Kolloquium zur Bachelorarbeit

Graphpartitionierung: Implementierung und Vergleich des Kernighan-Lin-Algorithmus und eines Multilevel-Ansatzes

Julian Eichen

27. Oktober 2020

Motivation

- Klassisches Problem: Graphpartitionierung

Motivation

- Klassisches Problem: Graphpartitionierung
- Anwedung in vielen Bereichen, in denen Graphen genutzt werden

Motivation

- Klassisches Problem: Graphpartitionierung
- Anwedung in vielen Bereichen, in denen Graphen genutzt werden
- insbesondere zur Fill-In Reduzierung bei direkten Lösungsverfahren

Motivation

- Klassisches Problem: Graphpartitionierung
- Anwedung in vielen Bereichen, in denen Graphen genutzt werden
- insbesondere zur Fill-In Reduzierung bei direkten Lösungsverfahren
- NP-schwer, daher Lösung mit Hilfe von Heuristiken

Graphpartitionierung

Problem

Gegeben:

Ungerichteter Graph $G = (V, E)$ *und eine Partitionsgröße* $p \in \mathbb{N}$

Gesucht:

Partition $V = V_1 \cup ... \cup V_p$ *mit:*

(P.1) $|V_l|$ *ist gleich groß für* $l = 1, ...p$

(P.2) $|S|$ *ist minimal für den induzierten Schnitt* S

Graphpartitionierung

Problem

Gegeben:

Ungerichteter Graph $G = (V, E)$ *und eine Partitionsgröße* $p \in \mathbb{N}$

Gesucht:

Partition $V = V_1 \cup ... \cup V_p$ *mit:*

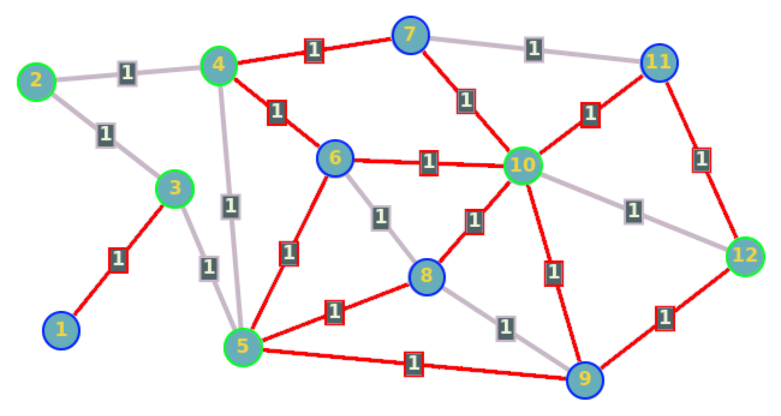
(P.1) $|V_l|$ *ist gleich groß für* $l = 1, ...p$

(P.2) $|S|$ *ist minimal für den induzierten Schnitt* S

Bemerkung

Falls man eine Partition mit $p > 2$ sucht, verwendet man im Allgemeinen rekursive Bisektion.

Beispiel



Graph mit Partitionierung für $p = 2$, Gewicht 1 auf allen Kanten und induziertem Schnitt der Größe 13

Fill-In

- Sei $Ax = b$ mit symmetrisch, positiv definitem $A = R^\intercal R$ und R obere Dreiecksmatrix.

Fill-In

- Sei $Ax = b$ mit symmetrisch, positiv definitem $A = R^\intercal R$ und R obere Dreiecksmatrix.
- Löse erst $R^\intercal c = b$ und dann $Rx = c$.

Fill-In

- Sei $Ax = b$ mit symmetrisch, positiv definitem $A = R^\intercal R$ und R obere Dreiecksmatrix.
- Löse erst $R^\intercal c = b$ und dann $Rx = c$.
- Ist A dabei dünn besetzt, kommt es bei der Berechnung von R zum sogenannten Fill-In und R ist nicht mehr dünn besetzt.

Fill-In

- Sei $Ax = b$ mit symmetrisch, positiv definitem $A = R^\intercal R$ und R obere Dreiecksmatrix.
- Löse erst $R^\intercal c = b$ und dann $Rx = c$.
- Ist A dabei dünn besetzt, kommt es bei der Berechnung von R zum sogenannten Fill-In und R ist nicht mehr dünn besetzt.
- Mit geschicktem Umordnen von A kann dieser verringert werden.

Fill-In

- Umordnung mit Hilfe von Graphpartitionierung:

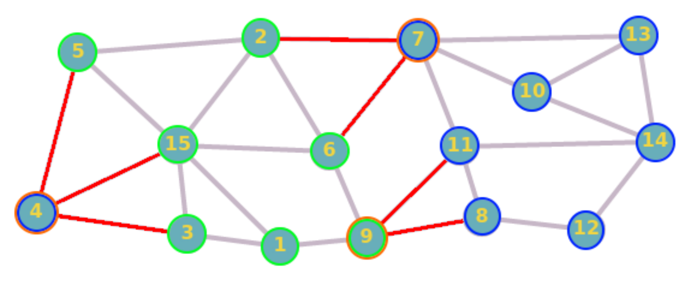
Fill-In

- Umordnung mit Hilfe von Graphpartitionierung:
- Im zu A gehörenden Graphen entsprechen die Knoten den Zeilen bzw. Spalten von A.

Fill-In

- Umordnung mit Hilfe von Graphpartitionierung:
- Im zu A gehörenden Graphen entsprechen die Knoten den Zeilen bzw. Spalten von A.
- Bestimme Partitionierung der Knotenmenge, eine Menge an Knoten welche die Partionen trennt und Ordne die Zeilen bzw. Spalten von A entsprechend an.

Beispiel



Partitionierter Graph, mit induziertem Schnitt und trennender Knotenmenge.

Kernighan-Lin

- Verbesserung einer initialen Partitionierung

Kernighan-Lin

- Verbesserung einer initialen Partitionierung
- Minimierung des induzierten Schnittes

Kernighan-Lin

- Verbesserung einer initialen Partitionierung
- Minimierung des induzierten Schnittes
- Tauschen von Knoten zwischen den Partitionen

Kernighan-Lin

- Verbesserung einer initialen Partitionierung
- Minimierung des induzierten Schnittes
- Tauschen von Knoten zwischen den Partitionen
- Finden einer optimalen Auswahl von Knoten ist NP-schwer

Kernighan-Lin

- Verbesserung einer initialen Partitionierung
- Minimierung des induzierten Schnittes
- Tauschen von Knoten zwischen den Partitionen
- Finden einer optimalen Auswahl von Knoten ist NP-schwer
- Beschränkung auf lokales Optimum

Kernighan-Lin [2]

- Es wird in zwei Schleifen gearbeitet:
  1. Äussere Schleife: Prüft ob die innere Schleife eine Verbesserung erzeugt und läuft solange diese ein Verbesserung erzeugt.
  2. Innere Schleife: Tauscht Knotenmengen, jeder Knoten darf dabei maximal einmal getauscht werden.

Kernighan-Lin [2]

- Es wird in zwei Schleifen gearbeitet:
  1. Äussere Schleife: Prüft ob die innere Schleife eine Verbesserung erzeugt und läuft solange diese ein Verbesserung erzeugt.
  2. Innere Schleife: Tauscht Knotenmengen, jeder Knoten darf dabei maximal einmal getauscht werden.

Kernighan-Lin [2]

- Es wird in zwei Schleifen gearbeitet:
  1. Äussere Schleife: Prüft ob die innere Schleife eine Verbesserung erzeugt und läuft solange diese ein Verbesserung erzeugt.
  2. Innere Schleife: Tauscht Knotenmengen, jeder Knoten darf dabei maximal einmal getauscht werden.
- Bei beiden Schleifen ergeben sich Möglichkeiten zur Modifikation.

Kernighan-Lin [2]

- Es wird in zwei Schleifen gearbeitet:
  1. Äussere Schleife: Prüft ob die innere Schleife eine Verbesserung erzeugt und läuft solange diese ein Verbesserung erzeugt.
  2. Innere Schleife: Tauscht Knotenmengen, jeder Knoten darf dabei maximal einmal getauscht werden.
- Bei beiden Schleifen ergeben sich Möglichkeiten zur Modifikation.
- Wir bewegen zum Beispiel immer den besten Knoten aus der größeren Partition in die kleinere.

Kernighan-Lin

- Bei der Auswahl der Knoten sind vor allem zwei Größen zu beachten:

Kernighan-Lin

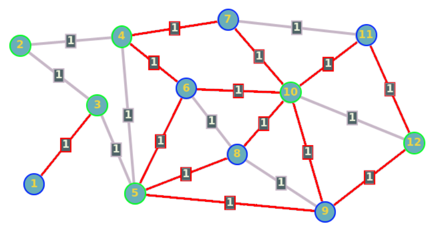
- Bei der Auswahl der Knoten sind vor allem zwei Größen zu beachten:

- $E_{\mathcal{V}}(a) = \sum_{e=\{a,b\} \in E} w(e),\ a \in A,\ b \in B,\ \textit{Äussere Kosten}$

- $I_{\mathcal{V}}(a) = \sum_{e=\{a,a'\} \in E} w(e),\ a, a' \in A,\ \textit{Innere Kosten}$

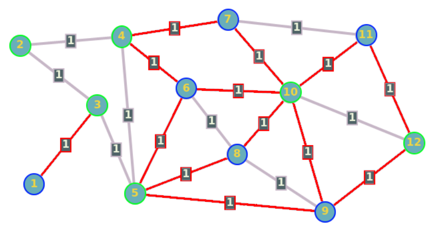
- mit $G = (V, E)$ ein Graph und $A, B \subset V$ eine Partitionierung von G

Beispiel



Graph mit initaler Partitionierung, Gewicht 1 auf allen Kanten und Schnitt der Größe 13

Beispiel



Graph mit initaler Partitionierung, Gewicht 1 auf allen Kanten und Schnitt der Größe 13

- Knoten 1:
  $E_{\mathcal{V}}(a) = 1$ und $I_{\mathcal{V}}(a) = 0$
- Knoten 2:
  $E_{\mathcal{V}}(a) = 0$ und $I_{\mathcal{V}}(a) = 2$
- Knoten 10:
  $E_{\mathcal{V}}(a) = 5$ und $I_{\mathcal{V}}(a) = 1$

Multilevel-Partitionierung [2]

- Prinzip zur Partitionierung von Graphen mit einer großen Anzahl an Knoten

Multilevel-Partitionierung [2]

- Prinzip zur Partitionierung von Graphen mit einer großen Anzahl an Knoten
- Besteht aus drei Teilen:

Multilevel-Partitionierung [2]

- Prinzip zur Partitionierung von Graphen mit einer großen Anzahl an Knoten
- Besteht aus drei Teilen:
  1. Vergröberung
  2. Partitionierung
  3. Verfeinerung

Multilevel-Partitionierung [2]

- Prinzip zur Partitionierung von Graphen mit einer großen Anzahl an Knoten
- Besteht aus drei Teilen:
  1. Vergröberung
  2. Partitionierung
  3. Verfeinerung
- Für die einzelnen Teile gibt es jeweils verschiedene Möglichkeiten der Umsetzung

Vergröberung [2]

- Matching als grundlegende Struktur.

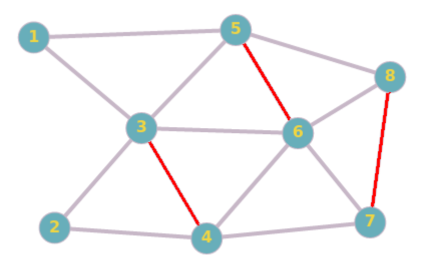
Vergröberung [2]

- Matching als grundlegende Struktur.
- Knotenpaare und die jeweilige verbindende Kante werden zu einem Knoten zusammengefasst.

Vergröberung [2]

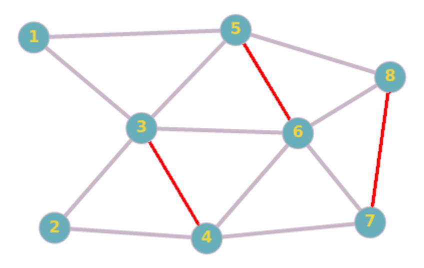
- Matching als grundlegende Struktur.
- Knotenpaare und die jeweilige verbindende Kante werden zu einem Knoten zusammengefasst.
- Die Nachbarschaftsbeziehungen der beiden Knoten werden vereinigt und auf den neuen Knoten übertragen.

Vergröberung

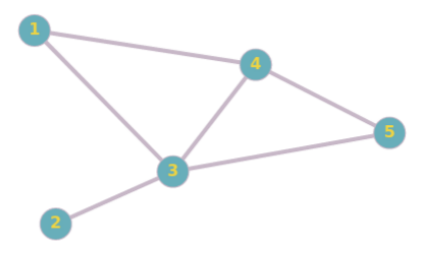


Matching im Ursprungsgraphen

Vergröberung



Matching im Ursprungsgraphen



Graph nach einer Vergröberung

Partitionierung [2]

- Die initiale Partitionierung wird auf dem gröbsten und damit kleinsten Graphen bestimmt.

Partitionierung [2]

- Die initiale Partitionierung wird auf dem gröbsten und damit kleinsten Graphen bestimmt.
- Aufgrund der geringen Anzahl an Knoten kann grundsätzlich ein beliebiger Algorithmus benutzt werden.

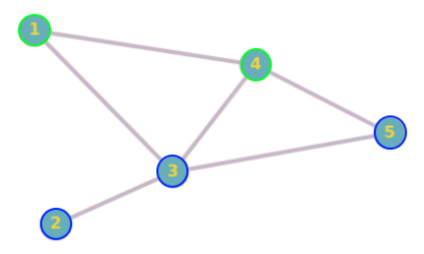
Partitionierung [2]

- Die initiale Partitionierung wird auf dem gröbsten und damit kleinsten Graphen bestimmt.
- Aufgrund der geringen Anzahl an Knoten kann grundsätzlich ein beliebiger Algorithmus benutzt werden.
- Da die Qualität der Partition jedoch den weiteren Verlauf der Multilevelpartitionierung beeinflusst sollte diese möglichst optimal sein.

Partitionierung [2]

- Die initiale Partitionierung wird auf dem gröbsten und damit kleinsten Graphen bestimmt.
- Aufgrund der geringen Anzahl an Knoten kann grundsätzlich ein beliebiger Algorithmus benutzt werden.
- Da die Qualität der Partition jedoch den weiteren Verlauf der Multilevelpartitionierung beeinflusst sollte diese möglichst optimal sein.
- Zum Beispiel bieten sich spektrale Bisektion oder Kernghan-Lin angewandt auf eine zufällige gewählte Partitionierung an.

Partitionierung



Partionierung des gröbsten Graphen

Verfeinerung [2]

- Projektion der initialen Partitionierung auf den Ursprungsgraphen

Verfeinerung [2]

- Projektion der initialen Partitionierung auf den Ursprungsgraphen
- Besteht für jede Vergröberungsstufe wiederum aus zwei Teilen:

Verfeinerung [2]

- Projektion der initialen Partitionierung auf den Ursprungsgraphen
- Besteht für jede Vergröberungsstufe wiederum aus zwei Teilen:
- Projektion der Partitionierung auf den nächst feineren Graphen

Verfeinerung [2]

- Projektion der initialen Partitionierung auf den Ursprungsgraphen
- Besteht für jede Vergröberungsstufe wiederum aus zwei Teilen:
- Projektion der Partitionierung auf den nächst feineren Graphen
  - Jeder Knoten wird wieder in seine Ursprungsknoten zerlegt

Verfeinerung [2]

- Projektion der initialen Partitionierung auf den Ursprungsgraphen
- Besteht für jede Vergröberungsstufe wiederum aus zwei Teilen:
- Projektion der Partitionierung auf den nächst feineren Graphen
  - Jeder Knoten wird wieder in seine Ursprungsknoten zerlegt
  - Ursprüngliche Nachbarschaftsbeziehungen werden wiederhergestellt

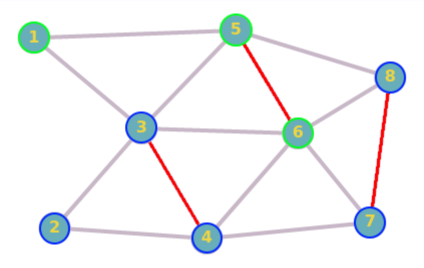
Verfeinerung [2]

- Projektion der initialen Partitionierung auf den Ursprungsgraphen
- Besteht für jede Vergröberungsstufe wiederum aus zwei Teilen:
- Projektion der Partitionierung auf den nächst feineren Graphen
  - Jeder Knoten wird wieder in seine Ursprungsknoten zerlegt
  - Ursprüngliche Nachbarschaftsbeziehungen werden wiederhergestellt
- Optimierung der projizierten Partitionierung

Verfeinerung [2]

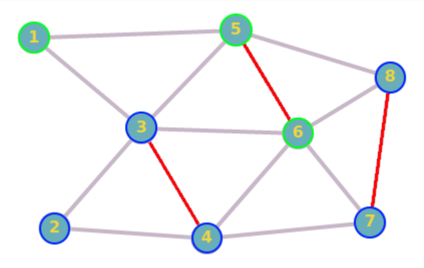
- Projektion der initialen Partitionierung auf den Ursprungsgraphen
- Besteht für jede Vergröberungsstufe wiederum aus zwei Teilen:
- Projektion der Partitionierung auf den nächst feineren Graphen
  - Jeder Knoten wird wieder in seine Ursprungsknoten zerlegt
  - Ursprüngliche Nachbarschaftsbeziehungen werden wiederhergestellt
- Optimierung der projizierten Partitionierung
  - Zum Beispiel durch den Kernighan-Lin Algorithmus

Verfeinerung

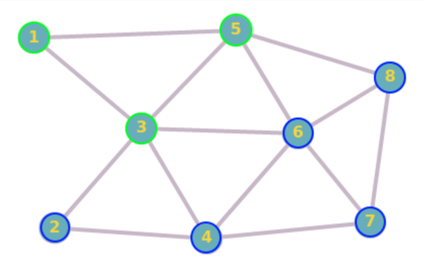


Projektion der Partitionierung auf den Ursprungsgraphen

Verfeinerung



Projektion der Partitionierung auf den Ursprungsgraphen



Ursprungsgraph mit optimierter Partitionierung

Anwendung bezüglich Fill-In

- Vier Varianten der Graphpartitionierung:

Anwendung bezüglich Fill-In

- Vier Varianten der Graphpartitionierung:
  1. Kernighan-Lin
  2. Multilevel mit KL zur initalen Partitionierung und Verfeinerung
  3. Multilevel mit nur einem Durchlauf der inneren Schleife von KL
  4. Multilevel mit nur einem Durchlauf der inneren Schleife,
     nur Knoten im Schnitt werden getauscht

Anwendung bezüglich Fill-In [2]

- Angewandt auf fünf verschiedene s.p.d. Matrizen wie folgt:

Anwendung bezüglich Fill-In [2]

- Angewandt auf fünf verschiedene s.p.d. Matrizen wie folgt:
  1. Matritzen nach dem oben beschriebenen Schema umsortieren
  2. Zerlegung in $A = R^\intercal R$, R obere Dreiecksmatrix
  3. Betrachten des Fill-In

Ergebnisse [1]

Anzahl der Einträge in R ungleich Null						
Matrix	$\|V\|$	KL	mlKL	mlKL1	mlKL1b	A ohne Perm.
bcsstk24	3562	1181441	1318372	1294697	1262387	2031722
sts4098	4098	1337218	1430365	1630803	13321013	4682946
bcsstk38	8032	2239028	2390788	2460518	2355263	1684342
fv3	9801	5424491	1869900	4571049	5120160	980001
bloweybq	10001	68174	51555	57870	50102	49993

Literatur

Website des Department of Computer & Information Science & Engineering der Universitity of Florida
https://www.cise.ufl.edu
letzter Zugriff 19.10.2020

George Karypis and Vipin Kumar. *A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs Army High Performance Computing Research Center, 1991*