# Distributed Decomposition Over Hyperspherical Domains

Aron Ahmadia[1], David Keyes[1], David Melville[2], Alan Rosenbluth[2], Kehan Tian[2]

[1] Department of Applied Physics and Applied Mathematics, Columbia University, New York, NY 10027, USA
[2] IBM T.J. Watson Research Center, Yorktown Heights, New York 10598, USA

**Summary.** We are motivated by an optimization problem arising in computational scaling for optical lithography that reduces to finding the point of minimum radius that lies outside of the union of a set of diamonds centered at the origin of Euclidean space of arbitrary dimension. A decomposition of the feasible region into convex regions suggests a heuristic sampling approach to finding the global minimum. We describe a technique for decomposing the surface of a hypersphere of arbitrary dimension, both exactly and approximately, into a specific number of regions of equal area and small diameter. The decomposition generalizes to any problem posed on a spherical domain where regularity of the decomposition is an important concern. We specifically consider a storage-optimized decomposition and analyze its performance. We also show how the decomposition can parallelize the sampling process by assigning each processor a subset of points on the hypersphere to sample. Finally, we describe a freely available C++ software package that implements the storage-optimized decomposition.

## 1 Global Optimization for Semiconductor Lithography Mask Design

In the newly heralded field of computational scaling [7], industrial scientists are now investigating a global minimization formulation of the problem of mask design for optimizing process windows in semiconductor lithography [5, 6]. We are considering the problem of forming an optimal mask design for the optical printing of a given 2-D target image, which is considered as a set of sampled target points that must be sufficiently illuminated for the image to correctly print. We attempt to minimize the total intensity of a set of $d$ exposure modes, with each axis $x_i$ corresponding to an intensity for mode $i$. In the space of the $d$ exposure modes, each of the $n$ samples of image features are represented as a $d$-dimensional diamond of infeasible space. Each sampled image feature is sufficiently illuminated by the set of exposure modes if their representative coordinate $\mathbf{x} \in \mathbf{R}^d$ lies outside the diamond. Sufficient illumination of all sampled target features in an exposure is achieved in all points outside of the union of the set of diamonds. Although a global minimum is ideal, the goal of the problem is to find good solutions in a reasonable amount of time. Additionally, global

minima that satisfy all the constraints may still be rejected due to manufacturability considerations, so a good solution method will provide the global minima and may provide a set of the best available local minima within each orthant of the search space.

### 1.1 Convex Partitions of the Feasible Domain

The semiconductor lithography problem can be considered as the search for the global minima of a linear optimization problem subject to nonconvex constraints:

$$\text{minimize } \|\mathbf{x}\|_1 \text{ subject to } A_i(\mathbf{x}) \cdot \mathbf{x} \geq \mathbf{b}_i \; i = 1 \ldots n \tag{1}$$

We define the $j^{\text{th}}$ normalized principal axis of diamond $A_i$ as $l_{i,j}\mathbf{c}_{i,j}$, with $\mathbf{c}_{i,j}$ representing the direction and $l_{i,j}$ the magnitude of the principal axis. We then note that each of the $2^d$ planes defining a half-space exterior of a diamond connects the $d$ principal axes of the diamond. $A_i(\mathbf{x})$ can be considered as a function of the choices of signs for the vectors representing the principal axes of the diamond, the choice of connection to the 'positive' or 'negative' end of each principal axis uniquely determines one of the planes.

Since the diamond is an intersection of half-spaces, a point $\mathbf{x}$ is considered feasible with respect to the diamond if it lies inside any of the reflected (pointing outwards from the origin) half-spaces $A^+_{i,k}$ arising from the hyperplanes $A_{i,k}$. Each combination of the positive and negative principal $\mathbf{c}_j$ axes in a given diamond $A_i$ is a set of $d$ points that uniquely determine a constraint half-space $A^+_{i,k}$. For any given $k$ and diamond $A_i$:

$$s_{i,k}(j) * \mathbf{c}_{i,j} \in A_{i,k} \quad j = \{1 \ldots d\} \tag{2}$$

Since all of the constraint planes are linear and do not contain the origin:

$$\text{if } \; \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d \in A_k, \text{ and } \theta \in \mathbf{R}^d, \; \theta_j \geq 0, \; \sum_j^d \theta_j \geq 1$$

$$(\theta_1 \mathbf{x}_1 + \theta_2 \mathbf{x}_2 + \cdots + \theta_d \mathbf{x_d}) \in A_k^+$$

In particular:

$$\theta \in \mathbf{R}^d, \theta_j \geq 0, \; \sum_j^d \theta_j \geq 1 \tag{3}$$

$$(\theta_1 s_k(1)\mathbf{c}_1 + \theta_2 s_k(2)\mathbf{c}_2 + \cdots + \theta_d s_k(d)\mathbf{c}_d) \in A_k^+ \tag{4}$$

Because the set of principal axes $l_j \mathbf{c}_j$ for each diamond forms an orthogonal basis on $\mathbf{R}^d$, we can express:

$$\mathbf{x} = \sum_j^d l_j \theta_j \mathbf{c_j}, \quad \theta_j = \mathbf{c}_j \cdot \mathbf{x} \tag{5}$$

Any combination of positive and negative $\theta_{i,j}$ can be converted to their absolute values with an appropriate choice of $k$, and we can say that $\mathbf{x}$ lies outside diamond $i$ (by satisfying constraint half-space $i,k$) if:

$$\sum_{j=1}^{d} \frac{|\theta_{i,j}|}{l_{i,j}} \geq 1 \tag{6}$$

We may enumerate a single plane $A_{i,k}$ with a tuple, or ordered list of $d$ signs $\mathbf{s_{i,k}}$, with the sign of the $j$th element $s_{i,k}(j)$ corresponding to the ends of the principal axes $\mathbf{c_{i,j}}$ it connects. We represent the concatenation of all $d$-tuples $s_{i,k}$ corresponding to a choice of plane for each of the $n$ diamonds into a single $n_l$-tuple: $\mathbf{s}$, with $n_l = n * 2^d$. Similarly, we concatenate the list of all principal axes into a list $\{\mathbf{c_i}\}$, with $i = 1 \cdots n_l$.

We define a set of nonoverlapping regions $R_\mathbf{s}$ that collectively exhaust $R^d$, with a total of $2^{n_l}$ potential tuples $\mathbf{s}$ and corresponding regions $R_\mathbf{s}$.

$$R_\mathbf{s} : \{\mathbf{x} \in \mathbf{R}^d \mid \mathbf{sign}(\mathbf{c_i} \cdot \mathbf{x}) = s_i,\ i = 1, \ldots, n_l\} \tag{7}$$

Each $\mathbf{s}$ defines a convex region containing $R_\mathbf{s}$:

$$\sum_{j=1}^{d} \frac{s_{i,j}(\mathbf{c_{i,j}} \cdot \mathbf{x})}{l_{i,j}} \geq 1\ \forall i \tag{8}$$

If we enumerate the set of convex regions $R_\mathbf{s}$, the global minima is the best local minima from each of the regions.

## 2 Partitions of n-Space

We expect the true number of convex regions $p$ in the space to be less than $2^{n_l}$, as some intersections of diamond half-spaces will be empty. We seek an upper bound on $p$ as a function of $d$, the dimensionality of the problem, and $n$, the number of diamonds. We consider each of the diamonds as a set of $d$ cutting planes, with $\mathbf{c_{i,j}}$ represented as a cutting plane that intersects the origin. Upper bounds for the number of regions generated by origin-centered cutting planes in general position were established independently by [1], Perkins, Willis, and Whitmore (unpublished), and [8]. "Partitions of N-Space by Hyperplanes", [9] allows tighter bounds based on the degeneracy of the planes. We consider the diamonds to be non-degenerate, so the upper bounds for cutting planes in general position are sufficient:

$$p \leq B_d^n = 2 \sum_{i=0}^{d-1} \binom{n-1}{i} \tag{9}$$

## 3 Incomplete Search Heuristics

When applied to the semiconductor lithography problem, the coordinate axes represent linear combinations of the underlying physical variables that are chosen for their

efficient average coupling to the features represented by the constraint diamonds. This selection process causes the modes to be strongly coupled to individual constraint features, which in turn causes the axes of the diamonds to be preferentially aligned with the coordinate axes. As a result the principal axes of the diamonds have a tendency to cluster together directionally, making the size distribution of the regions between intersecting ellipsoids very non-uniform, with the largest and deepest regions comprising a small fraction of the total. Empirically it is found that solutions which are adequately close to the global optimum can be found by searching only the largest regions. In [5], the authors suggest that directly sampling points on a hypersphere is a useful heuristic.

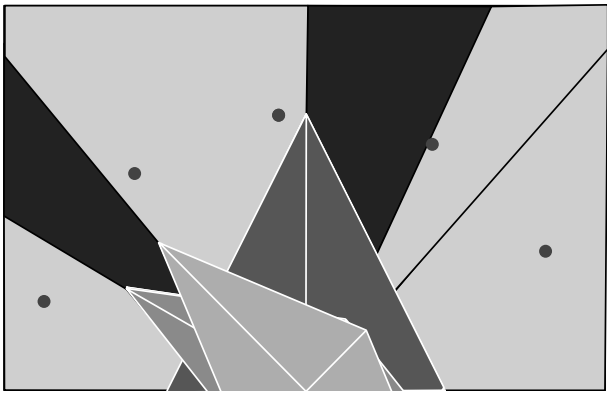

**Fig. 1.** Sampling reduces the number of regions to search by skipping small-angle regions, unsearched regions are darkly shaded.

The number of sampled regions $n_s$ using a decomposition of the $(d-1)$-sphere embedded in $R^d$ can be considered as a function of the search density per dimension $\rho$:

$$n_s(\rho) = \rho^d. \tag{10}$$

The number of sampled points is much more managable than the actual number of potential convex regions. We accept the currently unquantified risk of missing the global minimum in exchange for a more computationally tractable approximation to the problem.

## 4 Hypersphere Decomposition

### 4.1 Previous Work

We are now faced with the task of partitioning a $(d-1)$-sphere into regions of approximately equal area and small diameter. Fortunately, this topic has been well-studied in [3], based on a construction for 2-spheres introduced earlier in Zhou's

1995 PhD Thesis as well as unpublished work by E.B. Saff and I.H. Sloan. A domain decomposition method for particle methods on the 2-sphere was also introduced in [2]. A full introduction to the general algorithm for $d$-spheres into $n_s$ partitions is impossible here, but we generalize the algorithm as a partitioning of $m$-spheres into regions which are then recursively partitioned as $(m-1)$-spheres.

### 4.2 A Memory-Efficient Tree Storage Scheme for Equal Area Hypersphere Regions

We are motivated by the large value of $n_s$ to seek a compressed storage partition of $n_s$ points. At each level $j$ of the decomposition, starting at $j = d$, and ending at $j = 1$, we have some number of $j$-spheres embedded into $(j+1)$-space. We denote the number of $j$-spheres in the decomposition at level $j$ by $k_j$. We propose using the procedure's recursive tree as a storage scheme for the points, avoiding the costs of storing full coordinate information for each point. This idea was originally proposed in [4] in section in 2.5 for spherical coding, though its potential for compressed storage was not fully explored. We are interested in forming a loose bound on the total number of nodes required for the storage of the tree. Since the number of nodes on the tree at level $j$ is based on the number of spheres at level $k_j$, we seek an upper bound on $k_j$.

At each level $j$, we decompose each $j$-sphere into some finite number of $(j-1)$-spheres and 2 polar caps. The $(j-1)$-spheres correspond to "collars" of the $j$-spheres, and are assigned a number of regions proportional to their fractional area of the sphere. We impose an indexing on all the spheres for a given level $j$, such that if there are $k_j$ spheres at level $j$, then the spheres are indexed from $i = 1, \ldots, k_j$, and let $k_{i,j}$ equal the number of $(j-1)$-spheres to decompose the $i,j$-th sphere into. Finally, we also affix $z_{i,j}$ to every sphere in the system, denoting the number of regions contained by sphere $i,j$.

If we fix $d = d-1$ and decompose the $d$-sphere, we have $k_d = 1$. We can now recursively build an estimate for $k_{d-l-1}$ from $k_{d-l}$ for $l = 0, \ldots, d-1$. We claim that $k_{d-l} = O(n^{l/d})$. This is true for the base case $l = 0$, and is true for all $l$ if we can show that $k_{d-l-1} = O(n^{\frac{l+1}{d}})$. We use the fact that for a $d$-sphere being decomposed into $s$ regions, the number $k$ of $(d-1)$-spheres/collars it contains is:

$$k = \frac{\pi - 2\theta_{d,n}}{\frac{\sigma(S^d)}{s}^{\frac{1}{d}}} \tag{11}$$

Where $\theta_{d,n}$ is the polar cap angle for a $d$-sphere decomposed into $n_s$ regions and $\sigma(S^d)$ is the measure of surface area of a $d$-sphere. At each level $j$, we can account for all regions contained by summing over the assigned regions for each $j$-sphere and the polar caps for all $m$-spheres, where $m > j$:

$$\sum_{i=1}^{k_j} z_{i,j} + 2 \sum_{m=j+1}^{d+1} k_m = n \tag{12}$$

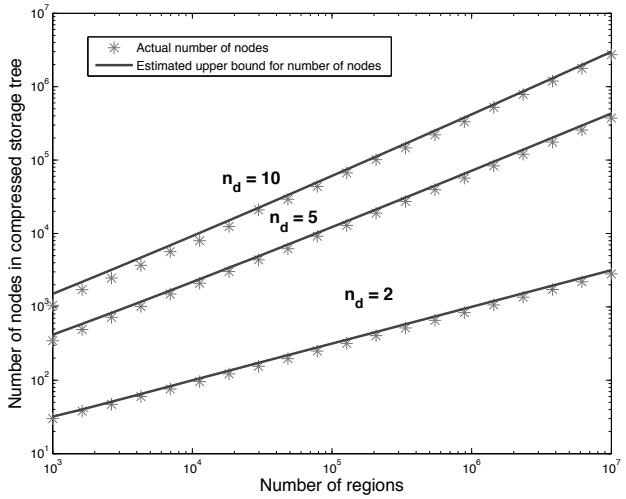


**Fig. 2.** Actual vs. Estimated Number of Nodes for $d = 2, 5, 10$ and between 1000 and 10,000,000 sample points

Trivially: $\sum_{i=1}^{k_j} z_{i,j} \leq n$. We now make the observation that the solution to max $\sum_{i=1}^{k} (p_i)^{1/j}$, subject to $\sum(p_i) \leq n$, is equal to $k(\frac{n}{k})^{1/j}$, and substitute:

$$k_{d-l-1} = O(n^{j/d})^{\frac{d-l-1}{d-l}} n^{\frac{1}{d-l}} \tag{13}$$

$$= O(n^{\frac{ld-l^2-l+d}{d(d-l)}}) \tag{14}$$

$$= O(n^{\frac{l+1}{d}}) \tag{15}$$

We substitute in equation (12) to obtain an estimate for the upper bound of the total nodes of the storage tree. We assume a constant $C = 1$, and:

$$K = \sum_{j=2}^{j=d} k_j = 2 \sum_{j=2}^{j=d-1} s^{j/d} + s^{\frac{d-1}{d}} \tag{16}$$

The estimates were computed for $d = 2, 5, 10$ and $n_s = 10^3$ to $n_s = 10^7$, then compared against actual tree structures in Fig. 3.

### 4.3 Parallel Decomposition

Finally, we introduce an algorithm for two-level decomposition suitable for massively parallel distributed sampling of the $n$-sphere. This algorithm successfully distributed a parallel search over 2,048 Blue Gene nodes.

1. Apply Leopardi's algorithm to generate a decomposition along some subspace of the original space.

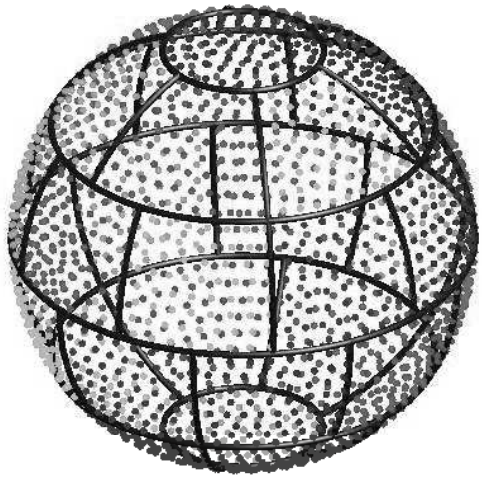

**Fig. 3.** Two-level distributed decomposition of a 2-sphere

2. For each decomposed region, apply Leopardi's algorithm again to sample points, along decomposed dimensions, the algorithm operates on region boundaries established in the first decomposition.

## 5 Ongoing Work

We wish to consider compressions of the tree structure by enforcing symmetry in the hypersphere decomposition. We are also interested in improving the performance of the tree structure code. A C++ implementation of the tree structure sampling code is available from http://aron.ahmadia.net/code.

*Acknowledgement.* The authors gratefully acknowledge Braxton Osting and Matias Courdurier for their insightful contributions to several sections of this paper.

## References

[1] Cameron, S.H.: An estimate of the complexity requisite in a universal decision network. *Bionics Symposium, WADD Report*, pages 60–600, 1960.
[2] Egecioglu, Ö., Srinivasan, A.: Domain Decomposition for Particle Methods on the Sphere. In *Proceedings of the Third International Workshop on Parallel Algorithms for Irregularly Structured Problems*, pages 119–130. Lecture Notes in Computer Science, 117. Springer, London, 1996.
[3] Leopardi, P.: A partition of the unit sphere into regions of equal area and small diameter. *Electron. Trans. Numer. Anal.*, 25:309–327, 2006.

[4] Leopardi, P.: *Distributing points on the sphere: Partitions, separation, quadrature and energy*. PhD thesis, University of New South Wales, 2007.
[5] Rosenbluth, A.E., Bukofsky, S., Fonseca, C., Hibbs, M., Lai, K., Molless, A.F., Singh, R.N., Wong, A.K.K.: Optimum mask and source patterns to print a given shape. *J. Microlithogr. Microfabrication, Microsyst.*, 1:13-30, 2002.
[6] Rosenbluth, A.E., Melville, D., Tian, K., Lai, K., Seong, N., Pfeiffer, D., Colburn, M.: Global optimization of masks, including film stack design to restore TM contrast in high NA TCC's. *Proc. SPIE*, 6520:65200P, 2007.
[7] Singh, V.: Computational lithography: the new enabler of Moore's Law. *Proc. SPIE*, 682768271Q, 2007.
[8] Winder, R.O.:. Single stage threshold logic. *Switching Circuit Theory and Logical Design*, AIEE Special Publ. S-134, pages 321–322, 1960.
[9] Winder, R.O.: Partitions of N-Space by Hyperplanes. *SIAM J. Appl. Math.*, 811–818, 1966.