

Parametrisierte Reduktionen

Hitting Set

Die Aufgabe besteht darin zu zeigen, dass Hitting Set $W[2]$ -Vollständig ist.

Schwere

Für die $W[2]$ -schwere reduzieren wir von Dominating Set. Tatsächlich muss man da nicht viel mehr tun, als die Definition der beiden Probleme anzuschauen:

- **Dominating Set:** Wähle Knoten aus, sodass für jede abgeschlossene Nachbarschaft mindestens ein Knoten gewählt wurde.
- **Hitting Set:** Wähle Elemente aus, sodass für jede Menge ein Knoten ausgewählt wurde.

Man kann also, gegeben die Instanz von Dominating Set, einfach die abgeschlossenen Nachbarschaften als Mengen wählen und erhält eine äquivalente Instanz.

Vollständigkeit

Für enthaltensein in $W[2]$ können wir beispielsweise auf Dominating Set reduzieren. Da wir bei Hitting Set Elemente und bei Dominating Set Knoten auswählen, erscheint es sinnvoll die Elemente als Knoten zu repräsentieren. Zusätzlich zu diesen Element-Knoten müssen wir jetzt für Mengen von Knoten fordern, dass mindestens ein Knoten aus der Menge gewählt wird. Das können wir erreichen, indem wir zusätzliche Knoten einfügen. Wir müssen dann sicherstellen, dass die Element-Knoten selbst immer automatisch dominiert werden und dass keine anderen Knoten ausgewählt werden.

Für jede zu hittenden Mengen kann man einfach einen Mengen-Knoten einfügen, der mit den Element-Knoten der Menge verbunden ist (wir repräsentieren die Hitting Set Instanz also als bipartiten Graphen). Außerdem verbinden wir die Element-Knoten zu einer Clique, damit die Element-Knoten selbst immer dominiert werden (sobald mindestens ein Element-Knoten ausgewählt wurde).

Wenn man jetzt eine Lösung der Hitting Set Instanz hat, so kann man einfach die zugehörigen Element-Knoten auswählen und hat offensichtlich eine Lösung.

Betrachte nun eine Lösung der Dominating Set Instanz. Falls dort Mengen-Knoten ausgewählt wurden, so kann man für jeden ausgewählten Mengen-Knoten stattdessen auch einen adjazenten Element-Knoten auswählen und erhält eine nicht schlechtere Lösung. Wir können also annehmen, dass nur Element-Knoten ausgewählt wurden. Die zugehörigen Elemente bilden dann eine Lösung der Hitting Set Instanz.

List Coloring (Baumweite)

Ziel ist es, von Multicolored Independent Set zu reduzieren. Wir kürzen die beiden Probleme im Folgenden mit MIS und LC ab. Wie so oft bei Reduktionen lohnt es sich auch hier die folgenden drei Fragen zu stellen:

1. Welche Entscheidungen treffe ich bei den Problemen MIS und LC?
2. Welche Bedingungen stellt eine MIS-Instanz an diese Entscheidungen?
3. Wie kann ich diese Bedingungen in LC erzwingen ohne zusätzliche Bedingungen zu generieren?

Zu treffende Entscheidungen

- Bei MIS muss man für jede Farbe einen Knoten auswählen.

- Bei LC muss man für jeden Knoten eine Farbe auswählen.

Das sieht erstmal so aus als wollten wir Farben in Knoten und Knoten in Farben übersetzen. Der erste Schritt unserer Reduktion besteht genau darin. Betrachte die MIS-Instanz bestehend aus dem Graphen $G = (V, E)$ und der k -Färbung $V = V_1 \cup \dots \cup V_k$. Für die LC-Instanz, erstelle zunächst einen Knoten v_i für jede Farbe $i \in [k]$ und setze V_i als die Liste der für v_i erlaubten Farben. Jede Lösung der LC-Instanz muss jetzt also für jedes der v_i ein Element aus V_i auswählen. Für die MIS-Instanz entspricht das der Auswahl eines Knotens aus V_i für die Farbe i .

Bedingungen an die Entscheidungen

Die Entscheidungen bei MIS sind jetzt insofern eingeschränkt, dass man keine zwei Knoten wählen darf, die mit einer Kante verbunden sind. für eine Kante $\{u, w\}$ mit $u \in V_i$ und $w \in V_j$ muss man also erzwingen, dass nicht gleichzeitig u für die Farbe i und w für die Farbe j gewählt wird.

Übersetzt auf die LC-Instanz heißt das, dass der Knoten v_i und der Knoten v_j die Farben u bzw. w in ihrer Liste an Farben enthalten, aber nicht beide Knoten zusammen gewählt werden dürfen. Das können wir erzwingen, indem wir einen Extraknoten x einfügen, der mit v_i und v_j verbunden ist und als mögliche Farben nur u und w hat. In jeder Lösung der LC-Instanz müssen wir dann eine der Farben u oder w für x wählen und können damit nicht beide Farben für v_j wählen.

Zusammenstecken

Jetzt muss man nur noch einmal die Dinge zusammenstecken und durchargumentieren, dass alles passt. Folgende Dinge sollte man dabei zeigen (sie sind alle recht leicht einzusehen sind):

- Der resultierende Graph hat Baumweite k .
- Eine Lösung der MIS-Instanz liefert eine Lösung für die LC-Instanz.
- Eine Lösung für die LC-Instanz liefert eine Lösung für die MIS-Instanz.

Strongly Connected Steiner Subgraph

Das SCSS Problem hat als Eingabe einen gerichteten Graphen zusammen mit einer Menge von Terminalknoten. Ziel ist es bis zu k Knoten auszuwählen, sodass der davon induzierte Teilgraph stark zusammenhängend ist und alle Terminalknoten enthält (die Terminalknoten sind in dem k enthalten).

Ziel ist es, von Multicolored Clique zu reduzieren und damit $W[1]$ -Schwere zu zeigen.

Erster Versuch

Analog zu vorher könnte man hier feststellen, dass bei Multicolored Clique als auch bei SCSS eine Knotenmenge gesucht wird. Wählt man diese direkte Übertragung, dann muss man noch folgende Bedingungen mittels SCSS modellieren:

- Aus jeder Farbmenge wird genau ein Knoten ausgewählt.
- Je zwei ausgewählte Knoten sind verbunden.

Die erste Bedingung ist leicht zu realisieren: Für jede Farbe kann man einen Terminalknoten erstellen, der mit allen Knoten dieser Farbe verbunden ist (in beide Richtungen gerichtet), aber sonst keinerlei Verbindungen bekommt. Da man jeden Terminalknoten mit dem Rest der Instanz verbinden muss, muss man wenigstens einen der Knoten aus der Farbe auswählen. Wenn wir insgesamt nur die Auswahl von $2k$ Knoten erlauben, dann erzwingen wir damit, dass aus jeder Farbe genau ein Knoten gewählt wird (zusätzlich zu dem Terminalknoten für die Farbe).

Die zweite Bedingung (je zwei ausgewählte Knoten sind verbunden) ist jedoch deutlich schwieriger zu erzwingen. Dass die ausgewählten Knoten irgendwie verbunden sind kann man sicherlich erzwingen, aber dafür zu sorgen, dass nicht schon weniger Kanten ausreichen um starken Zusammenhang zu bekommen erscheint unmöglich.

Auswahl von Knoten und Kanten

Statt nur Knoten auf Knoten zu mappen, bilden wir jetzt auch Kanten auf Knoten ab. Das heißt, wir wollen das SCSS-Problem nicht nur die Knoten der Clique, sondern auch die Kanten der Clique auswählen lassen. Dazu kann man auch die Kanten entsprechend der Farben partitionieren: Für jedes Paar an Farben gibt es eine Menge von Kanten, die Knoten dieser beiden Farben verbinden. Wir erstellen also, gegeben die Clique-Instanz G , für jeden Knoten und für jede Kante in G jeweils einen Knoten in der SCSS-Instanz und nennen diese Knoten-Knoten bzw. Kanten-Knoten. Jetzt müssen wir folgende Bedingungen erzwingen:

- Aus jeder Farbmenge von Knoten-Knoten wird genau ein Knoten-Knoten ausgewählt.
- Aus jeder Farbpaarmenge von Kanten-Knoten wird genau ein Kanten-Knoten ausgewählt.
- Die ausgewählten Kanten-Knoten passen zu den ausgewählten Knoten-Knoten (also: für jede Kante wurden auch die beiden Endpunkte ausgewählt).

Die ersten beiden Bedingungen können wir genauso wie oben erzwingen. Dazu müssen wir den neuen Parameter für die SCSS-Instanz entsprechend erhöhen (hängt aber weiterhin nur von k ab).

Für die letzte Bedingung, betrachte eine Kante $\{u, v\}$ und sei uv der zugehörige Kanten-Knoten. Außerdem seien u und v die Knoten-Knoten von u bzw. v . Dann erstellen wir in der SCSS-Instanz die Kanten (u, uv) und (uv, v) (die Richtung von u nach v ist hier beliebig gewählt).

Wenn wir jetzt den Kanten-Knoten uv wählen, aber nicht u , so hat der Teilgraph bestehend aus uv und dem benachbarten Terminalknoten keine eingehende Kante und damit ist der resultierende Graph nicht stark zusammenhängend. Analog hat er keine ausgehende Kante, falls v nicht gewählt wird.

Zusammengefasst: In jeder Lösung der SCSS-Instanz wählen wir für jedes Paar von Farben eine Kante, und wir wählen für jede Kante ihre Endpunkte, haben am Ende aber nur genau einen Knoten pro Farbe gewählt. Damit haben wir eine Clique.

Problem

Die Umkehrung gilt nicht so ohne weiteres. Es könnte sein, dass es eine Bunte Clique gibt, bei der wir die Orientierung der Kanten oben so gewählt haben, dass wir keinen starken Zusammenhang bekommen (z.B. wenn ein Knoten-Knoten ausschließlich eingehende Kanten hat).

Das Problem liegt hier darin, dass wir zwar die oben geforderten Bedingungen korrekt umgesetzt haben, dabei aber unbewusst zusätzliche Bedingungen eingebaut haben. Wir müssen jetzt also nochmal iterieren und versuchen dieses Problem zu lösen, ohne dabei die Bedingungen von oben kaputt zu machen.

Lösung

Die entscheidende Beobachtung besteht hier darin, dass die Bedingung "für jeder Farbe wurde ein Knoten gewählt" schon aus den anderen Bedingungen folgt. Wenn für jedes Farbenpaar eine Kante gewählt wird und für jede Kante ihre Knoten gewählt werden, so wurde auch für jede Farbmenge mindestens ein Knoten gewählt (und damit genau ein Knoten, weil wir sonst zu viele wählen). Wir können das also die Terminalknoten pro Farbe erstmal weglassen, was die Instanz etwas vereinfacht.

Um dann starken Zusammenhang zu bekommen können wir einen einzelnen Terminalknoten einfügen, der mit allen Knoten-Knoten verbunden ist (Kanten in beide Richtungen). Der Parameter muss entsprechend angepasst werden.

Jetzt kann man verifizieren, dass das passt: Die Bedingungen von oben sind weiterhin gegeben. Damit bekommen wir für eine Lösung der SCSS-Instanz auch eine bunte Clique. Umgekehrt, wenn wir eine Clique haben, so können wir den eben Eignefügten Terminalknoten nutzen, um starken Zusammenhang zu bekommen: Für jeden anderen Knoten finden wir einen gerichteten Pfad sowohl zu als auch von diesem Terminalknoten.