# Supplementary Material

This paper serves as a companion document to our main research paper, providing additional in-depth content. It is structured as follows: Section 1 presents detailed proofs of the propositions introduced in the first submission. In Section 2, we introduce a baseline CP model denoted as Baseline-CP-Jaccard, which encodes the maximum Jaccard constraint as a linear sum. Further experiments, designed to explore specific research queries related to the characteristics of the mined patterns, are provided in Sections 3.1 and 3.2. The evaluation of the proposed heuristic search strategies and their impact on diversity is presented in Sections 3.3. Complementary results are reported in the Appendix at Sections A, B, C and D.

## 1 Proofs of Propositions

This section presents detailed proofs of the propositions introduced in the main body of our research paper.

### Proof of Lemma 1 (Residual cover)

*Proof* As $Q \subseteq P$, we get $\mathcal{V}(P) \subseteq \mathcal{V}(Q)$. We have $\mathcal{V}(P) = \mathcal{V}_H^{pr}(P) \cup \{\mathcal{V}(P) \cap \mathcal{V}(H)\}$, with $\mathcal{V}_H^{pr}(P) \cap \{\mathcal{V}(P) \cap \mathcal{V}(H)\} = \emptyset$. Hence, $\mathcal{V}_H^{pr}(P) = \mathcal{V}(P) \setminus \{\mathcal{V}(P) \cap \mathcal{V}(H)\}$.

Now, let $t \in \mathcal{V}_H^{pr}(P)$. Thus, $t \in \mathcal{V}(P) \wedge t \notin \{\mathcal{V}(P) \cap \mathcal{V}(H)\} \cdots (1)$.
As $\mathcal{V}(P) \subseteq \mathcal{V}(Q)$, from (1), we get $t \in \mathcal{V}(P)$ and $t \in \mathcal{V}(Q) \cdots (2)$.
As $t \in \mathcal{V}(P) \wedge t \notin \{\mathcal{V}(P) \cap \mathcal{V}(H)\}$, we have $t \notin \mathcal{V}(H)$.
Since $t \notin \mathcal{V}(H) \Rightarrow t \notin \{\mathcal{V}(Q) \cap \mathcal{V}(H)\} \cdots (3)$
From (2) and (3) respectively, we get $t \in \mathcal{V}(Q)$ and $t \notin \{\mathcal{V}(Q) \cap \mathcal{V}(H)\} \cdots (4)$.

We have $\mathcal{V}(Q) = \mathcal{V}_H^{pr}(Q) \cup \{\mathcal{V}(Q) \cap \mathcal{V}(H)\}$. From (4) we conclude that $t \in \mathcal{V}_H^{pr}(Q)$.
Hence, $\forall t \in \mathcal{V}_H^{pr}(P)$, we have $t \in \mathcal{V}_H^{pr}(Q)$ and thus, $\mathcal{V}_H^{pr}(P) \subseteq \mathcal{V}_H^{pr}(Q)$. □

### Proof of Proposition 2 (New lower bound)

*Proof* $\forall H \in \mathcal{H}$ we have :

$$\begin{aligned} |\mathcal{V}(P)| \geq \theta &\Leftrightarrow |\mathcal{V}(H) \cap \mathcal{V}(P)| + |\mathcal{V}_H^{pr}(P)| \geq \theta \\ &\Leftrightarrow |\mathcal{V}(H) \cap \mathcal{V}(P)| \geq \theta - |\mathcal{V}_H^{pr}(P)| \end{aligned}$$

Since $|\mathcal{V}(H) \cup \mathcal{V}(P)| = |\mathcal{V}(H)| + |\mathcal{V}_H^{pr}(P)|$, we get

$$Jac(H,P) = \frac{|\mathcal{V}(H) \cap \mathcal{V}(P)|}{|\mathcal{V}(H) \cup \mathcal{V}(P)|} = \frac{|\mathcal{V}(H) \cap \mathcal{V}(P)|}{|\mathcal{V}(H)| + |\mathcal{V}_H^{pr}(P)|} \geq \frac{\theta - |\mathcal{V}_H^{pr}(P)|}{|\mathcal{V}(H)| + |\mathcal{V}_H^{pr}(P)|}$$

As $Jac(H,P) \geq 0$, if $|\mathcal{V}_H^{pr}(P)| \geq \theta$, the numerator becomes 0, and $\frac{0}{|\mathcal{V}(H)|+|\mathcal{V}_H^{pr}P|} = \frac{0}{\mathcal{V}(H)} = 0$. □

## Proof of Proposition 3 (Tightness of $LB_J(P,H)$ compared to $LB_J^{old}(P,H)$ introduced by Hien et al. (2020))

*Proof* The proof is straightforward

$$|\mathcal{V}(P)| \geq \theta \Leftrightarrow |\mathcal{V}(P)| + |\mathcal{V}(H)| + |\mathcal{V}_H^{pr}(P)| - \theta \geq |\mathcal{V}(H)| + |\mathcal{V}_H^{pr}(P)|$$
$$\Leftrightarrow \frac{1}{|\mathcal{V}(P)| + |\mathcal{V}(H)| + |\mathcal{V}_H^{pr}(P)| - \theta} \leq \frac{1}{|\mathcal{V}(H)| + |\mathcal{V}_H^{pr}(P)|}$$

we get $LB_J^{old}(P,H) \leq LB_J(P,H)$. □

## Proof of Proposition 4 (Monotonicity of $LB_J$)

*Proof* Since $|\mathcal{V}_H^{pr}(P)| \leq |\mathcal{V}_H^{pr}(Q)|$ (see lemma 1), we get

$$LB_J(H,P) = \frac{\theta - |\mathcal{V}_H^{pr}(P)|}{|\mathcal{V}(H)| + |\mathcal{V}_H^{pr}(P)|} \geq \frac{\theta - |\mathcal{V}_H^{pr}(Q)|}{|\mathcal{V}(H)| + |\mathcal{V}_H^{pr}(P)|}$$
$$\geq \frac{\theta - |\mathcal{V}_H^{pr}(Q)|}{|\mathcal{V}(H)| + |\mathcal{V}_H^{pr}(Q)|} = LB_J(H,Q)$$

□

## Proof of Proposition 6 (Upper bound)

*Proof* $\forall H \in \mathcal{H}$ we have:

$$\Rightarrow Jac(H,P) = \frac{|\mathcal{V}(H) \cap \mathcal{V}(P)|}{|\mathcal{V}(H) \cap \mathcal{V}(P)| + |\mathcal{V}_H^{pr}(P)| + |\mathcal{V}_P^{pr}(H)|}$$
$$\Rightarrow |\mathcal{V}_H^{pr}(P)| + |\mathcal{V}(H) \cap \mathcal{V}(P)| \geq \theta |\mathcal{V}_H^{pr}(P)| \geq \theta - |\mathcal{V}(H) \cap \mathcal{V}(P)|$$
$$\Rightarrow Jac(H,P) \leq \frac{|\mathcal{V}(H) \cap \mathcal{V}(P)|}{|\mathcal{V}(H) \cap \mathcal{V}(P)| + |\mathcal{V}_P^{pr}(H)| + \max\{0, \theta - |\mathcal{V}(H) \cap \mathcal{V}(P)|\}}$$
$$\Rightarrow Jac(H,P) \leq \frac{|\mathcal{V}(H) \cap \mathcal{V}(P)|}{|\mathcal{V}_P^{pr}(H)| + \max\{\theta, |\mathcal{V}(H) \cap \mathcal{V}(P)|\}}$$

□

## Proof of Proposition 7 (Anti-monotonicity of $UB_J$)

*Proof* $\forall Q \supset P, \mathcal{V}(Q) \subseteq \mathcal{V}(P)$, and history pattern $H \in \mathcal{H}$:

$$UB_J(H,P) = \frac{|\mathcal{V}(H) \cap \mathcal{V}(P)|}{|\mathcal{V}_P^{pr}(H)| + \max\{|\mathcal{V}(H) \cap \mathcal{V}(P)|, \theta\}}$$

We have to consider three cases:

1. $|\mathcal{V}(H) \cap \mathcal{V}(P)| > \theta, |\mathcal{V}(H) \cap \mathcal{V}(Q)| > \theta$:

$$UB_J(H,P) = \frac{|\mathcal{V}(H) \cap \mathcal{V}(P)|}{|\mathcal{V}_P^{pr}(H)| + |\mathcal{V}(H) \cap \mathcal{V}(P)|} = \frac{|\mathcal{V}(H) \cap \mathcal{V}(P)|}{|\mathcal{V}(H)|}$$

$$\geq \frac{|\mathcal{V}(H) \cap \mathcal{V}(Q)|}{|\mathcal{V}(H)|} = UB_J(H, Q)$$

2. $|\mathcal{V}(H) \cap \mathcal{V}(P)| > \theta, |\mathcal{V}(H) \cap \mathcal{V}(Q)| \leq \theta \Rightarrow \mathcal{V}_P^{pr}(H) + \theta \geq \mathcal{V}(H)$:

$$UB_J(H, P) = \frac{|\mathcal{V}(H) \cap \mathcal{V}(P)|}{|\mathcal{V}_P^{pr}(H)| + |\mathcal{V}(H) \cap \mathcal{V}(P)|} = \frac{|\mathcal{V}(H) \cap \mathcal{V}(P)|}{|\mathcal{V}(H)|}$$

$$\geq \frac{|\mathcal{V}(H) \cap \mathcal{V}(Q)|}{|\mathcal{V}_Q^{pr}(H)| + \theta} = UB_J(H, Q)$$

3. $|\mathcal{V}(H) \cap \mathcal{V}(P)| \leq \theta, |\mathcal{V}(H) \cap \mathcal{V}(Q)| \leq \theta \Rightarrow |\mathcal{V}(H) \cap \mathcal{V}(P)| \geq |\mathcal{V}(H) \cap \mathcal{V}(Q)| \mathcal{V}_P^{pr}(H) \leq \mathcal{V}_Q^{pr}(H)$:

$$UB_J(H, P) = \frac{|\mathcal{V}(H) \cap \mathcal{V}(P)|}{|\mathcal{V}_P^{pr}(H)| + \theta} \geq \frac{|\mathcal{V}(H) \cap \mathcal{V}(Q)|}{|\mathcal{V}_P^{pr}(H)| + \theta}$$
$$\geq \frac{|\mathcal{V}(H) \cap \mathcal{V}(Q)|}{|\mathcal{V}_Q^{pr}(H)| + \theta} = UB_J(H, Q)$$

As the anti-monotonicity holds for all three cases, it holds for the upper bound overall.

□

## Proof of Proposition 8 (ClosedDiversity Filtering rules)

*Proof* The proof of filtering rule (1) follows from proposition 5. We give the proof of rule (2). Let $P = x^+ \cup \{i\}$, $Q = x^+ \cup \{k\}$ s.t. $i \in x^*$ and $k \in x_{Div}^-$ and $H \in \mathcal{H}$.

$$LB_J(H, P) = \frac{\theta - |\mathcal{V}_H^{pr}(P)|}{|\mathcal{V}(H)| + |\mathcal{V}_H^{pr}(P)|}$$

$$|\mathcal{V}(P)| \leq |\mathcal{V}(Q)| \Rightarrow |\mathcal{V}_H^{pr}(P)| \leq |\mathcal{V}_H^{pr}(Q)|$$

$$\Rightarrow LB_J(H, P) \geq \frac{\theta - |\mathcal{V}_H^{pr}(Q)|}{|\mathcal{V}(H)| + |\mathcal{V}_H^{pr}(P)|} \geq \frac{\theta - |\mathcal{V}_H^{pr}(Q)|}{|\mathcal{V}(H)| + |\mathcal{V}_H^{pr}(Q)|} = LB_J(H, Q)$$

□

## Proof of Proposition 9 (Consistency and time complexity)

*Proof* **(i) GAC consistency.** Algorithm 1 embodies two kinds of filtering rules, especially (i) the closedness filtering rules that are already proved GAC (see (Lazaar et al., 2016)), and (ii) the rule of the $LB$ relaxation, for which any free item $i \in x^*$ that cannot be an extension of the current partial itemset $x^+$ to a diverse FCI is necessarily pruned (see Proposition 8). By construction, any values of $x_i \in x$, which do not belong to any solution of ClosedDiversity are deleted from $dom(x_i)$. Hence, the ClosedDiversity constraint enforces the Generalized Arc Consistency of $x$.
**(ii) Time complexity.** Line (4) runs in $\mathcal{O}(n \times m)$. Function $\mathcal{P}Growth_{LB}$ runs in $\mathcal{O}(m \times |\mathcal{H}|)$, since each $LB_J(H, x)$ call runs in $\mathcal{O}(m)$, thus filtering domains is achieved in $\mathcal{O}(n \times m \times |\mathcal{H}|)$. Now, if we suppose that $|\mathcal{H}| \leq n$, then we get $\mathcal{O}(n^2 \times m)$. Besides, handling the closedness filtering is $\mathcal{O}(\frac{n^2}{4} \times m)$, since (i) the pruning of infrequent

itemsets is $\mathcal{O}(n \times m)$ because in that case, handling the test in line 6 is $\mathcal{O}(m)$; (ii) the closure extension pruning is also $\mathcal{O}(n \times m)$ due to the test of line 8 that is achieved in $\mathcal{O}(m)$; (iii) the last closure filtering rule in lines 12−17 is $\mathcal{O}(\frac{n^2}{4} \times m)$ because the inclusion test in line 13 is $\mathcal{O}(m)$ and $|x^*| + |x^-|$ is at most $n$, thus this pruning rule is checked with at most $\frac{n}{2} \times \frac{n}{2}$ operations (i.e. $\mathcal{O}(\frac{n^2}{4})$). Since this last filtering rule is disjoint from the diversity filtering rule (this is why we "continue" at line 11), the overall time complexity is then $\mathcal{O}(n^2 \times m)$. In a similar way, we can easily find that the worst-case time complexity of Algorithm 3 is $\mathcal{O}(n \times m \times |\mathcal{H}|)$. □

# 2 Baseline Encoding for the Pairwise Jaccard Constraint

One of the key advantages of constraint programming solvers is their ability to go beyond the classical pattern mining monotonic and anti-monotonic principles, which are based on the traditional general-to-specific enumeration of 'true' items. Instead, CP requires propagators to be monotonic with respect to the 'domain' of the variables, encompassing both true, false, and free items. This is why constraint programming for itemset mining methods allow to combine constraints that are, in classical pattern mining, anti-monotone, monotone, convertible monotone and more.

To assess the interest of our relaxed version (i.e., lower bound) of the maximum pairwise Jaccard constraint in terms of performance, we propose a baseline encoding for this constraint as a linear sum. The main idea is to decompose the Jaccard index into simple operations on sets using intermediate Boolean variables representing the covers of the itemsets.

## 2.1 Encoding the maximum pairwise Jaccard constraint as a linear sum

In CP the cardinality of a set can be calculated by summing 0/1 variables representing the elements in the set. However, to deal with Jaccard similarity, we need to calculate the cardinality of a set which is the result of comparing two other sets.

Let $H$ be a member pattern of the history $\mathcal{H}$ and $P$ a given itemset. To evaluate the expression $|\mathcal{V}(H) \cap \mathcal{V}(P)|$, we would need to calculate the set $\mathcal{V}(H) \cap \mathcal{V}(P)$ and then sum the variables representing this set, while the expression $|\mathcal{V}(H) \cup \mathcal{V}(P)|$ can be rewritten as $|\mathcal{V}(H)| + |\mathcal{V}(P)| - |\mathcal{V}(H) \cap \mathcal{V}(P)|$.

In our CP formulation we have Boolean transaction variables $T_j$ that represent whether the transaction with id $j$ covers the itemset or not. We model the set $\mathcal{V}(P)$ by a vector $T^P$ of $m$ Boolean transaction variables, such that $(T^P_j = 1)$ iff itemset $P \subseteq t_j$. Now the expression $|\mathcal{V}(H) \cap \mathcal{V}(P)|$ can be rewritten as follows:

$$|\mathcal{V}(H) \cap \mathcal{V}(P)| = \sum_{j \in \mathcal{T}} (T^P_j T^H_j)$$

Here we count the number of transactions for which both $T_j^P$ and $T_j^H$ are 1. Since $\mathcal{V}(H)$ is a constant set, $T_j^H$ are also constant, leading to a linear sum. Proposition 1 gives our baseline CP encoding for the maximum Jaccard constraint.

**Proposition 1** (**Maximum Similarity constraint baseline encoding**) *Given a member pattern $H$ of the history $\mathcal{H}$, and an itemset $P$. Let $T^P$ be a vector of Boolean transaction variables associated with the set $\mathcal{V}(P)$ and $T^H$ be a binary constant vector representing the cover $\mathcal{V}(H)$, then*

$$Jac(H,P) \leq J_{max} \Leftrightarrow \sum_{j \in \mathcal{T}} \alpha_j T_j^P \leq \beta \tag{1}$$

*where,*

$$\alpha_j = T_j^H - J_{max}(1 - T_j^H)$$
$$\beta = J_{max} \sum_{j \in \mathcal{T}} T_j^H$$

*Proof* $\forall H \in \mathcal{H}$ we have :

$$\begin{aligned} Jac(H,P) \leq J_{max} &\Leftrightarrow |\mathcal{V}(H) \cap \mathcal{V}(P)| \leq J_{max} |\mathcal{V}(H) \cup \mathcal{V}(P)| \\ &\Leftrightarrow |\mathcal{V}(H) \cap \mathcal{V}(P)| - J_{max} |\mathcal{V}(H) \cup \mathcal{V}(P)| \leq 0 \end{aligned}$$

Since

$$|\mathcal{V}(H) \cap \mathcal{V}(P)| = \sum_{j \in \mathcal{T}} (T_j^P T_j^H)$$
$$|\mathcal{V}(P)| = \sum_{j \in \mathcal{T}} T_j^P$$
$$|\mathcal{V}(H)| = \sum_{j \in \mathcal{T}} T_j^H$$

we get,

$$\begin{aligned} Jac(H,P) \leq J_{max} &\Leftrightarrow \sum_{j \in \mathcal{T}} (T_j^P T_j^H) - J_{max} \left( \sum_{j \in \mathcal{T}} T_j^P + \sum_{j \in \mathcal{T}} T_j^H - \sum_{j \in \mathcal{T}} T_j^P y_j^H \right) \leq 0 \\ &\Leftrightarrow \sum_{j \in \mathcal{T}} (T_j^P T_j^H) - J_{max} \left( \sum_{j \in \mathcal{T}} T_j^P (1 - T_j^H) + \sum_{j \in \mathcal{T}} T_j^H \right) \leq 0 \\ &\Leftrightarrow \sum_{j \in \mathcal{T}} T_j^P \left( T_j^H - J_{max} \left( 1 - T_j^H \right) \right) - J_{max} \times \sum_{j \in \mathcal{T}} T_j^H \leq 0 \\ &\Leftrightarrow \sum_{j \in \mathcal{T}} \alpha_j T_j^P \leq \beta \end{aligned}$$

where,

$$\alpha_j = T_j^H - J_{max}(1 - T_j^H)$$
$$\beta = J_{max} \sum_{j \in \mathcal{T}} T_j^H$$

□

## 2.2 The ClosedCover global constraint

To exploit our baseline encoding of the maximum Jaccard constraint presented before, we need to represent the covers of itemsets. We propose the $\textsc{ClosedCover}_{\mathcal{D},\theta}(x,T)$ global constraint which extends ClosedPattern (Lazaar et al., 2016) by introducing a new vector $T$ of Boolean variables $(T_1,\ldots,T_{|\mathcal{T}|})$ to represent the covers of itemsets, where $T_j$ represents the presence/absence of the returned pattern in the transaction id $j$.

**Definition 1** (ClosedCover) Let $x$, $T$ be two vectors of Boolean variables, $\theta$ a support threshold and $\mathcal{D}$ a dataset. Given a complete assignment on variables $x$ and $T$, with $x^+ = \{i \in \mathcal{I} \mid x_i = 1\}$. $\textsc{ClosedCover}_{\mathcal{D},\theta}(x,T)$ holds iff $sup_{\mathcal{D}}(x^+) \geq \theta$, $x^+$ is closed, and $\mathcal{V}(x^+) = T^+$, with $T^+ = \{j \in \mathcal{T} \mid T_j = 1\}$.

The propagator we propose for ClosedCover exploits the filtering rules of ClosedPattern, and extends it with two additional rules for the Boolean variables of vector $T$. Given a partial assignment on $x$, for any $j \in T^*$, the following rules prune inconsistent values from $dom(T_j)$:

- $1 \notin dom(T_j)$ iff: $j \notin \mathcal{V}(x^+)$;
- $0 \notin dom(T_j)$ iff: $j \in (\mathcal{V}(x^+) \cap \mathcal{V}(x^*))$

The first rule filters transactions not covered by the present items $x^+$, while the second rule preserves transactions covered by both current items $x^+$ and free items $x^*$. As the diversity constraint can modify the domains of variables $T$, leading to inconsistencies with the covers of the current itemset computed in ClosedCover, we perform a consistency check before propagation and we return fail if the following condition holds: $\exists\, j \in \mathcal{T} : dom(T_j) = 0 \wedge j \in \mathcal{V}(x^+ \cup x^*)$.

Finally, the complete CP model connecting both the ClosedCover constraint with the linear sum encoding of the maximum pairwise Jaccard constraint is the following:

$$\textsc{Baseline-CP-Jaccard}_{\mathcal{D},\theta,J_{max}}(x,T^P,\mathcal{H}) \equiv \begin{cases} \textsc{ClosedCover}_{\mathcal{D},\theta}(x,T^P) \\ \forall H \in \mathcal{H} : \sum\limits_{j\in\mathcal{T}} \alpha_j T^P_j \leq \beta \end{cases}$$

## 2.3 Exploiting the baseline CP model

Initially, the history $\mathcal{H}$ is empty. It is incrementally updated with diverse frequent closed itemsets encountered during search. First, the constraint store is initialized by ClosedCover global constraint. Then, each time a new solution $H$ is found, it is added to the history $\mathcal{H}$ and we dynamically post a new pairwise Jaccard constraint between $H$ and the next mined itemset $x^+$. Let $H_i$ be the current solution found, i.e. $\mathcal{H} = \{H_1,\ldots,H_i\}$, we have that

$Jac(H_j, x^+) \leq J_{max}, \forall H_j \in \mathcal{H}$ s.t. $j \leq i$. This process stops when no solution exists satisfying the posted constraint model.

The Baseline-CP-Jaccard model requires $|\mathcal{H}|$ dynamic linear constraints to be posted during the search, in addition to the ClosedCover global constraint. As it is shown in our experiments, this model is very costly for CP solvers due to the huge number of linear sum constraints to be managed. Additionally, these constraints, which are typically large (contain $|\mathcal{T}|$ variables), have particularly weak propagation.

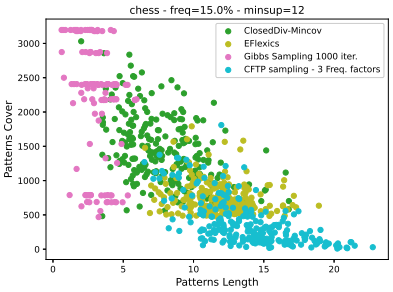
# 3 Additional Experimental Results

In this section, we present and discuss the results of supplementary experiments conducted to further validate and enhance the findings provided in the main body of our research article. The additional experiments were designed to explore specific research queries summarized as follows:

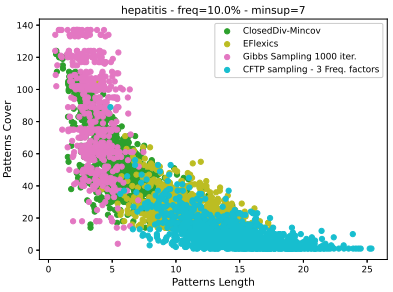
- **Q**$_1$ How (**in terms of length and cover size**) do patterns discovered by ClosedDiv compare to those mined by specialized approaches? (see Section 3.1)
- **Q**$_2$ How (**in terms of # of patterns**) does ClosedDiv compare to ClosedP, to exact Jaccard CP-based approaches and to specialized alternatives (i.e. Picker and PatternsTeam)? (see Section 3.2)
- **Q**$_3$ How (**in terms of running time and diversity of the pattern set**) does the branching heuristics impact the performance of the mining process and the quality of the resulting pattern set? (see Section 3.3)

## 3.1 Characteristics of the mined patterns

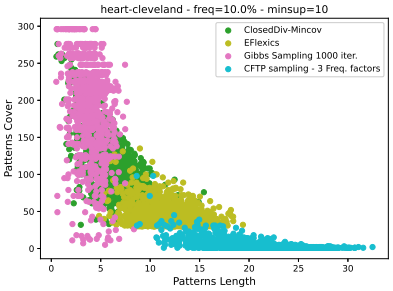
We investigate which kind of patterns are mined by each approach. We show in Figure 1 the scatter plots of the points obtained from two pattern descriptions, i.e., length and cover size. This plot tells us about the shape of patterns sampled by each approach. Gibbs favors sampling of short patterns widely distributed in terms of cover size, while cftp favors sampling of long patterns covering only a few transactions. This is the reason why the Jaccard indices between all the solutions returned by cftp are very small. With small covers (less than 10 transactions), likely, two covers will not intersect, hence leading to a Jaccard index of 0. ClosedDiv-Mincov is somewhere in the middle, which returns well-distributed patterns in terms of length and cover size. Flexics is also able to return patterns with diverse lengths but less distributed in terms of cover size. On very sparse datasets, although the length of sampled patterns by cftp and Gibbs varies, they, unfortunately, cover only one or two transactions.



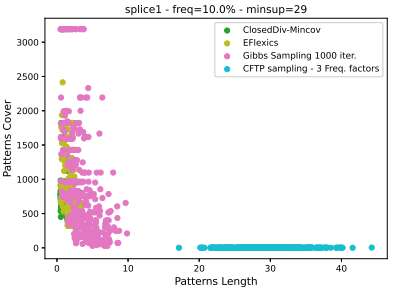

(a) Chess ($\theta = 15\%$)



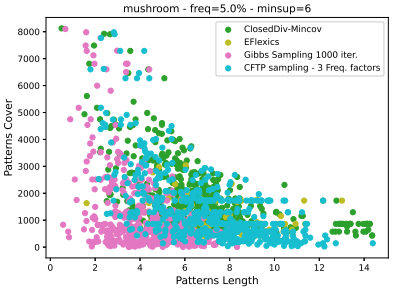

(b) Hepatitis ($\theta = 10\%$)



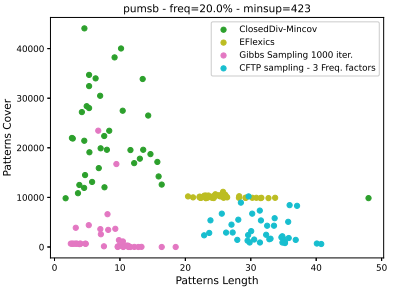

(c) Heart-cleveland ($\theta = 10\%$)



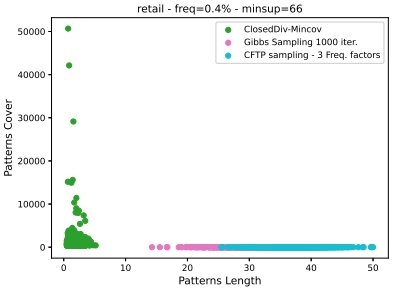

(d) Splice1 ($\theta = 10\%$)



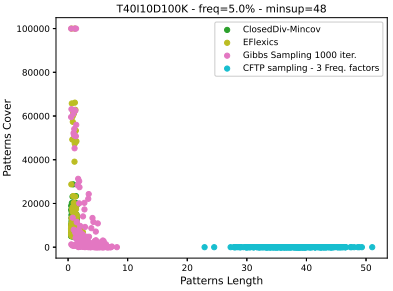

(e) Mushroom ($\theta = 5\%$)




(f) Pumsb ($\theta = 20\%$)




(g) Retail ($\theta = 0.4\%$)




(h) T40I10D100K ($\theta = 5\%$)

**Fig. 1**: Scatter plots of the sizes of the patterns and their covers across various datasets.

| Dataset<br>$\|\mathcal{I}\| \times \|\mathcal{T}\|$<br>$\rho$(%) | $\theta$(%) | #Patterns | | | | | | | #Nodes | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (1) | (2) | (3) | (4) | (5) |
| Chess | 30 | 5,316,468 | **2** | 14 | **2** (433,165) | **2** | OOM | OOM | 10,632,935 | 10,632,935 | **57** | 866,335 | 5,316,469 |
| 75 × 3,196 | 20 | 22,808,625 | **3** | 65 | **4** (194,270) | - | OOM | OOM | 45,617,249 | 45,617,249 | **318** | 388,545 | - |
| 49.33% | 15 | 50,723,131 | **4** | 238 | **5** (73,152) | - | OOM | OOM | 101,446,261 | 101,446,261 | **1,154** | 146,313 | - |
| | 10 | OOM | **8** | 1,622 | **8** (36,732) | - | OOM | OOM | 246,486,145 | **7,774** | 73,483 | - | |
| Hepatitis | 30 | 83,048 | **2** | 11 | **3** (3,485) | **2** | 10 | 11 | 166,095 | 166,095 | **28** | 6,972 | 83,049 |
| 68 × 137 | 20 | 410,318 | **4** | 45 | **4** (3,913) | **4** | 10 | 11 | 820,635 | 820,635 | **129** | 7,829 | 41,0321 |
| 50.00% | 10 | 1,827,264 | **9** | 1,018 | **9** (12,408) | **9** | 10 | 11 | 3,654,527 | 3,654,527 | **2,545** | 24,829 | 1,826,753 |
| Kr-vs-kp | 30 | 5,219,727 | **2** | 14 | **2** (432,426) | **2** | OOM | OOM | 10,439,453 | 10,439,453 | **57** | 864,857 | 5,219,728 |
| 73 × 3,196 | 20 | 21,676,719 | **3** | 64 | **4** (138,029) | **3** | OOM | OOM | 43,353,437 | 43,353,437 | **307** | 276,063 | 21,676,721 |
| 49.32% | 10 | OOM | **7** | 1,609 | **7** (27,601) | - | OOM | OOM | OOM | 218,876,381 | **7,703** | 55,221 | - |
| Connect | 30 | 460,357 | **1** | 19 | 3 (106,705) | - | OOM | OOM | 920,713 | 920,713 | **89** | 213,415 | - |
| 129 × 67,557 | 18 | 2,005,476 | **3** | 141 | 4 (542,858) | - | OOM | OOM | 4,010,951 | 4,010,951 | **699** | 1,085,725 | - |
| 33.33% | 15 | 3,254,780 | **4** | 297 | 5 (519,639) | - | OOM | OOM | 6,509,559 | 6,509,559 | **1,389** | 1,039,290 | - |
| | 10 | 8,035,412 | **7** | 1,907 | 9 (73,634) | - | OOM | OOM | 16,070,823 | 16,070,823 | **7,815** | 147,302 | - |
| Heart-cleveland | 10 | 12,774,456 | 9 | 1,470 | **8** (37,115) | 9 | 13 | OOM | 25,548,911 | 25,548,911 | **3,735** | 74,233 | 12,774,464 |
| 95 × 296 | 8 | 23,278,687 | 13 | 4,761 | **10** (4,817) | 13 | 14 | OOM | 46,557,373 | 46,557,373 | 11,441 | **9,648** | 23,278,699 |
| 47.37% | 6 | 43,588,346 | 19 | 20,490 | **12** (10,718) | 19 | 13 | OOM | 87,176,691 | 87,176,691 | 46,506 | **21,456** | 43,588,355 |
| Splice1 | 10 | 1,606 | **5** | 413 | 6 (1,042) | 5 | 50 | OOM | 3,211 | 3,211 | **825** | 2,083 | 1,610 |
| 287 × 3,190 | 5 | 31,441 | 99 | 7,920 | **89** (19,449) | 99 | 43 | OOM | 62,881 | 62,881 | **15,886** | 38,898 | 31,539 |
| 20.91% | 2 | 589,588 | 448 | - | **416** (355,612) | 448 | ?? | OOM | 1,179,175 | 1,179,175 | - | **711,229** | 590,035 |
| Mushroom | 5 | 8,977 | 13 | 548 | **11** (881) | 13 | 37 | 25 | 17,953 | 17,953 | **1,357** | 1,768 | 8,989 |
| 112 × 8,124 | 1 | 40,368 | 110 | 9,935 | **72** (2,119) | 110 | 41 | 25 | 80,735 | 80,735 | 20,924 | **4,249** | 40,477 |
| 18.75% | 0.8 | 47,765 | 199 | 12,743 | **90** (2,554) | 199 | 41 | 25 | 95,529 | 95,529 | 26,660 | **5,128** | 47,963 |
| | 0.5 | 62,334 | 278 | 23,931 | **121** (3,056) | 278 | 42 | 25 | 124,667 | 12,4667 | 49,406 | **6,120** | 62,611 |
| T40I10D100K | 8 | 138 | **7** | 125 | 10 (129) | **7** | 121 | 30 | 275 | 275 | 249 | 257 | **144** |
| 942 × 100,000 | 5 | 317 | **22** | 284 | 33 (289) | **22** | 123 | 30 | 633 | 633 | 567 | 577 | **338** |
| 4.20% | 1 | 65,237 | **373** | 7,217 | 395 (9,774) | - | 135 | OOM | 130,473 | 130,473 | **14,517** | 19,631 | - |
| Pumsb | 40 | - | - | **4** | - | - | OOM | OOM | - | - | **15** | - | - |
| 2,113 × 49,046 | 30 | - | - | **14** | - | - | OOM | OOM | - | - | **59** | - | - |
| 3.50% | 20 | - | - | **39** | - | - | OOM | OOM | - | - | **206** | - | - |
| T10I4D100K | 5 | 11 | **6** | 11 | 7 (77) | **6** | 10 | 10 | 21 | 21 | 21 | 21 | **16** |
| 870 × 100,000 | 1 | 386 | **83** | 360 | 93 (363) | **83** | 375 | 30 | 771 | 771 | 720 | 726 | **468** |
| 1.16% | 0.5 | 1,074 | **185** | 607 | 208 (650) | **185** | 555 | 30 | 2,147 | 2,147 | **1,238** | 1,323 | 1,259 |
| BMS1 | 0.15 | 1,426 | **69** | 592 | 73 (926) | **69** | 303 | 30 | 2,851 | 2,851 | **1,186** | 1,854 | 1,495 |
| 497 × 59,602 | 0.14 | 1,683 | **67** | 647 | 75 (980) | **67** | 311 | 30 | 3,365 | 3,365 | **1,298** | 1,961 | 1,750 |
| 0.51% | 0.12 | 2,374 | **76** | 778 | 79 (909) | **76** | 326 | 30 | 4,747 | 4,747 | **1,560** | 1,819 | 2,450 |
| retail | 5 | 17 | **2** | 12 | **2** (13) | **2** | 6 | 6 | 33 | 33 | 23 | 25 | **18** |
| 16470 × 88,162 | 1 | 160 | **60** | 105 | 63 (114) | **60** | 70 | 30 | 319 | 319 | **218** | 234 | 219 |
| 0.06% | 0.4 | 832 | 275 | 515 | **272** (550) | 275 | 317 | 30 | 1663 | 1663 | **1,071** | 1,143 | 1106 |

**Table 1**: Comparing the number of patterns and the number of nodes explored by CP approaches. (1): ClosedP (2): ClosedP+Jaccard (3) ClosedDiv-Mincov (4): ClosedDiv-Mincov+Jaccard (5): Baseline-CP-Jaccard (6): Picker (7): PatternsTeam. " – " is shown when the time limit is exceeded. OOM : Out Of Memory. In parenthesis, we give the number of intermediate solutions generated by ClosedDiv-Mincov+Jaccard before removing non-diverse patterns.

## 3.2 Number of patterns

We report in Table 1 the number of patterns extracted by ClosedP, ClosedDiv-Mincov, Picker, PatternsTeam, and exact Jaccard CP-based approaches for various datasets and minimum support threshold values. We also give the number of nodes explored by CP-based approaches during the search. This allows evaluation of the number of inconsistent values pruned by the filtering algorithm of each approach.

**A) ClosedDiv-Mincov vs ClosedP.** The results highlight the great discrepancy between the two models with a distinctly lower number of patterns generated by ClosedDiv-Mincov (in the thousands) in comparison to ClosedP (in the millions). On dense and moderately dense datasets (from Chess to Mushroom), the discrepancy is greatly amplified, especially for

small values of $\theta$. For instance, on Chess, the number of patterns for ClosedDiv-Mincov is reduced by 99.9% (from $\sim 50 \cdot 10^6$ solutions to 238) for $\theta = 15\%$. The density of the datasets provides an appropriate explanation for the good performance of ClosedDiv-Mincov. As the number of closed patterns increases with the density, redundancy among these patterns increases as well. On very sparse datasets, ClosedDiv-Mincov still outputs fewer solutions than ClosedP but the difference is less pronounced. This is explainable by the fact that on these datasets, where we have few solutions, almost all patterns are diverse.

**B) ClosedDiv vs exact Jaccard approaches.** As shown in Table 1, the number of patterns obtained with ClosedP+Jaccard is smaller than those of ClosedP and ClosedDiv-Mincov, particularly on dense datasets where the reduction is very impressive as compared to ClosedP (from $\sim 10^6$ solutions to less than 20). More importantly, ClosedP+Jaccard allows completing the extraction on instances where ClosedP runs into *Out Of Memory* (see Chess and Kr-vs-kp with $\theta$ of 10%). On sparse datasets, since the number of solutions is very low the reduction is not huge. Interestingly, on dense datasets, ClosedDiv-Mincov+Jaccard allows filtering out a large number of false positives leading to a high diversity of the final result set. Finally, on instances where Baseline-CP-Jaccard completes the extraction, both ClosedP+Jaccard and Baseline-CP-Jaccard obtain the same diverse pattern set since they explore the search space in the same way; the only difference resides in the way the diversity of the result set is ensured. As the results show, ClosedP+Jaccard and Baseline-CP-Jaccard usually extract fewer patterns than ClosedDiv and ClosedDiv-Mincov+Jaccard. The only exception is the three datasets Heart-cleveland, Splice1 and Mushroom, where ClosedDiv-Mincov+Jaccard extracts fewer patterns.

**C) ClosedDiv-Mincov vs Picker and PatternsTeam.** As the results show in Table 1, Picker usually extracts more patterns than ClosedDiv + Mincov + Jaccard, the most restrictive of the redundancy-minimizing techniques. We would therefore expect pattern sets extracted by Picker to show more redundancy, as minimizing redundancy is not Picker's goal, as mentioned above. The fact that PatternsTeam extracts fewer patterns than Picker is due to the inefficiency of the implementation and therefore cannot be used directly to try to infer statements about the redundancy of pattern sets derived in this manner.

## 3.3 Analysing the impact of the search order on diversity

In this experiment, we analyze the impact of the order in which patterns are processed on the resulting diverse pattern set of ClosedDiv. This evaluation took place in two ways:

1. We compare the two heuristics of variable selection Mincov and Witness and we study the effect of varying the $J_{max}$ parameter. We denote by WIT-FirstSOL our strategy of exploring the witness subtrees.

2. For the MINCOV heuristic, we evaluate the impact of the search order in terms of diversity. Thus, we compare two value selection strategies, one consisting of branching on the value 1 (resp. the value 0) of each variable before branching on the 0 (resp. the 1) denoted by MINCOV-1 (resp. MINCOV-0) which corresponds to the default value selection strategy of MINCOV.

### 3.3.1 Comparing variable selection heuristics

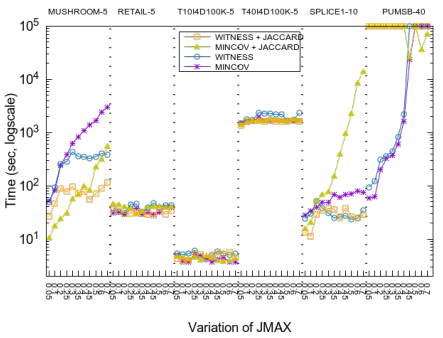
Figure 2 shows detailed results when varying $J_{max}$ from 0.05 to 0.7. These results show that the size of the history has a great impact on runtime. The size of the history $\mathcal{H}$ grows rapidly with the increase of $J_{max}$. This induces significant additional costs in the lower and upper-bound computations. Notice, in practice, the user will only be interested in small values of $J_{max}$ because the diversity of patterns is maximal and the number of patterns returned becomes manageable. This explains why we fixed the value of $J_{max}$ to 0.05 in our experiments. These results also show that using the exact Jaccard test becomes very efficient when dealing with a history of reasonable size.

Figures 2a and 2c compare MINCOV against WITNESS. On dense datasets, both heuristics perform very similarly, with a slight advantage for WITNESS for $J_{max} \geq 0.45$. Interestingly, the number of witness patterns mined is very low (less than 1% on most instances) compared to the number of diverse patterns extracted by both heuristics (see Figure 3). Consequently, WITNESS almost behaves as MINCOV. This is due to the WIT-FIRSTSOL strategy which avoids the complete exploration of the different witness subtrees encountered during the search.

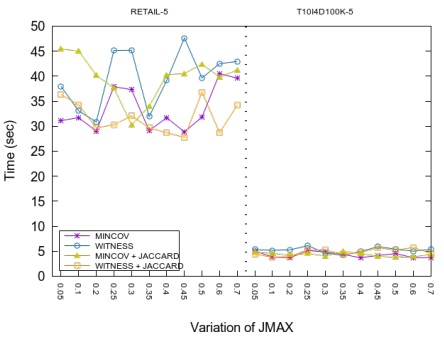
When comparing MINCOV and WITNESS against the two variants using the exact Jaccard test, in the beginning, we observe that CLOSEDDIV-MINCOV and CLOSEDDIV-WITNESS clearly dominate CLOSEDDIV-MINCOV+JACCARD and CLOSEDDIV-WITNESS+JACCARD. But as soon as the size of their history $\mathcal{H}$ becomes sufficiently large ($\geq 1000$ for MINCOV and $\geq 6500$ for WITNESS), the two last variants become very efficient, particularly for higher values of $J_{max}$. Moreover, as depicted in Figure 3, the size of the history $\mathcal{H}$ for MINCOV and WITNESS grows very quickly with the increase of $J_{max}$ (up to $\sim 9 \cdot 10^4$ for MINCOV), while for the two variants using the exact Jaccard test, the size of $\mathcal{H}$ remains reasonably small.

Finally, CLOSEDDIV-MINCOV+JACCARD provides runtime benefits compared to CLOSEDDIV-WITNESS+JACCARD. This is due to the large number of intermediate solutions generated by WITNESS, while for MINCOV this number is low, thus reducing the overhead induced by the test of the exact Jaccard constraint.

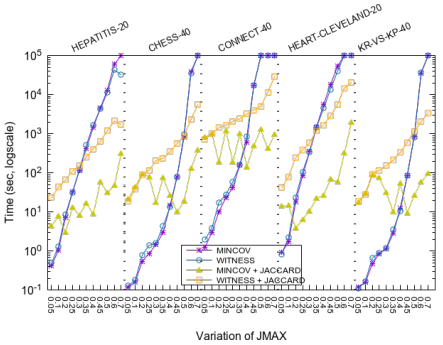
For the moderately dense datasets, CLOSEDDIV-WITNESS largely dominates CLOSEDDIV-MINCOV, particularly for $J_{max} \geq 0.25$. Importantly, CLOSEDDIV-WITNESS returns a few patterns and most of them are witness ones: $\sim$ 5% for MUSHROOM and 60% for SPLICE1. The behavior of CLOSEDDIV-MINCOV+JACCARD is a bit more complex, particularly for SPLICE1, due to the size of the history which increases exponentially with



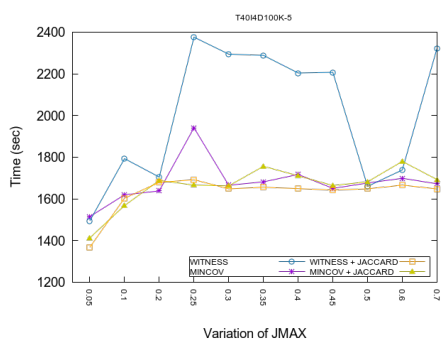

(a) Moderately dense and sparse datasets.




(b) Zoom view.




(c) Dense datasets.




(d) T40-5 instance : zoom view.

**Fig. 2**: CPU-time analysis of ClosedDiv-Mincov and ClosedDiv-Witness with and without exact Jaccard test.

the increase of $J_{max}$ (see Figure 3). This probably explains why this variant is more time-consuming. Finally, the runtimes of ClosedDiv-Witness and ClosedDiv-Witness+Jaccard are comparable.

For the sparse datasets, all heuristics are comparable in runtime and number of diverse patterns mined, with a slight advantage for the two variants using the exact Jaccard test (see zoom views in Figure 2).

### 3.3.2 Impact of the search order on diversity

Each of the three branching strategies (i.e. Mincov-1, Mincov-0 and Witness) induces a different search order of patterns, thus resulting in different outputs and the overall diversity of each result set may be impacted. To better understand the impact of this order, we represent in Figure 4 different graphics comparing the diversity of the pattern sets obtained by these strategies. Thus, for each dataset (Figures 4a, 4b and 4c), we show the Jaccard distribution of the patterns set (i.e. $CDF$) on the left side and the average pairwise Jaccard index between the $m$ first discovered patterns (denoted by

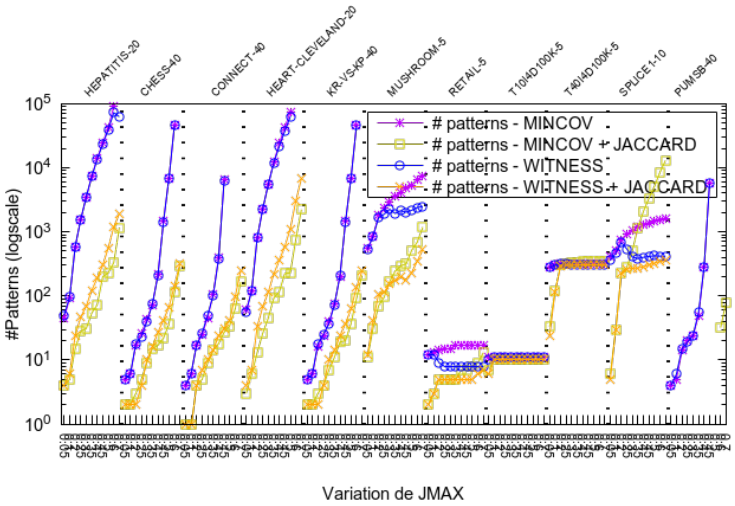


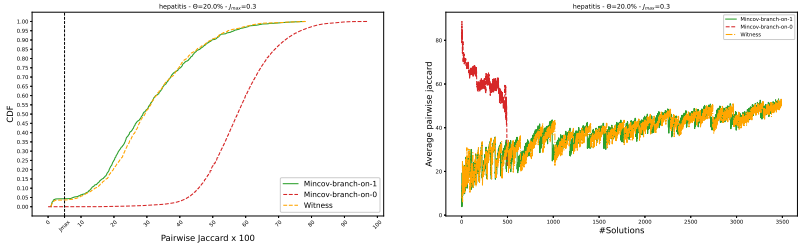

**Fig. 3**: Comparing # of patterns of ClosedDiv-Mincov and ClosedDiv-Witness with and without exact Jaccard test.

$\text{Jaccard}_{mean}$) on the right side. For this second evaluation, we consider the set $\mathcal{H}^m = \{H_1, \ldots, H_m\}$ of the $m$ first patterns discovered and we compute $\text{Jaccard}_{mean}(\mathcal{H}^m)$ as

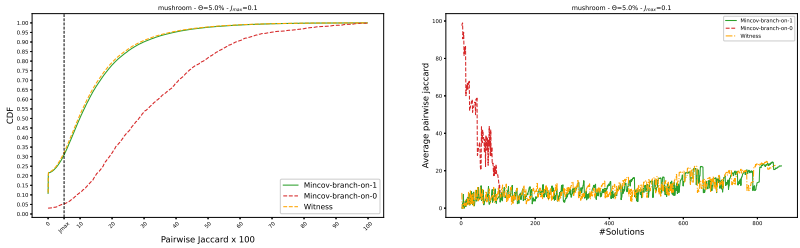
$$\text{Jaccard}_{mean}(\mathcal{H}^m) = \frac{2}{m \cdot (m-1)} \cdot \sum_{1 \leq i < j \leq m} Jac(H_i, H_j).$$

As we can see, the two strategies Mincov-1 and Witness have similar results in terms of diversity. On dense datasets (here Hepatitis, Figure 4a), the graphs thus show that the two strategies have similar Jaccard distributions, with an advantage sometimes for one and sometimes for the other. However, the analysis of the average of the pairwise patterns Jaccard gives a slight advantage to Witness compared to Mincov strategies. On the other hand, branching on "0" with Mincov-0 leads to a degradation of global diversity. This can be explained by the fact that this strategy first extracts patterns of small size. These patterns have large coverages, which accentuate the overlaps as shown by the average pairwise of the patterns Jaccard. However, when increasing the output sets, we observe an improvement in the diversity for Mincov-0, even if this diversity remains less interesting than that of Mincov-1 and Witness.

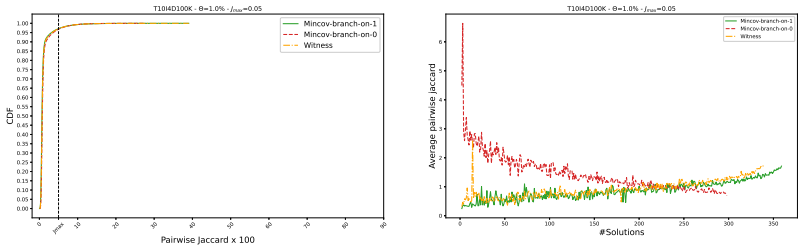
For the moderately dense datasets (here Mushroom, Figure 4b), we observe a similar behavior, with however a slight advantage for Mincov-1. For sparse datasets (here T10I4D100K, Figure 4c), the different strategies have very similar Jaccard index distributions. This is due to the limited number of




(a) Hepatitis.




(b) Mushroom.




(c) T10I4D100K.

**Fig. 4**: Analyzing the impact of the search order of patterns on the output sets in terms of diversity. We consider the distribution of pairwise Jaccard values of the pattern sets as well as the average pairwise Jaccard index between the m first discovered patterns. We evaluate the three branching strategies Mincov-1, Mincov-0 and Witness-WIT-FirstSOL.

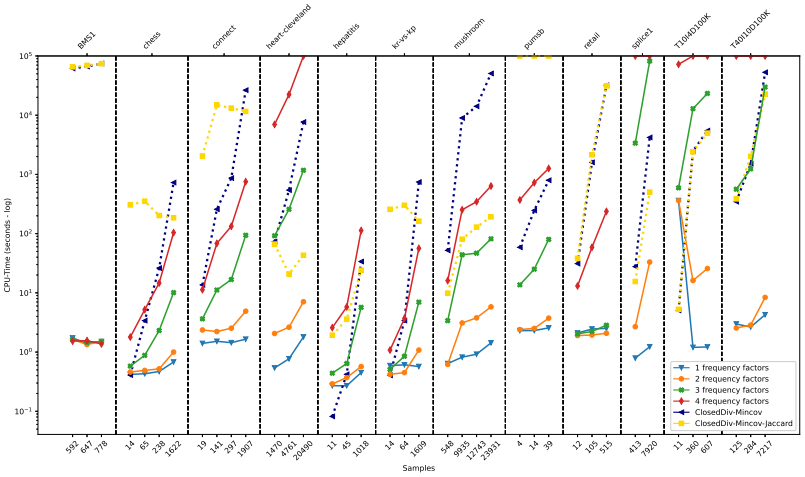
redundancies in these datasets which allows the extraction of diverse pattern sets regardless of the strategy used. However, we observe that the two strategies Mincov-1 and Witness obtain a better average diversity on the first 200 patterns extracted and that the result is reversed in favor of Mincov-0 from the 240th pattern.

## References

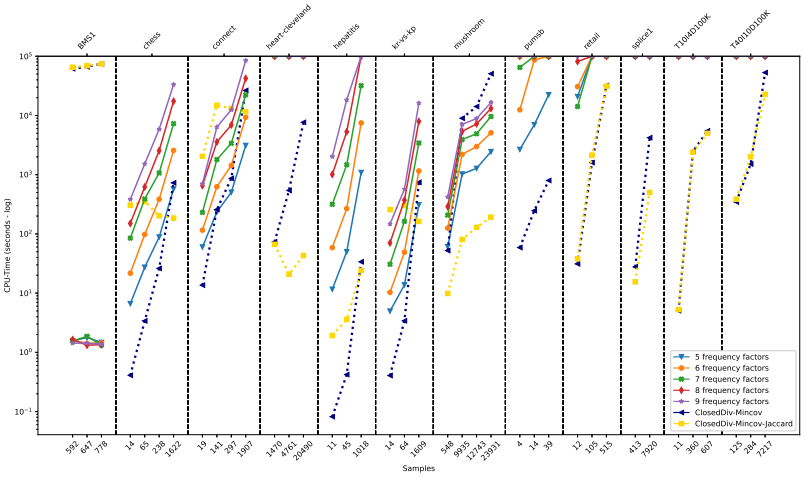
Hien A, Loudni S, Aribi N, et al. (2020) A relaxation-based approach for mining diverse closed patterns. In: Hutter F, Kersting K, Lijffijt J, et al. (eds) Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2020, Ghent, Belgium, September 14-18, 2020, Proceedings, Part I, Lecture Notes in Computer Science, vol 12457. Springer, pp 36–54, https://doi.org/10.1007/978-3-030-67658-2_3, URL https://doi.org/10.1007/978-3-030-67658-2_3

Lazaar N, Lebbah Y, Loudni S, et al. (2016) A global constraint for closed frequent pattern mining. In: Rueher M (ed) Principles and Practice of Constraint Programming - 22nd International Conference, CP 2016, Toulouse, France, September 5-9, 2016, Proceedings, Lecture Notes in Computer Science, vol 9892. Springer, pp 333–349, https://doi.org/10.1007/978-3-319-44953-1_22, URL https://doi.org/10.1007/978-3-319-44953-1_22

# A CPU-time Analysis: ClosedDiv-Mincov and CFTP




(a) frequency factor $c \in \{1, 2, 3, 4\}$

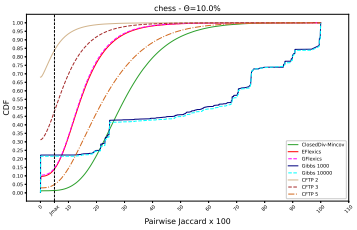



(b) frequency factor $c \in \{5, 6, 7, 8, 9\}$

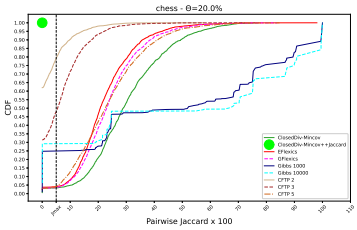
**Fig. 5**: CPU-time analysis (complete results): ClosedDiv ($J_{max} = 0.5$) *vs.* CFTP.

## B Overall Pairwise Redundancy of ClosedDiv, ClosedDiv+Jaccard, Flexics, CFTP and Gibbs

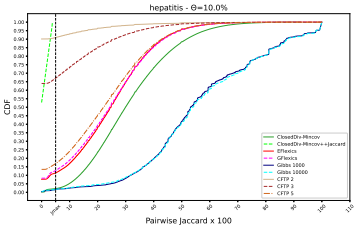
We have split the figures of this section into two sub-groups: Figure 6 shows the results for dense datasets and Figure 7 shows the results for moderately dense and sparse datasets.



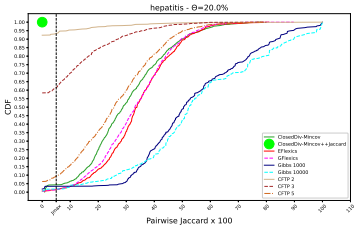

(a) Chess ($\theta = 10\%$)



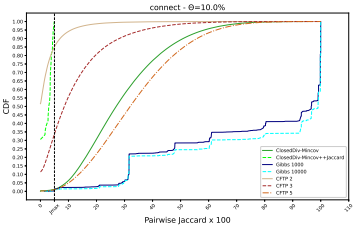

(b) Chess ($\theta = 20\%$)



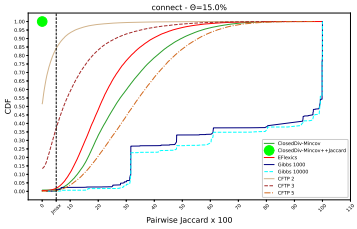

(c) Hepatitis ($\theta = 10\%$)




(d) Hepatitis ($\theta = 20\%$)

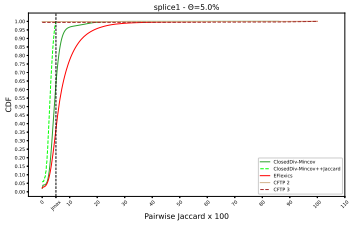



(e) Connect ($\theta = 10\%$)

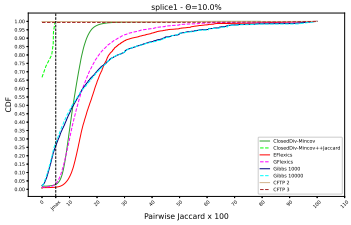



(f) Connect ($\theta = 15\%$)

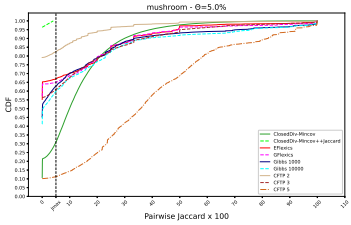
**Fig. 6**: Assessing overall pairwise redundancy for dense datasets: ClosedDiv-Mincov and ClosedDiv-Mincov+Jaccard vs sampling-based approaches.



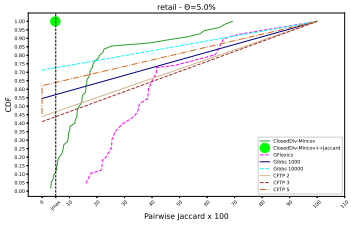

(a) SPLICE1 ($\theta = 5\%$)



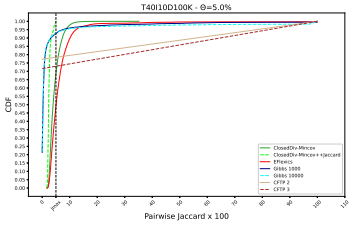

(b) SPLICE1 ($\theta = 10\%$)



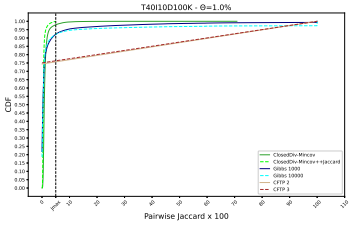

(c) MUSHROOM ($\theta = 5\%$)



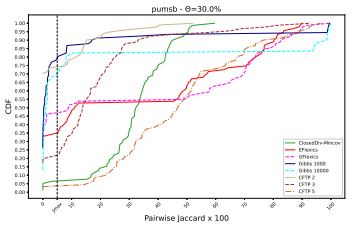

(d) RETAIL ($\theta = 5\%$)



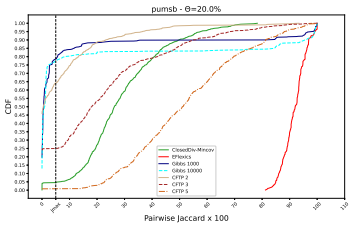

(e) T40I10D100K ($\theta = 5\%$)




(f) T40I10D100K ($\theta = 1\%$)



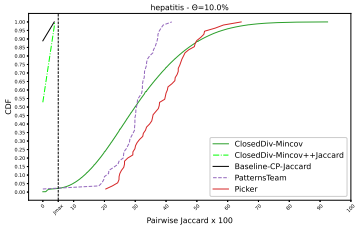

(g) PUMSB ($\theta = 30\%$)



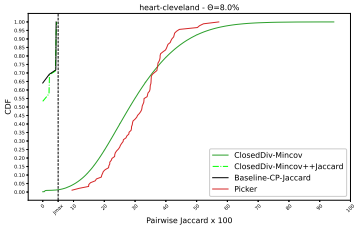

(h) PUMSB ($\theta = 20\%$)

**Fig. 7**: Assessing overall pairwise redundancy for moderately dense and sparse datasets: CLOSEDDIV-MINCOV and CLOSEDDIV-MINCOV+JACCARD vs sampling-based approaches.

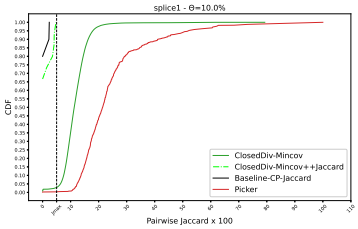
# C Overall Pairwise Redundancy of ClosedDiv, Picker, PatternsTeam and Baseline-CP-Jaccard



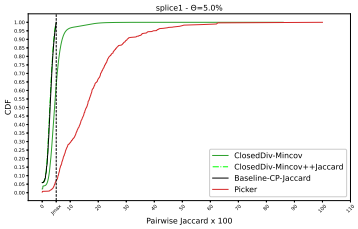

(a) Hepatitis ($\theta = 10\%$)



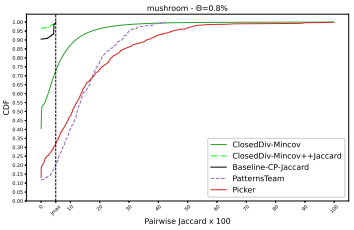

(b) Heart-cleaveland ($\theta = 8\%$)



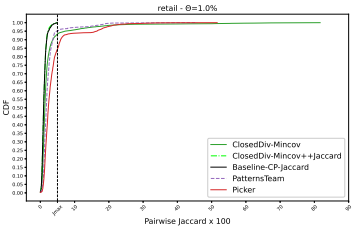

(c) Splice1 ($\theta = 10\%$)



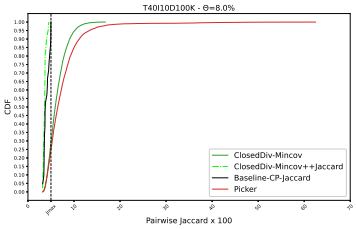

(d) Splice1 ($\theta = 5\%$)



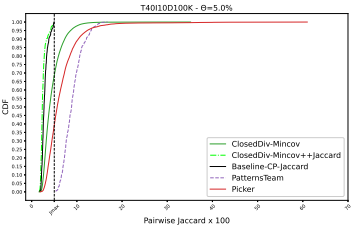

(e) Mushroom ($\theta = 8\%$)




(f) Retail ($\theta = 1\%$)




(g) T40I10D100K ($\theta = 8\%$)

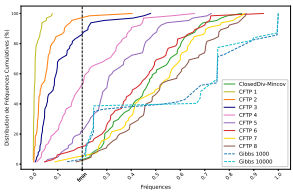



(h) T40I10D100K ($\theta = 5\%$)

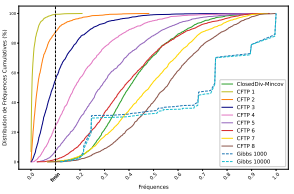
**Fig. 8**: Assessing overall pairwise redundancy of ClosedDiv, ClosedDiv-Mincov+Jaccard, Picker, PatternsTeam and Baseline-CP-Jaccard.

# D Frequency patterns analysis

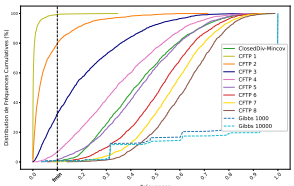
In this section, we compare the Cumulative Frequency Distribution of ClosedDiv patterns and those of the sampling methods.



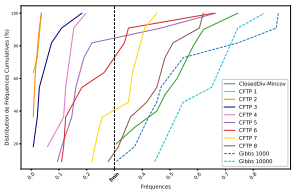
(a) Chess ($\theta = 20\%$)



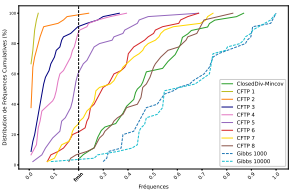
(b) Chess ($\theta = 10\%$)



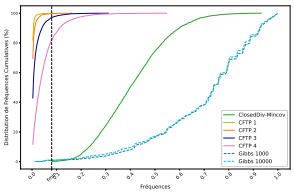
(c) Connect ($\theta = 10\%$)



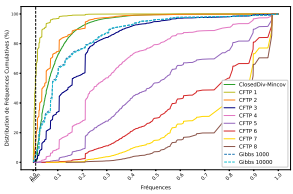
(d) Hepatitis ($\theta = 30\%$)



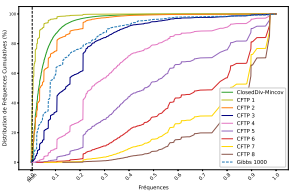
(e) Hepatitis ($\theta = 20\%$)



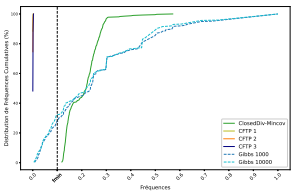
(f) heart-clevelan ($\theta = 8\%$)



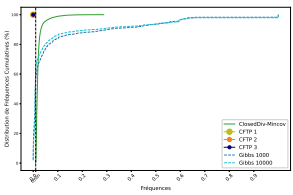
(g) Mushroom ($\theta = 1\%$)



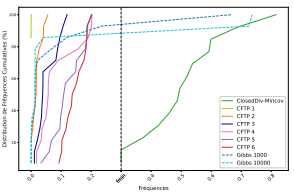
(h) Mushroom ($\theta = 0.5\%$)



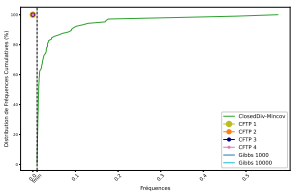
(i) Splice1 ($\theta = 10\%$)



(j) T40I10D100K ($\theta = 1\%$)



(k) Pumsb ($\theta = 30\%$)



(l) Retail ($\theta = 10\%$)

**Fig. 9**: Distribution of FCIs mined by both ClosedDiv and sampling-based approaches.