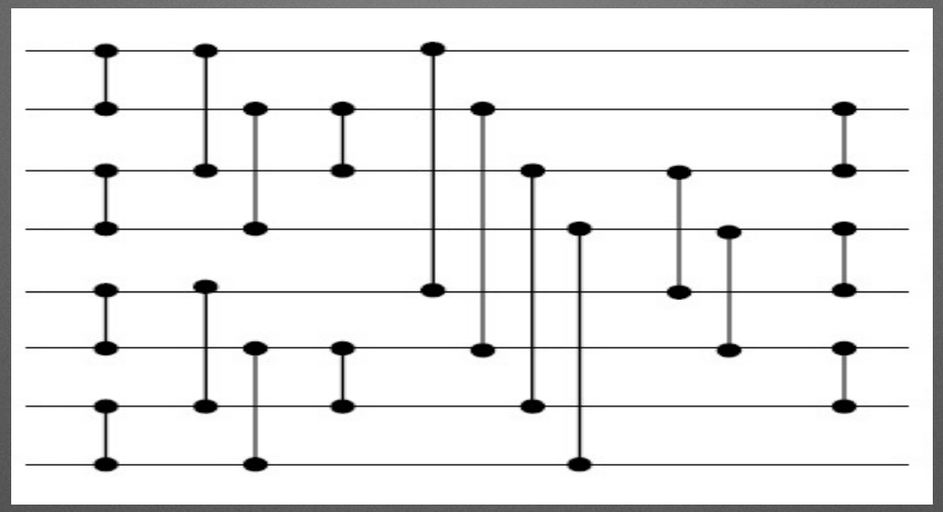
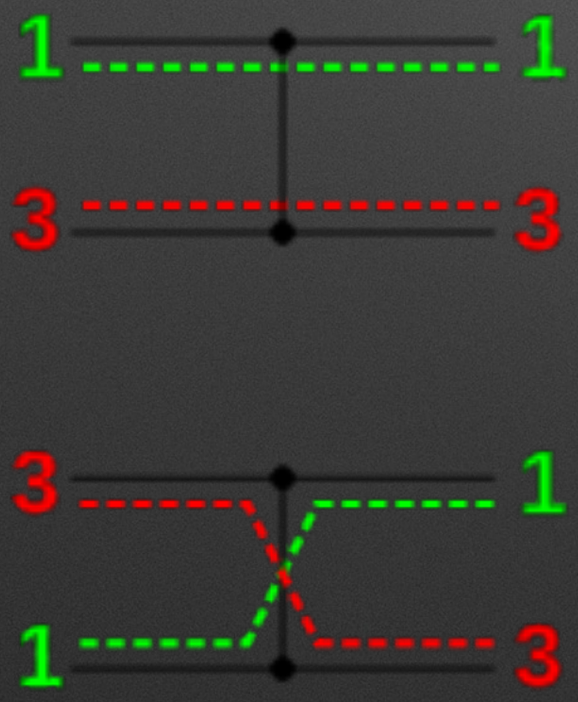
Sorteernetwerken van Optimale Grootte

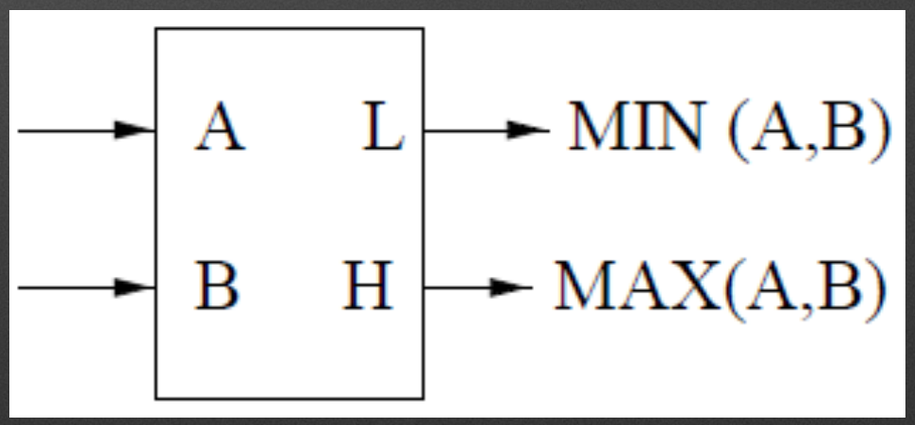
Mathias Dekempeneer
Vincent Derkinderen

Begeleider: Tom Schrijvers

Comparator Netwerk

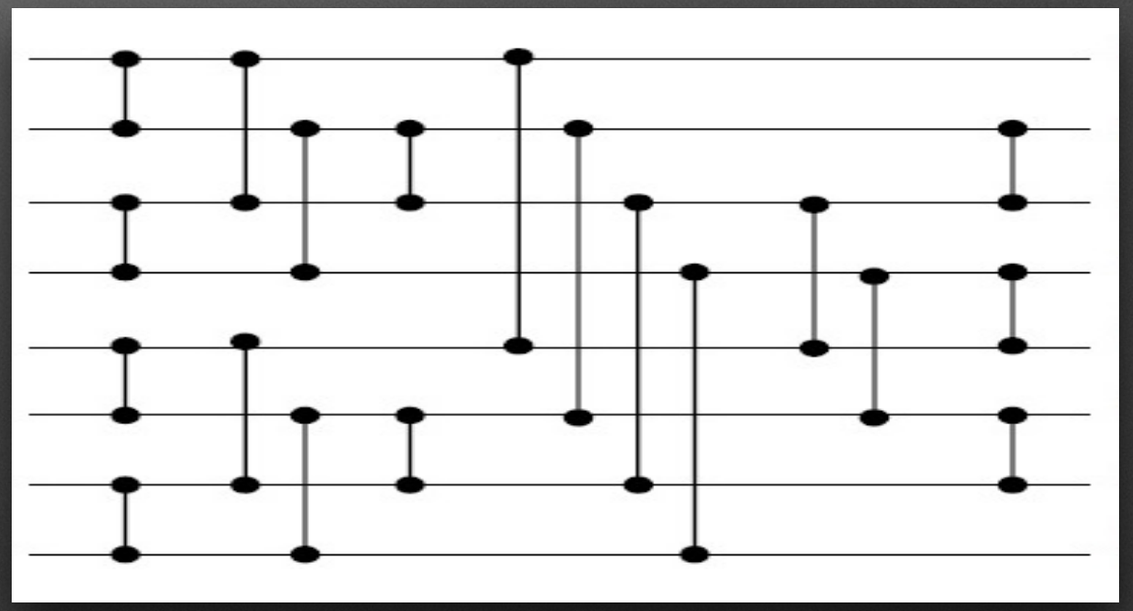




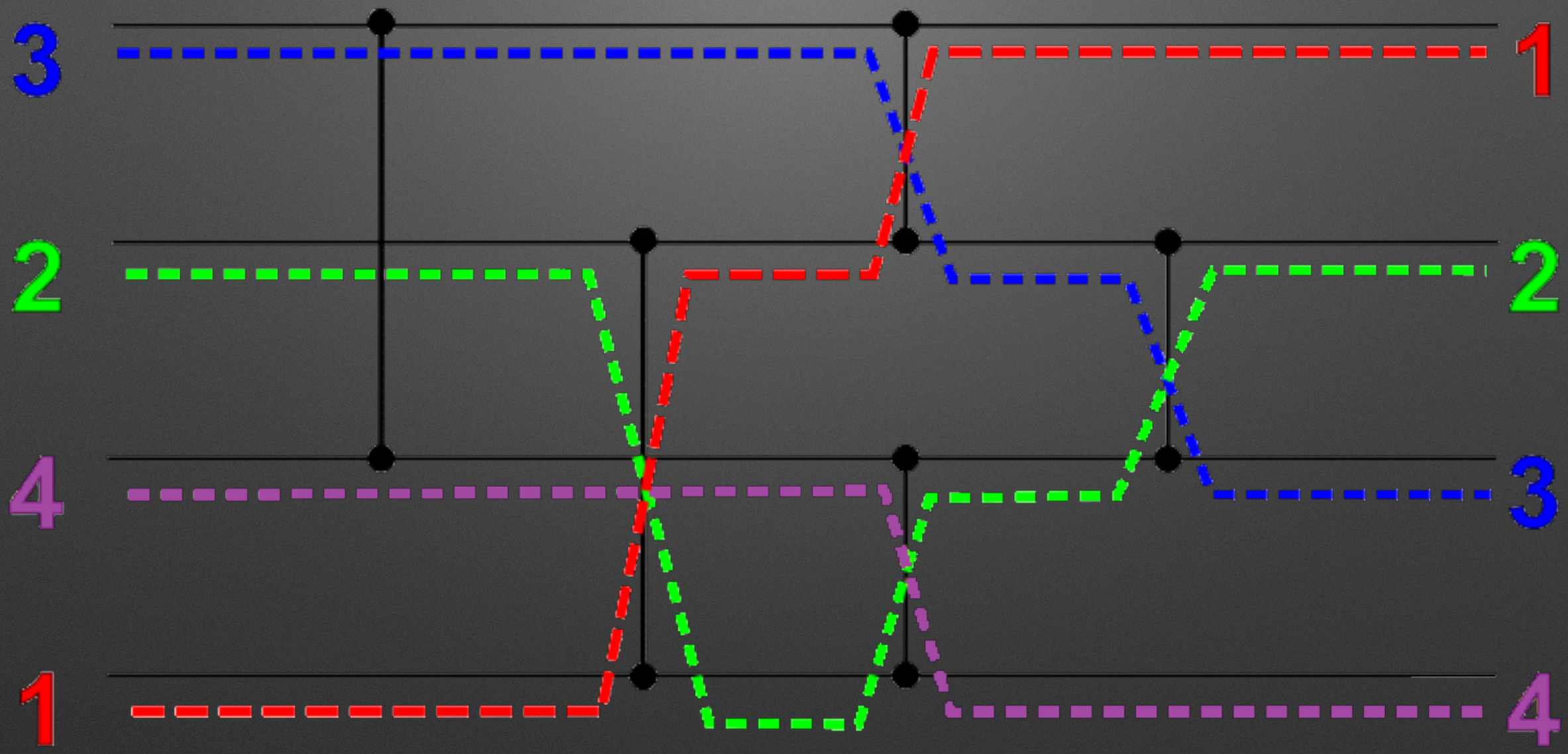


Comparator Netwerk

- Representatie comparatoren
  (1 2) (3 4) (5 6) (7 8) (1 3) (5 7)
  …

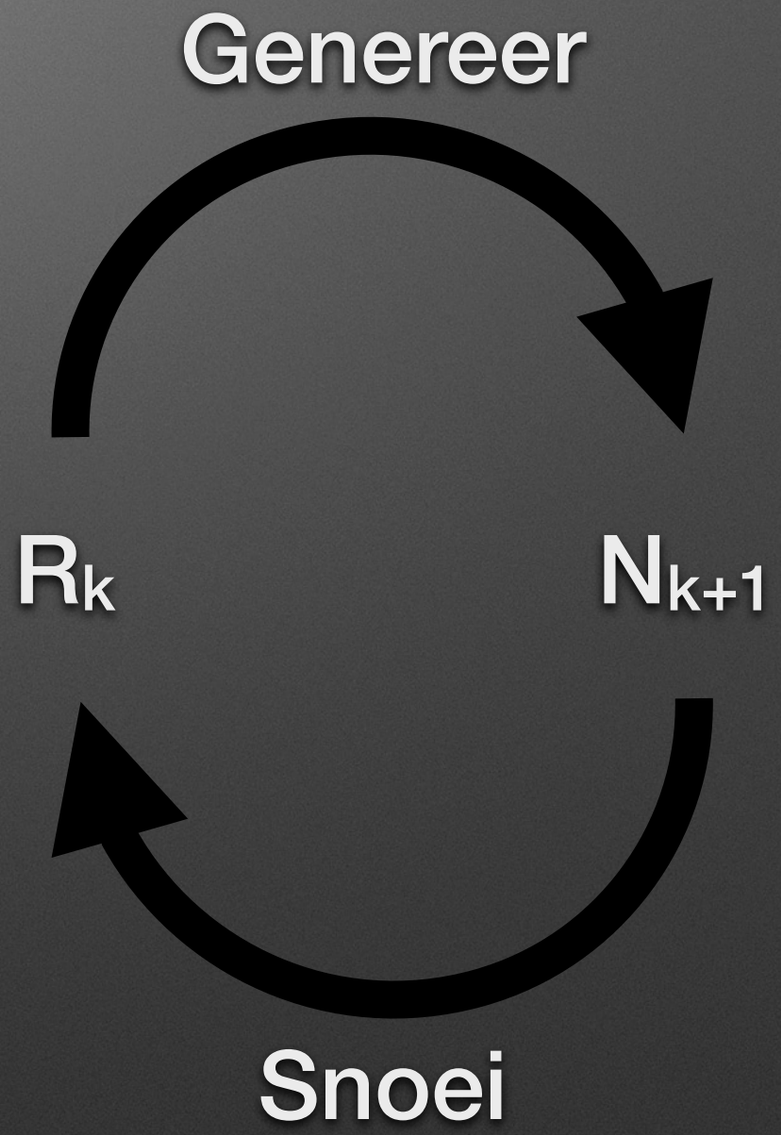


Sorteernetwerk



Genereer & Snoei

- Genereer:
  toevoegen alle mogelijke comparatoren
- Snoei:
  subsumes principe



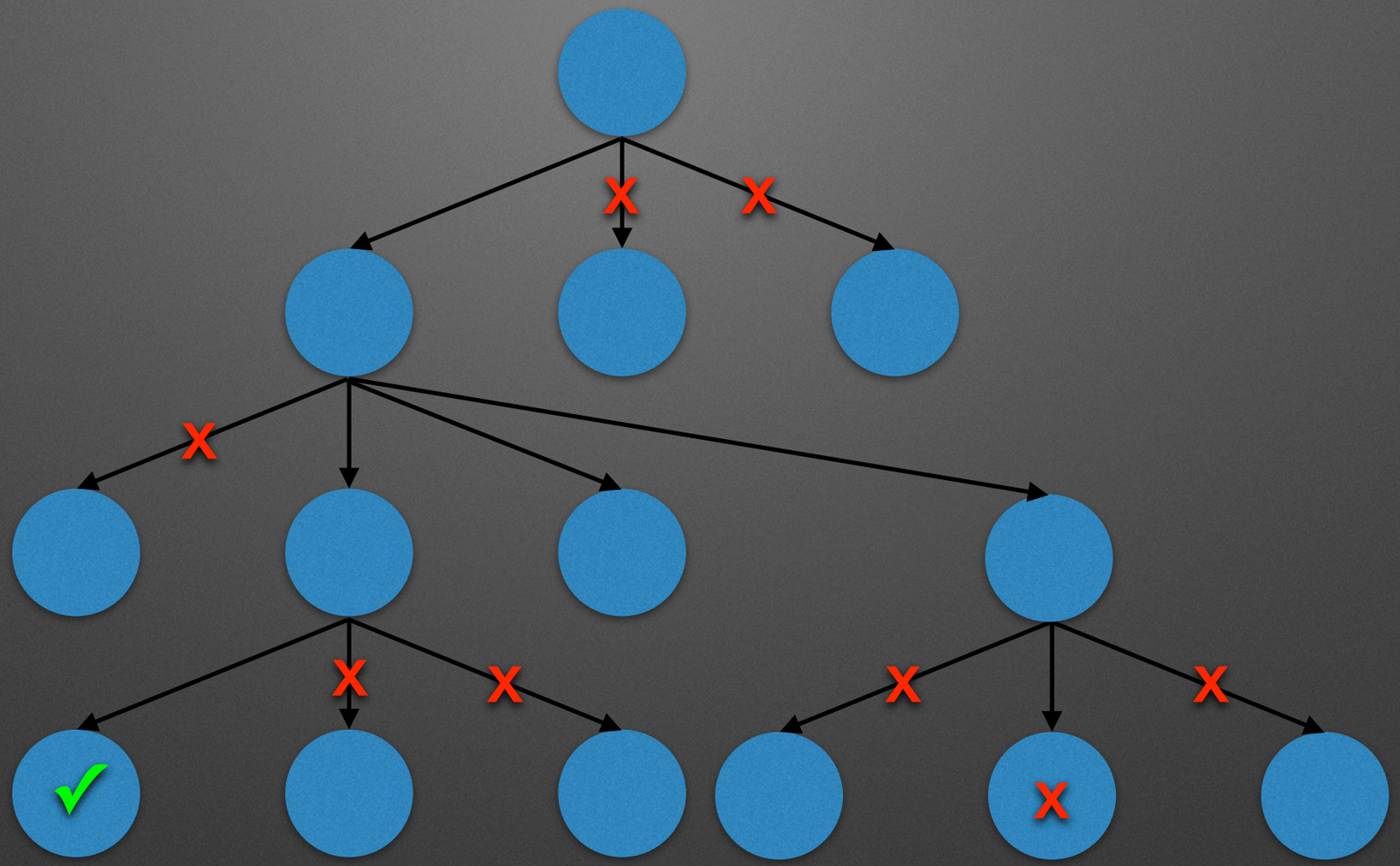
Genereer & Snoei

R_0

$N_1 - R_1$

$N_2 - R_2$

N_3



Subsumes

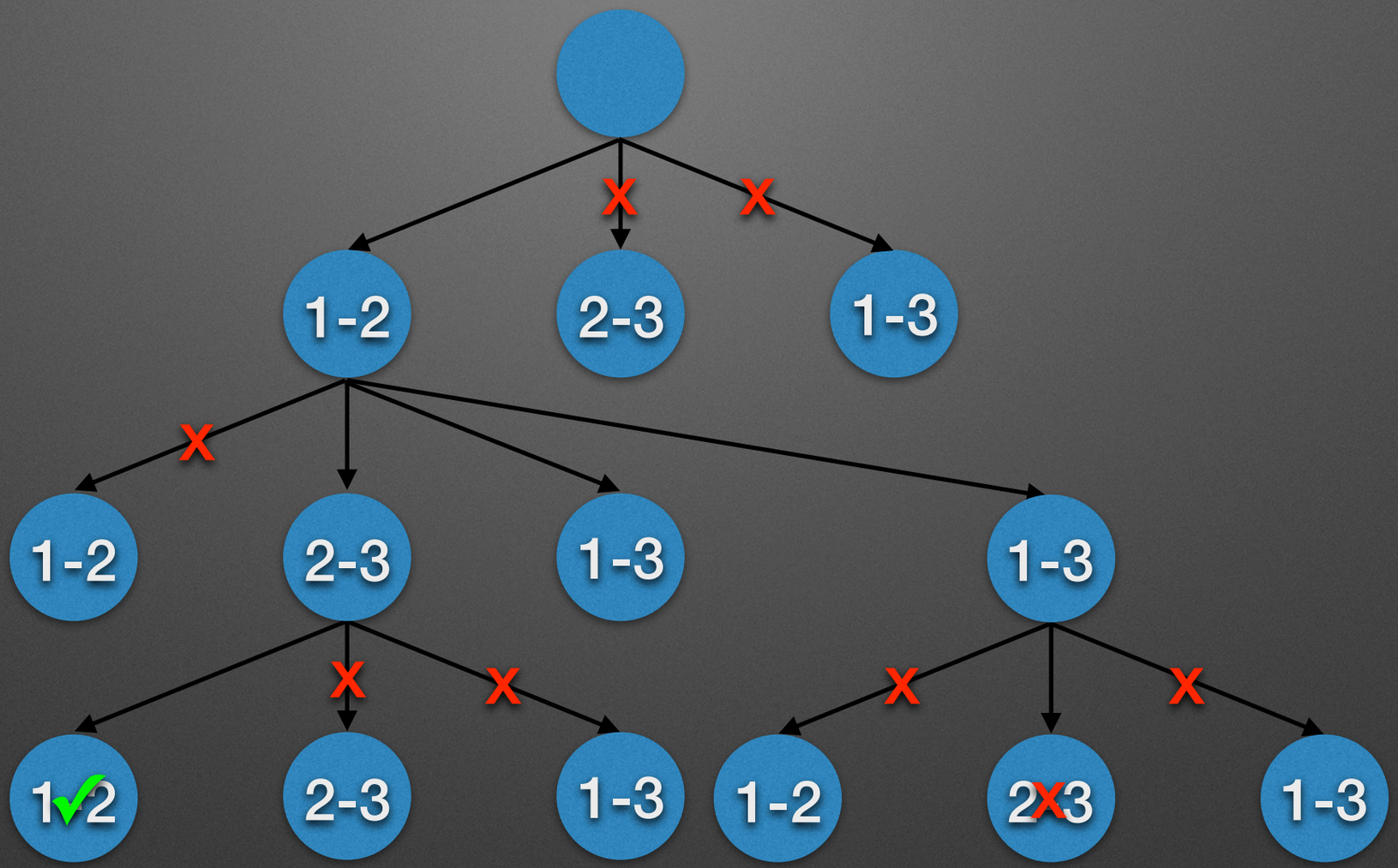
- Beschreven in "Twenty-Five Comparators is Optimal when Sorting Nine Inputs (and Twenty-Nine for Ten)" *(Codish et al.)*

- C_a subsumes $C_b \Leftrightarrow C_a$ wordt gedekt door C_b

  $\Leftrightarrow \pi(\text{Outputs}(C_a)) \subseteq \text{Outputs}(C_b)$

- Verwijder de netwerken die anderen dekken

Genereer & Snoei



Gevonden sorteernetwerk: (1-2) (2-3) (1-2)

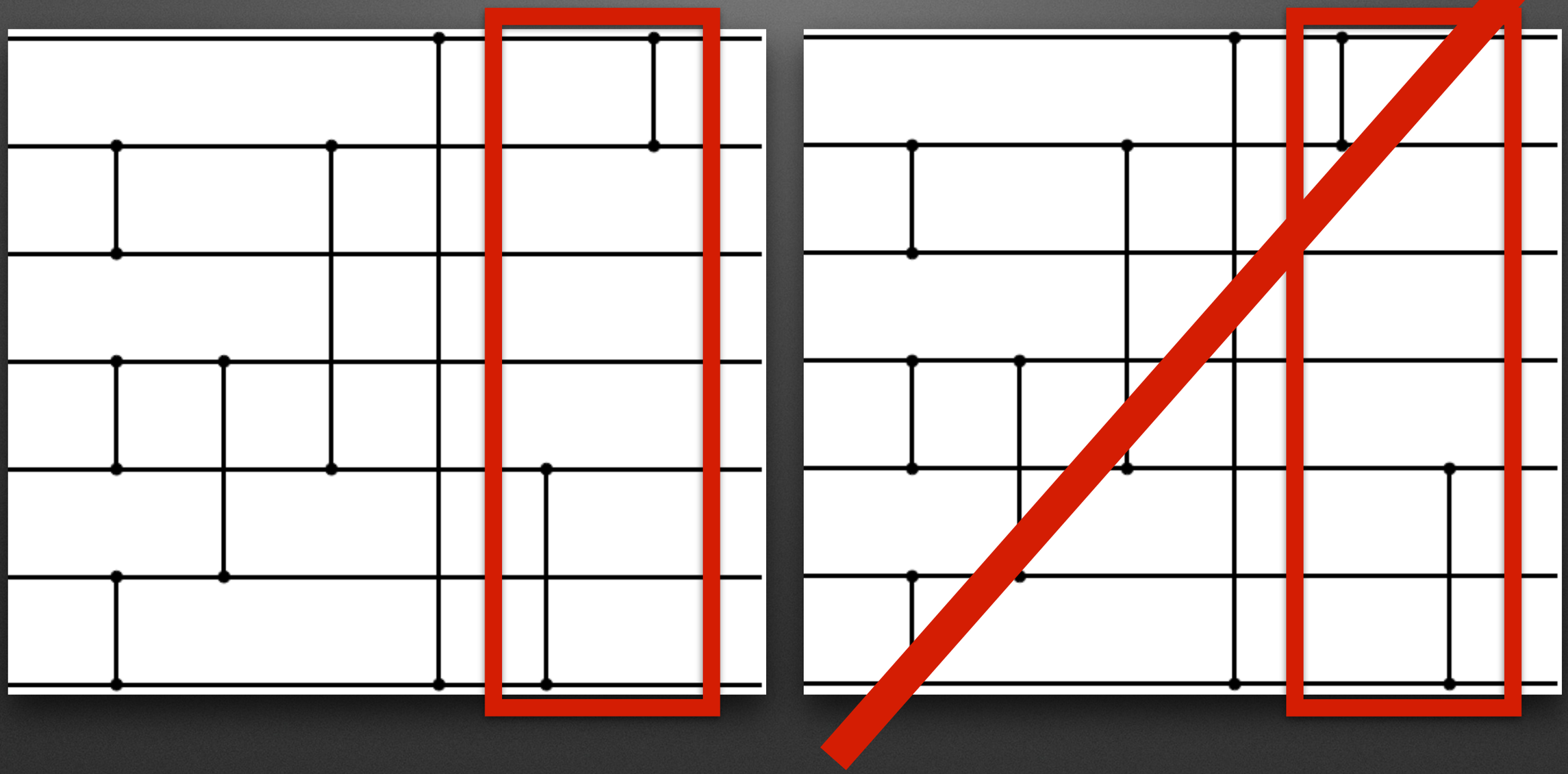
Genereer & Snoei

- Bottleneck: beslissing subsumes
  ⇒ methoden om sneller te beslissen

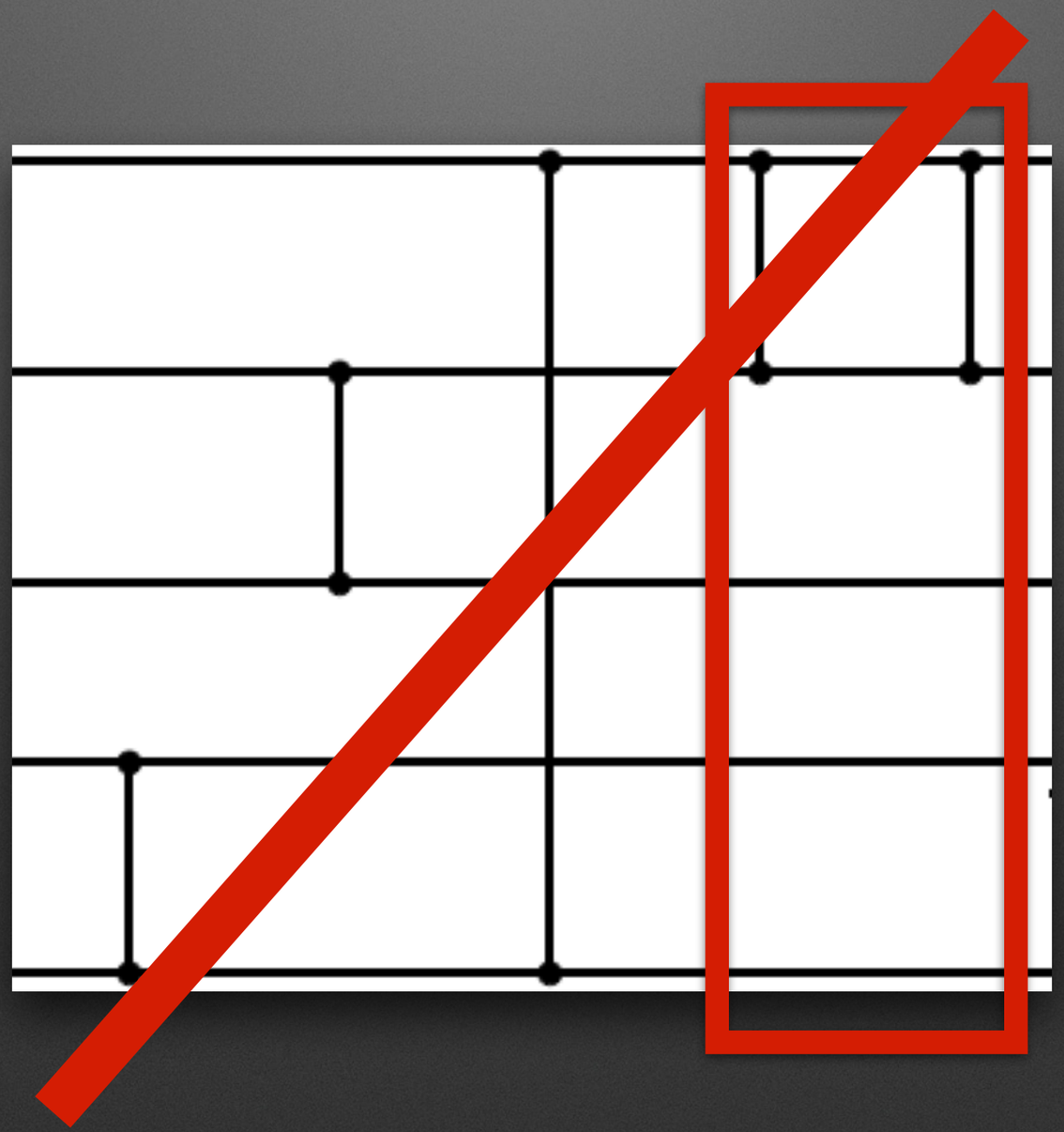
- Genereer (uniek, redundant)

- Snoei (kLengte, pLengte, lLengte …)

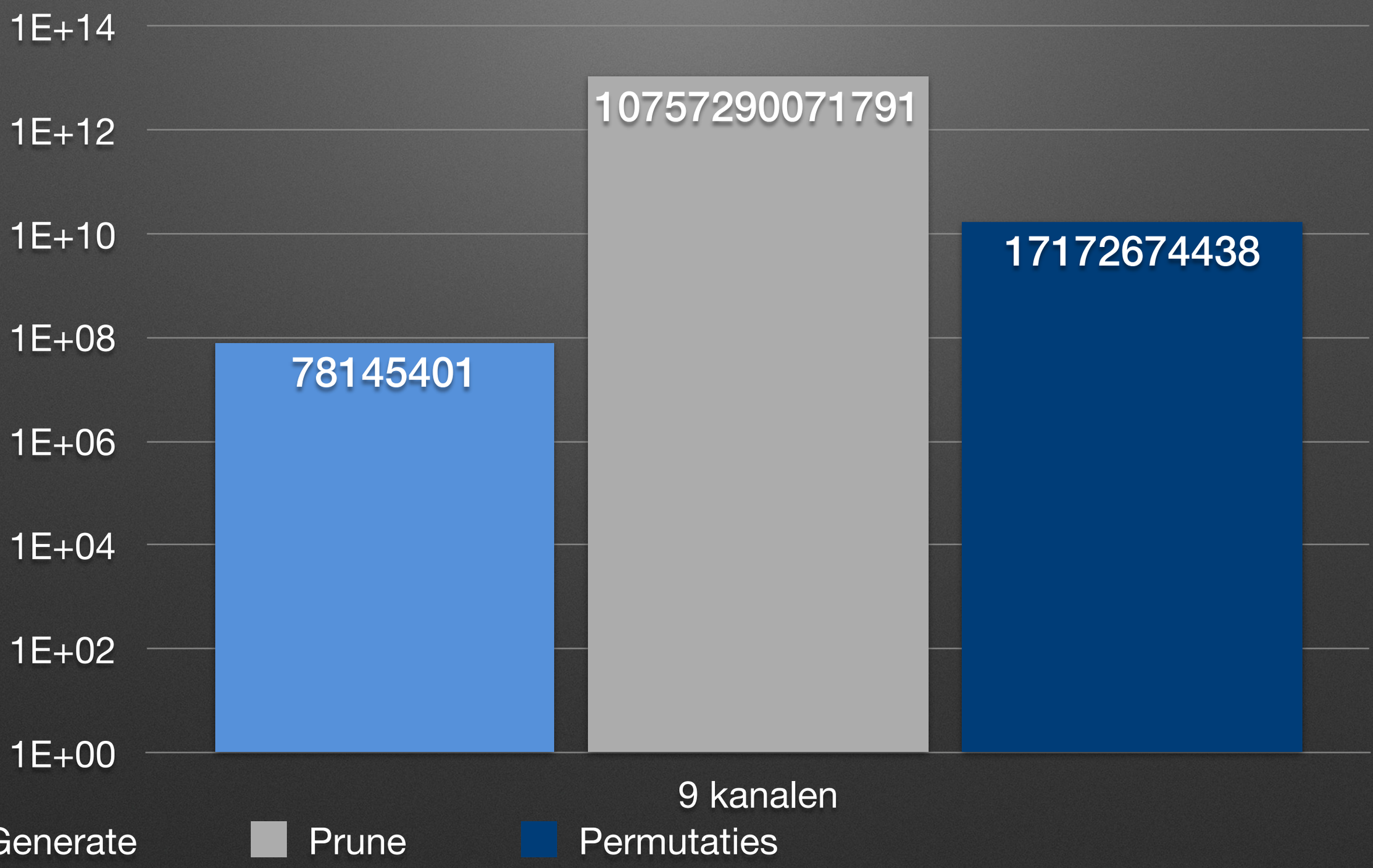
Methode uniek

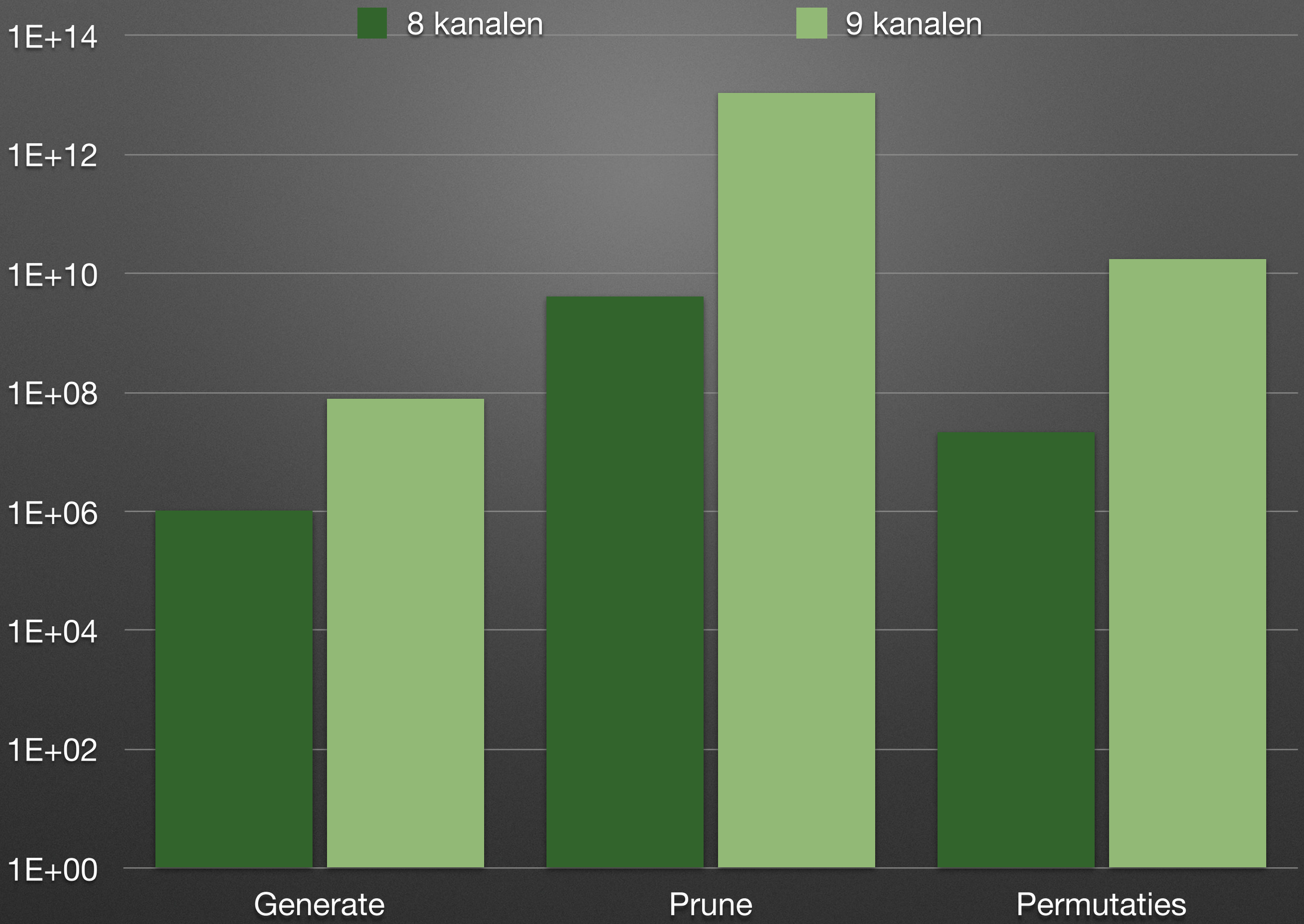


Methode redundant

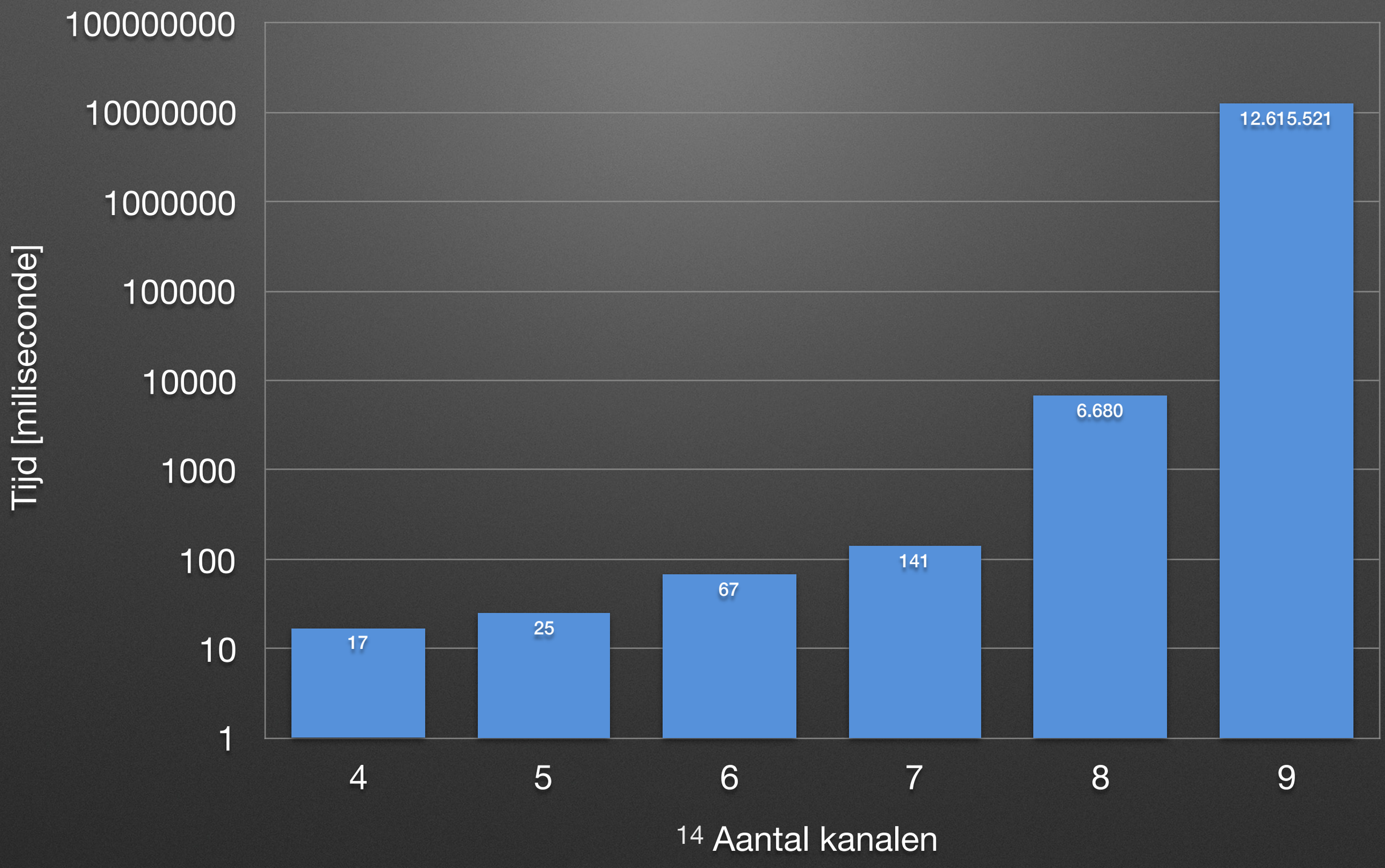


Beslissingen

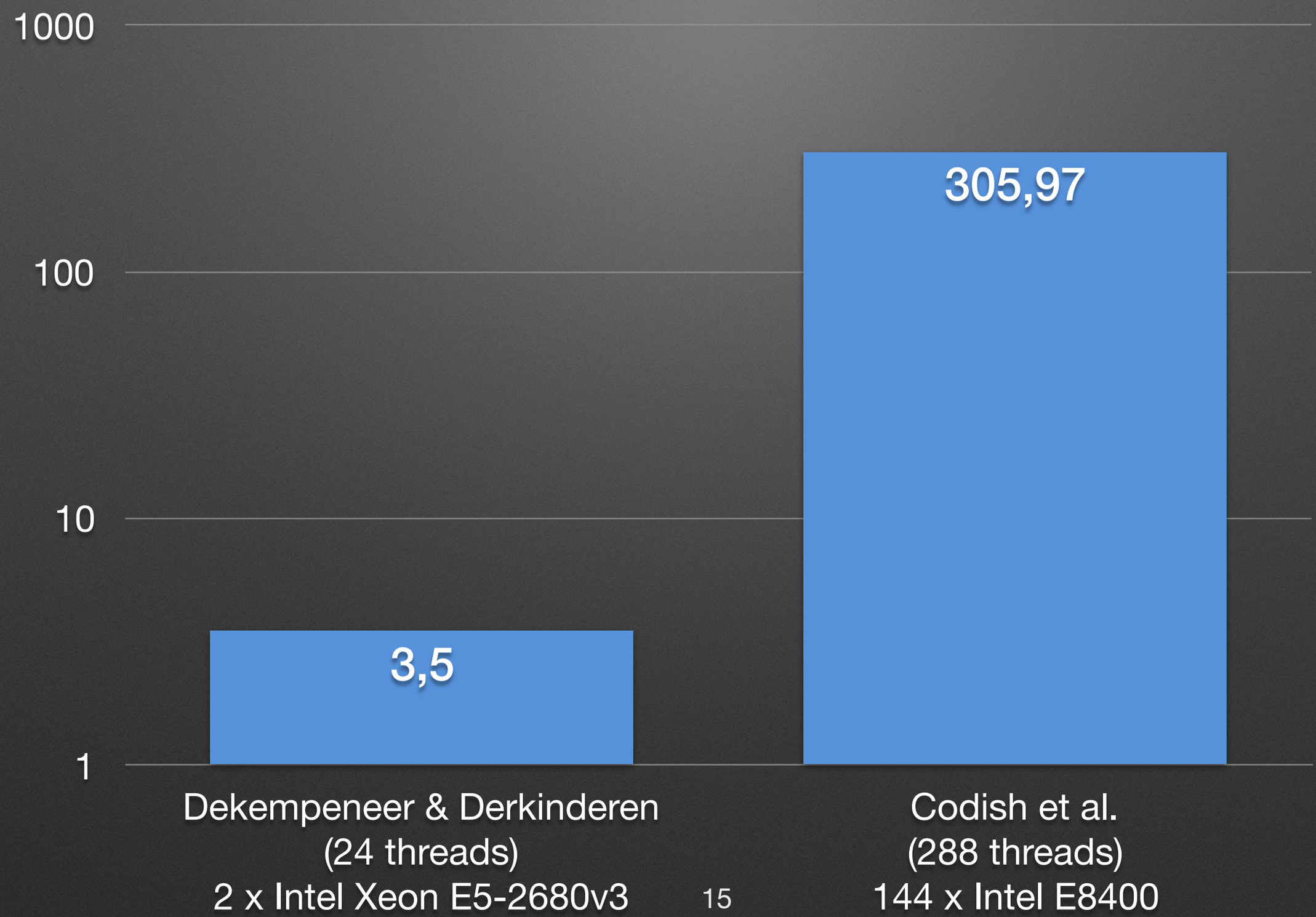



8 kanalen
9 kanalen
1E+14
1E+12
1E+10
1E+08
1E+06
1E+04
1E+02
1E+00
Generate
Prune
Permutaties

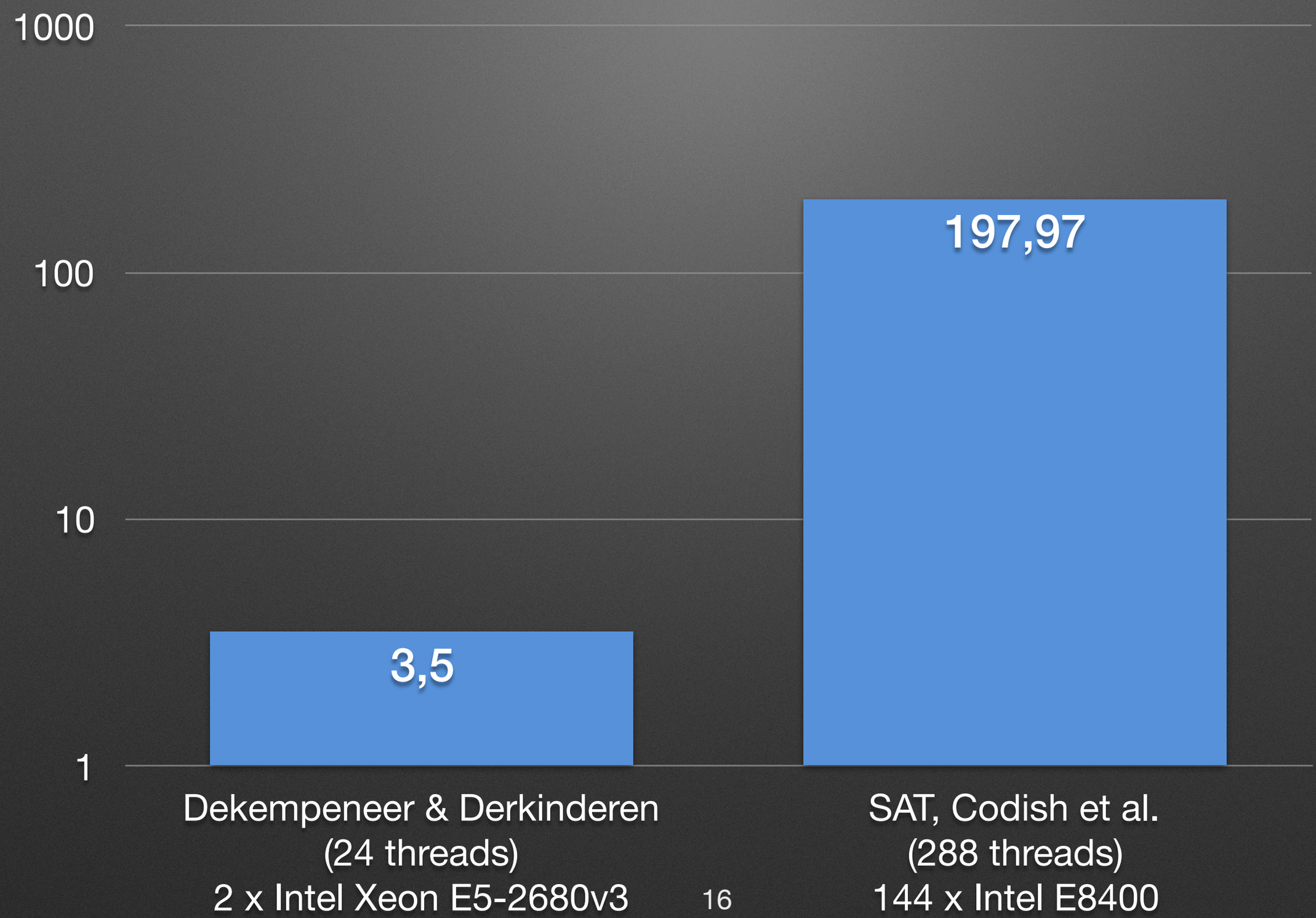
Resultaten


100000000
10000000
1000000
100000
10000
1000
100
10
1
Tijd [milisecon
onde]
17
25
67
141
6.680
12.615.521
4
5
6
7
8
9
14 Aantal kanalen

Resultaten


1000
100
10
1
Tijd [uur]
305,97
3,5
Dekempeneer & Derkinderen
(24 threads)
2 x Intel Xeon E5-2680v3
Codish et al.
(288 threads)
144 x Intel E8400

Resultaten


1000
100
10
1
Tijd [uur]
197,97
3,5
Dekempeneer & Derkinderen
(24 threads)
2 x Intel Xeon E5-2680v3
SAT, Codish et al.
(288 threads)
144 x Intel E8400

Conclusie

WAT?
Resultaten van de paper gereproduceerd

HOE?
Implementatie van paper
Verder bouwen op paper

Conclusie

WAAROM?
Bewijzen / vinden van efficiënt netwerk

WAT VOLGT?
Bekijken reden van verbetering
Implementatie voor meerdere nodes
Verbeteringen voor het algoritme zoeken