

A Spreadsheet-like User Interface for Combinatorial Multi-Objective Optimization



Derek Rayside
MIT CSAIL
[Waterloo 2010]

H.-Christian Eslter
University of Paderborn
[MIT 2008]



Concorde SST



Airbus A380



Boeing 787



Speed is King



Multiple Objectives

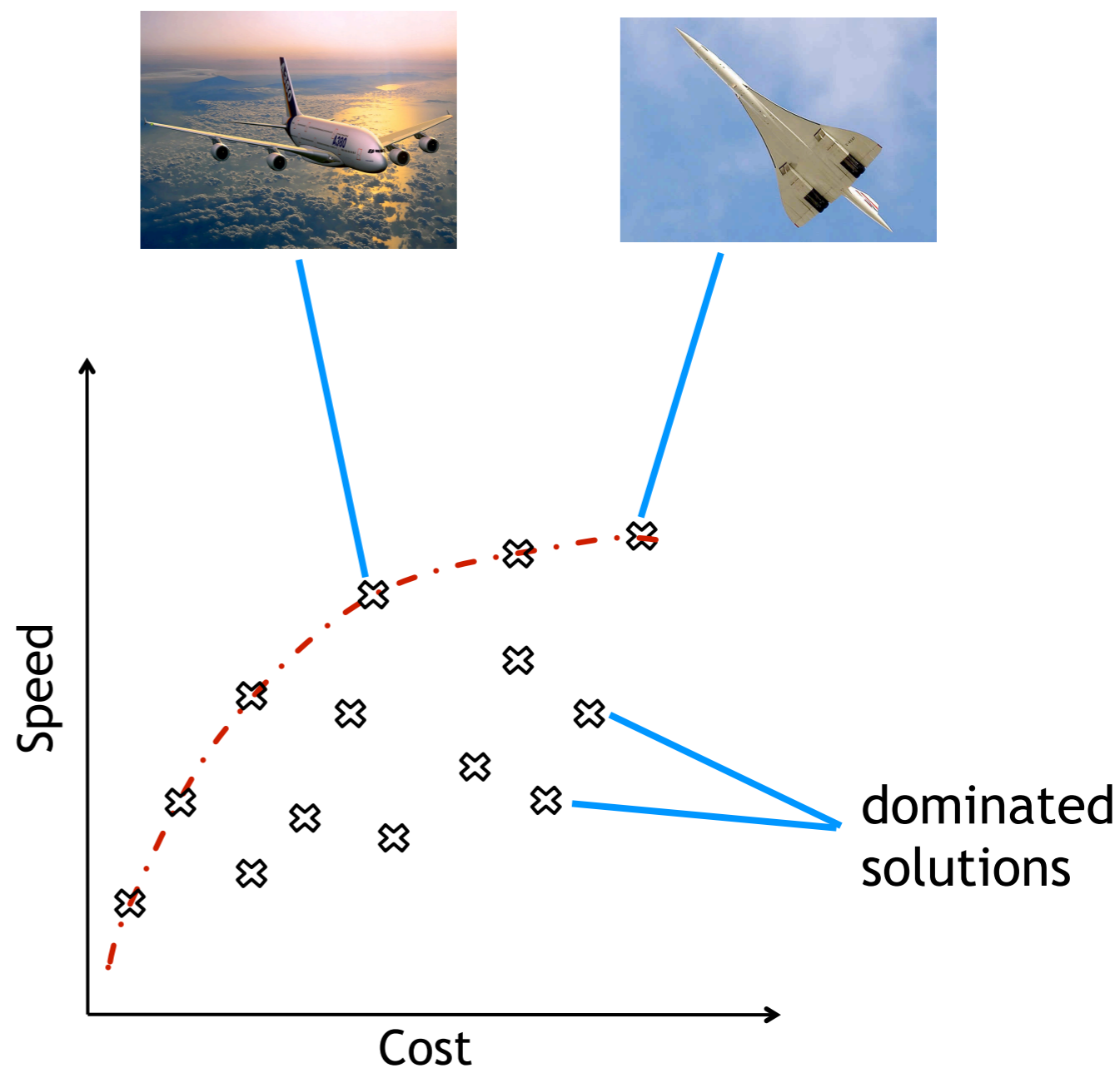


Choose Two



– Arthur C. Clarke

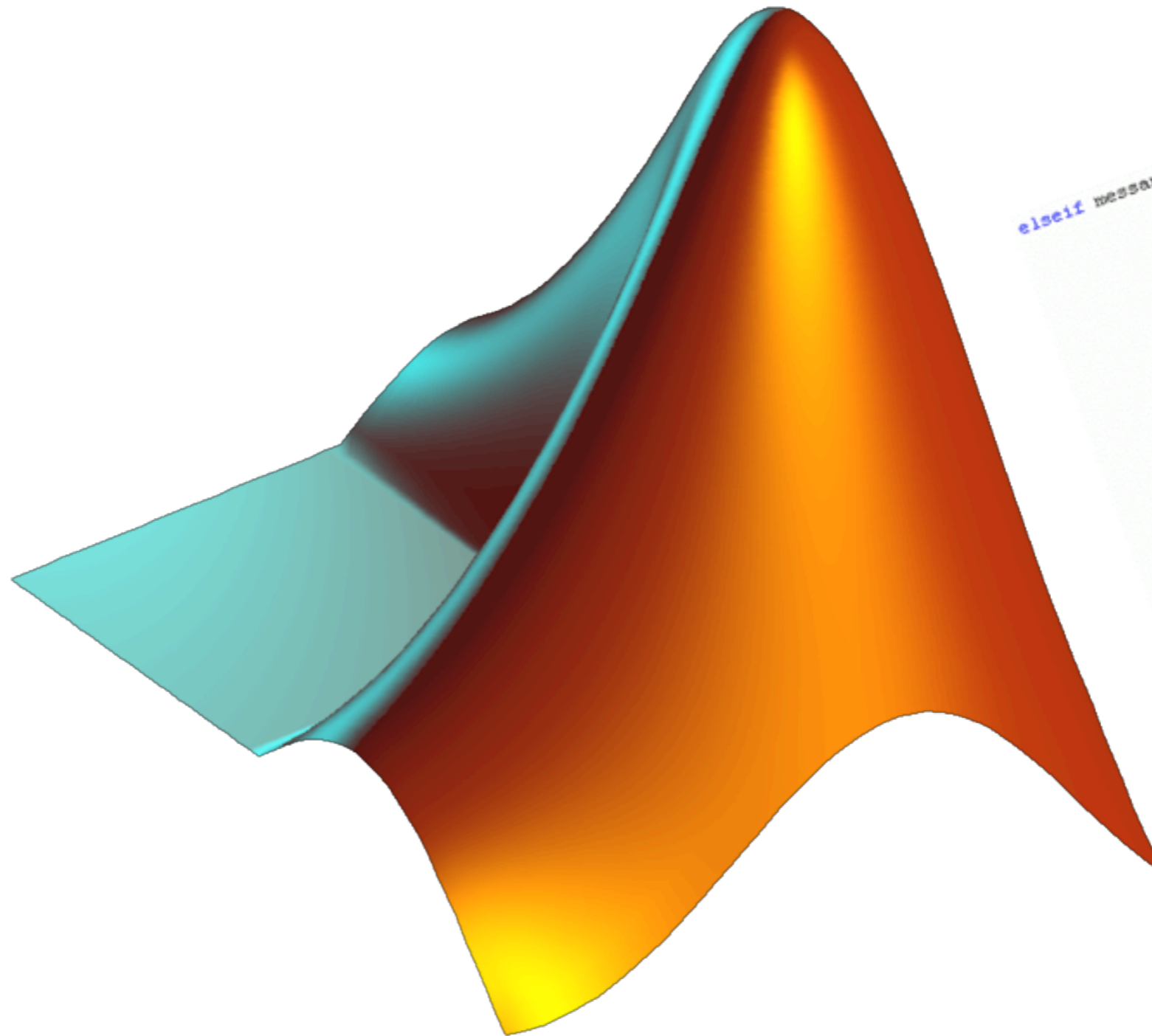
Pareto Front



How can I get started?

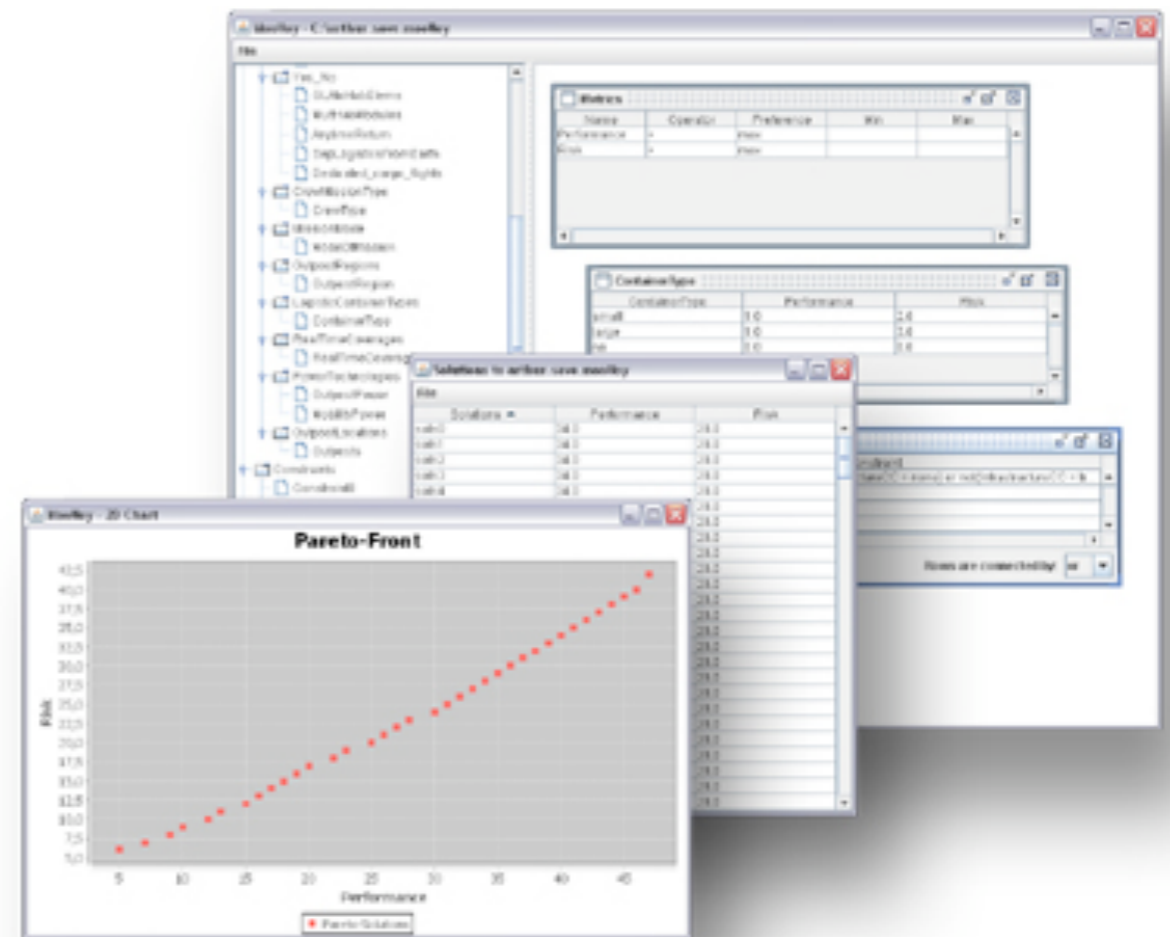


How can I get started?



```
elseif messart(ilag)==2
    [indz,inds] = find(C1(1:izanf+mikoord-1,
    C1(indz,NFG+1:2*NFG) = C1(indz,1:NFG);
    C1(indz,1:NFG) = zeros(length(indz),NFG)
    if Nas~=0
        switch language
            case 'D'
                fprintf('\n\n FEHLER: Geschw
            case 'E'
                fprintf('\n\n ERROR: Vel. c
            end
        end
    end
    return
end
end
izanf = izanf+mikoord;
end
% Recalculate controller matrix B in
if ANZKOP > 0
    B1 = B1*TC2;
end
% Transform B1*C1 to indices for
Tilting/Translation Control
%B1NFG = B1(1:2*NFG,:); % B1 for
%C1NFG = C1(:,1:2*NFG); % C1 for
%if Nas ~= 0
    B1Nas = B1(2*NFG+1:2*NFG+
    % C1Nas = C1(:,2*NFG+1:2*NFG+
    %end
```

Moolloy



Multi-Objective Optimization with Alloy

Outline



- Introduction
- Example: bicycle configuration
- Definition of a Multi-Objective Problem
- The Moolloy User Interface
- User Case Studies
- Conclusions & Future Work

Bicycle Configuration



Metrics: Speed vs Cost



Frame: aluminum or steel?



Forks: regular or shocks?



Handlebars: racing, etc, ?

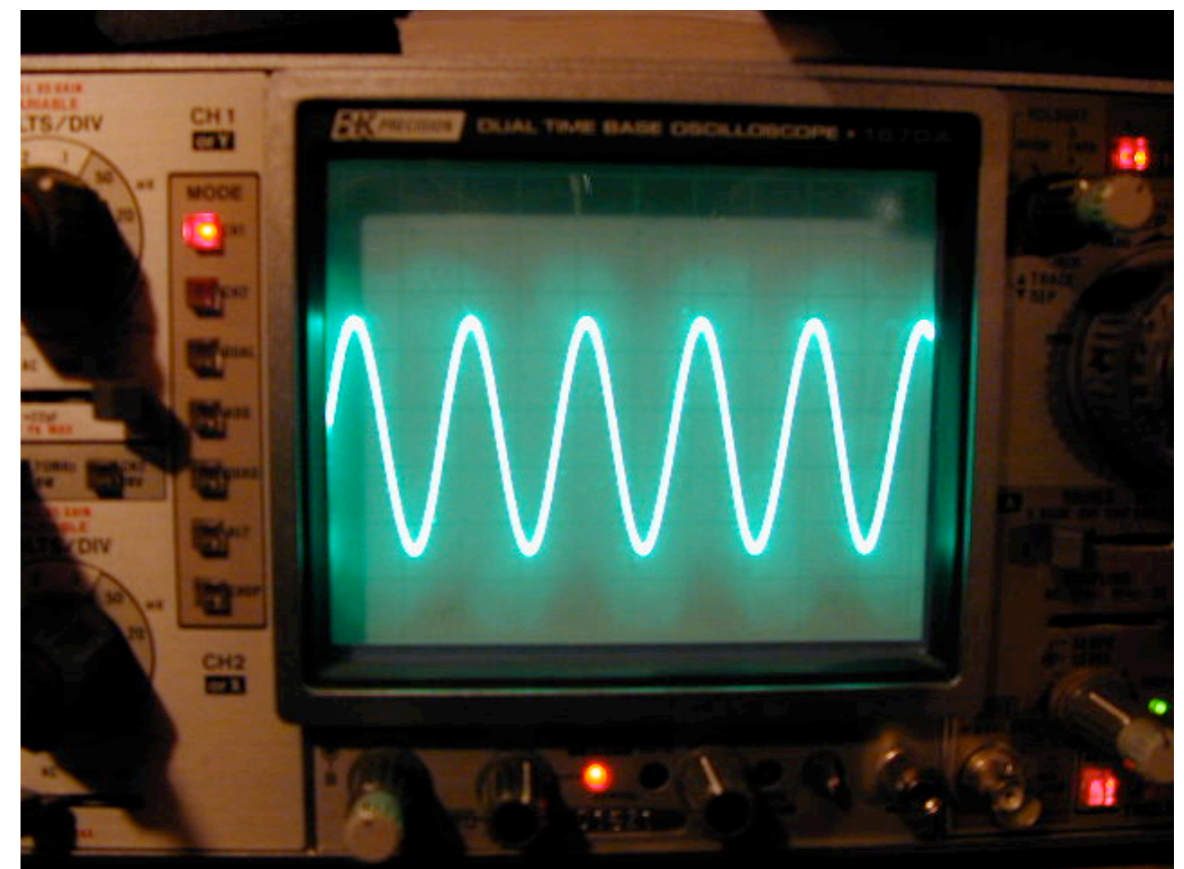
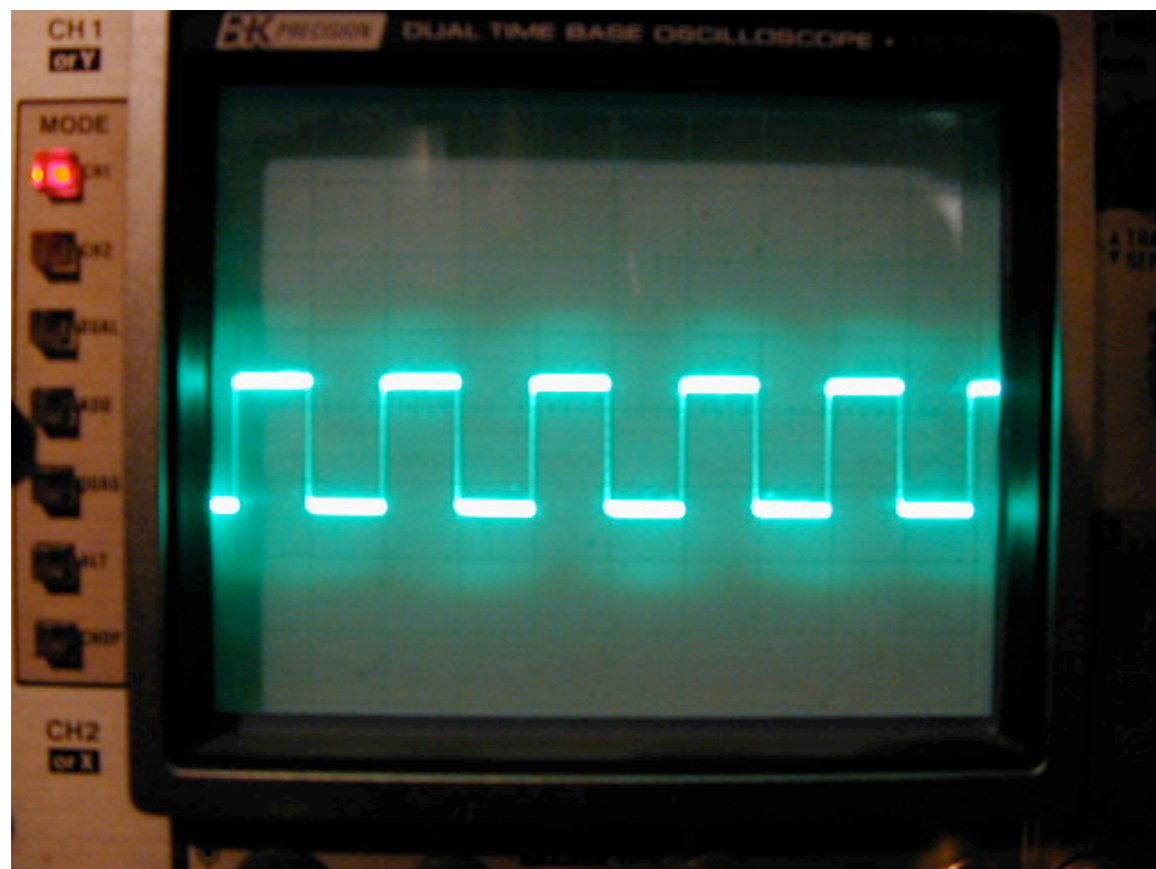


Definition of a MOOP

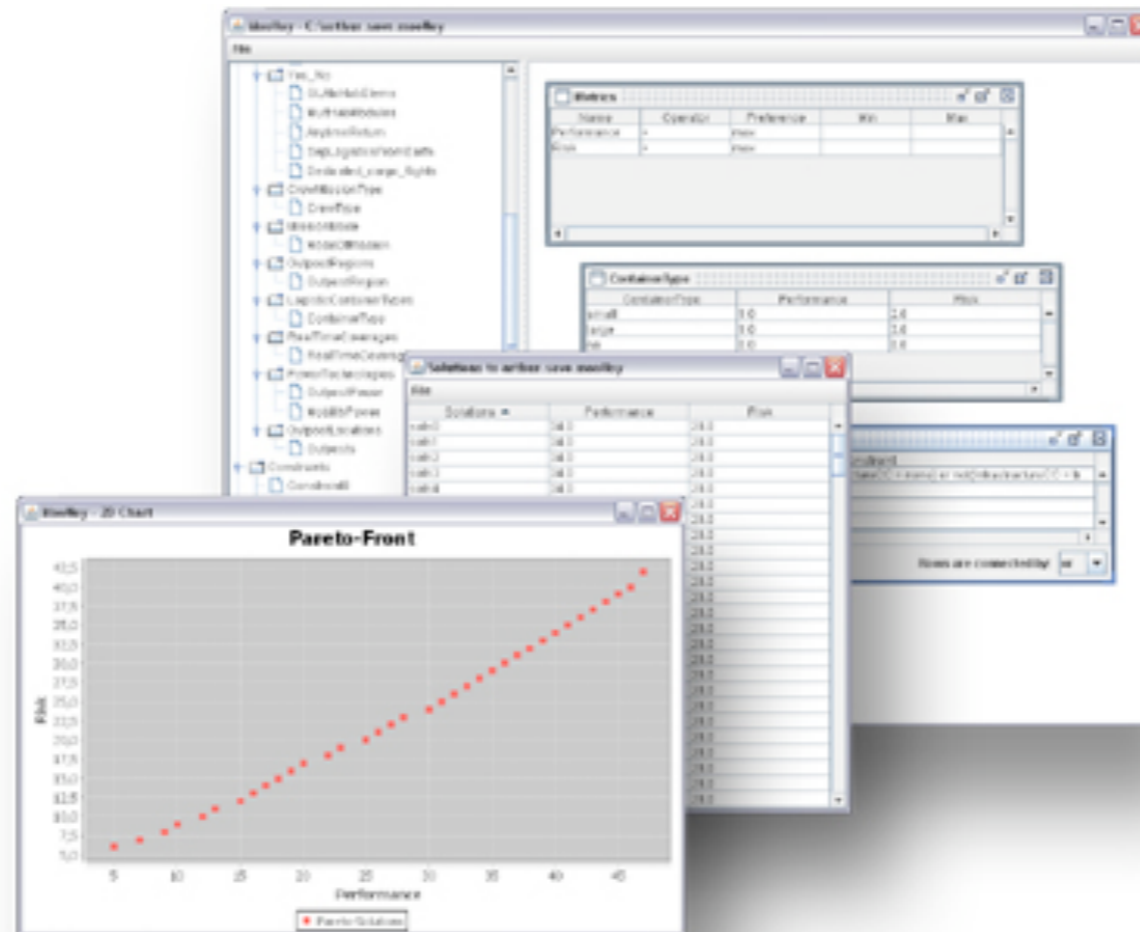


Elements	Examples
<i>Decision variables:</i> x_1, \dots, x_z	Frame; Fork; Front mech; Rear mech;
<i>Domains:</i> d_1, \dots, d_y	{Steel, Aluminum}; {RockShock_XL, LST_AIR}; {Srum9, Srum10, Shimanu_XR};
<i>Constraints:</i> c_1, \dots, c_p	if frame material is aluminum then fork model must be RockShock_XL
<i>Metrics:</i> m_1, \dots, m_q	Cost; Performance;

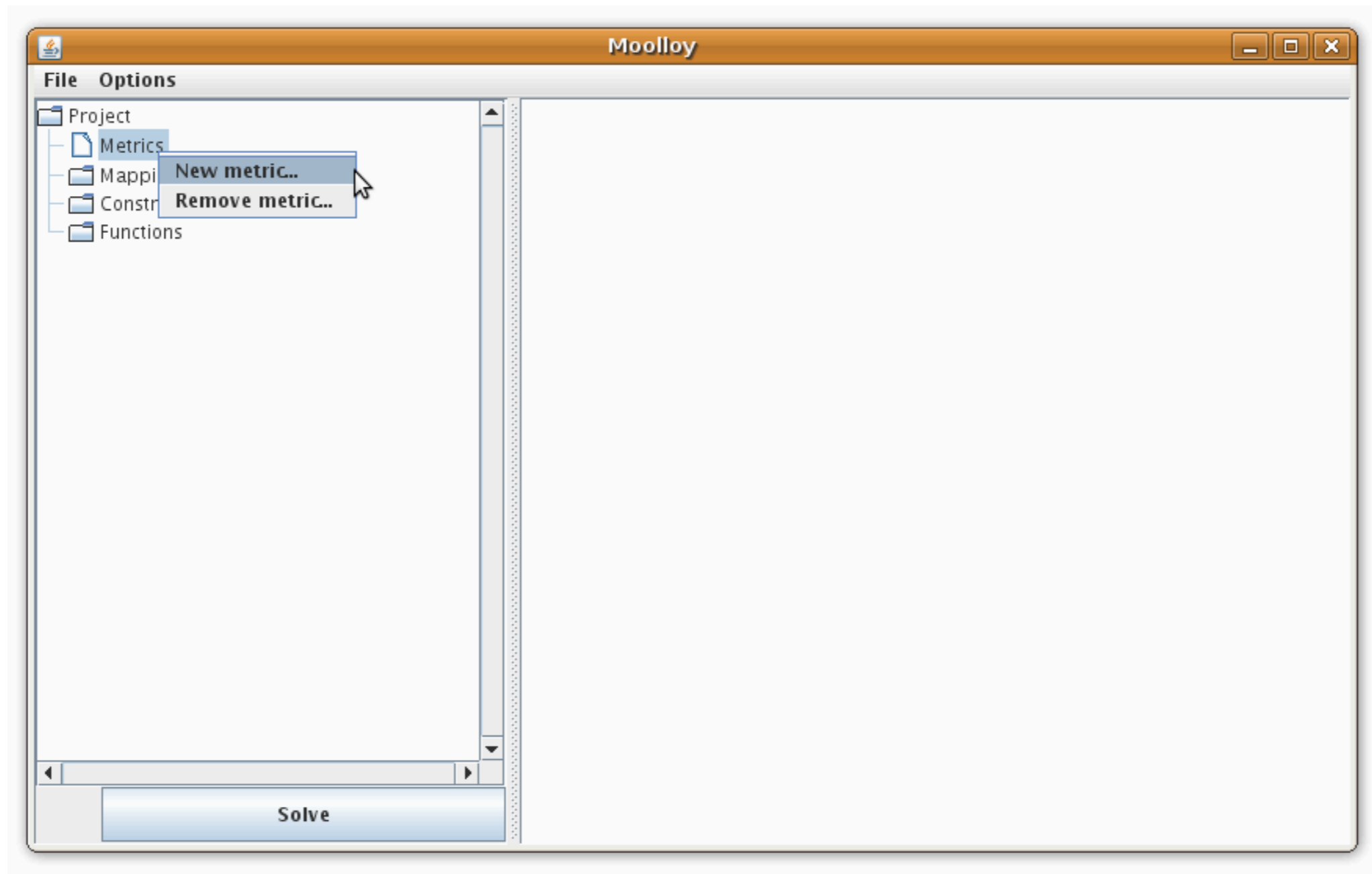
Discrete Domains



Moolloy User Interface



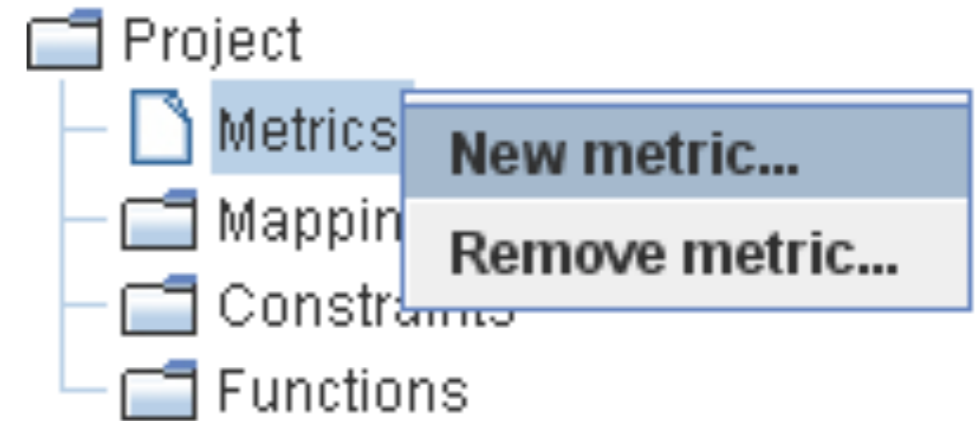
Project Tree



Declaring Metrics



- Start with declaring *Metrics*



- Choose *Operator* : *, +

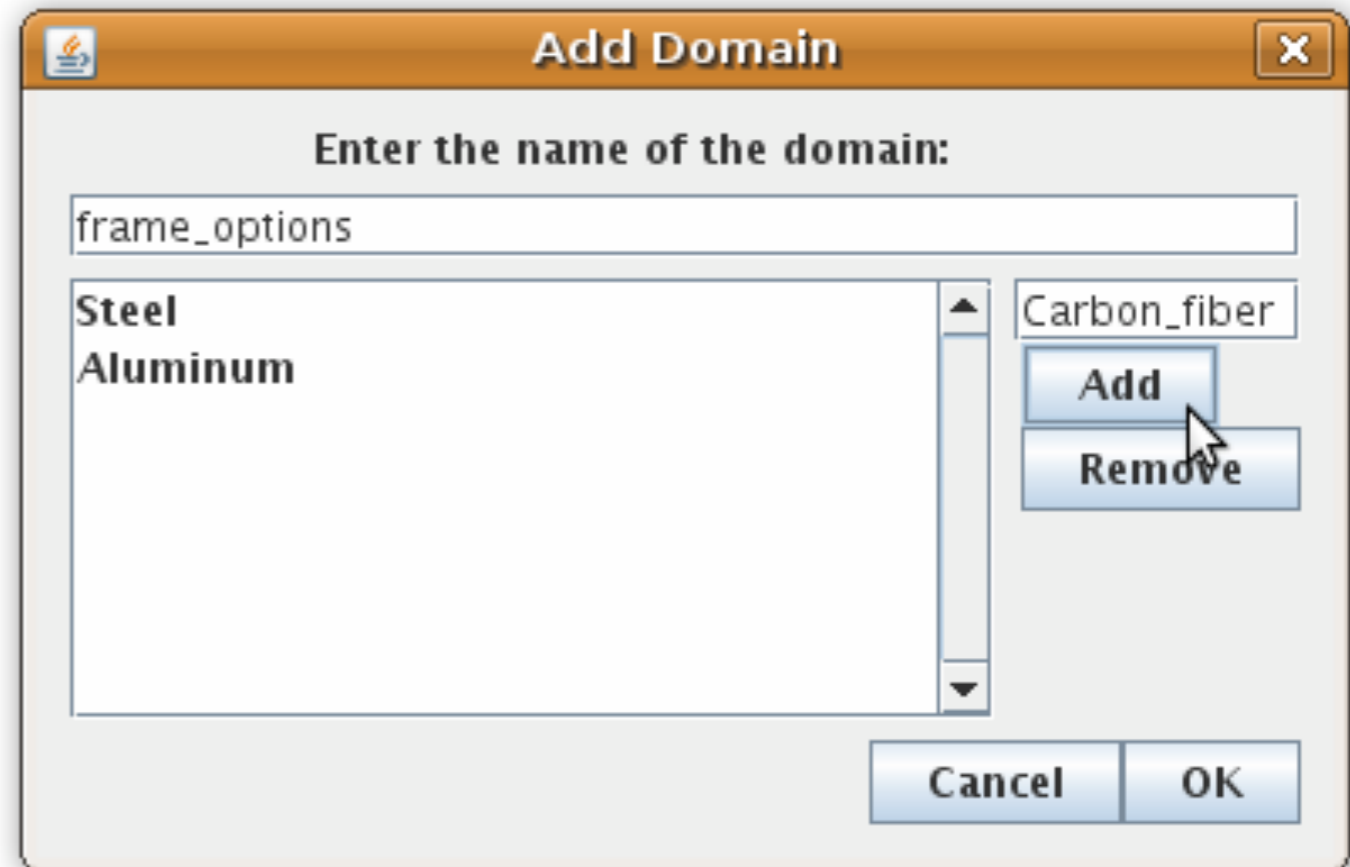
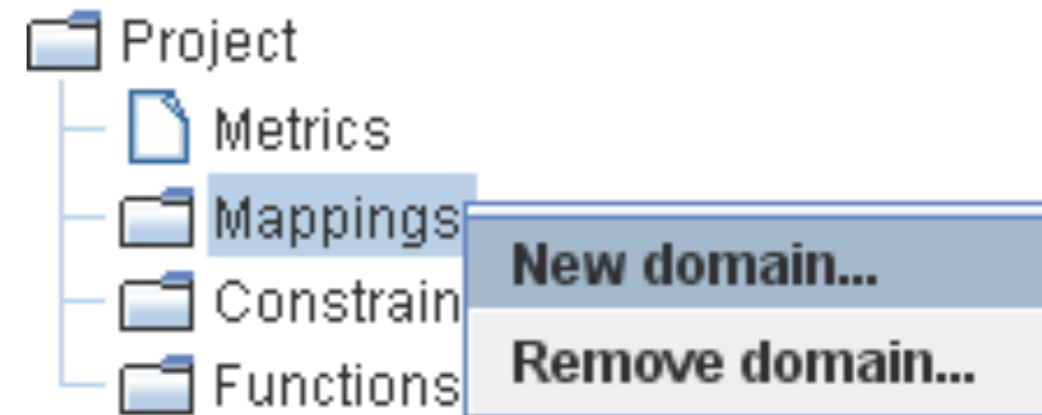
- Choose *Preference*: min, max, none

Name	Operator	Preference	Min	Max
Performance	+	max		
Cost	+	min		

Declaring Domains



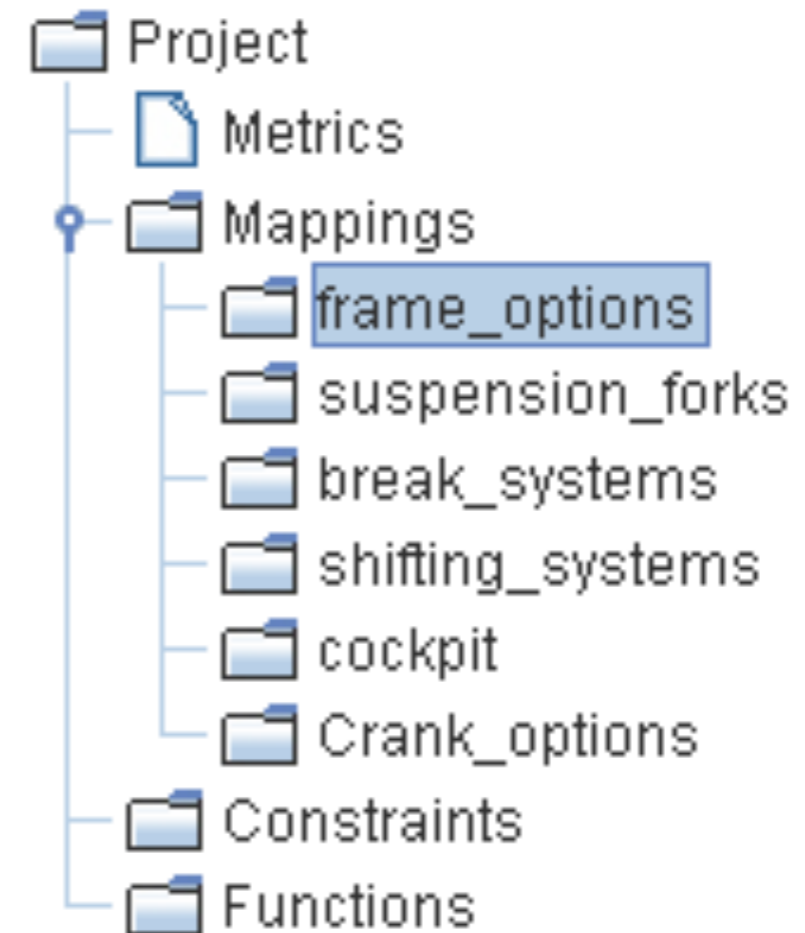
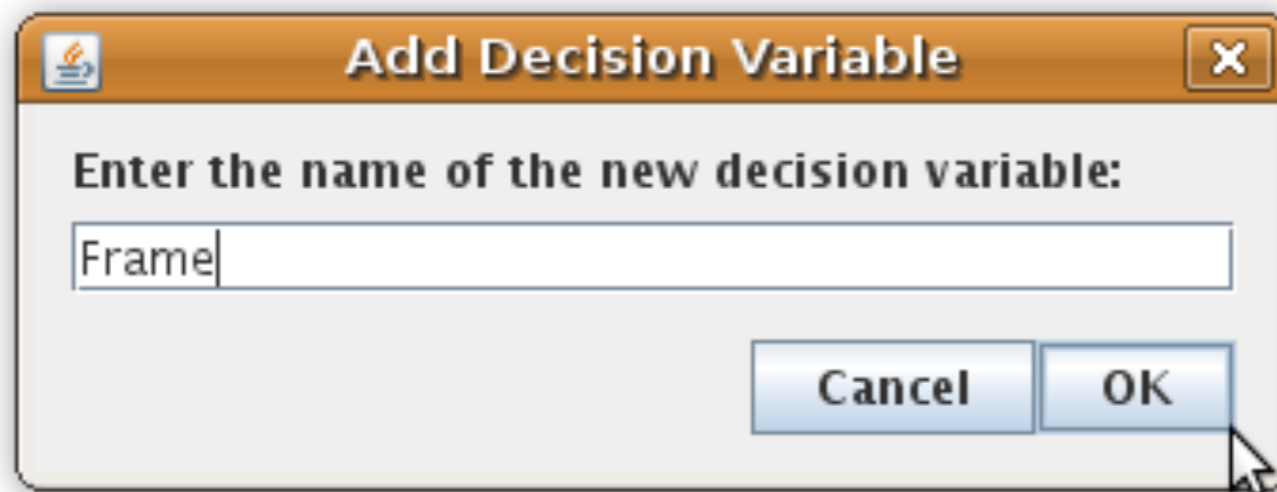
- Add domains and their values
- Examples:
 - *frame_options*:
{Steel,
Aluminum,
Carbon_fiber}
 - *forks*:
{regular,
shocks}



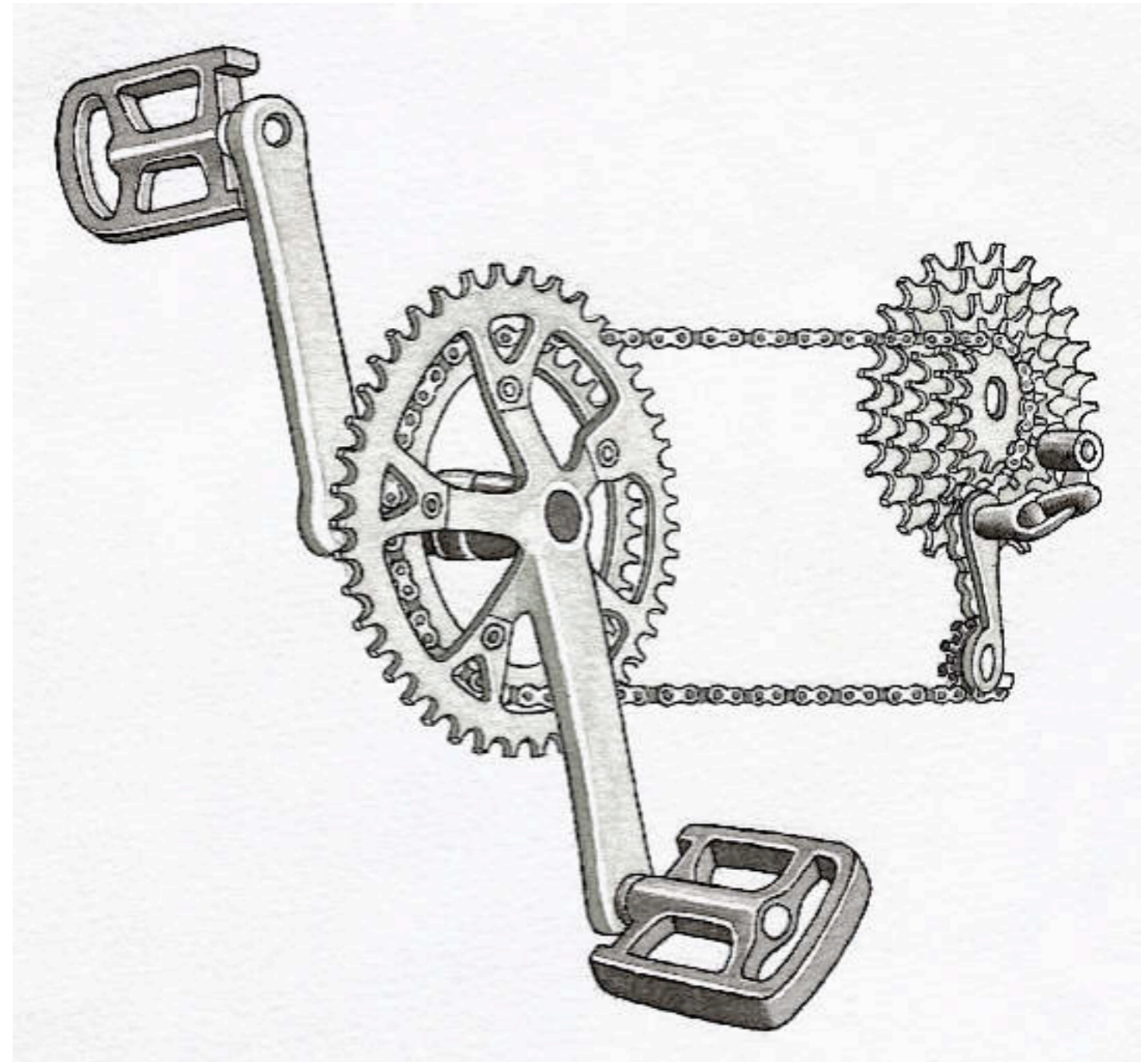
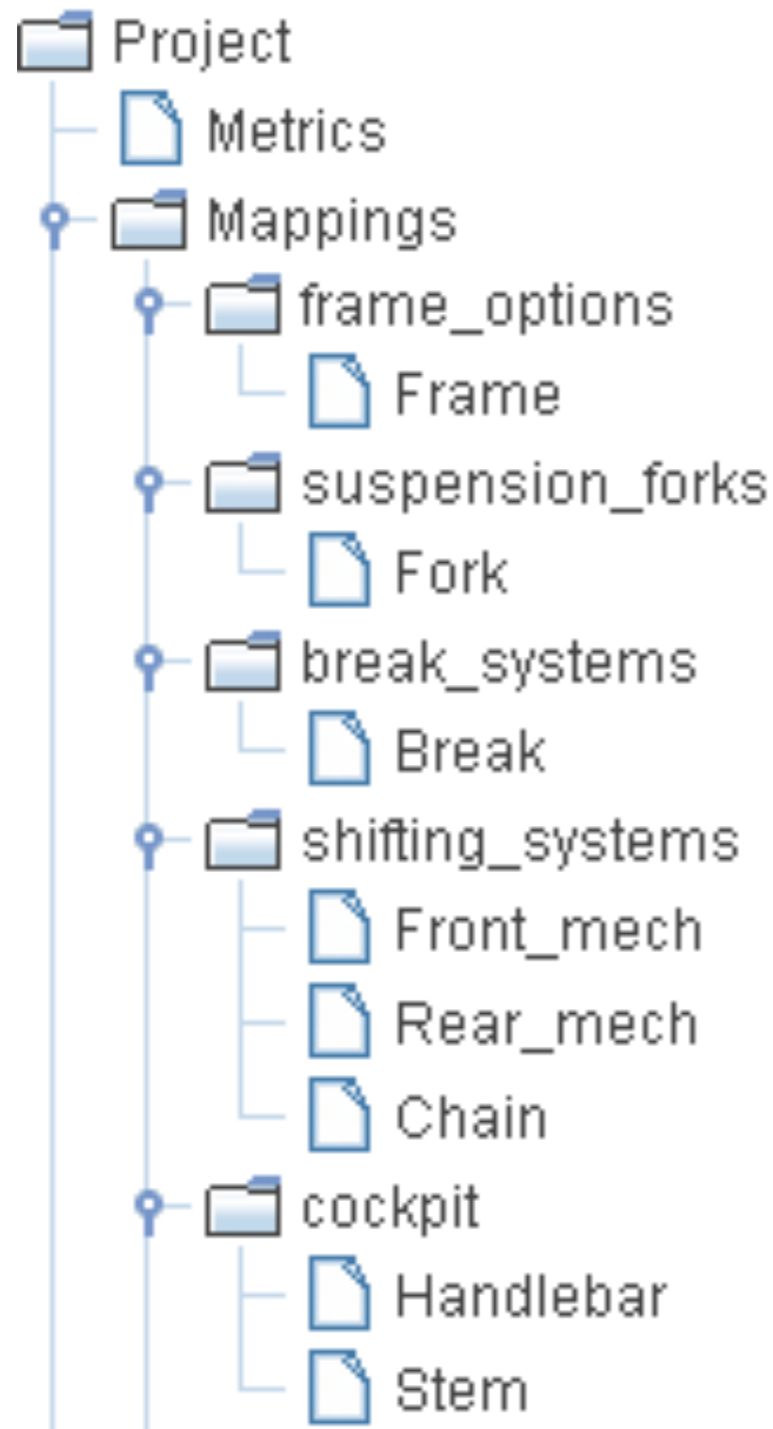
Decision Variables



- Add variables to the domains



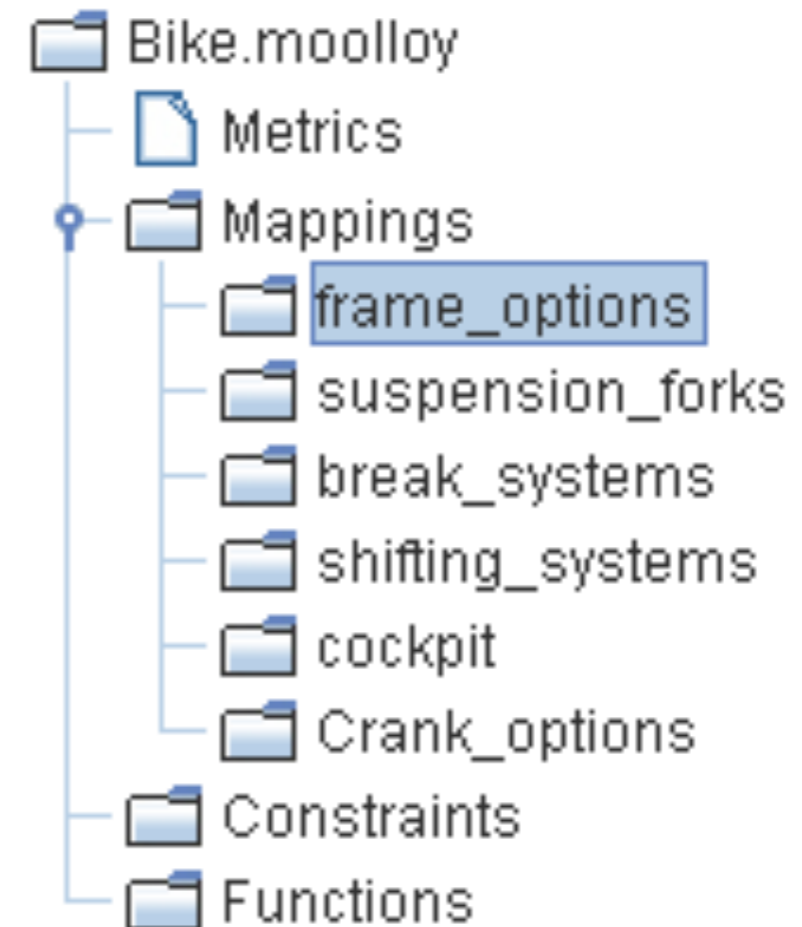
3 Decisions in 1 Domain



Metric Values



- Add variables to the domains
- Enter metric values for assignments



Frame	Performance	Cost
Steel	5.0	800.0
Aluminum	8.0	1100.0
Carbon_fiber		

Solve!



List of Pareto Solutions



- Each solution in list can be inspected
- Can plot 2D Pareto front

Solutions ▲	Cost	Performance
soln0	1835.0	49.0
soln1	1585.0	43.0
soln2	1480.0	43.0
soln3	1480.0	59.0
soln4	1910.0	62.0
soln5	2120.0	41.0
soln6	1455.0	65.0
soln7	2420.0	50.0
soln8	1610.0	50.0
soln9	1610.0	53.0
soln10	1685.0	67.0
soln11	2795.0	46.0
soln12	1525.0	46.0
soln13	1525.0	46.0
soln14	1525.0	46.0

Inspecting a single solution



Solution "soln35" - Moolloy

By value **By decision variable**

Front_mech -> Shimanu_XYZ	Chain -> Shimanu_XYZ	Break -> Magoron_HT44
Handlebar -> Synface_Race	Rear_mech -> Srum_10	Stem -> Rich_Pro
Crank -> Tunee_BigFeet	Fork -> LST_AIR	Frame -> Aluminum

Inspecting a single solution



Solution "soln35" - Moolloy

By value | By decision variable

suspension_forks

RockShock_XL	Mountain_July	LST_AIR Fork	none
--------------	---------------	-----------------	------

frame_options

Steel	Aluminum Frame	Carbon_fiber
-------	-------------------	--------------

cockpit

Synface_Race Handlebar	Synface_Mater	Rich_Pro Stem	Rich_M6	Rich_M5
---------------------------	---------------	------------------	---------	---------

break_systems

Shimano_XYZ	Shimano_XR	Shimano_XR_Dis	Magoron_Betty_I	Magoron_HT44 Break
-------------	------------	----------------	-----------------	-----------------------

Crank_options

Shimano_XR	Shimano_Deo	Tunee_BigFeet Crank
------------	-------------	------------------------

shifting_systems

Shimano_XYZ Front_mech Chain	Shimano_XR	Shimano_Dec	Srum_10 Rear_mech	Srum_9
------------------------------------	------------	-------------	----------------------	--------

UI Overview



The screenshot displays the Moolloy user interface for a bicycle model. The main window is titled "Moolloy - /home/hce/workspace/moolloy-ui-cason09/models/Bike.moolloy". On the left is a file explorer showing a tree structure of components like "Metrics", "Mappings", "frame_options", "suspension_forks", "break_systems", "shifting_systems", "cockpit", "Crank_options", "Constraints", and "Functions".

Four floating windows provide detailed data:

- Metrics:** A table with columns Name, Operator, Preference, Min, and Max. It lists "Perform..." with a "+" operator and "max" preference, and "Cost" with a "+" operator and "min" preference.
- Frame:** A table with columns Frame, Performance, and Cost. It lists "Steel" (5.0 performance, 800.0 cost), "Aluminum" (8.0 performance, 1100.0 cost), and "Carbon_fiber" (10.0 performance, 1900.0 cost).
- Fork:** A table with columns Fork, Performance, and Cost. It lists "RockShock_XL" (9.0 performance, 800.0 cost), "Mountain_July" (6.0 performance, 680.0 cost), "LST_AIR" (9.0 performance, 750.0 cost), and "none" (3.0 performance, 0.0 cost).
- Rear_mech:** A table with columns Rear_mech, Performance, and Cost. It lists "Shimanu_XYZ" (10.0 performance, 190.0 cost), "Shimanu_XR" (9.0 performance, 125.0 cost), "Shimanu_Deo" (7.0 performance, 75.0 cost), "Srum_10" (10.0 performance, 160.0 cost), and "Srum_9" (8.0 performance, 130.0 cost).

A "Solutions to Bike.moolloy" window shows a table of 15 solutions (soln0 to soln14) with columns for Solutions, Cost, and Performance. A mouse cursor is hovering over the Performance column for soln0.

A "Moolloy - 2D Chart" window displays a "Pareto-Front" chart. The y-axis is labeled "Performance" (ranging from 37.5 to 75.0) and the x-axis is labeled "Cost" (ranging from 1.500 to 4.000). Red square markers represent the Pareto-Solutions, showing a clear trade-off between cost and performance.

A "Solve" button is visible at the bottom of the main window.

Constraints



Constraints



- Three major types of constraints
 - i. Assignment constraints*
 - ii. Metric constraints*
 - iii. Functional constraints*
- *i) , ii)* are needed even for simple problems
- *iii)* more important for fine adjustments
 - (not yet supported by Moolloy UI)

Expressing Constraints



- Two ways of expressing constraints
 - 1) *Implicit*: using features built-in the UI
 - 2) *Explicit*: using a declarative expression language

Implicit Constraints



- Metric constraints

The 'Metrics' window displays a table with the following data:

Name	Operator	Preference	Min	Max
Performance	+	max	8	
Cost	+	min		1000

- Assignment constraints

The 'Frame' window displays a table with the following data:

Frame	Performance	Cost
Steel	5.0	800.0
Aluminum	8.0	1100.0
Carbon_fiber		

The 'frame_options' window displays a table with the following data:

frame_options	MinVars	MaxVars
Steel		
Aluminum		
Carbon_fiber		

Explicit Constraints



- Grammar for simple expression language:

```
constraint ::= '(' constraint ')'
            | '@' classId
            | negationOp constraint
            | constraint logicOp constraint
            | formula
            | logicConst

negationOp ::= 'not' | '!'

logicOp ::= 'and' | '&'
          | 'implies' | '=>'
          | 'iff' | '<=>'
          | 'or' | '|'

logicConst ::= 'true' | 'false'

formula ::= expr formulaOp expr
         | varId formulaOp (varId | valueId)
         | valueId formulaOp varId

formulaOp ::= '=' | '<' | '>' | '<=' | '>='

expr ::= '(' expr ')'
       | expr exprOp expr
       | sumExpr
       | multExpr
       | intConst

exprOp ::= '*' | '/' | '%' | '+' | '-'

sumExpr ::=
    '$sum(' metricId ',' domainId ','
           valueId ',' number ')

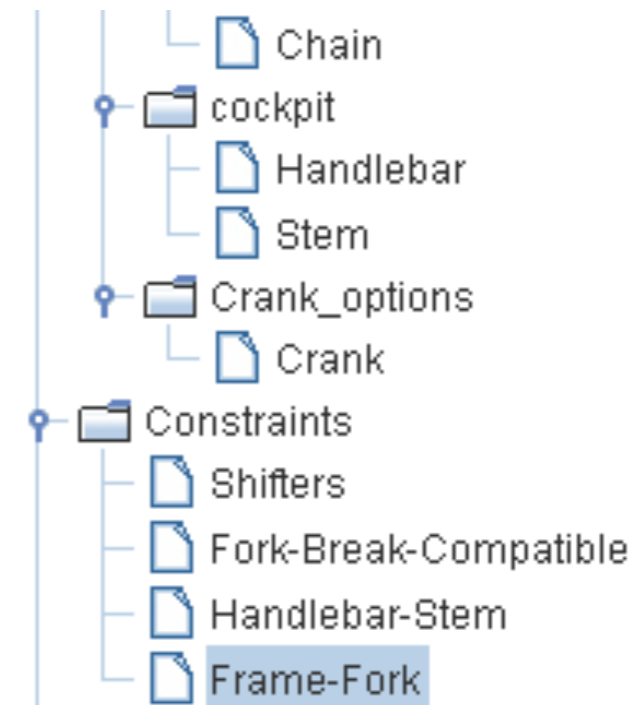
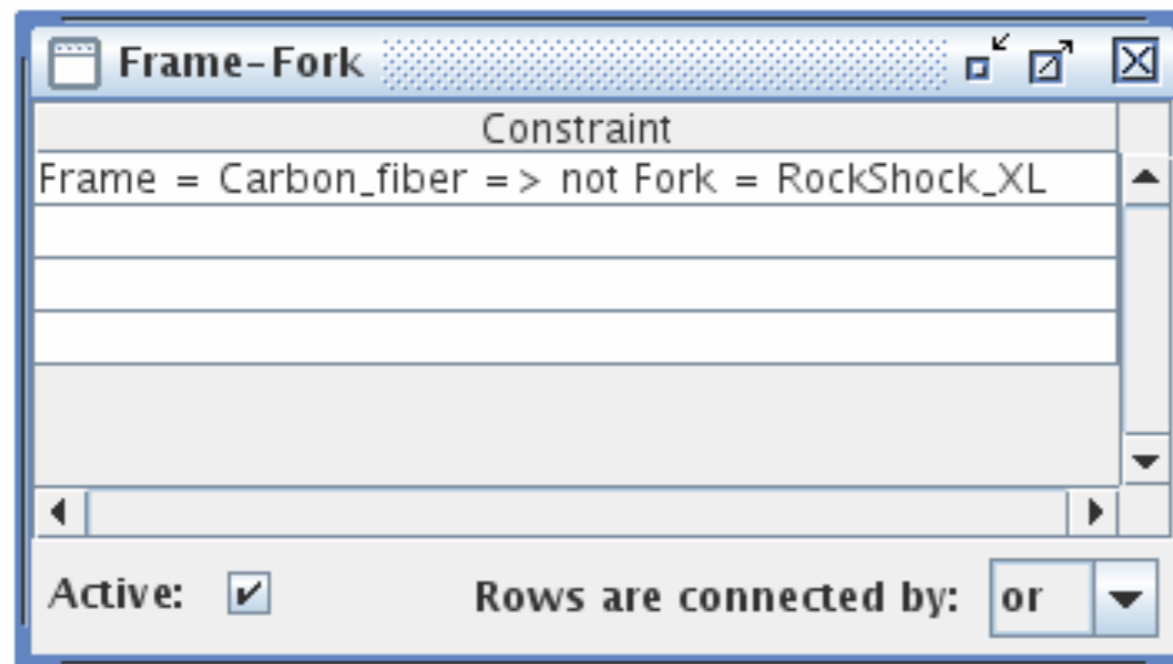
multExpr ::=
    '$mult(' metricId ',' domainId ','
            valueId ',' number ')

intConst ::= '$'number
```

An Explicit Constraint

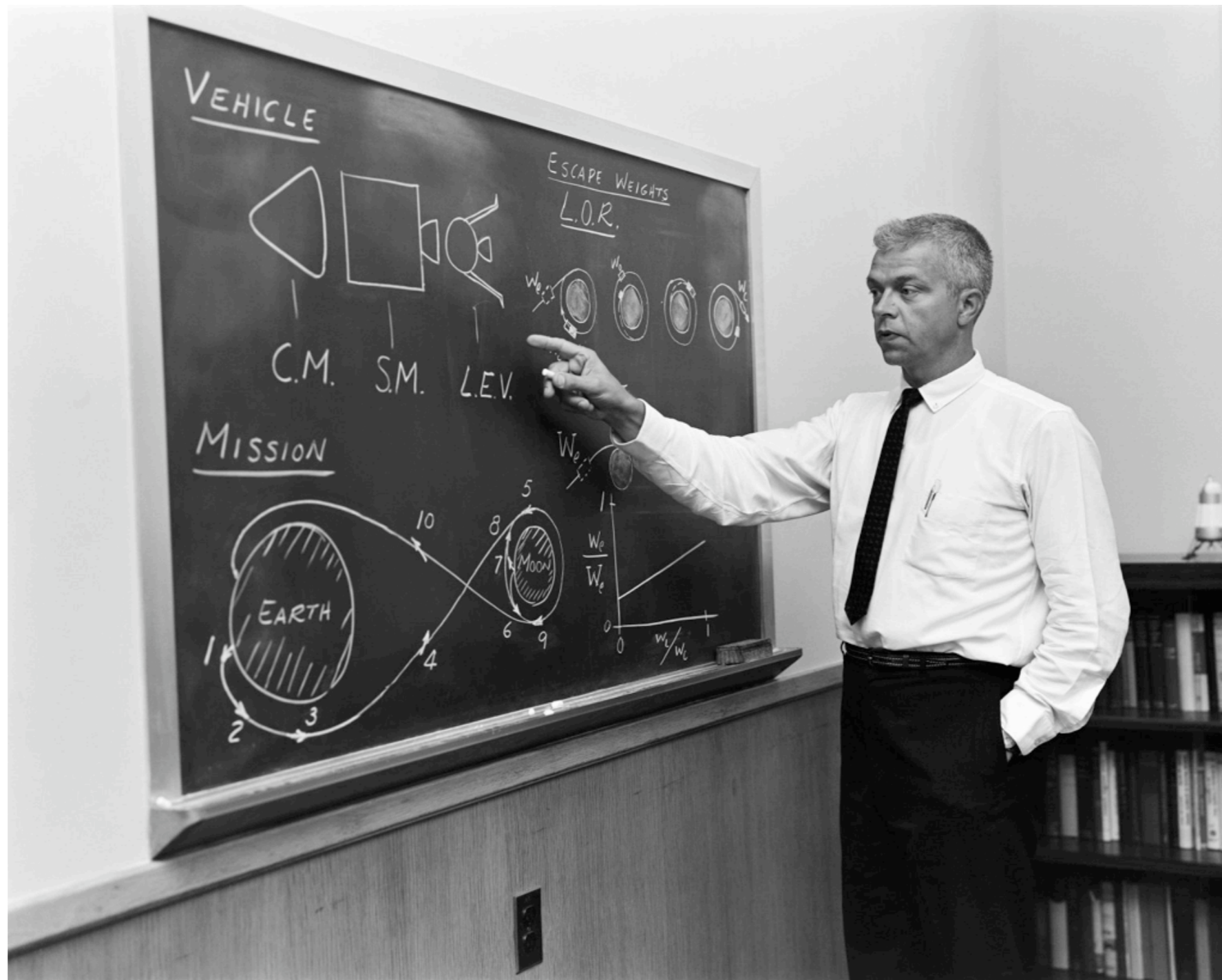



- **Frame=Carbon_fiber => not (Fork=Shocks)**



User Case Studies

Apollo Revisited



 John C. Houbolt
NASA Langley Research Center

7/10/1962

Image # EL-2000-00424

Production Planning



Decadel Launch Schedule



Decisions & Constraints



Decadel-Survey.moolloy

Metrics

Mappings

Year

ICESat-II

CLARREO

SMAP

XOVWM

PATH

3D-Winds

SCLP

GACM

GEO-CAPE

GRACE-II

SWOT

ASCENDS

Humidity

Constraints

Fairness_Water

Fairness_Earth

Fairness_Weather

Fairness_Climate

Fairness_Eco

SWOT_before_XOVWM

Budget

Fairness_Health

Metrics



Name	Operator	Preference	Min	Max
Health	+	max		
Water	+	max		
Eco	+	max		
Weather	+	max		
Earth	+	max		
Climate	+	max		
Cost	+	none		

Future Work

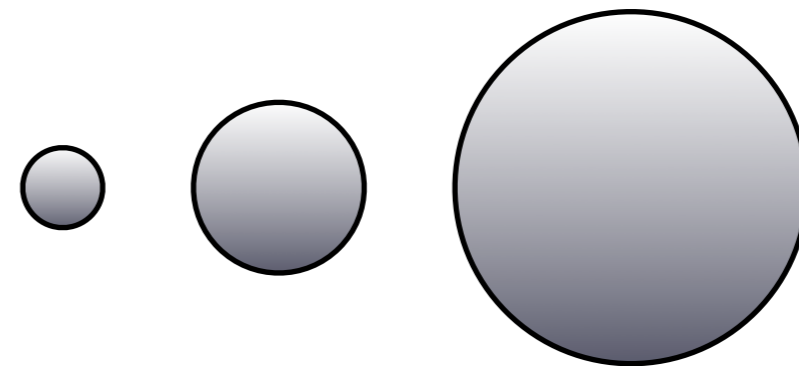


- Enhance UI with Schematic Tables
 - For expressing explicit constraints.
 - For inter-dependent metric functions.
 - [Edwards @ OOPSLA/Onward 2007]
- Visualizing Pareto-fronts in >2 dimensions
 - Discussion in workshop tomorrow afternoon
- Usability studies

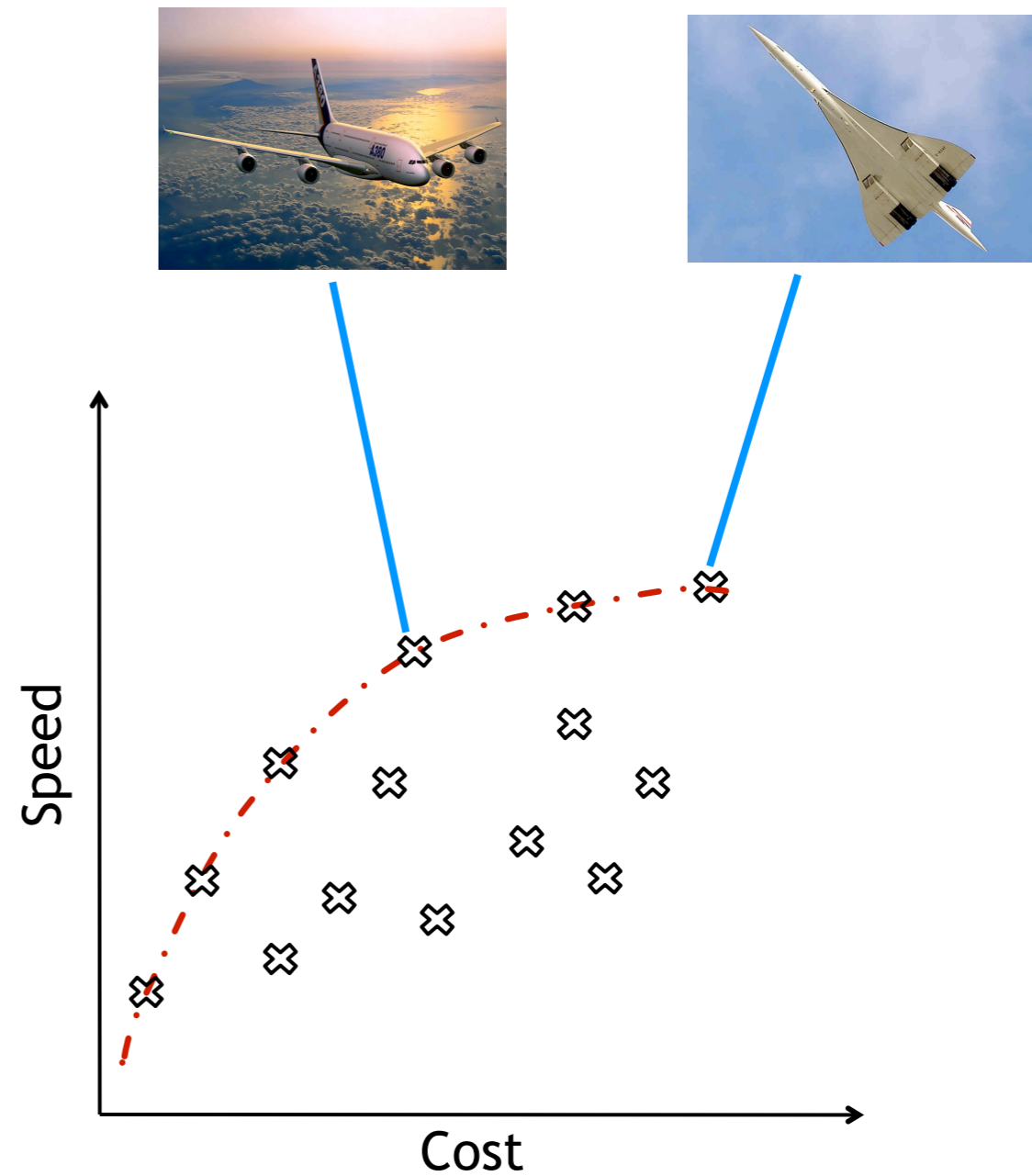
Conclusions



- Multi-objective optimization is not a hard idea.
 - Over 100 years old.
 - (But, designing algorithms & UIs is challenging.)
- It could be more widely used with better tools.
 - Potential users:



- <http://sdg.csail.mit.edu/moolloy>

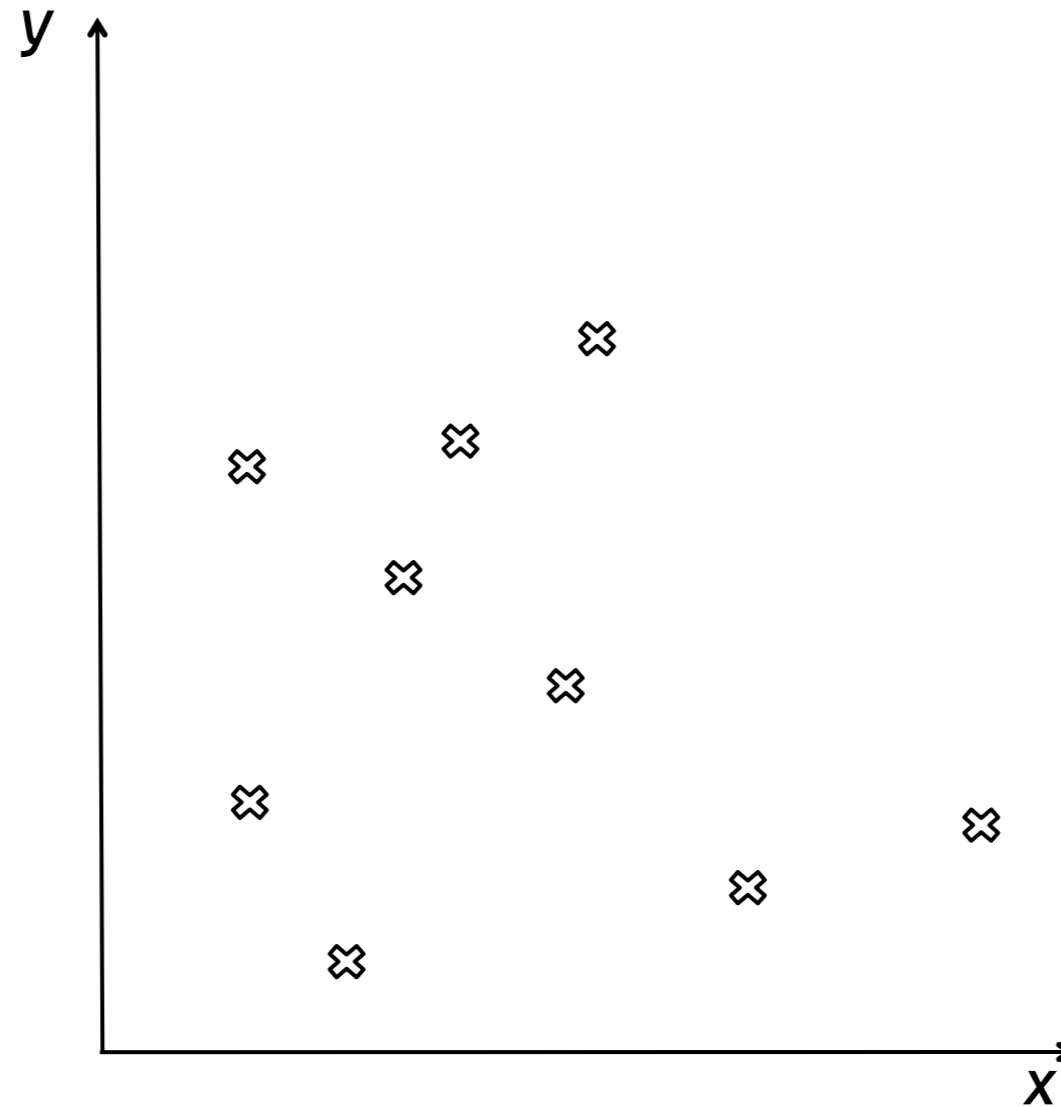


Extra Slides

Guided Improvement



maximize x
maximize y

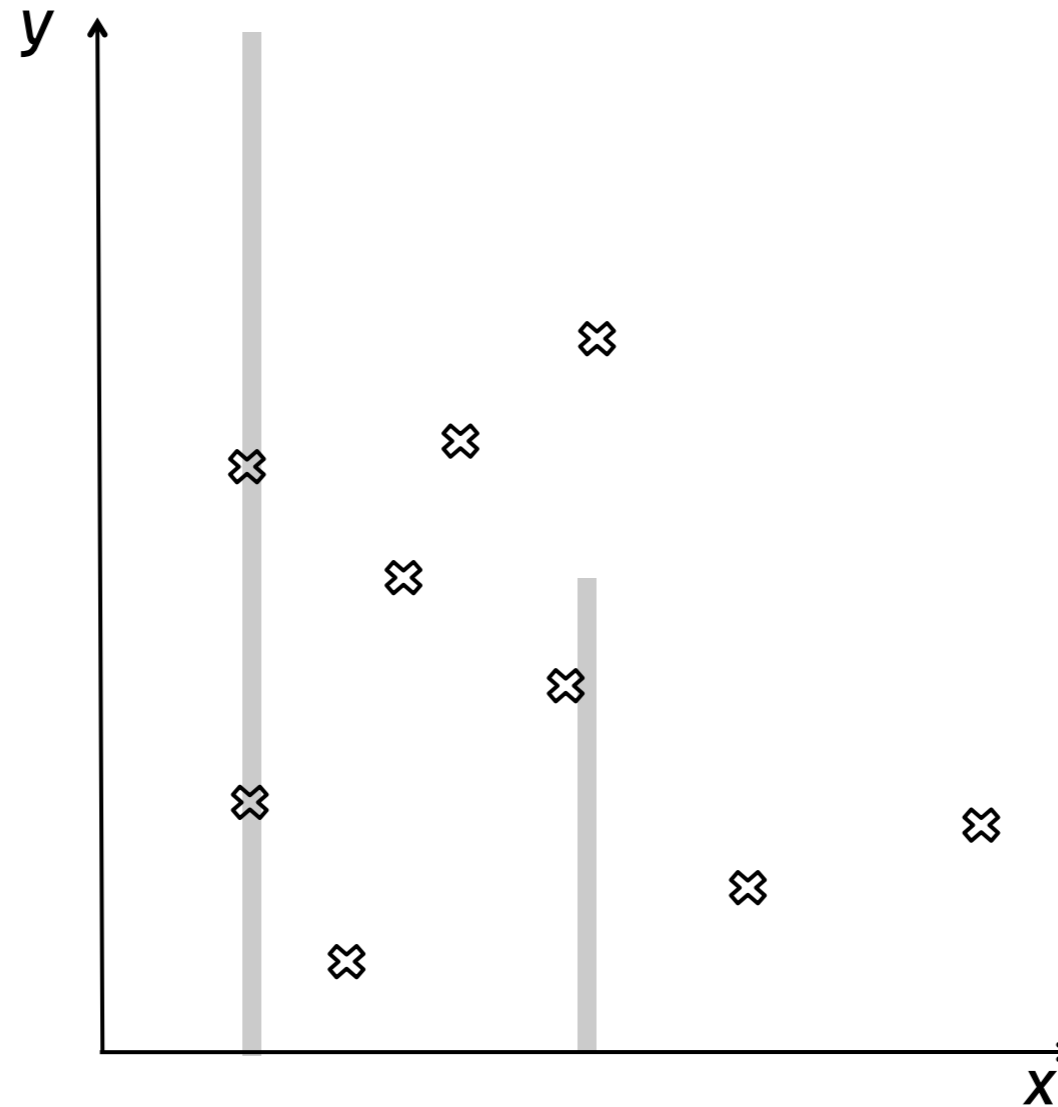


Output:

Guided Improvement



maximize x
maximize y

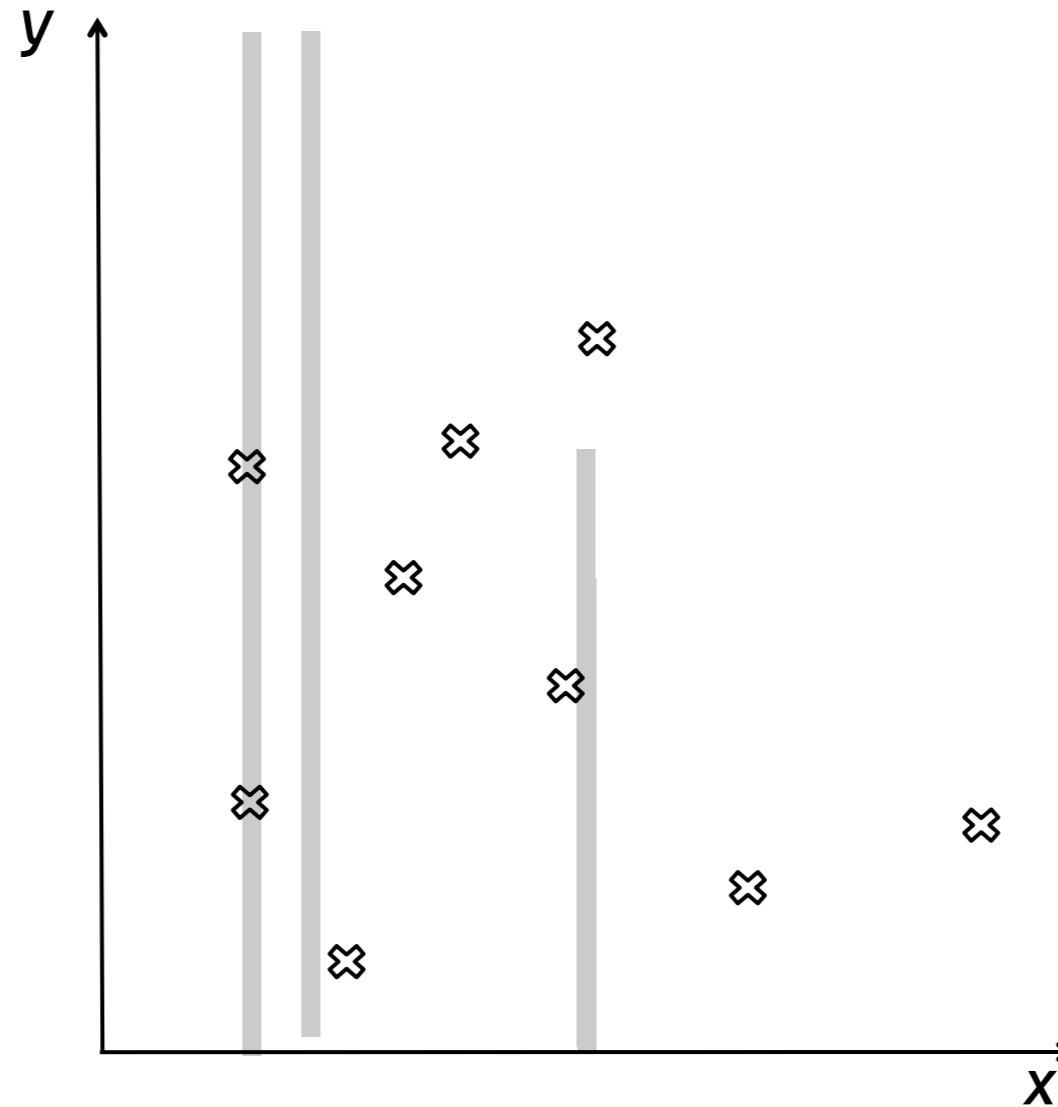


Output:

Guided Improvement



maximize x
maximize y

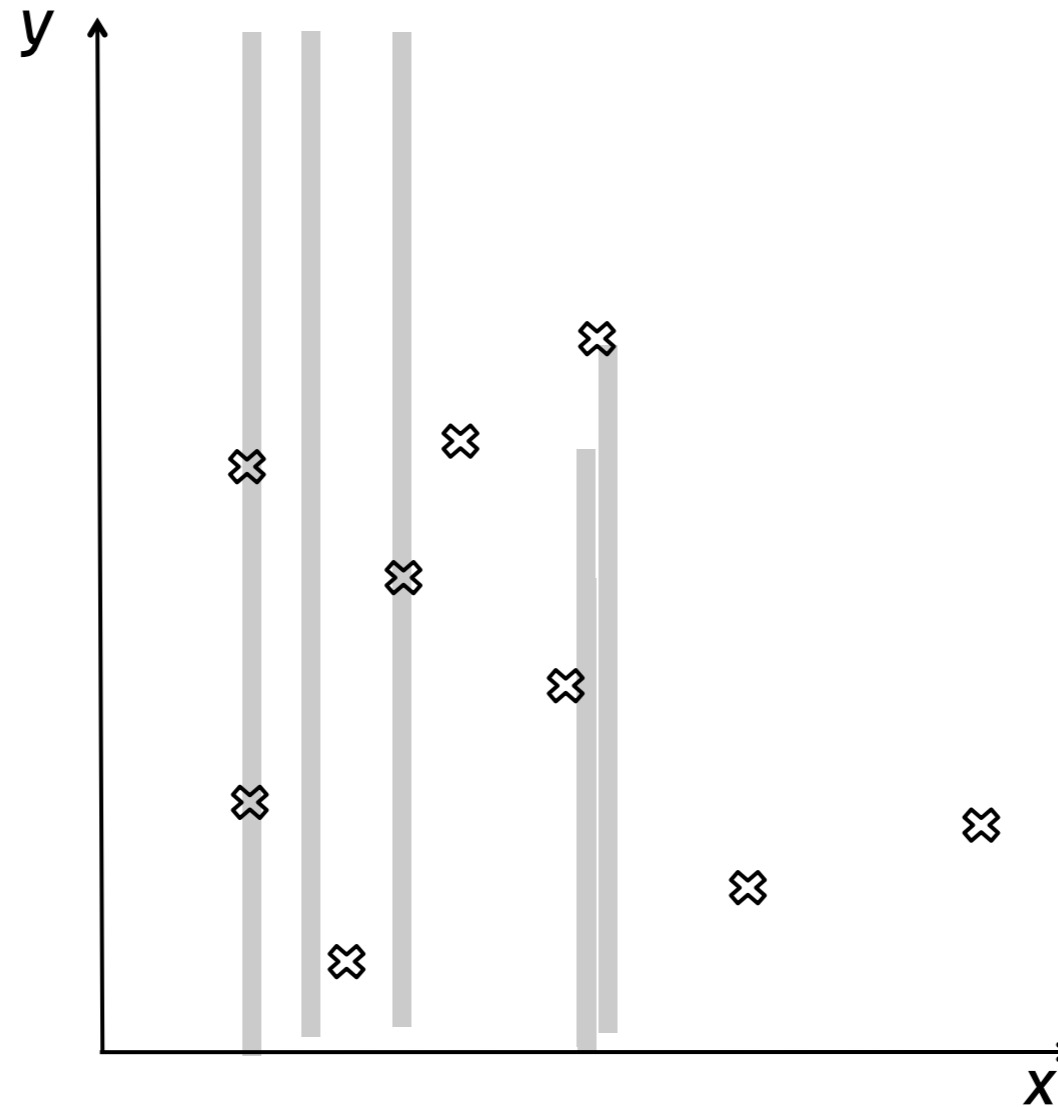


Output:

Guided Improvement



maximize x
maximize y

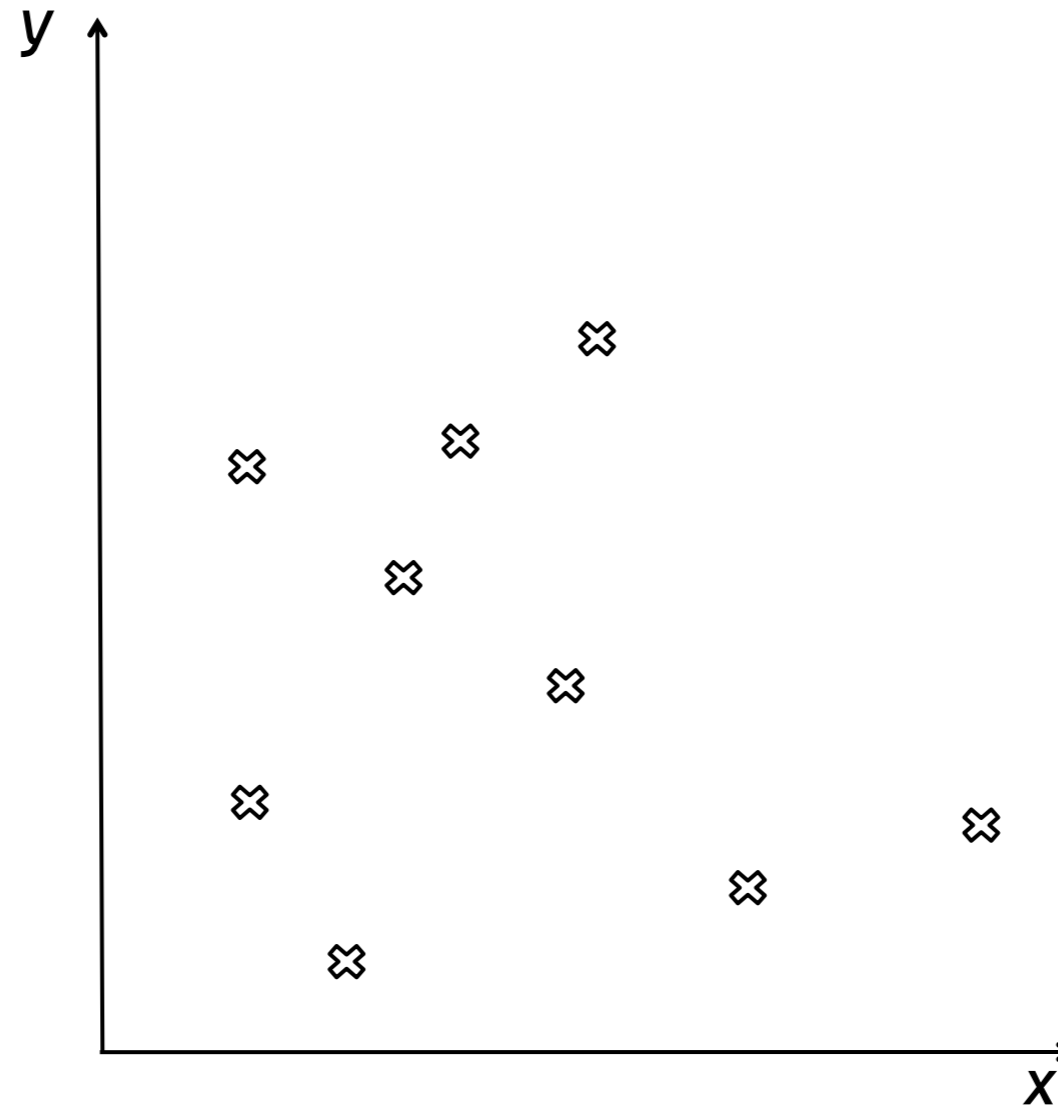


Output:

Guided Improvement



maximize x
maximize y

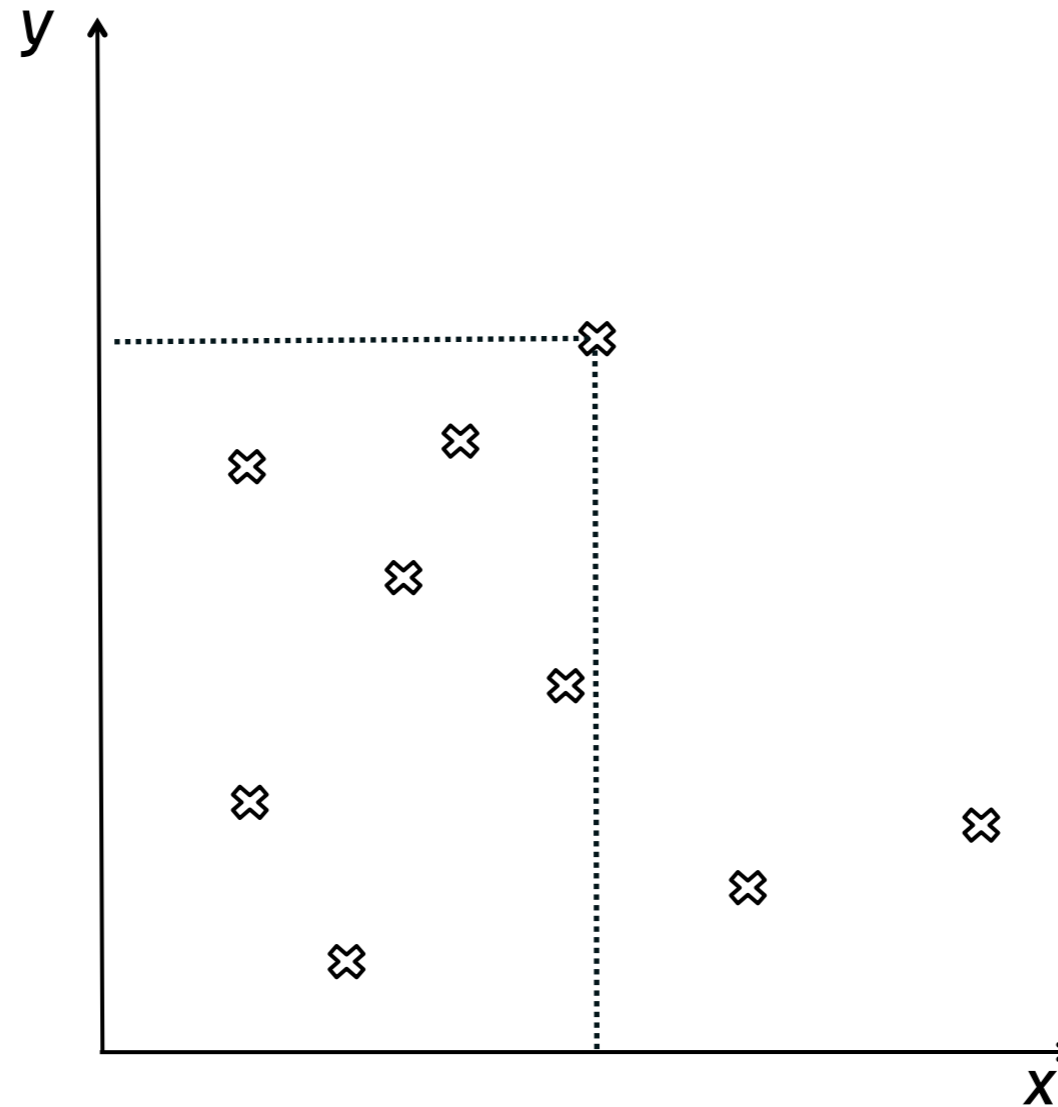


Output: S1, S2,

Guided Improvement



maximize x
maximize y

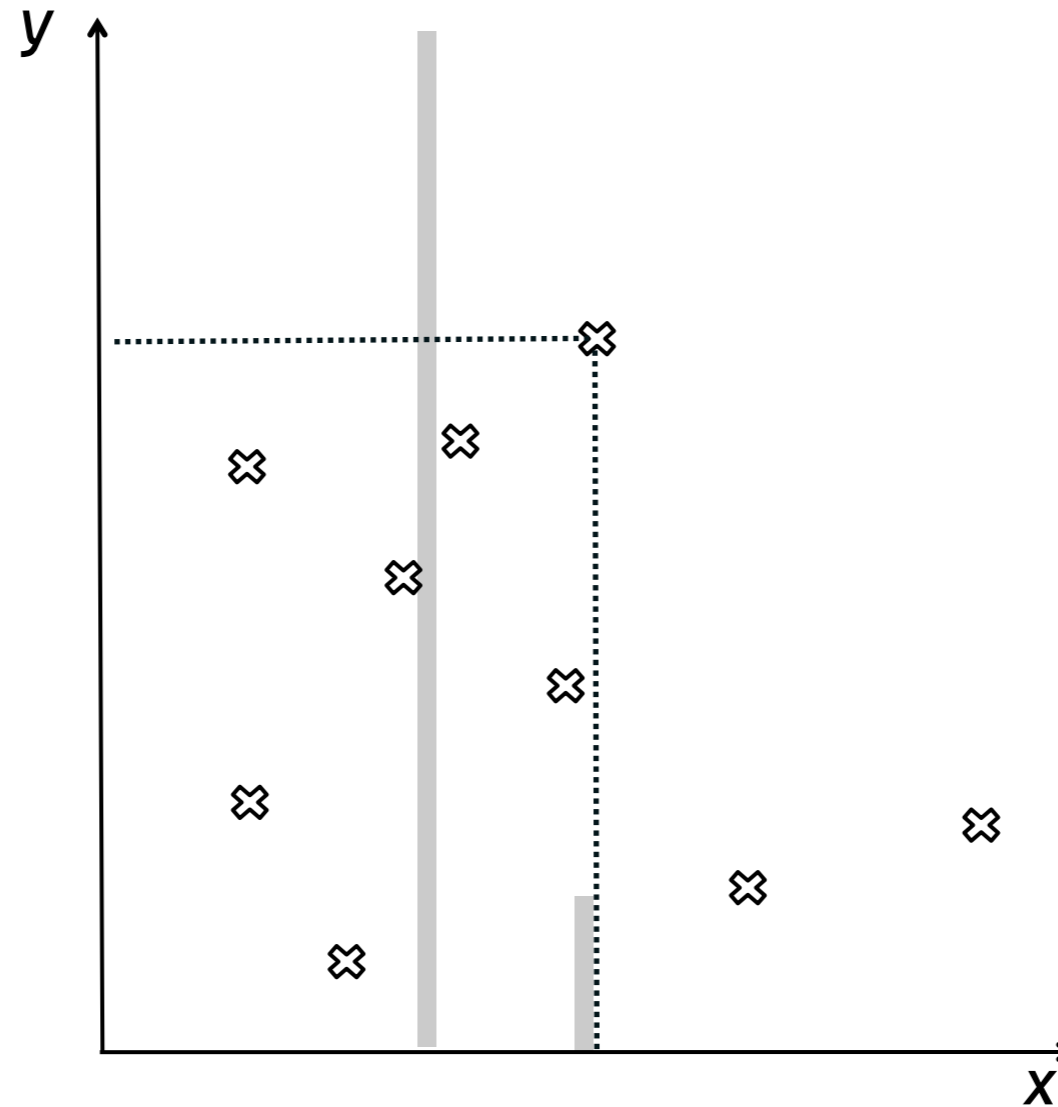


Output: S1, S2,

Guided Improvement



maximize x
maximize y

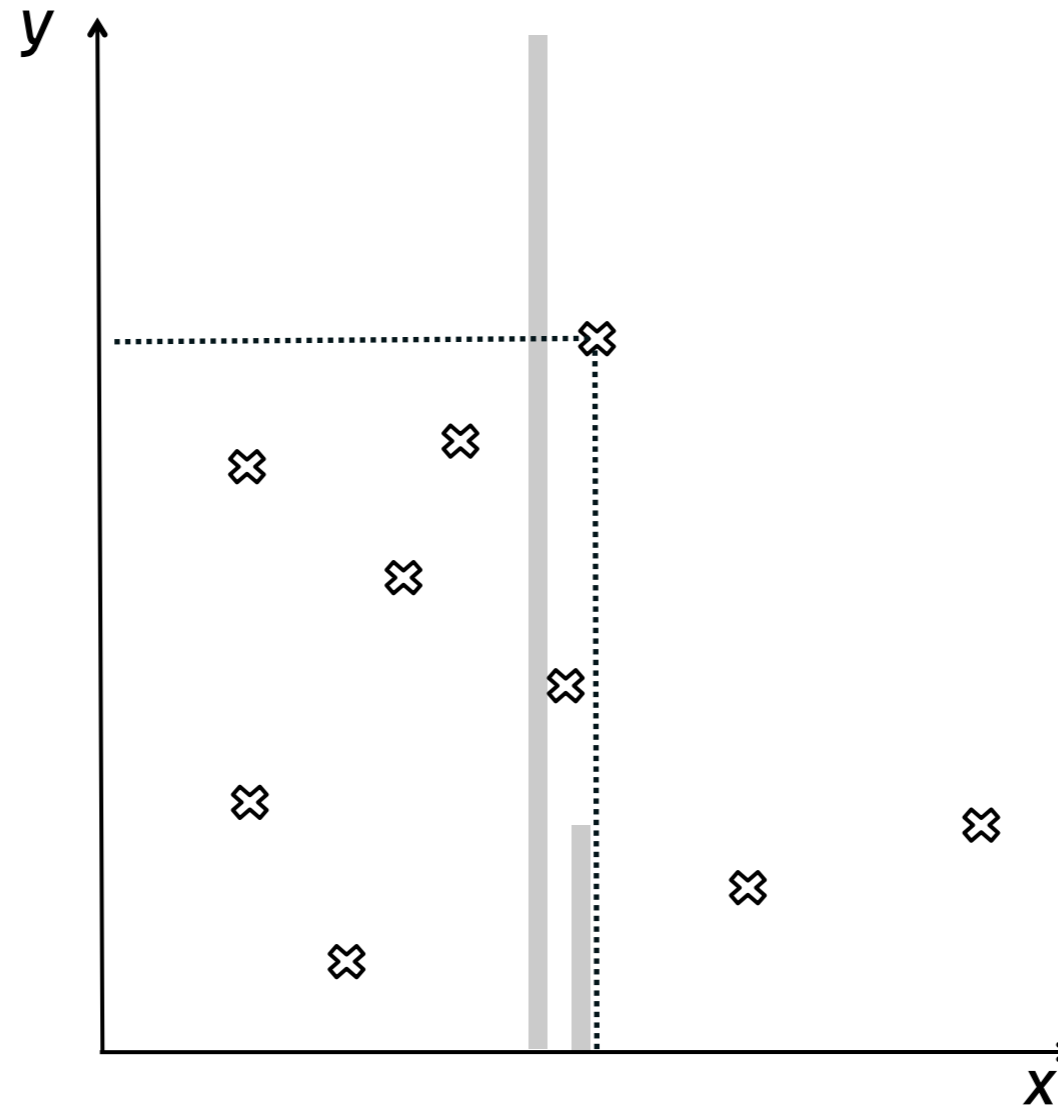


Output: $S_1, S_2,$

Guided Improvement



maximize x
maximize y

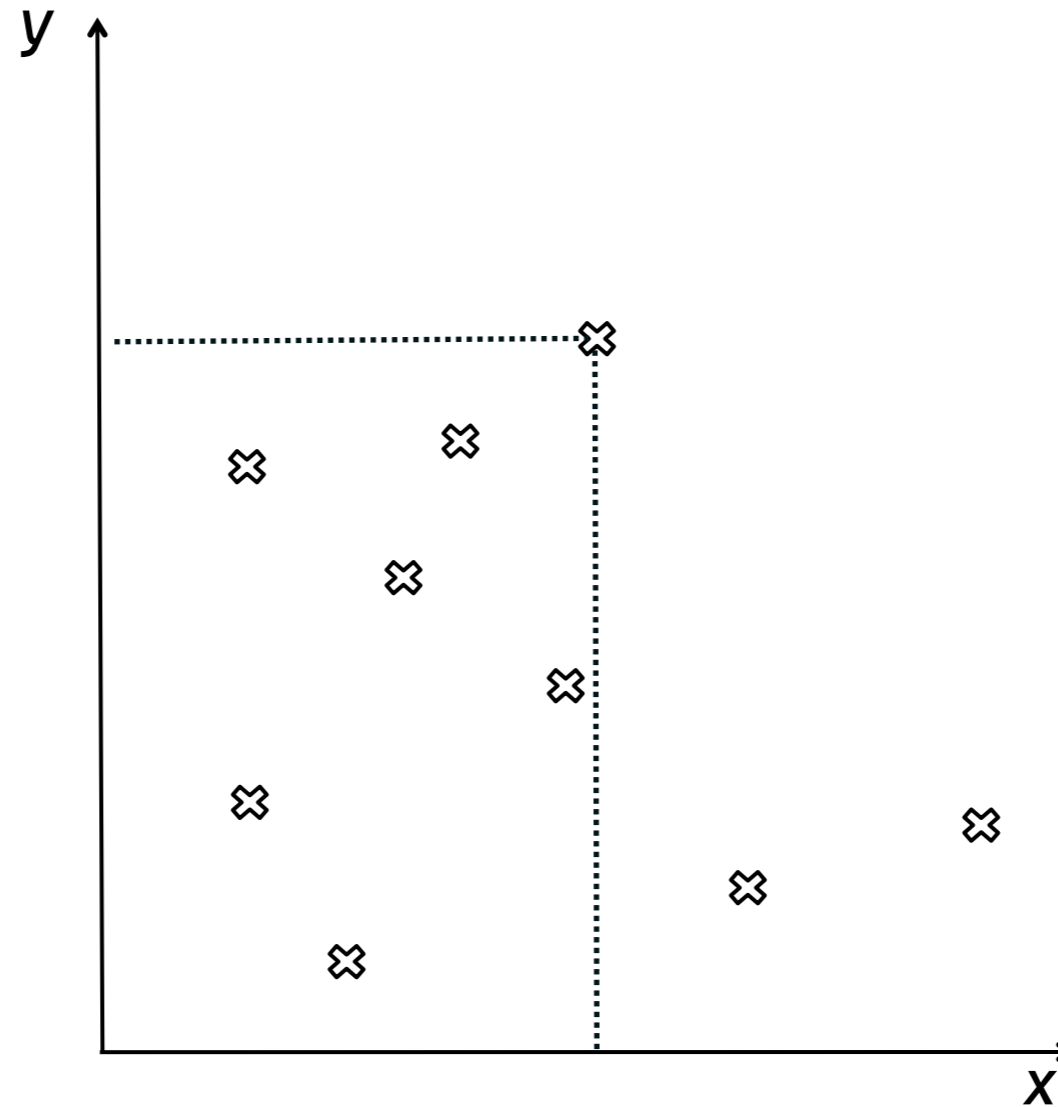


Output: $S_1, S_2,$

Guided Improvement



maximize x
maximize y

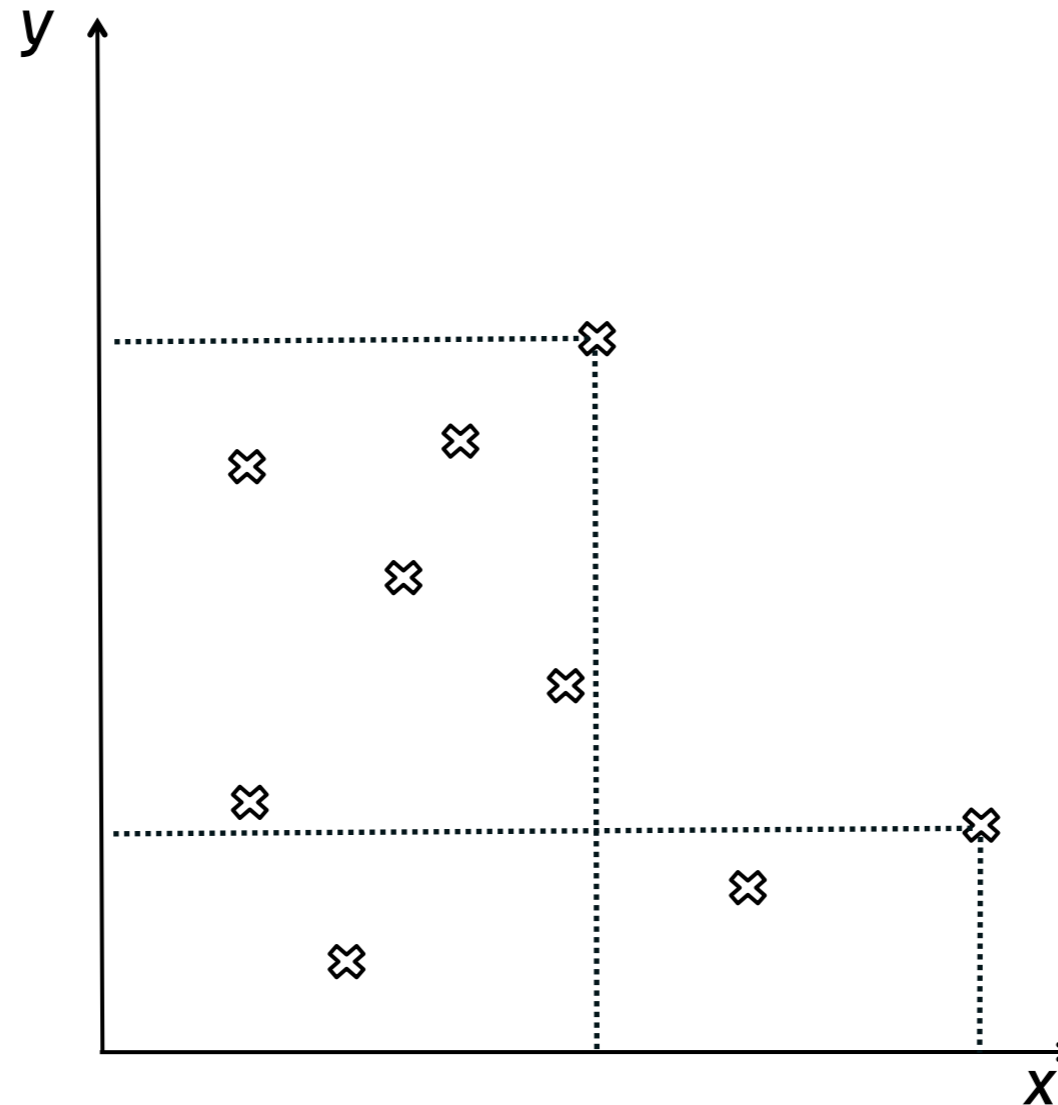


Output: S1, S2, S3,

Guided Improvement



maximize x
maximize y



Output: S1, S2, S3, Termination

Concorde SST



Motivation



- Finding *optimal solutions* is a key-problem
- Relevant for many areas (Science & Industry)
- Decades of research
 - Multitude of solving methods
 - Multitude of tools

Motivation



- Most tools solve **Single-Objective** problems
 - We optimize one criteria (e.g. *Cost*)
- Insufficient for many real-world applications
 - E.g. minimize *Cost* and maximize *Performance*



Multi-Objective Optimization (MOOP)

- Finding optimal solutions for multiple objectives

Motivation



- Solving MOOPs is hard
 - Sophisticated algorithms
 - Heuristic approaches
 - *(not covered here)*
- In this talk: How do we formulate MOOPs?
 - Make MOOP accessible to non-experts
 - Make MOOP as easy as working with Spreadsheets

Example MOOP

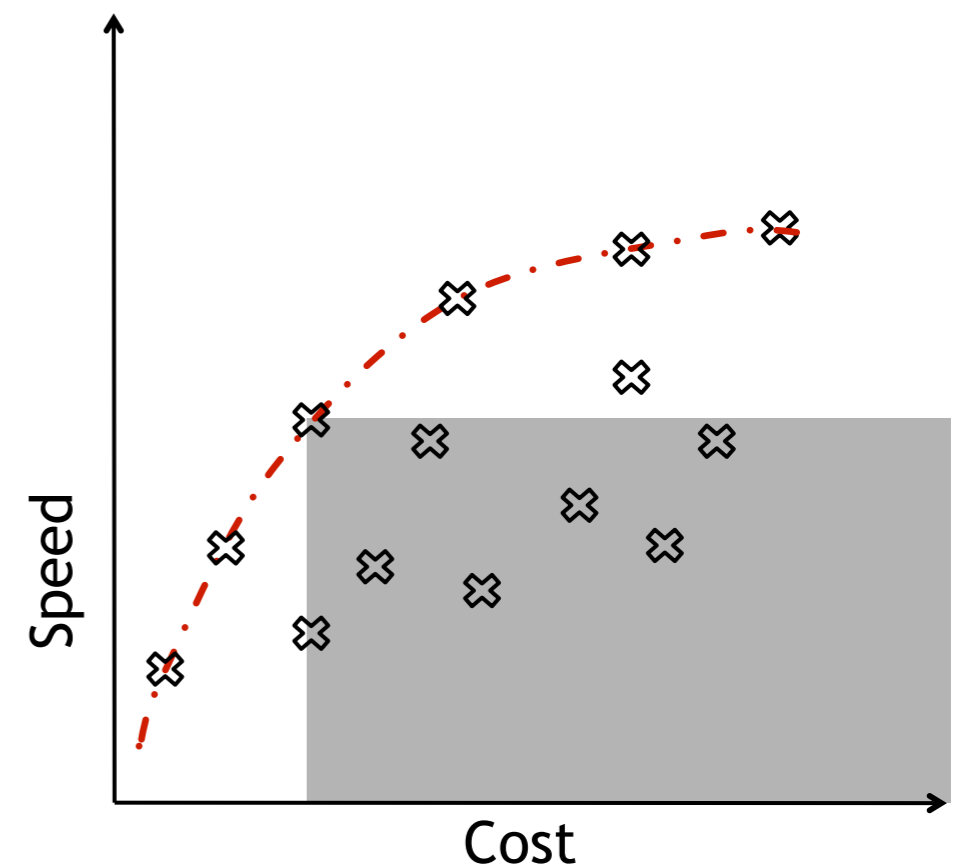


- Compose bicycle from custom-made parts
 - Frame: steel, aluminum, ...
 - Forks: RockShock XL, LST Air, ...
 - Front mech: Srum 9, Srum 10, Shimano XR, ...
 - Rear mech: Srum 9, Srum 10, Shimano XR, ...
 - Chain: Srum 9, Srum 10, Shimano XR, ...
- Objectives / Metrics:
 - minimize *Cost*
 - maximize *Performance*

Case Study



- Expect solutions with
 - High *Cost* & high *Performance*
 - Low *Cost* & low *Performance*
 - Medium *Cost* & medium *Performance*
- Some solutions *dominate* other ones
- Non-dominated solutions are *Pareto optimal*



Insight / Conjecture

