On Sum Coloring and Sum Multi-Coloring for Restricted Families of Graphs *

Allan Borodin, Ioana Ivan, Yuli Ye, and Bryce Zimny

Department of Computer Science, University of Toronto,
Toronto, ON, Canada M5S 3G4,
{bor,y3ye,zimny}@cs.toronto.edu
ioana.ivan@utoronto.ca

Abstract. We consider the sum coloring (chromatic sum) and sum multi-coloring problems for restricted families of graphs. In particular, we consider the graph classes of proper intersection graphs of axis-parallel rectangles, proper interval graphs, and unit disk graphs. All the above mentioned graph classes belong to a more general graph class of $(k+1)$-clawfree graphs (respectively, for $k = 4, 2, 5$).

We prove that sum coloring is NP-hard for penny graphs and unit square graphs which implies NP-hardness for unit disk graphs and proper intersection graphs of axis-parallel rectangles. We show a 2-approximation algorithm for unit square graphs, with the assumption that the geometric representation of the graph is given. For sum multi-coloring, we confirm that the greedy first-fit coloring, after ordering vertices by their demands, achieves a k-approximation for the preemptive version of sum multi-coloring on $(k+1)$-clawfree graphs. Finally, we study priority algorithms as a model for greedy algorithms for the sum coloring and sum multi-coloring problems. We show various inapproximation results under several natural input representations.

1 Introduction

The sum coloring problem (SC), also known as the chromatic sum problem, was formally introduced in [4]. For a given graph $G = (V, E)$, a proper coloring of G is an assignment of positive integers to its vertices $\phi : V \rightarrow Z^+$ such that no two adjacent vertices are assigned the same color. The sum coloring problem seeks a proper coloring such that the sum of colors over all vertices $\sum_{v \in V} \phi(v)$ is minimized. Sum coloring has many applications in job scheduling and resource allocation. Consider an instance of job scheduling in which one is given a set S of jobs, each requiring unit execution time. We construct the conflict graph G whose vertex set is in one-to-one correspondence with the set of input jobs S, and an edge exists between two vertices if and only if the corresponding jobs conflict for resources. Dividing the chromatic sum of the conflict graph $G = (V, E)$ by $n = |V|$ then determines the minimum average job completion

* This work was supported by the Natural Sciences and Engineering Research Council of Canada and the Department of Computer Science, University of Toronto.

time. Throughout this paper we will let n (respectively, m) denote the number of vertices (respectively the number of edges) in the input graph being considered.

The sum coloring problem has been studied extensively in the literature. The problem is NP-hard for general graphs [4], and cannot be approximated within $n^{1-\epsilon}$ for any $\epsilon > 0$ unless ZPP=NP [5][7]. The problem is polynomial time solvable for proper interval graphs [9] and trees [4]; however, it is APX-hard for both bipartite graphs [6] and interval graphs [11]. The best known approximation algorithm for interval graphs has approximation ratio 1.796 [16] and for bipartite graphs the best known is a $\frac{27}{26}$-approximation [12].

One well-studied extension of the sum coloring problem is sum multi-coloring (SMC): given graph $G = (V, E)$ and a demand function $x : V \rightarrow \{1, 2, ...\}$, color each vertex v with $x(v)$ different colors so as to minimize the sum of the maximum color assigned to each vertex while assigning distinct colors to adjacent vertices. That is, a coloring is now an assignment $\phi : V \rightarrow 2^{Z^+}$ such that $|\phi(v)| = x(v)$ for all v, and $\phi(u) \cap \phi(v) = \emptyset$ for all $(u, v) \in E$. There are two variants to the sum multi-coloring problem, namely the non-preemptive version (npSMC) where colors assigned to each vertex must be consecutive integers and the preemptive version (pSMC) in which colors need not be consecutive. One interesting fact about pSMC and npSMC is that there is a known polynomial time algorithm that solves npSMC for trees [13], while pSMC remains NP-hard for trees [14]. This is in contrast to other graph classes (see [19]) where the known approximation for pSMC is better than that for npSMC and also in contrast to the results in [10] that reduce pSMC to the weighted MIS problem, suggesting that npSMC is in general a harder problem. For a more complete review of previous results on sum coloring and sum multi-coloring, see [19].

In this paper, we consider the sum coloring and sum multi-coloring problem for restricted families of graphs with respect to both hardness and approximation algorithms. We shall always assume that the graph is connected since otherwise each connected component can be colored separately. The remainder of the paper is organized as follows. In Section 2, we discuss various classes of $(k+1)$-clawfree graphs. We prove the problem is NP-hard for penny graphs and unit square graphs, and show a 2-approximation for unit square graphs in Section 3. We study the sum multi-coloring problem for $(k + 1)$-clawfree graphs in Section 4 and priority inapproximations in Section 5. Section 6 concludes with some open problems.

2 $(k + 1)$-Clawfree Graphs

A graph is $(k+1)$-clawfree if every vertex has at most k independent neighbors. We follow the notation in [22] and let $\hat{G}(IS_k)$ denote the class of $(k+1)$-clawfree graphs. Similarly, we let $\hat{G}(VCC_k)$ denote the class of graphs for which the neighborhood of every vertex has a clique cover of size at most k. It is easy to see that $\hat{G}(VCC_k)$ is a subset of $\hat{G}(IS_k)$.

It turns out many interesting families of geometric intersection graphs are in the class $\hat{G}(VCC_k)$ and hence in the class of $\hat{G}(IS_k)$ for a small parameter k.

– *Proper Interval Graphs and Proper Circular Arc Graphs:* The vertices are intervals on the real line (respectively, arcs on a circle). Two vertices are adjacent if the two intervals (reps. arcs) intersect. The properness condition states that no interval (arc) is properly contained inside another interval (arc). Since the containment is proper, for any given interval (arc), any intersecting interval (arc) must intersect at one of its two end points. Therefore these graphs are in $\hat{G}(VCC_2)$.
– *Proper Intersection Graphs of Axis-Parallel Rectangles:* The vertices are axis-paralel rectangles. Two vertices are adjacent if the two rectangles intersect. Properness means that if rectangle R_1 interescts rectangle R_2 then the projection of R_1 (onto either the x or y-axis) is not properly contained in the projection of R_2. That is, the projection of the rectangles onto either the x or y axis becomes a proper interval graph. Since the containment is proper, for a given rectangle, every intersecting rectangle intersects it at one of its four corners. Therefore the underlying graph is in $\hat{G}(VCC_4)$. A special case is when the rectangles are axis parallel translates of a fixed axis parallel graph. In the case of translates of of a unit square, ther resulting graphs are called unit square graphs.

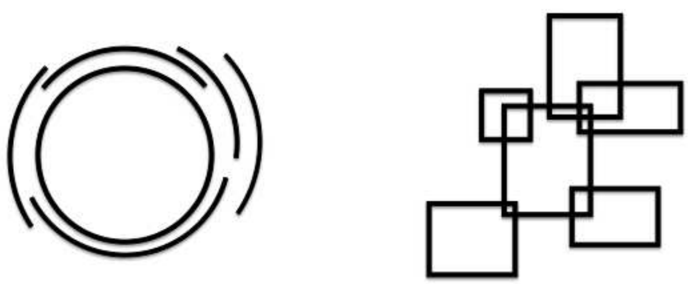


Fig. 1. Proper Circular Arc Graphs and Proper Intersection of Rectangles

– *Unit Disk Graphs:* The vertices are disks of unit size and two vertices are adjacent if the two disks intersect each other (including the boundary). For any disk, there are at most five pair-wise non-intersecting disks intersecting a single disk. Therefore the underlying graph is in $\hat{G}(IS_5)$. For any given unit disk, we can partition its conflicting region into six sectors so that any two unit disks whose centers lying in the same sector must intersect; see Figure 3 below. Therefore unit disk graphs are in $\hat{G}(VCC_6)$. It is not hard to show that this bound is tight; i.e., unit disk graphs are not in $\hat{G}(VCC_5)$. This gives a natural example which separates $\hat{G}(VCC_5)$ and $\hat{G}(IS_5)$. A special case of unit disk graphs is penny graphs where two disks cannot have a common interior point and two vertices are adjacent if the two disks touch each other at the boundary.

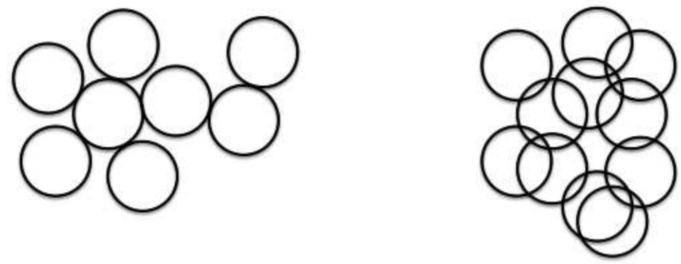


Fig. 2. Penny Graphs and Unit Disk Graphs

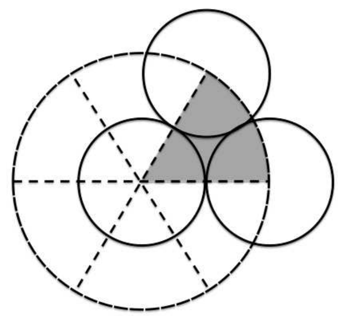


Fig. 3. Unit Disk Graphs are in $\hat{G}(VCC_6)$

- *Intersection of k-Sets:* The vertices are sets S_i of elements from some universe with $|S_i| \leq k$, and S_i and S_j are adjacent if and only if $S_i \cap S_j \neq \emptyset$. These graphs are in $\hat{G}(VCC_k)$.
- *Line Graphs:* The vertices are edges of an underlying graph and two vertices are adjacent if they share a common vertex in the underlying graph. It is easy to see that for a particular edge, it can have at most two non-intersecting edges intersect with it. Therefore, line graphs are in $\hat{G}(VCC_2)$.

With the exception of unit disk graphs, all the examples of $(k+1)$-clawfree graphs given are in the subclass $\hat{G}(VCC_k)$. However, we note that $\hat{G}(VCC_k)$ is a substantially different class of graphs than $\hat{G}(IS_k)$. In fact, based on a variation of Mycielski's construction [1], we can show that for every k, there is a 3-clawfree graph that is not in $\hat{G}(VCC_k)$. We also note that for fixed k, determining if G is in $\hat{G}(IS_k)$ can clearly be decided in time n^{k+2} whereas it is NP-hard to determine membership in $\hat{G}(VCC_k)$ for $k \geq 3$.

3 Sum Coloring for Unit Square Graphs

We show that the sum coloring problem for unit disk graphs and unit square graphs is NP-hard, and we develop a 2-approximation sum coloring algorithm for unit square graphs using a "strip" technique. For the hardness results, we use a reduction from the maximum independent set problem on planar graphs with maximum degree 3. We combine ideas in [17] and [13] and first show that sum coloring for penny graphs is NP-hard. We make use of the following observation from Valiant [3].

Lemma 1. *[3] A planar graph G with maximum degree 4 can be embedded in the plane using $O(|V|^2)$ area units in such a way that its vertices are at integer coordinates and its edges are drawn so that they are made up of line segments of the form $x = i$ or $y = j$, for integers i and j.*

Theorem 1. *Sum coloring is NP-hard for penny graphs.*

Proof. Given a planar graph G with maximum degree 3, we first apply the above lemma to draw its embedding onto integer coordinates, and without loss of generality we assume those coordinates are multiple of 8 units. We replace each vertex with a unit disk (a circle of diameter 1 unit), and for each edge uv, we replace it with l_{uv} tangent unit disks where l_{uv} is the Manhattan distance between u and v. We call the resulting penny graph G'. See figure 4. Note that there are three types of adjacent pair of unit disks. A corner pair refers two adjacent disks such that one of them is at the corner; an uneven pair refers two adjacent disks such that the center of one of them does not lie on the grid; the rest of the pairs are straight pairs. Let $\alpha(\cdot)$ denote the size of the maximum independent set. It is not hard to observe the following relationship between the maximum independent sets of the two graphs.

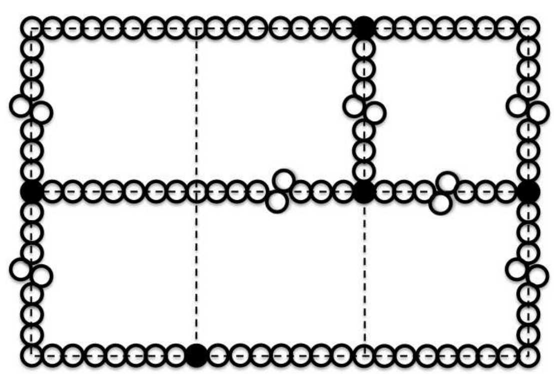


Fig. 4. Transformation from planar graphs with maximum degree 3 to penny graphs

Lemma 2. $\alpha(G') = \alpha(G) + \sum_{uv \in E} \frac{l_{uv}}{2}$.

Proof. We first show that $\alpha(G')$ is at least $\alpha(G) + \sum_{uv \in E} \frac{l_{uv}}{2}$. Given a maximum independent set I of G, then for any edge uv, at least one of u and v are not in I, hence we can add $\frac{l_{uv}}{2}$ alternating disks for each edge uv to form an independent set of G'. Therefore, $\alpha(G') \geq \alpha(G) + \sum_{uv \in E} \frac{l_{uv}}{2}$. On the other hand, given a maximum independent set I' of G', we can do the following modifications to I' without changing the size of I'. For each edge uv in G, if both u and v are in I', then the number of disks along the edge uv which are in I' must be less than $\frac{l_{uv}}{2}$, we can then remove, say v, from I' and increase the number of disks along the edge uv which are in I' by at least one. We keep doing that until for any edge uv in G there is at most one vertex in I'.

It is clear that after such modification, the vertices in $I' \cap G$ is an independent set for G, and hence $\alpha(G') \leq \alpha(G) + \sum_{uv \in E} \frac{l_{uv}}{2}$. □

We now do a second transformation. For each straight pair of adjacent unit disks, we do a transformation as shown in Fig. 5; for each uneven pair of adjacent unit disks, we do a transformation as shown in Fig. 6 and for each corner pair of adjacent unit disks, we do a transformation as shown in Fig. 7.

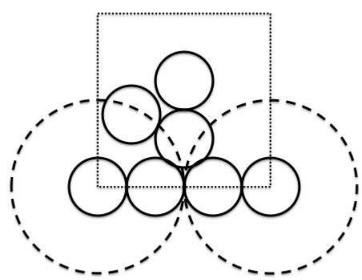


Fig. 5. Transformation for straight pairs

The purpose of the second transformation is that for each edge uv in G', we want to add an edge gadget as shown in Fig 8. Because the original graph is a planar graph with maximum degree 3, we can add these edge gadget in such a way that there are no overlapping disks and two disks in different gadgets do not touch each other. We call the resulting graph G''. Let m be the the number of edges in G'' and n the number of vertices, let $SC(G'')$ denote its chromatic sum. We now prove the following lemma to complete the reduction.

Lemma 3. $SC(G'') = 8m + 2n - \alpha(G')$.

Proof. We first show that the chromatic sum of G'' is at most $8m+2n-\alpha(G')$. To see that we give an explicit coloring of G''. Let I be the maximum independent

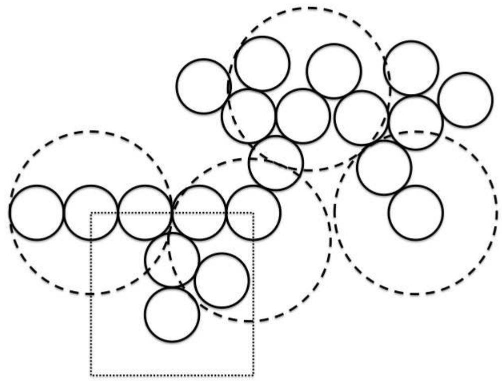


Fig. 6. Transformation for uneven pairs

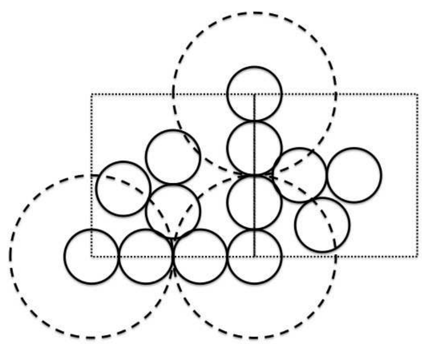


Fig. 7. Transformation for corner pairs

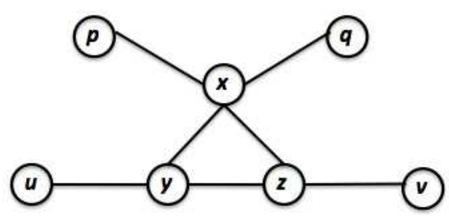


Fig. 8. The edge gadget

set of G', we coloring all vertices in I with color 1. We then color the remaining vertices in G' with color 2. Consider an edge gadget as depicted in figure 8. Since at least one of u and v is colored with 2, without loss of generality, assume u has color 2. We then color y with 1, z with 3, x with 2 and p, q with 1. Therefore the chromatic sum of G'' is at most $8m + 2n - \alpha(G')$.

We now show the chromatic sum of G'' is at least $8m + 2n - \alpha(G')$. Assume an optimal sum coloring, we first claim that all vertices in G' colored with 1 must form an independent set of G'. Suppose this is not the case and assume both u and v are colored with 1. There are two cases, the best possible choices of colors lead to Fig. 9 and 10, which achieves the sum of 13 and 12 respectively. If we recolor v with 2, we achieves the sum 11 as show in Fig. 11. However, recoloring v might lead to recoloring its other adjacent edge gadgets. We claim that we can coloring every other edge gadgets adjacent to v to maintain at least its original sum. Let u' be any other vertex adjacent to v in G', and y', z', x', p', q' be the corresponding vertices in the gadget, there are two cases:

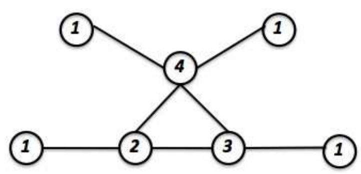


Fig. 9. Recoloring case 1

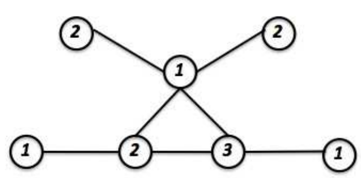


Fig. 10. Recoloring case 2

1. If u' is not colored with 2, then color z' with 1, y' with 2, x' with 3, p', q' with 1. This is the minimum possible, so it cannot exceed the original.
2. If u' is colored with 2, then color z' with 1, y' with 3, x' with 2, p', q' with 1. This is also the minimum possible, so it cannot exceed the original.

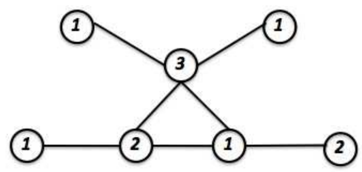


Fig. 11. Result coloring

Therefore by recoloring v with 2 and proper recoloring its neighborhood gadgets, we reduce the total sum, hence, all vertices in G' colored with 1 must form an independent set of G'. For the remaining vertices in G', we at least color them with 2 and for each gadgets, 8 is the best possible. Therefore the chromatic sum is at most $8m + 2n - \alpha(G')$. □

The NP-hardness follows immediately from Lemma 1, 2 and 3.

□

It follows immediately that sum coloring is NP-hard for unit disk graphs, since the class of penny graphs is a subclass of unit disk graphs.

Corollary 1. *Sum coloring is NP-hard for unit disk graphs.*

The transformation in the reduction to penny graphs also works for unit square graphs with a slight modification, i.e., by using unit squares instead of unit disks throughout the proof.

Theorem 2. *Sum coloring is NP-hard for unit square graphs.*

Since polynomial-time optimal algorithms are unlikely, we seek good approximations. For unit square graphs, we have the following observation: given a unit strip $\{(x, y)|y \in [i, i+1)\}$, consider the unit squares whose center lines lie inside this strip. Let H be the intersection graph induced by those unit squares, it is easy to observe the following:

Lemma 4. *H is a unit interval graph.*

It is known that unit interval graphs are the same graph class as proper interval graphs. Since sum coloring for proper interval graphs can be optimally solved in polynomial time [9] [1], we can optimally sum color H in polynomial time. Without loss of generality, we can assume that for a given geometric representation of a unit square graph G, the y-coordinates of all centers of the squares are in the range of $[0, h)$, so the plane can be partitioned into h horizontal strips. We now describe an algorithm which gives a proper coloring for the graph G.

[1] In fact, proper interval graphs can be optimally sum colored by a greedy algorithm running in time $O(n \log n)$ or just $O(n)$ if the n intervals are already sorted by non decreasing finishing time.

For each strip, color the graph induced by the squares in the strip with minimum sum. For each odd strip, and each color c used, use the new color $2c$. For each even strip, and each color c used, use the new color $2c-1$. This modified coloring is a proper coloring of the whole graph. This is because:

1. no two squares can intersect each other between two strips of the same parity,
2. the new coloring is still a proper coloring within each strip,
3. the new coloring does not create any violation between two adjacent strips.

Theorem 3. *Given a geometric representation, there is a simple greedy algorithm that achieves a 2-approximation to sum color an n node unit square graph. The running time is $O(n \log n)$.*

Proof. We use the above algorithm and first divide the graph into h strips. Since we assume connected graphs, h is bounded by the total number of unit squares. We optimally sum color each strip and let s_i be the chromatic sum of the graph induced by strip i. It is clear that the optimal solution is at least $\sum_i s_i$. Note that after the modification described above, the chromatic sum is at most $\sum_i 2s_i$. Therefore the algorithm achieves a 2-approximation.

Note that this algorithm is very efficient. The running time is $O(n \log n)$ if we are given the set of centers (in x and y coordinates) of the unit squares. The running time is dominated by sorting the x and y-coordinates. □

Extending an observation by Roberts [2], we show that the classes of proper intersection graphs of axis-parallel rectangles and unit square graphs coincide.

Theorem 4. *Proper intersection graphs of axis-parallel rectangles is the same class as that of unit square graphs.*

Proof. It is clear that unit square graphs are contained in the class of proper intersection graphs of axis-parallel rectangles. We only need to show the reverse direction. It is known by a result of Roberts [2] that the classes of proper interval graphs and unit interval graphs coincide. The proof of Bogart and West [8] gives an actual realization from a proper interval representation to a unit interval representation. This transformation can be done on both the x-axis and y-axis resulting in proper interval graphs, and then a unit square graph can be constructed by considering the unit intervals on both axes. Recent results of Gardi [18] and Lin et al. [21] show that such a transformation can be done efficiently in linear time and space. □

By Theorem 4, we immediately have the following corollary.

Corollary 2. *Given a geometric representation, there is an algorithm that achieves 2-approximation to sum color a proper axis parallel rectangle graph. The running time is $O(n \log n)$.*

4 Sum Multi-Coloring for $(k+1)$-Clawfree Graphs

A $(k+1)$-approximation to pSMC for the class of $\hat{G}(IS_k)$ was stated in [16] and a k-approximation was stated in [19]. Both refer to [10]. However, the proof in [10] (as well as the analogous result for sum coloring in [5]) seems only to extend to the class of $\hat{G}(VCC_k)$ as defined in Section 2. Similarly, the claim in [19] for npSMC on $k+1$ clawfree graphs only applies to the smaller class $\hat{G}(VCC_k)$. In this section, we provide a proof to confirm a k-approximation to pSMC for the class of $\hat{G}(IS_k)$. In contrast, we do not see how to extend the claimed result for npSMC. We follow the notation of [10]. For a given graph $G = (V, E)$, we denote

$$S(G) = \sum_{v \in V} x(v)$$

and

$$Q(G) = \sum_{(u,v) \in E} min(x(u), x(v))$$

In [10], the authors use the following greedy algorithm for pSMC: given a graph $G = (V, E)$, sort the vertices of G by non-decreasing demand and color the vertices in a first-fit manner. Bar-Noy et al. show that the sum of the multi-coloring obtained using this method is bounded above by $S(G) + Q(G)$ as an edge in the graph can only cause the incident vertex that is colored later to be given a higher color. We now seek to bound from below the cost of pSMC for a graph in $\hat{G}(IS_k)$ in terms of $S(G)$ and $Q(G)$.

Lemma 5. *For any graph $G = (V, E)$ in $\hat{G}(IS_k)$, the cost of a minimal sum multi-coloring of G is at least $\frac{1}{k} \cdot (S(G) + Q(G))$.*

Proof. Let $G = (V, E)$ be any graph in $\hat{G}(IS_k)$ and let ϕ be a multi-coloring of G with minimal sum. For a vertex v, denote by $f_\phi(v)$ the largest color assigned to v by ϕ. Again, we consider reconstructing G by adding back vertices in non-decreasing order of f_ϕ, breaking ties using some fixed ordering $<$of the vertices. lexicographically. In other words, we define a total ordering $\prec$, such that $u \prec v$ if and only if $f_\phi(u) < f_\phi(v)$ or $f_\phi(u) = f_\phi(v)$ and $u < v$. We consider a sequence of n distinct induced subgraphs of G in order of proper containment, with the vertex and edge sets growing based on a total ordering defined by $\prec$. When a vertex v is added to the graph, let $N'(v)$ denote the subset of $N(v)$ that is in the current induced subgraph. In other words:

$$N'(v) = \{u | u \in N(v) \text{ and } u \prec v\}$$

The total number of colors assigned to vertices in $N'(v)$ is equal to

$$\textstyle\sum_{u \in N'(v)} x(u)$$

and since the $k+1$ claw free property implies that no color can be used by more than k nodes, this impplies that the number of distinct colors used by vertices in $N'(v)$ is at least

$$\tfrac{1}{k} \cdot (\textstyle\sum_{u \in N'(v)} x(u))$$

From the above bound on the number of distinct colors, we may conclude:

$$\begin{aligned} f_\phi(v) &\geq x(v) + \frac{1}{k} \cdot \sum_{u \in N'(v)} x(u) \\ &\geq x(v) + \frac{1}{k} \cdot \sum_{u \in N'(v)} min(x(u), x(v)) \end{aligned}$$

Summing over all vertices and letting SMC(G) represent the multi-coloring sum, we obtain:

$$\begin{aligned} SMC(G) &= \sum_{v \in V} f_\phi(v) \\ &\geq \sum_{v \in V} (x(v) + \frac{1}{k} \cdot \sum_{u \in N'(v)} min(x(u), x(v))) \\ &\geq \frac{1}{k} \cdot (\sum_{v \in V} x(v) + \sum_{v \in V} \sum_{u \in N'(v)} min(x(u), x(v))) \\ &= \frac{1}{k} \cdot (S(G) + \sum_{(u,v) \in E} min(x(u), x(v))) \\ &= \frac{1}{k} \cdot (S(G) + Q(G)) \end{aligned}$$

□

In conjunction with the bound given by Bar-Noy et al. for greedy first-fit coloring, we conclude the following.

Theorem 5. *[10] For a graph $G \in \hat{G}(IS_k)$, $k \geq 2$, any multi-coloring obtained by a greedy first-fit coloring with respect to vertex demands is a k-approximation to pSMC on G.*

We note that Theorem 5 immediately implies the following corollary.

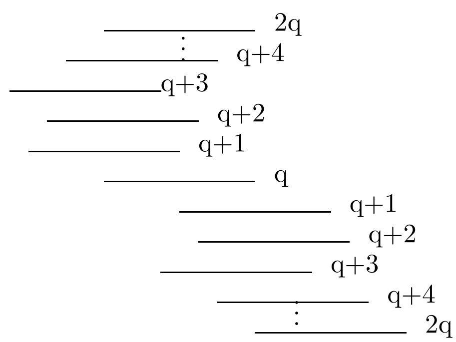
Corollary 3. *For a graph $G \in \hat{G}(IS_k)$, $k \geq 2$, a greedy first-fit coloring is a k-approximation to SC on G.*

5 Priority Inapproximation for SC and SMC

By Theorem 5, the greedy first-fit algorithm achieves a 2-approximation for proper interval graphs. It remains an open question whether or not these problems are NP-hard. Since the class of proper interval graphs is a very restricted family, it is natural to ask whether or not there is any greedy algorithm that can solve pSMC optimally or with an improved approximation. In what follows, we provide inapproximation results in the priority model as defined in [15]. For completeness we present the priority schema in Appendix A. We being with what might be considered the "natural input model" for the pSMC or npSMC problem on interval graphs. Namely, an input instance consists of a set of data items, each is represented by a time interval $[s_i, f_i)$ and its demand x_i, where s_i is its starting time and f_i its finishing time. We consider the adaptive priority algorithm model for which at each step the algorithm sees the data item with the highest priority based on a local ordering[2] and makes an irrevocable decision for that input item (i.e. interval). We have the following result for proper interval graphs.

Theorem 6. *There is no adaptive priority algorithm in the interval input model for pSMC or npSMC on proper interval graphs that can achieve approximation ratio better than* $\frac{5}{4}$.

Proof. We consider the following instance, see figure below:



All intervals are closed-open intervals with length $1+\delta$, for a very small δ. There is one interval with demand q starting at 1, and two intervals each for every $1 \leq i \leq q$ with demand $q+i$. For i odd, we have two intervals with demand $q+i$, one starting at $\frac{1}{2^i}$, which we will refer to as the top interval, and one starting at $1+\frac{1}{2^i}$, which we will refer to as the bottom interval. For i even, we have two intervals with demand $q+i$, a "bottom" one starting at $2-\frac{1}{2^i}$ and a "top" one starting at $1-\frac{1}{2^i}$. Let $f(I)$ be the highest color assigned to item I, there are three cases:

1. The algorithm first picks the interval I with demand q. If $f(I) \geq 2q$, the adversary removes all other intervals and we get an approximation ratio of 2. Otherwise, the adversary removes all items other than the top interval with demand $q+1$, and the bottom interval with demand $q+2$, which both intersect I, but not each other. We get $5q+3$, and the optimal value is $4q+5$. For large q, we can get an approximation ratio arbitrarily close to $\frac{5}{4}$.
2. The algorithm first picks one of intervals with demand $2q$, $2q-1$ or $2q-2$, call it I. If $f(I) \geq demand(I)+q$, the adversary removes all other intervals to get

[2] For the precise definition of a local ordering, refer to Appendix A.

an approximation ratio of at least $\frac{3}{2}$. Otherwise, it removes all intervals other than the one with demand q, and we get an approximation ratio arbitrarily close to $\frac{5}{4}$ for large q.

3. The algorithm first picks an interval I other than those mentioned above. If $f(I) \geq 2 \cdot demand(I)$, the adversary removes all other intervals and we get an approximation ratio of 2. Otherwise, there are four cases:
   - The interval I is a top interval which has demand $q+i$ with odd i. Then the adversary removes all items other than the top interval with demand $q+i+2$, and the bottom interval with demand $q+i$.
   - The interval I is a bottom interval which has demand $q+i$ with odd i. Then the adversary removes all items other than the top interval with demand $q+i$, and the bottom interval with demand $q+i+1$.
   - The interval I is a top interval which has demand $q+i$ with even i. Then the adversary removes all items other than the top interval with demand $q+i+1$, and the bottom interval with demand $q+i$.
   - The interval I is a bottom interval which has demand $q+i$ with even i. Then the adversary removes all items other than the top interval with demand $q+i$, and the bottom interval with demand $q+i+2$.

   For all the above cases, it is not hard to see that we get an approximation ratio arbitrarily close to $\frac{5}{4}$ for large q.

In all cases above, the algorithm would not get a better value by assigning non-consecutive colors to any interval, and the optimal solution assigns consecutive colors to each interval. Therefore, the statement holds for both pSMC and npSMC. □

For general (i.e. non proper) interval graphs, we have the following theorem.

Theorem 7. *There is no adaptive priority algorithm in the interval model for pSMC or npSMC on interval graphs that can achieve approximation ratio $\frac{3}{2}$.*

Proof. We start with an interval with demand q. We proceed inductively: for each interval with demand i, $q \leq i \leq pq-1$, we add m intervals with demand $i+1$, which are contained in it and does not intersect each other. Depending on the local ordering of the priority algorithm, there are two cases:

1. An item I with demand d, less than pq, has the highest priority. If $f(I) \geq 2d$, the adversary removes all other items and we get an approximation ratio of 2. Otherwise, we know there exist m intervals with demand $d+1$ which intersect I but not each other. If $f(I) < 2d$, the adversary removes all items other than these m intervals. We get an approximation ratio of $\frac{d+m(2d+1)}{m(d+1)+2d+1}$ $= \frac{2md+d+m}{md+m+2d+1}$, which is arbitrarily close to 2 for large m and d.
2. The item with highest priority has demand pq, call it I. If $f(I) \geq \frac{3}{2}pq$, the adversary removes all other items, and we get an approximation ratio of $\frac{3}{2}$. Otherwise, if $f(I) < pq+q$, the adversary removes all items except for the one with demand q. We get at least $q(2p+1)$, while the optimal value is $q(p+2)$, which gives us an approximation ratio of 2, since both p and q can be arbitrarily large. If $f(I) \geq pq+q$, there exists some item with demand

$f(I) - pq + 1$ which intersects I. The adversary removes all items except this one. We get an approximation ratio of $\frac{2f(I)+1}{2f(I)-pq+2} > \frac{3pq+1}{2pq+2}$, as $f(I) < \frac{3}{2}pq$. This approaches to $\frac{3}{2}$ for large pq.

As in the proof of the previous theorem, the algorithm would not get a better value by assigning non-consecutive colors to any interval, and the optimal solution assigns consecutive colors to each interval. Therefore, the statement holds for both pSMC and npSMC. □

The above two inapproximation results assume an interval representation of interval graphs. A common representation of graphs is the vertex adjacency representation [20] where an input item is a vertex, its weight (if any) and a list of adjacent vertices. Under such a vertex adjacency model, we have the following two inapproximation results.

Theorem 8. *There is no adaptive priority algorithm in the vertex adjacency model for SC (and hence for pSMC and npSMC) on planar* 4*-clawfree bipartite graphs that can achieve approximation ratio better than* $\frac{11}{10}$.

Proof. We borrow the example in [23], see figure 5 below. The graph 1 on the left has 7 vertices: five vertices have degree two and two vertices have degree three. The optimal solution for graph 1 is 10 by giving color 1 to B, F, G, D and 2 to everything else. The graph 2 on the right has 7 vertices; three vertices have degree two and four vertices have degree three. The optimal solution for graph 2 is also 10 by giving color 1 to A, G, F, E and 2 to everything else. The key vertex for each graph is the vertex A.

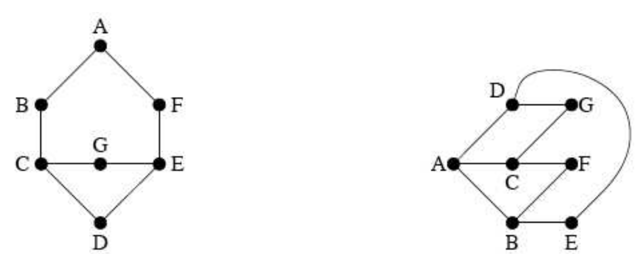


Fig. 12. Graph 1 to the left and graph 2 to the right

In the vertex adjacency model, any adaptive priority algorithm has to have an initial ordering on all possible data items. In particular, it has to rank in between vertices of degree 2 and 3. There are four cases:

– If the algorithm considers vertices of degree 2 first and is going to assign color 1 to it, then the adversary chooses graph 1 and presents vertex A to the algorithm. The solution obtained by the algorithm is at least 11.

– If the algorithm considers vertices of degree 2 first and is going to assign color other than 1 to it, then the adversary chooses graph 1 and presents vertex B to the algorithm. The solution obtained by the algorithm is at least 11.
– If the algorithm considers vertices of degree 3 first and is going to assign color 1 to it, then the adversary chooses graph 1 and presents vertex C to the algorithm. The solution obtained by the algorithm is at least 11.
– If the algorithm considers vertices of degree 3 first and is going to assign color other than 1 to it, then the adversary chooses graph 2 and presents vertex A to the algorithm. The solution obtained by the algorithm is at least 11.

In all the above cases, the algorithm cannot achieve approximation ratio better than $\frac{11}{10}$. □

Theorem 9. *There is no adaptive priority algorithm in the vertex adjacency model for pSMC or npSMC on proper interval graphs that can achieve approximation ratio better than* $\frac{11}{10}$.

Proof. We start with two graphs, solid vertices have demand 1 and hallow vertices have demand 2. The graph 1 on the left has 5 vertices, three of them have degree 2 and demand 1, two of them have degree 1 and demand 2. The graph 2 on the right has 2 vertices, both have degree 1 and demand 2. Note that there is a unique optimal solution of 10 for both pSMC and npSMC on graph 1. For

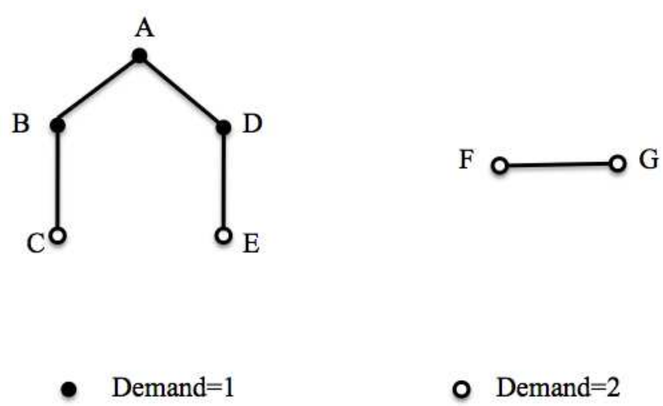


Fig. 13. Graph 1 to the left and graph 2 to the right

any adaptive priority algorithm, there are two cases:

– The algorithm first picks a vertex with demand 1 (and degree 2). If it is assigned color 1, we make it vertex A in the graph 1 above. If it is assigned

color greater than 1, we make it vertex B in the graph 1 above. In any case, the algorithm cannot obtain the unique optimal multicoloring, so it will get a sum of at least 11, resulting in an approximation of at least $\frac{11}{10}$.
– The algorithm first picks a vertex with demand 2 (and degree 1). If it is assigned any colors other than 2 and 3, we make it vertex C in the graph 1 above, resulting in an approximation ratio of at least $\frac{11}{10}$. If it is assigned the colors 2 and 3, we make it vertex F in the graph 2 above, resulting in an approximation ratio of at least $\frac{7}{6}$.

In any case, the algorithm cannot get an approximation ratio better than $\frac{11}{10}$. □

For (proper) interval graphs, a stronger priority model is to combine the two representations above; i.e., each data item is composed of a starting time, finishing time, its demand, and a list of its neighbors. We show that, even for such a more powerful "interval with vertex adjacency input" priority model, we can prove an inapproximation bound for pSMC or npSMC on proper interval graphs. This inapproximation is in contrast to the existence of an optimal priority algorithm for SC on proper interval graphs.

Theorem 10. *There is no adaptive priority algorithm in the interval with vertex adjacency model for pSMC or npSMC on proper interval graphs that can achieve approximation ratio better than* $\frac{14}{13}$.

Proof. We construct an instance with three types of intervals; see Figure 14 below. The adversary initially keeps a set of data items as shown in Figure 15.

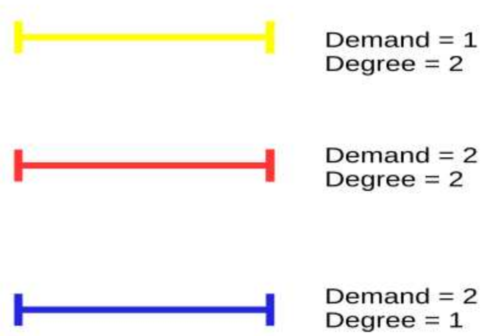


Fig. 14. Three types of data items

Note that this initial set of data items is not a valid input instance. However, the final instance constructed will be a proper interval graph. There are five cases:

1. Suppose the algorithm first selects the blue interval with start time 1.

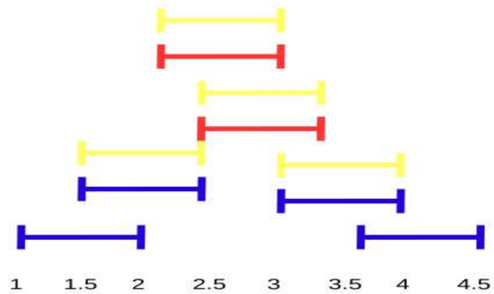


Fig. 15. The initial set of data items of the adversary

– If it is assigned any colors other than 1 and 2 or 1 and 3, the adversary presents the graph shown in Figure 16 below, making it interval A. We first focus on intervals B, D, E, F. It is not hard to verify that If E is not assigned color 1 then any sum multicoloring of B, D, E, F is at least 10. Since the multicoloring of A contains color greater or equal to 3, the sum multicoloring of the constructed graph is at least 14. Now suppose E is assigned color 1, then it is not hard to verify that any sum multicoloring (without assigning A (1,2) or (1,3)) of A, C is at least 6. Since the optimal sum multicoloring of B, D, E, F is 8, the sum multicoloring of the constructed graph is at least 14. However, the optimal solution (shown in the picture) for this graph assign colors 1 and 3 or 1 and 2 to interval A, and hence we get an approximation ratio of at least $\frac{14}{13}$.

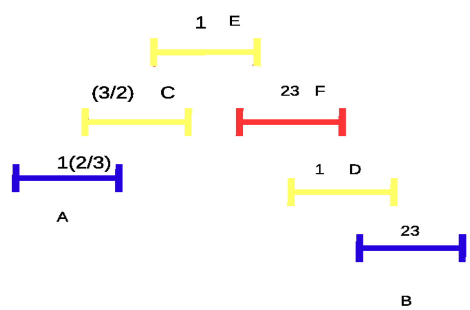


Fig. 16. The first case

– If it is assigned 1 and 2 or 1 and 3, the adversary presents the mirror image of the graph in Figure 16, making it interval B, and we again get an approximation ratio of at least $\frac{14}{13}$.

2. Suppose the algorithm first selects the yellow interval with start time 1.5.

  - If it is assigned any color other than 2 or 3, the adversary presents the graph in Figure 16, making it interval C, and we get approximation ratio of at least $\frac{14}{13}$.
  - If it is assigned the color 2 or color 3, the adversary presents the mirror image of the graph in Figure 16, making it interval D, and we get approximation ratio of at least $\frac{14}{13}$.
3. Suppose the algorithm first selects the blue interval with start time 1.5.
  - If it is assigned any color other than 2 and 3, the adversary makes it interval A in the graph shown in Figure 17. The graph has a min sum multi-coloring of 10, but if interval A is not assigned colors 2 and 3 we can get at best 11, so we get an approximation ratio of $\frac{11}{10}$.

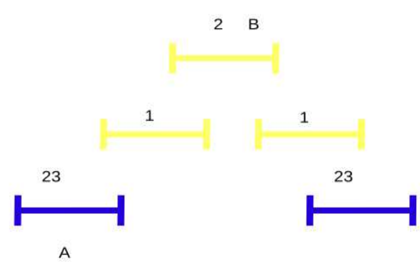


Fig. 17. The second case

  - If it is assigned colors 2 and 3, the adversary makes it interval A in the graph shown in Figure 18. The min sum multi-coloring is 12, but the lowest value we can get if interval A is assigned 2 and 3 is 14, so we get an approximation ratio of $\frac{7}{6}$.

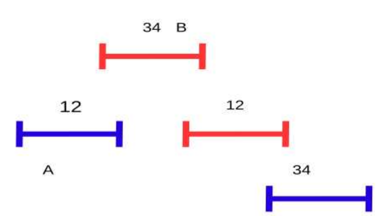


Fig. 18. The third case

4. The algorithm first selects the yellow interval with start time 2.

- If it is assigned any color other than 2, the adversary makes it interval B in the graph shown in Figure 17, resulting in an approximation ratio of at least $\frac{11}{10}$.
- If it is assigned color 2, the adversary makes it interval E in the graph shown in Figure 16, resulting in an approximation ratio of at least $\frac{14}{13}$.

5. The algorithm first selects the red interval with start time 2.
   - If it is assigned any colors other than 2 and 3, the adversary makes it interval F in the mirror image of the graph shown in Figure 16, resulting in an approximation ratio of at least $\frac{14}{13}$.
   - If it is assigned the colors 2 and 3, the adversary makes it interval B in the graph shown in Figure 18, resulting in an approximation ratio of at least $\frac{7}{6}$.

All other cases are symmetric to one of the cases discussed above. □

6 Conclusion

We have considered the sum coloring and sum multi-coloring problem for restricted families of graphs in this paper. We conclude by suggesting a few open questions:

1. The sum coloring problem can be optimally solved for proper interval graphs. Can sum multi-coloring (pSMC or npSMC) be optimally solved for proper interval graphs or are these problems NP-hard or APX-hard?
2. The best known sum coloring algorithm for chordal graphs is a 4-approximation derived from the repeated MIS approach. Can this bound be improved?
3. Is there a reduction of sum coloring to coloring in terms of approximability? Is there an APX hardness result for $(k+1)$-clawfree graphs and more generally how well can we sum color all $(k+1)$-clawfree graphs.
4. The best known sum coloring algorithm for unit disk graphs is a 5-approximation from 6-clawfreeness. This bound seems quite weak; can it be improved?

Acknowledgement

The authors wish to thank Magnús Halldórsson for his helpful comments.

References

1. J. Mycielski. Sur le coloriage des graphes. *Coll. Math.*, 3:161–162, 1955.
2. F. S. Roberts. Indifference graphs. *F. Harary (Ed.), Proof Techniques in Graph Theory*, pages 139–146, 1969.
3. Leslie G. Valiant. Universality considerations in vlsi circuits. *IEEE Trans. Computers*, 30(2):135–140, 1981.
4. E. Kubicka and A. J. Schwenk. An introduction to chromatic sums. In *CSC '89: Proceedings of the 17th conference on ACM Annual Computer Science Conference*, pages 39–45, New York, NY, USA, 1989. ACM.

5. Amotz Bar-Noy, Mihir Bellare, Magnús M. Halldórsson, Hadas Shachnai, and Tami Tamir. On chromatic sums and distributed resource allocation. *Inf. Comput.*, 140(2):183–202, 1998.
6. Amotz Bar-Noy and Guy Kortsarz. Minimum color sum of bipartite graphs. *Journal of Algorithms*, 28(2):339 – 365, 1998.
7. Uriel Feige and Joe Kilian. Zero knowledge and the chromatic number. *J. Comput. Syst. Sci.*, 57(2):187–199, 1998.
8. Kenneth P. Bogart and Douglas B. West. A short proof that "proper = unit". *Discrete Math.*, 201(1-3):21–23, 1999.
9. S. Nicoloso, Majid Sarrafzadeh, and X. Song. On the sum coloring problem on interval graphs. *Algorithmica*, 23(2):109–126, 1999.
10. Amotz Bar-Noy, Magnús M. Halldórsson, Guy Kortsarz, Ravit Salman, and Hadas Shachnai. Sum multicoloring of graphs. *Journal of Algorithms*, 37(2):422 – 450, 2000.
11. M. Gonen. Coloring problems on interval graphs and trees. *M.Sc. thesis*, 2001.
12. Krzysztof Giaro, Robert Janczewski, Marek Kubale, and Michal Malafiejski. A 27/26-approximation algorithm for the chromatic sum coloring of bipartite graphs. In *APPROX*, pages 135–145, 2002.
13. Magnús M. Halldórsson and Guy Kortsarz. Tools for multicoloring with applications to planar graphs and partial k-trees. *Journal of Algorithms*, 42(2):334 – 366, 2002.
14. Dániel Marx. The complexity of tree multicolorings. In *MFCS '02: Proceedings of the 27th International Symposium on Mathematical Foundations of Computer Science*, pages 532–542, London, UK, 2002. Springer-Verlag.
15. A. Borodin, M. Nielsen, and C. Rackoff. (Incremental) priority algorithms. *Algorithmica*, 37:295–326, 2003.
16. Magnús M. Halldórsson, Guy Kortsarz, and Hadas Shachnai. Sum coloring interval and k-claw free graphs with application to scheduling dependent jobs. *Algorithmica*, 37(3):187–209, 2003.
17. M.R. Cerioli, L. Faria, T.O. Ferreira, and F. Protti. On minimum clique partition and maximum independent set on unit disk graphs and penny graphs: complexity and approximation. *Electronic Notes in Discrete Mathematics*, 18:73 – 79, 2004. Latin-American Conference on Combinatorics, Graphs and Applications.
18. Frédéric Gardi. The Roberts characterization of proper and unit interval graphs. *Discrete Mathematics*, 307(22):2906–2908, 2007.
19. Rajiv Gandhi, Magnús M. Halldórsson, Guy Kortsarz, and Hadas Shachnai. Improved bounds for scheduling conflicting jobs with minsum criteria. *ACM Trans. Algorithms*, 4(1):1–20, 2008.
20. S. Davis and R. Impagliazzo. Models of greedy algorithms for graph problems. *Algorithmica*, 54(3), 2009. Preliminary version appeared in the Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2004.
21. Min Chih Lin, Francisco J. Soulignac, and Jayme L. Szwarcfiter. Short models for unit interval graphs. *Electronic Notes in Discrete Mathematics*, 35:247–255, 2009. LAGOS'09 - V Latin-American Algorithms, Graphs and Optimization Symposium.
22. Yuli Ye and Allan Borodin. Elimination graphs. In *ICALP '09: Proceedings of the 36th International Colloquium on Automata, Languages and Programming*, pages 774–785, Berlin, Heidelberg, 2009. Springer-Verlag.
23. Allan Borodin, Joan Boyar, Kim S. Larsen, and Nazanin Mirmohammadi. Priority algorithms for graph optimization problems. *Theor. Comput. Sci.*, 411(1):239–258, 2010.

A The priority algorithm schema

For completeness, we provide the schema for an adaptive priority algorithm introduced in Borodin, Nielsen and Rackoff [15] as a model for greedy algorithms.

Adaptive Priority
Input: A set $I = \{G_1, G_2 \ldots, G_n\}$ of items, $I \subseteq \mathcal{I}$
while not empty(I)
  Ordering: Choose, without looking at I, a total ordering $\mathcal{T}$ over $\mathcal{I}$
  next := first item in I according to ordering $\mathcal{T}$
  Decision: make a decision for item *next*
  remove *next* from I; remove from $\mathcal{I}$ any items preceding *next* in $\mathcal{T}$
end while

An algorithm is called an adaptive priority algorithm if it can be formulated using the template. Note that the algorithm has no knowledge of the iinout set $\{G_i | 1 \leq i \leq n\}$, but rather bases its choices (ii.e the "local orderings" and the irrevocable decisions) on information provided in the representation of the input item and the items previously considered (for which irrevocable decisions have already been made). For definiteness [3], i one can think of the ordering being induced by a function $f : \mathcal{I} \rightarrow \Re$ where $f(G_i)$ is then defining the priority of input item G_i.

In our application to the sum multi-coloring problem, there are several natural input models depending on the nature of the class of graphs being considered. For interval graphs, in the most basic input model, an input item G_i is an interval, represented by its end points s_i and f_i and (for the sum multi-coloring problem) the demand x_i. The irevocable decision made for an interval G_i is the set of x_i interger colors it is assigned. For arbitrary graphs, an input item G_i is a vertex, represented by its name, the demand and the names of its adjacent vertices. Once again, the irrevocable decision is the set of x_i colors assigned to the vertex. We also consider a more general input model for interval graphs, where now an interval is represented by both its end points as well as by the names of all adjacent (i.e. intersecting) intervals. For a dense interval graph, this is not a compact representation but this representation allows more algorithmic possibilities.

We also note that the irrevovable decision need not be a "greedy decision". The definition as to what constitutes a greedy decision depends on the application but loosely speaking greedy decisions are those that "live for today" in the sense of making an optimal decision as if there will be no further input items. Returning to the sum multi-coloring problem, the first fit coloring of a vertex is a greedy decision. Greedy algorithms are then modeled by priority algorithms that use greedy decisions and the more general concept of a priority algorithm

[3] This is strictly speaking a restriction on the implicit defnition of a "local ordering" as specified in the scheme. That is, based on the decisions made thus far, the algorithm can choose *any* total ordering on the set of all possible remaining input items $\mathcal{I}$. In practice this total ordering is induced by a function $f : \mathcal{I} \rightarrow \Re$.

models greedy-like or myopic algorithms. All of our inapproximation results are with respect to the more general concept of priority algorithms.