

Platz-Nr.: _____

Name: _____ Vorname: _____ Matrikel-Nr.: _____

BERGISCHE UNIVERSITÄT WUPPERTAL
FK 3: SCHUMPETER SCHOOL OF BUSINESS AND ECONOMICS

Master of Science

Wintersemester 2025/2026

Prüfungsgebiet: MWiWi 4.1 Advanced OR methods in Operations Management
Tag der Prüfung: 17.03.2026
Name des Prüfers: Prof. Dr. Bock
Erlaubte Hilfsmittel: Taschenrechner (nicht programmierbar)

Bearbeiten Sie **alle der fünf gegebenen Aufgaben** vollständig!

Die Lösungen zu den Aufgaben sollen gegliedert und in vollständigen zusammenhängenden Sätzen dargestellt und Rechnungen mit ihren Zwischenschritten nachvollziehbar sein. **Ein Ergebnis ohne nachvollziehbare Rechnung erhält keine Punkte.**

Die Darstellungsform und die Systematik der Gedankenführung gehen in die Bewertung ebenfalls ein. In Klammern ist für jede Aufgabe die Anzahl der maximal möglichen Punkte angegeben, die bei einer richtigen und vollständigen Bearbeitung erreicht werden können. Sie entspricht in etwa dem erwarteten Zeitbedarf in Minuten.

Insgesamt können **90 Punkte** erreicht werden. Für eine erfolgreiche Bearbeitung müssen wenigstens **45 Punkte** erworben werden.

Die Klausur besteht inklusive Deckblatt und Formelsammlung aus **6** Seiten.

Unterschrift: _____

Aufgabe 1 (Scheduling)**[22 Punkte]**

Die untenstehende Tabelle enthält die *Heads* a_i , die eigentliche Prozesszeiten p_i sowie die *Tails* q_i eines einstufigen Scheduling-Problems mit Vor- und Nachlauf bei dem insgesamt fünf Jobs zu bearbeiten sind. Zielsetzung ist die Minimierung des Makespan, der alle Nachläufe umfasst.

i	1	2	3	4	5
a_i	9	1	13	11	6
p_i	3	4	2	2	5
q_i	5	7	6	9	9

- a) Finden Sie mit dem Schrage-Algorithmus eine Lösung σ für obiges Scheduling-Problem. Bestimmen Sie den Critical Path und den Critical Job für diese Lösung. Ist die von Ihnen gefundene Lösung σ eine optimale Lösung? Falls ja, geben Sie den optimalen Zielfunktionswert an. Falls nein, geben Sie die nächste Verzweigung des Branch&Bound-Verfahrens von Carlier an. (12 Punkte)
- b) Nehmen Sie an, dass zu einem beliebigen, bereits optimal gelösten Scheduling-Problem mit n Jobs ein weiterer Job $i = 0$ hinzugenommen wird. Dieser Job hat oberste Priorität. Deshalb soll dieser Job 0 direkt bearbeitet werden, wenn er zum Zeitpunkt a_0 eintrifft. Der *Tail* des neuen Jobs ist gleich Null, d.h. es gilt $q_0 = 0$. Die Prozesszeit des Jobs ist p_0 . Geben Sie zwei unterschiedliche Konstellationen für die Werte von a_0 und p_0 sowie der gegebenen n Jobs an, sodass die Hinzunahme des neuen Jobs es nicht erforderlich macht, das Branch&Bound-Verfahren von Carlier erneut durchzuführen. (10 Punkte)

Aufgabe 2 (Cutting-Stock Problem)**[20 Punkte]**

Wir betrachten das folgende Cutting-Stock Problem, bei dem jede Rolle, die in Einheiten von insgesamt vier unterschiedlichen Finaltypen zerschnitten werden kann, die Kapazität $W = 17$ besitzt. Es gelten die folgenden Daten:

Produkttyp (Finaltyp) $i =$	1	2	3	4
Länge w_i	4	5	3	7
Anzahl b_i	11	13	21	9

- a) Überprüfen Sie, ob die ausschließliche Verwendung der Schnittmuster

$$\begin{pmatrix} 0 \\ 0 \\ 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

zu einer optimalen Lösung für die LP-Relaxierung des obigen Cutting Stock Problems führt. Zeigen Sie dazu die Optimalität der obigen Lösung oder geben Sie ein verbesserndes Schnittmuster an. Gehen Sie davon aus, dass die erste Dualvariable den Wert $\frac{1}{5}$ besitzt.

(15 Punkte)

- b) Wir unterstellen nun, dass die ausschließliche Verwendung der obigen Schnittmuster zu einer optimalen Lösung für die LP-Relaxierung des obigen Cutting Stock Problems führt. Geben Sie den Zielfunktionswert der ausschließlich mit den obigen Schnittmustern erzielbaren Lösung für die LP-Relaxierung des betrachteten Cutting Stock Problems an. Was bedeutet dieser Wert für die optimale Lösung des ursprünglichen Cutting Stock Problems? Wenn Sie unter a) keine Lösung ermitteln konnten, wählen Sie eine beliebige Lösung.

(5 Punkte)

Aufgabe 3 (Simulated Annealing)**[17 Punkte]**

Es sei das folgende *Quadratic Assignment Problem* gegeben:

	a	b	c
a	-	1	3
b	1	-	2
c	3	2	-

(Distanzmatrix)

	1	2	3
1	-	1	1
2	2	-	1
3	3	3	-

(Flussmatrix)

Führen Sie drei Iterationen (Übergangsversuche) des *Simulated Annealing* Verfahrens durch. Wählen Sie als Startlösung die Permutation $S_0 = (a, b, c)$ und nutzen Sie die lexikographische Reihenfolge für Tauschoperationen $\{(a, b), (a, c), (b, c), \dots\}$. Verwenden Sie in den einzelnen Iterationen k folgende Parameter für die Temperatur t_k und die Zufallsvariable X_k :

k	1	2	3
t_k	0,8	0,6	0,4
X_k	1,00	0,71	0,58

Aufgabe 4 (Thesen)**[15 Punkte]**

Nehmen Sie begründet Stellung zu den folgenden Thesen:

- Wir betrachten das Line-TSP mit N Kunden $1, \dots, N$ ohne release date (zero release dates) aber mit harten due dates und nehmen an, dass dieses Problem lösbar ist. Es gilt also $\forall i \in \{1, \dots, N\}: r_i = 0$ und d_i ist beliebig vorgegeben. Wenn es an jedem Kundenort eine gegebene von Null verschiedene konstante Servicezeit (handling time) gibt ($\forall i \in \{1, \dots, N\}: h_i = c > 0$), dann gibt es immer eine optimale zulässige Tour die jeden Kunden bedient, wenn der zugehörige Ort das erste Mal besucht wird. (5 Punkte)
- Wir betrachten die Zustände (states) (S, i, t) mit gegebenem $S \subseteq N' = \{1, 2, \dots, n\}$ und $i \in S$, die mit zulässigen Werten $t \in \mathbb{R}$ innerhalb des DP-Verfahrens von Dumas et al. (1995) zur Findung einer optimalen Lösung des TSPTW gebildet werden. Falls es einen Kunden $h \notin S$ gibt und es gilt $FIRST(S, i) + s_i > \min_{j \in S} LDT(i, j)$, dann können alle Zustände (S, i, t) für alle Werte $t \in \mathbb{R}$ gelöscht werden, da diese nicht mehr zu einer zulässigen optimalen Lösung erweitert werden können. (5 Punkte)

- c) Wir betrachten ein Rucksackproblem mit N Gütern und einem Rucksack mit Kapazität C . Wenn wir wissen, dass für dieses Rucksackproblem $C \leq N^4 + 123$ gilt lässt sich dieses Rucksackproblem für eine beliebig steigende Anzahl von Gütern in polynomieller Zeit optimal lösen. Mit anderen Worten: Es gibt ein optimales Lösungsverfahren, das jede Instanz dieses Problems mit N Gütern in asymptotischer Laufzeit $c \cdot N^k$ löst wobei c und k Konstanten sind. (5 Punkte)

Aufgabe 5 (Traveling Salesman Problem)

[16 Punkte]

Wir betrachten das asymmetrische Traveling Salesman Problem mit folgender Distanzmatrix

$$C = \begin{pmatrix} \infty & 2 & 2 & 1 & \infty \\ 2 & \infty & 6 & 4 & 1 \\ 6 & 3 & \infty & \infty & 4 \\ 6 & 5 & 3 & \infty & 6 \\ \infty & 2 & 4 & \infty & \infty \end{pmatrix}$$

- a) Zeichnen Sie das zugehörige Netzwerk. (4 Punkte)
- b) Führen Sie einen Branching Schritt des Branch&Bound-Verfahrens von Little et al (1963) durch. Bestimmen Sie dabei für jeden generierten Knoten die untere Schranke. Starten Sie mit dem folgenden Ergebnis aus der Initialisierung mit der Ungarischen Methode:

$$C = \begin{pmatrix} \infty & 1 & 1 & 0 & \infty \\ 0 & \infty & 5 & 3 & 0 \\ 2 & 0 & \infty & \infty & 1 \\ 2 & 2 & 0 & \infty & 3 \\ \infty & 0 & 2 & \infty & \infty \end{pmatrix}, \quad u = (1,0,0,0,0), \quad v = (1,1,3,3,2) \quad (12 \text{ Punkte})$$

Formeln:

$$\begin{aligned}
 J(j,l,k) &= \{i \mid j \leq i \leq l \wedge p_i < p_k\} \\
 V(\emptyset, t) &= 0 \text{ and } V(\{j\}, t) = \max\{0, t + p_j - d_j\} \\
 V(J(j,l,k), t) &= \min_{\delta} \left(\begin{aligned} &V(J(j, k' + \delta, k'), t) + \max\left(0, t + p_{k'} + \sum_{j \in J(j, k' + \delta, k')} p_j - d_{k'}\right) \\ &+ V\left(J(k' + \delta + 1, l, k'), t + \sum_{j \in J(j, k' + \delta, k')} p_j\right) \end{aligned} \right), \\
 \text{with } k' \in J(j,l,k) &\text{ is such that } p_{k'} = \max\{p_i \mid i \in J(j,l,k)\}
 \end{aligned}$$

$$\begin{aligned}
 LB &= \left\lfloor J\left(\frac{1}{2}, 1\right) \right\rfloor + \left\lfloor \frac{1}{2} \cdot \left\lfloor J\left[\frac{1}{2}, \frac{1}{2}\right] \right\rfloor \right\rfloor \\
 LB &= \left\lfloor \left\lfloor J\left(\frac{2}{3}, 1\right) \right\rfloor + \frac{2}{3} \cdot \left\lfloor J\left[\frac{2}{3}, \frac{2}{3}\right] \right\rfloor + \frac{1}{2} \cdot \left\lfloor J\left(\frac{1}{3}, \frac{2}{3}\right) \right\rfloor + \frac{1}{3} \cdot \left\lfloor J\left[\frac{1}{3}, \frac{1}{3}\right] \right\rfloor \right\rfloor \\
 C(M) &= \max \left\{ \sum_{i=0}^k t_{k, M+1-i} : k \in \left\{1, \dots, \left\lfloor \frac{N-1}{M} \right\rfloor\right\} \right\} \\
 LB &= \min\{M : C(M) \leq C\} \\
 E_j &:= \left\lfloor \frac{t_j + \sum_{h \in P_j^*} t_h}{C} \right\rfloor \quad \text{for } j = 1, \dots, N \\
 L_j(M) &:= M + 1 - \left\lfloor \frac{t_j + \sum_{h \in F_j^*} t_h}{C} \right\rfloor \quad \text{for } j = 1, \dots, N \\
 LB &= \min\{M \mid L_j(M) \geq E_j, \forall j\} \\
 L(j) &= \sum_{k \in \{h \mid h < j, h \in P_j^*\}} L(k) + 1
 \end{aligned}$$

$$\boxed{\text{If } x < e^{-\frac{\Delta}{t}}, \text{ then } S_0 = S}$$

$$\boxed{\forall i < i^* < j : U^+(i, j) = \min\{V^+(i, j-1) + x_j - x_{j-1}, V^-(i, j-1) + x_j - x_i\}}$$

$$\boxed{\forall i < i^* < j : U^-(i, j) = \min\{V^-(i+1, j) + x_{i+1} - x_i, V^+(i+1, j) + x_j - x_i\}}$$