Abteilung Algorithmik **Winter 2020/21**
Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig

Dr. Linda Kleist
Phillip Keldenich
Dominik Krupke

Mathematische Methoden der Algorithmik
Übungsblatt 2 vom 4. November 2020

Die Abgabe eurer Lösungen zu diesem Blatt muss bis zum 18. November 2020 um 16:00 Uhr per E-Mail an `sven.langner@tu-bs.de` erfolgen. Am besten gebt ihr eure Lösung als einzelnes PDF-Dokument (LaTeX, Word oder Scan) ab. Ihr könnt auch handschriftliche Lösungen scannen oder abfotografieren; nach Möglichkeit fasst ihr auch in diesem Fall am besten die einzelnen Seiten zu einem PDF-Dokument zusammen. Achtet dabei bitte einerseits auf Lesbarkeit und andererseits auf eine vernünftige Dateigröße (höchstens etwa 2 MB/Seite). Vermeidet bitte auch das Versenden von ZIP-Archiven, da es hier möglicherweise zu Problemen mit dem Virenscanner auf dem Mailserver der TU kommen kann.

> **WICHTIG:** Zur Bescheinigung der Studienleistung müssen am Ende des Semesters mindestens 50% der Hausaufgabenpunkte erreicht worden sein.

Aufgabe 0 (Prüfungsfragen): Die kleine Übung soll auch als kontinuierliche Vorbereitung auf die Prüfung dienen, indem jeweils mögliche Prüfungsfragen zu den aktuellen Themen diskutiert werden.

Überlegt euch in eurer Hausaufgabengruppe 2 leichte, 2 mittlere und 2 schwere mögliche Prüfungsfragen zum Stoff der Vorlesungen 1–3 und gebt diese mit eurer Abgabe ab. Die Fragen werden bis zur kleinen Übung zusammengetragen und die besten Fragen werden dann kollaborativ in der kleinen Übung diskutiert.

Aufgabe 1 (Matching und Vertex Cover): Sei $G = (V, E)$ ein einfacher Graph. Seien VC_{opt} ein kardinalitätsminimales Vertex Cover, M ein beliebiges und M_{max} ein inklusionsmaximales Matching in G. Aus der Vorlesung ist bekannt, dass die Ungleichung $|M| \leq |VC_{opt}|$ gilt.

(a) Zeige: $|VC_{opt}| \leq 2 \cdot |M_{max}|$.

(b) Gib eine Klasse von Graphen an, für die $|VC_{opt}|$ beliebig groß werden kann und für die die Ungleichung aus (a) für beliebige M_{max} immer mit Gleichheit erfüllt ist (mit kurzer Begründung).

(c) Gib in eigenen Worten wieder, was Dualität ist und warum dieses Konzept wichtig für Optimierungsprobleme ist.

(d) Gib einen Polynomialzeitalgorithmus an, der ein kardinalitätsminimales Vertex Cover für Bäume berechnet. Neben der Lösung soll dieser Algorithmus auch ein Zertifikat für die Optimalität der Lösung liefern.

(10+5+5+10 Punkte)

Aufgabe 2 (Praktisches Lösen von Weighted Vertex Cover): Im Problem *Weighted Vertex Cover* haben wir einen Graphen $G = (V, E)$ und eine Knotengewichtung $w : V \to \mathbb{Z}^+$ gegeben und suchen eine Knotenmenge $C \subseteq V$ minimalen Gesamtgewichts $\sum_{v \in C} w(v)$, sodass jede Kante $e \in E$ zu einem Knoten aus C inzident ist. Für den Graphen mit Knoten $V = \{v_0, v_1, v_2, v_3, v_4\}$, Kanten $E = \{v_0v_2, v_1v_2, v_2v_4, v_1v_3, v_3v_4, v_2v_3\}$, und den Gewichten $w(v_0) = 2, w(v_1) = 3, w(v_2) = 10, w(v_3) = 11, w(v_4) = 1$ lässt sich das folgende Integer Program in CPLEX-Syntax aufstellen.

```
Minimize
2 v0 + 3 v1 + 10 v2 + 11 v3 + 1 v4

Subject To
v0 + v2 >= 1
v1 + v2 >= 1
v1 + v3 >= 1
v2 + v4 >= 1
v3 + v4 >= 1
v2 + v3 >= 1

Bounds
0 <= v0 <= 1
0 <= v1 <= 1
0 <= v2 <= 1
0 <= v3 <= 1
0 <= v4 <= 1

Integers
v0
v1
v2
v3
v4
End
```

a) Löse das Problem zunächst fraktional, d.h., unter Verzicht auf die Ganzzahligkeitsbedingungen, und dann integral. Hierfür kannst du den Javascript-Solver unter `http://hgourvest.github.io/glpk.js/` verwenden. Das Problem wird von diesem Solver genau dann integral gelöst, wenn die Funktion MIP (für *Mixed Integer Programming*) aktiviert ist. **Hinweis:** Der sich ergebende Funktionswert sollte im fraktionalen Fall 13.5 sein und im integralen Fall 14.

b) Stelle das IP für die folgenden Graphen auf und gebe die entsprechenden fraktionalen und integralen Lösungen an.
   - $G_1 = (\{v_0, v_1, v_2\}, \{v_0v_1, v_1v_2, v_0v_2\})$, $w(v_0) = w(v_1) = w(v_2) = 1$.
   - $G_2 = (\{v_0, v_1, v_2, v_3, v_4\}, \{v_0v_2, v_1v_2, v_2v_4, v_1v_3, v_2v_3\})$,
     $w(v_0) = 2, w(v_1) = 3, w(v_2) = 10, w(v_3) = 11, w(v_4) = 1$.

c) Probiere weitere Instanzen aus. Welche Werte können die Variablen im fraktionalen Fall annehmen? Was fällt dir bei Einschränkung auf bipartite Graphen auf?

(3+7+5 Punkte)

Aufgabe 3 (LP-Formulierung eines Optimierungsproblems): Formuliere das folgende Optimierungsproblem als LP. Zu n gegebenen Punkten $(x_i, y_i), 1 \leq i \leq n$, in der Ebene suchen wir eine Gerade, die das Maximum (nicht die Summe!) der vertikalen Abstände zu den Punkten minimiert. Was ist die optimale Gerade für folgende Punkte?

$$\{(2,8),(4,8),(4,7),(7,5),(8,3),(10,2)\}$$

Hinweis: Die meisten Solver nehmen per default $x \geq 0$ an. Um dies zu unterbinden muss 'x free' zu den Bounds hinzugefügt werden. Der Javascript-Solver akzeptiert nur Konstanten auf der rechten Seite. Die Constraints müssen also gegebenenfalls dementsprechend umgeformt werden. **(15 Punkte)**