Masterarbeit im Fachbereich Mathematik der
FernUniversität Hagen
mit dem Thema

Minkonvexe Faktoren fixer Kantenzahl

Annabell Berger

23. Oktober 2006

Annabell Berger
Mendelssohnweg 7
07743 Jena
Matrikelnummer: 5014697
Geburtsdatum:

Betreuer: Professor Dr. Winfried Hochstättler

Inhaltsverzeichnis

1 Motivation - Gutschein für ein Essen zu zweit **3**

2 Das Minkonvex Problem **5**
2.1 Formalisierung 5
2.2 Der klassische Weg - Verbesserungspfade 8
2.3 Die schnelle Lösung - Der gewichtete Graph 13

3 Die Bestimmung eines maximal-gewichteten f-Faktors **20**
3.1 Das Knotensplitting von Tutte 21
3.2 Die Berechnung eines ungewichteten 1-Faktors 24
3.3 Das ungewichtete f-Faktor Problem 29
3.4 Das gewichtete 1-Faktor Problem als lineares Programm 33
3.5 Der Algorithmus für einen maximal-gewichteten f-Faktor 41

4 Verwandte Probleme **47**
4.1 Das Maxsquare Problem 47
4.2 Das Maxfixcover Problem 48

5 Zusammenfassung **49**

1 Motivation - Gutschein für ein Essen zu zweit

Vorliegende Arbeit behandelt diskrete Optimierungsprobleme auf der Knotenmenge von Graphen G auf folgende Weise. Eine Funktion $k : \mathbb{N}_0 \to \mathbb{R}$ von den nichtnegativen Zahlen auf die reellen Zahlen heißt *diskrete konvexe Funktion*, wenn für alle $i \in \mathbb{N}_0$ gilt

$$2k(i+1) \leq k(i) + k(i+2).$$

Wir betrachten den Graphen $G = (V, E)$. Bezeichne $H = (V, F)$ mit $F \subseteq E$ einen Teilgraphen von G und $d_H : V \to \mathbb{N}_0$ seine Knotengradfunktion. Wir wollen einen Teilgraphen $H = (V, F)$ mit $|F| = k$ bestimmen, für den

$$k(H) := \sum_{v \in V} k(d_H(v))$$

minimiert wird. Viele Anwendungsbeispiele für diese Optimierungsprobleme sind denkbar. Betrachten wir als einleitendes, intuitives Beispiel ein Problem, das sich im Alltag einiger Mathematiker ergeben könnte.
Eines Tages besuchte der Leiter des mathematischen Institutes seine fünf Mitarbeiter und sprach: „Ein ganzes Jahr anstrengende Arbeit liegt hinter Euch. Zum Dank dafür sollt Ihr gemeinsam Essen gehen. Ich schenke Euch vier Gutscheine für ein Essen zu zweit. Bitte teilt sie gerecht untereinander." Kurz bevor er den Raum verließ, hörte er, wie ein Mathematiker murmelte: „Gerecht wäre es, wenn jeder $1,6$ Menüs verspeist." Ein Lächeln zuckte um die Mundwinkel des Leiters und kurz bevor er die Tür schloss, sagte er verschmitzt: „Und teilt weise."
Nachdem sie wieder unter sich waren, fingen sie an zu diskutieren. Sie hatten zwar gut zusammengearbeitet, doch manche Mathematiker mochten sich nicht gern genug, um auch gemeinsam Essen zu gehen. Geht man davon aus, dass Sympathie auf Gegenseitigkeit beruht und schließen wir übertriebene Eigenliebe aus, dann ergibt sich für unsere Arbeitskollegen folgender schlichte Sympathiegraph.

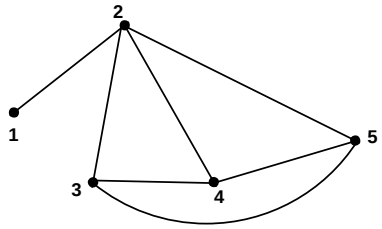

Abbildung 1: Der Sympathiegraph

Sie überlegten weiter, dass sich je zwei Mathematiker einen Gutschein teilen müssen, wobei jeder Mathematiker aus Gerechtigkeitsgründen mit möglichst einer ähnlichen Anzahl von Mathematikern Essen gehen sollte. Gerechtigkeit kann

wegen unseres Sympathiegraphen offenbar nicht bedeuten, dass jeder Mathematiker genau so oft wie ein anderer Mathematiker Essen gehen kann. Klar ist auch, dass kein Mathematiker einen Gutschein alleine in Anspruch nehmen darf. Angewandt auf unseren Sympathiegraphen ergibt sich folgende Problemstellung. Wie kann man vier Kanten als Gutscheine so markieren, dass eine möglichst gerechte Aufteilung der Gutscheine stattfindet. Recht plausibel scheint folgende Vorstellung von Gerechtigkeit - Wenn sich möglichst wenige Gutscheine im Graphen paaren, dann bekommen alle Mathematiker eine den Umständen entsprechend ähnliche Möglichkeit. Hier sieht man drei Lösungen, die ersten beiden sind optimal und die dritte ist nicht optimal.

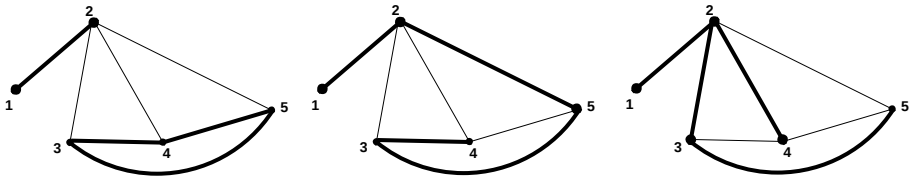

Abbildung 2: Zwei optimale und eine nicht-optimale Lösung

Intuitiv ist klar, dass aufgrund $8 \div 5$ drei Mathematiker je zwei gemeinsame Essen ((8 mod 5) und (8 div 5)+1) und zwei Mathematiker je ein gemeinsames Essen (8 div 5) teilen müssen. Ob dieses absolute Minimum, wie in unserem Fall möglich ist, hängt von den Beziehungen untereinander ab. Angenommen, der Institutsleiter hätte ihnen sechs Gutscheine gegeben, dann ist die bestmögliche Lösung, dass sich zwei Mathematiker drei Gutscheine und drei Mathematiker zwei Gutscheine teilen. Das ist aber wegen Mathematiker 1 nicht möglich. Die nächstbessere Lösung wäre, dass sich ein Mathematiker einen Gutschein, ein anderer zwei und die Restlichen drei Gutscheine teilen. In unserem Fall gibt es diese Lösung, und nur drei zu prüfende Möglichkeiten. Bei n Mathematikern und $|F|$ Gutscheinen könnte die Problemstellung eintreten, dass die Existenz eines absoluten Minimums zu prüfen ist, das für $n - (2|F| \bmod n)$ Knoten den Grad $(2|F| \operatorname{div} n)$ und für $(2|F| \bmod n)$ Knoten den Grad $((2|F| \operatorname{div} n) + 1)$ ergibt. Somit kommen $\binom{n}{k}$ mit $(k := 2|F| \bmod n)$ mögliche Kombinationen in Frage. Aufgrund der bekannten Beziehung $2^n = \sum_{k=0}^{n} \binom{n}{k}$, können es exponentiell viele Möglichkeiten sein. Denn angenommen, es gelte $\binom{n}{k} \in O(n^{s_k})$ für alle $k \in \{1, \ldots, n\}$ mit $s_k \in \mathbb{N}$. Dann folgt $2^n \in O(n^{s_l})$, wobei $s_l = \max\{s_k + 1 | k \in 1, \ldots, n\}$ ist. Widerspruch!
Es stellt sich die Frage, ob dieses Problem überhaupt in polynomieller Zeit lösbar ist. Eine positive Antwort, $O(|V|^{5,5})$, wurde in der Arbeit „Minsquare Factors and Maxfixcover Covers of Graphs " von Nicola Appolonio und Andras Sebö [3] gegeben. In vorliegender Arbeit wird im ersten Teil eine Reduktion des Problems zu einem gewichteten f-Faktor Problem vorgenommen. Diesen Weg wählten auch Nicola Appolonio und Andras Sebö in einer zweiten Arbeit [4]. Allerdings ist diese Reduktion sehr kompliziert und vergrößert zudem die Komplexität des Problems beträchtlich. Unsere Reduktion ermöglicht mit dem Algorithmus

von Gabow [9] eine Lösung des Problems in der Zeit $O(|E|^2 \log(|V|))$. Damit verbessern wir die Komplexität von Appolonio und Sebö. Im zweiten Teil der Arbeit stellen wir den Algorithmus für das gewichtete f-Faktor Problem vor, der in seinen einzelnen Bestandteilen auf den Arbeiten von Tutte [10], Edmonds [10], Ford, Fulkerson und Dantzig [12], Galil, Gabow und Micali [8] und Gabow [9] basiert. Entscheidend für die Komplexität unseres Problems sind die Arbeiten [9] und [8]. Wir haben die Ergebnisse hinsichtlich der Lösung des Minkonvex Problems zusammengetragen.

2 Das Minkonvex Problem

2.1 Formalisierung

Für das Verständnis der nächsten Abschnitte ist es notwendig, einige Konventionen bezüglich der Notation zu treffen. Ein Graph $G = (V, E)$ ist ein Paar von disjunkten endlichen Mengen, wobei die Elemente von E 2-Multimengen von V sind. Die Elemente von V nennt man Knoten (Vertex) des Graphen G und die Elemente von E bezeichnet man als Kanten (Edge) von G. Man schreibt für V auch $V(G)$ und für E auch $E(G)$. Ein Knoten v heißt mit einer Kante e *inzident*, wenn $v \in e$ gilt. Inzidiert nur ein Knoten v mit einer Kante e, so heißt e *Schleife*. Inzidieren zwei Knoten mit einer Kante e_1 und einer weiteren Kante e_2, so heißen e_1 und e_2 *parallele Kanten*. Besitzt ein Graph G keine Schleifen und keine parallelen Kanten, so bezeichnen wir ihn als *schlichten Graphen*. Gilt für zwei Knoten $v_1, v_2 \in E$, so heißen sie *benachbart* oder *adjazent*. Die Funktion $d_G : V \to \mathbb{N}$ mit

$$d_G(v) = |\{e | e \in E(G), \text{ v ist inzident zu } e\}|$$

heißt *Gradfunktion des schlichten Graphen* G und gibt für jeden Knoten v die Anzahl der inzidenten Knoten, den *Knotengrad* an. Summiert man die Knotengrade aller Knoten eines Graphen, so ist klar, dass jede Kante genau zweimal gezählt wird. Also gilt

$$\sum_{v \in V} d_G(v) = 2|E|.$$

Die Gradfunktion ist auch für Graphen mit Schleifen und parallelen Kanten definiert. Dabei ist zu beachten, dass eine Schleife an einem Knoten doppelt gezählt wird. Ein Graph $G' = (V', E')$ heißt *Teilgraph* von G, wenn $V' \subseteq V$ und $E' \subseteq E$ gilt. Man betrachte nun den Teilgraphen $H = (V, F)$ von G mit $F \subseteq E$. Gilt für eine Funktion $f : V \to \mathbb{N}_0$

$$f = d_H,$$

so nennt man den Teilgraphen H einen *f-Faktor von* G. Wir bezeichnen im Weiteren eine Kante $e \in E$ aus G als *F-Kante*, wenn sie in der Kantenmenge

des Teilgraphen $H = (V, F)$ liegt. Ein *Spaziergang auf G* ist ein nichtleerer Teilgraph $S = (V, W)$ von G der Form

$$W = \{\{v_0v_1\}, \{v_1v_2\}, \ldots, \{v_{k-1}v_k\}\}.$$

Sind die $v_i, v_j \in V$ paarweise verschieden, so bezeichnen wir den Spaziergang S als *Pfad P*. Der Pfad P heißt *Kreis*, wenn $v_0 = v_k$ gilt. Wir identifizieren im Folgenden einen Spaziergang S mit einer alternierenden Folge von Knoten und Kanten, schreiben also

$$S = (v_0, \{v_0, v_1\} \ldots, \{v_{k-1}, v_k\}, v_k)$$

und nennen S einen Spaziergang von v_0 nach v_k. Ist k eine gerade, ganze Zahl, so heißt der Spaziergang S *gerader Spaziergang*, sonst *ungerader Spaziergang*. Weiterhin definieren wir $S \cap H := (V, F \cap W)$ bzw. $S \cup H = (V, F \cup W)$ als Teilgraphen von G. Ein nicht leerer Teilgraph $H = (V', E')$ von $G = (V, E)$ heißt *zusammenhängend*, wenn für je zwei Knoten $v, v' \in V'$ ein Pfad $P = (v, \ldots, v')$ in H existiert. Ein maximal zusammenhängender Teilgraph H von G heißt *Komponente* von G. Ein Graph G, der keinen Kreis enthält, nennt sich *Wald*. Ein zusammenhängender Wald nennt sich *Baum*. Ein Wald ist demnach ein Graph, dessen Komponenten Bäume sind.
Sei nun in unserem Problem $G := (V, E)$ ein schlichter, ungerichteter Graph (der Sympathiegraph). Gesucht ist ein Teilgraph $H := (V, F)$ mit einer Kantenmenge $F \subseteq E$ konstanter Kardinalität $|F| =: k$ (die Gutscheine), bei welchem die Anzahl von Kantenpaaren minimal ist. D.h. es soll gelten

$$s(H) := \sum_{v \in V} \binom{d_H(v)}{2} = \frac{1}{2} \sum_{v \in V} d_H(v)^2 - \frac{1}{2} \sum_{v \in V} d_H(v) = \frac{1}{2} \sum_{v \in V} d_H(v)^2 - k = min.$$

Es ergibt sich das Optimierungsproblem

$$Minsquare(G, k) := \Big\{ H = (V, F) | \sum_{v \in V} d_H(v) = 2k \text{ und } \sum_{v \in V} d_H(v)^2 = min \Big\}.$$

Im Allgemeinen existieren mehrere optimale Lösungen, wie man bereits beim Einführungsbeispiel sehen konnte. Wir wollen das Problem verallgemeinern.

Definition 2.1.1. *Eine Funktion $k : \mathbb{N}_0 \to \mathbb{R}$ von den nichtnegativen Zahlen auf die reellen Zahlen heißt diskrete konvexe Funktion, wenn für alle $i \in \mathbb{N}_0$ gilt*

$$2k(i+1) \leq k(i) + k(i+2).$$

Bei dieser Definition handelt es sich um das eindimensionale diskrete Analogon zu einer reellwertigen konvexen Funktion $K : D \to \mathbb{R}$, wobei D eine konvexe Menge mit $D \subseteq \mathbb{R}^n$ ist. Für eine konvexe Funktion K gilt die Beziehung $K(\alpha x + \beta y) \leq \alpha K(x) + \beta K(y)$ für alle $x, y \in D$ und $\alpha, \beta \in \mathbb{R}$ mit $\alpha + \beta = 1$. Setzen wir für $\alpha, \beta := \frac{1}{2}$ und für $x := i, y := i + 2$, wobei die $i, i+1, i+2 \in D$ ganze Zahlen sind, so erhalten wir $2K(i+1) \leq K(i) + K(i+2)$.

Satz 2.1.2. *Für eine diskrete konvexe Funktion* $k : \mathbb{N}_0 \to \mathbb{R}$, $\xi_1, \xi_2 \in \mathbb{Z}$ *und* $i \in \mathbb{N}$ *mit* $i + (\xi_1 + \xi_2) \geq 0$ *gilt*

$$k(i + (\xi_1 + \xi_2)) - k(i) \geq k(i + \xi_1) - k(i) + k(i + \xi_2) - k(i),$$

wenn ξ_1, ξ_2 *das gleiche Vorzeichen besitzen.*

Beweis. Für positive ξ_1, ξ_2 gilt wegen 2.1.1

$$\begin{aligned}
k(i + (\xi_1 + \xi_2)) - k(i + \xi_1) &= \sum_{k=\xi_1+1}^{\xi_1+\xi_2} (k(i + \kappa) - k(i + (\kappa - 1)) \\
&\geq \sum_{\kappa=\xi_1+1}^{\xi_1+\xi_2} (k(i + (\kappa - 1)) - k(i + (\kappa - 2)) \\
&\geq \sum_{\kappa=\xi_1+1}^{\xi_1+\xi_2} (k(i + \kappa - 2) - k(i + (\kappa - 3)) \\
&\geq \ldots \geq \sum_{\kappa=\xi_1+1}^{\xi_1+\xi_2} (k(i + \kappa - \xi_1) - k(i + (\kappa - \xi_1 - 1)) \\
&= k(i + \xi_2) - k(i).
\end{aligned}$$

Nach Vertauschen der Summationsgrenzen, erfolgt der Fall mit negativen Vorzeichen analog.

□

Dieser Satz läßt sich zu einem für unsere Arbeit sehr wichtigen Fakt verallgemeinern.

Folgerung 2.1.3. *Es sei* $k : \mathbb{N}_0 \to \mathbb{R}$ *eine diskrete konvexe Funktion und* $i \in \mathbb{N}_0$. *Besitzen die ganzen Zahlen* $\xi_1, ..., \xi_s$ *das gleiche Vorzeichen und ist* $\left(i + \sum_{l=1}^{s} \xi_l\right) \geq 0$, *so gilt*

$$k(i + \sum_{l=1}^{s} \xi_l) - k(i) \geq \sum_{l=1}^{s} (k(i + \xi_l) - k(i)).$$

Beweis. Mit vollständiger Induktion nach s.

- Induktionsanfang: Für $s = 1$ gilt offenbar Gleichheit.
- Induktionsvoraussetzung: Die Behauptung gelte für alle $t < s$.
- Induktionsbeweis: Mit 2.1.2 ergibt sich:

$$\begin{aligned} k(i+\sum_{l=1}^{s}\xi_l)-k(i) &= k(i+\sum_{l=1}^{s-1}\xi_l+\xi_s)-k(i) \\ &\geq k(i+\sum_{l=1}^{s-1}\xi_l)-k(i)+l(i+\xi_s)-k(i) \\ &\geq \sum_{l=1}^{s-1}(k(i+\xi_l)-k(i))+k(i+\xi_s)-k(i) \\ &= \sum_{l=1}^{s}(k(i+\xi_l)-k(i)). \end{aligned}$$

□

Wir betrachten nun den Graphen $G=(V,E)$. Sei $k\in\mathbb{N}$ und $l_v:\mathbb{N}_0\to\mathbb{R}$ eine diskrete konvexe Funktion für alle $v\in V$. Sei weiterhin $l:\mathbb{Z}^V\to\mathbb{R}$ die *diskrete separable konvexe Funktion*

$$l(d_H):=\sum_{v\in V}l_v(d_H(v)).$$

Wir wollen einen Teilgraphen $H=(V,F)$ mit $|F|=k$ bestimmen, für den $l(d_H)$ minimiert wird. In der Notation von oben ergibt sich das Problem

$$Minkonvex(G,k)=\Big\{H=(V,F)|\sum_{v\in V}d_H(v)=2k \text{ und } l(d_H)=min\Big\}.$$

Offenbar handelt es sich bei der Funktion $s:\mathbb{N}_0\to\mathbb{R}$ mit $s(x):=x^2$ um eine diskrete konvexe Funktion, so dass Minsquare ein Spezialfall von Minkonvex ist.

2.2 Der klassische Weg - Verbesserungspfade

In vielen Algorithmen der sogenannten Matchingtheorie hat es sich bewährt, ausgehend von einer nicht optimalen Lösung schrittweise nach Verbesserungen zu suchen. Mit diesem Ansatz gelang es auch Apollonio und Sebö [3] zu zeigen, dass ein polynomieller Algorithmus für das Minsquare-Problem existiert.
Wir wählen einen beliebigen Teilgraphen $H=(V,F)$ von $G=(V,E)$, der nicht minimal ist. Dann können wir, ausgehend von einem Knoten v_i und einer F-Kante $\{v_i,v_j\}$, einen Spaziergang suchen, dessen Kanten abwechselnd in F und in $(E\setminus F)$ liegen und der mit einer $(E\setminus F)$-Kante $\{v_{i'},v_{j'}\}$ und dem Knoten $v_{j'}$ endet. Vertauschen wir die F-Kanten mit den $(E\setminus F)$-Kanten, dann erhalten wir einen neuen Teilgraphen $H'=(V,F')$. Ob es sich um eine Verbesserung der Minimalitätsbedingung handelt, hängt von der Gradsequenz d_H von H ab. Es ist

$$\sum_{v \in V} l_v(d_{H'}(v)) \quad = \quad \sum_{v \in V \setminus (\{v_i\} \cup \{v_{j'}\})} l_v(d_H(v)) + l_{v_i}((d_H(v_i) - 1)) + l_{v_{j'}}(d_H(v_{j'}) + 1).$$

Gilt $l_{v_i}((d_H(v_i) - 1)) + l_{v_{j'}}(d_H(v_{j'}) + 1) < l_{v_i}(d_{H'}(v_i)) + l_{v_{j'}}(d_H(v_{j'})$, dann haben wir eine Verbesserung vorgenommen. Für unser Minsquare-Problem bedeutet dies beispielsweise $d_H(v_{j'}) + 1 < d_H(v_i)$. Das folgende Beispiel zeigt eine solche Verbesserung.

Beispiel 2.2.1. *Wir verändern den nicht-optimalen Teilgraphen $H = (V, F)$ aus Abbildung 2 auf Seite 4 durch den Pfad $P = (2, 24, 4, 45, 5)$ zu folgendem Teilgraphen $H' = (V, F')$. Wegen $d_H(2) = 3$ und $d_H(5) = 1$ handelt es sich um eine Verbesserung.*

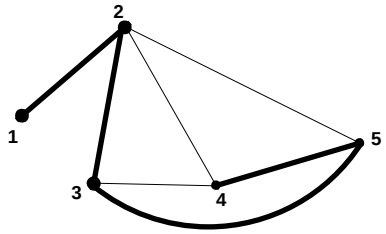

Abbildung 3: Verbesserung durch Pfad P

Bedauerlicherweise existiert ein solcher Spaziergang bzw. Pfad nicht immer, wenn H nicht minimal ist. Das folgende Beispiel beweist dies bereits im bipartiten Graphen.

Beispiel 2.2.2. *Ein verbessernder Spaziergang von 7 nach 6 existiert nicht. Das wäre wegen obiger Bedingung die einzige Möglichkeit. Allerdings gibt es die zwei Pfade $(1, 15, 5, 53, 3, 37, 7)$ und $(2, 26, 6)$, die nach Tausch der F- und $(E \setminus F)$-Kanten einen minimalen Teilgraphen $H' = (V, F')$ bereitstellen.*

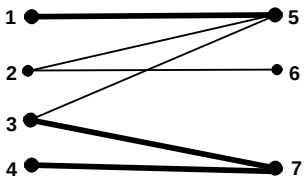

Abbildung 4: Nichtoptimaler bipartiter Graph

Es zeigt sich jedoch in 2.2.7, dass es eine der zwei Verbesserungsmöglichkeiten aus unseren Beispielen geben muss, wenn der Teilgraph $H = (V, F)$ nicht minimal ist. Es ergehen folgende Definitionen.

Definition 2.2.3. *Sei S ein Spaziergang, P ein Pfad auf G und $F \subseteq E$. Man nennt S einen $F-(E \setminus F)$-alternierenden Spaziergang und P einen $F-(E \setminus F)$-alternierenden Pfad, wenn die Kanten von S bzw. P abwechselnd in F und $(E \setminus F)$ liegen.*
Sei S ein F-$E \setminus F$-alternierender Spaziergang. Die Abbildung $\lambda_S : V \to \mathbb{Z}$ mit $\lambda_S(v) := d_{S \setminus H}(v) - d_{S \cap H}(v)$ heißt der Wechsel von F bezüglich S.

Ein alternierender Spaziergang kann an seinen Enden zwei F-Kanten, zwei $(E \setminus F)$-Kanten oder jeweils eine F-Kante und eine $(E \setminus F)$-Kante besitzen. Ein Vertauschen der Kanten wie in unseren Beispielen, kann demnach zu einem Überschuß von einer F-Kante bzw. einer $(E \setminus F)$-Kante führen. Für den Wechsel ergeben sich in diesen drei Fällen $\sum_{v \in V} \lambda_S(v) = -2$ oder $+2$ oder 0. Demnach beschreibt der Wechsel genau die Veränderung des Knotengrades nach Tausch der Kanten. Um einen Überschuss bzw. das Fehlen einer Kante nach dem Kantentausch auf einem alternierenden Spaziergang zu vermeiden, erscheint es sinnvoll, solche Spaziergänge zu paaren.

Definition 2.2.4.

1. *Ein Verbesserungsspaziergang erster Art ist ein gerader F-$(E \setminus F)$-alternierender Spaziergang Q, von Knoten v_i nach Knoten v_j, der mit einer F-Kante beginnt und mit einer $(E \setminus F)$-Kante endet, wobei $v_i \neq v_j$ ist und es gilt*

$$l(d_H + \lambda_Q) < l(d_H).$$

2. *Verbesserungsspaziergang zweiter Art sind zwei ungerade alternierende, kantendisjunkte Spaziergänge $P^+ = (v_i, ..., v_{i'})$ und $P^- = (v_j, ..., v_{j'})$ mit $v_i, v_{i'} \neq v_j, v_{j'}$, wobei die Endknoten von P^+ mit F-Kanten und die von P^- mit $(E \setminus F)$-Kanten inzidieren und es gilt*

$$l(d_H + \lambda_{P^+} + \lambda_{P^-}) < l(d_H).$$

Für unser Minsquare-Problem erkennt man durch leichtes Nachrechnen für Verbesserungsspaziergänge erster Art die Bedingung $d_H(v_i) + 1 < d_H(v_j)$ und für Verbesserungsspaziergänge zweiter Art die Bedingung $d_H(v_j) + d_H(v_{j'}) + 2 < d_H(v_i) + d_H(v_{i'})$, wenn $v_j \neq v_{j'}$ und $v_i \neq v_{i'}$ ist. Natürlich sind bei Verbesserungsspaziergängen zweiter Art auch Kreise zugelassen, denn in solch einem Kreis gibt es einen Knoten, der mit zwei F bzw. mit zwei $E \setminus F$-Kanten inzidiert. Somit wird nach Kantentausch der Knotengrad von v um $+2$ oder -2 verändert.

Hilfssatz 2.2.5. *Seien $F, F_0 \subseteq E$ mit $|F| = |F_0|$. $F \triangle F_0 := (F \setminus F_0) \cup (F_0 \setminus F)$ bezeichne die symmetrischen Differenz. Bezeichne $E(S_\nu)$ die Kantenmenge eines $(F - E \setminus F)$-alternierenden Spazierganges S_ν. Dann gilt:*

1. *Es gibt eine minimale Anzahl s kantendisjunkter Spaziergänge S_ν, für welche $F \triangle F_0 = (\cup_{\nu=1}^{s} E(S_\nu))$ ist.*

2. *Alle Spaziergänge aus 1. sind kompatibel, d.h. Spaziergänge S_ν, die mit dem festen Knoten v_i starten, beginnen alle mit einer F-Kante bzw. alle mit einer $(E \setminus F)$-Kante in v_i. Insbesondere heißt das $\lambda_{S_\nu}(v_i) \geq 0$ für alle S_ν oder $\lambda_{S_\nu}(v_i) \leq 0$ für alle S_ν.*

3. *Die Spaziergänge aus 1. bestehen aus t geraden alternierenden Spaziergängen Q_j mit $j \in \{1, \ldots, t\}$ und aus t' Paaren von ungeraden alternierenden Spaziergängen P_i^+, P_i^- mit $i \in \{1, \ldots, t'\}$, wobei die Endknoten der P_i^+ mit F-Kanten und die Endknoten der P_i^- mit $E \setminus F$-Kanten inzidieren.*

Beweis.

1. Eine Menge mit minimaler Anzahl $(F - E \setminus F)$-alternierender, disjunkter Spaziergänge, die genau aus den Kanten von $F \triangle F_0$ besteht, muss existieren, denn bereits alle Kanten aus $F \triangle F_0$ bilden eine Menge kantendisjunkter, alternierender Spaziergänge. (Die F_0-Kanten sind die $E \setminus F$-Kanten in $F \triangle F_0$.)

2. Angenommen, es existieren zwei Spaziergänge S und S', die beide mit dem Knoten v_i beginnen, wobei v_i in S mit einer F-Kante und in S' mit einer $(E \setminus F)$-Kante inzidiert. Dann ist die Vereinigung von S und S' ein alternierender Spaziergang. Widerspruch zur Minimalität von s in 1.!

3. Da F und F_0 die gleiche Mächtigkeit besitzen, folgt, dass in $F \triangle F_0$ die Anzahl der alternierenden Spaziergänge, deren Enden mit F-Kanten beginnen und enden, gleich der Anzahl der alternierenden Spaziergänge ist, die mit $(E \setminus F)$-Kanten beginnen und enden.

□

Folgerung 2.2.6. *Für die Gradsequenzen d_H, d_{H_0} von $H = (V, F)$ und $H_0 = (V, F_0)$ gilt*

$$d_{H_0} = \sum_{\nu=1}^{s} \lambda_{S_\nu} + d_H.$$

Beweis. Wegen $F \triangle F_0 = (\cup_{\nu=1}^{s} \Pi_\nu)$ und der Assoziativität der symmetrischen Differenz folgt $F_0 = F \triangle (F \triangle F_0) = F \triangle (\cup_{\nu=1}^{s} E(S_\nu))$.Also gilt:

$$\begin{aligned}
d_{H_0} &= d_{H \triangle (\cup_{\nu=1}^{s} S_\nu)} \\
&= d_{H \setminus (\cup_{\nu=1}^{s} S_\nu)} + d_{(\cup_{\nu=1}^{s} S_\nu) \setminus H} \\
&= d_{H \setminus (\cup_{\nu=1}^{s} S_\nu)} + d_{(\cup_{\nu=1}^{s} S_\nu) \setminus H} + d_{H \cap (\cup_{\nu=1}^{s} S_\nu)} - d_{(\cup_{\nu=1}^{s} S_\nu) \cap H} \\
&= (d_{(\cup_{\nu=1}^{s} S_\nu) \setminus H} - d_{(\cup_{\nu=1}^{s} S_\nu) \cap H}) + d_H = (d_{\cup_{\nu=1}^{s} (S_\nu \setminus H)} - d_{\cup_{\nu=1}^{s} (S_\nu \cap H)}) + d_H \\
&= \sum_{\nu=1}^{s} (d_{S_\nu \setminus H} - d_{S_\nu \cap H}) + d_H
\end{aligned}$$

$$= \sum_{\nu=1}^{s} \lambda_{S_\nu} + d_H.$$

□

Existenzsatz 2.2.7. *Sei $H = (V, F)$ ein Teilgraph von $G = (V, E)$ mit $|F| = k$. Genau dann existiert ein Verbesserungsspaziergang erster Art oder ein Verbesserungsspaziergang zweiter Art, wenn $l(d_H)$ nicht minimal ist.*

Beweis.
$\Longrightarrow$: Klar.

$\Longleftarrow$: Weil F nicht optimal ist, existiert ein optimaler Teilgraph $H_0 = (V, F_0)$ von $G = (V, E)$ und es gilt wegen 2.2.6 und 2.2.5.2.

$$\begin{aligned}
0 &> l(d_{H_0}) - l(d_H) \\
&= \sum_{v\in V} l_v(d_{H_0}(v)) - \sum_{v\in V} l_v(d_H(v)) \\
&= \sum_{v\in V} l_v((d_H(v) + \sum_{\nu=1}^{s} \lambda_{S_\nu}(v))) - \sum_{v\in V} l_v(d_H(v)). \\
&= \sum_{v\in V} l_v((d_H(v) + \sum_{j=1}^{t} \lambda_{Q_j}(v) + \sum_{i=1}^{t'} \lambda_{P_i^+\cup P_i^-}(v))) - \sum_{v\in V} l_v(d_H(v)). \\
&\geq \sum_{v\in V}\sum_{j=1}^{t}(l_v(d_H(v) + \lambda_{Q_j}(v) - l_v(d_H(v)) + \sum_{v\in V}\sum_{i=1}^{t'}(l_v(d_H(v) + \\
&\quad \lambda_{P_i^+\cup P_i^-}(v)) - l_v(d_H(v))) \\
&= \sum_{j=1}^{t}(\sum_{v\in V}(l_v(d_H(v) + \lambda_{Q_j}(v)) - l_v(d_H(v)))) + \sum_{i=1}^{t'}(\sum_{v\in V}(l_v(d_H(v) + \\
&\quad \lambda_{P_i^+\cup P_i^-}(v)) - l_v(d_H(v)))) \\
&= \sum_{j=1}^{t}(l(d_H + \lambda_{Q_j}) - l(d_H)) + \sum_{i=1}^{t'}(l(d_H + \lambda_{P_i^+\cup P_i^-}) - l(d_H))
\end{aligned}$$

Nach 2.2.4 gibt es einen Verbesserungsspaziergang zweiter oder erster Art.
□

Wir können durch die Suche nach Verbesserungsspaziergängen eine nicht optimale Kantenmenge F schrittweise verbessern. Verbesserungsspaziergänge lassen sich auf verschiedene Weise in polynomieller Zeit finden, was aber aufgrund der speziellen Situation des paarweisen Auftretens von alternierenden Spaziergängen nicht einfach ist. Im Minsquare-Problem wären wegen $\sum_{v\in V} d_H(v)^2 \leq$

$\sum_{v \in V} d_G(v)^2 \leq |V|(|V|-1)^2$ größenordnungsmäßig $O(|V|^3)$ Verbesserungsspaziergänge nötig, um einen optimalen Teilgraphen H zu finden. Das Minsquare-Problem lässt sich demnach in polynomieller Zeit lösen. Da jedoch die Anzahl der zu suchenden Verbesserungspfade offensichtlich von der diskreten, konvexen Funktion abhängt, kann es beim Lösen dieses Problems zu erheblichen Schwankungen der Zeitschranken kommen, bis dahingehend, dass ein Problem nicht in polynomieller Zeit lösbar ist. Um eine Unabhängigkeit von den jeweiligen konvexen Funktionen zu erreichen, ist demnach der Versuch naheliegend, alle alternierenden Pfade aus $F \triangle F_0$ zu finden, denn deren Anzahl kann den Wert $2|F|$ nicht überschreiten. Das ist bisher nicht gelungen.
Unser Ansatz vermeidet die Suche nach Verbesserungsspaziergängen. Stattdessen reduzieren wir das Problem auf ein klassisches Problem der Graphentheorie, auf die Suche nach einem maximal gewichteten f-Faktor.

2.3 Die schnelle Lösung - Der gewichtete Graph

Die Hauptidee der nachfolgenden Konstruktion ist es, die Bedingungen $|F| = k$ und $\sum_{v \in V} l_v(d_H(v)) = min$, die an einen gesuchten Teilgraphen $H = (V, F)$ von G gestellt werden, in einem Hilfsgraphen so unterzubringen, dass man nach einem konkreten f-Faktor, also nach einem Teilgraphen mit einer festgelegten Knotensequenz suchen kann. Das ist in G nicht der Fall. Wir kennen zwar die Anzahl der Kanten des gesuchten Teilgraphen, jedoch nicht die Knotensequenz eines solchen.
Wir konstruieren einen Hilfsgraphen $G' = (V \cup \{x\}, E \cup E^x)$ durch Hinzufügen eines Knotens x zu G, der zu allen $v \in V$ durch $d_G(v)$ parallele Kanten $e_{v,i}$ mit $1 \leq i \leq d_G(v)$ adjazent ist. Weiterhin setzen wir

$$E^x := \bigcup_{v \in V} \{e_{v,i} | 1 \leq i \leq d_G(v)\}.$$

Wir definieren eine Kantengewichtsfunktion auf der Kantenmenge $E' := E \cup E^x$ durch

$$w(e) := \begin{cases} 0 & \text{wenn } e \in E \\ l_v(i) - l_v(i-1) & \text{wenn } e = e_{v,i} \in E^x. \end{cases}$$

Weiterhin definieren wir eine Funktion $f : V \cup \{x\} \to \mathbb{N}_0$ auf der Knotenmenge von G' vermöge

$$f(v) := \begin{cases} 2|E| - 2k & \text{wenn } v = x \\ d_G(v) & \text{wenn } v \in V. \end{cases}$$

Wir bezeichnen einen f-Faktor $H' = (V \cup \{x\}, F')$ von G' als *G'-passend*, wenn

$$\forall v \in V \text{und i mit } \forall 1 \leq i \leq d_G(v) - 1 : (e_{v,i} \in F' \Rightarrow e_{v,i+1} \in F').$$

Bevor wir uns den entscheidenen Verbindungen zu unserem Minkonvex-Problem zuwenden, betrachten wir ein Beispiel.

Beispiel 2.3.1. *Wir betrachten den Sympathiegraphen G auf Seite 3 und konstruieren daraus den Hilfsgraphen G'.*

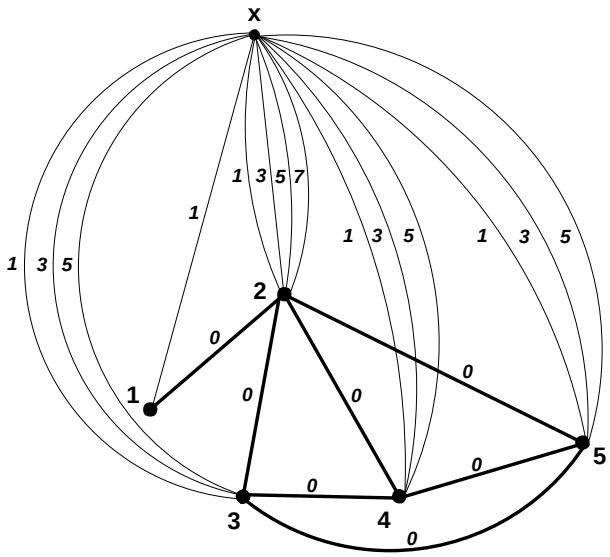

Abbildung 5: Hilfsgraph G'

Es sei nun ein beliebiger f-Faktor von G' gegeben. Betrachten wir die Kantenmenge F des f-Faktors, die durch die Knotenmenge V induziert wird, also den Teilgraphen $H = (V, F)$. Ein beliebiger Knoten v muss mit $d_H(v) \leq d_G(v)$ Kanten inzidieren, wobei $d_H(v)$ die Anzahl der zu v inzidenten F-Kanten bezeichnet. Deren Kantengewichte sind per Definition alle Null. Aufgrund der Knotenbedingung $f(v) = d_G(v)$ müssen weitere $(d_G(v) - d_H(v))$ Kanten von v mit x inzidieren. Wählen wir solche mit den größten Kantengewichten aus, das sind aufgrund der Konvexität von l_v die Kanten $e_{v,i}$ mit $d_H(v) < i \leq d_G(v)$, so ergibt die Kantensumme $l_v(d_G(v)) - l_v(d_H(v))$. Da der erste Summand konstant ist, gelingt es uns, den konvexen Funktionswert des Knoten v im Teilgraphen H mit der Kantenmenge F darzustellen.

Lemma 2.3.2. *Ist $H' = (V \cup \{x\}, F')$ ein f-Faktor von G', dann wird durch die Kantenmenge $F := E \cap F'$ ein Teilgraph $H = (V, F)$ von G mit $|F| = k$ bestimmt. Ist andererseits ein Teilgraph $H = (V, F)$ von G mit $|F| = k$ gegeben, dann wird durch*

$$F' : \quad = \quad F \bigcup_{v \in V} \left\{ e_{v,i} | d_H(v) + 1 \leq i \leq d_G(v) \right\}$$

ein passenden f-Faktor $H' = (V \cup \{x\}, F')$ von G' bestimmt.

Für einen passenden f-Faktor H' von G' gilt

$$l(d_H) \quad = \quad \sum_{e\in E'} w(e) - \sum_{e\in F'} w(e) + \sum_{v\in V} l_v(0) = w(G') - w(H') + \sum_{v\in V} l_v(0).$$

Beweis. Ist H' ein f-Faktor von G', dann bestimmt die Kantenmenge $F := E \cap F'$ einen Teilgraphen $H = (V, F)$ von G mit

$$\sum_{v\in V} d_H(v) \quad = \quad \sum_{v\in V} d_G(v) - (2|E| - 2k) = 2k.$$

Somit gilt $|F| = k$.
Sei nun ein Teilgraph $H = (V, F)$ von $G = (V, E)$ mit $|F| = k$ gegeben. Dann gilt für den Knotengrad des Graphen $H' = (V \cup \{x\}, F')$, wobei F' wie oben definiert ist,

$$\begin{aligned} d_{H'}(x) &= \sum_{v\in V}(d_G(v) - d_H(v)) = 2|E| - 2k \\ d_{H'}(v) &= d_H(v) + (d_G(v) - d_H(v)) = d_G(v) \text{ für alle } v \in V. \end{aligned}$$

Somit ist H' ein G'-passender f-Faktor.
Sei nun H' ein G'-passender f-Faktor, dann gilt

$$\begin{aligned} w(G') - w(H') &= \sum_{e\in E'} w(e) - \sum_{e\in F'} w(e) \\ &= \sum_{v\in V} \sum_{i=1}^{d_G(v)} (l_v(i) - l_v(i-1)) - \sum_{v\in V} \sum_{i=d_H(v)+1}^{d_G(v)} (l_v(i) - l_v(i-1)) \\ &= \sum_{v\in V} \sum_{i=1}^{d_H(v)} (l_v(i) - l_v(i-1)) \\ &= l(d_H) - \sum_{v\in V} l_v(0). \end{aligned}$$

□

Hauptsatz 2.3.3. *Eine Kantenmenge $F \subseteq E$ mit $|F| = k$ eines Teilgraphen $H = (V, F)$ von $G = (V, E)$ minimiert genau dann*

$$l(d_H) := \sum_{v\in V} l_v(d_H(v)),$$

wenn $H' = (V \cup \{x\}, F')$ ein f-Faktor auf $G' = (V \cup x, E')$ ist, der $w(H')$ maximiert.

Beweis. Ist F eine Kantenmenge, die aus k Kanten besteht, dann entsteht nach Konstruktion von F' in 2.3.2 ein passender f-Faktor $H' = (V \cup \{x\}, F')$ von G' mit $w(H') = w(G') - l(d_H) + \sum_{v \in V} l_v(0)$, der maximal ist, weil $l(d_H)$ minimal ist. Sei nun $H' = (V \cup \{x\}, F')$ ein f-Faktor, der $w(H')$ maximiert. Nach 2.3.2 wird durch $F = E \cap F'$ ein Teilgraph $H = (V, F)$, der aus k Kanten besteht, bestimmt. Weil die l_v diskrete konvexe Funktion sind, gilt

$$\forall i \in \mathbb{N} \text{ mit } i \geq 1 : l_v(i) - l_v(i-1) \leq l_v(i+1) - l_v(i).$$

Somit ist jedes $v \in V$ in H' mit dem Knoten x durch die Kanten $e_{v,i}$ mit $d_H(v) + 1 \leq i \leq d_G(v)$ verbunden, denn sonst gäbe es einen Widerspruch zur Maximalität von $w(H')$. Dabei ist zu beachten, dass mögliche parallele Kanten gleichen Gewichts mit kleinerem Index ausgetauscht werden können. Somit ist H' ein passender f-Faktor und demnach wird $l(d_H)$ nach 2.3.2 minimiert. □

Es genügt also einen maximal gewichteten f-Faktor $H' = (V \cup \{x\}, F')$ von $G' = (V \cup \{x\}, E')$ zu bestimmen. Vermöge $F := E \cap F'$ erhalten wir daraus einen minimalen Teilgraphen $H = (V, F)$ von G mit k Kanten. Lösen wir unser Gutscheinproblem mit dieser Vorgehensweise.

Beispiel 2.3.4. *Folgender fettgezeichneter Teilgraph ist ein maximal gewichteter f-Faktor von G' aus Beispiel 2.3.1 auf Seite 14. Betrachten wir nur die Kantenmenge, die durch die Knotenmenge V induziert wird, so erhalten wir eine optimale Lösung des Gutscheinproblems.*

Wie sich ein maximal-gewichteter f-Faktor bestimmen lässt, wird Gegenstand des nächsten Kapitels sein. Die parallelen Kanten in unserem Hilfsgraphen erweisen sich jedoch als ein Problem. Diese beseitigen wir durch eine weitere Modifikation von G', indem wir alle Kanten, die mit dem Knoten x inzidieren, durch je zwei weitere Knoten in drei Kanten unterteilen.
Wir konstruieren $G'' = (V \cup \{x\} \cup V^{(12)}, E \cup E_1^x \cup E_2^x \cup E_3^x)$ mit

$$\begin{aligned}
V^{(12)} &:= \bigcup_{v \in V} \{v_{(i)}^1, v_{(i)}^2 | 1 \leq i \leq d_G(v)\} \\
E_1^x &:= \bigcup_{v \in V} \{e_{v,i}^{(1)} | e_{v,i}^{(1)} = \{v, v_{(i)}^1\} \text{ mit } 1 \leq i \leq d_G(v)\} \\
E_2^x &:= \bigcup_{v \in V} \{e_{v,i}^{(2)} | e_{v,i}^{(2)} = \{v_{(i)}^1, v_{(i)}^{(2)}\} \text{ mit } 1 \leq i \leq d_G(v)\} \\
E_3^x &:= \bigcup_{v \in V} \{e_{v,i}^{(3)} | e_{v,i}^{(3)} = \{v_{(i)}^2, x\} \text{ mit } 1 \leq i \leq d_G(v)\}.
\end{aligned}$$

Wir definieren eine Kantengewichtsfunktion w' auf der Kantenmenge $E'' := E \cup E_1^x \cup E_2^x \cup E_3^x$ durch

$$w'(e) := \begin{cases} 0 & \text{wenn } e \in E \text{ und } e \in E_2^x \\ \frac{1}{2}(l_v(i) - l_v(i-1)) & \text{wenn } e \in E_1^x \text{ und } e \in E_3^x \end{cases}$$

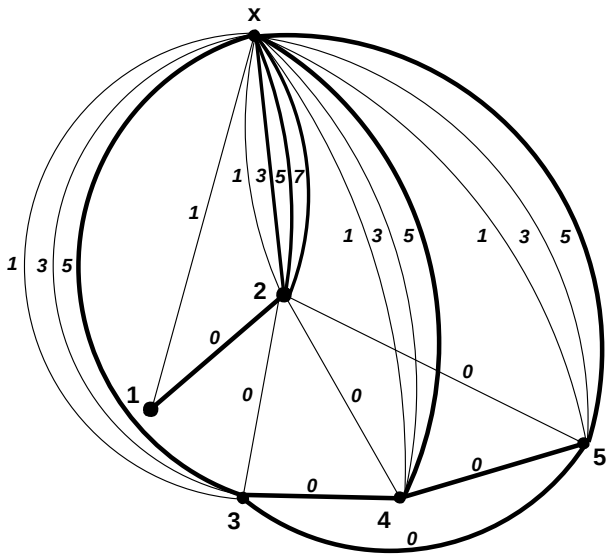

Abbildung 6: Die fetten Kanten bezeichnen einen maximal-gewichteten f-Faktor von G'.

Weiterhin definieren wir eine Funktion $f' : V \cup \{x\} \cup V^{(12)} \to \mathbb{N}_0$ auf der Knotenmenge von G'' vermöge

$$f'(v) := \begin{cases} 2|E| - 2k & \text{wenn } v = x \\ d_G(v) & \text{wenn } v \in V \\ 1 & \text{wenn } v \in V^{(12)}. \end{cases}$$

Die Definition eines passenden f-Faktors können wir übernehmen. Wir bezeichnen einen f'-Faktor $H'' = (V \cup \{x\} \cup V^{(12)}, F'')$ von G'' als G''-*passend*, wenn

$$\forall v \in V \text{und für } i \in \mathbb{N} \text{ mit } 1 \leq i \leq d_G(v) - 1 : (e^{(1)}_{v,i} \in F'' \Rightarrow e^{(1)}_{v,i+1} \in F'').$$

Aufgrund der Definition des f'-Faktors folgt aus $e^{(1)}_{v,i} \in F''$, dass $e^{(3)}_{v,i} \in F''$ und damit auch $e^{(3)}_{v,i+1} \in F''$ gilt.

Beispiel 2.3.5. *Wir betrachten Beispiel 2.3.1 auf Seite 14 und konstruieren den Graphen G''.*

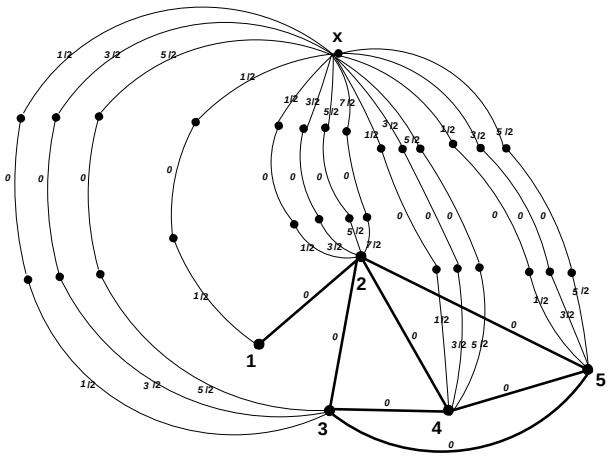

Abbildung 7: Der Hilfsgraph G''

Lemma 2.3.6. *Fordern wir*

$$\begin{aligned} e \in F'' \cap E &\Leftrightarrow e \in F' \cap E \\ (e_{v,i}^{(1)}, e_{v,i}^{(3)} \in F'') &\Leftrightarrow e_{v,i} \in F', \end{aligned}$$

dann ist $H'' = (V \cup \{x\} \cup V^{(12)}, F'')$ genau dann ein f'-Faktor von G'', der $w'(H'')$ maximiert, wenn $H' = (V \cup \{x\}, F')$ ein f-Faktor auf $G' = (V \cup \{x\}, E')$ ist, der $w(H')$ maximiert.

Beweis. f ist G'-passend, siehe Beweis 2.3.3. Analog folgt, dass f' wegen der Maximalität und Konvexität G''-passend ist. Außerdem ist klar, dass $d_{H'}(v) = d_{H''}(v)$ für alle $v \in V \cup \{x\}$ gilt. Es folgt

$$\begin{aligned} w'(H'') &= \sum_{e \in F''} w'(e) \\ &= \sum_{v \in V} \left(\sum_{i=d_{H''}(v)+1}^{d_G(v)} (w'(e_{v,i}^{(1)}) + w'(e_{v,i}^{(3)})) \right) \\ &= \sum_{v \in V} \left(\sum_{i=d_{H''}(v)+1}^{d_G(v)} (l_v(i) - l_v(i-1)) \right) \\ &= \sum_{v \in V} \left(\sum_{i=d_{H'}(v)}^{d_G(v)} (l_v(i) - l_v(i-1)) \right) \\ &= \sum_{e \in F'} w(e) \\ &= w(H'). \end{aligned}$$

Also folgt die Behauptung. □

Satz 2.3.7. *Eine Kantenmenge $F \subseteq E$ mit $|F| = k$ eines Teilgraphen $H = (V, F)$ von $G = (V, E)$ minimiert genau dann*

$$l(d_H) := \sum_{v \in V} l_v(d_H(v)),$$

wenn $H'' = (V \cup \{x\} \cup V^{(12)}, F'')$ ein f'-Faktor von G'' ist, der $w'(H'')$ maximiert.

Beispiel 2.3.8. *Folgender fettgezeichneter Teilgraph ist ein maximal gewichteter f'-Faktor von G'' aus 2.3.5 auf Seite 18. Betrachten wir nur die Kantenmenge, die durch die Knotenmenge V induziert wird, so erhalten wir eine optimale Lösung des Gutscheinproblems.*

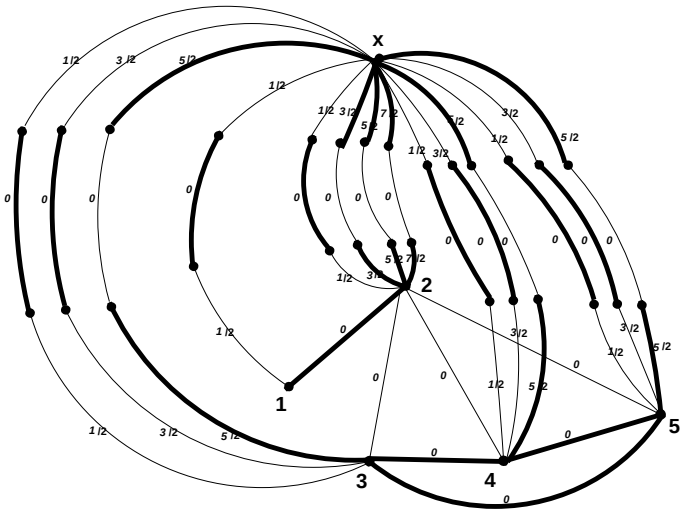

Abbildung 8: Ein minimal gewichteter f'-Faktor von G''

Wir haben also einen schlichten Hilfsgraphen konstruiert, für den ein minimal gewichteter f'-Faktor zu bestimmen ist. Allerdings kostet diese Konstruktion etwas. Für die Kantenmenge E'' gilt $|E''| = 7|E|$ und für die Knotenmenge V'' folgt $|V''| = |V| + 1 + 4|E|$. Der schnellstmögliche, uns bekannte Algorithmus zur Bestimmung eines minimal-gewichteten f'-Faktors besitzt nach [12] eine Komplexität von $O(\sum_{v \in V} f'(v)|E''|log|V''|)$. Wir erhalten demnach zur Lösung des Minkonvex-Problems eine Komplexität von $O(|E|^2 log|V|)$. Mehr dazu folgt im nächsten Kapitel.

3 Die Bestimmung eines maximal-gewichteten f-Faktors

Für die Bestimmung eines maximal-gewichteten f-Faktors gibt es zwei uns bekannte Herangehensweisen. Die erste Möglichkeit benutzt den Algorithmus von Gabow [9], der in seinen einzelnen Bestandteilen in den nächsten Abschnitten behandelt wird. Die zweite Möglichkeit bedient sich balancierter Netzwerke und ist nachzulesen in den Arbeiten von Jungnickel und Fremuth-Praeger. Das gewichtete f-Faktor Problem wird reduziert auf die Bestimmung einer balancierten Zirkulation mit minimalen Kosten auf einem balancierten Netzwerk $N = (V_N, E_N)$. Mit V_N bezeichnen wir die Knotenmenge und mit E_N die gerichtete Kantenmenge des Netzwerkes N. Die Konstruktion des Netzwerkes N lässt sich nachlesen in [6]. Auf dem Netzwerk N bestimmen wir zuerst eine Zirkulation mit minimalen Kosten und verändern diese schrittweise mit einem so genannten erweiterten primal-dualen Algorithmus [7] zu einer balancierten Zirkulation mit minimalen Kosten. Da parallele Kanten die Konstruktion stören würden, benötigen wir den Hilfsgraphen G'' aus dem letzten Abschnitt. Der erweiterte primal-duale Algorithmus benötigt eine Laufzeit von $O(|V_N||E_N|)$ und muss V_N mal durchgeführt werden. Bezeichnet $L(|V_N|, |E_N|)$ die Laufzeit zur Bestimmung einer minimalen Kostenzirkulation, so ergibt sich insgesamt zur Berechnung einer balancierten minimalen Kostenzirkulation eine Komplexität von

$$O(L(|V_N|, |E_N|) + |V_N|^2|E_N|).$$

Die uns bekannte beste Laufzeit von $L(|V_N|, |E_N|)$ beträgt nach Ahuja [1] $O(|E_N|^2 log|(V_N)| + |E_N|(log(|V_N|)^2))$. Insgesamt ergibt sich für unser gewichtetes f-Faktorproblem, und somit auch für unser Minkonvex Problem eine Laufzeit von

$$O(|E|^2 log(|V|) + |E|^3).$$

Fremuth-Paeger und Jungnickel geben einen Ausblick [7] auf eine Verbesserung der Komplexität der Laufzeit des erweiterten primal-dualen Algorithmus durch besondere Implementierungen auf $O(|E_N|^2 log|(V_N)|$. Für unser Minkonvex Problem würde das zu einer Laufzeitkomplexität von $O(|E|^2 log(|V|)$ führen. Es lohnt sich offenbar, diesen Weg im Auge zu behalten. Wir konzentrieren uns im Weiteren jedoch auf die andere Möglichkeit zur Bestimmung eines maximal-gewichteten f-Faktors. Die Ideen dazu stammen von Urquart und Gabow [8] und beruhen letztlich auf der Bestimmung eines maximal-gewichteten perfekten Matchings.
An dieser Stelle wollen wir einen Spezialfall, nämlich das Minsquare Problem auf einem bipartiten Graphen erwähnen. Dieser ist deswegen interessant, weil wir für einen bipariten Graphen eine abgewandelte Form der Reduktion auf ein

maximal gewichtetes f-Faktor Problem vorschlagen. Wir bewahren die bipartite Struktur auf dem Hilfsgraphen G', indem wir zwei Knoten x, x' zu G hinzufügen, je einen zu einer unabhängigen Knotenmenge. Die Knoten v der jeweils anderen unabhängigen Knotenmenge werden durch $d_G(v)$ parallele Kanten mit diesem Knoten verbunden und die Gewichte wie üblich gesetzt. Dann fordern wir für den f-Faktor $f(x) = f(x') = |E| - k$. Der Hilfsgraph bleibt bipartit. Das gewichtete f-Faktor Problem lässt sich auf einem bipartiten Graphen zu einem anderen Problem, nämlich zur Bestimmung eines maximalen Flusses mit minimalen Kosten auf einem Netzwerk reduzieren. Dabei stören allerdings parallele Kanten. Diese beseitigen wir nicht wie im letzten Kapitel durch Konstruktion des Graphen G''. Stattdessen splitten wir die zwei zusätzlichen Knoten x, x' in je $|E| - k$ Knoten x_i und x'_i auf, die alle mit einem weiteren Knoten y bzw. y' durch Kanten verbunden werden. Dann suchen wir einen gewichteten f'-Faktor auf dieser Konstruktion, wobei wir fordern $f'(y) = f'(y') = k$ und $f'(v) = 1$ für alle $v = x_i$ oder $v = x'_i$. Wieder gilt für die Komplexität der Knotenmenge $|V''| \in O(|E|)$. Die Bestimmung eines maximalen Flusses mit minimalen Kosten benötigt nach [1] die Zeit $O(|E|^2 log(|V|))$. Im bipartiten Fall haben wir die Möglichkeit, über Netzwerkflüsse unser Minsquare und unser Minkonvex Problem schnellstmöglich zu lösen.

3.1 Das Knotensplitting von Tutte

In der folgenden Konstruktion ersetzen wir jeden Knoten eines Graphen durch einen vollständigen bipartiten Graphen. Diese Vorgehensweise nennt sich Knotensplitting und wurde bereits 1954 von Tutte [13] entdeckt. Sie ermöglicht es, unser gewichtetes f-Faktor Problem auf die Bestimmung eines gewichteten 1-Faktors, d.h. eines gewichteten f-Faktors, für den $f(v) = 1$ für alle Knoten v gilt, zu reduzieren.
Sei $G = (V, E)$ ein schleifenloser Graph, der parallele Kanten enthalten darf, mit einer Kantengewichtsfunktion $w : E \to \mathbb{R}$ und einer Funktion $f : V \to \mathbb{N}$ mit $f(v) \leq d_G(v)$ für alle $v \in V$. Wir konstruieren einen Splitgraphen $G_f = (V_f, E_f)$, indem wir jeden Knoten $v \in V$ durch einen bipartiten vollständigen Graphen $G_v = (V_v \cup W_v, E_v)$ ersetzen mit

$$\begin{aligned} V_v &:= \{v_{1,v}, \ldots, v_{d_G(v),v}\} \\ W_v &:= \{w_{1,v}, \ldots, w_{d_G(v)-f(v),v}\}. \end{aligned}$$

Die Knoten aus V_v bezeichnen wir als *innere Knoten* und die Knoten in W_v als *äußere Knoten*. Die Kanten in E_v nennen wir *äußere Kanten*. *Innere Kanten* nennen sich die Kanten, die innere Knoten aus verschiedenen Knotenmengen $V_{v'}$ miteinander verbinden und zwar so, dass jeder innere Knoten mit genau einer inneren Kante inzidiert. Diese werden wie folgt gesetzt. Sind in G zwei Knoten v, v' durch s parallele Kanten $e^{(i)}_{v,v'}$ mit $1 \leq i \leq s$ miteinander verbunden, so werden s innere Knoten aus $V_{v'}$ mit s inneren Knoten aus V_v durch s innere

Kanten $\eta^{(i)}_{v,v'}$ miteinander verbunden. Wir nennen Kanten $e^{(i)}_{v,v'}$ und $\eta^{(i)}_{v,v'}$ *zueinander assoziiert*. Nun definieren wir eine Kantengewichtsfunktion $w_f : E_f \to \mathbb{R}$ vermöge

$$w_f(\eta) := \begin{cases} W := \max_{e \in E}\{w(e)\} & \text{wenn } \eta \text{ äußere Kante von} E_f \\ w(e^{(i)}_{v,v'}) & \text{wenn } \eta = \eta^{(i)}_{v,v'}. \end{cases}$$

Beispiel 3.1.1. *Wir betrachten unser gewichtetes f-Faktorproblem aus Beispiel 2.3.4 und konstruieren den Splitgraphen. Die fettgezeichneten Kanten kennzeichnen im oberen Graphen einen maximalen f-Faktor und im Splitgraphen darunter einen* 1*-Faktor.*

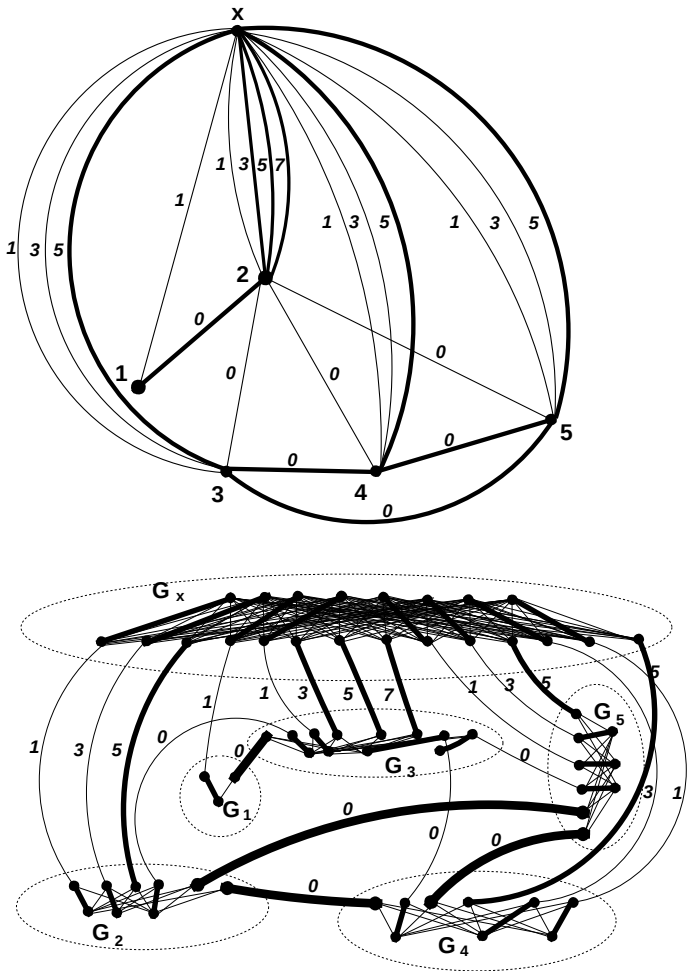

Abbildung 9: Der Splitgraph G_f zum Graphen G mit f-Faktor

Anhand der Abbildung lässt sich sehr gut erkennen, dass ein f-Faktor in G einem 1-Faktor in G_f entspricht. Der folgende Satz nach Tutte beweist diese Tatsache.

Satz 3.1.2. *Sei $G = (V, E)$ und G_f wie oben definiert. Besitzt G einen maximal gewichteten f-Faktor $H = (V, F)$, dann bestimmen die F-Kanten assoziierte innere Kanten M_I auf G_f, die so durch äußere Kanten M_A ergänzt werden können, dass $M := M_I \cup M_A$ in G_f einen maximal gewichteten 1-Faktor $H_f = (V_f, M)$ bildet. Umgekehrt bestimmt ein maximal gewichteter 1-Faktor $H_f = (V_f, M)$ von G_f durch alle inneren Kanten aus M die assozierte Kantenmenge $F \subseteq E$. Diese bildet den maximal gewichteten f-Faktor $H = (V, F)$ auf G.*

Beweis. Sei G ein maximaler f-Faktor $H = (V, F)$. Die innere Kantenmenge M_I besteht aus den zu F assozierten Kanten. Somit sind $f(v)$ innere Knoten in V_v inzident zu einer Kante aus M_I für jedes $v \in V$. Nach Definition verbleiben $d_G(v) - f(v)$ Knoten aus V_v übrig. Diese können wir, weil G_v ein vollständiger bipartiter Graph ist, mit äußeren Knoten so verbinden, dass jeder äußere Knoten in W mit genau einer äußeren Kante inzidiert. Somit wird ein 1-Faktor $H_f = (V_f, M)$ bestimmt. Angenommen dieser ist nicht maximal. Dann können die inneren Kanten des 1-Faktors nicht maximal sein, da alle äußeren Kanten konstantes Gewicht besitzen. Das heißt, die assozierten Kanten in G bilden keinen maximal gewichteten f-Faktor. Widerspruch!
Umgekehrt wird durch einen maximalen 1-Faktor $H_f = (V_f, M)$ durch die zu den inneren Kanten aus M_I assoziierten Kanten F ein f-Faktor $H = (V, F)$ bestimmt. Wäre dieser nicht maximal, so wäre auch der 1-Faktor nicht maximal. □

Unser Problem, die Berechnung eines maximalen f-Faktors, ist somit auf die Berechnung eines maximalen 1-Faktors auf G_f reduziert wurden. Leider erhöht diese Konstruktion die Größenordnung unserer Knoten- bzw. Kantenmengen. Es gelten

$$\begin{aligned} |V_f| &= \sum_{v \in V} d_G(v) + \sum_{v \in V} (d_G(v) - f(v)) = 4|E| - 2|F| \\ |E_f| &= |E| + \sum_{v \in V} d_G(v)(d_G(v) - f(v)) \leq |E| + 2|V||E|. \end{aligned}$$

Somit folgen $V_f \in O(|E|)$ und $E_f \in O(|E||V|)$. Das heißt, die Komplexität unseres Problems erfährt eine beträchtliche Vergrößerung. Im übernächsten Abschnitt umgehen wir dieses Problem mit einem Trick, der von Gabow [8] stammt. Er löst Teilschritte der Berechnung eines 1-Faktors auf Suchgraphen, deren Kantenmenge in $O(|E|)$ liegt, und verringert damit die Komplexität der Berechnung eines gewichteten 1-Faktors. Vorerst beschäftigen wir uns jedoch mit der Berechnung eines ungewichteten 1-Faktors.

3.2 Die Berechnung eines ungewichteten 1-Faktors

In diesem Abschnitt betrachten wir die Vorgehensweise zur Berechnung eines 1-Faktors. Dabei stützen wir uns auf die Betrachtungen in [12]. Um einen f-Faktor für einen Graphen G zu berechnen, können wir einen Splitgraphen G_f konstruieren und für diesen einen 1-Faktor berechnen, wie wir aus dem letzten Abschnitt wissen. Die Herangehensweise erinnert an den klassischen Ansatz, den wir in 2.2 für unser Minkonvex-Problem betrachtet haben. Hier benötigen wir alternierende Pfade. Der besseren Überschaubarkeit wegen betrachten wir das 1-Faktor Problem auf einem Graphen $G = (V, E)$ und modifizieren den Algorithmus erst später für unseren Splitgraphen G_f. Da wir bei unserem Problem von der Existenz eines 1-Faktors wissen, betrachten wir den Fall, dass ein solcher nicht existiert, nicht. Es ist klar, dass ein 1-Faktor genau $\frac{|V|}{2}$ Kanten besitzen muss. Die grobe Vorgehensweise ist Folgende. Wir betrachten einen Teilgraphen $H = (V, M)$, wobei für Knotengrade $d_H(v) = 0$ bzw. $d_H(v) = 1$ gelten, also $M < \frac{|V|}{2}$ folgt. Solch einen Teilgraphen H bezeichnen wir im folgenden als $(0,1)$-*Faktor* von G. Finden wir einen alternierenden Pfad $P = (v, \ldots, v')$ ungerader Länge dessen Kanten abwechselnd in M und in $E \setminus M$ liegen, wobei $d_H(v) = 0$ und $d_H(v') = 0$ gelten, und vertauschen die M- und $E \setminus M$-Kanten miteinander, dann erhalten wir einen $(0,1)$-Faktor von G, der eine Kante mehr besitzt, also unsere Lösung verbessert. Wieder stellt sich die Frage, ob ein solcher Pfad immer existiert, wenn der Teilgraph H kein 1-Faktor ist. Die Antwort ist positiv.

Satz 3.2.1. *Sei $G = (V, E)$ ein Graph, der einen 1-Faktor besitzt und $H = (V, M)$ ein $(0,1)$-Faktor von G. Genau dann existiert ein M-augmentierender Pfad, wenn H kein 1-Faktor von G ist.*

Beweis. Sei H ein 1-Faktor. Angenommen es existiert ein M-augmentierender Pfad P. Dann müssen dessen Endknoten den Knotengrad Null besitzen. Widerspruch!
Sei $H = (V, M)$ ein $(0,1)$-Faktor, der kein 1-Faktor ist, und $H' = (V, M')$ ein 1-Faktor. Offensichtlich gilt dann $|M'| > |M|$. Wir setzen $N := M \triangle M'$ gleich der symmetrischen Differenz von M und M' und betrachten den Teilgraphen $T := (V_T, N)$ von G, dessen Knotenmenge V_T aus den Endknoten aller Kanten von N besteht. Jeder Knoten in T besitzt dann den Knotengrad eins oder zwei. Somit besteht T aus Komponenten, die entweder aus M-alternierende Kreise oder M-alternierende Pfaden sind. Die Kreise müssen eine gerade Anzahl von Kanten besitzen, da zwei Kanten aus M (bzw. M') nicht benachbart sein können. Damit besitzen Kreise die gleiche Anzahl von M-Kanten und M'-Kanten. Da aber $|M|' > |M|$ gilt, gibt es einen M-alternierenden Pfad, der ein M-augmentierender Pfad ist. □

Es ergeben sich jedoch Schwierigkeiten bei der Suche nach solchen Pfaden. Betrachten wir folgendes Beispiel.

Beispiel 3.2.2. *Der Pfad $P = (0, 01, 1, 12, 2, 23, 3, 34, 4, 45)$ ist ein M-augmentierender Pfad, der nach Vertauschen der Kanten einen 1-Faktor auf G bestimmt.*

Fangen wir jedoch von Knoten 0 *an, einen alternierenden Pfad* $P' = (0, 01, 1, 12, 2, 35, 5, 54)$ *zu bestimmen, so enden wir erfolglos in einem Kreis.*

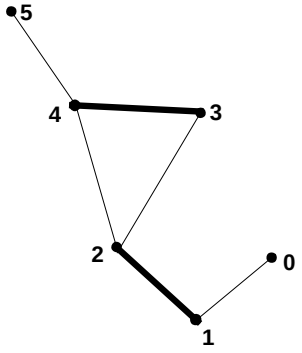

Abbildung 10: M-Augmentierender Pfad

Es ist offenbar wichtig, alternierende Kreise ungerader Länge zu erkennen, um den relevanten alternierenden Teilpfad innerhalb des Kreises zu bestimmen. Wir wollen unsere problematischen ungeraden Kreise beschreiben.

Definition 3.2.3. *Ein M-alternierender Spaziergang $W = (v_0, e_0, v_1, e_1, \ldots, e_t, v_t)$ heißt M-Blume, wenn folgende Bedingungen gelten*

- *t ist eine ungerade natürliche Zahl.*
- *$v_0, \ldots, v_{t-1}$ sind paarweise verschieden.*
- *$d_H(v_0) = 0$*
- *$v_t = v_i$ für eine gerade natürliche Zahl $i < t$.*

Der Kreis $B = (v_i, \ldots, v_t)$ heißt die zur M-Blume assoziierte M-Blüte.

Betrachten wir wieder unser Beispiel 3.2.2. Dann ist $W = (0, 01, 1, 12, 2, 24, 4, 43, 3, 32)$ eine M-Blume und $B = (2, 24, 4, 43, 3, 32, 2)$ die zu W assoziierte M-Blüte. Da eine Blüte immer ein Kreis ungerader Länge ist, kann es in bipartiten Graphen keine Blüten geben. In diesem Fall vereinfacht sich die Bestimmung von augmentierenden Pfaden. Wir betrachten jedoch weiter den allgemeinen Fall. Eine Blüte schrumpfen wir zu einem Knoten, d.h. wir identifizieren die Knotenmenge der Blüte mit einem einzigen Knoten, dem *Blütenknoten* B. Dabei bewahren wir alle Kanten, die mit genau einem Knoten aus $\mathcal{B}$ inzidieren und ignorieren Kanten, die mit zwei Knoten aus $\mathcal{B}$ inzidieren. Dann suchen wir einen M-augmentierenden Pfad P auf dieser Konstruktion. Ist der Knoten B in P enthalten, so blasen wir den Blütenknoten B wieder auf und kehren zum Ausgangsgraphen zurück. Auf diesem konstruieren wir aus P einen M-augmentierenden Pfad auf G.

Abbildung 11: Der geschrumpfte Graph $G \setminus B$

Beispiel 3.2.4. *Die Knotenmenge* $\{2, 4, 3\}$ *aus Beispiel 3.2.2 identifizieren wir mit dem Blütenknoten* B *(Schrumpfen) und erhalten folgenden Graphen.*

Wir finden den M*-augmentierenden Pfad* $P = (0, 01, 1, 1B, B, B5)$*. Im Ausgangsgraphen entspricht der Kante* $B5$ *die Kante* 45*. Also ersetzen wir* $(1B, B, B5)$ *mit* $(12, 2, 23, 3, 34, 4, 45)$ *(Aufblasen) und erhalten den augmentierenden Pfad* $P = (0, 01, 1, 12, 2, 23, 3, 34, 4, 45)$*.*

Formalisieren wir diese Vorgehensweise. Sei $G = (V, E)$ ein Graph und $\mathcal{B} \subseteq V$. $G/\mathcal{B}$ bezeichne den Graphen, der durch Kontraktion der Knotenmenge $\mathcal{B}$ zu einem Blütennoten B entsteht. Das heißt, $G/\mathcal{B}$ besitzt die Knotenmenge $(V \setminus \mathcal{B}) \cup B$. Nun identifizieren wir Kanten $e = (v, w)$, die mit genau einem Knoten $w \in \mathcal{B}$ inzidieren, mit Kanten $e = (v, B)$. Kanten $e \in E(\mathcal{B})$, die mit zwei Knoten aus $\mathcal{B}$ inzidieren, entfallen. Weiterhin definieren wir $M \setminus \mathcal{B} := M \setminus E(\mathcal{B})$ und $E \setminus \mathcal{B} := M \setminus E(\mathcal{B})$. Es ergibt sich folgender Satz.

Satz 3.2.5. *Sei* $\mathcal{B}$ *die Knotenmenge einer* M*-Blüte in* G*. Dann gibt es in* G *genau dann einen* M*-augmentierenden Pfad, wenn es in* $G/\mathcal{B}$ *einen* $M \setminus \mathcal{B}$*-augmentierenden Pfad gibt.*

Beweis. Sei $P = (x_0, \ldots, x_t)$ ein M-augmentierender Pfad auf G. Enthält P keinen Knoten aus $\mathcal{B}$, so ist $P/\mathcal{B}$ auch ein M-augmentierender Pfad auf $G/\mathcal{B}$. Enthält P Knoten von $\mathcal{B}$, so gilt $x_0 \notin \mathcal{B}$ oder $x_t \notin \mathcal{B}$. O.B.d.A. nehmen wir an $x_0 \notin \mathcal{B}$, sonst verwenden wir den Pfad P in entgegengesetzter Richtung. Sei nun x_j der erste Knoten von P, der in $\mathcal{B}$ vorkommt. Dann ist $(x_0, \ldots, x_{j-1}, B)$ ein $M \setminus \mathcal{B}$-augmentierender Pfad auf $G/\mathcal{B}$.
Sei nun P ein $M \setminus \mathcal{B}$-augmentierender Pfad auf G. Enthält P den Knoten B nicht, so ist P ein M-augmentierender Pfad auf G. Enthält hingegen P den Knoten B, so gibt es einen Knoten x in P mit $xB \notin M \setminus \mathcal{B}$. D.h. in der Blüte $B = (v_i, \ldots, v_t)$ in G gibt es ein v_j mit $\{x, v_j\} \notin M$. Setzen wir für den Knoten B in P die Sequenz

$$\begin{array}{ll} v_j, \ldots, v_t & \text{wenn j ungerade ist,} \\ v_j, \ldots, v_i & \text{wenn j gerade ist .} \end{array}$$

In beiden Fällen erhalten wir einen M-augmentierenden Pfad auf G. □

Um M-augmentierende Pfade zu finden, gehen wir also folgendermaßen vor. Ausgehend von einem Knoten, der nicht mit einer M-Kante inzidiert, suchen wir M-alternierende Spaziergänge ungerader Länge und schrumpfen solange M-Blüten, bis wir auf einem durch Schrumpfung entstandenen Graphen einen M-augmentierenden Pfad gefunden haben. Dazu wird zu einem gegebenen $(0, 1)$-Faktor H ein alternierender Wald wie folgt konstruiert.

Sei $G = (V, E)$ ein schlichter Graph und $H = (V, M)$ ein $(0, 1)$-Faktor von G. Sei weiter die Knotenmenge X definiert durch

$$X := \{v \in V | d_H(v) = 0\}.$$

Ein *M-alternierender Wald* $W = (V, F)$ ist ein Teilgraph von G, der ein Wald ist, und für den gelten

- $M \subseteq F$,
- jede Komponente von W enthält genau einen Knoten aus X oder besteht aus genau einer M-Kante,
- jeder Pfad in W, der in einem Knoten $v \in X$ startet, ist M-alternierend.

Weiter definieren wir drei Knotenmengen auf W vermöge

$$\begin{aligned} even(W) &:= \{v \in V | \text{ W enthält geraden Pfad von Knoten aus X nach v}\}, \\ odd(W) &:= \{v \in V | \text{ W enthält ungeraden Pfad von Knoten aus X nach v}\}, \\ free(W) &:= \{v \in V | \text{ W enthält keinen Pfad von Knoten aus X nach v}\}. \end{aligned}$$

Offensichtlich sind diese drei Knotenmengen aufgrund der Waldstruktur disjunkt. Folgender Algorithmus baut nun schrittweise solange alternierende Wälder auf, bis ein M-augmentierender Pfad gefunden wurde.

Algorithmus 3.2.6 (Augmentierender Pfad)**.**
Eingabe: Ein Graph $G = (V, E)$ mit einem $(0, 1)$-Faktor $H = (V, M)$ mit $|M| < \frac{1}{2}|V|$.
Ausgabe: Ein M-augmentierender Pfad auf G.
Initialisierung: Setze $W := (V, F)$ mit $F := M$. Weiterhin setzen wir $e_v := \{v, u\}$ für jeden Knoten $v \in V$, wenn es ein $e_v \in E$ gibt mit $u \in X$. Wir sammeln die e_v in einer Liste.
While (kein augmentierender Pfad gefunden ist) do

1. *Wähle ein $v \in V$, für das ein $e_v = (v, u)$ existiert.*
2. *If $(v \in free(W))$ then*
   - *Es gibt genau ein $w \in V$ mit $\{v, w\} \in M$. Für alle Knoten x, die mit w benachbart sind, setze $e_x := \{x, w\}$.*
   - *$F := F \cup \{e_v\}$.*
3. *If $(v \in even(W))$ then*
   - *Finde die geraden Pfade P, Q von einem Knoten $x \in X$ nach v und von einem Knoten aus $x' \in X$ nach w.*

- *If* $(x \neq x')$ *then* $P' = (P, e_v, Q)$ *ist ein augmentierender Pfad auf* G *else*
  (P, e_v, Q) *enthält eine* M*-Blüte.*
  Schrumpfe die Knotenmenge $\mathcal{B}$ *der Blüte durch Konstruktion des Graphen* $G := G/\mathcal{B}$ *und* $M := M \setminus \mathcal{B}$.

End While

Falls der Pfad P' *Blütenknoten enthält, konstruiere wie im Beweis zu 3.2.5 einen* M*-augmentierenden Pfad auf* G *durch Aufblasen.*

Beispiel 3.2.7. *Betrachten wir den Graphen* G *aus Beispiel 3.2.2. Wir bauen schrittweise mit dem Algorithmus alternierende Wälder auf, wobei wir eine Blüte zu einem Blütenknoten* B *schrumpfen müssen. Im dritten Wald finden wir einen augmentierenden Pfad.*

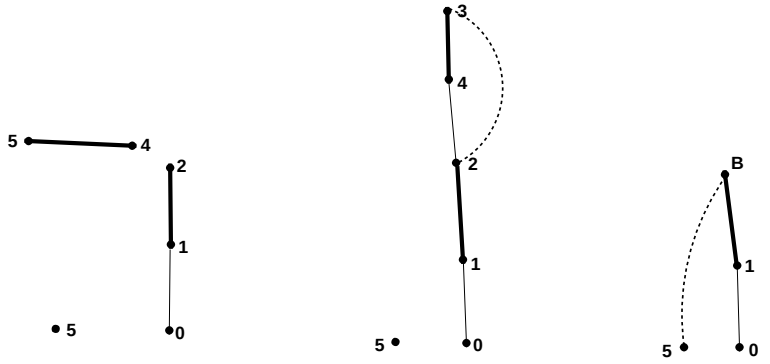

Abbildung 12: Alternierende Wälder

Für den Knoten u in $e_v = (v, u)$ gilt $u \in even(W)$ nach Konstruktion. Im Algorithmus wird entweder zu einem geraden X-Pfad im alternierenden Wald eine M-Kante aus $free(W)$ und eine $E \setminus M$-Kante e_x hinzugefügt. Damit entsteht ein neuer alternierender Wald W' mit einer größeren Kantenmenge. Dabei gilt für die Knotenmengen $free(W) = free(W') + 2$. Die Kanten e_x können in der Zeit $O(|V|)$ neu gesetzt werden. Oder zwei gerade Pfade aus W, deren Bestimmung $O(|V|)$ Zeit kostet, werden durch eine $E \setminus M$-Kante e_x aneinander gefügt. Wenn die X-Knoten in verschiedenen Komponenten liegen, entsteht ein augmentierender Pfad. Sonst haben wir eine Blüte gefunden. Im ersten Fall sind wir fertig, im zweiten Fall konstruieren wir den neuen Graphen $G/\mathcal{B}$ in der Zeit $O(|\mathcal{B}||V|)$ und beginnen den Algorithmus auf dem Graphen erneut. In diesem Fall enthält die Knotenmenge des neuen Graphen $G/\mathcal{B}$ mindestens zwei Knoten weniger. Somit kann der Algorithmus höchstens $\frac{1}{2}(|V| + |free(W)|) < |V|$ mal durchgeführt werden bis ein augmentierender Pfad gefunden wird. Insgesamt

ergibt sich für die Suche nach einem augmentierenden Pfad eine Laufzeit von $O(|V|^2)$. Für Graphen, für deren Kantenmengen E $|E| \in O(|V|)$ gilt, gibt es eine Implementierung mit einer verbesserten Komplexität. Gabow, Galil und Micali [9] erreichen mit Hilfe von intelligenten Datenstrukturen (Prioritätswarteschlangen) für die Suche nach einem augmentierenden Pfad die Komplexität $O(|E|log(|V|)$. Um die Korrektheit des Algorithmus zu zeigen, müssen wir sicher gehen, dass eine Kante $e_x \in E \setminus M$ existiert, die die Knotenmengen $even(W)$ und $even(W) \cup free(W)$ miteinander verbindet. Angenommen dem ist nicht so. Dann gilt für die Anzahl $o(G \setminus odd(W))$ der ungeraden Komponenten des Graphen $G \setminus odd(W)$

$$o(G \setminus odd(W)) \geq |even(W)| = |X| + |odd(W)| = (|V| - 2|M|) + |odd(W)|.$$

Denn jeder Knoten aus $even(W)$ ist in $G \setminus odd(W)$ isoliert, also eine ungerade Komponente. Wegen der Formel von Tutte und Berge [12] gilt für jeden Graphen G, der einen 1-Faktor besitzt

$$odd(W) \geq o(G \setminus odd(W)).$$

Insgesamt folgt $|M| = \frac{|V|}{2}$, also ist $H = (V, M)$ bereits ein 1-Faktor. Diesen Fall schließen wir jedoch bei der Eingabe für unseren Algorithmus aus. Somit existiert eine Kante e_x und unser Algorithmus arbeitet korrekt.

3.3 Das ungewichtete f-Faktor Problem

Betrachten wir nun ein ungewichtetes f-Faktor Problem auf dem Graphen $G = (V, E)$. Das Knotensplitting von Tutte reduziert uns dieses Problem zum ungewichteten 1-Faktor Problem auf dem Splitgraphen $G_f = (V_f, E_f)$. Da nach 3.1 $|V_f| \in O(|E(G)|)$ und $|E_f| \in O(|V(G)||E(G)|)$ liegen, erreichen wir zur Bestimmung eines augmentierenden Pfades eine Komplexität von $O(|E_f||log(|V_f|)|)$ bzw. $O(|E(G)||V(G)|log(|V(G)|))$ und somit zur Berechnung eines f-Faktors eine Komplexität von $O(|V_f||E_f||log(|V_f|))$ bzw. $O(|E(G)|^2|V(G)|log(|V(G)|))$. Gabow [8] entwickelte nun die raffinierte Idee, die augmentierenden Pfade von G_f schrittweise auf einem *Suchgraphen* $G_f(M)$ zu suchen. Dabei gilt $E(G_f(M))$ $\in O(E(G))$. Das heißt, die Bestimmung eines augmentierenden Pfades benötigt nur die Laufzeit von $O(|E(G)|log(|V(G))|)$.
Betrachten wir für alle $v \in V$ die vollständigen bipartiten Teilgraphen G_v in G_f. Wir wählen zu jedem äußeren Knoten eine äußere M-Kante und verlangen, dass alle inneren Knoten mit höchstens einer dieser M-Kanten inzidieren. Nach jeder Augmentierung bleibt diese Situation offensichtlich erhalten. Betrachten wir nun einen M-augmentierenden Pfad P, der durch einen bipartiten Graphen G_v von G_f verläuft. Teilpfade P_T von P, die vollständig durch G_v verlaufen, müssen eine gerade Anzahl von Kanten besitzen. Betrachten wir Teilpfade $P_T = (v_{i,v}, e_M, \ldots, e_{E\setminus M}, v_{i',v})$ in G_v. Da G_v ein vollständiger bipartiter Graph ist, existiert auch der Pfad $(v_{i,v}, e_M, v_{i',v})$ und kann P_T in P ersetzen. Wir können also den M-augmentierenden Pfad P zu einem M-augmentierenden Pfad kürzen, dessen Teilpfade in G_v nur aus zwei Kanten bestehen. Dieser Pfad

wiederum lässt sich zu einem M-augmentierenden Pfad kürzen, der höchstens zwei Teilpfade in G_v enthält, die je aus zwei Kanten bestehen, einer beginnend mit einer M-Kante und der andere beginnend mit einer $E \setminus M$-Kante. Wir bezeichnen einen M-augmentierenden Pfad P in G_f als *kurz*, wenn für alle $v \in V$ gilt, dass es höchstens zwei Teilpfade $(v_{i,v}, e_M, w_{j,v}, e_{E\setminus M}, v_{i',v})$ und $(v_{k,v}, e_{E\setminus M}, w_{j',v}, e_M, v_{k',v})$ in G_v gibt.

Beispiel 3.3.1. *Der Pfad* $P = (1, \ldots, 14)$ *ist ein augmentierender Pfad, der zu dem kurzen augmentierenden Pfad* $P' = (1, \ldots, 11, \{11, 14\}, 14)$ *verändert werden kann. Der Teilpfad* $(1, \ldots, 3)$ *von* P' *beginnt mit der* $E \setminus M$*-Kante* $\{1, 2\}$ *und der Teilpfad* $(10, \{10, 11\}, 11, \{11, 14\}, 14)$ *von* P' *beginnt mit der* M*-Kante* $\{10, 11\}$.

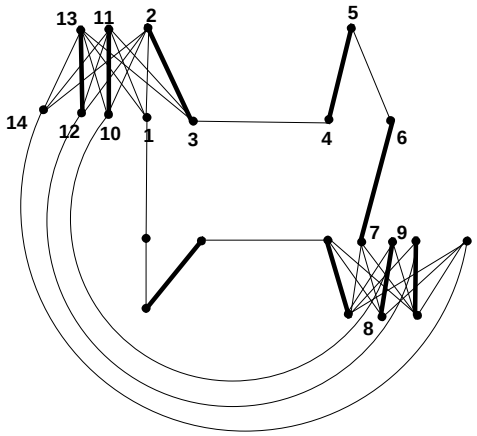

Abbildung 13: Gekürzter und ungekürzter augmentierender Pfad

Diese Beobachtung führt zu folgender Konstruktion von Gabow. Wir konstruieren einen Suchgraphen $G_f(M)$ bezüglich dem Knotensplitgraphen G_f und seinem $(0,1)$-Faktor $H_f = (V_f, M)$. Dabei setzen wir voraus, dass für alle äußeren Knoten $w_{j,v} \in W_v$ gilt $d_{H_f}(w_{j,v}) = 1$. Wir ersetzen in G_f die bipartiten Graphen G_v durch bipartite Graphen $G_M(v) := (V_M(v) \cup W_M(v), E_M(v))$ mit den zwei unabhängigen Knotenmengen

$$\begin{aligned} V_M(v) &:= V_v \cup \{V_{1,v}, V_{2,v}\} \\ W_M(v) &:= W_v \cup \{W_{1,v}, W_{2,v}\}. \end{aligned}$$

Dabei definieren wir Kanten $e \in E_M(v)$ wie folgt

- $\{V_{1,v}, W_{1,v}\}, \{V_{2,v}, W_{2,v}\}$ sind M-Kanten in $E_M(v)$.
- Gilt $d_{H_f}(v_{i,v}) = 0$, dann verbinde $v_{i,v}$ durch zwei Kanten, die nicht in M liegen mit $W_{1,v}$ und $W_{2,v}$.

- Gilt $d_{H_f}(v_{i,v}) = 1$ und liegt die zu $v_{i,v}$ inzidente M-Kante $\{v_{i,v}, w_{j,v}\}$ in G_v, dann verbinde $v_{i,v}$ mit $w_{j,v}$ durch eine M-Kante. Verbinde nun w_i durch zwei Kanten, die nicht in M liegen mit $V_{1,v}$ und $V_{2,v}$.
- Gilt $d_{H_f}(v_{i,2}) = 1$ und liegt die zu $v_{i,v}$ inzidente M-Kante $\{v_{i,v}, w_{j,v}\}$ nicht in G_v, dann verbinde $v_{i,v}$ durch zwei Kanten mit $W_{1,v}$ und $W_{2,v}$.

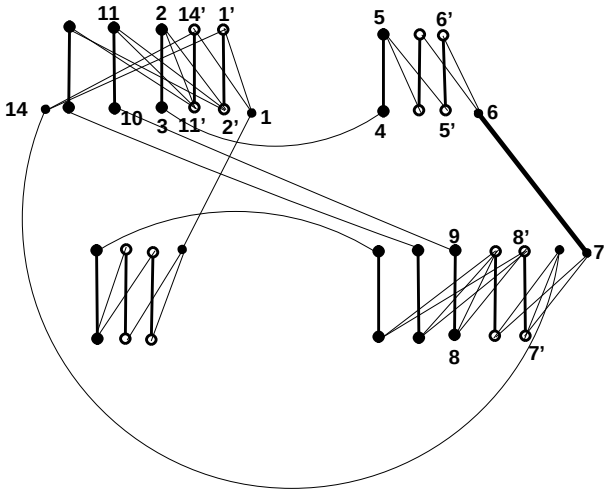

Abbildung 14: Der Suchgraph $G_f(M)$ zu Beispiel 3.3.1

Satz 3.3.2. *Genau dann gibt es einen M-augmentierenden Pfad P in G_f, wenn es einen M-augmentierenden Pfad P' in $G_f(M)$ gibt.*

Beweis. Sei P ein M-augmentierender Pfad in G_f. Dann können wir für alle $v \in V$ Teilpfade von P in G_f so abkürzen, dass ein kurzer augmentierender Pfad auf G_f entsteht. Für einen Teilpfad $(v_{i,v}, e_M, w_{j,v}, e_{E\setminus M}, v_{i',v})$ in G_v setzen wir in P $(v_{i,v}, e_M, w_{j,v}, e_{E\setminus M}, V_{1,v}, W_{1,v}, v_{i',v})$ und für einen Teilpfad $(v_{k,v}, e_{E\setminus M}, w_{j,v}, e_M, v_{k',v})$ in G_v setzen wir $(v_k, W_{2,v}, V_{2,v}, w_{j',v}, e_M, v_{k',v})$. Dann entsteht auf $G_f(M)$ ein M-augmentierender Pfad P'.
Sei umgekehrt P' ein M-augmentierender Pfad auf $G_f(M)$. Dann hat P' in $G_M(v)$ die zwei möglichen Teilpfade $(v_{k,v}, W_{2,v}, V_{2,v}, w_{j',v}, e_M, v_{k',v})$ und $(v_{i,v}, e_M, w_{j,v}, e_{E\setminus M}, V_{1,v}, W_{1,v}, v_{i',v})$. Ersetzen wir diese in P' durch die Teilpfade $(v_{k,v}, e_{E\setminus M}, w_{j',v}, e_M, v_{k',v})$ bzw. $(v_{i,v}, e_M, w_{j,v}, e_{E\setminus M}, v_{i',v})$ in G_v, dann entsteht ein M-augmentierender Pfad P auf G_f.

□

Wir können somit die Suche nach einem M-augmentierenden Pfad in G_f auf die Suche nach einem M-augmentierenden Pfad P' auf $G_f(M)$ beschränken. Finden wir in der Abbildung 14 den augmentierenden Pfad $(1, 1', 2', 2, 3, 4, 5, 5', 6', 6, 7, 7', 8', 8, 9, 10, 11, 11', 14', 14)$, dann entspricht dieser dem augmentierenden Pfad P' aus Beispiel 3.3.1. Die Anzahl der Kanten in $G_M(v)$ beträgt

$3d_G(v) - d_F(v) + 2$. Somit gilt $E_f(M) \in O(|E(G)|)$. Nach Auffinden eines M-augmentierenden Pfades muss in Abhängigkeit von der neuen Menge M wiederholt ein neuer Graph $G_f(M)$ konstruiert werden. Dies entspricht einer Augmentierung auf dem Graphen G_f. Das ist jedoch nicht aufwendig. Betrachten wir wieder einen M-augmentierenden Pfad P auf $G_f(M)$ und seine Teilpfade. Angenommen P enthält für ein $G_M(v)$ einen Teilpfad $(v_{i,v}, e_M, w_{j,v}, e_{E\setminus M}, V_{1,v}, e_M, W_{1,v}, e_{E\setminus M}, v_{i',v})$. Dann existiert in G_f der Teilpfad $(v_{i,v}, e_M, w_{j,v}, e_{E\setminus M}, v_{i',v})$ eines augmentierenden Pfades. Nach Vertauschen der Kanten entsteht der Teilpfad $(v_{i,v}, e_{E\setminus M}, w_{j,v}, e_M, v_{i',v})$. Das bedeutet nach Konstruktion unseres Hilfsgraphen, dass in $G_f(M')$ der Knoten $v_{i,v}$ mit den Knoten $W_{1,v}, W_{2,v}$ verbunden wird. Stattdessen ist nun $v_{i',v}$ durch eine M-Kante mit $w_{j,v}$ verbunden. Ob eine Verbesserung hin zu $f(v)$ an $G_M(v)$ stattgefunden hat, hängt von $v_{i',v}$ statt. Galt vor der Augmentierung $d_{H_f}(v_{i',v}) = 0$, dann ist dem so. Es stellt sich also heraus, dass wir den Splitgraphen G_f nicht mehr benötigen. Wir suchen lediglich M-augmentierende Pfade auf $G_f(M)$ und konstruieren daraus neue Graphen $G_f(M')$ mit $|M'| > |M|$, bis keine augmentierenden Pfade mehr existieren.

Beispiel 3.3.3. *Aus dem Suchgraphen $G_f(M)$ in Abbildung 14 konstruieren wir mit Hilfe des augmentierenden Pfades $(1, 1', 2', 2, 3, 4, 5, 5', 6', 6, 7, 7', 8', 8, 9, 10, 11, 11', 14', 14)$ den Suchgraphen $G_f(M')$.*

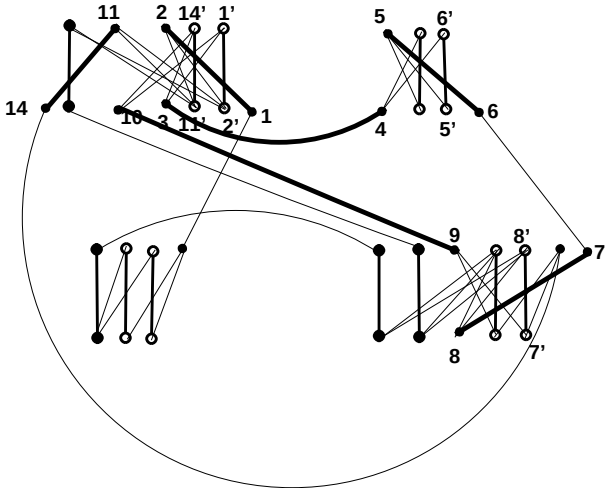

Abbildung 15: Der Suchgraph $G_f(M')$ zu Beispiel 3.3

Wir können also folgenden Algorithmus zum Auffinden eines f-Faktors formulieren.

Algorithmus 3.3.4 (f-Faktor)**.**
Eingabe: Ein Graph $G = (V, E)$ und eine Funktion $f : V \to \mathbb{N}_0$.
Ausgabe: Ein f-Faktor von $G = (V, E)$.

Initialisierung: Ein Graph $G_f(M)$ wie oben definiert. Dabei setzen wir die Menge M wie folgt.

- *In $V_M(v)$ inzidieren alle äußeren Knoten in W_v mit genau einer M-Kante.*
- *Weiterhin inzidieren genau $f(v)$ innere Knoten in V_v nur mit $E \setminus M$-Kanten.*

While (ein augmentierender Pfad P auf $G_f(M)$ existiert) do

Suche mit Algorithmus 3.2.6 einen augmentierenden Pfad P auf $G_f(M)$.
Konstruiere den neuen Graphen $G_f(M')$.
Setze $G_f(M) := G_f(M')$.

Nach [9] kann das f-Faktor Problem schneller, nämlich in der Zeit $O((\sum_{v \in V} f(v))^{0,5}|E|)$ gelöst werden. Hintergrund dieser Implementierung ist, dass kürzeste M-augmentierende Pfaden bestimmt werden. Bei dieser Vorgehensweise ergeben sich einige Schwierigkeiten mit unserem Suchgraphen, die jedoch behandelbar sind. Wir haben unseren Algorithmus jedoch hinsichtlich der Bestimmung eines maximal gewichteten f-Faktor Problems in den nächsten Abschnitten konstruiert und deswegen diesen Fall nicht in unsere Betrachtung einbezogen.

3.4 Das gewichtete 1-Faktor Problem als lineares Programm

Als eine Verallgemeinerung von Transport- und Netzwerkflussproblemen entwickelten Dantzig, Ford und Fulkerson 1956 ein Schema, das es ermöglicht, lineare Programme zu lösen. Ein kombinatorisches Optimierungsproblem, das sich zu einem linearen Programm reduzieren lässt, kann mit dieser Methode gelöst werden. Zunächst reduzieren wir unser gewichtetes 1-Faktor Problem zu einem linearen Programm.
Wie definieren auf einem Graphen $G = (V, E)$ und einer Kantenmenge $M \subseteq E$ die *charakteristische Funktion* $\chi^M : E \to \{0, 1\}$ wobei gilt

$$\chi^M(e) := \begin{cases} 1 & \text{wenn } e \in M \\ 0 & \text{wenn } e \notin M. \end{cases}$$

In $\mathbb{R}^n$ ist dann die konvexe Hülle der Menge $\{\chi^M | H = (V, M)$ ist ein 1-Faktor von G$\}$ ein Polytop, das wir als 1-*Faktor-Polytop* bezeichnen. Sei

$$\begin{aligned} \delta(v) &:= \{e \in E | v \text{ ist inzident zu e } \} \forall v \in V \\ E(U) &:= \{e \in E |, \text{ e inzidiert mit zwei Knoten aus U } \} \forall U \subseteq V. \end{aligned}$$

Nach Edmonds ergeht folgender Satz.

Satz 3.4.1. *Sei $x \in \mathbb{R}^E$ und $x(T) := \sum_{e \in T} x(e)$ für alle $T \subseteq E$. Für jeden Graphen $G = (V, E)$ ist das 1-Faktor-Polytop bestimmt durch die Menge aller x mit*

$$\begin{aligned} x(e) &\geq 0 \text{ für alle } e \in E \\ x(\delta(v)) &= 1 \text{ für alle } v \in V \\ x(E(U)) &\leq \lfloor \frac{1}{2}|U| \rfloor \text{ für alle } U \subseteq V \text{ mit } |\mathrm{U}| \text{ ungerade.} \end{aligned}$$

Beweis. Beweis siehe [12]. □

Wir erhalten für unser gewichtetes 1-Faktor Problem eine Formulierung als lineares Programm über dem 1-Faktorpolytop. Wir wissen, dass jede Eckenlösung mit einem 1-Faktor identifiziert werden kann. Schreiben wir unser lineares Programm in Matrixform auf, so erkennen wir intuitiv den Hintergrund dieser Formulierung. Wir nummerieren die Kanten eines Graphen $G = (V, E)$ vermöge $e_1, \ldots, e_m$, die Knoten mit $v_1, \ldots, v_n$, wobei $|E| = m$ und $|V| = n$ gilt. Weiterhin bezeichnen wir alle ungeraden Knotenmengen $O \subseteq V$, die mehr als einen Knoten enthalten mit $B_1, \ldots, B_k$, wobei $k = \sum_{\nu=1}^{\frac{n}{2}-1} \binom{n}{2\nu+1}$ gilt. Die *Inzidenzmatrix* $A = (a_{ij})$ ist eine $n \times m$-Matrix mit

$$a_{ij} := \begin{cases} 1 & \text{wenn } v_i \in e_j \\ 0 & \text{sonst.} \end{cases}$$

Die *Blossommatrix* $B = (b_{ij}) \in Mat_{k,m}(\mathbb{R})$ ist eine $k \times m$-Matrix mit

$$b_{ij} := \begin{cases} 1 & \text{wenn für zwei } v_\nu, w_\mu \in B(i) \text{ gilt } v_\nu, w_\mu \in e_j \\ 0 & \text{sonst.} \end{cases}$$

Sei nun ein Vektor $c \in \mathbb{R}^k$ gegeben mit

$$c_i := \frac{1}{2}(|B(i)| - 1) \text{ für } 1 \leq i \leq k.$$

Dann wird das 1-Faktor-Polytop bestimmt durch die Menge aller $x \in \mathbb{R}^m$, wobei gilt

$$\begin{aligned} Ax &= 1 && (1) \\ Bx &\leq c && (2) \\ x_j &\geq 0 \text{ für alle } j \in \{1, \ldots, m\}. && (3) \end{aligned}$$

Betrachten wir ein Beispiel, bei der Bedingung (2) verletzt wird. Dann existieren auf den Ecken optimale Lösungen unseres Problems, die nicht ganzzahlig sind.

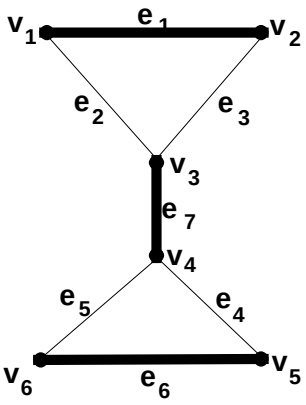

Abbildung 16: Ein 1-Faktor auf dem Graphen G

Beispiel 3.4.2. *Der abgebildete Graph mit Kantengewicht eins für alle Kanten besitzt die Inzidenzmatrix*

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0. \end{pmatrix}$$

Die fettgezeichneten Kanten bilden offenbar einen maximal gewichteten 1*-Faktor von* G*. Eine optimale Lösung lautet also* $x = (1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1)^t$*. Jedoch gibt es auch die rationale Lösung* $x = (0,5 \quad 0,5 \quad 0,5 \quad 0,5 \quad 0,5 \quad 0,5 \quad 0)^t$*, wenn die Bedingung* (2) *verletzt wird.*

Doch das ist nicht alles. Wie aus der linearen Optimierung bekannt [2], ist die Menge aller x, die Bedingung (1) und (3) erfüllen auch ein Polytop. Ein x ist genau dann eine Ecke, wenn die zu x gehörige Spaltenmenge, also die Menge aller Spalten $(a_{ij})_{i\in\{1,\ldots,n+k\}}$ von A mit $x_j > 0$, linear unabhängig ist. Kehren wir zu unserem Beispiel zurück. Sowohl die erste, sechste und siebente Spalte sind linear unabhängig bezüglich $\mathbb{R}$, als auch die erste bis sechste Spalte. Somit enthält unser Beispiel eine Lösung auf der Eckenmenge des Polytops, die optimal aber nicht ganzzahlig ist, also unbrauchbar für unser gewichtetes 1-Faktor Problem. Die Bedingung (2) verhindert offenbar genau diesen Fall, konkret passiert Folgendes. Betrachten wir einen nicht ganzzahligen Vektor, der im System die Bedingung (2) verletzt. Dann muss ein Knoten in G existieren, der mit mindestens zwei Kanten inzidiert, deren Kantenwert x_j kleiner als eins ist. Da aber jede Kante mit einem weiteren Knoten inzidiert, ist mindestens eine weitere Kante an diesem Knoten notwendig, um auf die Summe eins am

Knoten zu kommen. Somit enthält die Lösung Kreise. Gerade bzw. ungerade Kreise besitzen Inzidenzmatrizen der Form

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

bzw.

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1. \end{pmatrix}$$

Man erkennt leicht, dass die Spaltenmenge eines geraden Kreises immer linear abhängig ist, während die Spaltenmenge eines ungeraden Kreises linear unabhängig ist. Mit obigen Überlegungen folgt, dass mögliche optimale Lösungen x, die in der Eckenmenge des Polytops ohne Bedingung (2) liegen, niemals gerade sondern nur ungerade Kreise enthalten können. Das heißt insbesondere, dass für bipartite Graphen, die bekannterweise keine ungeraden Kreise enthalten, das System ohne die Bedingung (2) nur ganzzahlige Eckenmengen enthält. Für nichtbipartite Graphen müssen wir ungerade Kreise als Lösungen ausschließen. Das genau bewirkt unsere Blossommatrix. Kehren wir noch einmal zu unserem Beispiel 4 zurück.

Beispiel 3.4.3. *Die Knotenmenge* $B := \{v_1, v_2, v_3\}$ *ergibt in der Blossommatrix die Zeile* $(b_{i_0,j})_{1 \leq j \leq m} = (1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0)$ *mit dem Zeilenwert* $c_{i_0} = 1$. *Die Lösung* $x = (0,5 \quad 0,5 \quad 0,5 \quad 0,5 \quad 0,5 \quad 0,5 \quad 0)^T$ *multipliziert mit der Blossomzeile ergibt* $1,5 > c$. *Somit enthält das System diese Lösung nicht mehr.*

Betrachten wir nochmals die Blossommatrix. Eine Zeile beschreibt offenbar für eine ungerade Knotenmenge $B(i) \subseteq V$ die Anzahl der Kanten, die mit zwei Knoten aus $B(i)$ inzidieren. Bezeichne nun C die Matrix, welche die Kanten beschreibt mit denen genau ein Knoten aus $B(i)$ inzidiert, also die Matrix $C = (c_{ij})$ ist eine $k \times m$-Matrix mit

$$c_{ij} := \begin{cases} 1 & \text{wenn für genau ein } v \in B(i) \text{ gilt } v \in e_j \\ 0 & \text{sonst.} \end{cases}$$

Bezeichne weiterhin D die Matrix, welche die Kanten beschreibt, mit denen mindestens ein Knoten aus $B(i)$ inzidiert. Die Matrix $D = (d_{ij})$ ist eine $k \times m$-Matrix mit

$$d_{ij} := \begin{cases} 1 & \text{wenn für mindestens ein} v \in B(i) \text{ gilt } v \in e_j \\ 0 & \text{sonst.} \end{cases}$$

Offenbar gilt dann $B + C = D$. Betrachten wir aus unserem linearen Programm die Bedingung (2), $Bx \leq c$. Wegen der Bedingung $Ax = 1$ gilt für jeden Knoten v_i

$$\sum_{v_i \in e_j} x_j = 1.$$

Weil jeder Knoten in G mit mindestens einer Kante inzidiert und $|B(i)|$ ungerade ist, besitzt jede Zeile der Matrix D mindestens $\frac{1}{2}(|B(i)|-1)+1$ positive Einträge. Somit ergibt sich

$$Bx - Dx \leq c - (\frac{1}{2}(|B(i)| - 1) + 1)_{1 \leq i \leq k} \Leftrightarrow -Cx \leq -1.$$

Wir können somit ein äquivalentes lineares Programm formulieren vermöge

$$\begin{aligned} Ax &= 1 \\ Cx &\geq 1 \\ x_j &\geq 0 \text{ für alle } j \in \{1, \ldots, m\}. \end{aligned}$$

Im Folgenden definieren wir das Polytop P als

$$P := \left\{ x \in \mathbb{R}^{|E|} | Ax = 1, Cx \geq 1, x_j \geq 0 \qquad \forall j \in \{1, \ldots, m\} \right\}.$$

Sei $w \in \mathbb{R}^E$ eine Kantengewichtsfunktion, die wir mit dem Vektor $w \in \mathbb{R}^{|E|}$ identifizieren. Sei LP die Menge aller $x \in P$, für die $Wert(LP) := w^T x$ maximiert wird. (Abweichend von [12] behandeln wir die Berechnung eines maximal und nicht eines minimal gewichteten 1-Faktors.) Die Elemente von P nennen sich *zulässige Lösungen* von LP. Die Elemente von LP bezeichnet man als *primale optimale Lösungen*. Wir bestimmen das zugehörige duale Programm. Sei $y \in \mathbb{R}^n$, $z = (z_{B_1}, \ldots, z_{B_k})^T \in \mathbb{R}^k$. Dann bezeichne P_D das Polytop

$$P_D := \{ \begin{pmatrix} y \\ z \end{pmatrix} \in \mathbb{R}^{n+k} | z_{B_i} \geq 0 \ \forall i \in \{1, \ldots, k\}, (y^T, z^T) \begin{pmatrix} A \\ -C \end{pmatrix} \geq w \}.$$

Dann ist das duale Programm DP von LP gegeben durch die Menge aller $\binom{y}{z} \in P_D$, die $Wert(DP) := \sum_{i=1}^{n} y_i - \sum_{j=1}^{k} z_{B_i}$ minimieren. Die Elemente von P_D nennen sich *zulässige Lösungen* von DP. Die Elemente von DP bezeichnet man als *duale optimale Lösungen*. Da wir in unserem Fall von der Existenz eines maximal gewichteten 1-Faktors Kenntnis besitzen (wir gehen weiterhin von unserem reduzierten Minkonvex Problem aus), also einer optimalen Lösung $x \in LP$, folgt mit dem Hauptsatz der linearen Optimierung [12], dass auch das duale Programm DP eine optimale Lösung $\binom{y}{z}$ besitzt. Insbesondere gilt die starke Dualität, also $Wert(LP) = Wert(DP)$. Der primal-dualen Methode für lineare Programme liegt folgende Idee zugrunde. Wir beginnen mit einer dualen zulässigen Lösung $\binom{y}{z}$ und suchen nach einer primal zulässigen Lösung x, basierend auf den zwei folgenden Sätzen.

Satz 3.4.4 (Komplementärer Schlupf)**.** *Seien $x \in P$ und $(y,z) \in P_D$ zulässige Lösungen. Dann sind $x \in LP$ und $\binom{y}{z} \in DP$ genau dann , wenn gilt*

$$z_{B_i} > 0 \quad \Rightarrow \quad \sum_{j=1}^{m} c_{ij}x_j = 1 \text{ für alle } i \in \{1,\dots,k\}$$

$$x_j > 0 \quad \Rightarrow \quad \sum_{i=1}^{k}(-c_{ij}z_{B_i}) + y_\nu + y_\mu = w_j \text{ für alle } j \in \{1,\dots,m\} \text{ und } \{v_\nu, v_\mu\} = e_j.$$

Beweis. Siehe beispielsweise [12]. □

Lemma von Farkas 3.4.5. *Für $A' \in Mat_{m',n}(\mathbb{R})$ und $b \in \mathbb{R}^m$ sind folgende Aussagen äquivalent.*

1. *Es gibt ein $x' \in \mathbb{R}^{m'}$ mit $A'x' = b$ und $x' \geq 0$.*
2. *Für alle $y' \in \mathbb{R}^n$ mit $y^T A' \geq 0$ folgt $y^T b \geq 0$.*

Beweis. Siehe beispielsweise [12]. □

Die Idee ist nun Folgende. Wir beginnen mit einer zulässigen dualen Lösung $\binom{y}{z}$. Man betrachte die Untermatrix A' von $\binom{A}{-C}$, bestehend aus den Spalten j, die die Bedingung $\sum_{i=1}^{k}(-c_{ij}z_{B_i}) + y_\nu + y_\mu = w_j$ erfüllen. Um eine primale Lösung zu finden, löse man das eingeschränkte lineare Programm $A'x' = \binom{1}{-1}$ mit $x' \geq 0$. Für unseren Graphen heißt das konkret, dass wir auf einem Teilgraphen unseres Ausgangsgraphen, nämlich dem Teilgraphen dessen Kanten $e_j = \{v_\nu, v_\mu\}$ die Bedingung

$$w_\pi(j) := \sum_{i=1}^{k}(-c_{ij}z_{B_i}) + y_\nu + y_\mu - w_j = 0$$

erfüllen, einen 1-Faktor suchen. Es ergeben sich zwei Fälle.

1. Der Teilgraph enthält einen 1-Faktor, d.h. das eingeschränkte lineare Programm $A'x' = \binom{1}{-1}$ mit $x' \geq 0$ ist lösbar. Dann setzen wir für die fehlenden Komponenten von x jeweils eine Null. Es ergibt sich $Ax = \binom{1}{-1}$ und $x_j \geq 0$. Mit 3.4.4 ergibt sich, dass wir zwei optimale Lösungen gefunden haben, also die Lösung unseres Problems.

2. Der Teilgraph enthält keinen 1-Faktor, d.h. das eingeschränkte lineare Programm $A'x' = \binom{1}{-1}$ mit $x' \geq 0$ besitzt keine Lösung. Nach dem Lemma von Farkas 3.4.5 gibt es dann ein $\binom{y'}{z'}$ mit $(y'^T, z'^T)A' \geq 0$ und $(y'^T, z'^T)\binom{1}{-1} < 0$. Weil $z \geq 0$ gilt, gibt es ein $\alpha \in \mathbb{R}_+$, sodass $\binom{y}{z} + \alpha\binom{y'}{z'}$ eine weitere zulässige Lösung des dualen Programms ist. Es gilt $(y^T, z^T)\binom{1}{-1} > ((y+\alpha y')^T, (z+\alpha z')^T)\binom{1}{-1}$. Wir haben also eine bessere Lösung des dualen Programms gefunden. Wir wählen ein α und iterieren mit der neuen Lösung.

Das Prinzip ist bei linearen Programmen universell anwendbar und steht und fällt letztlich mit der Umsetzbarkeit der einzelnen Schritte. Wie stellen wir fest, ob $A'x' = \binom{1}{-1}$ mit $x' \geq 0$ lösbar ist? Uns interessieren nur ganzzahlige Lösungen. Wie wir wissen, befinden sich solche auf den Ecken des 1-Faktorpolytops. Wir wenden unser Prinzip aus dem letzten Kapitel an. Wir suchen mit Hilfe alternierender Wälder auf den Teilgraphen, dessen Kanten e_j die Bedingung $w_\pi(j) = 0$ erfüllen, augmentierende Pfade und augmentieren, solange solche Pfade existieren. Finden wir einen (ungewichteten) 1-Faktor, so sind wir fertig. Finden wir keinen 1-Faktor, so kommen durch die Modifikationen an der dualen Lösung weitere Kanten hinzu. D.h. stellen wir uns vor, dass die Konstruktion eines alternierenden Waldes $W = (V, F)$ nicht fortgesetzt, kein augmentierender Pfad gefunden und keine Blüte geschrumpft werden kann. Vorerst gehen wir der Einfachheit halber davon aus, dass keine Blütenknoten B im Wald vorhanden sind. Dann benötigen wir neue Kanten, die wir erhalten, indem wir duale Variablen verändern. Dabei wollen wir bisher zulässige Kanten bewahren, d.h. wir wollen den Wert $w_\pi(j) = 0$ von Kanten e_j nicht verändern. Das erreichen wir, indem wir für von jeder dualen Variablen y_i, wenn $v_i \in even(W)$ gilt, ein α subtrahieren bzw. ein α addieren, wenn $v_i \in odd(W)$ gilt. Das ist eindeutig möglich, weil diese Knoten in einem Wald W enthalten sind. Damit verändern wir allerdings die Werte $w_\pi(j) > 0$ von Kanten $e_j \notin F$, die bisher nicht im Wald enthalten sind. Wir wollen erreichen, dass für mindestens eine dieser Kanten $w_\pi(j) = 0$ gilt (eine neue mögliche Kante im Wald) und für die anderen weiterhin $w_\pi(j) > 0$, um die Zulässigkeit der neu entstehenden dualen Lösung zu garantieren. Es gibt verschiedene Möglichkeiten, wie eine Kante $e_j \notin F$ mit den Knoten aus W inzidieren kann. Verbindet sie zwei Knoten aus $odd(F)$, so vergrößert sich der Wert $w_\pi(j)$ um 2α, die Zulässigkeit ist also garantiert. Verbindet sie zwei Knoten aus $odd(F)$ und $even(F)$, so verändert sich ihr Wert nicht. Inzidiert sie jedoch mit zwei Knoten aus $even(F)$, so verringert sich ihr Wert um 2α. Eine Verkleinerung um α findet statt, wenn eine Kante mit zwei Knoten aus $even(F)$ und $free(F)$ inzidiert. Das sind offenbar uns interessierende Kanten. Der Wert α muss so gewählt werden, dass für alle Kanten die Zulässigkeit gewährleistet wird. Demnach bestimmen wir

$$\begin{aligned}
\alpha_1 &:= \min \frac{1}{2}\{w_\pi(j) | e_j \notin W \text{ und inzidiert mit zwei Knoten aus } even(F)\} \\
\alpha_2 &:= \min\{w_\pi(j) | e_j \notin W \text{ inzidiert mit je einem Knoten in} \\
&\quad even(F) \text{ und } free(F)\}
\end{aligned}$$

und setzen

$$\alpha := \min\{\alpha_1, \alpha_2\}.$$

Mit den hinzukommenden Kanten können wir nun solange einen alternierenden Wald aufbauen, bis wir einen augmentierenden Pfad gefunden haben oder wir nehmen erneut Veränderungen an den dualen Variablen vor. Nun müssen wir noch die Blütenknoten beachten. Nach dem Schrumpfen einer Blüte entsteht der neue Graph $G/\mathcal{B}$. Damit fallen die Knoten aus $\mathcal{B}$ weg und werden mit dem

Blütenknoten B ersetzt. Insbesondere werden die Kanten $e_j = \{v_i, v_i'\}$, die mit genau einem Knoten v_i aus $\mathcal{B}$ inzidieren, mit der Kante $\{B, v_i'\}$ identifiziert und demnach übernehmen wir auch die Werte $w_\pi(j)$ und $w(j)$ für diese Kanten. Finden wir auf einem geschrumpften Graphen G keinen augmentierenden Pfad, so verändern wir nach obigen Prinzip die dualen Variablen. D.h. wir verändern nun die Werte $w_\pi(j) = y_i + y_{i'} - \sum_{i=1}^{k} c_{ij} z_{B_i} - w_j$ durch die zwei dualen Variablen $y_{i'}$ und z_B. Damit behandeln wir den Blütenknoten B wie alle anderen Knoten auch, allerdings mit umgekehrten Vorzeichen. D.h. für $B_i \in even(W)$, setzen wir $z_{B_i} + \alpha$ und für $B_i \in odd(W)$ demnach $z_{B_i} - \alpha$. Es ergibt sich jedoch ein Problem für den Fall, dass $z_B = 0$ und $B \in odd(W)$ gelten. Dann würde $z_B - \alpha$ negativ werden und ist somit keine zulässige Lösung. Dieses Problem beheben wir, indem wir solche Blütenknoten B auflösen. Wir wählen dazu in der Blüte B einen Pfad P gerader Länge ausgehend von dem Knoten, der in der Blüte mit zwei Kanten aus $E \setminus M$ inzidiert. Den Knoten B ersetzen wir im Wald W mit dem Pfad P. Vom Restpfad der Blüte wählen wir alle M-Kanten und addieren auch diese zum Wald hinzu. Dann erst modifizieren wir die dualen Variablen.
Zur Bestimmung eines maximal gewichteten 1-Faktors setzen wir als initiale duale Variable

$$\begin{aligned} y_i &:= \max\{w_j/2 | e_j \text{ inzidiert mit } v_i\} \text{ für } i \in \{1, \ldots, n\} \\ z_{B_i} &:= 0 \text{ für } i \in \{1, \ldots, k\}. \end{aligned}$$

Offenbar handelt es sich um eine zulässige Lösung des dualen Programms. In der praktischen Umsetzung setzen wir für die dualen Variablen $z_{B_i} := 0$ im Algorithmus erst dann, wenn die Blüte B_i tatsächlich gefunden wurde. Nun verwenden wir Algorithmus 3.2.6, um einen augmentierenden Pfad auf dem Teilgraphen von G zu finden, für dessen Kanten e_j gilt $w_\pi(j) = 0$. Wir kehren nun wieder zu unserer unnummerierten Version der Variablen zurück, weil die Analogie zu Algorithmus 3.2.6 besser sichtbar wird.

Algorithmus 3.4.6 (Augmentierender Pfad,gewichtet)**.**
Eingabe: Ein Graph $G = (V, E)$ mit einem $(0,1)$-Faktor $H = (V, M)$ mit $|M| < \frac{1}{2}|V|$ und einer Kantengewichtsfunktion $w \in \mathbb{R}_+$. Für die Kanten $e \in M$ gilt $w_\pi(e) = 0$.
Ausgabe: Ein M-augmentierender Pfad auf G, für dessen Kanten e gilt $w_\pi(e) = 0$.
Initialisierung: Setze den alternierenden Wald $W := (V, F)$ mit $F := M$. Weiterhin setzen wir $e_v := \{v, u\}$ für jeden Knoten $v \in V$, wenn es ein $e_v \in E$ gibt mit $u \in X$ und wenn $w_\pi(e) = 0$ gilt. Wir sammeln die e_v in einer Liste.

While (kein augmentierender Pfad gefunden ist) do
If (es gibt ein $v \in V$, für das ein $e_v = (v, u)$ existiert) then

1. *If $(v \in free(W))$ then*

   - *Es gibt genau ein $w \in V$ mit $\{v, w\} \in M$. Für alle Knoten x, die mit w benachbart sind, setze $e_x := \{x, w\}$.*

- $F := F \cup \{e_v\}$.

2. *If* $(v \in even(W))$ *then*

   - *Finde die geraden Pfade* P, Q *von einem Knoten* $x \in X$ *nach* v *und von einem Knoten* $x' \in X$ *nach* w.
   - *If* $(x \neq x')$ *then* $P' = (P, e_v, Q)$ *ist ein augmentierender Pfad auf* G. *else*
     (P, e_v, Q) *enthält eine* M*-Blüte. Schrumpfe die Knotenmenge* $\mathcal{B}$ *der Blüte durch Konstruktion des Graphen* $G := G/\mathcal{B}$ *und* $M := M \setminus \mathcal{B}$. *Updaten der* e_v.

else

If (es gibt Blütenknoten $B \in odd(W)$ *mit* $z_B = 0$*) then*
setze $F := F \cup E(P) \cup N$ *für solche Blütenknoten* B *(wobei* $P = (v_i, \ldots, v_k)$ *ein gerader Teilpfad aus der Blüte* $B = (v_i, \ldots, v_t)$ *ist) und* N *aus den* M*-Kanten des restlichen Teilpfades besteht.*
Setze $y_v := y_v - \alpha$ *für alle* $v \in even(W)$,
$z_B := z_B + \alpha$ *für alle* $B \in even(W)$
$y_v := y_v + \alpha$ *für alle* $v \in odd(W)$
$z_B := z_B - \alpha$ *für alle* $B \in odd(W)$.
Updaten der e_v.

End While

Falls der Pfad P' *Blütenknoten enthält, konstruiere wie im Beweis zu 3.2.5 einen* M*-augmentierenden Pfad auf* G *durch Aufblasen der Blütenknoten.*

Das Updaten der dualen Variablen kann in der Zeit $O(|V|)$ durch geführt werden, so dass sich insgesamt zur Bestimmung eines augmentierenden Pfades ein Komplexität von $O(|V|^2)$ ergibt mit der gleichen Argumentation wie bei Algorithmus 3.2.6. Für Graphen, für deren Kantenmengen $|E| \in O(|V|)$ liegt, gibt es eine Implementierung mit einer verbesserten Komplexität. Gabow, Galil und Micali [8] erreichen mit Hilfe von intelligenten Datenstrukturen (Prioritätswarteschlangen) für die Suche nach einem augmentierenden Pfad die Komplexität $O(|E|log(|V|)$.

3.5 Der Algorithmus für einen maximal-gewichteten f-Faktor

Unser Algorithmus für einen maximal-gewichteten 1-Faktor kann wie im ungewichteten Fall auf den Splitgraphen $G_f = (V_f, E_f)$ angewendet werden, um einen maximal-gewichteten f-Faktor zu bestimmen. Es besteht auch hier die Möglichkeit, solange mit Algorithmus 3.4.6 nach augmentierenden Pfaden auf

dem Splitgraphen G_f zu suchen, bis ein 1-Faktor gefunden wurde. Dabei beginnen wir unsere Suche mit einem $(0,1)$-Faktor $H = (V_f, M)$, bei dem für alle äußeren Knoten $w \in V_f$ die Bedingung $d_H(w) = 1$ gilt. Dieser lässt sich bestimmen, indem in jedem bipartiten Graphen G_v von G_f genau $(d_G(v) - f(v))$ äußere Kanten als M-Kanten gewählt werden und zwar so, dass jeder äußere Knoten mit genau einer M-Kante inzidiert. Alle äußeren Kanten besitzen im Splitgraphen G_f nach Konstruktion maximales Kantengewicht W. Damit werden die dualen Variablen y_v alle mit dem Wert $y_v := W/2$ initialisiert. Insbesondere folgt für alle äußeren Kanten e_j die Bedingung $w_\pi(j) = 0$. D.h. die äußeren Kanten der bipartiten Graphen G_v stehen für den Aufbau alternierender Wälder zu jedem Zeitpunkt des Algorithmus zur Verfügung. Damit ist wegen Satz 3.3.2 klar, dass wir M-augmentierende Pfade auf dem Suchgraphen $G_f(M) = (V_f(M), E_f(M))$ finden können, wobei innere Kanten e mit $w_\pi(e) > 0$ nicht beachtet werden. Probleme bereiten uns jedoch die dualen Variablen und zwar dann, wenn die Suche nach einem M-augmentierenden Pfad nicht erfolgreich war. In diesem Fall verändern wir im Algorithmus 3.4.6 die dualen Variablen durch den zuletzt aufgebauten Wald W. Das geschieht bezüglich des Suchgraphen $G_f(M)$. Die dualen Variablen benötigen jedoch eine Modifikation bezüglich eines alternierenden Waldes von G_f. Klar ist, dass nach Veränderung von dualen Variablen nur innere Kanten von G_f bzw. $G_f(M)$ zum weiteren Aufbau eines Waldes hinzukommen können, welche einander eindeutig entsprechen. Es stellt sich die Frage, ob ein alternierender Wald W' von $G_f(M)$ einem alternierenden Wald W von G_f entspricht. Stellen wir uns vorerst vor, dass wir einen alternierenden Wald W' auf $G_f(M)$ konstruiert haben, der keine Blütenknoten B besitzt. Dann gibt es aufgrund der Waldstruktur von jedem Knoten v höchstens einen alternierenden Pfad zu einem anderen Knoten v' in W'. Enthält ein solcher Pfad die Knotensequenz $(w_{i,v}, e_{E\setminus M}, V_{i,v}, e_M, W_{i,v}, e_{E\setminus M}, v_{j,v})$, dann erhalten wir nach Streichen der Knoten $V_{i,v}$ und $W_{i,v}$ einen alternierenden Pfad auf G_f der die Knotensequenz $(w_{i,v}, e_{E\setminus M}, v_{j,v})$ enthält. Das ist möglich, weil G_f aus vollständigen bipartiten Graphen besteht. Dieses Prinzip führen wir gedanklich für alle alternierenden Pfade auf W' durch und erhalten einen alternierenden Wald W auf G_f. Damit gilt

$$\begin{aligned} even(W') \cap V_f &= even(W) \\ odd(W') \cap V_f &= odd(W) \\ free(W') \cap V_f &= free(W'). \end{aligned}$$

Somit können wir die dualen Variablen y_v mit Hilfe des alternierenden Waldes W' verändern und erhalten dann die dualen Variablen bezüglich des Splitgraphen G_f. Dabei bedenken wir, dass solche dualen Variablen nur bezüglich innerer Kanten verändert werden. Der Wert α lässt sich wie im letzten Abschnitt beschrieben, bestimmen. Schwieriger wird es, wenn Blütenknoten B im Wald W' enthalten sind. Es stellt sich die Frage, ob für jede Blüte im Suchgraphen eine Entsprechung im Splitgraphen existiert. Betrachten wir das folgende Beispiel.

Beispiel 3.5.1. *Der Blüte* $B = (v_4, W_1, V_1, w_2, v_2, \ldots, v_1, w_1, V_2, W_2, v_4)$ *entspricht im Splitgraphen die Blüte* $B = (v_4, w_2, v_2, \ldots, v_1, w_1, v_4)$.

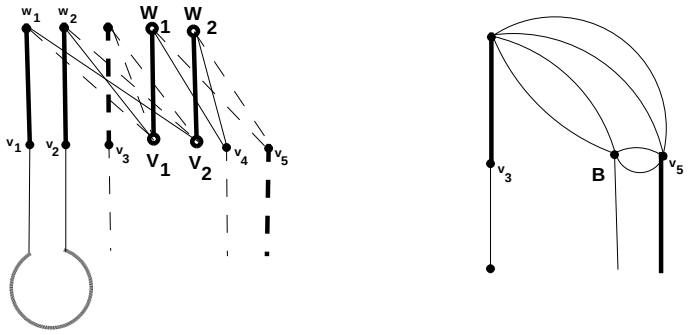

Abbildung 17: Die Blüte B in $G_f(M)$

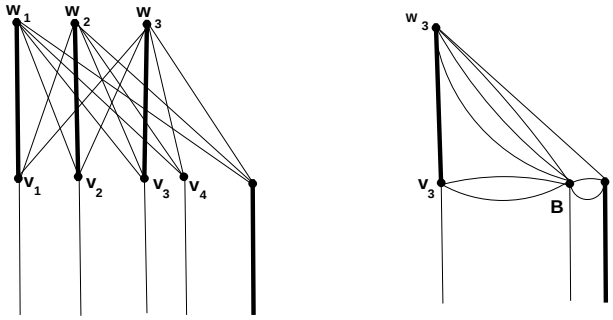

Abbildung 18: Die Blüte B in G_f

Für Blüten $B = (v_i, \ldots, v_t)$, für die $v_i \notin \{V_{1,v}, V_{2,v}, W_{1,v}, W_{2,v}\}$ ist, gibt es offensichtlich Entsprechungen in dem jeweiligen alternierenden Wald von G_f. Auch hier lassen sich die dualen Variablen mit dem Wald W von $G_f(M)$ mit den gleichen Argumenten ändern wie oben. Anders sieht es mit Blüten aus, bei denen v_i einem zusätzlichen Knoten entspricht. Sei eine Blüte $B = (v_i, \ldots, v_t)$ gegeben (siehe Definition 3.2.3), für deren Knoten $v_i \in \{V_{1,v}, V_{2,v}, W_{1,v}, W_{2,v}\}$ gilt, also ist v_i ein Knoten, der nicht in G_f enthalten ist. Solch eine Blüte nennen wir *Trugblüte*.

Beispiel 3.5.2. *Die Blüte $B = (V_1, w_1, v_1, \ldots, v_2, w_2, V_1)$ im Graphen G_f von Beispiel 3.5.1 ist eine Trugblüte von $G_f(M)$.*

Nach dem Schrumpfen einer Trugblüte zu einem Knoten B im alternierenden Wald zu $G_f(M)$ finden wir für G_f nach Streichen von nichtrelevanten Kanten nur einen Wald, der B als Knoten nicht enthält. Das heißt, eine Veränderung der dualen Variable z_B entspräche nicht unseren Erfordernissen. Dieser Fall tritt jedoch nicht ein. Wir wissen aus 3.4.6, dass die dualen Variablen z_B erst dann verändert werden, wenn ein alternierender Wald nicht weiter aufgebaut werden kann und kein augmentierender Pfad gefunden worden ist. Insbesondere

bedeutet das, dass wir duale Variablen verändern, die im alternierenden Wald W als Blütenknoten B enthalten sind, wenn dieser nicht weiter aufgebaut werden kann, keine weitere Blüte gefunden wird und kein M-augmentierender Pfad existiert. Solche Blüten nennen wir im folgenden *blockierende Blüten*.

Satz 3.5.3. *Ist B eine blockierende Blüte in einem M-alternierenden Wald W von $G_f(M)$, dann ist B keine Trugblüte.*

Beweis. Sei $B = (v_i, \ldots, v_t)$ eine Trugblüte in einem M-alternierenden Wald W auf $G_f(M)$. Dann gilt $v_i \in \{V_{1,v}, V_{2,v}, W_{1,v}, W_{2,v}\}$ für ein $v \in V$. Wir betrachten den Fall $v_i = V_{1,v}$. Solch eine Blüte muss innere Kanten des Graphen G enthalten, da in bipartiten Graphen keine Kreise ungerader Länge existieren. Insbesondere besitzt B dann mindestens fünf Knoten aus G_v. Außerdem gilt $V_{1,v} \in even(W)$, denn wäre $V_{1,v} \in odd(W)$, dann läge entweder v_{i+1} oder v_{t-1} in einer anderen Komponente als $V_{1,v}$. Dann wurde aber im Algorithmus ein Knoten aus $odd(W)$ durch eine Kante mit einer anderen Komponente verbunden. Widerspruch! Es gibt also den alternierenden Pfad P gerader Länge (ungleich Null) in W mit $P = (v_{i,v}, \ldots, W_{1,v}, V_{1,v})$. Wir nehmen im Folgenden an, dass die Knoten $V_{2,v}, W_{2,v}$ im Wald enthalten sind, also nicht durch eine Blüte geschrumpft wurden. Nun gibt es zwei Fälle.

1. $V_{2,v}$ befindet sich in einer anderen Komponente von W als $V_{1,v}$. Dann gilt entweder $V_{2,v}, W_{2,v} \in even(W) \cup odd(W)$ oder $V_{2,v}, W_{2,v} \in free(W)$. Im ersten Fall finden wir nach Schrumpfen von B einen augmentierenden Pfad, denn es gibt einen äußeren Knoten $w_{j,v} \in B$, der mit dem Knoten $V_{2,v}$ adjazent ist (oder es gibt eine Blüte B' in B, die ein $w_{j,v}$ enthält, dass mit $V_{2,v}$ adjazent ist), wenn $V_{2,v} \in even(W)$ gilt. Nach Schrumpfen der Blüte existiert dann die Kante $\{B, V_{2,v}\}$. Wir erhalten einen augmentierenden Pfad. Für $W_{2,v} \in even(W)$ gehen wir analog vor. Im zweiten Fall können wir den Wald vergrößern, durch die Kante $\{w_{j,v}, V_{2,v}\}$ die nach Schrumpfen von B zur Kante $\{B, V_{2,v}\}$ wird.

2. $V_{2,v}$ befindet sich in der gleichen Komponente von W wie $V_{1,v}$. Dann gibt es einen einen alternierenden Pfad von $v_{i,v}$ nach $V_{2,v}$ und demnach auch zu $W_{2,v}$. Es lässt sich in jedem Fall eine weitere Blüte konstruieren, denn wir wissen von der Existenz einer verbindenden Kante $\{w_{j,v}, V_{2,v}\}$ bzw. $\{v_{j,v}, W_{2,v}\}$ mit $w_{j,v}, v_{j,v} \in B$.

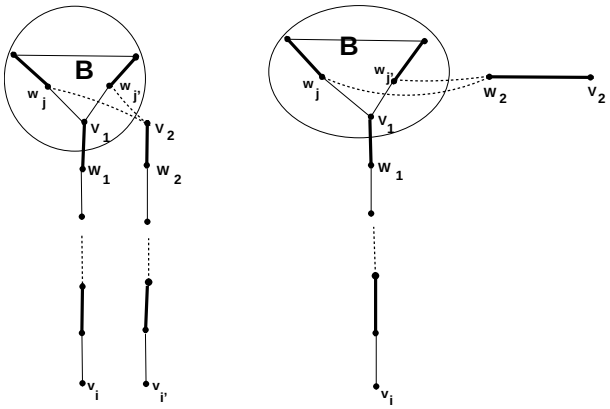

Abbildung 19: $V_2 \in even(W)$ und $V_2 \in free(W)$

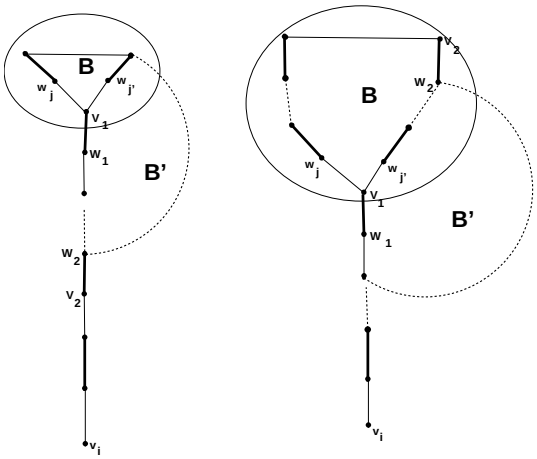

Abbildung 20: V_1 und V_2 liegen in einer Komponente

In beiden Fällen erweist sich, dass B keine blockierende Blüte sein kann. Angenommen die Knoten $V_{2,v}, W_{2,v}$ sind bereits durch Schrumpfen in Blütenknoten B' enthalten. Dann lösen wir diese Blüten auf, indem wir sie im Wald durch gerade Teilpfade ersetzen, so dass diese Knoten wieder im Wald enthalten sind. Die M-Kanten der restlichen Teilpfade solcher Blüten fügen wir als Komponenten von W hinzu. Nun behandeln wir unsere zwei Fälle von oben. Damit können wir insgesamt beweisen, dass B keine blockierende Blüte ist. □

Beispiel 3.5.4. *Die Blüte B' aus Beispiel 3.5.2 ist in der Blüte $B = (v_4, W_1, B', V_2, W_2, v_4)$ enthalten. Nach dem Schrumpfen von B erhalten wir die gleiche Situation wie in Beispiel 3.5.1.*

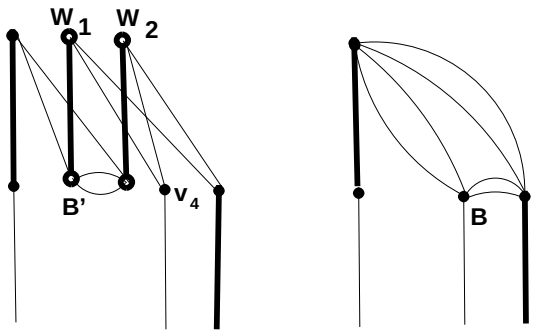

Abbildung 21: Blüte B in $G_f(M)$

Damit kann jede blockierende Blüte, die in $G_f(M)$ gefunden wurde, nach Streichen von Knoten, die nicht in G_f enthalten sind, zu einer Blüte in G_f konstruiert werden. Insgesamt bedeutet dies, dass wir duale Variablen einfach mit Hilfe alternierender Wälder auf $G_f(M)$ bestimmen können. Somit verwenden wir einfach das primale-duale Prinzip, um auf einem Suchgraphen $G_f(M)$, der nur Kanten e_j mit $w_\pi(j) = 0$ enthält, einen augmentierenden Pfad mit alternierenden Wäldern zu bestimmen. Endet diese Suche erfolglos, so ändern wir wie gewohnt unsere dualen Variablen mit Hilfe eines alternierenden Waldes W' von $G_f(M)$. Damit erhalten wir für die Suche nach einem augmentierenden Pfad von G_f die Komplexität $O(|E_f(M)|log(|V_f(M)|))$, also für das Auffinden eines f-Faktor die Komplexität $O(\sum_{v \in V} f(v)|E_f(M)|log(|V_f(M)|) = O(\sum_{v \in V} f(v)|E|log(|E|)) = O(\sum_{v \in V} f(v)|E|log(|V|))$.

4 Verwandte Probleme

4.1 Das Maxsquare Problem

Eine naheliegende Frage ist, ob sich die Konstruktion unseres Graphen G' aus Kapitel 2.3 auch zur Berechnung des Problems

$$Maxsquare(G,k) := \Big\{H=(V,F) | \sum_{v\in V} d_H(v) = 2k \text{ und } \sum_{v\in V} d_H(v)^2 = max\Big\}$$

eignet. Anders gefragt, was passiert bei der Bestimmung eines minimal gewichteten f-Faktors auf unserem Hilfsgraphen G'? Mit analogen Überlegungen ergibt sich für einen minimal gewichteten f-Faktor H' auf G'

$$w(G') - w(H') = -l(d_{G\setminus H}) + l(d_G).$$

Demnach berechnen wir mit einem minimal gewichteten f-Faktor das Minsquare Problem für einen Teilgraphen von G mit $(n-k)$ Kanten. Offensichtlich eignet sich der Hilfsgraph G' nicht zur Lösung des Maxsquare Problems. Tatsächlich stellt sich sogar heraus, dass dieses Problem NP-vollständig ist. Betrachten wir folgende Reduktion aus [3]. Wir konstruieren einen Hilfsgraphen $G_C = (V_C, E_C)$, indem wir jede Kante in G durch einen weiteren Knoten in zwei Kanten unterteilen. Sei $\triangle := \max\{d_G(v) | v \in V\}$. Dann fügen wir an jeden Knoten $v \in V$ weitere $\triangle - d_G(v)$ Kanten an. Somit besitzt jeder Knoten $v \in V \subseteq V_C$ aus G_C den Knotengrad $\triangle$. Betrachten wir nun eine Lösung des Problems Maxsquare$(G_C, r\triangle)$ mit $2 < r < \triangle|E_C|$. Wir erhalten einen Teilgraphen $H_C = (V_C, F_C)$ dessen Kantenmenge F_C aus den inzidenten Kanten von r Knoten mit Knotengrad $\triangle$ besteht, denn diese Knoten besitzen in G_C maximalen Knotengrad. Weiterhin müssen maximal viele dieser $r\triangle$ Kanten miteinander benachbart sein, da der Knotengrad von Knoten $v \in V_C \setminus V$ entweder eins oder zwei ist. Wenn $\frac{r(r-1)}{2}$ solcher Knoten den Knotengrad zwei besitzen, wird ein maximales Ergebnis erzielt. Das muss nicht existieren. Betrachten wir jedoch den Teilgraphen H auf G, der H_C entspricht, dann ist H in diesem Fall ein Teilgraph, bei dem r Knoten, den Knotengrad $(r-1)$ besitzen. Solch ein Teilgraph nennt sich *Clique* der Größe r. Angenommen Maxsquare$(G_C, r\triangle)$ wäre polynomiell lösbar, dann könnten wir auch in polynomieller Zeit bestimmen, ob G eine Clique der Größe $r > 2$ besitzt. Bei diesem Problem handelt es sich aber um ein bekanntes, klassisches NP-vollständiges Entscheidungsproblem. Demnach kann es für das Maxsquare Problem keinen polynomiellen Algorithmus geben, es sei denn, es gilt $P = NP$.

Beispiel 4.1.1. *Wir betrachten auf zwei verschiedenen Graphen G_C das Problem Maxsquare$(G_C, 3\triangle)$. Die Kantenmengen $\{1,\ldots,12\}$ sind jeweilige Lösungen. Die fettgezeichneten Kanten bilden den Teilgraphen H von G, dessen Knotenmenge aus den ausgefüllten Knoten besteht. Im ersten Graphen existiert keine Clique H der Größe drei, im zweiten gibt es eine.*

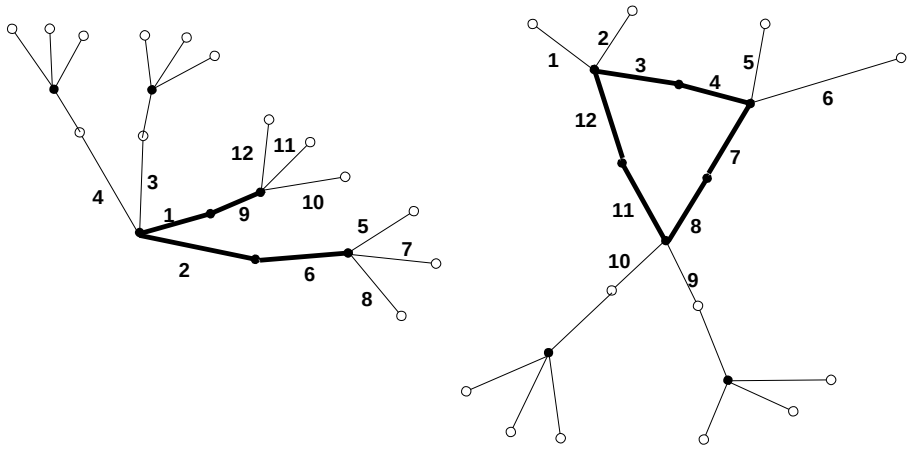

Abbildung 22: Reduktion des Maxsquare Problems

4.2 Das Maxfixcover Problem

Betrachten wir nochmals das Minsquare Problem auf einem Graphen $G = (V, E)$. Überschreitet die Größe der festen Kantenzahl k einen bestimmten Wert nicht, dann ist klar, dass ein $(0, 1)$-Faktor eine Lösung sein muss, denn dann wird die Knotengradsumme minimiert. Die Kantenmenge E' eines Teilgraphen $H = (V, E')$ von G, deren Kanten nicht benachbart sind, nennen wir *unabhängige Kantenmenge* oder auch *Matching*. Kennen wir $k' := |E'|$, dann ist Minsquare(G, k') die Bestimmung eines Matchings auf G. Unser Minsquare Problem verallgemeinert demnach die Bestimmung eines Matchings, also die Bestimmung einer unabhängigen Kantenmenge von G. Wie zu den meisten Problemen in der Graphentheorie gibt es auch in diesem Fall ein duales Problem, d.h. die Bestimmung einer unabhängigen Knotenteilmenge V' von einem zusammenhängenden Graphen G bei der zwei Knoten nicht durch eine Kante miteinander verbunden sind und demnach keine Knotenpaare bilden. Offenbar ist $V \setminus V'$ dann eine Knotenmenge bei der jeder Knoten mit mindestens einer Kante inzidiert. Das heißt diese Knotenmenge inzidiert mit allen Kanten von G. Die Knotenmenge $V \setminus V'$ nennt sich *Knotenüberdeckung* oder *Vertex Cover*. Analog zur Verallgemeinerung von Matchings gibt es auch zu Knotenüberdeckungen eine Verallgemeinerung. Wir suchen eine Knotenmenge $V'' := V \setminus V'$ mit $|V''| =: k''$, die mit maximal vielen Kanten aus G inzidiert. (D.h. dass sich möglichst wenige Knoten von V' paaren.) Dieses Problem nennt sich *MaxfixCover*. Sowohl Matchings, Knotenüberdeckungen als auch MaxfixCover lassen sich als Entscheidungsprobleme formulieren.

$$Matching(G, k') := \Big\{ E' \subseteq E || E'| = k', \text{E' ist unabhängige Kantenmenge von G} \}$$

$$VertexCover(G,k'') := \Big\{V'' \subseteq V || V''| = k'',\ \text{V" inzidiert mit allen Kanten aus E.}\Big\}.$$

$$Maxfix Cover(G,k'') :=$$
$$\Big\{V'' \subseteq V || V''| = k'',\ \text{V" inzidiert mit maximaler Anzahl Kanten aus E.}\Big\}$$

Matching(G,k') ist als Teilmenge von Minsquare(G,k') in polynomieller Zeit entscheidbar. (Das war schon vor [3] bekannt.) VertexCover(G,k'') ist ein bekanntes NP-vollständiges Problem und Maxfixcover(G,k'') als Obermenge demnach auch. Allerdings existieren polynomielle Lösungen für bestimmte Graphen, nämlich Kantengraphen. Ein Kantengraph $L(G)$ (Line Graph) von $G = (V,E)$ besitzt als Knotenmenge E. Weiter sind in $L(G)$ zwei Knoten genau dann durch eine Kante miteinander verbunden, wenn zwei Kanten $e \in E$ in G adjazent sind. Mit dieser Definition ist klar, dass Minsquare(G,k') äquivalent zum Problem MaxfixCover$(L(G),k'')$ ist. Damit lässt sich das MaxfixCover Problem auf einem Kantengraphen in polynomieller Zeit berechnen. Werden wir demnach mit einem MaxfixCover Problem konfrontiert, so lohnt es sich zu überprüfen, ob der zugrunde liegende Graph G ein Kantengraph ist.

5 Zusammenfassung

Das Minkonvex Problem ist sowohl als Modellierwerkzeug für praktische Probleme besonders gut geeignet, als auch für theoretische Fragen innerhalb der Graphentheorie sehr interessant. Kehren wir zu unserem Eingangsbeispiel zurück, so können wir das Minsquare Problem als gerechte Zuordnung der Größe k interpretieren. Mit der Anzahl n von Mathematikern und k Gutscheinen lässt sich ein ganzzahliger Mittelwert bestimmen, der angibt, wie häufig jeder Mathematiker essen gehen kann. Aufgrund der Beziehungen innerhalb der Mathematikerschaft, lässt sich dieser Wert nicht immer realisieren. Wir minimieren mit dem Minsquare Problem die Abweichung von diesem Wert bezüglich eines bestimmten Graphen G. Statt Gutscheinen könnten wir auch k Aufgaben, an denen je zwei Personen beteiligt sein müssen, gerecht verteilen. Erwähnenswert ist auch der Fall von bipartiten Graphen. Sehr bekannt ist das sogenannte Heiratsproblem, bei dem man möglichst viele Frauen und Männer unter Berücksichtigung von gegenseitigen Sympathien zweigeschlechtig miteinander verheiraten möchte. Denken wir an das letzte Kapitel, so handelt es sich um die Bestimmung eines maximalen Matchings auf einem bipartiten Graphen. Da das Minsquare Problem eine Verallgemeinerung darstellt, können wir das Heiratsproblem erweitern beispielsweise zu einem Tanzschulen Problem. An einem Abend sollen nacheinander Paare (Frau, Mann) unter Berücksichtigung ihrer Tanzfähigkeiten k verschiedene Tänze vorführen. Wenn k die Größe eines maximalen Matchings überschreitet, dann dürfen manche Personen mehr als einmal tanzen. Wieder

ist eine gerechte Lösung ein Minsquare Problem.
Besonders interessant am Minkonvex Problem ist die Interpretation als Verallgemeinerung von Matchingproblemen, die es ermöglicht, maximale Knotenüberdeckungen mit einer festen Anzahl von Knoten auf Kantengraphen in polynomieller Zeit zu bestimmen. Ein Problem, das für den allgemeinen Fall NP-vollständig ist. Weitere Verallgemeinerungen in unterschiedliche Richtungen durch weitere Bedingungen oder durch Lösen von der Graphenstruktur sind denkbar.

Danksagung

Herrn Prof. Hochstättler danke ich für das interessante Thema und die sehr gute Betreuung, Frau Dr. Ursula Berger dafür, dass Sie eine so hingebungsvolle Großmutter ist und meinen Freunden Bianca und Markus Pasche für die Versorgung mit Rotwein, Salat (und den damit verbundenen Privilegien) und guten Gesprächen.

Erklärung

1. Ich versichere, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.
2. Die eingereichte Arbeit ist nicht anderweitig als Prüfungsleistung verwandt worden.

Literatur

[1] Ahuja, R., Magnanti, T., Orlin, J., Network flows: Theory, Algorithms and Application, pp. 387.

[2] Aigner, M., Diskrete Mathematik, Friedr.Viehweg & Sohn Verlag/GWV Fachverlage Gmbh,pp. 295-322, Wiesbaden 2004.

[3] Apollonio, N., Sebö, A., Minsqare Factors and Maxfix Covers of Graphs. D.Bienstock and G. Nemhauser (Eds.): IPCO 2004, LNCS 3064, pp. 388-400, 2004.

[4] Appollonio, N., Sebö, A., Minkonvex Factors of Prescribed Size in Graphs. Technical Report 145, Les Cahlers du Laboratoire Leibniz, Grenoble, France, January 2006.

[5] Berger, A., Hochstättler, W., Minkonvex graph factors of prescribed size and a simpler reduction to weighted f-factors. http://www.fernuni-hagen.de/MATHEMATIK/DMO/pubs/feu-dmo 006-06.pdf, Technical Report 2006.

[6] Fremuth-Paeger, C., Jungnickel, D., Balanced Network Flows.1. A Unifying Framework for Design and Analysis of Matching Algorithms, John Wiley and Sons, Inc., pp. 4-5, 11-12, 1999.

[7] Fremuth-Paeger, C., Jungnickel, D., Balanced Network Flows. 7. Primal-Dual Algorithms, John Wiley and Sons, Inc., pp. 39-41, 2002.

[8] Gabow, H., An effizient reduction technique for degree-constrained subgraphs and bidirected network flow problems, 15th Annual ACM Symposium on the Theory of Computing, 25-27. April 1983, Boston, Massachusetts.

[9] Galil, Micali, Gabow, An O(|E||V|log|V|) algorithm for finding a maximal weighted matching in general graphs, SIAM Journal on Computing 15 (1986), 120-130.

[10] Gerards, A.M.H., Matching, in the Handbook of Operations Research and Management Science, Volume 7, Network Models, pp. 165-172, 1995.

[11] Pulleyblank, W.R., Matchings and stable sets, in Graham, Grötschel, Lovász (eds.), Handbook of Combinatorics (Elsevier Science, 1995).

[12] Schrijver, A., Combinatorial Optimization, Polyhedra and Efficiency, Volume A, Springer Verlag Berlin Heidelberg 2004.

[13] Tutte, W.T., A short proof of the factor theorem for finite graphs. Can. J. Math. 6 (1954), 347-352.