Time-Memory Tradeoffs für Subset Sum und Dekodierung

Masterarbeit

Alexander Kulpe

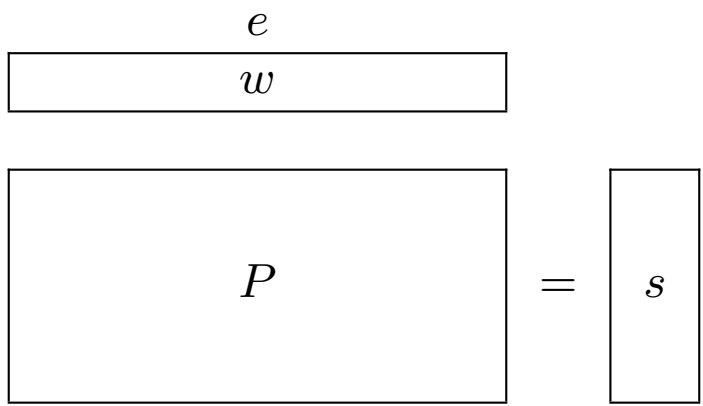
Ruhr-Universität Bochum

11. März 2024

Inhaltsverzeichnis

Inhaltsverzeichnis

Motivation: Codebasierte Kryptographie



- kann als vektorielle Subset Sum Variante aufgefasst werden

$\Rightarrow$ Verbesserungen für Subset Sum hilft bei Dekodierung

Inhaltsverzeichnis

Subset Sum

Problem: Random Subset Sum

- **Gegeben:** $((a_1, \ldots, a_n), t) \in (\mathbb{Z}_{2^n})^n \times (\mathbb{Z}_{2^n})^n$ mit $t = \sum_{i=1}^{n} \varepsilon_i a_i \bmod 2^n$, $\varepsilon \in \{0,1\}^n$ $\left(\frac{n}{2}\right)$
- **Gesucht:** ε

- Anwendung bei ISD-Algorithmen / Kryptanalyse
- Beste Algorithmen sehr speicherintensiv

$\Rightarrow$ Time-Memory Tradeoffs

Erste Algorithmen

Brute-Force

- **Laufzeit:** $\tilde{\mathcal{O}}\left(2^n\right)$
- **Speicher:** $\tilde{\mathcal{O}}\left(1\right)$

Meet-in-the-Middle

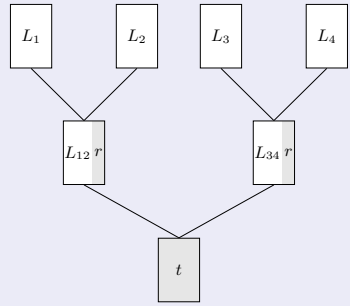
$$\sum_{i=1}^{n} \varepsilon_i a_i = t \bmod 2^n$$

$$\Leftrightarrow \sum_{i=1}^{\frac{n}{2}} \varepsilon_i a_i = t - \sum_{i=\frac{n}{2}+1}^{n} \varepsilon_i a_i \bmod 2^n$$

- **Laufzeit:** $\tilde{\mathcal{O}}\left(2^{\frac{n}{2}}\right)$
- **Speicher:** $\tilde{\mathcal{O}}\left(2^{\frac{n}{2}}\right)$

Schroeppel-Shamir

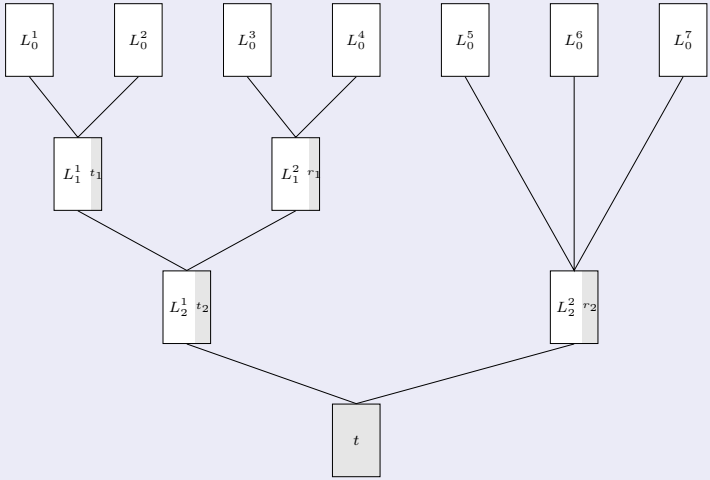
$$\sum_{i=1}^{\frac{n}{4}} \underbrace{\varepsilon_i a_i}_{L_1} + \sum_{i=\frac{n}{4}+1}^{\frac{n}{2}} \underbrace{\varepsilon_i a_i}_{L_2} = t - \sum_{i=\frac{n}{2}+1}^{\frac{3}{4}n} \underbrace{\varepsilon_i a_i}_{L_3} - \sum_{i=\frac{3}{4}n+1}^{n} \underbrace{\varepsilon_i a_i}_{L_4} \bmod 2^n$$



Laufzeit: $\tilde{\mathcal{O}}\left(2^{\frac{n}{2}}\right)$, **Speicher:** $\tilde{\mathcal{O}}\left(2^{\frac{n}{4}}\right)$

Erste Algorithmen II

7-Dissection



Laufzeit: $\tilde{\mathcal{O}}\left(2^{\frac{4}{7}n}\right)$, **Speicher:** $\tilde{\mathcal{O}}\left(2^{\frac{1}{7}n}\right)$

Lemma (7-Dissection-Tradeoff)

$\frac{1}{7} \leq \lambda \leq \frac{1}{4}$. Random Subset Sum *lösbar in erwarteter Speicher* $M = \tilde{\mathcal{O}}\left(2^{\lambda n}\right)$ *und erwarteter Zeit* $T = \tilde{\mathcal{O}}\left(2^{\frac{2}{3}(1-\lambda)n}\right)$.

Repräsentationen

- **Idee:** Betrachte größeren Suchraum mit noch mehr Lösungen
- Suchraum MITM: $\mathcal{S} = \{0,1\}^{\frac{n}{2}} \times \{0\}^{\frac{n}{2}}$
- Suchraum Repräsentationen: $\mathcal{S} = \{0,1\}^n \binom{n}{4}$
- Statt einer Lösung $\varepsilon \in \{0,1\}^n \binom{n}{2}$ nun $\binom{n/2}{n/4}$-viele Repräsentationen $(\varepsilon_1, \varepsilon_2) \in \mathcal{S}^2$ mit $\varepsilon = \varepsilon_1 + \varepsilon_2$

Beispiel ($n = 8$)

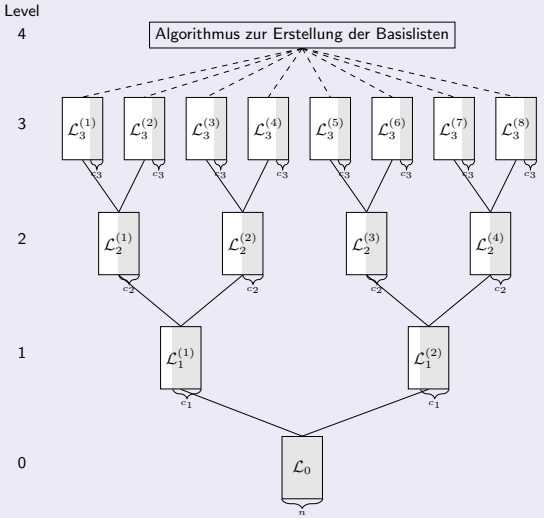
- MITM: $\varepsilon = 10100110$
- Repräsentationen:

$(10100000, 00000110)$ $(10000100, 00100010)$ $(10000010, 00100100)$
$(00100100, 10000010)$ $(00100010, 10000100)$ $(00000110, 10100000)$

	MITM	Repräsentationen
$\lvert\mathcal{S}\rvert$	$2^{\frac{n}{2}}$	$\binom{n}{n/4} = 2^{0,8113n}$
Lösungen	1	$\binom{n/2}{n/4} = 2^{n/2}$

$\Rightarrow$ Betrachte nur einen $2^{-n/2}$-Anteil des Suchraums für eine Lösung

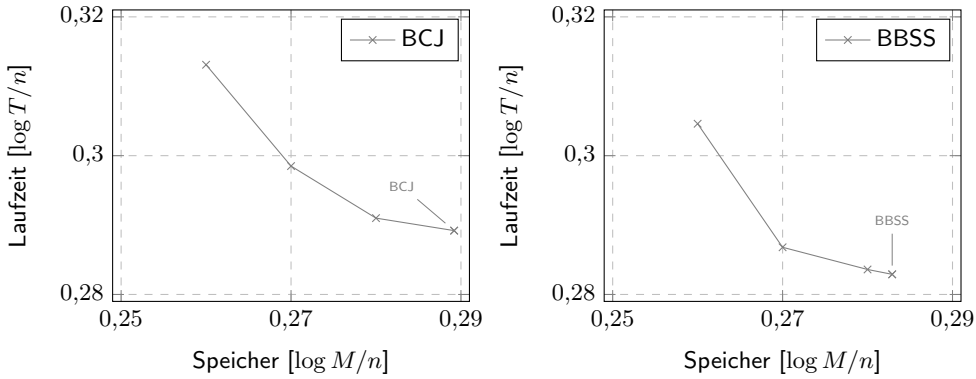
Howgrave-Graham-Joux



Rep.	T	M
$\{0,1\}$	$\tilde{\mathcal{O}}\left(2^{0,3373n}\right)$	$\tilde{\mathcal{O}}\left(2^{0,3113n}\right)$
$\{0,1,-1\}$	$\tilde{\mathcal{O}}\left(2^{0,2892n}\right)$	$\tilde{\mathcal{O}}\left(2^{0,2892n}\right)$
$\{0,1,-1,2\}$	$\tilde{\mathcal{O}}\left(2^{0,2829n}\right)$	$\tilde{\mathcal{O}}\left(2^{0,2829n}\right)$

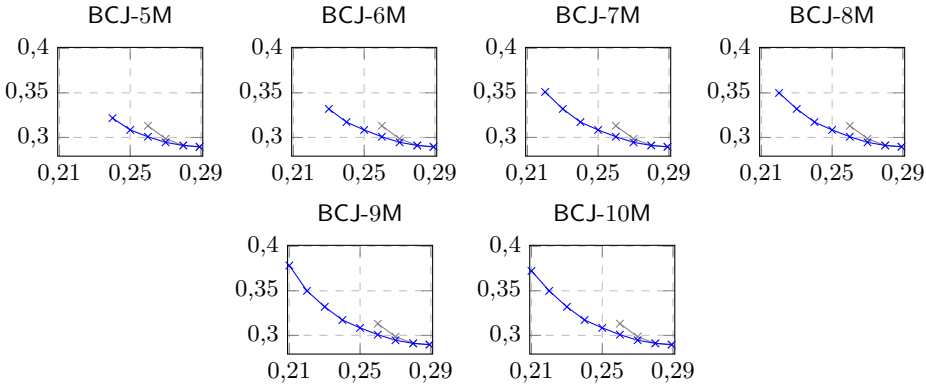
Inhaltsverzeichnis

Subset Sum Tradeoff: Impliziter Tradeoff



- **Beobachtung:** Obere Level dominieren Speicher- und Laufzeitkomplexität
- **Lösungsansätze:**
  - Erhöhe Tiefe des Suchbaums
  - Tausche Algorithmus zur Basislistenkonstruktion aus

Subset Sum Tradeoff: Höhere Tiefe I

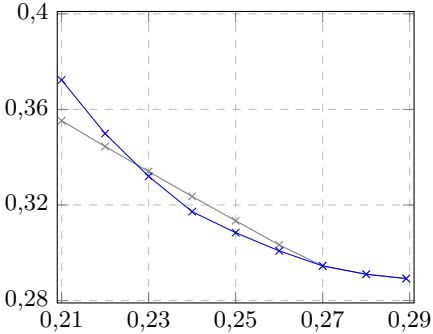


- monoton fallend* und konvergent

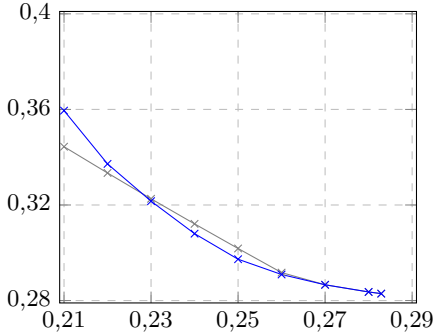
*Conditions apply

Subset Sum Tradeoff: Höhere Tiefe II

Aktuell bester Tradeoff [EZ23] vs BCJ-XM

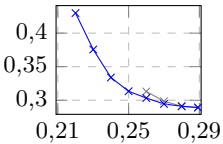


Aktuell bester Tradeoff [EZ23] vs BBSS-XM

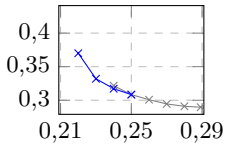


Subset Sum Tradeoff: Schroeppel-Shamir

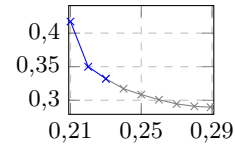
BCJ-4M vs BCJ-4S



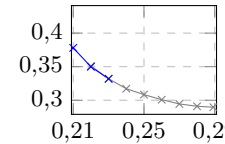
BCJ-5M vs BCJ-5S



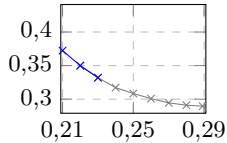
BCJ-6M vs BCJ-6S



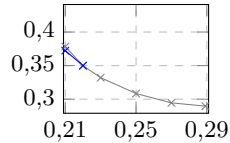
BCJ-7M vs BCJ-7S



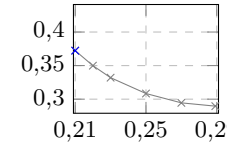
BCJ-8M vs BCJ-8S



BCJ-9M vs BCJ-9S



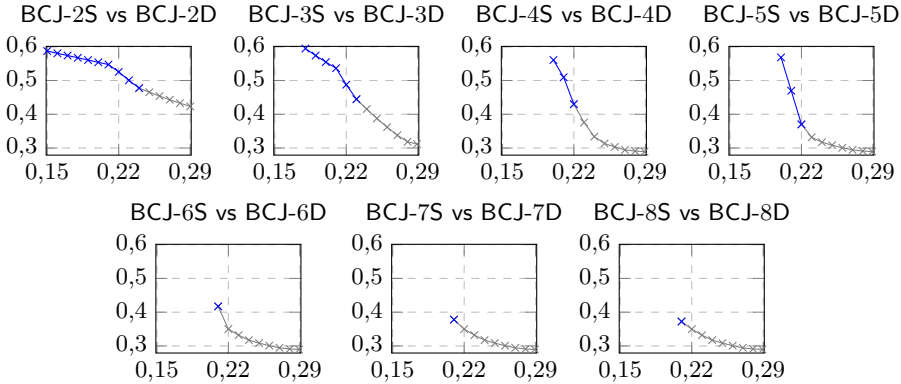
BCJ-10M vs BCJ-10S



- monoton fallend und konvergent
- Schroeppel-Shamir für feste Tiefe $X < 10$ besser als MITM
- BCJ-XM = BCJ-XS

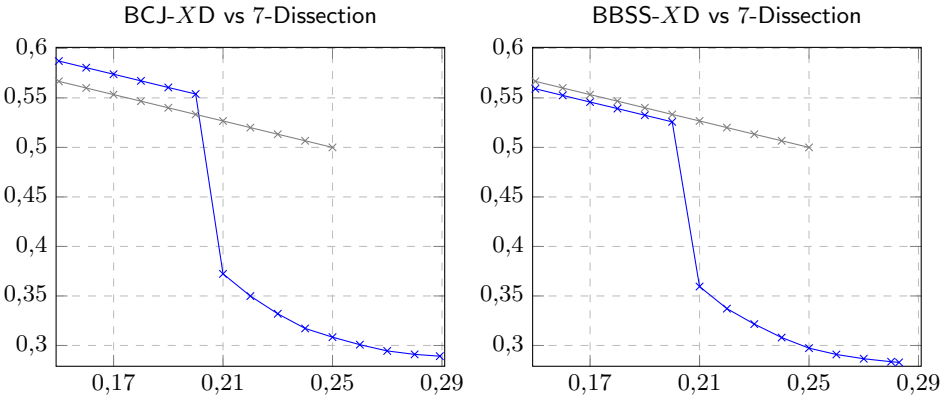
$\Rightarrow$ Tiefe wichtiger als Algorithmus zur Basislistenkonstruktion

Subset Sum Tradeoff: 7-Dissection



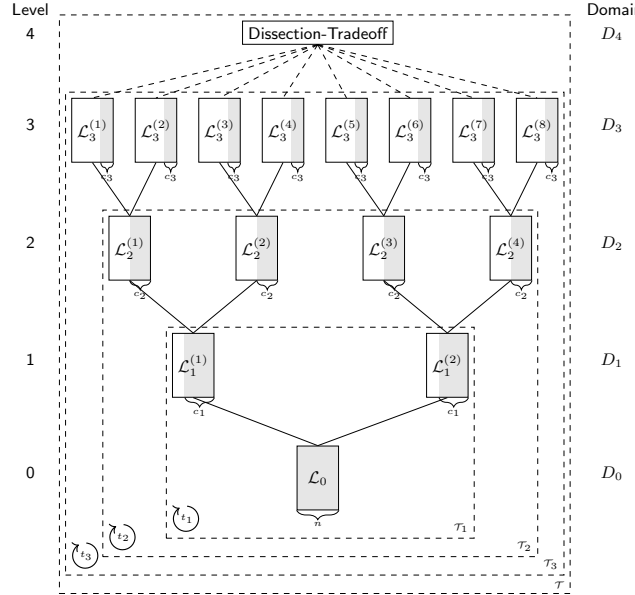
- $\log M \geq 0,21n$: monoton fallend und konvergent
- $\log M \leq 0,20n$: Basislistenkonstruktion dominiert Laufzeitkomplexität
  $\Rightarrow$ Geringere Tiefe besser (?)

Subset Sum Tradeoff: 7-Dissection II ($\log M \leq 0,20n$)

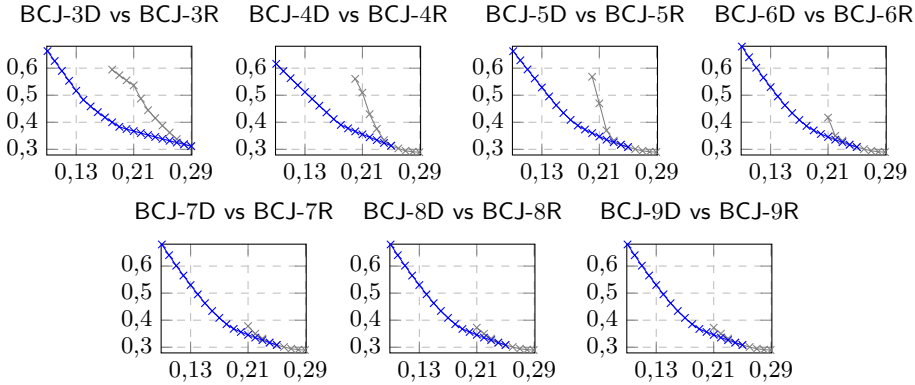


- BCJ: BCJ-XD schlechter als 7-Dissection
- BBSS: BBSS-XD besser als 7-Dissection mit optimaler Tiefe 3

Subset Sum Tradeoffs: Aktueller Tradeoff / Wiederverwendung bereits berechneter Subtrees

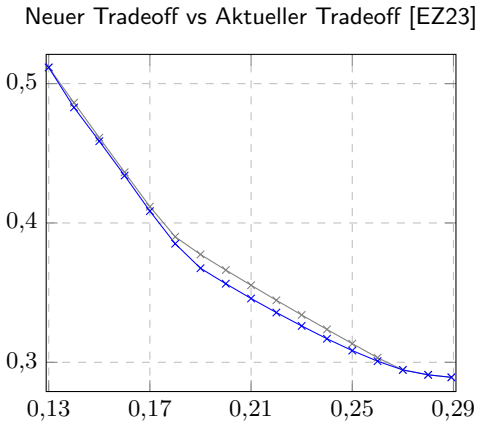


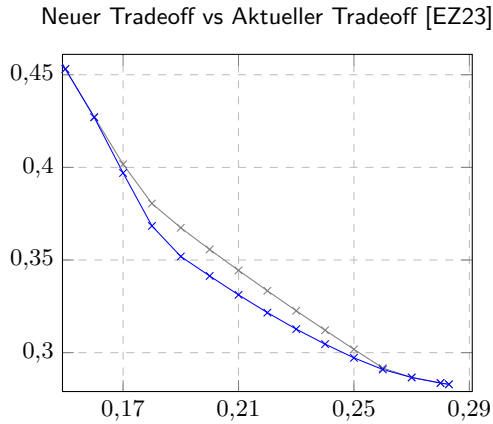
Subset Sum Tradeoffs: Wiederverwendung von bereits berechneten Subtrees



- $\log M \geq 0,19n$: monoton fallend und konvergent
- $0,16n \leq \log M \leq 0,18n$: monoton fallend und konvergent, geringere optimale Tiefe
- $\log M \leq 0,15n$: Basislistenkonstruktion und untere Liste in geringerer Tiefe besser balanciert (BCJ: Tiefe $3,4$, BBSS: Tiefe 4)

Subset Sum Tradeoff: Contribution



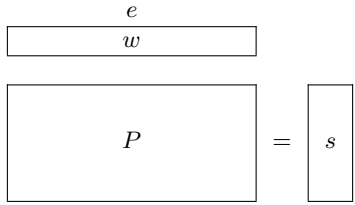


- BCJ: Verbesserung um bis zu $\tilde{\mathcal{O}}\left(2^{0,0099n}\right)$ bzw. $2,68\ \%$
- BBSS: Verbesserung um bis zu $\tilde{\mathcal{O}}\left(2^{0,0155n}\right)$ bzw. $4,22\ \%$

Inhaltsverzeichnis

Syndromdekodierung

- linearer $[n, k, d]$-Code C: C ist Untervektorraum von $\mathbb{F}_2^n$ mit Länge n, Dimension k und Distanz d
- Parity-Check-Matrix P: $C = \{c \mid c \in \mathbb{F}_2^n, Pc^t = 0\}$
- c Codewort, $x = c + e$ fehlerhaftes Codewort mit Fehlervektor e
- Syndrom s: $s = Px^t = P(c^t + e^t) = Pe^t$

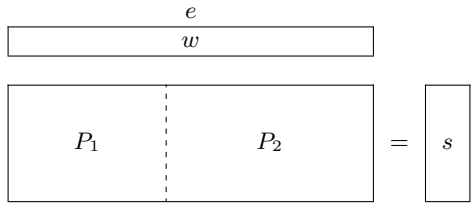


Syndromdekodierungsproblem

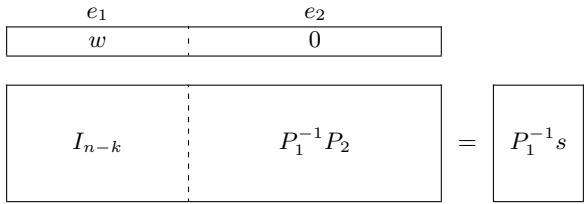
- **Gegeben:** Parity-Check-Matrix $P \in \mathbb{F}_2^{(n-k)\times n}$, Syndrom $s \in \mathbb{F}_2^{n-k}$, Gewicht w
- **Gesucht:** Fehlervektor $e \in \mathbb{F}_2^n(w)$ s.d. $Pe^t = s$

- half distance: $w = \lfloor \frac{d-1}{2} \rfloor$
- full distance: $w = d - 1$

Prange



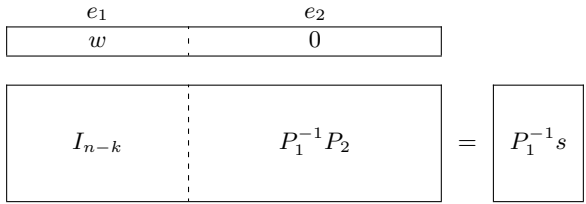
Prange



- $e_1 + P_1^{-1}P_2e_2 = P_1^{-1}s$
- Für $e_2 = 0^k$ gilt $e_1 = P_1^{-1}s$

$\Rightarrow$ Permutiere P, sodass $\mathrm{wt}(e_1) = w$

Prange

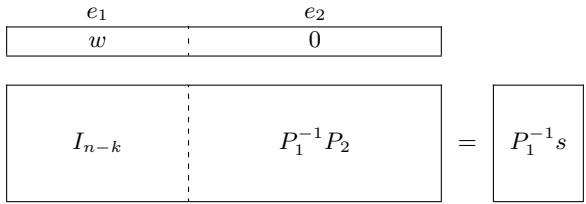


- $e_1 + P_1^{-1}P_2e_2 = P_1^{-1}s$
- Für $e_2 = 0^k$ gilt $e_1 = P_1^{-1}s$

$\Rightarrow$ Permutiere P, sodass $\mathrm{wt}(e_1) = w$

- Laufzeit: $T = \Pr[\text{gute Permutation}]^{-1}$

Prange

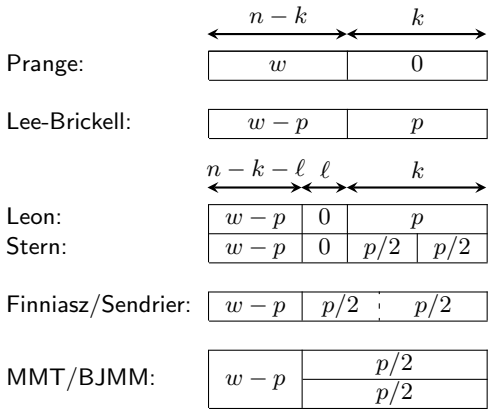


- $e_1 + P_1^{-1}P_2e_2 = P_1^{-1}s$
- Für $e_2 = 0^k$ gilt $e_1 = P_1^{-1}s$

$\Rightarrow$ Permutiere P, sodass $\mathrm{wt}(e_1) = w$

- Laufzeit: $T = \Pr[\text{gute Permutation}]^{-1}$
- Lässt sich die Wahrscheinlichkeit für eine gute Permutation erhöhen?

ISD



MMT

- Repräsentationen:

$$1 = 0 + 1$$
$$1 = 1 + 0$$
$$0 = 0 + 0$$

- optimale Tiefe: 2

BJMM

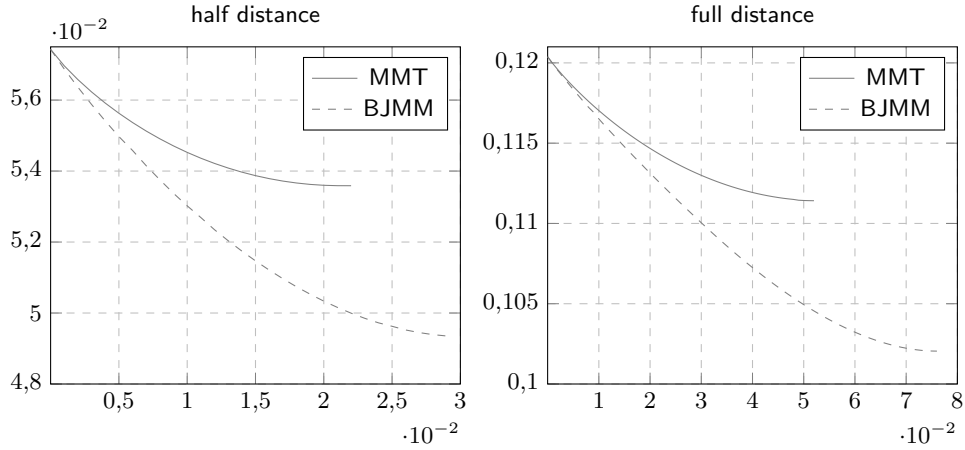
- Repräsentationen:

$$1 = 0 + 1$$
$$1 = 1 + 0$$
$$0 = 0 + 0$$
$$0 = 1 + 1$$

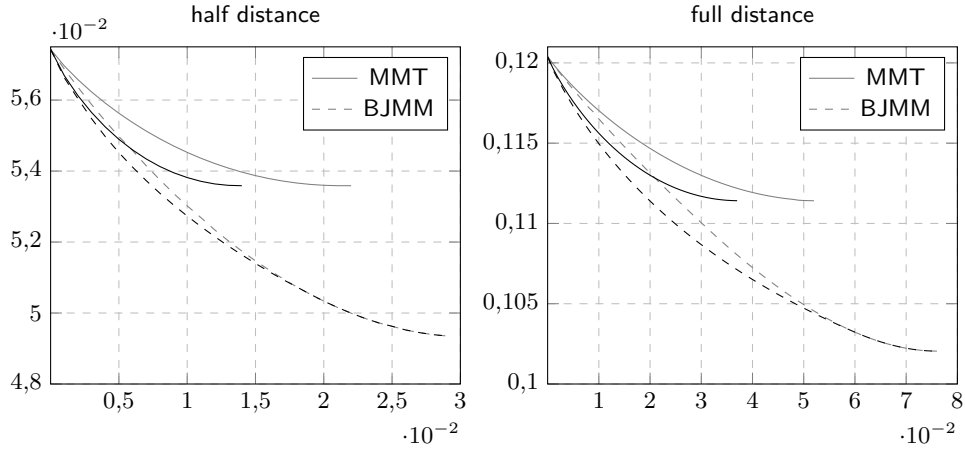
- optimale Tiefe: 3

Inhaltsverzeichnis

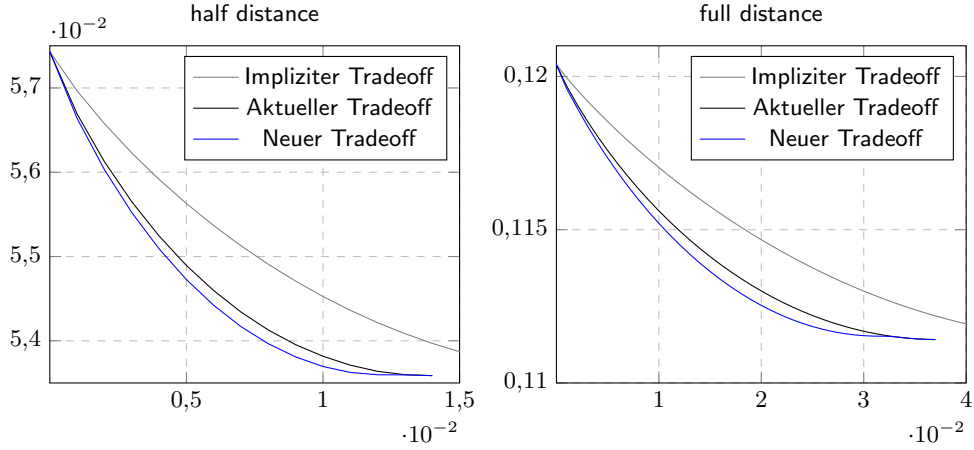
Dekodierung Tradeoff: Impliziter Tradeoff



Dekodierung Tradeoff: Wiederverwendung von bereits berechneten Subtrees [EZ23]



MMT Tradeoff: Contribution



- half distance: Verbesserung um bis zu $\tilde{\mathcal{O}}\left(2^{0,000175n}\right)$ / $0,32$ %
- full distance: Verbesserung um bis zu $\tilde{\mathcal{O}}\left(2^{0,000492n}\right)$ / $0,43$ %
- BJMM: keine Verbesserung

Zusammenfassung / Ausblick

Subset Sum

- Tiefe erhöhen
- Austausch Algorithmen zur Basislistenkonstruktion
- Wiederverwendung von bereits berechneten Subtrees
- BCJ: Verbesserung um bis zu $\tilde{\mathcal{O}}\left(2^{0,0099n}\right)$ bzw. $2,68\ \%$
- BBSS: Verbesserung um bis zu $\tilde{\mathcal{O}}\left(2^{0,0155n}\right)$ bzw. $4,22\ \%$

Dekodierung

- MMT: Verbesserung um bis zu $\tilde{\mathcal{O}}\left(2^{0,000492n}\right)$ / $0,43\ \%$
- BJMM: keine Verbesserung

$\Rightarrow$ BJMM asymptotisch besser, MMT in Praxis bevorzugt

Offene Fragen

- Weitere Anwendungen des neuen Subset Sum Tradeoffs
- Implementierung von neuer MMT-Variante und Analyse

Achtung

Optimale Algorithmenparameter sind im Allgemeinen nicht optimal für Tradeoffs!

Fragen?