Discrete Optimization 15 (2015) 26–36

Contents lists available at ScienceDirect

Discrete Optimization

journal homepage: www.elsevier.com/locate/disopt

Clique partitioning with value-monotone submodular cost

José R. Correa [a], Nicole Megow [b,*]

[a] Departamento Ingeniería Industrial, Universidad de Chile, Santiago, Chile

[b] Department of Mathematics, Technische Universität Berlin, Germany

ARTICLE INFO

Article history:
Received 22 July 2011
Received in revised form 10 September 2014
Accepted 3 November 2014
Available online 8 December 2014

Keywords:
Partition into cliques
Cost coloring
Submodular functions
Cardinality constraint
Max-coloring
Batch-scheduling with compatibilities

ABSTRACT

We consider the problem of partitioning a graph into cliques of bounded cardinality. The goal is to find a partition that minimizes the sum of clique costs where the cost of a clique is given by a set function on the nodes. We present a general algorithmic solution based on solving the problem variant without the cardinality constraint. We obtain constant factor approximations depending on the solvability of this relaxation for a large class of submodular cost functions which we call value-monotone submodular functions. For special graph classes we give optimal algorithms.

1. Introduction

We consider the problem of partitioning a graph into cliques of bounded cardinality. Given is a simple graph $G = (V, E)$, a set function $f\colon 2^V \to \mathbb{R}^+$, and a bound $B \in \mathbb{Z}^+$. The task is to find a partition of G into cliques $K_1, \ldots, K_\ell$ of cardinality at most B. The cost of a clique K_i is given by the set function $f(K_i)$ via an oracle. Our objective is to find a partition of minimum total cost, $\sum_{i=1}^{\ell} f(K_i)$. We denote our problem as *partition into cliques of bounded cardinality* PCliq(G, f, B).

The problem PCliq(G, f, B) is a generalization of the classical *graph partitioning problem* and thus $\mathcal{NP}$-hard [1]. For arbitrary graphs the problem is even $\mathcal{NP}$-hard to approximate within a factor $|V|^{1-\varepsilon}$ for all $\varepsilon > 0$ [2] since it contains *vertex coloring* as a special case. Indeed, our problem can be formulated equivalently as finding a partition into independent sets (coloring) of bounded cardinality and minimum total cost in $\overline{G}$, the complement of G, where the cost of each independent set (color class) is defined by some cost function f. In particular, PCliq(G, f, B) with $B = \infty$ and $f(S) \equiv 1$ is equivalent to the classical vertex coloring problem in $\overline{G}$. However, in the special case that the bound on the clique size B equals 2 then PCliq(G, f, B) can be transformed into a matching problem in G and can be solved optimally in polynomial time. Since cliques in a bipartite graph have size at most 2, the graph partitioning problem corresponds in this case to a matching problem.

Graph partitioning and coloring problems are among the fundamental problems in combinatorial optimization. Imposing bounds on the size of the cliques or color classes is very natural in many applications. Such problems are often formulated also in a scheduling context where compatibilities or conflicts between jobs are expressed in a graph, and the task is to find a partition into cliques or independent sets of jobs that minimize the total cost with respect to some scheduling cost function. The problems of partitioning or coloring with a cardinality bound have been mainly studied for two cost functions: $f(S) \equiv 1$ and the max-function. The coloring problem with the objective to minimize the number of colors, i.e., $f(S) \equiv 1$, is

* Corresponding author.
E-mail addresses: correa@uchile.cl (J.R. Correa), nmegow@math.tu-berlin.de (N. Megow).

http://dx.doi.org/10.1016/j.disopt.2014.11.001
1572-5286/

known as *mutual exclusion scheduling* [3–6] or *bounded vertex coloring* [7], and when the cost function is the max-function $f(S) = \max_{v \in S} w(v)$ for some weight function $w\colon V \to \mathbb{R}^+$ then the coloring problem is also called *bounded max-coloring problem* [8]. The complementary problem of partitioning a graph into cliques with a max-cost function has been studied also as *max-batch scheduling with compatible jobs* [9–12].

This body of previous work implies for our problem PCliq(G, f, B) that it is already for the simple cost function $f(S) \equiv 1$ $\mathcal{NP}$-hard on cographs, co-bipartite and co-interval graphs [3], on comparability and co-comparability graphs even when B is fixed ($B \geq 6$) [6], and for the complements of line graphs also for any fixed $B \geq 3$ [13]. However, there are polynomial-time optimal algorithms for split graphs [3,14], interval graphs [3,9], as well as for the complements of forests [5] and of line graphs [15].

The problem PCliq(G, f, B) with f being the max-function is strongly $\mathcal{NP}$-hard even for $B = \infty$ for co-bipartite graphs [16], co-interval graphs [17], split graphs [10,16], and interval graphs [12]. Interval graphs found particular interest in the literature because of their wide applicability in practical applications, e.g., describing compatible processing intervals in scheduling. For this graph class there is a polynomial-time approximation scheme (PTAS) for any fixed capacity bound $B \geq 3$ [12]. A PTAS is also known for graphs that are complements of trees [8]; here the complexity status remains unsolved. And for co-bipartite graphs there is a $17/11$-approximation [8].

The relaxed variant of our problem PCliq(G, f, B) without bounds on the clique size, i.e., $B = \infty$, has been widely studied in different settings. Various complexity and approximability results are known for particular cost functions and graph classes. Interestingly, many of these particular cost functions that appear in specific applications share the property of being submodular. Examples for such special problems and submodular cost functions are the above-mentioned *maximum function*, the *probabilistic cost function* [18,19], *partial q-coloring* [20,21], (modified) *chromatic entropy* [22,23], and more generally *non-decreasing weight-defined concave cost functions* [24]. Definitions and details will be given in Section 2.

While most research focused on particular cost functions, recently more general set functions have been considered in this context. Gijswijt, Jost, and Queyranne [25] introduce so-called *value-polymatroidal* set functions. They consider the graph partitioning problem without capacity restrictions, PCliq(G, f, ∞), and derive a polynomial time algorithm for interval graphs and circular arc graphs. Furthermore, Fukunaga, Halldórsson, and Nagamochi [24] assume a weight function $w\colon V \to \mathbb{R}^+$ and consider weight-defined *monotone concave* cost functions. They provide a general algorithmic scheme that yields a factor 4 approximation for clique partitioning without cardinality constraint, PCliq(G, f, ∞), in co-interval graphs, comparability and co-comparability graphs, and a 6-approximation for perfect graphs. In fact, they obtain a robust result in the sense that the solution they compute approximates all cost functions under consideration simultaneously.

Our results. We investigate the problem of partitioning a graph into cliques of bounded cardinality for a general class of cost functions. We consider *value-monotone submodular functions*. This subclass of submodular functions is only slightly more restricted than the class of value-polymatroidal functions introduced by Gijswijt et al. [25] in a similar context, and contains well-known cost functions for coloring or partitioning problems such as the max-function, probabilistic function, partial q-coloring, (modified) chromatic entropy, and more generally non-decreasing weight-defined concave cost functions.

We provide a very simple but general approximation algorithm for solving the graph partitioning problem for general graphs based on a solution to the relaxed problem without cardinality constraints on the cliques. We show that adding the bound on the cardinality does not diminish the solution quality too much. More precisely, if there is an α-approximate solution for the unbounded problem, then a Greedy algorithm turns this solution into an $(\alpha + 1)$-approximate solution that obeys the cardinality constraint. This general result implies several new or improved results for special graph classes. In particular, we obtain a 2-approximation for interval graphs and circular arc graphs for arbitrary value-monotone submodular cost functions. For special cost functions we obtain improved results such as, e.g., an $(e + 1)$-approximation for the general class of perfect graphs when f is the max-function. More generally, we obtain a 5-approximation for co-interval, comparability, and co-comparability graphs, and a 7-approximation for perfect graphs when f is a non-decreasing weight-defined concave function.

As a subproblem, we investigate the problem variant of partitioning a complete graph into cliques of bounded cardinality. We show that this seemingly simple problem is $\mathcal{NP}$-hard for polymatroid rank functions, a subclass of submodular functions. However, we also show that restricting to value-monotone cost functions allows a Greedy algorithm to solve the problem optimally in polynomial time.

Finally, we consider the partitioning problem on proper interval graphs. This graph class plays an important role in many applications, e.g., as compatibility graphs in the above mentioned max-batch scheduling problem [9]. We derive a dynamic programming algorithm that solves the problem optimally in running time that is exponential in the number w of different node types with respect to the cost function f. Thus, this algorithm computes an optimal solution in polynomial time if w is bounded by a constant. Moreover, using a recently shown technique to reduce the number w at the cost of a small increase in the objective function when f is the max-function [12], the algorithm leads in this case to a PTAS. Thus, we give a PTAS for max-batch scheduling with compatibilities that form a proper interval graph. This contrasts the previously known PTAS [12] which has polynomial running time only when the capacity bound B is constant. Note, however, that this result was actually derived for general interval graphs.

Outline. In Section 2 we define value-monotone submodular set functions, give examples, and discuss important properties. In Section 3 we investigate the clique partitioning problem on complete graphs. We prove optimality of a Greedy algorithm and show that the complexity status changes when allowing submodular cost functions that do not satisfy the

value-monotonicity properties. We investigate the problem on general graphs in Section 4 and give an approximation guarantee as a function of the approximation factor for the relaxed problem without cardinality restrictions on the cliques. Finally in Section 5, we consider proper interval graphs and provide a dynamic program that computes an optimal solution in polynomial time if the number of different node types is bounded. This leads to a PTAS for arbitrary node types when the cost function is the max-function.

2. Value-monotone submodular functions

2.1. Definition and examples

Let V be a finite set. The function $f\colon 2^V \to \mathbb{R}$ is called *submodular* if for all subsets $A, B \subseteq V$ holds $f(A)+f(B) \geq f(A\cup B)+f(A\cap B)$. We consider non-negative submodular functions with $f(\emptyset) = 0$, that satisfy the following two additional properties.

- *Value-monotone exchange property*. For all subsets $A, B \subseteq V$ such that $f(A) \geq f(B)$ and $u, v \in V \setminus (A\cup B)$ with $f(u) \geq f(v)$, or $v = \emptyset$, it holds
$$f(A+u)+f(B+v) \leq f(A+v)+f(B+u). \tag{E1}$$
- *Value-monotone enlargement*. For all sets $A, B \subseteq V$ such that $f(A) \geq f(B)$ and all elements $u \in V \setminus (A \cup B)$ holds that
$$f(A+u) \geq f(B+u). \tag{E2}$$

We denote non-negative submodular functions with $f(\emptyset) = 0$ that satisfy the value-monotone exchange property (E1) and the value-monotone enlargement (E2) as *value-monotone submodular functions*.

This class of set functions contains well-known cost functions that have been studied individually in the graph partitioning or coloring context. In the following we give some examples. The membership proofs can be found in the Appendix.

Maximum function: Let $w\colon V \to \mathbb{R}_+$ and $S \subseteq V$. The max-function is defined as $f(S) := \max_{v\in S} w(v)$. The *max-coloring problem* is a well-studied problem [8,16,17,26–29]. Here the task is to find a coloring of the graph (without restrictions on the cardinality) of minimum total cost, where the cost of each color class is given by the max-function. The partitioning variant of this problem has been investigated mainly as max-batch scheduling problem with compatibilities [9–12]. These results imply (in addition to the results mentioned in the introduction) that the unbounded graph partitioning problem PCliq(G, f, ∞) with f being the max-function is polynomial-time solvable for interval graphs [9,25], circular arc graphs [25], and for the complements of P_4-free graphs [16], whereas there is an e-approximation for perfect graphs [27].

Probabilistic cost function: Let $q\colon V \to [0, 1]$ and $S \subseteq V$. Then $f(S) := 1 - \prod_{v\in S} q(u)$ is called *probabilistic cost function*. It has been studied particularly in the context of stochastic models for combinatorial optimization problems where uncertain input is given with independent probabilities; see [18,19] for motivation and further references. When restricted to probabilistic cost functions, PCliq(G, f, ∞) is strongly $\mathcal{NP}$-hard for split graphs [19] and co-bipartite graphs [18], and it is polynomial-time solvable in bipartite graphs [18] and interval graphs [25]. These results were actually obtained for *probabilistic coloring* in the complement graphs, i.e., for finding a coloring of minimum total cost, where the cost of each color class is determined by the probabilistic cost function.

Partial q-coloring: Let $k \in \mathbb{N}$, $S \subseteq V$, and $f(S) := \min\{k, |S|\}$. Finding a coloring of minimum total cost, where the cost for each color class is given by f is called *partial q-coloring* [20,21]. The cost function f is a special case of the following class.

Non-decreasing weight-defined concave cost functions: Let $w\colon V \to \mathbb{R}_+$, $S \subseteq V$, and let $f(S) := h(\sum_{v\in S} w(v))$ for some concave, non-decreasing function $h\colon \mathbb{R} \to \mathbb{R}$. Such functions are also known as *submodular set utility function* and appear, e.g., when expressing decreasing marginal preferences in combinatorial auctions [30], as joint replenishment functions in supply chain management [31], or setup cost in facility location [32]. In a coloring context, such functions were considered by Fukunaga et al. [24] who derived the aforementioned robust constant-factor approximations for PCliq(G, f, ∞) in co-interval graphs, (co-)comparability, and perfect graphs.

(Modified) Chromatic entropy: Let $p : V \to [0, 1]$, $\sum_{v\in V} p(v) = 1$, and $S \subseteq V$. The *chromatic entropy cost function* is defined as $f(S) := -c(S)\log c(S)$ where $c(S) = \sum_{v\in S} p(v)$. This function is concave but not non-decreasing and thus not value-monotone. However, Gijswijt et al. [25] observed the following: the function $g(S) := f(S) + c(S)$ is non-decreasing concave and any partition into cliques $V = K_1 \cup \cdots \cup K_\ell$ has cost $\sum_{i=1}^{\ell} g(K_i) = \sum_{i=1}^{\ell} f(K_i) + c(V)$, which implies that both cost functions f and g yield the same optimal solution.

The problem of partitioning a graph into independent sets minimizing the total chromatic entropy is called *chromatic entropy coloring*. It finds applications in the field of data compression and zero-error coding theory [22,33]. Complexity and (additive) approximation results were derived in [23,34]. When f is a chromatic entropy cost function, PCliq(G, f, ∞) is strongly $\mathcal{NP}$-hard on co-interval graphs [34] and (even unweighted) co-chordal graphs [23].

2.2. Basic properties

Proposition 1. *A value-monotone submodular function f is non-decreasing, i.e., for any $A \subseteq V$ and $v \in A$ holds $f(A) \geq f(A-v)$.*

Proof. Since f is non-negative, we have $f(v) \geq f(\emptyset)$. Adding one by one all $u \in A - v$ on both sides proves the claim by (E2). □

Proposition 2. *Let f be a value-monotone submodular function. For any $A, B \subseteq V$ such that $f(A) \geq f(B)$ and $u, v \in V \setminus (A \cup B)$ such that $f(u) \geq f(v)$, holds*

$$f(A + u) \geq f(B + v).$$

Proof. With Proposition 1, f is non-decreasing, and we have $f(A \setminus B) \geq f(\emptyset)$. Adding element u on both sides gives with (E2) $f(A \setminus B + u) \geq f(u) \geq f(v)$. Now, adding B on both sides gives again with (E2) the desired result. □

Proposition 3. *Let f be a value-monotone submodular function. Consider any two sets $A, B \subseteq V$ with $|A| \geq |B|$. If each element $v \in B$ can be mapped to a distinctive element $u \in A$ such that $f(v) \leq f(u)$, then $f(A) \geq f(B)$.*

Proof. Starting from two empty sets, iterative application of Proposition 2 gives the proof for two sets A and B with the same cardinality. With f being non-decreasing (Proposition 1) the claim follows. □

Value-monotone submodular functions extend the definition of value-polymatroidal set functions introduced by Gijswijt et al. [25]. A *value-polymatroidal set function f* is a *polymatroid rank function* (i.e., non-decreasing, submodular, and $f(\emptyset) = 0$) that satisfies a weaker property than the value-monotone exchange property (E1): for every $A, B \subseteq V$ such that $f(A) \geq f(B)$ and every $u \in V \setminus (A \cup B)$, holds that $f(A+u)+f(B) \leq f(A)+f(B+u)$. Example 1 in the Appendix shows that there are value-polymatroidal set functions which do not satisfy our properties, neither the value-monotone exchange property (E1) nor the value-monotone enlargement property (E2). And we will show that both properties are necessary to obtain our results. However, we are not aware of any natural cost function that is value-polymatroidal but not value-monotone. The following follows trivially from (E1).

Observation 1. *A value-monotone submodular function is value-polymatroidal.*

We remark that the name *value-monotone submodular functions* intentionally contains some redundancy. In fact, any non-negative set function $f: 2^V \to \mathbb{R}$ that satisfies the value-monotonicity properties (E1) and (E2) is actually submodular. This follows immediately from the following well-known characterization of submodular functions: f is submodular if for all sets $B \subseteq A \subseteq V$ and any element $u \in V \setminus (A \cup B)$ holds

$$f(A + u) + f(B) \leq f(A) + f(B + u).$$

3. Complete graphs

In this section, we consider the problem of partitioning a complete graph K into sub-cliques, PCliq(K, f, B). First we show, that the problem is generally $\mathcal{NP}$-hard for non-negative submodular functions even when restricting to polymatroid rank functions. Then we show that with the value-monotonicity properties (E1) and (E2), the problem becomes solvable in polynomial time.

Theorem 4. *The problem* PCliq(K, f, B) *on a clique K is $\mathcal{NP}$-hard even if we restrict f to polymatroid rank functions.*

Proof. We show the theorem by reduction from *graph partitioning* with unit node weights. Given is a graph $G = (V, E)$ with edge weights $w(e) \in \mathbb{Z}^+$, for all $e \in E$, and two positive integers B and L. The question is, if there is a partition of V into disjoint sets $V_1, \ldots, V_\ell$ such that $|V_i| \leq B$ for $1 \leq i \leq \ell$ and such that $\sum_{e \in E_\delta} w(e) \leq L$, where E_δ denotes the set of edges that have endpoints in two *different* sets V_i. This problem is known to be $\mathcal{NP}$-hard [35].

Given an instance of graph partitioning, we build the complete graph $K = (V, E')$ by complementing G with all edges in the complement graph of G. We assign edge weights $w(e) = 0$, for $e \in E' \setminus E$. Abusing notation, let $e \in \delta(S)$ denote an edge e that has one or both endpoints in the set $S \subseteq K$. We define the set function $f: 2^V \to \mathbb{Z}^+$ as

$$f(S) = \sum_{e \in \delta(S)} w(e), \quad \text{for all } S \subseteq 2^V.$$

It can be easily verified that f is non-decreasing and submodular, and the empty set has value zero. Thus, f is a polymatroid rank function. Notice, that in a partition each edge $e \in E'_\delta$ with endpoints in two different sets contributes once to the value of each of these sets. Thus, any partition $\mathcal{S}$ of K has a total value

$$\sum_{S \in \mathcal{S}} f(S) = \sum_{S \in \mathcal{S}} \sum_{e \in \delta(S)} w(e) = \sum_{e \in E'} w(e) + \sum_{e \in E'_\delta} w(e).$$

By the choice of edge weights, there exists a partition $\mathcal{S}$ of clique K into sets of size at most B with total value $\sum_{S \in \mathcal{S}} f(S) - \sum_{e \in E'} w(e) \leq L$ if and only if there exists a graph partition for G with value $\sum_{e \in E_\delta} w(e) \leq L$. □

In case that f is a value-monotone submodular function, we can solve the problem optimally in polynomial time. In that case, an optimal solution can be characterized as follows.

Lemma 5. *Consider the problem* PCliq(K, f, B) *for a clique* K*, and let* $f\colon 2^V \to \mathbb{R}^+$ *be a value-monotone submodular function. There is an optimal partition* $\mathcal{S}$ *in which at most one set* $S \in \mathcal{S}$ *has cardinality less than* B*. If such a set* S *exists, then it has value* $f(S) = \min\{f(T) \mid T \in \mathcal{S}\}$.

Proof. Let the set family $\mathcal{S}^*$ be an optimal solution for the problem PCliq(K, f, B) with more than one set with cardinality less than B. We show how to transform it into an optimal partition where no set or only the set with minimum value has cardinality less than B.

Let $T \in \mathcal{S}^*$ be a set with $|T| < B$ with maximum value, and let $S \in \mathcal{S}^*$ denote the set with minimum value. If $|T|+|S| \leq B$, then by submodularity of f it holds $f(T \cup S) \leq f(T) + f(S)$. Therefore, merging sets T and S yields a new feasible solution of value no more than the optimal value.

Assume that $|T| + |S| > B$. Since f is non-decreasing, $f(T) \geq f(S) \geq f(S-u)$ for any $u \in S$. Such an element must exist, because otherwise $f(T) < f(S)$. It follows from Observation 1 that $f(T+u) + f(S-u) \leq f(T) + f(S)$. Therefore, moving elements $u \in S$ from S to T will not increase the total value of the partition. Repeat this procedure until all sets $T \in \mathcal{S}^*$ have reached cardinality B. Notice that the value $f(S)$ will not increase and therefore, S will be the set of minimum value at any step in the procedure. □

Using this lemma, we show that the problem PCliq$(K, f, B,)$ on a clique K can be solved optimally in polynomial time by the following the Greedy algorithm.

Algorithm 1: Greedy algorithm for partitioning a clique K

1 Order elements $u \in K$ in non-increasing order of $f(u)$ and group them greedily into sub-cliques of cardinality B beginning with the elements of largest value.

Theorem 6. *Algorithm* 1 *solves the problem PCliq*(K, f, B) *of partitioning a clique* K *into sub-cliques of cardinality at most* B *for value-monotone submodular functions to optimality.*

Proof. Let $\mathcal{S}^*$ be an optimal solution. We show how to transform it such that it coincides with the solution obtained by Algorithm 1. Order all sets in $\mathcal{S}^*$ in non-increasing order of set values. By Lemma 5 we can assume that at most one set in $\mathcal{S}$, namely the last set in this order, has cardinality less than B. If $\mathcal{S}^*$ does not coincide with the solution of the algorithm, then there exist two sets $S, T \in \mathcal{S}^*$ with $f(S) \leq f(T)$ that contain two elements $u \in S$ and $v \in T$ with $f(u) > f(v)$. Let S and T be the first such sets in the order. In case $f(S) = f(T)$, we simply exchange u and v without increasing the total value as property (E1) guarantees. Suppose now that $f(S) \neq f(T)$.

If $f(S-u) \geq f(T-v)$, then by Proposition 2 holds $f(S) \geq f(T)$ which is a contradiction. Therefore, we can assume that $f(S-u) < f(T-v)$. In that case the value-monotone exchange property (E1) implies $f(S)+f(T) \geq f(S-u+v)+f(T-v+u)$. Thus, exchanging elements u and v will not increase the objective function. Repeat this exchange procedure until no two sets S and T with elements $u \in S$ and $v \in T$ exist such that $f(u) > f(v)$ and $f(S) \geq f(T)$. This set partition coincides with the solution of Algorithm 1. □

For the interested reader we provide an example in which the Greedy algorithm fails for a value-polymatroidal function that does not satisfy properties (E1) and (E2) in the Appendix.

4. General graphs

Consider the following polynomial time algorithm for the problem of partitioning an arbitrary graph into cliques of bounded cardinality.

Algorithm 2: Partitioning into cliques of bounded cardinality

1 Find a partition into cliques $K_1, K_2 \ldots$ for the relaxed problem PCliq(G,f,∞) without cardinality restrictions.
2 **forall the** K_i **do**
3 | Solve the problem PCliq(K_i,f,B) on the clique K_i.
4 **end**

Theorem 7. *Let* $f\colon 2^V \to \mathbb{R}^+$ *be a value-monotone submodular function. If there exists an* α*-approximation algorithm for the problem* PCliq(G, f, ∞) *with set function* f*, then Algorithm* 2 *with Algorithm* 1 *as a subroutine is an* $(\alpha + 1)$*-approximation for problem* PCliq(G, f, B).

Proof. Let $K_1, \ldots, K_\ell$ be the solution of the relaxed problem PCliq(G, f, ∞) in Step 1 of the algorithm. For each of the cliques K_i let $K_i^1, \ldots, K_i^{\ell_i}$ denote the partition into sub-cliques of bounded cardinality in Step 2. Assume that all sets are indexed such that $f(K_i^j) \geq f(K_i^{j+1})$ for $j = 1, \ldots, \ell_i - 1$. Then the value of the algorithms solution is

$$\textsc{Alg} = \sum_{i=1}^{\ell}\sum_{j=1}^{\ell_i} f(K_i^j) = \sum_{i=1}^{\ell} f(K_i^1) + \sum_{i=1}^{\ell}\sum_{j=2}^{\ell_i} f(K_i^j) \leq \alpha\, \textsc{Opt}(G, \infty) + \sum_{i=1}^{\ell}\sum_{j=2}^{\ell_i} f(K_i^j). \tag{1}$$

Here, $\textsc{Opt}(G, \infty)$ denotes the optimal solution of the relaxed problem $\text{PCliq}(G, f, \infty)$ without cardinality constraints. The inequality holds true since the set function f is non-decreasing (Proposition 1), and thus, $f(K_i) \geq f(K_i^j)$ for any $j = 1, \ldots, \ell_i$.

We bound the second term in (1) by the optimal solution $\textsc{Opt}(K_n, B)$ of the relaxed problem $\text{PCliq}(K_n, f, B)$ on the complete graph K_n on node set V. To that end, let π denote a permutation of all nodes in which all $u \in V$ appear in non-increasing order of their values $f(u)$. Ties are broken by giving preference to the node with the smaller index. By Theorem 6, an optimal partition $\textsc{Opt}(K_n, B)$ consists of the sets $S_k^* = \{\pi((k-1)B+1), \pi((k-1)B+2), \ldots, \pi(kB)\}$, for $k = 1, \ldots, \lfloor |V|/B \rfloor$ and the set $S_{\lfloor |V|/B \rfloor + 1}$ of remaining nodes, if any.

Consider now the partial solution of the algorithm K_i^j for $i = 1, \ldots, \ell, j = 2, \ldots, \ell_i$. Recall that the algorithm sorts the sets K_i in non-increasing function value yielding sets K_i^j such that $|K_i^j| = B$, for $j \neq \ell_i$, and $|K_i^{\ell_i}| \leq B$ (and note that this implies that $B \sum_{i=1}^{\ell}(\ell_i - 1) < |V|$). Consider now all these sets K_i^j for $i = 1, \ldots, \ell$, $j = 2, \ldots, \ell_i$ sorted in non-increasing order of the function value of their first element. We argue that the kth set in this order, say K_i^j, satisfies that $f(K_i^j) \leq f(\{\pi((k-1)B+1), \pi((k-1)B+2), \ldots, \pi(kB)\}) = f(S_k^*)$. Indeed, let v be the first element in K_i^j. Let $\tau_s(v)$ be the total number of elements in sets K_s^j with $j \in \{2, \ldots, \ell_s\}$, with larger or equal f-value (excluding equal-value elements with larger index) and let $S(v) := \{s \mid \tau_s(v) > 0\}$. Hence, in K_s there are at most $\lceil \tau_s(v)/B \rceil$ sets K_s^j with first element having larger or equal function value when discarding the first set, K_s^1. As v is the first element in the k-th set in the partial solution, we have $\sum_{s \in S(v)} \lceil \tau_s(v)/B \rceil \geq k$. To conclude we note that the total number of elements with larger function value than v appearing in π before v (including those in K_s^1) is

$$\sum_{s \in S(v)} (\tau_s(v) + B) = B \sum_{s \in S(v)} (\tau_s(v)/B + 1) \geq B \sum_{s \in S(v)} \lceil \tau_s(v)/B \rceil \geq kB.$$

Using Proposition 3 we can finally bound the second term in (1) as follows:

$$\sum_{i=1}^{\ell} \sum_{j=2}^{\ell_i} f(K_i^j) \leq \sum_{k=1}^{\sum_{i=1}^{\ell}(\ell_i - 1)} f(S_k^*) \leq \sum_{k=1}^{\lfloor |V|/B \rfloor + 1} f(S_k^*) \leq \textsc{Opt}(K_n, B) \leq \textsc{Opt},$$

obtaining the desired approximation guarantee of $\alpha + 1$. □

Theorem 7 gives a general approximation guarantee as a function of the approximability of the relaxed problem setting with unbounded clique cardinality. Certainly, this implies several approximation results for specific graph classes and cost functions that have been considered in the unbounded setting. We mention in the following a few quite general implications.

Gijswijt et al. [25] provide a dynamic programming algorithm that solves the problem $\text{PCliq}(G, f', \infty)$ for G being an interval graph or a circular arc graph to optimality if f' is a value-polymatroidal set function. By Observation 1 a value-monotone submodular function is value-polymatroidal. Thus, combining the results in [25] with Theorem 7 implies the following result.

Corollary 8. *Algorithm 2 yields a 2-approximation for* $\text{PCliq}(G, f, B)$ *for value-monotone submodular functions f if G is an interval graph or a circular arc graph.*

Epstein and Levin [27] gave an e-approximate algorithm for the max-coloring problem in perfect graphs. Since the complement of a perfect graph is perfect, this implies together with Theorem 7 the following result.

Corollary 9. *Algorithm 2 yields an approximation factor of $e + 1 \approx 3.72$ for* $\text{PCliq}(G, \max, B)$ *in a perfect graph.*

Furthermore, Fukunaga, Halldórsson, and Nagamochi [24] consider generalized cost functions for coloring graphs that are monotone and concave in the total sum of node weights of the class. If non-decreasing, then such functions are value-monotone submodular and Theorem 7 implies the following result.

Corollary 10. *Algorithm 2 yields an approximation factor of 5 for* $\text{PCliq}(G, f, B)$ *for non-decreasing weight-defined concave functions f if G is a co-interval, comparability, or co-comparability graph. And it is a 7-approximation for perfect graphs.*

5. Proper interval graphs

In this section we consider proper interval graphs. It is a well-studied subclass of interval graphs with many practical applications. In Section 5.1 we design a dynamic programming algorithm for the clique partitioning problem with bounded cardinality. It runs in polynomial time when the number of different node types is bounded. Here we define that two nodes u, v are of the same *node type* with respect to the cost function f if $f(S+u) = f(S+v)$ for any set S. In Section 5.2 we apply recent results for the special case of the maximum-function to our dynamic program and obtain a PTAS for $\text{PCliq}(G, \max, B)$ when G being a proper interval graph. We remark that the complexity status of our problem remains open for proper interval graphs.

An *interval graph* $G = (V, E)$ is a graph where a continuous interval can be assigned to each node such that two nodes are adjacent if and only if their intervals intersect. The set of intervals assigned to the nodes in G is called *interval representation* of G. Numerous algorithms are known which recognize interval graphs in linear time. The first one was presented in [36]; this algorithm – as well as the majority of subsequent algorithms – utilizes the following characterization of interval graphs provided in [37].

Theorem 11 ([37])**.** *A graph is an interval graph if and only if its maximal cliques can be linearly ordered in such a way that for every vertex the maximal cliques to which it belongs occur consecutively in the linear order.*

Such a linear order of maximal cliques $(C_1, C_2, \ldots, C_k)$ of a graph G is called *clique path* of G.

A *proper interval graph* is an interval graph with an interval representation in which no interval is properly contained in another. Any connected proper interval graph has a unique clique path [38] and the number of maximal cliques is at most $|V| = n$.

5.1. Dynamic programming

Let $G = (V, E)$ be a proper interval graph. We may assume that it is connected, otherwise we consider each connected component separately. We denote by $(C_1, C_2, \ldots, C_k)$ the unique clique path of G which can be computed in linear time; e.g. [36]. In our further explanations, we consider the interval representation of the graph in connection to the clique path. We say an interval *starts (ends)* in the maximal clique C_i, if the corresponding node v satisfies $v \in C_i$ and $v \notin C_\ell$, for $\ell < i$ ($\ell > i$). An interval with nodes $v \in C_i$ which neither starts nor ends in C_i is said to *pass* C_i. Moreover, we say that an interval corresponding to node v has an *endpoint left of* the endpoint of another interval corresponding to node u, if and only if there exists some $i < \ell$ such that $v \in C_i$, $v \notin C_\ell$ and $u \in C_\ell$. The phrase *endpoint right of* is used accordingly. Since we consider proper interval graphs, an ordering of intervals according to start points is equivalent to an ordering by endpoints.

Let w be the number of different node types in graph G with respect to the cost function f. For each maximal clique C_i, $i = 1, \ldots, k$, and each node type $j = 1, \ldots, w$ we determine the three parameters $start^i_j$, $pass^i_j$, and end^i_j which denote the number of intervals of type j that start in, pass, and end in clique C_i, respectively.

A state in our dynamic program is defined as $S_i = [i, open^i_1, open^i_2, \ldots, open^i_w]$ which specifies a maximal clique C_i and a w-tuple of positive integers. A state defines a sub-solution $\mathcal{K}_i$, that is, a set of cliques of cardinality at most B, with the following properties:

(i) All intervals ending in cliques $C_1, C_2, \ldots, C_i$ in the clique path are assigned to solution cliques in $\mathcal{K}_i$.
(ii) The number of intervals of node type $j = 1, \ldots, w$ that are not assigned in $\mathcal{K}_i$, we say they are *open*, equals the value $open^i_j$ with $0 \le open^i_j \le \sum_{\ell=1}^{i} start^\ell_j - end^\ell_j$.
(iii) The open intervals of type j, $j = 1, \ldots, w$, are those with the most right endpoints among all intervals of type j passing C_i.

Notice that the number of states associated with a maximal clique C_i is at most n^w. We define an additional state $S_0 = [0, 0, \ldots, 0]$ as *start state*. Now, we establish a state graph F by linking two states S_i and S_{i+1}, $0 \le i \le k-1$, by an edge if

$$open^{i+1}_j \le start^{i+1}_j + \min\{open^i_j,\ pass^{i+1}_j\}, \quad \text{for all } 1 \le j \le w. \tag{L}$$

Lemma 12. *If a feasible solution $\mathcal{K}$ corresponding to state S_i can be augmented to a feasible solution $\mathcal{K}'$ to state S_{i+1} such that $\mathcal{K} \subseteq \mathcal{K}'$, then there is an edge between states S_i and S_{i+1}.*

Proof. Given the states S_i and S_{i+1} and numbers of open intervals $open^i_j$ and $open^{i+1}_j$ for each node type j, we compute the number of intervals fix_j that must be assigned to augmenting solution cliques $\mathcal{K}' \setminus \mathcal{K}$. This value equals the difference between the number of intervals that is available in clique C_{i+1} and the number of intervals that must be kept open:

$$fix_j = open^i_j + start^{i+1}_j - open^{i+1}_j.$$

The values $fix_1, fix_2, \ldots, fix_w$ and property (iii) define precisely the set V' of intervals that has to be assigned to solution cliques $\mathcal{K}' \setminus \mathcal{K}$.

Given a feasible solution to S_i then by Proposition (i) all intervals ending in cliques $C_1, \ldots, C_i$ are assigned to solution cliques in $\mathcal{K}$. To yield a feasible solution for S_{i+1} we need to assign all open intervals from state S_i that end in S_{i+1} to solution cliques in $\mathcal{K}' \setminus \mathcal{K}$ (and possibly additional open intervals). Using Property (L) we have that

$$\begin{aligned} fix_j &\ge open^i_j + start^{i+1}_j - \left(start^{i+1}_j + \min\{open^i_j,\ pass^{i+1}_j\}\right) \\ &= open^i_j - \min\{open^i_j,\ pass^{i+1}_j\}. \end{aligned}$$

Thus, by property (ii) it is ensured that fix_j is non-negative. And moreover, fix_j is at least as large as the number of intervals ending in C_{i+1} which are not assigned in $\mathcal{K}$. To see that, recall that S_i satisfies property (iii), that is, if intervals are left open in S_i then these are the ones with the most right endpoints. In case that $open^i_j > pass^{i+1}_j$, there exist open intervals of type j,

that end in C_{i+1}. They certainly must be assigned now, and in fact their number is $open_j^i - pass_j^{i+1}$, which equals the minimum number of intervals we request to assign to reach the state S_{i+1}. □

Let the cost of an edge (S_i, S_{i+1}) be the minimum cost for augmenting a feasible solution $\mathcal{K}$ for S_i to a feasible solution $\mathcal{K}'$ for S_{i+1}. From the previous lemma we know how to determine a set V' of intervals that constitute the new solution cliques $\mathcal{K}' \setminus \mathcal{K}$. Clearly, the set V' itself forms a clique, and we can run the Greedy Algorithm 1. By Theorem 6 Greedy finds a solution of minimum total cost for V', and thus, it directly implies the following corollary.

Corollary 13. *For two given states S_i and S_{i+1} that have an edge in the state graph, the minimum cost augmentation is $\mathcal{K}' \setminus \mathcal{K}$, and thus the cost of the edge, can be computed in $\mathcal{O}(n \log n)$ by the Greedy Algorithm* 1.

Lemma 14. *An optimal solution can be represented as a path in the state graph from the start state S_0 to a state $S_k = [k, 0, 0, \ldots, 0]$, and the cost of the solution is the sum of the edge cost on that path.*

Proof. Given an optimal solution $\mathcal{K}$ we construct a feasible solution $\mathcal{K}'$ of the same total cost such that we can partition it into subsets that can be associated with maximal cliques in the clique path and thus with states in our state graph.

Let v_1 be the node with the most left ending point over all nodes. Consider the solution clique K_1 that contains v_1. We want to rearrange the interval-to-clique assignment such that for each node type appearing in K_1, the intervals with the most left endpoints of this type are assigned. Suppose that there are two intervals of the same type associated with nodes $v \in K_1$ and $v' \in K_\ell,\ \ell \neq 1$, where v has an endpoint more to the right than v'. Choose the interval with most left endpoint among such v'. We can interchange the node-to-clique assignments of v and v', because the order of endpoints and start points of intervals in a proper interval is equal, and thus, v' forms a clique with the remaining intervals in K_1 if v did, and vice versa for v and K_ℓ. The total cost of the solution does not change, since we exchanged intervals of the same type. We repeat this interchange argument for all types appearing in K_1 and then remove the transformed K_1 from the sets under consideration. Repeating the procedure for any clique and node type transforms the optimal solution to another optimal solution $\mathcal{K}'$.

Now, we partition $\mathcal{K}'$ into disjoint subsets $\mathcal{K}'_1, \ldots, \mathcal{K}'_k$ such that $\mathcal{K}'_i$ consists of all cliques whose interval with the most left ending point ends in the maximal clique C_i. Observe that in any sub-solution $\bigcup_{\ell=1}^{i} \mathcal{K}'_\ell$, all intervals that end in maximal cliques $C_1, \ldots, C_i$ are assigned, which satisfies Proposition (i). Due to the interchange procedure the open intervals of a type j, $j = 1, \ldots, w$, are those with the most right endpoints among all intervals of type j passing C_i, which satisfies Proposition (iii). Moreover, the number of open intervals by definition satisfies Proposition (ii). Thus, any sub-solution $\bigcup_{\ell=1}^{i} \mathcal{K}'_\ell$ satisfies Properties (i)–(iii), and is therefore, represented by a valid state S_i in the state graph. By construction and Lemma 12, any two such states S_i and S_{i+1} are connected by an edge since the sub-solution $\bigcup_{\ell=1}^{i} \mathcal{K}_\ell$ corresponding to S_i can be augmented by $\mathcal{K}_{i+1}$ to an optimal sub-solution encoded in S_{i+1}, and the edge cost corresponds by Corollary 13 to the cost of $\mathcal{K}_i$. Thus, there is a minimum cost path from the start node to the state S_k representing the total optimal solution and its cost. □

Theorem 15. *Let $G = (V, E)$ be a proper interval graph, and let f be a value-monotone submodular function. There is a dynamic program that solves the problem* PCliq*(G, f, B) optimally in time $\mathcal{O}(n^{2w+1})$, where w is the number of different node types in V with respect to f.*

Proof. By Lemma 14, we find an optimal solution as follows: we construct the state graph with at most n^w states per maximal clique C_i, $0 \leq i \leq k \leq n$, which gives $\mathcal{O}(n^{w+1})$ nodes. This can be done in running time $\mathcal{O}(n^{2w+1})$. Then we find a minimum cost path from state S_0 to a state $[k, 0, \ldots, 0]$; we may use, e.g., Dijkstra's algorithm [39] since we have non-negative cost, which gives a total computation time of $\mathcal{O}(n^{2w+1})$. After choosing the minimum cost final solution we find the actual clique partitioning solution by backtracking. □

Corollary 16. *Let $G = (V, E)$ be a proper interval graph, let f be a value-monotone submodular function, and let the number w of different node types in V with respect to f be constant. A dynamic program solves the problem* PCliq*(G, f, B) optimally in polynomial time.*

We remark that this dynamic programming construction has significantly larger running time for general interval graphs. In that case, we cannot restrict the sub-solutions for a state to those that leave open the intervals with right most endpoints.

5.2. A polynomial time approximation scheme for the max-function

Let now f be the max-function. In this case, the number of different node types, w, is the number of distinct node values $f(u), u \in V$. It has been shown by Nonner [12,28] that we may assume this is constant by losing at most a factor $1+\varepsilon$, for any $\varepsilon > 0$. The procedure to achieve this runs in polynomial time. First, we geometrically round node values to powers of $1+\delta$, for some $\delta > 0$. Then we apply an adaptation of the Hochbaum and Maass' shifting technique [40]. We consider partitions of the total range of different node values into subranges $[(1+\delta)^{z+(j-1)/\varepsilon}, (1+\delta)^{z+j/\varepsilon})$, with $j \in \mathbb{N}_0$ and shift parameter $1 \leq z \leq 1/\varepsilon$, each containing at most $1/\varepsilon$ different node values. These partitions induce a decomposition of the graph into

polynomially many disjoint subgraphs each having constantly many different node values. Of course, any such subgraph of a proper interval graph is again a proper interval graph, so that on each of these subgraphs we run an algorithm for constantly many node values and output the union of all solution cliques. The value of this solution is within a factor $1 + \varepsilon\delta$ of the optimal clique partitioning solution for the instance after rounding the node values. Choosing δ and ε appropriately, the approximation ratio is arbitrarily close to 1. This leads to the following result.

Lemma 17 ([12,28]). *Consider the problem* PCliq(G, max, B), *where G is proper interval graph. For any $\varepsilon > 0$, we may assume that the number of different node types is bounded by a constant losing at most a factor $1 + \varepsilon$ in the approximation ratio.*

Combining Lemma 17 with the dynamic program in Theorem 15 gives directly a PTAS for the problem with arbitrary node types and the max-function. Notice that this holds for arbitrary capacity bounds B, which is in contrast to a previous PTAS (for general interval graphs) [12] that requires constant B.

Corollary 18. *There is a polynomial-time approximation scheme for solving the problem* PCliq(G, max, B) *when G is a proper interval graph.*

6. Conclusion

Our contribution is twofold: On the one hand, we solve traditional cost coloring problems with an additional size bound, which is a very relevant constraint in many practical applications. And on the other hand, we continue a line of research that started recently [24,25] on developing unifying solutions for larger classes of cost functions. Motivated by different applications, various cost functions have been considered for partitioning or coloring problems. It is not only of theoretical but also practical interest, to provide an algorithmic framework that yields provably good solutions for a large class of cost functions with similar properties.

As a far reaching goal, it is desirable to develop methods with provably good performance for more general cost functions as in our setting, e.g., submodular functions with weaker or without additional constraints. As a first step, we raise the following specific open questions related to our simple, but very general two-phase algorithm: is there a factor β approximation, for some constant β, for the clique partitioning problem on a complete graph for submodular functions (concerns Step 2 of our algorithm)? In case of a positive answer: can this algorithm be used to turn an α-approximate solution for the unbounded clique partitioning problem into a $g(\alpha, \beta)$-approximate solution for the bounded problem variant for some reasonable function g?

Acknowledgments

We thank Karol Suchan for valuable discussion and Rajiv Raman for finding examples that distinguish value-monotone submodular functions from value-polymatroidal functions. The research was partially supported by Núcleo Milenio Información y Coordinación en Redes ICM/FIC P10-024F and by the German Science Foundation (DFG) under contract ME 3825/1.

Appendix

A.1. Special cases of value-monotone submodular functions

In this section we give the proofs showing that several above-mentioned and well-studied cost functions f belong to the class of value-monotone submodular functions. We show for each f under consideration that it satisfies for all subsets $A, B \subseteq V$ with $f(A) \geq f(B)$ and elements $u, v \in V \setminus (A \cup B)$ with $f(u) \geq f(v)$ the two properties

(E1) $f(A + u) + f(B + v) \leq f(A + v) + f(B + u)$, and
(E2) $f(A + u) \geq f(B + u)$.

Maximum function. Let $w\colon V \to \mathbb{R}_+$, $S \subseteq V$, and $f(S) := \max_{s\in S} w(s)$. The max-function f satisfies (E2) since $\max_{s\in A} w(s) = f(A) \geq f(B) = \max_{s\in B} w(s)$ implies $f(A + u) = \max\{\max_{s\in A} w(s), w(u)\} \geq \max\{\max_{s\in B} w(s), w(u)\} = f(B + u)$. To see that f satisfies (E1) consider two cases:

(i) If $\max_{s\in A} w(s) \geq w(u)$, then $f(A+u) = \max_{s\in A} w(s) \leq \max\{\max_{s\in A} w(s), w(v)\} = f(A+v)$. Furthermore, $f(B+v) \leq f(B+u)$ which follows from $f(v) \leq f(u)$ and adding all elements in B one by one using (E2). Thus, $f(A+u)+f(B+v) \leq f(A+v)+f(B+u)$.
(ii) In case that $\max_{s\in A} w(s) < w(u)$, we have $f(A + u) = w(u) \leq \max\{w(u), \max_{s\in B} w(s)\} = f(B + u)$ and $f(B + v) \leq f(A + v)$ by (E2), which implies that f satisfies (E1).

Probabilistic cost function. Let $q\colon V \to [0,1]$ and for $S \subseteq V$, $f(S) := 1 - \prod_{s\in S} q(s)$. Notice that $f(u) \geq f(v)$ implies $q(u) - q(v) \leq 0$. Then f satisfies (E1) because

$$\begin{aligned} f(A+u) - f(A+v) &= 1 - \prod_{s\in A} q(s)\cdot q(u) - \left(1 - \prod_{s\in A} q(s)\cdot q(v)\right) \\ &= \left(1 - \prod_{s\in A} q(s)\right)\cdot q(u) - q(u) + 1 - \left(\left(1 - \prod_{s\in A} q(s)\right)\cdot q(v) - q(v) + 1\right) \\ &= f(A)\cdot(q(u) - q(v)) - q(u) + 1 - (-q(v) + 1) \\ &\leq f(B)\cdot(q(u) - q(v)) - q(u) + 1 - (-q(v) + 1) \\ &= f(B+u) - f(B+v). \end{aligned}$$

The argument that f satisfies (E2) is similar. With $f(A) \geq f(B) \geq 0$ and $q(u) \geq 0$ we have

$$\begin{aligned} f(A+u) &= 1 - \prod_{s\in A} q(s)\cdot q(u) = \left(1 - \prod_{s\in A} q(s)\right)\cdot q(u) - q(u) + 1 = f(A)\cdot q(u) - q(u) + 1 \\ &\geq f(B)\cdot q(u) - q(u) + 1 = \left(1 - \prod_{s\in B} q(s)\right)\cdot q(u) - q(u) + 1 = 1 - \prod_{s\in B} q(s)\cdot q(u) \\ &= f(B+u). \end{aligned}$$

Non-decreasing weight-defined concave function. Let $w\colon V \to \mathbb{R}_+$, $S \subseteq V$, and let $f(S) := h(\sum_{v\in S} w(v))$ for some concave, non-decreasing function $h\colon \mathbb{R} \to \mathbb{R}$. The f satisfies (E2) since it is non-decreasing. To see that f satisfies (E1) we use the property that the marginal increase of concave functions h is decreasing, i.e., for $x < x'$ and $t \geq 0$ holds $h(x'+t) - h(x') \leq h(x+t) - h(x)$. Set $x = w(B) + w(v)$, $x' = w(A) + w(v)$, and $t = w(u) - w(v)$. Then we can conclude that

$$\begin{aligned} f(A+u) - f(A+v) &= h(w(A) + w(u)) - h(w(A) + w(v)) = h(x'+t) - h(x') \\ &\leq h(x+t) - h(x) = h(w(B) + w(u)) - h(w(B) + w(v)) \\ &= f(B+u) - f(B+v). \end{aligned}$$

Partial q-coloring and **modified chromatic entropy** cost functions are special cases of non-decreasing weight-defined concave functions and thus value-monotone submodular functions.

A.2. Value-polymatroidal vs. value-monotonicity

A natural question is to ask if the restriction of value-polymatroidal functions to value-monotone ones that satisfy the two value-monotonicity properties (E1) and (E2) is (i) a true restriction and (ii) indeed necessary, e.g., for the Greedy algorithm in Section 3 to be optimal. We give an example that answers both questions affirmatively. We give an (artificial) cost function that is value-polymatroidal but violates both properties. Furthermore, the Greedy algorithm for partitioning a complete graph into cliques fails to be optimal on this example. This shows that for optimality Greedy indeed requires that the cost function satisfies both additional value-monotonicity properties.

To simplify notation, we denote $f(\{x,y\})$ by $f(xy)$.

Example 1. The function f is defined on 4 elements a, b, c and d. The function values are as follows $f(\emptyset) = 0, f(a) = 4, f(b) = 5, f(c) = 6, f(d) = 7, f(ab) = 9, f(ac) = 10, f(ad) = 9, f(bc) = 9, f(bd) = 10, f(cd) = 10$, and all remaining values for sets of three or four nodes equal 11.

- *Value-polymatroidal.* It is easy to verify that f is value-polymatroidal, i.e., non-decreasing, submodular, $f(\emptyset) = 0$, and for every $A, B \subseteq V$ such that $f(A) \geq f(B)$ and every $u \in V \setminus (A \cup B)$ holds that $f(A+u) + f(B) \leq f(A) + f(B+u)$.
- *Property (E1) is violated.* Since $f(a) < f(b)$ and $f(c) < f(d)$, it must hold that $f(ac) + f(bd) \leq f(ad) + f(bc)$ to satisfy (E1). However, $f(ac) + f(bd) = 20$ while $f(ad) + f(bc) = 18$.
- *Property (E2) is violated.* Since $f(a) < f(b)$, adding element c on both sides must yield $f(ac) < f(bc)$ to satisfy (E2). However, $f(ac) = 10$, while $f(bc) = 9$.
- *Greedy is not optimal.* Suppose the capacity is $B = 2$. Then the algorithm yields the solution sets $\{c,d\}$ and $\{a,b\}$ with cost $10 + 9 = 19$. On the other hand, the solution $\{ad\}, \{bc\}$ has cost $9 + 9 = 18$ and is thus optimal.

References

[1] M.R. Garey, D.S. Johnson, Computers and Intractability, A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, New York, 1979.
[2] D. Zuckerman, Linear degree extractors and the inapproximability of max clique and chromatic number, Theory Comput. 3 (1) (2007) 103–128.
[3] H.L. Bodlaender, K. Jansen, Restrictions of graph partition problems. Part I, Theoret. Comput. Sci. 148 (1) (1995) 93–109.
[4] F. Gardi, Mutual exclusion scheduling with interval graphs or related classes. I, Discrete Appl. Math. 157 (1) (2009) 19–35.
[5] B.S. Baker, E.G. Coffman, Mutual exclusion scheduling, Theoret. Comput. Sci. 162 (2) (1996) 225–243.
[6] K. Jansen, The mutual exclusion scheduling problem for permutation and comparability graphs, Inform. and Comput. 180 (2) (2003) 71–81.
[7] P. Hansen, A. Hertz, J. Kuplinsky, Bounded vertex colorings of graphs, Discrete Math. 111 (1993) 305–312.

[8] E. Bampis, A. Kononov, G. Lucarelli, I. Milis, Bounded max-colorings of graphs, in: Proc. of the 21st International Symposium on Algorithms and Computation, (ISAAC 2010), in: Lecture Notes in Computer Science, vol. 6506, Springer, 2010, pp. 353–365.
[9] G. Finke, V. Jost, M. Queyranne, A. Sebő, Batch processing with interval graph compatibilities between tasks, Discrete Appl. Math. 156 (5) (2008) 556–568.
[10] M. Boudhar, Scheduling on a batch processing machine with split compatibility graphs, J. Math. Model. Algorithms 4 (4) (2005) 391–407.
[11] M. Boudhar, G. Finke, Scheduling on a batch machine with job compatibilities, Belg. J. Oper. Res. Stat. Comput. Sci. 40 (2000) 69–80.
[12] T. Nonner, Capacitated max-batching with interval graph compatibilities, in: H. Kaplan (Ed.), Proc. of the 12th Scandinavian Symposium and Workshops on Algorithm Theory, (SWAT 2010), in: Lecture Notes in Computer Science, vol. 6139, Springer, 2010, pp. 176–187.
[13] E. Cohen, M. Tarsi, NP-completeness of graph decomposition problems, J. Complexity 7 (2) (1991) 200–212.
[14] Z. Lonc, On complexity of some chain and antichain partition problems, in: Graph-Theoretic Concepts in Computer Science, Proc. 17th Int. Workshop, in: Lecture Notes in Computer Science, vol. 570, Springer, 1992, pp. 97–104.
[15] N. Alon, A note on the decomposition of graphs into isomorphic matchings, Acta Math. Hungar. 42 (1983) 221–223.
[16] M. Demange, D. de Werra, J. Monnot, V.T. Paschos, Time slot scheduling of compatible jobs, J. Sched. 10 (2) (2007) 111–127.
[17] B. Escoffier, J. Monnot, V.T. Paschos, Weighted coloring: further complexity and approximability results, Inform. Process. Lett. 97 (3) (2006) 98–103.
[18] C. Murat, V.T. Paschos, On the probabilistic minimum coloring and minimum k-coloring, Discrete Appl. Math. 154 (3) (2006) 564–586.
[19] N. Bourgeois, F.D. Croce, B. Escoffier, C. Murat, V.T. Paschos, Probabilistic graph-coloring in bipartite and split graphs, J. Comb. Optim. 17 (3) (2009) 274–311.
[20] C. Berge, Minimax relations for the partial q-colorings of a graph, Discrete Math. 74 (1989) 3–14.
[21] K. Cameron, A min–max relation for the partial q-colorings of a graph. Part II: Box perfection, Discrete Math. 74 (1989) 15–27.
[22] N. Alon, A. Orlitsky, Source coding and graph entropies, IEEE Trans. Inform. Theory 42 (5) (1996) 1329–1339.
[23] J. Cardinal, S. Fiorini, G. Joret, Minimum entropy combinatorial optimization problems, Theory Comput. Syst. 51 (1) (2012) 4–21.
[24] T. Fukunaga, M.M. Halldórsson, H. Nagamochi, Robust cost colorings, in: Proceedings of the 19th Annual ACM–SIAM Symposium on Discrete Algorithms, SODA'08, 2008, pp. 1204–1212.
[25] D. Gijswijt, V. Jost, M. Queyranne, Clique partitioning of interval graphs with submodular costs on the cliques, RAIRO Oper. Res. 41 (2007) 275–287.
[26] S.V. Pemmaraju, R. Raman, Approximation algorithms for the max-coloring problem, in: ICALP'05, in: Lecture Notes in Computer Science, vol. 3580, 2005, pp. 1064–1075.
[27] L. Epstein, A. Levin, On the max coloring problem, in: C. Kaklamanis, M. Skutella (Eds.), Proc. of the 5th International Workshop in Approximation and Online Algorithms, (WAOA 2007), in: Lecture Notes in Computer Science, vol. 4927, Springer, Berlin, Heidelberg, 2008, pp. 142–155.
[28] T. Nonner, Clique clustering yields a PTAS for max-coloring interval graphs, in: Proc. of the 38th International Colloquium on Automata, Languages and Programming, (ICALP 2011), in: Lecture Notes in Computer Science, vol. 6755, Springer, 2011, pp. 183–194.
[29] D. de Werra, M. Demange, B. Escoffier, J. Monnot, V.T. Paschos, Weighted coloring on planar, bipartite and split graphs: Complexity and approximation, Discrete Appl. Math. 157 (4) (2009) 819–832.
[30] B. Lehmann, D.J. Lehmann, N. Nisan, Combinatorial auctions with decreasing marginal utilities, Games Econom. Behav. 55 (2) (2006) 270–296.
[31] A. Federgruen, Y.-S. Zheng, The joint replenishment problem with general joint cost structures, Oper. Res. 40 (2) (1992) 384–403.
[32] M.T. Hajiaghayi, M. Mahdian, V.S. Mirrokni, The facility location problem with general cost functions, Networks 42 (1) (2003) 42–47.
[33] J. Körner, A. Orlitsky, Zero-error information theory, IEEE Trans. Inform. Theory 44 (6) (1998) 2207–2229.
[34] J. Cardinal, S. Fiorini, G. Joret, Minimum entropy coloring, J. Comb. Optim. 16 (4) (2008) 361–377.
[35] L. Hyafil, R.L. Rivest, Graph partitioning and constructing optimal decision trees are polynomial complete problems. Rapport de Recherche 33, IRIA-Laboria, Domaine de Voluceau, Rocquencourt, 78150 Le Chesnay, France, 1973. Cited in [1].
[36] K.S. Booth, G.S. Lueker, Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms, J. Comput. System Sci. 13 (3) (1976) 335–379.
[37] P.C. Gilmore, A.J. Hoffman, A characterization of comparability graphs and of interval graphs, Canad. J. Math. 16 (1964) 539–548.
[38] L. Ibarra, The clique-separator graph for chordal graphs, Discrete Appl. Math. 157 (8) (2009) 1737–1749.
[39] E. Dijkstra, A note on two problems in connexion with graphs, Numer. Math. 1 (1959) 269–271.
[40] D.S. Hochbaum, W. Maass, Approximation schemes for covering and packing problems in image processing and VLSI, J. ACM 32 (1) (1985) 130–136.