

Sorteernetwerken van Optimale Grootte*

Mathias Dekempeneer, Vincent Derkinderen

Bachelor Informatica

Katholieke Universiteit Leuven

voornaam.achternaam@student.kuleuven.be

Abstract

Korte samenvatting van wat we doen en wat de conclusie is.

Verder werken op paper van Codish et al. Sorteert optimal size sorting network.

Tijdsverbetering van x?

Optimale grootte, nul één principe, huidige tabel van resultaten optimale grootte (en diepte), formule voor ondergrens

n	6	7	8	9	10	11	12
bovengrens	12	16	19	25	29	35	39
ondergrens	12	16	19	25	29	33	37

Tabel 1: Minimaal aantal comparatoren bij n kanalen.

1 Introductie

Situering + bijdrage.

Sorting Network (high level), Optimal Size (high level), contributies andere papers rond deze twee, enkele getallen rond grootte orde van het probleem, wat er al geprobeerd is (SAT, generate & prune,...), hoe wij het probleem zullen aanpakken (hoe wij prunen (high level)), gebruikte hardware...

2 Probleemstelling

Een *comparator network* C_k^n bestaat uit n kanalen en k *comparatoren*. Een comparator (i, j) verbindt twee verschillende kanalen i en j waarbij $0 < i < j \leq n$. We nemen x_l^m als waarde op kanaal m net voor comparator l . De l^{de} comparator vergelijkt de huidige waarden van beide kanalen en plaatst de kleinste waarde op kanaal i en de grootste waarde op kanaal j zodat $x_{l+1}^i = \min(x_l^i, x_l^j)$ en $x_{l+1}^j = \max(x_l^i, x_l^j)$. De uitvoer van een comparator netwerk verwijst naar de partieel geordende vector $\vec{x} = \{x_{k+1}^1 \dots x_{k+1}^n\}$.

Een *sorteernetwerk* is een comparator netwerk met als eigenschap dat de uitvoer gesorteerd is ongeacht de invoer.

Een sorteernetwerk C_k^n van optimale grootte houdt in dat er geen ander sorteernetwerk C_l^n bestaat waarbij $l < k$. Om te onderzoeken of een comparator netwerk een sorteernetwerk is, kunnen we gebruik maken van het *nul - één principe*.

HIER KOMT NOG EEN FOTO VAN DE WERKING VAN EEN SORTEERNETWERK.

Definities + basisuitleg + evaluatiecriteria

Comparator netwerk, sorteernetwerk

3 Voorgestelde oplossing

Ontwerp, wat (algoritme)

Generate & prune (en hoe we dit doen) + de getallen hier rond (zoals aantal comparatoren). Het prune idee uitleggen. Benadruk de slechte grootte orde en de nood aan snellere beslissingen om de uitkomst van de prune check op voorhand te weten.

3.1 Representatie van sorteernetwerken

Bijgehouden informatie van netwerken

3.2 Genereren

Uitleg hoe we de generate doen.

Redundant (of de comparator die je toevoegt, wel iets verandert? Uitleggen wat wij specifiek doen), unique_if uitleggen

3.3 Snoeien

Uitleg hoe we de prune doen.

Checken van alle netwerken met alle netwerken voor de prune stap.

- Aantal 1en bij $C_a > C_b \Rightarrow C_a$ subsumes niet C_b
- $|W(C_a, x, k)| > |W(C_b, x, k)| \Rightarrow C_a$ subsumes niet C_b
- Uitleggen reductie van permutaties

3.4 Parallellisatie

Parallellisatie uitleggen

Uitleg hoe generate and prune verandert door elke thread zijn stuk te laten generate en prunen en vervolgens in een groter geheel te prunen en hoe dit groter geheel prunen werkt zonder locks.

*Dankwoord

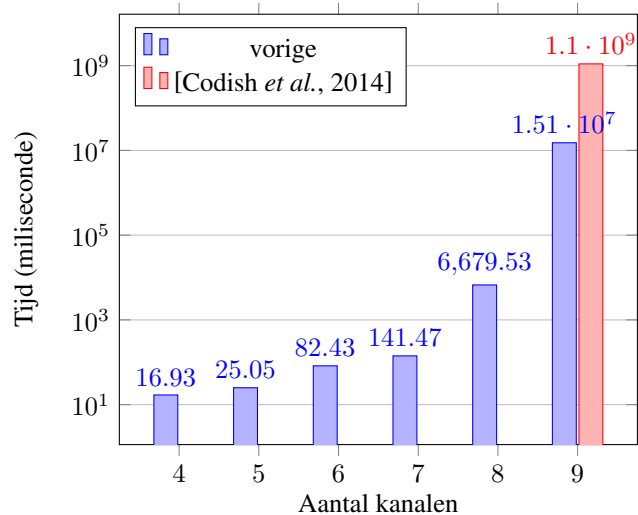
4 Evaluatie

Empirische evaluaties + grafiekjes

Tabel geven van hoeveel beslissingen er op welke plaats genomen worden.

Vergelijken runtime voor 9 kanalen met Codish.

Schatting runtime voor 10 kanalen.



5 Conclusies

Conclusie[Codish et al., 2014]

Conclusie van wat er bereikt is en hoe er verder aan gewerkt kan worden.[Codish et al., 2015]

Erkenning

De rekeninfrastructuur en dienstverlening gebruikt in dit werk, werd voorzien door het VSC (Vlaams Supercomputer Centrum), gefinancierd door het FWO en de Vlaamse regering - departement EWI

Referenties

[Codish et al., 2014] Michael Codish, Luis Cruz-Filipe, Michael Frank, and Peter Schneider-Kamp. Twenty-five comparators is optimal when sorting nine inputs (and twenty-nine for ten). Technical report, IEEE International Conference on Tools with Artificial Intelligence (ICTAI), November 2014.

[Codish et al., 2015] Michael Codish, Luis Cruz-Filipe, and Peter Schneider-Kamp. Sorting networks: the end game. In *Proceedings of the 9th International Conference on Language and Automata Theory and Applications, LATA, LNCS*, 2015.

[Knuth, 1973] D. E. Knuth. *The art of computer programming. Vol.3: Sorting and searching*. 1973.