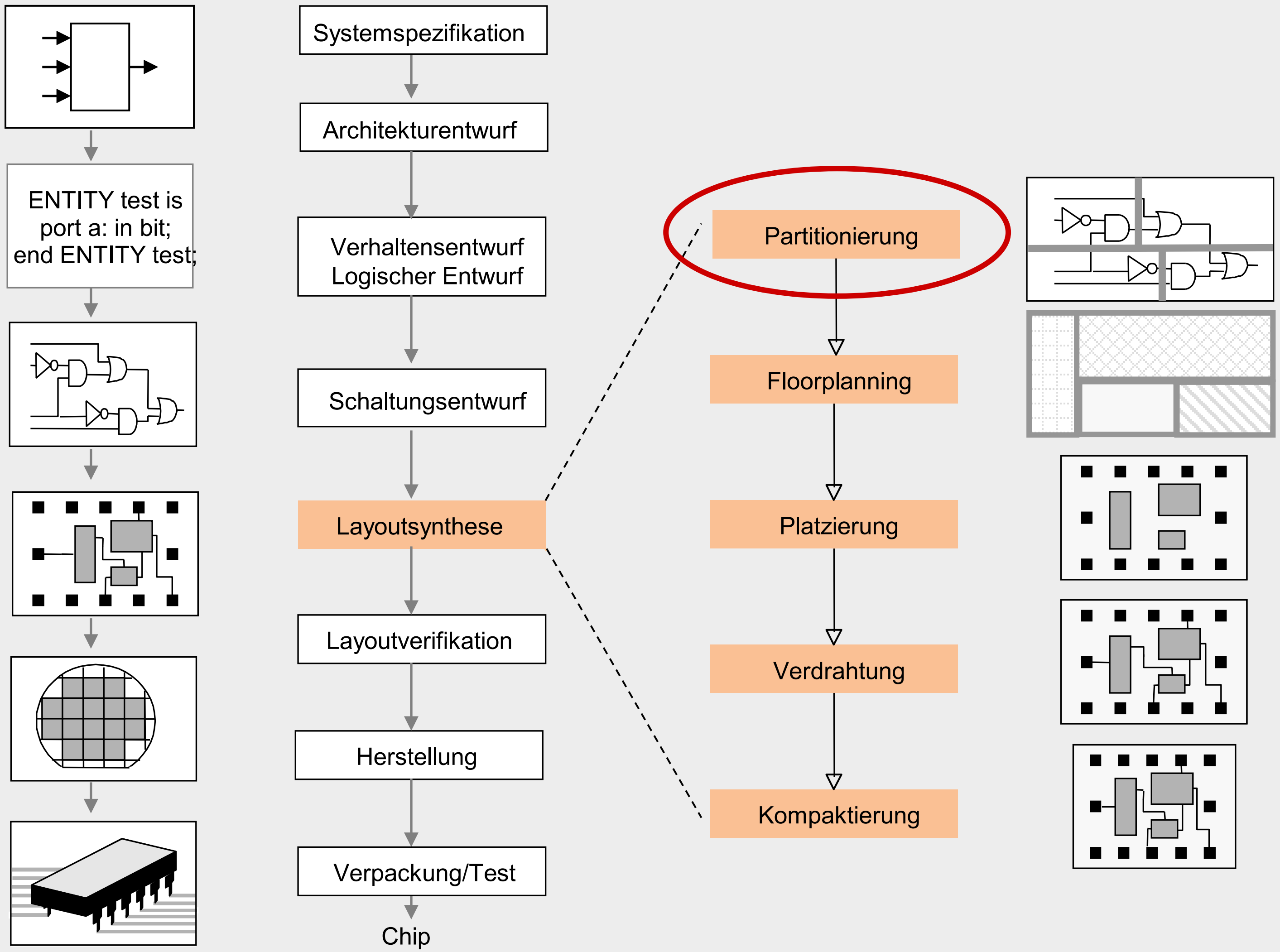
Gliederung Kapitel 2 – Partitionierung

2.1 Einführung



2.1 Einführung: Partitionierung auf Hierarchieebenen

Systemebene

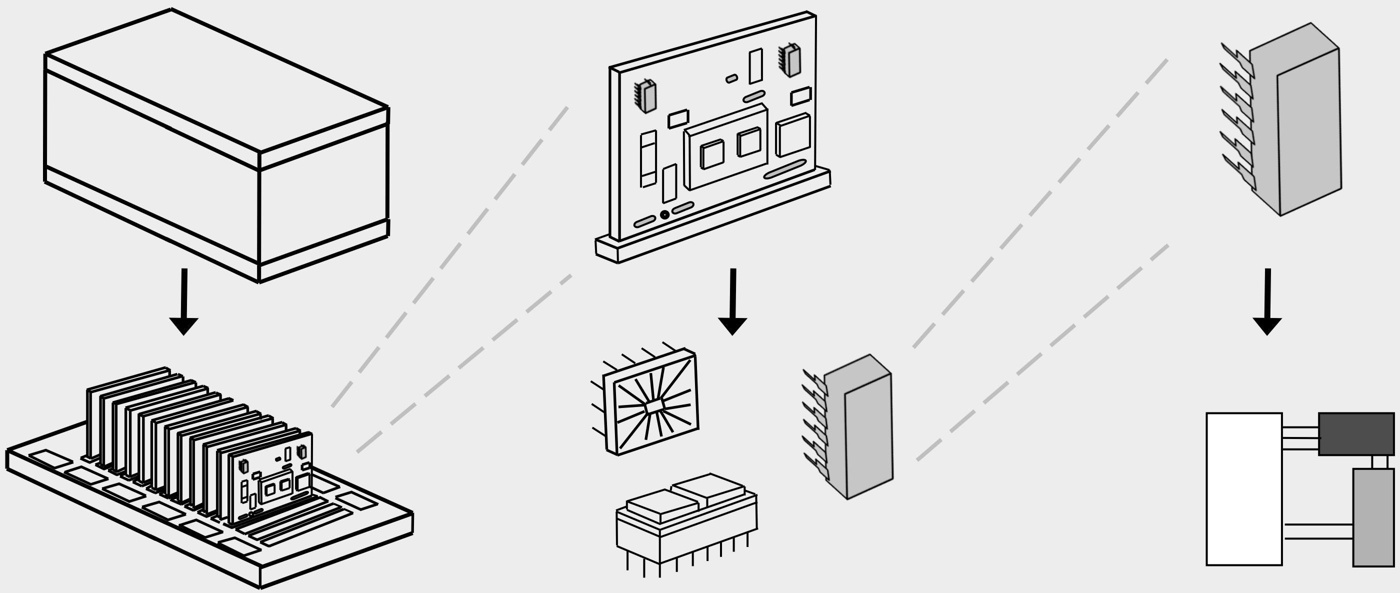
Jedes Subsystem (Leiterplatte) kann unabhängig entworfen und gefertigt werden.

Leiterplattenebene

Ermittlung von Teilschaltungen auf einer Leiterplatte, die als IC/MCM realisierbar sind.

IC-Ebene

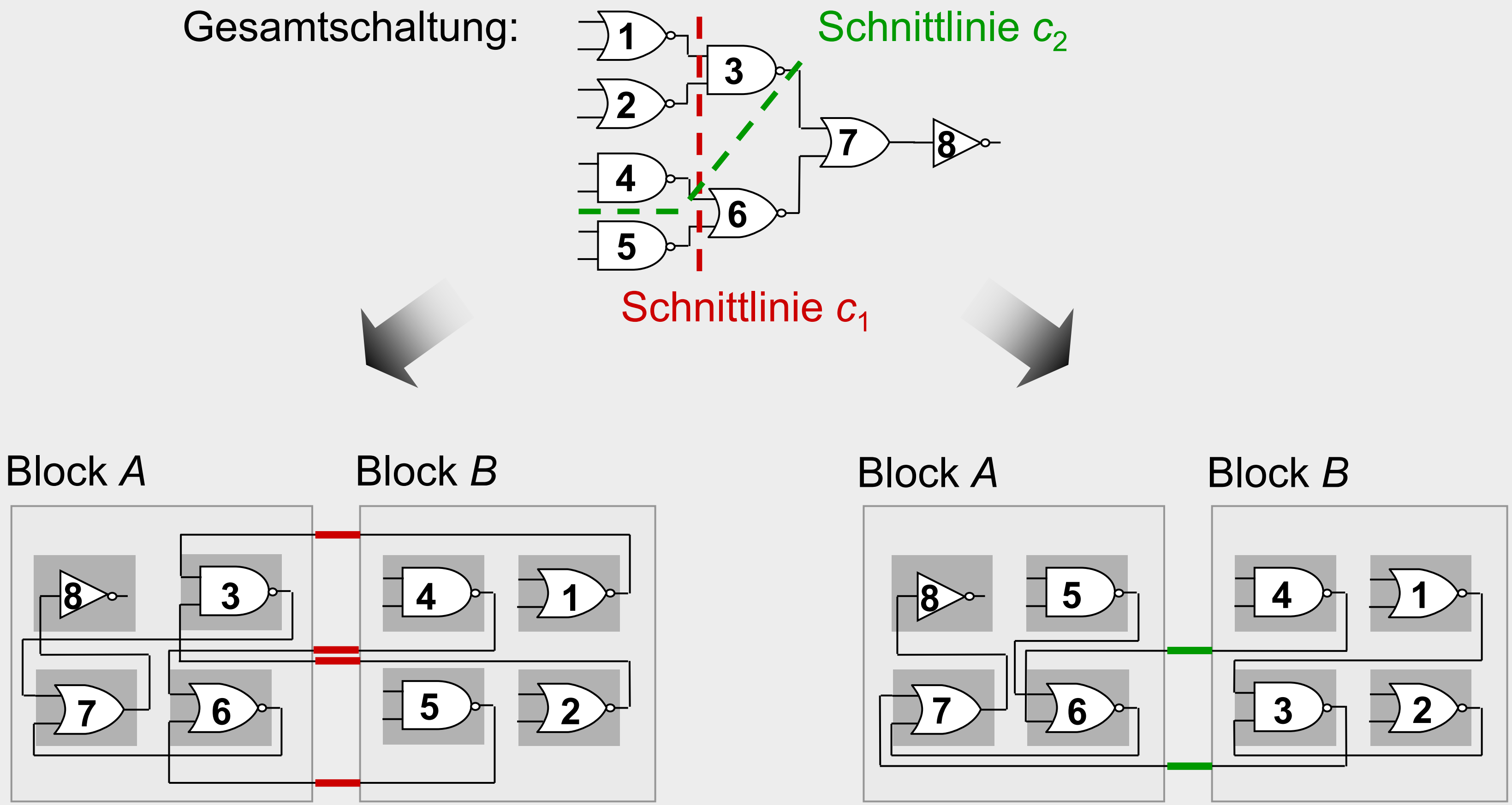
IC-Schaltungen werden aus Komplexitätsgründen oft in Blöcke aufgeteilt, die sich unabhängig voneinander entwerfen lassen.



2.1 Einführung

Die Aufgabe der Partitionierung besteht darin, eine Schaltung in Teilschaltungen, sog. Partitionen oder Blöcke, aufzuteilen (zu partitionieren), wobei die Verknüpfungen der Blöcke untereinander zu minimieren sind.

Gesamtschaltung:

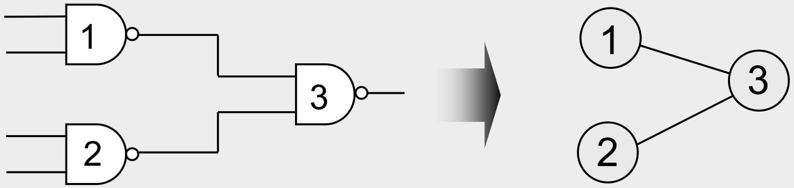


Schnittlinie c_1: vier externe Verbindungen

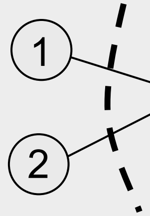
Schnittlinie c_2: zwei externe Verbindungen

2.1 Einführung: Prinzipielle Vorgehensweise

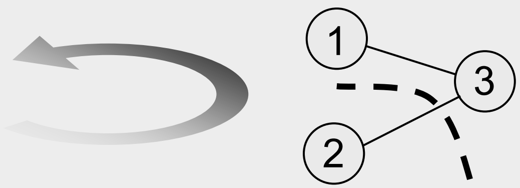
1. Abbildung der Schaltung in einem Graphen (Elemente = Knoten, Verbindungen = Kanten)



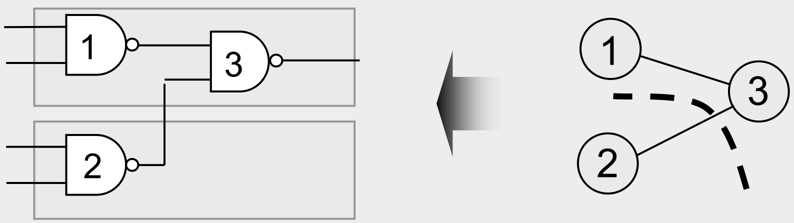
2. Ermitteln einer Anfangs-Partitionierung



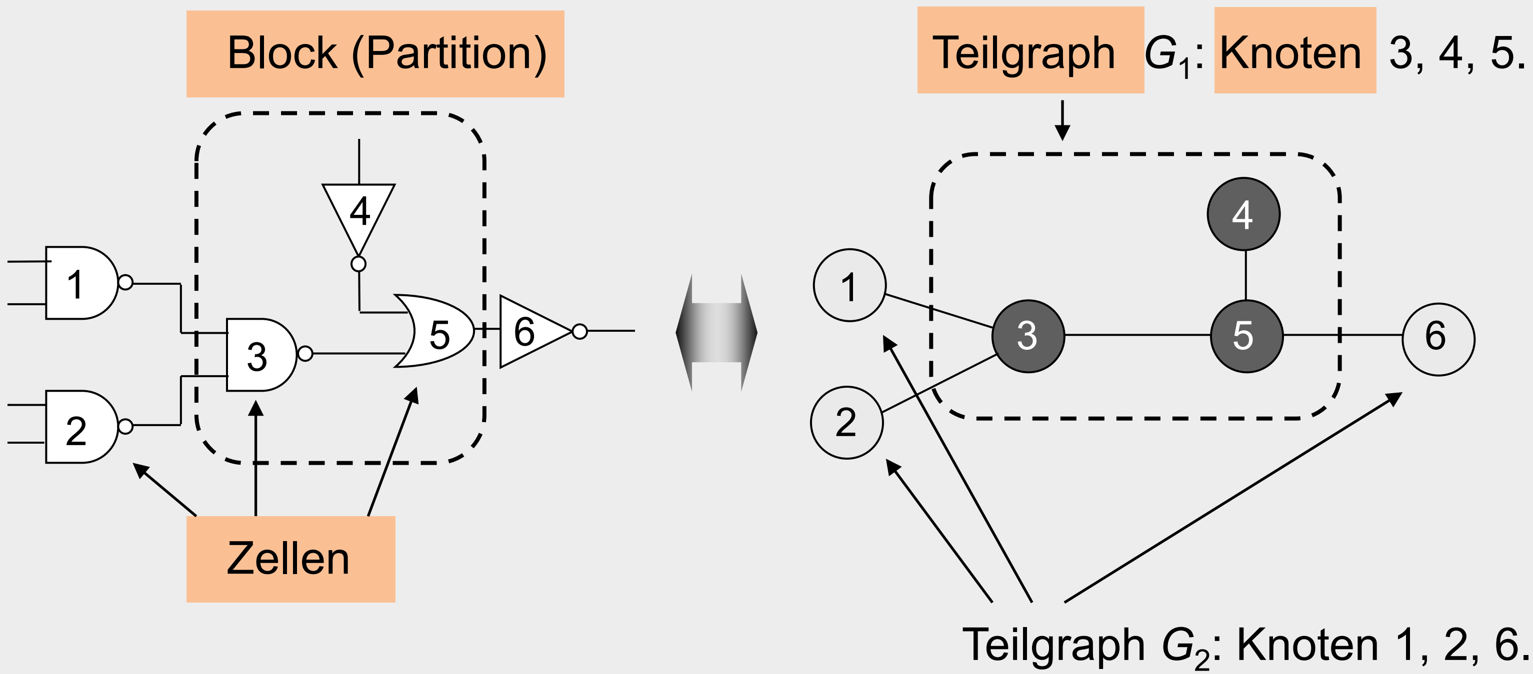
3. Iterative Partitionierungsverbesserungen unter Berücksichtigung der Optimierungsziele



4. Rücküberführung der Graphendarstellung in die Schaltungsabbildung



2.2 Begriffsbestimmungen



Geschnittene Kanten bzw.
Schnittmenge (Cutset):
(1,3), (2,3), (5,6),

Kapitel 2 – Partitionierung

2.3 Optimierungsziele

- Minimierung der Anzahl der externen Verbindungen
- Gleichmäßige bzw. vorgegebene Flächengröße der Blöcke (Bounded-Size-Partitionierung)

2.3 Optimierungsziel: Externe Verbindungen

- Netze der Schnittmenge (cutset) ψ verkörpern i.d.R. externe Verdrahtung
- Minimierung dieser Schnittmenge angestrebt
- Die Wichtung *w*(*e*) einer Kante *e* im Schaltungsgraphen stellt die Kosten dar, die bei der Realisierung dieser Verbindung als externe Verdrahtung entstehen.
- Damit ist Zielfunktion *Z* während der Partitionierung zu minimieren:

$$Z = \sum_{e \in \psi} w(e) \rightarrow \min .$$

2.3 Optimierungsziel: Bounded-Size-Partitionierung

- $s(v)$ sei die Fläche der Zelle v, V_i die Menge der Zellen des i-ten Blocks
- Vorgabe eines oberen Grenzwertes (Upper bound) für die Flächengröße einer jeden Teilschaltung (Block), wobei Fläche des i-ten Blocks mit $\sum_{v \in V_i} s(v)$ angegeben wird
- Ziel: Schaltung in k Blöcke ungefähr gleicher Größe zu teilen

$$|V_i| = \sum_{v \in V_i} s(v) \leq \frac{1}{k} \sum_{v \in V} s(v) = \frac{1}{k} |V|$$

mit $|V_i|$ = Flächengröße der Menge V_i (i-ter Block) und
$|V|$ = Flächengröße der Menge V (Gesamtschaltung).

- Wenn alle Schaltungselemente (Zellen, Bauelemente usw.) die gleiche Größe besitzen, reduziert sich die (Un-)Gleichung zu

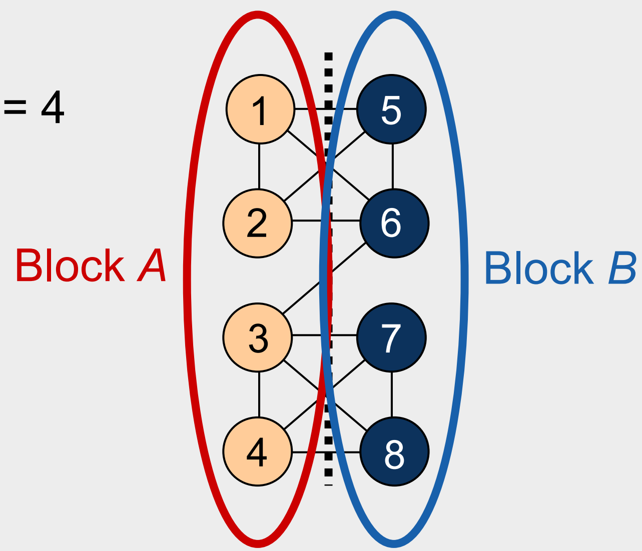
$$n_i \leq \frac{n}{k}$$

mit n_i = Anzahl der Elemente in V_i, n = Anzahl der Elemente in V, k = Anzahl der Blöcke.

Gegeben ist ein Graph mit insgesamt $2n$ Knoten und gewichteten Kanten zwischen beliebigen Knoten.

Gesucht ist die Aufteilung des Graphen in zwei Teilgraphen (Blöcke) mit jeweils n Knoten (Zellen) mit minimalen Schnittkosten (Summe der Gewichte aller geschnittenen Kanten).

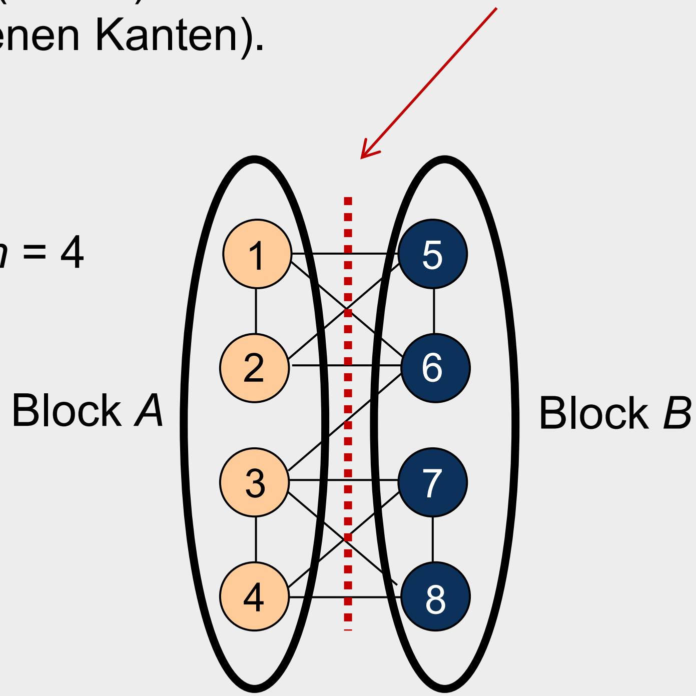
Beispiel: $n = 4$



Gegeben ist ein Graph mit insgesamt 2*n* Knoten und gewichteten Kanten zwischen beliebigen Knoten.

Gesucht ist die Aufteilung des Graphen in zwei Teilgraphen (Blöcke) mit jeweils *n* Knoten (Zellen) mit minimalen Schnittkosten (Summe der Gewichte aller geschnittenen Kanten).

Beispiel: $n = 4$



2.4.1 Kernighan-Lin (KL)-Algorithmus: Begriffe

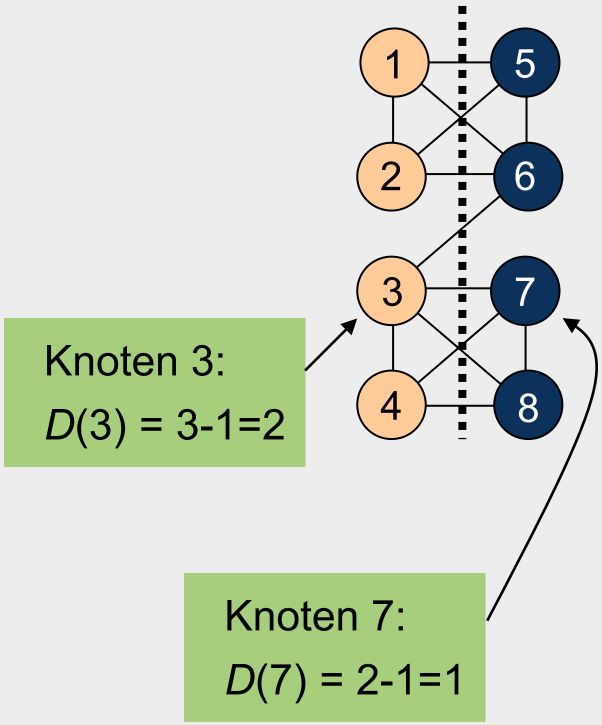
Kosten $D(v)$ eines Knotens v

$D(v) = \sum v_{extern} - \sum v_{intern}$,

wobei

$\sum v_{extern}$ die Anzahl der von der Schnittlinie geschnittenen Kanten des Knotens v und

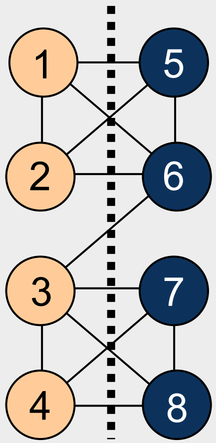
$\sum v_{intern}$ die Anzahl der nicht von der Schnittlinie erfassten Kanten des Knotens sind.



Hohe Kosten (positiver *D*-Wert) geben damit einen „Drang“ an, die Teilmenge zu wechseln, während niedrige Kosten (negativer *D*-Wert) ein gewisses „Bleiberecht“ verkörpern.

2.4.1 Kernighan-Lin (KL)-Algorithmus: Begriffe

Gewinnwert zweier Knoten *a* und *b*



$$\Delta g = D(a) + D(b) - 2 * c(a,b),$$

wobei

$D(a)$, $D(b)$ die Kosten der Knoten a, b

$c(a,b) = 1$, wenn zwischen a und b eine Kante existiert, andernfalls gilt $c(a,b) = 0$.

Δg gibt damit an, wie sinnvoll es hinsichtlich der Schnittkosten des Graphen ist, die beiden betrachteten Knoten auszutauschen.

Je größer Δg ist, umso mehr wird die Gesamtanzahl der geschnittenen Kanten verringert.

2.4.1 Kernighan-Lin (KL)-Algorithmus: Begriffe

Gewinnwert zweier Knoten *a* und *b*

$\Delta g = D(a) + D(b) - 2 * c(a,b)$,

wobei

$D(a)$, $D(b)$ die Kosten der Knoten *a*, *b*

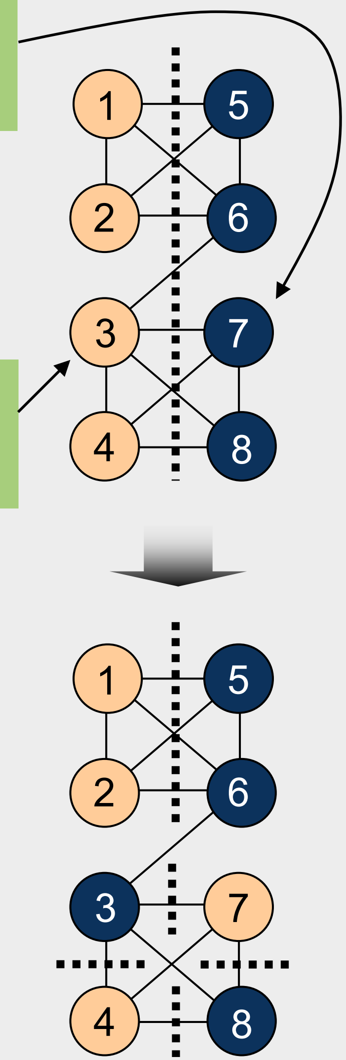
$c(a,b) = 1$, wenn zwischen *a* und *b* eine Kante existiert, andernfalls gilt $c(a,b) = 0$.

$\Delta g\,(3,7) = D(3) + D(7) - 2 * c(a,b) = 2 + 1 – 2 = 1$

=> Der Austausch der Knoten 3 und 7 würde die Schnittkosten um 1 verringern.

Knoten 7:
$D(7) = 2\text{-}1=1$

Knoten 3:
$D(3) = 3\text{-}1=2$



2.4.1 Kernighan-Lin (KL)-Algorithmus: Begriffe

Gewinnwert zweier Knoten *a* und *b*

$\Delta g = D(a) + D(b) - 2 * c(a,b),$

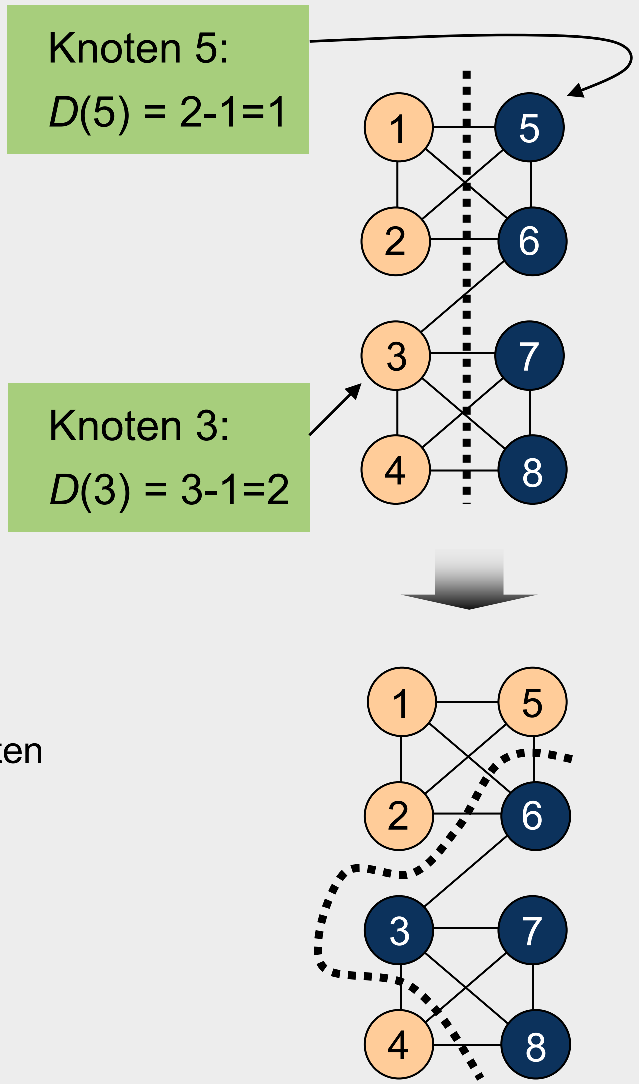
wobei

$D(a)$, $D(b)$ die Kosten der Knoten *a*, *b*

$c(a,b) = 1$, wenn zwischen *a* und *b* eine Kante existiert, andernfalls gilt $c(a,b) = 0$.

$\Delta g\,(3,5) = D(3) + D(5) - 2 * c(a,b) = 2 + 1 – 0 = 3$

=> Der Austausch der Knoten 3 und 5 würde die Schnittkosten um 3 verringern.



2.4.1 Kernighan-Lin (KL)-Algorithmus: Begriffe

Gewinnwert zweier Knoten *a* und *b*

Das Ziel besteht darin, die zwei Knoten *a* und *b* zu finden, deren Gewinnwert Δg am größten von allen möglichen Knotenkombinationen ist, und diese dann auszutauschen.

2.4.1 Kernighan-Lin (KL)-Algorithmus: Begriffe

Maximaler positiver Gewinn G_m eines Passes

Der maximale positive Gewinn G_m gibt die Folge von m Vertauschungen innerhalb eines Passes an, die zu einer maximalen Schnittkostenminimierung des Graphen führt.

Konkret ergibt sich G_m aus den aufaddierten Gewinnwerten Δg von m hintereinander erfolgten Vertauschungen innerhalb eines Passes.

2.4.1 Kernighan-Lin (KL)-Algorithmus

Schritt 0:

- V = Menge der $2n$ Knoten
- $\{A, B\}$ sei eine willkürliche Anfangspartitionierung

Schritt 1:

- $i = 1$
- Berechnung von $D(v)$ für alle Knoten $v \in V$

Schritt 2:

- Auswahl von a_i und b_i mit maximalem Gewinnwert $\Delta g_i = D(a_i) + D(b_i) - 2 * c(a_i b_i)$
- Vertauschen und Fixieren von a_i und b_i

Schritt 3:

- Wenn alle Knoten fixiert sind, weiter mit Schritt 4, andernfalls
- Neuberechnung der D-Werte für alle Knoten, welche nicht fixiert und mit a_i und b_i verbunden sind
- $i = i + 1$
- Weiter mit Schritt 2

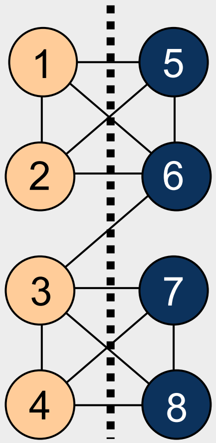
Schritt 4:

- Bestimmung der Vertauschungssequenz 1 bis m ($1 \leq m \leq i$), so dass $G_m = \sum_{i=1}^{m} \Delta g_i$ maximiert wird
- Wenn $G_m > 0$, weiter mit Schritt 5, andernfalls ENDE

Schritt 5:

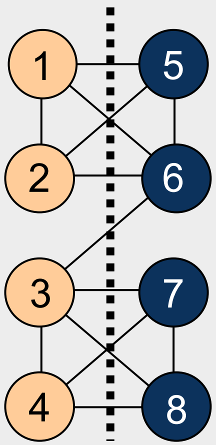
- Durchführen aller m Vertauschungen, Beseitigen aller Knotenfixierungen
- Weiter mit Schritt 1.

2.4.1 Kernighan-Lin (KL)-Algorithmus: Beispiel



Schnittkosten: 9
Nicht fixiert:
1,2,3,4,5,6,7,8

2.4.1 Kernighan-Lin (KL)-Algorithmus: Beispiel



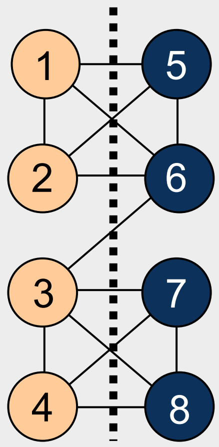
Schnittkosten: 9
Nicht fixiert:
1,2,3,4,5,6,7,8

Kosten $D(v)$ jedes Knotens:

$D(1) = 1$	$D(5) = 1$
$D(2) = 1$	$D(6) = 2$
$D(3) = 2$	$D(7) = 1$
$D(4) = 1$	$D(8) = 1$

Auswahl der Knoten für maximalen Gewinnwert

2.4.1 Kernighan-Lin (KL)-Algorithmus: Beispiel



Schnittkosten: 9
Nicht fixiert:
1,2,3,4,5,6,7,8

Kosten $D(v)$ jedes Knotens:

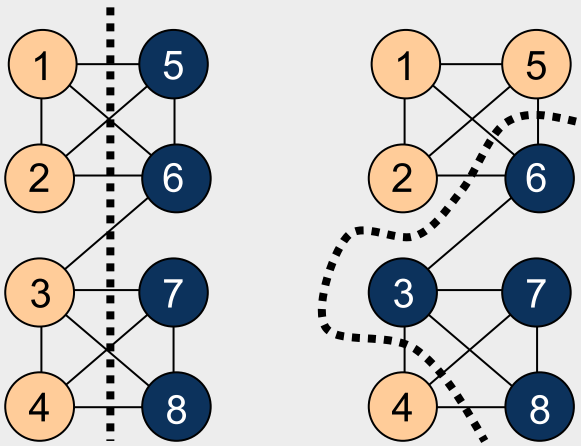
$D(1) = 1$	$\boldsymbol{D(5) = 1}$
$D(2) = 1$	$D(6) = 2$
$\boldsymbol{D(3) = 2}$	$\cancel{D(7) = 1}$
$D(4) = 1$	$D(8) = 1$

$\Delta g_1 = 2+1\text{-}0 = 3$ ← Gewinnwert bei Knotentausch

Austausch (3,5)

$G_1 = \Delta g_1 = 3$ ← Bisheriger Gewinn des Passes

Auswahl der Knoten für maximalen Gewinnwert



Schnittkosten: 9
Nicht fixiert:
1,2,3,4,5,6,7,8

$D(1) = 1$ **$D(5) = 1$**
$D(2) = 1$ $D(6) = 2$
$D(3) = 2$ $D(7) = 1$
$D(4) = 1$ $D(8) = 1$

Auswahl der Knoten für maximalen Gewinnwert

$\Delta g_1 = 2+1-0 = 3$

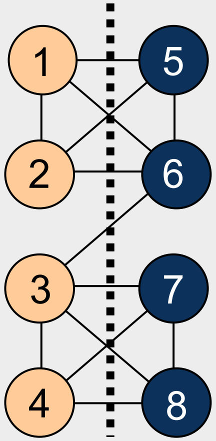
Gewinnwert bei Knotentausch

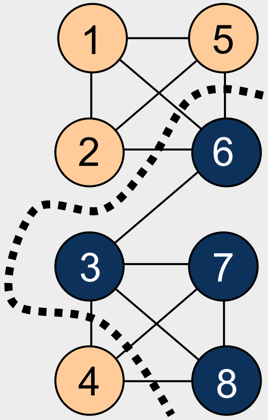
Austausch (3,5)

$G_1 = \Delta g_1 = 3$

Bisheriger Gewinn des Passes

2.4.1 Kernighan-Lin (KL)-Algorithmus: Beispiel





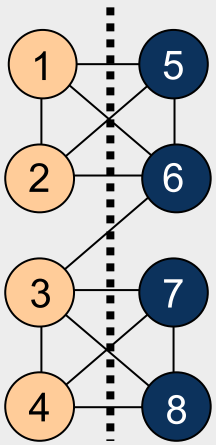
Schnittkosten: 9
Nicht fixiert:
1,2,3,4,5,6,7,8

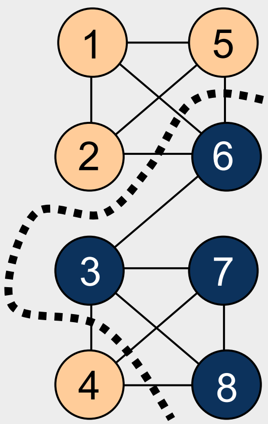
Schnittkosten: 6
Nicht fixiert:
1,2,4,6,7,8

$D(1) = 1$	$\boldsymbol{D(5) = 1}$
$D(2) = 1$	$D(6) = 2$
$\boldsymbol{D(3) = 2}$	$D(7) = 1$
$D(4) = 1$	$D(8) = 1$

$\Delta g_1 = 2+1-0 = 3$
Austausch (3,5)
$G_1 = \Delta g_1 = 3$

2.4.1 Kernighan-Lin (KL)-Algorithmus: Beispiel





Schnittkosten: 9
Nicht fixiert:
1,2,3,4,5,6,7,8

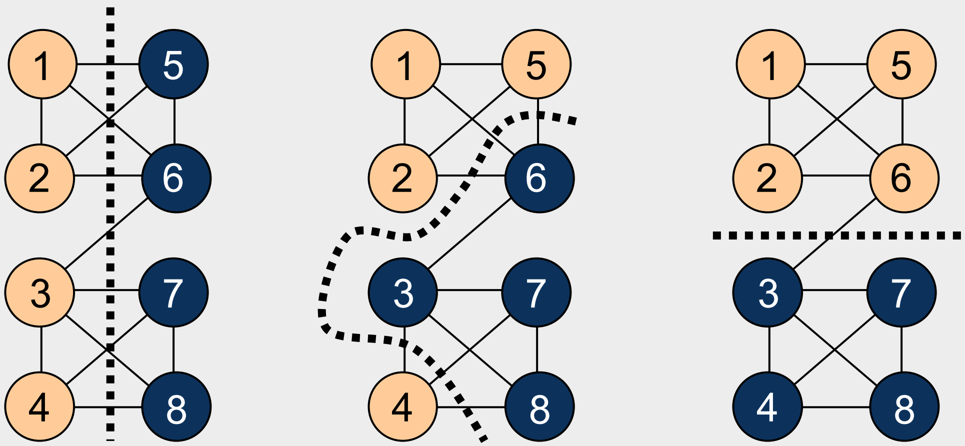
Schnittkosten: 6
Nicht fixiert:
1,2,4,6,7,8

$D(1) = 1$	$\boldsymbol{D(5) = 1}$
$D(2) = 1$	$D(6) = 2$
$\boldsymbol{D(3) = 2}$	$D(7) = 1$
$D(4) = 1$	$D(8) = 1$

$\Delta g_1 = 2+1-0 = 3$
Austausch (3,5)
$G_1 = \Delta g_1 = 3$

$D(1) = -1$	$\boldsymbol{D(6) = 2}$
$D(2) = -1$	$D(7) = -1$
$\boldsymbol{D(4) = 3}$	$D(8) = -1$

2.4.1 Kernighan-Lin (KL)-Algorithmus: Beispiel



Schnittkosten: 9
Nicht fixiert:
1,2,3,4,5,6,7,8

Schnittkosten: 6
Nicht fixiert:
1,2,4,6,7,8

$D(1) = 1$ $\quad$ $\boldsymbol{D(5) = 1}$
$D(2) = 1$ $\quad$ $D(6) = 2$
$\boldsymbol{D(3) = 2}$ $\quad$ $D(7) = 1$
$D(4) = 1$ $\quad$ $D(8) = 1$

$\Delta g_1 = 2+1-0 = 3$
Austausch (3,5)
$G_1 = \Delta g_1 = 3$

$D(1) = -1$ $\quad$ $\boldsymbol{D(6) = 2}$
$D(2) = -1$ $\quad$ $D(7) = -1$
$\boldsymbol{D(4) = 3}$ $\quad$ $D(8) = -1$

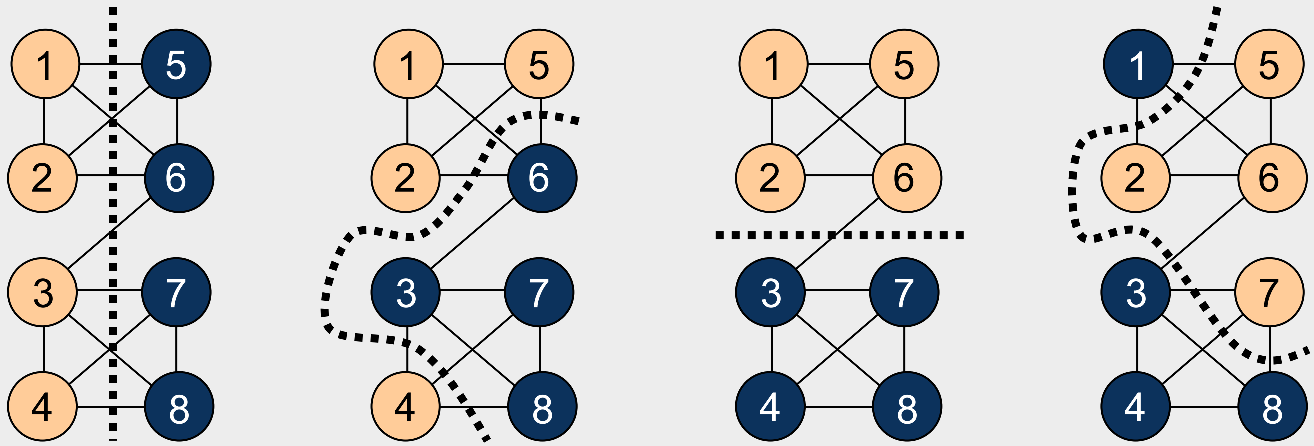
$\Delta g_2 = 3+2-0 = 5$
Austausch (4,6)
$G_2 = G_1 + \Delta g_2 = 8$

Auswahl der Knoten für maximalen Gewinnwert

Gewinnwert bei Knotentausch

Bisheriger Gewinn des Passes

2.4.1 Kernighan-Lin (KL)-Algorithmus: Beispiel



Schnittkosten: 9
Nicht fixiert:
1,2,3,4,5,6,7,8

Schnittkosten: 6
Nicht fixiert:
1,2,4,6,7,8

Schnittkosten: 1
Nicht fixiert:
1,2,7,8

Schnittkosten: 7
Nicht fixiert:
2,8

$D(1) = 1$	$\boldsymbol{D(5) = 1}$
$D(2) = 1$	$D(6) = 2$
$\boldsymbol{D(3) = 2}$	$D(7) = 1$
$D(4) = 1$	$D(8) = 1$

$\Delta g_1 = 2+1-0 = 3$
Austausch (3,5)
$G_1 = \Delta g_1 = 3$

$D(1) = -1$	$\boldsymbol{D(6) = 2}$
$D(2) = -1$	$D(7) = -1$
$\boldsymbol{D(4) = 3}$	$D(8) = -1$

$\Delta g_2 = 3+2-0 = 5$
Austausch (4,6)
$G_2 = G_1 + \Delta g_2 = 8$

$\boldsymbol{D(1) = -3}$	$\boldsymbol{D(7) = -3}$
$D(2) = -3$	$D(8) = -3$

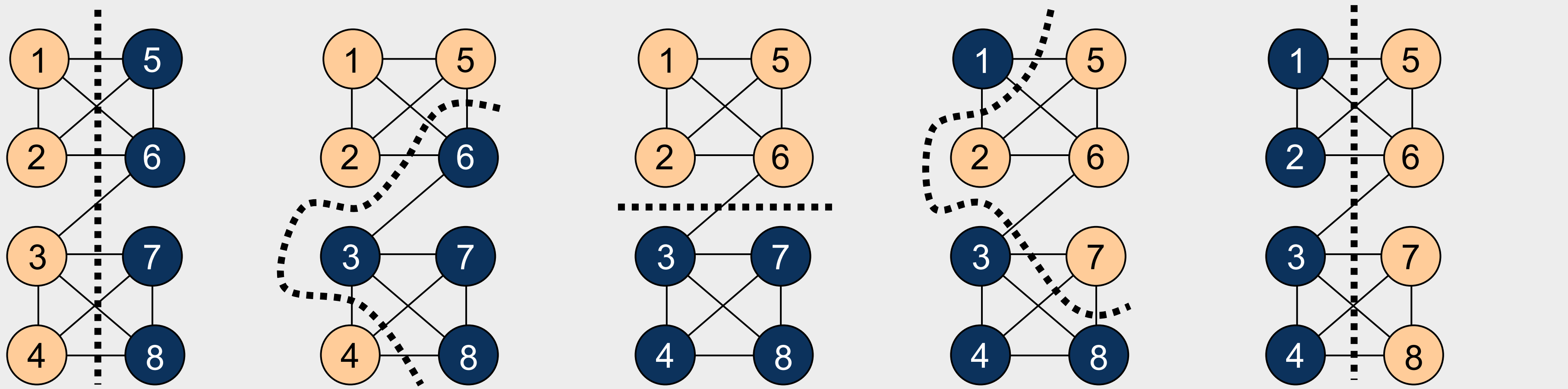
$\Delta g_3 = -3-3-0 = -6$
Austausch (1,7)
$G_3 = G_2 + \Delta g_3 = 2$

Auswahl der Knoten für maximalen Gewinnwert

Gewinnwert bei Knotentausch

Bisheriger Gewinn des Passes

2.4.1 Kernighan-Lin (KL)-Algorithmus: Beispiel



Schnittkosten: 9
Nicht fixiert:
1,2,3,4,5,6,7,8

Schnittkosten: 6
Nicht fixiert:
1,2,4,6,7,8

Schnittkosten: 1
Nicht fixiert:
1,2,7,8

Schnittkosten: 7
Nicht fixiert:
2,8

Schnittkosten: 9
Nicht fixiert:
–

$D(1) = 1$	$\mathbf{D(5) = 1}$
$D(2) = 1$	$D(6) = 2$
$\mathbf{D(3) = 2}$	$D(7) = 1$
$D(4) = 1$	$D(8) = 1$

$\Delta g_1 = 2+1-0 = 3$
Austausch (3,5)
$G_1 = \Delta g_1 = 3$

$D(1) = -1$	$\mathbf{D(6) = 2}$
$D(2) = -1$	$D(7)=-1$
$\mathbf{D(4) = 3}$	$D(8)=-1$

$\Delta g_2 = 3+2-0 = 5$
Austausch (4,6)
$G_2 = G_1+\Delta g_2 = 8$

$\mathbf{D(1) = -3}$	$\mathbf{D(7)=-3}$
$D(2) = -3$	$D(8)=-3$

$\Delta g_3 = -3-3-0 = -6$
Austausch (1,7)
$G_3 = G_2 + \Delta g_3 = 2$

$\mathbf{D(2) = -1}$	$\mathbf{D(8)=-1}$

$\Delta g_4 = -1-1-0 = -2$
Austausch (2,8)
$G_4 = G_3 + \Delta g_4 = 0$

2.4.1 Kernighan-Lin (KL)-Algorithmus: Beispiel

$D(1) = 1$	$\boldsymbol{D(5) = 1}$	$D(1) = -1$	$\boldsymbol{D(6) = 2}$	$\boldsymbol{D(1) = -3}$	$\boldsymbol{D(7)=-3}$	$\boldsymbol{D(2) = -1}$	$\boldsymbol{D(8)=-1}$
$D(2) = 1$	$D(6) = 2$	$D(2) = -1$	$D(7)=-1$	$D(2) = -3$	$D(8)=-3$		
$\boldsymbol{D(3) = 2}$	$D(7) = 1$	$\boldsymbol{D(4) = 3}$	$D(8)=-1$				
$D(4) = 1$	$D(8) = 1$						
$\Delta g_1 = 2+1-0 = 3$		$\Delta g_2 = 3+2-0 = 5$		$\Delta g_3 = -3-3-0 = -6$		$\Delta g_4 = -1-1-0 = -2$	
Austausch (3,5)		**Austausch (4,6)**		**Austausch (1,7)**		**Austausch (2,8)**	
$G_1 = \Delta g_1 =3$		$G_2 = G_1+\Delta g_2 =8$		$G_3= G_2 +\Delta g_3 = 2$		$G_4 = G_3 +\Delta g_4 = 0$	

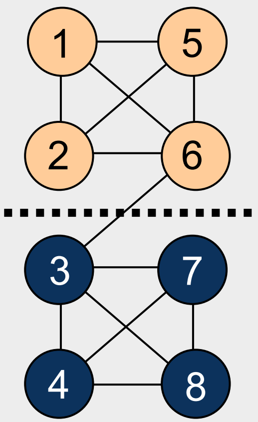
Maximaler positiver Gewinn $G_m = 8$ bei $m = 2$.

2.4.1 Kernighan-Lin (KL)-Algorithmus: Beispiel

$D(1) = 1$ **$D(5) = 1$** $D(2) = 1$ $D(6) = 2$ **$D(3) = 2$** $D(7) = 1$ $D(4) = 1$ $D(8) = 1$ $\Delta g_1 = 2+1-0 = 3$ **Austausch (3,5)** $G_1 = \Delta g_1 = 3$	$D(1) = -1$ **$D(6) = 2$** $D(2) = -1$ $D(7)=-1$ **$D(4) = 3$** $D(8)=-1$ $\Delta g_2 = 3+2-0 = 5$ **Austausch (4,6)** $G_2 = G_1+\Delta g_2 = 8$	**$D(1) = -3$** **$D(7)=-3$** $D(2) = -3$ $D(8)=-3$ $\Delta g_3 = -3-3-0 = -6$ **Austausch (1,7)** $G_3 = G_2 + \Delta g_3 = 2$	**$D(2) = -1$** **$D(8)=-1$** $\Delta g_4 = -1-1-0 = -2$ **Austausch (2,8)** $G_4 = G_3 + \Delta g_4 = 0$

Maximaler positiver Gewinn $G_m = 8$ bei $m = 2$.

Da $G_m > 0$, werden die m Vertauschungen (3,5) und (4,6) durchgeführt.

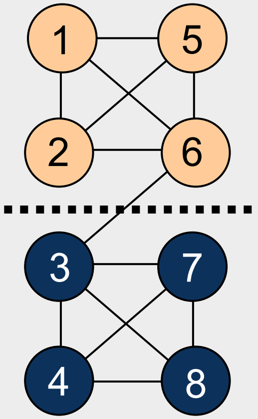


2.4.1 Kernighan-Lin (KL)-Algorithmus: Beispiel

$D(1) = 1$ $D(2) = 1$ $\boldsymbol{D(3) = 2}$ $D(4) = 1$ $\boldsymbol{D(5) = 1}$ $D(6) = 2$ $D(7) = 1$ $D(8) = 1$ $\Delta g_1 = 2+1-0 = 3$ **Austausch (3,5)** $G_1 = \Delta g_1 = 3$	$D(1) = -1$ $D(2) = -1$ $\boldsymbol{D(4) = 3}$ $\boldsymbol{D(6) = 2}$ $D(7) = -1$ $D(8) = -1$ $\Delta g_2 = 3+2-0 = 5$ **Austausch (4,6)** $G_2 = G_1 + \Delta g_2 = 8$	$\boldsymbol{D(1) = -3}$ $D(2) = -3$ $\boldsymbol{D(7) = -3}$ $D(8) = -3$ $\Delta g_3 = -3-3-0 = -6$ **Austausch (1,7)** $G_3 = G_2 + \Delta g_3 = 2$	$\boldsymbol{D(2) = -1}$ $\boldsymbol{D(8) = -1}$ $\Delta g_4 = -1-1-0 = -2$ **Austausch (2,8)** $G_4 = G_3 + \Delta g_4 = 0$

Maximaler positiver Gewinn $G_m = 8$ bei $m = 2$.

Da $G_m > 0$, werden die m Vertauschungen (3,5) und (4,6) durchgeführt.



Da $G_m > 0$, weiterer Pass notwendig, bis $G_m \leq 0$ während aller Vertauschungen.

2.4.1 Kernighan-Lin (KL)-Algorithmus: Anmerkungen

- Die notwendige Anzahl der zu durchlaufenden äußeren Schleifen (Pässe) ist nicht abhängig von der Problemgröße; oft reichen bereits vier Pässe aus, um die erzielbare beste Lösung zu finden.

- Ein Teil der Knoten-Vertauschungen kann mit einem negativen Gewinnwert Δg behaftet sein, der durch folgende Vertauschungen innerhalb des Passes kompensiert wird.
  Auch bei negativen Gewinnwerten sollten die noch verbleibenden Vertauschungen innerhalb eines Passes durchgeführt werden.

- Der Algorithmus findet nicht immer die optimale Lösung, ist aber eine schnelle Heuristik.

2.4.3 Fiduccia-Mattheyses (FM)-Algorithmus

- Näher zur schaltungstechnischen Realität: Zelle (anstelle Knoten), Block (anstelle Teilgraph), Netz (anstelle Kante)
- Es wird immer nur eine Zelle verschoben, d.h. kein Knoten-/Zellenpaar getauscht. Damit anwendbar für Partitionierungen ungleicher Größe.
- Das Konzept der Schnittkosten wird erweitert, um auch Hypergraphs (Mehrpunkt-Netze) einzuschließen.
  - KL-Algorithmus: Minimierung der geschnittenen <u>Kanten</u>
  - FM-Algorithmus: Minimierung der geschnittenen <u>Netze</u>
- Die Größen der einzelnen Zellen werden berücksichtigt.

2.4.3 Fiduccia-Mattheyses (FM)-Algorithmus

Gegeben ist ein Graph mit Zellen (Knoten) und gewichteten Netzen (Kanten) zwischen beliebigen Zellen.

Gesucht ist die Aufteilung des Graphen in zwei Blöcke (Teilgraphen) *A* und *B* derart, dass die Anzahl der Netze zwischen beiden Blöcken minimiert und gleichzeitig ein vorgegebenes Größenverhältnis der Blöcke eingehalten wird.

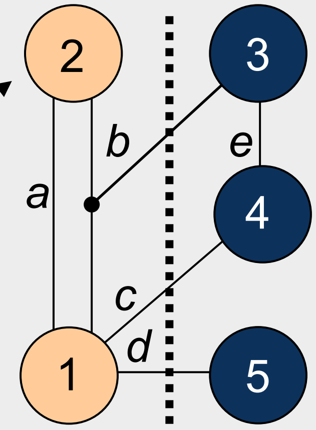
Gewinnwert einer Zelle $\Delta g(c)$

$$\Delta g(c) = FS(c) - TE(c) ,$$

wobei

$FS(c)$ die Anzahl der mit der Zelle c verbundenen Netze darstellt, die nicht mit anderen Zellen im gegenwärtigen Block der Zelle c verbunden sind (durch die Schnittlinie kommend nur die Zelle c verbinden) und

$TE(c)$ die Anzahl der mit der Zelle c verbundenen Netze ist, die nicht die Schnittlinie kreuzen



Zelle 2: $FS(2) = 0$ $TE(2) = 1$ $\Delta g(2) = -1$

Je höher der Gewinnwert $\Delta g(c)$ einer Zelle c ist, umso sinnvoller ist es hinsichtlich einer Minimierung der Schnittmenge, diese Zelle in den anderen Block zu verschieben.

Gewinnwert einer Zelle $\Delta g(c)$

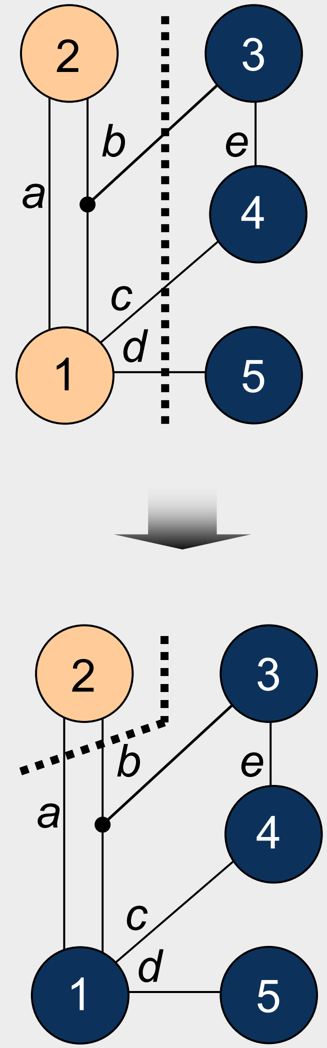
$$\Delta g(c) = FS(c) - TE(c) ,$$

wobei

FS(*c*) die Anzahl der mit der Zelle *c* verbundenen Netze darstellt, die nicht mit anderen Zellen im gegenwärtigen Block der Zelle *c* verbunden sind (durch die Schnittlinie kommend nur die Zelle c verbinden) und

TE(*c*) die Anzahl der mit der Zelle *c* verbundenen Netze ist, die nicht die Schnittlinie kreuzen

Zelle 1:	$FS(1) = 2$	$TE(1) = 1$	$\Delta g(1) = 1$
Zelle 2:	$FS(2) = 0$	$TE(2) = 1$	$\Delta g(2) = -1$
Zelle 3:	$FS(3) = 1$	$TE(3) = 1$	$\Delta g(3) = 0$
Zelle 4:	$FS(4) = 1$	$TE(4) = 1$	$\Delta g(4) = 0$
Zelle 5:	$FS(5) = 1$	$TE(5) = 0$	$\Delta g(5) = 1$



2.4.3 Fiduccia-Mattheyses (FM)-Algorithmus: Begriffe

Maximaler positiver Gewinn G_m eines Passes

Der maximale positive Gewinn G_m gibt die Folge von m Verschiebungen innerhalb eines Passes an, die zu einer maximalen Schnittkostenminimierung führt.

Konkret ergibt sich G_m aus den aufaddierten Gewinnwerten Δg von m hintereinander erfolgten Verschiebungen innerhalb eines Passes.

2.4.3 Fiduccia-Mattheyses (FM)-Algorithmus: Begriffe

Verhältnisfaktor und Gleichgewichtskriterium

Da beim FM-Algorithmus die Zellen einzeln verschoben werden, ist eine Berücksichtigung der Partitionierungsgrößen bei jeder Zellenverschiebung notwendig.

Mittels eines sog. Verhältnisfaktors *r* wird ein angestrebtes Größenverhältnis der Flächen *A* und *B* ausgedrückt:

$$r = \frac{|A|}{|A| + |B|}$$

Mit einem vom Verhältnisfaktor *r* ableitbaren Gleichgewichtskriterium lässt sich jede Zellenverschiebung auf Einhaltung des durch *r* vorgegebenen Größenverhältnisses beider Teilflächen verifizieren.

Basiszelle

Die zur Verschiebung ausgewählte Zelle wird als Basiszelle bezeichnet.

Diese besitzt den maximalen Zellengewinnwert $\Delta g(c)$ unter allen verschiebbaren Zellen und verletzt durch ihre Verschiebung nicht das Gleichgewichtskriterium.

Basiszelle

Zelle 1:	$FS(1) = 2$	$TE(1) = 1$	$\Delta g(1) = 1$
Zelle 2:	$FS(2) = 0$	$TE(2) = 1$	$\Delta g(2) = -1$
Zelle 3:	$FS(3) = 1$	$TE(3) = 1$	$\Delta g(3) = 0$
Zelle 4:	$FS(4) = 1$	$TE(4) = 1$	$\Delta g(4) = 0$

2.4.3 Fiduccia-Mattheyses (FM)-Algorithmus

Schritt 0: Errechnen des Gleichgewichtskriteriums

Schritt 1: Ermitteln des Gewinns Δg_1 jeder Zelle

Schritt 2: $i = 1$

- Auswahl der Basiszelle c_1 mit maximalem Gewinnwert Δg_1 ,Verschiebung dieser Zelle

Schritt 3:

- Fixierung der Basiszelle c_i
- Gewinnwert-Aktualisierung der Zellen, die durch kritische Netze mit der Basiszelle c_i verbunden sind

Schritt 4:

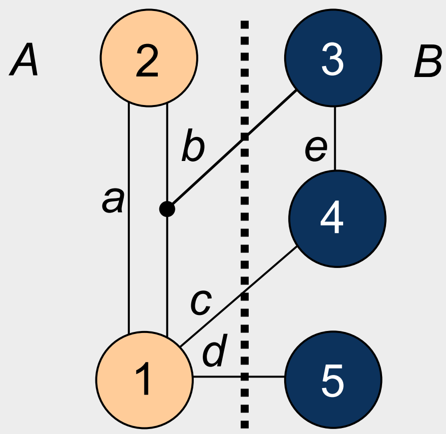
- Wenn alle Zellen fixiert sind, weiter mit Schritt 5, andernfalls
- $i = i + 1$
- Auswahl der nächsten Basiszelle c_i mit maximalem Gewinnwert Δg_i und Verschiebung dieser Zelle
- Weiter mit Schritt 3

Schritt 5:

- Ermitteln der besten Verschiebungsfolge $c_1, c_2, .., c_m$ $(1 \leq m \leq i)$, so dass $G_m = \sum_{i=1}^{m} \Delta g_i$ maximiert wird
- Wenn $G_m > 0$, weiter mit Schritt 6, andernfalls ENDE

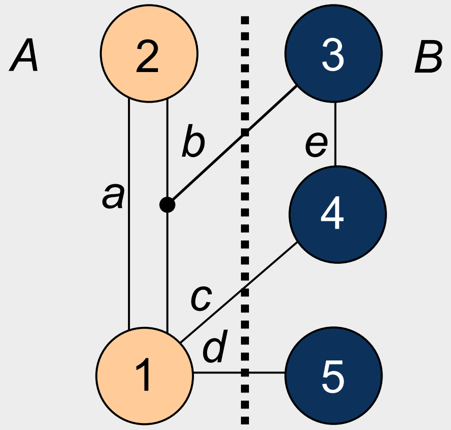
Schritt 6:

- Durchführen aller m Vertauschungen, beseitigen aller Zellenfixierungen
- Neuer Pass, dazu weiter mit Schritt 1.



Gegeben:
Verhältnisfaktor r = 0,375
s(Zelle_1) = 2
s(Zelle_2) = 4
s(Zelle_3) = 1
s(Zelle_4) = 4
s(Zelle_5) = 5.

2.4.3 Fiduccia-Mattheyses (FM)-Algorithmus: Beispiel



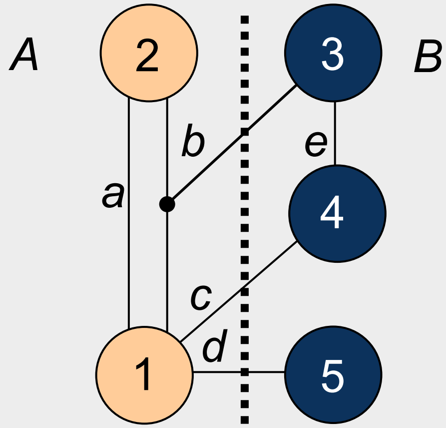
Gegeben:
Verhältnisfaktor r = 0,375
s(Zelle_1) = 2
s(Zelle_2) = 4
s(Zelle_3) = 1
s(Zelle_4) = 4
s(Zelle_5) = 5.

Schritt 0: Errechnen des Gleichgewichtskriteriums

$$r \cdot |V| - s_{\max} \le |A| \le r \cdot |V| + s_{\max}$$

0,375 * 16 – 5 = $1 \le A \le 11$ = 0,375 * 16 +5.

2.4.3 Fiduccia-Mattheyses (FM)-Algorithmus: Beispiel

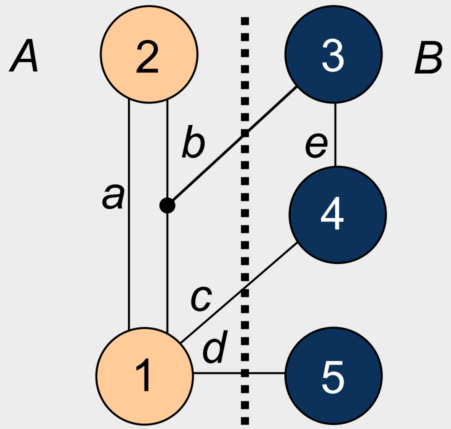


Schritt 1: Ermitteln der Gewinnwerte jeder Zelle

Zelle 1:

- Zwei Netze (*c*, *d*) sind nur mit Zelle 1 in deren Block verbunden und sind geschnitten, d.h. *FS*(Zelle_1) = 2
- Ein Netz (*a*) ist mit Zelle 1 verbunden und wird nicht geschnitten, d.h. *TE*(Zelle_1) = 1
- Δg(Zelle_1) = 2 - 1 =1, d.h. Anzahl der geschnittenen Netze reduziert sich um eins (von drei auf zwei), sollte Zelle 1 von *A* nach *B* verschoben werden

2.4.3 Fiduccia-Mattheyses (FM)-Algorithmus: Beispiel



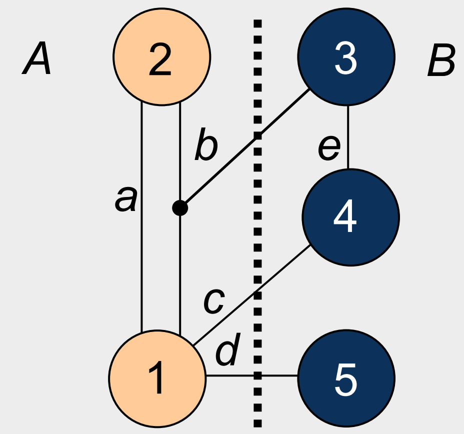
Schritt 1: Ermitteln der Gewinnwerte jeder Zelle

Zelle 1:

- Zwei Netze (*c*, *d*) sind nur mit Zelle 1 in deren Block verbunden und sind geschnitten, d.h. *FS*(Zelle_1) = 2
- Ein Netz (*a*) ist mit Zelle 1 verbunden und wird nicht geschnitten, d.h. *TE*(Zelle_1) = 1
- Δg(Zelle_1) = 2 - 1 =1, d.h. Anzahl der geschnittenen Netze reduziert sich um eins (von drei auf zwei), sollte Zelle 1 von *A* nach *B* verschoben werden
- Analog werden die Gewinnwerte der verbleibenden Zellen berechnet:

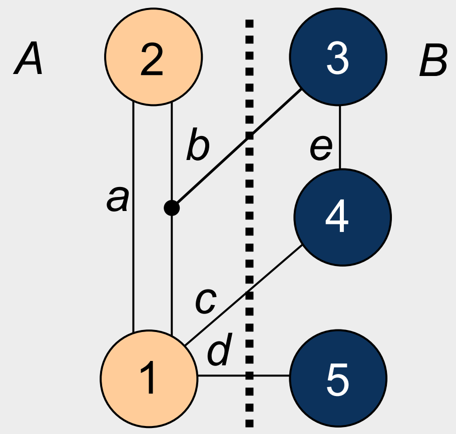
Zelle 2:	*FS*(Zelle_2) = 0	*TE*(Zelle_2) = 1	Δg(Zelle_2) = -1
Zelle 3:	*FS*(Zelle_3) = 1	*TE*(Zelle_3) = 1	Δg(Zelle_3) = 0
Zelle 4:	*FS*(Zelle_4) = 1	*TE*(Zelle_4) = 1	Δg(Zelle_4) = 0
Zelle 5:	*FS*(Zelle_5) = 1	*TE*(Zelle_5) = 0	Δg(Zelle_5) = 1 .

2.4.3 Fiduccia-Mattheyses (FM)-Algorithmus: Beispiel



Gewinnwerte der Zellen:

Zelle 1:	FS(Zelle_1) = 2	TE(Zelle_1) = 1	Δg(Zelle_1) = 1
Zelle 2:	FS(Zelle_2) = 0	TE(Zelle_2) = 1	Δg(Zelle_2) = -1
Zelle 3:	FS(Zelle_3) = 1	TE(Zelle_3) = 1	Δg(Zelle_3) = 0
Zelle 4:	FS(Zelle_4) = 1	TE(Zelle_4) = 1	Δg(Zelle_4) = 0
Zelle 5:	FS(Zelle_5) = 1	TE(Zelle_5) = 0	Δg(Zelle_5) = 1 .



Gewinnwerte der Zellen:

Zelle 1:	*FS*(Zelle_1) = 2	*TE*(Zelle_1) = 1	Δg(Zelle_1) = 1
Zelle 2:	*FS*(Zelle_2) = 0	*TE*(Zelle_2) = 1	Δg(Zelle_2) = -1
Zelle 3:	*FS*(Zelle_3) = 1	*TE*(Zelle_3) = 1	Δg(Zelle_3) = 0
Zelle 4:	*FS*(Zelle_4) = 1	*TE*(Zelle_4) = 1	Δg(Zelle_4) = 0
Zelle 5:	*FS*(Zelle_5) = 1	*TE*(Zelle_5) = 0	Δg(Zelle_5) = 1 .

Schritt 2: Auswahl der Basiszelle und Verschiebung

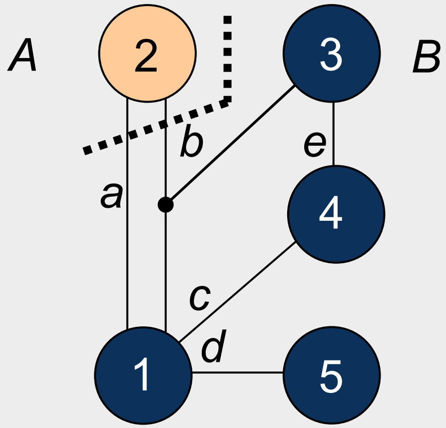
Mögliche Basiszellen: Zelle 1 und Zelle 5

- Gleichgewichtskriterium nach Verschiebung von Zelle 1: $|A| = s(2) = 4$
- Gleichgewichtskriterium nach Verschiebung von Zelle 5: $|A| = s(1) + s(2) + s(5) = 11$.

Beide Verschiebungen verletzen somit nicht das Gleichgewichtskriterium $1 \leq |A| \leq 11$;
Zelle 1 wird aufgrund der besseren Erfüllung als Basiszelle ausgewählt und verschoben.

Schritt 3: Zellenfixierung und Gewinnwert-Aktualisierung

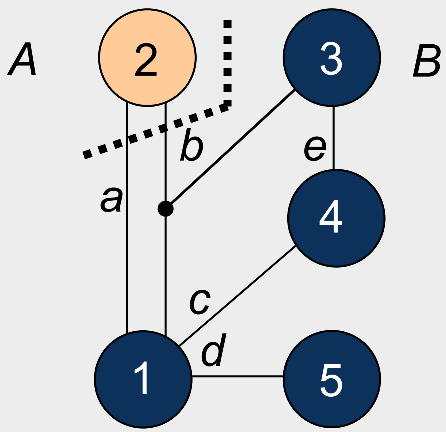
1. Zelle 1 fixieren, neue Anordnung:



2.4.3 Fiduccia-Mattheyses (FM)-Algorithmus: Beispiel

Schritt 3: Zellenfixierung und Gewinnwert-Aktualisierung

1. Zelle 1 fixieren, neue Anordnung:

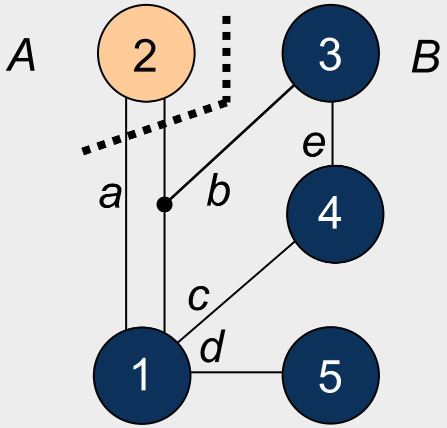


2. Gewinnwert aktualisieren von allen verschiebbaren Zellen, die mittels kritischer Netze mit Zelle 1 verbunden sind:

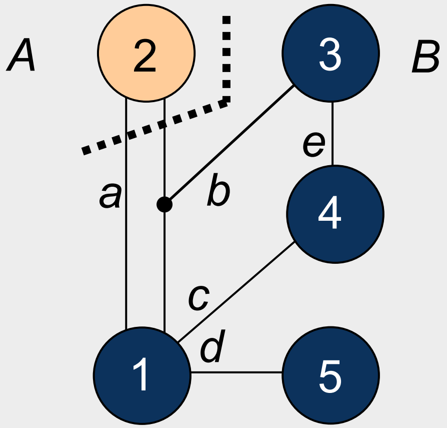
Zelle 2:	FS(Zelle_2) = 2	TE(Zelle_2) = 0	Δg(Zelle_2) = 2
Zelle 3:	FS(Zelle_3) = 0	TE(Zelle_3) = 1	Δg(Zelle_3) = -1
Zelle 4:	FS(Zelle_4) = 0	TE(Zelle_4) = 2	Δg(Zelle_4) = -2
Zelle 5:	FS(Zelle_5) = 0	TE(Zelle_5) = 1	Δg(Zelle_5) = -1

Nach Iteration i = 1: Mengenverteilung $A_1 = \{2\}$, $B_1 = \{1,3,4,5\}$, davon fixiert $\{1\}$.

2.4.3 Fiduccia-Mattheyses (FM)-Algorithmus: Beispiel



Zelle 2:	FS(Zelle_2) = 2	TE(Zelle_2) = 0	Δg(Zelle_2) = 2
Zelle 3:	FS(Zelle_3) = 0	TE(Zelle_3) = 1	Δg(Zelle_3) = -1
Zelle 4:	FS(Zelle_4) = 0	TE(Zelle_4) = 2	Δg(Zelle_4) = -2
Zelle 5:	FS(Zelle_5) = 0	TE(Zelle_5) = 1	Δg(Zelle_5) = -1



Zelle 2:	*FS*(Zelle_2) = 2	*TE*(Zelle_2) = 0	Δg(Zelle_2) = 2
Zelle 3:	*FS*(Zelle_3) = 0	*TE*(Zelle_3) = 1	Δg(Zelle_3) = -1
Zelle 4:	*FS*(Zelle_4) = 0	*TE*(Zelle_4) = 2	Δg(Zelle_4) = -2
Zelle 5:	*FS*(Zelle_5) = 0	*TE*(Zelle_5) = 1	Δg(Zelle_5) = -1

Schritt 4: Auswahl und Verschiebung einer Basiszelle; Zellenfixierung; Gewinnwert-Aktualisierung

- Iteration $i = 2$
  Gemäß obigem Schritt 3: Zelle 2 mit maximalem Gewinnwert $\Delta g_2 = 2$, $|A| = 0$, d.h. Gleichgewichtskriterium nicht erfüllt
  Zelle 3 mit nächstem maximalen Gewinnwert $\Delta g_2 = -1$, $|A| = 5$, d.h. Gleichgewichtskriterium erfüllt
  Zelle 5 mit nächstem maximalen Gewinnwert $\Delta g_2 = -1$, $|A| = 9$, d.h. Gleichgewichtskriterium erfüllt
  Zelle 3 wird verschoben, Mengenverteilung $A_2 = \{2,3\}$, $B_2 = \{1,4,5\}$, davon fixiert $\{1,3\}$.
- Iteration $i = 3$

2.4.3 Fiduccia-Mattheyses (FM)-Algorithmus: Beispiel

Schritt 4: Auswahl und Verschiebung einer Basiszelle; Zellenfixierung; Gewinnwert-Aktualisierung

- Iteration $i = 2$
  Gemäß obigem Schritt 3: Zelle 2 mit maximalem Gewinnwert $\Delta g_2 = 2$, $|A| = 0$, d.h. Gleichgewichtskriterium nicht erfüllt
  Zelle 3 mit nächstem maximalen Gewinnwert $\Delta g_2 = -1$, $|A| = 5$, d.h. Gleichgewichtskriterium erfüllt
  Zelle 5 mit nächstem maximalen Gewinnwert $\Delta g_2 = -1$, $|A| = 9$, d.h. Gleichgewichtskriterium erfüllt
  Zelle 3 wird verschoben, Mengenverteilung $A_2 = \{2,3\}$, $B_2 = \{1,4,5\}$, davon fixiert $\{1,3\}$.
- Iteration $i = 3$

Schritt 4: Auswahl und Verschiebung einer Basiszelle; Zellenfixierung; Gewinnwert-Aktualisierung

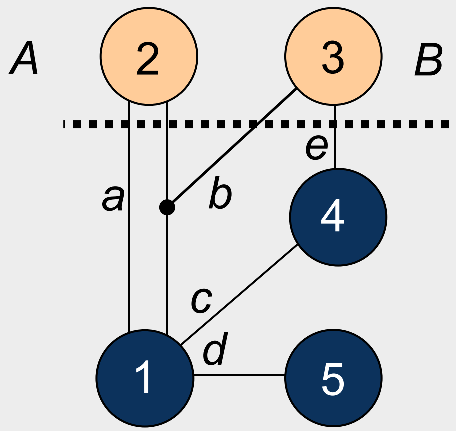
- Iteration $i = 2$

  Gemäß obigem Schritt 3: Zelle 2 mit maximalem Gewinnwert $\Delta g_2 = 2$, $|A| = 0$, d.h. Gleichgewichtskriterium nicht erfüllt

  Zelle 3 mit nächstem maximalen Gewinnwert $\Delta g_2 = -1$, $|A| = 5$, d.h. Gleichgewichtskriterium erfüllt

  Zelle 5 mit nächstem maximalen Gewinnwert $\Delta g_2 = -1$, $|A| = 9$, d.h. Gleichgewichtskriterium erfüllt

  Zelle 3 wird verschoben, Mengenverteilung $A_2 = \{2,3\}$, $B_2 = \{1,4,5\}$, davon fixiert $\{1,3\}$.
- Iteration $i = 3$



Schritt 4: Auswahl und Verschiebung einer Basiszelle; Zellenfixierung; Gewinnwert-Aktualisierung

- Iteration $i = 2$

  Gemäß obigem Schritt 3: Zelle 2 mit maximalem Gewinnwert $\Delta g_2 = 2$, $|A| = 0$, d.h. Gleichgewichtskriterium nicht erfüllt

  Zelle 3 mit nächstem maximalen Gewinnwert $\Delta g_2 = -1$, $|A| = 5$, d.h. Gleichgewichtskriterium erfüllt

  Zelle 5 mit nächstem maximalen Gewinnwert $\Delta g_2 = -1$, $|A| = 9$, d.h. Gleichgewichtskriterium erfüllt

  Zelle 3 wird verschoben, Mengenverteilung $A_2 = \{2,3\}$, $B_2 = \{1,4,5\}$, davon fixiert $\{1,3\}$.

- Iteration $i = 3$

  $\Delta g_3(\text{Zelle\_2}) = 1$ $\Delta g_3(\text{Zelle\_4}) = 0$ $\Delta g_3(\text{Zelle\_5}) = -1$

  Zelle 2 mit maximalem Gewinnwert $\Delta g_3 = 1$, $|A| = 1$, d.h. Gleichgewichtskriterium erfüllt

  Zelle 2 wird verschoben, Mengenverteilung $A_3 = \{3\}$, $B_3 = \{1,2,4,5\}$, davon fixiert $\{1,2,3\}$.

2.4.3 Fiduccia-Mattheyses (FM)-Algorithmus: Beispiel

Schritt 4: Auswahl und Verschiebung einer Basiszelle; Zellenfixierung; Gewinnwert-Aktualisierung

- Iteration $i = 2$
  Gemäß obigem Schritt 3: Zelle 2 mit maximalem Gewinnwert $\Delta g_2 = 2$, $|A| = 0$, d.h. Gleichgewichtskriterium nicht erfüllt
  Zelle 3 mit nächstem maximalen Gewinnwert $\Delta g_2 = -1$, $|A| = 5$, d.h. Gleichgewichtskriterium erfüllt
  Zelle 5 mit nächstem maximalen Gewinnwert $\Delta g_2 = -1$, $|A| = 9$, d.h. Gleichgewichtskriterium erfüllt
  Zelle 3 wird verschoben, Mengenverteilung $A_2 = \{2,3\}$, $B_2 = \{1,4,5\}$, davon fixiert $\{1,3\}$.
- Iteration $i = 3$
  $\Delta g_3(\text{Zelle\_2}) = 1$ $\Delta g_3(\text{Zelle\_4}) = 0$ $\Delta g_3(\text{Zelle\_5}) = -1$
  Zelle 2 mit maximalem Gewinnwert $\Delta g_3 = 1$, $|A| = 1$, d.h. Gleichgewichtskriterium erfüllt
  Zelle 2 wird verschoben, Mengenverteilung $A_3 = \{3\}$, $B_3 = \{1,2,4,5\}$, davon fixiert $\{1,2,3\}$.
- Iteration $i = 4$
  $\Delta g_4(\text{Zelle\_4}) = 0$ $\Delta g_4(\text{Zelle\_5}) = -1$
  Zelle 4 mit maximalem Gewinnwert $\Delta g_4 = 0$, $|A| = 5$, d.h. Gleichgewichtskriterium erfüllt
  Mengenverteilung $A_4 = \{3,4\}$, $B_4 = \{1,2,5\}$, davon fixiert $\{1,2,3,4\}$.
- Iteration $i = 5$
  $\Delta g_5(\text{Zelle\_5}) = -1$
  Zelle 5 mit maximalem Gewinnwert $\Delta g_5 = -1$, $|A| = 10$, d.h. Gleichgewichtskriterium erfüllt
  Mengenverteilung $A_5 = \{3,4,5\}$, $B_5 = \{1,2\}$, alle Zellen fixiert
  Pass 1 beendet; weiter mit Schritt 5.

Schritt 5: Ermittlung der besten Verschiebungsfolge 1 … *m*

$G_1 = \Delta g_1 = 1$
$G_2 = \Delta g_1 + \Delta g_2 = 0$
$G_3 = \Delta g_1 + \Delta g_2 + \Delta g_3 = 1$
$G_4 = \Delta g_1 + \Delta g_2 + \Delta g_3 + \Delta g_4 = 1$
$G_5 = \Delta g_1 + \Delta g_2 + \Delta g_3 + \Delta g_4 + \Delta g_5 = 0.$

Schritt 5: Ermittlung der besten Verschiebungsfolge 1 … *m*

$G_1 = \Delta g_1 = 1$
$G_2 = \Delta g_1 + \Delta g_2 = 0$
$G_3 = \Delta g_1 + \Delta g_2 + \Delta g_3 = 1$
$G_4 = \Delta g_1 + \Delta g_2 + \Delta g_3 + \Delta g_4 = 1$
$G_5 = \Delta g_1 + \Delta g_2 + \Delta g_3 + \Delta g_4 + \Delta g_5 = 0.$

- Maximaler positiver Gewinn

  $$G_m = \sum_{i=1}^{m} g_i = 1$$

  in Iterationen 1, 3 und 4.
- Aufgrund des ausgewogeneren Gleichgewichtskriteriums ($|A| = 5$) wird Iteration 4 gewählt, d.h. $m = 4$.

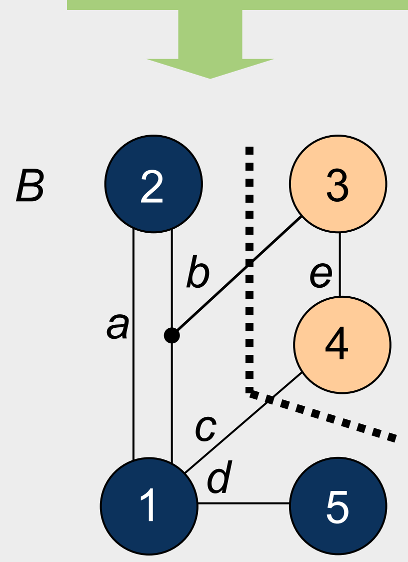
2.4.3 Fiduccia-Mattheyses (FM)-Algorithmus: Beispiel

Schritt 6: Verschiebung der ersten *m* Zellen tatsächlich durchführen

– Da $m = 4$, werden nur die ersten vier ausgewählten Zellen (Zellen 1, 3, 2, 4) verschoben.

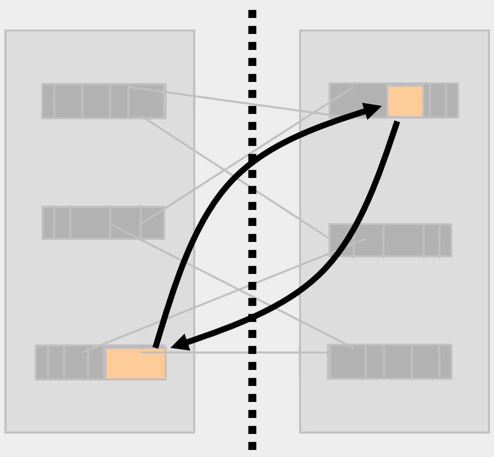
Schritt 6: Verschiebung der ersten *m* Zellen tatsächlich durchführen

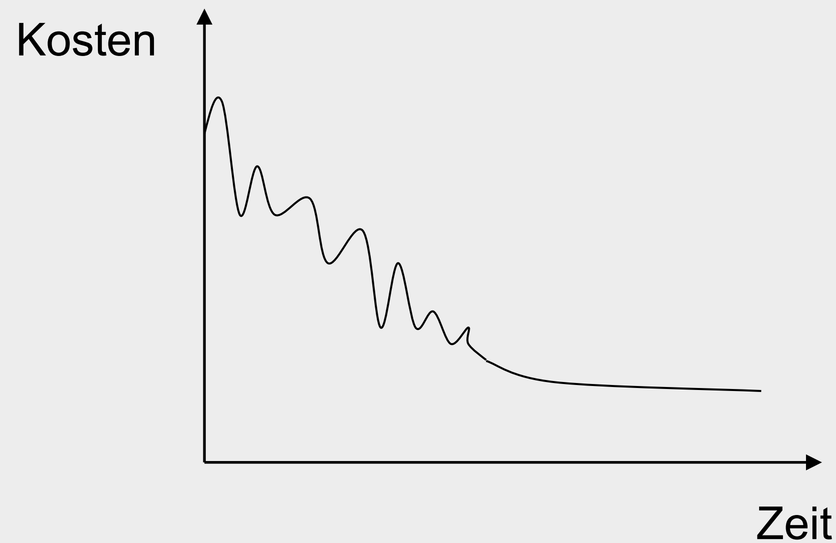
- Da m = 4, werden nur die ersten vier ausgewählten Zellen (Zellen 1, 3, 2, 4) verschoben.
- Ergebnis von Pass 1:
  Mengenverteilung $A = \{3,4\}$, $B = \{1,2,5\}$, Schnittkosten von 3 auf 2 reduziert.



2.4.4 Simulated-Annealing (SA)-Algorithmus

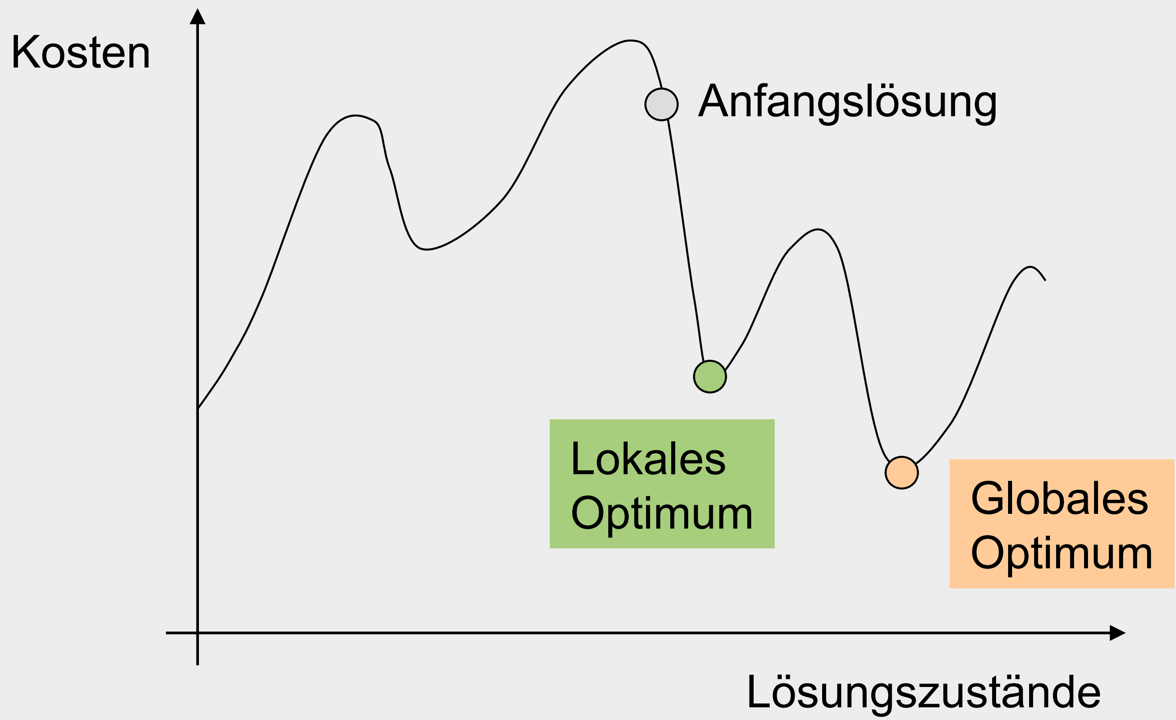
- Analogie zum Abkühlungsprozess von metallischen Schmelzen (energieminimales Atomgitter)
- Modifikation einer Anfangspartitionierung durch Tausch von zufällig ausgewählten Zellen





- Wenn sich Kosten verbessern, wird Tausch ausgeführt
- Bei keiner Kostenverbesserung wird Tausch mit temperaturabhängiger (d.h. abnehmender) Wahrscheinlichkeit ausgeführt

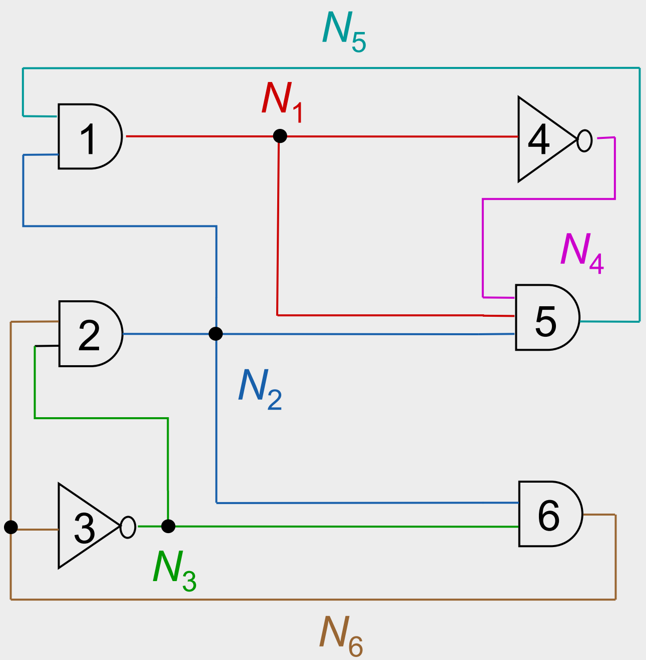
2.4.4 Simulated-Annealing (SA)-Algorithmus



```
begin
    T = T0, i = 0                                   /* Initialisierung */
    cur_part = init_part                            /* Anfangspartitionierung */
    cur_cost = COST(cur_part)                       /* Anfangsqualität */
    repeat
        repeat
            i = i + 1
            a_i = SELECT(A), b_i = SELECT(B)
            trial_part = EXCHANGE(a_i, b_i, cur_part)   /* Tauschversuch */
            trial_cost = COST(trial_part)
            Δcost = trial_cost – cur_cost
            if( Δcost < 0 ) then                    /* wenn Verbesserung */
                cur_cost = trial_cost               /* Tauschdurchführung */
                cur_part = MOVE(a_i, b_i)
            else                                    /* ansonsten */
                r = RANDOM(0,1)                     /* Zufallszahl */
                if( r < e^(–Δcost/T) ) then         /* bedingter */
                    cur_cost = trial_cost           /* Tausch */
                    cur_part = MOVE(a_i, b_i)
        until( Abbruchkriterium, z.B. Gleichgewicht bei T, erreicht )
        T = α * T          /* 0 < α < 1 */          /* Temperatur-Reduktion */
    until( T < Tmin )
end
```

Gegeben: Schaltung mit sechs Zellen und sechs Netzen, wobei alle Zellen identische Größe besitzen, sowie Netzliste mit Netzgewichten

Gesucht: Aufteilung in zwei Blöcke *A* und *B* mit jeweils drei Zellen und minimalen Kosten

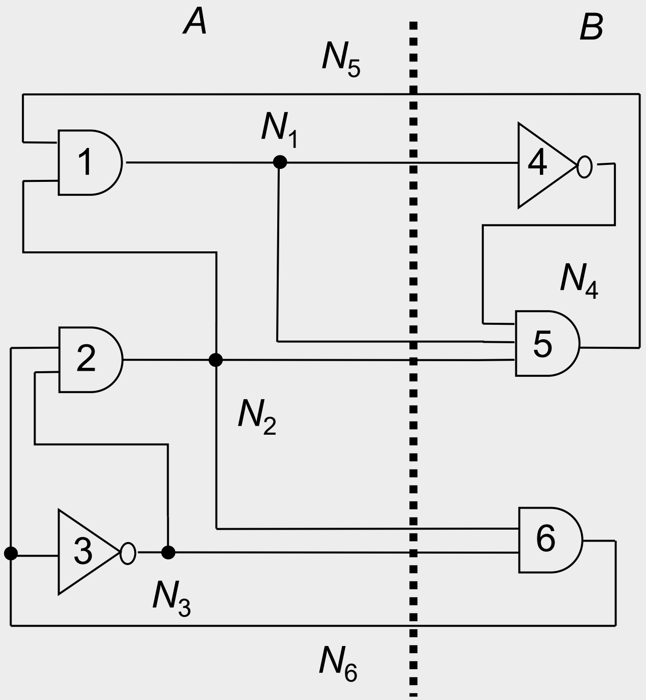


Netze	Gewichte
$N_1 = (1, 4, 5)$	$w_1 = 4$
$N_2 = (1, 2, 5, 6)$	$w_2 = 2$
$N_3 = (2, 3, 6)$	$w_3 = 2$
$N_4 = (4, 5)$	$w_4 = 3$
$N_5 = (1, 5)$	$w_5 = 1$
$N_6 = (2, 3, 6)$	$w_6 = 4$

2.4.4 Simulated-Annealing (SA)-Algorithmus: Beispiel

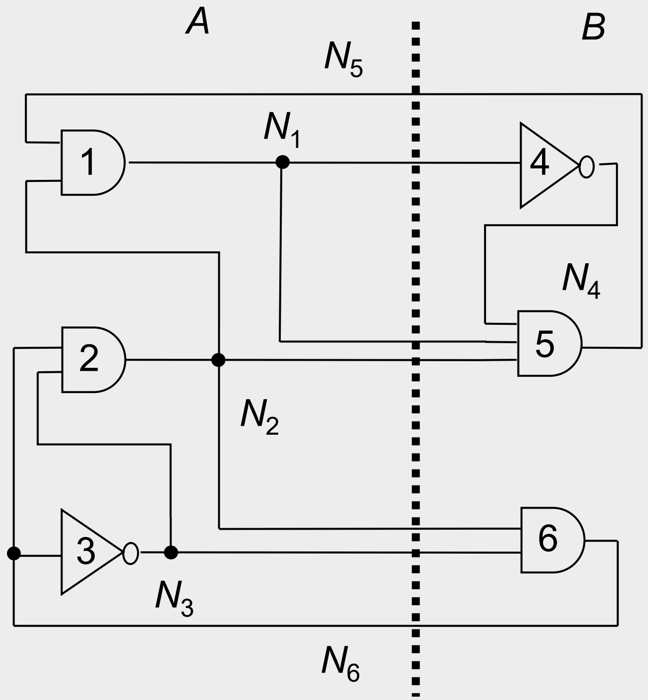
Parameter des SA-Algorithmus:

- Anfangstemperatur: $T_0 = 10$, Abbruch bei $T < 3{,}5$
- Anfangspartitionierung: init_part $A\{1,2,3\}$, $B\{4,5,6\}$
- Anzahl der Austauschversuche pro Temperaturschritt T: 4
- Temperaturabnahme: $\alpha = 0{,}7$
- Austauschfunktion EXCHANGE(): Paarweiser Tausch der Zellen zwischen A und B
- Minimierung der Zielfunktion COST(): Summe der Netzgewichte der geschnittenen Netze



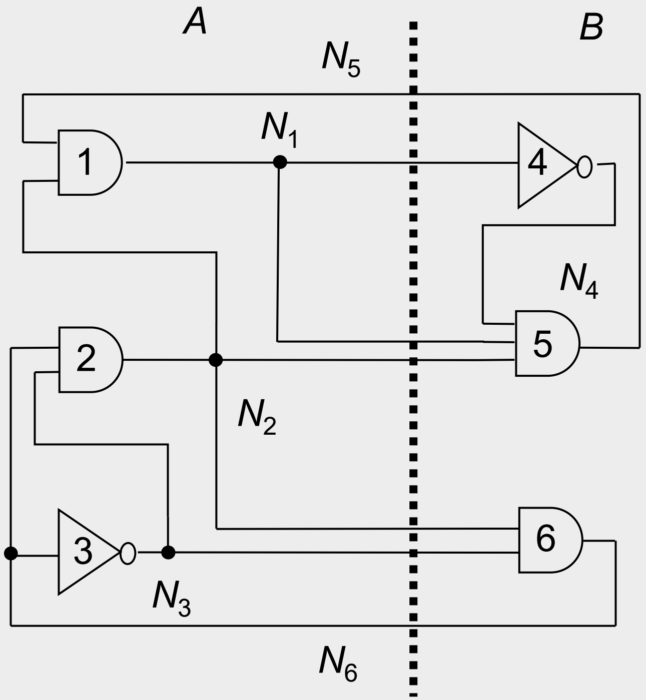
Anfangspartitionierung:
A {1,2,3}, *B* {4,5,6}
Kosten: 13

Zähler i	T	a_i, b_i	*cur_cost*	*trial_cost*	Zufallszahl r	$e^{-\frac{\Delta cost}{T}}$	Tausch
1	10,00		13				



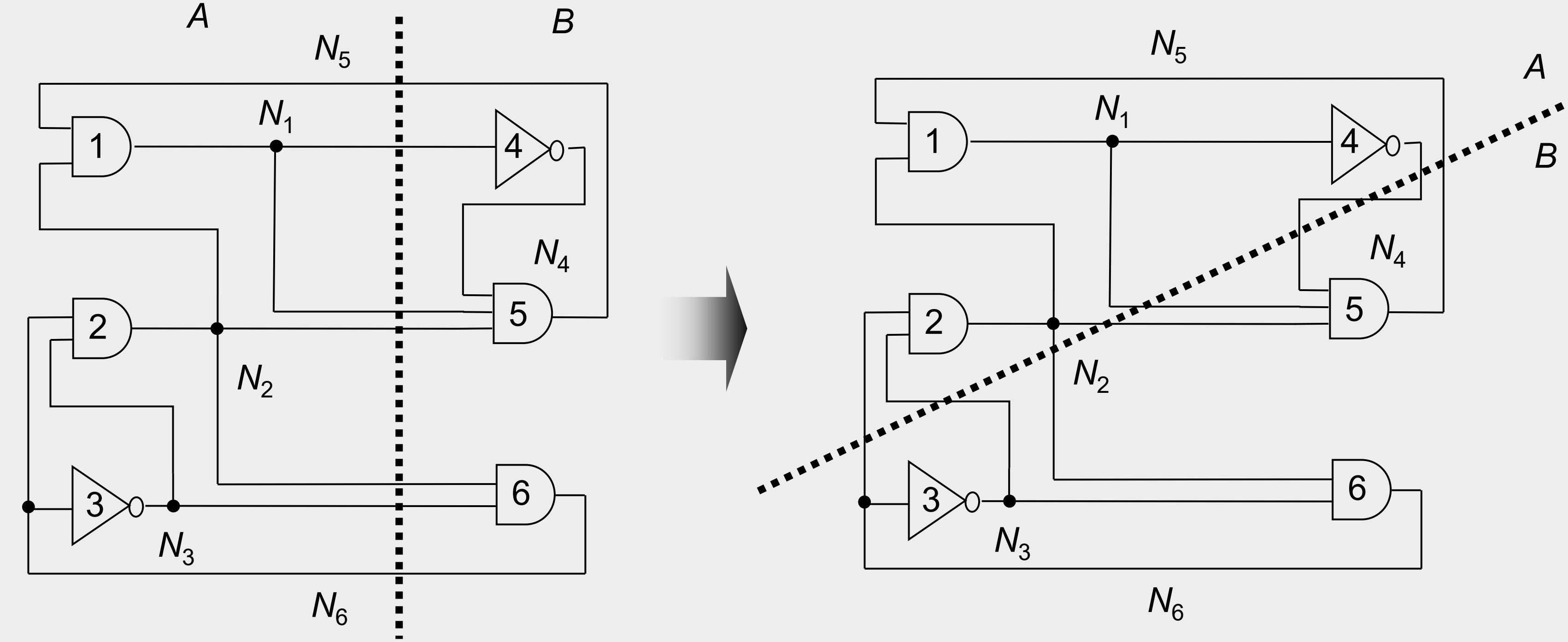
Anfangspartitionierung:
A {1,2,3}, *B* {4,5,6}
Kosten: 13

Zähler i	T	a_i, b_i	*cur_cost*	*trial_cost*	Zufallszahl r	$e^{-\frac{\Delta cost}{T}}$	Tausch
1	10,00	3,4	13				



Anfangspartitionierung:
A {1,2,3}, *B* {4,5,6}
Kosten: 13

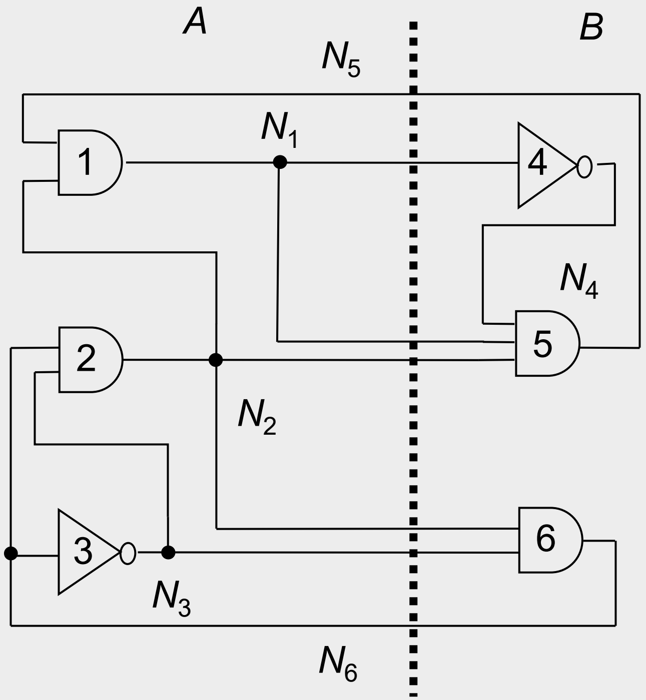
Zähler i	T	a_i, b_i	*cur_cost*	*trial_cost*	Zufallszahl r	$e^{-\frac{\Delta cost}{T}}$	Tausch
1	10,00	3,4	13	16			



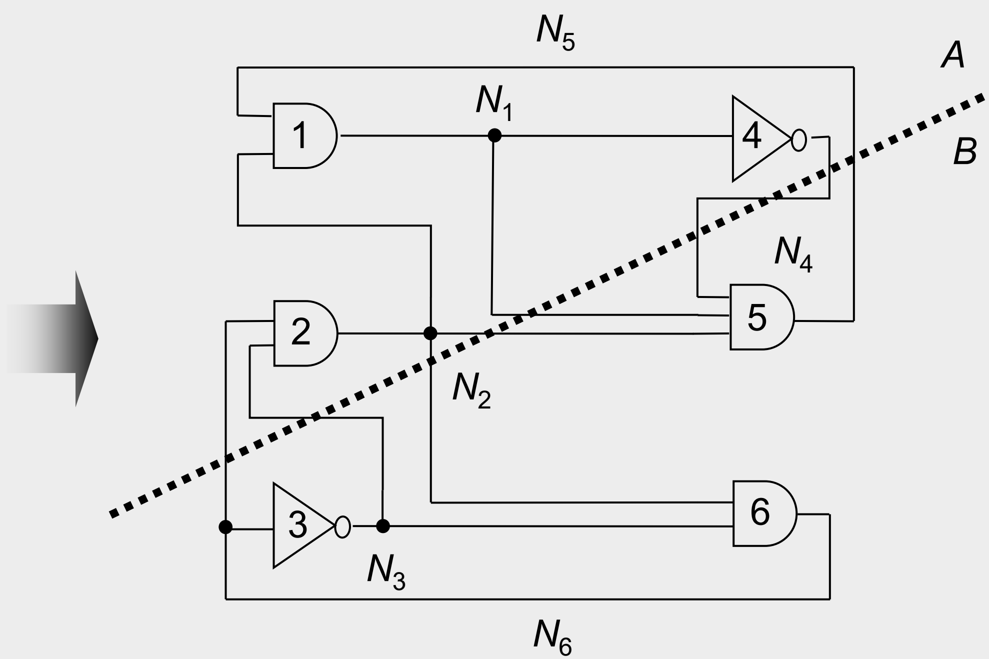
Anfangspartitionierung:
A {1,2,3}, *B* {4,5,6}
Kosten: 13

A {1,2,4}, *B* {3,5,6}
Kosten: 16

Zähler i	T	a_i, b_i	*cur_cost*	*trial_cost*	Zufallszahl r	$e^{-\frac{\Delta cost}{T}}$	Tausch
1	10,00	3,4	13	16	0,21713	0,74082	Ja



Anfangspartitionierung:
A {1,2,3}, *B* {4,5,6}
Kosten: 13



A {1,2,4}, *B* {3,5,6}
Kosten: 16

Zähler i	T	a_i, b_i	*cur_cost*	*trial_cost*	Zufallszahl r	$e^{-\frac{\Delta cost}{T}}$	Tausch
1	10,00	3,4	13	16	0,21713	0,74082	Ja

Zähler i	T	a_i, b_i	*cur_cost*	*trial_cost*	Zufallszahl r	$e^{-\frac{\Delta cost}{T}}$	Tausch
1	10,00	3,4	13	16	0,21713	0,74082	Ja
2	10,00	2,3	16	16	0,66138	1	Ja
3	10,00	4,6	16	13			Ja
4	10,00	1,2	13	**2**			Ja
5	7,00	2,5	2	16	0,11209	0,13534	Ja
6	7,00	1,5	16	13			Ja
7	7,00	1,4	13	15	0,33190	0,75148	Ja
8	7,00	2,6	15	15	0,12564	1	Ja
9	4,90	1,3	15	16	0,70105	0,81540	Ja
10	4,90	3,4	16	13			Ja
11	4,90	1,6	13	2			Ja
12	4,90	5,6	2	16	0,49375	0,05743	Nein
13	3,43	1,3	2	13	0,86493	0,04048	Nein
14	3,43	2,5	2	16	0,34371	0,01688	Nein
15	3,43	1,6	2	13	0,73232	0,04048	Nein
16	3,43	1,2	2	13	0,02093	0,04048	Ja

Zähler i	T	a_i, b_i	*cur_cost*	*trial_cost*	Zufallszahl r	$e^{-\frac{\Delta cost}{T}}$	Tausch
1	10,00	3,4	13	16	0,21713	0,74082	Ja
2	10,00	2,3	16	16	0,66138	1	Ja
3	10,00	4,6	16	13			Ja
4	10,00	1,2	13	**2**			Ja
5	7,00	2,5	2	16	0,11209	0,13534	Ja
6	7,00	1,5	16	13			Ja
7	7,00	1,4	13	15	0,33190	0,75148	Ja
8	7,00	2,6	15	15	0,12564	1	Ja
9	4,90	1,3	15	16	0,70105	0,81540	Ja
10	4,90	3,4	16	13			Ja
11	4,90	1,6	13	2			Ja
12	4,90	5,6	2	16	0,49375	0,05743	Nein
13	3,43	1,3	2	13	0,86493	0,04048	Nein
14	3,43	2,5	2	16	0,34371	0,01688	Nein
15	3,43	1,6	2	13	0,73232	0,04048	Nein
16	3,43	1,2	2	13	0,02093	0,04048	Ja

Veranschaulichung ausgewählter Iterationen i:

i = 4 (T = 10):

- Selektierte Zellen zum Tausch sind 1, 2; *cur_cost* = 13, *trial_cost* = 2, d.h. Tausch wird akzeptiert

Zähler i	T	a_i, b_i	*cur_cost*	*trial_cost*	Zufallszahl r	$e^{-\frac{\Delta cost}{T}}$	Tausch
1	10,00	3,4	13	16	0,21713	0,74082	Ja
2	10,00	2,3	16	16	0,66138	1	Ja
3	10,00	4,6	16	13			Ja
4	10,00	1,2	13	**2**			Ja
5	7,00	2,5	2	16	0,11209	0,13534	Ja
6	7,00	1,5	16	13			Ja
7	7,00	1,4	13	15	0,33190	0,75148	Ja
8	7,00	2,6	15	15	0,12564	1	Ja
9	4,90	1,3	15	16	0,70105	0,81540	Ja
10	4,90	3,4	16	13			Ja
11	4,90	1,6	13	2			Ja
12	4,90	5,6	2	16	0,49375	0,05743	Nein
13	3,43	1,3	2	13	0,86493	0,04048	Nein
14	3,43	2,5	2	16	0,34371	0,01688	Nein
15	3,43	1,6	2	13	0,73232	0,04048	Nein
16	3,43	1,2	2	13	0,02093	0,04048	Ja

Veranschaulichung ausgewählter Iterationen *i*:

$i = 4$ ($T = 10$):

- Selektierte Zellen zum Tausch sind 1, 2; *cur_cost* = 13, *trial_cost* = 2, d.h. Tausch wird akzeptiert

$i = 5$ ($T = 7$):

- Selektierte Zellen zum Tausch sind 2, 5; *cur_cost* = 2, *trial_cost* = 16, d.h. Zufallszahl wird generiert: 0,11209
- 0,11209 < $e^{-14/7}$ = 0,13534, d.h. Tausch wird akzeptiert.

Zähler i	T	a_i, b_i	*cur_cost*	*trial_cost*	Zufallszahl r	$e^{-\frac{\Delta cost}{T}}$	Tausch
1	10,00	3,4	13	16	0,21713	0,74082	Ja
2	10,00	2,3	16	16	0,66138	1	Ja
3	10,00	4,6	16	13			Ja
4	10,00	1,2	13	**2**			Ja
5	7,00	2,5	2	16	0,11209	0,13534	Ja
6	7,00	1,5	16	13			Ja
7	7,00	1,4	13	15	0,33190	0,75148	Ja
8	7,00	2,6	15	15	0,12564	1	Ja
9	4,90	1,3	15	16	0,70105	0,81540	Ja
10	4,90	3,4	16	13			Ja
11	4,90	1,6	13	2			Ja
12	4,90	5,6	2	16	0,49375	0,05743	Nein
13	3,43	1,3	2	13	0,86493	0,04048	Nein
14	3,43	2,5	2	16	0,34371	0,01688	Nein
15	3,43	1,6	2	13	0,73232	0,04048	Nein
16	3,43	1,2	2	13	0,02093	0,04048	Ja

Veranschaulichung ausgewählter Iterationen i:

i = 4 (T = 10):

- Selektierte Zellen zum Tausch sind 1, 2; *cur_cost* = 13, *trial_cost* = 2, d.h. Tausch wird akzeptiert

i = 5 (T = 7):

- Selektierte Zellen zum Tausch sind 2, 5; *cur_cost* = 2, *trial_cost* = 16, d.h. Zufallszahl wird generiert: 0,11209
- 0,11209 < $e^{-14/7}$ = 0,13534, d.h. Tausch wird akzeptiert.

i = 12-15 (T = 4,9 bzw. T = 3,43):

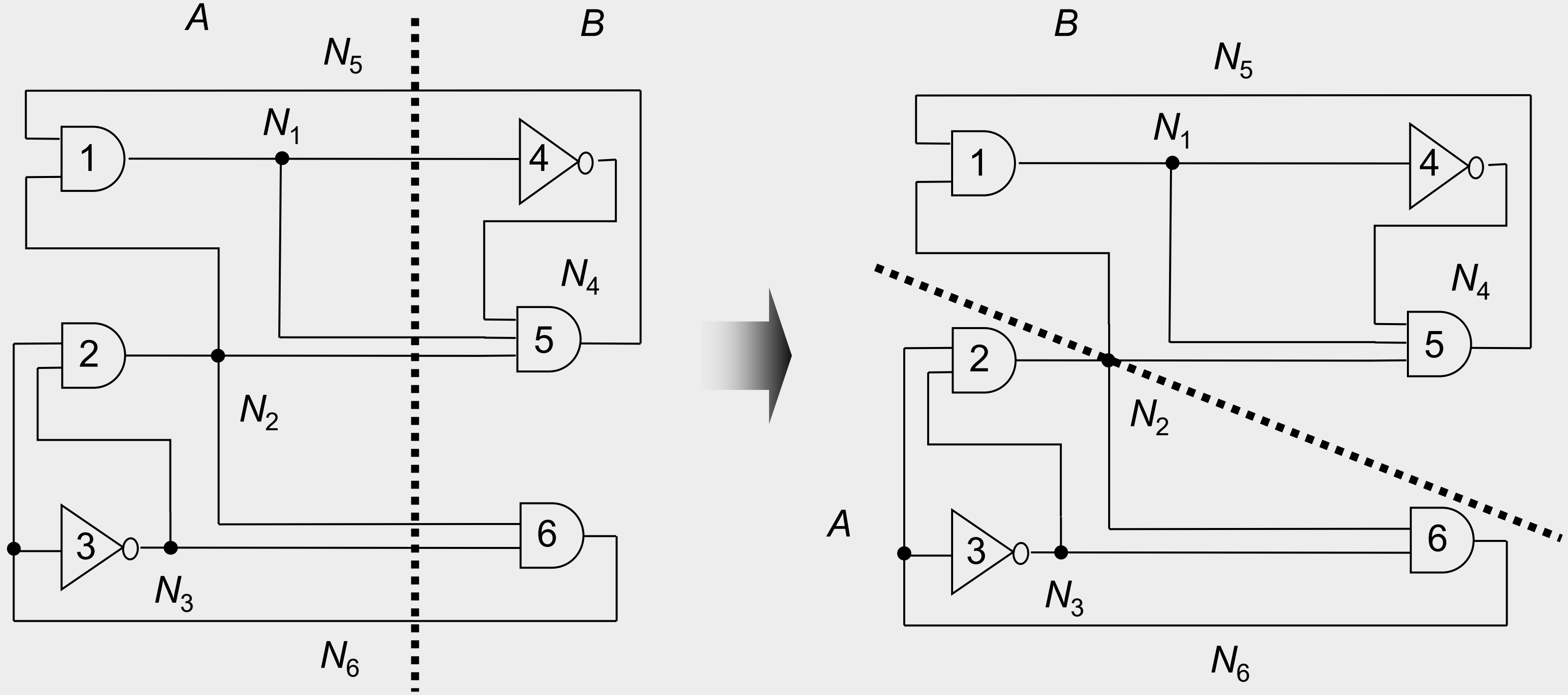
- *trial_cost* jeweils größer als *cur_cost* und Zufallszahl r jeweils größer als $e^{-\Delta \cos t / T}$, d.h. Vertauschungen nicht akzeptiert

Zähler i	T	a_i, b_i	*cur_cost*	*trial_cost*	Zufallszahl r	$e^{-\frac{\Delta cost}{T}}$	Tausch
1	10,00	3,4	13	16	0,21713	0,74082	Ja
2	10,00	2,3	16	16	0,66138	1	Ja
3	10,00	4,6	16	13			Ja
4	10,00	1,2	13	**2**			Ja
5	7,00	2,5	2	16	0,11209	0,13534	Ja
6	7,00	1,5	16	13			Ja
7	7,00	1,4	13	15	0,33190	0,75148	Ja
8	7,00	2,6	15	15	0,12564	1	Ja
9	4,90	1,3	15	16	0,70105	0,81540	Ja
10	4,90	3,4	16	13			Ja
11	4,90	1,6	13	2			Ja
12	4,90	5,6	2	16	0,49375	0,05743	Nein
13	3,43	1,3	2	13	0,86493	0,04048	Nein
14	3,43	2,5	2	16	0,34371	0,01688	Nein
15	3,43	1,6	2	13	0,73232	0,04048	Nein
16	3,43	1,2	2	13	0,02093	0,04048	Ja

- In der 4. Iteration (i = 4) erreicht der SA-Algorithmus einen Kostenwert von 2, welcher den besten erzielbaren Wert darstellt.
- Dieser repräsentiert die Summe der Netzgewichte der geschnittenen Netze (hier Netz 2 mit w_2 = 2).
- Wie dem detaillierten Ablauf zu entnehmen ist, wurde dieses Optimum durch die vorherige Akzeptanz von einer schlechteren Lösung in der 1. Iteration erreicht.

Die beste Lösung entspricht einer Partitionierung von A{2, 3, 6} und B{1, 4, 5}.

Die beste Lösung entspricht einer Partitionierung von *A* {2,3,6} und *B* {1,4,5}



Anfangspartitionierung:
A {1,2,3}, *B* {4,5,6}
Kosten: 13

Erzielte Lösung:
A {2,3,6}, *B* {1,4,5}
Kosten: 2

Zusammenfassung Kapitel 2 – Partitionierung