



# EIN TEILCHENSCHWARM- OPTIMIERTES VERFAHREN ZUR GENERIERUNG VON FEHLERFREIEN DATENPFADEN

Vortrag im Hauptseminar

Markus Mieth – [markus.mieth@mailbox.tu-dresden.de](mailto:markus.mieth@mailbox.tu-dresden.de)

Dresden, 8. Oktober 2015

- 1 Problembeschreibung
- 2 Grundlagen des Lösungsansatzes
- 3 Lösung nach Nach Sengupta und Sedaghat, 2015
- 4 Zusammenfassung

- 1 Problembeschreibung
- 2 Grundlagen des Lösungsansatzes
- 3 Lösung nach Nach Sengupta und Sedaghat, 2015
- 4 Zusammenfassung

# Transient Faults

- plötzlich auftretende und vorübergehende Störungen
- zum Beispiel hervorgerufen durch Strahlung
- Verkleinerung der Architekturbreite steigert Häufigkeit

# Beispiel SDFG

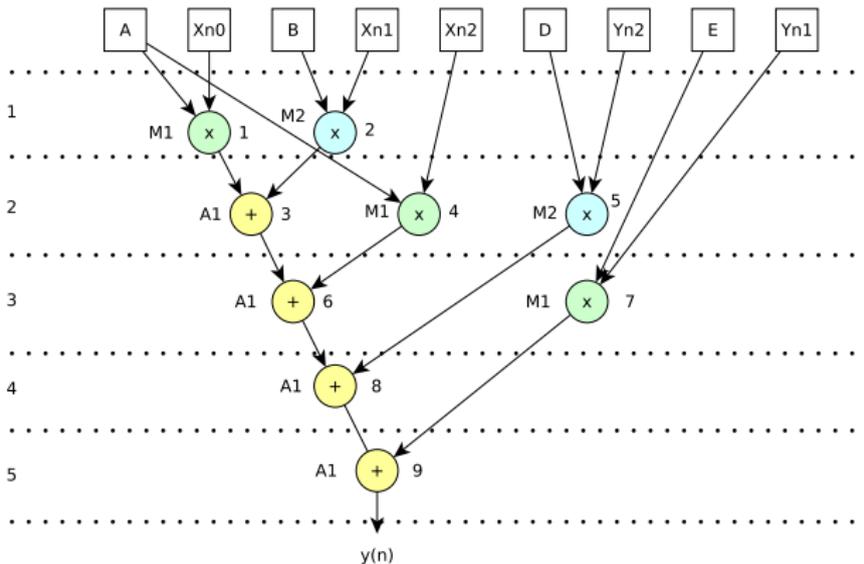


Abbildung: Scheduled Data Flow Graph (SDFG) mit 2M, 1A

- 1 Problembeschreibung
- 2 Grundlagen des Lösungsansatzes**
- 3 Lösung nach Nach Sengupta und Sedaghat, 2015
- 4 Zusammenfassung

# Beispiel DMR SDFG

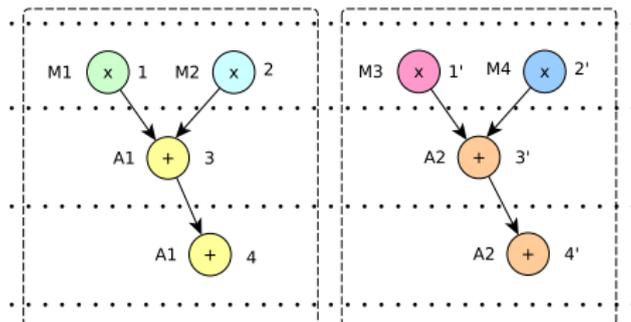


Abbildung: Beispiel Double Modular Redundant (DMR) mit 4M und 2A

# Schwarmintelligenz

- Erste Arbeit dazu von Beni und Wang, 1989
- Beschreibung des Konzepts (nach David W. Corne, 2012):
  - „nützliches Verhalten“ in dem kooperative Effekte in einer Gruppe individueller Agenten auftauchen
  - individuelle Agenten
    - sind homogen
    - handeln asynchron und parallel
    - werden in ihrer Kommunikation durch Stigmergie beeinflusst
  - es herrscht eine kleine bis gar keine Kontrolle
  - das „nützliche Verhalten“ soll relativ einfach sein (z.B. Futtersuche, Nestbau, ...)

# Beispielalgorithmen Schwarmintelligenz

- Ameisenkolonie Optimierung (ACO) (Dorigo, 1992)
- Künstliches Immunsystem (AIS) (Kephart, 1994)
- **Partikel Schwarm Optimierung (PSO)** (Kennedy und Eberhart, 1995)
- Bakterienstamm Optimierung (I. Cohen und Ben-Jacob, 1999)
- Künstlicher Bienenvolk Algorithmus (ABC) (Karaboga, 2005)
- Altruistischer Algorithmus (Waibel, Floreano und Keller, 2011)

# Partikelschwarmoptimierung (PSO)

- Heuristische Suchmethode, die eine Nachahmung der Futtersuche von Vogelschwärmen darstellt
- interne Repräsentation: Population von Partikeln, welche sich durch einen Multidimensionalen Raum bewegen
- jedes Partikel besitzt eine Position und Geschwindigkeit

# Partikelschwarmoptimierung (PSO)

- Position  $x$  (Repräsentiert durch  $n$  Dimensionen) eines Partikels  $i$  zum Zeitpunkt  $t + 1$ :

$$x_i(t + 1) = x_i(t) + v_i(t + 1)$$

- Update-Regel für die Geschwindigkeit  $v$  eines Partikels  $i$ :

$$v_i(t + 1) = \omega v_i(t) + b_1 r_1 (x_i^{lb} - x_i(t)) + b_2 r_2 (x^{gb} - x_i(t))$$

mit:

- $\omega$  ... Massenträgheit
- $b_1$  ... kognitive Lernfaktor
- $b_2$  ... soziale Lernfaktor
- $r_1, r_2$  ... zufällige Zahlen im Bereich  $[0, 1]$
- $x_i^{lb}$  ... beste Position des  $i$ -ten Partikels im Zusammenhang mit dem Minimierungsproblem
- $x^{gb}$  ... zurzeit global beste gefundene Position

- 1 Problembeschreibung
- 2 Grundlagen des Lösungsansatzes
- 3 Lösung nach Nach Sengupta und Sedaghat, 2015**
- 4 Zusammenfassung

# Problemdefinition

Gegeben: ein Kontrollflussdatengraph (engl. *Control Data Flow Graph* (CDFG))

Gesucht: optimale Ressourcenmenge  $R_x = \{N(R_1), N(R_2), \dots, N(R_d) \dots N(R_D)\}$

- mit minimalen hybriden Kosten  $\text{Cost}(P_T^{\text{DMR}}, T_E^{\text{DMR}})$
- mit  $P_T^{\text{DMR}} \leq P_{\text{cons}}$ ,  $T_E^{\text{DMR}} \leq T_{\text{cons}}$  und  $k_c$  als Fehler einschränkung (multi-cycle (k-cycle) transient fault)

# Evaluationsmodell I

i) Energie Modell:

$$P_T^{\text{DMR}} = P_S^{\text{DMR}} + P_D^{\text{DMR}} \text{ mit}$$
$$P_S^{\text{DMR}} = \sum_{d=1}^D N(R_d) \cdot K(R_d) \cdot p_c \quad \text{und} \quad P_D^{\text{DMR}} = \frac{E_{FU}^{\text{DMR}}}{T_E^{\text{DMR}}}$$

$N(R_d)$ ... Anzahl an Ressourceninstanzen  $R$  vom Typ  $d$

$K(R_d)$ ... Fläche an belegten Ressourcen  $R$  vom Typ  $i$

$E_{FU}^{\text{DMR}}$  ... gesamter Energieverbrauch der Ressourcen in DMR

$P_S^{\text{DMR}}$  ... statischer Leistungsverbrauch der Ressourcen in DMR

$P_D^{\text{DMR}}$  ... dynamischer Leistungsverbrauch der Ressourcen in DMR

$P_T^{\text{DMR}}$  ... gesamter Leistungsverbrauch der Ressourcen in DMR

$T_E^{\text{DMR}}$  ... gesamte Ausführungszeit eines DMR Systems

# Evaluationsmodell

ii) Ausführungszeit:

$$T_E^{\text{DMR}} = \sum_{c.s=1}^n \text{Max}(D(\text{op}_i), \dots, D(\text{op}_n), D(\text{op}_{i'}), \dots, D(\text{op}_{n'}))$$

mit  $1 \leq i \leq n$  und  $1' \leq i' \leq n'$

$D(\text{op}_n)$ ... Verzögerung von Operation  $n$

$i, i'$ ... ohne Kennzeichnung „'“ aus dem Originalen, mit aus der Kopie

c.s... Kontrollschritt

# Evaluationsmodell

iii) Fitnessfunktion:

$$C_f(x_i) = \varphi_1 \frac{P_T^{\text{DMR}} - P_{\text{cons}}}{P_{\text{max}}^{\text{DMR}}} + \varphi_2 \frac{T_E^{\text{DMR}} - T_{\text{cons}}}{T_{\text{max}}^{\text{DMR}}}$$

$C_f(x_i)$  ... Kosten eines Partikels mit Ressourcenmenge  $x_i$

$P_{\text{cons}}$  ... Benutzerspezifizierter Leistungsverbrauch

$T_{\text{cons}}$  ... Benutzerspezifizierte Ausführungszeit

$T_{\text{max}}^{\text{DMR}}$  ... maximale Ausführungszeit eines DMR-Systems in dem Designraum

$P_{\text{max}}^{\text{DMR}}$  ... maximale Leistung eines DMR-Systems in dem Designraum

# Evaluationsmodell

i) Energie-Modell:

$$P_T^{\text{DMR}} = P_S^{\text{DMR}} + P_D^{\text{DMR}}$$

ii) Ausführungszeit:

$$T_E^{\text{DMR}} = \sum_{c.s=1}^n \text{Max}(D(\text{op}_i), \dots, D(\text{op}_n), D(\text{op}'_i), \dots, D(\text{op}'_n))$$

iii) Fitnessfunktion:

$$C_f(x_i) = \varphi_1 \frac{P_T^{\text{DMR}} - P_{\text{cons}}}{P_{\text{max}}^{\text{DMR}}} + \varphi_2 \frac{T_E^{\text{DMR}} - T_{\text{cons}}}{T_{\text{max}}^{\text{DMR}}}$$

# PSO DSE Framework

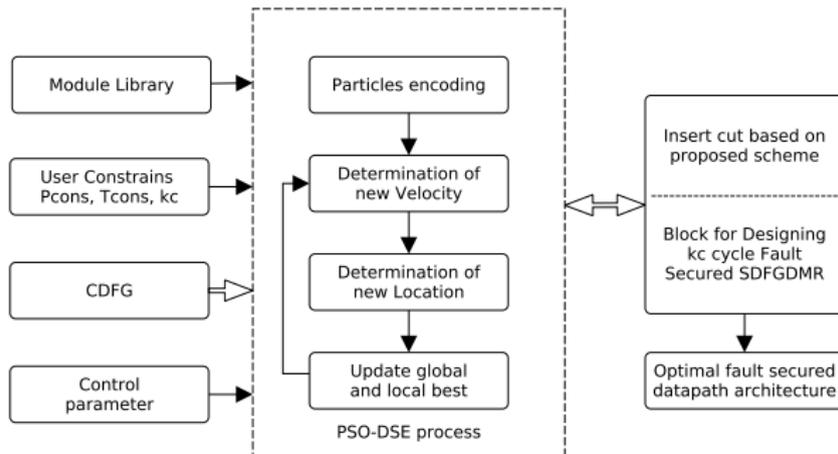


Abbildung: Nach Sengupta und Sedaghat, 2015

# Particle Encoding

- Position von Partikel  $X_i = (N(R_1), N(R_2), \dots, N(R_d), \dots, N(R_D))$
- $X_1 = (\min(R_1), \min(R_2), \dots, \min(R_D))$
- $X_2 = (\max(R_1), \max(R_2), \dots, \max(R_D))$
- $X_3 = (\frac{\min(R_1)+\max(R_1)}{2}, \frac{\min(R_2)+\max(R_2)}{2}, \dots, \frac{\min(R_D)+\max(R_D)}{2})$
- Rest:  $N(R_d) = \text{random}(\min(R_d), \max(R_d))$  (Gleichverteilt)

$N(R_d) \dots$  Anzahl an Instanzen vom Typ  $d$

$\min, \max \dots$  minimale. bzw. maximale Anzahl ( $\in \mathbb{N}$ ) an Instanzen vom Typ  $d$

# Initialwerte

- Geschwindigkeit  $v_1 = 0$
- Kognitiver Lernfaktor  $b_1$  und Sozialer Lernfaktor  $b_2$ : zufällige Werte mit  $b_1, b_2 \in [1, 2]$
- Gewichtungsfaktor  $\omega \in [0.1, 0.9]$

# Berechnung neuer Positionen und Geschwindigkeiten

- neue Geschwindigkeit:

$$V_{d_i}^+ = \omega V_{d_i} + b_1 r_1 [R_{d_{lbi}} - R_{d_i}] + b_2 r_2 [R_{d_{gbl}} - R_{d_i}]$$

- neue Position:

$$R_{d_i}^+ = f(V_{d_i}^+, R_{d_i}) = R_{d_i} + V_{d_i}^+$$

# Lokale und Beste Position

globale beste Position mit minimalen Kosten aller Partikel ( $X_1, \dots, X_n$ ):

$$X_{gb} = X_i [\text{Min}(C_{f_{1b1}}^{X_1}, C_{f_{1b2}}^{X_2}, C_{f_{1b3}}^{X_3}, \dots, C_{f_{1bn}}^{X_n})]$$

$C_{1bi}^{X_i}$  ... lokale beste Bewertung (Anhand der Fitnessfunktion) des Partikels  $X_i$

# Beispiel SDFGDMR

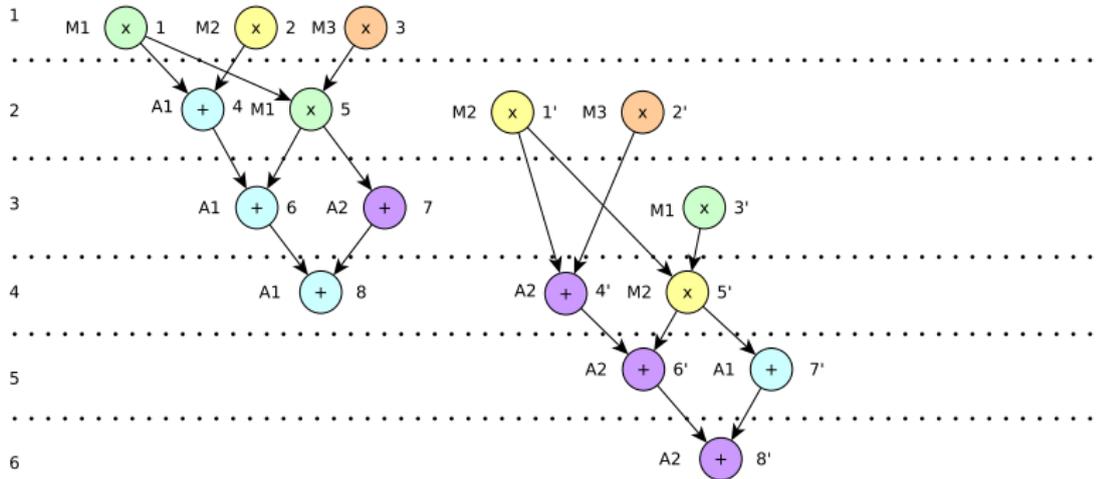


Abbildung: SDFGDMR mit  $X_i = (3M, 2A)$  und  $k_C \geq 1$

- 1 Problembeschreibung
- 2 Grundlagen des Lösungsansatzes
- 3 Lösung nach Nach Sengupta und Sedaghat, 2015
- 4 Zusammenfassung**

# Zusammenfassung

- Einführung in Transient Faults mit üblicher Lösung DMR
- Einführung in Particle Swarm Optimization
- Vorzeigen einer Lösung für DSE aus beiden
- laut Benchmarks mit Infinite Impulse Response (IIR), Berkeley Paket Filter (BPF), MPEG MV u.a. haben fast alle getesteten, propagierten Designs mit den k-cycle transient fault secured datapath bessere Werte in der Fitnessfunktion

# Quellen I

-  Beni, G. und J. Wang (1989). "Swarm Intelligence in Cellular Robotic Systems". In: *Proceed. NATO Advanced Workshop on Robots and Biological Systems*. Tuscany, Italy.
-  David W. Corne Alan P. Reynolds, Eric Bonabeau (2012). "Swarm Intelligence". In: *Handbook of Natural Computing*. Springer Berlin Heidelberg.
-  Dorigo, M. (1992). "Optimization, Learning and Natural Algorithms". Diss. Politecnico di Milano.
-  I. Cohen I. Golding, Y. Kozlovsky und E. Ben-Jacob (1999). "Continuous and discrete models of cooperation in complex bacterial colonies". In: S. 235–247.
-  Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization*. Techn. Ber. Erciyes University, Engineering Faculty, Computer Engineering Department.

# Quellen II

-  Kennedy, J. und R. Eberhart (1995). "Particle swarm optimization". In: *Neural Networks, 1995. Proceedings., IEEE International Conference on*. Bd. 4, 1942–1948 vol.4.
-  Kephart, Jeffrey O. (1994). "A Biologically Inspired Immune System for Computers". In: *In Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*. MIT Press, S. 130–139.
-  Sengupta, Anirban und Reza Sedaghat (2015). "Swarm intelligence driven design space exploration of optimal k-cycle transient fault secured datapath during high level synthesis based on user power–delay budget". In: *Microelectronics Reliability* 55.6, S. 990–1004. ISSN: 0026-2714. URL: <http://www.sciencedirect.com/science/article/pii/S0026271415000608>.

# Quellen III

-  Waibel, Markus, Dario Floreano und Laurent Keller (2011). "A Quantitative Test of Hamilton's Rule for the Evolution of Altruism". In: *PLoS Biol* 9.5, e1000615. URL: <http://dx.doi.org/10.1371/journal.pbio.1000615>.

Vielen Dank für die  
Aufmerksamkeit.