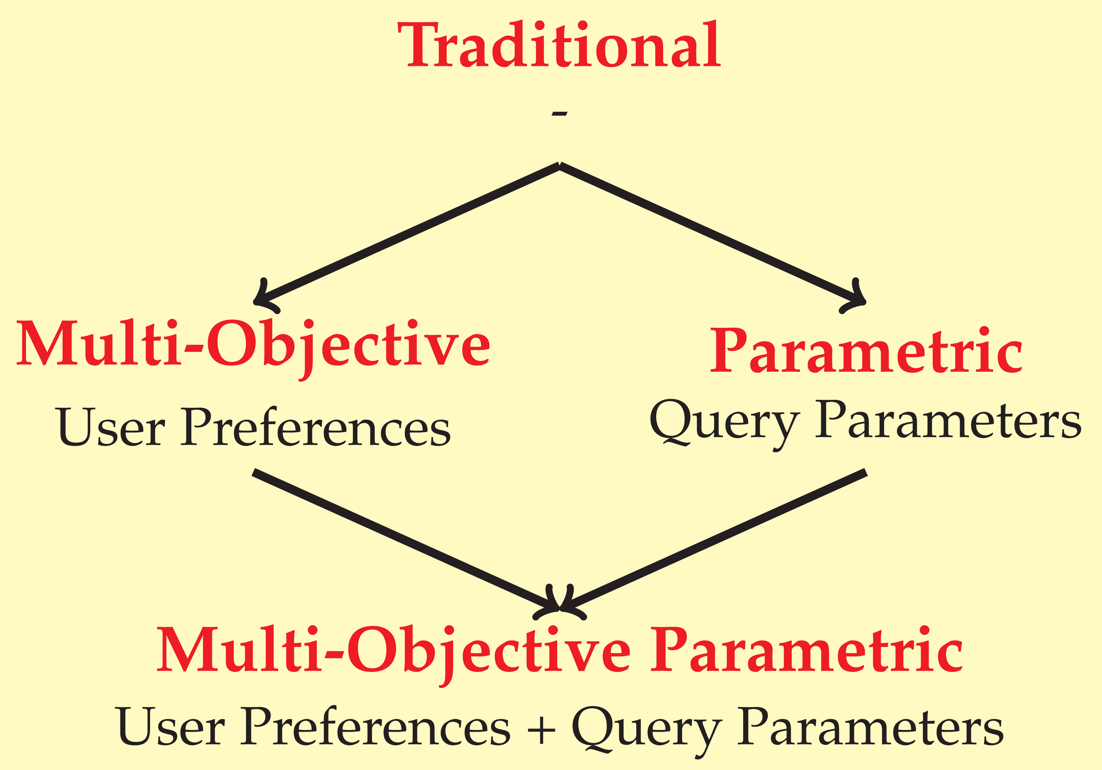
# Multi-Objective Parametric Query Optimization

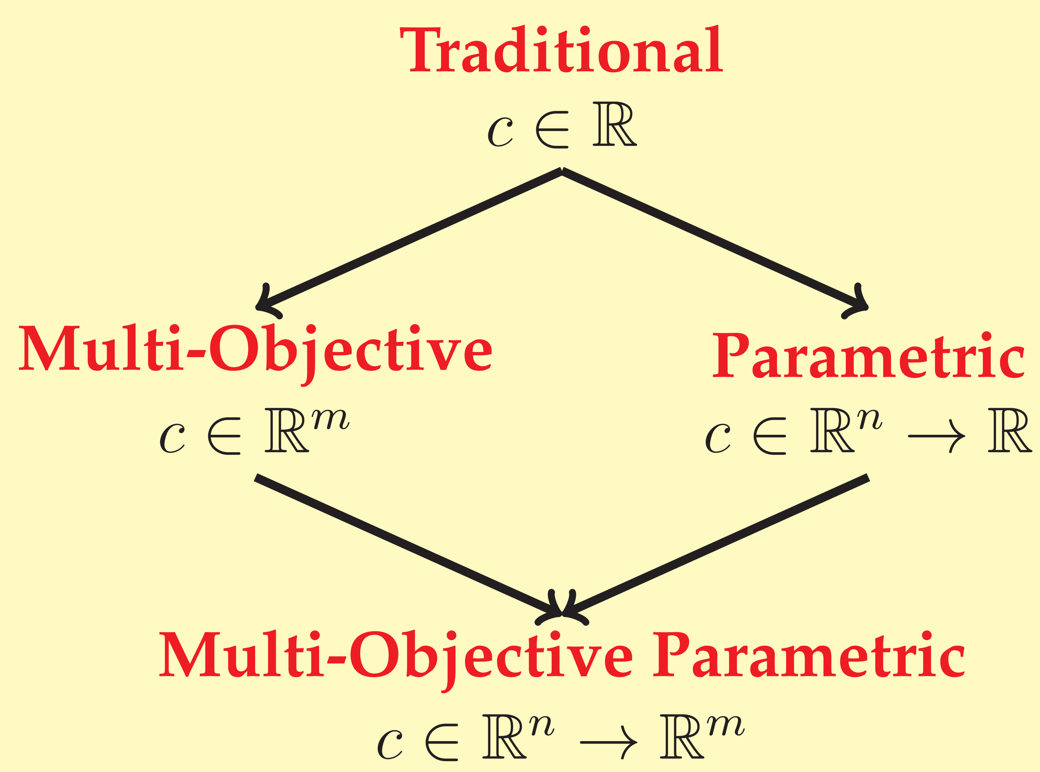
Immanuel Trummer and Christoph Koch
{immanuel.trummer, christoph.koch}@epfl.ch

## Problem Statement

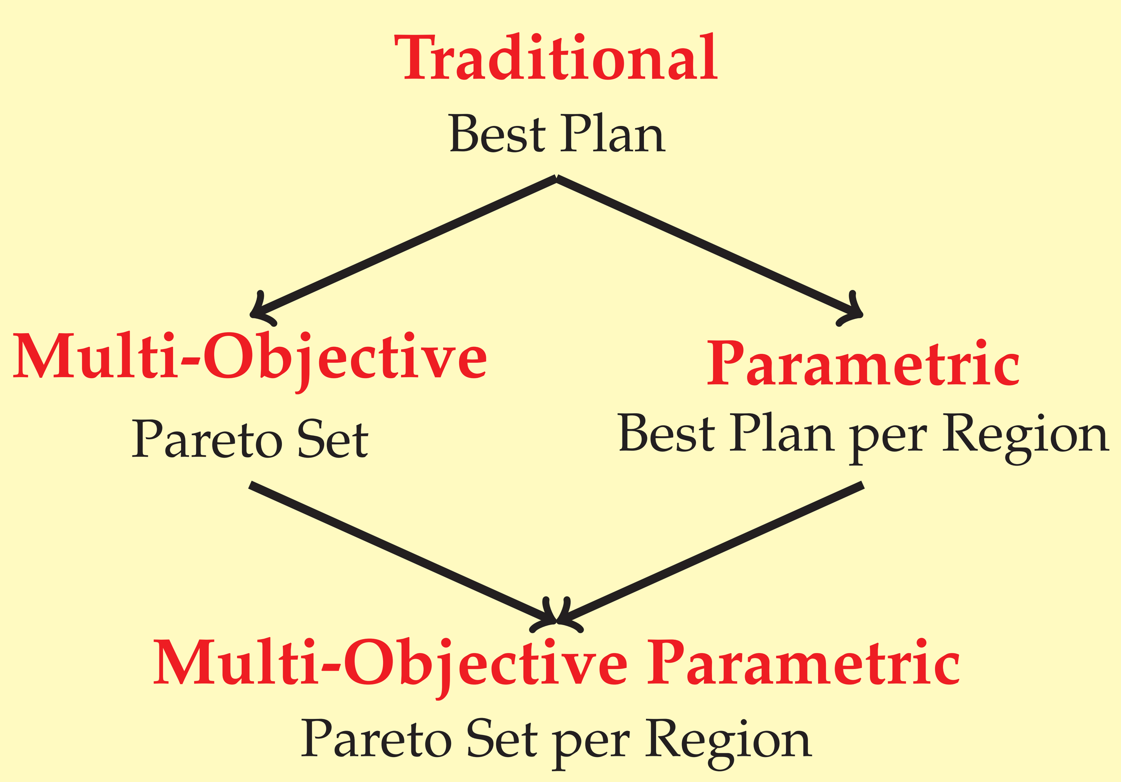
In traditional query optimization, the optimizer has all required information to compare alternative query plans. In multi-objective query optimization, we lack information on user preferences required to prioritize between plan cost metrics. In parametric query optimization, we lack information about parameters that influence the cost of alternative plans. Multi-objective parametric query optimization allows to model both kinds of uncertainty:




Traditional query optimization assigns each query plan to a scalar cost value. Multi-objective query optimization assigns each plan to a cost vector, representing cost according to different metrics. Parametric query optimization models the cost of a plan as function from a multi-dimensional parameter space to a one-dimensional cost space. Multi-objective parametric query optimization generalizes all of those variants and models plan cost as function from a multi-dimensional parameter space to a multi-dimensional cost space:

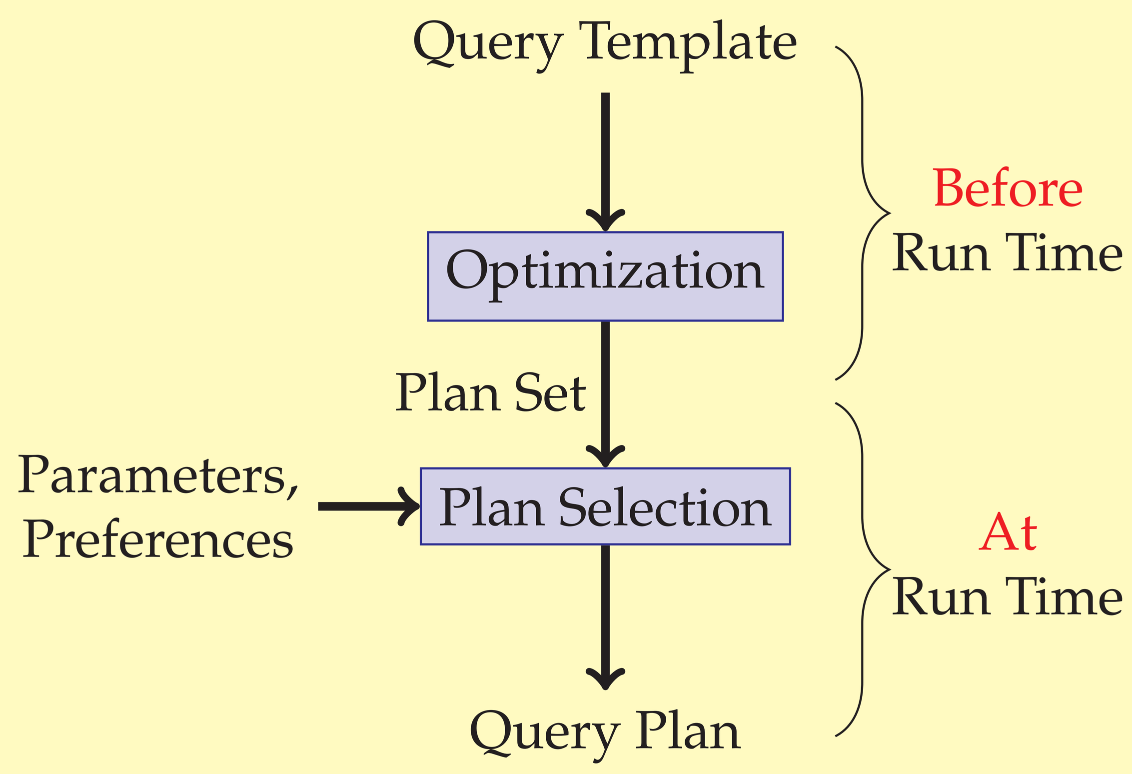



The goal in traditional query optimization is to find the best query plan. In multi-objective query optimization, we want to a find a set of Pareto-optimal plans. We want to find the optimal plan for each parameter space region in parametric query optimization. In multi-objective parametric query optimization, the goal is to find the set of Pareto-optimal plans for each parameter space region:




## Context

Multi-objective parametric query optimization happens before run time: based on a parameterized query template, the set of query plans that could be relevant for arbitrary parameter values and user preferences is calculated. At run time, once parameter values and user preferences become known, the best plan is selected out of that set. This avoids query optimization at run time:




## Problem Properties

Considering multiple cost metrics instead of only one changes the properties of the parametric query optimization problem fundamentally. We show that the “guiding principle of parametric query optimization” on which many algorithms have been based does not generalize to multi-objective parametric query optimization.

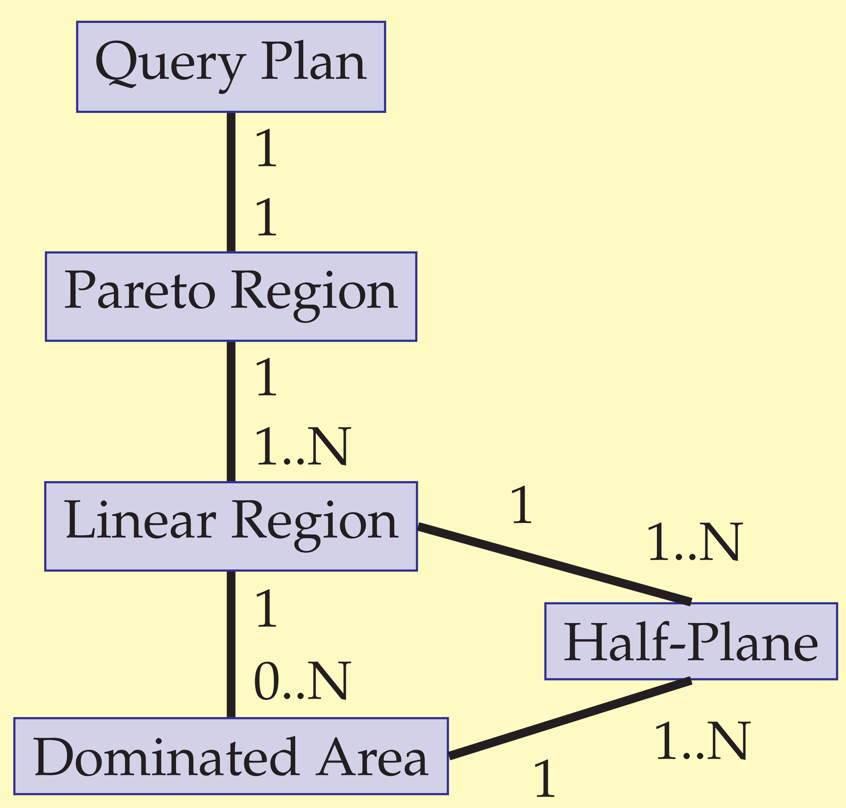
## Algorithm Overview

- Based on dynamic programming
- Each query plan maps to a Pareto region
- Regions are reduced during pruning
- Plans with empty region are discarded

## Region Representation

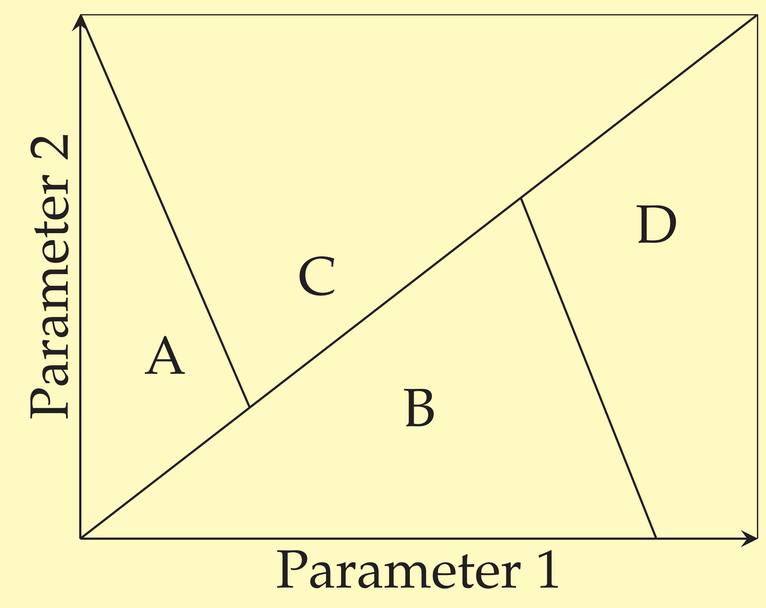
Each query plan is associated with a parameter space region for which it is Pareto-optimal. Assuming that plan cost functions are piecewise linear, we prove that all Pareto regions can be represented as follows.

Each Pareto region is the union of several linear regions. Each linear region is a convex polytope from which other convex polytopes, representing sub-regions in which other plans dominate, have been cut out. Each convex polytope is an intersection of half-planes.

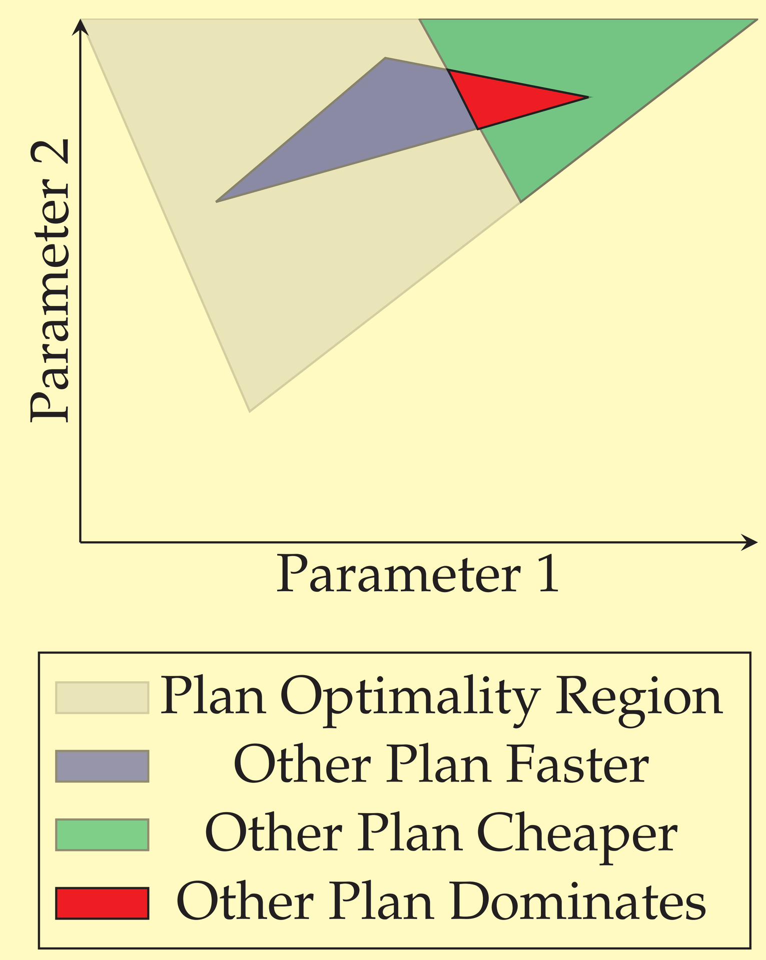



## Pruning

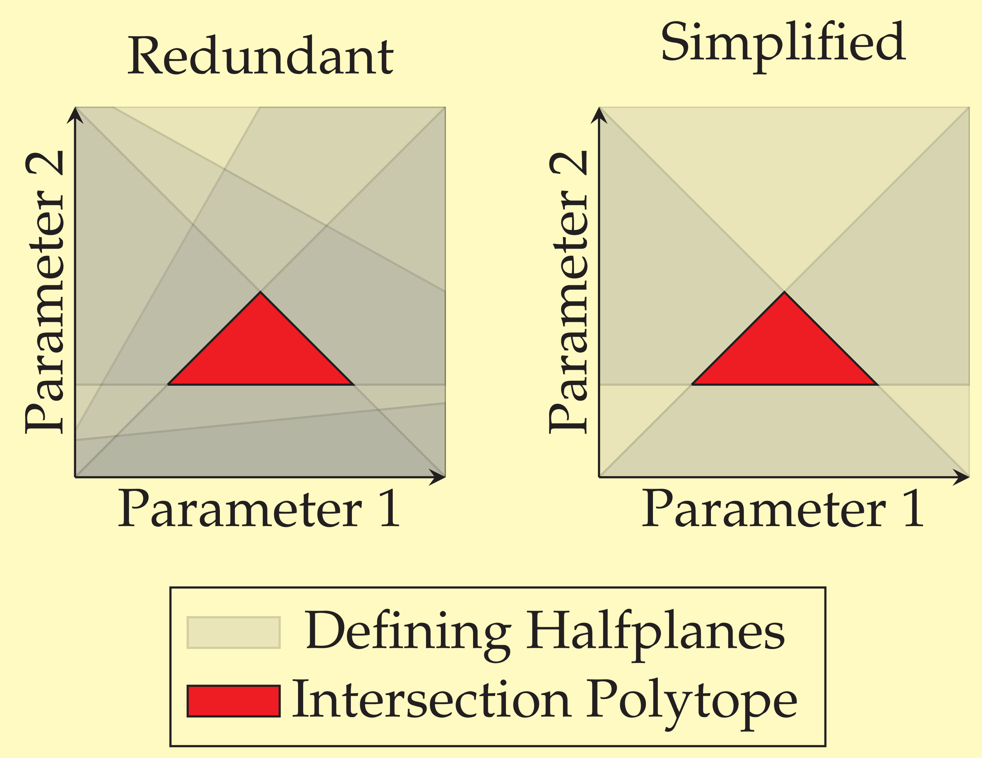
**1.** We divide the parameter space into linear regions (here: A to D) in which all plans have linear cost functions:




**2.** We compare plains pair-wise and remove parameter space regions in which a plan is dominated from its Pareto-optimality region:




**3.** Consecutive pruning steps often lead to redundant representations of parameter space regions. We regularly try to simplify region representations to increase efficiency:




## Experimental Results (Extract)

We measured the optimization time required by a Java-based implementation of our algorithm for optimizing randomly generated queries. We consider two plan cost metrics in a Cloud scenario and assume linear plan cost functions: