Fast Multilevel Algorithms for Linear Ordering Problems

Ilya Safro Dorit Ron Achi Brandt

The Weizmann Institute of Science
POB 26, Rehovot 76100, Israel

Linear ordering problems are combinatorial optimization problems which deal with the minimization of different functionals in which the graph vertices are mapped onto $(1, 2, ..., n)$. These problems are widely used and studied in many practical and theoretical applications. In this review we summarize a variety of linear-time algorithms for these problems inspired by the Algebraic Multigrid approach which is based on weighted edge contraction. The experimental results for four such problems turned out to be better than every known results in almost all cases, while the short running time of the algorithms enables applications on very large graphs.

1 Introduction

The objective of the class of linear ordering problems is to minimize different functionals that map the set of the graph vertices onto $(1, 2, ..., n)$. This class contains many graph (or matrix) layout problems such as : the minimum p-sum (MpSP), the bandwidth, the workbound reduction, the wavefront, the envelope size, etc. Some problems, such as finding the minimum linear arrangement [27] or the bandwidth [21], appear in many applications for solving problems in the large sparse matrix computation. Some other are closely related to the problem of calculating the envelope size of a symmetric matrix or, more precisely, to the amount of work needed in the Cholesky factorization of such a matrix [15]. Linear ordering problems may also be motivated as a model used in VLSI design [10] and may be used in several biological applications, graph drawing and other fields (see [13, 16, 21, 29]). Commonly for general graphs (or matrices) these problems are NP-hard and their decision versions are NP-complete [14].

Since these problems have a practical significance, many heuristic algorithms were developed in order to achieve near optimal solution. Among the most successful are spectral sequencing [18], optimally oriented decomposition tree [1], multilevel based [17,20], simulated annealing [22] and others. Some of these algorithms have proven themselves superior in solution quality while others in execution time.

One of the most popular and exploitable method designed to achieve a suitable linear ordering for different problems is the spectral sequencing (SS) [18]. This approach consists of ordering the graph vertices according to the sorted coordinates of the second eigenvector of the graph Laplacian. The heuristic argumentation of SS is based on the fact that the *continuous* version of the minimum 2-sum problem can be solved by this method to the optimum [18]. In practice, for the (discrete) minimum 2-sum it was shown in [28] that the direct application of SS (without additional reinforcement postprocessing) on ”real world instances” does not achieve good enough results, while the lower bounds based on SS are very far from the best known ordering costs. Rather poor results of the *exact* SS were presented in [11] for the minimum bandwidth problem. Better results were shown there by using different *approximated* SS, i.e., by calculating the second eigenvector *less* precisely. In fact, they have tested 19 algorithms (17 of which are different versions of SS) and presented the best achieved results among all. In Section 4 we show the significant improvement achieved by our algorithm over all those algorithms: on the average our results were better by 34%.

In this review we present a general framework of multilevel algorithms especially designed for linear ordering problems. Our strategy is based on the Algebraic MultiGrid scheme (AMG) [3,5,6,9,25,31,32]. We demonstrate how the building blocks of the general multilevel approach can be used in several ways to make it suitable for solving various functionals. In particular, three approaches are presented: (1) the basic multilevel scheme for solving the *minimum* 2*-sum* problem; (2) a continuation method in which a sequence of increasingly p-sum problems are involved until a desired p is reached, e.g., the *bandwidth* of a graph can be approximated by large enough p that can be considered infinite for practical purposes; (3) in addition, we propose to use the ordering obtained by the minimum 2-sum problem as a first approximation for other linear ordering problems, which is then improved by a postprocessing of local minimizations with actual use of the relevant functional. This approach is demonstrated for the *workbound* reduction problem and for the *wavefront* reduction problem.

The main objective of a multilevel based algorithm is to create a hierarchy of problems, each representing the original problem, but with fewer degrees of freedom. General multilevel techniques have been successfully applied to various areas of science (e.g. physics, chemistry, engineering, etc.) [4,7]. AMG methods were originally developed for solving linear systems of equations resulting from the discretization of partial differential equations. Lately they have been applied to various other fields, yielding for example novel methods for image segmentation [30] and for the linear arrangement problem [27]. In the context of graphs it is the Laplacian matrix that represents the related set of equations. The main difference between our approach to most other multilevel approaches (related to various graph optimization problems) is the coarsening scheme. While the previous approaches may be viewed as *strict* aggregation process, the AMG coarsening is actually a $weighted$ aggregation : each node may be divided into *fractions*, and different fractions belong to different aggregates. This enables more freedom in solving the coarser levels and avoids making hardened local decisions, such as edge contractions, before accumulating the relevant global information.

One of the important achievements of our work is the general coarsening scheme that turns out to be suitable for all the different functionals we have tested. This fact can be explained by the way the hierarchy of problems is constructed: variables are eliminated within the coarsening phase only and exactly when they show strong dominant connections to the remaining (non-eliminated) variables, this in turn guarantees that the solution of the eliminated variables is naturally obtained once the non-eliminated variables are solved. The various algorithms thus differ only in the *disaggregation* process which follows by projecting to a finer level the final arrangement obtained on a coarser level. This initial fine level arrangement is being further improved by applying different local reordering methods. We have developed a simultaneous minimization of several vertices called Window Minimization. In its basic application (for the 2-sum problem [28]) it involves the minimization of a quadratic form. More evolved functionals can be used after quadratization. Also, we suggest the use of numerical calculation rather than analytic, for instance, in calculating derivatives. Finally, our postprocessing is intensified by Simulated Annealing (SA) [19] which is a general stochastic method to escape local minima. In the multilevel framework SA is aimed at searching only for *local* changes that guarantee the preservation of large-scale solution features inherited from coarser levels. This stochastisity turned out to be important especially for the *minimum linear arrangement* problem.

The discussion about theoretical complexity issues, such as lower and upper bounds for the solution cost, is beyond of the scope in this review. We are not interested in worst possible scenarios nor in random instances. Our focus is on practical high-performance and low computational cost algorithms that will outperform existing algorithms by providing better results in less running time. For that purpose we used a known benchmark [12] from which we took graphs of various origins and sizes including very large instances. Our multilevel algorithm exhibit linear complexity, i.e., the computational cost is proportional to $|V| + |E|$.

We compared the results obtained by our multilevel algorithms with many previously described algorithms. In this chapter we summarize the results of the minimum 2-sum, the bandwidth and the workbound problems and show that our results are on the average better than previous ones by about 30%, while the running time for graphs with about 10^4 nodes and 10^5 edges is less than one minute on 1.7GHz PC. In general, our experimental results show that the AMG framework can be used for linear ordering problems to obtain high quality results in linear time while using the exact same set of parameters. The implemented algorithm can be downloaded from [26].

The review is arranged as follows. The various functionals and their generalizations are described in Section 2. The multilevel algorithm along with additional optimization techniques are presented in Section 3. A comparison of our results with other works is finally summarized in Section 4.

2 Definitions and generalizations

Given a weighted graph $G = (V,\ E)$, where $V = \{1, 2, ..., n\}$, denote by w_{ij} the non-negative weight of the edge ij between nodes i and j (if $ij \notin E$ then $w_{ij} = 0$). The MpSP is defined by

$$\text{minimize over } \pi \quad \sigma_p(G,\pi) = \sum_{ij} w_{ij} |\pi(i) - \pi(j)|^p \quad , \tag{1}$$

where π is a permutation over V. During the multilevel solver, each vertex i is assigned with a *volume* v_i, thus (1) becomes $\min_\pi \sigma_p(G,x) = \sum_{ij} w_{ij}|x_i - x_j|^p$, where $x_i = \frac{v_i}{2} + \sum_{k,\pi(k)<\pi(i)} v_k$. The bandwidth problem is defined by

$$bw(G) = \min_{\pi} \max_{ij} |\pi(i) - \pi(j)| = \min_{\pi} \lim_{p \to \infty} (\sigma_p(G,\pi))^{1/p}, \tag{2}$$

and the workbound reduction problem by

$$\text{minimize over } \pi \quad wb(G,\pi) = \sum_i \max_{j,\pi(j)<\pi(i)} w_{ij}(\pi(i) - \pi(j))^2 \quad , \tag{3}$$

where the $\max$ function may be approximated by

$$wb(G,x) \approx \sum_i \Big(\sum_{j:x_j<x_i} w_{ij}(x_i - x_j)^p \Big)^{2/p}. \tag{4}$$

3 General multilevel scheme

In the multilevel framework a hierarchy of decreasing size graphs : $G_0, G_1, ..., G_k$ is constructed. Starting from $G_0 = G$, create by *coarsening* the sequence $G_1, ..., G_k$, then solve the coarsest level directly, and finally *uncoarsen* the solution back to G. This entire process is called a *V-cycle*.

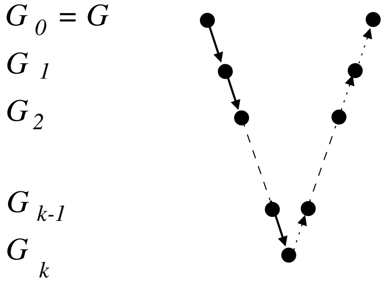

Figure 1: The Scheme of a V-cycle. Solid arrows stand for coarsening, dotted for uncoarsening.

Coarsening: Weighted Aggregation. The construction of a coarse graph is divided into three stages: (a) a subset of the fine nodes is chosen to serve as the *seeds* of the aggregates (which form the nodes of the coarser level), (b) the rules for interpolation are determined, and (c) the weights of the edges between the aggregates are calculated.

Coarse Nodes. 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. Starting from $C = \emptyset$ and $F = V$, nodes are transferred from F to C until all remaining $i \in F$ satisfy $\sum_{j \in C} w_{ij} / \sum_{j \in V} w_{ij} \geq 0.4$.

The Coarse Problem. Define for each $i \in F$ a coarse neighborhood N_i consisting of C-nodes to which i is connected. Let $I(j)$ 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 j. The classical AMG interpolation matrix P is defined by

$$P_{iI(j)} = \begin{cases} w_{ij} / \sum\limits_{k \in N_i} w_{ik} & \text{for } i \in F,\ j \in N_i \\ 1 & \text{for } i \in C,\ j = i \\ 0 & \text{otherwise} \ . \end{cases} \tag{5}$$

$P_{iI(j)}$ thus represents the likelihood of i to belong to the $I(j)$-th aggregate. The edge connecting two coarse aggregates p and q, is assigned with the weight $w_{pq} = \sum_{k \neq l} P_{kp} w_{kl} P_{lq}$. The volume of the p-th coarse aggregate is $\sum_j v_j P_{jp}$.

The coarsest level. Solving the appropriate functional at the coarsest level is performed by trying all possible arrangements. Since the amount of work investigated at the coarsest levels is negligible compared with that of the finest levels, many solutions can in fact be kept at each level whose graph is relatively small with respect to G.

Disaggregation (uncoarsening). Having solved a coarse problem, the solution to the next-finer-level problem is initialized by first placing the seeds according to the coarse order and then adjusting all other F-nodes while aiming at the minimization of the arrangement cost. This first approximation is subsequently improved by several $relaxation$ sweeps, first compatible, then regular. Finally, the arrangement is improved by strict minimization. Details follow below.

Initialization. Given is the arrangement of the coarse level aggregates in its generalized form, put first each seed $j \in C$ at $y_j = x_{I(j)}$. Next define $V' \subset V$ to be the subset of nodes that have already been placed, so we start with $V' = C$. Then proceed by positioning each fine node $i \in V \setminus V'$ at the coordinate y_i in which the cost of the arrangement, at that moment when i is being placed, is minimized. For the MpSP: if $p = 1$ then $y_i \in \{y : |\sum_{y_j < y,\, j \in V'} w_{ij} - \sum_{y_j > y,\, j \in V'} w_{ij}| \text{ is minimal}\}$; if $p = 2$ then $y_i = \sum_{j \in V'} y_j w_{ij} / \sum_{j \in V'} w_{ij}$; and for a general (even) p y_i has to minimize

$$\sum_{j \in V'} w_{ij} (y_i - y_j)^p$$

(can be achieved by several Newton-Rhapson steps). Then $V' \leftarrow V' \cup \{i\}$ and the process continues until $V' = V$. Finally, in order to take volumes into account, each position y_i is changed to

$$x_i = \frac{v_i}{2} + \sum_{y_k < y_i} v_k \quad . \tag{6}$$

Relaxation. The arrangement obtained after the initialization is a first feasible solution for the MpSP which is then improved by employing several sweeps of $relaxation$, first $compatible$ then $Gauss$-$Seidel$-$like$. These two types of relaxation are very similar to the above initialization: The compatible relaxation, improves the positions of the F-nodes one by one according to the minimization criteria above (where $V' = V$), while keeping the positions of the seeds (C-nodes) unchanged. The Gauss-Seidel-like relaxation is similarly performed, but for all nodes (including C). Each such sweep is again followed by (6).

Window Minimization (WM). The cost of the arrangement can be further reduced by *strict minimization*, a sequence of rearrangement that accepts only changes which decrease the arrangement cost. We first describe the basic WM involving the quadratic form for $p = 2$. Given a current approximation $\tilde{x}$ to the arrangement of the graph, denote by δ_i a $correction$ to $\tilde{x_i}$. Let $\mathfrak{W} = \{\pi^{-1}(s+1), ..., \pi^{-1}(s+q)\}$ be a $window$ of q sequential vertices in the current arrangement. The local minimization problem of the $p = 2$ functional is

$$\text{minimize } \sigma_2(\mathfrak{W}, \tilde{x}, \delta) = \sum_{i,j\in\mathfrak{W}} w_{ij}(\tilde{x_i}+\delta_i-\tilde{x_j}-\delta_j)^2 + \sum_{i\in\mathfrak{W}, j\notin\mathfrak{W}} w_{ij}(\tilde{x_i}+\delta_i-\tilde{x_j})^2. \quad (7)$$

To prevent the possible convergence of many coordinates to one point, and, more precisely, to express the aim of having $\{x_i + \delta_i\}_{i\in\mathfrak{W}}$ an approximate permutation of $\{x_i\}_{i\in\mathfrak{W}}$ one should add constraints of the form

$$\sum_{i\in\mathfrak{W}} (\tilde{x_i} + \delta_i)^m v_i \;=\; \sum_{i\in\mathfrak{W}} \tilde{x_i}^m v_i \;, \quad m = 1, 2 \quad (8)$$

where for $m = 2$ we have neglected the quadratic term in δ_i. Note that the sums $\sum_{i\in\mathfrak{W}} \tilde{x_i}^m v_i$ for $m = 1, 2$ are invariant under permutations. Using Lagrange multipliers λ_1 and λ_2, the final formulation of the WM for $p = 2$ is the following system of equations:

$$\begin{cases} \sum_{j\in\mathfrak{W}} w_{ij}(\delta_i - \delta_j) + \delta_i \sum_{j\notin\mathfrak{W}} w_{ij} + \lambda_1 v_i + \lambda_2 v_i \tilde{x_i} = \sum_j w_{ij}(\tilde{x_j} - \tilde{x_i}) & \text{for } i \in \mathfrak{W} \\ \sum_i \delta_i v_i = 0 \\ \sum_i \delta_i v_i \tilde{x_i} = 0 \quad . \end{cases} \quad (9)$$

The use of WM for non-quadratic functional is achieved by quadratization. For $p > 2$, define $\widehat{w}_{ij} = w_{ij}(\tilde{x_i} - \tilde{x_j})^{p-2}$ and the WM follows by substituting w_{ij} with $\widehat{w}_{ij}$ in (7) and (9).

For the workbound reduction problem the WM of a concrete $\mathfrak{W}$ can be approximated by

$$wb_p(\mathfrak{W}, \tilde{x}, \delta) = \sum_{i\in\mathfrak{W}} \Big(\sum_{j\in\mathfrak{W},\ \tilde{x}_j<\tilde{x}_i} w_{ij}(\tilde{x_i}+\delta_i-\tilde{x_j}-\delta_j)^p + \sum_{j\notin\mathfrak{W},\ \tilde{x}_j<\tilde{x}_i} w_{ij}(\tilde{x_i}+\delta_i-\tilde{x_j})^p \Big)^{2/p}. \quad (10)$$

The quadratization of (10) is achieved by Taylor expansion up to the third term.

Adding stochastisity. A general method to escape false local minima and advance to lower costs is to replace the strict minimization by a process that still accepts each candidate change which lowers the cost, but also assigns a positive probability for accepting a candidate step which increases the cost of the arrangement. The probability assigned to a candidate step is equal to $exp(-\Delta/T)$, where $\Delta > 0$ measures the $increase$ in the arrangement cost and $T > 0$ is a *temperature-like* control parameter which is gradually decreased toward zero. This process, known as $Simulated\ Annealing$ (SA) [19], in large problems would usually need to apply $very\ gradual$ cooling (decrease of temperatures), making it extremely slow and inefficient for approaching the global optimum.

In the multilevel framework, however, the role of SA is somewhat different. At each level it is assumed that the $global$ arrangement of aggregates has been inherited from the coarser levels, and thus only $local$, small-scale changes are needed. For that purpose, we have started at relatively high T, lowered it $substantially$ at each subsequent sweep, until strict minimization is employed.

Repeated heating and cooling is successively employed for better search over the local landscape. This search is further enhanced by the introduction of a "memory"-like tool consisting of an additional permutation which stores the *Best-So-Far* (BSF) observed arrangement, which is being occasionally updated by a procedure called *Lowest Common Configuration* (LCC) [8]. LCC enables the systematic accumulation of sub-permutations over a sequence of different arrangements, such that each BSF sub-permutation exhibits the best (minimal) sub-order encountered so far. The cost of the obtained BSF is at most the lowest cost of all the arrangements it has observed, and usually it is lower. The use of LCC becomes more important for larger graphs, where it is expected that the optimum of a subgraph is only weakly dependent on other subgraphs. The complete description of the LCC algorithm is given in [27].

We have tried to apply SA for all mentioned functionals as well as in [24, 27, 28]. The influence of this method on the bandwidth and workbound functionals was insignificant. The final results were changed within their standard deviation. However, in our previous work on the minimum linear arrangement [27] and in [24], we observed quite an impressive improvement while applying simulated annealing. Thus, we suggest to use it in the general multilevel scheme for various functionals.

Continuation method. The disaggregation scheme for the minimization of $\sigma_p(G, x)$ is based on $continuation$ in the parameter p, such that $p = 2$ is used to exactly solve the coarsest level, then at each subsequent finer level p is increased. Thus, every level l minimizes $\sigma_p(G_l, x)$ by initialization from $\sigma_{p-2}(G_{l+1}, x)$. The successive increase in p is continued, if needed, at the end of the V-cycle during the strict minimization.

4 Results and Related Works

The Minimum linear arrangement [27]. We have tested our algorithm on the benchmarks provided by Petit [22] and Koren [20]. Most successful competitive heuristics were : Spectral Sequencing, Optimally Oriented Decomposition Tree, Multilevel based, Simulated Annealing, Genetic Hillclimbing and some of their combinations. The test suite provided in [22] contains rather small graphs for which our algorithm gave the best costs (in almost all cases) in comparison to all previously listed heuristics. The running time was so negligible, that comparison was meaningless. The most interesting result was the comparison of our AMG-like algorithm with the combination of spectral and multilevel approaches [20] on very large graphs (introduced there). The fast version of our algorithm which run only a fifth of the time of [20] exhibited an average improvement of 7%. Our slower but more evolved (which basically applies more simulated annealing) version improved the costs of [20] by 12% (for complete list of results see [27]).

Another algorithm that was able to deal with these large-scale instances has been recently published by Rodriguez-Tello et al. [23]. This algorithm is based on a sophisticated evaluation of the σ_1 functional that was used in a two-stage simulated annealing framework. Spending on the average at least *thirteen* times more computational time than our slower version, their results show an average improvement of 1.3%.

The Minimum 2-sum [28]. We have found only one article [15] with an implemented algorithm and numerical results for the minimum 2-sum problem. The algorithm is based on the spectral approach. Since their test suite is relatively small to provide enough information regarding the problem, we have launched a new, much larger test suite and compared our results to the spectral approach. Our multilevel algorithm without any minimization at the finest level provided much better results (better by an average of 31.4%) than the spectral one, see Figure 2. Finally, the minimization process applied after both strategies has proven itself to be good enough for both of the approaches and almost equalized the results. For complete list of results see [28].

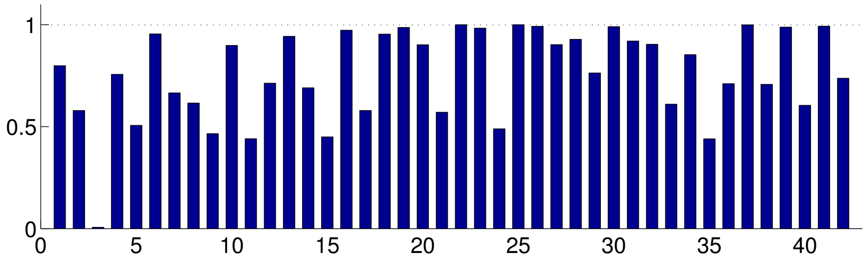

Figure 2: Minimum 2-sum ratios for the 43 graphs of [28]. Each bar represents a ratio between our σ_2 average of 100 trials and previously best known σ_2 values.

Bandwidth. We have chosen to test our algorithm on the test suites of [2, 11] which also include large enough inputs to make the picture complete. The ratios of our results to the best known values from [2, 11] are depicted in Fig. 3. The experimental results of each graph are described by a triple of bars, where every bar corresponds to a V-cycle with different number of WM sweeps. The first (second, third) bar corresponds to the V-cycle with 5 (10, 200) WM sweeps at all (all, the finest) levels. Each bar represents a ratio between our σ_∞ average of 100 trials and previously best known σ_∞ value. The third bar shows improvement of 34% on the average over the previously best known values.

Workbound reduction. We present the results of the workbound reduction problem as ratios to the best values from [11] (Fig. 4). Each graph is represented by a pair of

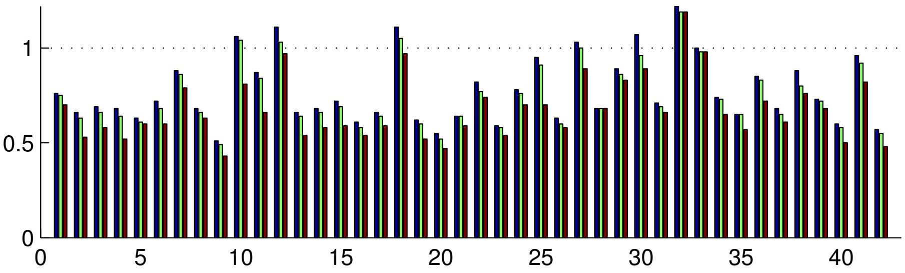

Figure 3: Bandwidth ratios for the 43 graphs of [2] and [11].

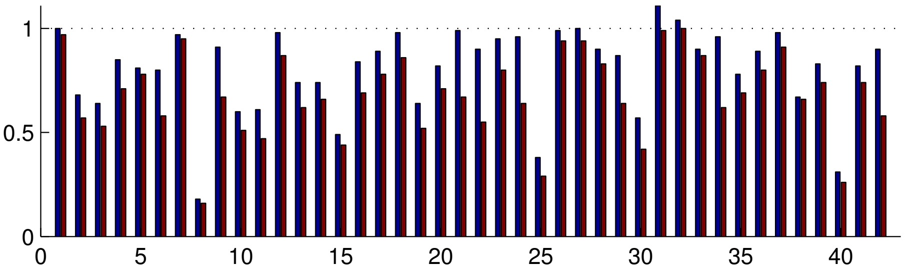

Figure 4: Workbound ratios for the 43 graphs of [2] and [11].

bars. Each bar represents a ratio between our wb average of 100 trials to those in [11]. The left bar corresponds to the first approximation obtained by evaluating the workbound functional over the solution of $\sigma_2(G)$ without any postprocessing. The second bar includes a postprocessing of several iterations of WM showing an improvement of 31% on the average over previously known values.

Wavefront reduction. As an additional experiment aimed at checking whether the M2SP may provide a good first approximation for another functional, we tested it for the $wavefront\ reduction$ problem defined by $wf(G,\pi) = \left(\sum_i |f_i|^2/n\right)^{1/2}$, where $f_i = adj(\{\cup_{j=1}^{i}\pi^{-1}(j)\})\bigcup\{\pi^{-1}(i)\}$ and $adj(X) = \bigcup_{j\in X}\{k : kj \in E\}\backslash X$. We have $evaluated$ $wf(G,\pi)$ for 15 graphs on the solution of $\sigma_2(G)$, without further optimizations, and obtained similar results to those presented in [17].

5 Conclusion

We have presented a variety of multilevel algorithms for the class of linear ordering problems for general graphs. These algorithms are based on the general principle that during coarsening each vertex may be associated to more than just one aggregate according to some “likelihood” measure. The uncoarsening initialization, which produces the first arrangement of the fine graph nodes, strongly relies on energy considerations (unlike usual interpolation in classical AMG). This initial order is further improved by Gauss-Seidel-like relaxation, window minimization and possibly by employing stochasticity, i.e., simulated annealing. The running time of the algorithms is linear, thus it can be applied to very large graphs. In addition, we have proposed two general principles that can be used for different functionals: (1) the continuation approach in which functionals that contain an evaluation of power p are successively approximated by a sequence of similar but with lower power functionals; (2) a first approximation can be obtained from the arrangement produced by one V-cycle of the minimum 2-sum problem instead of using the very popular spectral approach.

Since our algorithms were developed for practical purposes we compared them to many different heuristics : Spectral Sequencing, Optimally Oriented Decomposition Tree, Multilevel based, Simulated Annealing, Genetic Hillclimbing and other. For almost all instances we observed significant improvement either of the results or of the computational time compared various state-of-the-art methods. Our algorithms have proven themselves to be very stable (i.e., small standard deviations) and of high quality both as a first approximation (using "light" V-cycles) and as more aggressive energy minimizers (with more "heavy" cycles and possibly postprocessing).

We recommend our multilevel algorithms as a general practical method for solving graph (matrix) linear ordering problems and as a fast and high-quality method for obtaining first approximation for them. The implemented algorithms can be obtained at [26].

References

[1] R. Bar-Yehuda, G. Even, J. Feldman, and J. Naor. Computing an optimal orientation of a balanced decomposition tree for linear arrangement problems. *Journal of Graph Algorithms and Applications*, 5(4):1–27, 2001.

[2] S. Barnard, A. Pothen, and H. Simon. A spectral algorithm for envelope reduction of sparse matrices. *Numerical Linear Algebra with Applications*, 2(4):317–334, 1995.

[3] A. Brandt. Algebraic multigrid theory: The symmetric case. *Jornal of Appl. Math. Comp.*, 19:23–56, 1986. Preliminary proceedings of the International Multigrid Conference, April 6–8, 1983, Copper Mountain, CO.

[4] A. Brandt. Multiscale scientific computation: Review 2001. In T. Barth, R. Haimes, and T. Chan, editors, *Multiscale and Multiresolution methods (Proceeding of the Yosemite Educational Symposium, October 2000)*. Springer-Verlag, 2001.

[5] 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, POB 1852, 1982.

[6] A. Brandt, S. McCormick, and J. Ruge. Algebraic multigrid (AMG) for sparse matrix equations. In D. J. Evans, editor, *Sparsity and its Applications*, pages 257–284, 1984.

[7] A. Brandt and D. Ron. Chapter 1 : Multigrid solvers and multilevel optimization strategies. In J. Cong and J. R. Shinnerl, editors, *Multilevel Optimization and VLSI-CAD*. Kluwer, 2003.

[8] A. Brandt, D. Ron, and D. Amit. Multi-level approaches to discrete-state and stochastic problems. In W. Hackbush and U. Trottenberg, editors, *Multigrid Methods II*, pages 66–99. Springer-Verlag, 1986.

[9] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A multigrid tutorial: second edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[10] C.-K. Cheng. Linear placement algorithm and applications to VLSI design. *Netw.*, 17(4):439–464, 1987.

[11] G. M. D. Corso and F. Romani. Heuristic spectral techniques for the reduction of bandwidth and work-bound of sparse matrices. *Numerical Algorithms*, 28(1–4):117–136, Dec. 2001.

[12] T. Davis. University of florida sparse matrix collection. *NA Digest*, 97(23), 1997.

[13] J. Díaz, J. Petit, and M. Serna. A survey of graph layout problems. *ACM Comput. Surv.*, 34(3):313–356, 2002.

[14] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified np-complete problems. In *STOC '74: Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 47–63, New York, NY, USA, 1974. ACM Press.

[15] A. George and A. Pothen. An analysis of spectral envelope reduction via quadratic assignment problems. *SIAM Journal on Matrix Analysis and Applications*, 18(3):706–732, 1997.

[16] S. B. Horton. *The Optimal Linear Arrangement Problem: Algorithms And Approximation*. PhD thesis, Georgia Institute of Technology, May 1997.

[17] Y. F. Hu and J. A. Scott. A multilevel algorithm for wavefront reduction. *SIAM J. Sci. Comput.*, 23(4):1352–1375, 2001.

[18] M. Juvan and B. Mohar. Optimal linear labelings and eigenvalues of graphs. *Discrete Appl. Math.*, 36(2):153–168, 1992.

[19] S. Kirkpatrick. Models of disordered systems. In C. Castellani, editor, *Lecture Notes in Physics 149*. Springer-Verlag, 1981.

[20] Y. Koren and D. Harel. Multi-scale algorithm for the linear arrangement problem. *Proceedings of 28th Inter. Workshop on Graph-Theoretic Concepts*, 2002.

[21] Y. Lai and K. Williams. A survey of solved problems and applications on bandwidth, edgesum, and profile of graphs. *J. Graph Theory*, 31:75–94, 1999.

[22] J. Petit. Experiments on the minimum linear arrangement problem. *ACM Journal of Experimental Algorithmics, 8*, 2003.

[23] E. Rodriguez-Tello, J.-K. Hao, and J. Torres-Jimenez. An effective two-stage simulated annealing algorithm for the minimum linear arrangement problem. *accepted in the Computers and Operations Research*, 2007.

[24] 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, 2005.

[25] J. Ruge and K. Stüben. *Algebraic Multigrid*, pages 73–130. SIAM, 1987.

[26] I. Safro. Homepage of our projects. *http://www.wisdom.weizmann.ac.il/˜safro*.

[27] I. Safro, D. Ron, and A. Brandt. Graph minimum linear arrangement by multilevel weighted edge contractions. *Journal of Algorithms*, 60(1):24–41, 2006.

[28] I. Safro, D. Ron, and A. Brandt. Multilevel algorithm for the minimum 2-sum problem. *Journal of Graph Algorithms and Applications*, 10(2), 2006.

[29] F. Shahrokhi, O. Sýkora, L. A. Székely, and I. Vrto. On bipartite drawings and the linear arrangement problem. *SIAM Journal on Computing*, 30(6):1773–1789, 2001.

[30] E. Sharon, A. Brandt, and R. Basri. Fast multiscale image segmentation. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pages 70–77, 2000.

[31] K. Stüben. *An introduction to algebraic multigrid*, pages 413–532. Academic Press, 2001.

[32] K. Stüben. A review of algebraic multigrid. *J. Comput. Appl. Math.*, 128(1-2):281–309, 2001.