

k -Zusammenhang im Semi-Streaming Modell

Mariano Zelke

Humboldt-Universität zu Berlin, Institut für Informatik, 10099 Berlin

zelke@informatik.hu-berlin.de

Das Semi-Streaming Modell wurde 2003 von Muthukrishnan¹ eingeführt und erweist sich als praktisch bei der Lösung von graphentheoretischen Problemen auf riesigen Graphen. Solche Graphen können während des Rechenvorganges aufgrund ihrer Größe nicht komplett im Hauptspeicher abgelegt werden, vielmehr sind sie auf externen Speichereinheiten deponiert. Diese geben den gespeicherten Graphen als Datenstrom seiner Kanten in der Reihenfolge aus, in der sie gespeichert sind. Ein *Semi-Streaming Algorithmus* kann diesen Datenstrom verarbeiten, nimmt also die Kanten eines Graphen $G = (V, E)$ in beliebiger Reihenfolge entgegen, ist dabei allerdings auf einen Speicherbedarf von $\mathcal{O}(|V| \cdot \text{polylog } |V|)$ Bits beschränkt. Insbesondere heißt dies, dass der Algorithmus sich nicht alle Kanten eines dichten Graphen mit $|V| = o(|E|)$ merken kann.

Ein Graph G ist k -zusammenhängend, wenn die Entfernung von mindestens k Knoten nötig ist, damit G in mehrere Komponenten zerfällt. Es sind Semi-Streaming Algorithmen bekannt, die für $k = 1, \dots, 4$ bestimmen, ob ein Graph k -zusammenhängend ist². Wir präsentieren zwei verschiedene Semi-Streaming Algorithmen, die dies für beliebiges konstantes k beherrschen.

Der erste Algorithmus liest den Datenstrom D der Kanten des Eingabegraphen genau einmal und benötigt eine Zeit von $\mathcal{O}(k^2 n)$ für die Verarbeitung einer Kante. Der zweite Algorithmus liest D $k + 1$ mal und verarbeitet eine Kante in $\mathcal{O}(k + \alpha(n))$. Dabei ist $\alpha(n)$ das Inverse der Ackermannfunktion.

Beide Algorithmen bauen auf der gleichen Grundidee auf: Während sie die Kanten des Eingabegraphen G lesen, bauen sie einen Subgraphen von G auf, der linear viele Kanten besitzt und in den beschränkten Speicherplatz eines Semi-Streaming Algorithmus passt. Dieser Subgraph wird schließlich benutzt, um den k -Zusammenhang des Eingabegraphen zu entscheiden.

¹S. Muthukrishnan. *Data streams: Algorithms and applications*. 2003. <http://athos.rutgers.edu/~muthu/stream-1-1.ps>

²J. Feigenbaum, S. Kannan, A. McGregor, S. Suri und J. Zhang. *Graph Distances in the Streaming Model: the Value of Space*. SODA 2005: 745-754.