

Vorlesung: Informationstechnik (IT)

Prof. Dr.-Ing. K.D. Müller-Glaser

Institutsleitung

Prof. Dr.-Ing. K. D. Müller-Glaser

Prof. Dr.-Ing. J. Becker

Prof. Dr. rer. nat. W. Stork

Institut für Technik der Informationsverarbeitung (ITIV)



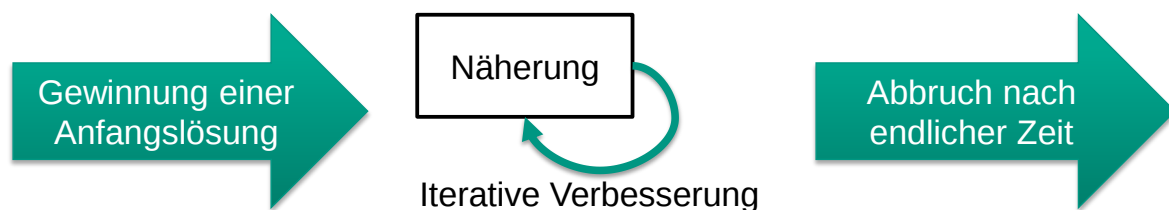
Optimierungsalgorithmen

KIT – Universität des Landes Baden-Württemberg und
nationales Forschungszentrum in der Helmholtz-Gemeinschaft

www.kit.edu

Problemstellung

- Bei komplexen Problemen ist exakte Lösung oft nicht möglich
- Gründe dafür:
 - Formulierung als Algorithmus ist nicht möglich
 - Exakter Algorithmus hat Rechenzeit der Ordnung $O(n^x)$ oder $O(x^n)$ oder sogar $O(n!)$
- Pragmatische Lösung: Heuristische Verfahren



Exakte Lösung häufig nicht möglich wegen der Rechenzeit

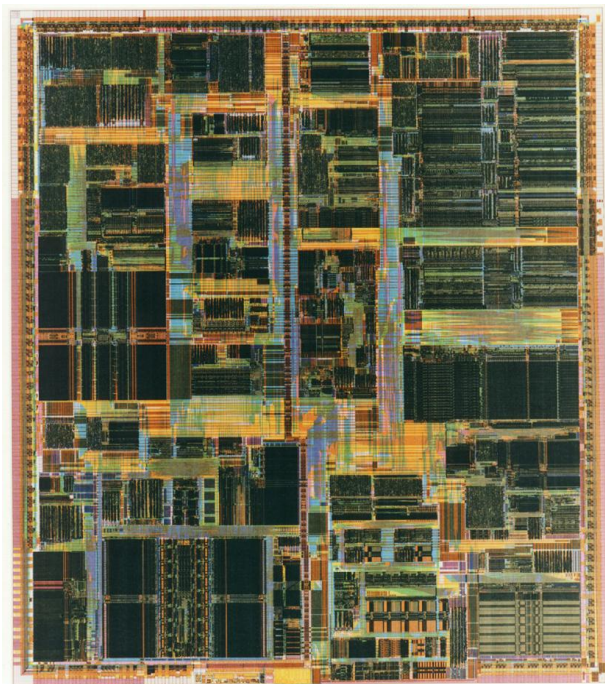
Ordnung $O(n^x)$ oder $O(x^n)$ oder sogar $O(n!)$

Pragmatische Lösung: Heuristische Verfahren

Heuristische Verfahren aus 2 Schritten:

1. Gewinnung einer Anfangslösung
2. Verfahren zur iterativen Verbesserung

Iterative Verfahren sind: Greedy, Probabilistisch, Hill climbing



P6 Dieshot

■ Teilaufgaben beim Entwurf eines Chiplayout:

- Partitionierung
- Platzierung
- Verdrahtung

■ Viele Nebenbedingungen:

- Geringe Gesamtfläche
- Temperaturverhalten
- ...

Partitionierung

Aufteilung in kleine Teilschaltungen

Platzierung

Hauptbedingungen:

Geringe Gesamtfläche

Geringe Gesamtnetzlänge

Nebenbedingungen:

Zeit- und Temperaturverhalten

Halo-Generierung

Platzierungsstrategien

Konstruktive Verfahren

Iterative Verfahren

Verdrahtung

Globales Verdrahten

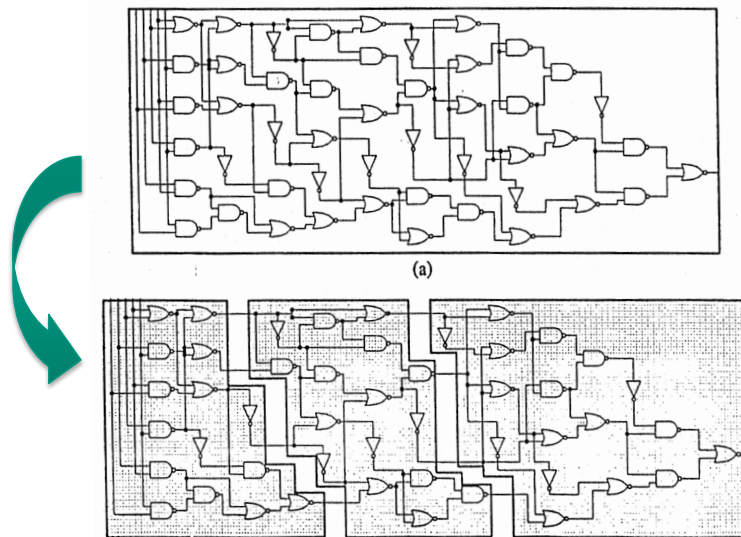
Aufteilung der Verdrahtungsfläche in Kanäle

Zuweisung der Netze auf Kanäle

Detailliertes Verdrahten

Partitionierung

- Aufteilung der (geschätzten) Fläche eines Chips in einzelne funktionale Blöcke, die mit anderen Blöcken durch möglichst wenige kurze Leitungen verbunden sind:



Partitionierung – Aufgabenstellung

- Aufgabe: Finde eine Partition P der Modulmenge $M = \{ m_1, \dots, m_i, \dots, m_n \}$ unter Randbedingungen:
 - Zahl der Verbindungen
 - Flächen der Module
 - ...
- Vereinfachung: Finde eine Bipartition $P = \{ B_1, B_2 \}$ mit minimaler Kostenfunktion
- Heuristische Verfahren:
 - Konstruktive Methoden: Anlagerungsverfahren
 - Iteratives Verbessern: Random Interchange, Kernighan-Lin, Simulated Annealing, Evolutionäre Algorithmen

- **Aufgabe:** Finde eine Partition P der Modulmenge $M = \{ m_1, \dots, m_i, \dots, m_n \}$ unter verschiedenen Randbedingungen (Zahl der Verbindungen, Flächen der Module, usw.)
- **Vereinfachung:** Finde eine Bipartition $P = \{ B_1, B_2 \}$, für die gilt: Die Kostenfunktion ist minimal.
- C_{ij} : Zahl der Netze, die den Modul m_i mit dem Modul m_j verbinden, und m_i und m_j liegen in verschiedenen Blöcken B_1 und B_2 .
- Für die Blöcke B_1 und B_2 der Partition P gilt:
 - Heuristische Vorgehensweise:
 - Konstruktive Methoden (Anlagerungsverfahren, paralleles Verfahren)
 - Iteratives Verbessern (Zufälliges Vertauschen, Kernighan-Lin, Simulated Annealing, Evolutionäre Algorithmen)

Partitionierung – Überdeckung

■ Mathematische Definition

■ Für Partitionierung gilt:

Modulmenge $M = \{ m_1, \dots, m_i, \dots, m_n \}$

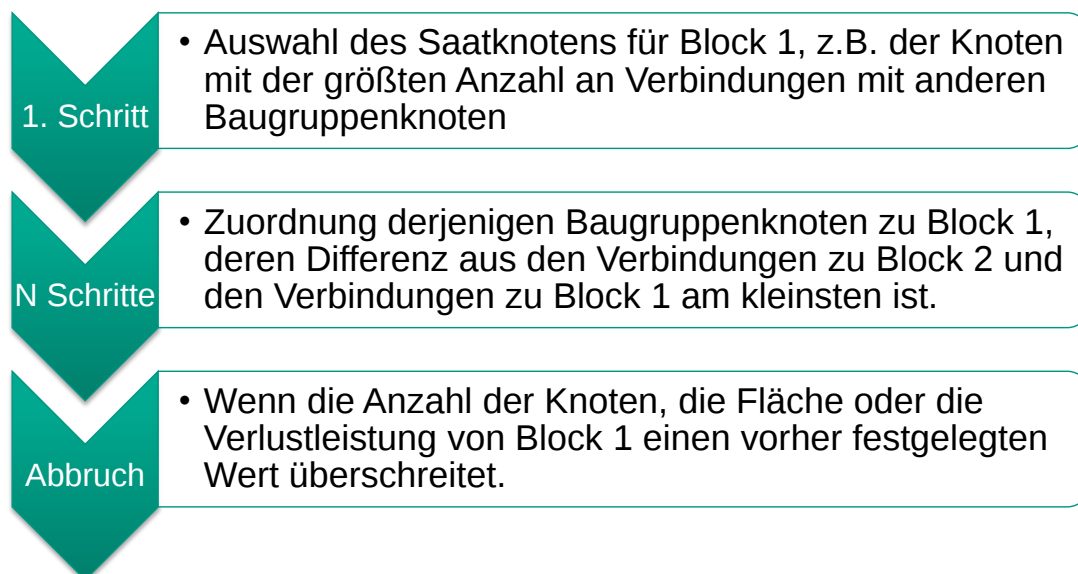
Partition $P = \{ B_1, B_2, B_3, \dots, B_k \}$

$$\bigcup_{i=1}^k B_i = M \quad \text{und} \quad \forall i, j | i \neq j \rangle B_i \cap B_j = \emptyset$$

■ Für Überdeckung gilt lediglich

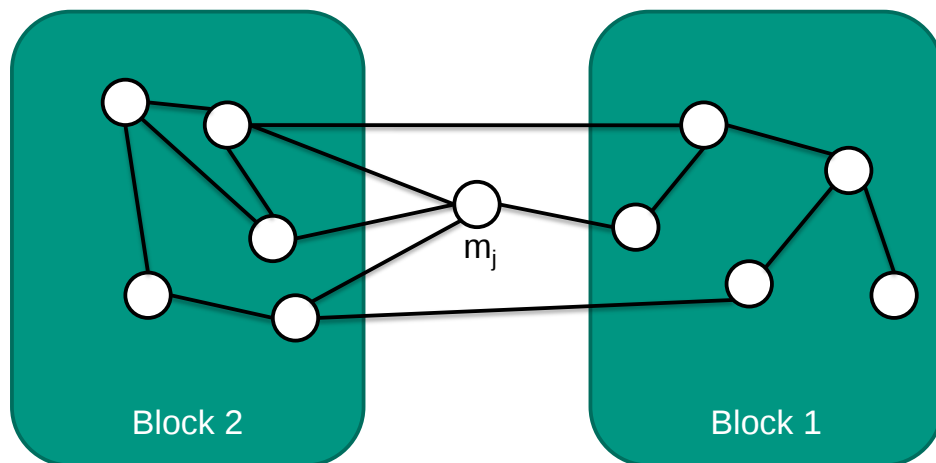
$$\bigcup_{i=1}^k B_i = M$$

Anlagerungsverfahren



■ Problem: Kann in lokalem Minimum der Kostenfunktion enden

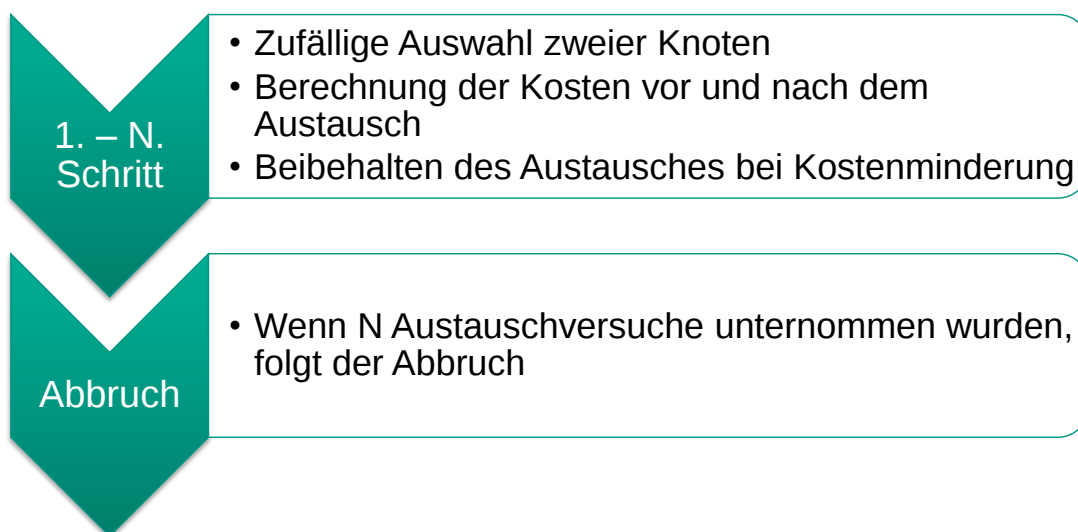
Beispiel: Anlagerungsverfahren



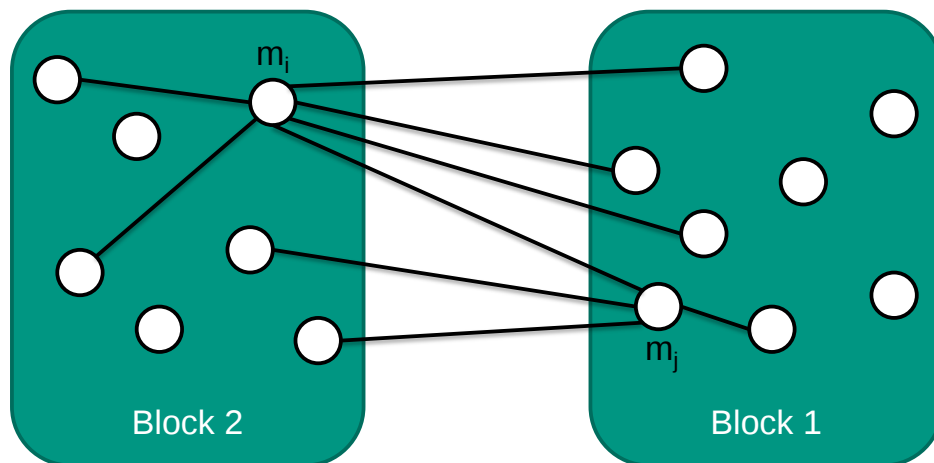
- Verbindungen zu Block 1: 1
- Verbindungen zu Block 2: 3
- Folglich Zuordnung zu Block 2

Random Interchange

- Iteratives Verbessern einer bestehenden Bipartition durch zufälliges Vertauschen von Knoten:



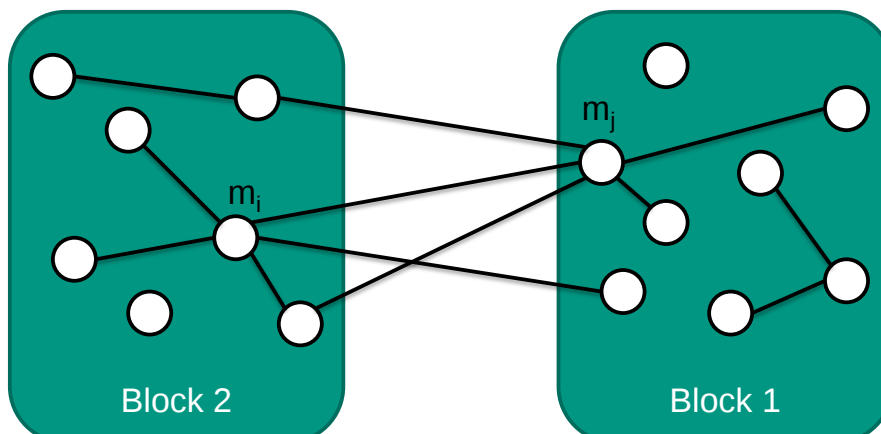
Beispiel: Random Interchange



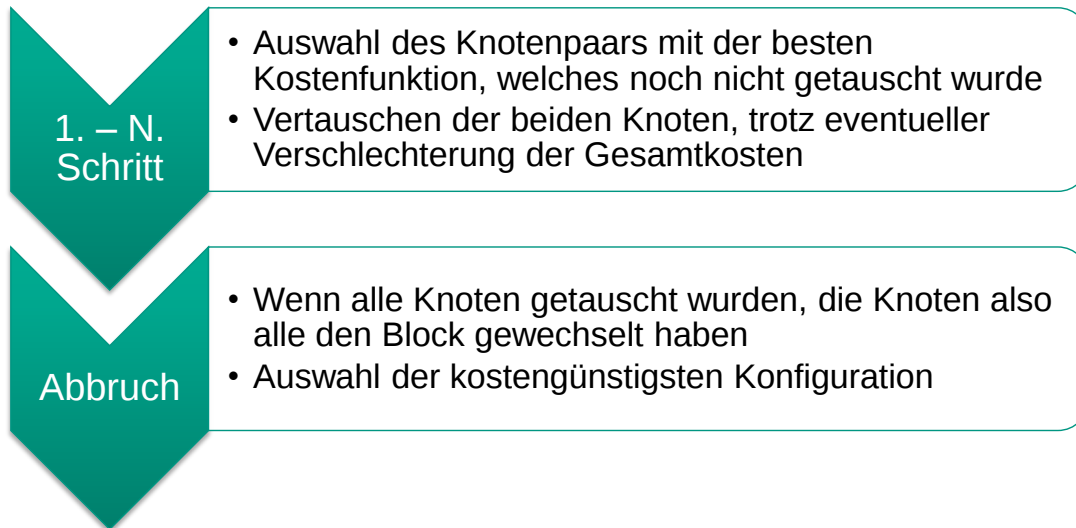
- Knoten m_i : innere: 2 externe: 4 c_{ij} : 1
- Knoten m_j : innere: 1 externe: 3 c_{ij} : 1
- Bewertungsfunktion: $D = E(m_i) + E(m_j) - I(m_i) - I(m_j) - 2 c_{ij}$
- $D=2 > 0$: Verbesserung, also Austausch von m_i und m_j

Zwischenübung 01: Random Interchange

- Bringt die Vertauschung von Knoten m_i und m_j einen Kostenvorteil?

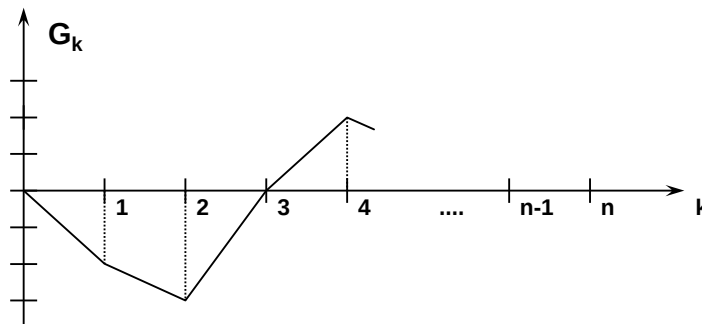


- Systematisches Vertauschen aller Knoten der beiden Partitionsblöcke und Erstellen einer Kostenfunktion:



- Systematisches Vertauschen aller Module zweier Blöcke einer Partition mit Abspeicherung der Kosten vor und nach dem Tausch.
- Kostengünstigste Auswahl aller Austauschmöglichkeiten, auch wenn sich eine Erhöhung der Kosten ergeben sollte.
- Darauf folgende Schritte: Betrachtung nur noch der nicht ausgetauschten Module.
- Ende, wenn alle Module ausgetauscht wurden, d.h. die Elemente der beiden Blöcke ihre Plätze getauscht haben.
- Auswahl der besten Partition anhand der abgespeicherten Veränderung der Gesamtkosten.
- Bewertungsfunktion wie bei Random Interchange:
- $D(m_i, m_j) = E(m_i) + E(m_j) - I(m_i) - I(m_j) - 2 c_{ij}$

Kernighan-Lin-Algorithmus



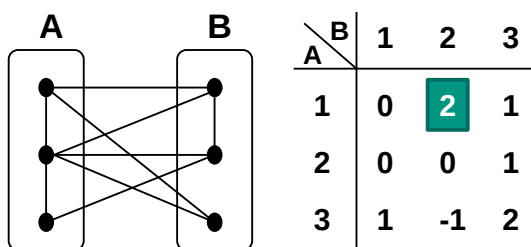
$$G_k = \sum_{i=1}^k D_i$$

- Gleiche Bewertungsfunktion wie bei Random Interchange:

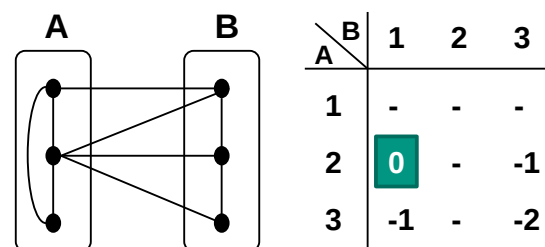
$$D(m_i, m_j) = E(m_i) + E(m_j) - I(m_i) - I(m_j) - 2 c_{ij}$$
- Verlauf der Bewertungsfunktion über der Anzahl der Vertauschungen k, mit D_i = Gewinn beim i-ten Austausch
- Der ausgewählte Satz von Vertauschungen korrespondiert mit dem maximalen Gewinn $[\max G_k]$

Beispiel: Kernighan-Lin

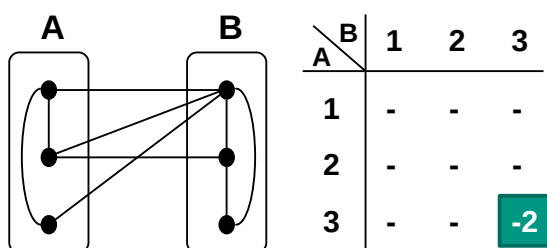
1.



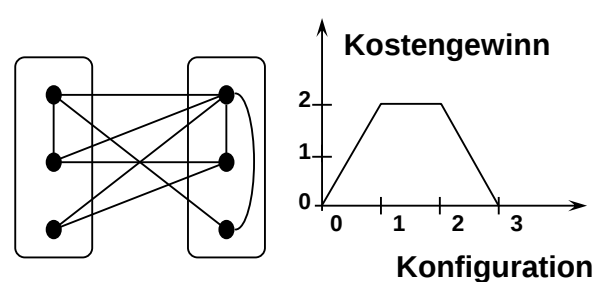
2.



3.

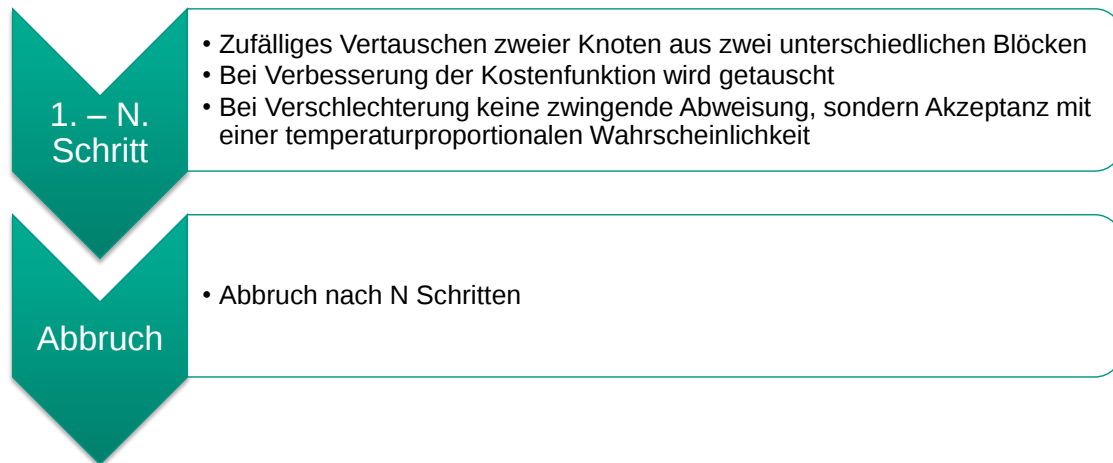


4.



Simulated Annealing

- Verbesserung des Random Interchange-Algorithmus
- Vorbild: Kristallbildung aus einer Schmelze bei langsamer Temperatursenkung



Link zur Demonstration:

<http://www-i1.informatik.rwth-aachen.de/~algorithmus/algo41.php>

- Zufälliges Tauschen eines Moduls aus zwei Blöcken einer Partition
- Berechnung des Kostenunterschieds vorher und nachher
- Bei Verbesserung wird getauscht
- Bei Verschlechterung keine Abweisung, sondern Akzeptanz mit einer temperaturabhängigen Wahrscheinlichkeit
- Wahrscheinlichkeit, einen verschlechternden Tausch durchzuführen, wird mit sinkender Temperatur geringer

Noch umfangreichere Demonstration anhand des Problem des Handlungsreisenden:

<http://www.math.tu-clausthal.de/Arbeitsgruppen/Stochastische-Optimierung/tspapplet/applet.html>

Pseudo Code: Simulated Annealing

```

simulate_annealing() {
    T = T0; //Anfangswert der Temperatur
    X = j0; //Anfangskonfiguration

    while( !stop_criterion ) { //Endtemperatur, oder weniger als
                                //Min Austauschungen
        while( !loop_criterion ) { //Mindestanzahl Versuche bei
                                    //einer Temperatur
            j = generate( X ); //Generiere neue Konfiguration,
                                //also Austausch zweier Module
            if( accept( cost( j ), cost( X ), T ) ) //spezielle Akzeptanzfunktion
                X = j; //akzeptierte Lösung abspeichern
        }
        T = update( T ); //langames Senken der Temperatur
    } //z.B. T = 0.9T
}

```

Pseudo Code: Simulated Annealing

```

accept( cost( j ), cost( i ), T ) { //liefert eine 1, wenn ein Austausch
                                    //akzeptiert wird, wenn nicht, eine 0
    diff_cost = cost( j ) - cost( i ); //Differenz der Kosten berechnen

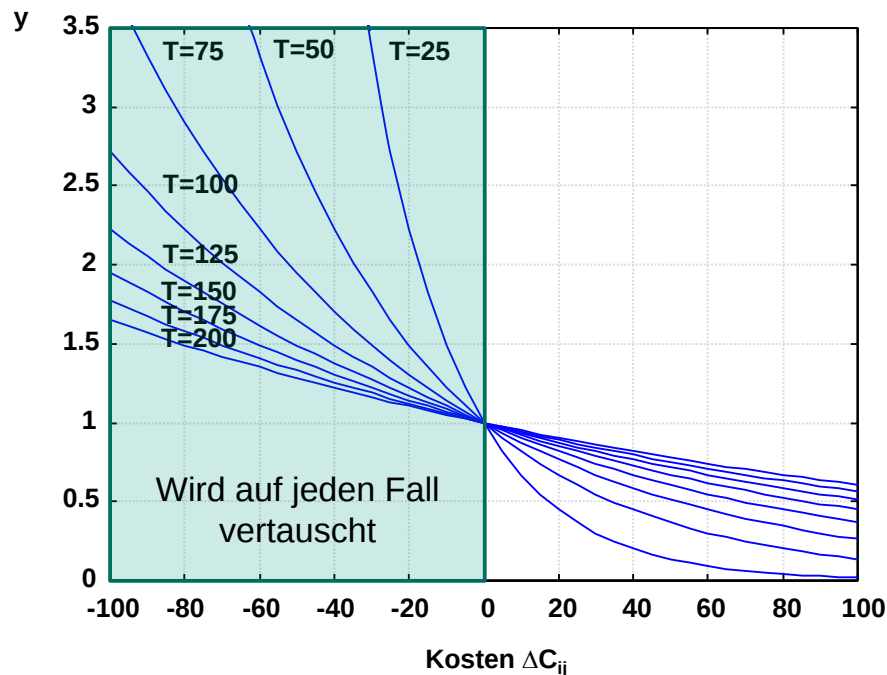
    y = exp( -diff_cost / T ); //je höher die Temperatur, desto größer wird y

    r = random; //Zufallszahl zwischen 0 und 1

    if( r < y )
        return 1; //akzeptiert
    else
        return 0; //oder nicht
}

```

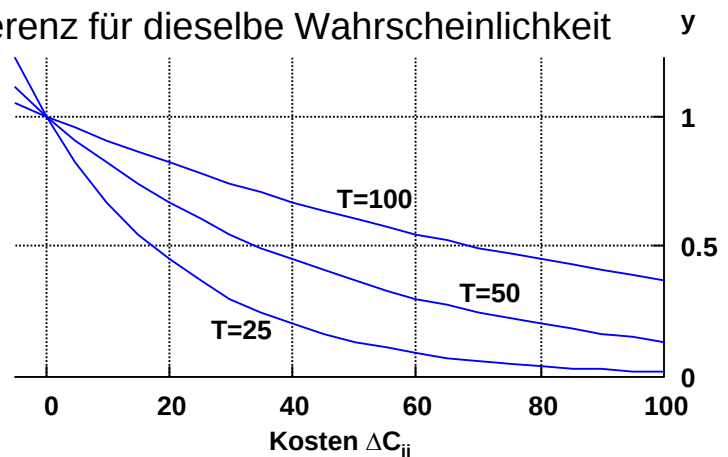
Simulated Annealing Akzeptanz



Zwischenübung02: Simulated Annealing



- Sie sehen die grafische Darstellung für den Fall einer Kostenverschlechterung bei verschiedenen Temperaturen. Beantworten Sie die folgenden Fragen:
- Welche Kostendifferenz wird für $T=25$ mit einer Wahrscheinlichkeit von 50% akzeptiert?
- Wie groß darf die Kostendifferenz für dieselbe Wahrscheinlichkeit bei $T=100$ maximal sein?

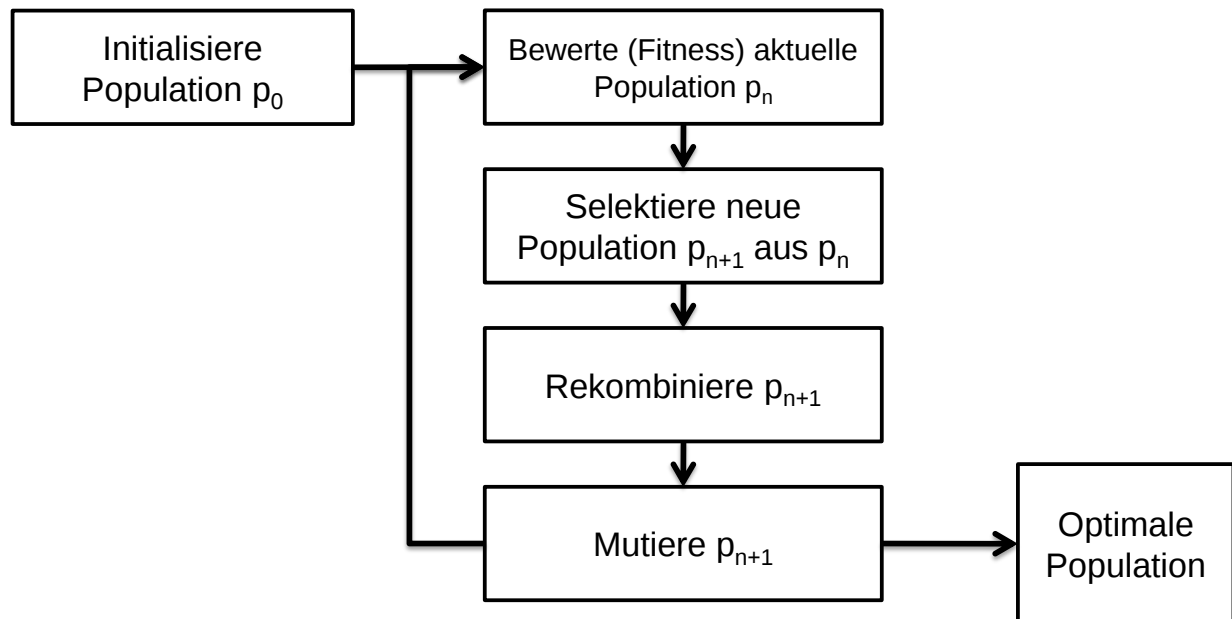


■ Merkmale evolutionärer Algorithmen:

- Außer Zielfunktion keinerlei weitergehende Informationen notwendig.
- Es wird mit Codierungen von Konfigurationen, nicht mit Konfigurationen selbst, gearbeitet.
- Es findet keine Verbesserung von Punkt zu Punkt, sondern von Population zu Population statt, d.h. es wird stets aus einer Menge (Population) von Konfigurationen (sog. Individuen) eine neue Population (sog. Generation) erzeugt.
- Die Übergangsregeln sind probabilistisch, nicht deterministisch.
- Der Übergang von einer Population zur nächsten, d.h. die Erzeugung einer neuen Generation, erfolgt dabei mit Hilfe von Operatoren. Die Veränderungen finden dabei stets anhand der Codierungen statt.

■ Operatoren:

- Selektion (Reproduktion)
ungeschlechtliche Fortpflanzung: Hierbei werden identische Nachkommen (Clone) gebildet. Je besser die Zielfunktion (engl. fitness) eines Individuums, um so häufiger kann es seine Erbmasse, d.h. sich selbst reproduzieren. Dieses Prinzip entspricht dem Prinzip des Überlebens des Besten (engl. survival of the fittest).
- Rekombination (Kreuzung)
Geschlechtliche Fortpflanzung: Zwei Individuen vermischen ihre Erbanlagen und "zeugen" zwei Nachkommen, die die Position ihrer Eltern in der Population übernehmen, d.h. Nachkommen ersetzen ihre Vorfahren.
- Mutation
Zufällige Änderungen der Erbmasse: Dieser Operator ist von untergeordneter Bedeutung, kann bei bestimmten Problemen jedoch zur Überwindung lokaler Optima hilfreich sein.



Parameter:

Codierung Code stellt Individuum J dar

Population P : Menge von (nicht notwendigerweise unterschiedlichen) Codierungen

g : die zu optimierende Funktion Fitnessfunktion

Konstanten:

Kreuzungswahrscheinlichkeit p_c

Mutationswahrscheinlichkeit p_m

Anfangspopulation P_0

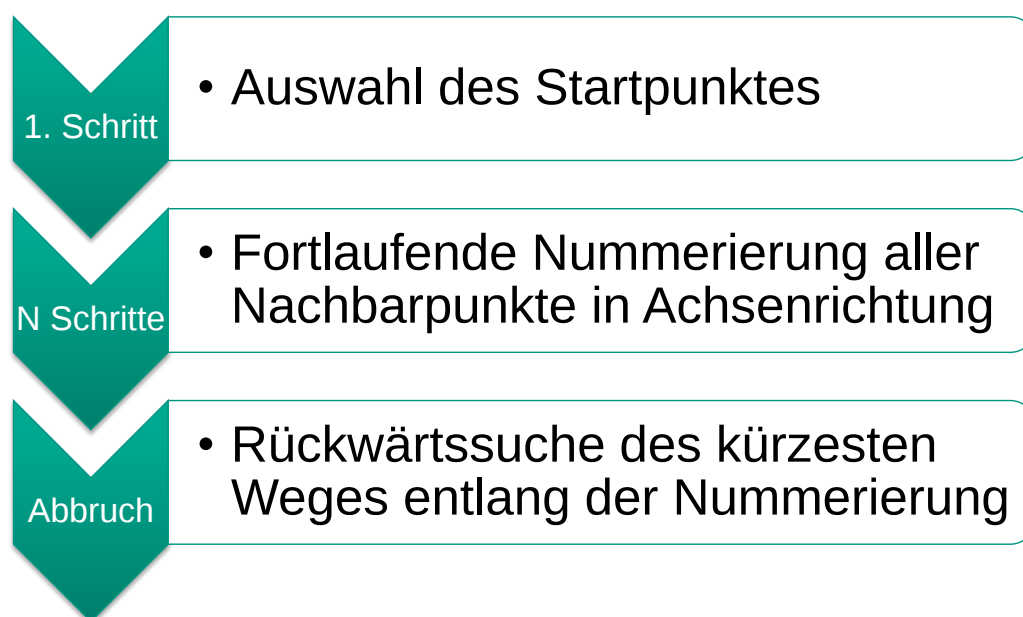
Gesucht:

"optimale" bzw. "bessere" Population. Eine optimale Population enthält eine Codierung mit der besten Fitness.

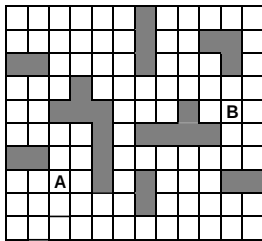
Verdrahtung

- *Aufgabe: „Bestimmung der geometrischen Strukturen für die Verbindungen zwischen den Zellen gemäß der Netzliste unter Berücksichtigung technologie- und schaltungsspezifischer Designregeln“*
- Technologieabhängigkeit:
 - Standard- und Makrozellen
 - Verdrahtungsfläche in einzelne Verdrahtungskanäle aufteilen
 - Globale Wegesuche
 - Festlegen der Verdrahtungsreihenfolge der Kanäle
 - Erzeugen der Endverdrahtung für jeden Kanal
 - Moderne Gate Arrays (Sea of Gates)
 - Aufteilung in Blöcke
 - Block für Block behandeln

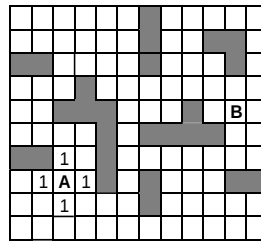
LEE-Algorithmus (maze routing)



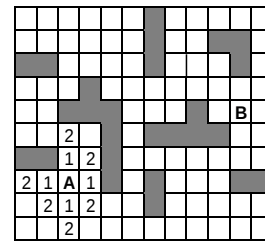
Beispiel für LEE-Algorithmus



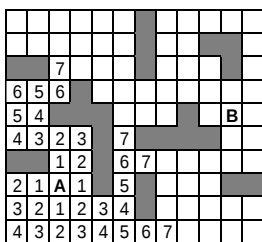
Verbindungspunkte



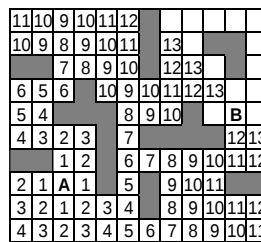
1. Welle



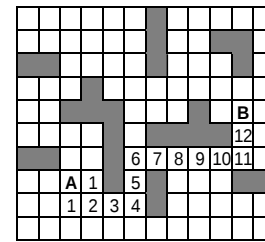
2. Welle



3. bis 7. Welle

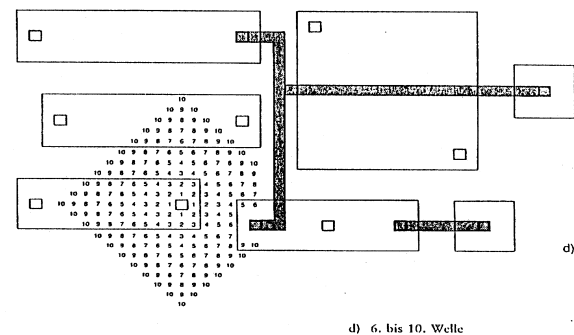
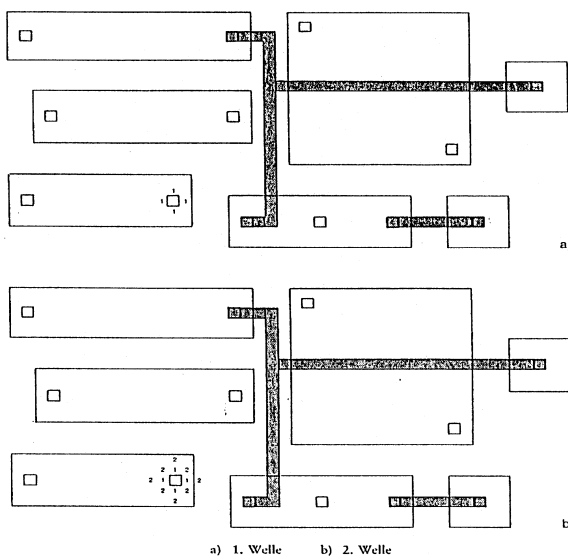


8. bis 13. Welle

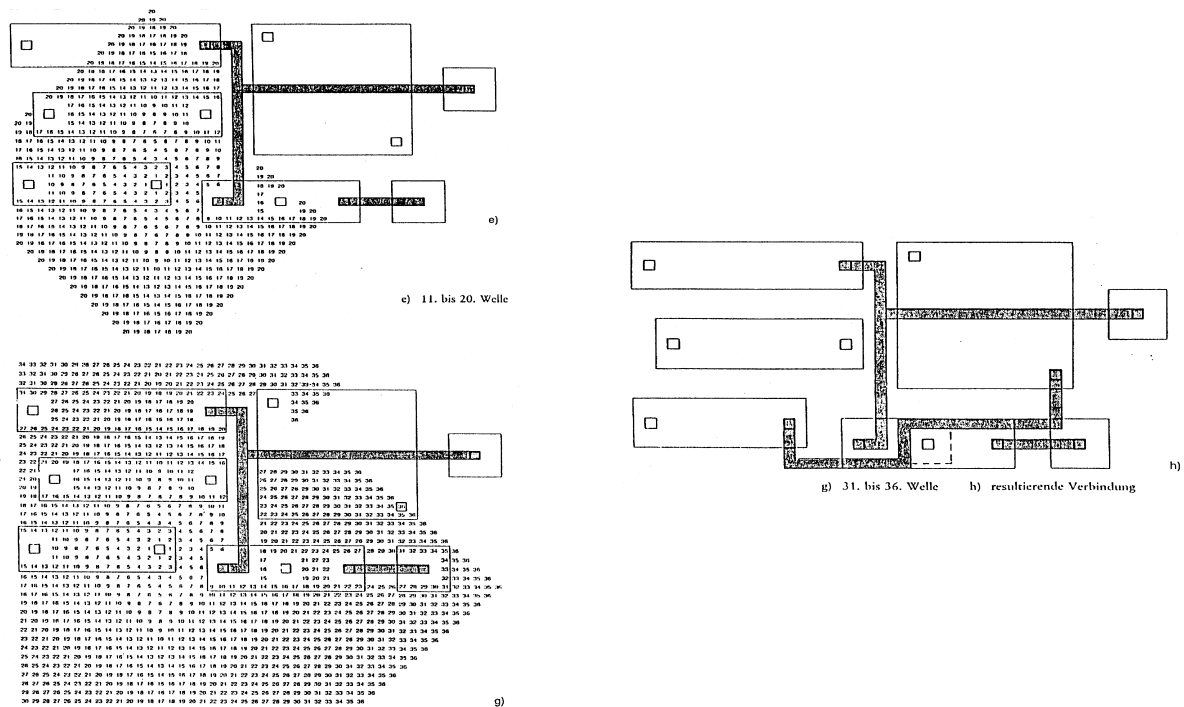


Rückverfolgung

Beispiel: LEE-Algorithmus



Beispiel: LEE-Algorithmus



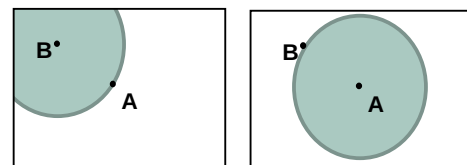
11-34 10.07.2013

Prof. Dr.-Ing. K.D. Müller-Glaser - Informationstechnik (IT)
Optimierungsalgorithmen

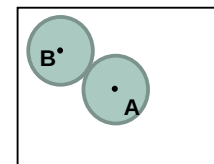
Institut für Technik der Informationsverarbeitung (ITIV)

Modifikation des LEE-Algorithmus

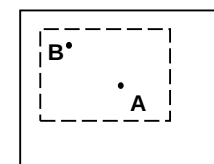
- Günstige Wahl des Startpunktes:



- Gleichzeitige Wegsuche von Start- und Zielpunkt:



- Einführung eines Suchrahmens:



- Übergang von Zweipunkt- zu Mehrpunktverbindungen
- Dreidimensionale Wegesuche möglich (Mehrlagenverbindungen)

11-35 10.07.2013

Prof. Dr.-Ing. K.D. Müller-Glaser - Informationstechnik (IT)
Optimierungsalgorithmen

Institut für Technik der Informationsverarbeitung (ITIV)