REPORTS
IN
INFORMATICS

ISSN 0333-3590

Graph Coloring in Optimization Revisited

Assefaw H. Gebremedhin, Fredrik Manne,
Alex Pothen

REPORT NO 226 January 2002

Department of Informatics

UNIVERSITY OF BERGEN

Bergen, Norway

This report has URL `http://www.ii.uib.no/publikasjoner/texrap/ps/2002-226.ps`

Reports in Informatics from Department of Informatics, University of Bergen, Norway, is available at `http://www.ii.uib.no/publikasjoner/texrap/`.

Requests for paper copies of this report can be sent to:

Department of Informatics, University of Bergen, Høyteknologisenteret,
P.O. Box 7800, N-5020 Bergen, Norway

Graph Coloring in Optimization Revisited

Assefaw Hadish Gebremedhin[*] Fredrik Manne[†] Alex Pothen[‡]

Abstract

We revisit the role of graph coloring in modeling problems that arise in efficient estimation of large sparse Jacobian and Hessian matrices using both finite difference (FD) and automatic differentiation (AD) techniques, in each case via direct methods. For Jacobian estimation using column partitioning, we propose a new coloring formulation based on a bipartite graph representation. This is compared with the known *column intersection* graph based formulation. Moreover, we analyze existing graph coloring formulations for efficient estimation of Hessians using FD and Jacobians using AD. We identify distance-2 coloring as a unifying general problem across the different cases. We also present and analyze greedy graph coloring heuristics corresponding to each case.

Key words: Sparsity, symmetry, Jacobians, Hessians, finite differences, automatic differentiation, matrix partition problems, graph coloring problems, NP-completeness, heuristics

1 Introduction

Optimization algorithms often require the estimation of the unsymmetric Jacobian matrix $F'(x)$ of a mapping $F : R^n \to R^m$ or the symmetric Hessian matrix $\nabla^2 f(x)$ of a function $f : R^n \to R$. In large scale problems these matrices are typically sparse, a property that can be exploited in the efficient determination of the entries of the matrix.

There are two widely used numerical techniques in the estimation of Jacobian and Hessian matrices: *finite differencing* (FD) and *automatic differentiation* (AD). In both techniques the scheme followed in the evaluation of the matrices can be either a *direct* or an *indirect* (*substitution*) method. The former corresponds to a diagonal system and the latter to a triangular system of equations. There is a trade-off in using these methods. A substitution method is less restrictive and thus is expected to require fewer function evaluations but it may be subject to approximation difficulties and numerical instability. A direct method, on the other hand, is more restrictive, implying that more function evaluations are required, but the computation of the matrix entries is straightforward and numerically stable. Moreover, direct methods offer straightforward parallelization, whereas substitution methods have less parallelism since the function values computed have to be communicated from one processor to another for other function values to be obtained.

[*]Dept. of Informatics, University of Bergen, N-5020 Bergen, Norway. assefaw@ii.uib.no

[†]Dept. of Informatics, University of Bergen, N-5020 Bergen, Norway. fredrikm@ii.uib.no

[‡]Dept. 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

Since the early 70's efficient computation of Jacobian and Hessian matrices using FD techniques have been studied extensively [2, 3, 4, 6, 8, 12, 13]. Curtis, Powell and Reid [6] were the first to observe that *sparsity* can be exploited in reducing the number of function evaluations required in the estimation of Jacobian matrices. In the Hessian case, Powell and Toint [13] were the first to exploit *symmetry.* Several authors later extended these works and studied the matrix estimation problems from a graph theoretic viewpoint. Coleman and Moré [3], the first to adopt this view, showed that the estimation of a sparse Jacobian matrix using a direct method can be seen as a *distance-1 graph coloring* problem. Later, McCormick [12] showed that the approximation of a Hessian matrix using a direct method is equivalent to a *distance-2 graph coloring* problem. McCormick's graph theoretic formulation ignores symmetry and a formulation that overcomes this was later suggested by Coleman and Moré [4] and subsequently extended to indirect methods in [2], in each case resulting in new variants of the graph coloring problem.

The literature on matrix estimation using the relatively more recent technique AD is less extensive. Examples of graph theoretic studies in this area include the work on Jacobian matrix estimation by Coleman and Verma [5] and by Hossain and Steihaug [10].

In both the FD and AD cases, a graph theoretic approach has proved to be superior to a matrix oriented one for several reasons. The approach often provides better insight into the problem and the resulting graph problem becomes easier to analyze thereby leading to good algorithms.

The first objective of the current paper is to *unify* existing graph theoretic formulations of matrix partition problems that arise in direct estimation of symmetric and unsymmetric matrices using both FD and AD techniques. This is done to identify a *general* problem and show that the different cases can be seen as relaxed variants thereof. The second objective is to use this insight to develop *effective algorithms* for the different cases based on the general formulation. The unified view, in particular, eases the task of developing efficient parallel algorithms for the various problems, a task that has been a primary motivation for the current work.

Specifically, in this paper, we show that the *distance-2 graph coloring* problem, and its relaxed variants, are equivalent formulations of the different matrix partition problems. For the column partitioning problem of an unsymmetric matrix, it is well known that the *column intersection* graph representation of the matrix results in the equivalent *distance-1 graph coloring* problem [3]. By using a *bipartite* graph representation, we show that the problem can be seen as a distance-2 coloring problem, thereby fitting into the desired unifying framework. The column intersection graph representation of a matrix A requires the (indirect) computation and storage of A^TA whereas our graph representation requires no further computation and no more space than that required to efficiently store A. When A is of very large size, the storage requirement of A^TA can be a cause for concern. Moreover, unlike column intersection graph, bipartite graph representation is *robust*: the representation only captures the sparsity structure of the matrix and is decoupled from the eventual technique used to manipulate it.

Motivated by the effectiveness of existing distance-1 coloring heuristics, we show that some of the ideas used in these heuristics can be extended to the distance-2 coloring case. We also show very simple ways of adapting a distance-2 coloring algorithm to handle relaxed variants of the distance-2 coloring problem.

The rest of this paper is organized as follows. In Section 2 we introduce and formally state the matrix partition problems of our concern and in Section 3 we develop their graph theoretic characterizations. In Section 4 we first review some preliminary facts related to the coloring problems and then present and analyze several known and new greedy heuristic algorithms for distance-2 coloring and its relaxed variants. Finally, we conclude the paper in Section 5 with some remarks and point out avenues for further work.

2 Matrix Problems

2.1 Finite Differences (FD) and Partition Problems

Given a continuously differentiable function $F : R^n \to R^m$, the Jacobian of F at x is the $m \times n$ matrix whose (i, j)th entry is $F'(x)_{ij} = \frac{\partial f_i}{\partial x_j}(x)$, where $f_1(x), f_2(x), \ldots, f_m(x)$ are the components of $F(x)$. Let A denote the Jacobian matrix $F'(x)$. Then an estimation of the jth column of A can be obtained from the *forward difference* approximation,

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

where e_j is the jth Cartesian basis vector and h is a positive step-length. Assuming that $F(x)$ is already evaluated, an approximation to a_j can be obtained by one extra evaluation of F. If the kth entry of a_j (element a_{kj}) is known to be zero, then we would like to avoid the computation of the kth component of F at $x + he_j$. The problem is then to utilize the known sparsity structure in estimating A using (1) under the assumption that evaluating $F(x)$ is more efficient than evaluating the components $f_i(x)$, $1 \leq i \leq m$, separately. Notice that by using (1), A can be estimated trivially using n extra function evaluations. The goal here is to estimate A with as few function evaluations as possible.

Let d_i be the sum of a subset of vectors from the set $\{e_1, e_2, \ldots, e_n\}$ and be scaled by h. The problem of estimating a Jacobian matrix using FD can thus be stated as: Given the sparsity structure of an $m \times n$ matrix A, find vectors $d_1, d_2, \ldots, d_p$ such that $Ad_1, Ad_2, \ldots, Ad_p$ determine (all nonzero unknowns in)A uniquely.

Specifying $Ad_1, Ad_2, \ldots, Ad_p$ gives rise to a system of mp linear equations in the unknowns a_{ij}. If the choice of $d_1, d_2, \ldots d_p$ is such that the system of equations that defines the unknowns can be ordered to a triangular form, then from $Ad_1, Ad_2, \ldots, Ad_p$ the nonzero entries of A (the unknowns) can be determined via *substitution*. If, on the other hand, the choice of $d_1, d_2, \ldots, d_p$ is such that the system of equations that defines the unknowns can be ordered to diagonal form, then $Ad_1, Ad_2, \ldots, Ad_p$ determine A *directly*. In such a case, for each nonzero a_{ij} there is a d in $\{d_1, d_2, \ldots, d_p\}$ such that

$$a_{ij}\delta_j = (Ad)_i,$$

where $\delta_j = h$ is the jth component of d. Notice that minimizing the number of function evaluations corresponds to minimizing p. The first problem we are concerned with is thus the following.

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

Given a twice continuously differentiable function $f : R^n \to R$, the Hessian of f at x is the $n \times n$ symmetric matrix whose (i, j)th entry is $H(x)_{ij} = \nabla^2 f(x)_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$. When ∇f is available, $\nabla^2 f$ can be approximated by applying formula (1) to the function $F = \nabla f$. In this case, the problem of estimating a Hessian matrix becomes a special case of Problem 2.1 in which the matrix A is *symmetric*.

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

Curtis et al. [6] observed that, in the direct determination of matrix A, 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, *i.e.*, the columns in the group are pairwise *structurally orthogonal*. Powell and Toint [13] later extended the method of Curtis et al. to the symmetric case and showed that the number of function evaluations (p) can be reduced further by taking symmetry into consideration. Coleman and Morè [3, 4] formalized the approaches of Curtis et al. and of Powell and Toint by introducing the concepts of *consistent* and *symmetrically consistent* partition.

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

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

Let $\{C_1, C_2, \ldots, C_p\}$ be a consistent partition of the columns of A. The nonzero entries of A can be determined with p evaluations of Ad by associating with each group C_k a vector d having components $\delta_j \neq 0$ if a_j belongs to C_k, and $\delta_j = 0$ otherwise. Then,

$$Ad = \sum_{a_j \in C_k} \delta_j a_j.$$

If $a_{ij} \neq 0$ and column $a_j \in C_k$ then,

$$(Ad)_i = \delta_j a_{ij},$$

and thus a_{ij} is determined. Every nonzero element of A is directly determined in this way.

If A is symmetric, a symmetrically consistent partition is sufficient to determine A directly. This is because, if column a_j is the only column in its group with a nonzero in row i then a_{ij} can be determined as discussed above, and if column a_i is the only column in its group with a nonzero in row j then a_{ji} can be determined. Hence, every nonzero of A is directly determined with p evaluations of Ad.

If A is symmetric, but a consistent partition (ignoring symmetry) is used, 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 non-zeros of the matrix and determining the other half by symmetry is preferable both in terms of exploiting and enforcing symmetry.

Using Definitions 2.3 and 2.4, Problems 2.1 and 2.2 can now be stated as partition problems as follows.

Problem 2.5 (*UNSYMCOLPART*) *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 (*SYMCOLPART*) *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.*

2.2 Automatic Differentiation and Partition Problems

Automatic Differentiation (AD) is a chain rule based technique for evaluating the derivatives of functions defined by computer programs written in some high level language. 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. Due to space limitation, we do not give the technical details of the technique here but refer the interested reader to [9].

As in the FD setting, the criterion for partitioning columns into groups in the AD setting is structural orthogonality. In particular, one can use the forward mode to compute a group of columns from the matrix-vector product Ad, where the vector d has non-zeros 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 estimated by the vector-matrix product $d^T A$. This shows that one can potentially take advantage of the sparsity both in columns and in rows.

One way of exploiting sparsity in both rows and columns is to *separately* partition the rows and columns of the matrix and use the partition which yields the minimum number of groups. However, such an approach may not be satisfactory for matrices with a few dense rows and columns. For such matrices considering a *combined* row and column partition may be worthwhile. Hossain and Steihaug [10] studied this approach and formalized the resulting matrix partition problem and its equivalent graph problem.

Note that by considering the vectors $d_1, d_2, \ldots, d_p$ as the p columns of the $n \times p$ matrix D, Problem 2.1 can also be written as: given the structure of $A \in R^{m\times n}$, find a matrix $D \in R^{n \times p}$ such that the product AD determines A directly with the least value of p. Similarly, the problem that arises in the efficient computation of a Jacobian using both column *and* row partitions can be posed as follows.

Problem 2.7 *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 uniquely and $p = p_1 + p_2$ is minimized.*

Using the following two definitions from [10] we shall shortly restate Problem 2.7 in its partition form.

Definition 2.8 A *row-column partition* of A is a division of the columns into one set of groups $C_1, C_2, \ldots, C_{p_1}$, and the rows into another set of groups $R_1, R_2, \ldots, R_{p_2}$, such that each column and row belongs to exactly one group.

Definition 2.9 A row-column partition of A is said to be *consistent* with the direct determination of A if whenever a_{ij} is a nonzero element of A then either the group containing column j has no other column with a nonzero in row i, or the group with row i has no other row with a nonzero in column j.

Notice that given a row-column consistent partition of a matrix A using p_1 column groups and p_2 row groups, all the nonzero entries of A can be determined directly with $p = p_1 + p_2$ function evaluations (corresponding to the p_1 matrix-vector products and the p_2 vector-matrix products). 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. By requiring p to be as small as possible, we make sure that the nonzero matrix entries are determined with the minimum number of function evaluations. Problem ROWCOLPART is thus a reformulation of Problem 2.7.

Problem 2.10 (*ROWCOLPART*) *Given the sparsity structure of $A \in R^{m \times n}$, find a row-column consistent partition such that the sum of the number of column- and row-groups is minimized.*

Coleman and Verma [5] have also studied the estimation of Jacobians using AD and they identify a similar row-column partition problem. Their notion of partition differs from that in Definition 2.8 in that it partitions only a *subset* of the columns and the rows of a matrix. They use a certain *preprocessing* step to identify the columns and rows that need to be included in the respective subsets to be partitioned such that the matrix entries can be determined uniquely.

In the case of estimating a Hessian matrix using AD, we note that, due to symmetry, a row-column partition of the matrix does not make sense. In other words, using either the forward or reverse mode of AD alone suffices. Table 1 summarizes the partition problems considered so far.

	FD/AD using one mode	AD using both modes
Jacobian	UNSYMCOLPART	ROWCOLPART
Hessian	SYMCOLPART	-

Table 1: Summary of partition problems in direct methods

3 Graph Problems

The problems summarized in Table 1 are combinatorial and have been successfully solved by finding equivalent graph theoretic formulations [2, 3, 4, 5, 10, 12]. Our objective here is to unify these approaches and identify a general problem. A crucial step in finding a graph theoretic formulation of each of these matrix partition problems is the determination of the graph used to model the matrix. In Section 3.1 we describe our graph representations of unsymmetric and symmetric matrices and in Section 3.2 we use these representations to derive equivalent graph problems. An important feature of our representations is that a graph can be readily extracted from the sparsity structure of a matrix, irrespective of the eventual method used to determine its entries. Moreover, our representations are attractive both in terms of ease of construction and storage space. The relative merits of our graph formulation for the unsymmetric case compared to a column intersection graph based formulation is discussed in Section 3.3.

3.1 Graph Representation

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*. Thus $E \subset \{(u, v) : u \neq v, u, v \in V\}$. 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 k or less; otherwise they are called *non-distance-k neighbors*.

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

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 $1 \leq i \leq m$, $1 \leq j \leq n$.

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

If A is symmetric and all its diagonal elements are nonzero, the *adjacency* graph $G(A)$ is an alternative, symmetry-exploiting, representation of A. In $G(A) = (V, E)$, a vertex v_i in V is obtained by 'merging' row vertex r_i and column vertex a_i of the corresponding bipartite graph $G_b(A)$. Specifically, 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 $1 \leq i \leq n$, $1 \leq j \leq n$, $i \neq j$. Notice 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 $|V|$ in $G(A)$ is n and the number of edges $|E|$ 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 usually is positive definite [2].

In this paper, in order not to use too many symbols, we use the same notation, say a_i, both when referring to the ith column of matrix A and the corresponding vertex in $G(A)$ or $G_b(A)$.

3.2 Graph Coloring

A *distance-k p-coloring* (or (k, p)-coloring for short) of a graph $G = (V, E)$ is a mapping $\phi : V \to \{1, 2, \ldots, p\}$ such that $\phi(u) \neq \phi(v)$ whenever u and v are distance-k neighbors. 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 $\phi : W \to \{1, 2, \ldots, p\}$ 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*, denoted by $\chi_k(G)$.

The *distance-k graph coloring problem* (DkGCP), stated below, asks for an optimal (k, p)-coloring of a graph.

Problem 3.1 *Given a graph $G = (V, E)$ and an integer k, find a (k, p)-coloring of G such that p is minimized.*

Notice that a (k, p)-coloring of G induces a partition of the vertex set V into p groups (also called *color classes*) $U_1, U_2, \ldots, U_p$, 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. A natural question that arises is then how the partition problems in Table 1 could be related to the DkGCP for some k.

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

Lemma 3.2 *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) in 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; an identical argument can be used to prove the case for rows. Assume that vertices a_i and a_j in $G_b(A)$ 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 the 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 converse, 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 3.3 *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 3.2 and 3.3 provide a partial answer to our question regarding the relationship between the problems in Table 1 and DkGCP. By Lemma 3.2, finding a consistent partition of the columns of a matrix A (see Definition 2.3) 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 3.4 *Let A be an unsymmetric 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 3.4, Problem UNSYMCOLPART is equivalent to the following problem.

Problem 3.5 (*PartialD2COLOR*) *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, exploiting sparsity along the rows instead of along the columns might be preferable. In other words, the matrix problem one needs to solve is Problem 2.5 applied on A^T, instead of A. In such cases, our graph formulation becomes handy: the equivalent problem is solving distance-2 coloring on the vertex set V_1, instead of V_2.

We now look for an analogous graph formulation of Problem SYMCOLPART. Consider a symmetric matrix A with nonzero diagonal elements and let a_{ij}, $i \neq j$, be a nonzero element. By Definition 2.4, in a symmetrically consistent partition of A, (1) since both a_{ii} and a_{jj} are nonzero, a_i and a_j should belong to two different groups, and (2.1) for any nonzero a_{ki}, $k \neq i, j$, a_j and a_k should belong to two different groups, or (2.2) for any nonzero a_{jl}, $l \neq i, j$, a_i and a_l should belong to two different groups.

Coleman and Moré [4] gave an equivalent graph characterization of the aforementioned conditions using the concept of *distance-$\frac{3}{2}$ coloring*[1], defined below.

Definition 3.6 *A mapping $\phi : V \to \{1, 2, \ldots, p\}$ is a $(\frac{3}{2}, p)$-coloring of a graph $G = (V, E)$ if ϕ is $(1, p)$-coloring of G and every path of length 3 uses at least 3 colors.*

The name distance-$\frac{3}{2}$ is chosen to reflect that a $(\frac{3}{2}, p)$-coloring is a relaxed $(2, p)$-coloring, and a restricted $(1, p)$-coloring. To see this, observe that a $(1, p)$-coloring requires 2 colors for every path of length 1, a $(2, p)$-coloring requires 3 colors for every path of length 2, and $(\frac{3}{2}, p)$-coloring is a $(1, p)$-coloring further restricted to require 3 colors for every path of length 3.

The following theorem formalizes the connection between symmetrically consistent partition and distance-$\frac{3}{2}$ coloring. The proof is omitted as it follows directly from the discussion preceding the definition of $(\frac{3}{2}, p)$-coloring.

Theorem 3.7 [Coleman and Moré [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 a $(\frac{3}{2}, p)$-coloring of $G(A)$ if and only if ϕ induces a symmetrically consistent partition of the columns of A.*

By Theorem 3.7, the following problem is equivalent to Problem SYMCOLPART.

Problem 3.8 ($D\frac{3}{2}$*COLOR*) *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.*

The last matrix problem we want to address is Problem ROWCOLPART, dealing with row-column consistent partition of an unsymmetric matrix. Observe the similarity between the definitions of symmetrically consistent partition (Definition 2.4) and row-column consistent partition (Definition 2.9). Of course, in the latter case, a group always consists either entirely of columns or rows. Using similar arguments as that used in the case of symmetrically consistent partition, one can arrive at a graph theoretic characterization of row-column consistent partition using the concept of *distance-$\frac{3}{2}$ bicoloring*, defined below. Distance-$\frac{3}{2}$ bicoloring, albeit with a different name, was introduced by Hossain and Steihaug [10] and a related concept is used by Coleman and Verma [5].

Definition 3.9 *A mapping $\phi : V \to \{1, 2, \ldots, p\}$ is a $(\frac{3}{2}, p)$-bicoloring of a bipartite graph $G_b = (V_1, V_2, E)$ if every path of length 3 uses at least 3 colors and $\{\phi(v_1) : v_1 \in V_1\} \cap \{\phi(v_2) : v_2 \in V_2\} = \emptyset$.*

The connection between row-column consistent partition and graph coloring is thus given by the next theorem.

[1] They used the term *path-coloring*

Theorem 3.10 [Hossain and Steihaug [10]] *Let A be an unsymmetric matrix and $G_b(A) = (V_1, V_2, E)$ be its bipartite graph representation. A mapping ϕ is a $(\frac{3}{2}, p)$-bicoloring of $G_b(A)$ if and only if ϕ induces a row-column consistent partition of A.*

Finally, using Theorem 3.10, we state the following equivalent problem to Problem ROWCOLPART.

Problem 3.11 ($D\frac{3}{2}$*BICOLOR*) *Given the bipartite graph $G_b(A) = (V_1, V_2, E)$ representing the sparsity structure of an unsymmetric matrix $A \in R^{m\times n}$, find a $(\frac{3}{2}, p)$-bicoloring of $G_b(A)$ with the least value of p.*

In summary, the mapping from matrix problems to graph problems that we have been establishing in this section is given in Table 2.

	FD/AD using one mode	AD using both modes
Jacobian	USYMCOLPART/PartialD2COLOR	ROWCOLPART/D$\frac{3}{2}$BICOLOR
Hessian	SYMCOLPART/D$\frac{3}{2}$COLOR	-

Table 2: Summary of mapping from matrix to graph problems

From the results of this section, we see that efficient and direct estimation of Jacobian and Hessian matrices using both FD and AD techniques are related to the distance-2 coloring problem or some relaxed variant of it.

3.3 Bipartite versus Intersection Graphs

In [3] it has been shown that Problem UNSYMCOLPART is equivalent to the distance-1 graph coloring problem on the *column intersection graph* of the matrix. The column intersection graph $G_i(A)$ of a matrix A is constructed in the following way. A column i of A represents a *set* consisting of the row numbers of the non-zeros in column i and corresonds to vertex i in $G_i(A)$. An edge (i, j) in G_i exists whenever the sets in vertices i and j have non-empty intersection. In other words, there is an edge between two vertices in $G_i(A)$ whenever the corresponding columns in A are not structurally orthogonal. Figure 1 depicts a matrix A, and the corresponding bipartite graph $G_b(A) = (V_1, V_2, E)$ and intersection graph $G_i(A) = (V_2, E')$.

Note that we have augmented the graph $G_i(A)$ with *weights* on the edges. The number $w(i, j)$ is the size of the intersection of the sets represented by vertices i and j. In terms of the matrix A, $w(i, j)$ shows the number of rows where both columns i and j have non-zero entries. In the bipartite graph $G_b(A)$ this corresponds to the *multiplicity* of the paths of length two edges between vertices i and j.

There are different aspects involved in a comparison between the two graph formulations.

Storage space Given a matrix A, the graph $G_b(A) = (V_1, V_2, E)$ and the weighted graph $G_i(A) = (V_2, E')$, let the degree (number of distance-1 neighbors) of a vertex u in G_b be denoted by $d(u)$ and let the average degrees in the sets V_1 and V_2 of G_b

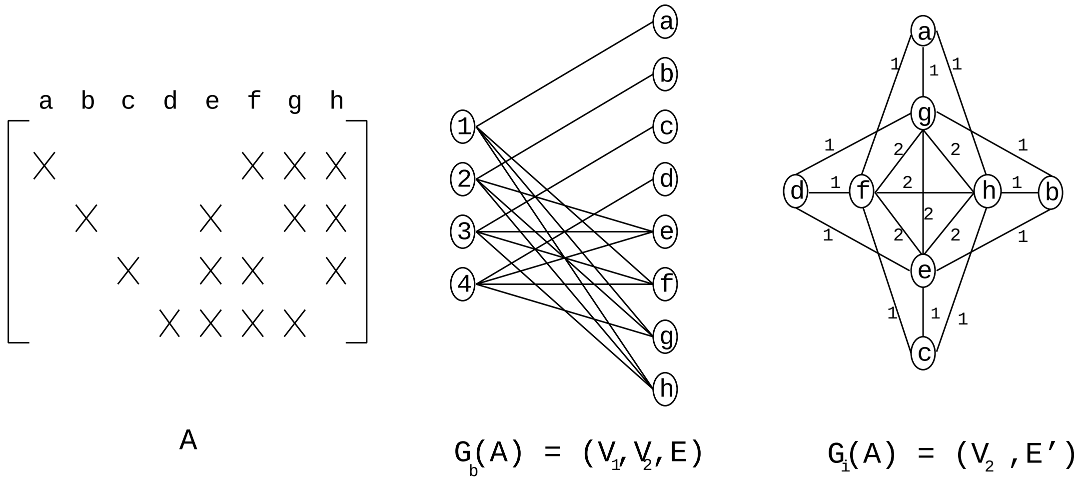

Figure 1: A matrix and its bipartite and intersection graph rep.

be $\overline{\delta}(V_1)$ and $\overline{\delta}(V_2)$, respectively. Further, let $\overline{w}$ denote the *average edge weight* in graph G_i. Then,

$$
\begin{aligned}
\sum_{e \in E'} w(e) &= \frac{1}{2} \cdot \sum_{u \in V_2} d(u)(d(v)-1), \quad \forall (u,v) \in E &(2)\\
|E'| \cdot \overline{w} &= \frac{1}{2} \cdot |V_2| \cdot \overline{\delta}(V_2) \cdot (\overline{\delta}(V_1) - 1) &(3)\\
&= \frac{1}{2} \cdot |E| \cdot (\overline{\delta}(V_1) - 1) &(4)\\
|E'| &= |E| \cdot (\frac{\overline{\delta}(V_1) - 1}{2\overline{w}}) &(5)
\end{aligned}
$$

Therefore, as long as $\frac{\overline{\delta}(V_1)-1}{2\overline{w}} > 1$, the column intersection graph will have more edges than the bipartite graph of the matrix. How likely is this for a typical large sparse matrix? We first do a randomized analysis to get an estimate for $\overline{w}$. Consider two columns i and j of A that are not structurally orthogonal ($w(i,j) \geq 1$ in $G_i(A)$). W.o.l.g., let i and j have non-zero entries at row 1. Assume that the rest of the non-zeros of columns i and j are distributed randomly. Noting that the probability that a non-zero entry of column i (similarly j) appears in row k, $2 \leq k \leq |V_1|$, is $\frac{d(i)-1}{|V_1|-1}$, the *expected* number of rows where i and j both have non-zero entries is

$$E[w(i,j)] = \overline{w} = 1 + \frac{(\overline{\delta}(V_2) - 1)^2}{|V_1| - 1}.$$

Thus, roughly speaking, for matrices (graphs) where $\overline{\delta}(V_1) \geq 5$ and $\overline{\delta}(V_2) < \sqrt{|V_1|}$, the bipartite graph representation would save memory space compared to column intersection graph. We expect this structural cut-off values to be satisfied by most matrices of practical interest.

Flexibility Notice that the column intersection graph is a 'compressed' representation of the structure of the underlying matrix. Clearly, some information is missed in the compression process. In particular, given an edge in G_i between two columns, we cannot determine at which particular row they share non-zero values. In contrast, the bipartite graph is an equivalent representation of the structure of the matrix. This provides flexibility. For example, if one decides to perform a row partitioning alone or combine it with a column partitioning, then the bipartite graph representation can be used directly, whereas the column intersection graph cannot be used. In this sense, in the case of bipartite graph formulation, the representation is decoupled from the eventual technique to be employed.

Computation The non-zero structure of a matrix A directly, without any further computation, provides the corresponding bipartite graph $G_b(A)$. In fact, in principle, the data structure used to represent A can be used for implementing algorithms that use $G_b(A)$. In contrast, $G_i(A)$ has to be computed. The time required for the computation of $G_i(A)$ is proportional to the number of edges in $G_i(A)$, i.e.,

$$T_{const} = O(|V_2|(\overline{\delta}(V_2)(\overline{\delta}(V_1) - 1)) \tag{6}$$

For the aforementioned reasons and to better fit in the unified view, we concentrate on the bipartite graph based formulation in the remainder of this paper.

4 Coloring Algorithms

Motivated by existing effective greedy distance-1 coloring heuristics, we now present and analyze greedy algorithms for finding usable solutions for the general D2GCP and for Problems PartialD2COLOR, D$\frac{3}{2}$COLOR, and D$\frac{3}{2}$BICOLOR. We begin by introducing some notations and presenting preliminary facts.

4.1 Notations and Preliminaries

Let $adj_k(u) = \{w | w \text{ is a distance-}k \text{ neighbor of } u\}$ define the distance-k neighborhood of a vertex u in $G = (V, E)$. The size of the set $adj_k(u)$, called degree-k of u, is denoted by $d_k(u)$. Let Δ denote the maximum degree-1 in G. One can easily give a bound on the maximum degree-2 in G in terms of Δ. Thus, for any vertex u in G,

$$d_2(u) = \sum_{w \in adj_1(u)} d_1(w) \leq (\Delta) d_1(u) \leq \Delta^2. \tag{7}$$

An alternative view of DkGCP can be obtained by using the notion of *power* of a graph. 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 4.1 *Let G^k be the kth power of 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 4.1 is that distance-2 coloring of a graph is equivalent to distance-1 coloring of the square of the graph. This establishes the equivalence of the bipartite graph and column intersection graph based formulations of Problem UNSYMCOLPART. Specifically, as has been shown in [3], the column intersection $G_i(A)$ of a matrix A is isomorphic to the adjacency graph of A^TA. We note that $G_i(A)$ is in fact the subgraph of $G_b(A)^2$ induced by the vertices in V_2.

Observe that distance-1 neighbors of a vertex in a graph 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 $\chi_2(G)$.

Lemma 4.2 $\chi_2(G) \geq \Delta + 1$.

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

It is worth mentioning that Lemma 4.2 captures an important feature of the matrix A in Problem UNSYMCOLPART or its equivalent Problem PartialD2COLOR. To see this, observe that each evaluation of Ad_i gives rise to only one equation consisting of the unknowns in the ith row. If there are $\alpha(i)$ unknowns in the ith row then we must have $\alpha(i) \leq p$, and hence $max\{\alpha(i) : 1 \leq i \leq m\}$ is a lower bound on the number of evaluations of Ad needed to determine A. Noting that $\alpha(i)$ is the same as degree-1 of the vertex r_i in $G_b(A)$, Lemma 4.2, excluding the additive term 1, states precisely this lower bound. The additive term 1 is excluded since a row vertex in the graph is not colored (the rows of the matrix are not partitioned).

McCormick [12] has shown that for any fixed $k \geq 2$, DkGCP is NP-hard. Prior to this, D1GCP—the"usual" graph coloring problem—was known to be NP-hard [7]. Coleman and Moré [4] sketch a proof showing that D$\frac{3}{2}$GCP, the problem of finding a $(\frac{3}{2}, p)$-coloring with the minimum p, is NP-hard, even if the graph is bipartite. This in turn implies that the problem of finding a $(\frac{3}{2}, p)$-bicoloring such that the value of p is minimized, is also NP-hard. The implication follows since a $(\frac{3}{2}, p)$-bicoloring is a $(\frac{3}{2}, p)$-coloring with the further restriction that the color sets used by the disjoint vertex sets are also disjoint. Before leaving the discussion on NP-hardness, we observe that the relationships among distance-1 coloring, distance-$\frac{3}{2}$ coloring, 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 $\chi_{kb}(G_b)$, respectively.

Theorem 4.3 *For a general graph $G = (V, E)$,*

$$\chi_1(G) \leq \chi_{\frac{3}{2}}(G) \leq \chi_2(G).$$

Furthermore, for a bipartite graph $G_b = (V_1, V_2, E)$,

$$\chi_{\frac{3}{2}}(G_b) \leq \chi_{\frac{3}{2}b}(G_b) \leq min\{\chi_2(G_b[V_1]), \chi_2(G_b[V_2])\},$$

where $\chi_2(G_b[W])$ denotes the minimum number of colors required for partial distance-2 coloring on the vertex set W.

Since the graph coloring problems of our concern are all NP-hard, in practical applications we are bound to rely on using heuristic approaches in solving them. There exist several sequential heuristics for the distance-1 coloring problem which are quite effective in coloring graphs arising from practical applications, see Coleman and Moré [3]. Distance-2 coloring and its relaxed variants, distance-$\frac{3}{2}$ coloring and distance-$\frac{3}{2}$ bicoloring, have been studied to a much lesser extent and we do not know of any practical heuristics for these problems. In the next subsections we show that some of the ideas used in distance-1 coloring heuristics can be adapted to the new cases by extending the notion of neighborhood. The resulting algorithms for the general D2GCP, PartialD2COLOR, D$\frac{3}{2}$COLOR, and D$\frac{3}{2}$BICOLOR are presented in that order.

We have noted a few recent studies on D2GCP in contexts other than derivative matrix estimation. For example, D2GCP and its application to channel assignment in radio networks has been studied by Krumke et al. [11], and Agnarsson and Halldórsson [1] have studied coloring powers of planar graphs.

4.2 Distance-2 Coloring Algorithms

A simple heuristic approach for distance-2 coloring is to visit the vertices of a graph in some order, each time assigning a vertex the smallest color that is not used by any of the vertex's distance-2 neighbors (see Algorithm Greedy-D2Coloring). Notice that since the maximum degree-2 in the graph is bounded by Δ^2, it is always possible to color a vertex using a color from the set $\{1, 2, \ldots, \Delta^2 + 1\}$.

Greedy-D2Coloring($G = (V, E)$)

for $i = 1$ to $|V|$
    $allowedColors = \{1, 2, \ldots, \Delta^2 + 1\}$
    for each colored $u \in adj_2(v_i)$
        $allowedColors = allowedColors - \{color(u)\}$
    $color(v_i) = \min\{c | c \in allowedColors\}$

The output of Greedy-D2Coloring depends on the order in which the vertices are visited. In our case, a random ordering is assumed. A better result (with fewer number of colors) can be expected if a more elaborate criterion is used in selecting the next vertex to be colored. One such ordering, which is among the well-known orderings in the distance-1 coloring literature, is *largest degree first* ordering (LFO). In *LFO-D2Coloring*, the vertices are visited in descending order of their degree-2 values. Note that other more effective orderings, such as *incidence degree ordering* [3], could also be used.

Let the number of colors used by Greedy-D2Coloring on a graph $G = (V, E)$ be ${\chi_2}^{greedy}(G)$. Then from Lemma 4.2 and Equation (7), the following theorem follows immediately.

Theorem 4.4 $\Delta + 1 \leq \chi_2(G) \leq {\chi_2}^{greedy}(G) \leq \Delta^2 + 1$.

Due to the lower bound on χ_2 above, Greedy-D2Coloring is an $O(\sqrt{|V|})$ approximation algorithm. See [12] for proof.

In a graph $G = (V, E)$, let

$$\overline{\delta}_2 = \frac{1}{|V|} \sum_{u \in V} d_1(u)^2, \tag{8}$$

denote the average degree-2 in G. The following lemma gives the time complexity of Greedy-D2Coloring.

Lemma 4.5 Greedy-D2Coloring *has time complexity* $O(|V|\overline{\delta}_2)$.

Proof: In step i of the algorithm, deleting a forbidden color from the set $allowedColors$ can be done in constant time by *marking* the element to be deleted (lazy deletion). Then in finding the minimum color to be assigned to vertex v_i, the set $allowedColors$ is scanned sequentially until the first unmarked element is encountered. Therefore, the number of operations performed during the ith step of the algorithm is proportional to $d_2(v_i)$—the distance-2 degree of vertex v_i. The total time is thus proportional to

$$\sum_{v \in V} d_2(v) = \sum_{v \in V} \sum_{u \in adj_1(v)} d_1(u) = \sum_{u \in V} (d_1(u))^2 = |V|\overline{\delta}_2.$$

The average degree-2, $\overline{\delta}_2$, in a graph can be related to the average degree-1, $\overline{\delta}_1 = \frac{1}{|V|}\sum_{v \in V} d_1(v)$, thereby providing an alternative complexity expression to that of Lemma 4.5. The expression, as we shall see, reflects the dependence of the run time on the *distribution* of the degree-1 values of the vertices of the graph.

Let the *standard deviation* of the degrees in $G = (V, E)$ be given by,

$$\sigma^2 = \frac{1}{|V|} \sum_{v \in V} (d_1(v) - \overline{\delta}_1)^2. \tag{9}$$

Then,

$$\begin{aligned}
|V|\sigma^2 &= \sum_{v \in V} d_1(v)^2 + \sum_{v \in V} \overline{\delta}_1^{\,2} - 2\sum_{v \in V} d_1(v)\overline{\delta}_1 && (10)\\
&= |V|\overline{\delta}_2 + |V|\overline{\delta}_1^{\,2} - 2\overline{\delta}_1 \sum_{v \in V} d_1(v) && (11)\\
&= |V|\overline{\delta}_2 + |V|\overline{\delta}_1^{\,2} - 2|V|\overline{\delta}_1^{\,2} && (12)\\
&= |V|\overline{\delta}_2 - |V|\overline{\delta}_1^{\,2} && (13)
\end{aligned}$$

Rewriting we get,

$$\overline{\delta}_2 = \overline{\delta}_1^{\,2} + \sigma^2. \tag{14}$$

We thus have the following corollary to Lemma 4.5.

Corollary 4.6 Greedy-D2Coloring *has time complexity* $O(|V|(\overline{\delta}_1^{\,2} + \sigma^2))$.

Since $\overline{\delta}_1 = \frac{2|E|}{|V|}$, for graphs where $\sigma < \overline{\delta}_1$, the complexity expression in Corollary 4.6 reduces to $O(\frac{|E|^2}{|V|})$. In particular, for sparse graphs, where $|E| = O(|V|)$, the time complexity of Greedy-D2Coloring becomes $O(|V|)$.

If the vertices of the graph are sorted in descending order of their degree-2 values using a linear time sorting algorithm prior to their coloring, the time complexity of LFO-D2Coloring is the same as that of Greedy-D2Coloring. Notice that since the degree-2 values of the vertices are integers (and bounded), a linear time sorting algorithm can be used.

4.3 Partial Distance-2 Coloring Algorithms

In Section 3.2, it was shown that Problem PartialD2COLOR corresponds to Problem UNSYMCOLPART. The input to PartialD2COLOR is a bipartite graph $G_b = (V_1, V_2, E)$ and the question is to find a partial distance-2 coloring on the set V_2.

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. 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 $\{1, 2, \ldots, \Delta(V_2)(\Delta(V_1)-1))+1\}$. Using this, a slight modification of Algorithm Greedy-D2Coloring gives Greedy-PartialD2Coloring.

Lemma 4.7 Greedy-PartialD2Coloring *has time complexity* $O(|V_2|\overline{\delta}(V_2)(\overline{\delta}(V_1) - 1))$.

From Lemma 4.7 and Equation (6), 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 distance-1 coloring is asymptotically the same as the time required for the direct computation of distance-2 coloring.

Greedy-PartialD2Coloring$(G_b = (V_1, V_2, E))$

for $i = 1$ to $|V_2|$
    $allowedColors = \{1, 2, \ldots, \Delta(V_2)(\Delta(V_1)-1)+1\}$
    for each $u \in adj_1(v_i \in V_2)$
        for each colored $w \in adj_1(u)$
            $allowedColors = allowedColors - \{color(w)\}$
    $color(v_i) = \min\{c | c \in allowedColors\}$

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

Recall that a $(\frac{3}{2}, p)$-coloring is a $(1, p)$-coloring where every path of length 3 uses at least 3 colors. Recall also that it is a relaxed distance-2 coloring. As an illustration, suppose $uvwx$ is a path in a graph. A coloring ϕ in which $\phi(u) = \phi(w) = 2$, $\phi(v) = 1$ 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 uvw be a path in G and suppose u and v are colored and we want to determine the color of w. We allow $\phi(w)$ to be equal to $\phi(u)$ as long as $\phi(v) < \phi(u)$. To see that this coloring can always be extended to yield a valid distance-$\frac{3}{2}$ coloring, consider the path $uvwx$, an extension of path uvw in one direction. Now, since $\phi(w) = \phi(u) > \phi(v)$, we cannot let $\phi(x)$ be equal to $\phi(v)$. Obviously, $\phi(x)$ should be different from $\phi(w)$, otherwise it will not be a legal distance-1 coloring. Thus the path $uvwx$ uses three colors, ϕ is a distance-1 coloring and therefore it is a valid distance-$\frac{3}{2}$ coloring. We use the idea just described to modify Greedy-D2Coloring into Greedy-D$\frac{3}{2}$Coloring.

Greedy-D$\frac{3}{2}$Coloring(G=(V,E))

for $i = 1$ to $|V|$
  $allowedColors = \{1, 2, \ldots, \Delta^2 + 1\}$
  for each $u \in adj_1(v_i)$
    if u is colored
      $allowedColors = allowedColors - \{color(u)\}$
    for each colored $w \in adj_1(u)$
      if u is colored
        if $(color(w) < color(u))$
          $allowedColors = allowedColors - \{color(w)\}$
      else
        $allowedColors = allowedColors - \{color(w)\}$
  $color(v_i) = \min\{c | c \in allowedColors\}$

Similarly, LFO-D2Coloring can be modified to yield LFO-D$\frac{3}{2}$Coloring. We point out that an essentially similar heuristic to LFO-D$\frac{3}{2}$Coloring, formulated in the language of matrices, was first suggested by Powell and Toint [13].

Observe that Greedy-D$\frac{3}{2}$Coloring has the same time complexity as Greedy-D2Coloring while the number of colors used by the former is expected to be less than that of the latter.

4.5 Distance-$\frac{3}{2}$ Bicoloring Algorithms

Notice that a $(\frac{3}{2}, p)$-bicoloring is a $(\frac{3}{2}, p)$-coloring in which the graph under consideration is bipartite and the set of colors used by each vertex set in the graph are disjoint.

Suppose the graph to be colored is $G_b = (V_1, V_2, E)$ and the vertices in each vertex set are ordered in some fashion. The following procedure describes a trivial way of achieving a valid distance-$\frac{3}{2}$ bicoloring. Color all the vertices in, say V_1, using one color that is not to be used for coloring a vertex in V_2. Then, color the vertices in V_2 such that the coloring is distance-2 consistent. Notice that this procedure gives exactly a partial distance-2 coloring on V_2. In terms of the underlying matrix, this corresponds to column partitioning. This observation suggests that, if we are to benefit from row-column partitioning, then the bicoloring algorithm of the corresponding bipartite graph has to alternate from one set to the other in its order of visiting the vertices.

We describe our heuristic which makes use of this observation. In our distance-$\frac{3}{2}$ bicoloring heuristic, we rely on the fact that enforcing a coloring to be distance-1 consistent can be temporarily postponed. With this in mind, we can use the same color set for both V_1 and V_2, zig-zag from one set to the other, each time picking the next vertex in the list and coloring it. Later, some *offset* can be added to the color of each of the vertices in one of the vertex sets to ensure that the coloring is distance-1 consistent. The details of the heuristic are outlined in Algorithm Greedy-D$\frac{3}{2}$BiColoring.

Greedy-D$\frac{3}{2}$BiColoring($G_b = (V_1, V_2, E)$)

{ $V_{min} = \min\{V_1, V_2\}$; $V_{max} = \max\{V_1, V_2\}$; }
for $i = 1$ to $|V_{min}|$
    $assignSmallestColor(v_i)$ {v_i is the ith vertex in V_{min}}
    $assignSmallestColor(w_i)$ {w_i is the ith vertex in V_{max}}
for $i = |V_{min}| + 1$ to $|V_{max}|$
    $assignSmallestColor(v_i)$ {v_i is the ith vertex in V_{max}}
$offset = \max_{v_i \in V_{min}}(color(v_i))$
for $i = 1$ to $|V_{max}|$
    $color(v_i) = color(v_i) + offset$ {v_i is the ith vertex in V_{max}}

assignSmallestColor(v)

$allowedColors = \{1, 2, \ldots, \Delta^2 + 1\}$
for each $u \in adj_1(v)$
    for each colored $w \in adj_1(u)$
        if u is colored
            if $(color(w) < color(u))$
                $allowedColors = allowedColors - \{color(w)\}$
        else
            $allowedColors = allowedColors - \{color(w)\}$
$color(v_i) = \min\{c | c \in allowedColors\}$

5 Conclusion

We have studied the efficient estimation of Jacobian and Hessian matrices using both FD and AD techniques, in each case using direct methods. In the case of Jacobian estimation using FD, a new graph theoretic formulation has been suggested. In the other cases, known graph formulations have been reviewed in a manner that exposes the inter-relationship among the different cases. The table below summarizes the relationship. Distance-$\frac{3}{2}$ coloring is a relaxed distance-2 coloring, and distance-$\frac{3}{2}$

	FD/AD using one mode	AD using both modes
Jacobian	distance-2 coloring	distance-$\frac{3}{2}$ bicoloring
Hessian	distance-$\frac{3}{2}$ coloring	-

bicoloring is a restricted distance-$\frac{3}{2}$ coloring.

It is well known that, when a *column intersection* graph is used, the direct estimation of a Jacobian using FD is equivalent to (standard) distance-1 graph coloring problem. Our formulation as distance-2 coloring problem is based on a bipartite graph representation of the sparsity structure of the underlying matrix. The column intersection graph representation of a matrix A requires the (indirect) computation and storage of $A^T A$ whereas our graph representation requires no further computation and no more space than that required to efficiently store matrix A. When A is of very large size, the storage requirement of $A^T A$ can be a problem. Moreover, unlike a column intersection graph, a bipartite graph representation is *robust*: the representation only captures the sparsity structure of the matrix and is decoupled from the eventual technique used to manipulate it. For example, suppose partitioning the rows, instead of the columns, is of interest. Then, the bipartite graph representation can still be used, whereas the column intersection graph cannot. It should also be noted that, given a matrix A the computational cost involved in the construction of the column intersection graph $G_i(A)$ and then distance-1 coloring is of the same order as that of directly performing distance-2 coloring on $G_b(A)$.

We have also presented and analyzed several effective greedy heuristics for distance-2 coloring, distance-$\frac{3}{2}$ coloring, and distance-$\frac{3}{2}$ bicoloring problems. Our emphasis has been on showing how the techniques used in distance-1 coloring heuristics can be extended to the new coloring variants.

Developing practical parallel algorithms for these problems, is an interesting direction for future work and is currently being pursued by the authors of this paper.

A similar review of substitution methods is also another interesting direction to follow.

References

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

[2] 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.

[3] 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.

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

[5] 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.

[6] 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.

[7] M.R. Garey and D.S. Johnson. *Computers and Intractability.* W.H. Freeman, New York, 1979.

[8] D. Goldfarb and Ph. L. Toint. Optimal estimation of Jacobian and Hessian matrices that arise in finite difference calculations. *Mathematics of Computation*, 43(167):69–88, July 1984.

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

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

[11] S. O. Krumke, M. V. Marathe, and S.S. Ravi. Approximation algorithms for channel assignment in radio networks. In *Dial M for Mobility, 2nd Internationla Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, Dallas, Texas, September 30 – October 1 1998.

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

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