Graph Coloring in Optimization Revisited

Assefaw Hadish Gebremedhin * Fredrik Manne Alex Pothen†

Abstract

We revisit the role of graph coloring in modeling a variety of matrix partitioning problems that arise in numerical determination of large sparse Jacobian and Hessian matrices. The problems considered in this paper correspond to the various scenarios under which a matrix computation, or estimation, may be carried out, i.e., the particular problem depends on whether the matrix to be computed is symmetric or nonsymmetric, whether a one-dimensional or a two-dimensional partition is to be used, whether a direct or a substitution based evaluation scheme is to be employed, and whether all nonzero entries of the matrix or only a subset need to be computed. The resulting complex partitioning problems are studied within a unified graph theoretic framework where each problem is formulated as a variant of a coloring problem. Our study integrates existing coloring formulations with new ones. As far as we know, the estimation of a subset of the nonzero entries of a matrix is investigated for the first time. The insight gained from the unified graph theoretic treatment is used to develop and analyze several new heuristic algorithms.

Key words: Sparsity, symmetry, Jacobians, Hessians, finite differences, automatic differentiation, matrix partitioning problems, graph coloring problems, NP-completeness, approximation algorithms

*Department of Informatics, University of Bergen, N-5020 Bergen, Norway. {assefaw, fredrikm}@ii.uib.no

†Department of Computer Science, Old Dominion University, Norfolk, VA 23529-0162 USA. pothen@cs.odu.edu and ICASE, NASA Langley Research Center, Hampton, VA 23681-2199 USA. pothen@icase.edu

1 Introduction

Algorithms for solving nonlinear systems of equations and numerical optimization problems that rely on derivative information require the repeated estimation of Jacobian and Hessian matrices. Since this usually constitutes an expensive part of the entire computation, efficient methods for estimating these matrices via finite difference (FD) or automatic differentiation (AD) techniques are needed. It should be noted that FD techniques find an approximation whereas AD techniques enable exact (within the limits of machine precision) computation. For brevity, we use the term 'estimation' in referring to both cases. In applying AD and FD techniques, if the sparsity structure of the desired matrix is known a priori, or can be determined easily, the nonzero entries can be estimated efficiently. The objective in such an efficient estimation is to minimize the number of function evaluations or AD passes required.

This objective calls for a variety of *matrix partitioning problems*. The particular problem depends on whether the required matrix is symmetric or nonsymmetric, whether a one-dimensional partition (involving only columns or rows) or a two-dimensional partition (involving both columns and rows) is used, whether the entries are evaluated using a direct method or via substitution, and finally, whether all the nonzero entries of the matrix or only a subset of them need to be determined.

Several studies have demonstrated the usefulness of *graph coloring* in modeling the matrix partitioning problems of our concern [4, 7, 8, 9, 17, 18, 23]. However, these studies have been rather disintegrated: a typical study in this field focuses on one type of matrix, a specific numerical method, and a particular evaluation scheme. This has at least two consequences. First, the inherent similarity among the various partitioning problems gets obscured. Second, it makes the identification of a generic formulation difficult thereby hindering the development of algorithms and software in a flexible manner.

The main purpose of this paper is to study the various matrix partitioning problems within a *unified graph theoretic framework*. We consider eight different partitioning problems. For each matrix partitioning problem, we develop an equivalent *graph coloring formulation*. To our knowledge, the problems in partial matrix estimation, where a specified *subset* of the nonzero entries is to be determined, are studied for the first time. In full matrix estimation, the case in which all the nonzero entries are to be determined, we use known coloring formulations for all problems except for the estimation of a nonsymmetric matrix using a one-dimensional partition via a direct method. For this problem, we propose *distance-2 graph coloring* as

Matrix	1D Partition	2D Partition	Method
Jacobian	distance-2 coloring	distance-$\frac{3}{2}$ bicoloring	Direct
Hessian	distance-$\frac{3}{2}$ coloring	NA	Direct
Jacobian	NA	acyclic bicoloring	Substitution
Hessian	acyclic coloring	NA	Substitution

Table 1: Graph coloring formulations for estimating *all* nonzero entries of derivative matrices. The Jacobian and the Hessian are represented by their bipartite and adjacency graphs, respectively. NA stands for not applicable.

an alternative formulation to the known distance-1 coloring formulation [7]. The motivation for the distinction between full and partial matrix estimation is the fact that the computation in the latter case can be carried out even more efficiently.

Table 1 summarizes the coloring formulations used in this paper. The formulations for one-dimensional estimation of the Hessian are due to Coleman and Moré [8] and Coleman and Cai [4] and those for two-dimensional estimation of the Jacobian are due to Coleman and Verma [9].

In our work, for all the matrix partitioning problems, we rely on a *bipartite* graph representation for a nonsymmetric matrix and an *adjacency* graph representation for a symmetric matrix. We show that these representations are robust and flexible—they are decoupled from the eventual technique to be employed and the matrix entries to be computed. This is in contrast with the *column intersection* graph representation [7] of a nonsymmetric matrix which targets at determining *all* the nonzero entries of the matrix using a one-dimensional column partition. Moreover, the space required for storing the column intersection graph of matrix A is proportional to the number of non-zeros in A^TA, whereas the space required for storing the bipartite graph of A is proportional to the number of non-zeros in A.

Using our graph representations, we identify the distance-2 graph coloring problem as a unifying generic model for the various one-dimensional matrix partitioning problems, i.e., the coloring problem in a particular case is some relaxed variant of the distance-2 coloring problem. In the case of two-dimensional partitioning problems, we build upon the known relationship to graph *bicoloring* and observe a connection to finding a *vertex cover* in a graph.

All of the problems listed in Table 1 are known to be NP-hard [4, 8, 9, 21]. We use the insight gained from the interrelationship among these problems to develop several simple heuristic algorithms for finding sub-optimal solutions.

The rest of this paper is organized as follows. Section 2 is a detailed introduction to the partitioning problems that arise in full matrix estimation. In Section 3 we rigorously develop the equivalent graph problem formulations and discuss their interrelationship. Section 4 deals with the graph theoretic formulation of partitioning problems in partial matrix estimation. In Section 5 we compare formulations based on bipartite graph with formulations based on column intersection graph. In Section 6 we present and analyze various greedy heuristic algorithms. We conclude the paper in Section 7 with some remarks and point out avenues for further work.

2 Background

2.1 Finite Differences and Partitioning Problems

Given a continuously differentiable function $F : R^n \rightarrow R^m$, the **Jacobian** of F at the point x is the $m \times n$ matrix whose (i, j) entry $J(x)_{ij} = F'(x)_{ij} = \frac{\partial f_i}{\partial x_j}(x)$, where $f_1(x), f_2(x), \ldots, f_m(x)$ are the components of $F(x)$. Let A denote the Jacobian matrix $F'(x)$. An estimate for the jth column of A, denoted henceforth by a_j, can be obtained from the *forward difference* approximation,

$$Ae_j = a_j = \frac{\partial}{\partial x_j}F(x) \approx \frac{1}{h}[F(x + he_j) - F(x)], \quad 1 \leq j \leq n, \tag{1}$$

where e_j is the jth unit vector and h is a positive step length. Other finite difference approximations of higher order, such as *central differences*, could also be used to estimate A. In any case, if $F(x)$ is already evaluated, an approximation to a_j is obtained with one additional function evaluation. Thus, if each column of A is computed independently, n additional function evaluations will be required. However, by exploiting the sparsity structure of A, the required number of function evaluations can be reduced significantly. The sparsity structure of A is often easily available and the goal here is to exploit this to estimate the nonzero entries of A using as few function evaluations as possible under the assumption that evaluating $F(x)$ is more efficient than evaluating the components $f_i(x)$, $1 \leq i \leq m$, separately.

Given a twice continuously differentiable function $f : R^n \rightarrow R$, the **Hessian** of f at the point x is the $n \times n$ symmetric matrix whose (i, j) entry $H(x)_{ij} = \nabla^2 f(x)_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$. When ∇f is available, $\nabla^2 f$ can be approximated by applying Formula (1) to the function $F = \nabla f$. Again, we assume that evaluating the gradient $\nabla f(x)$ as a single entity is more desirable than evaluating the components $\partial_1 f(x), \ldots, \partial_n f(x)$ separately.

Let d_i be the binary vector obtained by adding some unit vectors e_j, $j \in \{1, 2, \ldots, n\}$, together. The problem of estimating a sparse derivative matrix, Jacobian or Hessian, using FD can then be stated as follows. Given the sparsity structure of a matrix A find vectors $d_1, d_2, \ldots, d_p$ such that the products $Ad_1, Ad_2, \ldots, Ad_p$ enable the determination of all the nonzero entries of A.

Specifying the vectors $Ad_1, Ad_2, \ldots, Ad_p$ gives rise to a system of linear equations where the unknowns are the nonzero elements of A. If the choice of the vectors d_i is such that the resulting system of equations can be ordered to a diagonal form, then we say that A is *directly* determined by the vectors d_i. If, on the other hand, the vectors d_i are chosen such that the system of equations can be ordered to a triangular form, then the unknowns can be determined via *substitution*.

In both a direct and a substitution based determination, minimizing the number of function evaluations corresponds to minimizing the number of vectors p. There is a trade-off in the choice of methods. A direct method is more restrictive and hence requires more function evaluations compared to a substitution method. On the other hand, a substitution method is subject to numerical instability, whereas a direct method is not. Moreover, in terms of parallel computation, direct methods offer straightforward parallelization since the estimates can be read off directly from each row of a matrix-vector product, whereas substitution methods have less parallelism since there are more dependencies among the computations required to obtain the matrix entries. In the next two paragraphs (Sections 2.1.1 and 2.1.2), we consider minimizing p in a direct and substitution based evaluation, respectively.

2.1.1 Direct estimation

In a direct determination of a matrix A, note that for each nonzero element a_{ij}, there is a vector d in the set $\{d_1, d_2, \ldots, d_p\}$ such that $a_{ij} = (Ad)_i$, where $(Ad)_i$ is the ith component of the vector Ad. Thus, each nonzero matrix element a_{ij} can be read off from some component of the vector Ad.

The problems that arise in the direct, efficient estimation of sparse Jacobian and Hessian matrices can thus be stated as follows.

Problem 2.1 *Given the sparsity structure of a general matrix $A \in R^{m \times n}$, find the fewest vectors $d_1, d_2, \ldots, d_p$ such that $Ad_1, Ad_2, \ldots, Ad_p$ determine A directly.*

Problem 2.2 *Given the sparsity structure of a symmetric matrix $A \in R^{n \times n}$, find the fewest vectors $d_1, d_2, \ldots, d_p$ such that $Ad_1, Ad_2, \ldots, Ad_p$ determine A directly.*

Curtis et al. [12] were the first to observe that, while using a direct method, a group of columns can be determined by one evaluation of Ad if no two columns in this group have a nonzero in the same row position. Such columns are *structurally orthogonal*, since their pairwise inner products are zero. Powell and Toint [26] later showed that, in the case of Hessian estimation, the number of function evaluations can be reduced further by considering symmetry. The following two notions define the underlying *partitions* used in these methods [8].

Definition 2.3 A partition of the columns of a matrix A is said to be *consistent* with a direct determination of A if whenever a_{ij} is a nonzero element of A then the group containing a_j has no other column with a nonzero in row i.

Definition 2.4 A partition of the columns of a symmetric matrix A is *symmetrically consistent* with a direct determination of A if whenever a_{ij} is a nonzero element of A then either the group containing a_j has no other column with a nonzero in row i, or the group containing a_i has no other column with a nonzero in row j.

Let $\{C_1, C_2, \ldots, C_p\}$ be a consistent partition. With each group C_k, associate a binary vector d_k having components $\delta_j = 1$ if a_j belongs to C_k, and $\delta_j = 0$ otherwise. Then,

$$Ad_k = \sum_{a_j \in C_k} a_j.$$

If $a_{ij} \neq 0$ and column $a_j \in C_k$, then $a_{ij} = (Ad_k)_i$. Thus, all the nonzero entries of A can be determined with p evaluations of Ad_k.

When A is symmetric, a symmetrically consistent partition is sufficient to determine A directly: if a_j is the only column in its group with a nonzero in row i then a_{ij} can be determined as discussed above; alternatively, if a_i is the only column in its group with a nonzero in row j then a_{ji} can be determined.

If a consistent partition (rather than a symmetrically consistent one) is used to compute a symmetric matrix A, the estimate for a_{ij} may actually be different from that of a_{ji} due to truncation error. Thus, using a symmetrically consistent partition to compute half of the nonzero elements of a matrix and determining the other half by symmetry is preferable both in terms of reducing computational work and ensuring that the computed matrix is indeed symmetric.

Using Definitions 2.3 and 2.4, Problems 2.1 and 2.2 can be restated as follows. In the remainder of this paper, we shall use the acronym MPP to refer to a matrix partitioning problem.

Problem 2.5 (*MPP1*) *Given the sparsity structure of a matrix $A \in R^{m \times n}$, find a consistent partition of the columns of A with the fewest number of groups.*

Problem 2.6 (*MPP2*) *Given the sparsity structure of a symmetric matrix $A \in R^{n \times n}$, find a symmetrically consistent partition of the columns of A with the fewest number of groups.*

2.1.2 Estimation via substitution

In estimating a matrix A via a substitution method, the vectors $d_1, \ldots, d_p$ are chosen such that the system of equations defined by the products $Ad_1, \ldots, Ad_p$ allows the determination of the unknowns via a substitution process. A partition suitable for a substitution method needs to fulfill more relaxed requirements compared to a direct method and hence results in smaller number of groups. In the FD context, this fact has been especially used when estimating a symmetric matrix since substitution can be effectively combined with the exploitation of symmetry [4]. For a nonsymmetric matrix, the advantage offered by a substitution method over a direct method is not so pronounced. An example of a substitution method for a nonsymmetric matrix can be found in [18].

In this paper, we concentrate on using a substitution method for a symmetric matrix. To illustrate the fact that a partition used in a substitution method requires fewer groups than that used in a direct method, consider the 4×4 symmetric matrix H shown in Figure 1.

$$\begin{bmatrix} \mathrm{x} & \mathrm{x} & & \\ \mathrm{x} & \mathrm{x} & \mathrm{x} & \\ & \mathrm{x} & \mathrm{x} & \mathrm{x} \\ & & \mathrm{x} & \mathrm{x} \end{bmatrix}$$

Figure 1: The structure of a 4×4 symmetric matrix

For matrix H, any partition consistent with a direct determination requires at least three groups. One such partition is $\{(h_1, h_4), (h_2), (h_3)\}$, where h_j is the jth column of H. However, if we do not insist on determining the elements directly, two groups would suffice. For example $\{(h_1, h_3), (h_2, h_4)\}$. The two matrix-vector products corresponding to the two groups yield a system of eight equations involving the nonzero entries of H. Note that, due to symmetry, nonzero element h_{ij} can be identified with h_{ji}, and hence, there are effectively seven unknowns in the system. This system can be ordered to a triangular form and be solved via substitution.

In general, a partition of the columns of a symmetric matrix induces a substitution method if there is an ordering of the matrix unknowns such that all unknowns can be solved for, in that order, using symmetry and previously solved elements.

We now formally define such a partition and then state the corresponding partitioning problem.

Definition 2.7 A partition of the columns of a symmetric matrix A is said to be *substitutable* if there exists an ordering on the elements of A such that for every nonzero a_{ij}, either a_j is in a group where all the nonzeros in row i, from other columns in the same group, are ordered before a_{ij} or a_i is in a group where all the nonzeros in row j, from other columns in the same group, are ordered before a_{ij}.

Problem 2.8 (*MPP3*) *Given the sparsity structure of a symmetric matrix* $A \in R^{n \times n}$*, find a substitutable partition of the columns of* A *with the fewest number of groups.*

2.2 Automatic Differentiation and Partitioning Problems

Automatic differentiation (AD) is a chain rule based technique for evaluating the derivatives of functions defined by computer programs. The two basic modes of operation of AD, known as *forward* and *reverse* mode, correspond to a bottom-up and a top-down strategy of accumulating partial derivatives of elementary functions that define the computational scheme of the function to be differentiated. A treatment of the technical details of AD is beyond the scope of this paper, but the interested reader is referred to, for instance, the books [14] and [10].

What is of interest for us is that, as in the FD setting, the efficient computation of matrices using AD gives rise to partitioning problems in which structural orthogonality continues to be the partition-criterion. In particular, one can use the forward mode to compute a group of columns of a matrix A from the product Ad_1, where the vector d_1 has nonzeros in positions corresponding to columns of A that are structurally orthogonal. Furthermore, in the reverse mode, a group of structurally orthogonal rows of A can be computed from the vector-matrix product ${d_2}^T A$, where d_2 is an appropriately defined vector. This means that one can potentially take advantage of the sparsity available in columns and in rows.

One way of exploiting sparsity in columns and rows is to *separately* partition the columns and rows of the matrix and use the partition which gives the minimum number of groups. For a symmetric matrix, a row partition is equivalent to a column partition, but for a nonsymmetric matrix, the two partitions may differ considerably. For example, consider an $n \times n$ matrix where all the entries on the diagonal and the first row are nonzero, and the rest of the matrix entries are all zero. For such a matrix structure, a column partition requires n groups whereas a row partition requires just two groups.

However, an approach based on a separate row and column partition is not always satisfactory. For example, consider an $n \times n$ matrix where all of the elements in the first row, first column, and the diagonal are nonzero and the rest of the entries are all zero. For such a structure, a row partition requires n groups and so does a column partition. However, using a *combined* row and column partition, three groups are enough to determine all the nonzero entries of the matrix. First, separately evaluate the entries in the first column and the first row (two groups). Then, since the remaining $(n-1) \times (n-1)$ matrix is diagonal, determine all entries by one evaluation. Thus, three groups (two column and one row) suffice to determine all the nonzero entries.

In Sections 2.2.1 and 2.2.2 we consider such a computation of a nonsymmetric matrix using the combined modes of AD via direct and substitution methods. We call a partition that involves both rows and columns a *two-dimensional* partition as opposed to a *one-dimensional* partition in which either only rows or only columns are involved. Note that a two-dimensional partition does not make sense for computing a symmetric matrix. In particular, a symmetry-exploiting one-dimensional partition is sufficient.

2.2.1 Direct computation

Consider the vectors $d_1, d_2, \ldots, d_p$ in Problem 2.1 as the p columns of the $n \times p$ matrix D. An alternative way of posing the problem would then be: given the structure of a matrix $A \in R^{m \times n}$, find a matrix $D \in R^{n \times p}$ with the least value of p such that the product AD determines A directly. By the same token, the problem that arises in the two-dimensional efficient direct computation of a Jacobian can be posed as follows.

Problem 2.9 *Given the sparsity structure of the matrix $A \in R^{m \times n}$, find matrices $D_1 \in R^{n \times p_1}$ and $D_2 \in R^{m \times p_2}$ such that AD_1 and $D_2^T A$ together determine A directly and the value $p = p_1 + p_2$ is minimized.*

Hossain and Steihaug [17] studied Problem 2.9 and reformulated it as a partitioning problem by using the notion of *consistent row-column partition* in which the *entire* set of rows and columns is partitioned into two respective set of groups. Coleman and Verma [9] also studied the same problem and identified a similar two-dimensional partition problem. Their notion of partition differs from that of Hossain and Steihaug in that it partitions only a *subset* of the columns and the rows of the matrix that suffice for the direct determination of the entries. The following concepts were used to formalize the requirements.

Definition 2.10 A *bipartition* of a matrix A is a pair (Π_C, Π_R) where Π_C is a column partition of a subset of the columns of A and Π_R is a row partition of a subset of the rows of A.

Definition 2.11 A bipartition (Π_C, Π_R) of a matrix A is *consistent with a direct determination* if for every nonzero entry a_{ij} of A, either column j is in a group of Π_C which has no other column having a nonzero in row i, or row i is in a group of Π_R which has no other row having a nonzero in column j.

The number of column and row groups in a consistent bipartition corresponds to the number of forward and reverse AD passes, respectively, required to compute the nonzero entries directly. To see this, observe that a nonzero a_{ij} can be determined either from a column group where column j is the only column with a nonzero in row i, or from a row group where row i is the only row with a nonzero in column j. Hence, assuming that the computational costs involved in the forward and reverse modes of AD are of the same order, in an efficient method, the value $|\Pi_C| + |\Pi_R|$ is required to be as small as possible[1].

Thus Problem 2.9 can be reformulated as follows.

Problem 2.12 (*MPP4*) *Given the sparsity structure of a matrix* $A \in R^{m \times n}$*, find a bipartition* (Π_C, Π_R) *of* A *consistent with a direct determination such that* $|\Pi_C| + |\Pi_R|$ *is minimized.*

2.2.2 Computation via Substitution

In using a substitution method in the AD context, the requirement on the bipartition can be relaxed so as to obtain fewer number of groups compared

[1] In this paper, we are concerned only with computational cost; however, in general, the forward mode requires less memory space than the reverse mode.

Matrix	1D Partition	2D Partition	Method
Jacobian	MPP1	MPP4	Direct
Hessian	MPP2	NA	Direct
Jacobian	NA	MPP5	Substitution
Hessian	MPP3	NA	Substitution

Table 2: Partitioning problems in estimating/computing *all* nonzero entries of derivative matrices using FD and AD. The entry NA denotes that the case is not applicable.

with that in a direct method. We state the following definition used by Coleman and Verma [9] to subsequently give the fifth matrix partitioning problem of our concern.

Definition 2.13 A bipartition (Π_C, Π_R) of a matrix A is *consistent with a determination by substitution* if there exists an ordering on the elements of A such that for every nonzero entry a_{ij}, either column j is in a group where all nonzeros in row i, from other columns in the group, are ordered before a_{ij} or row i is in a group where all the nonzeros in column j, from other rows in the group, are ordered before a_{ij}.

Problem 2.14 (*MPP5*) *Given the sparsity structure of a matrix* $A \in R^{m \times n}$*, find a bipartition* (Π_C, Π_R) *of* A *consistent with a determination by substitution such that* $|\Pi_C| + |\Pi_R|$ *is minimized.*

Problems MPP1 through MPP5 are summarized in Table 2. Note that the problems in Table 2 are formulated independent of the numerical technique used. For example, MPP1 could arise in using FD or only the forward mode of AD.

2.3 Other Methods

The matrix estimation methods considered in this paper rely on a one-dimensional or a two-dimensional *partition* (symmetrically) consistent with a direct or a substitution-based determination. However, approaches where this is not necessarily the case have also been suggested. For instance, direct methods that allow structurally non-orthogonal columns to reside in the same group and/or allow columns to reside in several groups have been suggested [8, 25, 26]. McCormick [23] gives a classification of direct methods for estimating a symmetric matrix, including those that do not necessarily rely on consistent partitions.

In another direction, Hossain [16] suggests a technique for a direct estimation of a Jacobian in which the rows are first grouped into blocks that define 'segmented' columns and then the segments are partitioned into groups each of which consists of structurally independent segments. It is shown that, for some matrix structures, such an approach may reduce the number of function evaluations compared with an approach that does not use segmentation.

3 Graph Formulations

Problems MPP1 through MPP5 are combinatorial and several previous studies have demonstrated the usefulness of graph theoretic approaches in analyzing and solving them [4, 7, 8, 9, 17, 23]. In this section, we integrate these approaches within a unified framework. In addition, we propose a new, more flexible, graph formulation for problem MPP1. The graph formulation of each matrix problem depends on the choice of the graph used to represent the matrix structure. In Section 3.2 we describe our graph representations which will be used to develop the equivalent graph problems in Sections 3.3 through 3.5. In Section 3.6 we show how the various graph problems obtained relate with one another. We start this section by defining some basic graph theoretic concepts. Other graph concepts will be defined later as required.

3.1 Basic Definitions

A *graph* G is an ordered pair (V, E) where V is a finite and nonempty set of *vertices* and E is a set of unordered pairs of distinct vertices called *edges*. If $(u, v) \in E$, vertices u and v are said to be *adjacent*; otherwise they are called *non-adjacent*. A *path* of *length* l in a graph is a sequence $v_1, v_2, \ldots, v_{l+1}$ of distinct vertices such that v_i is adjacent to v_{i+1}, for $1 \leq i \leq l$. Two distinct vertices are said to be *distance-k neighbors* if the shortest path connecting them has length *at most* k; otherwise they are called *non-distance-k neighbors*. The number of distance-k neighbors of a vertex u is referred to as the *degree-k* of u.

In a graph $G = (V, E)$, a set of vertices $C \subseteq V$ is said to *cover* a set of edges $F \subseteq E$ if for every edge $e \in F$, at least one of the endpoints of e is in C. If the set C covers the entire E, it is called a *vertex cover*. The set of vertices $I \subseteq V$ is called an *independent set* if no pair of vertices in I are adjacent to each other.

A graph $G = (V, E)$ is *bipartite* if its vertex set V can be partitioned into two disjoint sets V_1 and V_2 such that every edge in E connects a vertex from V_1 to a vertex from V_2. We denote a bipartite graph by $G_b = (V_1, V_2, E)$.

3.2 Representing Matrix Structures Using Graphs

Bipartite graph Let A be an $m \times n$ matrix with rows $r_1, r_2, \ldots, r_m$ and columns $a_1, a_2, \ldots, a_n$. We define the bipartite graph $G_b(A)$ of A as $G_b(A) = (V_1, V_2, E)$ where $V_1 = \{r_1, r_2, \ldots, r_m\}$, $V_2 = \{a_1, a_2, \ldots, a_n\}$, and $(r_i, a_j) \in E$ whenever a_{ij} is a nonzero element of A, for $1 \le i \le m$, $1 \le j \le n$.

Note that $G_b(A)$ is a space-efficient representation for a nonsymmetric matrix A. To see this, notice that the number of vertices $|V_1| + |V_2| = m + n$, and the number of edges $|E| = nnz(A)$, where $nnz(A)$ is the number of nonzeros in A. Also, note that the graph can be constructed by reading off the entries of the matrix without any further computation.

Adjacency graph Let $A \in R^{n\times n}$ be a symmetric matrix with nonzero diagonal elements and let its columns be $a_1, a_2, \ldots, a_n$. The adjacency graph of A is $G(A) = (V, E)$ where $V = \{a_1, a_2, \ldots, a_n\}$, and $(a_i, a_j) \in E$ whenever a_{ij} is a nonzero element of A, for $1 \le i \le n$, $1 \le j \le n$, $i \ne j$. Note that $G(A)$ is a space-efficient, symmetry-exploiting, graph representation of the symmetric matrix A with no explicit representation for the edges corresponding to the nonzero diagonal elements. In particular, the number of vertices is n and the number of edges is $\frac{1}{2}(nnz(A) - n)$. This is in contrast to $2n$ and $nnz(A)$, respectively, had a bipartite graph representation been used.

For the symmetric matrix of our interest, the Hessian matrix, the assumption that the diagonals are nonzero is reasonable in many contexts. In particular, the Hessian is usually positive definite [4].

To simplify notation, we shall use a_i both when referring to the ith column of the matrix A and the corresponding vertex in $G(A)$ or $G_b(A)$.

Column intersection graph Our graph formulations for problems MPP1 through MPP5 are based on the bipartite and adjacency graph representations discussed above. However, in the literature, other graph representations have also been used. In particular, Coleman and Moré [7] used the column intersection graph $G_c(A)$ to represent a nonsymmetric matrix A. In $G_c(A) = (V, E)$, the columns of A constitute the vertex set V, and an edge (a_i, a_j) is in E whenever columns a_i and a_j have nonzero entries at the same row position, i.e. whenever a_i and a_j are not structurally orthogonal.

3.3 Distance-k Graph Coloring

A *distance-k p-coloring*, or (k,p)-coloring for short, of a graph $G=(V,E)$ is a mapping $\phi: V \to \{1,2,\dots,p\}$ such that $\phi(u) \neq \phi(v)$ whenever u and v are distance-k neighbors. The minimum possible value of p in a (k,p)-coloring of a graph G is called its *k-chromatic number*, and is denoted by $\chi_k(G)$. A (k,p)-coloring of $G=(V,E)$ is called *partial* if it involves only a subset of the vertices; in particular, a partial (k,p)-coloring of $G=(V,E)$ on W, $W \subset V$, is a mapping $\phi: W \to \{1,2,\dots,p\}$ such that $\phi(u) \neq \phi(v)$ whenever u and v are distance-k neighbors.

The *distance-k graph coloring problem* (DkGCP) asks for an optimal (k,p)-coloring of a graph: given a graph G and an integer k, find a (k,p)-coloring of G such that p is minimized.

Notice that a (k,p)-coloring of $G=(V,E)$ partitions the set V into p groups (called *color classes*) $U_1, U_2, \dots, U_p$, where $U_i = \{u \in V : \phi(u) = i\}$. Each color class is a *distance-k independent set*, i.e., no pair of distinct vertices consists of distance-k neighbors. Noting this, a natural question arises—can the matrix partitioning problems MPP1 through MPP5 be related to the DkGCP for some value of k?

Recall that structural orthogonality is the criterion used in a consistent partition of the columns (or rows) of a matrix. The following two simple observations provide the graph theoretic equivalents of structural orthogonality in matrices.

Lemma 3.1 *Let $A \in R^{m\times n}$ be a matrix and $G_b(A) = (V_1, V_2, E)$ be its bipartite graph. Two columns (or rows) of A are structurally orthogonal if and only if the corresponding vertices in $G_b(A)$ are non-distance-2 neighbors.*

Proof: We prove the statement for columns; a similar argument can be used to prove the case for rows. Assume that vertices a_i and a_j in V_2 are non-distance-2 neighbors. Thus, by definition, there is no path a_i, r_k, a_j in G_b for any $r_k \in V_1$, $1 \leq k \leq m$. In terms of matrix A, this means that there is no $k \in [1,\ m]$ such that both a_{ki} and a_{kj} are nonzero. Hence, by definition, a_i and a_j are structurally orthogonal.

To prove the 'only if' part of the statement, assume that columns a_i and a_j are structurally orthogonal. Then, by definition, there is no $k \in [1,\ m]$ such that $a_{ki} \neq 0$ and $a_{kj} \neq 0$. This implies that there is no path a_i, r_k, a_j in $G_b(A)$, for any $1 \leq k \leq m$. Hence, by definition, a_i and a_j are non-distance-2 neighbors. □

Similarly, one can prove the following statement for the case of a symmetric matrix and its adjacency graph representation.

Lemma 3.2 *Let $A \in R^{n \times n}$ be a symmetric matrix with nonzero diagonal elements and let $G(A) = (V, E)$ be its adjacency graph. Two columns in A are structurally orthogonal if and only if the corresponding vertices in $G(A)$ are non-distance-2 neighbors.*

Lemmas 3.1 and 3.2 provide a partial answer to our question regarding the relationship between the matrix partitioning problems and the DkGCP. As discussed in the forthcoming sections, distance-2 coloring is a generic model in efficient derivative matrix estimation using methods that rely on one-dimensional column or row partition.

3.4 Coloring Problems in Direct Methods

In this subsection we consider problems MPP1, MPP2, and MPP4.

By Lemma 3.1, finding a consistent partition of the columns of a matrix A is equivalent to finding a *partial* distance-2 coloring of $G_b(A) = (V_1, V_2, E)$ on V_2. The following result formalizes the equivalence.

Theorem 3.3 *Let A be a nonsymmetric matrix and $G_b(A) = (V_1, V_2, E)$ be its bipartite graph representation. A mapping ϕ is a partial distance-2 coloring of $G_b(A)$ on V_2 if and only if ϕ induces a consistent partition of the columns of A.*

In view of Theorem 3.3, Problem MPP1 is equivalent to the following graph coloring problem (GCP).

Problem 3.4 (*GCP1*) *Given the bipartite graph $G_b(A) = (V_1, V_2, E)$ representing the sparsity structure of a matrix $A \in R^{m \times n}$, find a partial (2,p)-coloring of $G_b(A)$ on V_2 with the least value of p.*

For matrices with a few dense rows, a row partition may yield fewer groups than a column partition. Consequently, the matrix problem one needs to solve is MPP1 applied on A^T. In such cases, our graph formulation becomes handy—the equivalent problem is to find a partial distance-2 coloring on the vertex set V_1.

By Lemma 3.2, finding a consistent partition of the columns of a symmetric matrix A is equivalent to finding a distance-2 coloring of the adjacency graph $G(A)$. This equivalence was in fact first observed by McCormick [23]. However, as has been stated earlier, the symmetry present in A can be exploited to further reduce the number of groups (colors) required. Thus, we now consider the graph coloring formulation of the partitioning problem where A is symmetric (MPP2).

Consider a symmetric matrix A with nonzero diagonal elements and let a_{ij}, $i \neq j$, be any nonzero element in A. Further, let a_{ki}, $k \neq i, j$ and a_{jl}, $l \neq i, j, k$ be any other two nonzero elements. By Definition 2.4, in a symmetrically consistent partition of A,

- columns a_i and a_j should belong to two different groups (this is because both a_{ii} and a_{jj} are nonzero), and
- columns a_j and a_k should belong to two different groups, or columns a_i and a_l should belong to two different groups.

Coleman and Moré [8] gave an equivalent characterization of the aforementioned conditions in terms of a coloring of the associated adjacency graph. Specifically, they introduced the notion of *distance-$\frac{3}{2}$ coloring*[2], defined below.

Definition 3.5 A mapping $\phi : V \rightarrow \{1, 2, \ldots, p\}$ is a $(\frac{3}{2}, p)$-*coloring* of the graph $G = (V, E)$ if ϕ is $(1, p)$-coloring of G and every path of length three uses at least three colors.

The name 'distance-$\frac{3}{2}$ coloring' is chosen to reflect that it is in a sense 'in-between' distance-1 and distance-2 colorings. In particular, a distance-$\frac{3}{2}$ coloring is a relaxed distance-2 and a restricted distance-1 coloring. As an illustration, observe that a distance-1 coloring requires two colors for every path of length one, a distance-2 coloring requires three colors for every path of length two, and a distance-$\frac{3}{2}$ coloring is a distance-1 coloring further restricted to require three colors for every path of length three (see Figure 2). Note that a 4-cycle requires two, three, and four colors in a distance-1, a distance-$\frac{3}{2}$ and a distance-2 coloring, respectively.

The following theorem formalizes the connection between symmetrically consistent partition and distance-$\frac{3}{2}$ coloring. The result follows directly from the discussion that led to the definition of $(\frac{3}{2}, p)$-coloring.

Theorem 3.6 [Coleman and Moré [8]]
Let A be a symmetric matrix with nonzero diagonal elements and $G(A) = (V, E)$ be its adjacency graph representation. A mapping ϕ is a $(\frac{3}{2}, p)$-coloring of $G(A)$ if and only if ϕ induces a symmetrically consistent partition of the columns of A.

[2] Coleman and Moré used the term *path-coloring*.

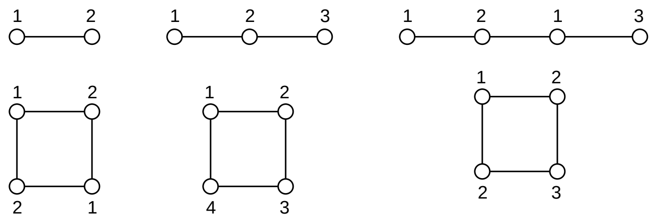

Figure 2: Distance-1, 2, and $\frac{3}{2}$ coloring of paths and a 4-cycle.

By Theorem 3.6, the following problem is equivalent to MPP2.

Problem 3.7 (*GCP2*) *Given the adjacency graph $G(A) = (V, E)$ representing the sparsity structure of a symmetric matrix $A \in R^{n\times n}$ with nonzero diagonal elements, find a $(\frac{3}{2}, p)$-coloring of $G(A)$ with the least value of p.*

Our next problem, MPP4, aims at finding a bipartition with the fewest number of groups consistent with a direct determination. When a nonsymmetric matrix A is represented by its bipartite graph $G_b(A) = (V_1, V_2, E)$, we have shown that a column partition consistent with a direct determination can be obtained by finding a partial distance-2 coloring of G_b on V_2. We now consider how this coloring has to be modified to capture a bipartition consistent with a direct determination. Notice that the coloring we are looking for should meet the following conditions.

- The sets V_1 and V_2 should use disjoint set of colors.
- Some vertices may not be involved in the determination of any nonzero entry of the underlying matrix. Such vertices are assigned a 'neutral' color (say color zero).
- Since every nonzero matrix entry has to be determined, for every edge in E, at least one of the endpoints has to be assigned a nonzero color.
- A nonzero matrix entry may be determined either from a positively colored column vertex or a positively colored row vertex. This suggests that the coloring condition sought here is some relaxation of the distance-2 coloring requirement imposed in the case of one-dimensional partition.

The following definition, introduced by Coleman and Verma [9], albeit using a different terminology, formalizes the conditions listed above. The subsequent theorem establishes the equivalence between the matrix and graph problems.

Definition 3.8 Let $G_b = (V_1, V_2, E)$ be a bipartite graph. A mapping $\phi : [V_1, V_2] \rightarrow \{0, 1, \ldots, p\}$ is a *distance-$\frac{3}{2}$ bicoloring* of G_b if the following conditions hold.

1. If $u \in V_1$ and $v \in V_2$, then $\phi(u) \neq \phi(v)$ or $\phi(u) = \phi(v) = 0$.
2. If $(u, v) \in E$, then $\phi(u) \neq 0$ or $\phi(v) \neq 0$.
3. If vertices u and v are adjacent to vertex w with $\phi(w) = 0$, then $\phi(u) \neq \phi(v)$.
4. Every path of three edges uses at least three colors.

Theorem 3.9 [Coleman and Verma[9]]
Let A be an $m \times n$ matrix and $G_b(A) = (V_1, V_2, E)$ be its bipartite graph. The mapping $\phi : [V_1, V_2] \rightarrow \{0, 1, \ldots, p\}$ is a distance-$\frac{3}{2}$ p-bicoloring if and only if ϕ induces a bipartition (Π_C, Π_R) of A, with $|\Pi_C| + |\Pi_R| = p$, consistent with a direct determination.

Thus MPP4 is equivalent to the following graph problem.

Problem 3.10 (*GCP4*) *Given the bipartite graph $G_b(A) = (V_1, V_2, E)$ representing the sparsity structure of an $m \times n$ matrix A, find a distance-$\frac{3}{2}$ p-bicoloring of $G_b(A)$ with the least value of p.*

3.5 Coloring Problems in Substitution Methods

In this subsection, we consider problems MPP3 and MPP5. To formulate MPP3 as a graph problem, we introduce the notion of *acyclic* coloring. Coleman and Cai [4] established the connection between acyclic[3] coloring and the estimation of a symmetric matrix using a substitution method. Acyclic coloring had been studied earlier by Grünbaum [15] in a different context.

Definition 3.11 *A mapping $\phi : V \rightarrow \{1, 2, \ldots, p\}$ is an acyclic p-coloring of a graph $G = (V, E)$ if ϕ is $(1, p)$-coloring and every cycle in G uses at least three colors.*

Theorem 3.12 [Coleman and Cai [4]]
Let A be a symmetric matrix with nonzero diagonal elements and $G(A) = (V, E)$ be its adjacency graph representation. A mapping ϕ is an acyclic p-coloring of $G(A)$ if and only if ϕ induces a substitutable partition of the columns of A.

[3]Coleman and Cai use the term cyclic coloring to refer to what is known as acyclic coloring in the graph theoretic literature.

Consider the 5×5 symmetric matrix and its adjacency graph depicted in Figure 3. An optimal acyclic coloring that uses three colors is shown. It can be verified that this induces a substitutable partition of the columns. By contrast, a symmetric partition consistent with a direct determination (a distance-$\frac{3}{2}$ coloring) would have required four groups (colors).

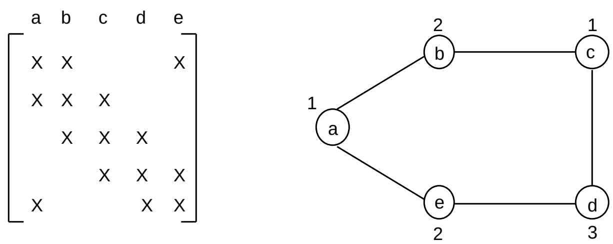

Figure 3: A 5×5 symmetric matrix and an acyclic coloring of its adj. graph

In view of Theorem 3.12, MPP3 is equivalent to the following graph problem.

Problem 3.13 (*GCP3*) *Given the adjacency graph $G(A) = (V, E)$ representing the sparsity structure of a symmetric matrix $A \in R^{n \times n}$ with nonzero diagonal elements, find an acyclic p-coloring of $G(A)$ with the least value of p.*

The relationship between bicoloring and bipartition, established by Theorem 3.9, coupled with that between acyclic coloring and substitutable partition, established by Theorem 3.12, suggests that 'acyclic bicoloring' might be the right graph model for MPP5. Coleman and Verma [9] showed that this was indeed the case.

Definition 3.14 Let $G_b = (V_1, V_2, E)$ be a bipartite graph. A mapping $\phi : [V_1, V_2] \rightarrow \{0, 1, \ldots, p\}$ is an *acyclic bicoloring* of G_b if the following conditions hold.

1. If $u \in V_1$ and $v \in V_2$, then $\phi(u) \neq \phi(v)$ or $\phi(u) = \phi(v) = 0$.
2. If $(u, v) \in E$, then $\phi(u) \neq 0$ or $\phi(v) \neq 0$.
3. If vertices u and v are adjacent to vertex w with $\phi(w) = 0$, then $\phi(u) \neq \phi(v)$.
4. Every cycle uses at least three colors.

Matrix	1D Partition	2D Partition	Method
Jacobian	distance-2 coloring	distance-$\frac{3}{2}$ bicoloring	Direct
Hessian	distance-$\frac{3}{2}$ coloring	NA	Direct
Jacobian	NA	acyclic bicoloring	Substitution
Hessian	acyclic coloring	NA	Substitution

Table 3: Graph coloring formulations for estimating *all* nonzero entries of derivative matrices. The Jacobian and the Hessian are represented by their bipartite and adjacency graphs, respectively. NA stands for not applicable.

Theorem 3.15 [Coleman and Verma [9]]
Let A be an $m \times n$ matrix and $G_b(A) = (V_1, V_2, E)$ be its bipartite graph. The mapping $\phi : [V_1, V_2] \to \{0, 1, \ldots, p\}$ is an acyclic p-bicoloring if and only if ϕ induces a bipartition (Π_C, Π_R) of A, with $|\Pi_C| + |\Pi_R| = p$, consistent with determination by substitution.

By Theorem 3.15, the following coloring problem is equivalent to MPP5.

Problem 3.16 (*GCP5*) *Given the bipartite graph $G_b(A) = (V_1, V_2, E)$ representing the sparsity structure of an $m \times n$ matrix A, find an acyclic p-bicoloring of $G_b(A)$ with the least value of p.*

Table 3 (the same table was given in Section 1) summarizes the graph coloring problems that arise in efficient derivative matrix estimation.

3.6 Distance-k Chromatic Numbers

In this subsection, we expose the inter-relationship among the various colorings introduced thus far and show that distance-2 coloring is the most *general* among them.

The *power* of a graph gives an alternative view to the DkGCP. The kth power of a graph $G = (V, E)$ is the graph $G^k = (V, F)$ where $(u, v) \in F$ if and only if u and v are distance-k neighbors in G. The following equivalence follows immediately.

Lemma 3.17 *Let G^k be the kth power of graph G. A mapping ϕ is a (k, p)-coloring of G if and only if it is a $(1, p)$-coloring of G^k.*

A particular implication of Lemma 3.17 is that distance-2 coloring of a graph is equivalent to distance-1 coloring of the square of the graph. This establishes the equivalence between GCP1, our bipartite graph based formulation of MPP1, and the distance-1 coloring formulation, which uses the column intersection graph, suggested by Coleman and Moré [7]. Specifically, as has been shown in [7], the column intersection graph $G_c(A)$ of a matrix A is isomorphic to the adjacency graph of $A^T A$. We note that $G_c(A)$ is in fact the subgraph of $G_b(A)^2$ induced by the vertices in V_2. For a graph $G = (V, E)$, let the graph induced by $U \subseteq V$ be denoted by $G[U]$.

Lemma 3.18 *Let $G_b(A) = (V_1, V_2, E)$ and $G_c(A) = (V_2, E')$ be the bipartite and column intersection graphs of matrix A. Then, $G_c = {G_b}^2[V_2]$.*

Observe that the distance-1 neighbors of a vertex in a graph G form a *clique* in the square of the graph. A clique is a set of vertices in which the vertices are mutually adjacent to each other. This observation immediately provides a lower bound on $\chi_2(G)$, the 2-chromatic number of G. Let Δ denote the maximum degree-1 in G.

Lemma 3.19 *For any graph G, $\chi_2(G) \geq \Delta + 1$.*

Proof: Observe that the cardinality of a maximum clique in the square graph G^2 is $\Delta + 1$. □

Further, the conditions required by distance-1 coloring, acyclic coloring, distance-$\frac{3}{2}$ coloring, acyclic bicoloring, distance-$\frac{3}{2}$ bicoloring, and distance-2 coloring imply the following relationships among their respective chromatic numbers. The chromatic numbers for distance-k coloring (and bicoloring) of a general graph G (and bipartite graph G_b) are denoted by $\chi_k(G)$ (and $\chi_{kb}(G_b)$). Similarly the chromatic numbers for acyclic coloring (and bicoloring) of a graph G (and a bipartite graph G_b) are denoted by $\chi^a(G)$ (and $\chi^{ab}(G_b)$).

Theorem 3.20 *For a general graph $G = (V, E)$,*

$$\chi_1(G) \leq \chi^a(G) \leq \chi_{\frac{3}{2}}(G) \leq \chi_2(G) = \chi_1(G^2).$$

Proof: Observe that a distance-2 coloring is a distance-$\frac{3}{2}$ coloring; a distance-$\frac{3}{2}$ coloring is an acyclic coloring; and an acyclic coloring is a distance-1 coloring. □

Theorem 3.21 *For a bipartite graph $G_b = (V_1, V_2, E)$,*

$$\chi^{ab}(G_b) \leq \chi_{\frac{3}{2}b}(G_b) \leq min\{\chi_1({G_b}^2[V_1]), \chi_1({G_b}^2[V_2])\},$$

where $({G_b}^2[W])$ is the sub-graph of ${G_b}^2$ induced by W.

Proof: The first inequality is obvious. For the second inequality, observe that a partial distance-2 coloring on V_2 is a valid distance-$\frac{3}{2}$ bicoloring where all the vertices in V_1 are specified to be colored with 0. A similar argument, with the roles of V_1 and V_2 interchanged, can be used to complete the proof. □

In the context of matrix estimation using numerical methods, the implication of Theorem 3.21 is that, an optimal two-dimensional partition, irrespective of the structure of the matrix, yields fewer (or at most as many) groups compared to an optimal one-dimensional partition, and hence potentially results in a more efficient computation.

The results in this section show that distance-2 coloring is an archetypal model in the estimation of Jacobian and Hessian matrices using techniques that rely on one-dimensional partition via direct and substitution methods.

Distance-2 coloring has other applications. Examples include channel assignment [20] and facility location problems (see Chapter 5 in [27]). From a more theoretical perspective, distance-2 coloring for planar graphs has been studied in [2] and a similar study for chordal graphs is available in [1].

4 Partial Matrix Estimation

In many PDE constrained optimization contexts, the Jacobian or the Hessian is formed only for preconditioning purposes. For preconditioning, it is often common to compute a *subset* of the matrix elements. Computing a good preconditioner is critical for fast convergence to the solution. The recent survey article [19] discusses various applications where methods known as "Jacobian-free Newton-Krylov" are used. A basic ingredient in the use of these methods is an approximate computation of *some* elements of the Jacobian. Also, there are examples in which only certain elements of the Hessian need to be updated in an iterative procedure, while others do not because they are unlikely to change in value [3].

In this section we develop the graph coloring formulations of the partitioning problems that arise in the estimation of a *specified subset* of the nonzero entries of a matrix via direct methods. We call this *partial* matrix estimation as opposed to *full* matrix estimation, where all nonzero entries are required to be determined.

The coloring formulations in this section are new and more sophisticated than the coloring formulations in full matrix estimation. The motivation for developing the new graph formulations is that efficient partial matrix estimation can be used to further reduce the number of colors needed to estimate the required elements. For example, if only the diagonal elements of a Hessian are needed, then we need only the *distance-1 coloring* of the adjacency graph, rather than the distance-$\frac{3}{2}$ coloring required for full matrix estimation.

The colorings defined in this section allow a vertex to have the color zero. A vertex with color zero signifies the fact that it would not be used to estimate any element of the matrix represented by the graph, i.e., columns or rows that correspond to the color zero are not used to estimate any elements in those columns or rows.

The rest of this section is organized in three parts. Each part deals with a scenario defined by the kind of matrix under consideration (symmetric or nonsymmetric) and the type of partition employed (one-dimensional or two-dimensional). In each case, the required entries are assumed to be determined using a direct method. The problems that correspond to estimation via substitution are not considered in this paper.

4.1 Nonsymmetric matrix, One-dimensional partition

Let $A \in R^{m \times n}$ be a nonsymmetric matrix, and S denote the set of nonzero elements of A required to be estimated. A partition $\{C_1, \ldots, C_p\}$ of a subset of the columns of A is *consistent with a direct determination of* S if for every $a_{ij} \in S$, column a_j is included in some group that contains no other column with a nonzero in row i.

Let $G_b(A) = (V_1, V_2, E)$ be the bipartite graph of A, and $F \subseteq E$ correspond to the elements of S. A mapping $\phi : V_2 \rightarrow \{0, 1, \ldots, p\}$ is a *distance-2 coloring of* G_b *restricted to* F if the following conditions hold for every $(v, w) \in F$, where $v \in V_1$, $w \in V_2$.

1. $\phi(w) \neq 0$, and
2. for every path (u, v, w), $\phi(u) \neq \phi(w)$.

Theorem 4.1 *The mapping ϕ is a distance-2 p-coloring of $G_b(A)$ restricted to F if and only if ϕ induces a column partition $\{C_1, \ldots, C_p\}$ consistent with a direct determination of S.*

Proof: Assume that ϕ is a distance-2 p-coloring of $G_b(A)$ restricted to F. We show that the groups $\{C_1, \ldots, C_p\}$ where $C_\alpha = \{a_j : \phi(a_j) = \alpha\}$, $1 \leq \alpha \leq p$, constitute a partition of a subset of the columns of A consistent with a direct determination of S. First, observe that by coloring condition 1, for every $a_{ij} \in S$ ($(r_i, a_j) \in F$), $\phi(a_j) \neq 0$ and thus column a_j belongs to group $C_{\phi(a_j)}$ and hence is involved in the partition. Assume now that the partition induced by the coloring is not consistent with a direct determination of S. This occurs only if there exist nonzero elements a_{ij} and a_{ik}, $k \neq j$, such that $a_{ij} \in S$ and both a_j and a_k belong to group $C_{\alpha'}$ for some α', $1 \leq \alpha' \leq p$. But this contradicts coloring condition 2, and hence cannot occur.

Conversely, assume that the partition $C = \{C_1, \ldots, C_p\}$ is consistent with a direct determination of S. Construct a coloring ϕ of $G_b(A)$ as follows. $\phi(a_j) = \alpha$ if $a_j \in C_\alpha$, and $\phi(a_j) = 0$ if $a_j \notin C$. We claim that ϕ is a distance-2 p-coloring of $G_b(A)$ restricted to F. Each vertex in V_2 incident to an edge in F corresponds to a column with an entry in S and thus gets a nonzero color. Thus ϕ satisfies coloring condition 1. Consider any path (a_j, r_i, a_k) where $(r_i, a_j) \in F$. Note that such a path in $G_b(A)$ exists whenever entries a_{ij} and a_{ik} are nonzero. The partition condition implies that column a_k cannot be in the same group as a_j. Thus, by construction, $\phi(a_j) \neq \phi(a_k)$, satisfying coloring condition 2. □

4.2 Symmetric matrix, One-dimensional partition

Let $A \in R^{n \times n}$ be a symmetric matrix with nonzero diagonal elements, and S denote the set of nonzero elements of A required to be estimated. A partition $\{C_1, \ldots, C_p\}$ of a subset of the columns of A is *symmetrically consistent with a direct determination of* S if for every $a_{ij} \in S$ at least one of the following two conditions are met.

1. The group containing a_j has no other column with a nonzero in row i.

2. The group containing a_i has no other column with a nonzero in row j.

Let $G(A) = (V, E)$ be the adjacency graph of A, $F_{od} \subseteq E$ correspond to the off-diagonal elements in S, and F_d correspond to the diagonal elements in S, i.e., $F_d = \{(u, u) : u \in U\}$ where $U \subseteq V$. Let $F = F_{od} \cup F_d$. A mapping $\phi : V \to \{0, 1, 2, \ldots, p\}$ is a *distance-*$\frac{3}{2}$ *coloring of* G *restricted to* F if the following conditions hold.

1. For every $(u, u) \in F_d$,

    1.1. $\phi(u) \neq 0$, and

    1.2. for every $(u, v) \in E$, $\phi(u) \neq \phi(v)$.

2. For every $(v, w) \in F_{od}$,

    2.1. $\phi(v) \neq \phi(w)$, and

    2.2. at least one of the following two conditions holds:

2.2.1. $\phi(v) \neq 0$ and for every path (v, w, x), $\phi(v) \neq \phi(x)$ or

2.2.2. $\phi(w) \neq 0$ and for every path (u, v, w), $\phi(u) \neq \phi(w)$.

Theorem 4.2 *The mapping ϕ is a distance-$\frac{3}{2}$ p-coloring of $G(A)$ restricted to F if and only if ϕ induces a column partition $\{C_1, \ldots, C_p\}$ symmetrically consistent with a direct determination of S.*

Proof: Assume that ϕ is a distance-$\frac{3}{2}$ p-coloring of $G(A)$ restricted to F. We show that the groups $\{C_1, \ldots, C_p\}$ where $C_\alpha = \{a_j : \phi(a_j) = \alpha\}$, $1 \leq \alpha \leq p$, constitute a partition of a subset of the columns of A symmetrically consistent with a direct determination of S.

By coloring conditions 1 and 2, for every $a_{ij} \in S$ ($(a_i, a_j) \in F$), at least one of a_i or a_j has a nonzero color and hence is involved in the partition $\{C_1, \ldots, C_p\}$. Let $a_{ij} \in S$ be a diagonal entry ($i = j$). Then, coloring condition 1 ensures that $\phi(a_i) \neq 0$ and that $\phi(a_i) \neq \phi(a_k)$ for every $(a_i, a_k) \in E$. Thus, by construction, column a_i belongs to group $C_{\phi(a_i)}$ and no column a_k, $k \neq i$ with $a_{ik} \neq 0$ is in $C_{\phi(a_i)}$. This clearly satisfies the partition condition. Let $a_{ij} \in S$ now be an off-diagonal entry ($i \neq j$). Assume without loss of generality that $\phi(a_j) \neq 0$. By condition 2.1, $\phi(a_j) \neq \phi(a_i)$. By condition 2.2.2, there is no path (a_k, a_i, a_j) in $G(A)$, for any $k \neq i, j$ such that $\phi(a_j) = \phi(a_k)$. The last two statements together imply that column a_j belongs to group $C_{\phi(a_j)}$ and that no column a_k, $k \neq j$ with $a_{ik} \neq 0$ is in $C_{\phi(a_j)}$. This satisfies partition condition 1. A similar argument applies to the case where $\phi(a_i) \neq 0$ which implies the satisfaction of the alternative partition condition.

To prove the converse, assume that the partition $C = \{C_1, \ldots, C_p\}$ is symmetrically consistent with a direct determination of S. Construct a coloring ϕ of $G(A)$ as follows. $\phi(a_j) = \alpha$ if $a_j \in C_\alpha$, and $\phi(a_j) = 0$ if $a_j \notin C$. We claim that ϕ is a distance-$\frac{3}{2}$ p-coloring of $G(A)$ restricted to F.

Consider a diagonal element $a_{ii} \in S$. The partition conditions ensure that a_i is in some group $C_{\alpha'}$ and that there is no column $a_k \in C_{\alpha'}$, $k \neq i$ such that $a_{ik} \neq 0$. Thus, by construction, $\phi(a_i) \neq 0$ and $\phi(a_i) \neq \phi(a_k)$ for every $(a_i, a_k) \in E$, satisfying coloring condition 1. Consider now the case where $a_{ij} \in S$ is an off-diagonal element. First, observe that since all diagonal elements are nonzero, a_i and a_j cannot belong to the same group. Thus $\phi(a_i) \neq \phi(a_j)$, satisfying coloring condition 2.1. Second, observe that there are two possibilities by which the partitioning conditions have been satisfied. We consider only one of these; the second one can be treated in a similar manner. Suppose a_j belongs to some group $C_{\alpha'}$ and that there is no other column $a_k \in C_{\alpha'}$, $k \neq j$ such that $a_{ik} \neq 0$. Thus, by construction,

$\phi(a_j) \neq 0$ and $\phi(a_j) \neq \phi(a_k)$ for every path (a_k, a_i, a_j) in $G(A)$, satisfying coloring condition 2.2.2. □

A special case of Theorem 4.2 is the problem of estimating *only* the diagonal elements of A, i.e., $F_d = \{(v, v) : v \in V\}$ and $F_{od} = \emptyset$. For this problem, condition 1 is the only applicable condition, and states that a distance-1 coloring of $G(A)$ is sufficient.

4.3 Nonsymmetric matrix, Two-dimensional partition

Let $A \in R^{m \times n}$ be a nonsymmetric matrix, and S denote the set of nonzero elements of A required to be estimated. A bipartition (Π_C, Π_R) of a subset of the columns and rows of A is *consistent with a direct determination of* S if for every $a_{ij} \in S$ at least one of the following conditions are met.

1. The group (in Π_C) containing column j has no other column having a nonzero in row i.

2. The group (in Π_R) containing row i has no other row having a nonzero in column j.

Let $G_b(A) = (V_1, V_2, E)$, and $F \subseteq E$ correspond to the elements in S. A mapping $\phi : [V_1, V_2] \rightarrow \{0, 1, \ldots, p\}$ is said to be a *distance-$\frac{3}{2}$ bicoloring of G_b restricted to F* if the following conditions are met.

1. Vertices in V_1 and V_2 receive disjoint colors, except for color 0; i.e., for every $u \in V_1$ and $v \in V_2$, either $\phi(u) \neq \phi(v)$ or $\phi(u) = \phi(v) = 0$.

2. At least one endpoint of an edge in F receives a nonzero color; i.e., for every $(v, w) \in F$, $\phi(v) \neq 0$ or $\phi(w) \neq 0$.

3. For every $(v, w) \in F$,

   3.1. if $\phi(v) = 0$, then, for every path (u, v, w), $\phi(u) \neq \phi(w)$,

   3.2. if $\phi(w) = 0$, then, for every path (v, w, x), $\phi(v) \neq \phi(x)$,

   3.3. if $\phi(v) \neq 0$ and $\phi(w) \neq 0$, then for every path (u, v, w, x), either $\phi(u) \neq \phi(w)$ or $\phi(v) \neq \phi(x)$.

Theorem 4.3 *The mapping ϕ is a distance-$\frac{3}{2}$ p-bicoloring of G_b restricted to F if and only if ϕ induces a bipartition (Π_C, Π_R), $|\Pi_C| + |\Pi_R| = p$, consistent with a direct determination of S.*

Proof: Let the construction of a partition given a coloring, and vice-versa, be done in a similar manner as in the proof of Theorem 4.1.

Assume that ϕ is a distance-$\frac{3}{2}$ bicoloring of $G_b(A)$ restricted to F. Let the induced bipartition be (Π_C, Π_R). Coloring condition 1 implies that (Π_C, Π_R) is a bipartition. By condition 2, for every $a_{ij} \in S$, either $a_j \in \Pi_C$ or $r_i \in \Pi_R$ (or both). Assume now that (Π_C, Π_R) is not consistent with a direct determination of S. This occurs only if one of the following cases hold for any $a_{ij} \in S$:

- $\phi(r_i) = 0, \phi(a_j) \neq 0$ and there exists a column a_k, $k \neq j$ with $a_{ik} \neq 0$ such that $\phi(a_j) = \phi(a_k)$. But this contradicts coloring condition 3.1, and hence cannot occur.

- $\phi(a_j) = 0, \phi(r_i) \neq 0$ and there exists a row r_l, $l \neq i$ with $a_{lj} \neq 0$ such that $\phi(r_i) = \phi(r_l)$. But this contradicts coloring condition 3.2, and hence cannot occur.

- $\phi(r_i) \neq 0, \phi(a_j) \neq 0$, and there exist column a_k, $k \neq j$ with $a_{ik} \neq 0$, and row r_l, $l \neq i$ with $a_{lj} \neq 0$ such that $\phi(a_j) = \phi(a_k)$ and $\phi(r_i) = \phi(r_l)$. But this contradicts coloring condition 3.3, and hence cannot occur.

Hence, (Π_C, Π_R) is consistent with a direct determination of S.

Conversely, assume that (Π_C, Π_R) is a bipartition consistent with a direct determination of S. Clearly, the constructed coloring ϕ satisfies conditions 1 and 2. To complete the proof, we show that ϕ also satisfies condition 3. Assume that ϕ violates condition 3. Then one of the following cases must have happened:

- There exists a path (a_k, r_i, a_j) for some $(r_i, a_j) \in F$ such that $\phi(r_i) = 0$ and $\phi(a_j) = \phi(a_k)$. But this implies that element a_{ij} cannot be determined directly, contradicting the assumption that (Π_C, Π_R) is consistent with a direct determination of S.

- There exists a path (r_i, a_j, r_l) for some $(r_i, a_j) \in F$ such that $\phi(a_j) = 0$ and $\phi(r_i) = \phi(r_l)$. Again this implies that element a_{ij} cannot be determined directly, a contradiction of our assumption.

- There exists a path (a_k, r_i, a_j, r_l) for some $(r_i, a_j) \in F$ such that $\phi(r_i) = \phi(r_l) \neq 0$ and $\phi(a_j) = \phi(a_k) \neq 0$. But this implies that element a_{ij} cannot be determined directly, contradicting the assumption.

□

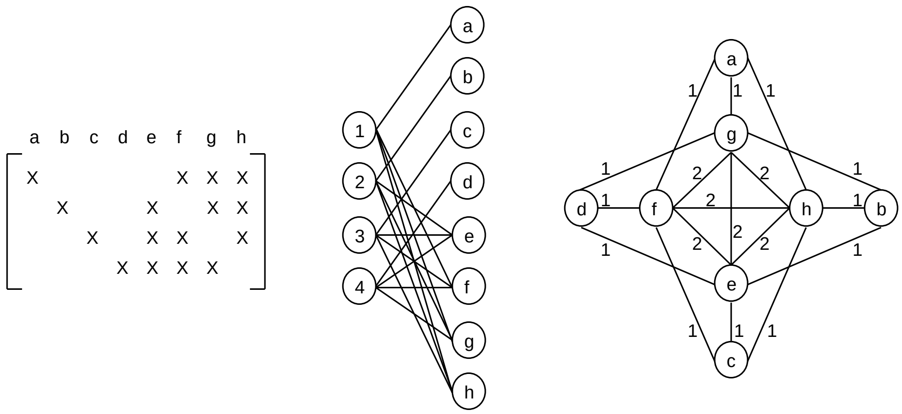

Figure 4: A matrix and its bipartite and intersection graph, resp.

5 Column Intersection vs. Bipartite Graph Formulation

In this section we compare our bipartite graph based formulations with formulations based on a column intersection graph.

Figure 4 depicts a matrix A, the corresponding bipartite graph $G_b(A) = (V_1, V_2, E)$, and the column intersection graph $G_c(A) = (V_2, E')$. Note that we have augmented the graph $G_c(A)$ with *edge weights*—$w(a_i, a_j)$ is the size of the intersection of the sets represented by vertices a_i and a_j. In terms of matrix A, $w(a_i, a_j)$ is the total number of rows where both columns a_i and a_j have nonzero entries. In the bipartite graph $G_b(A)$, it corresponds to the number of common neighbors of vertices a_i and a_j.

In Sections 5.1 to 5.4, we compare and contrast the two graph formulations in terms of flexibility, storage space requirement, ease of graph construction, and use of existing software.

5.1 Flexibility

Notice that the column intersection graph is a 'compressed' representation of the structure of the underlying matrix. Clearly, some information is lost in the compression process. In particular, given an edge in G_c between two columns, we cannot determine the row at which they share nonzero entries. By contrast, the bipartite graph is an equivalent representation of the structure of the matrix. This provides flexibility. As an illustration, notice that the bipartite graph can be used in a one-dimensional (row or column) partition as well as a two-dimensional (combined row and column) partition. The column intersection graph, on the other hand, is applicable only to a column partition. Moreover, the graph formulations for partial matrix estimation problems would have required new types of 'intersection' graphs to be defined. In general, the advantage of the bipartite graph representation is that the representation is decoupled from the eventual technique to be employed and the matrix entries to be determined.

5.2 Storage Space Requirement

Although Lemma 3.18 correlates the bipartite graph of a matrix with its column intersection graph, one cannot immediately deduce that one graph is denser than the other. The density of the respective graph namely depends on the structure of the matrix. Here, we make a rough analysis to show that for sparse matrices of practical interest, the column intersection graph is likely to be denser than the bipartite counterpart.

Given a matrix A, the graph $G_b(A) = (V_1, V_2, E)$ and the weighted graph $G_c(A) = (V_2, E')$, for a vertex u in G_b, let $N_1(u) = \{v : (u,v) \in E\}$, $d_1(u) = |N_1(u)|$, and let the average degree-1 in the sets V_1 and V_2 of G_b be $\overline{\delta}_1(V_1)$ and $\overline{\delta}_1(V_2)$, respectively. Further, let $\overline{w}$ denote the *average edge weight* in G_c. Then,

$$\begin{aligned}
\sum_{e\in E'} w(e) &= \frac{1}{2}\cdot \sum_{u\in V_2}\sum_{v\in N_1(u)} (d_1(v)-1)\\
|E'|\cdot\overline{w} &\approx \frac{1}{2}\cdot |V_2|\cdot\overline{\delta}_1(V_2)\cdot(\overline{\delta}_1(V_1)-1)\\
&= \frac{1}{2}\cdot |E|\cdot(\overline{\delta}_1(V_1)-1)\\
|E'| &= |E|\cdot(\frac{\overline{\delta}_1(V_1)-1}{2\overline{w}})
\end{aligned}$$

Therefore, as long as $\frac{\overline{\delta}_1(V_1)-1}{2\overline{w}} > 1$, the column intersection graph is likely to have more edges (and hence requires more storage space) than the bipartite graph of the matrix.

5.3 Ease of Construction

The sparsity structure of matrix A directly, without any further computation, gives the corresponding bipartite graph $G_b(A)$. In principle, the data structure used to represent A can be used for implementing algorithms that use $G_b(A)$. By contrast, $G_c(A)$ has to be computed. As the following Lemma states, the time required for the computation of $G_c(A)$ is proportional to the number of edges in $G_c(A)$.

Lemma 5.1 *Given a graph $G_b(A) = (V_1, V_2, E)$, the time required for constructing $G_c(A)$ is $T_{const} = O(|V_2|(\overline{\delta}_1(V_2)(\overline{\delta}_1(V_1) - 1))$.*

It should however be noted that once the graph $G_c(A)$ is computed, a subsequent distance-1 coloring of $G_c(A)$ can be done faster than a distance-2 coloring of $G_b(A)$. As we shall show in Section 6, the overall time required for constructing and distance-1 coloring of $G_c(A)$ is of the same order as the time required for a direct distance-2 coloring of $G_b(A)$.

5.4 Use of Existing Software

Serial program packages that implement various practically effective distance-1 coloring heuristics exist [5, 6]. For matrix partitioning problems where a column intersection graph based formulation can be applied, these packages can be readily used. On the other hand, since distance-2 coloring is a prototypical model in our context, efficient programs, including parallel ones, for the distance-2 coloring problem need to be developed.

6 Distance-k Coloring Algorithms

In this section we present several fast algorithms for the various coloring problems considered in this paper. The algorithm design is greatly simplified by taking distance-2 coloring as a generic starting point.

For any fixed integer $k \geq 1$, DkGCP is NP-hard [21]. A proof sketch showing that the problem of finding a $(\frac{3}{2}, p)$-coloring with the minimum p is NP-hard, even if the graph is bipartite, is given in [8]. The problem of finding a distance-$\frac{3}{2}$ bicoloring with the fewest number of colors is also NP-hard [9]. Further, finding an acyclic p-coloring with the least value of p is NP-hard [4].

Since all the graph coloring problems of our concern are NP-hard, in practical applications, we are bound to rely on using *approximation algorithms* or *heuristics*. An algorithm $\mathcal{A}$ is said to be a γ-approximation algorithm for a minimization problem if its runtime is polynomial in the input size and if for every problem instance $\mathcal{I}$ with an optimal solution $OPT(\mathcal{I})$, the solution $\mathcal{A}(\mathcal{I})$ output by $\mathcal{A}$ is such that $\frac{\mathcal{A}(\mathcal{I})}{OPT(\mathcal{I})} \leq \gamma$. The *approximation ratio* $\gamma \geq 1$, and the goal is to make γ as close to unity as possible. If no such guarantee can be given for the quality of an approximate solution obtained by a polynomial time algorithm, the algorithm is usually referred to as a heuristic.

In the case of distance-1 coloring, there exist several, practically effective heuristics [7]. In this section we show that some of the ideas used in the distance-1 coloring heuristics can be adapted to the cases considered in this paper by extending the notion of *neighborhood*. The algorithms we present are *greedy* in nature, i.e., the vertices of a graph are processed in some order and at each step a decision that looks best at the moment (and that will not be reversed later) is made.

In Section 6.1, we present a generic greedy distance-2 coloring algorithm and give a detailed analysis of its performance both in terms of computation time and number of colors used. In Sections 6.2 to 6.5, adaptations of this algorithm, tailored to the various coloring problems of our concern are presented.

Notations Some notations used in the rest of this section are first in order. Recall that for a vertex u in a graph G, a vertex $w \neq u$ is a *distance-k neighbor* of u if the shortest path connecting u and w has length $\leq k$. Let $N_k(u) = \{w : w \text{ is a distance-}k \text{ neighbor of } u\}$. Let $d_k(u) = |N_k(u)|$ denote the degree-k of u; Δ be the maximum degree-1 in G; and $\overline{\delta}_k = \frac{1}{|V|}\sum_{u \in V} d_k(u)$ denote the average degree-k in G.

Further, in a bipartite graph $G_b = (V_1, V_2, E)$, let the maximum degree-1 in the vertex sets V_1 and V_2 be denoted by $\Delta(V_1)$ and $\Delta(V_2)$, respectively. Similarly, let the average degree-k in the sets V_1 and V_2 be denoted by $\overline{\delta}_k(V_1)$ and $\overline{\delta}_k(V_2)$, respectively.

6.1 Distance-2 Coloring Algorithms

A simple approach for an approximate distance-2 coloring of a graph $G = (V, E)$ is to visit the vertices in some order, each time assigning a vertex the smallest color that is not used by any of its distance-2 neighbors. Note that the degree-2 of a vertex u in G is bounded by Δ^2, i.e., $d_2(u) \leq \sum_{w \in N_1(u)} d_1(w) \leq \Delta \cdot d_1(u) \leq \Delta^2$. Thus, since the vertices in G can always be distance-2 colored trivially using $|V|$ different colors, it is always possible to color a vertex using a value from the set $\{1, 2, \ldots, min\{\Delta^2 + 1, |V|\}\}$. Algorithm GreedyD2Coloring, outlined below, uses this as it colors the vertices of the graph in an arbitrary order. In the algorithm, color(v) is the color assigned to vertex v and forbiddenColors is a vector of size $C_{max} = \min\{\Delta^2 + 1, |V|\}$ used to mark the colors that cannot be assigned to a particular vertex. Specifically, forbiddenColors(c) $= v$ indicates that color c cannot be assigned to vertex v.

Lemma 6.1 *GreedyD2Coloring finds a distance-2 coloring in time $O(|V|\overline{\delta}_2)$.*

GreedyD2Coloring($G = (V, E)$)

for each $v_i \in V$ **do**
  for each colored vertex $u \in N_2(v_i)$ **do**
    forbiddenColors(color(u)) $= v_i$
  end-for
  color(v_i) $= \min\{c : \text{forbiddenColors}(c) \neq v_i\}$
end-for

Proof: We first show correctness. In step i of the algorithm, the color used by each of the distance-2 neighbors of vertex v_i is marked (using v_i) in the vector forbiddenColors. Thus, at the end of the inner for loop, the set of colors that are allowed for vertex v_i is the set of indices in forbiddenColors where the mark used is different from v_i. The minimum value in this set is thus the smallest allowable color for vertex v_i. Notice that the vector forbiddenColors does not need to be initialized at every step as the marker v_i is used *only* in step i.

Turning to complexity, note that marking the forbidden colors at step i of the algorithm takes $O(d_2(v_i))$ time. Finding the smallest allowable color to v_i can be done within the same order of time by scanning forbiddenColors sequentially until the first index c where a value other than v_i is stored is found. The total time is thus proportional to $\sum_{v \in V} d_2(v) = O(|V|\overline{\delta}_2)$. □

We now analyze the quality of the solution provided by GreedyD2Coloring. Let the number of colors used by GreedyD2Coloring on a graph $G = (V, E)$ be ${\chi_2}^{greedy}(G)$. Then recalling the lower bound given in Lemma 3.19, we get the following theorem and its corollary.

Theorem 6.2 $\Delta + 1 \leq \chi_2(G) \leq {\chi_2}^{greedy}(G) \leq min\{\Delta^2 + 1, |V|\}$.

Corollary 6.3 *GreedyD2Coloring is an $O(\sqrt{|V|})$-approximation algorithm.*

Proof: The approximation ratio γ is at most $\frac{1}{\Delta+1} \cdot min\{\Delta^2 + 1, |V|\}$. There are two possibilities to consider. In the first case $\Delta^2 + 1 < |V|$. This implies $\Delta = O(\sqrt{|V|})$ and $\gamma = \frac{\Delta^2+1}{\Delta+1} = O(\Delta) = O(\sqrt{|V|})$. In the second case $|V| < \Delta^2 + 1$. This implies $\Delta = \Omega(\sqrt{|V|})$ and $\gamma = \frac{|V|}{\Delta+1} = O(\sqrt{|V|})$. □

Note that for practical problems, such as problems that arise in solving PDEs using good finite element discretizations, $\Delta^2 + 1 \lll |V|$, making GreedyD2Coloring an $O(\Delta)$-approximation algorithm.

The actual number of colors used by GreedyD2Coloring depends on the order in which the vertices are visited. In GreedyD2Coloring, an arbitrary ordering is assumed. A solution with fewer number of colors can be expected if a more elaborate ordering criterion is used. For example, the ideas in *largest degree first* and *incidence degree ordering* for distance-1 coloring [7] can be adapted to the distance-2 coloring case.

As a final remark on the complexity of GreedyD2Coloring, we show that the algorithm runs in linear time in the number of vertices for certain sparse graphs. Let $\delta_2 = \frac{1}{|V|}\sum_{u \in V} d_1(u)^2$ and let the *standard deviation* of degree-1 in $G = (V, E)$ be given by

$$\sigma^2 = \frac{1}{|V|}\sum_{v \in V} (d_1(v) - \overline{\delta}_1)^2.$$

Then,

$$\begin{aligned} |V|\sigma^2 &= \sum_{v \in V} d_1(v)^2 + \sum_{v \in V} \overline{\delta}_1^{\,2} - 2\sum_{v \in V} d_1(v)\overline{\delta}_1 \\ &= |V|\delta_2 + |V|\overline{\delta}_1^{\,2} - 2\overline{\delta}_1 \sum_{v \in V} d_1(v) \\ &= |V|\delta_2 + |V|\overline{\delta}_1^{\,2} - 2|V|\overline{\delta}_1^{\,2} \\ &= |V|\delta_2 - |V|\overline{\delta}_1^{\,2} \end{aligned}$$

Rewriting we get,

$$\delta_2 = \overline{\delta}_1^{\,2} + \sigma^2.$$

Noting that $\delta_2 \geq \overline{\delta}_2$, we get the following corollary to Lemma 6.1.

Corollary 6.4 *GreedyD2Coloring has time complexity* $O(|V|(\overline{\delta}_1^{\,2} + \sigma^2))$.

Since $\overline{\delta}_1 = \frac{2|E|}{|V|}$, the complexity expression in Corollary 6.4 reduces to $O(\frac{|E|^2}{|V|})$ for graphs where $\sigma < \overline{\delta}_1$. In particular, for sparse graphs, where $|E| = O(|V|)$, the time complexity of GreedyD2Coloring becomes $O(|V|)$.

6.2 Partial Distance-2 Coloring Algorithms

Here we modify GreedyD2Coloring slightly to make it suitable for solving the partial distance-2 coloring problem GCP1 (our graph formulation of MPP1).

For any vertex $v \in V_2$, the number of vertices at distance *exactly* two units from v is at most $\Delta(V_2)(\Delta(V_1) - 1)$. Thus, vertex v can always be assigned a color from the set $\{1, 2, \ldots, C_{max}\}$, where $C_{max} = min\{\Delta(V_2)(\Delta(V_1) - 1)) + 1, |V_2|\}$. In Algorithm GreedyPartialD2Coloring, given below, the vector forbiddenColors is of size C_{max}.

The following result is straightforward.

GreedyPartialD2Coloring($G_b = (V_1, V_2, E)$)

for each $v \in V_2$ **do**
  for each $u \in N_1(v)$ **do**
    for each colored vertex $w \in N_1(u)$ **do**
      forbiddenColors(color(w)) $= v$
    end-for
  end-for
  color(v) $= \min\{c :$ forbiddenColors(c) $\neq v\}$
end-for

Lemma 6.5 *GreedyPartialD2Coloring has time complexity* $O(|V_2|\overline{\delta}_1(V_2)(\overline{\delta}_1(V_1) - 1))$.

As stated earlier, a distance-1 coloring formulation for MPP1 was provided in [7] using the column intersection graph. From Lemmas 5.1 and 6.5 and noting that greedy distance-1 graph coloring is linear in the number of edges, it follows that the time required for the construction of the column intersection graph plus the computation of a (greedy) distance-1 coloring is asymptotically the same as the time required for the direct computation of a (greedy) partial distance-2 coloring. This means the two formulations are (asymptotically) comparable in terms of overall computation time.

6.3 Distance-$\frac{3}{2}$ Coloring Algorithms

Recall that a distance-$\frac{3}{2}$ coloring, which was used to model MPP2, is a distance-1 coloring where every path of length three uses at least three colors. We propose two algorithms for this problem. In finding a valid color to assign a vertex, the first algorithm visits the distance-3 neighbors of the vertex while the second algorithm visits only the distance-2 neighbors. In both algorithms the vector forbiddenColors is of size $C_{max} = min\{\Delta^2 + 1, |V|\}$. GreedyD$\frac{3}{2}$ColoringAlg1 outlines the first algorithm.

Figure 5 graphically shows the decision made during one of the $|V|$ steps of GreedyD$\frac{3}{2}$ColoringAlg1. The root of the tree corresponds to the vertex v to be colored at the current step. The neighbors of v that are one, two, and three edges away are represented by the nodes at level u, w, and x, respectively. Each tree node corresponds to many vertices of the input graph. A darkly shaded node signifies that the vertex is already colored. The forbidden colors are marked by an f and '?' indicates that whether the color is forbidden or not depends on the color used at level x. The correspondence between the figure and Lines 1, 2 and 3 of the algorithm is

$\mathsf{GreedyD\frac{3}{2}ColoringAlg1}(G = (V, E))$

```
for each v ∈ V do
    for each u ∈ N_1(v) do
        if u is colored                                   (1)
            forbiddenColors(color(u)) = v
        for each colored vertex w ∈ N_1(u) do
            if u is not colored                           (2)
                forbiddenColors(color(w)) = v
            else
                for each colored vertex x ∈ N_1(w), x ≠ u do
                    if (color(x) == color (u))            (3)
                        forbiddenColors(color(w)) = v
                        break
                    end-if
                end-for
            end-if
        end-for
    end-for
    color(v) = min{c : forbiddenColors(c) ≠ v}
end-for
```

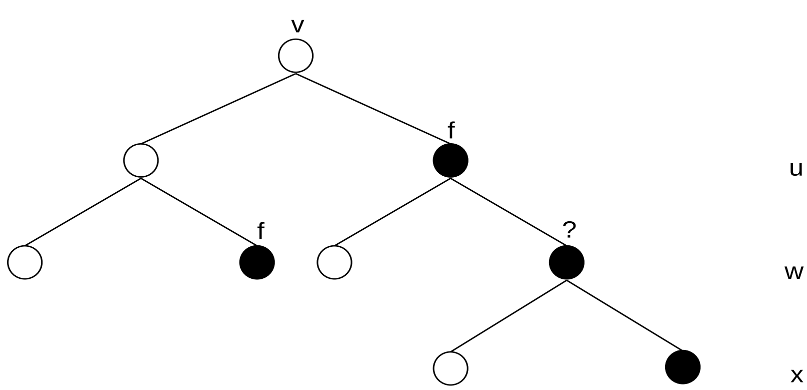

Figure 5: Visualizing a step in GreedyD$\frac{3}{2}$ColoringAlg1

obvious.

Notice that in Line 2 of the algorithm, the color of the vertex w in the path (v, u, w) where u is not yet colored is forbidden for vertex v. Later on, when vertex u is colored, the test in Line 1 ensures that u gets a color different from both v and w, making the path use three different colors. Had the requirement in Line 2 not been imposed, a situation in which a path (v, u, w, x) is two-colored could arise. Thus from its construction, the output of GreedyD$\frac{3}{2}$ColoringAlg1 is a valid distance-$\frac{3}{2}$ coloring. The amount of work done in each step of the algorithm is proportional to $d_3(v)$. Thus we get the following result.

Lemma 6.6 *GreedyD$\frac{3}{2}$ColoringAlg1 finds a distance-$\frac{3}{2}$ coloring in time $O(|V|\overline{\delta}_3)$.*

We shall now present the second distance-$\frac{3}{2}$ coloring algorithm in which the coloring at each step is obtained by considering only the distance-2 neighborhood (in contrast to distance-3 neighborhood of the previous case). The idea behind the algorithm (formulated in terms of matrices) was first suggested by Powell and Toint [26].

Recall that distance-$\frac{3}{2}$ coloring is a relaxed distance-2 coloring. As an illustration, suppose v, u, w, x is a path in a graph. A coloring ϕ in which $\phi(v) = \phi(w) = 2$, $\phi(u) = 1$ and $\phi(x) = 3$ is a valid distance-$\frac{3}{2}$ (but not distance-2) coloring on this path.

One way of relaxing the requirement for distance-2 coloring so as to obtain a distance-$\frac{3}{2}$ coloring is to let two vertices at distance of exactly two units from each other share a color as long as the vertex in between them has a color of lower value. More precisely, let v, u, w be a path in G and suppose v and u are colored and we want to determine the color of w. We allow $\phi(w)$ to be equal to $\phi(v)$ as long as $\phi(u) < \phi(v)$. To see that this coloring can always be extended to yield a valid distance-$\frac{3}{2}$ coloring, consider the path v, u, w, x, an extension of path v, u, w in one direction. Now, since $\phi(w) = \phi(v) > \phi(u)$, we cannot let $\phi(x)$ be equal to $\phi(u)$. Obviously, $\phi(x)$ should be different from $\phi(w)$, otherwise it will not be a valid distance-1 coloring. Thus the path v, u, w, x uses three colors, ϕ is a distance-1 coloring and therefore it is a valid distance-$\frac{3}{2}$ coloring. The algorithm that makes use of this idea is given in GreedyD$\frac{3}{2}$ColoringAlg2.

Clearly, the runtime of GreedyD$\frac{3}{2}$ColoringAlg2 is $O(|V|\overline{\delta}_2)$. Notice, however, that GreedyD$\frac{3}{2}$ColoringAlg1 may use smaller number of colors than GreedyD$\frac{3}{2}$ColoringAlg2. For instance, Figure 6 shows an example where the first algorithm uses three colors in coloring the vertices in their alphabetical order while the second one uses four in doing the same.

```
GreedyD3/2ColoringAlg2(G = (V, E))

for each v ∈ V do
    for each u ∈ N1(v) do
        if u is colored
            forbiddenColors(color(u)) = v
        for each colored vertex w ∈ N1(u) do
            if u is not colored
                forbiddenColors(color(w)) = v
            else
                if (color(w) < color(u))
                    forbiddenColors(color(w)) = v
            end-if
        end-for
    end-for
    color(v) = min{c : forbiddenColors(c) ≠ v}
end-for
```

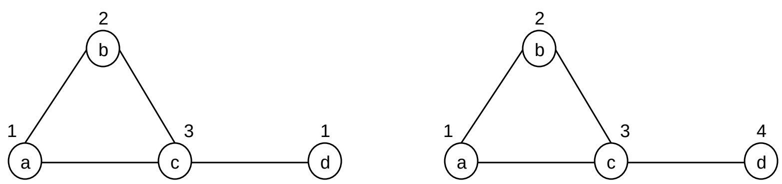

Figure 6: GreedyD$\frac{3}{2}$ColoringAlg1 vs. Alg2

6.4 Acyclic Coloring Algorithms

Here we present an effective algorithm called GreedyAcyclicColoring to find an approximate solution for the acyclic coloring problem (GCP3).

The basic idea in our algorithm is to detect and 'break' a two-colored cycle while an otherwise distance-1 coloring of the graph proceeds. Specifically, the algorithm colors the vertices of a graph $G = (V, E)$ while making a Depth First Search (DFS) traversal. Recall that a *back-edge* in the *DFS-tree* of an undirected graph defines a unique cycle. For more information on DFS, refer to the book [11].

In GreedyAcyclicColoring, the *DFS-tree* $T(G)$ of the graph G is implicitly constructed as the algorithm proceeds. Let $\phi(v)$ denote the color of vertex v and $s(v)$ denote the order in which v is first visited in the DFS traversal of G ($1 \leq s(v) \leq |V|$). The root r of $T(G)$ has $s(r) = 1$. Further, let $p(v)$ be a pointer to the *parent* of v in $T(G)$, and $l(v)$ be a pointer to the lowest ancestor of v in $T(G)$ such that $\phi(l(v)) \neq \phi(v)$ and $\phi(l(v)) \neq \phi(p(v))$.

The latter pointer will be used in the detection of a two-colored cycle. In particular, the algorithm proceeds in such a way that the path in $T(G)$ from v up to, but not including, $l(v)$ is two-colored. At the beginning of the algorithm, for every vertex u, $l(u)$ is set to point to null.

Consider the step in GreedyAcyclicColoring where vertex v is first visited. To start with, v is assigned the smallest color different from all of its distance-1 neighbors in G, including its parent $p(v)$ in $T(G)$. If there exists a back-edge $b = (v, w)$ in the current $T(G)$ such that $s(l(p(v))) < s(w)$ and $\phi(v) = \phi(p(p(v)))$, then this implies that the cycle corresponding to b is two-colored (see Figure 7 which shows a partial view of the DFS-tree at the step where vertex v is to be colored). To break the cycle, v is assigned a new color—the smallest color different from $\phi(p(v))$ and $\phi(p(p(v)))$—and $l(v)$ is set to point to $p(p(v))$. Otherwise, if no such back-edge exists, the color of v is declared final and $l(v)$ is updated in the following manner. If $\phi(v) \neq \phi(p(p(v)))$, then $l(v) = p(p(v))$; otherwise $l(v) = l(p(v))$.

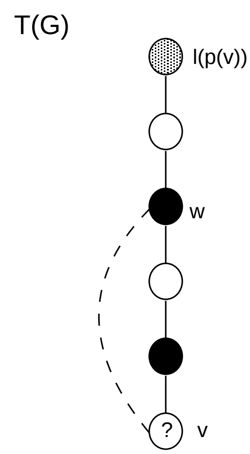

Figure 7: Visualizing a step in GreedyAcyclicColoring

Notice that the amount of work done in each DFS visit of a vertex v in the graph is proportional to the degree-1 of v. This makes GreedyAcyclicColoring an $O(|E|)$-time algorithm.

We note that Coleman and Cai [4] have proposed an algorithm for the acyclic coloring problem. The idea in their algorithm is to first transform a given graph $G = (V, E)$ to a 'completed' graph $G' = (V, E')$ such that a distance-1 coloring of G' is equivalent to an acyclic coloring of G, and then use a known distance-1 coloring heuristic on G'. The construction of G' is done in the following way. Start by setting $E' = E$; visit the vertices in V in a *predefined order*; at each step i, if vertex v_i is adjacent to vertices v_j and

v_k such that both v_j and v_k are ordered before v_i then add the edge (v_j, v_k) to E'.

Our approach differs from that of Coleman and Cai in at least two ways. First, the graph G' used in the latter approach may require substantially more storage space than the original graph G used in our approach. Second, an edge in $E' \setminus E$ in the latter approach may actually be redundant. For example, a distance-1 coloring of an odd-length cycle in G uses at least three colors and hence is a valid acyclic coloring whereas the Coleman and Cai approach adds one redundant edge to the cycle.

6.5 Bicoloring Algorithms

Here we consider problems GCP4 and GCP5, introduced in Sections 3.4 and 3.5, respectively.

Recall that in a distance-$\frac{3}{2}$ p-bicoloring, some of the vertices are assigned the neutral color 0. We make the following crucial observation which helps us identify a possible set of such vertices. The observation is a direct consequence of Condition 2 of Definition 3.8.

Observation 6.7 *Let $G_b = (V_1, V_2, E)$ be a bipartite graph and $\phi : [V_1, V_2] \to \{0, 1, \ldots, p\}$ be a distance-$\frac{3}{2}$ bicoloring of G_b. Then,*

- *the set $C = \{v : \phi(v) \neq 0\}$ is a* ***vertex cover*** *in G_b, and*
- *the set $I = \{v : \phi(v) = 0\}$ is an* ***independent set*** *in G_b.*

One consequence of Observation 6.7 is that $|I| + |C| = |V_1| + |V_2|$. Thus, minimizing the cardinality of the vertex cover C corresponds to maximizing the cardinality of the independent set I.

6.5.1 An algorithm for GCP4

In light of Observation 6.7, we suggest GCP4Algorithm as a scheme for solving the coloring problem GCP4.

GCP4Algorithm$(G_b = (V_1, V_2, E))$

1. Find a vertex cover C in G_b.
2. Assign the vertices in the set $I = (V_1 \cup V_2) \setminus C$ the color 0.
3. Color the vertices in C such that the result is a distance-3/2 bicoloring of G_b.

In Step 1 of GCP4Algorithm, any vertex cover can be used. However, the choice of the vertex cover affects the subsequent coloring in Step 3, both in terms of number of colors used and coloring time spent. To reduce the coloring time in Step 3, the size of the vertex cover should be minimized. Furthermore, minimizing the potential number of colors to be used imposes an additional requirement: the vertex cover should include those vertices from V_1 and V_2 with relatively high number of distance-1 neighbors. (Recall the introductory discussion in Section 2.2 used to motivate the need for a two-dimensional partition: matrices with a few dense rows and columns benefit from a two-dimensional partition.)

In a bipartite graph, a minimum cardinality vertex cover can be obtained via finding a *maximum matching* in polynomial time [22, 28]. In fact, it can be computed practically in effectively linear time in the number of edges [24].

Once Steps 1 and 2 are carried out, Step 3 can be done by a suitable adaptation of GreedyD$\frac{3}{2}$ColoringAlg1. GreedyD$\frac{3}{2}$BiColoring, given here only pictorially, is such an adaptation. One of the differences between the coloring and bicoloring algorithms is that in the latter case, two disjoint set of colors are used in coloring the vertices in V_1 and V_2 of the bipartite graph $G_b = (V_1, V_2, E)$. Another difference is that at a step of the bicoloring algorithm where v is colored, a vertex within the distance-3 neighborhood of v may be one of *three* types: it is colored with a positive value, it is colored with 0, or it is not yet colored. The choice of color for v thus needs to consider these three options.

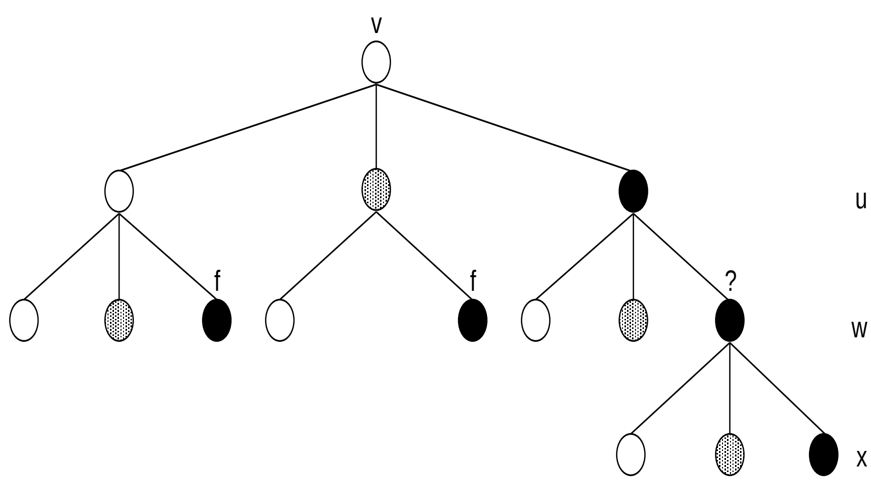

Figure 8: Visualizing a step in GreedyD$\frac{3}{2}$BiColoring

Figure 8 shows a visual presentation of the ith step of GreedyD$\frac{3}{2}$BiColoring. Note the similarity with Figure 5. Notice also that, in choosing a color for vertex v in $G_b = (V_1, V_2, E)$, since the colors for vertices in V_1 and V_2 are drawn from two disjoint sets, we need only consider colors of vertices two edges away from v. In the figure, dark shaded nodes correspond to colored vertices, light shaded nodes show vertices with color zero, and unshaded nodes correspond to uncolored vertices. Observe that the node with color 0 at level u has only two children. The colors of the vertices in the nodes marked by an f indicate forbidden colors and whether the color at the node marked by '?' is forbidden or not depends on the color used at node x: if $\phi(u) = \phi(x)$, $\phi(w)$ is forbidden, otherwise, it is not.

The time complexity of GreedyD$\frac{3}{2}$BiColoring is $O((|V_1| + |V_2|)\overline{\delta}_3)$, which is also the overall time complexity of GCP4Algorithm assuming that step 1 is done using a greedy algorithm that is linear in the number of edges.

Notice that a partial distance-2 coloring of G_b on V_2 is a just special case of the scheme GCP4Algorithm. To see this, consider the trivial choice of vertex cover $C = V_2$ in Step 1. This implies that, in Step 2, the vertices in the set $I = V_1$ will be colored with zero. By Condition 3 of Definition 3.8, vertices adjacent to a vertex colored with zero are required to be assigned different colors. Thus, the result is effectively a partial distance-2 coloring of G_b on V_2.

We note that Hossain and Steihaug [17] and Coleman and Verma [9] have each proposed an algorithm for GCP4. These algorithms can be interpreted in light of GCP4Algorithm. The algorithm of Hossain and Steihaug (HS) *implicitly* finds a vertex cover while the coloring of the graph proceeds. Using our terminology, the vertices that remain uncolored at the end of the HS-algorithm form an independent set in the graph and can thus be assigned the neutral color 0.

The algorithm of Coleman and Verma uses a preprocessing step to identify the rows and columns of the underlying matrix that eventually need to be colored with positive values. The preprocessing step uses a non-straightforward matrix-based procedure. It appears that the procedure effectively produces a small sized vertex cover (however, this is not clearly stated in the paper). After the preprocessing step, a certain 'column intersection' graph, adapted to the distance-$\frac{3}{2}$ bicoloring requirements, is constructed to finally use known distance-1 coloring heuristics on the resulting graph.

6.5.2 An algorithm for GCP5

Similarly, by virtue of Observation 6.7, the approach we suggest for solving the acyclic bicoloring problem is given in GCP5Algorithm.

GCP5Algorithm($G_b = (V_1, V_2, E)$)

1. Find a vertex cover C in G_b.
2. Assign the vertices in the set $I = (V_1 \cup V_2) \setminus C$ the color 0.
3. Color the vertices in C such that the result is an acyclic bicoloring of G_b.

7 Conclusion

We have studied the efficient estimation of sparse Jacobian and Hessian matrices using FD and AD techniques. We considered methods that rely on a one-dimensional as well as a two-dimensional partition to be used in an evaluation based on a direct or a substitution scheme. We introduced partial matrix estimation problems in distinction from full matrix estimation problems. In doing so, we developed a unified graph theoretic framework to cope with a variety of complex matrix partitioning problems.

At the basis of our graph problem formulations lies a robust graph representation of the sparsity structure of a matrix: a nonsymmetric matrix is represented by its bipartite graph and a symmetric matrix by its adjacency graph.

We showed that the distance-2 graph coloring problem is a generic model for the various one-dimensional matrix partitioning problems.

Our unified graph theoretic approach enabled us to provide some fresh insight into the matrix problems and as a result we developed several simple and effective algorithms. Our emphasis has been on greedy algorithms. Other algorithmic techniques need to be explored in the future. For example, it could be interesting to find a distance-2 coloring algorithm that uses asymptotically the same time as the greedy algorithm discussed in this paper and balances the number of vertices in each color class. Finding a random color, rather than the smallest color, from an allowable set could be an idea to consider in this regard.

In the case of two-dimensional partition problems, based on the known relationship to graph bicoloring, we argued that finding a 'small'-size vertex cover as a preprocessing step contributes to making the overall computation more efficient. A more precise characterization of the 'optimum' vertex cover required is a worthwhile issue.

We have not developed any special algorithms for the restricted coloring problems arising in partial matrix estimation. The ideas used in our algorithms for the coloring problems in full matrix estimation can be adapted to the restricted cases by observing the particular coloring conditions.

In general, most of the algorithms in the literature for solving the coloring problems considered in this paper rely on first transforming the input graph $G = (V, E)$ to some denser graph $G' = (V, E')$, $E' \supseteq E$, such that a distance-1 coloring of G' is equivalent to the particular coloring problem on G. In contrast, the algorithms proposed in this paper solve the particular coloring problem directly on G. As has been argued, the main advantages offered by our approach are the possibility to mix-and-match methods, less storage space requirement, and ease of developing flexible software.

One of the motivations for the current study has been the need for the development of parallel algorithms for solving partitioning problems in large-scale PDE-constrained optimization contexts. In a recent work [13], we have shown some parallel algorithms (using the shared-memory programming model) for the distance-2 and distance-$\frac{3}{2}$ coloring problems. Our results, theoretical as well as experimental, were promising. We believe that this study lays a foundation for further work on the development and implementation of not only shared-memory but also distributed-memory parallel algorithms.

Acknowledgement We thank Trond Steihaug for interesting discussions.

References

[1] G. Agnarsson, R. Greenlaw, and M. M. Halldórsson. On powers of chordal graphs and their colorings. *Congress Numerantium*, 100:41–65, 2000.

[2] G. Agnarsson and M. M. Halldórsson. Coloring powers of planar graphs. In *11th. Ann. ACM-SIAM symp. on Discrete Algorithms*, pages 654–662, 2000.

[3] O. Axelsson and U. Nävert. On a graphical package for nonlinear partial differential equation problems. In B. Gilchrist, editor, *Proceedings of IFIP Congress 77*, Information Processing, pages 103–108. North-Holland, 1977.

[4] T. F. Coleman and J. Cai. The cyclic coloring problem and estimation of sparse Hessian matrices. *SIAM J. Alg. Disc. Meth.*, 7(2):221–235, April 1986.

[5] T. F. Coleman, B. Garbow, and J. J. Moré. Software for estimating sparse Jacobian matrices. *ACM Trans. Mathematical Software*, 10:329–347, 1984.

[6] T. F. Coleman, B. Garbow, and J. J. Moré. Software for estimating sparse Hessian matrices. *ACM Trans. Mathematical Software*, 11:363–377, 1985.

[7] T. F. Coleman and J. J. Moré. Estimation of sparse Jacobian matrices and graph coloring problems. *SIAM J. Numer. Anal.*, 20(1):187–209, February 1983.

[8] T. F. Coleman and J. J. Moré. Estimation of sparse Hessian matrices and graph coloring problems. *Math. Program.*, 28:243–270, 1984.

[9] T. F. Coleman and A. Verma. The efficient computation of sparse Jacobian matrices using automatic differentiation. *SIAM J. Sci. Comput.*, 19(4):1210–1233, July 1998.

[10] G. Corliss, C. Faure, A. Griewank, L. Hascoét, and U. Naumann. *Automatic Differentiation of Algorithms: From Simulation to Optimization*. Springer, New York, 2002.

[11] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. Massachusetts Institute of Technology, 1990.

[12] A. R. Curtis, M. J. D. Powell, and J. K. Reid. On the estimation of sparse Jacobian matrices. *J. Inst. Math. Appl.*, 13:117–119, 1974.

[13] A.H. Gebremedhin, F. Manne, and A. Pothen. Parallel distance-k coloring algorithms for numerical optimization. In B. Monien and R. Feldmann, editors, *Euro-Par 2002 Parallel Processing*, volume 2400 of *LNCS*, pages 912–921. Springer Verlag, 2002.

[14] A. Griewank and G. F. Corliss. *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*. SIAM, Philadelphia, 1991.

[15] B. Grünbaum. Acyclic colorings of planar graphs. *Israel J. Math.*, 14:390–408, 1973.

[16] S. Hossain. On the computation of sparse Jacobian matrices and Newton steps. Technical Report 146, Department of Informatics, University of Bergen, Norway, March 1998.

[17] S. Hossain and T. Steihaug. Computing a sparse Jacobian matrix by rows and columns. *Optimization Methods and Software*, 10:33–48, 1998.

[18] S. Hossain and T. Steihaug. Reducing the number of AD passes for computing a sparse Jacobian matrix. In G. Corliss, C. Faure, A. Griewank, L. Hascoét, and U. Naumann, editors, *Authomatic Differentiation of Algorithms: From Simulation to Optimization*, pages 263 – 270. Springer, 2002.

[19] D. A. Knoll and D. E. Keyes. Jacobian-free Newton-Krylov methods: A survey of appoaches and applications. Available at http://www.math.odu.edu/˜ keyes/.

[20] S. O. Krumke, M. V. Marathe, and S.S. Ravi. Approximation algorithms for channel assignment in radio networks. In *Dial M for Mobility*, Dallas, Texas, October 1998.

[21] Y. Lin and S. S. Skiena. Algorithms for square roots of graphs. *SIAM J. Disc. Math.*, 8:99–118, 1995.

[22] L. Lovász and M.D. Plummer. *Matching Theory*. North-Holland, Amsterdam, 1986.

[23] S. T. McCormick. Optimal approximation of sparse Hessians and its equivalence to a graph coloring problem. *Math. Program.*, 26:153–171, 1983.

[24] R. Motwani. Average-case analysis of algorithms for matchings and related problems. *Journal of the ACM*, 41:1329–1356, 1994.

[25] G. N. Newsam and J. D. Ramsdell. Estimation of sparse Jacobian matrices. *SIAM J. Alg. Disc. Meth.*, 4:404–418, 1983.

[26] M. J. D. Powell and P. L. Toint. On the estimation of sparse Hessian matrices. *SIAM J. Numer. Anal.*, 16(6):1060–1074, December 1979.

[27] V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.

[28] L. A. Wolsey. *Integer Programming*. Wiley-Interscience Publication, New York, USA, 1998.