Minsquare Factors and Maxfix Covers of Graphs

Nicola Apollonio[1]* and András Sebő[2]

[1] Department of Statistics, Probability, and Applied Statistics University of Rome
"La Sapienza", Rome, Italy
[2] CNRS, Leibniz-IMAG, Grenoble, France

Abstract. We provide a polynomial algorithm that determines for any given undirected graph, positive integer k and various objective functions on the edges or on the degree sequences, as input, k edges that minimize the given objective function. The tractable objective functions include linear, sum of squares, etc. The source of our motivation and at the same time our main application is a subset of k vertices in a line graph, that cover as many edges as possible (maxfix cover). Besides the general algorithm and connections to other problems, the extension of the usual improving paths for graph factors could be interesting in itself: the objects that take the role of the improving walks for b-matchings or other general factorization problems turn out to be edge-disjoint unions of pairs of alternating walks. The algorithm we suggest also works 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.

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 pregiven 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 3.1), including also linear functions (Section 3.2). 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 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 [14] under the name of *max vertex cover*, obviously

* On leave from Università di Roma "La Sapienza", postdoc fellowship supported by the European project DONET while the author visited Laboratoire Leibniz (IMAG)

D. Bienstock and G. Nemhauser (Eds.): IPCO 2004, LNCS 3064, pp. 388–400, 2004.

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 ?

A maxfix cover for $L(G)$ is $T \subseteq E(G)$ minimizing the number of incident pairs of edges in the remaining edge-set $F = E(G) \setminus T$, $|F| = k := n - t$. Clearly, the number of such paths 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 the *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 $b : V(G) \longrightarrow \mathbb{N}$, a *b-matching* 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)-factor, where $f, g : V(G) \longrightarrow \mathbb{N}$, 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 'is $\alpha \geq k$?'.) Similarly, minfix covers are the complements of maxfix induced subgraphs.

As we will see in 3.4, among 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 the most number (highest weight) of hyperedges. For *Edge-Path* hypergraphs, that is a 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 Apollonio et al. 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 [9] that convex function can be minimized in polynomial time on convex polytopes under natural and general conditions. Hence convex functions can be optimized on 'b-matching polytopes' and intersections of such polytopes with hyperplanes (or any other solvable polyhedron in the sense of [9]). However, the optima are not necessarily integer, neither when minimizing on the polytope itself, nor for the intersection.

Minimizing a convex function on 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 [8], [15] do the job. However, for our problem, where the set of b-matchings is intersected with a hyperplane, single paths do no more 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 on the intersection of (f, g)-factor 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. 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* $t : V(G) \longrightarrow \mathbb{N}$ *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;
- $\sum_{v \in V} t^2(v) < \sum_{v \in V} d_F^2(v)$, 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 [8], [15], [12]), 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 checked:

(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. Optimizing linear functions over the degree sequences of subgraphs of fixed cardinality k presents already the difficulties of the general case (see Section 3.2); on the other hand the methods apply for a quite general set of objective functions (see Section 3). We have chosen to put in the center minsquare factors because of the origins of the problem and because they are a representative example of the new problems that can be solved.

The rest of the paper is organized as follows: in Section 2 we develop the key lemmas that are behind the main result and make the algorithm work. Then in Section 2.2 we prove the main result and state a polynomial algorithm that solves the minsquare problem. In Section 3 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.

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 3.2 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 $\kappa_P : V(G) \longrightarrow \mathbb{Z}$ 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. □

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 $\delta(a, \lambda_1 + \lambda_2) \geq \delta(a, \lambda_1) + \delta(a, \lambda_2)$.

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

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

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 $\kappa : V(G) \longrightarrow \mathbb{Z}$ 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 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) $F \varDelta F_0$ is the disjoint union of $m \in \mathbb{N}$ F-alternating paths $P_1, \ldots, P_m$. In other words, $F_0 = F \varDelta P_1 \varDelta \ldots \varDelta P_m$, 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. 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. *The minsquare and the maxfix-cover problem can be solved in polynomial time.*

Proof. 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. □

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 matching-equivalent algorithm is sufficient for improving by one. However, the goal of this paper is merely to prove polynomial solvability. Some remarks on more refined methods can be found in 3.2.

Various polynomial algorithms are known for testing whether a given function $b : V \longrightarrow \mathbb{N}$ is a degree sequence of the given graph $G = (V, E)$. Such algorithms are variants or extensions of Edmonds' algorithm for 1-matchings [6], and have been announced in [7]. The same problem can also be reduced to matchings. A variety of methods for handling these problems can be found in [12], [8], [15].

The complexity of a variant of the matching algorithm is bounded by $O(n^{2.5})$, and can be used for making an improving step; n^3 calls for this algorithm are sufficient, so if we are a bit more careful, $O(n^{5.5})$ operations are sufficient to find a minsquare factor.

3 Special Cases and Extensions

3.1 Characterizing when It Works

We will see here that the proof of Theorem 1 works if we replace squares by any set of functions $f_v : \mathbb{N} \longrightarrow \mathbb{R}$ $(v \in V)$ for which $\delta(F, \lambda) := \sum_{v \in V} f_v(d_F(v) + \lambda) - f_v(d_F(v))$ satisfies (4). This is just a question of checking. However, a real new difficulty arises for proving Corollary 2: 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 $(v \in V)$ on the vertices, and $f_v(x) := p_v x$. The input is $O(\log \max\{|p_v| : v \in V\})$, but if we cannot make sure a bigger improvement than by a constant, then we may need $O(\max\{|p_v| : v \in V\})$ 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. Along with the following standard trick the polynomial bound for the length of an algorithm minimizing $\sum_{v \in V} f_v(d_F(v))$ among subgraphs $F \subseteq E(G)$ can be achieved:

(5) *Starting with an arbitrary F_0 and choosing repeatedly an improving vector κ with maximum $|\delta(F, \kappa)|$, that is, minimum $\delta(F, \kappa) < 0$ value, there are at most $O(n^2 \log \max\{|p_v| : v \in V\})$ improving steps.*

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 $w : E(G) \longrightarrow \mathbb{Z}$ 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 $w(F) = \sum_{v \in V} p_v d_F(v)$.) 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 3.2.)

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

A function $f : D \longrightarrow \mathbb{R}$ $(D \subseteq \mathbb{R}^n)$ 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 [13]. Let us state the equivalence of these inequalities in a form useful for us:

Lemma 3. *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 $f(a + x) - f(a) \geq f(a + x_1) - f(a) + f(a + x_2) - f(a)$.*

Proof. 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). Until now 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: $f(z+r)-f(z) = f(z+r)-f(z+r-1)+f(z+r-1)-f(z+r-2)+\ldots+f(z+1) - f(z) \geq r(f(z+1) - f(z))$, 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. □

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 $f : \mathbb{R}^2 \to \mathbb{R}$ 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

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

3.2 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, (u, l)-factors with parity constraints:

Theorem 4. *Let $G = (V, E)$ be a graph, $k \in \mathbb{N}$, $l, u : V \longrightarrow \mathbb{N}$ and $T \subseteq V$. Then $F \subseteq E$, $|F| = k$ minimizing $\sum_{v \in V} d_F^2(v)$ under the constraint $l(v) \geq d_F(v) \geq u(v)$ for all $v \in V$ and such that $d_F(v)$ has the same parity as $u(v)$ 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 [16] and in Exercise 6.9 of [4] – about the integrality of the intersection of f-factor polyhedra with the hyperplane $x_1 + \ldots + x_n = k$ to which we provided thus one simple solution, and another through our main result, both different from the one suggested in [4].)

3.3 Jump Systems

A *jump system* [3] is a set $J \subseteq \mathbb{Z}^n$ with the property that for all $x, y \in J$ and each vector $x' \notin J$ at one step from x towards y there exists a vector $x'' \in J$ at one step from x' towards y. We say that x' is at one step from x towards y if $x' - x$ is ± 1 times a unit vector and the non-zero coordinate of $x' - x$ has the same sign as the corresponding coordinate of $y - x$. More generally, x' is at s steps from x towards y if $|x' - x| = s$ and $|x' - x| + |y - x'| = |y - x|$, that is, the l_1-distance of x' from x is s, and it is on the 'l_1-line' that joins x and y.

Jump systems generalize, among others, the family of bases or independent sets of matroids, delta-matroids, and degree sequences of the subgraphs of a graph (some of these statements are easy, some others are more difficult to prove, see [3], [5]). Their convex hulls constitute the most general class of polyhedra for which the greedy algorithm works.

In the proof of Theorem 1 the key-fact is that starting from a given graph factor one can arrive at any other one by small changes in the right direction (3). This property makes sense for sets of vectors in general, but it requires a more restrictive property than the definition of jump systems:

Let us say that $L \subseteq \mathbb{Z}^V$ is a leap-system if for all $x, y \in L$, $y-x = l_1+\ldots+l_m$ $(m, \in \mathbb{N}, l_i \in \mathbb{Z}^V,\ i = 1, \ldots, m)$ where, $|y-x| = |l_1| + \ldots |l_m|$, $|l_i| \leq 2$, $(i = 1, \ldots, m)$ and $x + \sum_{i \in Q} l_i \in L$, for $Q \subseteq \{1, \ldots, m\}$, $|Q| \leq 2$.

It is straightforward to check that Theorem 1 and the algorithms are valid without any change if degree sequences are replaced by leap-systems .

The definition of an improving vector is now the following:

Let L be a leap-system and $l \in L$. Then $\kappa \in \mathbb{Z}^V$ is an *improving vector* with respect to l if $b = l + \kappa \in L$, and $\sum_{v \in V} |\kappa(v)| \leq 4$, $f(b) < f(l)$, while $\sum_{v \in V} b(v) = \sum_{v \in V} l(v)$.

Theorem 5. *Let L be a leap system. If $l \in L$ is not optimal, then there exists an improving vector.*

As a consequence the sum of one dimensional convex functions can be optimized on leap systems intersected with hyperplanes, in polynomial time.

3.4 Weighted Minsquare, Maxsquare, Minfix, or Maxfix Cover

Let us first see what we can say about the weighted minsquare problem. Let $a_1, \ldots, a_n$ be an instance of a partition problem. Define a graph $G = (V, E)$ on $n+2$ vertices $V = \{s, t, 1, \ldots, n\}$, 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, $i = 1, \ldots, n$) to have exactly one incident edge in the factor, that is, the upper and lower bounds (see Section 3.2) are 1. Then the contribution of these vertices to the sum of squares of the degrees is fix and the sum of the contributions of s and t is at least $((a_1 + \ldots + a_n)/2)^2$, 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 hyperedges that are covered is to be maximized with 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 *path-weighted maxfix cover is NP-hard for stars and even for* $0-1$ *weights.*

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 $\hat{G} = (\hat{V}, \hat{E})$ (equivalently, to a min cover problem in $L(\hat{G})$) where $\hat{G}$ 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, $\hat{G}$ is a bipartite graph, where the two classes are $A := V$ and $B := \hat{V} \setminus V$. In A all the degrees are equal to Δ, and in B they are all at most 2.

If $K \subseteq V$ is a clique of size r in G, then let $F_K \subseteq E(\hat{G})$ be defined as $F_K := \delta_{\hat{G}}(K)$ where $K \subseteq V = A \subseteq \hat{V}$. Clearly, $|F_K| = r\Delta$. The interested reader may check the following assertions: F_K *is a maxsquare subgraph of cardinality* $r\Delta$, and conversely, *a subgraph* $F \subseteq V$ *of cardinality* $r\Delta$ where the sum of the squares of the degrees is at least that of F_K above, satisfies $F = F_K$ for some $K \subseteq V$, $|K| = r$ that induces a complete subgraph. Therefore the problem of deciding whether G has a clique of size r can be polynomially reduced to a maxsquare problem, showing that the latter is also NP-complete.

Among all these problems the most interesting is maybe the one we could solve: indeed, it generalizes the maximum matching problem and the methods are also based on 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 Section 3.1.

References

1. N. Apollonio, L. Caccetta, B. Simeone, Cardinality Constrained Path Covering Problems in Trees, Tech.Rep. 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*, to appear.
3. A. Bouchet, W. Cunningham, Delta-matroids, Jump-systems and Bisubmodular polyhedra, SIAM J. DIscrete Mathematics, February 1995.
4. W. Cook, W. Cunningham, W. Pulleyblank, A. Schrijver, Combinatorial Optimization, John Wiley & Sons, Inc, 1998.
5. W.H. Cunningham and J. Green-Krótki, b-matching degree-sequence polyhedra, *Combinatorica* **11** (1991) 219-230.
6. J.R. Edmonds, *Maximum matching and a polyhedron with 0,1-vertices*, J. Res. Nat. Bur. Standards Sect. B (1968), 125–130.
7. 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.
8. A. M. H. Gerards, ‘Matching’, in the Handbook of Operations Research and Management Science, Volume 7, ‘Network Models’ (1995), 135–224.
9. M. Grötschel, L. Lovász, A. Schrijver, “Geometric Algorithms and Combinatorial Optimization”, Springer Verlag, Berlin Heidelberg (1988).

10. L. Lovász, On the structure of factorizable graphs, *Acta Math. Acad. Sci. Hungar.* 23, 1972, 179–195.
11. L. Lovász, The factorization of graphs II, *Acta Math. Acad. Sci. Hungar.* 23, 1972, 223–246.
12. L. Lovász, M.D. Plummer, Matching Theory, Akadémiai Kiadó, Budapest, 1986.
13. K. Murota, Discrete convex analysis, *Mathematical Programming*, **83** (1998), 313-371
14. E. Petrank, The Hardness of Approximation: Gap Location, *Computational Complexity* **4** (1994) 133-157.
15. W. R. Pulleyblank, 'Matchings and stable sets', in Graham, Grötschel, Lovász (eds.), Handbook of Combinatorics (Elsevier Science, 1995).
16. A. Schrijver, A Course in Combinatorial Optimization (lecture notes), http://homepages.cwi.nl/ lex/, October, 2000.