On Chromatic Sums and Distributed Resource Allocation

Amotz Bar-Noy* Mihir Bellare† Magnús M. Halldórsson‡
Hadas Shachnai§ Tami Tamir¶

*Department of Electrical Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel. Part of this work was done at IBM T.J. Watson, Yorktown Heights, NY.

†Department of Computer Science and Engineering, Mail Code 0114, University of California at San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA.

‡Science Institute, University of Iceland, IS-107 Reykjavik, Iceland. Part of this work was done at Japan Advanced Institute of Science and Technology – Hokuriku, Ishikawa 923-12, Japan.

§The Department of Computer Science, Technion, Haifa 32000, Israel. Part of this work was done while the author was at IBM T.J. Watson, Yorktown Heights, NY.

¶Israel Software Lab, Intel, Haifa, Israel.

Proposed Running Head:

A. BAR-NOY ET AL., ON CHROMATIC SUMS

Address for Proofs:

Dr. Hadas Shachnai
Department of Computer Science
The Technion IIT
Haifa 32000
Israel.

Abstract

This paper studies an optimization problem that arises in the context of distributed resource allocation: Given a conflict graph that represents the competition of processors over resources, we seek an allocation under which no two jobs with conflicting requirements are executed simultaneously. Our objective is to minimize the *average response time* of the system. In alternative formulation this is known as the *Minimum Color Sum (MCS)* problem [25].

We show, that the algorithm based on finding iteratively a maximum independent set (*MaxIS*) is a 4-approximation to the MCS. This bound is tight to within a factor of 2. We give improved ratios for the classes of bipartite, bounded-degree, and line graphs. The bound generalizes to a 4ρ-approximation of MCS for classes of graphs for which the maximum independent set problem can be approximated within a factor of ρ. On the other hand, we show that an $n^{1-\epsilon}$-approximation is NP-hard, for some $\epsilon > 0$.

For some instances of the resource allocation problem, such as the *Dining Philosophers*, an efficient solution requires *edge* coloring of the conflict graph. We introduce the *Minimum Edge Color Sum (MECS)* problem which is shown to be NP-hard. We show that a 2-approximation to MECS(G) can be obtained distributively using *compact coloring* within $O(\log^2 n)$ communication rounds.

Key words. distributed resource allocation, response time, graph coloring, maximum independent sets

List of Symbols

χ	chi
δ	delta
Δ	Delta
l	ell
μ	mu
o	oh
O	Oh
1	one
Ψ	Psi
0	zero

1 Introduction

Chromatic Sums of Graphs

Given a graph $G = (V, E)$, a *vertex coloring* is a function $\Psi : V \to \mathbf{N}$ such that adjacent vertices are assigned distinct numbers (*colors*). The Minimum Color problem is to find a vertex coloring which uses the minimum number of colors. In this paper we consider a related problem known as *Minimum Color Sum (MCS)* problem [24, 25].

> Given a graph $G = (V, E)$, find a vertex coloring $\Psi : V \to \mathbf{N}$ for G such that $\sum_{v \in V} \Psi(v)$ is minimized.

We note that the problems are not equivalent. For instance, bipartite graphs can be colored with two colors. However, for any integer k, there exist bipartite graphs (in fact, trees) for which the best MCS uses k colors [25].

In case each of the nodes $v \in V$ has a weight $w(v)$ associated with it, we refer to the *Minimum Weighted Color Sum (MWCS)* problem in which the objective is to minimize $\sum\limits_{v \in V} w(v) \cdot \Psi(v)$.

Similarly, we define the *Minimum Edge Color Sum (MECS)* problem.

> Given a graph $G = (V, E)$, find an edge coloring $\Psi : E \to \mathbf{N}$ for G (i.e. edges with common endpoints are assigned distinct colors) such that $\sum_{e \in E} \Psi(e)$ is minimized.

In case each of the edges $e \in E$ has a weight $w(e)$ associated with it, we refer to the *Minimum Weighted Edge Color Sum (MWECS)* problem in which the objective is to minimize $\sum\limits_{e \in E} w(e) \cdot \Psi(e)$.

Applications

Our main application is the problem of resource allocation with constraints imposed by *conflicting resource requirements*. In a common representation of the distributed resource allocation problem [11, 27], the constraints are given by a conflict graph G, in which the nodes represent processors, and the edges indicate competition on resources, i.e., two nodes are adjacent if the corresponding processors cannot run their jobs simultaneously. We focus on the *one shot* resource allocation problem [32, 4], in which we have to allocate resources to one batch of requests. The allocation of resources should satisfy the two following conditions:

- *Mutual exclusion*: No two conflicting jobs are executed simultaneously.
- *No starvation*: The request of any processor is eventually granted.

Our objective is to minimize the *average response time*, or equivalently to minimize the sum of the job completion times. Assuming some *fixed* execution time for the jobs, this problem is the MCS problem.

For some resource allocation problems, such as the classic *Dining Philosophers*, efficient solution requires an *edge* coloring of the conflict graph (see, e.g., [27, 28, 34]). The measure used for these problems is the *maximal waiting chain*, which is the number of colors needed to edge color the conflict graph G. For these problems, minimizing the average response time corresponds to finding the Minimum Edge Color Sum.

Further applications of the MCS problem have been studied in the contexts of compiler design and VLSI routing [29]. In a VLSI design problem, known as Over-The-Cell Routing, we are given a set of two-terminal nets and a set of parallel, horizontal tracks of distances $d = 1, 2, 3, \ldots$ from the baseline where the terminals lie. The nets are routed with a vertical connection from each terminal to the assigned track along with a horizontal connection within the track. No overlapping nets can be routed within the same track. The objective is to minimize the total wiring length, which, in addition to the fixed and pre-determined horizontal costs, equals twice the sum of the distances from the nets to the assigned tracks.

Main Results

In this paper we present the following results for the MCS problem.

- Hardness: Finding an $n^{1-\epsilon}$-approximation for the MCS is NP-hard, for some $\epsilon > 0$, where n is the number of vertices.
- The algorithm based on finding iteratively a maximum independent set (which we call *MaxIS*) is shown to provide a 4-approximation to the MCS (MWCS). This bound is tight to within a factor of 2. For a large subclass of graphs this algorithm is polynomial.
- A modified version of the MaxIS is shown to achieve a bound of $\frac{9}{8}$ to the optimum for the subclass of bipartite graphs.

For the MECS problem we derive the following results:

- The problem of finding MECS for a given graph G is NP-hard.
- A restricted version of the edge coloring problem called *compact edge coloring* is introduced. It is shown that for a given conflict graph G any compact edge coloring provides a 2-approximation to MECS(G). Compact edge coloring can be found for any graph G in time that is linear in the size of G.

Previous works on distributed resource allocation refer to the *maximal response time per processor* and thus aim at devising algorithms with good local performance (where each processor waits at most for the execution of its neighbors). Applying the above results, we quantify the behavior of these algorithms with respect to the average response time of the system. In particular, we show that any schedule which guarantees that a processor waits only if one of its conflicting neighbors executes a job, provides a $\frac{\Delta+2}{3}$-approximation to the optimal schedule, where Δ is the maximal degree in the graph. We show that for a general conflict graph, finding an optimal schedule is

NP-hard. The results in Section 2 imply that the minimal average response time is also hard to approximate.

Using the MaxIS algorithm, a 4-approximation to the minimal average response time can be found in polynomial time for a large subclass of conflict graphs, including *bipartite* graphs, *interval* graphs and *line* graphs. This implies, that the MaxIS is a 4-approximation for instances of the resource allocation problem, where we seek to minimize the edge color sum. For a conflict graph G of size n and maximal degree Δ, the MaxIS can be implemented distributively within $O(\Delta \cdot \log^2 n)$ communication rounds, by using iteratively a randomized distributed algorithm for finding a maximum matching in G (see in [21]). We show that *compact edge coloring*, that can be implemented distributively in $O(\log^2 n)$ communication rounds, yields a 2-approximation to MECS(G).

Related Work

The minimum color sum problem was introduced by Kubicka in [23]. In [25] it is shown that computing the MCS of a given graph is NP-hard. A polynomial time algorithm is given for the case where G is a tree. Jansen shows in [22] that the MCS is solvable in polynomial time for partial k-trees. In [24] it is shown that approximating MCS within an *additive* constant factor is NP-hard, and that a *first-fit* algorithm yields a $\frac{\overline{d}}{2} + 1$-approximation for graphs of average degree $\overline{d}$.

The wide literature on resource allocation problems, starting with the early work of Lynch [27], studies algorithms that minimize the *maximal response time* per processor, or alternatively – the maximal waiting chain in the system, in solutions for the *Dining Philosophers* version of the problem [2, 5, 11, 27]. In this context, the term of *one shot resource allocation problem* was coined by Rhee [32]. The one shot problem is used in his work to show that it is NP-hard to minimize the maximal response time for a static conflict graph.

Bar-Noy et al. consider in [6] the problem of scheduling *persistent* tasks with conflicting resource requirement. Since the tasks are scheduled repeatedly, the response time for a given task is the maximal time that elapses between two successive schedules of that task.

Other works related to the present context address the more general problem of scheduling under constraints. Typical examples are a predetermined partial order [13, 36] or resource constrained scheduling. In the latter case, each of the jobs is associated with a vector of requirement for resources, and jobs cannot be scheduled simultaneously, if the sum of their requirements for a specific resource exceeds the total amount of that resource (see in [16, 33]).

Outline of the Paper

The rest of this paper is organized as follows: In Section 2 we give some definitions and prove a hardness result. In Section 3 we define the MaxIS algorithm and show that MaxIS is a 4-approximation for the MCS and that this bound holds for the MWCS problem as well. In addition, we show that the MaxIS has an asymptotic lower bound of 2. In Section 3.3 we give a $\frac{9}{8}$-approximation algorithm for the case where G is a bipartite graph. In Section 4 we introduce the *compact coloring problem.* We show in Section 4.1, that if G is a line graph then any compact coloring of G is a 2-approximation to MCS(G). This result is applied in Section 4.2 to the minimum edge color sum problem. Section 5 presents the applications of the above results to the resource allocation problem. We show that for a general conflict graph, finding a schedule that minimizes the average response time is NP-hard, and that a 2-approximation to the MECS problem can be obtained by a randomized distributed algorithm in $O(\log^2 n)$ communication rounds[1].

2 Preliminaries

2.1 Definitions and Notation

For a given undirected graph $G = (V, E)$, let n denote the number of vertices, and Δ the maximum degree of the graph.

An *independent set (IS)* in G is a subset V' of V such that every vertex in V' has no neighbor in V'. A *maximal independent set* is an IS which is not contained in a strictly larger IS, and a *maximum independent set (MaxIS)* is an IS of maximum size in G.
A *c-coloring* of G is a partition of V into c independent sets. A c-coloring is specified by a mapping $\Psi : V \to \{1, \ldots, c\}$. The IS that consists of vertices with $\Psi(v) = i$, is denoted by C_i. The *chromatic number* of a graph, denoted by $\chi(G)$ is the smallest possible c for which there exists a c-coloring of G. A *c-edge-coloring* of G is a partition of E into c sets, such that no two edges in the same set share an endpoint. The *chromatic index* of a graph, denoted by $I(G)$ is the smallest possible c for which there exists a c-edge-coloring of G.

Definition 2.1 *Given a graph $G(V, E)$ and a valid coloring of G, $\Psi : V \to \mathbf{N}$, the* color sum of G with respect to Ψ *is*

$$\text{CS}(G, \Psi) = \sum_{v \in V} \Psi(v) = \sum_{i=1}^{\infty} i \cdot |C_i|$$

Definition 2.2 *The* minimum color sum *of a graph G, denoted by* MCS(G)*, is the minimum* CS(G, Ψ) *over all the legal colorings Ψ* [2]*.*

Our results extend to apply to the minimum weighted color sum problem:

Definition 2.3 *Given a graph $G(V, E)$, a weight function $W : V \to \mathbf{R}$ and a valid coloring of G, $\Psi : V \to \mathbf{N}$, the* weighted color sum of G with respect to Ψ *is*

$$\text{WCS}(G, \Psi) = \sum_{v \in V} w(v) \cdot \Psi(v)$$

The minimum weighted color sum *of a graph G, denoted by* MWCS(G)*, is the minimum* WCS(G, Ψ) *over all the legal colorings Ψ.*

[1]Remark: The current paper is a merger and extension of the two papers [7] and [18].

[2]Throughout the paper we use also the term *chromatic sum* when referring to the minimum color sum of a graph.

2.2 Hardness of the MCS

In this section we give a strong lower bound on the approximability of MCS (assuming P $\neq$ NP).

Theorem 2.1 *Suppose there exists an $f(n)$-approximate algorithm for* MCS *for a given hereditary class of graphs. Then there exists an $g(n)$-approximate algorithm for Graph Coloring on the same class of graphs, where $g(n) = O(f(n)\log n)$. If further $f(n) = \Omega(n^c)$ for some $c > 0$, then $g(n) = O(f(n))$.*

Proof: Let G be a graph and k be its chromatic number. Let A be a $f(n)$-approximate algorithm for MCS. Now, $MCS(G) \leq kn$, and A yields a coloring with a sum of at most $knf(n)$. At least half of the vertices must be colored with the first $2kf(n)$ colors. Use those $2kf(n)$ color classes, and recursively color the remaining at most $n/2$ vertices. The recursion is of depth $\log n$, and the total number of colors used is at most $2kf(n)\log n$, for a performance ratio of at most $2f(n)\log n$.

More accurately, the total number of colors used is at most

$$2k\sum_{i=0}^{\infty} f(n/2^i).$$

If $f(n) \geq n^c$, for some constant $c > 0$, then this convex sum is at most

$$2k\sum_{i=0}^{\infty} \frac{1}{(2^c)^i} f(n) \leq O(kf(n)).$$

Thus, we obtain a performance ratio of $O(f(n))$. □

Feige and Kilian have recently shown that Graph Coloring (of general graphs) is hard to approximate within $n^{1-\epsilon}$ factor [14]. We thus obtain the same hardness bound for MCS.

Corollary 2.2 MCS *cannot be approximated within $n^{1-\epsilon}$, for any $\epsilon > 0$, unless $NP = ZPP$.*

3 The MaxIS Algorithm

3.1 Upper Bound

A natural approach for solving the MCS yields the following algorithm: Iteratively, find a maximum independent set IS_i for $i \geq 1$, color IS_i with i and omit from G the nodes and edges of IS_i, until $G = \emptyset$.
We call this algorithm *MaxIS*.

Theorem 3.1 *The MaxIS algorithm is a* 4*-approximation to the* MCS.

We use in the proof the following technical lemma.

Lemma 3.2 *For any positive real numbers* $a_1, \ldots, a_n$ *and* $0 < q < 1$

$$\sum_{i=1}^{n} q^i a_i \sum_{j=1}^{i-1} a_j \leq \frac{1+3q}{2(1-q)} \sum_{i=1}^{n} q^i a_i^2 \ . \qquad (1)$$

Proof: Observe, that for any $x, y, c > 0$ and a positive integer k

$$c^{-k}x^2 - 2xy + c^k y^2 = (c^{-k/2}x - c^{k/2}y)^2 \geq 0 \ , \qquad (2)$$

or

$$xy \leq \frac{c^{-k}x^2 + c^k y^2}{2} \ ,$$

thus, for some $c > 0$, and $k = i - j - 1$

$$\sum_{j=1}^{i-1} a_i a_j \leq \frac{1}{2} \sum_{j=1}^{i-1} \left(c^{-(i-j-1)} a_i^2 + c^{i-j-1} a_j^2 \right) \qquad (3)$$

and

$$\begin{aligned}
\sum_{i=1}^{n} q^i a_i \sum_{j=1}^{i-1} a_j &\leq \frac{1}{2} \sum_{i=1}^{n} q^i \sum_{j=1}^{i-1} \left(c^{-(i-j-1)} a_i^2 + c^{i-j-1} a_j^2 \right) \\
&= \frac{1}{2} \sum_{i=1}^{n} \left(\sum_{j=1}^{i-1} c^{-(j-1)} q^i a_i^2 + \sum_{j=1}^{n-i} c^{j-1} q^{i+j} a_i^2 \right) \\
&= \frac{1}{2} \sum_{i=1}^{n} q^i a_i^2 \left(\sum_{j=0}^{i-2} c^{-j} + q \sum_{j=0}^{n-i-1} (cq)^j \right) \\
&\leq \frac{1}{2} \sum_{i=1}^{n} q^i a_i^2 \left(\frac{c}{c-1} + \frac{q}{1-cq} \right) \equiv \frac{1}{2} \sum_{i=1}^{n} q^i a_i^2 f(c) \ .
\end{aligned}$$

The first inequality follows directly from Inequality (3); the second and third equations are achieved by rearranging the summations; the last inequality uses infinite summation as upper bound for the given sums. The value c^* which minimizes $f(c)$ is $\frac{q+1}{2q}$. Substituting c^* into the last inequality gives Inequality (1). □

Proof of Theorem 3.1: Let $V = \{V_1, \ldots, V_k\}$ be the partition of the nodes by their colors $1, \ldots, k$ using some optimal coloring. We obtain the upper bound by dividing each of the optimal sets V_j, $1 \leq j \leq k$, to smaller subsets V_{ji}, $1 \leq i \leq L$, for some $L \geq 1$. Under the optimal coloring the nodes in V_{ji} are covered at the j-th iteration, therefore the cost incurred by the i-th strip, defined as $\cup_{j=1}^{k} V_{ji}$ is $\sum_{j=1}^{k} j|V_{ji}|$.

Consider the operation of the MaxIS algorithm on G. Number the nodes in V_j, $1 \leq j \leq k$, by $1, \ldots, |V_j|$, such that their colors with respect to the MaxIS algorithm are in nondecreasing order.

Then for some $m \geq k$ and $1 \leq i \leq \log|V_1| \equiv L$, we call *strip* i and denote it by $l_i(m)$ the subset of nodes numbered with the indices $\lceil(\frac{m-1}{m})^i|V_1|\rceil, \ldots, \lfloor(\frac{m-1}{m})^{i-1}|V_1|\rfloor$. For $1 \leq j \leq k$, the number of nodes in strip i which belong to V_j and denoted by $l_i(m,j)$, satisfies

$$|l_i(m,j)| \leq |V_1| \left[\left(\frac{m-1}{m}\right)^{i-1} - \left(\frac{m-1}{m}\right)^{i}\right] = |V_1|\frac{(m-1)^{i-1}}{m^i} \tag{4}$$

Let $c_i(m)$ be the number of sets among $\{V_1, \ldots, V_k\}$ which contain nodes in $l_i(m)$, then

$$|l_i(m)| = \sum_{j=1}^{k} |l_i(m,j)| \leq \frac{(m-1)^{i-1}}{m^i}|V_1|c_i(m)\ . \tag{5}$$

Assuming that $l_i(m)$ is full, i.e., equality in the RHS of (5), we increase the size of each of the optimal sets by at most a factor of $\frac{m}{m-1}$. We note that the optimal coloring uses $c_i(m)$ colors for covering strip i, and thus the cost incurred by that strip is

$$|V_1|\frac{(m-1)^{i-1}}{m^i}\sum_{j=1}^{c_i(m)} j = |V_1|\frac{(m-1)^{i-1}}{m^i} \cdot \frac{c_i(m)(c_i(m)+1)}{2}\ . \tag{6}$$

Hence, the optimal color sum satisfies:

$$\mathrm{MCS}(G) \geq \frac{m-1}{m}|V_1|\sum_{i=1}^{L}\binom{c_i(m)+1}{2}\frac{(m-1)^{i-1}}{m^i}\ . \tag{7}$$

In obtaining an upper bound for CS(MaxIS) we use the following two claims:

Claim 3.3 *MaxIS starts coloring* $l_i(m)$ *after using at most* $\left\lfloor \sum_{j=1}^{i-1} \frac{c_j(m)}{m-1} \right\rfloor$ *colors.*

Proof: By induction on i.
Basis: The claim holds for $i = 1$ (with empty sum equals to 0).
Induction step: Assume that the claim holds for i, then we observe, that while the nodes in $l_i(m)$ were not fully covered, MaxIS finds in G an IS of size at least $\left\lceil \left(\frac{m-1}{m}\right)^i |V_1| \right\rceil$. Hence, the amount of colors used until $l_i(m)$ is fully covered is bounded by

$$\frac{\sum_{j=1}^{i-1} c_j(m)}{m-1} + \frac{l_i(m)}{\left\lceil \left(\frac{m-1}{m}\right)^i |V_1| \right\rceil} \leq \frac{\sum_{j=1}^{i} c_j(m)}{m-1}\ , \tag{8}$$

and at most $\left\lfloor \sum_{j=1}^{i} \frac{c_j(m)}{m-1} \right\rfloor$ colors precede the first color used for covering nodes in $l_{i+1}(m)$. $\square$

Claim 3.4 *The average cost for coloring a node in $l_i(m)$ is bounded by* $\left\lfloor \frac{\sum_{j=1}^{i-1} c_j(m)}{m-1} \right\rfloor + \frac{\left\lceil \frac{c_i(m)}{m-1} \right\rceil + 1}{2}$.

Proof: Observe, that the upper bound on the cost of coloring $l_i(m)$ is obtained for the case where $\left\lceil \left(\frac{m-1}{m}\right)^i |V_1| \right\rceil$ divides $l_i(m)$, and MaxIS covers in each iteration exactly $\left\lceil \left(\frac{m-1}{m}\right)^i |V_1| \right\rceil$ nodes in $l_i(m)$. The number of such iterations is $\left\lceil \frac{c_i(m)}{m-1} \right\rceil$. Thus, the cost incurred by $l_i(m)$ is bounded by

$$\sum_{k=1}^{\left\lceil \frac{c_i(m)}{m-1} \right\rceil} \left\lceil \left(\frac{m-1}{m}\right)^i |V_1| \right\rceil \cdot \left(\left\lfloor \frac{\sum_{j=1}^{i-1} c_j(m)}{m-1} \right\rfloor + k \right)$$

$$= \left\lceil \frac{c_i(m)}{m-1} \right\rceil \cdot \left\lceil \left(\frac{m-1}{m}\right)^i |V_1| \right\rceil \cdot \left(\left\lfloor \frac{\sum_{j=1}^{i-1} c_j(m)}{m-1} \right\rfloor + \frac{\left\lceil \frac{c_i(m)}{m-1} \right\rceil + 1}{2} \right)$$

$$\leq l_i(m) \cdot \left(\left\lfloor \frac{\sum_{j=1}^{i-1} c_j(m)}{m-1} \right\rfloor + \frac{\left\lceil \frac{c_i(m)}{m-1} \right\rceil + 1}{2} \right)$$

and the claim follows. □

From Inequality (5) and Claims 3.3, 3.4,

$$\begin{aligned} \mathrm{CS}(MaxIS) &\leq |V_1| \sum_{i=1}^{L} c_i(m) \frac{(m-1)^{i-1}}{m^i} \left(\frac{\left\lceil \frac{c_i(m)}{m-1} \right\rceil + 1}{2} + \left\lfloor \frac{\sum_{j=1}^{i-1} c_j(m)}{m-1} \right\rfloor \right) \\ &\leq |V_1| \sum_{i=1}^{L} c_i(m) \frac{(m-1)^{i-1}}{m^i} \left(\frac{c_i(m)}{2(m-1)} + \frac{\sum_{j=1}^{i-1} c_j(m)}{m-1} + 1 \right) . \end{aligned}$$

Setting $q = (m-1)/m$, $n = L$, and $a_i = c_i(m)$ for $1 \leq i \leq L$ in (1), we have

$$\sum_{i=1}^{L} \frac{(m-1)^i}{m^i} c_i(m) \sum_{j=1}^{i-1} c_j(m) \leq \frac{4m-3}{2} \sum_{i=1}^{L} c_i^2(m) \frac{(m-1)^i}{m^i} . \tag{9}$$

Using inequality (7) we have

$$\begin{aligned} \mathrm{CS}(MaxIS) &\leq |V_1| \sum_{i=1}^{L} \frac{(m-1)^{i-1}}{m^i} \left(\frac{4m-2}{2} c_i^2(m) + c_i(m) \right) \\ &\leq \frac{4m-2}{m-1} \mathrm{MCS}(G) - \frac{m}{m-1} |V_1| \sum_{i=1}^{L} c_i(m) . \end{aligned}$$

Hence,

$$\frac{\mathrm{CS}(MaxIS)}{\mathrm{MCS}(G)} \leq 4 + O(\frac{1}{m}) \ . \tag{10}$$

□

For the general case, where the independent set algorithm is not exact but finds a ρ-approximate solution, we note, that the average cost for coloring a node in $l_i(m)$ is bounded by $\rho\left(\left\lfloor \frac{\sum_{j=1}^{i-1} c_j(m)}{m-1} \right\rfloor + \frac{\left\lceil \frac{c_i(m)}{m-1} \right\rceil + 1}{2}\right)$. Thus we have

Corollary 3.5 *When using a ρ-approximate independent set algorithm, the MaxIS algorithm is a 4ρ-approximation to the* MCS*.*

This immediately gives us a fairly good characterization of the approximability of MCS on various classes of graphs: $O(n/\log^2 n)$ on general graphs [10], $O(\Delta \log\log\Delta/\log\Delta)$ on graphs of maximum degree Δ [35], $O(n^{.2134})$ on 3-colorable graphs [8], and at most 4 on all perfect graphs and partial k-trees, among others.
We show below, that the bound in Theorem 3.1 applies also to the MWCS problem. In that case, the weighted MaxIS algorithm (*W_MaxIS*) chooses iteratively an IS with *maximum weight* in G.

Theorem 3.6 *The W_MaxIS algorithm is a 4-approximation to the* MWCS*.*

Proof: Assuming first integer weights, the proof follows the steps of the proof of Theorem 3.1, except that we replace the set V_j, $j = 1, \ldots, k$ with a set V_j' of $W(V_j)$ nodes, where $W(V_j) = \sum_{v \in V_j} w(v)$, such that for any $v \in V_j'$ $w(v) = 1$. Thus, each node in V_j is represented by $w(v)$ nodes in V_j'. We define a partition of the sets V_j' to the subsets V_{ji}', $1 \leq i \leq L$, with $L \equiv \lg W(V_1)$. Applying the MaxIS algorithm to G, each node in V_j' gets the color of the node $v \in V_j$ to which it belongs. Then we number the nodes in V_j' in such a way that their colors are in nondecreasing order, and define $l_i(m)$ as the subset of nodes numbered with indices $\lceil (\frac{m-1}{m})^i W(V_1) \rceil, \ldots, \lfloor (\frac{m-1}{m})^{i-1} W(V_1) \rfloor$. Replacing in each step of the proof $|V_1|$ with $W(V_1)$ we have the statement of the theorem.
The proof can be easily extended to apply to non-integral weights. Thus we omit the details. □

3.2 Lower Bound

In this subsection we show that there exist graphs for which MaxIS is at least 2-approximation to the MCS. We construct a family of k-partite graphs G_k for $k \geq 2$, such that MaxIS is a $\frac{2k}{k+1} - o(1)$-approximation to MCS(G_k).

Intuitively, we construct a balanced k-partite graph that recursively has the property that the largest independent set contains equally many vertices from each partition. Hence, MaxIS colors n/k vertices with the first color, $n(k-1)/k^2$ with the second color, $n(k-1)^2/k^3$ with the third color, and so on. The total cost sums up to be nk, while the cost of the balanced k-coloring is $n(k+1)/2$, for a ratio of $2 - O(1/k)$.

Throughout our analysis we use the following known equalities:

$$\sum_{i=0}^{n} \left(\frac{k-1}{k}\right)^{i} = k - \frac{(k-1)^{n+1}}{k^{n}} \ ,$$

and

$$\sum_{i=1}^{n} i \left(\frac{k-1}{k}\right)^{i-1} = k^2 - (n+k)\frac{(k-1)^{n}}{k^{n-1}} \ .$$

We describe first the construction for $k = 2$: Let G_2 be a bipartite graph with two large independent sets A and B of the same size. The edges between A and B will be chosen in a way that will force MaxIS to pick x vertices from both A and B (i.e., $2x > |A|$). In the second stage, MaxIS will pick y vertices from the remains of A and B. (i.e., $2y > |A| - x$). This process continues until MaxIS picks all the vertices of G_2.

More formally, we define the graph G_2^m as follows. The vertices are composed of two independent sets A and B each of size $2^m - 1$. Let $A = A_0 \cup A_1 \cdots \cup A_{m-1}$ and $B = B_0 \cup B_1 \cdots \cup B_{m-1}$ such that $|A_i| = |B_i| = 2^i$ for $0 \le i \le m-1$. Indeed,

$$\sum_{i=0}^{m-1} |A_i| = \sum_{i=0}^{m-1} 2^i = 2^m - 1 = |A| \ .$$

The edges of G_2^m are all the possible edges between A and B except those edges from A_i to B_i for $0 \le i \le m-1$.

Since G_2^m is a bipartite graph we can color A with 1 and B with 2 and it follows that

$$MCS(G_2^m) \le 3(2^m - 1) \ .$$

We now compute the cost of MaxIS. For $0 \le i \le m-1$, let $D_i = A_i \cup B_i$. The set D_i is independent and $|D_i| = 2^{i+1}$. The largest IS of G_2^m is D_{m-1}. By induction one can verify that the largest IS of $G_2^m - \{D_{m-1} \cup \cdots \cup D_{i+1}\}$ is D_i. Therefore, MaxIS colors the set D_i with color $m-i$ for $0 \le i \le m-1$. Consequently,

$$CS(MaxIS) = \sum_{i=1}^{m} i 2^{m+1-i} = 4(2^m - 1) - 2m \ .$$

It follows that for G_2^m,

$$\frac{CS(MaxIS)}{\mathrm{MCS}(G_2^m)} \ge \frac{4}{3} - \frac{2m}{3(2^m - 1)} \ ,$$

and for large m we get

$$\frac{CS(MaxIS)}{\mathrm{MCS}(G_2^m)} \ge \frac{4}{3} - o(1) \ .$$

We now generalize the above construction to $k \geq 2$: Informally, G_k will be a k-partite graph with k large independent sets $A^1, \ldots, A^k$ of the same size. The edges between the k independent sets will be chosen in a way that will force MaxIS to pick x vertices from each of them (i.e., $kx > |A^1|$). In the second stage, MaxIS will pick y vertices from the remain of each of the sets (i.e., $ky > |A^1| - x$). This process will continue until MaxIS picked all the vertices of G_k.

In the following formal description we choose $|A^1|$, x, and y in a way that facilitates the analysis. However, even the best choice cannot produce a better lower bound than $\frac{2k}{k+1}$. We define the graph G_k^m as follows. The vertices are composed of k independent sets $A^1, \ldots, A^k$ each of size $k^m - (k-1)^m$. Let $A^j = A_0^j \cup A_1^j \cdots \cup A_{m-1}^j$ for $1 \leq j \leq k$ such that $|A_i^j| = (k-1)^{m-1-i}k^i$ for $1 \leq j \leq k$ and $0 \leq i \leq m-1$. We note, that

$$\sum_{i=0}^{m-1} |A_i^j| = \sum_{i=0}^{m-1} (k-1)^{m-1-i}k^i = k^m - (k-1)^m = |A^j| \ .$$

The edges of G_2^m are all the possible edges between the A^js except those edges between the corresponding subsets, i.e., between A_i^j and $A_i^{j'}$ for $1 \leq j \neq j' \leq k$ and $0 \leq i \leq m-1$.

Since G_k^m is a k-partite graph we can color A^j with j for $1 \leq j \leq k$ and it follows that

$$MCS(G_k^m) \leq \frac{k(k+1)}{2}(k^m - (k-1)^m) \ .$$

We now compute the cost of MaxIS. For $0 \leq i \leq m-1$, let $D_i = A_i^1 \cup \cdots \cup A_i^k$. The set D_i is independent and $|D_i| = (k-1)^{m-1-i}k^{i+1}$. The largest IS of G_k^m is D_{m-1}. It can be shown inductively, that the largest IS of $G_k^m - \{D_{m-1} \cup \cdots \cup D_{i+1}\}$ is D_i. Therefore, MaxIS colors the set D_i with color $m-i$ for $0 \leq i \leq m-1$. Consequently,

$$CS(MaxIS) = \sum_{i=1}^{m} i(k-1)^{i-1}k^{m+1-i} = k^2(k^m - (k-1)^m) - km(k-1)^m \ .$$

It follows that for G_k^m,

$$\frac{CS(MaxIS)}{\mathrm{MCS}(G_k^m)} \geq \frac{2k}{k+1} - \frac{2m(k-1)^m}{(k+1)(k^m - (k-1)^m)} \ ,$$

and for large m

$$\frac{CS(MaxIS)}{\mathrm{MCS}(G_k^m)} \geq \frac{2k}{k+1} - o(1) \ .$$

3.3 Approximating the Chromatic Sum for Bipartite Graphs

In the following we describe an algorithm that achieves a ratio of $\frac{9}{8}$ for the MCS of bipartite graphs. That is, given a bipartite graph G, the algorithm generates a coloring Ψ such that $\mathrm{CS}(G, \Psi) \leq \frac{9}{8} \cdot \mathrm{MCS}(G)$.

The algorithm colors the graph in two ways, and then chooses the coloring with the smaller sum. One coloring is any two-coloring. The other coloring colors a maximum independent set with the first color, and then two-colors the remaining vertices. Note that a maximum independent set of a bipartite graph can be found in polynomial time by computing a maximum matching [15].

Theorem 3.7 *The above algorithm achieves a ratio of $\frac{9}{8}$ to the MCS for any bipartite graph.*

Proof: Let α be the size of the maximum independent set of the graph. The cost of our former coloring is at most $3n/2$ and the latter coloring is at most $\alpha + (n-\alpha)\cdot 5/2 = 5n/2 - 3\alpha/2$. The cost of the optimal coloring is at least $\alpha + 2(n-\alpha) = 2n - \alpha$. Hence, the ratio is at most

$$\min\{\frac{3n/2}{2n-\alpha}, \frac{5n/2-3\alpha/2}{2n-\alpha}\} \;=\; 1+\min\{\frac{\alpha-n/2}{2n-\alpha}, \frac{n/2-\alpha/2}{2n-\alpha}\}$$

which is maximized when $\alpha - n/2 = n/2 - \alpha/2$ or $\alpha = 2n/3$, in which case the ratio is $9/8$. □

4 Using Compact Coloring to Approximate the Chromatic Sum

A coloring $\Psi : V \to \{1\ldots k\}$ is *compact* if $C_i = \{v \in V \,|\, \Psi(v) = i\}$ comprises a maximal independent set in $G \setminus \bigcup_{j<i} C_j$, for every $1 \le i \le k$. This definition provides a simple *greedy* polynomial time algorithm for compact coloring of any graph G. The algorithm consists of at most $\Delta+1$ phases: In phase i we color with i the subset of vertices C_i, that is a maximal independent set in $G \setminus \bigcup_{j<i} C_j$. Indeed, Theorem 3.1 implies that the above algorithm produces a 4Δ-approximation to MCS(G), since the size of a MaxIS in G is at most Δ times the size of any maximal independent set in G. We derive below a tighter bound of $\frac{\Delta+2}{3}$.
The following observation, which gives an alternative definition for compact coloring, can be easily verified.

Lemma 4.1 *A coloring Ψ is compact if and only if every vertex v with $\Psi(v) = i$ has a neighbor u with $\Psi(u) = j$ for all $1 \le j \le i-1$.*

This suggests an alternative formulation of the greedy algorithm, often referred to as *first-fit*: Process the vertices in an arbitrary order and assign a vertex to the smallest color with which none of its preceding neighbors have been colored. This method has the advantage of being *on-line*, processing resource requests as they arrive.

The following general upper bound on the chromatic sum has been observed several times in the past. Let m denote the number of edges in the graph.

Lemma 4.2 ([9, 24]) *The sum of any compact coloring is at most $m+n$.*

This bound is tight for disjoint collection of cliques. It can be attained by a parallel algorithm [17].

Theorem 4.3 *Any compact coloring of a graph $G = (V, E)$ provides a $\frac{\Delta+2}{3}$-approximation to* MCS(G)*, and that is tight.*

Proof: All edges have at least one endpoint outside the first color class of the optimal solution. Thus, when maximum degree is bounded by Δ, there are at least $\lceil m/\Delta \rceil$ vertices outside the first color class. That is, we have:

$$MCS(G) \ \geq \ n + m/\Delta \tag{11}$$

Thus, by Lemma 4.2, the performance ratio of a compact coloring is at most

$$\frac{m+n}{n+m/\Delta} = \frac{\overline{d}/2+1}{1+\overline{d}/(2\Delta)}.$$

This is maximized at $\overline{d} = \Delta$ (i.e., when G is regular with degree Δ), for a ratio of $(\Delta+2)/3$.

This ratio is tight for the graph $B_{p,p}$ formed by a complete bipartite graph from which a single bipartite matching has been removed. Namely, the graph contains vertex set $\{v_1, \ldots v_p, u_1, \ldots u_p\}$ and the edge set $\{(v_i, u_j) \mid 1 \leq i, j \leq p, i \neq j\}$. One compact coloring contains p classes with 2 vertices each, for a cost of $2\binom{p}{2} = p(p+1)$ versus an optimal coloring of cost $3p$, for a ratio of $(p+1)/3 = (\Delta+2)/3$. $\square$

4.1 Compact Coloring of Line Graphs

While for general graphs we have the approximation ratio $\frac{\Delta+2}{3}$, we show below that for the subclass of *line graphs*, compact coloring is a 2-approximation to the chromatic sum.

Given a graph $G = (V, E)$, the *line graph* of G, denoted by $L(G)$ is the intersection graph of E: The vertices in $L(G)$ are the edges of G. Two vertices in $L(G)$ are adjacent whenever the corresponding edges in G are. We say that G is a line graph, if there exists some graph G', such that $G = L(G')$.

The following property of line graphs is used in the proof of the next theorem:

Property 4.4 *[19] If $G = (V, E)$ is a line graph, then E can be partitioned into cliques, such that each vertex belongs to at most two cliques.*

Theorem 4.5 *If G is a line graph, then any compact coloring of G is a 2-approximation to* MCS(G)*.*

Proof: We prove a stronger ratio of $2 - 4/(\overline{d}+4)$, which follows from the combination of Lemma 4.2 and the following lemma.

Lemma 4.6 *For a line graph G, $MCS(G) \geq (m+2n)/2$.*

Proof: Let $Q_1, Q_2, \ldots Q_l$ be the clique partition of G, with q_i denoting the size of each clique. Extend the partition so that each vertex appears exactly twice, by adding singleton cliques for those vertices that appeared only once. Let Q denote the set of all $2n$ pairs (i, v) where v is contained in clique Q_i.

We define a *clique labeling* to be an assignment of positive integers to the pairs of Q such that, for each Q_i and each distinct u, w in Q_i, (i, v) and (i, w) have different labels. The *cost* of a clique labeling is the sum of the labels. Let $CL(G)$ denote the optimal clique labeling of line graph G. The minimum cost clique labeling has the labels involving a given clique Q_i arranged to be exactly the first q_i positive integers. Hence,

$$CL(G) = \sum_i \binom{q_i + 1}{2} = \sum_i \binom{q_i}{2} + q_i = \sum_i |E(Q_i)| + |V(Q_i)| = m + 2n. \tag{12}$$

Intuitively, we have a labeling of the vertices, where each vertex may receive two labels, one for each of its cliques. An ordinary vertex coloring can easily be extended to a clique labeling by doubling each label. Thus, the optimal chromatic sum is at least half the cost of an optimal clique labeling, i.e. $CL(G) \leq 2 \cdot MCS(G)$. The lemma now follows from (12). □

□

4.2 The Minimum Edge Color Sum Problem

We now introduce the minimum edge color sum problem:

Definition 4.1 *Given a graph $G(V, E)$ with a valid edge coloring $\Psi : E \to \mathbf{N}$, let B_i denote the set of edges $e \in E$, with $\Psi(e) = i$. The* edge color sum of G with respect to Ψ *is*

$$\mathrm{ECS}(G, \Psi) = \sum_{e \in E} \Psi(e) = \sum_{i=1}^{\infty} i \cdot |B_i|$$

Definition 4.2 *The* minimum edge color sum *of a graph G, denoted by* $\mathrm{MECS}(G)$, *is the minimum* $\mathrm{ECS}(G, \Psi)$ *over all legal edge colorings Ψ.*

In this section we show that the minimum edge color sum problem is NP-hard, and that the results in the previous sections imply the existence of polynomial time algorithms for approximating $\mathrm{MECS}(G)$ to within a constant factor.

Theorem 4.7 *The problem of finding* $\mathrm{MECS}(G)$ *for a given graph G is NP-hard.*

Proof: The proof is by reduction from the Chromatic Index [15]:
Given a graph $G = (V, E)$ and the question: 'Is $I(G) = \Delta$?', denote by S the following subset of edges in E:

$$S = \{e_i = (v_i, u_i) \mid d(v_i) + d(u_i) \geq \Delta\} \ .$$

We number the edges in S with $1, \ldots, |S|$ and construct the extended graphs $G'_1, \ldots, G'_{|S|}$. In G'_i we replace the edge i with $|E|^2 + 1$ parallel edges. Choosing $w = \frac{|E|^4}{2} + |E|^2(\Delta + \frac{1}{2}) + \frac{|E|(\Delta+1)}{2}$, it can be verified that

1. If $I(G) = \Delta$ then, for every i, $1 \leq i \leq |S|$, $\mathrm{MECS}(G'_i) \leq w$.
2. If $I(G) = \Delta + 1$ then, there exists some i, $1 \leq i \leq |S|$, such that $\mathrm{MECS}(G'_i) > w$.

By a theorem of Vizing, $I(G)$ is always either Δ or $\Delta + 1$. □

Definition 4.3 *An edge coloring* $\Psi : E \to \{1, \ldots, k\}$ *is* compact *if and only if every edge* e *with* $\Psi(e) = j$*, has neighboring edges with all colors* $1, \ldots, j - 1$.

Theorem 4.8 *Any compact edge coloring of a graph* G *is a* 2*-approximation to* $\mathrm{MECS}(G)$.

Proof: Let $\Psi : E \to \{1, \ldots, k\}$ be some compact edge coloring of a graph G. Let $G' = (V', E')$ be the line graph of G. Then Ψ induces a compact vertex coloring $\Psi' : V' \to \{1, \ldots, k\}$ on G'. By Theorem 4.5 any compact coloring of G' is a 2-approximation to $\mathrm{MCS}(G') = \mathrm{MECS}(G)$. □

5 Application to the Resource Allocation Problem

The *resource allocation* problem was introduced by Chandy and Misra [11] as the abstracted drinking philosophers problem. An instance of the resource allocation problem is a *resource allocation graph* G. The vertices represent processors, and there is an edge between any pair of processors that may compete on some resource. The requirements of processors for resources may vary over time. The current requirements are represented by a dynamic *conflict graph* C, where the vertices are processors currently waiting to execute their jobs, and there is an edge between two processors that compete on some resource. Clearly $C \subseteq G$. An algorithm for the resource allocation problem is called *a scheduler*. We denote by μ the maximum execution time of any job.

Any scheduler needs to satisfy the properties of safety and liveness mentioned above.
The total time that elapses from a processor's request for resources until it can execute its job is regarded as the *response time* for that processor. We seek solutions for the problem that minimize the average response time of the system.

Theorem 5.1 *It is NP-hard to find a schedule for the resource allocation problem that achieves the minimum average response time.*

Proof: We use in the proof a special instance of the resource allocation problem.

Definition 5.1 *The following is the* one-shot resource allocation problem*:*
Input: A set of k jobs $J_1, \ldots, J_k$ with the execution times $\mu_1, \ldots, \mu_k$ respectively and the corresponding conflict graph C.
Output: A legal schedule for $J_1, \ldots, J_k$ satisfying the safety and liveness properties.

Definition 5.2 *Given a set of jobs $J_1, \ldots, J_k$ with the execution times $1 \leq \mu_i \leq \mu \; \forall \, 1 \leq i \leq k$,* a slow execution of the resource allocation problem *is an execution in which all the processors use the resources for exactly μ time units.*

We show that every sequential algorithm which finds the optimal schedule, can be used to find in polynomial time the chromatic sum of a graph.
Given a graph G, an integer k and the question 'Is MCS$(G) < k$?', construct a conflict graph $C = G$ and apply the optimal scheduling algorithm on a slow execution of the one-shot resource allocation problem.

Claim 5.2 MCS$(G) < k$ *if and only if the average response time for C is less than $(\frac{k}{N} - 1)\mu$.*

Proof: As the scheduler is optimal, there is no delay between executions of two successive jobs of competing processors. Combining that with the fact that all execution times equal to μ implies that the schedule is a partition of the conflicting processors into non-overlapping execution sets. The members of the first set execute their jobs exactly at the interval $[0, \mu)$, the members of the second set execute their jobs exactly on the interval $[\mu, 2\mu)$ etc.

For a given optimal execution, associate with each processor P_i a label $L(P_i)$, such that $L(P_i) = c$ if and only if P_i executes its job in the interval $[(c-1)\mu, c\mu)$. The mutual exclusion property yields that L is a coloring of G. In addition, for every $P_i \in C$, response-time$(P_i) = (c-1)\mu$ if and only if $L(P_i) = c$. Thus, CS$(G, L) < k$ if and only if $\sum_{P_i \in C}$ response-time $(P_i) < (k - N)\mu$, or the average response time is less than $(\frac{k}{N} - 1)\mu$. □

The above reduction is clearly polynomial, and since it is NP-hard to determine the chromatic sum of a given graph [25], the problem of scheduling the jobs so as to minimize the average response time is NP-hard. □

In applying the results in Sections 3 and 4 to the resource allocation problem, we model a distributed network as a communication graph G where each vertex represents a processor and there is a bidirectional communication link connecting every pair of adjacent processors. We assume a synchronous system which operates in rounds. Thus, a message sent at round k from processor p_i to a neighboring processor p_j arrives to p_j at round $k+1$. Messages may be of arbitrary length and local computation is instantaneous and unlimited. We assume that processors have unique numerical id's.
Some resource allocation algorithms [12] use a preprocessing which results in a legal coloring of

the communication graph. The color of a processor indicates the maximal length of a waiting chain for this processor. Achieving small number of colors in the preprocessing guarantees small maximal response time. The same holds when minimizing the *average* response time, if we replace the original preprocessing with a preprocessing that minimizes the color sum of the graph. That is, during the preprocessing each processor picks a label for itself. The entire labeling of G is a legal coloring that approximates the chromatic sum of G. As stated above, a compact coloring of G provides a $\frac{\Delta+2}{3}$-approximation to MCS(G).

Theorem 5.3 *[4] For any graph G, a compact coloring of G can be found distributively within $O(\log^2 n)$ communication rounds.*

Hence, for the one shot resource allocation problem we have

Corollary 5.4 *If the execution time of any job is μ then a schedule which approximates the minimal average response time within a factor of $\frac{\Delta+2}{3}$ can be found distributively in $O(\log^2 n)$ communication rounds.*

For the subclass of conflict graphs for which a maximum independent set can be found in polynomial time, the minimum average response time can be approximated to within a factor of 4 using the MaxIS algorithm.

Some known resource allocation algorithms [27, 34] conduct a preprocessing in which an *edge* coloring of the communication graph is found. In these algorithms, the response time for a processor depends on the colors of its neighboring edges. A preprocessing that finds a coloring that minimizes the edge color sum, would yield a resource allocation algorithm that achieves small average response time. Using iteratively a randomized distributed algorithm for finding a maximum matching in the conflict graph G (see in [21]), the MaxIS algorithm can be implemented to yield a 4-approximation within $O(\Delta \cdot \log^2 n)$ communication rounds. By a reduction from compact coloring, the algorithm presented in [4] can be used to obtain distributively a 2-approximation to MECS(G) within $O(\log^2 n)$ communication rounds.

Acknowledgments

We are indebted to Jaikumar Radhakrishnan for his important contributions, including the precursor to Theorem 3.1. We would like to thank Hagit Attiya for many stimulating discussions on this paper, and Sara Nicoloso for introducing to us various applications of the chromatic sum problem. The anonymous referee provided many helpful comments that improved the presentation of our results.

References

[1] Arora, S., Lund, C., Motwani, R., Sudan, M., and Szegedy, M. (1992), Proof verification and hardness of approximation problems, in *Proceedings of the Thirty Third Annual IEEE Symposium on Foundations of Computer Science*, pp. 14–23.

[2] Awerbuch, B., and Saks, M. (1990), A Dining Philosophers Algorithm with Polynomial Response Time, in *Proceedings of the Thirty First IEEE Symp. on Foundation of Computer Science*, pp. 65–74.

[3] Awerbuch, B., Cowen, and Smith (1994), Efficient Asynchronous Distributed Symmetry Breaking, in *Proceedings of the Twenty Sixth on the Theory of Computing*, pp. 214-223.

[4] Attiya, H., Shachnai, H., and Tamir, T. (1994), Local Labeling and Resource Allocation Using Preprocessing, in *Proceedings of the 8th International Workshop on Distributed Algorithms*, pp. 194–208.

[5] Bar-Ilan, J. and Peleg, D. (1992), Distributed Resource Allocation Algorithms, in *Proceedings of the Sixth International Workshop on Distributed Algorithms*, pp. 276–291.

[6] Bar-Noy, A., Mayer, A., Schieber, B., and Sudan, M. (1995), Guaranteeing Fair Service to Persistent Dependent Tasks, in *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 243–252.

[7] Bar-Noy, A., Shachnai, H., and Tamir, T. (1996), On Chromatic Sums and Distributed Resource Allocation, in *Proceedings of the Fourth Israel Symposium on Theory and Computing Systems.*

[8] Blum, A., and, Karger, D. An $O(n^{0.2143})$ coloring for 3-colorable graphs, *Inf. Proc. Let.*, to appear.

[9] Boppana, R. B. (1989), Optimal separations between concurrent-write parallel machines, in *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pp. 320–326.

[10] Boppana, R. B. and Halldórsson, M. M. (1992), Approximating maximum independent sets by excluding subgraphs, *BIT*, **32**, pp. 180–196.

[11] Chandy, K., and Misra, J. (1984), The Drinking Philosophers Problem, *ACM Trans. Programming Languages and Systems*, **6**, pp. 632–646.

[12] Choy, M., and Singh, K. (1992), Efficient Fault Tolerant Algorithms in Distributed Systems, in *Proceedings of the Twenty Forth Annual Symposium on the Theory of Computing*, pp. 593–602.

[13] Cheng, S.C., Stankovic, J.A., and Ramamritham, K. (1987), Hard Real-Time Systems, IEEE Computer Society Press, pp. 150–173.

[14] Feige, U. and Kilian, J. (1996), Zero-knowledge and the Chromatic Number, in *Proceedings of the IEEE Conference on Structure in Complexity Theory.*

[15] Garey, M.R. and Johnson, D.S. (1979), Computers and intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman.

[16] Garey, M.R. and Graham, R. (1975), Bounds for Multiprocessor Scheduling with Resource Constraints, *SIAM J. on Computing*, **4**, pp. 187–200.

[17] Goldberg, M. and Spencer, T. H. (1990), An efficient parallel algorithm that finds independent sets of guaranteed size, in *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 219–225.

[18] Halldórsson, M.M., and Radhakrishnan, J. (1993), Approximating the Chromatic Sum. *Japan Advanced Institute of Science and Technology (JAIST), Research Report*, IS-RR-93-0002F.

[19] Harary, F. (1969), Graph Theory. Addison-Wesley.

[20] Irani, S. and Leung, V. (1996), Scheduling with Conflicts, and Applications to Traffic Signal Control, in *Proceedings of the Seventh ACM-SIAM Symposium on Discrete Algorithms*, pp. 85–94.

[21] JáJá, J. (1992), An Introduction to Parallel Algorithms. Addison-Wesley.

[22] Jansen, K. (1997), The Optimum Cost Chromatic Partition Problem. *Proc. of the Third Italian Conference on Algorithms and Complexity (CIAC '97)*, LNCS 1203.

[23] Kubicka, E. (1989), The Chromatic Sum of a Graph, PhD thesis, Western Michigan University.

[24] Kubicka, E., Kubicki, G. and Kountanis, D. (1989), Approximation Algorithms for the Chromatic Sum, in *Proceedings of the First Great Lakes Computer Science Conf.*, Springer LNCS 507, pp. 15–21.

[25] Kubicka, E. and Schwenk, A.J. (1989), An Introduction to Chromatic Sums, in *Proceedings of the ACM Computer Science Conf.*, pp. 39-45.

[26] Lund, C. and Yannakakis, M. (1993), On the hardness of approximating minimization problems, in *Proceedings of the Twenty Fifth Annual ACM Symposium on Theory of Computing.*

[27] Lynch, N. (1981), Upper Bounds for Static Resource Allocation in a Distributed System, *J. of Computer and System Sciences*, **23**, pp. 254–278.

[28] Naor, M. and Stockmeyer, L. (1993), What Can be Computed Locally?, in *Proceedings of the Twenty Fifth Annual Symposium on the Theory of Computing*, pp. 184–193.

[29] Nicoloso, S., Sarrafzadeh, M. and Song, X. (1994), On the Wire-Length Minimization Problem, Manuscript.

[30] Nicoloso, S., Sarrafzadeh, M. and Song, X. (1994), On the Sum Coloring Problem on Interval Graphs, Istituto di Analisi dei Sistemi ed Informatica (IASI-CNR), R. 390.

[31] Panconesi, A. and Srinivasan, A. (1992), Improved Distributed Algorithms for Coloring and Network Decomposition problems, in *Proceedings of the Twenty Fourth Annual Symposium on the Theory of Computing*, pp. 581–592.

[32] Rhee, I. (1994), Efficiency of Partial Synchrony, and Resource Allocation in Distributed Systems. PhD thesis, University of North Carolina at Chapel Hill.

[33] Srivastav, A. and Stangier, P. (1994), Tight Approximations for Resource Constrained Scheduling Problems, in *Proceedings of the Second European Symposium on Algorithms*, pp. 307–318.

[34] Steyer, E. and Peterson, G. (1988), Improved Algorithms for Distributed Resource Allocation, in *Proceedings of the Seventh Annual Symposium on Principles of Distributed Computing* pp. 105–116, 1988.

[35] Vishwanathan, S. (1995), Private communication.

[36] Wang, Q. and Cheng, K.H. (1992), A Heuristic of Scheduling Parallel Tasks and its Analysis, *SIAM J. on Computing*, **21**, pp. 281–294.