6

Fest-Parameter-Algorithmen für ausgewählte Graphenprobleme

In den Abschnitten 3.4 und 5.1 wurden einige Probleme vorgestellt, die NP-vollständig oder NP-hart sind, die sich also nicht in Polynomialzeit lösen lassen, außer wenn die für unwahrscheinlich gehaltene Gleichheit P = NP gelten würde. Dazu gehören auch viele wichtige Graphenprobleme, wie zum Beispiel die in Abschnitt 3.4 definierten NP-vollständigen Probleme UNABHÄNGIGE MENGE, KNOTENÜBERDECKUNG, PARTITION IN k UNABHÄNGIGE MENGEN (auch bekannt als k-FÄRBBARKEIT, siehe Satz 5.26), PARTITION IN CLIQUEN, DOMINIERENDE MENGE, DOMATISCHE ZAHL und das TRAVELING SALESPERSON PROBLEM sowie das NP-harte UNIQUE OPTIMAL TRAVELING SALESPERSON PROBLEM, für das Papadimitriou [Pap84] zeigte, dass es vollständig für die Komplexitätsklasse P^{NP} ist.

Wie in Abschnitt 5.2 erläutert wurde, gibt es verschiedene Möglichkeiten, mit der Härte solcher Probleme umzugehen. Ein Ansatz, der in Kapitel 7 auf Färbbarkeitsprobleme für Graphen angewandt wird, besteht in der Verbesserung der vorhandenen Exponentialzeit-Algorithmen. Andere Ansätze, wie etwa der Entwurf von effizienten Approximations-, randomisierten oder heuristischen Algorithmen für harte Probleme, werden in diesem Buch nicht betrachtet; wir beschränken uns (bis auf wenige Ausnahmen – siehe z. B. die Abschnitte 7.1.3, 7.1.4 und 7.4.2) auf *exakte*, *deterministische* und stets *korrekte* Algorithmen für harte Graphenprobleme. In diesem Kapitel wenden wir uns einem weiteren Ansatz zu, den harten Graphenproblemen beizukommen, nämlich durch den Entwurf von Fest-Parameter-Algorithmen.

Wir wiederholen zunächst die Definition von parametrisierten Algorithmen (FPT-Algorithmen), siehe Definition 5.45.

Es sei (Π, κ) ein parametrisiertes Problem, d. h., Π ist ein Entscheidungsproblem mit der Instanzenmenge $\mathscr{I}$ und

$$\kappa : \mathscr{I} \to \mathbb{N}$$

eine Parametrisierung (ein Parameter) von Π. Ein Algorithmus A für Π heißt *parametrisierter Algorithmus* (FPT-*Algorithmus*) bezüglich der Parametrisierung κ, falls es eine berechenbare Funktion

$$f : \mathbb{N} \mapsto \mathbb{N}$$

F. Gurski et al., *Exakte Algorithmen für schwere Graphenprobleme*, eXamen.press,
DOI 10.1007/978-3-642-04500-4_6,

und ein Polynom p gibt, sodass für jede Instanz $I \in \mathscr{I}$ die Laufzeit von A bei Eingabe von I durch

$$f(\kappa(I)) \cdot p(|I|)$$

oder durch

$$f(\kappa(I)) + p(|I|)$$

abgeschätzt werden kann. Parametrisierte Probleme, die durch einen parametrisierten Algorithmus gelöst werden können, heißen *fest-Parameter-berechenbar* und werden in der Komplexitätsklasse FPT zusammengefasst.

6.1 Knotenüberdeckung

In diesem Abschnitt stellen wir einige parametrisierte Algorithmen für das folgende parametrisierte Problem vor, das bereits in Beispiel 5.44 eingeführt wurde:

p-KNOTENÜBERDECKUNG	
Gegeben:	Ein Graph $G = (V,E)$ und eine Zahl $k \in \mathbb{N}$.
Parameter:	k.
Frage:	Gibt es in G eine Knotenüberdeckung der Größe höchstens k?

Dabei beschreiben wir unterschiedliche Entwurfstechniken für parametrisierte Algorithmen.

6.1.1 Problemkernreduktion

Unser erster Algorithmus für p-KNOTENÜBERDECKUNG beruht auf der Idee, eine gegebene Instanz I so auf eine Instanz I' zu reduzieren, dass die Größe von I' funktional nur noch vom Parameter und *nicht* von der Instanzengröße $|I|$ abhängig ist. Formal bezeichnet man dieses Vorgehen als Reduktion auf einen Problemkern.

Definition 6.1 (Problemkernreduktion). *Es seien (Π, κ) ein parametrisiertes Problem, $\mathscr{I}$ die Instanzenmenge von Π und $\kappa : \mathscr{I} \to \mathbb{N}$ eine Parametrisierung für Π. Eine in Polynomialzeit berechenbare Funktion $f : \mathscr{I} \times \mathbb{N} \to \mathscr{I} \times \mathbb{N}$ heißt* Problemkernreduktion *für (Π, κ), falls f bei Eingabe $(I, \kappa(I))$ eine Ausgabe $(I', \kappa(I'))$ berechnet, sodass die folgenden drei Eigenschaften erfüllt sind:*

Problemkernreduktion

1. *Für alle $I \in \mathscr{I}$ ist $(I, \kappa(I))$ genau dann eine „Ja"-Instanz von Π, wenn $(I', \kappa(I'))$ eine „Ja"-Instanz von Π ist.*
2. *Es gibt eine Funktion $f' : \mathbb{N} \to \mathbb{N}$, sodass $|I'| \leq f'(\kappa(I))$.*
3. *$\kappa(I') \leq \kappa(I)$.*

Die Instanz $(I', \kappa(I'))$ heißt Problemkern *für (Π, κ) und $f'(\kappa(I))$ heißt die* Größe des Problemkerns.

Größe des Problemkerns

Die Zugehörigkeit eines parametrisierten Problems (Π, κ) zur Klasse FPT und die Existenz einer Problemkernreduktion für (Π, κ) sind in gewisser Weise äquivalent.

Satz 6.2. *1. Falls ein parametrisiertes Problem (Π, κ) eine Problemkernreduktion besitzt, so liegt es in der Klasse* FPT.
2. Jedes Problem (Π, κ) aus der Klasse FPT *kann durch eine Problemkernreduktion gelöst werden.*

Beweis.

1. Es sei (Π, κ) ein parametrisiertes Problem. Falls (Π, κ) eine Problemkernreduktion besitzt, so kann man in polynomieller Zeit aus $(I, \kappa(I))$ eine Instanz $(I', \kappa(I'))$ mit $|I'| \leq f'(\kappa(I))$ bestimmen und in dieser mittels erschöpfender Suche alle möglichen Lösungen testen. Somit erhält man einen FPT-Algorithmus für (Π, κ).
2. Es sei (Π, κ) ein parametrisiertes Problem, das durch einen FPT-Algorithmus mit der Laufzeit $f(\kappa(I)) \cdot p(|I|)$ gelöst werden kann. Wir unterscheiden für jede Instanz I die folgenden beiden Fälle:
   - Gilt $f(\kappa(I)) \leq |I|$, so löse die Instanz in der Zeit $f(\kappa(I)) \cdot p(|I|) \leq |I| \cdot p(|I|)$.
   - Gilt $f(\kappa(I)) > |I|$, so haben wir einen Problemkern der Größe höchstens $f(\kappa(I))$ gefunden.

Damit ist der Beweis abgeschlossen. ❑

Wir geben nun zwei Aussagen zur Problemkernreduktion für das parametrisierte Problem p-KNOTENÜBERDECKUNG an, die wir anschließend verwenden, um einen parametrisierten Algorithmus für dieses Problem zu entwerfen.

Lemma 6.3 (Knoten vom Grad 0). *Es seien $G = (V, E)$ ein Graph und $v \in V$ ein Knoten vom Grad* 0*. G hat genau dann eine Knotenüberdeckung der Größe k, wenn $G - v$ eine Knotenüberdeckung der Größe k hat.*

Beweis. Knoten vom Grad 0 sind an keiner Kante beteiligt und müssen deshalb nicht betrachtet werden. ❑

Die folgende Aussage liefert die *Problemkernreduktion von Buss*.

Problemkernreduktion von Buss

Lemma 6.4 (Knoten vom Grad größer als k). *Es seien $G = (V, E)$ ein Graph und $v \in V$ ein Knoten vom Grad größer als k. G hat genau dann eine Knotenüberdeckung der Größe k, wenn $G - v$ eine Knotenüberdeckung der Größe $k - 1$ hat.*

Beweis. Es seien G ein Graph und C eine Knotenüberdeckung für G mit $|C| \leq k$. Es sei v ein Knoten aus G, der $\ell > k$ Nachbarn und damit ℓ inzidente Kanten besitzt. Um alle diese ℓ Kanten zu überdecken, ohne v in die Knotenüberdeckung C aufzunehmen, müsste man alle ℓ Nachbarn von v in die Menge C aufnehmen. Da dies nicht möglich ist, liegt v (und ebenso jeder andere Knoten vom Grad größer als k) in einer Knotenüberdeckung der Größe höchstens k. ❑

In Definition 3.3 wurde der maximale Knotengrad eines Graphen $G = (V,E)$ durch $\Delta(G) = \max_{v \in V} \deg_G(v)$ eingeführt.

Lemma 6.5. *Es sei $G = (V,E)$ ein Graph ohne Knoten vom Grad 0, der eine Knotenüberdeckung der Größe höchstens k besitzt, und es sei $\Delta(G) \leq d$. Dann hat G höchstens $k \cdot (d+1)$ Knoten.*

Beweis. Es seien G ein Graph und C eine Knotenüberdeckung für G mit $|C| \leq k$. Jeder Knoten aus C kann höchstens d Knoten abdecken. Zusammen mit den Knoten aus C hat G somit höchstens $k \cdot d + k = k \cdot (d+1)$ Knoten. ❑

Am Beispiel des $K_{1,n}$ (mit $n+1$ Knoten, dem maximalen Knotengrad $\Delta(K_{1,n}) = d = n$ und der Größe $|C| = k = 1$ einer kleinsten Knotenüberdeckung C) sieht man, dass die angegebene Abschätzung der Knotenanzahl auch erreicht werden kann.

Korollar 6.6. *Das Problem p-*KNOTENÜBERDECKUNG *hat einen Problemkern der Größe höchstens $k(k+1)$, wobei k der Parameter dieses Problems ist.*

Beweis. Sei (G,k) mit $G = (V,E)$ und $k \in \mathbb{N}$ eine Instanz für KNOTENÜBERDECKUNG. Ohne Beschränkung der Allgemeinheit können wir annehmen, dass G keine Knoten vom Grad 0 und keine Knoten vom Grad größer als k enthält; andernfalls modifizieren wir die Instanz gemäß Lemma 6.3 und Lemma 6.4 (Problemkernreduktion).

Gilt $|V| > k(k+1)$, so gibt es nach Lemma 6.5 ($d = k$) keine Knotenüberdeckung der Größe höchstens k in G.

Gilt $|V| \leq k(k+1)$, so existiert ein Problemkern der Größe höchstens $k(k+1)$. ❑

Damit ist p-KNOTENÜBERDECKUNG nach Satz 6.2 in der Klasse FPT. Der folgende parametrisierte Algorithmus für p-KNOTENÜBERDECKUNG nimmt zuerst alle Knoten in die Knotenüberdeckung auf, die aufgrund ihres hohen Knotengrades darin enthalten sein müssen (Problemkernreduktion von Buss), und testet anschließend für alle möglichen Knotenmengen, ob sie den Restgraphen überdecken:

1. Es sei $H = \{v \in V \mid \deg(v) > k\}$.
   Gilt $|H| > k$, so gibt es nach Lemma 6.4 keine Knotenüberdeckung der Größe höchstens k.
   Gilt $|H| \leq k$, so sei $k' = k - |H|$, und es sei G' der Teilgraph von G ohne die Knoten aus H und deren inzidenten Kanten.
2. Entferne alle Knoten vom Grad 0 aus G'.
3. Falls G' mehr als $k'(k+1)$ Knoten enthält (d. h., falls $|V - H| > k'(k+1)$ gilt), gibt es nach Lemma 6.5 wegen $\Delta(G') \leq k$ keine Knotenüberdeckung der Größe höchstens k' in G'.
4. Suche eine Knotenüberdeckung der Größe höchstens k' in G' durch Testen aller k'-elementigen Knotenteilmengen. Besitzt G' eine Knotenüberdeckung der Größe höchstens k', so hat G eine Knotenüberdeckung der Größe höchstens k, und sonst nicht.

Satz 6.7. *Der obige Algorithmus entscheidet für jeden Graphen G mit n Knoten in der Zeit $\mathcal{O}(k \cdot n + k^{2k})$, ob G eine Knotenüberdeckung der Größe höchstens k besitzt.*

Beweis. Zunächst sei bemerkt, dass der obige Algorithmus das Problem p-KNOTENÜBERDECKUNG löst. Falls nämlich der Algorithmus die Eingabe in Schritt 1 verwirft, da $|H| > k$ gilt, ist dies nach Lemma 6.4 sicher richtig. Gilt dagegen $|H| \leq k$, so nimmt der Algorithmus alle Knoten vom Grad mindestens k in die Knotenüberdeckung auf, da diese Knoten nach Lemma 6.4 in jeder Knotenüberdeckung enthalten sein müssen. Damit kann die Frage, ob G eine Knotenüberdeckung der Größe höchstens k besitzt, auf die äquivalente Frage reduziert werden, ob der Graph G' eine Knotenüberdeckung der Größe höchstens $k' = k - |H|$ besitzt. Der Graph G' hat nach Konstruktion einen maximalen Knotengrad von k und keinen Knoten vom Grad 0, denn alle Knoten vom Grad 0 oder größer als k wurden ja aus G entfernt. Aus Lemma 6.5 folgt nun, dass der Algorithmus in Schritt 3 korrekterweise die Eingabe verwirft, falls G' zu viele Knoten (also mehr als $k'(k+1)$ Knoten) hat, um eine Knotenüberdeckung der Größe k' besitzen zu können. In Schritt 4 schließlich testet der Algorithmus alle möglichen Knotenteilmengen darauf, ob diese eine Knotenüberdeckung der Größe höchstens k' für G' bilden, und akzeptiert entsprechend.

Nun betrachten wir die Laufzeit des Algorithmus. Die Schritte 1, 2 und 3 können für Graphen G mit n Knoten jeweils in der Zeit $\mathcal{O}(k \cdot n)$ realisiert werden. Das Testen aller k'-elementigen Teilmengen, ob diese eine Knotenüberdeckung in einem Graph mit höchstens $k'(k+1) \leq k(k+1)$ Knoten bilden, benötigt die Zeit

$$\mathcal{O}\left(k \cdot k' \cdot (k+1) \cdot \binom{k' \cdot (k+1)}{k'}\right), \tag{6.1}$$

da die Knotenmenge des Graphen G' höchstens $\binom{k' \cdot (k+1)}{k'}$ verschiedene Teilmengen der Größe k' hat und G' höchstens $k \cdot k' \cdot (k+1)$ Kanten besitzt. Die Zeitschranke aus (6.1) liegt aber in

$$\mathcal{O}\left(k^3 \cdot \binom{k \cdot (k+1)}{k}\right) \subseteq \mathcal{O}(k^{2k}), \tag{6.2}$$

wie man sich leicht überlegt. Somit ist die Laufzeit des Algorithmus in $\mathcal{O}(k \cdot n + k^{2k})$. ❑

Übung 6.8. (a) Zeigen Sie die Inklusion (6.2):

$$\mathcal{O}\left(k^3 \cdot \binom{k \cdot (k+1)}{k}\right) \subseteq \mathcal{O}(k^{2k}).$$

(b) Wenden Sie die Problemkernreduktion von Buss für jedes $k \in \{1,2,3,4\}$ auf die folgenden Graphen an. Wie sieht jeweils der Problemkern aus?

1. Der Gittergraph $G_{10,5}$.
2. Der vollständige Graph K_{20}.
3. Das Rad R_{10}. (Ein Rad R_n besteht aus einem Kreis C_n, in den ein Knoten eingefügt wird, der mit allen n Kreisknoten verbunden wird.)

Mit der folgenden Beobachtung kann die Kerngröße etwas verringert werden.

Lemma 6.9 (Knoten vom Grad 1). *Es seien $G = (V,E)$ ein Graph und $v \in V$ ein Knoten vom Grad 1 und mit der Nachbarschaft $N(v) = \{u\}$. G hat genau dann eine Knotenüberdeckung der Größe k, wenn $G - \{u,v\}$ eine Knotenüberdeckung der Größe $k-1$ hat.*

Beweis. Die Kante zwischen v und seinem Nachbarn u kann nur durch v oder u abgedeckt werden. Da u möglicherweise noch weitere Kanten abdeckt, fügt man u zur Knotenüberdeckung hinzu. ❑

Satz 6.10. *Das Problem p-KNOTENÜBERDECKUNG hat einen Problemkern der Größe höchstens k^2, wobei k der Parameter dieses Problems ist.*

Beweis. Sei (G,k) mit $G = (V,E)$ und $k \in \mathbb{N}$ eine Instanz für p-KNOTENÜBERDECKUNG. Ohne Beschränkung der Allgemeinheit können wir annehmen, dass G keine Knoten vom Grad 0, keine Knoten vom Grad 1 und keine Knoten vom Grad größer als k enthält; andernfalls modifizieren wir die Instanz gemäß Lemma 6.3, Lemma 6.4 und Lemma 6.9 (Problemkernreduktion).

Jeder Knoten einer Knotenüberdeckung der Größe höchstens k kann höchstens k Kanten abdecken. Somit gibt es im Graphen noch höchstens k^2 Kanten. Da es keine Knoten vom Grad 0 oder 1 gibt, folgt, dass es im Graphen auch nur höchstens k^2 Knoten gibt. ❑

Mit Hilfe von Matchings kann man sogar einen Problemkern linearer Größe nachweisen.

Satz 6.11. *Das Problem p-KNOTENÜBERDECKUNG hat einen Problemkern der Größe höchstens $2k$, wobei k der Parameter dieses Problems ist.* **ohne Beweis**

Man vermutet, dass man den Faktor von 2 hier nicht weiter verbessern kann, da dies auch einen bislang nicht erzielten Durchbruch in der Approximierbarkeit des Optimierungsproblems MIN KNOTENÜBERDECKUNG liefern würde.

Da die in Satz 6.7 angegebene Laufzeit schon für kleine Werte von k recht groß wird, geben wir in Abschnitt 6.1.2 einen weiteren FPT-Algorithmus für p-KNOTENÜBERDECKUNG an.

Übung 6.12. Geben Sie einen Problemkern-Algorithmus für das Problem p-UNABHÄNGIGE MENGE auf planaren Graphen[33] an. Welche Größe hat der Problemkern?
Hinweis: In jedem planaren Graphen gibt es einen Knoten vom Grad höchstens 5.

[33] Ein Graph G heißt *planar*, falls G kreuzungsfrei in die Ebene einbettbar ist. Das heißt, falls man die Knoten von G so in $\mathbb{R}^2$ einbetten kann, dass die Kanten ohne Überschneidung zwischen den Knoten geführt werden können.

6.1.2 Verbesserter Suchbaum mit beschränkter Höhe

In diesem Abschnitt geben wir einen weiteren FPT-Algorithmus für das parametrisierte Problem p-KNOTENÜBERDECKUNG an. Zunächst sei an Satz 5.48 in Abschnitt 5.2.1 erinnert. Im Beweis dieses Satzes wurde ein $\mathcal{O}(2^k \cdot n)$-Algorithmus für dieses Problem angegeben, dessen Grundidee in einer Teile-und-herrsche-Strategie besteht, die einen Suchbaum mit beschränkter Höhe liefert, der die folgende Verzweigungsregel verwendet:

V0 Wähle eine Kante $\{v_1, v_2\}$ und füge entweder den Knoten v_1 oder den Knoten v_2 zur bisher konstruierten Knotenüberdeckung hinzu.

Verwendet man anstelle der Verzweigungregel V0 (Auswahl einer Kante) die folgenden drei Verzweigungsregeln, so erhält man einen wesentlich besseren Suchbaum mit beschränkter Höhe.

V1 Falls es einen Knoten v vom Grad 1 gibt, so wird dessen Nachbar $u \in N(v)$ in die Knotenüberdeckung aufgenommen.
V2 Falls es einen Knoten v vom Grad 2 gibt, so sind entweder beide Nachbarn von v oder es sind v und alle Nachbarn der zwei Nachbarn von v in einer optimalen Knotenüberdeckung.
V3 Falls es einen Knoten v vom Grad mindestens 3 gibt, so ist entweder v oder es sind alle Nachbarn von v in einer optimalen Knotenüberdeckung.

In einer Verzweigung nach (V1) hat der zugehörige Suchbaumknoten nur ein Kind und in einer Verzweigung nach (V2) und (V3) verzweigt der Suchbaum in zwei neue Teilinstanzen.

Satz 6.13 (Niedermeier [Nie06]). *Der Suchbaum zu den Verzweigungsregeln V1, V2 und V3 hat für den Parameter k des Problems p-*KNOTENÜBERDECKUNG *eine Größe von $\mathcal{O}(1.47^k)$.* **ohne Beweis**

Durch weitere trickreiche Fallunterscheidungen erzielte man weitere verbesserte FPT-Algorithmen für p-KNOTENÜBERDECKUNG. Der zur Zeit beste Algorithmus stammt von Chen, Kanj und Xia.

Satz 6.14 (Chen, Kanj und Xia [CKX06]). *Das Problem p-*KNOTENÜBERDECKUNG *kann für Instanzen (G,k), wobei G ein Graph mit n Knoten und $k \in \mathbb{N}$ ist, in der Zeit $\mathcal{O}(kn + 1,2738^k)$ gelöst werden.* **ohne Beweis**

6.2 Hitting Set für Mengen der Größe drei

Das folgende Problem verallgemeinert das Problem KNOTENÜBERDECKUNG. Da es keine gebräuchliche Übersetzung ins Deutsche gibt, bleiben wir bei der englischen Bezeichnung für das Problem.

3-HITTING SET (3-HS)	
Gegeben:	Eine Menge S, eine Menge $\mathscr{C}$ von drei-elementigen Teilmengen der Menge S und eine Zahl $k \in \mathbb{N}$.
Frage:	Gibt es eine Teilmenge $S' \subseteq S$, $\lvert S' \rvert \leq k$, sodass S' mindestens ein Element aus jeder der Mengen in $\mathscr{C}$ enthält?

3-HITTING SET
3-HS

Eine solche Menge $S' \subseteq S$ mit $|S'| \leq k$ und $S' \cap C \neq \emptyset$ für alle $C \in \mathscr{C}$ heißt – dem englischen Sprachgebrauch folgend – „*Hitting Set der Größe höchstens k für* $\mathscr{C}$“.

2-HITTING SET
2-HS

Das analog definierte Problem (2-HS) 2-HITTING SET, in dem die Familie $\mathscr{C}$ zwei-elementige anstelle von drei-elementigen Mengen enthält, entspricht genau dem Problem KNOTENÜBERDECKUNG. Das Problem 3-HS kann als eine Verallgemeinerung von KNOTENÜBERDECKUNG auf Hypergraphen mit „Dreierkanten“ angesehen werden.

Beispiel 6.15. Es sei eine Instanz $I = (S, \mathscr{C}, k)$ von 3-HITTING SET durch

$$S = \{1,2,3,4,5,6,7,8,9\},$$
$$\mathscr{C} = \{\{1,2,3\},\{1,4,7\},\{2,4,5\},\{2,6,9\},\{3,5,7\},\{5,6,8\}\} \text{ und}$$
$$k = 3$$

gegeben. Es ist nicht schwer zu sehen, dass die Teilmenge $S' = \{3,4,6\}$ von S eine Lösung dieser Instanz ist.

Satz 6.16. 3-HITTING SET *ist* NP*-vollständig.* **ohne Beweis**

p-3-HITTING SET
p-3-HS

Das entsprechende parametrisierte Problem hat die folgende Gestalt:

p-3-HITTING SET (p-3-HS)	
Gegeben:	Eine Menge S, eine Menge $\mathscr{C}$ von drei-elementigen Teilmengen der Menge S und eine Zahl $k \in \mathbb{N}$.
Parameter:	k.
Frage:	Gibt es eine Teilmenge $S' \subseteq S$, $\lvert S' \rvert \leq k$, sodass S' mindestens ein Element aus jeder der Mengen in $\mathscr{C}$ enthält?

Wir schätzen nun die Größe des Problemkerns für p-3-HITTING SET ab, indem wir zeigen, dass die Menge $\mathscr{C}$ in der Eingabe des Problems höchstens k^3 drei-elementige Teilmengen der Menge S enthält.

Lemma 6.17. *Das Problem* p-3-HITTING SET *hat einen Problemkern der Größe höchstens* k^3*, wobei* k *der Parameter dieses Problems ist.*

Beweis. Für zwei feste $x, y \in S$ gibt es höchstens k Mengen $\{x, y, z'\}$ für ein beliebiges $z' \in S$. Denn gäbe es mehr als k Möglichkeiten, das Element z' zu wählen, so müsste wegen der Bedingung $|S'| \leq k$ (für eine Lösung S' des Problems) stets x oder y in S' enthalten sein, und damit könnten alle Mengen $\{x, y, z'\}$ aus $\mathscr{C}$ entfernt werden.

Weiterhin gibt es für ein festes $x \in S$ höchstens k^2 Mengen $\{x, y', z'\}$ für beliebige $y', z' \in S$, da wir nach der ersten Beobachtung wissen, dass x zusammen mit einem Element y nur in k Mengen vorkommen kann. Gäbe es nun mehr als k^2 Mengen, die x enthalten, so müsste wegen $|S'| \leq k$ stets x in S' liegen, und damit könnten alle Mengen $\{x, y', z'\}$ aus $\mathscr{C}$ entfernt werden.

Aus der letzten Beobachtung folgt, dass jedes Element x in k^2 drei-elementigen Teilmengen auftreten kann. Daraus folgt, wieder wegen $|S'| \leq k$, dass $\mathscr{C}$ aus höchstens $k \cdot k^2 = k^3$ drei-elementigen Teilmengen bestehen kann. ❑

Die Idee unseres parametrisierten Algorithmus für p-3-HITTING SET ist eine Problemkernreduktion der Instanz, die dann in einem höhenbeschränkten Suchbaum entschieden wird. Der Algorithmus basiert auf der folgenden einfachen Beobachtung (die sich unmittelbar aus der Definition des Problems ergibt): Ist $I = (S, \mathscr{C}, k)$ eine Instanz von p-3-HITTING SET, so muss mindestens ein Element einer jeden drei-elementigen Teilmenge aus $\mathscr{C}$ in der gesuchten Lösung liegen. Demgemäß nutzt unser Suchbaum die folgende Verzweigungsregel:

V Wähle eine drei-elementige Teilmenge $\{x, y, z\}$ aus $\mathscr{C}$ und füge entweder x, y oder z in die Lösung ein.

Der Algorithmus arbeitet bei Eingabe einer Instanz $I = (S, \mathscr{C}, k)$ von p-3-HITTING SET so:

1. Führe gemäß Lemma 6.17 eine Problemkernreduktion durch, sodass $|\mathscr{C}| \leq k^3$ gilt.
2. Konstruiere einen vollständigen trinären Wurzelbaum T der Höhe k, d. h., jeder innere Knoten von T hat drei Kinder.
3. Markiere den Wurzelknoten in T mit $(\mathscr{C}, k)$.
4. Die vom Wurzelknoten verschiedenen Knoten in T werden in Abhängigkeit von der Markierung ihrer Vorgängerknoten wie folgt markiert. Es sei u ein mit $(\mathscr{C}', k')$ markierter innerer Baumknoten, dessen Kinder noch unmarkiert sind. Für $x \in \{a, b, c\}$ bezeichne $\mathscr{C}'^{-x}$ die Menge $\mathscr{C}'$ ohne die Mengen, die x enthalten. Wähle eine beliebige drei-elementige Teilmenge $\{a, b, c\}$ aus $\mathscr{C}'$ und
   a) markiere ein Kind von u mit $(\mathscr{C}'^{-a}, k-1)$;
   b) markiere ein anderes Kind von u mit $(\mathscr{C}'^{-b}, k-1)$;
   c) markiere das verbleibende dritte Kind von u mit $(\mathscr{C}'^{-c}, k-1)$.
5. Falls es einen Baumknoten mit der Markierung $(\emptyset, k')$ gibt, dann besitzt $\mathscr{C}$ ein *Hitting Set* der Größe höchstens k; andernfalls gibt es für $\mathscr{C}$ kein *Hitting Set* der Größe höchstens k.

Satz 6.18. *Das Problem p-3-HITTING SET kann für Instanzen $I = (S, \mathscr{C}, k)$, wobei n die Größe von I und $k \in \mathbb{N}$ der Parameter ist, in der Zeit $\mathcal{O}(n + 3^k \cdot k^3)$ gelöst werden.*

Beweis. Der im Algorithmus aufgebaute Suchbaum hat eine Größe von 3^k. An jedem Baumknoten sind Operationen auszuführen, die jeweils die Zeit $\mathcal{O}(|\mathscr{C}|)$ benötigen, also nach Lemma 6.17 die Zeit $\mathcal{O}(k^3)$. Die Konstruktion des Problemkerns gemäß diesem Lemma kann in der Zeit $\mathcal{O}(n)$ ausgeführt werden. ❑

Übung 6.19. Zeichnen Sie den vollständigen Suchbaum für die Instanz von 3-HS aus Beispiel 6.15.

Korollar 6.20. p-3-HITTING SET $\in$ FPT.

Anmerkung 6.21 (Entwurfstechniken). Wir haben bisher die folgenden Techniken zum Entwurf von parametrisierten Algorithmen kennen gelernt:

- in Abschnitt 6.1.1 eine Problemkernreduktion,
- in Abschnitt 6.1.2 einen Algorithmus, der einen höhenbeschränkten Suchbaum verwendet, und
- in Abschnitt 6.2 eine Problemkernreduktion, gefolgt von einem Algorithmus, der einen höhenbeschränkten Suchbaum verwendet.

Eine weitere nützliche Methode besteht in einer Verflechtung von Problemkernreduktion und Suchbaumtechnik:

- Führe an jedem mit (I',k') markierten Suchbaumknoten mit Nachfolgerknoten zu den Instanzen $(I'_1,k''),\ldots,(I'_d,k'')$ eine Problemkernreduktion durch, falls dies möglich ist.

```
1:  if (|I'| > c · κ(I))
2:      then führe eine Problemkernreduktion für (I',k') durch
3:  verzweige von (I',k') in die Instanzen (I'_1,k''),...,(I'_d,k'')
```

6.3 Graphmodifikation

Zur Definition des Graphmodifikationsproblems benötigen wir einige Begriffe. Wir wollen uns hier auf solche Grapheigenschaften beschränken, die von der Anordnung bzw. Bezeichnung der Knoten im Graphen unabhängig sind. Das heißt, die durch solche Grapheigenschaften induzierten Graphklassen sind unter Graphisomorphie (siehe (3.1) und (3.2) in Abschnitt 3.1) abgeschlossen: Sind zwei Graphen isomorph, so haben sie entweder beide eine Eigenschaft oder sie haben sie beide nicht. Wir identifizieren dabei eine Grapheigenschaft mit der durch sie induzierten (und unter Isomorphie abgeschlossenen) Graphklasse.

Grapheigenschaft

Definition 6.22 (Grapheigenschaft). *Eine* Grapheigenschaft *ist eine Graphklasse, die unter Graphisomorphie abgeschlossen ist.*

Beispiel 6.23 (Grapheigenschaft). Die Klasse der

- zusammenhängenden Graphen,
- Bäume,

- Graphen mit einer Clique der Größe 77,
- planaren Graphen,
- regulären Graphen

sind Beispiele für Grapheigenschaften. Endliche Mengen von Graphen können hingegen nie eine Grapheigenschaft bilden.

Definition 6.24 (vererbbare Grapheigenschaft). *Eine Grapheigenschaft $\mathscr{E}$ ist eine* vererbbare Grapheigenschaft*, falls gilt: Ist $G \in \mathscr{E}$ und ist H ein induzierter Teilgraph von G, so ist auch $H \in \mathscr{E}$.*

vererbbare Grapheigenschaft

Anders gesagt, vererbbare Grapheigenschaften sind solche Grapheigenschaften, die unter der Bildung von induzierten Teilgraphen abgeschlossen sind.

Beispiel 6.25 (vererbbare Grapheigenschaft). Die Klasse der

- bipartiten Graphen,
- vollständigen Graphen,
- planaren Graphen,
- Wälder und
- Co-Graphen

sind Beispiele für vererbbare Grapheigenschaften.

Dagegen sind die Klasse der Bäume und die Klasse aller Graphen mit mindestens 400 Knoten Beispiele für nicht vererbbare Grapheigenschaften.

Definition 6.26 (Charakterisierung durch Ausschlussmengen). *Eine Grapheigenschaft $\mathscr{E}$ besitzt eine* Charakterisierung durch Ausschlussmengen*, wenn es eine Graphklasse $\mathscr{F}$ gibt, sodass ein Graph G genau dann zu $\mathscr{E}$ gehört, wenn G keinen Graphen aus $\mathscr{F}$ als induzierten Teilgraphen enthält.*

Charakterisierung durch Ausschlussmengen

Die Elemente von $\mathscr{F}$ bezeichnet man auch als verbotene induzierte Teilgraphen*.*

verbotener induzierter Teilgraph

Charakterisierungen durch Ausschlussmengen gibt es für jede vererbbare Grapheigenschaft und für keine nicht vererbbare Grapheigenschaft.

Lemma 6.27. *Eine Grapheigenschaft $\mathscr{E}$ ist genau dann vererbbar, wenn sie eine Charakterisierung durch Ausschlussmengen besitzt.*

Beweis. Es sei $\mathscr{E}$ eine vererbbare Grapheigenschaft. Wähle als Ausschlussmenge $\mathscr{F}$ die Menge aller Graphen, die nicht zu $\mathscr{E}$ gehören.

Es sei $\mathscr{E}$ eine Grapheigenschaft mit Charakterisierung durch eine Ausschlussmenge $\mathscr{F}$. Weiter seien $G \in \mathscr{E}$ und H ein induzierter Teilgraph von G. Da G keinen Graphen aus $\mathscr{F}$ als induzierten Teilgraphen enthält, gilt dies auch für H, und somit liegt auch H in $\mathscr{E}$. ❑

Interessant sind jedoch im Folgenden nur die Charakterisierungen durch endliche Ausschlussmengen.

Charakterisierung durch endliche Ausschlussmengen

Definition 6.28 (Charakterisierung durch endliche Ausschlussmengen). *Eine Grapheigenschaft $\mathscr{E}$ besitzt eine* Charakterisierung durch endliche Ausschlussmengen*, wenn sie eine Charakterisierung $\mathscr{F}$ durch Ausschlussmengen besitzt und $\mathscr{F}$ nur endlich viele paarweise nicht isomorphe Graphen enthält.*

Beispiel 6.29 (Charakterisierung durch endliche Ausschlussmengen). Ein Beispiel für eine Grapheigenschaft $\mathscr{E}$, die eine Charakterisierung durch endliche Ausschlussmengen besitzt, ist die Klasse der Co-Graphen, da hier die Ausschlussmenge $\mathscr{F} = \{P_4\}$ gewählt werden kann.
Weiterhin lassen sich

- die Klasse der trivialerweise perfekten Graphen (englisch: *trivially perfect graphs*) durch die endliche Ausschlussmenge $\mathscr{F} = \{C_4, P_4\}$ und
- die Klasse der Schwellenwert-Graphen (englisch: *threshold graphs*) durch die endliche Ausschlussmenge $\mathscr{F} = \{C_4, P_4, 2K_2\}$ charakterisieren.

Offensichtlich sind diese beiden Graphklassen Teilklassen der Klasse der Co-Graphen.

Anmerkung 6.30. Es besitzt jedoch nicht jede vererbbare Grapheigenschaft $\mathscr{E}$ eine Charakterisierung durch endliche Ausschlussmengen.

$\mathscr{E}$-GRAPHMODIFIKATION

Das Graphmodifikationsproblem ist wie folgt definiert:

$\mathscr{E}$-GRAPHMODIFIKATION	
Gegeben:	Ein Graph $G = (V,E)$ und Zahlen $i, j, k \in \mathbb{N}$.
Frage:	Kann man durch Löschen von bis zu i Knoten und von bis zu j Kanten aus G und durch Hinzufügen von bis zu k Kanten zu G einen Graphen aus $\mathscr{E}$ erhalten?

Satz 6.31. $\mathscr{E}$-GRAPHMODIFIKATION *ist* NP*-hart für jede nicht triviale*[34] *vererbbare Grapheigenschaft $\mathscr{E}$.* **ohne Beweis**

Anmerkung 6.32. 1. Offenbar ist $\mathscr{E}$-GRAPHMODIFIKATION sogar NP-vollständig, sofern die nicht trivial vererbbare Grapheigenschaft $\mathscr{E}$ in Polynomialzeit getestet werden kann.
2. Bereits die folgenden Spezialfälle des Problems $\mathscr{E}$-GRAPHMODIFIKATION sind NP-hart:
   - **Kantenlöschungsproblem**: Kann man durch Löschen von bis zu k Kanten aus G einen Graphen aus $\mathscr{E}$ erhalten?
   - **Knotenlöschungsproblem**: Kann man durch Löschen von bis zu k Knoten (und deren inzidenten Kanten) aus G einen Graphen aus $\mathscr{E}$ erhalten?

[34] Zur Erinnerung: Eine Grapheigenschaft $\mathscr{E}$ ist *nicht trivial*, falls es unendlich viele Graphen gibt, die in $\mathscr{E}$ liegen, und es unendlich viele Graphen gibt, die nicht in $\mathscr{E}$ liegen.

- **Knoten-/Kantenlöschungsproblem**: Kann man durch Löschen von bis zu k Knoten (und deren inzidenten Kanten) und von bis zu ℓ Kanten aus G einen Graphen aus $\mathscr{E}$ erhalten?
- **Kantenadditionsproblem**: Kann man durch Hinzufügen von bis zu k Kanten zu G einen Graphen aus $\mathscr{E}$ erhalten?

Wir betrachten nun die folgende parametrisierte Version des Graphmodifikationsproblems.

p-$\mathscr{E}$-GRAPHMODIFIKATION

p-$\mathscr{E}$-GRAPHMODIFIKATION	
Gegeben:	Ein Graph $G=(V,E)$ und Zahlen $i,j,k \in \mathbb{N}$.
Parameter:	$i+j+k$.
Frage:	Kann man durch Löschen von bis zu i Knoten und von bis zu j Kanten aus G und durch Hinzufügen von bis zu k Kanten zu G einen Graphen aus $\mathscr{E}$ erhalten?

Um mittels einer Problemkernreduktion zu beweisen, dass p-$\mathscr{E}$-GRAPHMODIFIKATION fest-Parameter-berechenbar ist, geben wir zunächst eine Hilfsaussage an.

Lemma 6.33. *Es sei $\mathscr{E}$ eine vererbbare Grapheigenschaft, die für einen Graphen $G=(V,E)$ in der Zeit $t(G)$ überprüfbar ist. Dann kann für jeden Graphen $G \notin \mathscr{E}$ in der Zeit $\mathscr{O}(|V| \cdot t(G))$ ein minimaler verbotener induzierter Teilgraph gefunden werden.*

Beweis. Es seien $G=(V,E)$ ein Graph und $V=\{v_1,\ldots,v_n\}$. Die folgende einfache Schleife bestimmt einen minimalen verbotenen induzierten Teilgraphen H.

```
1:  H = G;
2:  for i = 1 to n do
3:      if H − v_i ∉ E then H = H − v_i
4:  return H
```

Für die Korrektheit ist zu beachten, dass $\mathscr{E}$ eine vererbbare Grapheigenschaft ist. ❑

Satz 6.34 (Cai [Cai96]). *Es sei $\mathscr{E}$ eine Grapheigenschaft mit einer Charakterisierung durch endliche Ausschlussmengen. Dann ist das $\mathscr{E}$-Graphmodifikationsproblem für Graphen mit n Knoten in der Zeit $\mathscr{O}(N^{i+2j+2k} \cdot n^{N+1})$ lösbar. Hierbei ist N die maximale Knotenanzahl der Graphen in der Ausschlussmenge.*

Beweis. Offensichtlich ist die Zahl N eine durch das Problem gegebene Konstante. Wir geben nun einen FPT-Algorithmus für p-$\mathscr{E}$-GRAPHMODIFIKATION an. Durch eine Reduktion auf den Problemkern muss die Modifikation nicht auf den gesamten Graphen ausgeübt werden, sondern nur auf einen minimalen induzierten Teilgraphen H in G.

Es sei $G=(V,E)$ ein Graph mit n Knoten. Die folgende Methode entscheidet, ob das Problem p-$\mathscr{E}$-GRAPHMODIFIKATION für den Eingabegraphen G lösbar ist.

```
GraphMod(G = (V,E), i, j, k){
 1:  if G ∈ 𝒢
 2:      return true;
 3:  H := (V_H, E_H) minimaler verbotener induzierter Teilgraph von 𝒢 in G
 4:  if i > 0 then
 5:      for all v ∈ V_H do
 6:          if(GraphMod((V − v, E), i − 1, j, k)) then
 7:              return true;
 8:  if j > 0 then
 9:      for all e = {v, w} ∈ E_H do
10:          if (GraphMod((V, E − e), i, j − 1, k)) then
11:              return true;
12:  if k > 0 then
13:      for all (v, w) ∈ V_H × V_H, {v, w} ∉ E_H do
14:          if (GraphMod((V, E ∪ {{v, w}}), i, j, k − 1)) then
15:              return true;
16:  return false; }
```

Die Laufzeit der Methode ergibt sich wie folgt. Nach Lemma 6.33 kann H in der Zeit $\mathcal{O}(n \cdot n^N)$ gefunden werden. Es gibt N Möglichkeiten, einen Knoten aus H zu löschen, $\binom{N}{2}$ Möglichkeiten, eine Kante aus H zu löschen, und $\binom{N}{2}$ Möglichkeiten, eine Kante in H einzufügen. Somit werden höchstens $\binom{N}{2}^{j+k} \cdot N^i \in \mathcal{O}(N^{i+2j+2k})$ Graphen mit der obigen Methode erzeugt. Ob G zu $\mathscr{E}$ gehört, kann in der Zeit $\mathcal{O}(n^N)$ getestet werden: Für jeden Graphen H aus $\mathscr{F}$ müssen alle $\binom{n}{|H|} \leq n^N$ Teilgraphen aus G der Größe $|H|$ untersucht werden. Damit folgt die Laufzeit $\mathcal{O}(|\mathscr{F}| \cdot n^N) \in \mathcal{O}(n^N)$.

Die totale Laufzeit ergibt sich somit als $\mathcal{O}(N^{i+2j+2k} \cdot n^{N+1})$. ❑

Korollar 6.35. p-$\mathscr{E}$-GRAPHMODIFIKATION $\in$ FPT.

Übung 6.36. Wenden Sie den Algorithmus für p-$\mathscr{E}$-GRAPHMODIFIKATION im Beweis von Satz 6.34 mit der Grapheigenschaft $\mathscr{E} = \{G \mid G \text{ ist ein Co-Graph}\}$ auf die folgenden Gittergraph-Instanzen an:

1. $G = G_{2,3}$ und $i = 1$, $j = 0$, $k = 0$,
2. $G = G_{2,3}$ und $i = 1$, $j = 1$, $k = 0$ und
3. $G = G_{2,4}$ und $i = 1$, $j = 1$, $k = 1$

Unmittelbar aus der Laufzeitabschätzung in Satz 6.34 ergibt sich das folgende Korollar.

Korollar 6.37. *Es sei $\mathscr{E} = \{G \mid G \text{ ist ein Co-Graph}\}$. Dann ist das Problem p-$\mathscr{E}$-*GRAPHMODIFIKATION *für Graphen mit n Knoten und den Parameter $\ell = i + j + k$ (wie in der Problemdefinition) in der Zeit $\mathcal{O}(4^{i+2j+2k} \cdot n^5)$ lösbar.*

Anmerkung 6.38. 1. Bei der angegebenen Lösung kann auf die Endlichkeit der Ausschlussmenge nicht verzichtet werden.
2. Es gibt auch Graphmodifikationsprobleme, die nicht durch endliche Ausschlussmengen charakterisiert werden können, aber dennoch fest-Parameter-berechenbar sind, z. B. p-CHORDAL GRAPH COMPLETITION (auch als p-MINIMUM FILL IN bekannt):

p-CHORDAL GRAPH COMPLETITION
p-MINIMUM FILL IN

	p-CHORDAL GRAPH COMPLETITION
Gegeben:	Ein Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$.
Parameter:	k.
Frage:	Kann man durch Hinzufügen von höchstens k Kanten einen chordalen Graphen erhalten (d. h. einen Graphen, in dem jeder Kreis mit mindestens vier Knoten eine Sehne/Abkürzung enthält)?

6.4 Parameterwahl

Die Fest-Parameter-Berechenbarkeit eines Problems Π bezüglich einer Parametrisierung κ impliziert leider noch nicht unmittelbar die praktische Lösbarkeit des Problems. Eine Laufzeit von

$$2^{2^{2^{2^{2^{\kappa(I)}}}}} \cdot |I|^2$$

ist zwar eine zulässige Laufzeit eines FPT-Algorithmus, praktisch jedoch auch für kleine Werte des Parameters nicht brauchbar.

Weiterhin kann es sein, dass der gewählte Parameter κ groß im Vergleich zur Instanzengröße ist. Dies ist zum Beispiel im Problem p-SAT durch den Parameter „Anzahl der Variablen“ der Fall.

Aus diesen Gründen sollte man stets versuchen, die folgenden beiden Bedingungen bei der Wahl des Parameters κ zu erfüllen:

1. Die Parametrisierung κ sollte es ermöglichen (zumindest für kleine Werte des Parameters), einen praktisch brauchbaren parametrisierten Algorithmus zu entwerfen.
2. Die in der Praxis typischerweise auftretenden Probleminstanzen I sollten einen nicht zu großen Parameterwert $\kappa(I)$ besitzen.

6.5 Offene Probleme

Für die folgenden Probleme ist es bislang offen, ob sie zur Klasse FPT gehören oder ob sie W$[t]$-schwer sind:

p-$K_{t,t}$-TEILGRAPH	
Gegeben:	Ein Graph $G = (V,E)$ und eine Zahl $t \in \mathbb{N}$.
Parameter:	t.
Frage:	Enthält G den vollständig bipartiten Graphen $K_{t,t}$ als einen Teilgraphen?

p-$K_{t,t}$-TEILGRAPH

p-GERADE MENGE	
Gegeben:	Ein Graph $G = (V,E)$ und eine Zahl $k \in \mathbb{N}$.
Parameter:	k.
Frage:	Enthält G eine Menge S von Knoten mit $1 \leq \lvert S\rvert \leq k$, sodass für jeden Knoten u in G die Zahl $\lvert N(u) \cap S\rvert$ gerade ist?

p-GERADE MENGE

6.6 Literaturhinweise

Bereits im letzten Kapitel wurden die Bücher von Downey und Fellows [DF99], Flum und Grohe [FG06] sowie Niedermeier [Nie06] zur Fest-Parameter-Algorithmik und parametrisierten Komplexität erwähnt, in denen man zahlreiche weitere parametrisierte Algorithmen findet.

In zwei Ausgaben der Zeitschrift *Computer Journal* aus dem Jahre 2008 werden weitere interessante Forschungsergebnisse über FPT-Algorithmen zusammengefasst. Einige dieser Artikel werden im Folgenden aufgezählt. Slopper und Telle [ST08] geben einen Überblick über Entwurfstechniken für parametrisierte Algorithmen für Probleme wie z. B. p-KNOTENÜBERDECKUNG und p-DOMINIERENDE MENGE an. Mit schweren paramatrisierten Problemen beschäftigen sich Chen und Meng [CM08]. Marx [Mar08] betrachtet Zusammenhänge zwischen parametrisierten Algorithmen und Approximationsalgorithmen und Cai [Cai08] untersucht die parametrisierte Komplexität von Optimierungsproblemen. Gramm, Nickelsen und Tantau [GNT08] stellen den Einsatz von parametrisierten Algorithmen in der Bioinformatik vor. Weiterhin betrachten Gutin und Yeo [GY08] zahlreiche Ergebnisse über parametrisierte Algorithmen für gerichtete Graphen.