Searching by transpositions

G. Popescu *

November 14, 2006

1 Abstract

A graph optimization problem arising from topological construction of optimal interconnection networks is the embedding of an interconnection graph of n nodes in a graph of $m > n$ nodes. Interconnection networks are graphs with low diameter (logarithmic with the number of vertices), and sublogarithmic degree, which make them efficient for distributed indexing applications. These networks are used to connect parallel computing system components. Another application is the construction of communication networks with a deBruijn graph structure, where the mapping of data objects to nodes is given by a hash function. To route or look-up data from the network, the address of data items is first obtained from the hash table. Network nodes route the request for data objects or packets addressed to distributed objects such that to minimize some network metric (e.g. network load, delay).

The optimal topology construction can be formulated as an embedding of an interconnection network graph in a complete graph with edge weights representing the metric to be minimized: 'given an interconnection network graph $G(V,E)$ and a complete graph $G_c(V_c, E_c, w)$ with the weight function defined over the edges of the complete graph, find a minimum average weight embedding of G in G_c'. This is equivalent with selecting a set of $|E|$ edges from the $N(N-1)/2$ edges with minimum average weight such that the connectivity rules of the interconnection network are satisfied.

We propose here an efficient graph embedding algorithms and provide an asymptotic analysis of algorithm performance. The optimal topology embedding search uses two matrices: the cost matrix and the interconnection topology adjancency matrix. The cost matrix - $A(i,j)$ - is a symmetric **nxn** matrix containing delay measurements between all pairs of overlay nodes. Each overlay node is identified by an index $N_i, i = 1..n$. The interconnection network topology matrix $C(k,l)$ is the adjacency matrix constructed from the interconnection topology graph structure. Note that the adjacency matrix of an interconnection topology represented as un undirected graph (e.g. hypercube) is symmetric and that the sum of all elements of the constraints matrix is the twice the number of edges in the interconnection topology. For interconnection networks represented as directed graphs (deBruijn), the sum of matrix elements is equal to the number of interconenction topology edges.

Let $\pi : I -> I$ be a permutation over the set of interconnection topology indices, $I = \{I_j, j = 1..n\}$. Given the two overlay and adjacency matrices A and C, the edge cost function can be written

*George.Popescu@acm.org

as:

$$F(\Pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} c_{\pi(i)\pi(j)} \tag{1}$$

.

The permutation matrix $\mathbf{\Pi}$ is obtained by arranging the columns of the identity matrix in the order given by the current index permutation π. By multiplying to the left and right of the transposed adjacency matrix, its rows and columns indices are permuted in the order of π.

In the matrix formulation, the transposition of pairs of node indices coresponds to exchanging two rows and two columns of the interconnection network adjacency matrix. Evaluating only the changes of cost function reduces the computations required for each step of the search iteration to a constant factor of the maximum per node connectivity of the interconnection topology.

Consider the transposition $t_{p^*q^*}$ applied to the current index permutation π_k, which leads to a new permutation of indices $\pi_{k+1} = \pi_k * t_{p^*q^*}$. The transposition search selects the direction of search by computing first the local gradient in the index permutation space. The local neighbourhood of a permutation is the set of all permutations accesible through transpositions of node indices. Each step of the gradient descent algorithm evaluates all transpositions of current indices, exchanging the indices corresponding to the minimum cost, while allowing cost increases for a maximum number of steps estimated from the structure of cost matrix. We compute algorithm complexity of transposition search and provide an asymptotic analysis of its performance.

References

[1] Aiken, W., *Total Relative Displacement of Permutations*, J of Combinatorial Theory, Series A87, 1999, pp. 1-21.

[2] G. Constatine, *Combinatorial Theory and Statictical Design* , J. Wiley & Sons, 1987.

[3] D. Cvetkovic, M. Doob, H. Sachs *Spectra of Graphs. Theory and Applications* , Academic Press, 1980.

[4] K. B. Reid, *Total relative displacement of Vertex Permutations of* $K_{n_1,n_2,\ldots n_t}$ Journal of Graph Theory, 41(2), 2002, pp. 85-100.

[5] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, and Hypercubes* , Morgan Kaufman, 1992.

[6] I. Tomescu, *Problems in Combinatorics and Graph Theory,* J. Wiley & Sons, 1985.