Scalable Incremental Package Maintenance

Vasileios Vittis*, Azza Abouzied■, Peter Haas*, Alexandra Meliou*

* University of Massachusetts, Amherst
■ New York University, Abu Dhabi

Background

Can I have the perfect meal plan?

- Each meal is gluten-free — **Base** Constraint
- Has 3 to 6 meals — **Cardinality** Constraint
- The total calories are between 2.000 and 2.500 — **Global** Constraint
- **With minimal saturated fats** — **Objective**

Language Specification: Package Query Language (PaQL)

```
SELECT       PACKAGE ( R ) AS P
FROM         Recipes R REPEAT 0
WHERE        R.gluten = 'free'
SUCH THAT    COUNT( P.* ) BETWEEN 3 and 6 AND
             SUM( P.kcal ) BETWEEN 2 and 2.5
MINIMIZE     SUM( P.saturated_fat )
```

Translation: PaQL to Integer Linear Programming

COUNT(P.*) >= 3 ⇨ $\sum_{i=1}^{n} x_i \geq 3$

SUM(P.kcal) >= 2 ⇨ $\sum_{i=1}^{n} x_i \cdot t_{i,kcal} \geq 2$

Package...

...is a collection of tuples with certain ***global properties*** define on the ***collection as a whole***

Package Builder*

scalable prototype system to find optimal packages

Challenge

What if I add more recipes?

Do I have to calculate everything from the scratch?

Let's update Package Builder to support dynamic environments

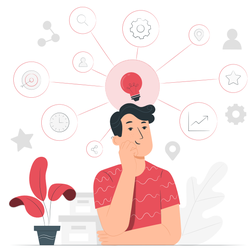
Problem in Streaming Setting:
Recomputing of the optimal solution from scratch is expensive for big data.

SCAN ME

Intuition

Let's make two rules!

1. **Reduce the data to the most "favorable" meals using Multiple Layers Skyline (MLS)**
2. **Evaluate new tuples based on their trade-offs**



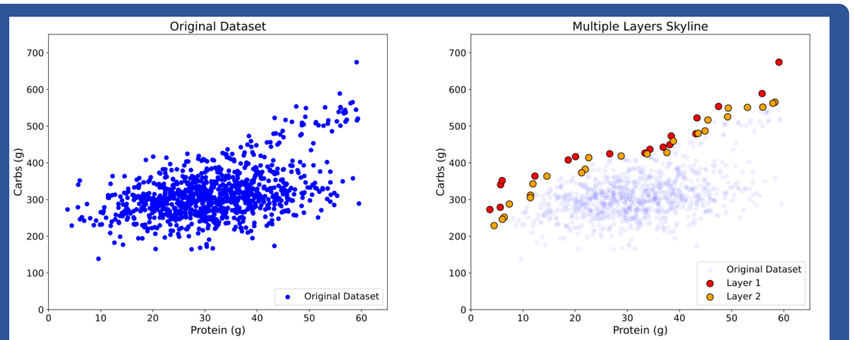
Warm-up

One time Data Reduction

Favorable points have:

- **HIGH Carbs (g)**
- **LOW Protein (g)**

Also applicable for offline (static) setting

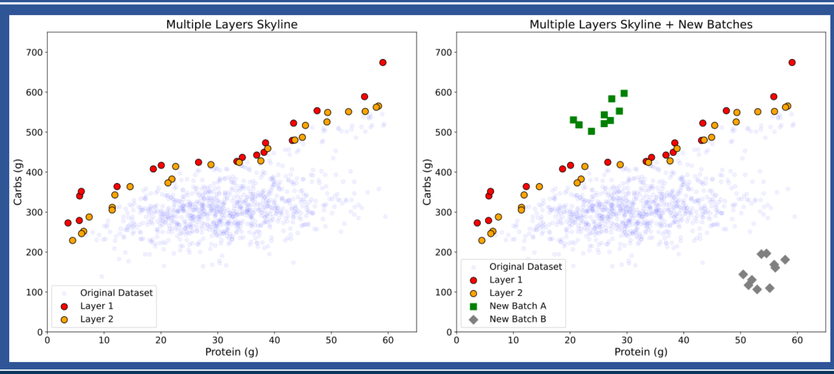


Data Size: 1000 | Execution Time: 0.105 sec | Objective Value: 1262.973

Data Size: 42 | Execution Time: 0.015 sec | Objective Value: 1262.973

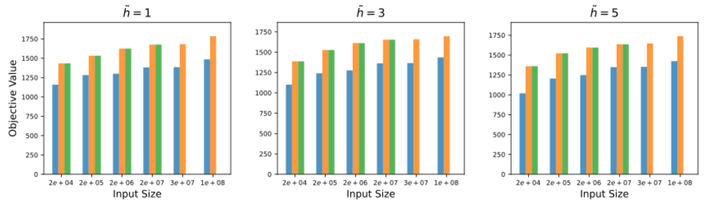
Online Stream

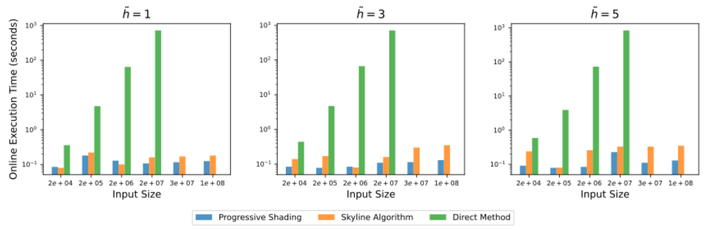
1. **Incrementally Maintain Online MLS**
2. **Update the MLS when necessary (based on their characteristics)**



Experimental Evaluation

```
SELECT       PACKAGE ( R ) AS P
FROM         Recipes R REPEAT 0
SUCH THAT    SUM( P.protein ) <= ub* AND
             COUNT(P.*) = 2
MAXIMIZE     SUM( P.carbs );
```





*ub value changes based on the hardness level (h)

+Progressive Shading

Results

- ✓ Scaling Package Queries to Millions of Tuples
- ✓ Adaptability to different Package Queries
- ✓ Incremental Package Evaluation for table insertions
- ✓ Exact Guarantees for the Objective Value
- ✓ Orders-of-magnitude Faster Running Time

Future Work

What if I change my requirements or my objectives?

→ IPM for small query changes

Are there any strategies with approximation bounds?

→ Keep track of an effective set of high-quality tuples

Database setting → Manage Updates and Deletions

* Scalable computation of high-order optimization queries, M.Brucato, A.Abouzied, A.Meliou (2019)

\+ Scaling Package Queries to a Billion Tuples via Hierarchical Partitioning and Customized Optimization A. L.Mai, P.Wang, M.Brucato, A.Abouzied, P.J.Haas, A.Meliou(2023)