Algorithmen für Chip-Entwurfswerkzeuge

Analytische Placer

TECHNISCHE UNIVERSITÄT DARMSTADT

Vorlesung
WS 2013/2014

Florian Stock, Andreas Koch

Eingebette Systeme und Anwendungen
Technische Universität Darmstadt

Embedded Systems & Applications

Kräftebasierte Placer

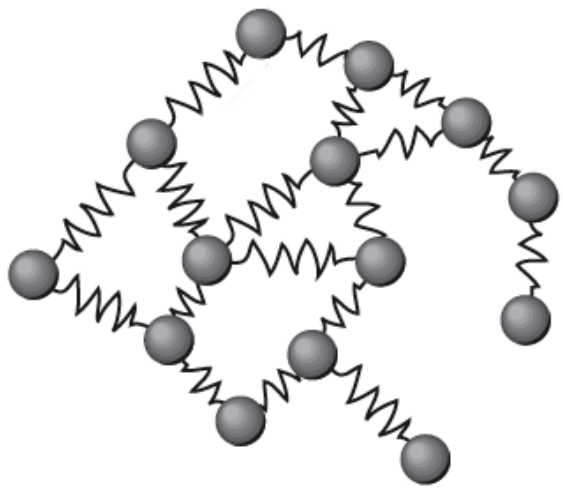
TECHNISCHE UNIVERSITÄT DARMSTADT

- Zielkostenfunktion: Quadratische Verdrahtungslänge

  Vorteil: Im Gegensatz zu HPWL stetig differenzierbar

  Nachteil: Lange Netze gegenüber kurzer übergewichtet

- Entspricht physikalisch einer Anordnung von Objekten die mittels Federn verbunden sind
  $\Rightarrow$ Kräftebasierte (Force-Driven) Platzierung
- Optimum entspricht Kräftegleichgewicht (Überlappung erstmal ignoriert)
- Optimum finden durch lösen eines LGS
  $\Rightarrow$ sehr einfach $\Rightarrow$ sehr beliebt
- Für zusätzliche Ziele zusätzliche Kräfte möglich



[Quelle: Wikimedia]

Kräftebasierte Placer

TECHNISCHE UNIVERSITÄT DARMSTADT

- Überlappungsfreiheit herstellen
  - Erst ignorieren, hinterher legaliserien
    (z.B. Tetris Legalisierung)
  - Zustäzliche Kräfte die für Verteilung wirken hinzufügen
    (Schwerpunkte oder andere explizite Kräfte)
    Iterativ wiederholen bis keine Überlappung
- Implementierungen Kräftebasierter Platzierer:
  - GORDIAN (Kleinhans, 1991)
  - BonnPlace (Brenner, 2005)
  - hATP (Nam, 2006)

Analytische Placer
StarPlace

TECHNISCHE UNIVERSITÄT DARMSTADT

- StarPlace(Xu, Grewal, Areibi 2010)
- Star+-Netzmodel
- Benutzt CG- oder SOR-(Successive Over-Relaxation)-Löser
- Zur Vermeidung von Triviallösungen:
  ShrubPlace (vorplatziert alle I/O-Logik am Außenrand)

StarPlace
Star+

TECHNISCHE UNIVERSITÄT DARMSTADT

- Basierend auf Star-Netzmodell:
  Multipin-Netz wird ersetzt durch fiktiven Knoten und alle Pins verbinden zu diesem
- Fast lineare Funktion
- Schnell: Nach Zügen ein Update in $\mathcal{O}(1)$ möglich
- Schwerpunkt x_g für Netz l mit k_l Blocks
  $x_g := \frac{1}{k_l} \sum_{i \in Netz_l} x_i$
  $||Netz_l||_x := \gamma \sqrt{\sum_{i \in Netz_l} (x_i - x_g)^2 + \phi}$
  ϕ Konstante > 0, so dass es immer diffbar
  γ Faktor zur Kompensation von Unterschätzung
- y-Koordinate analog
- Unterschied zu normalem Star-Modell: $\sqrt{\ldots + \phi}$

StarPlace
Star+ $\mathcal{O}(1)$-Update

TECHNISCHE UNIVERSITÄT DARMSTADT

- Kann auch für SA benutzt werden
- Schnell: Nach Zügen ein Update in $\mathcal{O}(1)$ möglich

$$||Netz_l||_x = \gamma \sqrt{\sum_{i \in Netz_l} (x_i - x_g)^2 + \phi} = \gamma \sqrt{\sum_{i \in Netz_l} x_i^2 - k_l x_g^2 + \phi}$$

$U_l := \sum_{i \in Netz_l} x_i^2$, $V_l := \sum_{i \in Netz_l} x_i = k_l x_g$

$$||Netz_l||_x = \gamma \sqrt{U_l - \frac{V_l^2}{k_l} + \phi}$$

Block $b \in Netz_l$ wird bewegt, dann Update in $\mathcal{O}(1)$

$$U_l^{new} = U_l - x_b^2 + (x_b^{new})^2$$
$$V_l^{new} = V_l - x_b + x_b^{new}$$

StarPlace
Star+ Performance

TECHNISCHE UNIVERSITÄT DARMSTADT

- In VPR: Star+ vs. HPWL
  - Benutzt mit empirisch ermittelten Werten

    $\gamma =$ 1.59
    $\phi =$ 1.0
  - Ergebnisse

Verdrahtbarkeit:	Ähnlich gut
Verdrahtungsressourcen:	Ähnlich gut
	(2.4% besser als VPR-`fast`)
Kritischer Pfad:	$6\% - 9\%$
Laufzeit:	Kaum ein Unterschied

StarPlace
Algorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

```
APlace()
begin
    Convert_Graph_Star+() ;
    PrePlace() ;
    repeat
        CG_Iteration() ;
        Legalize() ;
    until Max Anzahl Iterationen;
```

1. Graph mittels Star+ konvertieren
2. Vorplatzieren mit shrubPlace (I/O am Rand, Blöcke mit viel I/O nahe beieinander)
3. Konjugiertes Gradientenverfahren
4. In jeder Iteration Lösung legalisieren (Partitionsbasierter Ansatz)

StarPlace
Zusammenfassung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Benutzt zur Vorplatzierung shrubPlace $\Rightarrow$
  - 1.5% besseres Ergebnis bei Vedrahtungsressourcen gegenüber zufälliger Vorplatzierung
  - Benötigt 5% der Laufzeit von StarPlace
- Benutzt CG-Löser
  2% besserer kritischer Pfad, 4% mehr Verdrahtungsressourcen
  bei 56% längerer Laufzeit
- Besserer Löser: SOR (Successive Over-Relaxation)
  (CG *schwächelt* bei nichtquadratischen Funktionen)
  9% besserer kritischer Pfad bei 1% mehr Verdrahtungsressourcen
  bei 78% weniger Laufzeit

Analytische Placer
APlace

TECHNISCHE UNIVERSITÄT DARMSTADT

- Khang & Wang, 2005
- Zufällige Startplatzierung
- LSE-Zielkosten
- Zusätzliche Kosten/Kräfte zur Vermeidung von Überlappung
- Top-Down hierarchisches Vorgehen
- CG-Löser

APlace
Kostenfunktion

TECHNISCHE UNIVERSITÄT DARMSTADT

- LSE-Zielkosten
- Verteilung der Zellen (Reduzierung der Überlappung):
  Aufteilung der vorhandenen Fläche in Gitter g

$$DichteStrafe := \sum_{\text{Gitterelemente } g} (\text{GesamtZellFläche}(g) - \text{DurchschnittZellenFläche})^2$$

- Ziel: Möglichst gleichmäßige Verteilung aller Zellen über gesamte Fläche
- Nicht differenzierbar, darum Potenialfunktion

$$Potenial(c, g) := K_c \cdot p(Zelle_x - Gitter_x) \cdot p(Zelle_y - Gitter_y)$$

K_c Normierungsfaktor, so dass
$\sum_g Potenial(c, g) = A_c$ mit A_c Fläche der Zelle c

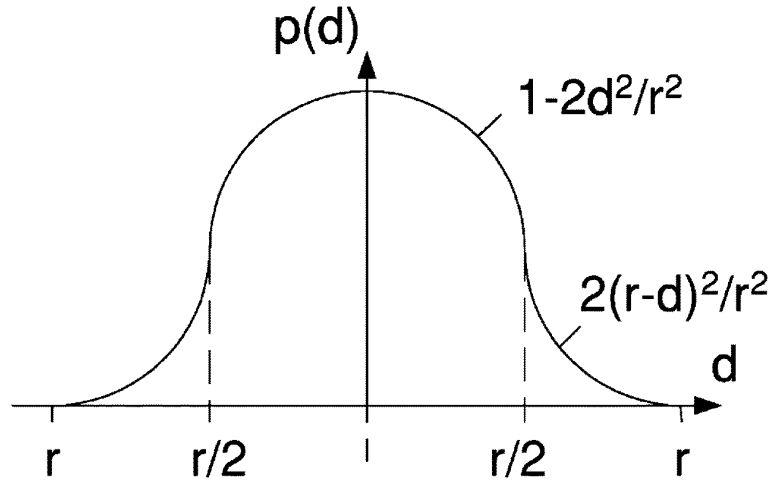
APlace
Glockenfunktion *p*

TECHNISCHE UNIVERSITÄT DARMSTADT

- Naylor et al., 2001
- Glockenfunktion

$$p(d) = \begin{cases} 1 - 2d^2/r^2 & 0 \leq d \leq r/2 \\ 2(d-r)^2/r^2 & r/2 \leq d \leq r \\ 0 & \text{sonst} \end{cases}$$

- r ist der Einflußradius (empirisch $r = 4$)



[Quelle: Kahng]

APlace
Kostenfunktion

TECHNISCHE UNIVERSITÄT DARMSTADT

- Differenzierbare Straffunktion:

$$DichteStrafe := \sum_{\text{Gitterelemente } g} \left(\left(\sum_{\text{Zellen c}} Potenial(c, g) \right) - ExpPotenial(g) \right)^2$$

- Erwartetes Potenial von Gitterelement g:

$$ExpPotenial(g) := \frac{\text{GesamtZellFläche}}{\text{AnzahlGitterelemente}}$$

- Kombination von Verdrahtungslänge und Dichte:

$$Gewicht_{WL} \cdot c_{LSE} + Gewicht_{Dichte} \cdot DichteStrafe$$

APlace
Zusammenfassung

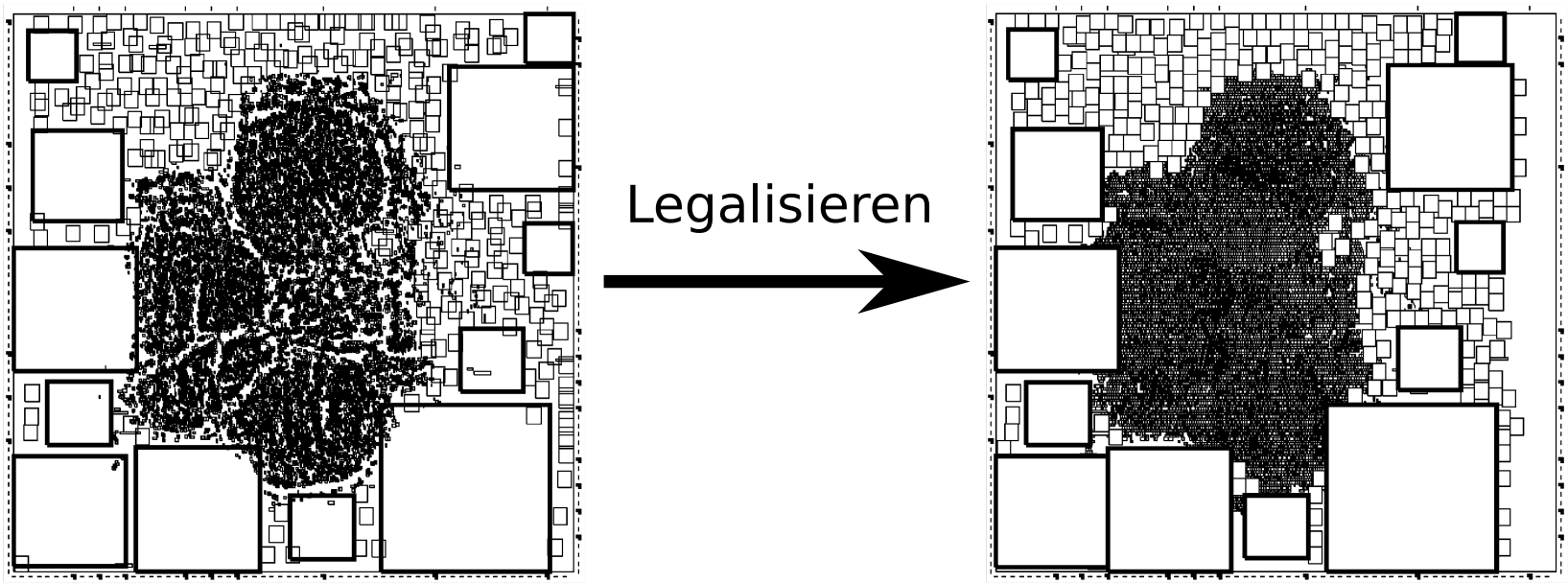
TECHNISCHE UNIVERSITÄT DARMSTADT

- LSE-Kosten
- Dichtefunktion zur Überlappungsreduzierung
- Ergebnisse bis zu 10% besser als bei anderen Placern
- Laufzeit −5% bis +1300% höher
- Varianten:
  - Auch andere Verdrahtungslängenfunktionen
  - Timing-Driven
  - Congestion-Driven
  - Mixed-Size-Design

Legalisierung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Legalisierung von Platzierungen mit Überlappungen
- Weit verbreitet bei analytischen Platzierern
  - Erst Problem ohne Nebenbedingungen optimieren
  - Hinterher gefundene Lösung legalisieren



[Quelle: Kahng]

Legalisierung
Tetris-Algorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

- Hill, 2002 für Standardzellen, verwendet z.B. bei APlace
- Greedy-Algorithmus:
  1. Module in absteigender x-Koordinate sortieren (Breitere bei unentschieden) $\rightarrow$ Abarbeitungsreihenfolge
  2. Für jedes Modul: Mögliche Kandidatenposition, ist die am weitesten links liegende Position in jeder Reihe $\Rightarrow$ die mit kleinstem Kosten (d.h. Abstand) wird Zielposition
- Sehr schnell
- Funktioniert auch gut bei Mixed-Size-Designs
- Varianten:
  - Erleichterte vertikale Bewegung $\Rightarrow$ Ausgeglichenere Reihen
  - Diffusionsbasierter Ansatz
    $\Rightarrow$ Ergebnis näher an Ausgangslösung

Legalisierung
Partionsbasierter Ansatz

TECHNISCHE UNIVERSITÄT DARMSTADT

- Verwendung u.a. bei StarPlace oder GORDIAN
- Rekursiv abwechselnded horizontal und vertikal unterteilen
- Solange bis nur noch eine Zelle in einer Partition
  Diese eine Zelle legalisieren
- Details später bei Partitionierer

Zusammenfassung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Analytische Placer
  - Kräftebasiertes Platzieren
  - StarPlace
  - APlace
- Legalisierung