Ein einfacher max-cut Algorithmus für planare Graphen

F. Liers and G. Pardella

Institut für Informatik, Universität zu Köln,
Pohligstraße 1, D-50969 Köln, Germany,
{liers, pardella}@informatik.uni-koeln.de,
`http://cophy.informatik.uni-koeln.de`

Sei $G = (V, E)$ ein (gewichteter) Graph mit Knotenmenge V und Kantenmenge E. Das MAX-CUT Problem besteht darin, eine Partitionierung der Knotenmenge V in zwei Knotenmengen $W \subseteq V$ und $V \backslash W$ zu finden, so dass die Summe der Kantengewichte $w(e)$ der Kanten $e = (v, w)$ mit $v \in V \backslash W,\ w \in W$ maximal ist. Partitionierungsprobleme bzw. Schnittprobleme finden sich in vielfältigen Anwendungsproblemen. Darunter sind zum Beispiel VIA Minimierung im Schaltkreisentwurf, Physik von ungeordneten Systemen oder auch in der Funktionssicherheit von Netzwerken. Desweiteren ist das MAX-CUT Problem äquivalent zu quadratischer 0-1 Optimierung ohne Nebenbedingungen.

Für beliebig gewichtete allgemeine Graphen ist das MAX-CUT Problem (und durch Vorzeicheninvertierung der Kantengewichte auch das MIN-CUT Problem) NP-schwer. Jedoch ist es für gewisse Graphklassen polynomiell lösbar, insbesondere existieren diverse polynomielle Lösungsstrategien für planare Graphen. Dabei folgen diese Ansätze alle dem selben Schema: auf einem geeigneten Hilfsgraphen wird ein gewichtetes perfektes Matching bestimmt. Dieses induziert dann einen Schnitt im Originalgraphen. Wir folgen diesem Schema und präsentieren einen neuen und einfachen Algorithmus für das MAX-CUT Problem auf beliebig gewichteten planaren Graphen. Die Laufzeit unseres Verfahrens kann mit $O(|V|^{\frac{3}{2}} \log |V|)$ im schlimmsten Fall abgeschätzt werden. Diese Laufzeitschranke entspricht der des Verfahrens von Shih, Wu und Kuo, welche den derzeit schnellsten Algorithmus für das Problem vorgestellt haben. Jedoch arbeitet unsere Methode auf einem wesentlich kleineren Hilfsgraphen, welcher in linearer Zeit berechnet werden kann und eine einfache Struktur aufweist. Da die meiste Rechenzeit auf die Berechnung des Matchings im Hilfsgraphen abfällt, ist es einsichtig, dass unser Verfahren in der Praxis schneller und ressourcenschonender ist. Wir können mit unserem Programm Schnitte in großen realistischen sowie zufälligen Instanzen mit über 10^6 Knoten effizient zu berechnen.

Hierzu erstellen wir den Dualgraphen, welchen wir geeignet transformieren und dabei jeden Knoten durch einen K_4 Subgraphen (sogenannte *Kasteleyn cities*, basierend auf einer Arbeit von Kasteleyn aus den sechzigern Jahren) ersetzen. Auf diesem so konstruierten Graphen berechnen wir ein maximal (minimal) gewichtetes perfektes Matching. Dieses Matching liefert maximal (minimal) gewichtete Euler-Subgraphen im Dualgraphen, welche zu einem optimalen Schnitt im Originalgraphen korrespondieren. Die Erstellung des Hilfsgraphen erfolgt in linearer Zeit. Mit Hilfe des *planar separator* Theorems von Lipton und Tarjan erreichen wir die erwähnte Laufzeitschranke.