

Bachelor-/Masterarbeit

Evolutionary Hypergraph Partitioning Revisited

Beschreibung

Das *Hypergraphpartitionierungsproblem* ist ein klassisches Optimierungsproblem, das einen Hypergraphen in k (ein Eingabeparameter) ungefähr gleich große Blöcke zerlegt und dabei die Anzahl der Kanten zwischen den Blöcken (der *Schnitt* der Partition) minimiert. Das Problem hat zahlreiche praktische Anwendungen von der Lastverteilung auf Parallelrechnern, über den Entwurf von Mikroprozessor-Chips bis hin zur Simulation von Quanten-Schaltkreisen.

Hypergraphpartitionierung ist NP-schwer, selbst wenn man nur gute Approximationen anstatt einer exakten Lösung finden möchte. Große Instanzen können in der Praxis deshalb nur über heuristische Verfahren gelöst werden. Hierbei gibt es ein breites Spektrum an verschiedenen Trade-Offs zwischen Laufzeit und Lösungsqualität. Ein bekanntes Verfahren zum Erreichen hoher Lösungsqualität ist die Verwendung eines *evolutionären Algorithmus*¹⁾. Hierbei wird ein grundlegendes Partitionierungsverfahren (basierend auf dem Multilevel-Ansatz) wiederholt ausgeführt, unter Nutzung von *Mutationsoperationen* zur Modifikation einer Partition in der aktuellen Population sowie *Kombinationsoperationen* zur Kombination zweier Partitionen.

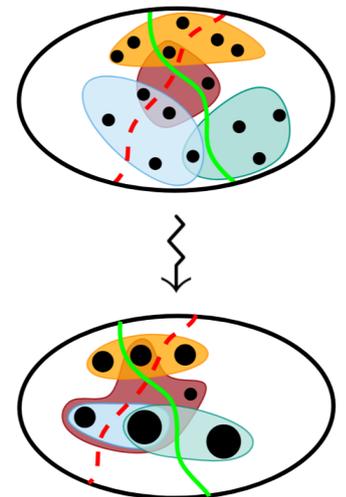
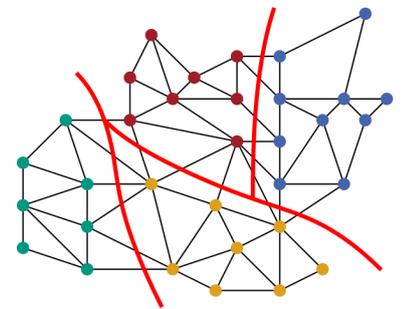
Der existierende evolutionäre Ansatz ist rein sequentiell – die Laufzeit lässt sich also auf Multicore-Systemen deutlich verbessern, indem eine parallele Variante entwickelt wird. Aus algorithmischer Sicht ist außerdem eine signifikante Schwäche des bestehenden Ansatzes, dass dieser in lokalen Minima stecken bleiben kann. In diesem Fall wird über zahlreiche Operationen hinweg keine Verbesserung mehr erzielt, obwohl die aktuelle Partition noch nicht optimal ist. Mögliche Techniken, um dies zu beheben, beinhalten Ansätze zur *Diversifizierung* der Population sowie *Neustarts*, die in regelmäßigen Abständen komplett neue Lösungen ohne Beachtung der bisherigen Population berechnen.

Thema der Arbeit

Ziel der Arbeit ist die Entwicklung eines evolutionären Ansatzes basierend auf einem modernen, parallelen Hypergraphpartitionierer. Hierbei soll Multicore-Parallelität möglichst effizient genutzt werden. Weiterhin sollen algorithmische Techniken entwickelt werden, die entweder das Steckenbleiben in lokalen Minima verhindern oder anderweitig zu besserer Performance beitragen.

Voraussetzungen

- Interesse an und solide Kenntnisse von Algorithmen und Datenstrukturen
- Gute Programmierkenntnisse in C++
- Grundlegende Kenntnisse von paralleler Programmierung



¹⁾ Robin Andre, Sebastian Schlag, and Christian Schulz. "Memetic Multilevel Hypergraph Partitioning." Genetic and Evolutionary Computation Conference, 2018.