Generering og komprimering af slutspilstabeller i skak med hurtig tilgang v.h.a. OBDD'er

Følgende punkter vil blive berørt:

- Introduktion — målsætningen opridses.
- Generering af slutspilstabeller (kortfattet).
- Indekserings skemaer (kortfattet).
- Representation v.h.a. OBDD'er.
- Diverse optimeringer — vigtigste er tildeling af *don't cares*
- Resultater — sammenligning med alternative repræsentationer.
- Optimal tildeling af don't cares er NP-hårdt — selv at approksimere.
- Beviset herfor generaliserer tidligere resultater af Sauerhoff og Wegener og Hirata, Shimozono og Shinohara.

Introduktion

Type

- En tabel for hver kombination af brikker og hver spiller til at trække.
- ATS : Afstand Til Skakmat. Værdien uafgjort samt . . . , -M1, -M0, M1, M2, . . .
- VTU : Vundet/Tabt/Uafgjort

Målsætning — opslagstid.

- Opslag skal kunne foretages uden at nedsætte søgehastigheden nævneværdigt.
- D.v.s. i omegnen af $\frac{1}{2}\mu s$ på en 3 GHz P4, som er den tid et state-of-the-art skak program bruger per stilling.

Målsætning — pladsforbrug. Sammenligning med

- Nalimov: Mest udbredte format. 3-6 brikkers slutspil. Kun ATS
- EgmProbe: 3-5 brikkers slutspil. Kun VTU.
- Run length encoding: Benyttet af *Chinook*. Kun VTU.

Målsætning — understøttelse

- 3-4 brikkers slutspil : ATS
- 5 brikkers slutspil : VTU. Enkelte svære slutspil: ATS

Generering af slutspilstabeller

Hukommelses ekstern optimering ikke anvendt. 2 GB ram nok for $\leq$ 5 briks slutspil.

Fremadrettet konstruktion

- $+$ Simpel. Almindelig trækgenerator kan anvendes.
- $\div$ Langsom. Gentagne analyser af skakstillingerne indtil en stilling med højeste matdybde er fundet.

Bagudrettet konstruktion

- $+$ Hurtig. Lineær i størrelsen af spilgrafen.
- $\div$ Mere kompliceret. Kræver entydig repræsentation af stillinger. Pladsforbrug 2 bytes per stilling.

Sammenligning

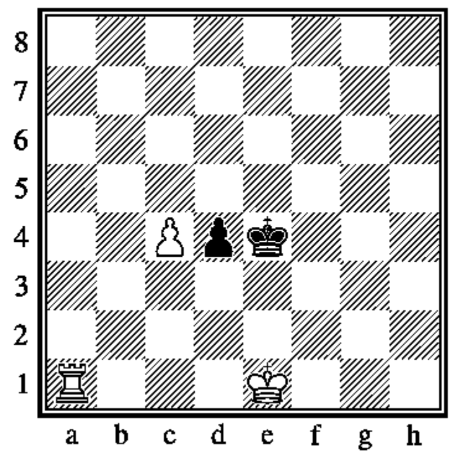
Testet på en 3 GHz Pentium 4 med 1 GB hukommelse

Slutspil konstruerede	Antal stillinger	Fremadrettet konstruction		Bagudrettet konstruktion	
		tid	tid/stilling	tid	tid/stilling
Alle 2 og 3 briks slutspil	411.382	0m19.564s	47.6μs	0m3.572s	8.6μs
Alle K+2 vs K	78.617.952	64m59.912s	49.6μs	17m37.094s	13.4μs
Alle K+1 vs K+1	78.822.912	179m5.106s	136.3μs	11m28.075s	8.7μs
KQQQK	38.497.536	5m24.448s	8.4μs	10m30.698s	16.4μs
KBBKN	119.218.176	832m41.147s	419.1μs	20m25.832s	10.3μs

Indekserings skemaer

Triviel indeksering:

	a	b	c	d	e	f	g	h	
8	56	57	58	59	60	61	62	63	8
7	48	49	50	51	52	53	54	55	7
6	40	41	42	43	44	45	46	47	6
5	32	33	34	35	36	37	38	39	5
4	24	25	26	27	28	29	30	31	4
3	16	17	18	19	20	21	22	23	3
2	8	9	10	11	12	13	14	15	2
1	0	1	2	3	4	5	6	7	1
	a	b	c	d	e	f	g	h	



$$
\begin{aligned}
\text{Indeks} &= (2^6)^4 \cdot HK + (2^6)^3 \cdot SK + (2^6)^2 \cdot HT + (2^6)^1 \cdot HB + (2^6)^0 \cdot SB \\
&= (2^6)^4 \cdot 4 + (2^6)^3 \cdot 28 + (2^6)^2 \cdot 0 + (2^6)^1 \cdot 26 + (2^6)^0 \cdot 27 \\
&= 000100\ 011100\ 000000\ 011010\ 011011 \text{ (i binær)}
\end{aligned}
$$

En passant: Var sidste træk c2-c4, flyt da bønderne til felt c3.

Rokade: Kan hvid rokere langt, flyt da hvidt tårn til sort konges placering.

Optimering. Kun en reduktion af antallet af lovlige stillinger overfører til OBDD'er.

$+$ Begræns fx hvid konge til a1-d1-d4 eller a1-d8. k ens brikker.
Kun stillinger med hvid til at trække.

$\div$ Nummerering af kongeplaceringerne. Ingen bønder på forreste eller bageste række.

Repræsentation v.h.a. OBDD'er

Complete Ordered Binary Decision Diagrams.

- Hver variabel skal testes på hver vej fra roden til et blad.
- Dette giver en opdeling i lag.

Repræsentation:

- Domimerende faktor er repræsentationen af kanterne.
- $\lceil \log(n) \rceil$ bits nødvendigt for at identificere en kant til en knude i et lag med n knuder.

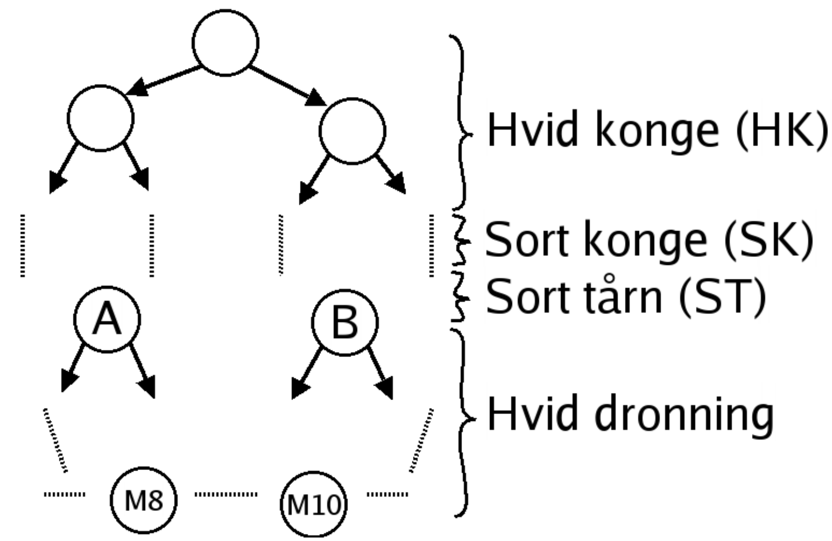
Kompakt repræsentation — reduktion af frihedsgrader.

- *Ideel* knude: Knude hvor udkanterne peger på 2 på hinanden følgende knuder.
- For en ideel knude behøves kun venstre udkant defineret — maksimer antallet af ideelle knuder.
- Modificeret max parring algoritme giver $\sim$ 40% ideelle knuder
- Pladsbesparelse: $\sim$ 20%

Optimeringer - Heuristik tildeling af don't cares

Eksempel:

- Vi har slutspillet KQKR — hvid konge og dronning mod sort konge og tårn.
- Slutspillet er for hvid til at trække.
- Variabelordningen er som illustreret til højre.
- Betragt delmængderne som knuderne A og B repræsenterer.
- Bemærk: $A[h5 \mapsto M10] = B[h6 \mapsto M10]$



A:

	a	b	c	d	e	f	g	h	
8	M8	M10	M10	*	M8	M10	M10	*	8
7	*	M10	M10	*	M10	M10	*	M10	7
6	M8	*	M8	*	M10	*	M10	M10	6
5	M8	M8	*	*	*	M10	M10	-**HK**-	5
4	-**ST**-	*	*	-**SK**-	*	*	*	*	4
3	M8	M8	*	*	*	M10	M10	M10	3
2	M8	*	M8	*	M10	*	M10	M10	2
1	*	M10	M10	*	M10	M10	*	M10	1
	a	b	c	d	e	f	g	h	

B:

	a	b	c	d	e	f	g	h	
8	M8	M10	M10	*	M8	M10	M10	*	8
7	*	M10	M10	*	M10	M10	*	M10	7
6	M8	*	M8	*	M10	*	M10	-**HK**-	6
5	M8	M8	*	*	*	M10	M10	M10	5
4	-**ST**-	*	*	-**SK**-	*	*	*	*	4
3	M8	M8	*	*	*	M10	M10	M10	3
2	M8	*	M8	*	M10	*	M10	M10	2
1	*	M10	M10	*	M10	M10	*	M10	1
	a	b	c	d	e	f	g	h	

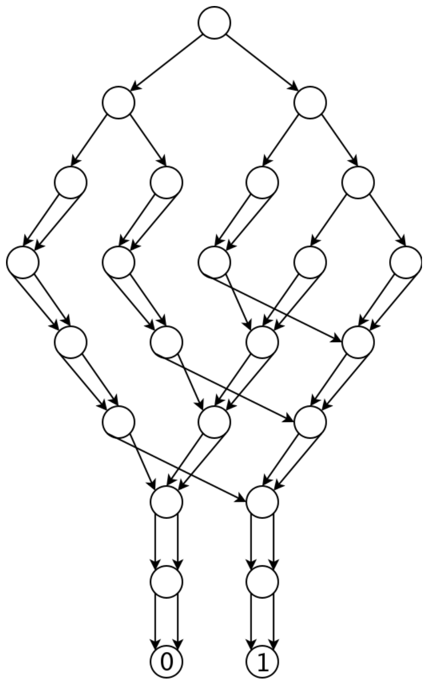
Optimeringer - Heuristik overskrivning af don't cares

Forbedring: $\sim 25\%$. For simple slutspil langt mere. Men:

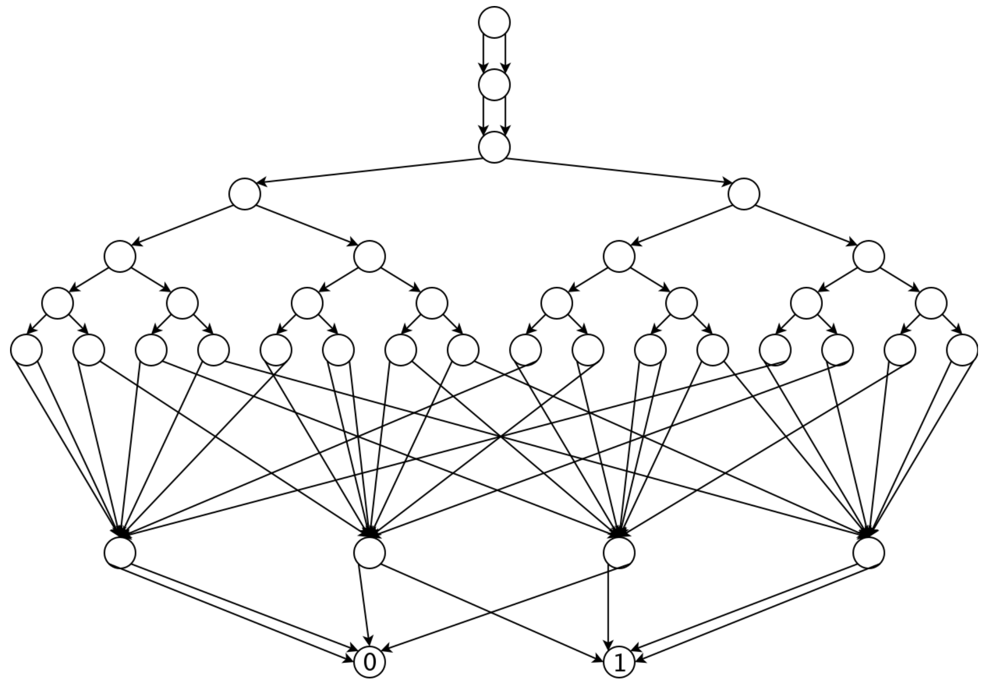
$t =$

```
***0 ***1 ***0 ***1 ***0 ***1 ***0 ***1 ***0 ***1 ***0 ***1 ***0 ***1 ***0 ***1
**0* **0* **1* **1* **0* **0* **1* **1* **0* **0* **1* **1* **0* **0* **1* **1*
*0** *0** *0** *0** *1** *1** *1** *1** *0** *0** *0** *0** *1** *1** *1** *1**
0*** 0*** 0*** 0*** 0*** 0*** 0*** 0*** 1*** 1*** 1*** 1*** 1*** 1*** 1*** 1***
```

Optimal løsning D:



OBDD D' fundet af algoritmen:



Inapproksimabilitet (for denne heuristik): $\frac{|D'|}{|D|} \geq 2^{\sqrt{|D|}}$ eller $\frac{|D'|}{|D|} \geq |t|^{1-\epsilon}$ for alle $\epsilon > 0$.

Optimeringer - Omnummering af skakbrædtets felter

Algoritmer fejlede.

Effektive nummereringer af felterne. 2 Klare vindere:

Z kurve
(bruges til kongerne)

	a	b	c	d	e	f	g	h	
8	21	23	29	31	53	55	61	63	8
7	20	22	28	30	52	54	60	62	7
6	17	19	25	27	49	51	57	59	6
5	16	18	24	26	48	50	56	58	5
4	5	7	13	**15**	37	39	45	47	4
3	4	6	**12**	**14**	36	38	44	46	3
2	1	**3**	**9**	**11**	33	35	41	43	2
1	**0**	**2**	**8**	**10**	32	34	40	42	1
	a	b	c	d	e	f	g	h	

Modificeret Z kurve
(bruges til alle andre brikker)

	a	b	c	d	e	f	g	h	
8	16	18	24	26	58	56	50	48	8
7	17	19	25	27	59	57	51	49	7
6	20	22	28	30	62	60	54	52	6
5	21	23	29	31	63	61	55	53	5
4	5	7	13	15	47	45	39	37	4
3	4	6	12	14	46	44	38	36	3
2	1	3	9	11	43	41	35	33	2
1	0	2	8	10	42	40	34	32	1
	a	b	c	d	e	f	g	h	

- Forbedring: $\sim$ 15%

Bedst at udnytte symmetri ved at begrænse sort konge (til a1-d1-d4 trekanten h.h.v. a1-d8 rektanglet).

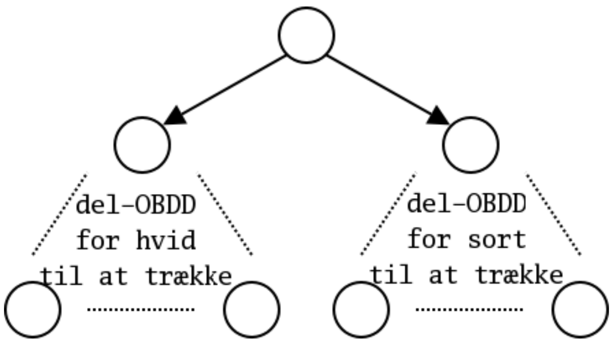
- Forbedring: $\sim$ 5-10%

Optimeringer

Variabelordning — sifting algoritmen.

- Baseret på lokal søgning
- Skal udføres før tildeling af *don't cares*
- Forbedring: $\sim$ 10%

Klyngeopdeling.



$$t = \text{0101 001* 11*1 10*1 0000 010* 11** 11*1 0111 1*01}$$

$$t_0 = \text{0101 001* **** **** 0000 010* **** **** **** ****}$$
$$t_1 = \text{**** **** 11*1 10*1 **** **** 11** 11*1 0111 1*01}$$
$$c = \langle 0, 1, 0, 1, 1 \rangle$$
$$t_{c[\lfloor \frac{i}{8} \rfloor]}[i] = t[i]$$

- Forbedring: $\sim$ 1%. Kun ganske få slutspil kan med fordel repræsenteres af flere OBDD'er.

Strengere definition af gyldig skakstilling. Antallet af stillinger reduceres med:

3 briks slutspil	4 briks slutspil	5 briks slutspil
0.137%	0.593%	1.53%

Sammenligning med Nalimov og EgmProbe

Slutspil	Afstand til skakmat		Vundet/tabt/uafgjort	
	Dette speciale	Nalimov	Dette speciale	EgmProbe 2.0
KPK	19563+**15491**	**17654**+16589	**1172+1336**	3962
KNK	551+551	**186+187**	551+551	**56**
KBK	**551**+551	1503+**187**	551+551	**56**
KRK	7666+10196	**7059+7051**	551+708	**92**
KQK	**4556**+6998	7605+**5961**	551+1076	**84**
KPPK	435415+**442704**	**384915**+610687	**6216+11488**	20392
KPKP	1339738	**1116299**	**122913**	213592
KNPK	**1277110**+1383360	1547889+**1227423**	**14860+34944**	108612
KQKR	861440+1098842	**562344+641119**	4341+14837	**10970**
KQKQ	**197637**	257632	42189	**6316**
...	...	...	...	...
Alle 3-4 briks s.	**13890243**+17409488	14814890+**15768949**	1039436	**970710**

Endgame	Afstand til skakmat	
	This thesis	Nalimov
KBBBK	**2267775+3007899**	4599013+6450288
KNNNK	**4479412+4599705**	5127367+6831165
KQQQK	**217341+969960**	1620714+4302377
KRRRK	**244943+1321303**	1942079+4665085

Sammenligning med run length encoding

Kun for VTU. Slår klart run length encoding — med en faktor $\sim$ 5-10!

HRLE+ : Run length encoding med (kanonisk) Huffman komprimering af længderne
: og tildeling af *don't cares* til hyppigst forekomne værdi.

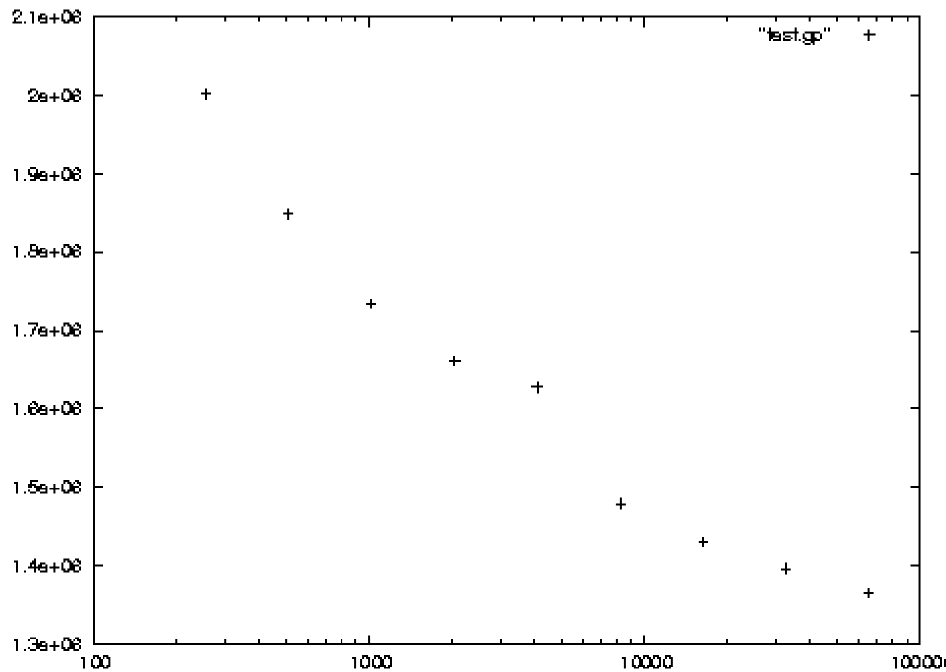
Komprimering:	KRK	KQK	KPK	KRKP	KQKR
HRLE+	5+1213	5+1262	7413+8498	177521+553326	10430+199697
OBDD	551+**708**	551+**1076**	**1172+1336**	**31197+59133**	**4341+14837**

Opslagstid

Opslag i	KBBBK	KRRRK	KRKP	KQKR	KQK
OBDD	1.91μs	1.71μs	0.72μs	0.69μs	0.48μs
Nalimov, komprimeret	78.1μs	61.3μs	28.4μs	84.1μs	-

Slutspillet KRRRK er en anelse mindre end KQKR. Men flere af lagene giver cache fejl. En 8192 bytes blokkomprimering er 35 - 120 gange langsommere end at bruge OBDD'er.

Blokstørrelse	KRKP
65536	1364599
32768	1396658
16384	1431193
8192	**1479273**
4096	1627879
2048	1662050
1024	1734596
512	1848682
256	2001733



Optimal tildeling af don't cares

Baggrund

- En OBDD giver en kompakt repræsentation for en funktion $f : \{0,1\}^n \mapsto \{0, \ldots, k-1\}$.
- I mange anvendelser haves kun en partielt defineret funktion f_p.
- Problem: Find en minimal OBDD som er konsistent med f_p.

Tidligere resultater

- Sauerhoff og Wegener har vist NP-hårdhed og inapproksimabilitet for f_p givet som en OBDD.
- Tilsvarende har Hirata, Shimozono og Shinohara vist NP-hårdhed og inapproksimabilitet for f_p givet som en mængde af positive instanser og en mængde af negative instanser.
- Begge beviser er en reduction fra graffarvningsproblemet.
- Her kendes følgende inapproksimationsresultater (underforstået: $\forall \epsilon > 0$)
  $\mathrm{P} \neq \mathrm{NP} \Rightarrow |V|^{\frac{1}{7}-\epsilon}$ og $\mathrm{NP} \not\subseteq \mathrm{ZPP} \Rightarrow |V|^{1-\epsilon}$.

Ny udfordring

- I dette speciale er f_p givet som en tabel.
- Denne repræsentation kunne potentielt set være for simpel til at konstruere små instanser som indfanger kompleksiteten af graffarvningsproblemet.
- Dette speciale afkræfter denne mistanke, og giver samtidig et stærkt inaproksimationsresultat.

Sammenligning med tidligere resultater

Sauerhoff og Wegener:

- Baggrund: Ufuldstændigt definerede funktioner/kredsløb.
- Input: Definition 1: OBDD D_f og D_c med værdier i $\{0,1\}$. D_c angiver *care* instanserne af D_f
  Definition 2: OBDD D med værdier i $\{0,1,*\}$. $*$ angiver *don't care*.
- Inapproksimationsresultat: Def. 2: P $\neq$ NP $\Rightarrow |D|^{\frac{1}{21}-\epsilon}$, NP $\not\subseteq$ ZPP $\Rightarrow |D|^{\frac{1}{3}-\epsilon}$.
  Def. 1 $\leq_p$ Def. 2. Reduktionen konstruerer D_f og D_c af samme størrelse som D.

Hirata, Shimozono og Shinohara:

- Baggrund: Opstil mindste hypotese konsistent med en mængde positive og negative instanser.
- Input: P, $N \in \{0,1\}^n$ med $P \cap N = \emptyset$. P er de positive instanser.
- Inapproksimationsresultat: P $\neq$ NP $\Rightarrow n^{\frac{1}{7}-\epsilon}$, NP $\not\subseteq$ ZPP $\Rightarrow n^{1-\epsilon}$.
  Bemærk n er antal variabler og der kan gælde $n \ll |N \cup P| \leq 2^n$.

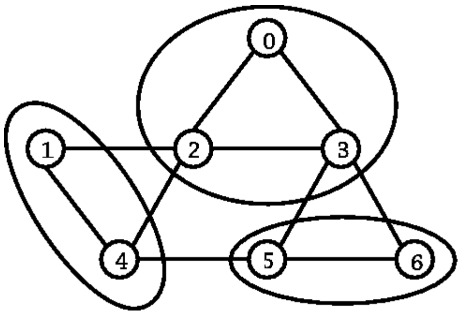
Dette speciale:

- Baggrund: Slutspilstabeller for skak. Stor tæthed af værdier.
- Input: Ufuldstændig sandhedstabel t med værdier fra $\{0,1,*\}$. $|t| = 2^n$
- Inapproksimationsresultat: P $\neq$ NP $\Rightarrow |t|^{\frac{1}{28}-\epsilon}$, NP $\not\subseteq$ ZPP $\Rightarrow |t|^{\frac{1}{4}-\epsilon}$.

Fællesbetegnelse: Minimal Konsistent OBDD, MKO

Fælles for beviserne

Samme ide er anvendt — her illustreret for reduktionen fra klike problemet.



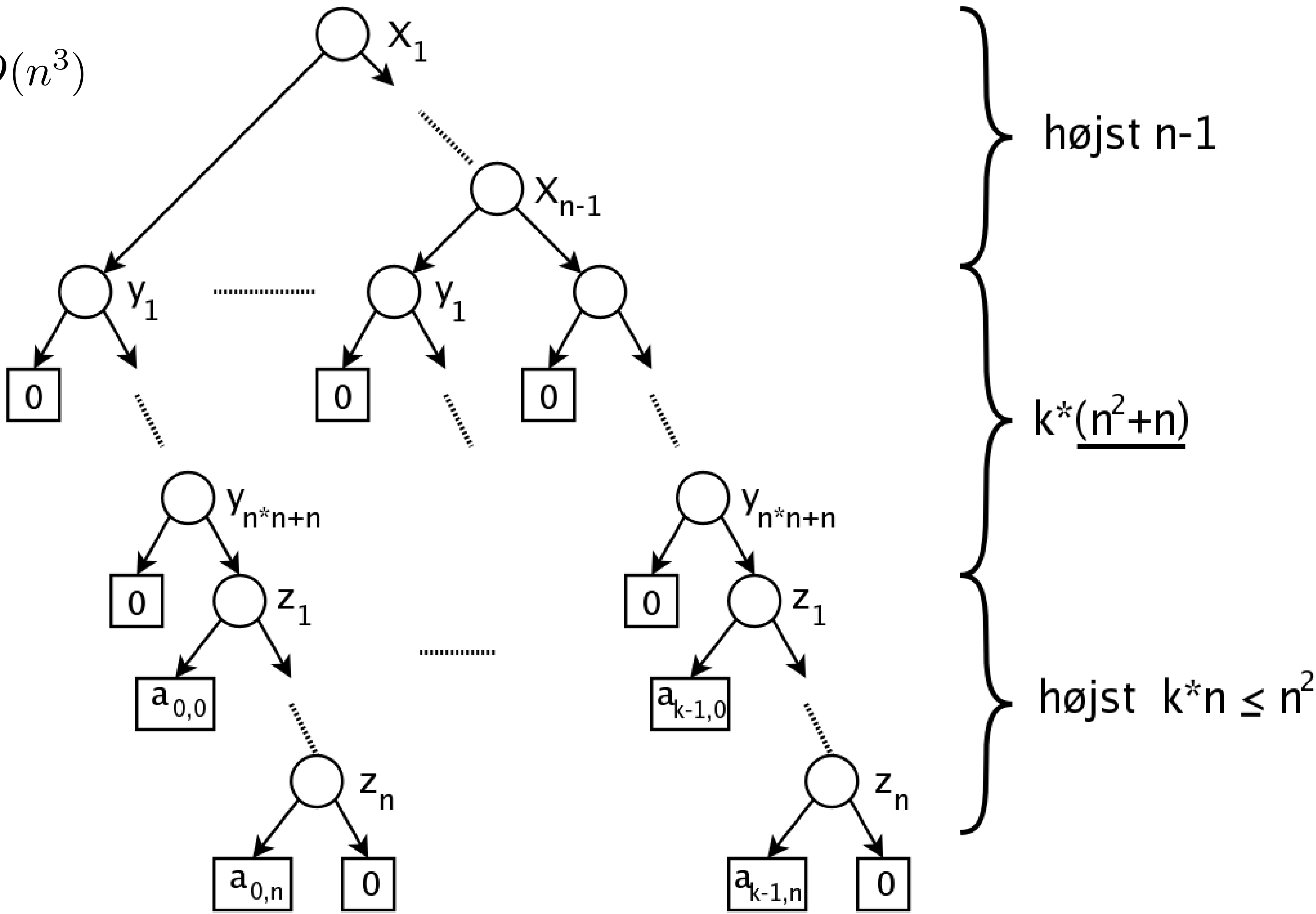
s_0 = 10**000 dækkes af 1011000 = s'_0
s_1 = 01*0*00 dækkes af 0100100 = s'_1
s_2 = **1**00 dækkes af 1011000 = s'_0
s_3 = *0*10** dækkes af 1011000 = s'_0
s_4 = 0**01*0 dækkes af 0100100 = s'_1
s_5 = 000**1* dækkes af 0000011 = s'_2
s_6 = 000*0*1 dækkes af 0000011 = s'_2

Reduktion:

- NP-hårdhed: Karp reduktion. Grafproblemet har en løsning af størrelse k hviss MKO har en løsning af størrelse $g(k)$.
- Problem: Størrelsen af en optimal løsning kan ikke vurderes præcist. Uvished $< n$.
  Løsning: oppumpning af kontrolleret del med faktor n. Gør inapproksimationsresultatet svagere.
- Inapproksimabilitet: Der er en effektiv algoritme til at konstruere en løsning til grafproblemet af størrelse k givet en løsning til MKO af størrelse $\leq g(k)$.
- D.v.s. en algoritme der approksimerer MKO kan bruges til at lave en algoritme der approksimerer grafproblemet.

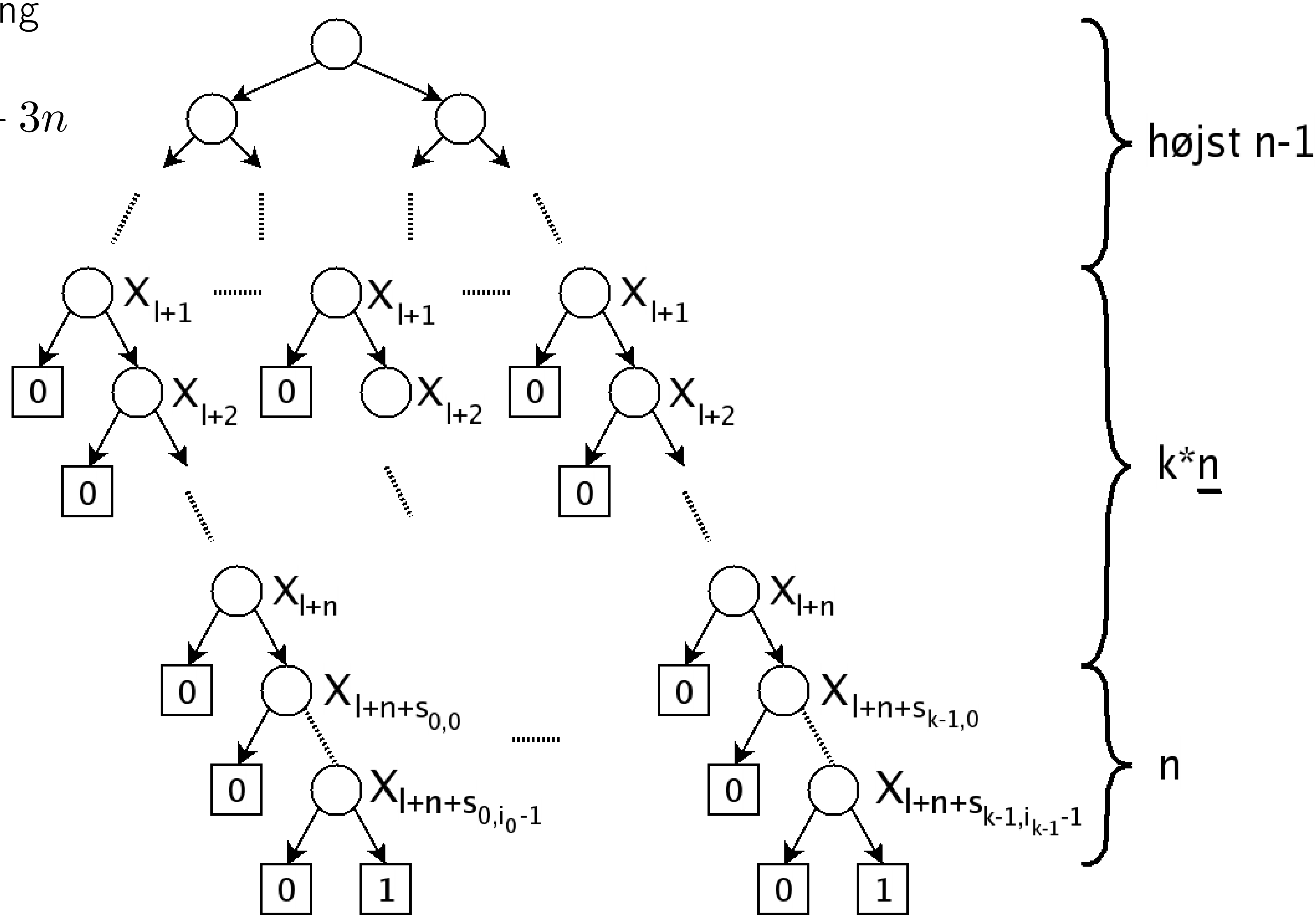
Sauerhoff og Wegener

k : optimal løsning
$n = |V|$
$|D| \in O(n^3)$



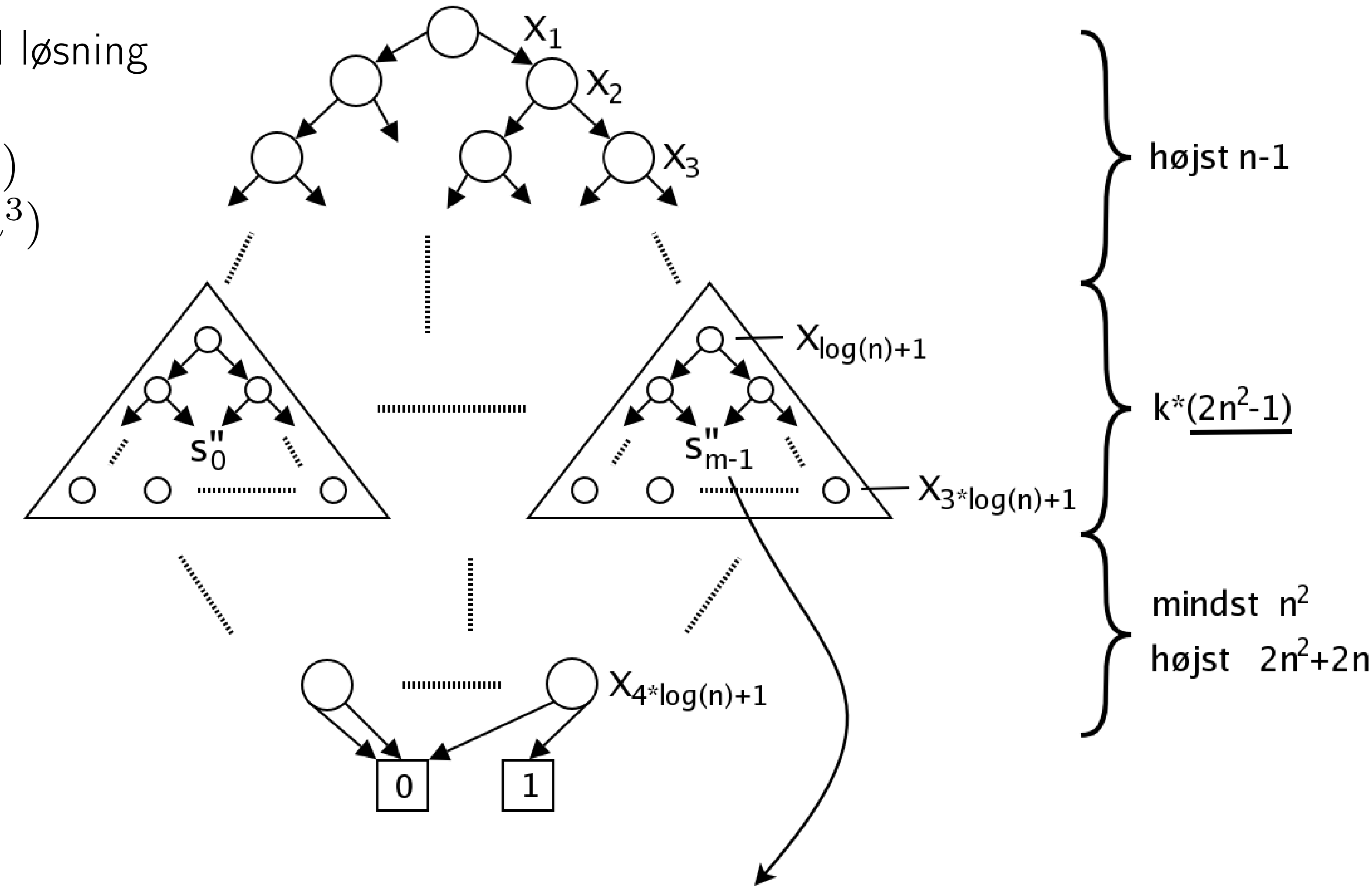
Hirata, Shimozono og Shinohara

k : optimal løsning
$n = |V|$
$|N \cup P| \leq n^2 + 3n$
$|D| \in O(n^3)$



Dette speciale

k : optimal løsning
$n = |V|$
$|t| \in O(n^4)$
$|D| \in O(n^3)$

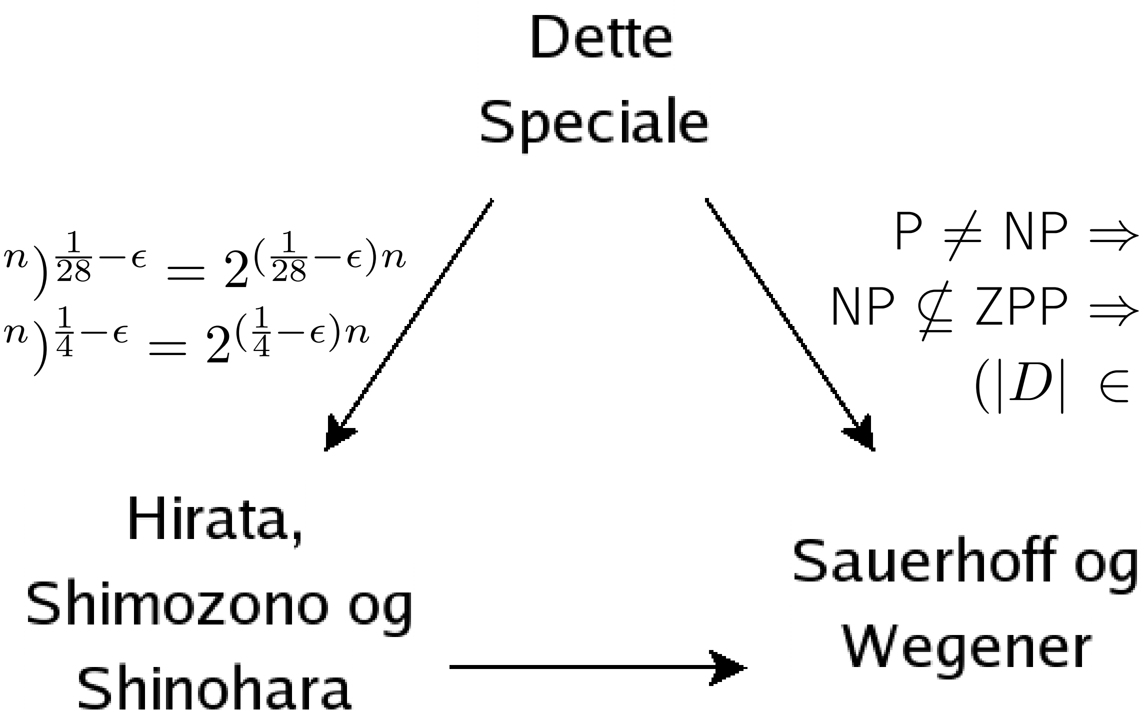


$$s''_i = b_0 s'_i b_1 s'_i b_2 s'_i \ldots b_{n^2-1} s'_i$$

b_j er en binær repræsentation af j hvor de betydende cifre er opdelt rekursivt.

Generaliseringer og forbedring af tidligere resultater

$$\mathsf{P} \neq \mathsf{NP} \Rightarrow |t|^{\frac{1}{28}-\epsilon}$$
$$\mathsf{NP} \not\subseteq \mathsf{ZPP} \Rightarrow |t|^{\frac{1}{4}-\epsilon}$$



$$\mathsf{P} \neq \mathsf{NP} \Rightarrow (2^n)^{\frac{1}{28}-\epsilon} = 2^{(\frac{1}{28}-\epsilon)n}$$
$$\mathsf{NP} \not\subseteq \mathsf{ZPP} \Rightarrow (2^n)^{\frac{1}{4}-\epsilon} = 2^{(\frac{1}{4}-\epsilon)n}$$

$$\mathsf{P} \neq \mathsf{NP} \Rightarrow |D|^{\frac{1}{21}-\epsilon}$$
$$\mathsf{NP} \not\subseteq \mathsf{ZPP} \Rightarrow |D|^{\frac{1}{3}-\epsilon}$$
$$(|D| \in O(|t|^{\frac{4}{3}}))$$

$$\mathsf{P} \neq \mathsf{NP} \Rightarrow n^{\frac{1}{7}-\epsilon}$$
$$\mathsf{NP} \not\subseteq \mathsf{ZPP} \Rightarrow n^{1-\epsilon}$$

$$\mathsf{P} \neq \mathsf{NP} \Rightarrow \log(|D|)^{\frac{1}{7}-\epsilon}$$
$$\mathsf{NP} \not\subseteq \mathsf{ZPP} \Rightarrow \log(|D|)^{1-\epsilon}$$

$$\mathsf{P} \neq \mathsf{NP} \Rightarrow |D|^{\frac{1}{21}-\epsilon}$$
$$\mathsf{NP} \not\subseteq \mathsf{ZPP} \Rightarrow |D|^{\frac{1}{3}-\epsilon}$$

$$\left(\begin{array}{r} \mathsf{P} \neq \mathsf{NP} \Rightarrow |D|^{\frac{1}{21}-\epsilon} \\ \mathsf{NP} \not\subseteq \mathsf{ZPP} \Rightarrow |D|^{\frac{1}{3}-\epsilon} \end{array}\right)$$

Konklusion

- OBDD'er kan optimeres til at matche Lempel-Ziv baseret komprimering og er run length encoding suverænt overlegne.
- Opslagstiden er med $< 2\mu s$ nede på noget acceptabelt.
- Alle 3-4 briks slutspil er blevet komprimeret som OBDD'er, men 5 briks slutspil ligger på grænsen af almindelige computeres formåen.
- Alle optimeringer har tilsammen mere end halveret størrelsen af OBDD'erne.
- Største optimering skyldes tildeling af *don't cares*
- Beviset for at dette problem er NP-hårdt generaliserer tidligere relaterede resultater.
- Inapproksimationsresulatet viser endvidere at vi må acceptere ekstreme worst case eksempler og i stedet fokusere på, hvad der virker godt i praksis.
- Målt som funktion af antal variable er inapproksimationsresultatet exponentielt meget kraftigere end de tidligere resultater.