

Utility-Peak Itemset Mining with Constraint Programming

université
PARIS-SACLAY
SPATTR-2025-12

 FIDD ANR-24-CE23-0711  anr[©]

Nadjib Lazaar

Paris-Saclay University — lazaar@lisn.fr

C. Hamdi, N. Belmecheri, D. Bekkoucha, S. Jabbour, L. Sais



CP26 — Lisbon — 23/07/1826



HIGH UTILITY ITEMSET MINING (HUIM)

.....

t1:	A		C	D	
t2:	A	B			E
t3:		B	C	D	E
t4:	A		C		E

HIGH UTILITY ITEMSET MINING (HUIM)

t1: A(1) C(2) D(1)

t2: A(3) B(1) E(2)

t3: B(2) C(1) D(4) E(1)

t4: A(1) C(3) E(2)

HIGH UTILITY ITEMSET MINING (HUIM)

.....

t1:	A(1)		C(2)	D(1)	
t2:	A(3)	B(1)			E(2)
t3:		B(2)	C(1)	D(4)	E(1)
t4:	A(1)		C(3)		E(2)
Utility	4	2	1	3	2

HIGH UTILITY ITEMSET MINING (HUIM)

t1:	A(1)		C(2)	D(1)	
t2:	A(3)	B(1)			E(2)
t3:		B(2)	C(1)	D(4)	E(1)
t4:	A(1)		C(3)		E(2)
Utility	4	2	1	3	2

HIGH UTILITY ITEMSET MINING (HUIM)

t1: A(1) C(2) D(1)

t2: A(3) B(1) E(2)

t3: B(2) C(1) D(4) E(1)

t4: A(1) C(3) E(2)

Utility	4	2	1	3	2

$$freq(AC) = 2$$

$$u(AC) = u(AC, t_1) + u(AC, t_4)$$

$$u(AC) = 6 + 7 = 13$$

HIGH UTILITY ITEMSET MINING (HUIM)

- Aims at finding regularities in transactional databases
- **Given:**
 - Items $I = \{i_1, \dots, i_n\}$ **with utility values**
 - Transactions $T = \{t_1, \dots, t_m\}$ **with item quantities**
 - Minimum utility threshold θ
- **The need:**
 - All high-utility itemsets (HUI) satisfying utility $\geq \theta$

SPECIALIZED APPROACHES — AND THEIR LIMITATIONS

- **Specialized algorithms:**
 - **Two-Phase / IHUP / UP-Growth:** Candidate generation + TWU pruning (Transaction-Weighted Utility) [Liu et al., 2005, Ahmed et al., 2009; Tseng et al., 2013]
 - **HUI-Miner / D2HUP:** Vertical structures (utility-lists) [Liu et al., 2012; Liu et al., 2012]
 - **EFIM / FHM:** Projection + remaining utility pruning [Zida et al., 2015; Fournier-Viger et al., 2014]
- **Limitations:**
 - **Combinatorial explosion:** Low threshold $\theta \rightarrow$ millions of extracted HUIs
 - **Interpretability:** A massive output is impractical to use
 - **Redundancy:** Many HUIs overlap or are poorly informative
 - **Rigidity:** User constraints require algorithm redesign

SPECIALIZED APPROACHES — AND THEIR LIMITATIONS

- **Specialized algorithms:**

- **Two-Phase / IHUP / UP-Growth:** Candidate generation + TWU pruning (Transaction-Weighted Utility) [Liu et al., 2005, Ahmed et al., 2009; Tseng et al., 2013]
- **HUI-Miner / D2HUP:** Vertical structures (utility-lists) [Liu et al., 2012; Liu et al., 2012]
- **EFIM / FHM:** Projection + remaining utility pruning [Zida et al., 2015; Fournier-Viger et al., 2014]

- **Limitations:**

Condensed Representation

- **Combinatorial explosion:** Low threshold $\theta \rightarrow$ millions of extracted HUIs
- **Interpretability:** A massive output is impractical to use
- **Redundancy:** Many HUIs overlap or are poorly informative
- **Rigidity:** User constraints require algorithm redesign

SPECIALIZED APPROACHES — AND THEIR LIMITATIONS

- **Specialized algorithms:**

- **Two-Phase / IHUP / UP-Growth:** Candidate generation + TWU pruning (Transaction-Weighted Utility) [Liu et al., 2005, Ahmed et al., 2009; Tseng et al., 2013]
- **HUI-Miner / D2HUP:** Vertical structures (utility-lists) [Liu et al., 2012; Liu et al., 2012]
- **EFIM / FHM:** Projection + remaining utility pruning [Zida et al., 2015; Fournier-Viger et al., 2014]

- **Limitations:**

Condensed Representation

- **Combinatorial explosion:** Low threshold $\theta \rightarrow$ millions of extracted HUIs
- **Interpretability:** A massive output is impractical to use
- **Redundancy:** Many HUIs overlap or are poorly informative

- **Rigidity:** User constraints require algorithm redesign

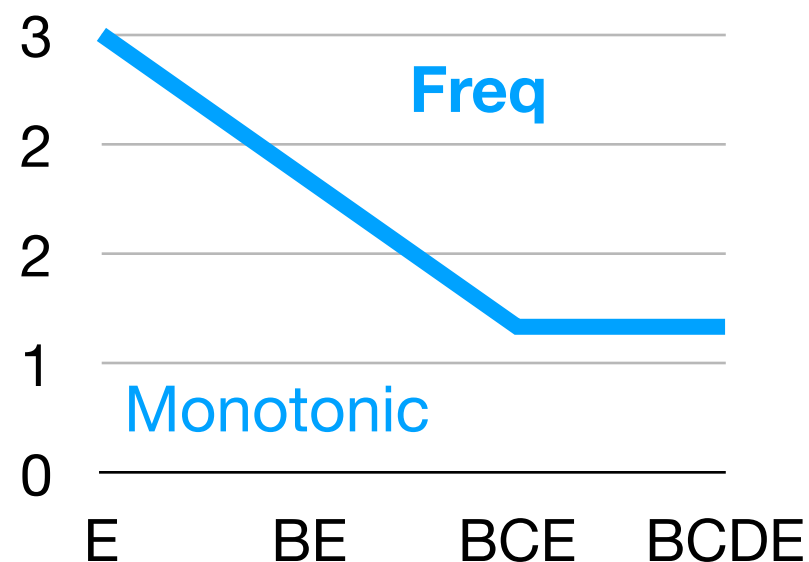
Declarative Approach

EXISTING CONDENSED REPRESENTATIONS

- **Maximal/Closed Itemsets:**
 - Based on support closure
 - **Criterion:** no superset with the same support
 - **X** Structural criterion, not utility-related
 - **X** Structural maximality $\neq$ maximum utility

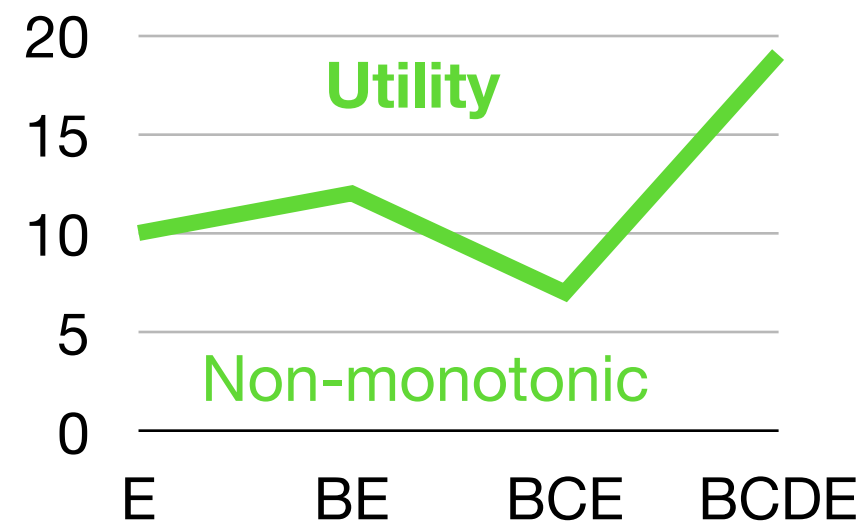
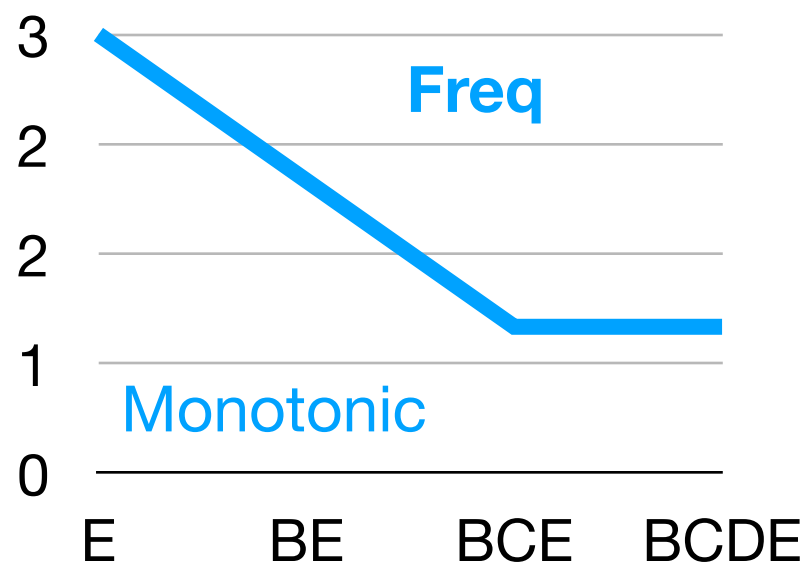
EXISTING CONDENSED REPRESENTATIONS

- **Maximal/Closed Itemsets:**
 - Based on support closure
 - **Criterion:** no superset with the same support
 - **✗** Structural criterion, not utility-related
 - **✗** Structural maximality $\neq$ maximum utility



EXISTING CONDENSED REPRESENTATIONS

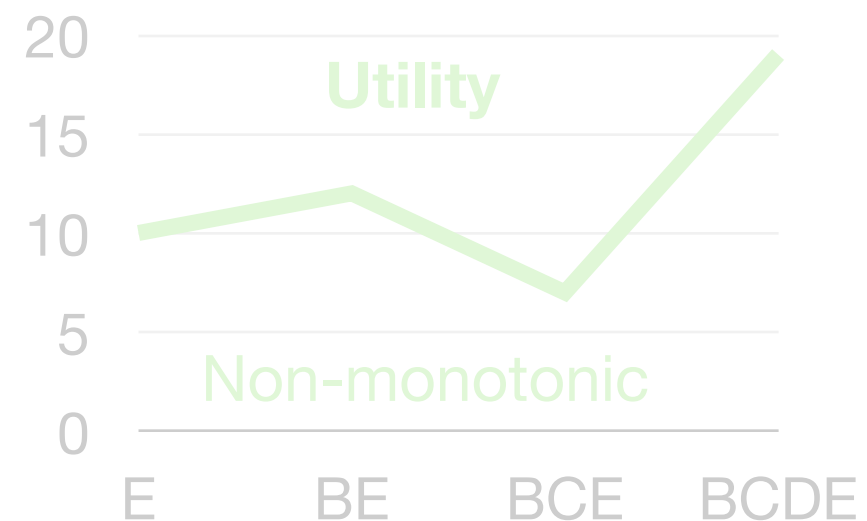
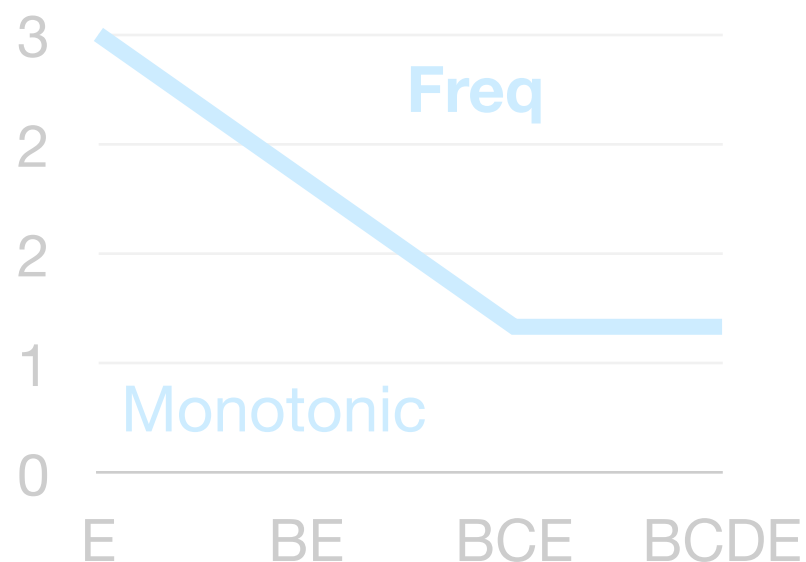
- **Maximal/Closed Itemsets:**
 - Based on support closure
 - **Criterion:** no superset with the same support
 - **✗** Structural criterion, not utility-related
 - **✗** Structural maximality $\neq$ maximum utility



EXISTING CONDENSED REPRESENTATIONS

- Maximal/Closed Itemsets:
 - Based on support closure
 - **Criterion:** no superset with the same support
 - ✗ Structural criterion, not utility-related

No existing representation captures utility-based maximality



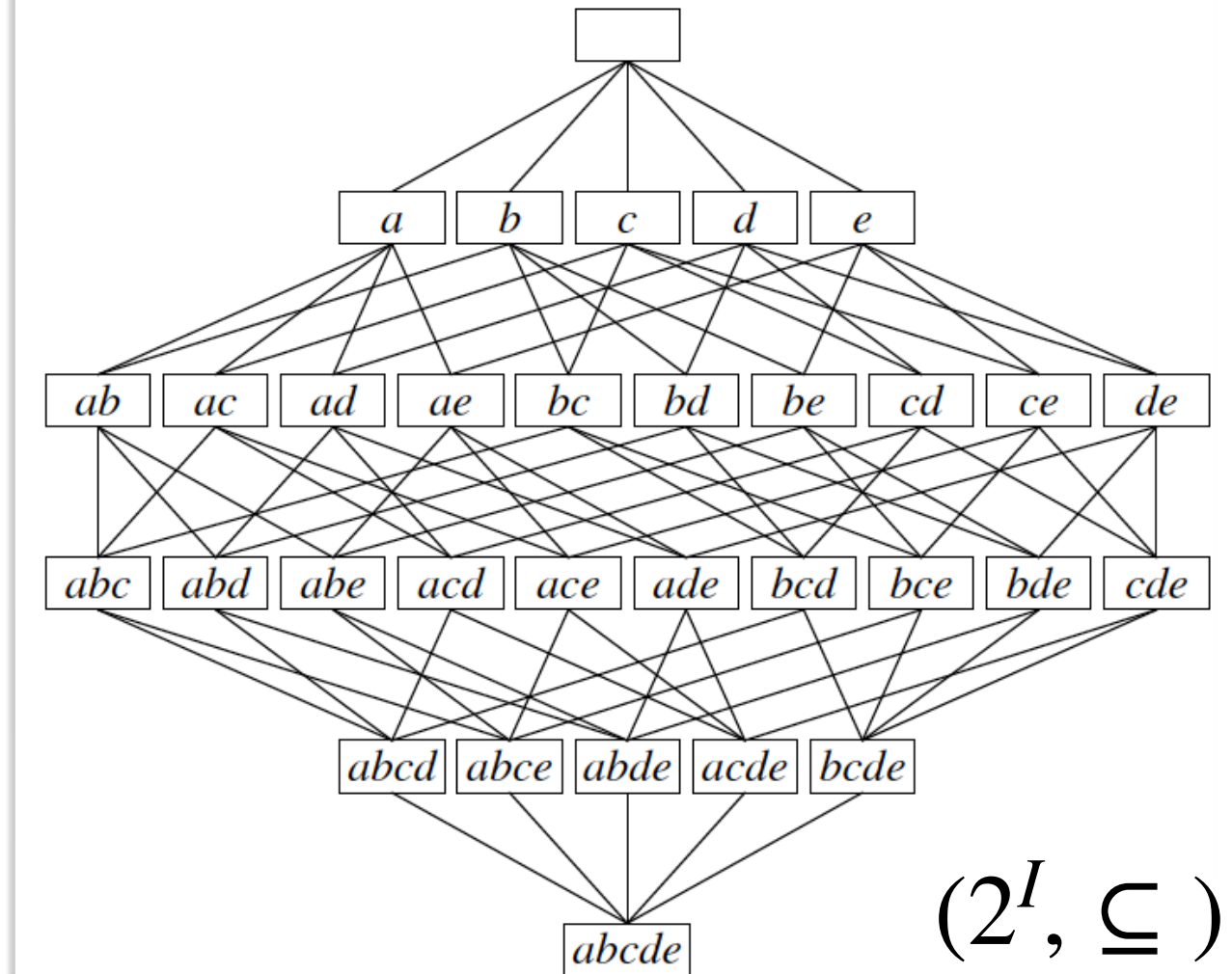
UTILITY-MAXIMAL ITEMSET (UMI)

- **An itemset is Utility-Maximal iff:**
 - $\forall Q \supset P : u(Q) \leq u(P)$
 - $\forall Q \subset P : u(Q) \leq u(P)$

UTILITY-MAXIMAL ITEMSET (UMI)

- An itemset is Utility-Maximal iff:

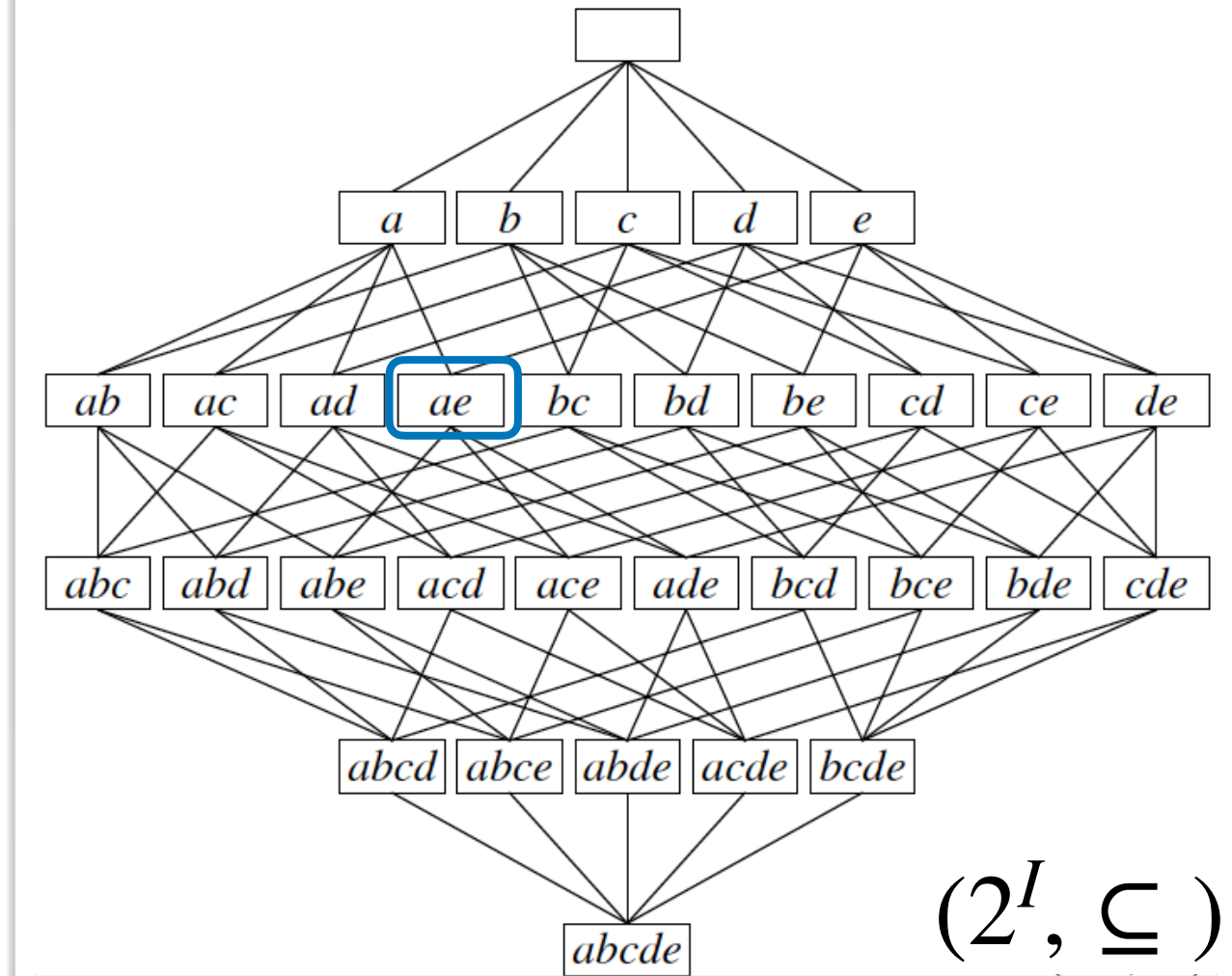
- $\forall Q \supset P : u(Q) \leq u(P)$
- $\forall Q \subset P : u(Q) \leq u(P)$



UTILITY-MAXIMAL ITEMSET (UMI)

- An itemset is Utility-Maximal iff:

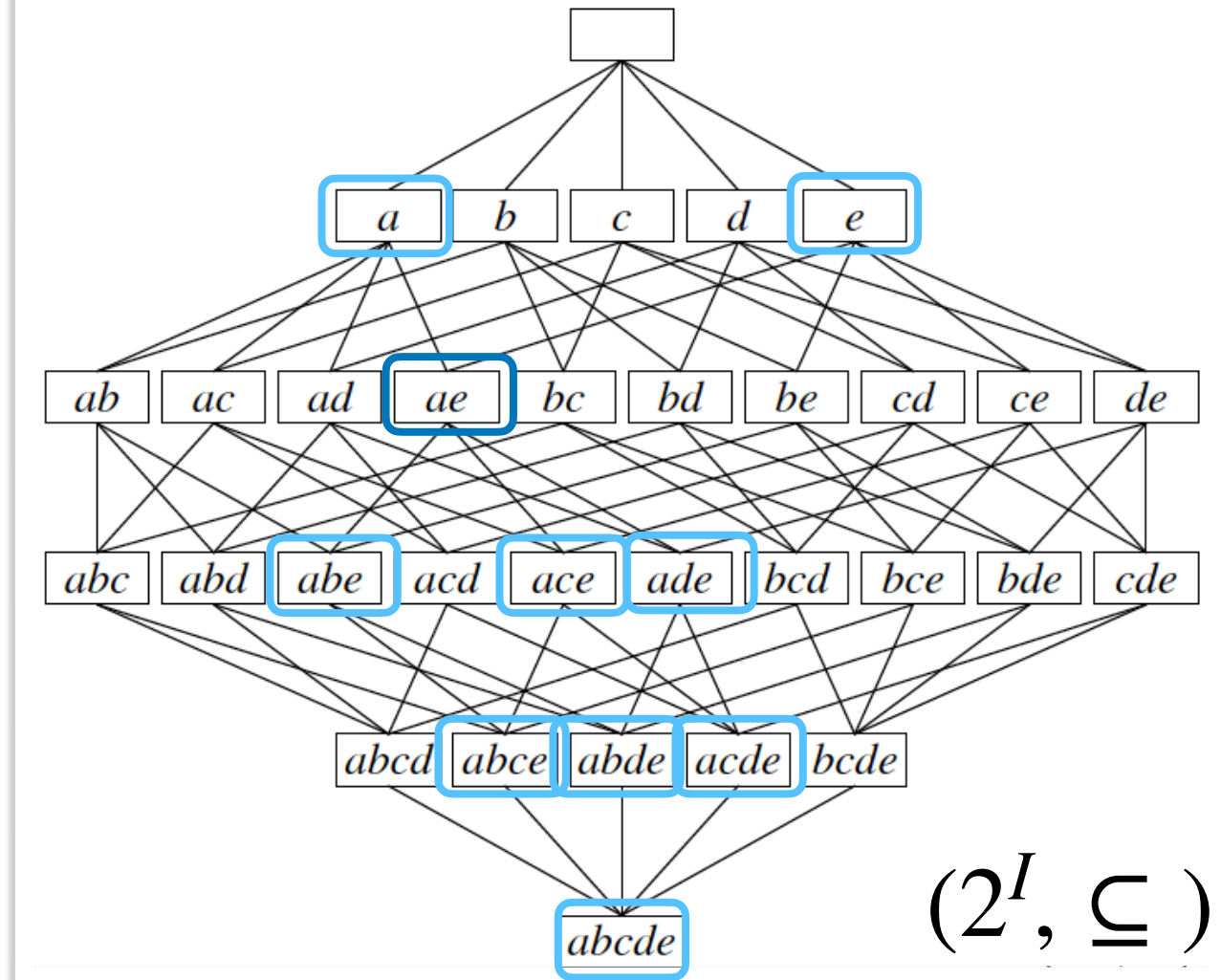
- $\forall Q \supset P : u(Q) \leq u(P)$
- $\forall Q \subset P : u(Q) \leq u(P)$



UTILITY-MAXIMAL ITEMSET (UMI)

- An itemset is Utility-Maximal iff:

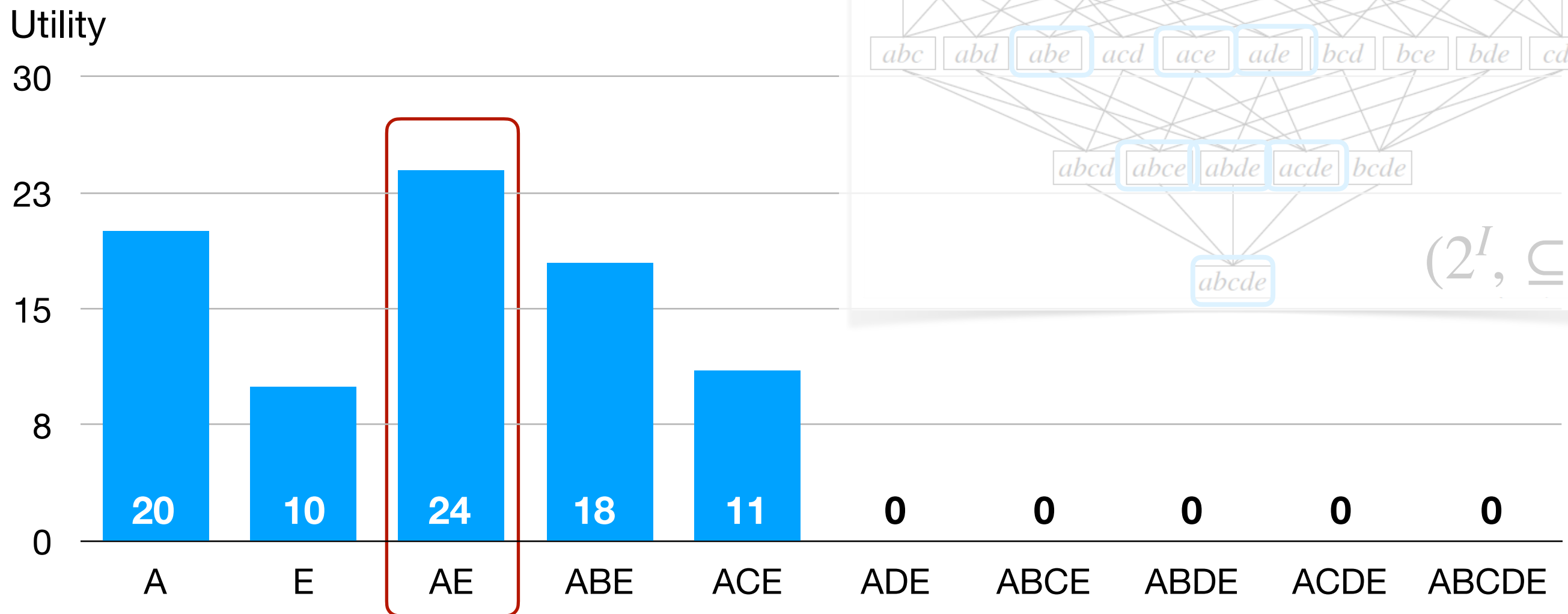
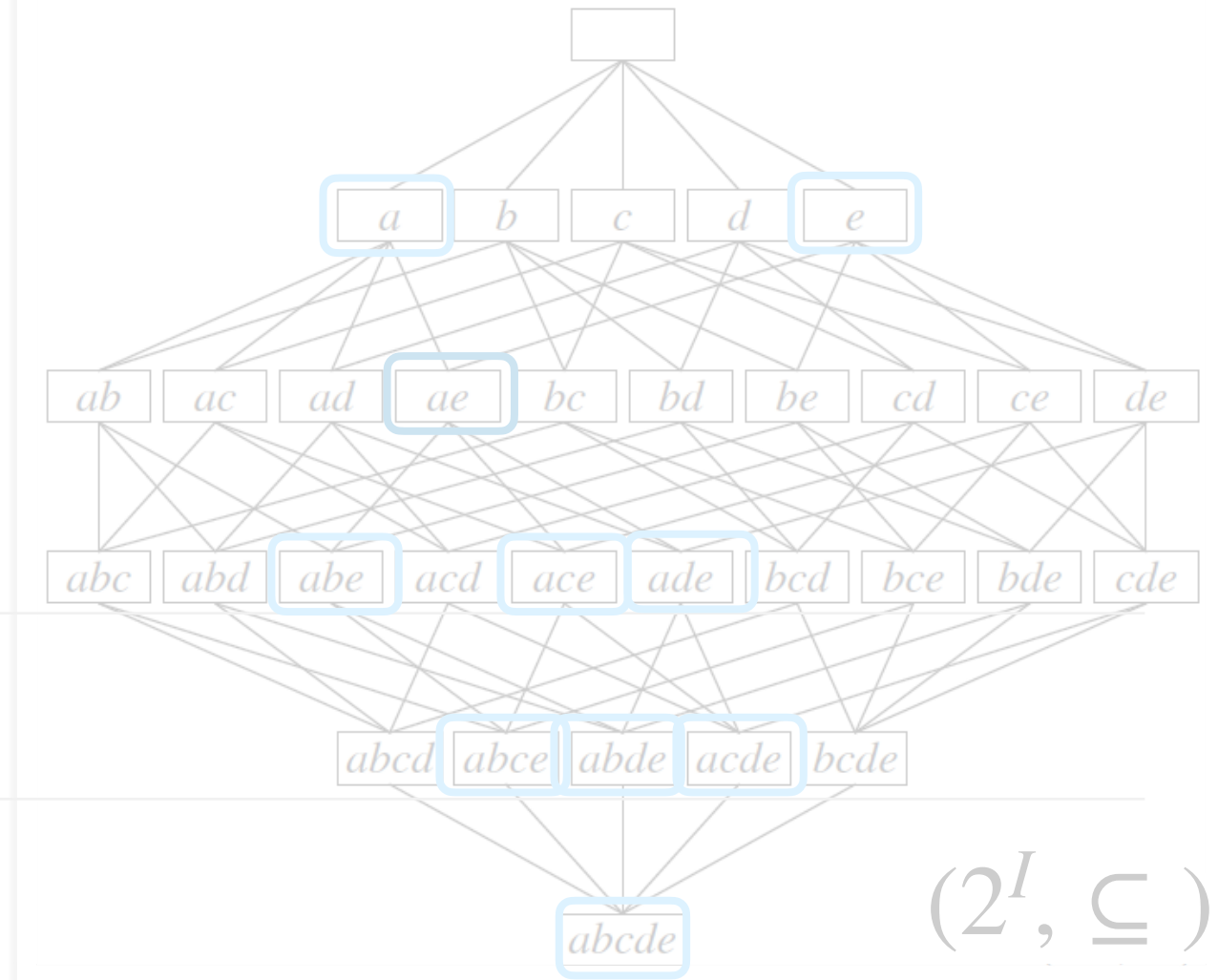
- $\forall Q \supset P : u(Q) \leq u(P)$
- $\forall Q \subset P : u(Q) \leq u(P)$



UTILITY-MAXIMAL ITEMSET (UMI)

- An itemset is **Utility-Maximal** iff:

- $\forall Q \supset P : u(Q) \leq u(P)$
- $\forall Q \subset P : u(Q) \leq u(P)$



UTILITY-PEAK ITEMSET (UPI)

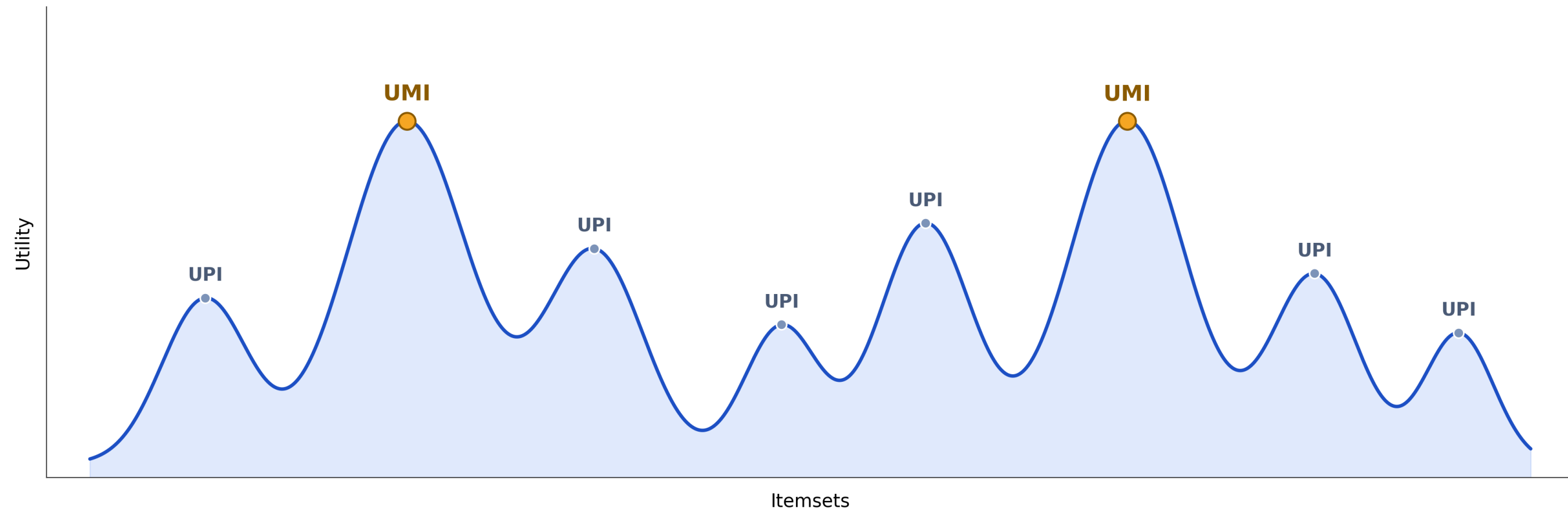
- **An itemset is Utility-Peak iff:**

- $\forall i \in P : u(P \setminus \{i\}) \leq u(P)$
- $\forall i \notin P : u(P \cup \{i\}) \leq u(P)$

UTILITY-PEAK ITEMSET (UPI)

- An itemset is **Utility-Peak** iff:

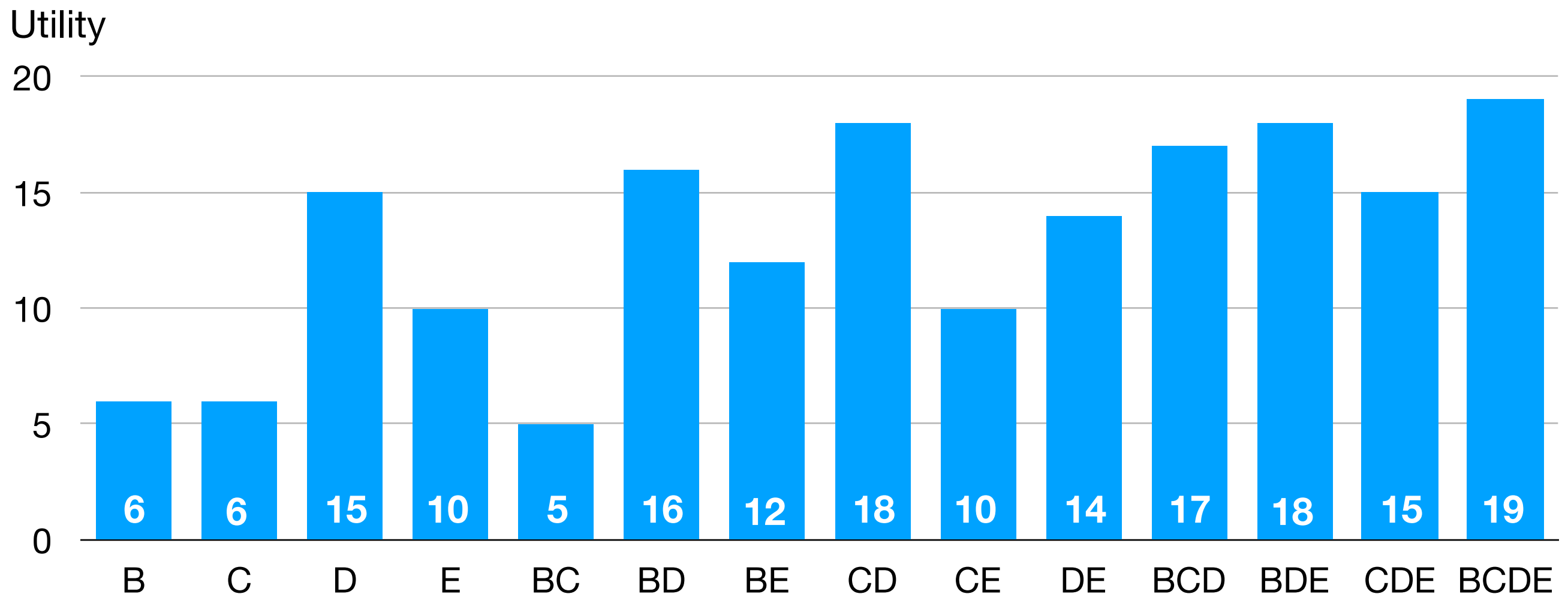
- $\forall i \in P : u(P \setminus \{i\}) \leq u(P)$
- $\forall i \notin P : u(P \cup \{i\}) \leq u(P)$



UTILITY-PEAK ITEMSET (UPI)

- An itemset is Utility-Peak iff:

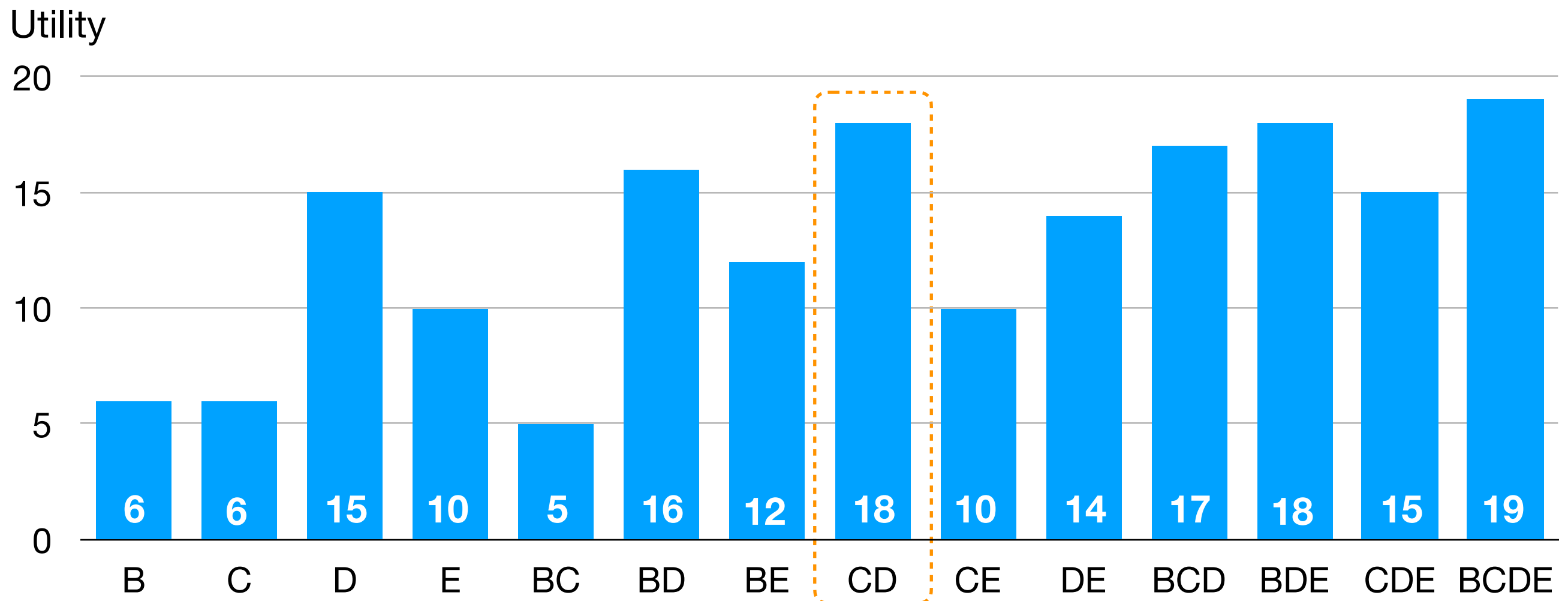
- $\forall i \in P : u(P \setminus \{i\}) \leq u(P)$
- $\forall i \notin P : u(P \cup \{i\}) \leq u(P)$



UTILITY-PEAK ITEMSET (UPI)

- An itemset is **Utility-Peak** iff:

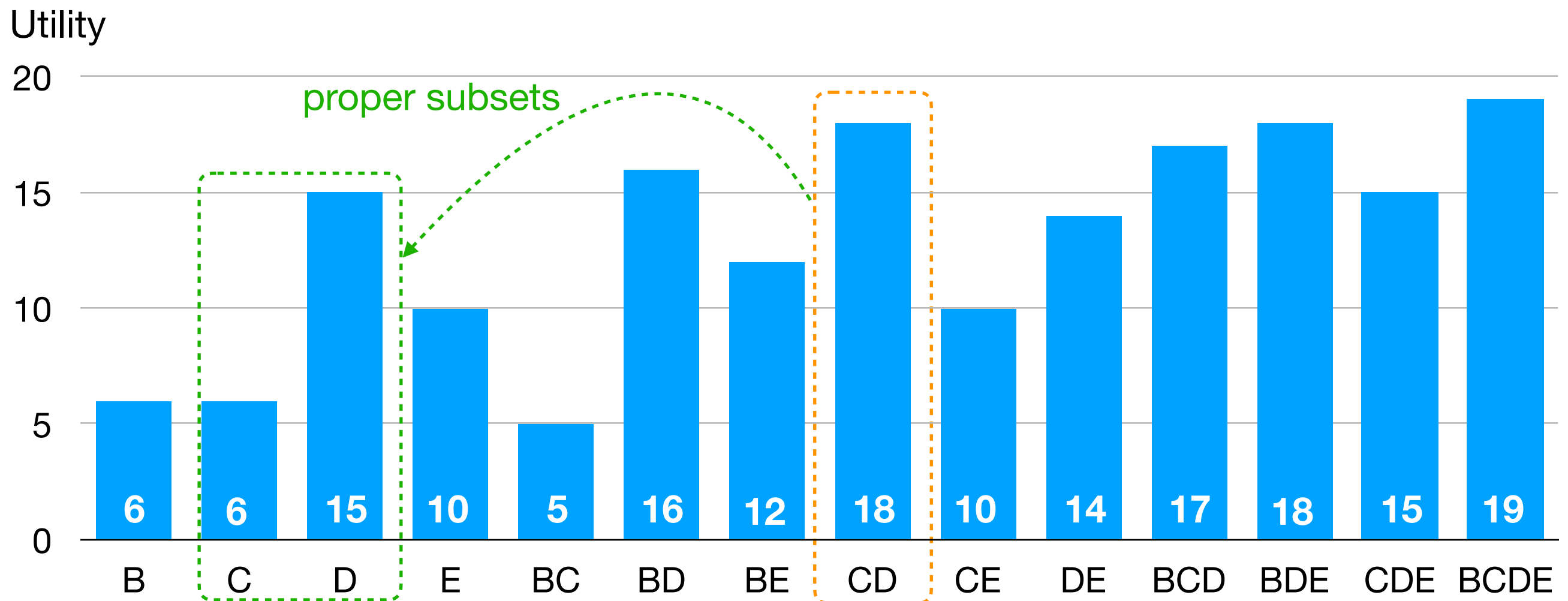
- $\forall i \in P : u(P \setminus \{i\}) \leq u(P)$
- $\forall i \notin P : u(P \cup \{i\}) \leq u(P)$



UTILITY-PEAK ITEMSET (UPI)

- An itemset is **Utility-Peak** iff:

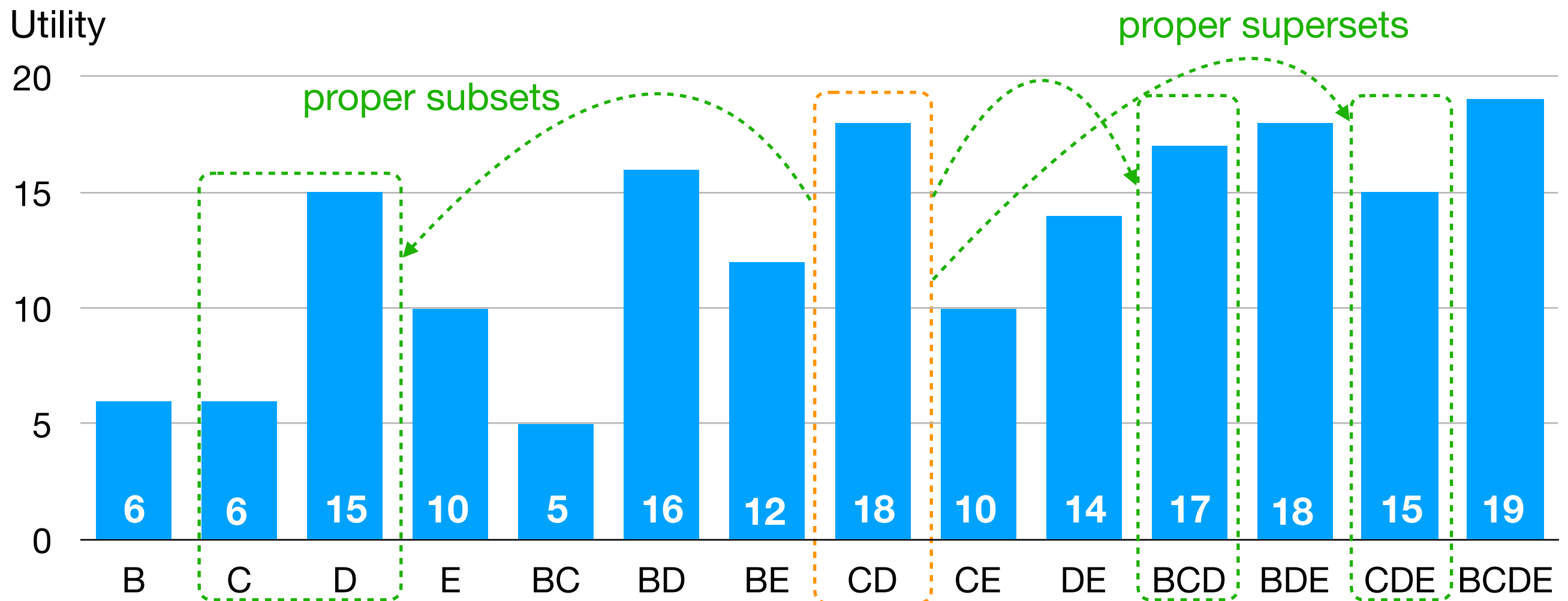
- $\forall i \in P : u(P \setminus \{i\}) \leq u(P)$
- $\forall i \notin P : u(P \cup \{i\}) \leq u(P)$



UTILITY-PEAK ITEMSET (UPI)

- An itemset is **Utility-Peak** iff:

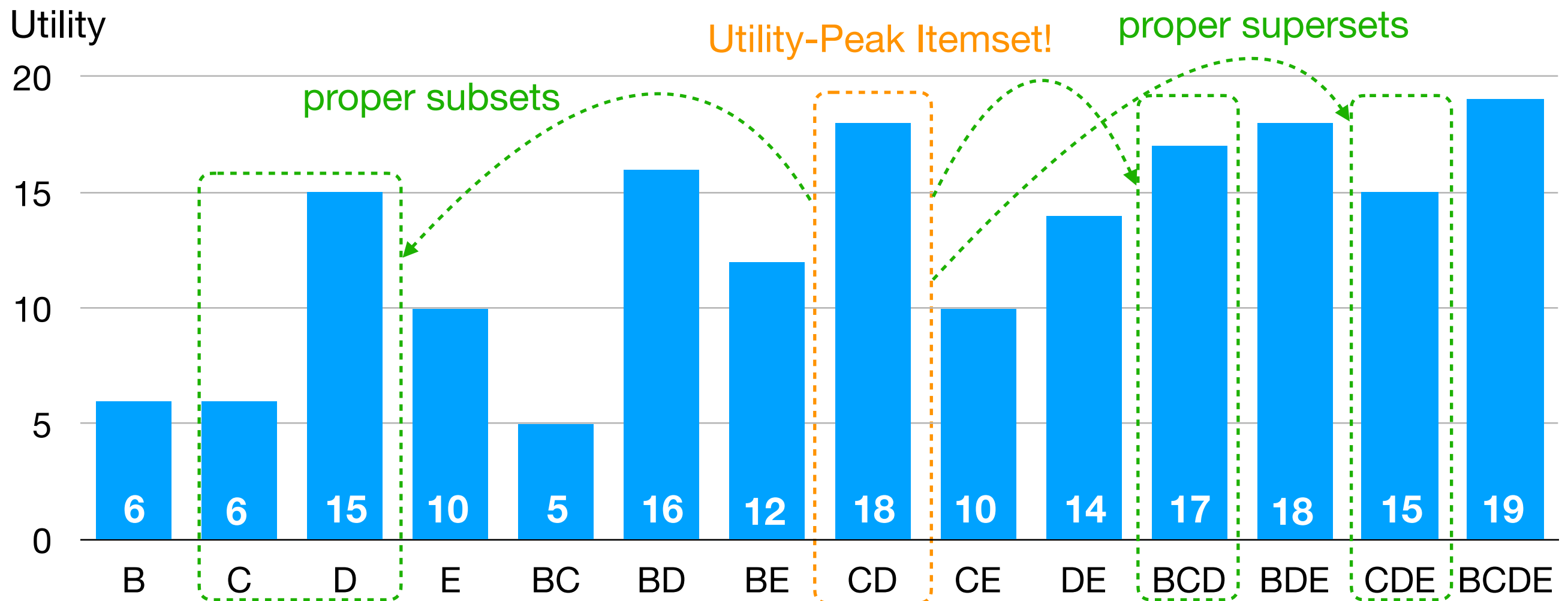
- $\forall i \in P : u(P \setminus \{i\}) \leq u(P)$
- $\forall i \notin P : u(P \cup \{i\}) \leq u(P)$



UTILITY-PEAK ITEMSET (UPI)

- An itemset is **Utility-Peak** iff:

- $\forall i \in P : u(P \setminus \{i\}) \leq u(P)$
- $\forall i \notin P : u(P \cup \{i\}) \leq u(P)$



UTILITY-PEAK ITEMSET (UPI)

- An itemset is **Utility-Peak** iff:
 - $\forall i \in P : u(P \setminus \{i\}) \leq u(P)$
 - $\forall i \notin P : u(P \cup \{i\}) \leq u(P)$
- **Local maximality:** A UPI is locally optimal in the lattice: neither adding nor removing an item can improve utility
- **Complementary to HUIs:** The result is much more compact and actionable
- **Complexity [Th10-13]:**
 - Existence of a UPI: always true (P)
 - Find a UPI : PLS-complete (from Max-2SAT-Flip)
 - Existence of a θ -UPI: NP-complete (from One-in-Three-3SAT)

UTILITY-PEAK ITEMSET (UPI)

- An itemset is Utility-Peak iff:

- $\forall i \in P : u(P \setminus \{i\}) \leq u(P)$
- $\forall i \notin P : u(P \cup \{i\}) \leq u(P)$

- **Local maximality:** A UPI is locally optimal in the lattice: neither adding nor removing items increases utility

**UPIs capture locally utility-optimal patterns,
regardless of size or support**

- **Conciseness:** active

- **Complexity [Th10-13]:**

- Existence of a UPI: always true (P)
- Find a UPI : PLS-complete (from Max-2SAT-Flip)
- Existence of a θ -UPI: NP-complete (from One-in-Three-3SAT)

A GENERAL CP MODEL FOR HUI AND UPI

- Reified Decomposition

- **Variables:** $x_i \in \{0,1\} (\forall i \in I)$ and $b_{it} \in \{0,1\} (\forall i \in I, t \in T)$

- **HUI Constraints:**

- $b_{it} \leftrightarrow (x_i = 1 \wedge i \in t) \quad \forall i \in I, t \in T$

- $\sum_{t \in T} u_t \geq \theta \quad \text{where } u_t = \sum_{i \in t} b_{it} \cdot u(i, t)$

- **UPI Constraints:**

- $u(x^{-1}(1) \setminus \{i\}) \leq u(x^{-1}(1)) \quad \forall i \in x^{-1}(1)$

- $u(x^{-1}(1) \cup \{i\}) \leq u(x^{-1}(1)) \quad \forall i \in I \setminus (x^{-1}(1) \cup x^{-1}(0))$

A GENERAL CP MODEL FOR HUI AND UPI

- Reified Decomposition

- Variables: $x_i \in \{0,1\} (\forall i \in I)$ and $b_{it} \in \{0,1\} (\forall i \in I, t \in T)$

- HUI Constraints:

- $b_{it} \leftrightarrow (x_i = 1 \wedge i \in t) \quad \forall i \in I, t \in T$

- - Large Number of Auxiliary Variables

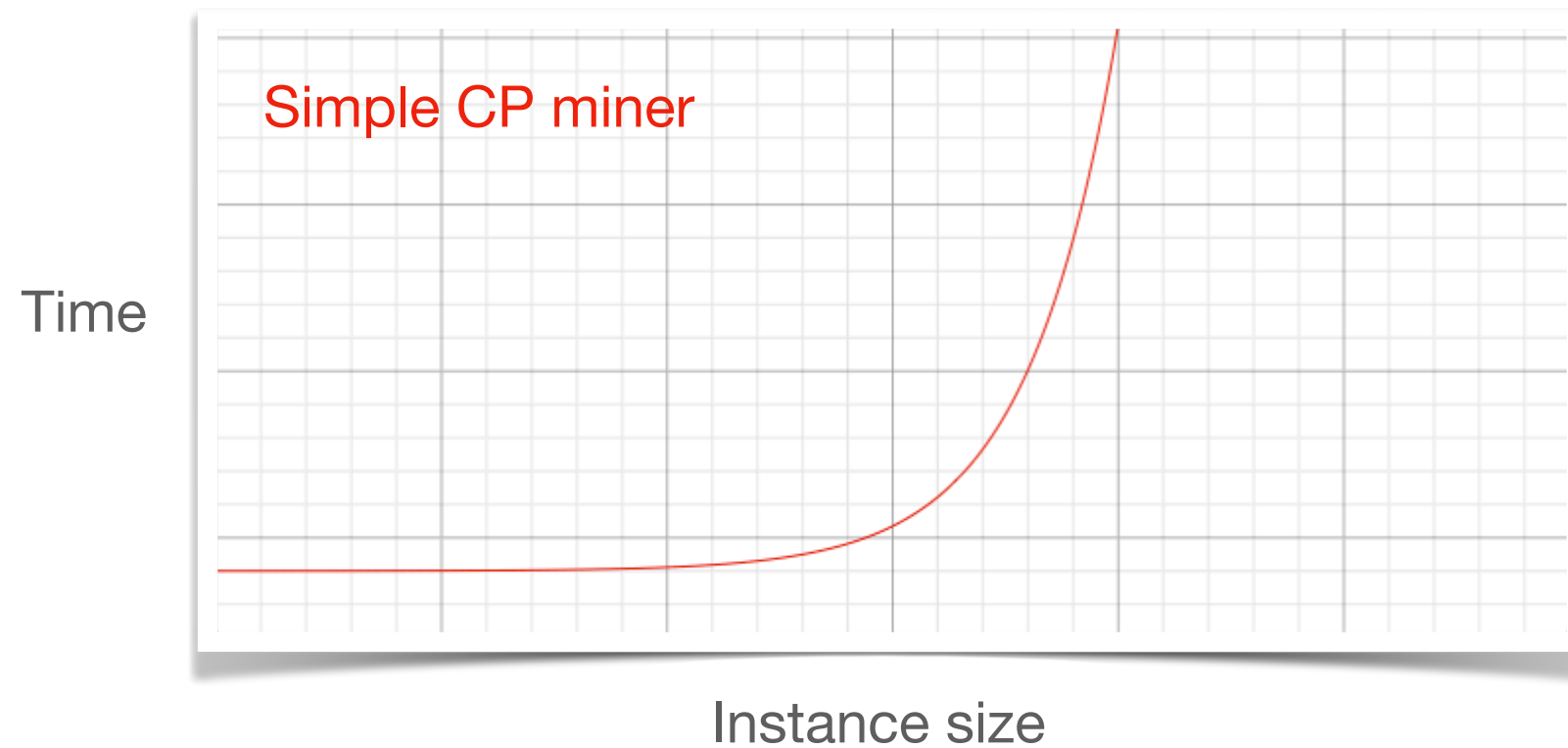
- - High number of reified constraints with significant arity

- UPI Constraints:

- $u(x^{-1}(1) \setminus \{i\}) \leq u(x^{-1}(1)) \quad \forall i \in x^{-1}(1)$

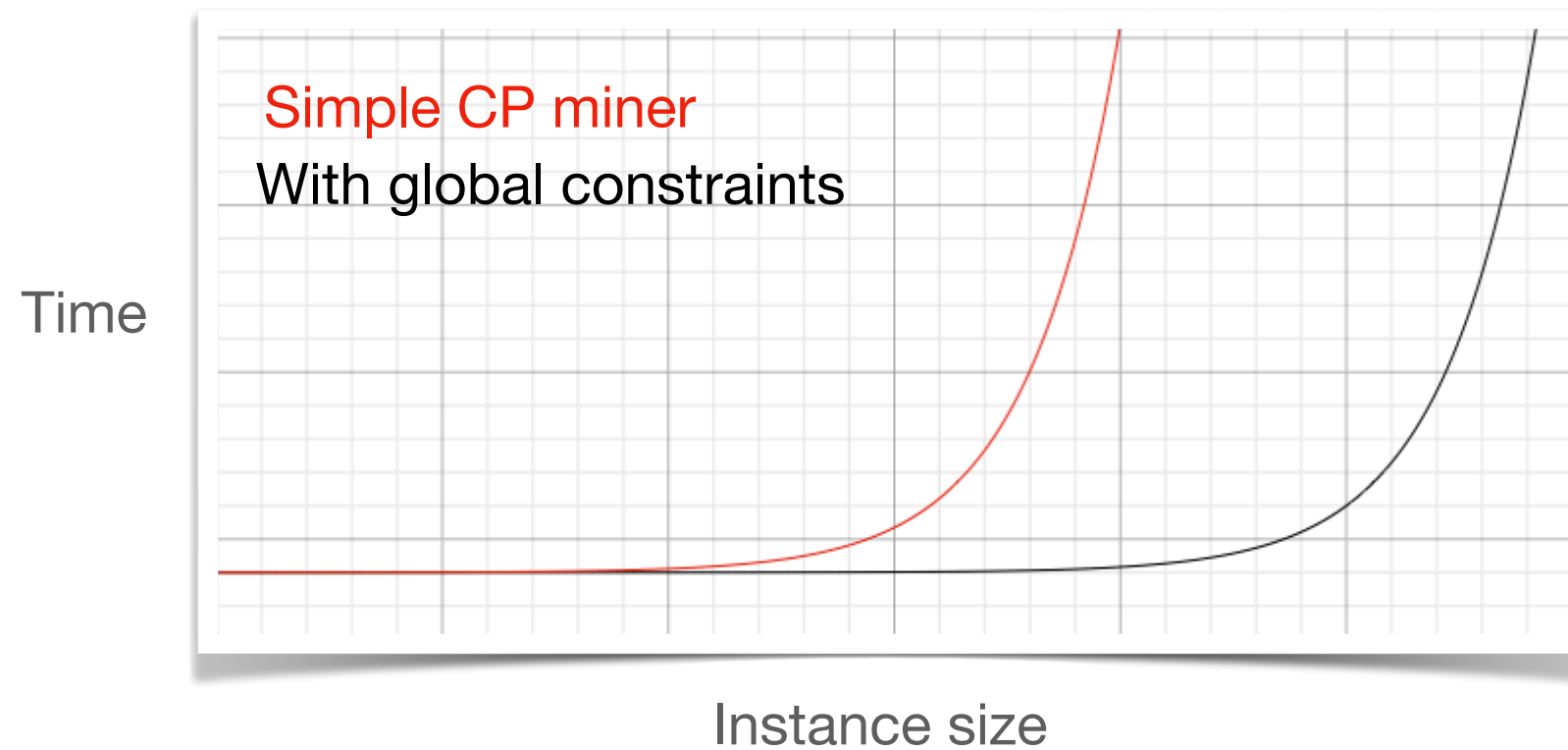
- $u(x^{-1}(1) \cup \{i\}) \leq u(x^{-1}(1)) \quad \forall i \in I \setminus (x^{-1}(1) \cup x^{-1}(0))$

A GENERAL CP MODEL FOR HUI AND UPI



A GENERAL CP MODEL FOR HUI AND UPI

.....



PEAKUTILITY CONSTRAINT

► Definition:

Let X be a vector of boolean variables, D be a dataset, θ a minimum utility and a given HUI/UPI mode. Given a complete and P the itemset induced by X , $PeakUtility_{D,\theta,mode}(X)$ holds iff:

1. $u(P) \geq \theta$ (High-Utility condition)

2. If mode= UPI:

$$\forall i \in P, \quad u(P \setminus \{i\}) \leq u(P), \quad \forall i \in I \setminus P, \quad u(P \cup \{i\}) \leq u(P)$$

(Peak condition)

- Mining all UPIs with $u(P) \geq \theta$ is NP-hard =>
- enforcing full domain consistency on PeakUtility is intractable unless $P = NP$ => Approximate propagation rules

PEAKUTILITY PROPAGATOR

- Given a partial assignment $(\sigma^+, \sigma^-, \sigma^*)$, we have 8 filtering rules:

PEAKUTILITY PROPAGATOR

➤ Given a partial assignment $(\sigma^+, \sigma^-, \sigma^*)$, we have 8 filtering rules:

➤ **Rule 1:** $\sigma^+ \neq \emptyset \wedge \text{cover}(\sigma^+) = \emptyset \Rightarrow \text{failure}$

➤ **Rule 2:** $\forall i \in \sigma^*, \text{cover}(\sigma^+ \cup \{i\}) = \emptyset \Rightarrow \text{remove } 1 \text{ from } \text{dom}(x_i)$

➤ **Rule 3:** $TWU(\sigma^+) < \theta \Rightarrow \text{failure}$

➤ **Rule 4:** $u(\sigma^+) + ru(\sigma^+, \sigma^*) < \theta \Rightarrow \text{failure}$

HUI

PEAKUTILITY PROPAGATOR

➤ Given a partial assignment $(\sigma^+, \sigma^-, \sigma^*)$, we have 8 filtering rules:

➤ **Rule 1:** $\sigma^+ \neq \emptyset \wedge \text{cover}(\sigma^+) = \emptyset \Rightarrow \text{failure}$

➤ **Rule 2:** $\forall i \in \sigma^*, \text{cover}(\sigma^+ \cup \{i\}) = \emptyset \Rightarrow \text{remove } 1 \text{ from } \text{dom}(x_i)$

➤ **Rule 3:** $TWU(\sigma^+) < \theta \Rightarrow \text{failure}$

➤ **Rule 4:** $u(\sigma^+) + ru(\sigma^+, \sigma^*) < \theta \Rightarrow \text{failure}$

➤ **Rule 5:** $u(\sigma^+) + ru(\sigma^+ \cup \{i\}, \sigma^* \setminus \{i\}) < \theta \Rightarrow \text{remove } 1 \text{ from } \text{dom}(x_i)$

➤ **Rule 6:** $\sigma^* = \emptyset \wedge \exists j \in \sigma^+, u(\sigma^+ \setminus \{j\}) \geq u(\sigma^+) \Rightarrow \text{failure}$

➤ **Rule 7:** $\sigma^* = \emptyset \wedge \exists i \in \sigma^-, u(\sigma^+ \cup \{i\}) \geq u(\sigma^+) \Rightarrow \text{failure}$

➤ **Rule 8:** $|\sigma^*| = k \wedge (\forall S \subseteq \sigma^*, S \neq \emptyset, u(\sigma^+ \cup S) < \theta) \Rightarrow \forall i \in \sigma^*, 1 \notin \text{dom}(x_i)$

HUI

UPI

PEAKUTILITY CONSTRAINT (SOME RESULTS)

- **PeakUtility** and **CP models** implemented in Choco-Solver 4.10 (gitlab.lisn.upsaclay.fr/dpm/peakutility)
- Intel Xeon E5-2680 v4 (2.4 GHz, 64 GB RAM) under Ubuntu 20.04
- 900s timeout and 256GB memory limit
- Bitset encoding for transactions
- Comparison with EFIM (Specialized depth-first algorithm for HUI mining) [Zida et al., 2015]

PEAKUTILITY CONSTRAINT (SOME RESULTS)

.....

CP4HUIM			PeakUtility	
Dataset	#HUI	nodes	#HUI	nodes
BMS	0	134K	3M	6M
Chess	120	237K	28M	58M
Connect	0	9K	319K	4M
Foodmart	0	1M	230K	463K
Mushroom	2	487K	10M	21M
Retail	0	33K	17M	869K

TO= 900s, 256 GB memory limit

PEAKUTILITY CONSTRAINT (SOME RESULTS)

EFIM-PP				PeakUtility		
Dataset	#HUI	#UPI	T(s)	#HUI	#UPI	T(s)
Sparse						
S01	3K	462	(0 , 0)	3K	462	(1, 1)
S08	10K	2K	(0 , 1)	10K	2K	(0 , 0)
S14	40K	7K	(4 , 6)	40K	7K	(5 , 13)
R01_foodmart	54K	1K	(1 , 13)	54K	1K	(15 , 26)
R02_retail	>10 ⁹	—	OOM	17K	1K	TO
Dense						
S23	6K	85	(0 , 0)	6K	85	(0 , 1)
S27	38K	470	(1 , 1)	38K	470	(2 , 3)
S39	6K	237	(0 , 0)	6K	237	(2 , 1)
R03_connect	>10 ⁹	—	OOM	319K	0	TO
R04_mushroom	>10 ¹⁰	—	OOM	10M	0	TO
R06_chess	>10 ¹⁰	—	OOM	28M	0	TO

TO= 900s, 256 GB memory limit

CONCLUSION

- New **utility-driven** condensed representation: **Utility-Peak Itemsets (UPIs)** — itemsets that are locally maximal w.r.t. utility (not size, not inclusion)
- Complexity study: deciding whether a θ -UPI exists is **NP-complete**
- Global constraint **PeakUtility**: unifies HUI and UPI enumeration through 8 propagation rules with utility upper-bound pruning
- Experiments on 52 datasets: competitive with EFIM on tractable instances, **robust where EFIM runs out of memory**; direct UPI mining avoids exhaustive HUI enumeration