Parameterisierte Algorithmen

WS 2011/12 in Trier

Henning Fernau
fernau@uni-trier.de

Parameterisierte Algorithmen Gesamtübersicht

- Einführung
- Grundbegriffe
- Problemkerne
- Suchbäume
- Graphparameter
- Weitere Methoden
- Komplexitätstheorie—parameterisiert

Themen für heute und nächste Woche:

- Win-Win
- iteratives Verbessern
- Farbkodierungen
- Wohl-Quasi-Ordnungen

Win-Win

Unter Win-Win wollen wir einen Algorithmus verstehen, der Fallunterscheidungen vornimmt; jeder der entstehenden Fälle ist auf seine eigene Art “gut”.

Die Trade-off-Berechnungen aus der vorigen Vorlesung kann man als ein Beispiel ansehen:

- Entweder hat unser planarer Graph “wenige Schichten” (und damit beschränkte Baumweite)
- oder wir finden einen “kleinen Schicht-Separator”.

Diese Strategie funktioniert oft in der parameterisierten Algorithmik.

Win-Win mit Baumweiten

Satz 1 *Gegeben seien ein Graph G und eine Zahl k.*
Dann lässt sich in Polynomzeit eines der folgenden finden:

1. *Ein Kreis der Länge mindestens k.*

2. *Eine Baumzerlegung mit Baumweite höchstens k.*

Beweisidee: $k + 1$ *Polizisten gehen im Gänsemarsch durch einen "Tiefensuchbaum" von G: entweder finden Sie besagten Kreis, oder aber sie bezeugen eine Baumzerlegung (entlang des Suchbaums).*

Mehr zum Gänsemarsch durch G.

- Auf jedem Graphknoten wird eine Glühbirne installiert.
- Wenn (irgendwann) jeder Polizist auf einem Knoten steht und der Gänsemarsch weitergeht, knipst der letzte Polizist beim Betreten eines (für ihn) neuen Knotens die Birne an.
- Der erste Polizist leitet den Gänsemarsch:
  Er sucht sich wenn möglich einen noch unbesuchten Nachbarknoten seines Knotens aus und betritt ihn; die anderen Polizisten "rücken nach".
  Gibt es keinen unbesuchten Nachbarknoten, geht der erste Polizist zurück zum zweiten Polizisten (und knipst beim Weggehen sein Licht an) und sucht dort weiter.
  Beachte: Ein Knoten ist besucht, wenn seine Birne brennt oder ein Polizist darauf steht.
- Dann schaut der erste Polizist (auf seinem neuen Knoten) entlang "seiner" Kanten:
  Sieht er eine Birne am anderen Ende brennen, wurde ein Kreis mit mind. k verschiedenen Knoten entdeckt.
  Dazu Tiefensuchbaumeigenschaft: Bereits besuchte Knoten, die von "neuem" Knoten n aus mit Kante verbunden sind, liegen auf dem Pfad von der Wurzel zu n.
- Sollte kein Kreis auf diese Weise entdeckt werden, so bilde Bag zu Knoten n des Baumes aus den "letzten $k+1$ Knoten" im beschriebenen Tiefensuchbaum (Pfad zur Wurzel). $\leadsto$ Baumzerlegung der Weite k, insbesondere liegt jede Kante von G in einem Bag.

Win-Win mit Baumweiten

Folgerung 2 *Das Problem, einen Kreis der Länge mindestens* k *zu finden, liegt in FPT.*

Dazu muss man sich noch überlegen, wie man solche Kreise mit Hilfe einer Baumzerlegung findet...
Ähnlich sieht man:

Folgerung 3 *Das Problem, einen Weg der Länge mindestens* k *zu finden, liegt in FPT.*

Iteratives Verbessern

Wird woanders und in der Literatur auch zumeist induktives Komprimieren oder iteratives Komprimieren genannt. Diese Bezeichnungen passen sehr gut bei Problemen, die auf Minimierungsprobleme zurückgehen.

Da sich ähnliche Verfahren auch für Maximierungsprobleme anbieten, dort aber eher ein Expansions- denn ein Kompressions-Schritt vonnöten sind, wähle ich eine allgemeinere Bezeichnung.

Iteratives Verbessern—Das Knotenüberdeckungsproblem

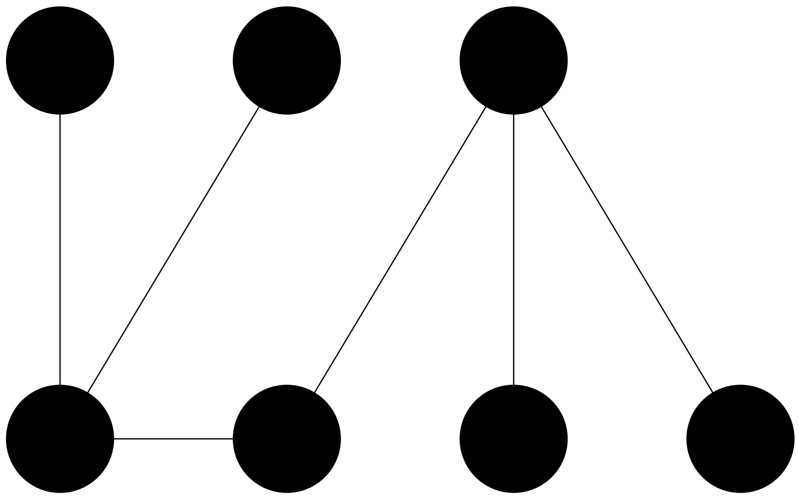
Sei $G = (V, E)$ mit $V = \{v_1, \ldots, v_n\}$, sowie $G_i = G[\{v_1, \ldots, v_i\}]$.
Eine (optimale) Knotenüberdeckung für G_1 ist leer.
Nach Induktionsannahme ist “kleine” Knotenüberdeckung C_i für G_i bekannt.
Dann ist $C'_{i+1} = C_i \cup \{v_{i+1}\}$ eine gültige Knotenüberdeckung für G_{i+1}.
Falls $|C'_{i+1}| = k + 1$, muss ein Kompressionsschritt erfolgen.
Somit erhalten wir in jedem Fall eine Knotenüberdeckung C_{i+1} der Maximalgröße k, oder aber wir haben nachgewiesen, dass eine derart kleine Knotenüberdeckung nicht existiert.

Der Kompressionsschritt beim Knotenüberdeckungsproblem

Lemma 4 *Der Kompressionsschritt lässt sich in Zeit $\mathcal{O}^*(2^\ell)$ durchführen, liegt eine Knotenüberdeckung C' der Größe $\ell > k$ vor.*

Der Algorithmus für den Kompressionsschritt ist leicht skizziert:
Setze $C = C'$.
Betrachte alle Teilmengen X von C'.
Setze $C'' = (C \setminus X) \cup N(X)$, wobei $N(X)$ alle Nachbarn von Knoten aus X aufsammelt.
Ist C'' kleiner als C, so setze $C = C''$.
Ist nach Betrachtung aller Teilmengen immer noch $C = C'$, so NEIN-Instanz.
Andernfalls liefere C zurück.

Ein Beispiel



Es seien v_1, v_2, v_3 von links nach rechts die Knoten in der oberen Reihe und v_4, v_5, v_6, v_7 die in der unteren (von links nach rechts).
Welche Knoten werden nach und nach hinzugenommen, soll eine Knotenüberdeckung der Größe 2 bestimmt werden ?

Der Kompressionsschritt beim Knotenüberdeckungsproblem

Die behauptete Laufzeit für das Verfahren ist evident.

Warum stimmt das Verfahren ? Wir machen folgende Beobachtungen:
(A) Da wir alle Nachbarn von Knoten aus X aufsammeln (was bedeutet das für zwei benachbarte Knoten in C' ?), ist C'' ebenfalls Knotenüberdeckung.
(B) Ist D irgendeine Knotenüberdeckung, so lässt sich C' aufteilen in $D \cap C'$ und $C' \setminus D$. Behauptung: $N(C' \setminus D) \subseteq D \setminus C'$.
Wegen (A) ist $(D \cap C') \cup N(C' \setminus D)$ Knotenüberdeckung.
Wäre die Beh. falsch, so wäre durch die Menge D eine Kante nicht abgedeckt.
Also “sehen” wir insbesondere auch eine kleinstmögliche Überdeckung bei der Durchführung des Kompressionsschrittes.

Folgerung 5 *Das Knotenüberdeckungsproblem lässt sich durch iteratives Verbessern in Zeit $\mathcal{O}^*(2^k)$ lösen.*

Anmerkungen

Wie Sie (möglicherweise) wissen, gibt es Polynomzeitverfahren, die eine optimale Knotenüberdeckung bis auf einen Faktor 2 annähern.
Z.B. kann man den “Nemhauser-Trotter-Kern” als so eine Approximation ansehen, indem einfach alle Knoten in die Überdeckung getan werden.
Gibt es (ganz allgemein) eine Polynomzeit-Approximation mit konstantem Faktor, so lässt sich diese als Ausgangspunkt eines (einzigen !) Kompressionsschrittes ansehen.
Nachteil: Of schlechtere Konstanten als bei der iterativen Methode.
Beispielsweise führt dieser Ansatz zu einem $\mathcal{O}^*(4^{\mathrm{k}})$-Algorithmus für das Knotenüberdeckungsproblem.

Heuristische Anmerkungen

Es muss durchaus nicht sein, dass ein naiv implementierter Suchbaumalgorithmus der iterativen Verfeinerung überlegen ist, auch nicht für das Knotenüberdeckungsproblem:
Tricks: sortiere zunächst die Knoten nach ihrem Grad und fange mit dem größtgradigen an.
Anstatt zuerst den neuen Knoten hinzuzufügen, prüfen wir als allererstes, ob die "alte Überdeckung" auch schon eine ist für den "neuen Graphen".
Bedenke: Die Laufzeit eines Verfeinerungsverfahrens hängt von der Größe der kleinsten Knotenüberdeckung ab und nicht (in erster Linie) vom Parameter k; das kann man ausnutzen, indem ein Kompressionsschritt auch in den ersten k Durchläufen eingebaut wird.

Iteratives Verbessern—Das allgemeine Schema.

Betrachte wachsende Instanzen-Folge $x_0, x_1, x_2, \ldots, x_N$, gebildet aus der Eingabe-Instanz x.
Diese soll die Eingabe ausschöpfen, d.h., $x = x_N$.
Das Folgende ist wieder spezieller für die Kompression...
<u>Induktionsanker</u>: Bestimme triviale Lösung für x_0 der Größe höchstens k.
<u>Induktionsannahme</u>: eine Lösung der Größe $\leq k$ ist für x_i bekannt.
<u>Induktionsschritt</u>: Aus einer Lösung der Größe höchstens k sowie “$x_{i+1} - x_i$” lässt sich “leicht” eine Lösung für x_{i+1} bestimmen, die “nur wenig” größer als k ist. [Ist diese Lösung höchstens gleich k groß, so sind wir hier fertig.]
In einem Kompressionsschritt lässt sich die so gefundene Lösung zu einer der Größe $\leq k$ umformen, sofern das vorgelegte Problem überhaupt eine Lösung der Größe $\leq k$ besitzt.
Die eigentliche “Arbeit” (*FPT*-Zeit) steckt “meist” im Kompressionsschritt.

BIPARTISIERUNG

Eingabe: Graph G

Parameter: natürliche Zahl k

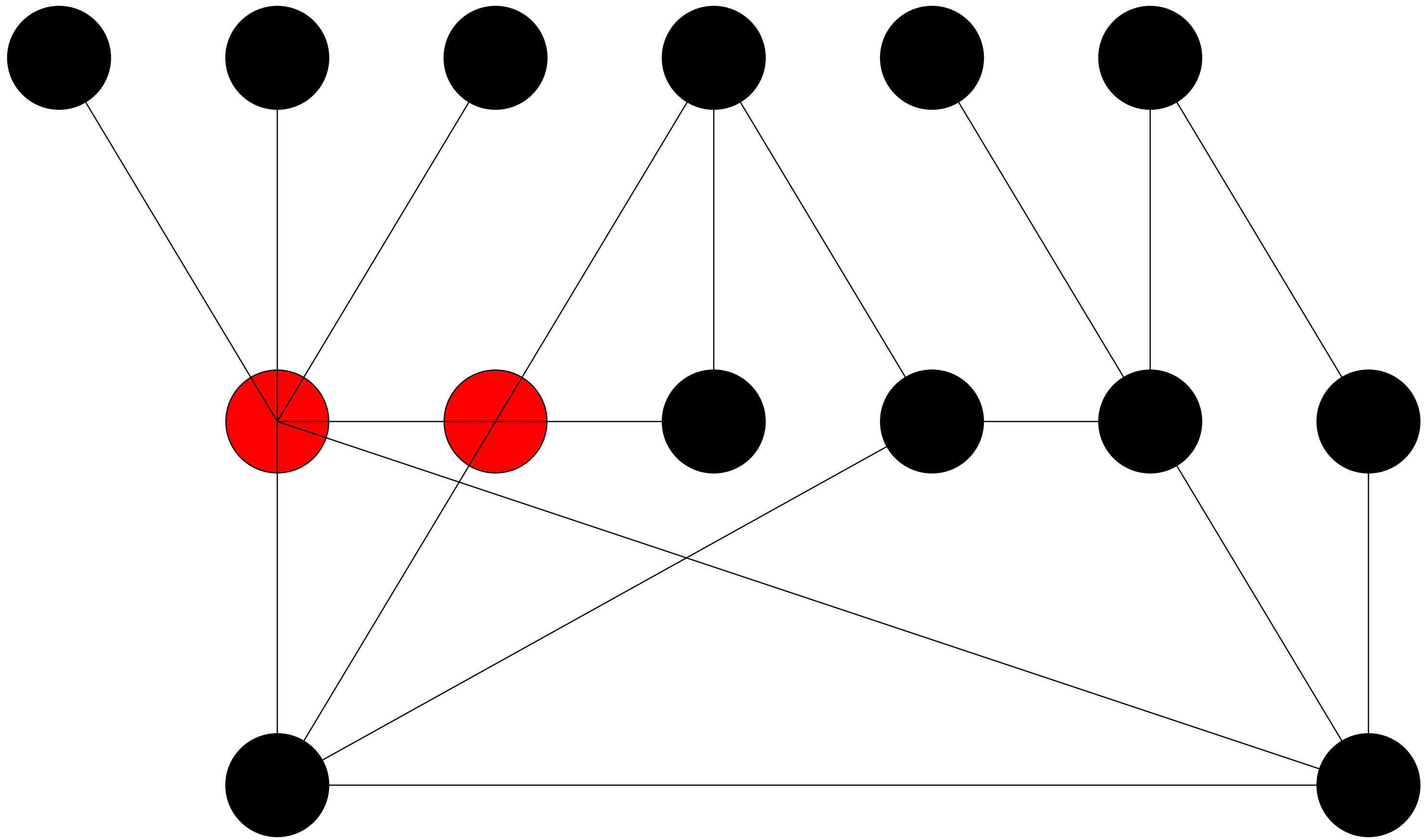
Frage: Gibt es k Knoten B, sodass $G \setminus B$ bipartit ist ?

Dies ist offenbar gleichwertig mit der Aufgabe, k Knoten zu finden, die alle Kreise ungerader Länge abdecken.

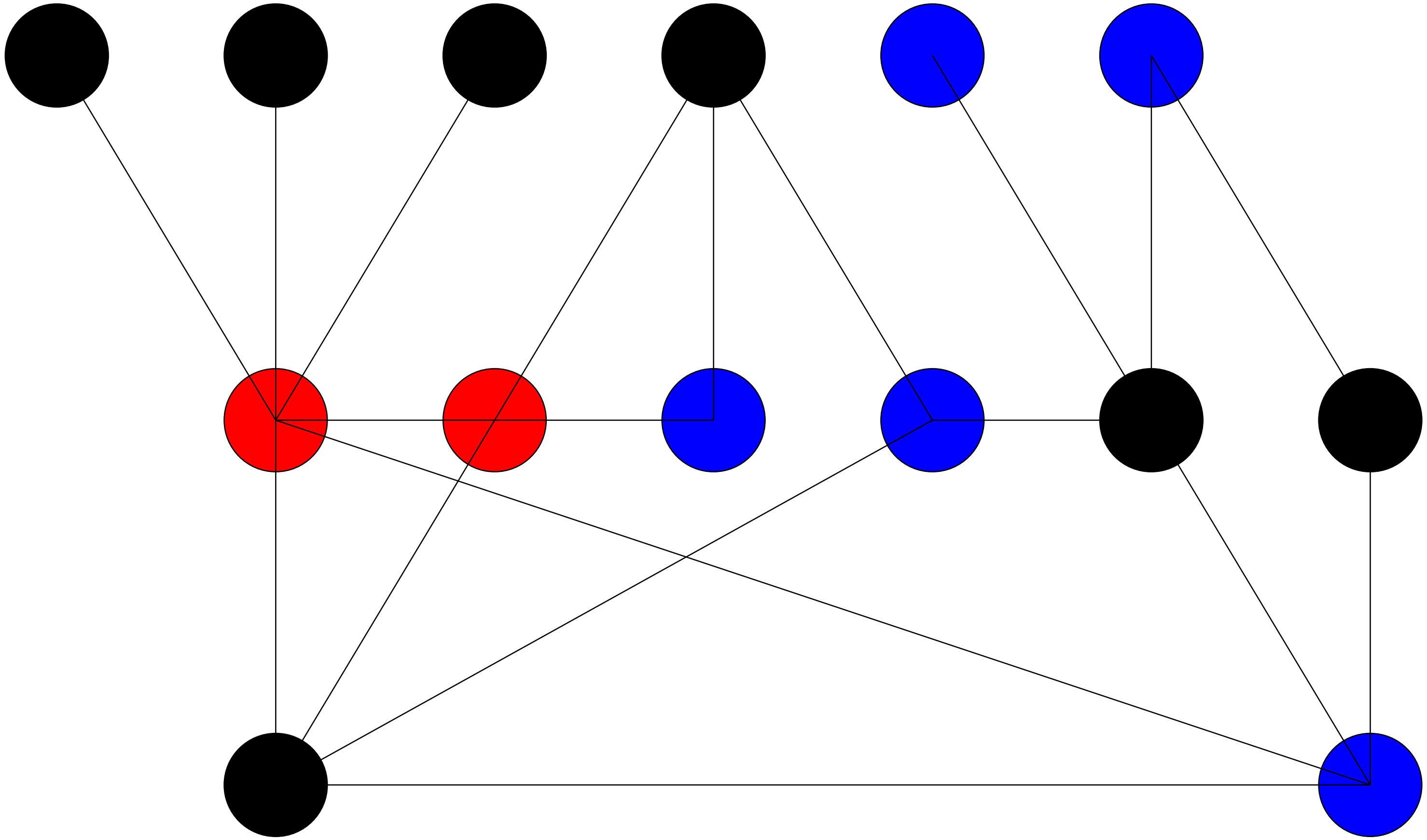
In der Literatur heißt dieses Problem daher auch “Odd cycle cover”.

Erinnerung: bipartit $=$ paar $=$ zweifärbbar

Ein Beispiel: Durch Löschen der roten Knoten wird der Graph bipartit.



Ein Beispiel: noch deutlicher . . .



Bipartisierung—Der Kompressionsschritt

Lemma 6 *Der Kompressionsschritt lässt sich in Zeit* $\mathcal{O}^*(3^\ell)$ *durchführen, liegt eine Bipartisierung* B' *der Größe* $\ell > k$ *vor.*

Folgerung 7 *Das Bipartisierungsproblem lässt sich durch iteratives Verbessern in Zeit* $\mathcal{O}^*(3^k)$ *lösen.*

Übersicht über folgende Folien:

$*$ Definition eines Hilfsgraphen H, der aus einem Graphen G und einer Bipartisierungsmenge B' gewonnen wird.

$*$ Definition einer “gültigen Partition” in H.

$*$ Kompressionsschritt arbeitet auf H.

Bipartisierung—Der Hilfsgraph $H = (V^H, E^H)$

Ausgangslage: Graph $G = (V, E)$, Bipartisierungsmenge B'

B' induziert Bipartition (S_1, S_2) auf $G[V \setminus B']$.

Knotenmenge $V^H = (V \setminus B') \cup \{v_1^H, v_2^H \mid v \in B'\}$

Kantenmenge E^H ist gegeben durch folgende Regeln für Kante $uv \in E$:

(1) Gilt $u, v \in V \setminus B'$, so $uv \in E^H$.

(2) Gilt $u \in S_i$ für $i = 1, 2$, und gilt $v \in B'$, so $uv_{3-i}^H \in E^H$.

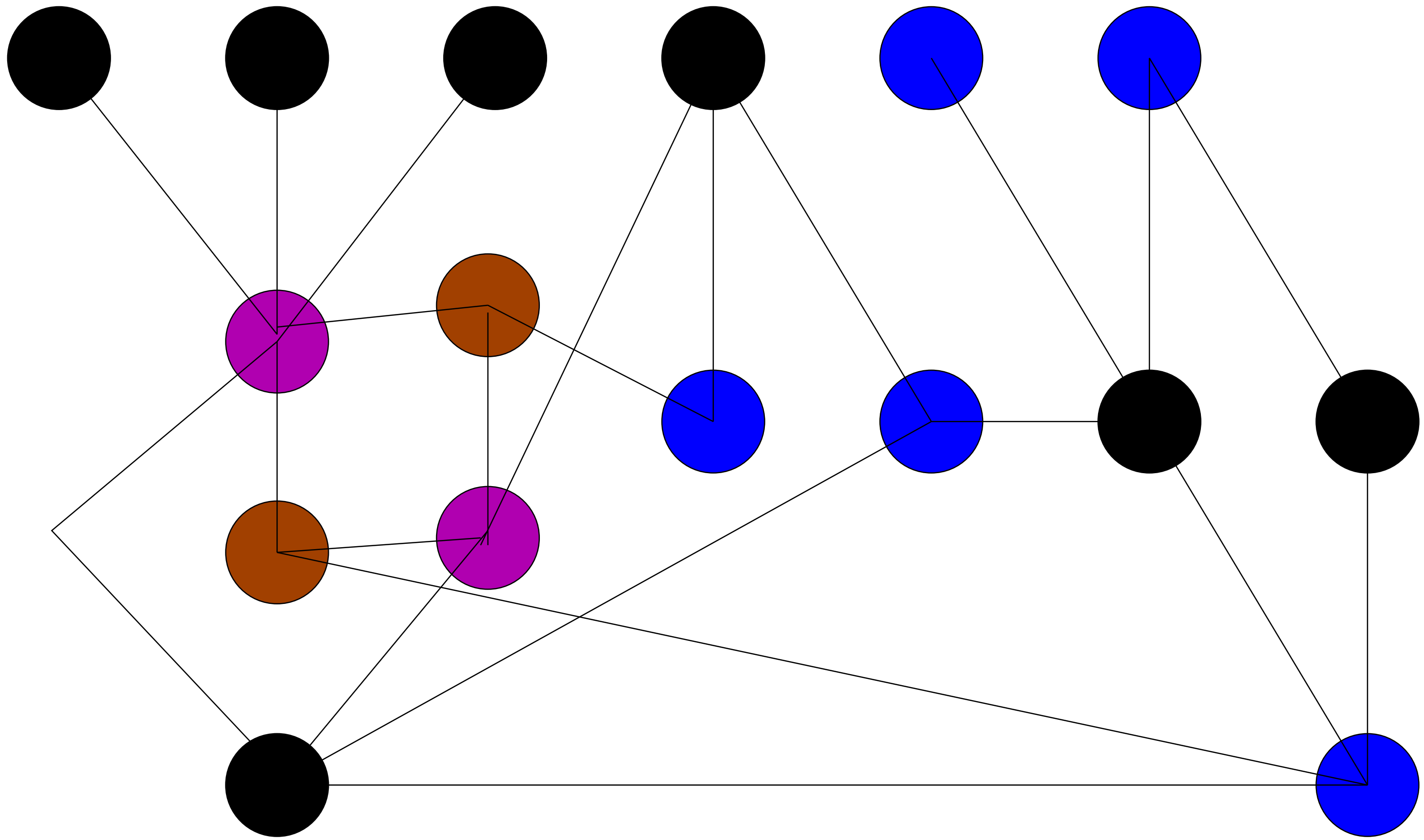
(3) Gilt $u, v \in B'$, so $u_1^H v_2^H \in E^H$. $H[\{v_1^H, v_2^H \mid v \in B'\}]$ ist vollständig bipartit.

Lemma 8 *Die Zweifärbung (S_1, S_2) lässt sich zu einer Zweifärbung von H erweitern: $(S_1 \cup \{v_1^H \mid v \in B'\}, S_2 \cup \{v_2^H \mid v \in B'\})$.*

Eine gültige Partition (X_A^H, X_B^H) zu $X \subseteq B'$ erfüllt für jedes $x \in X$ genau einen der folgenden zwei Fälle:

(1) $x_1^H \in X_A^H$ und $x_2^H \in X_B^H$ ODER (2) $x_1^H \in X_B^H$ und $x_2^H \in X_A^H$.

Ein Beispiel: Der Hilfsgraph mit zweigeteilten Bipartisierungsknoten



Ein Lemma für den Kompressionsschritt

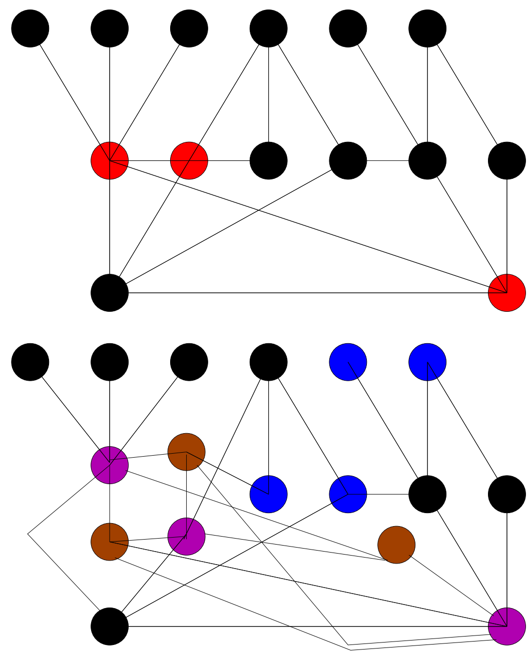
Lemma 9 *Die folgenden beiden Aussagen sind gleichwertig:*
(1) Es gibt eine Bipartisierungsmenge B'' *mit* $|B''| < |B'|$.
(2) Es gibt eine Teilmenge $X \subseteq B'$ *und eine gültige Partition* (X_A^H, X_B^H)*, sodass gilt: Es gibt in* $H[V \setminus B' \cup X^H]$ *(mit* $X^H = X_A^H \cup X_B^H$*) keine* $|X|$ *in* $V \setminus B'$ *knotendisjunkten Wege, die von den Knoten aus* X_A^H *zu den Knoten aus* X_B^H *führen.*

Dabei heißen die Wege $p_1, \ldots, p_r$ in Z knotendisjunkt, wenn für ihre Knotenmengen $P_1, \ldots, P_r$ gilt: $\bigcap_{i=1}^{r}(P_i \cap Z) = \emptyset$.
Diese Aussage kann man im Kompressionsschritt wie folgt benutzen:
Falls es *keine* $|X|$ knotendisjunkten Wege gibt zwischen X_A^H und X_B^H, so bestimme Menge Y von Knoten, $|Y| < |X|$ derart, dass jeder Weg X_A^H und X_B^H über Y führen muss. (Netzflussalgorithmen)
Falls $|(B' \setminus X) \cup Y| < |B|$, setze "Rekordvariable" B auf $(B' \setminus X) \cup Y$.

Zurück zum Beispiel

Angenommen, der Kompressionsschritt würde z.B. mit den beiden “richtigen” Knoten sowie dem Knoten “rechts unten” aufgerufen.
In dem entstehenden Hilfsgraphen wäre die “braune Variante” zu keinem Knoten verbunden (außer zu violetten Varianten der anderen beiden aufgespaltenen Knoten).

Betrachten wir X lediglich bestehend aus jenem Knoten “rechts unten”, so gibt es offenbar überhaupt keinen Weg zwischen der braunen und violetten Variante dieses Knotens in H', der zwischendurch nur Knoten aus dem Ursprungsgraphen benutzt.
Daher können wir jenen Knoten ganz aus der Bipartisierungsmenge entfernen.



Algorithm 1 A greedy algorithm for Bipartization (BP), called GBP

Input(s): a graph $G = (V, E)$, a positive integer k

Output(s): if possible: a subset $B \subset V$, $|B| \leq k$ whose removal produces a bipartite graph or NO if no such set exists.

if $V = 0$ AND $k \geq 0$ **then**
  return $\emptyset$
else
  pick a vertex v
  $B :=$ GBP$(G - v, k)$
  if $B =$NO **then**
    return NO
  else if $|B| < k$ **then**
    return $B \cup \{v\}$
  else
    return BP-improve$(G, B \cup \{v\}, k)$

Algorithm 2 A search algorithm to improve solutions for BP, called improve-BP

Input(s): a graph $G = (V, E)$, a bipartization set B', an integer k, $|B'| = k + 1$

Output(s): if possible: a subset with at most k vertices whose removal produces a bipartite set or NO if no such set exists.

Let $H = (V', E')$ be the auxiliary graph; found:=NO
for all mappings $\alpha : B' \to \{0, 1, 2\}$ AND not found **do**
  {$\alpha(c) = 0$: keep c in the bipartization set}
  {$\alpha(c) \neq 0$: color c with i; for all such vertices to go into the color sets, we have to find "substitutes" that go into the bipartization set instead.}
  Let $X := \{c \in B \mid \alpha(c) \neq 0\}$. $B[i] := \{c_i \mid c \in B'\}$. for $i = 1, 2$.
  Let $X_A := \{c_i \mid c \in B', \alpha(c) = i\}$. Let $X_B := \{c_i \mid c \in B', \alpha(c) = 3 - i\}$.
  Let NOT$(B) := (B[1] \cup B[2]) \setminus (X_A \cup X_B)$. $H' := H -$ NOT(B).
  {By construction, X_A and X_B form a valid partition.}
  if there are less than $|X|$ vertex disjoint paths from X_A to X_B in H' **then**
    found:=YES
    Let Y' be a cutset that separates X_A from X_B in H', with $|Y'| < |X|$.
    Let Y be the set of vertices in G that correspond to vertices in Y'.
    {By definition, $|Y| \leq |Y'|$.}
    {$Y \cup (B' \setminus X)$ is a bipartization set of G}
if found **then**
  return $Y \cup (B' \setminus X)$
else
  return NO

Laufzeit für den Kompressionsschritt

Wie beim Knotenüberdeckungsproblem werden von der “aktuellen Bipartisierungsmenge” B' beliebige Teilmengen X betrachtet.
Weiter werden nun aber beliebige gültige Partitionen von X durchgegangen.
Mit $|B'| = \ell$ liefert dies 3^ℓ Möglichkeiten, denn

$$\sum_{X \subseteq B'} \sum_{X' \subseteq X} 1 = \sum_{i=0}^{\ell} \binom{\ell}{i} 2^i = 3^\ell$$

Kombinatorische Begründung und Interpretation von X_1:
Betrachte Abbildungen $f : B' \to \{0, 1, 2\}$ mit der Bedeutung: $f(x) = 0$ gdw. $x \notin X$, $f(x) = 1$ gdw. $x \in X_1$ gdw. $x_1^H \in X_A^H$ (und damit $x_2^H \in X_B^H$), sowie $f(x) = 2$ gdw. $x \in X \setminus X_1$ gdw. $x_2^H \in X_A^H$ (und damit $x_1^H \in X_B^H$).
Beobachte: das so durch X_1 festgelegte Paar (X_A^H, X_B^H) ist eine gültige Partition, und jede gültige Partition lässt sich so darstellen.
Das führt auf die behauptete Laufzeitabschätzung.

Ein Lemma für den Kompressionsschritt: Beweis 1.

Lemma 10 *Die folgenden beiden Aussagen sind gleichwertig:*
(1) Es gibt eine Bipartisierungsmenge B'' *mit* $|B''| < |B'|$.
(2) Es gibt eine Teilmenge $X \subseteq B'$ *und eine gültige Partition* (X_A^H, X_B^H), *sodass gilt: Es gibt in* $H[V \setminus B' \cup X^H]$ *(mit* $X^H = X_A^H \cup X_B^H$*) keine* $|X|$ *knotendisjunkten Wege, die von den Knoten aus* X_A^H *zu den Knoten aus* X_B^H *führen.*

Beweis: "$\Leftarrow$:" Sei $X \subseteq B'$ nicht leer und (X_A^H, X_B^H) eine gültige Partition von X, sodass es keine $|X|$ knotendisjunkten Wege zwischen X_A^H und X_B^H gibt.
$\rightsquigarrow$ Es gibt Knotenmenge Y, $|Y| < |X|$, $Y \cap B' = \emptyset$, sodass jeder Weg von X_A nach X_B in $H' = H[(V \setminus B') \cup X_A \cup X_B]$ über Y führt.
$\rightsquigarrow$ Jeder ungerade Kreis in G führt durch Knoten aus $B' \setminus X$ oder aus Y.
$\rightsquigarrow$ $(B' \setminus X) \cup Y$ ist eine kleinere Bipartisierungsmenge als B'.

Ein Lemma für den Kompressionsschritt: Beweis 2.

Lemma 11 *Die folgenden beiden Aussagen sind gleichwertig:*
(1) Es gibt eine Bipartisierungsmenge B'' mit $|B''| < |B'|$.
(2) Es gibt eine Teilmenge $X \subseteq B'$ und eine gültige Partition (X_A^H, X_B^H), sodass gilt: Es gibt in $H[V \setminus B' \cup X^H]$ (mit $X^H = X_A^H \cup X_B^H$) keine $|X|$ knotendisjunkten Wege, die von den Knoten aus X_A^H zu den Knoten aus X_B^H führen.

Beweis: "$\Rightarrow$": Sei B'' eine Bipartisierungsmenge mit $|B''| < |B'|$.
Setze $X := B' \setminus (B'' \cap B')$ und $Y := B'' \setminus (B'' \cap B')$. $\rightsquigarrow |Y| < |X|$,
(S_1, S_2): Zweifärbung von $G[V \setminus B']$ bzw. ihre Erweiterung auf H.
$(\hat{S}_1, \hat{S}_2)$: Zweifärbung von $G[V \setminus B'']$.
Beobachte: Folgendes liefert eine gültige Partition von X:
$X_A^H := \{x_1^H \mid x \in X \cap \hat{S}_1\} \cup \{x_2^H \mid x \in X \cap \hat{S}_2\}$,
$X_B^H := \{x_1^H \mid x \in X \cap \hat{S}_2\} \cup \{x_2^H \mid x \in X \cap \hat{S}_1\}$.
Nämlich: Nach Konstruktion gilt für $x \in X$: $x \notin B''$, und daher muss $x \in \hat{S}_1$ oder $x \in \hat{S}_2$ zutreffen.
Beweise abschließend Behauptung: "Jeder Weg zwischen X_A und X_B führt über Y." durch Widerspruch: "Es gibt einen Weg von $u^H \in X_A^H$ nach $v^H \in X_B^H$, der nicht durch Knoten aus Y führt."
Durch Betrachtung verschiedener Fälle kann man zeigen:

Wegen der Zweifärbung (S_1, S_2) gibt es einen Weg ungerader / gerader Länge und wegen der Zweifärbung $(\hat{S}_1, \hat{S}_2)$ gibt es einen Weg gerader / ungerader Länge.

Wir unterscheiden vier Fälle für die entsprechenden Knoten u, v in G:
(1) $u, v \in X \cap \hat{S}_1$. Nach Konstruktion von X_A gilt daher genauer $u^H = u_1^H$ und $v^H = v_2^H$.
Also liegen u^H und v^H in verschiedenen Farbklassen bzgl. (S_1, S_2).
Der Weg von u^H nach v^H hat daher ungerade Länge.
Andererseits liegen u^H und v^H in derselben Farbklasse $\hat{S}_1$ (bgzl. $(\hat{S}_1, \hat{S}_2)$ in G).
Da der betrachtete Weg zwischen u^H und v^H keine Y-Knoten enthält, hat er gerade Länge. $\lightning$
(2) $u, v \in X \cap \hat{S}_2$ sieht man ganz genauso ein.
(3) $u \in X \cap \hat{S}_1$, $v \in X \cap \hat{S}_2$.
Wegen $u^H, v^H \in S_1$ hat ein Weg zwischen u^H und v^H gerade Länge.
Andererseits gilt $u \in \hat{S}_1$ und $v \in \hat{S}_2$.
Da der betrachtete Weg zwischen u^H und v^H keine Y-Knoten enthält, hat er ungerade Länge. $\lightning$
(4) $u \in X \cap \hat{S}_2$, $v \in X \cap \hat{S}_1$ sieht man genauso ein.

Durch Kontraposition ergibt sich aus dem Lemma:

Folgerung 12 *Die vorzuliegende Bipartisierungsmenge B' ist kleinstmöglich gdw. es für jede Wahl $X \subseteq B'$ und für jede gültige Partition (X_A^H, X_B^H) von X (wenigstens) $|X|$ knotendisjunkte Wege zwischen X_A^H und X_B^H gibt.*

Abschließendes zum Iterativen Verbessern

Ausgangspunkt: B. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32:299–301, 2004.

Dieser Arbeit folgt auch unsere Darstellung.

Für das Bipartisierungsproblem war zuvor lange unbekannt, ob es einen parameterisierten Algorithmus dafür gibt.

Andere “Kreiszerstörungsprobleme” können ebenso behandelt werden:
edge bipartization, feedback vertex set, . . .

Einen guten Überblick über die Anwendungen dieser Technik bietet H. Mosers Vortrag.

Schöne Beispiele für Iteratives Verbessern bei Maximierungsproblemen fehlen mir (noch).