MULTISCALE MODEL. SIMUL.
Vol. 9, No. 1, pp. 407–423

RELAXATION-BASED COARSENING AND MULTISCALE GRAPH ORGANIZATION*

DORIT RON†, ILYA SAFRO‡, AND ACHI BRANDT†

Abstract. In this paper we generalize and improve the multiscale organization of graphs by introducing a new measure that quantifies the "closeness" between two nodes. The calculation of the measure is linear in the number of edges in the graph and involves just a small number of relaxation sweeps. A similar notion of distance is then calculated and used at each coarser level. We demonstrate the use of this measure in multiscale methods for several important combinatorial optimization problems and discuss the multiscale graph organization.

Key words. relaxation-based coarsening, multiscale and multilevel methods for graphs, algebraic distance, partitioning, graph linear ordering

AMS subject classifications. 68R10, 90C06, 90C35, 99C15, 65M55, 68W01

DOI. 10.1137/100791142

1. Introduction.

A general approach for solving many large-scale graph problems, as well as most other classes of large-scale computational science problems, is through multilevel (multiscale, multiresolution, etc.) algorithms. This approach generally involves *coarsening* the problem, producing from it a sequence of progressively coarser levels (smaller, hence simpler, related problems), then recursively using the (approximate) solution of each coarse problem to provide an initial approximation to the solution at the next-finer level. At each level, this initial approximation is first improved by what we generally call "local processing" (LP). This is an inexpensive sequence of short steps, each involving only a few unknowns, together covering all unknowns of that level several times over. The usual examples of LP are a few sweeps of classical (e.g., Gauss–Seidel or Jacobi) relaxation in the case of solving a system of equations, or a few Monte Carlo passes in statistical-physics simulations. Following the LP, the resulting approximation may be further improved by one or several cycles, each using again a coarser-level approximation followed by LP, applying them at each time to the *residual* problem (the problem of calculating the *error* in the current approximation). See, for example, references [6, 7, 11, 12, 13, 14, 38, 42].

At each level of coarsening one needs to define the set of coarse unknown variables and the equations (or the stochastic relations) that they should satisfy (or the energy that they should minimize). Each coarse unknown is defined in terms of the next-finer-level unknowns (*defined*, not *calculated*: they are all unknowns until the coarse

*Received by the editors April 5, 2010; accepted for publication (in revised form) January 24, 2011; published electronically March 3, 2011. This work was partially funded by the CSCAPES Institute, a DOE project, and in part by DOE contract DE-AC02-06CH11357. This article was created in part by UChicago Argonne, LLC, operator of Argonne National Laboratory. Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under contract DE-ACO2-06CH11357. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.

http://www.siam.org/journals/mms/9-1/79114.html

†Faculty of Mathematics and Computer Science, The Weizmann Institute of Science, Rehovot, 76100, Israel (dorit.ron@weizmann.ac.il, achi.brandt@weizmann.ac.il).

‡Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439 (safro@mcs.anl.gov).

level is approximately solved and the fine level is interpolated from that solution). The following are examples:

- The set of coarse unknowns can simply represent a chosen subset of the fine-level set.
- If the fine-level variables are real numbers or vectors, each coarse variable can represent a weighted average of several of them.
- If the fine-level variables are Ising spins (having only values of $+1$ or -1), each coarse variable can again be an Ising spin, representing the *sign* of the sum of several fine spins.
- A coarse variable can be defined from several fine variables by a stochastic process ([5], for example).
- In the case of graph problems, each node of the coarse graph can represent an *aggregate* of several fine-level nodes or a *weighted aggregate* of such nodes, that is, allowing each fine-level node to be split between several aggregates.

The choice of an adequate LP at a fine level and the choice of an adequate set of variables at the next-coarser level are strongly coupled. The general guiding rule [10] is that this pair of choices is good if (and to the extent that) a fine-level solution can always be recovered from the corresponding set of coarse variables by a short iterative use of a suitably modified version of the LP. That version is called *compatible LP* (CLP). Examples are compatible Monte Carlo (CMC), introduced in [13], and compatible relaxation (CR), introduced in [8].

The CLP, needed in several important upscaling procedures (such as the selection of the coarse variables, the acceleration of the fine-level simulations, and the processing of fine-level windows within coarse simulations; see [10]) can also be used for performing the interpolation from the coarse solution to obtain the first approximation at the fine level. When possible, however, the construction of a more explicit interpolation is desired in order to apply it for the direct formulation of equations (or an energy functional) that should govern the coarse level, as in Galerkin coarsening.

In the process of defining the set of coarse variables and in constructing an explicit interpolation, it is important to know how "close" two given fine-level variables are to each other at the stage of switching to the coarse level. We need to know, in other words, to what extent the value after the LP of one variable implies the value of the other. If they are sufficiently close, they can, for example, be aggregated to form a coarse variable.

The central issue addressed in the present article is how to measure this "closeness" between two variables in a system of equations or between two nodes in a given graph. (We consider the latter to be a special case of the former, by associating the graph with the system formed by its Laplacian.) More generally, we want to define the distance of one variable x_i from a small subset S of *several* variables, in order to measure how well x_i can be interpolated from S following the LP.

In classical algebraic multigrid (AMG), aimed at solving the linear system

$$Ax = b \quad \text{or} \quad \sum_{j=1}^{n} a_{ij}x_j = b_i \quad , \quad (i = 1, \ldots, n), \tag{1.1}$$

the closeness of two unknowns x_i and x_j is measured simply by the relative size of their coupling a_{ij}, for example, by the quantity

$$|a_{ij}|/\max\left(\sum_k |a_{ik}|, \sum_k |a_{kj}|\right) \tag{1.2}$$

(or similarly by the relative size of their coupling in some *power* of A). Although this definition has worked well for the coarsening procedures of discretized scalar elliptic differential equations, it is not really effective, and sometimes meaningless, for systems lacking sufficient diagonal dominance (including many discretized *nonscalar* elliptic systems). Moreover, even for systems with a fully dominant diagonal (such as the Laplacian of a graph), the classical AMG definition may result in wrong coarsening, for example, in graphs with nonlocal edges (see example in section 3).

Instead, we propose to define the "closeness" between two variables exactly, by measuring how well their values are correlated at the coarsening stage, namely, following the LP relaxation sweeps. Since the coarse level is actually applied to the *residual* system, the two variables will be considered close if their *errors* have nearly the same ratio in all relaxed vectors. We will thus create a sequence of K *normalized relaxed error vectors* $x^{(1)}, \ldots, x^{(K)}$, each obtained by relaxing the homogeneous system $Ax = 0$ from some (e.g., random) start and then normalizing the result. We will then define the *algebraic distance* (reciprocal of "closeness") between any two variables x_i and x_j as

$$\min_\eta \left(\sum_{k=1}^{K} |\eta x_i^{(k)} - |\eta^{-1}| x_j^{(k)}|^p \right)^{1/p}, \tag{1.3}$$

where $p \geq 2$ in order to attach larger weights to larger differences (using usually either $p = 2$ or the maximum norm ($p \to \infty$)). This use of η gives a symmetric measure of how well x_i can be interpolated from x_j or vice versa. For the graph Laplacian (and other zero-sum A) this can be simplified to a distance defined as

$$\left(\sum_{k=1}^{K} (x_i^{(k)} - x_j^{(k)})^2 \right)^{1/2} \quad \text{or} \quad \max_{k=1}^{K} \left| x_i^{(k)} - x_j^{(k)} \right|. \tag{1.4}$$

The algebraic distance defined by (1.4) measures how strongly in the graph any pair of nodes i and j is connected, not just directly, but possibly even more *indirectly*, through their common neighbors, neighbors of neighbors, etc. Starting from a random vector and applying a small number of relaxation sweeps, it is expected that in case i and j are strongly connected, their values in the relaxed vector will be similar. Such a similarity can accidentally occur also between two nodes which are not strongly connected to each other in the graph, due to the random initialization. However, since we use several such test vectors, it is not likely that such an accidental similarity will occur in all of them. In other words, similarity of *all* values between two nodes (hence small value in (1.4)) indicates they are strongly (directly and indirectly) connected. Conversely, it is enough that one vector results with a large difference between the values of two nodes to conclude that they are not strongly coupled. In our coarsening procedure, the algebraic distance main function is to avoid aggregating together two nodes that *seem* to be strongly coupled (having a large weight on their connecting edge) but actually are far apart (having very different sets of neighbors, hence weak algebraic coupling), as occurs, for example, in the graph in Figure 3.1: (as shown in Table 3.1 for $K = 10$, $r = 20$, or 50 and $w_{ij} = 2$) the coupling between i and j, which seems to be *twice* as strong as all other couplings in the graph, turns out to be *half* the strength of other couplings of either i or j when measured by the algebraic distance proposed here.

More generally, the distance of a node i from a subset S of several nodes can similarly be defined as the deviation of the best-fitted interpolation from S to i, where the deviation is the L_2 norm of the vector of K errors obtained upon applying the

interpolation to our K normalized relaxed error vectors, and the best-fitted interpolation is the one having the minimal deviation. (This least-square interpolation is the one introduced in bootstrap AMG (BAMG) [9] for the coarse-to-fine explicit interpolation.)

An essential aspect of the "algebraic distance" defined here is that it is a crude *local distance*. It measures meaningful closeness only between neighboring nodes; the closer they are the less fuzzy is their measured distance. For nodes that should not be considered as neighbors, their algebraic distance just detects the fact that they are far apart; its exact value carries no further meaning. The important point is that this crude local definition of distance is fast to calculate and is all that is required for the coarsening purposes. A similar notion of distance is then similarly calculated at each coarser level.

Indeed, we argue that meaningful distances in a general graph should, in principle, be *defined* (not just *calculated*) only in such a multiscale fashion. This essential viewpoint, and relations to diffusion distances and spectral clustering are discussed in section 5. In particular, we advocate the replacement of spectral methods by AMG-like multilevel algorithms, which are both faster and more tunable to define better solutions to many fuzzy graph problems (see, for example, [41, 42]).

The paper is organized as follows. The graph problems we use to demonstrate our approach are introduced in section 2. The calculation of the "algebraic distance" and its use within the multiscale algorithm is described in section 3. Results of tests are summarized in section 4. Finally, the relations of our approach to diffusion distances and spectral clustering are discussed in section 5.

2. Notation and problem definitions. Given a weighted graph $G = (V,\ E)$, where $V = \{1, 2, \ldots, n\}$ is the set of nodes (vertices) and E is the set of edges, denote by w_{ij} the nonnegative weight (coupling) of the undirected edge ij between nodes i and j; if $ij \notin E$, then $w_{ij} = 0$. We consider as our examples the following two optimization problems.

2.1. Linear ordering. Let π be a bijection

$$\pi :\ V \ \longrightarrow\ \{1, 2, \ldots, n\}.$$

The purpose of linear ordering problems is to minimize some functional over all possible permutations π. The following functional should be minimized for the minimum p-sum problem[1] (MpSP):

$$\sigma_p(G, \pi) = \sum_{ij} w_{ij} |\pi(i) - \pi(j)|^p. \tag{2.1}$$

In the generalized form of the problem that emerges during the multilevel solver, each vertex i is assigned with a *volume* (or *length*), denoted v_i. Given the vector of all volumes, v, the task now is to minimize the cost

$$\sigma_p(G, \pi, v) \stackrel{def}{=} \sigma_p(G, x) = \sum_{ij} w_{ij} |x_i - x_j|^p,$$

where $x_i = \frac{v_i}{2} + \sum_{k, \pi(k) < \pi(i)} v_k$; that is, each vertex is positioned at its center of mass, capturing a segment on the real axis that equals its length. The original form

[1] We use this definition for simplicity. The usual definition of the functional is $\sigma_p(G, \pi) = (\sum_{ij} w_{ij} |\pi(i) - \pi(j)|^p)^{1/p}$, which yields the same minimization problem.

of the problem is the special case where all the volumes are equal. In particular, we would like to concentrate on the minimum linear arrangement (where $p = 1$) and the minimum 2-sum problem (M2SP) that were proven to be nondeterministic polynomial time complete (NP-complete) in [22, 24] and whose solutions can serve as an approximation for many different linear ordering problems replacing the spectral approaches [41, 42].

2.2. Partitioning. The goal of the 2-partitioning problem is to find a partition of V into two disjoint nonempty subsets Π_1 and Π_2 such that

$$\text{minimize} \sum_{i\in\Pi_1, j\in\Pi_2} w_{ij} \quad \text{subject to} \quad |\Pi_k| \le (1+\alpha)\cdot\frac{|V|}{2}, \ (k=1,2), \tag{2.2}$$

where α is a given *imbalance factor*.

Graph partitioning is an NP-hard problem [23] used in many fields of computer science and engineering. Applications include VLSI design, minimizing the cost of data distribution in parallel computing, and optimal tasks scheduling. Because of its practical importance, many different heuristics (spectral [36], combinatorial [31, 21], evolutionist [15], etc.) have been developed to provide an approximation in a reasonable (and, one hopes, linear) computational time. However, only the introduction of multilevel methods for partitioning [30, 35, 2, 34, 45, 3, 37, 4, 27, 29, 1] has really provided a breakthrough in efficiency and quality.

3. The coarsening algorithm. In the multilevel framework a hierarchy of decreasing size graphs $G_0, G_1, \ldots, G_k$ is constructed. Starting from the given graph, $G_0 = G$, we create by recursive *coarsening* the sequence $G_1, \ldots, G_k$, and then solve the coarsest level directly, and uncoarsen the solution back to G.

In general, the AMG-based coarsening is interpreted as a process of weighted aggregation of the graph nodes to define the nodes of the next-coarser graph. In weighted aggregation each node can be divided into fractions, and different fractions belong to different aggregates. The construction of a coarse graph from a given one is divided into three stages. First a subset of the fine nodes is chosen to serve as the *seeds* of the aggregates (the nodes of the coarse graph). Then the rules for aggregation are determined, thereby establishing the fraction of each nonseed node belonging to each aggregate. Finally, the graph couplings (or edges) between the coarse nodes are calculated. The entire coarsening scheme is shown in Algorithm 1.

The AMG-based multilevel framework for graph optimization problems is discussed, for example, in [42]. In the present work we generalize the coarsening part of the AMG-based framework. The problem-dependent solution of the coarsest level and the uncoarsening are not changed here. They are fully described in [42] and references therein.

The principal difference between the previous AMG-based coarsening approaches [42, 28, 17] and the new *relaxation-based* approach is the improved measure, the *algebraic coupling*, assigned to each edge, or, more generally, between any two nodes in the graph. The algebraic coupling is the reciprocal of the calculated *algebraic distance* introduced below.

Algebraic distance and coupling. We start with a simple example which illustrates the need for an improved measure while coarsening. Consider Figure 3.1: one additional edge ij (connecting nodes i and j) is added to a regular two dimensional mesh. The usual set of seeds chosen by some AMG-based criterion (see, for example, Algorithm 2) will include about half of the nodes, roughly, those which belong to one

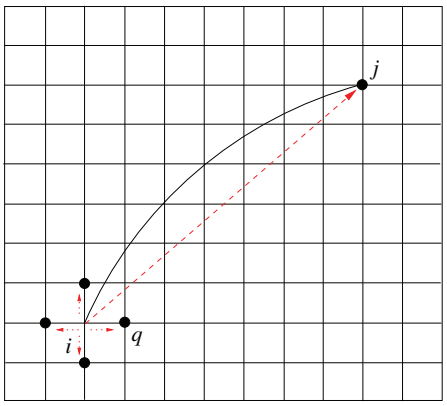


FIG. 3.1. *Mesh graph with an additional edge between nodes i and j. The black dots mark some of the nodes selected to serve as the seeds of the coarse aggregates; see Algorithm* 2.

color of a red-black chess board; see, for instance, the five black dots in the figure. The main question is then to which of those five seeds should node i be aggregated. Clearly, nodes i and j should not belong to the same aggregate unless their coupling is much stronger than other graph couplings. However, if the weight of ij is just somewhat larger than all other graph edges, node i will tend to be aggregated with node j, rather than with any of its immediate geometric neighbors. Such a decision will create bad coarse-level approximations in many optimization problems (e.g., linear ordering and partitioning). Moreover, at the next-coarser levels the approximation may further deteriorate by making similar wrong decisions, making the entire neighborhood of i close to j, thereby promoting linear arrangements in which *many* local couplings would unnecessarily become long-range ones. To prevent this situation we would like to have a measure that not only evaluates the coupling between i and j according to the *direct* coupling between them but also takes into account the contribution of connections between the *neighborhoods* of i and j. In other words, if i and j are not only connected directly, but are also connected via other nodes so that they can be reached from each other by a short path (with only a small number of intermediate nodes), the coupling between them should be stronger than if i and j are connected only directly, while all other possible paths are much longer. Thus, it is clear that the new measure should indicate that the coupling between i and j is in fact weaker than the coupling of i to its immediate neighbor (e.g., q). That is, if the immediate (graph) neighbors of i are connected to those of j, the coupling between i and j should be enhanced, while if i's neighbors are not connected to those of j, as in Figure 3.1, a significant weakening of the ij coupling is due. This measure will prevent possible errors while coarsening. As presented below (in Table 3.1) the new calculated coupling of node i to q turned out to be almost *twice* as strong as the coupling to j even with $w_{iq} = 1$ and $w_{ij} = 2$ (using $K = 10$ and $r = 20$).

We introduce the notion of *algebraic distance*, which is based on the same set of *test vectors* (TVs) being used in the BAMG [9]. The key new ingredient of the adaptive BAMG setup is the use of several TVs, collectively representing algebraically smooth error, to define the interpolation weights. When a priori knowledge of the nature of this error is not available, slightly relaxed random vectors are used for this task. A set of some K low-residual TVs $\{x^{(k)}\}_{k=1}^{K}$ can first be obtained by relaxation. Namely,

TABLE 3.1

Statistical results (over 100 runs) for the average (and, in parentheses, the standard deviation) of $ln(d_{ij}/d_{i*})$ (see 3.6), calculated with K TVs and r Jacobi relaxation sweeps for different relative strengths of w_{ij}.

K	r	$w_{uv}=1$ for (u,v) nearest neighbors			
		$w_{ij}=1$	$w_{ij}=2$	$w_{ij}=3$	$w_{ij}=4$
1	10	2.47(1.51)	1.88(1.74)	1.38(1.85)	1.14(1.69)
	20	2.74(1.74)	2.1(1.59)	1.4(1.26)	1.44(1.59)
	50	2.65(1.36)	1.98(1.76)	1.92(1.41)	1.5(1.59)
	100	3.03(1.72)	2.14(1.32)	1.51(1.42)	1.16(1.78)
5	10	1(0.502)	0.628(0.416)	0.24(0.417)	-0.0484(0.397)
	20	1.34(0.442)	0.825(0.415)	0.435(0.358)	0.208(0.332)
	50	1.68(0.342)	1.04(0.338)	0.643(0.306)	0.362(0.296)
	100	1.78(0.467)	1.06(0.392)	0.743(0.369)	0.396(0.359)
10	10	0.821(0.281)	0.443(0.294)	0.022(0.293)	-0.244(0.313)
	20	1.09(0.268)	0.624(0.239)	0.298(0.235)	0.0126(0.253)
	50	1.49(0.263)	0.86(0.235)	0.504(0.226)	0.2(0.204)
	100	1.69(0.315)	1.01(0.275)	0.572(0.264)	0.285(0.271)

each $x^{(k)}$ is a result of r fine-level relaxation sweeps on the homogeneous equation $Ax = 0$, starting from a *random* approximation, where A is the Laplacian of the graph. In particular, we have used a small number (usually r=10) of Jacobi under relaxation sweeps with $\omega = 0.5$. That is, the new value for each $x^{(k)}$, $k = 1, \ldots, K$ (in our tests $K = 20$) is

$$x^{(k)}_{NEW} = (1-\omega)x^{(k)} + \omega x^{(k)}_{JAC} \ , \tag{3.1}$$

where

$$x^{(k)}_{JAC} = D^{-1}(D-A)x^{(k)}, \tag{3.2}$$

D being the diagonal of A. The algebraic distance from node i to node j is defined over the K relaxed TVs by

$$d_{ij} = \max_{k=1}^{K} \left| x_i^{(k)} - x_j^{(k)} \right|. \tag{3.3}$$

Other definitions, such as

$$d_{ij} = \sum_{k=1}^{K} \left(x_i^{(k)} - x_j^{(k)} \right)^2 \tag{3.4}$$

are also possible. Hence, *only* if d_{ij} is small may nodes i and j be aggregated into the same coarse node. The algebraic coupling between i and j, c_{ij} is defined as the reciprocal of d_{ij}:

$$c_{ij} = 1/d_{ij} \ . \tag{3.5}$$

We return to the example in Figure 3.1 and demonstrate the outcome of definition (3.3). We show that (*a*) i will *not* tend to be connected to j unless w_{ij} equals the sum of i's other couplings, (*b*) K, the number of the relaxed TVs, should be larger than 1, and (*c*) that even if i *is* connected to j as a result of strong w_{ij}, i's other neighbors

Data: Q , ν
Result: coarse graph
For every edge ij derive its algebraic distance d_{ij} (3.3) or (3.4) and algebraic coupling c_{ij} (3.5);
SelectCoarseNodes(Q , ν);
Define the coarse graph using the matrix P in (3.9).

Algorithm 1: Coarsening scheme

will not tend to be connected to i as well but will prefer other neighbors; hence the neighborhoods of i and j will not tend to be connected to each other.

(a) Consider Table 3.1. The number K of TVs is given in the leftmost column. The number r of Jacobi relaxation sweeps varies from 10 to 100 as shown in the second to the left column. Each of the four columns to the right presents the (natural) logarithm of d_{ij}/d_{i*}, where

$$d_{i*} = \min\{d_{is} | s \text{ a nearest neighbor of } i\}. \tag{3.6}$$

Each number is averaged over 100 independent runs (obtained by random initialization of the TVs using different reordering of the graph nodes), for graph couplings $w_{uv} = 1$ when u and v are nearest neighbors, and $w_{ij} = 1, 2, 3$, or 4 as shown. The numbers in parentheses are the corresponding standard deviations. Clearly the strength of the coupling between i and j is relatively decreased when measured by the algebraic distance. For instance, if the graph coupling between i and j is 1 (as are all other couplings in the graph), then after 20 relaxation sweeps (with $K = 10$) d_{ij} is three times bigger than the minimum of the (algebraic distance of the) edges to i's four nearest neighbors. Thus, the algebraic coupling between i and j is *not* the strongest coupling of i (not even close to it), and hence it is guaranteed that i and j will not belong to the same coarse node.

(b) The importance of using more than just 1 TV can be seen from the values of the standard deviations: the use of 1 TV results in standard deviations similar to the average, which means that $ln(d_{ij}/d_{i*})$ has a significant chance to become negative, so ij has a significant chance to be the strongest coupling of i. With 10 TVs this chance becomes much smaller, at least for $w_{ij} \leq 2$. Even with 10 TVs, however, the chance grows with the strength of w_{ij}, becoming more than 50% roughly when $w_{ij} \geq 4$. Thus, the aggregation of i with j becomes likely. This by itself is fine and justified. What we really need to avoid is that entire neighborhoods of i and j will, as a result, be aggregated at some coarser level.

(c) In the case that i is aggregated with j we want to check whether the neighbors of i will tend to be aggregated with i (and thus also with j) or will prefer their other neighbors. To see that we calculate the (natural) logarithm of d_{qi}/d_{q^i*}, where

$$d_{q^i*} = \min\{d_{qs} | s \text{ a nearest neighbor of } q \text{ other than } i\}. \tag{3.7}$$

As shown in Table 3.2, q would rather be aggregated with one of its other-than-i neighbors. For example, for $K = 10$, $r = 20$, and $w_{ij} = 4$ out of the 100 runs, in 95 q would have been connected with i. The main conclusion is that nodes i and j do not tend to be connected as long as w_{ij} is smaller than the sum of all other couplings of i or of j. When the coupling is of the same strength, they will be connected about half the time, but then, not less importantly, the neighbors of i (and similarly of j)

TABLE 3.2

Statistical results (over 100 runs) for the average (and, in parentheses, the standard deviation) of $ln(d_{qi}/d_{q^i*})$ (see Figure 3.1 and (3.7)), calculated with K TVs and r Jacobi relaxation sweeps for different relative strengths of w_{ij}.

		$w_{uv}=1$ for (u,v) nearest neighbors			
K	r	$w_{ij}=1$	$w_{ij}=2$	$w_{ij}=3$	$w_{ij}=4$
1	10	0.975(1.67)	0.939(1.7)	1.14(1.63)	1.07(1.89)
	20	0.911(1.62)	1.09(1.31)	1.02(1.46)	0.931(1.64)
	50	1.37(1.77)	1.14(1.79)	1.28(1.49)	1.24(1.45)
	100	0.897(1.55)	1.23(1.45)	1.29(1.53)	1.31(1.44)
5	10	0.382(0.534)	0.482(0.428)	0.416(0.52)	0.587(0.487)
	20	0.434(0.444)	0.472(0.366)	0.592(0.486)	0.663(0.458)
	50	0.498(0.436)	0.755(0.53)	0.784(0.526)	0.813(0.455)
	100	0.501(0.522)	0.746(0.544)	0.812(0.549)	0.816(0.535)
10	10	0.283(0.312)	0.299(0.316)	0.376(0.307)	0.401(0.357)
	20	0.362(0.281)	0.419(0.295)	0.449(0.288)	0.441(0.327)
	50	0.448(0.311)	0.531(0.35)	0.672(0.351)	0.604(0.333)
	100	0.464(0.377)	0.682(0.348)	0.839(0.374)	0.749(0.39)

will *not* tend to join them but will prefer to be connected to other nearest neighbors nodes. Similar results are obtained when using (3.4) to calculate d_{ij}.

With the notion of the algebraic coupling in mind, the coarse nodes selection and the calculation of the aggregation weights are modified as follows.

Seed selection. The construction of the set of seeds C and its complement F is guided by the principle that each F-node should be "strongly coupled" to C. We will include in C nodes with exceptionally large volume or nodes expected (if used as seeds) to aggregate around them an exceptionally large total volume of F-nodes. We start with $C=\emptyset$—hence $F=V$—and then sequentially transfer nodes from F to C as follows. As a measure of how large an aggregate seeded by $i\in F$ might grow, we define its *future volume* ϑ_i by

$$\vartheta_i = v_i + \sum_{ij\in E} v_j \cdot \frac{c_{ji}}{\sum_{jk\in E} c_{jk}}. \tag{3.8}$$

Nodes with future volume larger than ν times the average of the ϑ_i's are first transferred to C as most "representative" (in our tests $\nu=2$). The insertion of additional F-nodes to C depends on a "strength of coupling to C" threshold Q (in our tests $Q=0.5$), as specified in Algorithm 2.

Data: Q, ν
Result: set of seeds C
$C\leftarrow\emptyset,\ F\leftarrow V$;
Calculate ϑ_i (3.8) for each $i\in F$, and their average $\overline{\vartheta}$;
$C\leftarrow$nodes i with $\vartheta_i > \nu\cdot\overline{\vartheta}$;
$F\leftarrow V\setminus C$;
forall $i\in F$ *in descending order of* ϑ_i **do**
  if $(\sum_{j\in(C\cap N(i))} c_{ij}/\sum_{j\in N(i)} c_{ij})\le Q$ *or*
  $(\sum_{j\in(C\cap N(i))} w_{ij}/\sum_{j\in N(i)} w_{ij})\le Q$ **then** move i from F to C;
end

Algorithm 2: SelectCoarseNodes(Q, ν)

Coarse nodes. Each node in the chosen set C becomes the seed of an aggregate that will constitute one coarse-level node. Next it is necessary to determine for each $i \in F$ a list of $j \in C$ to which i will belong. Define *caliber*, l, to be the maximal number of C-points allowed in that list. The selection we propose here is based on both measures: the graph couplings w_{ij}'s and the algebraic couplings c_{ij}'s. Define for each $i \in F$ a coarse neighborhood $\bar{\bar{N}}^C(i) = \{j \in C : ij \in E\}$. Set D to be the maximal c_{ij} in $\bar{\bar{N}}^C(i)$. Construct a possibly smaller coarse neighborhood by including only nodes with strong algebraic coupling $\bar{N}^C(i) = \{j \in \bar{\bar{N}}^C(i) : c_{ij} \geq \beta * D\}$; we use $\beta = 0.5$. If $\bar{N}^C(i) > l$, then the final coarse neighborhood $N^C(i)$ will include the first l largest w_{ij}'s in $\bar{N}^C(i)$. If $\bar{N}^C(i) \leq l$, then $N^C(i) \leftarrow \bar{N}^C(i)$. This construction of the coarse neighborhood $N^C(i)$ of $i \in F$ is summarized in Algorithm 3. (In the results below we have used only $l = 1$ and $l = 2$.) The classical AMG interpolation matrix P (of size $|V| \times |C|$) is then defined by

$$P_{ij} = \begin{cases} w_{ij} / \sum_{k \in N^C(i)} w_{ik} & \text{for } i \in F,\ j \in N^C(i), \\ 1 & \text{for } i \in C,\ j = i, \\ 0 & \text{otherwise.} \end{cases} \tag{3.9}$$

P_{ij} represents the fraction of i that will belong to the jth aggregate.

Data: l, i, β
$\bar{\bar{N}}^C(i) \leftarrow \{j \in C \ : \ ij \in E\}$;
$D \ = \max_{j \in \bar{\bar{N}}^C(i)} c_{ij}$;
$\bar{N}^C(i) = \{j \in \bar{\bar{N}}^C(i) : c_{ij} \geq \beta * D\}$;
if $l < |\bar{N}^C(i)|$ **then**
  $N^C(i) \leftarrow$ the l largest w_{ij}'s in $\bar{N}^C(i)$;
if $l \geq |\bar{N}^C(i)|$ **then**
  $N^C(i) \leftarrow \bar{N}^C(i)$.

Algorithm 3: The coarse neighborhood $N^C(i)$

Coarse graph couplings. The coarse couplings are constructed as follows. Let $I(k)$ be the ordinal number in the coarse graph of the node that represents the aggregate around a seed whose ordinal number at the fine level is k. Following the weighted aggregation scheme used in [43], the edge connecting two coarse aggregates, $p = I(i)$ and $q = I(j)$, is assigned with the weight $w_{pq}^{(coarse)} = \sum_{k \neq l} P_{ki} w_{kl} P_{lj}$. The volume of the ith coarse aggregate is $\sum_j v_j P_{ji}$. Note that if the given graph is connected, it is guaranteed that the coarse graph will also be connected.

4. Computational results. We demonstrate the power of our new relaxation-based coarsening scheme by comparing its experimental results with those of the classical AMG-based coarsening for three important NP-hard optimization problems: the M2SP ((2.1) with $p = 2$), the minimum linear arrangement ((2.1) with $p = 1$), and the minimum 2-partitioning (2.2) problems. In all cases the results are obtained by taking the lightest possible uncoarsening schemes, so that differences due to the different coarsening schemes are least blurred.

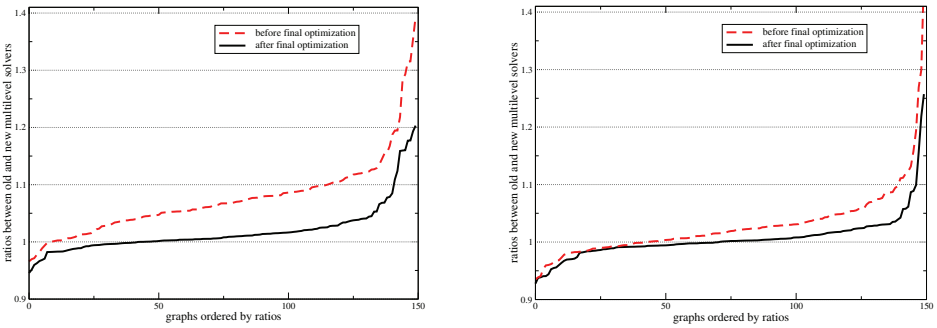
We have implemented and tested the new coarsening scheme by using the linear ordering packages developed in [41] and in [40] and the SCOTCH package [35] on a Linux machine. The implementation is nonparallel and has not been optimized. The

results should be considered only qualitatively and can certainly be improved by more advanced uncoarsening. Thus, no intensive attempt to achieve the best known results for the particular test sets was done. The details regarding the uncoarsening schemes for the above problems are given in [41, 40, 17].

4.1. The minimum *p*-sum problem. We present the numerical comparison for two M*p*SPs: the M2SP and the minimum linear arrangement. For these problems we have designed a full relaxation-based coarsening solver and evaluated it on a test set of 150 graphs of different nature, size ($|V| \leq 5 \cdot 10^6$ and $|E| \leq 10^7$), and properties. The test graphs are taken from [19] and from real-life network data such as social networks, power grids, and peer-to-peer connections. Our solvers are free and can be downloaded with detailed solutions for every graph from [39]. To emphasize the difference in the minimization results between the two coarsening schemes (the relaxation-based and the classical AMG-based schemes), we measure the results obtained at the end of the multilevel cycle *before* the final local optimization postprocessing (Gauss–Seidel relaxation and the LP in [41, 40]), as well as after its application. Moreover, we use small calibers, $l = 1, 2$, since these demonstrate more sharply the quality of matching between the F-points and the C-points. For higher calibers it is also important to use the adaptive BAMG scheme [9] for calculating the interpolation weights, which is beyond the scope of this work. Small calibers are important for maintaining the low complexity of the multilevel framework, which is vital, for example, for hypergraphs and expanders.

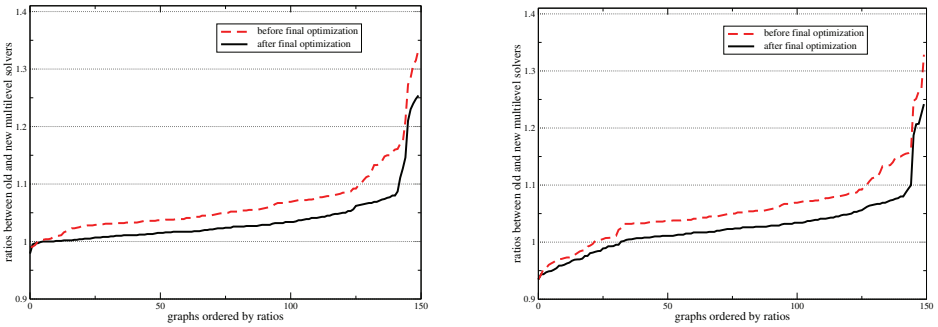
The minimum 2-sum problem (M2SP). A comparison of the relaxation-based and AMG-based coarsenings with calibers 1 and 2 is presented in Figures 4.1(a) and 4.1(b), respectively. Each x-axis scale division corresponds to one graph from the test set. The y-axis corresponds to the ratio between the average cost obtained by 100 runs of the AMG-based coarsening and the one obtained by 100 runs of the relaxation-based coarsening. (To obtain different results for a given graph we have reordered the nodes. That is, while coarsening the nodes are scanned in different orders. This reordering affects the set of seeds being chosen for the next coarse level and thus the entire coarsening itself.) Each figure contains two curves: the dashed curves with cost measurements before applying the postprocessing of local optimization (e.g., Gauss–Seidel relaxation, window minimization [41]) and the regular curves with cost measurements after adding such optimization steps. Clearly most graphs benefit from the relaxation-based coarsening, showing a ratio greater than 1. The ratio decreases when more optimization is used, especially since the Gauss–Seidel relaxation is a powerful algorithmic component for this problem and thus brings the results of the two coarsening schemes closer to each other as is indicated by the regular curves. All these results were obtained with $K = 10$ TVs. When we lowered K to 5, we observed no significant change in the results. Our number of Jacobi overrelaxation sweeps $r = 20$ cannot be reduced by more than twice since this relaxation scheme is expected to evolve slowly. The detailed analysis of the convergence properties is presented in [16].

The minimum linear arrangement problem. Similarly to the previous problem, we designed a relaxation-based solver and established a series of experiments for the minimum linear arrangement problem. The experimental setup was identical to that of the M2SP. It was based on the solver designed in [40]. In this case we can observe even more significant improvement when employing the relaxation-based coarsening than for the M2SP.



(a) $l = 1$ (b) $l = 2$

FIG. 4.1. *Results for the M2SP.*



(a) $l = 1$ (b) $l = 2$

FIG. 4.2. *Results for the minimum* 1*-sum (linear arrangement) problem.*

Which graphs are most beneficial? It is remarkable that the most beneficial graphs in our test set come from VLSI design and general optimization problems. We know that these graphs are very irregular (compared, for example, with finite-element graphs and with those that pose 2D/3D geometry). Thus, we may conclude that the algebraic couplings help to identify the weakness of nonlocal connections and prevent them from being aggregated. In several examples, we achieved the best known results with caliber 1, while using classical AMG-based approaches they can be achieved with bigger calibers only.

An algebraic coupling-based algorithm. We have also tried a straightforward algorithm in which, during coarsening, the weights of the graph are simply replaced by their algebraic couplings. That is, in the **if** statement at the end of Algorithm 2, only the first term is taken into account (namely, $(\sum_{j\in(C\cap N(i))} c_{ij}/\sum_{j\in N(i)} c_{ij}) \le Q$). Similarly, in Algorithm 3, w_{ij} (in the first **if**) is replaced by c_{ij}. We present the comparison of the obtained simple algebraic coupling-based coarsening scheme with the mixed scheme described in Algorithms 2 and 3 and in Figure 4.3. The comparison was done for the M2SP including postprocessing (of local optimization) using the same experimental setup. The bold curve corresponds to the ratios between the classical AMG-based results and the simple algebraic coupling-based coarsening scheme. To see the difference between this algorithm and the more elaborate one, we add a copy of its results, that is, the bold curve from Figure 4.1(a). The mixed version is clearly better: in about 25 more graphs the results are improved. The average improvement was 1.5%.

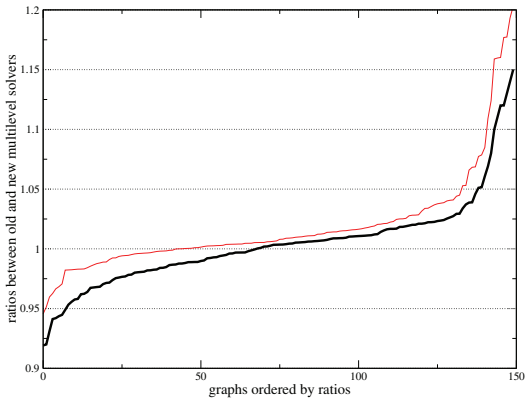


FIG. 4.3. *Results for the M2SP. Comparison of the algebraic distance based only and mixed full relaxation-based algorithms.*

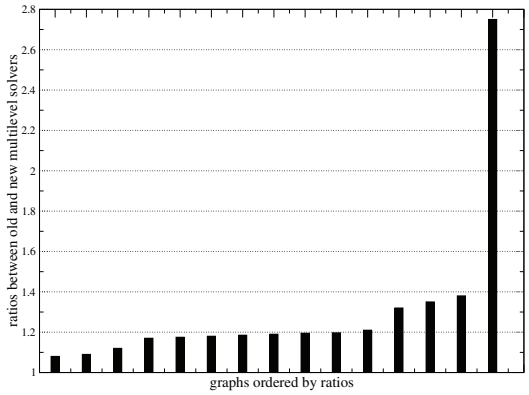


FIG. 4.4. *Results for the minimum 2-partitioning problem.*

4.2. The minimum 2-partitioning problem. We compared the relaxation-based coarsening and the classical AMG-based coarsening by combining two packages. The coarsening part was the same as in the M*p*SPs. The uncoarsening was based on the SCOTCH package; details of its fastest version can be taken from [17].

The comparison of the relaxation-based and the AMG-based coarsenings with caliber 1 is presented in Figure 4.4. The interpretation of x- and y-axes is similar to Figure 4.1. Included are 15 graphs of different nature and size. The details regarding the numerical results can be obtained from [39]. The four best ratios are obtained for graphs with power-law degree distributions. More results for the graph and hypergraph partitioning problems are reported in [16]. Even though the algorithm used there only substitutes the original given couplings by their algebraic couplings, it is already clear that better results are observed for most tested instances of both graphs and hypergraphs.

4.3. Running time. The implementation of stationary iterative processes and their running times is a well studied issue. These topics are beyond the scope of this paper; we refer the reader to two books in which one can find discussions about sequential and parallel matrix-vector multiplications and general relaxations [25, 26].

Typical running time of an AMG-based framework for linear ordering problems on graphs can be found in [40, 41, 42]. The introduction of the algebraic distance did not increase significantly those running time estimations.

5. Multiscale distance definition and hierarchical organization. As mentioned in the introduction, the algebraic distance defined above is only a crude *local distance*, measuring meaningful relative distances only between neighboring nodes while also detecting which nodes should not be considered as close neighbors. This fuzzy local distance, which can be calculated rapidly, is all we need for coarsening. A similar distance is then calculated at each coarser level, thus yielding a multiscale definition of distances through the entire graph, where at large distances one defines the distances only between (usually large) aggregates of nodes, not between any individual pair of distant nodes. Such multiscale distances are not only far less expensive to calculate: we next list several reasons why, *in principle*, distances in a general large graph should be *defined* better in such a multiscale fashion.

- At large distances the detailed individual distances (the exact travelling time from each house in Baltimore to each house in Boston, say) are usually not of interest.
- The distance in a general graph is a fuzzy notion, whose definition is to a certain extent arbitrary. Consider two neighboring nodes a and b and a third node c which is far from both. The difference between the distances between a and c and between b and c is not really meaningful for a few reasons: it will come out differently when measured by two different legitimate distance definitions, and it might change by much when a slight change is introduced into the graph data (e.g., into its edge weights), or upon inaccurately solving the equations that define these distances (e.g., when measuring diffusion maps; see below).
- The most important reason is that at different scales different factors should in principle enter into the distance definition. In particular, at increasingly larger distances, intrinsic properties of increasingly larger aggregates should play a progressively more important role. For example, in image segmentation, while at the finest level the "closeness" of two neighboring pixels (i.e., their chance to belong to the same segment) can be defined by their color similarity, at larger scales the closeness of two neighboring patches should be defined in terms of the similarity in their *average* color (which is different from the direct color similarity of neighboring pixels along the boundary between the patches) and also in terms of similarity in various texture measures (color variances, shape moments of subaggregates, average orientation of fine embedded edges, etc.) and other aggregative properties [43, 44]. Another example is that in the problem of identifying clusters in a large set of points in R^d, at the finest level the distance between data points can simply be their Euclidean distance, while at coarser levels the distance between two aggregates of points should also take into account similarity in terms of aggregative properties, such as density, orientation, and dimensionality [32].
- The multiscale definition of distance also brings much needed flexibility into the way distances at one level are converted into distances at coarser levels. For example, in a graph whose finest level consists of face images and their similarity scores, if at some coarse-level node A is the union of two fine-level nodes A_1 and A_2, and node B is the union of B_1 and B_2, then the coarse weight w_{AB} of the edge (A, B) can be defined either as some *average* of

$w_{A_1B_1}$, $w_{A_1B_2}$, $w_{A_2B_1}$, and $w_{A_2B_2}$, or alternatively as the *maximum* (or L_p average with large p) of those four weights. The former choice (average) is more suitable if one wants to cluster faces having a *similar pose*, while the latter choice (max or L_p) is more suitable if we need clusters of images each belonging to the *same person* (or, generally, when the clustering should be based on transitive similarity).

An ingenious rigorous definition of distances in a general graph, introduced in [18], is called *diffusion distance*. Denoting by $p(t, y|x)$ the probability of a random walk on the graph starting at x to reach y after t steps, the diffusion distance between two nodes x_i and x_j is defined by

$$d(x_i, x_j, t)^2 = \sum_y w(y)[p(t, y|x_i) - p(t, y|x_j)]^2 \tag{5.1}$$

with some suitable choice of the node weights w. This is, in fact, a multiscale definition of distance, with the diffusion time t serving as the scaling parameter. And indeed the definition is used for hierarchical organizations of graphs (even though large-scale distances are still defined in detail for any pair of nodes). The calculation of our "algebraic distance" can be viewed as just a fast way to compute a crude approximation to diffusion distances at some small t.

The essential practical point is that this crude and inexpensive "algebraic distance" is all one needs for solving graph problems by repeated coarsening. The calculation of the *diffusion map* (the diffusion distances at various scales t) for a large graph is, on the other hand, quite expensive, requiring computing (possibly many) eigenvectors of the graph Laplacian. The fast way to calculate them should involve using a multiscale algorithm such as AMG (which is likely to work well in those cases where hierarchical organizations of the graph are meaningful; the AMG solver can, by the way, calculate *many* eigenvectors for nearly the same work of calculating only one [33]). However, instead of calculating the diffusion map and then using it for organizing the graph, *the AMG structure can itself be used directly, and more efficiently for any such organization.*

Indeed, as pointed out in [9], the same coarsening procedures used by the AMG solver can directly be used for efficient hierarchical organizations (such as multiscale clustering) of a graph (as in [32]) or for multiscale segmentation of an image (as in [43, 44]). As exemplified in this article (and also in [41, 42]), this kind of procedure can also be used for many other types of graph problems; in particular, it can also be used for detecting small hidden cliques in random graphs [20].

Thus, for discrete graphs, and analogously also for related continuum field problems, although the diffusion map is a useful theoretical concept, it is often not the most practical tool. We believe this to be true for most if not all spectral graph methods (using eigenvectors of the graph Laplacian): the same AMG structure that would rapidly calculate the eigenvectors can be better used to directly address the problem at hand. As pointed out in the discussion of multiscale distances, this can yield not just faster solutions, but also, and more importantly, better definitions and more tunable treatments for many practical problems.

6. Conclusions. We have proposed a new measure that quantifies the "closeness" between two nodes in a given graph. The calculation of the measure is linear in the number of edges in the graph and involves just a small number of relaxation sweeps. The calculated measure is all that is required for coarsening purposes. A similar notion of distance is then calculated and used at each coarser level. We demonstrate

the use of this new measure for the minimum (1,2)-sum linear ordering problem and for the minimum 2-partitioning problem. The improvement in the results shows that this measure indeed detects the most important couplings in the graph and helps in producing a better coarsening, while at the same time preventing nonlocal vertices from belonging to the same coarse aggregate.

Acknowledgment. We express our gratitude to Dr. Cédric Chevalier for providing us with a modification of Scotch software and helping to design the minimum 2-partitioning problem experiments.

REFERENCES

[1] A. Abou-Rjeili and G. Karypis, *Multilevel algorithms for partitioning power-law graphs*, in Proceedings of the IEEE International Parallel and Distributed Processing Symposium, 2006.

[2] C. J. Alpert, J.-H. Huang, and A. B. Kahng, *Multilevel circuit partitioning*, in Proceedings of the Design Automation Conference, 1997, pp. 530–533.

[3] R. Banos, C. Gil, J. Ortega, and F. G. Montoya, *Multilevel heuristic algorithm for graph partitioning*, in Proceedings of the International Conference on Applications of Evolutionary Computing, 2003, Springer-Verlag, Berlin, pp. 143–153.

[4] S. T. Barnard and H. D. Simon, *A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems*, Concurrency: Practice and Experience, 6 (1994), pp. 101–107.

[5] H. W. J. Blöte, J. R. Heringa, A. Hoogland, E. W. Meyer, and T. S. Smit, *Monte Carlo renormalization of the 3D Ising model: Analyticity and convergence*, Phys. Rev. Lett., 76 (1996), 2613.

[6] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.

[7] A. Brandt, *Algebraic multigrid theory: The symmetric case*, Appl. Math. Comput., 19 (1986), pp. 23–56.

[8] A. Brandt, *General highly accurate algebraic coarsening*, Electron. Trans. Numer. Anal., 10 (2000), pp. 1–20.

[9] A. Brandt, *Multiscale scientific computation: Review* 2001, in Multiscale and Multiresolution Methods: Theory and Applications, Lecture Notes in Comput. Sci. 20, T. J. Barth, T. Chan, and R. Haimes, eds., Springer-Verlag, New York, 2001.

[10] A. Brandt, *Principles of systematic upscaling*, in Bridging the Scales in Science and Engineering, J. Fish, ed., Oxford University Press, Oxford, 2008, pp. 193–216.

[11] A. Brandt, S. McCormick, and J. Ruge, *Algebraic Multigrid (AMG) for Automatic Multigrid Solution with Application to Geodetic Computations*, Technical report, Institute for Computational Studies, Fort Collins, CO, 1982.

[12] A. Brandt, S. McCormick, and J. Ruge, *Algebraic multigrid (AMG) for sparse matrix equations*, in Sparsity and its Applications, D. J. Evans, ed., Cambridge University Press, Cambridge, 1984, pp. 257–284.

[13] A. Brandt and D. Ron, *Renormalization multigrid (RMG): Statistically optimal renormalization group flow and coarse-to-fine Monte Carlo acceleration*, J. Stat. Phys., 102 (2001), pp. 231–257.

[14] A. Brandt and D. Ron, *Chapter 1 : Multigrid solvers and multilevel optimization strategies*, in Multilevel Optimization and VLSICAD, J. Cong and J. R. Shinnerl, eds., Kluwer, Norwell, MA, 2003, pp. 1–69.

[15] T. N. Bui and B. R. Moon, *Genetic algorithm and graph partitioning*, IEEE Trans. Comput., 45 (1996), pp. 841–855.

[16] J. Chen and I. Safro, *Algebraic Distance on Graphs*, Technical report ANL/MCS-P1696-1009, Argonne National Laboratory, Chicago, 2009.

[17] C. Chevalier and I. Safro, *Comparison of coarsening schemes for multilevel graph partitioning*, in Proceedings of the 4th Learning and Intelligent Optimization Conference, 2009.

[18] R. R. Coifman and S. Lafon, *Diffusion maps*, Appl. Comput. Harmon. Anal., 21 (2006), pp. 5–30.

[19] T. Davis, *University of florida sparse matrix collection*, NA Digest, 97, 1997.

[20] U. Feige and D. Ron, *Finding hidden cliques in linear time*, Discrete Math. Theor. Comput. Sci., 1 (2010), pp. 189–204.

[21] C. M. Fiduccia and R. M. Mattheyses, *A linear-time heuristic for improving network partitions*, in Proceedings of the 19th Design Automation Conference, 1982, IEEE, pp. 175–181.
[22] M. R. Garey, D. S. Johnson, and L. Stockmeyer, *Some simplified NP-complete problems*, in Proceedings of the Sixth Annual ACM Symposium on Theory of Computing, 1974, ACM Press, pp. 47–63.
[23] M. R. Garey and D. S. Johnson, *Computers and Interactability. A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Fransisco, 1979.
[24] A. George and A. Pothen, *An analysis of spectral envelope reduction via quadratic assignment problems*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 706–732.
[25] A. Grama, G. Karypis, A. Gupta, and V. Kumar, *Introduction to Parallel Computing: Design and Analysis of Algorithms*, Addison-Wesley, Reading, MA, 2003.
[26] M. T. Heath, *Scientific Computing*, McGraw-Hill, Inc., New York, 2002.
[27] B. Hendrickson and R. W. Leland, *A multi-level algorithm for partitioning graphs*, in Proceedings of the ACM/IEEE Conference on Supercomputing, 1995.
[28] Y. F. Hu and J. A. Scott, *A multilevel algorithm for wavefront reduction*, SIAM J. Sci. Comput., 23 (2001), pp. 1352–1375.
[29] G. Karypis and V. Kumar, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM J. Sci. Comput., 20 (1998), pp. 359–392.
[30] G. Karypis and V. Kumar, *METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices*, University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Center, Minneapolis, MN, 4 edition, 1998.
[31] B. W. Kernighan and S. Lin, *An efficient heuristic procedure for partitioning graphs*, BELL Syst. Tech. J., 49 (1970), pp. 291–307.
[32] D. Kushnir, M. Galun, and A. Brandt, *Fast multiscale clustering and manifold identification*, Pattern Recognition, 39 (2006), pp. 1876–1891.
[33] D. Kushnir, M. Galun, and A. Brandt, *Efficient multilevel eigensolvers with applications to data analysis tasks*, IEEE Trans. Pattern Anal. Mach. Intell., 32 (2010), pp. 1377–1391.
[34] H. Meyerhenke, B. Monien, and T. Sauerwald, *A new diffusion-based multilevel algorithm for computing graph partitions of very high quality*, in Proceedings of the 22nd International Parallel and Distributed Processing Symposium, 2008, IEEE.
[35] Scotch, *Static Mapping, Graph Partitioning, and Sparse Matrix Block Ordering Package*, http://www.labri.fr/~pelegrin/scotch/.
[36] A. Pothen, H. D. Simon, and K.-P. Liou, *Partitioning sparse matrices with eigenvectors of graphs*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 430–452.
[37] D. Ron, S. Wishko-Stern, and A. Brandt, *An Algebraic Multigrid Based Algorithm for Bisectioning General Graphs*, Technical report MCS05-01, Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot, Israel, 2005.
[38] J. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid Methods, S. F. McCormick, ed., SIAM, Philadelphia, 1987, pp. 73–130.
[39] I. Safro, *Homepage of our Projects*, http://www.mcs.anl.gov/~safro.
[40] I. Safro, D. Ron, and A. Brandt, *Graph minimum linear arrangement by multilevel weighted edge contractions*, J. Algorithms, 60 (2006), pp. 24–41.
[41] I. Safro, D. Ron, and A. Brandt, *Multilevel algorithm for the minimum 2-sum problem*, J. Graph Algorithms Appl., 10 (2006), pp. 237–258.
[42] I. Safro, D. Ron, and A. Brandt, *Multilevel algorithms for linear ordering problems*, ACM J. Exp. Algorithmics, 13 (2008), pp. 1.4–1.20.
[43] E. Sharon, A. Brandt, and R. Basri, *Fast multiscale image segmentation*, in Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, 2000, pp. 70–77.
[44] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt, *Hierarchy and adaptivity in segmenting visual scenes*, Nature, 442 (2006), pp. 810–813.
[45] C. Walshaw, *Multilevel refinement for combinatorial optimisation problems*, Ann. Oper. Res., 131 (2004), pp. 325–372.