SUPPLEMENTARY DOCUMENT

The supplementary material for Paper # 490 is present in this document.

A. SYMMETRIC REFINEMENT - MINIMUM RBS

Symmetric Refinement strategy ensures that for a PASs $\mathbb{A}$, the number of resultant blocks is kept to a minimum. Let the domain of b along $\mathbb{A}$ be denoted as $D^{\mathbb{A}}(b)$. Further, let $S_b^{\mathbb{A}}$ be a relation associated with the points in $D^{\mathbb{A}}(b)$. For a pair of points $t_1, t_2 \in D^{\mathbb{A}}(b)$, we say $t_1 S_b^{\mathbb{A}} t_2$ iff the projection t_1 and t_2 along the rest of the attributes i.e. $\mathbb{U} \setminus \mathbb{A}$ is identical. It is easy to verify that $S_b^{\mathbb{A}}$ forms an *equivalence relation.* For an equivalence relation, the *quotient set* of the relation gives the minimum partition.

Lemma 11.1. *The Symmetric Refinement algorithm returns the quotient set of* $D^{\mathbb{A}}(b)$ *by* $S_b^{\mathbb{A}}$.

The proof follows from the fact that Symmetric Refinement algorithm uses a hashmap, which enables grouping of points in $D^{\mathbb{A}}(b)$ together such that their projection on $\mathbb{U} \setminus \mathbb{A}$ are identical. Hence, for a PAS, the symmetric refinement algorithm produces the quotient set of $S_b^{\mathbb{A}}$, and hence returns the refinement with minimum number of blocks.

B. PROJECTION SUBSPACE DIVISION

Projection Regions - Poset

Let $\mathbb{V}$ represent the set of all possible vectors. Further, let $\mathbb{Q}$ denote the collection of CPBs, where there is a projection-block q associated with each vector $v \in \mathbb{V}$. Therefore, $\mathbb{P}^* \subseteq \mathbb{Q}$. Let the subset of $\mathbb{V}$ corresponding to the elements in $\mathbb{P}^*$ be denoted as $\mathbb{V}^*$. Each position in vector v can have one of the three possibilities among $0, 1, \times$, and at least one position needs to mandatorily be 1. Therefore, $\mathbb{Q}$ comprises $3^m - 2^m$ elements. Note that $\mathbb{Q}$ forms a *partial-order* with respect to the subset relation, and can therefore be represented by a Hasse Diagram. As an exemplar, the Hasse Diagram for an $m = 3$ case is shown in Figure 8 (for simplicity, the elements of $\mathbb{V}$ are shown instead of $\mathbb{Q}$).

`Opt-PSD` Algorithm

The degree of the DG has a proportional impact on the number of CPBs constructed. To see this behaviour, the number of CPBs for `Opt-PSD` for a few general DGs are shown in Table 8.

Proof of Correctness

The correctness of `Opt-PSD` algorithm follows from the following:

- it starts from the top nodes of the Hasse diagram and recursively refines them. Therefore, it continues to cover all the elements of $\overline{\mathbb{R}}$.
- the PRBs that are related to a common constraint are split by restricted powerset enumeration ensuring that they are mutually disjoint.

Hence, the algorithm does restricted enumeration depending on vertex's neighbours, or in other words it takes into account which PRBs co-appear in a constraint.

Algorithm 2: Optimal Projection Subspace Division

```
Input: Division Graph $G$
Output: Optimal Vectors-set $\mathbb{V}^*$
$toBeSplit \leftarrow \emptyset$;
$visited \leftarrow \emptyset$;
for $\overline{r}$ in $\overline{\mathbb{R}}$ do
    $visited \leftarrow visited \cup \overline{r}\ v_{init} \leftarrow \{\times\}^m, v_{init}(\overline{r}) \leftarrow 1$;
    $toBeSplit \leftarrow \{v_{init}\}$;
    while $toBeSplit \neq \emptyset$ do
        $v \leftarrow toBeSplit.pop()$;
        $pivot, targets \leftarrow getPivot(G, v)$;
        if $pivot$ exists then
            $toBeSplit \leftarrow$
              $toBeSplit \cup Split(v, pivot, targets, visited)$;
        else
            $\mathbb{V}^* \leftarrow \mathbb{V}^* \cup \{v\}$;
return $\mathbb{V}^*$;
```

```
Function Split($v, pivot, targets, visited$):
    $splitSet \leftarrow \emptyset$;
    for $\overline{r} \in targets$ do
        if $\overline{r} \in visited$ then
            $v_r \leftarrow 0$;
            remove $\overline{r}$ from $targets$;
    if $targets = \emptyset$ then
        return $v$;
    $powerset \leftarrow$ generate powerset enumeration of $targets$;
    for $s \in powerset$ do
        $new\_v \leftarrow v$;
        $new\_v_r \leftarrow 1, \forall \overline{r} \in s$;
        $new\_v_r \leftarrow 0, \forall \overline{r} \in targets \setminus s$;
        $splitSet \leftarrow splitSet \cup new\_v_r$;
    return $splitSet$;
```

Table 8: No. of CPBs in `Opt-PSD`

Division Graph	No. of CPBs
Empty Graph ($\overline{K_m}$)	m
Path Graph (P_m)	$\frac{1}{2}m(m+1)$
Cycle Graph (C_m)	$m^2 - m + 1$
Star ($K_{1,m-1}$)	$2^{m-1} + m - 1$
Complete Graph (K_m)	$2^m - 1$

Proof of Optimality

We now prove that `Opt-PSD` produces the optimal division. Firstly, state the following lemma.

Lemma 11.2. *If a pair of CPBs in* $\mathbb{P}$, p_1 *and* p_2, *map to identical sets in* $\overline{\mathbb{R}}$, *they can be combined into a single element* $p_1 \cup p_2$, *without violating either condition.*

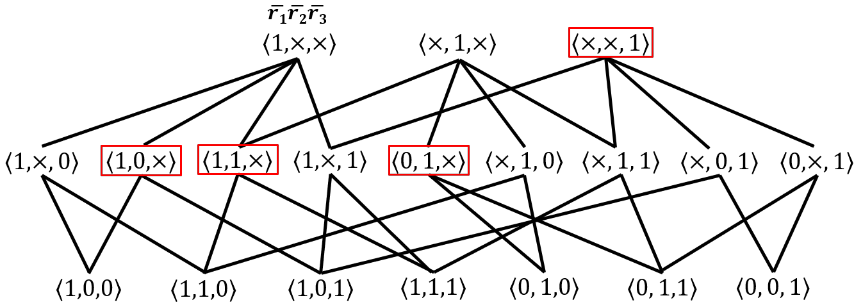

Figure 8: Hasse Diagram

Proof. We are given that p_1 and $p_2 \in P$ are such that $p_1 L\bar{r} \Leftrightarrow p_2 L\bar{r}$ for $s \in S$. We need to prove that replacing p_1 and p_2 with $p_{1,2} = p_1 \cup p_2$ in P does not violate any of the two conditions.

- **Condition 1:** It is required that each $\bar{r} \in \overline{\mathbb{R}}$ is expressible as union of related elements of P through L.
  If $(p_1, \bar{r}) \notin L$, then $(p_2, \bar{r}) \notin L$ (and vice versa). Hence, the expression for $\bar{r}$ remains unaltered.
  If $(p_1, \bar{r}) \in L$, then $(p_2, \bar{r}) \in L$ (and vice versa). Let $\rho = \{p \in P \setminus \{p_1, p_2\} : pL\bar{r}\}$. Then, $\bar{r} = p_1 \cup p_2 \bigcup_{p\in\rho} p$. After replacing p_1 and p_2 with $p_{1,2}$, the expression would become $\bar{r} = p_{1,2} \bigcup_{p\in\rho} p$.
- **Condition 2:** Let c be any $c \in \mathbb{C}$ such that $(p_1, c) \in M \circ L^{\mathbb{A}}$ (and $(p_2, c) \in M \circ L^{\mathbb{A}}$). It is easy to see that (from Condition 2) p_1 will be disjoint with all the other elements of P that are related to c through $M \circ L^{\mathbb{A}}$. That is,
  $$p_1 \cap p' = \emptyset, \forall p' \in P \setminus \{p_1\} : (p, c) \in M \circ L^{\mathbb{A}}$$
  Likewise, p_2 will also be disjoint with all the other elements of P that are related to c. Therefore, on replacing p_1 and p_2 with their union $p_{1,2}$, $p_{1,2}$ will continue to remain disjoint with all the other elements of P that are related to c.

□

For a CPB $p \in \mathbb{P}$, consider the subset s of points:

$$s = \bigcap_{\bar{r}: v_p(\bar{r})=1} \bar{r} \setminus \bigcup_{\bar{r}': v_p(\bar{r}')=0,\times} \bar{r}'$$

Note that with this definition, $s \subseteq p$ and cannot overlap with any $p' \in \mathbb{P} \setminus \{p\}$. This restriction leads to the following lemma:

Lemma 11.3. *Given* $(\mathbb{P}, L)$ *returned by* `Opt-PSD`, $\forall p \in \mathbb{P}$, *there exists a point* $u \in p$ *such that* $u \notin p', \forall p' \in \mathbb{P} \setminus \{p\}$.

We use this observation to prove that `Opt-PSD` returns an optimal division, and further, that this optimal division is *unique*.

Lemma 11.4. `Opt-PSD` *returns the unique optimal division.*

Proof. We give a brief sketch of the proof here.
Let $(\mathbb{P}, L)$ be the division provided by `Opt-PSD`, and let there be another division $(\mathbb{P}', L')$ such that $|\mathbb{P}'| \leq \mathbb{P}$.

$$\implies \exists u \in p_1, v \in p_2 (\neq p_1) \text{ for some } p_1, p_2 \in \mathbb{P}, \text{ where } p_1 L\bar{r}_1, p_2 L\bar{r}_2,$$
$$\bar{r}_1, \bar{r}_2 \in \overline{\mathbb{R}}, \text{ such that } u, v \in p', p'L'\bar{r}_1, p'L'\bar{r}_2 \text{ for some } p' \in \mathbb{P}'.$$

Case (1) $\bar{r}_1 = \bar{r}_2 = \bar{r}$: Since $p_1 L\bar{r}$ and $p_2 L\bar{r}$,

$$\implies \exists c \in C \text{ such that } \bar{r}Mc, \bar{r}'Mc, \text{ for some } \bar{r}' \in \overline{\mathbb{R}} \text{ and}$$
$$(p_1, \bar{r}') \in L, (p_2, \bar{r}') \notin L \text{ (wlog) (using Lemma 11.2)}$$
$$\implies v \notin \bar{r}', \text{ otherwise there would exist } p_3 \in \mathbb{P} \text{ such that } v \in p_3;$$
$$p_2 \cap p_3 \neq \emptyset \text{ and } p_3 L\bar{r}' \text{ would imply Condition 2 violation.}$$
$$\implies \exists p'' \in \mathbb{P}' \text{ such that } p''L'\bar{r}', u \in p'' \text{ and } v \notin p''.$$
$$Since, p' \cap p'' \neq \emptyset \text{ and } (p', c), (p'', c) \in M \circ L'$$
$$\text{Hence, contradiction (Condition 2 violation).}$$

Case (2) $\bar{r}_1 \neq \bar{r}_2$:
(2a): $u \in p_1 \setminus p_2$ (or $v \in p_2 \setminus p_1$, wlog)
Since, $u \in p_1, p_1 L\bar{r}_2$, therefore $u \in \bar{r}_2$
$\implies \exists p_3 \in \mathbb{P}$ such that $u \in p_3$ and $p_3 L\bar{r}_2$
p_2, p_3, p' are such that $u \in p_3, v \in p_2, u, v \in p', p_2 L\bar{r}_2, p_3 L\bar{r}_2, p'L'\bar{r}_2$. This is not possible using result of Case (1). Contradiction.
(2b): $u, v \in p_1 \cap p_2$
p_1, p_2 has at least one point each that is absent in all the other CPBs (using Lemma 11.3). Therefore, if u, v, which are present in $p_1 \cap p_2$ are merged in $\mathbb{P}'$, then $|\mathbb{P}'| > |\mathbb{P}|$. Contradiction.
Hence, `Opt-PSD` gives the optimal division. □

C. WORKLOAD DECOMPOSITION

Template-based Decomposition (TD). Here, the decomposition algorithm assumes conflicting pairs are defined at a template level. That is, two constraints conflict if their PASs partially intersect. The reason we consider TD is to remove any coincidental performance benefit that may have been obtained thanks to the specific filter predicate constants present in the original workload. Table 9 shows the number of workloads obtained for the four tables with this artificially expanded definition of conflict. We observe that even here, just 8 sub-workloads are sufficient for producing compatibility. Finally, again thanks to decomposition, both the summary generation times and the summary sizes are extremely small.

D. PRIOR WORK COMPARISON

DataSynth's Unconstructible Solution. Consider a toy example with the following pair of projection-inclusive constraints (PICs)

Table 9: Workload Decomposition - TD

Table	Sub-Workload Sizes	Aggregate Summary Time	Aggregate Summary Size
SS	10,10,8,8,5,5,4,3	70 s	109 kB
CS	9,7,4,4,4	14 s	117 kB
WS	9,9,6,5	7 s	41 kB
INV	6,2	2 s	16 kB

on the item table from TPC-DS:

$$PIC\ 1 : \langle 4 \le i\_class\_id < 12, i\_class\_id, 6876, 8 \rangle$$

$$PIC\ 2 : \langle 8 \le i\_class\_id < 16, i\_class\_id, 4490, 8 \rangle$$

For this scenario, DataSynth produced the following interval-based solution:

Table 10: LP Solution from DataSynth

Interval	Range	Total, Distinct Row Card.
I_1	$i\_class\_id < 4$	**11124,0**
I_2	$4 \le i\_class\_id < 8$	**2386,0**
I_3	$8 \le i\_class\_id < 12$	4490,8
I_4	$12 \le i\_class\_id < 16$	0, 0
I_5	$i\_class\_id > 16$	0,0

Here, in the first two intervals, the distinct row cardinality is zero while the total row cardinality is positive, a clear impossibility.