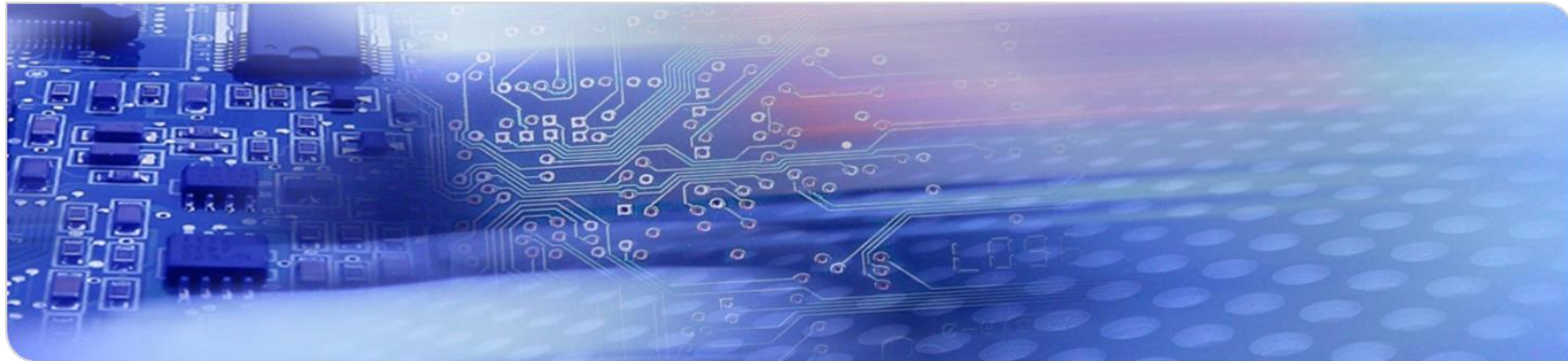


Hardware/Software Co-Design

Übung 7 - Wiederholung
M.Sc. Fabian Lesniak



Agenda

- Wiederholung ausgewählter Themen
 - Profiling
 - Partitionierungsalgorithmen
 - Hierarchical Clustering
 - Tabu-Search

3.5.3 Optimal Profiling Instrumentation

- How do we instrument the code to minimize overheads?
 - „Optimally Profiling and Tracing Programs“ – Thomas Ball, James Larus
- **QPT**, Programm-Profiler und Tracing-System
 - Instrumentierungs-basierte profiling und tracing Algorithmen
 - **Profiling:** Es werden die **Verzweigungs- bzw. Ausführungshäufigkeiten** jedes sequentiellen Programm-Basisblocks gemessen.
 - **Tracing:** Die **Ausführungsreihenfolge** der Programm-Basisblöcke wird protokolliert.
- Diese daraus gewonnenen Daten erlauben die Berechnung der **Ausführungskosten der Prozeduren** in einem Programm.

3.5.3 Profiling & Tracing mit QPT (I)

■ Definition: Basisblock

- Ein Basisblock ist eine **Sequenz von aufeinanderfolgenden Befehlen**, in welche der **Kontrollfluß eintritt und diese wieder verläßt**, ohne Halt oder Möglichkeit daraus zu verzweigen, ausgenommen an deren Ende.

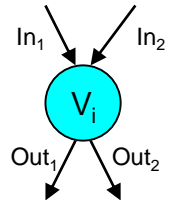
■ Definition: Kontrollflußgraph (CFG)

- Ein **Kontrollflußgraph (CFG)** ist ein gerichteter Graph (mit Wurzel) $G=(V, E)$ mit einem **speziellen EXIT-Knoten** (\neq Wurzel), so daß dieser Graph mit einem Programm auf folgende Weise korrespondiert:

Jeder Knoten in V repräsentiert einen **Basisblock** und jede Kante in E repräsentiert eine **Kontrollverzweigung** von einem Basisblock zu einem anderen.

■ Definition: Gewichte W eines CFG

- Als **Gewichte W des CFG** wird allen Kanten aus E ein nicht-negativer Wert zugeordnet, wobei die **Summe der Gewichte** der **eingehenden bzw. ausgehenden** Kanten eines Knotens **gleich** sind (Bilanz: $\sum in_i = \sum out_i$).
- Das **Gewicht eines Knotens** aus V ist einfach die **Summe der eingehenden bzw. nur ausgehenden Kanten** (=Ausführungshäufigkeit B_i).

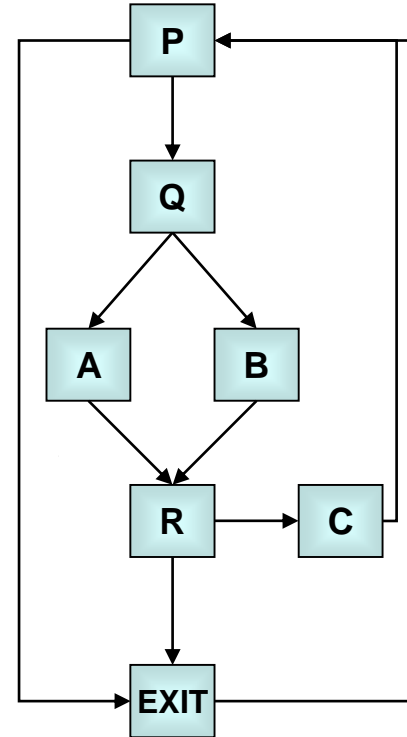


3.5.3 Profiling mit QPT (IV)

1. CFG erstellen
2. Initiale Messung der Häufigkeiten

```
program
  while P do
    begin
      if Q then
        A;
      else
        B;
      if R then
        break;
      C;
    end;
  end
```

Programmbeispiel



CFG mit Gewichten W

3.5.3 Profiling mit QPT (IV)

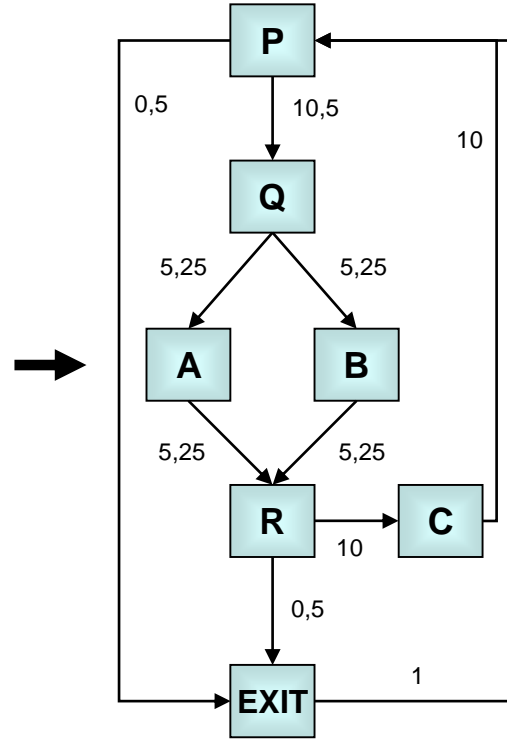
1. CFG erstellen
2. Initiale Messung der Häufigkeiten
3. Generierung des MAB

```

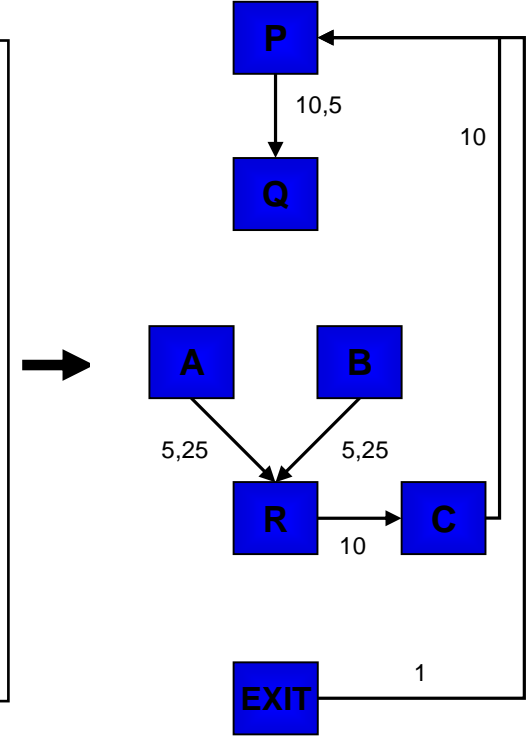
program
  while P do
    begin
      if Q then
        A;
      else
        B;
      if R then
        break;
      C;
    end;
  end

```

Programmbeispiel



CFG mit Gewichten W



Maximal-aufspannender Baum
(zusammenhängend + zyklensfrei !)

- epl liefert eine **minimale Lösung** bzgl. $EF(G, epl) \Leftrightarrow (E - epl)$ bildet einen **maximal aufspannenden Baum** G' von G

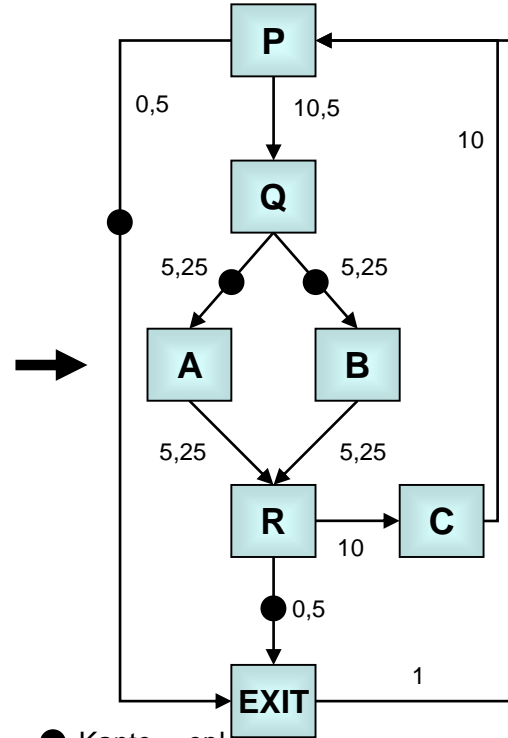
3.5.3 Profiling mit QPT (IV)

1. CFG erstellen
2. Initiale Messung der Häufigkeiten
3. Generierung des MAB
4. Platzierung der Zähler auf den fehlenden Kanten bzgl. CFG
5. Häufige Ausführung zur Generierung der Statistik und Ermittlung der mittleren Häufigkeiten.

```

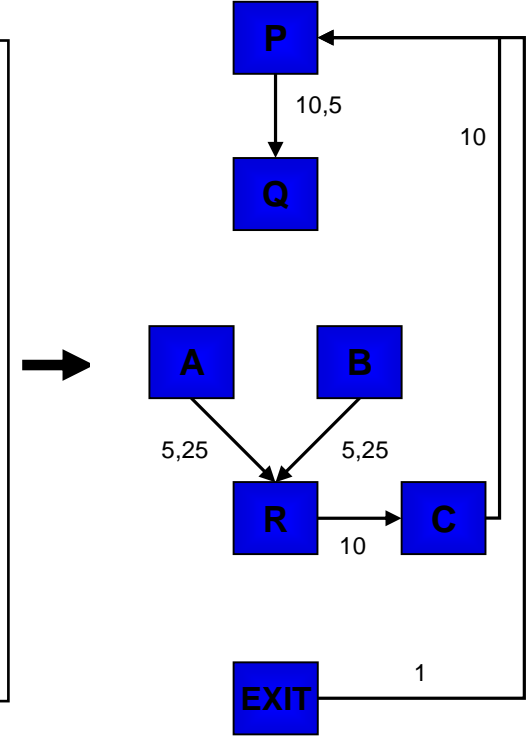
program
  while P do
    begin
      if Q then
        A;
      else
        B;
      if R then
        break;
      C;
    end;
  end
end
    
```

Programmbeispiel



● Kante \in epl

CFG mit Gewichten W



Maximal-aufspannender Baum
(zusammenhängend + zyklensfrei !)

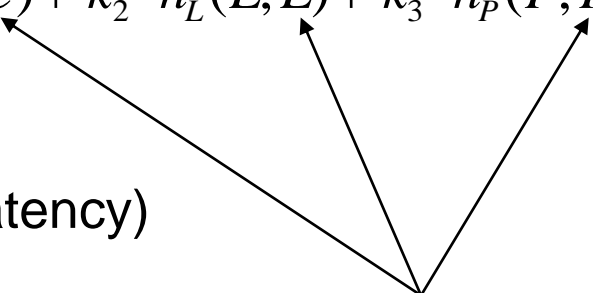
- epl liefert eine **minimale Lösung** bzgl. $EF(G, epl) \Leftrightarrow (E - epl)$ bildet einen **maximal aufspannenden Baum** G' von G

Partitionierung - Überblick

- **4.1 Einführung, Partitionierungsansätze, Komplexität**
- 4.2 Klassifikation von Partitionierungsalgorithmen
- 4.3 Konstruktive Algorithmen
 - 4.3.1 Hierarchisches Clustering
- 4.4 Iterative Algorithmen
 - 4.4.1 Tabu-Search
 - 4.4.2 Kernighan Lin
 - 4.4.3 Fiduccia Mattheyses
 - 4.4.4 Integer Linear Programming - ILP
 - 4.4.5 Simulated Annealing
 - 4.4.6 Genetische Algorithmen
- 4.5 Hardware/Software Partitionierungsverfahren und Co-Design-Systeme

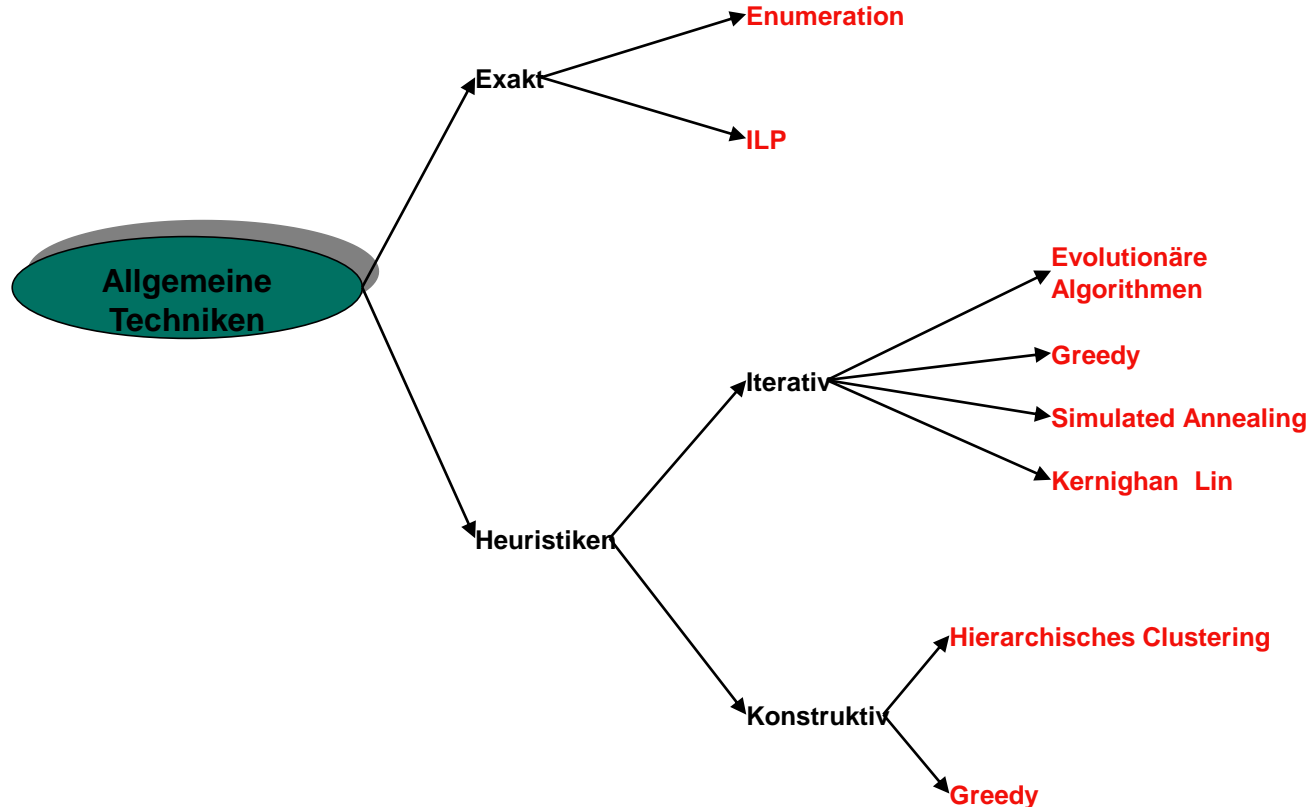
4.1 Kostenfunktionen zur Hardware/Software-Partitionierung

- Beispiel für eine Kostenfunktion:

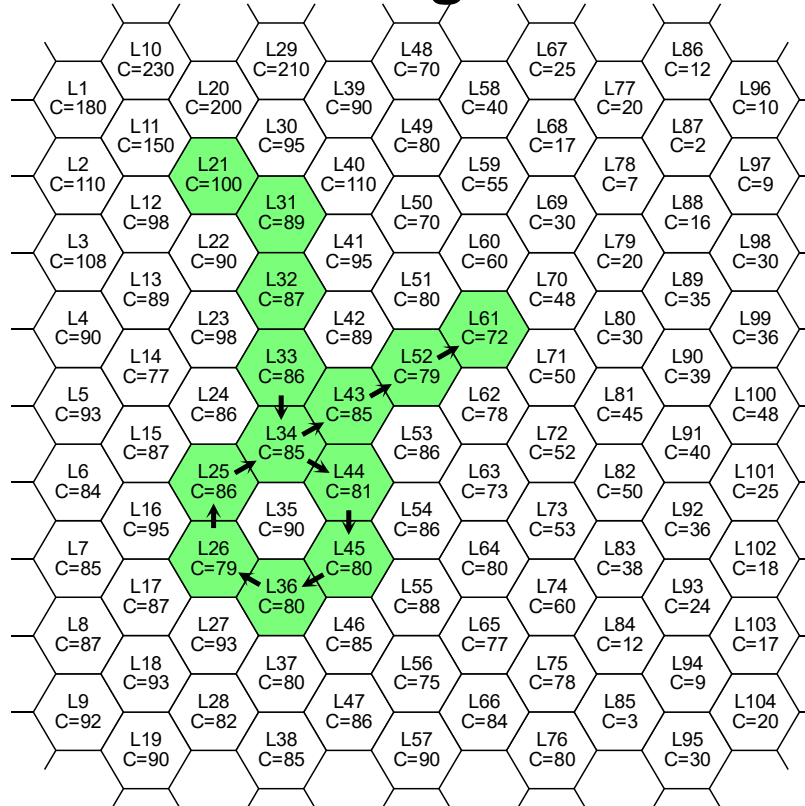
$$f(C, L, P) = k_1 \cdot h_c(C, \bar{C}) + k_2 \cdot h_L(L, \bar{L}) + k_3 \cdot h_p(P, \bar{P})$$


- $C \cong$ Systemkosten in [Euro]
- $L \cong$ Ausführungszeit in [sec] (Latency)
- $P \cong$ Leistungsaufnahme in [W]
- h_C, h_L, h_P geben an, wie stark C, L, P die Entwurfsbedingungen (Constraints $\bar{C}, \bar{L}, \bar{P}$ überstrichen) verletzen.
- $k_1, k_2, k_3 \cong$ Gewichtung und Normierung

4.2 Klassifikation von Partitionierungsalgorithmen (I)



4.4.1 Tabu-Search Algorithmus: Beispiel (I)


 $n_{\text{Tabu_Liste}}=10 \quad k=6$

Stop_Cnd: BS < 5

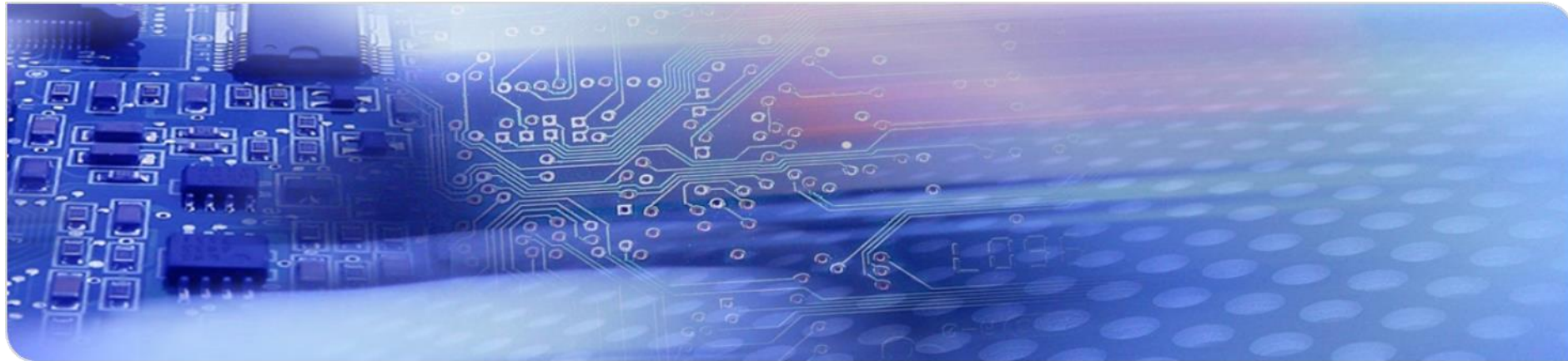
BS:	CS:	Step:
100	100	Start: L21
89	89	L21 \Rightarrow L31
87	87	L31 \Rightarrow L32
86	86	L32 \Rightarrow L33
85	85	L33 \Rightarrow L34
81	81	L34 \Rightarrow L44
80	80	L44 \Rightarrow L45
80	80	L45 \Rightarrow L36
79	79	L36 \Rightarrow L26
79	86	L26 \Rightarrow L25
79	85	L25 \Rightarrow L34
79	85	L34 \Rightarrow L43
79	79	L43 \Rightarrow L52
72	72	L52 \Rightarrow L61

Arbeitsphase

- Aufgabe 3.07: Profiling
 - Kanten-Frequenzproblem
- Aufgabe 4.01: Hierarchical Clustering
 - Verschmelzung von Knoten
- Aufgabe 4.0.2: Tabu-Search

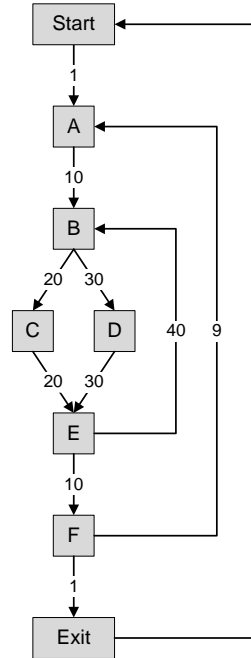
Hardware/Software Co-Design

Übung 7 - Lösung
M.Sc. Fabian Lesniak



Aufgabe 3.07: Profiling

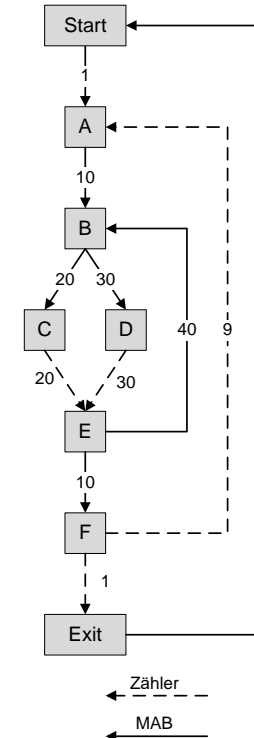
■ Gegeben ist folgender Kontrollflussgraph mit Gewichten:



- Was ist der Unterschied zwischen WCET und Profiling? Welches findet zur Laufzeit bzw. Compilezeit statt?
- Welches Graphenproblem wurde in der Vorlesung bei Profiling vorgestellt und gelöst?
- Bestimmen Sie den maximal aufspannenden Baum (MAB) des Kontrollflussgraphen. Sie können in der Zeichnung die Kanten, die zum MAB gehören, mit einem X markieren.
- Auf welchen Kanten werden jetzt die Zähler für das Profiling platziert? Markieren Sie diese mit einem Kreis.
- Warum wird in diesem Verfahren der Maximale und nicht der Minimale Aufspannende Baum verwendet?
- Wie können die restlichen Zählerwerte aus den platzierten Zählern bestimmt werden?

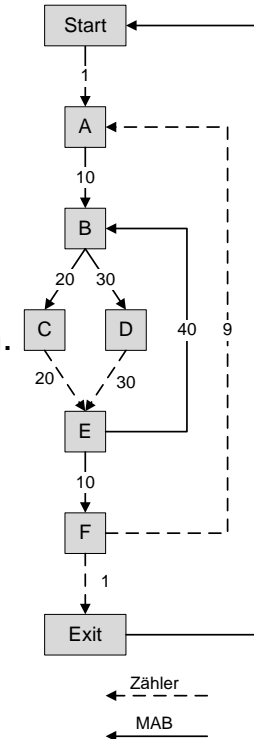
Lösung Aufgabe 3.07: Profiling

- Was ist der Unterschied zwischen WCET und Profiling? Welches findet zur Laufzeit bzw. Compilezeit statt?
 - WCET bestimmt die maximale Ausführungszeit anhand einer statischen Quellcodeanalyse. Profiling hingegen findet zur Laufzeit statt und misst die Verzweigungs- bzw. Ausführungshäufigkeiten jedes Basisblockes.
- Welches Graphenproblem wurde in der Vorlesung bei Profiling vorgestellt und gelöst?
 - Kanten-Frequenzproblem
- Bestimmen Sie den maximal aufspannenden Baum (MAB) des Kontrollflussgraphen. Sie können in der Zeichnung die Kanten, die zum MAB gehören, mit einem X markieren.
- Auf welchen Kanten werden jetzt die Zähler für das Profiling platziert? Markieren Sie diese Kanten.



Lösung Aufgabe 3.07: Profiling

- e) Warum wird in diesem Verfahren der maximal und nicht der minimal aufspannende Baum verwendet?
- Ziel ist das Platzieren der Zähler auf Kanten mit möglichst wenigen Aufrufen, um den Overhead für das Profiling möglichst gering zu halten. Da die Zähler auf den Kanten platziert werden, die vom Aufspannenden Baum nicht verwendet werden, sollten die Kanten des MAB möglichst maximal sein, damit die niedrigen Kanten übrig bleiben.
- f) Wie können die restlichen Zählerwerte aus den platzierten Zählern bestimmt werden?
- Die freien Zählerstellen bilden einen Baum. An den Blättern des Baumes sind alle bis auf einen Zählerwert bekannt und durch die Flussgleichungen kann dieser Zählerwert bestimmt werden. Somit ist es sukzessive möglich sämtliche Werte zu bestimmen.



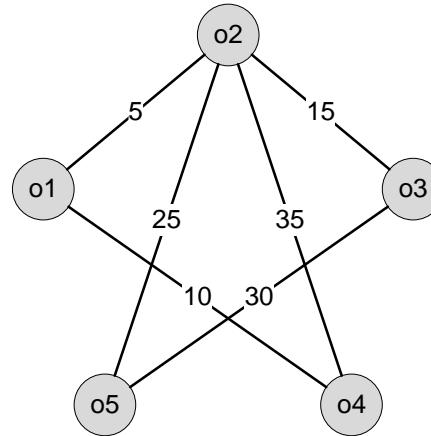
■ Profiling

- Unterschied zu WCET und Tracing?
- Welches Problem wird gelöst?
- Warum wird ein MAB verwendet?
- Welche Eigenschaften hat dieser?
- Wie wird dieser angewendet?
- Was bedeuten die Zahlen an den Kanten?

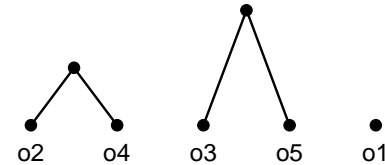
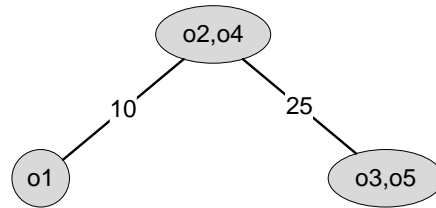
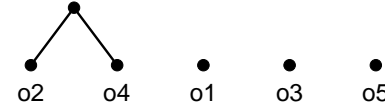
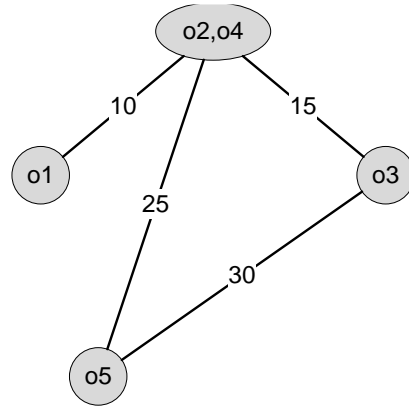


Aufgabe 4.01: Hierarchical Clustering

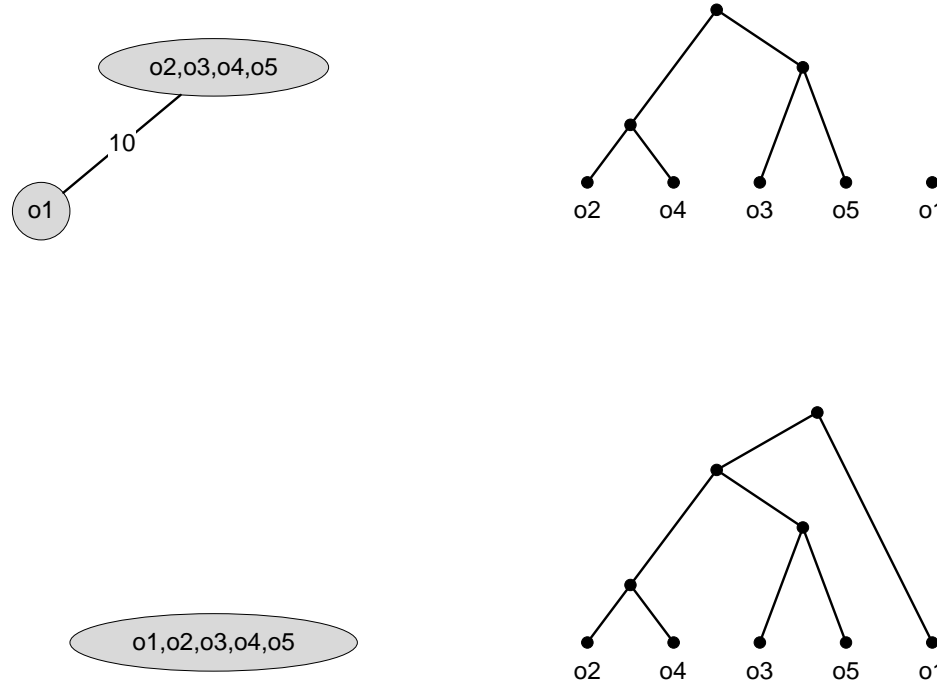
- Führen Sie das Hierarchical Clustering Verfahren anhand des gegebenen Graphen durch. Bei der Verschmelzung der Knoten soll die Maximum-Metrik angewandt werden, d.h. das Kantengewicht einer neuen Kante entspricht dem Maximum der vorherigen Kantengewichte.



Lösung Aufgabe 4.01: Hierarchical Clustering



Lösung Aufgabe 4.01: Hierarchical Clustering



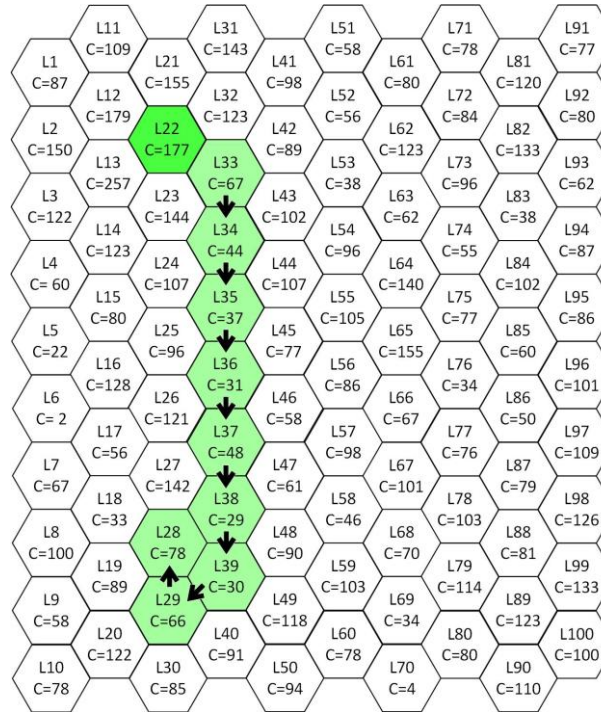
- Hierarchical Clustering
 - In welche Klasse von Verfahren gehört Hierarchical Clustering?
 - Wie werden Knoten verschmolzen?
 - Wie kann eine Bi-Partitionierung durchgeführt werden?



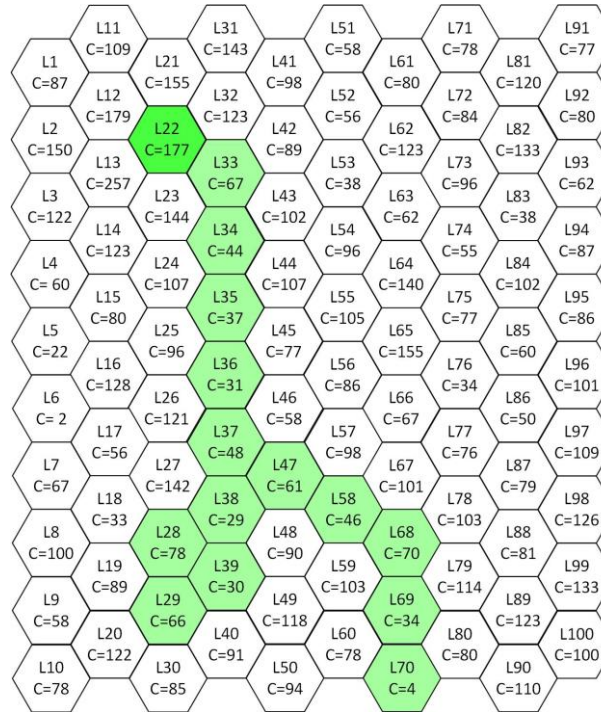
Lösung Aufgabe 4.02: Tabu-Search

- a) Führen Sie den Tabu-Search Algorithmus auf der in Figure 4.4. gegebenen Lösungsmenge jeweils für die zwei unterschiedlichen Startpunkte „L22“ und „L71“ aus. Der Algorithmus soll mit der Stop-Condition $C < 5$ beendet werden und verfügt über eine Tabu-Liste der Länge 8.
- b) Kann das Verfahren Hill-Climbing?
- c) Findet das Verfahren die global beste Lösung?

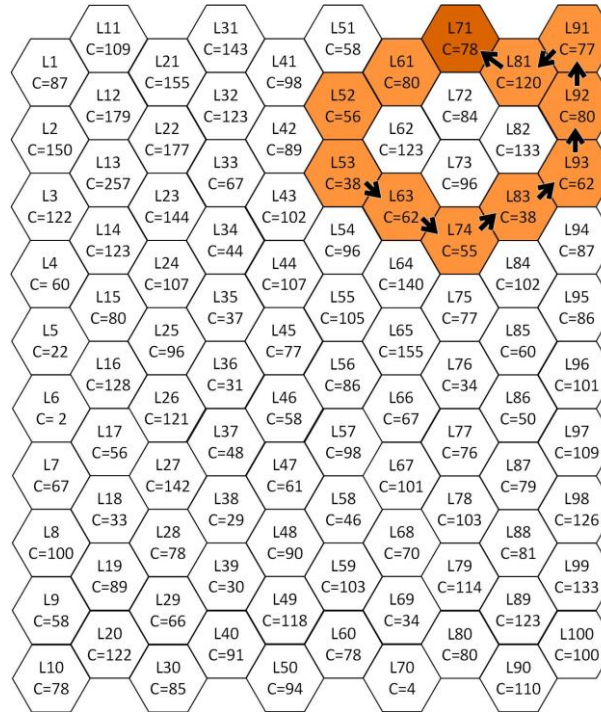
Aufgabe 4.02: Tabu-Search



Aufgabe 4.02: Tabu-Search



Aufgabe 4.02: Tabu-Search



Lösung Aufgabe 4.02: Tabu-Search

- a) Führen Sie den Tabu-Search Algorithmus auf der in Figure 4.4. gegebenen Lösungsmenge jeweils für die zwei unterschiedlichen Startpunkte „L22“ und „L71“ aus. Der Algorithmus soll mit der Stop-Condition $C < 5$ beendet werden und verfügt über eine Tabu-Liste der Länge 8.
- b) Kann das Verfahren Hill-Climbing?
- Grundsätzlich verfügen alle iterative Verfahren über eine Hill-Climbing-Eigenschaft (also die Fähigkeit, auch Verschlechterungen in Kauf zu nehmen um aus lokalen Extrema herauszukommen). In diesem Beispiel wäre dies an mehreren Stellen z.B. bei L36->L37 oder bei L39->L29.
- c) Findet das Verfahren die global beste Lösung?
- Dies ist möglich, es kann jedoch nie garantiert werden. Im vorgestellten Beispiel wird das globale Optimum nicht gefunden (dies liegt bei L6). Da man das globale Optimum aber im Voraus nicht kennt, kann man nur den Algorithmus mit unterschiedlichen Startkonfigurationen neu laufen lassen und dann (üblicherweise anhand der Stop-Condition) einen Wert definieren ab welchem man mit dem Ergebnis zufrieden ist.

- Für was kann TS gut eingesetzt werden?
- Wie geht der Algorithmus vor?
- Was ist die Tabu-Liste?
- Wie wird diese upgedated?
- Welche Komplexität hat der Algorithmus?

