| | |
|---|---|
| Title | 埋め込みパターンとその関連操作に基づくグラフ埋め込み最適化 |
| Author(s) | 佐藤, 円 |
| Citation | |
| Issue Date | 2010-03 |
| Type | Thesis or Dissertation |
| Text version | author |
| URL | http://hdl.handle.net/10119/8948 |
| Rights | |
| Description | Supervisor:金子 峰雄, 情報科学研究科, 修士 |

# Optimization of Graph Embedding Based on Pattern-Embedding and Relevant Operations

Madoka Satou (0810029)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 9, 2010

**Keywords:** graph embedding, pattern-embedding, mesh-graph.

For an interconnection network and a parallel program, it is common to represent both the parallel program and network topology as undirected graphs $G$ and $H$. For the program, the set of vertices $V(G)$ represents the tasks and the set of edges $E(G)$ represents the communication requirements between these tasks. For the processor network, the set of vertices $V(H)$ represents the PEs, and the edges $E(H)$ represents the communication links between these PEs. Using these notations, it is possible to define an execution of the parallel program $G$ on a processor network $H$ as a one-to-one mapping: $V(G) \longrightarrow V(H)$ which is called graph embedding. A major ploblem with graph embedding is to assign a program graphs into a different interconnection networks with a fixed topology and a limited number of PEs. Some measures of efficiency of this embedding are the dilation which measures the maximal distance in $H$ between the images of adjacent vertices of $G$; and the expansion which measures the fraction of the PE's that are not used in the embedding.

Each of conventional studies on graph embedding focused on a specific pair of guest graph topology and host graph topology, and its discussion was deeply topology-dependant. In addition, solutions relyed mainly on the experience and discovery of experts. In a practical situation, we will encounter various graph topologies, various graph sizes and various shapes of graph boundary, and it is not practical to ask exparts on demand to solve one among such various instances. It motivates us to develop a generic and algorithmic method to solve various instances of graph embedding. In principle, we can construct a search method of $\Phi_V : V(G) \longrightarrow V(H)$ and $\Phi_E : E(G) \longrightarrow P(H)$, where $P(H)$ is a set of all paths in $H$. However in such a primitive search, a huge size of solution space, such as $|V(H)|^{|V(G)|}$ or $|V(H)|!/(|V(H) - V(G)|)!$ has to be handled, and it is intractable.

In this research, we pay attention to a regularity of guest graph and host graph, and we introduce a macroscopic treatment of graph embedding called pattern-embedding rather than individual vertex mapping. Also several operations on pattern-embedding are studied for yielding a diversity of graph-embedding solutions. As the first attempt to developping an algorithmic method for graph embedding, we have limited our discussions on mesh-graph for both guest graph and host graph. For a mesh-graph, (1)there are a number of edge types, and each vertex except boundary vertex has one outgoing edge and one ingoing edge from each edge type. A subgraph composed of one vertex, its outgoing edges and their end vertices is now called a unit subgraph. A mesh-graph has another feature that (2)for every vertex, the graph can be seen equivalently (except outside graph boundary). From these two properties, we can repeatedly apply the equivalent embedding for all unit subgraphs, which is defined as a pattern-embedding. Remark that a pattern-embedding is specified by a set of edge embedding for all edge types, without depending on the number of vertices nor the number of edges. Next, in order to generate embedding solutions, an operation on pattern-embedding called pattern-switching has been introduced, and the condition for a pattern-switching to preserve the guest graph topology has been shown. When we apply a pattern-switching, one pattern-embedding may possibly interfere with other pattern-embedding in terms of vertex-congestion. Hence, the condition that two or more pattern-embedding do not interfere each other is also derived. As a result of these studies, we have constructed a framefork to generate a diversity of embedding solutions with relatively small amount of design parameters. A practical search algorithm working on this framework is an important future work. Extensions and modifications of pattern-embedding and relevant operations in order to applying our approach to a larger class of graph embedding are also exciting future challenges.