ルール重みが変動するルール順序最適化問題に対する発見的解法

原田 崇司[1]　田中 賢[1]　三河 賢治[2]

[1] 神奈川大学大学院 理学研究科 理学専攻 情報科学領域

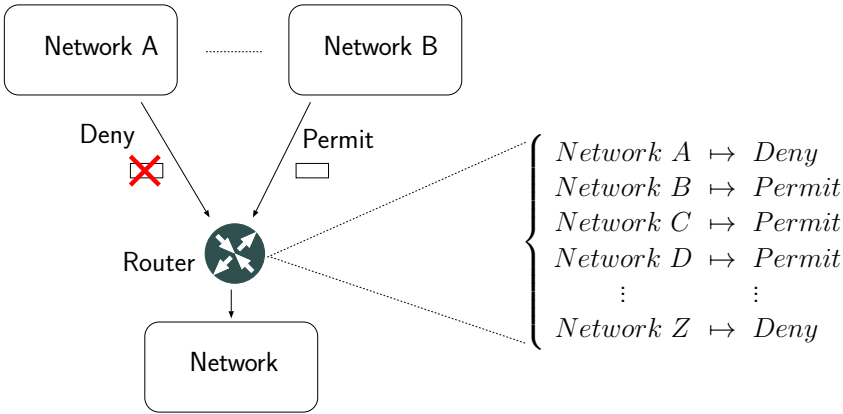
[2] 新潟大学学術情報機構情報基盤センター

2018 年 1 月 29 日

AL・FPAI, 大濱信泉記念館

本日の内容

パケットフィルタリング



ポリシーに従ってパケットをフィルタリング

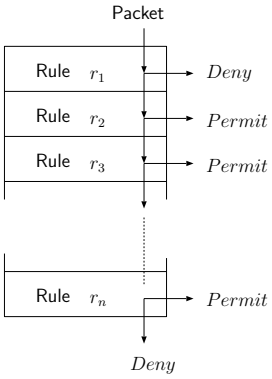
パケットフィルタリング

ポリシー : プログラムにおける**仕様**のようなもの
ルールリスト : プログラムにおける**実装**のようなもの

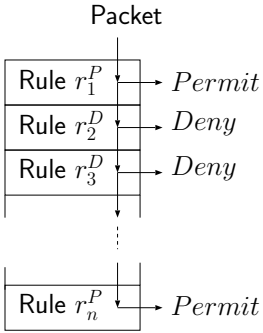
ポリシーを満たすルールリストを作成

ルールリストを線型探索してフィルタリング
($r_1, r_2, \ldots, r_n$ の順でパケットと照合)

ルール数が多いと遅延が発生
ルール数は数百 $\sim$ 数千



フィルタリングルールの形式



フィルタリングモデル

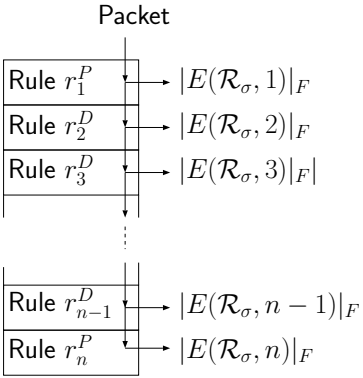
Filter $\mathcal{R}$
$r_1^P = * \; 0 \; * \; 1$
$r_2^P = 0 \; 0 \; 0 \; 0$
$r_3^P = 0 \; * \; 0 \; 0$
$r_4^D = 0 \; * \; 1 \; *$
$r_5^P = * \; 1 \; * \; 1$
$r_6^P = * \; * \; * \; 1$
$r_7^D = * \; * \; * \; *$

ルールリストの例

ルールの形式： $r_i^e = b_1 b_2 \ldots b_w,\ b_k \in \{0, 1, *\},\ e \in \{P, D\},\ i \in \mathbb{N}$

ルール番号 i はただの識別子，ルールリスト中の位置を表してはいない

ネットワーク機器の遅延



i 番目のルールによって評価型が決まる
パケットは i 回の照合を受ける

1 回の照合を遅延 1 と考え,
ネットワーク機器の遅延を定義

$$L(\mathcal{R}_\sigma, F) = \sum_{i}^{n-1} \sigma(i)|E(\mathcal{R}_\sigma, i)|_F + (n-1)|E(R_\sigma, n)|_F$$

σ はルールの順序，F はパケットの頻度分布

$|E(\mathcal{R}_\sigma, i)|_F$ は順序 σ においてルール r_i^e により評価型が決まるパケットの数

$|E(\mathcal{R}_\sigma, i)|_F$ は r_i に合致するパケットの数ではないことに注意

評価パケット数（ルール重み）

評価パケット数 $|E(\mathcal{R}_\sigma, i)|_F$

分布 F において，順序 σ のルールリスト $\mathcal{R}_\sigma$ の $1 \sim (\sigma(i)-1)$ 番目のルールに合致せず，r_i^e に合致するパケットの数

Filter $\mathcal{R}$	$\lvert E(\mathcal{R}_{id}, i)\rvert_F$
$r_1^P = * \; 0 \; * \; 1$	4
$r_2^P = 0 \; 0 \; 0 \; 0$	1
$r_3^P = 0 \; * \; 0 \; 0$	1
$r_4^D = 0 \; * \; 1 \; *$	3
$r_5^P = * \; 1 \; * \; 1$	3
$r_6^P = * \; * \; * \; 1$	0
$r_7^D = * \; * \; * \; *$	4
$L(\mathcal{R}_{id}, F) = 60$	

e.g. F：一様分布，σ：id

r_4^D に合致するパケットは

$$\{0010, 0011, 0110, 0111\}.$$

しかし，0011 は r_1^P に評価されるので，r_4^D によって評価型が決まるパケットは

$$\{0010, 0110, 0111\}$$

の 3 つ．以上より $|E(\mathcal{R}_{id}, 4)| = 3$

ルールリストとポリシー

Filter $\mathcal{R}_{id}$	$\lvert E(\mathcal{R}_{id}, i)\rvert_F$
$r_1^P = * \; 0 \; * \; 1$	4
$r_2^P = 0 \; 0 \; 0 \; 0$	1
$r_3^P = 0 \; * \; 0 \; 0$	1
$r_4^D = 0 \; * \; 1 \; *$	3
$r_5^P = * \; 1 \; * \; 1$	3
$r_6^P = * \; * \; * \; 1$	0
$r_7^D = * \; * \; * \; *$	4
$L(\mathcal{R}_{id}, F) = 60$	

Filter $\mathcal{R}_{\sigma}$	$\lvert E(\mathcal{R}_{\sigma}, i)\rvert_F$
$r_1^P = * \; 0 \; * \; 1$	4
$r_4^D = 0 \; * \; 1 \; *$	3
$r_5^P = * \; 1 \; * \; 1$	3
$r_3^P = 0 \; * \; 0 \; 0$	2
$r_2^P = 0 \; 0 \; 0 \; 0$	0
$r_6^P = * \; * \; * \; 1$	0
$r_7^D = * \; * \; * \; *$	4
$L(\mathcal{R}_{\sigma}, F) = 51$	

$$0000 \mapsto P, 0001 \mapsto P, 0010 \mapsto D, 0011 \mapsto P, 0100 \mapsto P, 0101 \mapsto P, 0110 \mapsto D, 0111 \mapsto D,$$
$$1000 \mapsto D, 1001 \mapsto P, 1010 \mapsto D, 1011 \mapsto P, 1100 \mapsto D, 1101 \mapsto P, 1110 \mapsto D, 1111 \mapsto P \quad (1)$$

フィルタ $\mathcal{R}_{id}$ と $\mathcal{R}_{\sigma=(1\ 5\ 4\ 2\ 3\ 6\ 7)}$ は同じポリシー (1) を表現

⇒ ルールを並び替えるとフィルタリング遅延を減らせる可能性有り

ルール順序最適化問題

ルール順序最適化問題

入力： ルールリスト $\mathcal{R}$，頻度分布 F

出力： $\sum_{i=1}^{n-1} \sigma(i)|E(\mathcal{R}_\sigma, i)|_F + (n-1)|E(\mathcal{R}_\sigma, n)|$ を最小化し，$\mathcal{R}$ のポリシーを維持するルールの順序 σ

Filter $\mathcal{R}_{id}$	$\lvert E(\mathcal{R}_{id}, i)\rvert_F$
$r_1^P = * \; 0 \; * \; 1$	4
$r_2^P = 0 \; 0 \; 0 \; 0$	1
$r_3^P = 0 \; * \; 0 \; 0$	1
$r_4^D = 0 \; * \; 1 \; *$	3
$r_5^P = * \; 1 \; * \; 1$	3
$r_6^P = * \; * \; * \; 1$	0
$r_7^D = * \; * \; * \; *$	4

Filter $\mathcal{R}_{\sigma}$	$\lvert E(\mathcal{R}_{\sigma}, i)\rvert_F$
$r_1^P = * \; 0 \; * \; 1$	4
$r_4^D = 0 \; * \; 1 \; *$	3
$r_5^P = * \; 1 \; * \; 1$	3
$r_3^P = 0 \; * \; 0 \; 0$	2
$r_2^P = 0 \; 0 \; 0 \; 0$	0
$r_6^P = * \; * \; * \; 1$	0
$r_7^D = * \; * \; * \; *$	4

σ によって，ルールの重み $E(\mathcal{R}_\sigma, i)$ が変動しうることに注意

重複関係 $\mathcal{O} \subseteq \mathcal{R} \times \mathcal{R}$ と従属関係 $\mathcal{D} \subseteq \mathcal{R} \times \mathcal{R}$

ルール順序最適化問題を考えるにあたっては，ルール間の重複関係と従属関係が重要

重複関係 $\mathcal{O} \subseteq \mathcal{R} \times \mathcal{R}$

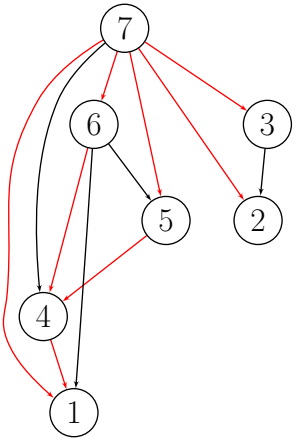
ルール $r_i^{e_i}, r_j^{e_j} \in \mathcal{R}$ **の両方に合致するパケット** $p \in \mathcal{P}$ **が存在するとき，** $r_i^{e_i}$ **と** $r_i^{e_i}$ **は重複するといい，** $\mathcal{O}(r_i, r_j)$ **と表す．**

従属関係 $\mathcal{D} \subseteq \mathcal{R} \times \mathcal{R}$

ルール $r_i^{e_i}, r_j^{e_j} \in \mathcal{R}$ **について，** $O(r_i, r_j) \ \wedge e_i \neq e_j \ \wedge \ i < j$ **のとき，** $r_j^{e_j}$ **は** $r_i^{e_i}$ **に従属するといい，** $\mathcal{D}(r_i, r_j)$ **と表す．**

重複関係 $\mathcal{O} \subseteq \mathcal{R} \times \mathcal{R}$ と従属関係 $\mathcal{D} \subseteq \mathcal{R} \times \mathcal{R}$

Filter $\mathcal{R}_{id}$	$\lvert E(\mathcal{R}_{id}, i)\rvert_F$
$r_1^P = * \; 0 \; * \; 1$	4
$r_2^P = 0 \; 0 \; 0 \; 0$	1
$r_3^P = 0 \; * \; 0 \; 0$	1
$r_4^D = 0 \; * \; 1 \; *$	3
$r_5^P = * \; 1 \; * \; 1$	3
$r_6^P = * \; * \; * \; 1$	0
$r_7^D = * \; * \; * \; *$	4



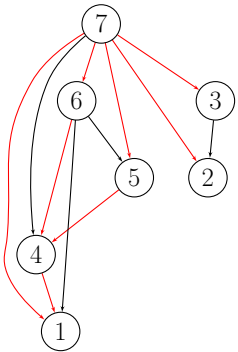
黒 ： 重複
赤 ： 従属

e.g. r_6^P **は** r_5^P **に重複，** r_6^P **は** r_4^D **に従属**

重複関係 $\mathcal{O} \subseteq \mathcal{R} \times \mathcal{R}$ と従属関係 $\mathcal{D} \subseteq \mathcal{R} \times \mathcal{R}$

Filter $\mathcal{R}_{id}$
$r_1^P = * \; 0 \; * \; 1$
$r_2^P = 0 \; 0 \; 0 \; 0$
$r_3^P = 0 \; * \; 0 \; 0$
$r_4^D = 0 \; * \; 1 \; *$
$r_5^P = * \; 1 \; * \; 1$
$r_6^P = * \; * \; * \; 1$
$r_7^D = * \; * \; * \; *$

黒 : 重複
赤 : 従属



1. $\forall \mathcal{R}.\ \big(\mathcal{O}(r_i, r_j) \Rightarrow \sigma(i) < \sigma(j)\big)$
   $\Rightarrow \sigma$ **はポリシーを保持**

2. $\forall \mathcal{R}.\ \big(\mathcal{D}(r_i, r_j) \Rightarrow \sigma(i) < \sigma(j)\big)$
   $\Rightarrow \sigma$ **はポリシーを保持**

以上より重複関係と従属関係は重要だが，正確には..

先行研究

順序 σ がポリシーを満たすことを下記のように定義したルール順序最適化問題に対する研究が存在

1. σ **はポリシーを保持** $\iff \forall i, j \ \left(\mathcal{O}(r_i, r_j) \Rightarrow \sigma(i) < \sigma(j)\right)$
   (Fulp 2005, Hamed 2006, El-Alfy 2007, Tapdiya 2009, **日景** 2016)
2. σ **はポリシーを保持** $\iff \forall i, j \ \left(\mathcal{D}(r_i, r_j) \Rightarrow \sigma(i) < \sigma(j)\right)$ (**田中** 2013)

けれども 1 も 2 も $\Rightarrow$ が成り立たない．即ち，

1. $\mathcal{O}(r_i, r_j)$ **はポリシー違反の必要条件であって十分条件ではない**
2. $\mathcal{D}(r_i, r_j) \ \wedge \ \sigma(j) < \sigma(i)$ **であってもポリシー違反を起こさない** $\mathcal{R}, \sigma$ **が存在**

$\mathcal{O}(r_i, r_j)$ でも $e_i \neq e_j$ ならば交換可能

Filter $\mathcal{R}_{id}$	$\lvert E(\mathcal{R}_{id}, i)\rvert_F$
$r_1^P = * \; 0 \; * \; 1$	4
$r_2^P = 0 \; 0 \; 0 \; 0$	1
$r_3^P = 0 \; * \; 0 \; 0$	1
$r_4^D = 0 \; * \; 1 \; *$	3
$r_5^P = * \; 1 \; * \; 1$	3
$r_6^P = * \; * \; * \; 1$	0
$r_7^D = * \; * \; * \; *$	4

Filter $\mathcal{R}_{\sigma}$	$\lvert E(\mathcal{R}_{\sigma}, i)\rvert_F$
$r_3^P = 0 \; * \; 0 \; 0$	2
$r_1^P = * \; 0 \; * \; 1$	4
$r_2^P = 0 \; 0 \; 0 \; 0$	0
$r_4^D = 0 \; * \; 1 \; *$	3
$r_5^P = * \; 1 \; * \; 1$	3
$r_6^P = * \; * \; * \; 1$	0
$r_7^D = * \; * \; * \; *$	4

0000 が r_2^P と r_3^P に合致するので，r_2^P と r_3^P は重複

$P = e_2 = e_3$ なので r_3^P は r_2^P よりも上位に配置 $(\sigma(j) < \sigma(i))$ 可能

$\mathcal{O}(r_i, r_j) \wedge \sigma(j) < \sigma(i)$ ならば重み変動の可能性

Filter $\mathcal{R}_{id}$	$\lvert E(\mathcal{R}_{id}, i)\rvert_F$
$r_1^P = * \ 0 \ * \ 1$	4
$r_2^P = 0 \ 0 \ 0 \ 0$	1
$r_3^P = 0 \ * \ 0 \ 0$	1
$r_4^D = 0 \ * \ 1 \ *$	3
$r_5^P = * \ 1 \ * \ 1$	3
$r_6^P = * \ * \ * \ 1$	0
$r_7^D = * \ * \ * \ *$	4

Filter $\mathcal{R}_{\sigma}$	$\lvert E(\mathcal{R}_{\sigma}, i)\rvert_F$
$r_3^P = 0 \ * \ 0 \ 0$	2
$r_1^P = * \ 0 \ * \ 1$	4
$r_2^P = 0 \ 0 \ 0 \ 0$	0
$r_4^D = 0 \ * \ 1 \ *$	3
$r_5^P = * \ 1 \ * \ 1$	3
$r_6^P = * \ * \ * \ 1$	0
$r_7^D = * \ * \ * \ *$	4

r_2^P で評価されていた 0000 が，r_3^P に評価されるので重み $E(\mathcal{R}, i)$ が変動

⇩

重複しているルール $r_i^{e_i}$ と $r_j^{e_j}$ を交換すると重みが変動する可能性あり

$\mathcal{D}(r_i, r_j) \wedge \sigma(j) < \sigma(i)$ でもポリシー違反にならない例

Filter $\mathcal{R}_{id}$	$\lvert E(\mathcal{R}_{id}, i)\rvert_F$
$r_1^P = * \; 0 \; * \; 1$	4
$r_2^P = 0 \; 0 \; 0 \; 0$	1
$r_3^P = 0 \; * \; 0 \; 0$	1
$r_4^D = 0 \; * \; 1 \; *$	3
$r_5^P = * \; 1 \; * \; 1$	3
$r_6^P = * \; * \; * \; 1$	0
$r_7^D = * \; * \; * \; *$	4

Filter $\mathcal{R}_{\sigma}$	$\lvert E(\mathcal{R}_{\sigma}, i)\rvert_F$
$r_1^P = * \; 0 \; * \; 1$	4
$r_3^P = 0 \; * \; 0 \; 0$	2
$r_4^D = 0 \; * \; 1 \; *$	3
$r_5^P = * \; 1 \; * \; 1$	3
$r_7^D = * \; * \; * \; *$	4
$r_2^P = 0 \; 0 \; 0 \; 0$	0
$r_6^P = * \; * \; * \; 1$	0

r_2^P と r_7^D は重複しており，$e_2 = P \neq D = e_7$ なので，r_7^D は r_2^P に従属

r_2^P により評価されるパケットはないので，r_7^D を r_2^P より上位に配置可能

大体の先行研究にある問題点と (Yuan 2008) らの手法

1. $\mathcal{O}(r_i, r_j) \Rightarrow \neg(\sigma(j) < \sigma(i))$ としていた
2. $\mathcal{D}(r_i, r_j) \Rightarrow \neg(\sigma(j) < \sigma(i))$ としていた
3. $e_i = e_j \wedge E(\mathcal{R}, i) \cap E(\mathcal{R}, j) \neq \emptyset \wedge \sigma(j) < \sigma(i)$ のとき，ルール重みの変動を無視していた

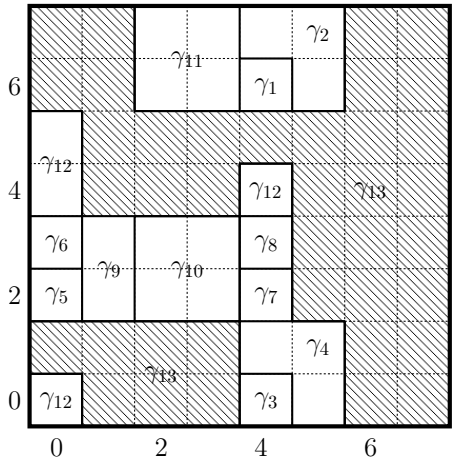
これに対して

(Yuan 2008) らは，$\mathcal{R}$ によりパケット空間 $\mathcal{P}$ を分割してブロック γ_i を生成 γ_i に基づいて，ルール順序最適化問題を上の 3 点を解決した整数計画問題として形式化

ルールリスト $\mathcal{R}$ によるパケット空間 $\mathcal{P}$ の分割

Filter $\mathcal{R}$
$r_1^P = 1\ 0 * 1\ 1 *$
$r_2^P = 1\ 0 * 0\ 0 *$
$r_3^D = * \ 0\ 0\ 0\ 1 *$
$r_4^D = 0 * * \ 0\ 1 *$
$r_5^P = 0\ 1 * * \ 1 *$
$r_6^P = * \ 0\ 0 * * \ 0$
$r_7^D = * * * * * *$

BDD によって集合 γ_i を表現



評価パケット数算出問題の計算複雑さ

デフォルトルール r_n の評価パケット数算出問題

入力：ルールリスト $\mathcal{R}$，頻度分布 F，順序 σ
出力：頻度分布 F のもとで r_n に合致するパケットの数 $|E(\mathcal{R}_\sigma, n)|_F$

頻度分布 F が一様のとき，この問題は $\#\mathcal{P}$-complete（原田 2018）

$n = 1000$ 程度のルールリストでは $E(\mathcal{R}_\sigma, i)$ を現実的な時間で計算可能

研究目的

ルール順序最適化問題に対する従来の研究は評価パケット数の変動を無視
Yuan らの手法は，厳密解法であって，n が大きいと手に負えない

⇩

ルール重みの変動を計算するアルゴリズムが必要
現実的なサイズのルールリストに対する発見的解法が必要

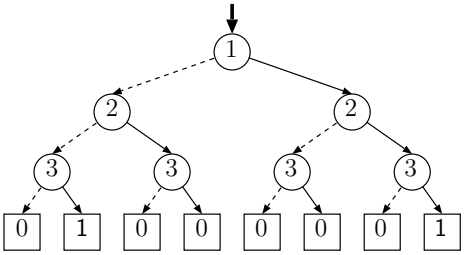
⇩

ZDD を用いたルール重みの変動を計算する手法を提案
これを用いた重み変動を考慮したルール順序最適化法に対する SA を提案

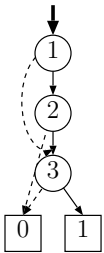
ZDD

ZDD (Zero-Suppressed Binary Decision Diagram)

疎な組合せ集合を効率よく扱えるデータ構造
ZDD 同士の和集合，差集合などの演算が存在



場合分け二分木

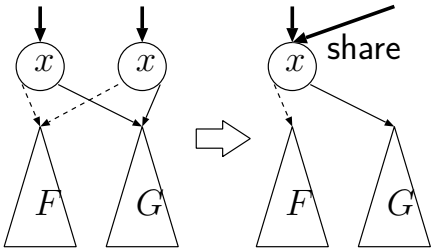


ZDD

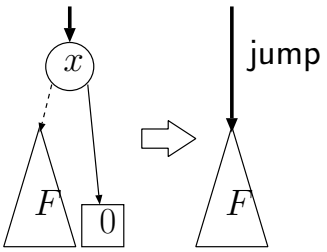
場合分け二分木と ZDD は組合せ集合 $\{001,\ 111\}$ を表現

ZDD の節点削除規則

論理函数（組合せ集合）に対する場合分け二分木に対して，以下の節点削除規則を既約になるまで適用すれば ZDD の出来上がり



等価節点の共有



冗長節点の削除

場合分け二分木を構築してから簡約することによって ZDD を構築するのは遅すぎるので，ZDD を高速に構築するための技法が数多存在

提案手法

ルール重みの変動を考慮したルール順序最適化問題に対する焼きなまし法

ルール重みの変動を考慮するためには，$M(F, r_i)$ と $E(\mathcal{R}_\sigma, F, i)$ が必要

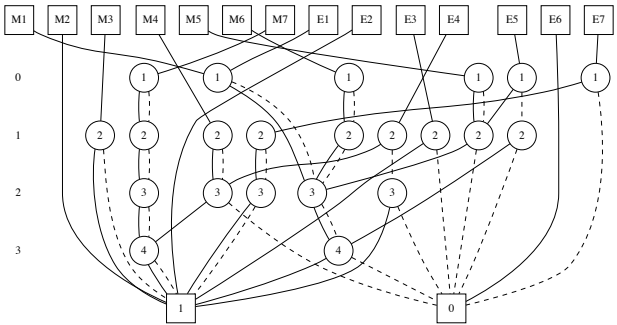
$M(r_i)$	ルール r_i^e に合致するパケットの集合
$M(F, r_i)$	頻度分布 F においてルール r_i^e に合致するパケットの集合
$E(\mathcal{R}_\sigma, i)$	順序 σ において，r_i^e に評価されるパケットの集合
$E(\mathcal{R}_\sigma, F, i)$	頻度分布 F，順序 σ において，r_i^e に評価されるパケットの集合

焼きなまし法を適用するためには，近傍 $N(\sigma)$，解 σ の評価函数が必要

$M(r_i)$ と $E(\mathcal{R}, i)$ に対する ZDDs

$M(r_i)$	ルール r_i^e に合致するパケットの集合
$M(F, r_i)$	頻度分布 F においてルール r_i^e に合致するパケットの集合
$E(\mathcal{R}_\sigma, i)$	順序 σ において，r_i^e に評価されるパケットの集合
$E(\mathcal{R}_\sigma, F, i)$	頻度分布 F，順序 σ において，r_i^e に評価されるパケットの集合

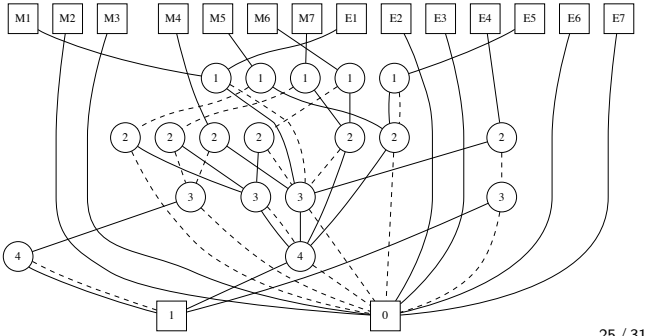
$F(p) = 1$



$M(r_i)$ と $E(\mathcal{R}, i)$ に対する ZDDs

$M(r_i)$	ルール r_i^e に合致するパケットの集合
$M(F, r_i)$	頻度分布 F においてルール r_i^e に合致するパケットの集合
$E(\mathcal{R}_\sigma, i)$	順序 σ において，r_i^e に評価されるパケットの集合
$E(\mathcal{R}_\sigma, F, i)$	頻度分布 F，順序 σ において，r_i^e に評価されるパケットの集合

$$F(p) = \begin{cases} 1 & \text{if } p \text{ is prime} \\ 0 & \text{otherwise} \end{cases}$$



焼きなまし法の適用（近傍）

解 σ に対して式 (2) の近傍を使用

$$N(\sigma) = \left\{ \sigma' \;\middle|\; \left| \{k \mid \sigma(k) \neq \sigma'(k)\} \right| = 2 \right\} \tag{2}$$

ランダムな順序 $o : [|N(\sigma)|] \to [|N(\sigma)|]$ を生成
順序 o で近傍 $N(\sigma)$ 内の改善解を探索

焼きなまし法の適用（評価函数）

i と j 番目を入れ替えた際の $i \sim j$ 番目のルールで評価されるパケット集合

$$
\begin{aligned}
&E\big(\mathcal{R}_{\tau(\sigma^{-1}(i),\sigma^{-1}(j))\circ\sigma}, \sigma^{-1}(i)\big) \\
&= E\big(\mathcal{R}_{\sigma}, \sigma^{-1}(j)\big) \cup \big(E\big(\mathcal{R}_{\sigma}, \sigma^{-1}(i)\big) \cap M\big(r_{\sigma^{-1}(j)}\big)\big) \cup \cdots \\
&\quad \cup \big(E\big(\mathcal{R}_{\sigma}, \sigma^{-1}(j-1)\big) \cap M\big(r_{\sigma^{-1}(j)}\big)\big)
\end{aligned} \tag{3}
$$

$$
\begin{aligned}
&E\big(\mathcal{R}_{\tau(\sigma^{-1}(i),\sigma^{-1}(j))\circ\sigma}, \sigma^{-1}(j)\big) \\
&= E\big(\mathcal{R}_{\sigma}, \sigma^{-1}(i)\big) \setminus M\big(r_{\sigma^{-1}(i+1)}\big) \setminus \cdots \setminus M\big(r_{\sigma^{-1}(j)}\big)
\end{aligned} \tag{4}
$$

$$
\begin{aligned}
&E\big(\mathcal{R}_{\tau(\sigma^{-1}(i),\sigma^{-1}(j))\circ\sigma}, \sigma^{-1}(k)\big) \\
&= E\big(\mathcal{R}_{\sigma}, \sigma^{-1}(k)\big) \setminus M\big(r_{\sigma^{-1}(j)}\big) \cup \big(E\big(\mathcal{R}_{\sigma}, \sigma^{-1}(i)\big) \setminus M\big(r_{\sigma^{-1}(i+1)}\big) \setminus \\
&\quad \cdots \setminus M\big(r_{\sigma^{-1}(k-1)}\big)\big) \cap M\big(r_{\sigma^{-1}(k)}\big)
\end{aligned} \tag{5}
$$

焼きなまし法の適用（評価函数）

解 σ と σ' の比較函数 h

$$h(\sigma, \sigma') = \sum_{k=i}^{j} k \cdot \left(|E(\mathcal{R}_\sigma, \sigma^{-1}(k))|_F - |E(\mathcal{R}_{\sigma'}, \sigma^{-1}(k))|_F \right)$$

ただし， $\sigma' = \tau_{(\sigma^{-1}(i), \sigma^{-1}(j))} \circ \sigma$

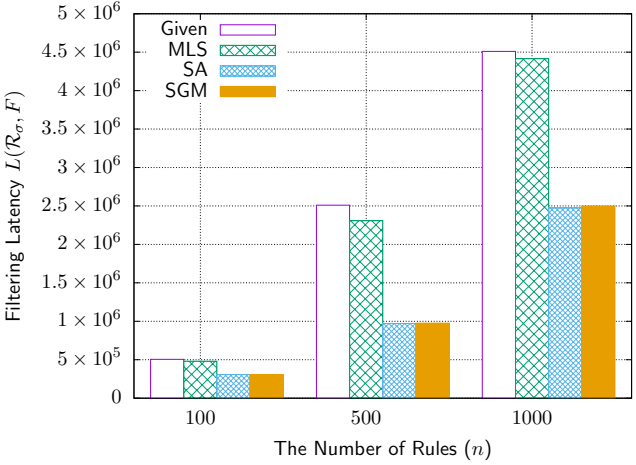
原稿 $|E(\mathcal{R}_\sigma, k)|_F$ となっていますが，正しくは $|E(\mathcal{R}_\sigma, \sigma^{-1}(k))|_F$ です
間違えていました．すみません

実験環境

OS	:	Mac OSX 10.9.5
CPU	:	Intel Core i5 1.4 GHz
主記憶容量	:	4GB
実装言語	:	C++
ZDD **ライブラリ**	:	CUDD

- **ルール長** 120**，ルール数が百** $\sim$ **千のルールリストを** ClassBench **で生成**
- **単純局所探索法，焼きなまし法，**SGM(Tapdiya 2009) **を実装**
- **それぞれの遅延** $(L(\mathcal{R}_\sigma, F))$ **を計測**

実験結果



SA は既存手法において平均的に最良の手法である SGM よりも遅延を削減

まとめと今後の課題

まとめ

- ルール順序最適化問題では重みの変動に考慮が必要
- 重みが変動するルール順序最適化問題に対する焼きなまし法を提案

今後の課題

- 重みの計算の高速化による，提案手法の高速化
- $\mathcal{D}(r_i, r_j)$ でも，$\sigma(j) < \sigma(i)$ が可能かを高速に判定する手法の提案
- ルール順序最適化問題に対する厳密解法の提案 $(n > 141)$