High Quality (Hyper-)Graph Partitioning

Algorithm Engineering for NP-hard Graph Problems
Yaroslav Akhremtsev, Peter Sanders, Sebastian Schlag and Christian Schulz

1. The \( \lambda \)-balanced (Hyper-)Graph Partitioning Problem

- Graph: \( G = (V, E) \)
  - models relationships between objects
  - dyadic (2 ary) relationships

- Hypergraph: \( H = (V, E) \)
  - generalization: edges connect \( \geq 2 \) vertices
  - arbitrary (\( d \) ary) relationships

**Task:**
- Partition \( G / H = (V, E : V \rightarrow \mathbb{R}_+, \omega : E \rightarrow \mathbb{R}_+) \) into \( k \) disjoint blocks \( V_1, \ldots, V_k \) such that
  - blocks \( V_i \) are roughly equal-sized: \( \epsilon(V_i) \leq (1 + \epsilon) \frac{|V|}{k} \)
  - objective: the cut is minimized. \( \text{cut} = \sum_{e \in \text{cut}} w(e) \), connectivity = \( \sum_{e \in \text{cut}} (\lambda - 1) w(e) \), where \( \lambda = \# \text{blocks connected by edge } e \)

2. Applications

- VLSI Design
- Warehouse Optimization
- Complex Networks
- Route Planning
- Simulation
- Scientific Computing

3. Multilevel (Hyper-)Graph Partitioning

- Input (Hyper-)Graph
- Output Partition
- match / cluster
- contract
- refine
- uncontract
- 1. Coarsening
- 2. Initial Partitioning
- 3. Uncoarsening

4. Algorithmic Innovations

- Min-Hash Based Sparsification
- Community-Aware Coarsening
- Max-Flow Min-Cut Refinement
- Engineered FM Local Search
- Global Search Strategies
- Advanced Memory Models / Parallelization

5. Open Source (Hyper-)Graph Partitioning Software

- KaHIP - Karlsruhe High Quality Partitioning:
  - \url{http://algo2.iti.kit.edu/kahip/}
  - multilevel graph partitioning algorithms
  - \( n \)-level approach
  - parallel evolutionary algorithms
  - cut and connectivity optimization
  - perfectly balanced partitioning
  - recursive bisection & direct \( k \)-way

- KaHyPar - Karlsruhe Hypergraph Partitioning:
  - \url{http://www.kahypar.org}
  - multilevel graph partitioning algorithms
  - \( n \)-level approach
  - parallel evolutionary algorithms
  - cut and connectivity optimization
  - perfectly balanced partitioning
  - recursive bisection & direct \( k \)-way

6. Experimental Results

- Extensive experiments on large benchmark sets of graphs and hypergraphs confirm that KaHIP and KaHyPar compute high-quality solutions – outperforming competing state-of-the-art tools.

Selected References


Sebastian Schlag
Institute of Theoretical Informatics, Algorithms II
sebastian.schlag@kit.edu