



Ostfalia
Hochschule für angewandte
Wissenschaften

Fakultät Informatik

Hadi Dahnoun, 70486348

Forschungsaufgabe im Modul Quantitative Entscheidungslehre

an der
Ostfalia Hochschule für angewandte Wissenschaften
– Hochschule Braunschweig/Wolfenbüttel
Fachrichtung: Wirtschaftsinformatik (M. Sc.)

Erste/-r Prüfer/-in:

Prof. Dr. Torsten Sander

Diese Arbeit wurde im Rahmen der Lehrveranstaltung Quantitative Entscheidungslehre bearbeitet.

Aufgabenstellung

Mit dieser Forschungsaufgabe sollen Sie lernen, wie man mit Computerhilfe explorativ neue Erkenntnisse gewinnen diese als Vermutungen festhalten und idealerweise dann stringent logisch beweisen kann. Als Forschungsgegenstand betrachten wir Teilerfremdheitsgraphen. Wir konstruieren den Teilerfremdheitsgraph $T(n)$ mit n Ecken wie folgt: Als Eckenmenge nehmen wir die Zahlen $1, \dots, n$. Wir ziehen zwischen zwei unterschiedlichen Ecken x und y eine Kante genau dann, wenn die Zahlen x und y teilerfremd sind (also $\gcd(x, y) = 1$ gilt). Beispiel $n = 9$: Die Ecke 1 ist mit allen anderen Ecken verbunden, die Ecke 2 mit den Ecken 1, 3, 5, 7, 9, die Ecke 3 mit den Ecken 1, 2, 4, 5, 7, 8, die Ecke 4 mit den Ecken 1, 3, 5, 7, 9 und so weiter. Manche strukturellen Eigenschaften von (unnummerierten) Graphen prägen sich in den algebraischen Eigenschaften ihrer Adjazenzmatrix aus. Hierbei sind nur solche Eigenschaften interessant, die nicht von der konkreten Indizierung der Zeilen/Spalten der Matrix abhängen (entsprechend der Nummerierung der Ecken des betrachteten Graphen). Ein Beispiel dafür ist der Rang der Adjazenzmatrix. Diesen kennen Sie noch aus der linearen Algebra. Komplementär dazu ergibt sich die Nullität einer Matrix mit Format $n \times n$ und Rang r als die Zahl $n - r$. Wir wollen einen Graphen nullig nennen, wenn seine Nullität größer als Null ist. Bereits im Skript klang an, dass fast keine Graphen nullig sind. Insofern ist es interessant, Klassen nulliger Graphen aufzuspüren und die konkreten Nullitäten für die Graphen der Klasse vorsagen oder sogar aufgrund ihrer Struktur erklären zu können. Bearbeiten Sie folgende Teilaufgaben:

- a) Ermitteln Sie für aufsteigendes n für insgesamt möglichst viele Graphen $T(n)$ den jeweiligen Rang bzw. die Nullität der Adjazenzmatrix.
- b) Finden Sie ein Muster für die Entwicklung des Rangs abhängig von n . Schauen Sie dabei auf die Primfaktorzerlegung der jeweiligen Zahl n , um zu vorherzusagen, wie sich der Rang der Adjazenzmatrix von $T(n)$ von demjenigen für $T(n-1)$ unterscheidet.
- c) Betrachten Sie einige Beispielgraphen und gehen Sie wie folgt vor. Der betrachtete Graph G habe n Ecken und Adjazenzmatrix A . Berechnen Sie $y = Ax$ für einen beliebigen Vektor $x \in \mathbb{R}^n$. Alternativ schreiben Sie die Werte von x an die Ecken des Graphen G (gemeint: den Wert der i -ten Komponente an die Ecke i). Ermitteln Sie für jede Ecke die Summe der Werte aller zu dieser Ecke benachbarten Ecken. Vergleichen Sie Ihr Ergebnis mit y . Was fällt auf? Finden Sie eine allgemeingültige Erklärung dafür.

Teilaufgabe A

In diesem Teil wurde mithilfe eines Python-Skripts die Adjazenzmatrizen für den **Teilerfremdheitsgraph $T(n)$** erstellt. Hier ist, wie es funktioniert:

1. **Matrixinitialisierung:** Die Matrix A wird als $n \times n$ Matrix mit Nullen initialisiert. Diese Konfiguration entspricht einem Graphen mit n Eckpunkten und keinen Kanten.
2. **Betrachtung der Eckpunkte:** Das Skript iteriert über alle Paare von Eckpunkten i und j von 1 bis n . Im Kontext dieser Aufgabe entsprechen diese Eckpunkte den ganzen Zahlen von 1 bis n .
3. **Kantenbedingung:** Für jedes Paar (i,j) prüft es, ob i und j teilerfremd sind, also ob $\gcd(i,j)=1$. Wenn sie teilerfremd sind und $i \neq j$ wird das entsprechende Element in der Matrix $A[i-1][j-1]$ auf 1 gesetzt. Dieser Schritt stellt eine Kante zwischen den Eckpunkten i und j im Graphen dar, gemäß der Regel, dass eine Kante genau dann existiert, wenn die entsprechenden Zahlen teilerfremd sind.

```
def create_adjacency_matrix(n):
    A = np.zeros((n, n), dtype=int)
    for i in range(1, n+1):
        for j in range(1, n+1):
            if i != j and gcd(i, j) == 1:
                A[i-1][j-1] = 1
    return A
```

mit Berücksichtigung der Symmetrie:

```
def create_adjacency_matrix(n):
    A = np.zeros((n, n), dtype=int)
    for i in range(1, n + 1):
        for j in range(i + 1, n + 1):
            if gcd(i, j) == 1:
                A[i-1][j-1] = 1
                A[j-1][i-1] = 1 # Use symmetry
    return A,n
```

danach erfolgt die Berechnung von Rang und Nullität auch mit Python (np Library). Das beschriebene Skript wurde für $n=10$, $n=100$ und $n=1000$ ausgeführt. (siehe Abbildungen 1, 2 und 3).

ABBILDUNG 1 N=10

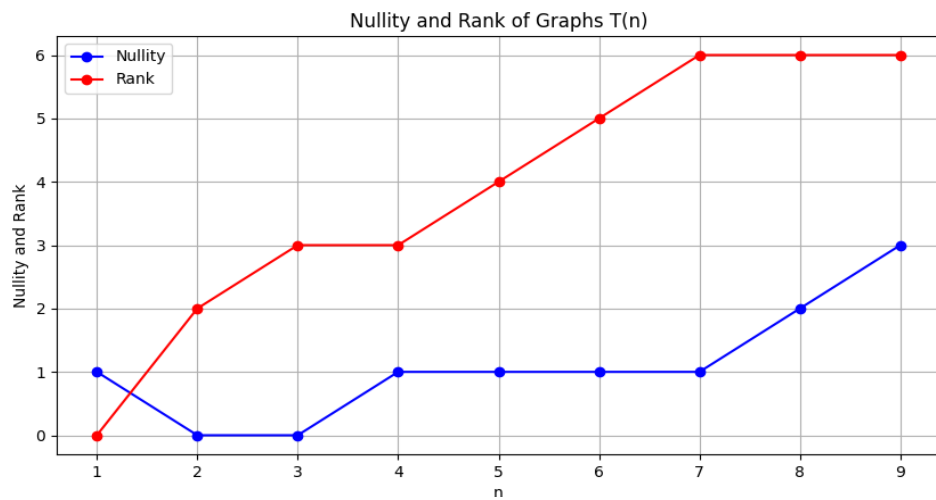


ABBILDUNG 2 N=100

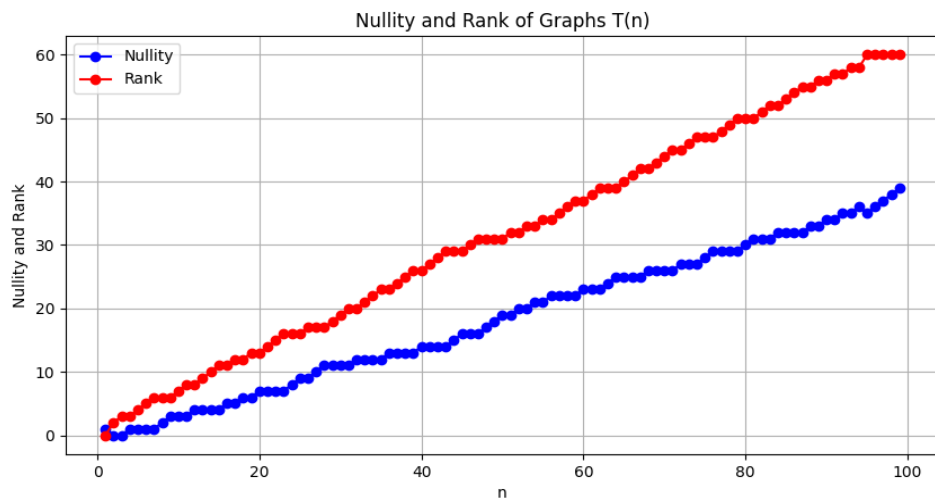
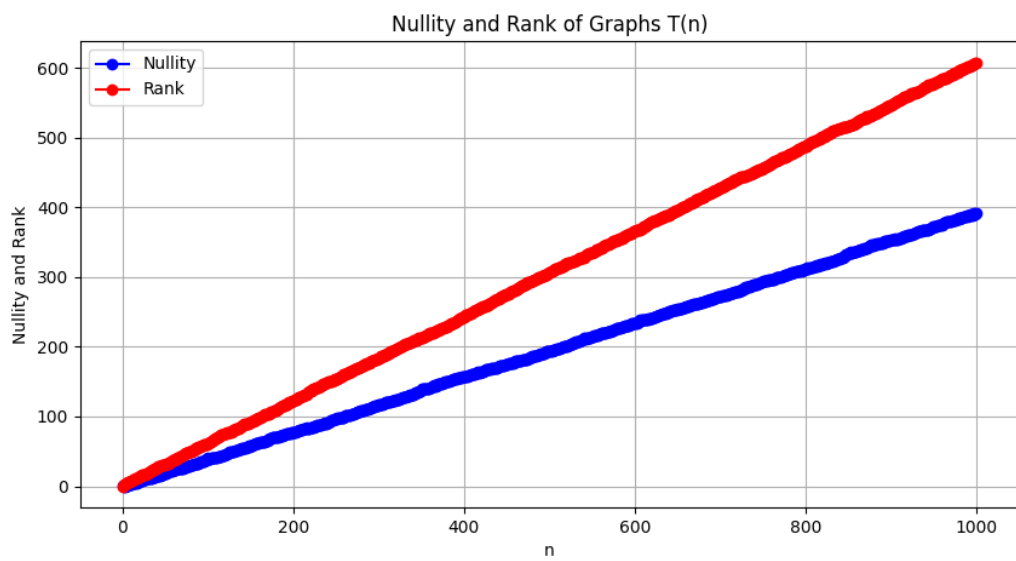


ABBILDUNG 3 N=1000



Teilaufgabe B

Aus dem Diagramm geht hervor, dass der Rang der Adjazenzmatrix linear zunimmt, wenn n zunimmt. Dies deutet darauf hin, dass die Anzahl der linear unabhängigen Zeilen in der Adjazenzmatrix im Allgemeinen zunimmt, wenn dem Graphen mehr Knoten (Zahlen) hinzugefügt werden. Dies ist wahrscheinlich, weil jeder neue Knoten n Verbindungen (Kanten) zu anderen Knoten einführt, die teilerfremd sind zu n . Um das Muster des Rangs der Adjazenzmatrix zu analysieren für zunehmende n und zu sagen, wie sich der Rang von $T(n-1)$ bis $T(n)$, insbesondere unter Berücksichtigung der Primfaktorzerlegung von n , wird eine Tabelle erstellt, um auch Änderungen im Rang von $n-1$ bis n zu analysieren und diese Änderungen mit der Primfaktorzerlegung von n in Beziehung zu setzen. Es werden Graphen mit $n=1$ bis zu $n=18$ berücksichtigt.

n	Primfaktorzerlegung	Rang
1	1	0
2	2	2
3	3	3
4	2^2	3
5	5	4
6	$2 \cdot 3$	5
7	7	6
8	2^3	6
9	3^2	6
10	$2 \cdot 5$	7
11	11	8
12	$2^2 \cdot 3$	8
13	13	9
14	$7 \cdot 2$	10
15	$5 \cdot 3$	11
16	2^4	11
17	17	12
18	$2 \cdot 3^2$	12

	Primzahl
	neue Kombination von Primzahlen

Die Tabelle zeigt deutlich, dass die Betrachtung einer Primzahl zu einer Verbindung mit allen anderen Ecken führt, mit Ausnahme der Verbindung zu sich selbst, was typischerweise zu einer Erhöhung des Ranges beiträgt.

Der Rang erhöht sich auch, wenn sich die Struktur seiner Primfaktorzerlegung erheblich verändert. Basierend auf den gegebenen Daten (bis $n=18$) sind hier die spezifischen Fälle, unter denen der Rang steigt (Außer wenn n eine Primzahl ist):

neue Primfaktor-Kombination wird eingeführt:

- $N=6$ (Erstes Auftreten von 2×3) → Rang steigt auf **5**
- $N=10$ (Erstes Auftreten von 2×5) → Rang steigt auf **7**
- $N=14$ (Erstes Auftreten von 7×2) → Rang steigt auf **10**
- $N=15$ (Erstes Auftreten von 5×3) → Rang steigt auf **11**

Es ist ebenfalls auffallend, dass der Rang nicht erhöht wird:

- Wenn n eine Potenz einer bereits vorhandenen Primzahl ist (z. B. $4=2^2$, $8=2^3$, 3^2 , $16=2^4$), dann erhöht sich der Rang **nicht**.
- Wenn n eine zusammengesetzte Zahl ist, die nur aus bereits bekannten Primfaktoren besteht, **ohne eine neue Struktur einzuführen**, bleibt der Rang ähnlich. Bei $n=2 \times 3 \times 3=18$ wird beispielsweise der Rang nicht erhöht, wenn die Adjazenzmatrix die durch die Primzahlen 2, 3, und die Primfaktor-Kombination 2×3 verursachte Konnektivität bereits angemessen darstellt.

Allgemeiner Beispiel für $n=9$: Da die Teilerfremdheit einer Zahl mit 3 oder 9 durch ihre Teilbarkeit durch 3 definiert wird (d.h., wenn eine Zahl nicht durch 3 teilbar ist, wird sie sowohl mit 3 als auch mit 9 teilerfremd sein), werden sowohl die Ecke 3 als auch die Ecke 9 mit demselben Satz von Ecken verbunden sein, die nicht durch 3 teilbar sind.

Fazit:

Der Rang der Adjazenzmatrix erhöht sich typischerweise, wenn n entweder eine **neue Primzahl** einführt, die zuvor nicht in der Matrix aufgetreten ist, oder wenn n eine **neue Kombination von Primzahlen** bildet, die in dieser Form bisher nicht in der Struktur der Matrix enthalten war. Diese Erhöhung des Rangs tritt auf, weil solche neuen Primzahlen oder Primfaktorkombinationen **neue, linear unabhängige Verbindungsmuster** erzeugen, die die bisherigen Strukturen erweitern.

$$\text{Rank}(n) = (\text{Anzahl Primzahlen} \leq n) + (\text{Primfaktor-Kombinationen} \leq n)$$

Anwenden für $n=17$:

Anzahl Primzahlen kleiner oder gleich 17 (1,2,3,5,7,11,13,17)= 8

Anzahl Primfaktor-Kombinationen kleiner oder gleich 17 (2×3 , 2×5 , 2×7 , 3×5)= 4

➤ $\text{Rang}(17) = 8 + 4 = 12$

Teilaufgabe C

Interpretation der Matrix-Vektor-Multiplikation:

- Jede Zeile i von A stellt die Verbindungsbeziehungen der Ecke i mit allen anderen Knoten j dar.

$$a_{ij} = \begin{cases} 1 & , \text{ falls } \gcd(i,j) = 1 \\ 0 & \text{sonst} \end{cases}$$

- Das Skalarprodukt der i -ten Zeile mit x :

$$y_i = \sum_{j=1}^n a_{ij} x_j$$

y_i = Skalarprodukt von $[a_{i1}, a_{i2}, \dots, a_{in}]$ und $[x_1, x_2, \dots, x_n]$.

Dies berechnet die Summe von x_j für alle Knoten j , die mit Knoten i verbunden sind (da $a_{ij}=1$ für benachbarte Knoten und 0 sonst).

Repräsentation manuelle Addition \rightarrow

$$y_i = \sum_{\substack{j=1 \\ \gcd(i,j)=1}}^n x_j = \sum_{j=1}^n a_{ij} \cdot x_j$$

$y=Ax$ berechnet daher für jeden Knotenpunkt die Summe der Werte seiner Nachbarn.

Explizit abstrakter Beispiel für $n=7$

- Schritt 1: Konstruktion von $T(7)$ bzw. Adjazenzmatrix A für $T(7)$

Wir betrachten den Graphen $T(7)$ mit den **Knoten** $\{1,2,3,4,5,6,7\}$, wobei eine **Kante** zwischen zwei Knoten i und j existiert, wenn: $\gcd(i, j) = 1$

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

- Schritt 2: Berechnung von $y=Ax$ für einen Beispielvektor x

für beliebigen Vektor x , $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$ ergibt sich $y_i = \sum_{j=1}^7 a_{ij} x_j$

$$y_1 = x_2 + x_3 + x_4 + x_5 + x_6 + x_7$$

$$y_2 = x_1 + x_3 + x_5 + x_7$$

$$y_3 = x_1 + x_2 + x_4 + x_5 + x_6$$

$$y_4 = x_1 + x_3 + x_5 + x_7$$

$$y_5 = x_1 + x_2 + x_3 + x_4 + x_6$$

$$y_6 = x_1 + x_3 + x_5 + x_7$$

$$y_7 = x_1 + x_2 + x_4 + x_6$$

das zeigt, dass y eine Linearkombination der Nachbarschaftssummen ist, da Adjazenzrelation (hier 1 für Adjazent & 0 für nicht Adjazent) mit Eckknotenwerte der Nachbarn multipliziert wird.

$\text{Rang}(A)=6$; für $n=7 \rightarrow$ der 7-dimensionalen Ausgabe y ist tatsächlich auf einen 6-dimensionalen Unterraum beschränkt. Man sieht, dass $y_2 = y_4$

Höherer Rang bedeutet, dass mehr Zeilen unabhängige Gleichungen zum System $y=Ax$ beitragen \rightarrow größere Vielfalt an möglichen Ausgaben.

Der Rang von A bestimmt, **wie viele unabhängige Nachbarschaftssummen** im Graphen wirklich eine **neue Information** liefern. Je höher der Rang, desto unabhängiger sind die Berechnungen im System $y=Ax$.