Dynamics of Continuous, Discrete and Impulsive Systems
Series **A:** *Mathematical Analysis* **11 (2004)** 339-367

HYPERGRAPH PARTITIONING TECHNIQUES

Dorothy Kucar, Shawki Areibi[2] and Anthony Vannelli

Department of Electrical and Computer Engineering
University of Waterloo, Waterloo, Ontario N2L 3G1

[2]School of Engineering
Guelph University, Guelph, Ontario N1G 2W1

Abstract. Graph and hypergraph partitioning are important problems with applications in several areas including parallel processing, VLSI design automation, physical mapping of chromosomes and data classification. The problem is to divide the vertices of a hypergraph into k similar–sized blocks such that a given cost function defined over the hypergraph is optimized. The intent of this paper is to expose our readers to the theoretical and implementation issues of hypergraph partitioning techniques.

Keywords. Hypergraph Partitioning, Iterative Improvement, Multilevel Techniques, ILP Model, Eigenvector Methods, Simulated Annealing, Genetic Algorithms, Tabu Search

1 Introduction

Partitioning is used to divide a hypergraph into smaller, more manageable blocks while minimizing the number of connections between the blocks, called *cutnets*. Vertices that are strongly "related" are assigned to the same block; vertices that are weakly "related" are assigned to different blocks. The relationship between vertices is determined by the number of, or the weight of edges connecting the vertices. From this description, it is evident that partitioning is a generic tool applicable to problems in a wide variety of scientific disciplines:

- **Large–Scale Dynamical Systems**: Given a large dynamical system of the form

$$\mathbf{z}' = \hat{\mathbf{A}}\mathbf{z} + \hat{\mathbf{B}}\mathbf{u}$$

where $\mathbf{z}$ represents the *state* of the system and $\mathbf{u}$ the input to the system: the goal is to decouple the directed graph associated with the system into smaller, highly–connected subgraphs. The system takes on a block–triangular structure where the blocks correspond to strongly–connected structures. It is then computationally easy to determine whether the blocks are individually stable, hence whether the overall system is stable [52, 51].