Multiscale Methods

In many complex systems, a big scale gap can be observed between micro- and macroscopic scales of problems because of the difference in physical (social, biological, mathematical, etc.) models and/or laws at different scales.
Introduction to Network Analysis
In many problems, notwithstanding the fact that elementary parts of the system have a complicated (and even nondeterministic) behavior, their ensembles represent much more structured systems.

Two body scratch model
Even when differences between models at different scales are not observed, an efficient approximation of the microscopic scale can be achieved by looking at the macroscopic scale with its substantially smaller number of elementary objects (such as variables and particles).
The Multiscale Method
Multiscale ≈ Multilevel ≈ Multigrid ≈ Multiresolution

- The **Multiscale method** is a class of algorithmic techniques for solving efficiently large-scale computational and optimization problems.

- A multivariable problem defined in some space can have an approximate description at any given length scale of that space: a continuum problem can be discretized at any given resolution, multiparticle system can be represented at any given characteristic length, etc.

- The multiscale algorithm recursively constructs a sequence of such descriptions at increasingly larger (coarser) scales, and combines *local* processing at each scale with various *inter-scale* interactions.
Computational Multiscale Methods

Joseph Fourier
Functional analysis at multiple resolutions (1768-1830)

Radiy Fedorenko
Smoothing, finite elements, two-level multigrid (1930-2009)

Achi Brandt
Popularization, first basic research (1977), algebraic multigrid (1980), ...
A suitable relaxation can always reduce the information content of the error (by smoothing it), and quickly make it approximable by far fewer variables (which are related to the smooth error modes).

**Observation**

Original error

$k = 5$

$k = 10$

$k = 500$

**Reminder:** Iterative relaxation for solving $Ax = b$

$$x^{(k+1)} = Tx^{(k)} + v$$
Cycles

A. Coarsening
Defines the hierarchical structure (P=P₀, P₁, ..., Pₖ)

B. Exact solution

C. Interpolation and Relaxation
Produces an initial solution of Pᵢ₋₁ from the solution Sᵢ of Pᵢ and constructs final solution Sᵢ₋₁ from Pᵢ₋₁
Computational work

Total complexity, $\sum_i c_i m_i \approx H_{m_0}$, is **linear**, where $c_i$ is computational work per variable and $m_i$ is a number of variables at level $i$.

A. Coarsening
Defines the hierarchical structure ($P=P_0, P_1, ..., P_k$)

B. Exact solution

C. Interpolation and Relaxation
Produces an initial solution of $P_{i-1}$ from the solution $S_i$ of $P_i$ and constructs final solution $S_{i-1}$ from $P_{i-1}$
Coarsening

\[ G \]

\[ G \]

Coarsest graph

\[ G \]

Solution for coarsest graph

Interpolation
Relaxation
Local improvements

solve exactly
Algebraic Multigrid

Brandt, McCormick, Rudge, ”Algebraic Multigrid (AMG) for automatic multi-grid solution with application to geodetic computations”, 1982

• Given: $A \in \mathbb{R}^{n \times n}$ positive definite, symmetric.

• Goal: solve $Ax = b$.

• Claim: If $A$ is positive definite, then

$$x \text{ minimizes } P(x) = \frac{1}{2} x^T A x - x^T b \text{ iff } Ax = b.$$ 

• $\tilde{x}$ - current approximation

• $e(rror) = x - \tilde{x}$ (hard to estimate)

• $b - A\tilde{x} = r(esidual) = A(x - \tilde{x}) = Ae$
Algebraic Multigrid

At all levels: solve $Ae = r$, where $e(rror) = x - \tilde{x}$ and $r(esidual) = b - A\tilde{x}$

$$\min \frac{1}{2} e^T A e - e^T r =$$

$$\min \frac{1}{2} (\tilde{e} + \uparrow^f_c e^c)^T A (\tilde{e} + \uparrow^f_c e^c) - (\tilde{e} + \uparrow^f_c e^c)^T r \leftrightarrow \ldots \leftrightarrow$$

$$\min \frac{1}{2} (e^c)^T \left[(\uparrow^f_c)^T A \uparrow^f_c\right] e^c - (e^c)^T (\uparrow^f_c)^T (r - A\tilde{e}) =$$

$$\min \frac{1}{2} (e^c)^T A^c e^c - (e^c)^T r^c$$

- $\tilde{e}$ - initial fine level error
- $e^c$ - coarse level error
- $\uparrow^f_c$ - coarse-to-fine interpolation operator
Multilevel Algorithms for Combinatorial Optimization Problems

- **Examples:** VLSI Placement, Partitioning, Minimum Linear Ordering, Clustering, Segmentation, Visualization.

- **Quality:** Usually exhibit superior results to other methods on practical test suites.

- **Time:** Usually exhibit linear time complexity.

- No theoretical results for discrete problems.
Coarsening

\[ G \]

Interpolation
Relaxation
Local improvements

-solve exactly-

Coarsest graph
Solution for coarsest graph
Minimum Linear Arrangement, $p=1$

Given:

- Graph $G = (V, E)$
- Weighting function on edges $w : E \rightarrow \mathbb{R}$
- Permutation of vertices $\pi : V \rightarrow \{1, 2, ..., |V|\}$

Graph weight: $\sigma_p(G, \pi) = \left( \sum_{ij \in E} w_{ij}|\pi(i) - \pi(j)|^p \right)^{1/p}$

Goal: minimize over all $\pi$ $\sigma_p(G, \pi)$

If $p = \infty$ minimize over all $\pi$ $\sigma_\infty(G) = \max_{ij \in E} w_{ij}|\pi(i) - \pi(j)|$
*k*-partitioning

Process which consists of dividing graph vertices into a given number of sets *k*, while enforcing two typical constraints:

1. **Boundary objective**: the size of the interface between parts should be as small as possible (usually edgcut)
2. **Balance constraint**: all sets should be evenly weighted

Additional constraints can be considered, such as:

- Connectivity of parts, especially in VLSI design
- Compactness of parts (aspect ratio), in domain decomposition
  
  . . .
Simple Case: Coarsening by Contractions

Intuitive explanation: two or more vertices are merged if they have a good chance to share common properties.

Common properties

- $k$-partitioning: $i$ and $j$ belong to the same part
- Linear arrangement: $|\pi(i) - \pi(j)|$ is small
- Coloring: $\chi(i) = \chi(j)$
Common disadvantage of all strict coarsening methods

They make local decisions (i.e., merging) before accumulating the relevant global information. It creates additional difficulty for solving irregular instances when local decision contradicts global picture.

Existing multilevel solvers

- **CHACO** by Hendrickson and Leland, since 1993
- **METIS** by Karypis and Kumar, since 1995
- **SCOTCH** by Pellegrini, since 1996
- **JOSTLE** by Walshaw, since 1995
Weighted Aggregation (inspired by Algebraic Multigrid)

Weighted Aggregation

- The main new algorithmic property (inspired by AMG) is to make a "soft" coarsening in which each node may be divided into fractions and different fractions form a coarse node.

Examples

- Sharon, Brandt, Basri, “Fast multiscale image segmentation”, 2000
- S, Ron, Brandt, “Graph minimum linear arrangement by multilevel weighted edge contractions”, 2002
Minimum Linear Arrangement

Original problem

Generalized problem

\[ \mathcal{E}_\pi(X) = \left( \sum_{ij \in E} w_{ij} \cdot |x_i - x_j|^p \right)^{1/p} \]
Coarse nodes construction: C-points selection

- Choose a dominating set $C \subset V$ s.t. all others from $F = V \setminus C$ are “strongly coupled” to $C$.

- Each chosen node will serve as the seed of a coarse aggregate.
Interpolation weights

\[(f_c)_{ij} = \begin{cases} \frac{w_{ij}}{\sum_{k \in N(i)} w_{ik}} & i \in F, j \in N(i) \\ 1 & i \in C, j = i \\ 0 & \text{otherwise} \end{cases} \]

- Define the interpolation weights of all vertices
- In some sense, the interpolation weights (iw) are the probabilities of a vertex to share a common property with the aggregates it belongs to.
Coarse Graph

Aggregate $J$ includes its seed $j$ and the (normalized) parts $iw_{ij}$ of all fine $x_i$. Thus its volume $\text{vol}(J) = \text{vol}(j) + \sum_i iw_{ij} \text{vol}(j)$.

The volumes of the coarse nodes satisfy

$$
\sum_{v \in G_c} \text{vol}(v) = \sum_{u \in G_f} \text{vol}(u)
$$
Coarse Graph

- Let $iw(\|k\|$ be the interpolation weight of fine vertex $k$ to aggregate $l$, then

$$w_{lJ} = \sum_{l,k} iw(\|l\|) \cdot w_{lk} \cdot iw(kJ)$$

- Time: $O(|E_f|Q^2)$, where $Q$ is the interpolation order.
Coarse Graph

$\uparrow_c^f$ – Fine(f)-to-coarse(c) interpolation operator

$L_f$ – Weighted Laplacian of $G$ at level $f$

\[(\uparrow_c^f)^T L_f \uparrow_c^f = L_c + \{ \text{self loops at the diagonal} \} \]

\[ w_{IJ} = \sum_{l,k} iw(ll) \cdot w_{Ik} \cdot iw(kJ) \]
**Problem independent:**
Minimum k-Partitioning
Minimum 2-sum
Minimum Bandwidth
Minimum Workbound
Minimum Wavefront
Minimum Linear Arrangement

In contrast to classical multigrid approaches where
\[ L_f = P^T L_c P \]
we will use an information about all graph connections
Uncoarsening: Interpolation for Minimum Linear Arrangement

1) Place the seeds according to their aggregates

2) Place other vertices by minimizing their local contribution to the total energy:
   - $p = 1$: at their medians
   - $p = 2$: at their weighted averages
   - $p > 2$: solve minimization numerically
Relaxation

Two types of pointwise relaxation that improve current solution:

- **Compatible Relaxation:** keep coarse vertices (seeds) invariant minimizing the energy of other vertices one-by-one wrt to the problem,
- **Gauss-Seidel Relaxation:** Improve all vertices.

Initial legal coordinates $x_i, \forall i \in V$

for all $i \in V$ $y_i \leftarrow x_i$

for all $i \in F$ (Compatible) / $i \in V$ (Gauss-Seidel) do

$$y_i = \arg\min_{y_i} \begin{cases} \\
|\sum_{j<i} w_{ij} - \sum_{j>i} w_{ij}|, & \text{if } p = 1 \\
\sum_{j \in V} y_j w_{ij} / \sum_{j \in V} w_{ij}, & \text{if } p = 2 \\
\sum_{j \in V} w_{ij} (y_i - y_j)^p, & \text{if } p > 2 \\
\end{cases}$$

end

for all $i \in V$ $x_i = \frac{v_i}{2} + \sum_{y_k < y_i} v_k$
Lemma: Improving the ordering cost of $W$ (a subset of consecutive vertices) cannot increase the cost of total ordering.

Window minimization

$$\text{minimize } \sigma_2(W, \tilde{x}, \delta) = \sum_{i,j \in W} w_{ij}(\tilde{x}_i + \delta_i - \tilde{x}_j - \delta_j)^2 + \sum_{i \in W, j \notin W} w_{ij}(\tilde{x}_i + \delta_i - \tilde{x}_j)^2$$

- $\tilde{x}$ - current approximation
- $\delta$ - correction
Uncoarsening: Local Refinement, \( p=2 \)

\[
\text{minimize } \sigma_2(W, \tilde{x}, \delta) = \sum_{i,j \in W} w_{ij}(\tilde{x}_i + \delta_i - \tilde{x}_j - \delta_j)^2 + \sum_{i \in W, j \notin W} w_{ij}(\tilde{x}_i + \delta_i - \tilde{x}_j)^2
\]

To prevent the possible convergence of many coordinates to one point add

\[
\sum_{i \in \mathcal{W}} (\tilde{x}_i + \delta_i)^m v_i = \sum_{i \in \mathcal{W}} \tilde{x}_i^m v_i , \quad m = 1, 2
\]

Final system of equations

\[
\begin{cases}
\sum_{j \in \mathcal{W}} w_{ij}(\delta_i - \delta_j) + \delta_i \sum_{j \notin \mathcal{W}} w_{ij} + \lambda_1 v_i + \lambda_2 v_i \tilde{x}_i = \sum_{j} w_{ij}(\delta_i - \delta_j) \\
\sum_i \delta_i v_i = 0 \\
\sum_i \delta_i v_i \tilde{x}_i = 0
\end{cases}
\]
Linear Arrangement: Spectral Approach

minimize over real $x$
subject to

$$E(x) = \sum_{i,j} w_{ij} (x_i - x_j)^2$$
$$\sum_i x_i^2 = 1, \quad \sum_i x_i = 0.$$  \[
\begin{align*}
E(x) &= x^T A x \\
x^T B x &= 1, \quad \sum_i x_i = 0
\end{align*}
\]
where
$$a_{ij} = -w_{ij}, \quad a_{ii} = \sum_j w_{ij}, \quad b_{ij} = \delta_{ij}$$

$x$ is the second eigenvector of
$$Ax = \lambda B x.$$  

Heuristics: order the vertices according to the eigenvector of the second smallest eigenvalue.
Experimental Results: Linear Arrangement, $p=2$

[SRB] ”Multilevel algorithm for the minimum 2-sum problem”, 2006

Ratios between multilevel and spectral algorithms

Large-scale graphs
Linear Arrangement, Larger powers

**POSTPROCESSING:**
Additional iterations of Window Minimization with sequentially growing $p$

- Solve for $p = 2k$
- Solve for $p = 2k - 2$
- Solve for $p = 4$
- Solve for $p = 2$
• Define $\hat{w}_{ij} = w_{ij}(\tilde{x}_i - \tilde{x}_j)^{p-2}$

• Substitute $w_{ij}$ with $\hat{w}_{ij}$ in

\[
\text{minimize } \sigma_p(W, \tilde{x}, \delta) = \\
= \sum_{i,j \in W} w_{ij}(\tilde{x}_i + \delta_i - \tilde{x}_j - \delta_j)^p + \sum_{i \in W, j \not\in W} w_{ij}(\tilde{x}_i + \delta_i - \tilde{x}_j)^p = \\
= \sum_{i,j \in W} \hat{w}_{ij}(\tilde{x}_i + \delta_i - \tilde{x}_j - \delta_j)^2 + \sum_{i \in W, j \not\in W} \hat{w}_{ij}(\tilde{x}_i + \delta_i - \tilde{x}_j)^2 \approx \\
\approx \hat{\sigma}_2(W, \tilde{x}, \delta)
\]
Experimental Results: Linear Arrangement, $p=\infty$

Scalability. Dependence of running time on the graph size.

blue line - whole set of graphs.
bold line - 25 largest graphs.
dashed line - line with slope 1
The Minimum Workbound Problem

Goal: minimize over all $\pi$

$$wb(G, \pi) = \sum_i \max_j w_{ij} (\pi(i) - \pi(j))^2$$

$$\pi(j) < \pi(i)$$

Generalization:

$$wb(G, x) = \sum_i \max_{j:x_j < x_i} w_{ij} (x_i - x_j)^2 \approx \sum_i \left( \sum_{j:x_j < x_i} w_{ij} (x_i - x_j)^p \right)^{2/p}$$

Window Minimization for the minimum workbound problem (Taylor exp.):

$$wb_p(W, \tilde{x}, \delta) \approx wb_p(W, \tilde{x}, 0) + \sum_{i \in W} \frac{\partial wb_p}{\partial \delta_i} (W, \tilde{x}, 0) \delta_i + \sum_{i,j \in W} \frac{\partial^2 wb_p}{\partial \delta_i \partial \delta_j} (W, \tilde{x}, 0) \delta_i \delta_j$$
Experimental Results: Minimum Workbound

[SRB] ”Multilevel algorithms for linear ordering problems”, 2008

Ratios between multilevel and spectral algorithms

Large-scale graphs
Refinement for $k$-partitioning

The diagram illustrates the process of refining a solution for $k$-partitioning. The solution at the coarse level (LEVEL 0) is inherited and used as a basis for the refined solution at the next level (LEVEL 1). The connections and weights between nodes are marked with arrows and approximate values, such as $\sim 0.6$ and $\sim 0.8$, to indicate the strength of the connections.
More Examples: Dimensionality Reduction

Given a set of high dimensional data represented by vectors \( x_1, \ldots, x_n \) in \( \mathbb{R}^m \), the task is to represent these with low dimensional vectors \( y_1, \ldots, y_n \in \mathbb{R}^d \) with \( d \ll m \), such that nearby points remain nearby, and distant points remain distant.

[FSS] “Multilevel Nonlinear Dimensionality Reduction for Manifold Learning”
More Examples: Segmentation

Segmentation: The pixel graph

Low contrast - strong coupling, High contrast - weak coupling;
Segmentation \equiv \text{Low-energy cut}

\[
\text{minimize } \Gamma(u) = \frac{\sum_{i>j} w_{ij}(u_i - u_j)^2}{\sum_{i>j} w_{ij}u_i u_j}
\]

Any boolean \( u \) that yields a low-energy \( \Gamma(u) \) corresponds to a salient segment
Segmentation: Multiscale Approach

Figure 2 | The multiscale normalized cut graph approach.  

*Figure 2* | The multiscale normalized cut graph approach.  

- **a**, A simple image.  
- **b**, Pixels of the image are nodes, represented by filled circles; strong coupling is represented by thick red lines, and weak coupling by thin blue lines.  
- **c**, Adaptive coarsening. Each pixel in **b** is strongly coupled to one of the chosen seeds shown here (thus, pixels strongly coupled to a given seed form an aggregate). Couplings between the seeds are shown.  
- **d**, An additional coarsening level. In this case, this is the level at which the salient segment is detected.
Two-dimensional layout problem

Find an optimal layout of 2D objects such that

1. the total length of the given connections between these objects will be minimal
2. the two-dimensional space will be well utilized and
3. the overlapping between objects will be as little as possible
Two-dimensional layout problem

minimize \[ \text{Total edge length (quadratic functional)} \]
subject to \[ \forall \text{ small squares } s \text{ the amount of the material inside } s \text{ is less than its area (linear inequality constraints).} \]
Material movement problem

\[
\min_{u,v} \frac{1}{2} \sum_{ij \in E} w_{ij} \left[ \left( \tilde{x}_i + \sum_{p \in c(i)} \alpha_{pi} u_p - \tilde{x}_j - \sum_{p \in c(j)} \alpha_{pj} v_p \right)^2 + \left( \tilde{y}_i + \sum_{p \in c(i)} \alpha_{pi} v_p - \tilde{y}_j - \sum_{p \in c(j)} \alpha_{pj} u_p \right)^2 \right]
\]
∀s, eqd(s) =

\[
\frac{\Upsilon(s) + \Upsilon_r(s)}{2A} \cdot h_y \cdot \frac{u_{rt}(s) + u_{rb}(s)}{2} - \frac{\Upsilon(s) + \Upsilon_l(s)}{2A} \cdot h_y \cdot \frac{u_{lt}(s) + u_{lb}(s)}{2} + \\
\frac{\Upsilon(s) + \Upsilon_t(s)}{2A} \cdot h_x \cdot \frac{v_{rt}(s) + v_{lt}(s)}{2} - \frac{\Upsilon(s) + \Upsilon_b(s)}{2A} \cdot h_x \cdot \frac{v_{rb}(s) + v_{lb}(s)}{2}
\]

\[
\leq M(s) - \Upsilon(s)
\]

\(\Upsilon(s)\) \quad \text{total area of nodes overlapping with } s

\(h_x\) \text{ and } \(h_y\) \quad \text{width and height of } s

\(A\) \quad \text{area of } s

\(\Upsilon_r(s)\) \quad \text{area of nodes overlapping with right neighbor square}

\(\Upsilon_l(s)\) \quad \ldots

\(\Upsilon_t(s)\) \quad \ldots

\(\Upsilon_b(s)\) \quad \ldots
Two-dimensional layout problem: coarsening

Variables

Constraints
Two-dimensional layout problem: example
Mesh 64x64 + Random Edges
Two-dimensional layout problem: VLSI Chip

Original

Multiscale Solver
By Brandt, Ron