

$$f(a) + f(a + 2) \geq 2f(a + 1),$$

that is,

$$f(a + 2) - f(a + 1) \geq f(a + 1) - f(a).$$

We are not surprised to get this inequality, which characterizes supermodularity, strictly related to discrete convexity, see Murota's work [14]. Notice indeed that $g : 2^{E(G)} \rightarrow \mathbb{Z}$ defined by $g(F) = \sum_{v \in V(G)} f_v(d_F(v))$ is supermodular on $E(G)$, when the f_v 's are convex. Let us state the equivalence of the above inequalities in a form useful for us: Let us state the equivalence of these above inequalities in a form useful for us:

Lemma 2.1. *The following statements are equivalent about the function f whose domain is a (possibly infinite) interval:*

- (i) f is convex

- (ii) For every integer i so that $i, i-1, i+1 \in D(f) : f(i) \leq (f(i-1) + f(i+1))/2$.
- (iii) If $x = x_1 + x_2$, where x_1, x_2 have the same sign, then $f(a+x) - f(a) \geq f(a+x_1) - f(a) + f(a+x_2) - f(a)$.

Indeed, (i) implies (ii) since the latter is a particular case of the defining inequality for convexity. Suppose now that (ii) holds, that is, $f(i+1) - f(i) \geq f(i) - f(i-1)$, and $x = x_1 + x_2$, where x_1 and x_2 have the same sign as x . Then applying this inequality $|x_1|$ times we get $f(a+x) - f(a+x_1) \geq f(a+x_2) - f(a)$. (If $x > 0$, this follows directly; if $x < 0$ then in the same way $f(a) - f(a+x_2) \geq f(a+x_1) - f(a+x)$, which is the same.) This inequality (after rearranging) is the same as (iii). So far we have not even used the assumption about the domain.

We finally prove that (iii) implies (i). Let $x, y, z \in D$, $z = \lambda x + (1-\lambda)y$. Suppose without loss of generality $x = z+r, y = z-s, r, s \in \mathbb{N}$, and prove

$$(s+r)f(z) \leq sf(x) + rf(y).$$

Since by the condition all integers between $z+r$ and $z-s$ are in the domain of f , we have: $f(z+r) - f(z) = f(z+r) - f(z+r-1) + f(z+r-1) - f(z+r-2) + \dots + f(z+1) - f(z) \geq r(f(z+1) - f(z))$, and similarly $f(z) - f(z-l) \leq s(f(z+1) - f(z))$, whence $f(z) - f(z-l) \leq (s/r)(f(z+r) - f(z))$. Rearranging, we get exactly the inequality we had to prove. \square

We need (4) to hold only for improving vectors, and this property does not imply convexity. Conversely, convexity is also not sufficient for (4) to hold: define $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ with $f(x, y) := \max(x, y)$, and let $u = (0, 0)$, $\lambda := (2k+1, 2k+1)$, $\lambda_1 := (k+1, k)$, $\lambda_2 := (k, k+1)$, where $k \in \mathbb{N}$. Then

$$f(u+\lambda) - f(u) = 2k+1 < (k+1) + (k+1) = (f(u+\lambda_1) - f(u)) + (f(u+\lambda_2) - f(u)).$$

2.4 Nonseparable Convex Functions

The convex functions that are the sums of one-dimensional convex functions will be called *separable*. We showed in the previous section that our results can be generalized to separable convex functions but they do not hold for general convex functions. We prove here that the problems themselves are

actually NP-complete for these. Indeed, convexity is not a strong constraint: any function on the hypercube $\{0, 1\}^V$ is convex.

Theorem 2.2. *For any graph $G = (V, E)$, $V = \{1, \dots, n\}$ and a number $k \in \mathbb{N}$ as inputs, it is NP-hard to compute the minimum of $f(d_F(1), \dots, d_F(n))$ ($F \subseteq E, |F| = k$), given an oracle that computes f in polynomial time.*

Proof. We reduce the Hamiltonian cycle problem for undirected graphs to the problem stated in the theorem. Let $G = (V, E)$ be an arbitrary graph.

Split the vertices of the complete graph K_V on V into $n - 1$ vertices, that is, consider the graph H on $n \times (n - 1)$ vertices, and the function f , where

- $V(H)$ is the set of ordered pairs (i, j) , $i \neq j$ where (i, j) represents the (split) endpoint of the edge ij of K_V at vertex i .
- the edge set is a perfect matching: it consists of edges between (i, j) and (j, i) for every pair i, j , $i \neq j$.
- Define $f : V(H) \rightarrow \{0, 1\}$ to be 0 if and only if $f(i, j) = 1$ implies $f(j, i) = 1$ and $\{ij : f(i, j) = 1\}$ represents a Hamiltonian cycle of G , and let it be 1 otherwise.

Clearly, f can be computed in polynomial time, and the minimization problem for f is equivalent to the Hamiltonian cycle problem for G . \square

This result is of course somewhat banal: any combinatorial search problem where the searched object is a subset of edges can be encoded into function f ; if the problem is in NP, the function f will be computable in polynomial time, and will be convex as any function whose domain is the hypercube.

2.5 Minsquare factors under classical constraints

If we want to solve the minsquare problem for constrained subgraphs, that is to determine the minimum of the sum of squares of the degrees of subgraphs satisfying some additional requirements we do not really get significantly more difficult problems. This is at least the case if the requirements are the ‘classical’ upper, lower bound or parity constraints for a subset of vertices.

For such problems (3) can still be applied and the improving path theorems hold. We state the most general consequence concerning the complexity of the minsquare of constrained graph factors, that is, (f, g) -matchings with parity constraints:

Theorem 2.3. *Let $G = (V, E)$ be a graph, $k \in \mathbb{N}$, $l, u : V \rightarrow \mathbb{N}$ and $T \subseteq V$. Then $F \subseteq E$, $|F| = k$ minimizing $\sum_{v \in V} d_F^2(v)$ under the constraint $l(v) \leq d_F(v) \leq u(v)$ for all $v \in V$ and such that $d_F(v)$ has the same parity as $l(v)$ for all $v \in T$, can be found in polynomial time.*

The sum of squares objective function can be replaced here by any objective function mentioned in the previous subsection. The cardinality constraint can actually be replaced by a degree constraint on an added new vertex x_0 . Again, the linear case is much easier. (For instance the minimum weight k -cardinality matching problem can be solved by adding a new vertex, joining it to every $v \in V(G)$ and requiring it to be of degree $n - 2k$ and requiring every $v \in V$ to be of degree 1. In polyhedral terms this is an exercise on Schrijver’s web page [18] and Exercise 6.9 in [5] – about the integrality of the intersection of f -factor polyhedra with the hyperplane $x_1 + \dots + x_n = k$ to which we provided thus one simple solution, and another through our main result, both different from the one suggested in [5].)

The variants we mention can either be proved directly or with some simple and well-known gadgets such as loops for parity constraints.

The results can also be generalized to more abstract structures such as particular jump systems. In [3] this is worked out for “leap systems”, which by now turned out to be a too restrictive structure and those results have been subsumed by Murota’s results in [15] concerning the minimization of M -convex functions. These seem to reach furthest possible in the direction of generalizing our results to abstract structures.

3 Another Way, Another Generalization

In this section we describe another polynomial algorithm for finding a min-convex factor of a graph, where the given convex function is separable. Instead of n b -matching problems in (essentially) the original graph, this solution needs to solve just one b -matching problem in a graph that may have n times as many vertices. The complexity of this solution is therefore higher,

but it has the advantage of providing a framework into which other interesting extensions can be encoded. Moreover it confirms that the “level of difficulty” of our problem and a range of variants is the same as that of b -matchings.

Given an arbitrary graph $G = (V, E)$ we define a graph \hat{G} with the help of the following main gadget (see Figure 3): for every $v \in V$ let $K_v := (A_v, B_v)$ be a complete bipartite graph on new vertices A_v and B_v , where $A_v := \{a_{v,1}, \dots, a_{v,d(v)}\}$, $B_v := \{b_{v,e} : e \in \delta_G(v)\}$, $|A_v| = |B_v| = d(v)$. Now join

- Every $v \in V$ to all the vertices of A_v
- For every $e = uv \in E$, join $b_{u,e} \in B_u$ with $b_{v,e} \in B_v$.

To finish the definition of \hat{G} add a new vertex s , and join it with every vertex $v \in V$ with $d(v)$ parallel edges. This vertex s will serve to control the cardinality of the subgraphs we will be considering.

Given $G = (V, E)$ and $k \in \mathbb{N}$ define a g -factor problem (in the goal of reducing the existence of a minconvex factor of size k to it) with $g : V(\hat{G}) \rightarrow \mathbb{N}$ as follows: $g(s) := 2(|E| - k)$, $g(v) := d(v)$ for all $v \in V$, and 1 on every other vertex, that is on vertices of K_v .

Note first that each g -factor \hat{F} of \hat{G} corresponds to a (unique) subset F , $|F| = k$ of G that will be called the *natural image* of the g -factor in G . The definition is simple:

$$F := \{e = uv \in E(G) : b_{u,e}b_{v,e} \in \hat{F}\}$$

We have to check that $|F| = k$. Indeed, since there are $g(s) = 2(|E| - k)$ edges incident to s , there are $\sum_{v \in V} d_G(v) - g(s) = 2|E| - 2(|E| - k) = 2k$ edges from v to A_v ($v \in V$) altogether. The rest of the vertices of A_v is matched in K_v with the vertices of B_v , and then the rest of the vertices of B_v is matched to other vertices of $B := \cup_{u \in V} B_u$. Recall $|A_v| = |B_v|$ ($v \in V$) and that $g(x) = 1$ for all $x \in V(K_v)$. It follows that there are altogether $2k$ vertices of B matched to other vertices of B by k edges; therefore $|F| = k$ as claimed.

Conversely, $F \subseteq E$, $|F| = k$ is the *natural image* of (several) g -matchings of \hat{G} , and one can get them by reversing the above correspondence.

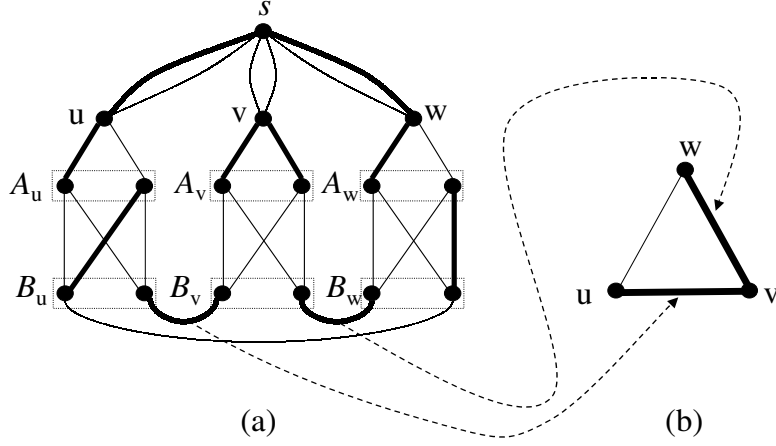


Figure 3: (a): The graph \hat{G} for the triangle G on $\{u, v, w\}$ with one of its g -factors \hat{F} (thick lines); (b): the natural image of \hat{F} in G .

Recall that a separable convex function on the degree sequences of a graph is the sum of functions $f_v : \{0, 1, \dots, d(v)\} \rightarrow \mathbb{R}$ on degree sequences, such that for all $v \in V$:

$$f_v(i+1) - f_v(i) \geq f_v(i) - f_v(i-1)$$

for all $i = 1, \dots, d(v) - 1$ (obviously equivalent to Lemma 2.1 (ii)).

Theorem 3.1. *Suppose we are given a graph $G = (V, E)$ and a separable convex function on its degree sequences given by the convex functions f_v ($v \in V$). Then the natural image of minimum weight g -factors of \hat{G} , where the weights are defined by $w(va_{v,i}) := f_v(i) - f_v(i-1)$, ($i = 1, \dots, d(v) - 1$) minimize $\sum_{v \in V} f_v(d_F(v))$ among subgraphs of G of size k .*

Proof. Let \hat{F} be a minimum weight g -factor of \hat{G} . By definition of natural image it follows that the number of endpoints of edges of \hat{F} in A_v is $d_F(v)$, F being the natural image of \hat{F} . We can suppose without loss of generality that the set of the endpoints of \hat{F} in A_v is $\{a_{v,1}, \dots, a_{v,d_F(v)}\}$.

Indeed, if there exists $i \in \mathbb{N}$ such that $va_{v,i} \notin \hat{F}$ and $va_{v,i+1} \in \hat{F}$, then let $b \in B_v$ be the unique vertex (because of $g(a_{v,i}) = 1$) for which $a_{v,i}b \in \hat{F}$, and

define

$$F' := (\hat{F} \setminus \{va_{v,i+1}, a_{v,i}b\}) \cup \{va_{v,i}, a_{v,i+1}b\}.$$

Because of convexity (see Lemma 2.1 (ii)):

$$w(va_{v,i+1}) := f_v(i+1) - f_v(i) \geq f_v(i) - f_v(i-1) = w(va_{v,i}),$$

and therefore $w(\hat{F}) \geq w(F')$, where F' is also a g -factor in \hat{G} . By the minimality of \hat{F} equality must hold throughout and we can replace \hat{F} by F' . Thus we will suppose now that *the endpoints of \hat{F} itself in A_v are $\{a_{v,1}, \dots, a_{v,d_F(v)}\}$* . We have then

$$w(\hat{F}) = \sum_{v \in V} \sum_{i=1}^{d_F(v)} w(va_{v,i}) = \sum_{v \in V} \sum_{i=1}^{d_F(v)} (f_v(i) - f_v(i-1)) = \sum_{v \in V} f_v(d_F(v)) - \sum_{v \in V} f_v(0),$$

and the last (subtracted) term is a constant (independent on $d_F(v)$).

Conversely, for any $F \subseteq E(G)$, $|F| = k$, any $\hat{F} \subseteq E(\hat{G})$ whose image is F has weight at least $\sum_{v \in V} f_v(d_F(v)) - \sum_{v \in V} f_v(0)$. \square

The model \hat{G} is of course able to include a wider range of optimization problems. In the above proof all weights are 0: we used only the possibility of putting a weight on the edges of \hat{G} incident to v , and under the constraint of convexity (it should be noticed however that any weight function on $E(\hat{G})$ which is non decreasing in i on the edges $va_{v,i}$ and zero elsewhere would define a separable convex function on the degree sequences of the original graph, according to Lemma 2.1). One can optimize on releasing these constraints and putting weights on other edges as well. The minconvex objective function plus a linear objective function can be minimized in this way, a case that has been shown by Murota to be polynomially solvable as it is a particular case of M-convex function minimization, see [15]. However, the following problem does not fit any more into the framework of usual optimization problems: assign weights to the edges of K_v ($v \in V$) and to the edges that join B_u and B_v ($u, v \in V$), and let the edges incident to v ($v \in V$) have 0 weight. We get the following problem in terms of the original graph G :

Find $F \subseteq E$ of minimum cost where the costs are associated with triples (v, uv, i) , $u, v \in V$, $uv \in E$ and $i \in [0, d_G(u)] \cap \mathbb{N}$, and express the “price” of edge uv if it is the i -th in the order of the edges of F at vertex v ; furthermore

one can add to this latter “price” an ordinary linear weight function on the edges. This model actually contains the one of Theorem 3.1 (even though the weights of edges incident to $v \in V$ are 0) : one gets this theorem as a special case by putting on all edges of K_v incident to $a_{v,i}$ the weight $-w(va_{v,i})$ defined in the statement of the theorem. (In terms of the problem we defined on G this means that the values assigned to (v, vu, i) do not depend on v .)

Combining the two kinds of conditions, and releasing the convexity assumption one gets again different results that may be interesting. One can also add some big weight to a group of edges of the auxiliary graph to express the priority of some component of the cost. We leave to the reader the trial of some of the quite big variety of possibilities, and especially the pleasure of finding the appropriate one whenever a need occurs.

4 Weighted minsquare, maxsquare, minfix or maxfix cover

Let us first see what we can say about the weighted minsquare problem. Let a_1, \dots, a_n be an instance of a partition problem. Define a graph $G = (V, E)$ on $n + 2$ vertices $V = \{s, t, 1, \dots, n\}$, and join both s and t to i with an edge of weight a_i . (The degree of both s and t is n and that of all the other vertices is 2.)

Prescribe the vertices of degree 2 (that is, the vertices $i, i = 1, \dots, n$) to have exactly one incident edge in the factor, that is, the upper and lower bounds (see Section 2.5) are 1. Then the contribution of these vertices to the sum of squares of the degrees is constant and the sum of the contributions of s and t is at least $((a_1 + \dots + a_n)/2)^2$, with equality if and only if the PARTITION problem has a solution with these data. (NP-completeness may hold without degree constraints as well.)

We showed in the Introduction (Section 1) that the maxfix cover problem in the line graph of G can be reduced to the minsquare problem in G , which in turn is polynomially solvable. We also exhibited how the relation between transversals and stable sets extends to our more general problems. The following two extensions arise naturally and both turn out to be NP-hard:

In the context of maxfix covers it is natural to put weights on the hyperedges. Associate weights to the hyperedges and the total weight of hyper-

edges that are covered is to be maximized by a fixed number of elements. The edge-weighted maxfix cover problem is the graphic particular case of this, and even this is NP-hard, and even for cliques: the maxfix (vertex) cover problem for a graph $G = (V, E)$ is the same as the weighted maxfix cover problem for the complete graph on V with edge-weights 1 if $e \in E$, and 0 otherwise. Furthermore, a clique is a line graph (for instance of a star) so *edge-weighted maxfix cover is NP-hard for line graphs of stars and even for 0 – 1 weights.*

In the same way as the maxfix cover problem is equivalent to the min-square factor problem, minfix covers are equivalent to *maxsquare factors*. This consists in determining for a given graph and number k a subset of edges that *maximizes* the sum of the squares of the degrees.

The maxsquare problem (and accordingly the minfix cover problem in line graphs) is NP-hard ! Indeed, let's reduce the problem of deciding whether a clique of size r exists in the graph $G = (V, E)$ to a maxsquare problem in $\tilde{G} = (\tilde{V}, \tilde{E})$ with $k := r\Delta_G$ (equivalently, to a min cover problem in $L(\tilde{G})$) where \tilde{G} is defined as follows: subdivide every edge of G into two edges with a new vertex, and for all $v \in V$ add $\Delta_G - d_G(v)$ edges to new vertices of degree 1 each. We suppose that G does not have loops or parallel edges.

Clearly, \tilde{G} is a bipartite graph, where the two classes are $A := V$ and $B := \tilde{V} \setminus V$. In A all the degrees are equal to $\Delta = \Delta_G$, and in B they are all at most 2. Then the edges of the factor we have to select forms a bipartite graph as well, and therefore the size $k = r\Delta$ of the set F we have to choose is equal to $\sum_{a \in A} d_F(a) = \sum_{b \in B} d_F(b) = |F| = k$. Among sets of numbers (bounded by Δ) whose sum is fixed (to $k = r\Delta$) the largest sum of squares is reached if each of the numbers is maximum – that is, we cannot do better than selecting a set F of edges of \tilde{G} where $d_F(a) = \Delta$ for any $a \in A$ incident to some edge of F (and of course the number of such vertices is r), and $d_F(b) = 2$ for any $b \in B$ incident to some edge of F . This is of course not always possible, but clearly, when it is, it provides the unique maximum, and the subgraphs F with these properties are in one-to-one correspondence with the cliques of size r of G .

The problem of deciding whether G has a clique of size r is thus polynomially reduced to the a maxsquare problem on \tilde{G} with $k = r\Delta$, showing that the maxsquare problem is also NP-hard.

One may think of the “fixed size” constraint in the minsquare factor

problem as a constraint given on the rank of the factor F in the uniform matroid $U_{n,t}$ on $E(G)$. So a direct generalization of the minsquare problem would be the following: given a matroid M on $E(G)$ find a basis F of M minimizing $\|d_F\|^2$ in G . Also this generalization turns out to be NP-complete (in the ordinary sense): *a hamiltonian path of a connected graph G is a basis of the graphic matroid $M(G)$ of G and minimizes $\|d_F\|^2$ among all $F \subseteq E(G)$ that are bases of $M(G)$.*

Among all these problems the most interesting is maybe the one we could solve: indeed, it generalizes the maximum matching problem and the methods extend those of matching theory.

Acknowledgment: We are thankful to Maurice Queyranne and Akihisa Tamura for their questions about the relation of (4) to convexity that generated the remarks in 2.3.

References

- [1] N. Apollonio, L. Caccetta, B. Simeone, Cardinality Constrained Path Covering Problems in Trees, Technical Report n. 18, Department of Statistics, Probability, and Applied Statistics, University of Rome “La Sapienza”, 2002.
- [2] N. Apollonio, L. Caccetta, B. Simeone, Cardinality Constrained Path Covering Problems in Grid Graphs, *Networks* 44(2),2004.
- [3] Apollonio, Sebő, Minsquare Factors and Maxfix Covers of Graphs, in: D. Bienstock and G. Nemhauser eds, *Integer Programming and Combinatorial Optimization*, Lecture Notes in Computer Science, 3064, Springer-Verlag, 2004, 388-400.
- [4] A. Bouchet, W. Cunningham, Delta-matroids, Jump-systems and Bisubmodular polyhedra, *SIAM J. Discrete Mathematics*, February 1995.
- [5] W. Cook, W. Cunningham, W. Pulleyblank, A. Schrijver, *Combinatorial Optimization*, John Wiley & Sons, Inc, 1998.

- [6] W.H. Cunningham and J. Green-Krótki, b -matching degree-sequence polyhedra, *Combinatorica* **11** (1991) 219-230.
- [7] J.R. Edmonds, *Maximum matching and a polyhedron with 0,1-vertices*, J. Res. Nat. Bur. Standards Sect. B (1968), 125–130.
- [8] J.R. Edmonds, E.L. Johnson, *Matching: a well-solved class of integer linear programs*, in *Combinatorial Structures and Their Applications*, Calgary, Alberta, (1969), Guy, Hanani, Sauer, Schönheim eds.
- [9] A. M. H. Gerards, ‘Matching’, in the *Handbook of Operations Research and Management Science*, Volume 7, ‘Network Models’ (1995), 135–224.
- [10] M. Grötschel, L. Lovász, A. Schrijver, “Geometric Algorithms and Combinatorial Optimization”, Springer Verlag, Berlin Heidelberg (1988).
- [11] L. Lovász, On the structure of factorizable graphs, *Acta Math. Acad. Sci. Hungar.* **23**, 1972, 179–195.
- [12] L. Lovász, The factorization of graphs II, *Acta Math. Acad. Sci. Hungar.* **23**, 1972, 223–246.
- [13] L. Lovász, M.D. Plummer, *Matching Theory*, Akadémiai Kiadó, Budapest, 1986.
- [14] K. Murota, Discrete convex analysis, *Mathematical Programming*, **83** (1998), 313-371
- [15] K. Murota, M -convex Functions on Jump Systems: Generalization of Minsquare Factor Problem, Proceedings of the Japanese-Hungarian workshop, Budapest, 2005.
- [16] E. Petrank, The Hardness of Approximation: Gap Location, *Computational Complexity* **4** (1994) 133-157.
- [17] W. R. Pulleyblank, ‘Matchings and stable sets’, in Graham, Grötschel, Lovász (eds.), *Handbook of Combinatorics* (Elsevier Science, 1995).
- [18] A. Schrijver, *A Course in Combinatorial Optimization* (lecture notes), <http://homepages.cwi.nl/~lex/>, October, 2000.