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

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

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

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?

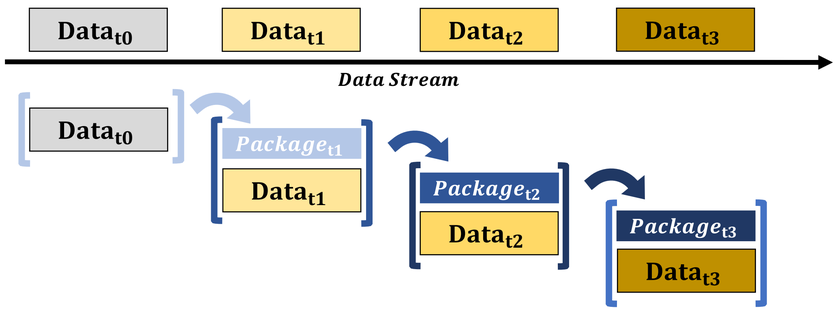
Do I have to calculate everything from the scratch?

Let's update Package Builder to support dynamic environments

Problem:
Recomputing the optimal solution from scratch is expensive for big data.

Incremental Solution Algorithms

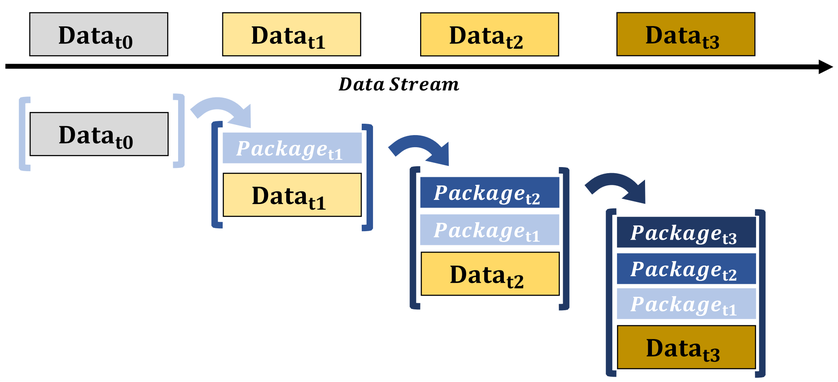
Greedy Algorithm → Last Solution + New Data



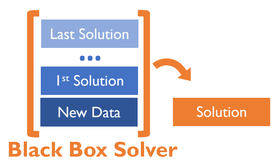
Intuition

Create new package from good tuples in the new data plus the high-quality tuples in the prior package

Accumulative Algorithm → All Past Solutions + New Data



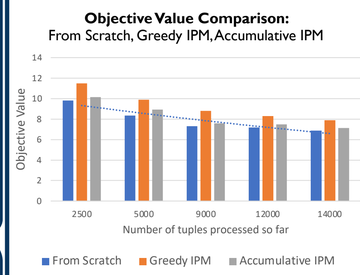
Intuition

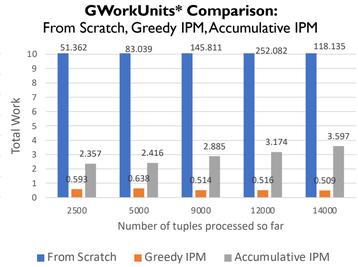


Create new package from good tuples in the new data plus the high-quality tuples in all packages seen, so more likely to find a superior set of tuples

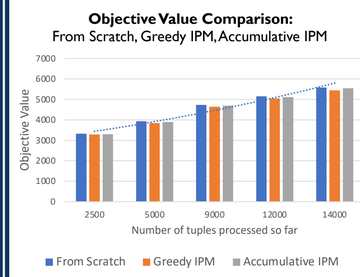
Experimental Evaluation

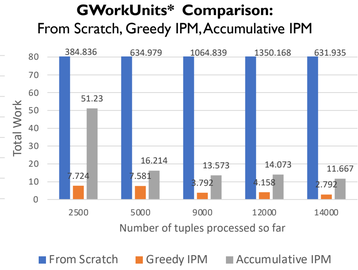
```
SELECT      PACKAGE ( R ) AS P
FROM        Recipes R REPEAT 0
SUCH THAT   SUM( P.calories ) BETWEEN 2200 AND 2450 AND
            SUM( P.carbs ) BETWEEN  225 AND 325 AND
            SUM( P.fat ) BETWEEN 0 AND 50
MINIMIZE    SUM( P.cost );
```






```
SELECT      PACKAGE ( R ) AS P
FROM        Recipes R REPEAT 0
SUCH THAT   SUM( P.protein ) BETWEEN 70 AND 80 AND
            SUM( P.carbs ) BETWEEN  0 AND 70 AND
            SUM( P.fat ) BETWEEN 90 AND 110
            COUNT( P.* ) BETWEEN 5 AND 20
MAXIMIZE    SUM( P.calories );
```





Good approximation in objective value with orders-of-magnitude faster running time

*Gurobi Work Unit ≈ deterministic performance metric based on the hardware

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)