Aufgabe 3-4-4

Job Shop Scheduling ist der englische Ausdruck für Maschinenbelegung. Im weitesten Sinne sind Arbeitsgänge, Aufträge, Tätigkeiten auf Abteilungen, Anlagen oder eben Maschinen zeitlich zu verteilen. Für jeden Job gibt es eine Auswahl möglicher Bearbeitungsfolgen; die Wahl einer konkreten Folge kann die Bearbeitungszeiten auf einzelnen Shops beeinflussen. Maschinenseitig gesehen kann die Job-Reihenfolge zu verschiedenen Rüstzeiten oder -kosten führen.

Job Shop Scheduling ist nun die Aufgabe, die Zeitplanung – welcher Arbeitsgang wann auf welcher Maschine – so vorzunehmen, dass ein vorher festzulegendes Zielkriterium erfüllt ist, wie etwa hier: Minimale Durchlaufzeit aller Jobs.

In dieser Aufgabe sei folgende Situation gegeben: Vier Jobs $j = 1$ bis 4 sollen auf drei Maschinen $\mathbf{M}i$, $i = 1, 2, 3$, eingeplant werden; jeder Job wird auf den drei Maschinen in der Reihenfolge erst $i = 1$, dann $i = 2$, dann $i = 3$ bearbeitet; die Operationszeiten sind in Tabelle 1 zusammengestellt:

Tabelle 1: Operationszeiten

[ZE]	**Job 1**	**Job 2**	**Job 3**	**Job 4**
M1	**1**	**2**	**3**	**4**
M2	**4**	**3**	**4**	**2**
M3	**2**	**1**	**3**	**4**

Für die Bearbeitungsfolge (**4 3 2 1**) ergibt sich beispielsweise eine Gesamtdurchlaufzeit von 20 [ZE]. Teil a) der Aufgabe zeigt, wo Verzögerungen bei der Bearbeitung auftreten.

In Vorbereitung auf die Anwendung eines Nachbarschaftssuchverfahrens sei die Nachbarschaftsstruktur zu einer Lösung wie folgt definiert: Der in der Bearbeitungsreihenfolge an Position 4 als letzter angegebene Job wird jeweils mit den übrigen Positionen in wie folgt getauscht: Position 4 mit 3, 4 mit 2, 4 mit 1; jede Reihenfolge hat somit genau drei Nachbarn. Die Durchlaufzeiten der neu entstehenden Reihenfolgen sind jeweils zu berechnen.

a) Zeichnen Sie ein Balkendiagramm zur Bearbeitungsreihenfolge (**4 3 2 1**), in dem an der x-Achse die Zeiteinheiten abgetragen sind.

b) Zeichnen Sie den Nachbarschaftsgraphen zu dem oben formulierten Maschinenbelegungsproblem. Es ist hinreichend, wenn Sie die (vollständigen) Nachbarschaftsbeziehungen für die vier, zu den Reihenfolgen (**1 2 3 4**), (**2 1 4 3**), (**3 2 4 1**) und (**4 1 2 3**) gehörenden Knoten eintragen. Verwenden Sie das vorgegebene Schema in Abbildung 1.

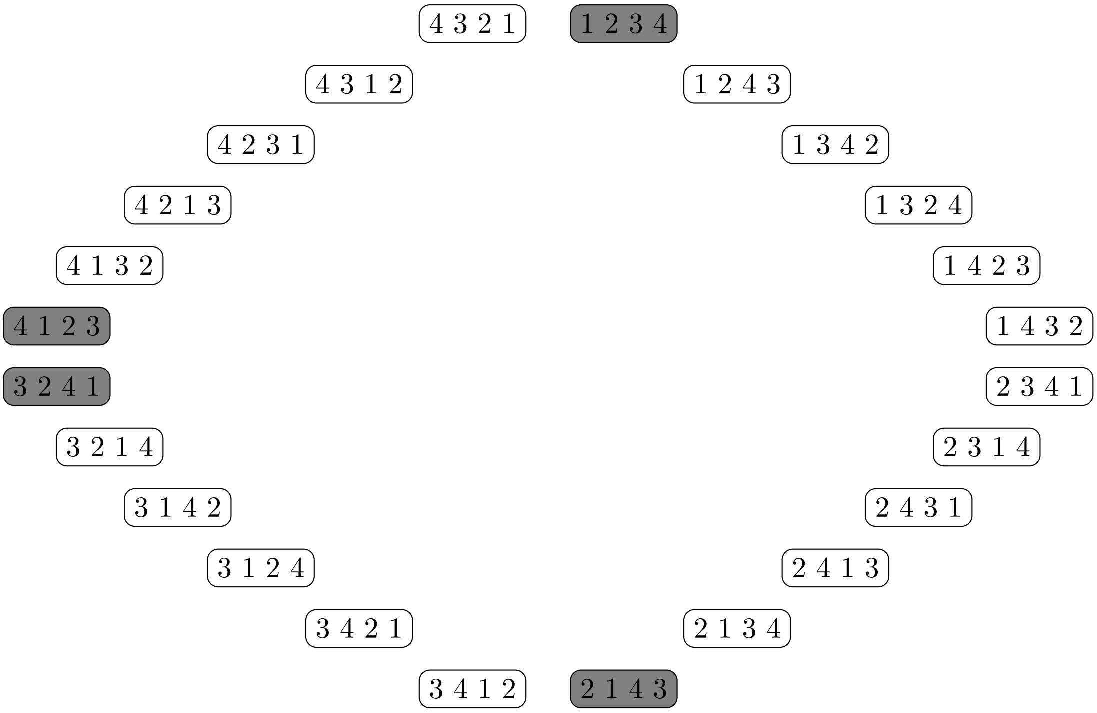

Abbildung 1: Schema des Nachbarschaftsgraphen

c) Führen Sie Simulated Annealing unter Anwendung des in Einheit 3 des Moduls beschriebenen Pseudocodes (Algorithmus 4.1) aus. Beenden Sie das Verfahren nach drei vollständigen Iterationen, d.h. nachdem dreimal eine Lösung akzeptiert wurde, oder brechen Sie ab, nachdem alle Elemente der direkten Nachbarschaft ohne Akzeptanz untersucht worden sind. Verwenden Sie den Kühlplan $T_{k+1} = \frac{1}{2} \cdot T_k$, $T_0 = 20$ sowie folgende Zufallszahlen R: 0,95 / 0,31 / 0,67 / 0,10 / 0,88. Notieren Sie Ihre Ergebnisse gemäß dem Schema in Tabelle 2.

Tabelle 2: Schema zur Erfassung der Ergebisse aus Simulated Annealing

aktuelle Lösung	f_{akt}	Nachbarlösungen	f_N	Δf	T_k	$p(\Delta f, T_k)$	R	akzept.
4 3 2 1	20				20			

Lösungshinweise

a) Das Balkendiagramm in Abbildung 2 zeigt deutlich, dass für die Bearbeitungsfolge (**4 3 2 1**) die Maschine 2 zwar bereits zum Zeitpunkt 6 frei ist, aber Job 3 auf **M**1 noch nicht abgeschlossen ist und somit erst zum Zeitpunkt 7 auf **M**2 wechseln kann. Insgesamt ergibt sich eine Durchlaufzeit für die vier Jobs von 20 [ZE].

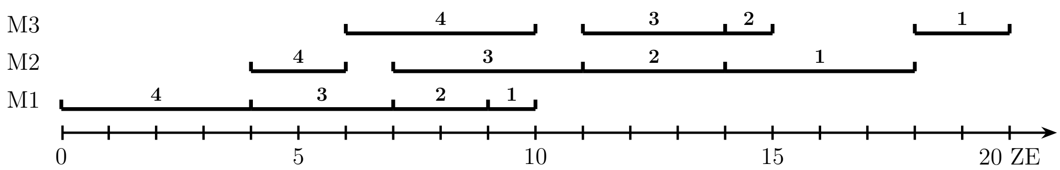

Abbildung 2: Balkendiagramm zur Bearbeitungsreihenfolge (**4 3 2 1**)

b)

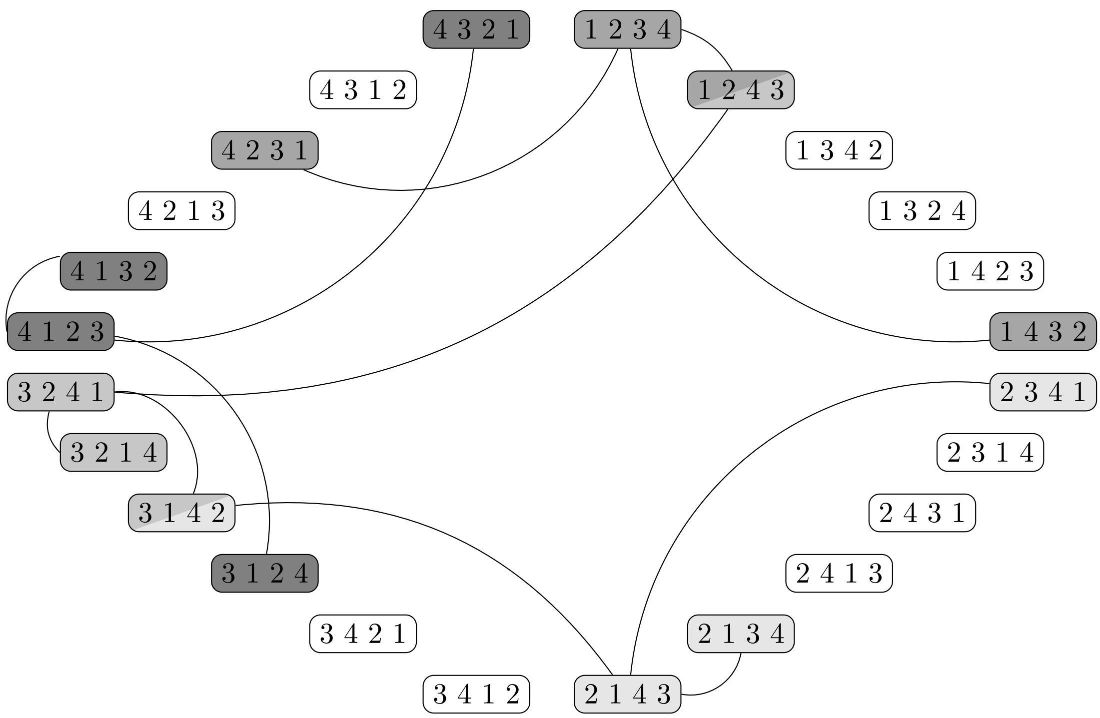

Abbildung 3: Nachbarschaftsbeziehungen (**1 2 3 4**), (**2 1 4 3**), (**3 2 4 1**), (**4 1 2 3**)

In Tabelle 3 sind zu den geforderten Bearbeitungsreihenfolgen zunächst die resultierenden Nachbarn notiert.

Tabelle 3: Lösung mit Simulated Annealing

(**1 2 3 4**)	(**1 2 4 3**)	(**1 4 3 2**)	(**4 2 3 1**)
(**2 1 4 3**)	(**2 1 3 4**)	(**2 3 4 1**)	(**3 1 4 2**)
(**3 2 4 1**)	(**3 2 1 4**)	(**3 1 4 2**)	(**1 2 4 3**)
(**4 1 2 3**)	(**4 1 3 2**)	(**4 3 2 1**)	(**3 1 2 4**)

In Abbildung 3 sind die Verbindungen eingezeichnet. Zu beachten ist, dass bspw. (**1 2 4 3**) sowohl Nachbar von (**1 2 3 4**) als auch von (**3 2 4 1**) ist.

c) Nachdem in Teil b) der Aufgabe der Generierung der Nachbarschaft verdeutlicht wurde, sind nun in Tabelle 4 die Iterationen in Simulated Annealing vollständig durchgeführt worden. Aufwendig ist die Bestimmung der Durchlaufzeiten zu den jeweiligen Reihenfolgen. Da die Auswahl einer Nachbarlösung zufällig erfolgt, wäre es an dieser Stelle hinreichend, den Wert für eine Reihenfolge zu ermitteln und die mögliche Akzeptanz gemäß Algorithmus zu prüfen. In Iteration zwei wurde zunächst ein Element abgelehnt und dann ein zweiter Nachbar überprüft. Nach dreimaliger Akzeptanz einer Lösung, wurde das Verfahren nicht weiter fortgesetzt.

Tabelle 4: Lösung mit Simulated Annealing

aktuelle Lösung	f_{akt}	Nachbarlösungen	f_N	Δf	T_k	$p(\Delta f, T_k)$	R	akzept.
4 3 2 1	20	**4 3 1 2**	19	-1	20			ja
		4 1 2 3	20					
		1 3 2 4	18					
4 3 1 2	19	**4 3 2 1**	20	1	10	0,91	0,95	nein
		4 2 1 3	20	1		0,91	0,31	ja
		2 3 1 4	19					
4 2 1 3	20	**4 2 3 1**	19	-1	5			ja
		4 3 1 2	19					
		3 2 1 4	22					