Multiple Objective Optimization

The man who, though exceedingly hungry and thirsty and both equally, yet being equidistant from food and drink, is therefore bound to stay where he is.

Aristotle, On the Heavens (Book II)

Introduction

Multi-objective optimization (MOO) problems involve multiple objective functions to be optimized simultaneously.

$$\begin{aligned} \min \quad & \mathbf{F}(x) = (f_1(x), f_2(x), \ldots, f_k(x)) \\ \text{s.t.} \quad & x \in X \end{aligned}$$

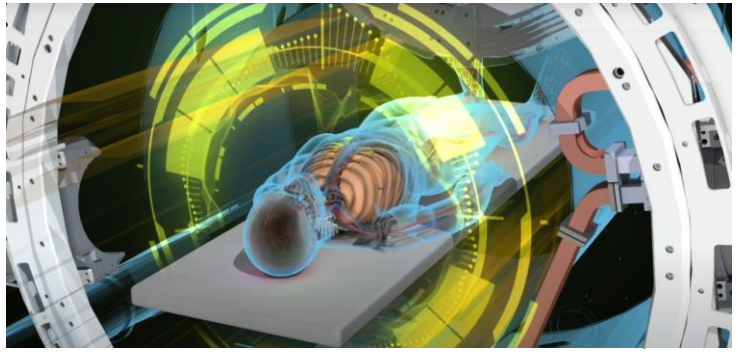
Often, the $k \geq 2$ objectives are conflicting (i.e., competing against each other for the same resources)

Multiple-objective Optimization Examples

- Portfolio optimization:
  - maximize revenue
  - minimize risk

- Scheduling truck routes:
  - minimize deadhead miles (driving empty truck)
  - equalize workload among drivers

- Radiation therapy:
  - minimize normal tissue dose
  - maximize does to tumor regions



Commensurate vs. non-commensurate

Multi-objective optimization (MOO) problems involve multiple objective functions to be optimized simultaneously.

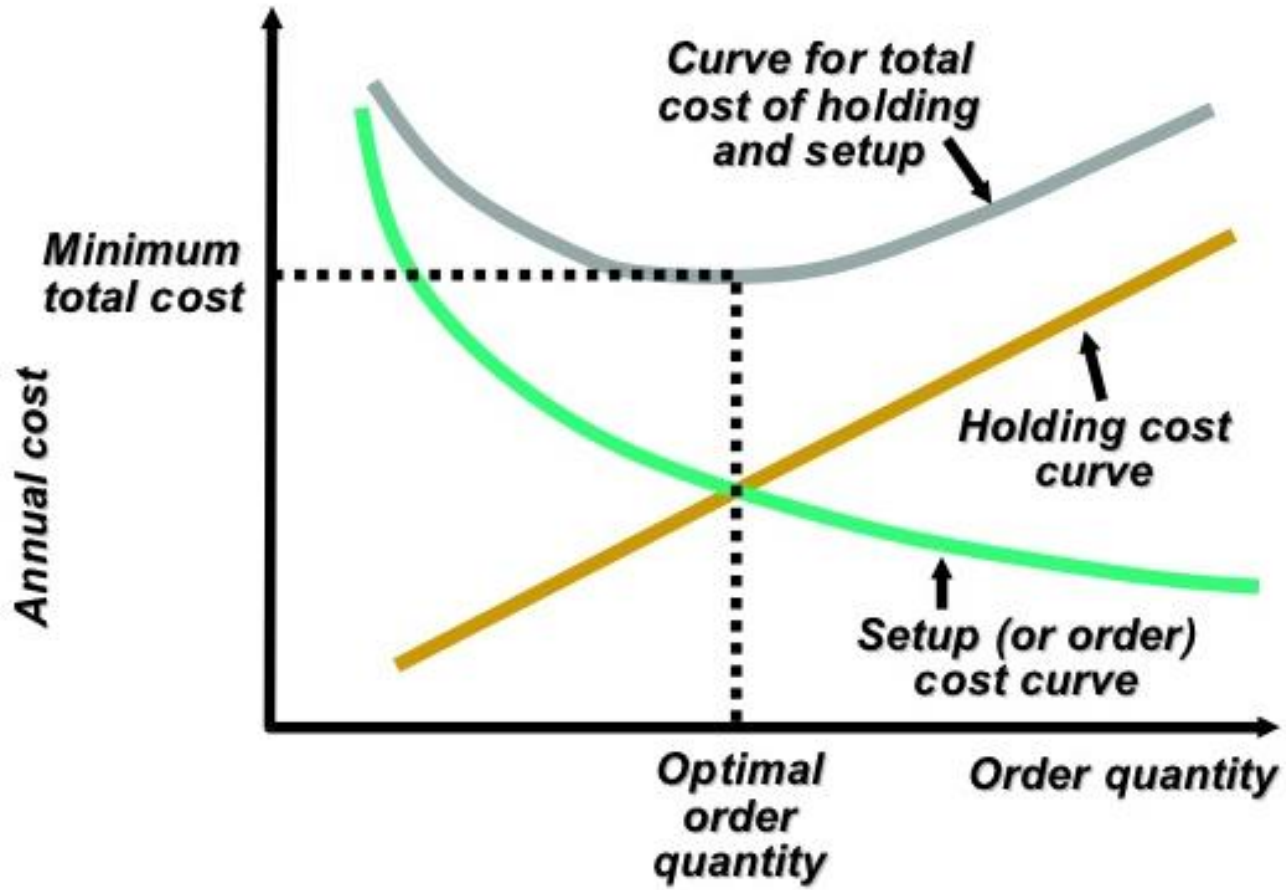
$$\begin{aligned} \min \quad & \mathbf{F}(x) = (f_1(x), f_2(x), \ldots, f_k(x)) \\ \text{s.t.} \quad & x \in X \end{aligned}$$

Often, the $k \geq 2$ objectives are conflicting (i.e., competing against each other for the same resources)

The objective functions may be commensurable or non-commensurable

Commensurate Objectives Example

- Inventory control, i.e., minimize: Order Cost and Holding Cost



Multiple objective optimization

For commensurable objectives, we can typically use one objective function by linking the two objectives with their common unit.

But, some problems cannot be reduced and <u>must</u> be solved using pure MO techniques.

We are most interested in problems with multiple non-commensurable and conflicting objectives.

Non-Commensurate Example

- Suppose you need to fly on a long trip: should you choose the cheapest ticket (more connections) or shortest flying time (more expensive)?
- time and money – fundamentally different
- Also, the relative importance will vary.
  - maybe a business emergency that you need to go fix quickly
  - or, maybe you are on a very tight budget

Pareto-Optimal Solutions

- A constrained MOO problem will have many solutions in the feasible region
- Even though we may not be able to assign numerical relative importance to the multiple objectives, we can still classify some possible solutions as better than others

Pareto-Optimal Solutions Example

Suppose in our airplane-trip example from earlier, we find the following tickets:

Ticket	Travel Time (hrs)	Ticket Price ($)
A	10	1700
B	9	2000
C	8	1800
D	7.5	2300
E	6	2200

Pareto-Optimal Solutions Example

Suppose in our airplane-trip example from earlier, we find the following tickets:

Ticket	Travel Time (hrs)	Ticket Price ($)
A	10	1700
B	9	2000
C	8	1800

For tickets A & B, we can't say that either is superior without knowing the relative importance of travel time vs. price.

Pareto-Optimal Solutions Example

However, comparing tickets B & C shows that C is better than B in both objectives → **C dominates B**

Ticket	Travel Time (hrs)	Ticket Price ($)
A	10	1700
B	9	2000
C	8	1800
D	7.5	2300

As long as C is a feasible option, there is no reason we would choose B.

Pareto-Optimal Solutions Example

Similarly, **E dominates D**

Ticket	Travel Time (hrs)	Ticket Price ($)
A	10	1700
B	9	2000
C	8	1800
D	7.5	2300
E	6	2200

Pareto-Optimal Solutions Example

The other options (A, C, & E) have a trade-off associated with time vs. price, so none is clearly superior to the others.

Ticket	Travel Time (hrs)	Ticket Price ($)
A	10	1700
C	8	1800
E	6	2200

This is the **non-dominated** set of solutions.

Pareto Optimality

- MOO → tradeoffs between competing objectives
- Pareto approach → exploring the **tradeoff surface**, yielding a set of possible solutions

Pareto Optimum: Definition

- A solution is Pareto optimal iff:
  - It is <u>at least</u> as good as all other candidates for all objectives, and
  - It is <u>better</u> than all other candidates for at least one objective.

Pareto Dominance

Given vector of objective functions and assuming minimization:

$$(f_1(x), f_2(x), \ldots, f_k(x))$$

Solution x_1 dominates x_2 if:

$$f_i(x_1) \leq f_i(x_2) \quad \forall i \in \{1, \ldots, k\}$$

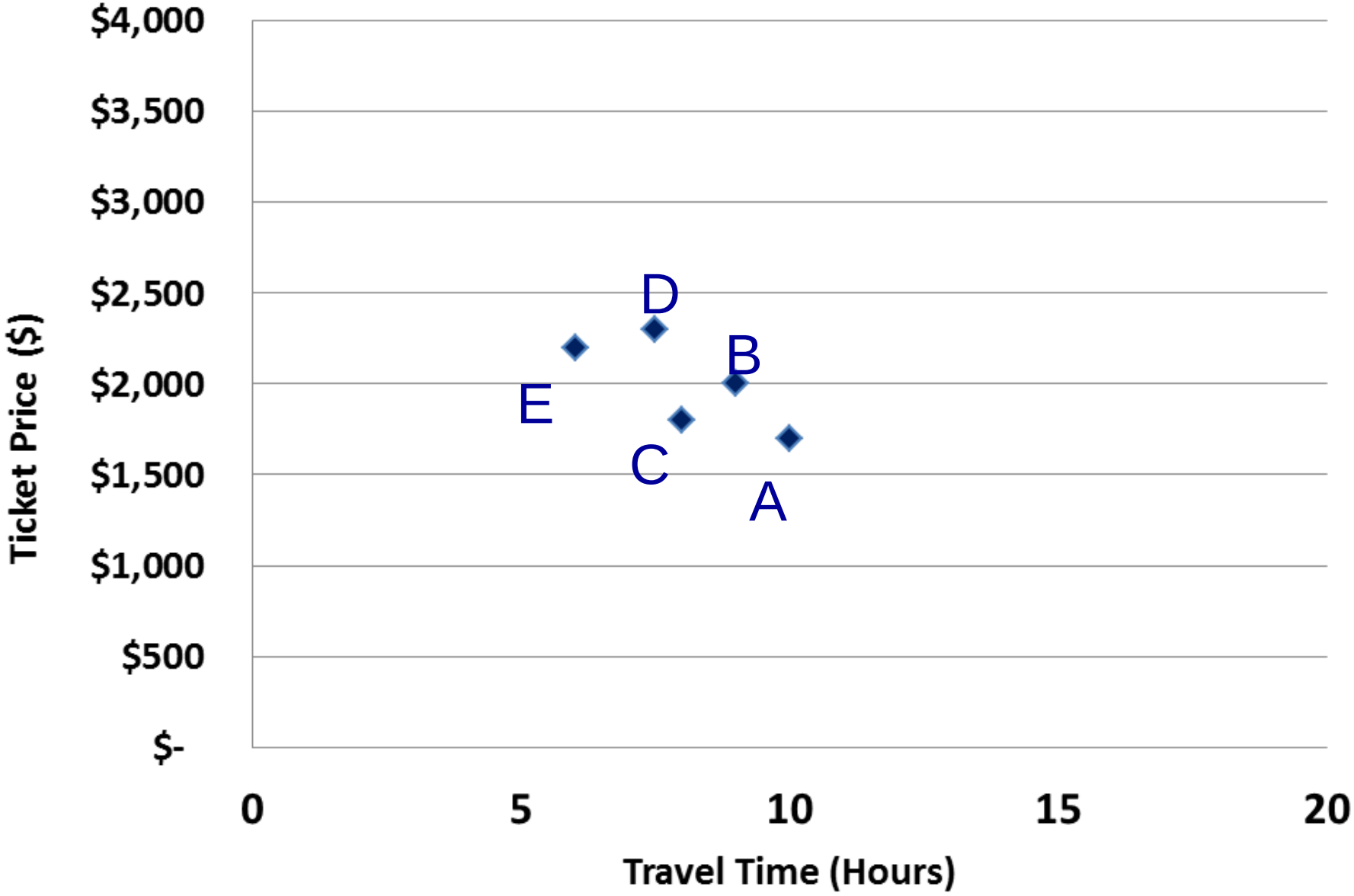
and

$$\exists i \in \{1, \ldots, k\} : f_i(x_1) < f_i(x_2)$$

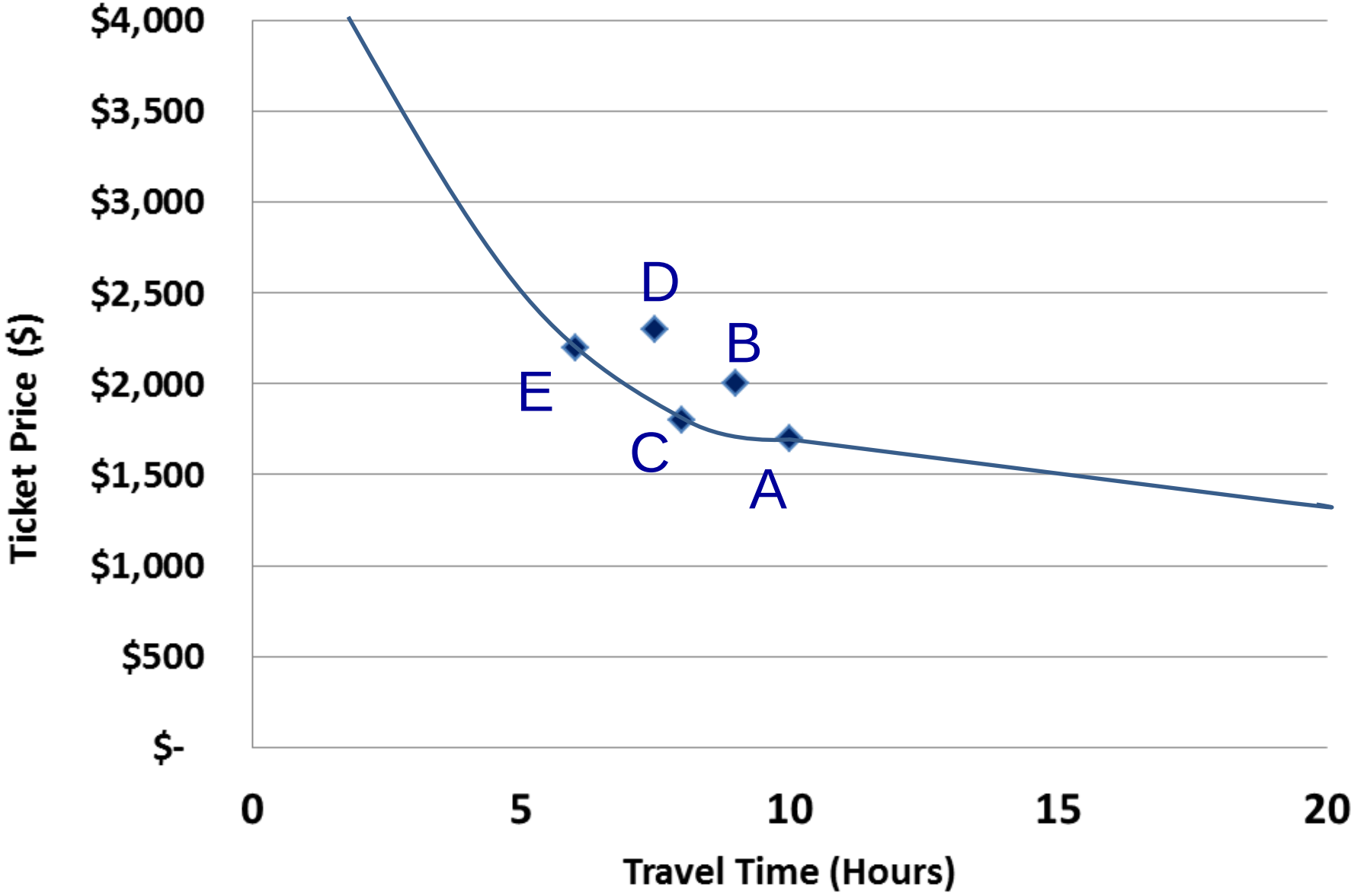
Pareto Optimal Solution

- **Non-dominated solution set**
  - set of all solutions *that are not dominated* by any member of the solution set
- The non-dominated set of the entire feasible decision space is called the Pareto optimal set
- A Pareto optimal solution cannot be improved w.r.t. any objective without degrading one or more other objectives

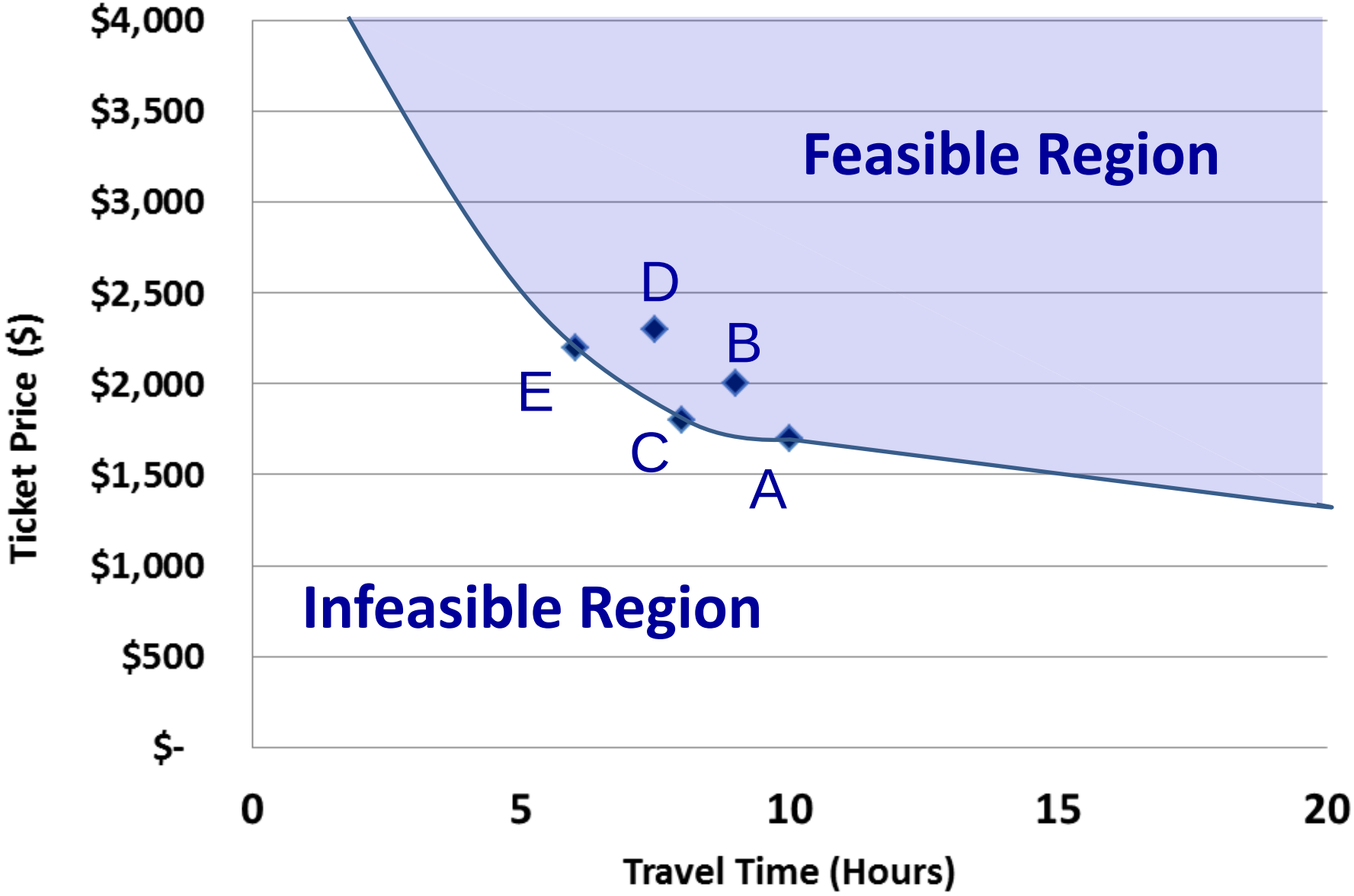
Graph of Solutions



Graph of Solutions



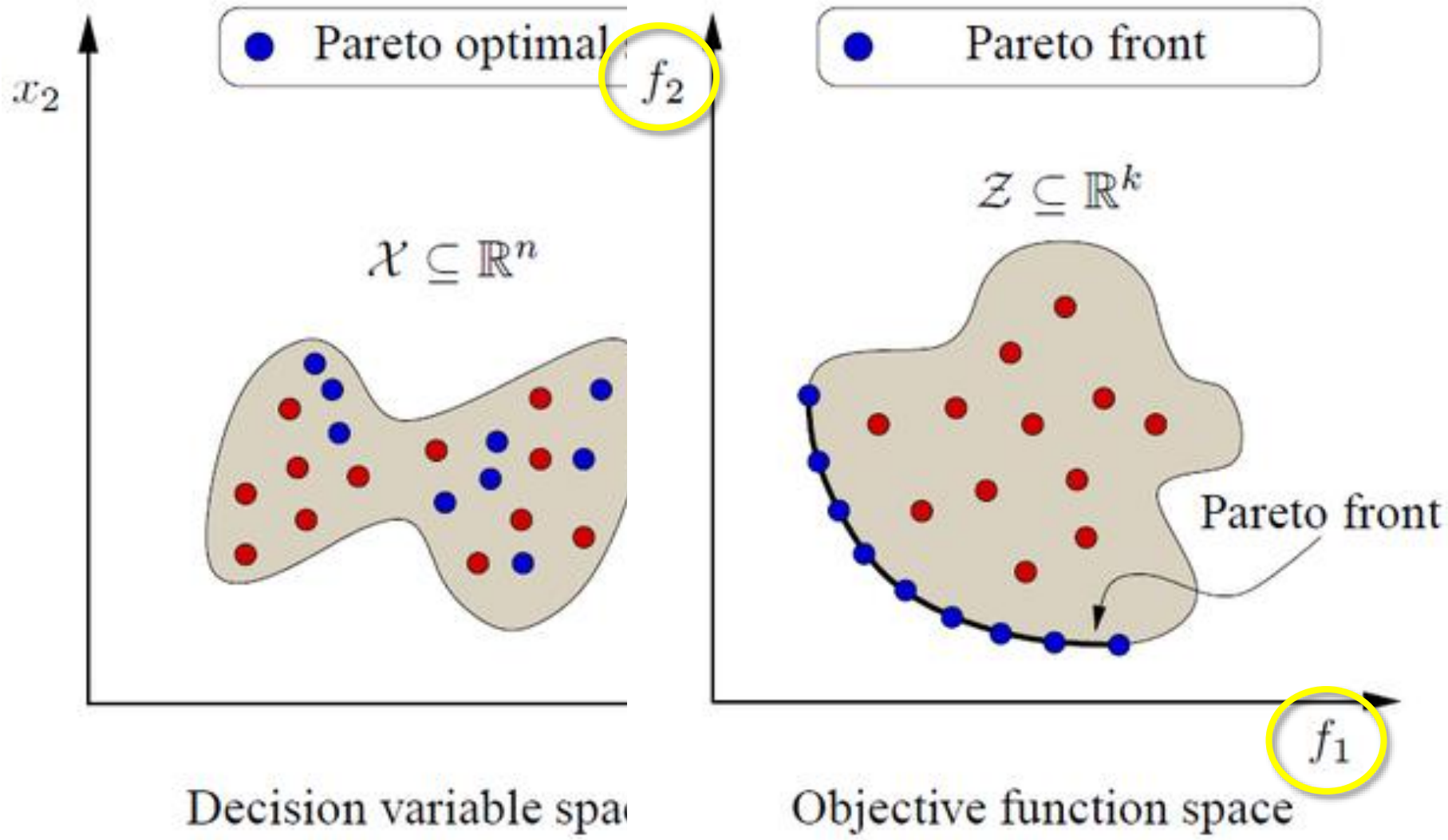
Graph of Solutions



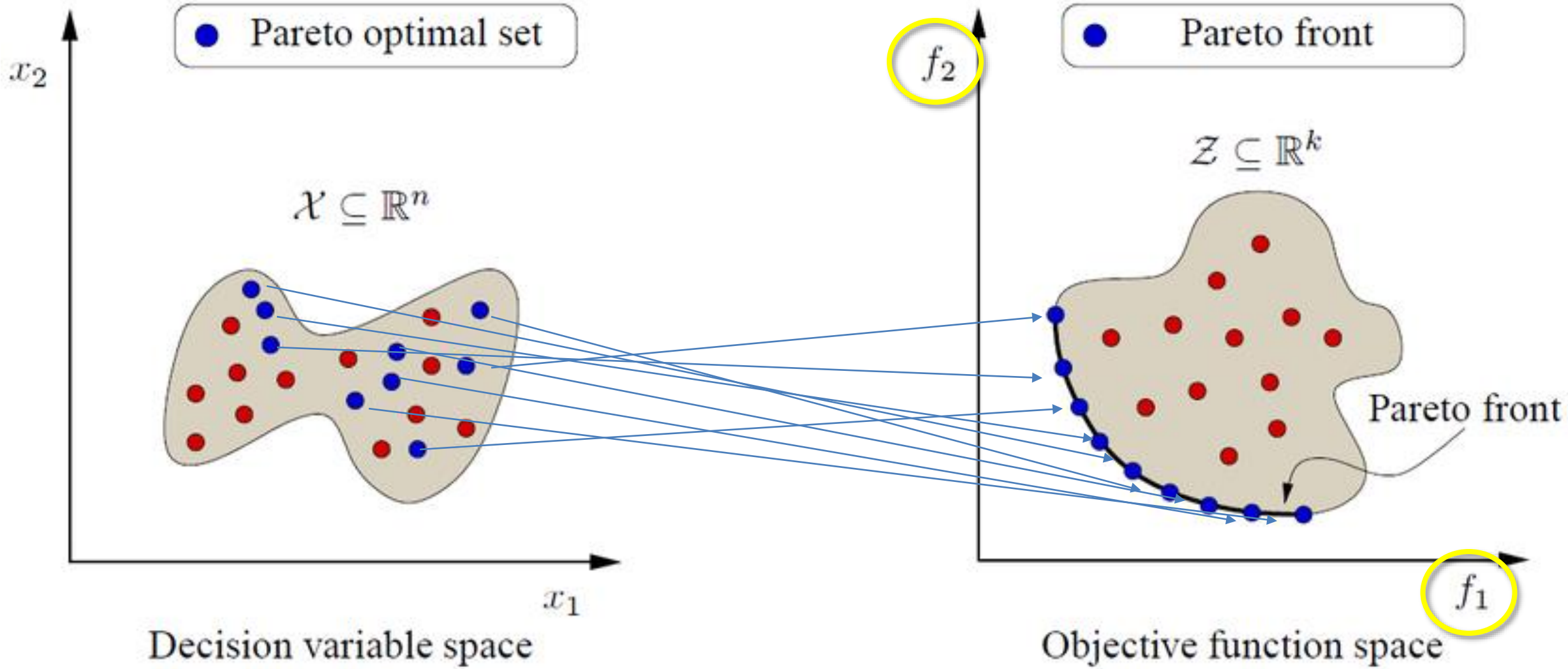
Pareto-optimal Solutions

- The curve is called the **Pareto front** and solutions on it are called Pareto-optimal
- This front is at the boundary of feasible and infeasible solutions in **objective space**
- Thus, it is important in MOO to find many solutions as close as possible to the Pareto front

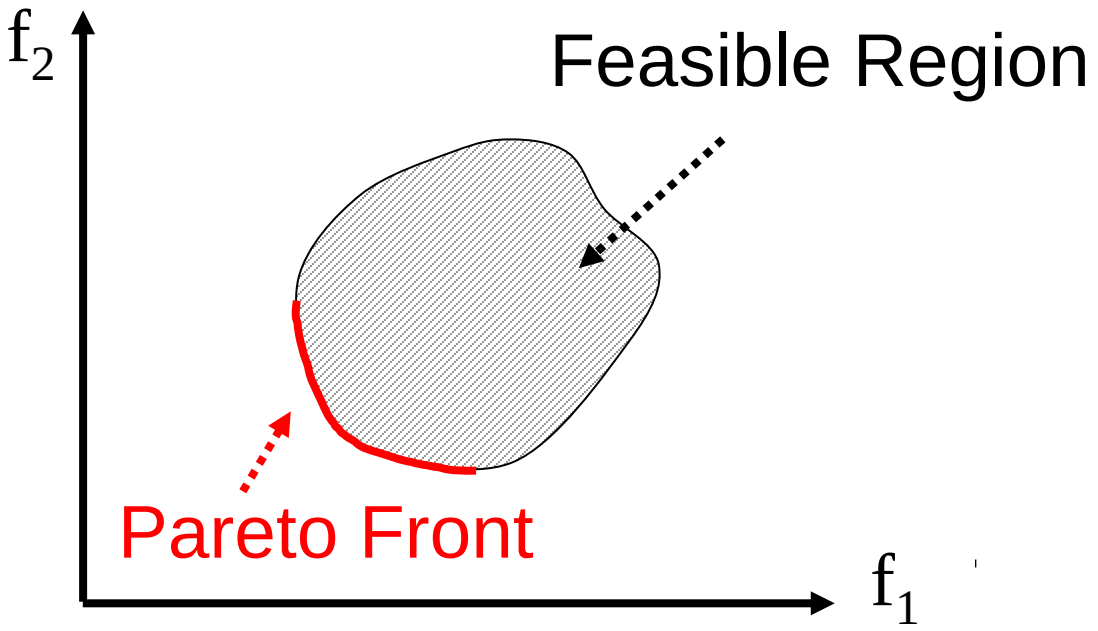
Objective space

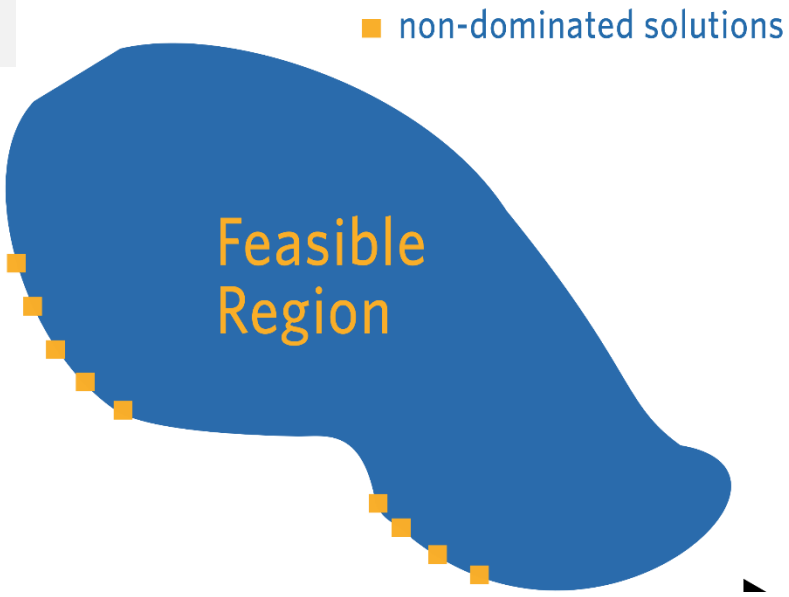


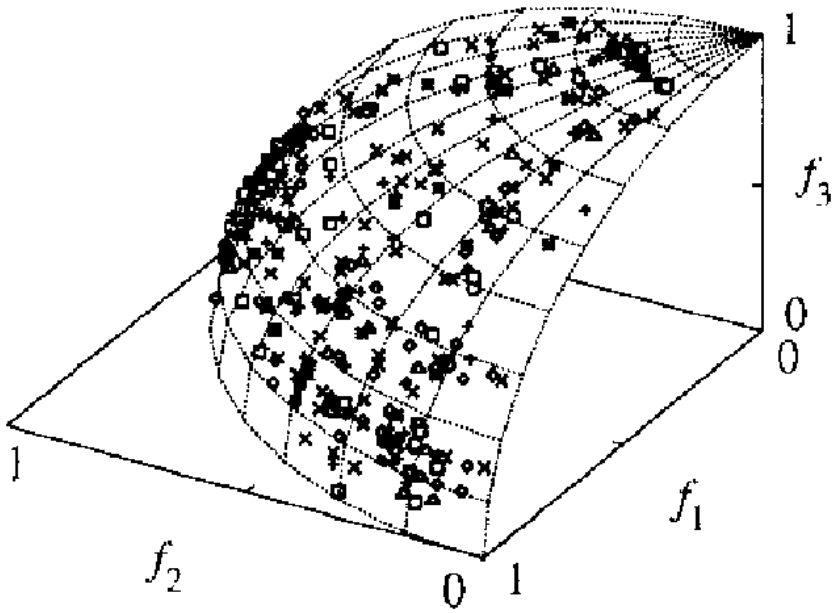
Objective space



Graphical Example







Three General Approaches

Traditional methods

- **Preemptive Optimization**
  - sequential optimization of individual objectives (in priority order)
- **Single Objective Techniques**
  - Method I: weighted sum of objectives
  - Method II: constraint method

- **Purely Multi-Objective**
  - many population-based methods

Metaheuristics methods

Preemptive Optimization Steps

1. **Prioritize objectives** according to predefined criteria (problem-specific)
2. Optimize highest-priority objective function
3. Introduce new constraint based on optimum value just obtained
4. Repeat steps 2 & 3 for every other objective function, in succession

Preemptive Optimization Example

Objectives in priority order: (i) maximize revenue, $f_1(x)$ (ii) minimize risk, $f_2(x)$

Step 1: solve the primary objective problem,

max revenue: $f_1(x)$
s.t. $x \in X$ ← *this is shorthand to represent all the original problem constraints*

→ *Assume the optimal solution produces a maximum revenue of* $\$15{,}000$

Step 2: solve the secondary objective problem with an additional constraint,

min risk: $f_2(x)$
s.t. $x \in X$
s.t. $f_1(x) \geq \$15{,}000$ ← *additional constraint*

Single Objective Techniques: scalarization a.k.a. weighted sum

1. **Assign weights** to each objective function according to predefined criteria (e.g., decision-maker preference)
2. Sum weighted functions to create new composite objective function
3. Solve as a regular, single-objective optimization problem with the new composite objective function
4. Optionally, repeat with different weights

Scalarization (weighted-sum technique)

$$\min \quad \sum_{i=1}^{k} \gamma_i f_i(x)$$

$$\text{s.t.} \quad \sum_{i=1}^{k} \gamma_i = 1$$

$$\gamma_i > 0, \quad i = 1, \ldots, k$$

$$x \in X$$

Scalarization Example

Objectives: (i) maximize revenue, $f_1(x)$ (ii) minimize risk, $f_2(x)$

Step 1: decide on weights, e.g., $\gamma_1 = 0.8, \gamma_2 = 0.2$

Steps 2 and 3: create new composite objective function and solve

$$\begin{aligned} \max \quad & 0.8 f_1(x) - 0.2 f_2(x) \\ \text{s.t.} \quad & x \in X \end{aligned}$$

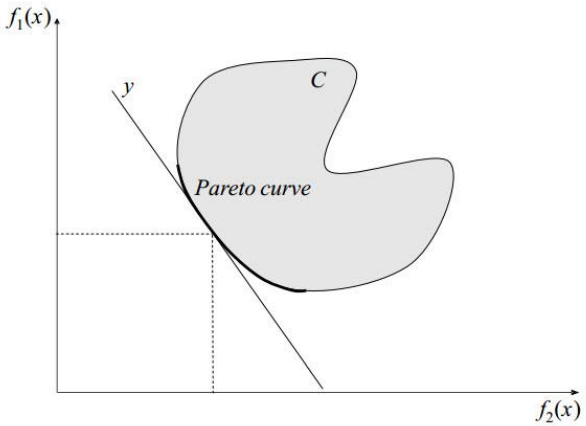
Step 4 (optional): choose *new weights* γ_1, γ_2 goto Step 2

**if you want to estimate Pareto frontier, repeat many times for different weights*

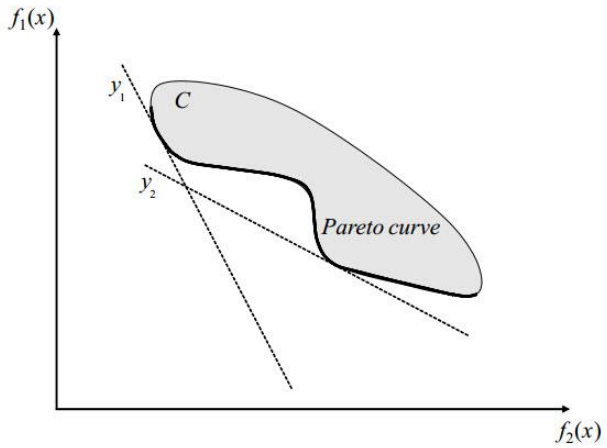
Scalarization Notes

- Need to consider the units when deciding weights for objectives.
  - May need to normalize objective value scales or apply some conversion prior to weighting.

- Scalarization works here.



- Scalarization does not work here.



Single objective techniques: ε-constraint Method

1. Choose one objective to be optimized
2. The remaining objectives are modeled as constraints, e.g., to be less than some target value ϵ.

$$
\begin{aligned}
\min \quad & f_j(x) \\
\text{s.t.} \quad & f_i(x) \leq \epsilon_i \ \forall i \in \{1, \ldots, k\} \setminus j \\
& x \in X
\end{aligned}
$$

ε-constraint Method example

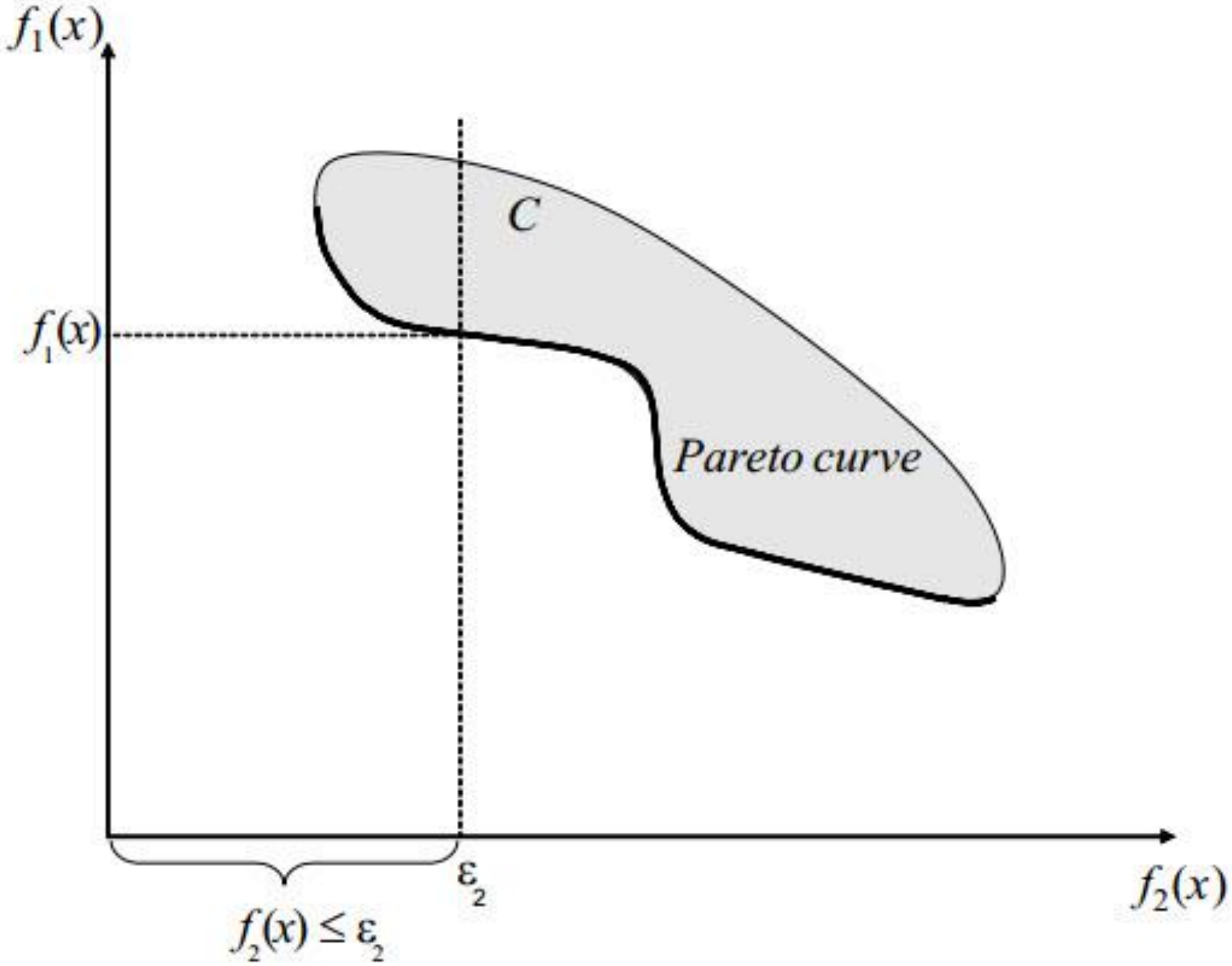
Objectives: (i) maximize revenue, $f_1(x)$ (ii) minimize risk, $f_2(x)$

Step 1: choose one objective to be optimized, e.g., $f_1(x)$
Step 2: choose a value for ε and create a constraint for the other objective

$$\begin{aligned} \max\ & f_1(x) \\ \text{s.t.}\quad & x \in X \\ & f_2(x) \leq \varepsilon \end{aligned}$$

Step 3 (optional): increase ε, go to Step 2.

**if you want to estimate Pareto frontier, complete steps 2-3 many times*


$f_1(x)$
C
$f_1(x)$
Pareto curve
ε_2
$f_2(x)$
$f_2(x) \le \varepsilon_2$

Traditional Approaches

- Advantages:
  - Easy to understand and formulate
  - Simple to solve (using standard techniques)
- Disadvantages:
  - *A priori* prioritization/weighting can end up being arbitrary (due to insufficient understanding of problem): oversimplification
  - Not suited to certain types of MOOs
- Estimation of front can be computationally intensive

EXAMPLES FROM RESEARCH APPLICATIONS

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

Resilience

Center of Excellence for Risk-Based Community Resilience Planning

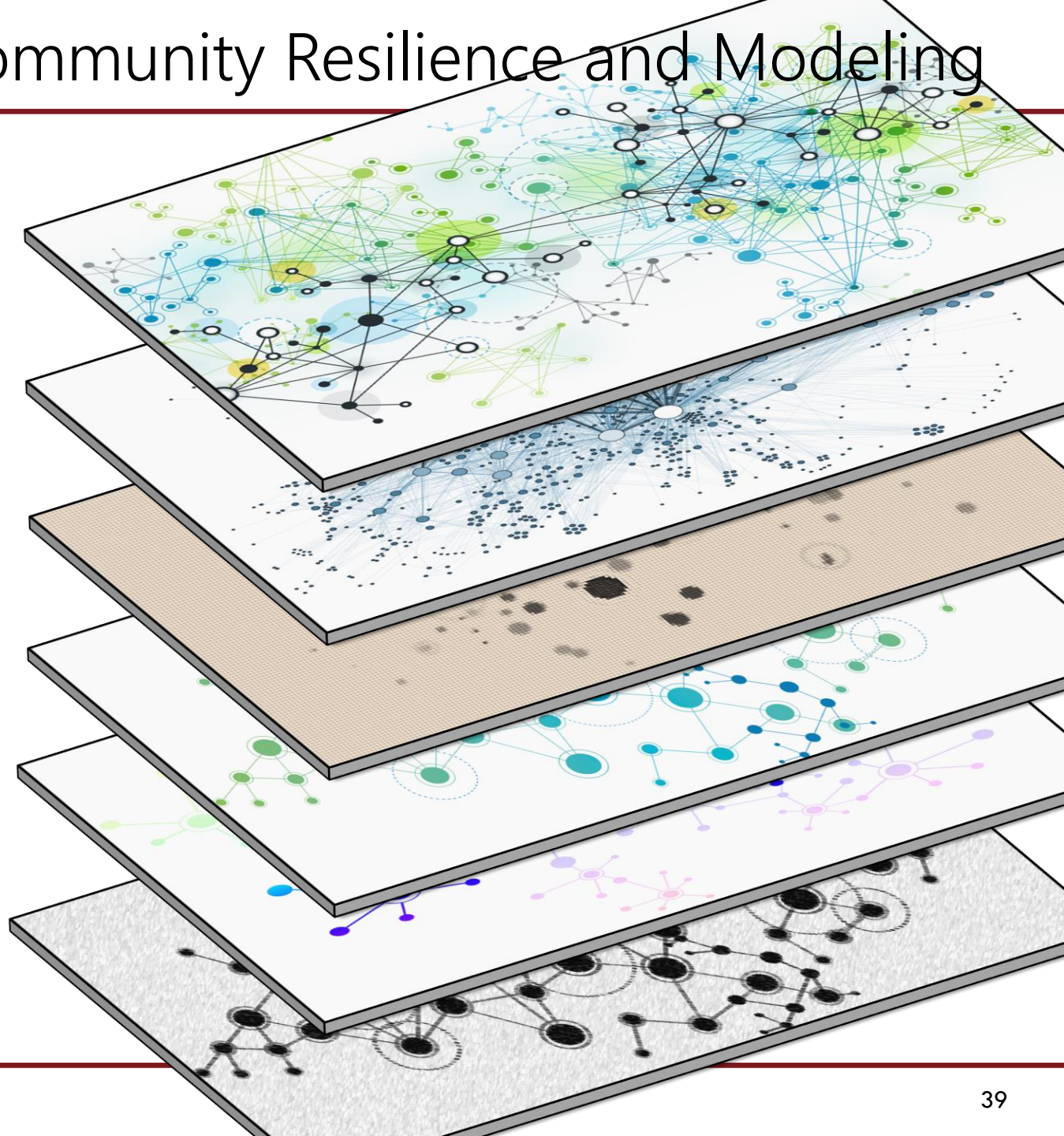
http://resilience.colostate.edu

Colorado State University
The UNIVERSITY of OKLAHOMA
ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
TEXAS A&M UNIVERSITY
Oregon State UNIVERSITY OSU
KU THE UNIVERSITY OF KANSAS
RICE
UNIVERSITY of WASHINGTON
CAL POLY POMONA
University of Colorado Boulder

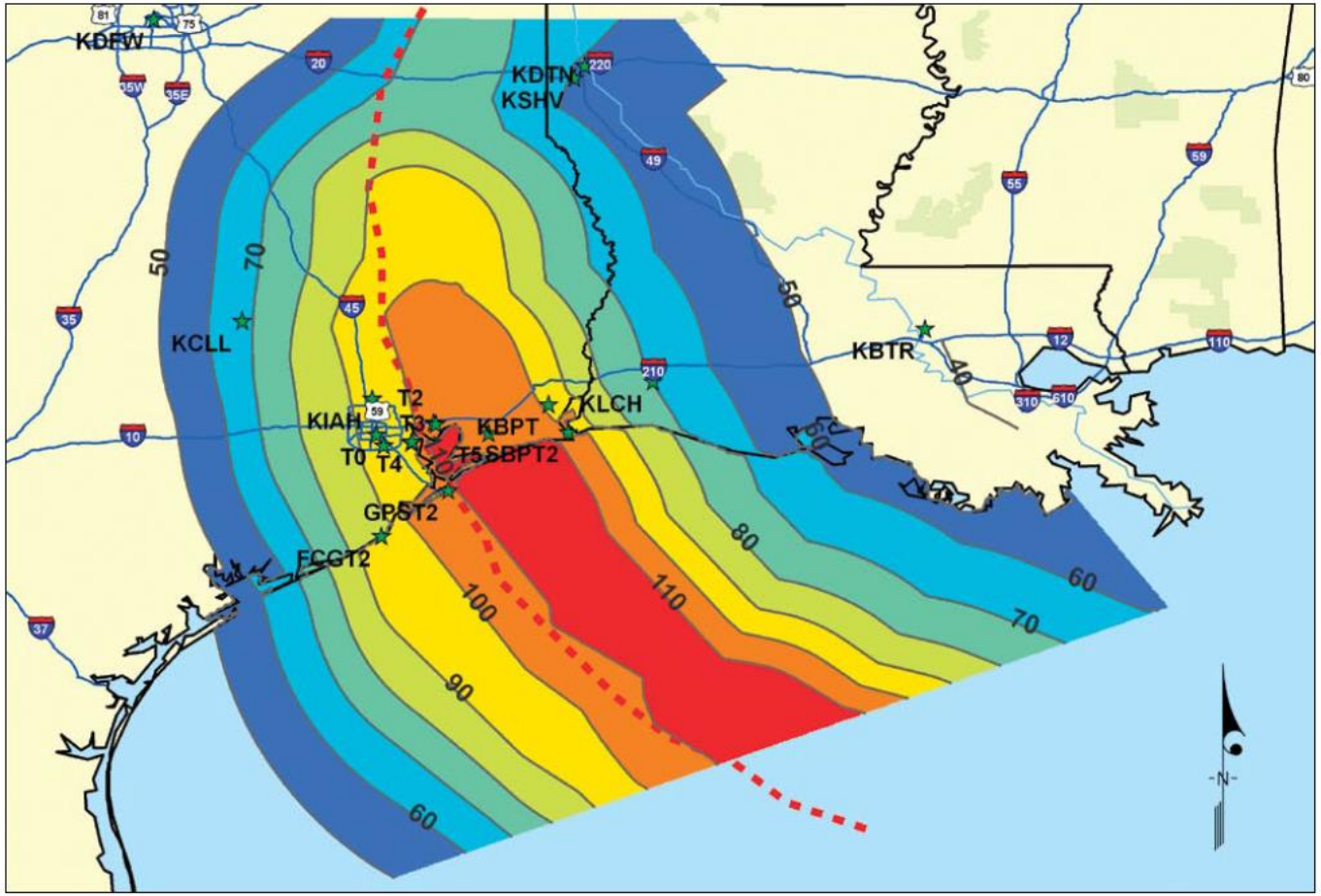
Community Resilience and Modeling

- Social and Economic Systems

- Built environment
  - Building portfolio
  - Transportation system
  - Power network
  - Water/waste water
  - Telecommunications

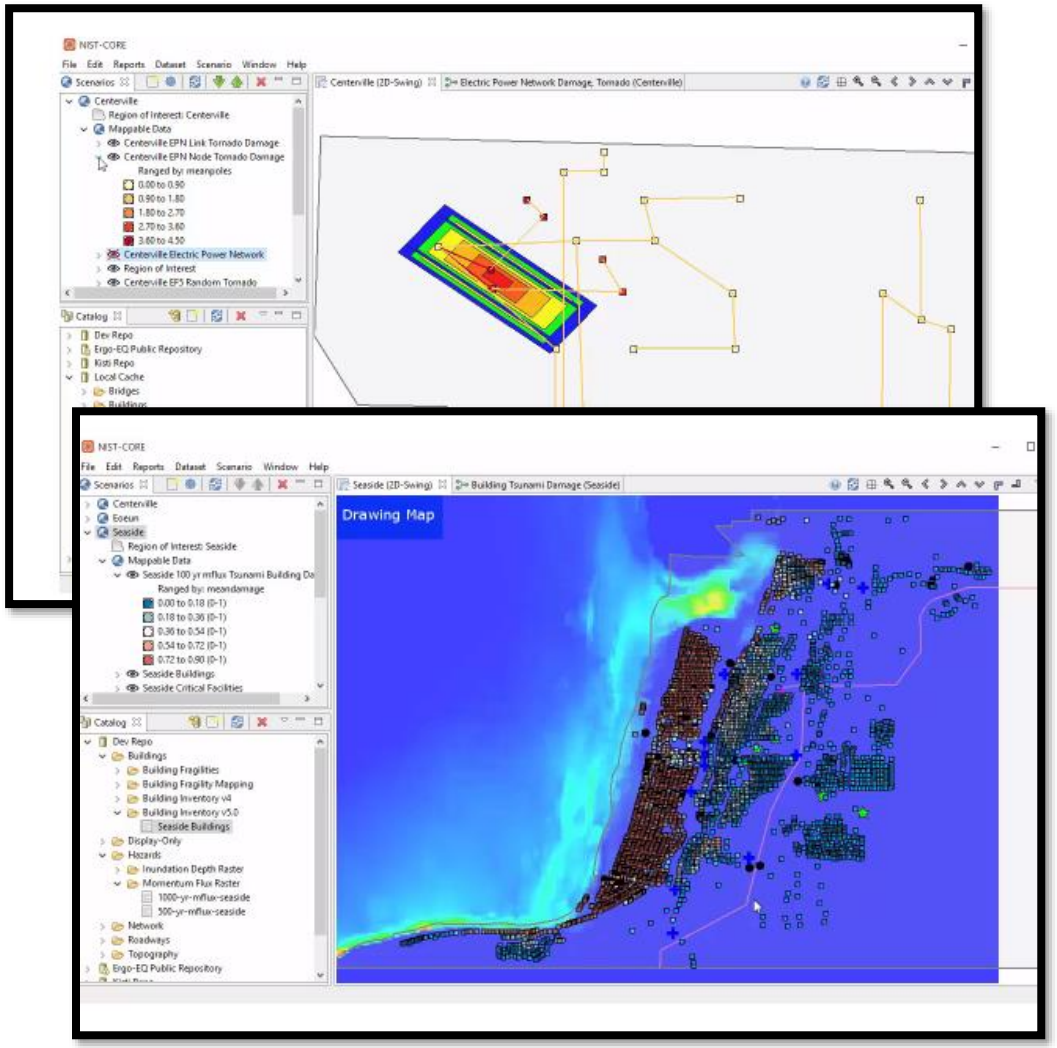


Hurricanes

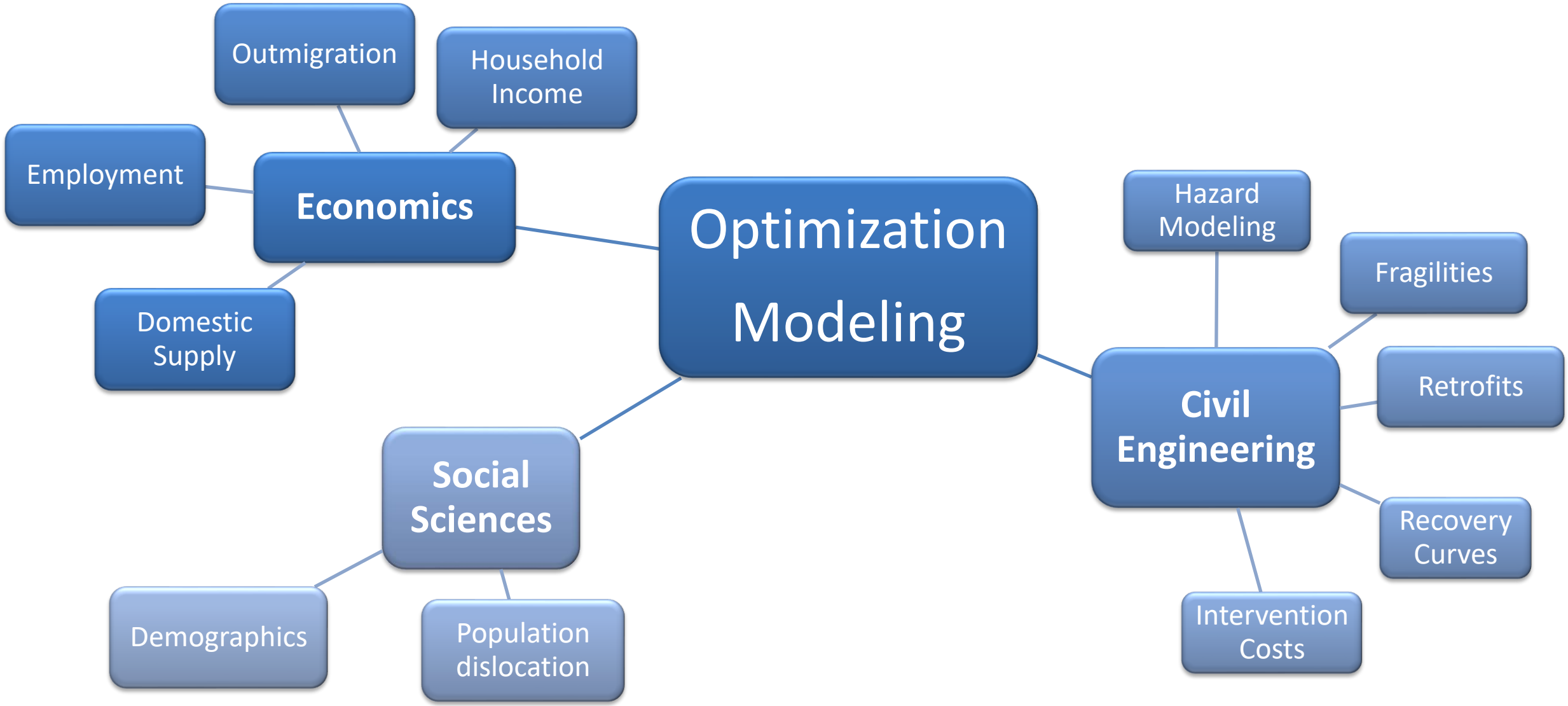

KDFW
KDTN
KSHV
KCLL
KBTR
KIAH
KLCH
KBPT
T2
T3
T0
T4
T5
SBPT2
GPST2
FCGT2
50
60
70
80
90
100
110
N

Interdependent Networked Community Resilience Modeling Environment (IN-CORE)

…optimize community disaster resilience planning and post-disaster recovery strategies using physics-based models of inter-dependent physical systems combined with socio-economic systems.



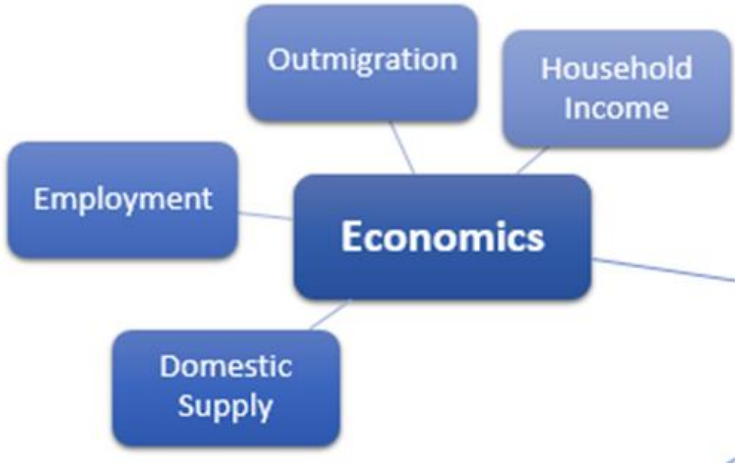
Interdisciplinary methodologies



Computable general equilibrium model

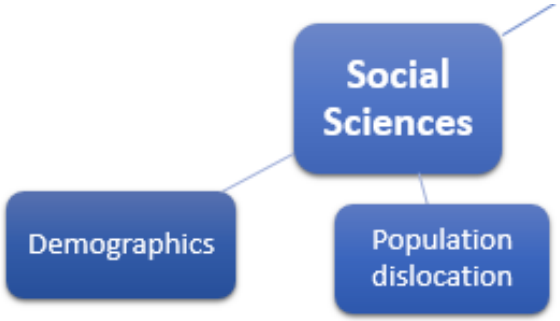
Computable general equilibrium (CGE) widely used economic approach to facilitate counterfactual quantitative policy analysis

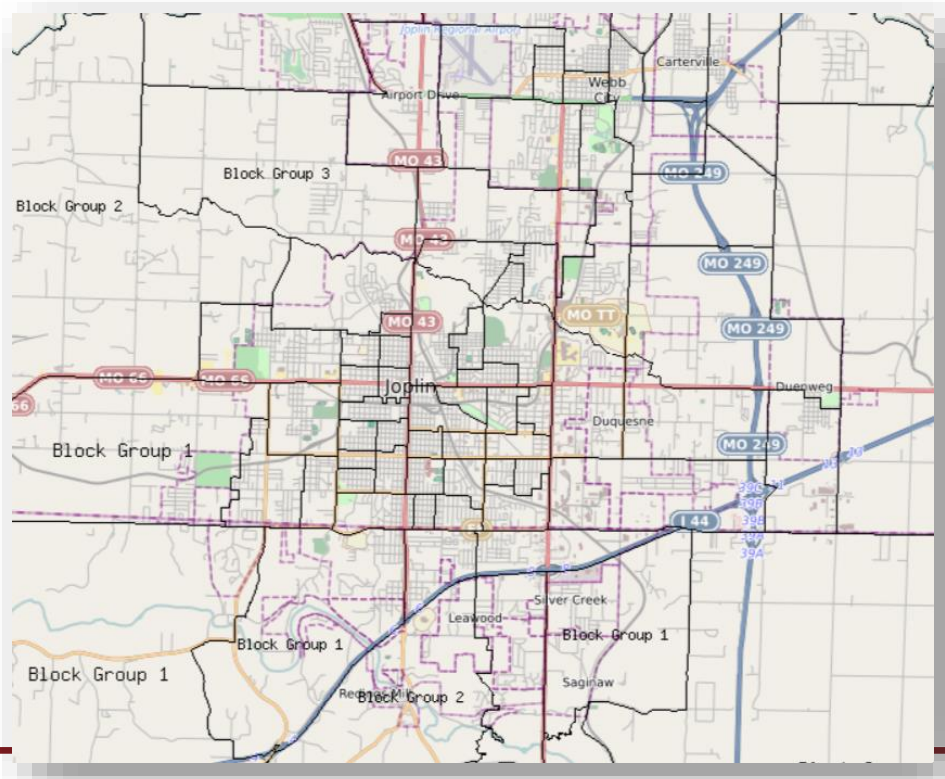
- highly non-linear, “black-box”, numerical method
- calibrated and benchmarked on observed socioeconomic data
- custom built by experts for specific markets
  - elasticities of substitution
  - spatial/non-spatial
  - dynamic/static



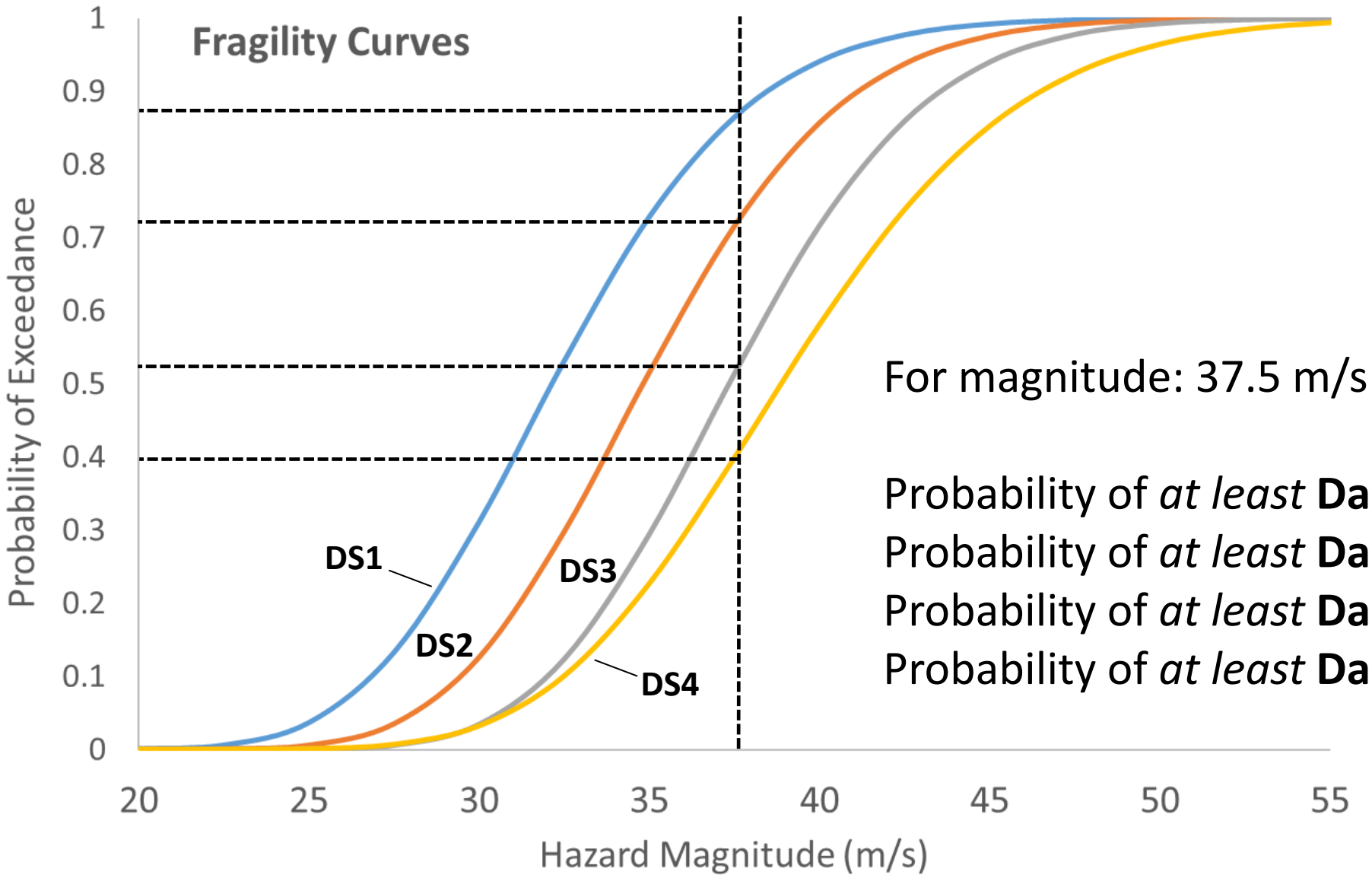
Population Dislocation Model

- Probability model of household dislocating derived empirically from post-disaster observations
- Logistic regression model fit based on:
  i. loss due to building damage
  ii. residential structure type
  iii. census block group demographics





Damage and Structure Specific Fragility Curves



For magnitude: 37.5 m/s

Probability of *at least* **Damage State 1**: 0.865
Probability of *at least* **Damage State 2**: 0.716
Probability of *at least* **Damage State 3**: 0.514
Probability of *at least* **Damage State 4**: 0.399

Decision variables

Pre-event intervention…

x_{ijk} : # of buildings to retrofit

location element

e.g., building id, parcel id, block id, zip code, census tract, PUMA

structure type

e.g., archetype, framing/material

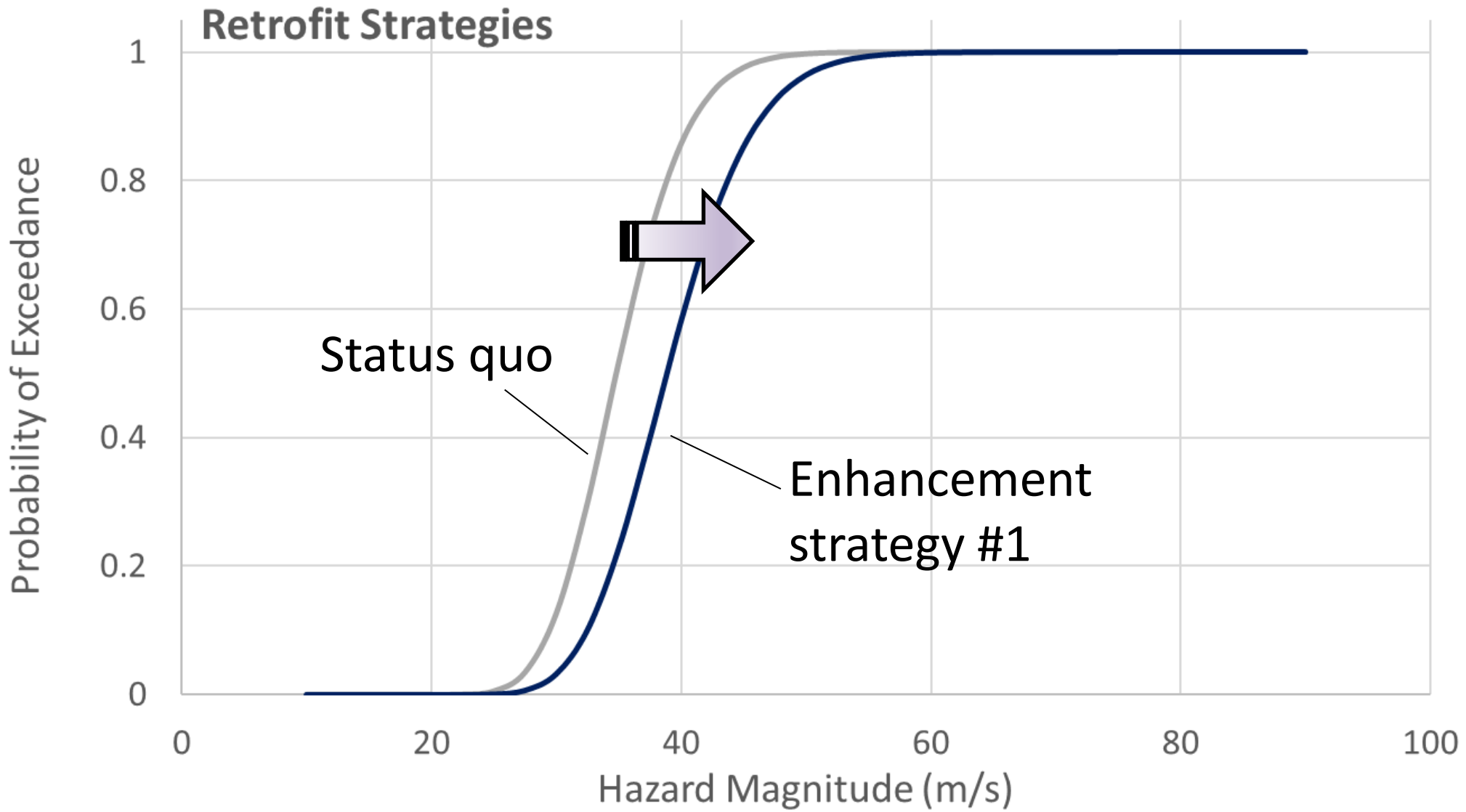
controllable robustness

e.g., building code, enhancement strategy

Optimization Modeling

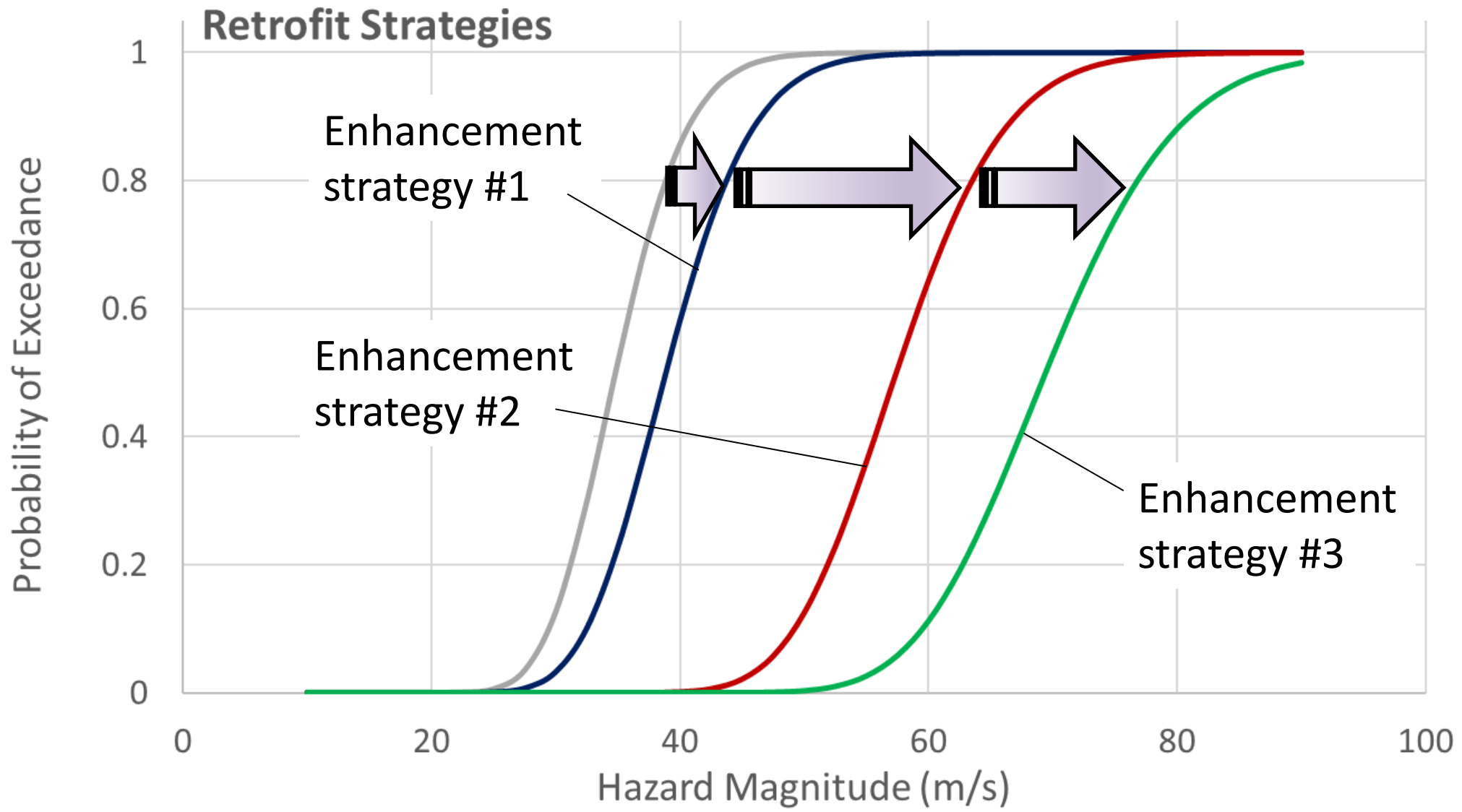
Retrofit Strategies

Damage State 4 Exceedance Probabilities



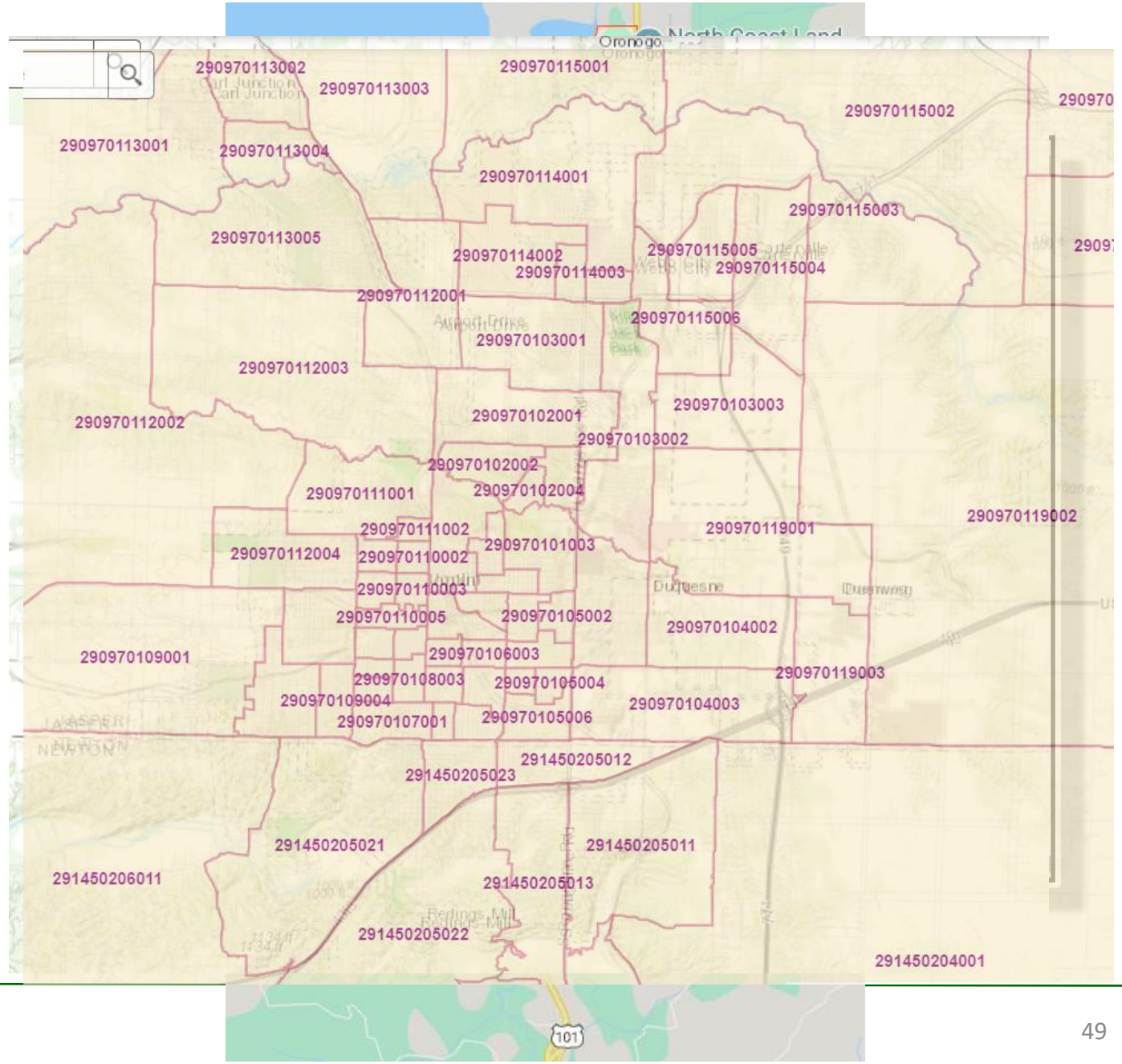
Retrofit Strategies

Damage State 4
Exceedance
Probabilities



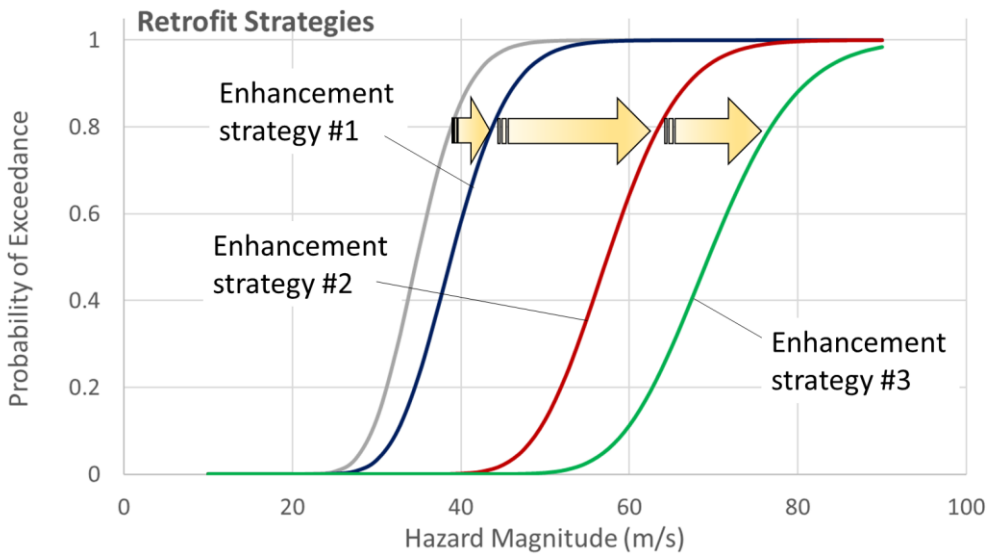
Spatial granularity

- Supports modeling down to individual structure and up through any aggregation-level
- Decision on granularity should be based on:
  - Researcher/stake-holder needs
  - Data availability

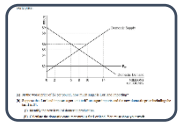


Quantifiable benefit of decisions

x_{ijk} →



Direct economic loss


Total domestic supply

Population dislocation

Employment

Post-event, future building functionality

Availability of services

HH income

Objectives may be competing for the limited resources, i.e., we need to consider tradeoffs

Objective: Minimize Population Dislocation

$$\min d_i = \sum_{j \in S} \sum_{k \in K} x_{ijk} \times \left\lceil \psi_{ijk} - p^{threshold} \right\rceil \times H_{ij}$$

where,

$$\psi_{ijk} = \frac{1}{1 + e^{-(\alpha_0 + \alpha_1 \frac{l_{ijk}^{-c}}{\bar{M}_{ij}} + \alpha_2 DS_j + \alpha_3 B_i + \alpha_4 H_i)}}, \forall i \in \mathcal{Z}, \forall j \in \mathcal{S}, \forall k \in \mathcal{K}$$

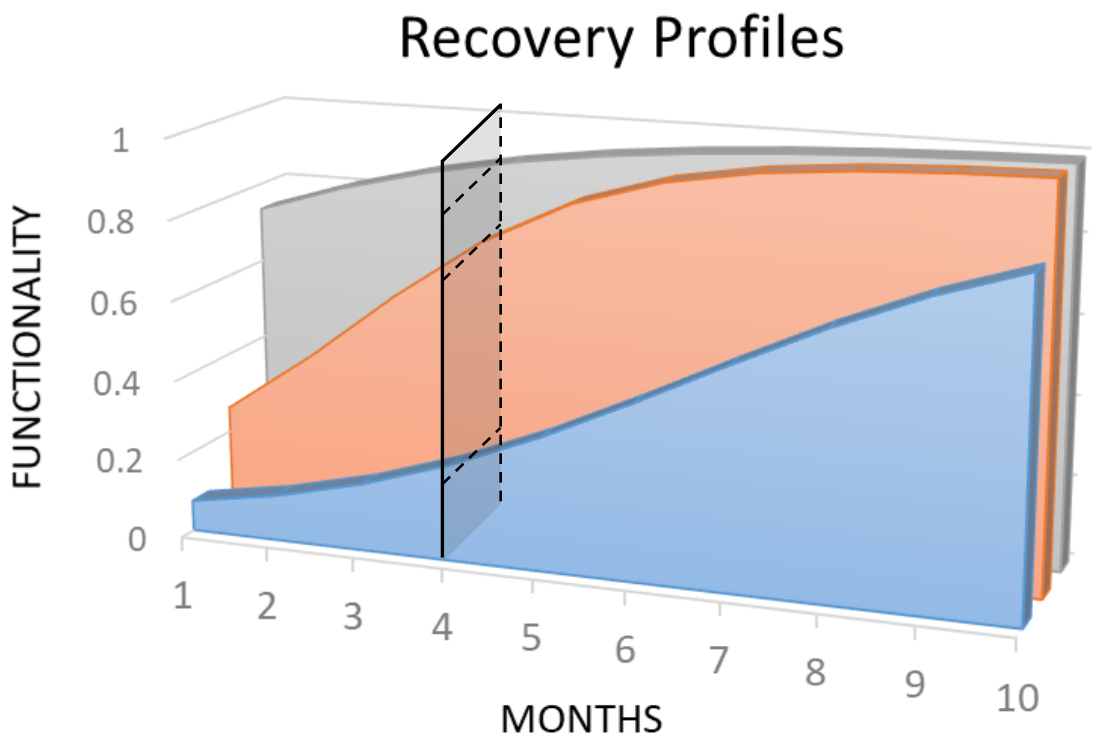
Y.-S. Lin, W. Peacock, J. Lu, Y. Zhang, Household dislocation algorithm 3: A logistic regression approach, Report, Hazard Reduction and Recovery Center, Texas A&M University (2008).

Objective: Maximize Building Functionality

$$\max \frac{\sum_{i \in \mathcal{Z}} \sum_{j \in \mathcal{S}} \sum_{k \in \mathcal{K}} Q^t_{ijk} x_{ijk}}{\sum_{i \in \mathcal{Z}} \sum_{j \in \mathcal{S}} \sum_{k \in \mathcal{K}} b_{ijk}}$$

where,

$$Q^t_{ijk} = \frac{1}{1 + (\frac{1}{Q^0_{ijk}} - 1)e^{-r_{ijk}t}}$$



Koliou, M., & van de Lindt, J. W. (2019). Development of Building Restoration Functions for use in Community Recovery Planning to Tornadoes. Natural Hazards Review (in-press)

Mathematical framework

Minimize $\sum_{i\in\mathcal{Z}}\sum_{j\in\mathcal{S}}\sum_{k\in\mathcal{K}} p_i(l_{ijk}x_{ijk})$ → **Direct economic Loss**

Minimize $\sum_{i\in\mathcal{Z}}\sum_{j\in\mathcal{S}}\sum_{k\in\mathcal{K}} p_i(d_{ijk}x_{ijk})$ → **Population dislocation**

Maximize $\sum_{i\in\mathcal{Z}}\sum_{j\in\mathcal{S}}\sum_{k\in\mathcal{K}} p_i(\hat{Q}^t_{ijk}\, x_{ijk})$ → **Estimated building functionality at time *t***

Subject to $\sum_{i\in\mathcal{Z}}\sum_{j\in\mathcal{S}}\sum_{k\in\mathcal{K}}\sum_{k'\in\mathcal{K}} SC_{ijkk'}\, y_{ijkk'} \le B$ → **Budget constraints**

$$\sum_{k\in\mathcal{K}} x_{ijk} = \sum_{k\in\mathcal{K}} b_{ijk} \quad \forall i\in\mathcal{Z}, j\in\mathcal{S}$$

→ **Building quantity and location consistency**

$$x_{ijk} = \sum_{k':(k',k)\in\mathcal{L}} y_{ijk'k} + b_{ijk} - \sum_{k':(k,k')\in\mathcal{L}} y_{ijkk'} \quad \forall i\in\mathcal{Z},\ j\in\mathcal{S}, k,k'\in\mathcal{K}$$

$$x_{ijk} \ge 0\ \forall i\in\mathcal{Z}, j\in\mathcal{S}, k\in\mathcal{K}$$

$$y_{ijkk'} \ge 0\ \forall i\in\mathcal{Z}, j\in\mathcal{S}, k,k'\in\mathcal{K}$$

→ **Auxiliary constraints, non-negativity**

Hazard agnostic, optimization framework integrating economic, social, and physical system objectives

Flooding

Earthquake

Tsunami

Tornado

Inputs

l_{ijk} — Monetary loss information by location, archetype, enhancement levels

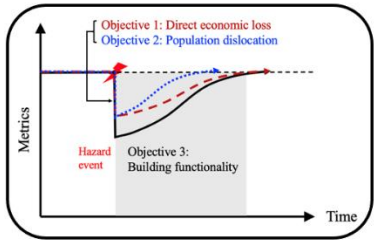
d_{ijk} — Population information by location, archetype, enhancement levels

$\hat{Q}^t_{ijk}$ — Building information by location, archetype, enhancement levels

$SC_{ijkk'}$ — Enhancement cost by location, archetype, enhancement levels

p_i — Hazard probability parameter

Optimization model



$$\text{Minimize} \quad \sum_{i\in\mathcal{Z}}\sum_{j\in\mathcal{S}}\sum_{k\in\mathcal{K}} p_i(l_{ijk}x_{ijk})$$

$$\text{Minimize} \quad \sum_{i\in\mathcal{Z}}\sum_{j\in\mathcal{S}}\sum_{k\in\mathcal{K}} p_i(d_{ijk}x_{ijk})$$

$$\text{Maximize} \quad \sum_{i\in\mathcal{Z}}\sum_{j\in\mathcal{S}}\sum_{k\in\mathcal{K}} p_i(\hat{Q}^t_{ijk}\, x_{ijk})$$

$$\text{Subject to} \quad \sum_{i\in\mathcal{Z}}\sum_{j\in\mathcal{S}}\sum_{k\in\mathcal{K}}\sum_{k'\in\mathcal{K}} SC_{ijkk'}\, y_{ijkk'} \le B$$

$$x_{ijk} = \sum_{k':(k',k)\in\mathcal{L}} y_{ijk'k} + b_{ijk} - \sum_{k':(k,k')\in\mathcal{L}} y_{ijkk'}$$

$$\forall i\in\mathcal{Z},\ j\in\mathcal{S},\ k,k'\in\mathcal{K}$$

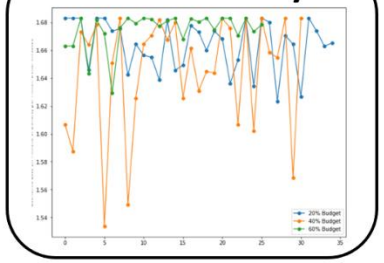
$$\sum_{k\in\mathcal{K}} x_{ijk} = \sum_{k\in\mathcal{K}} b_{ijk} \quad \forall i\in\mathcal{Z}, j\in\mathcal{S}$$

$$x_{ijk} \ge 0\ \forall i\in\mathcal{Z}, j\in\mathcal{S}, k\in\mathcal{K}$$

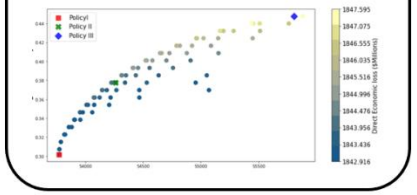
$$y_{ijkk'} \ge 0\ \forall i\in\mathcal{Z}, j\in\mathcal{S}, k,k'\in\mathcal{K}$$

Output

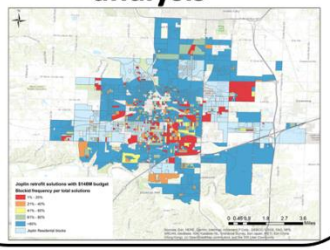
Investment analysis



Tradeoff analysis



Spatial and Temporal analysis



Centerville Optimization

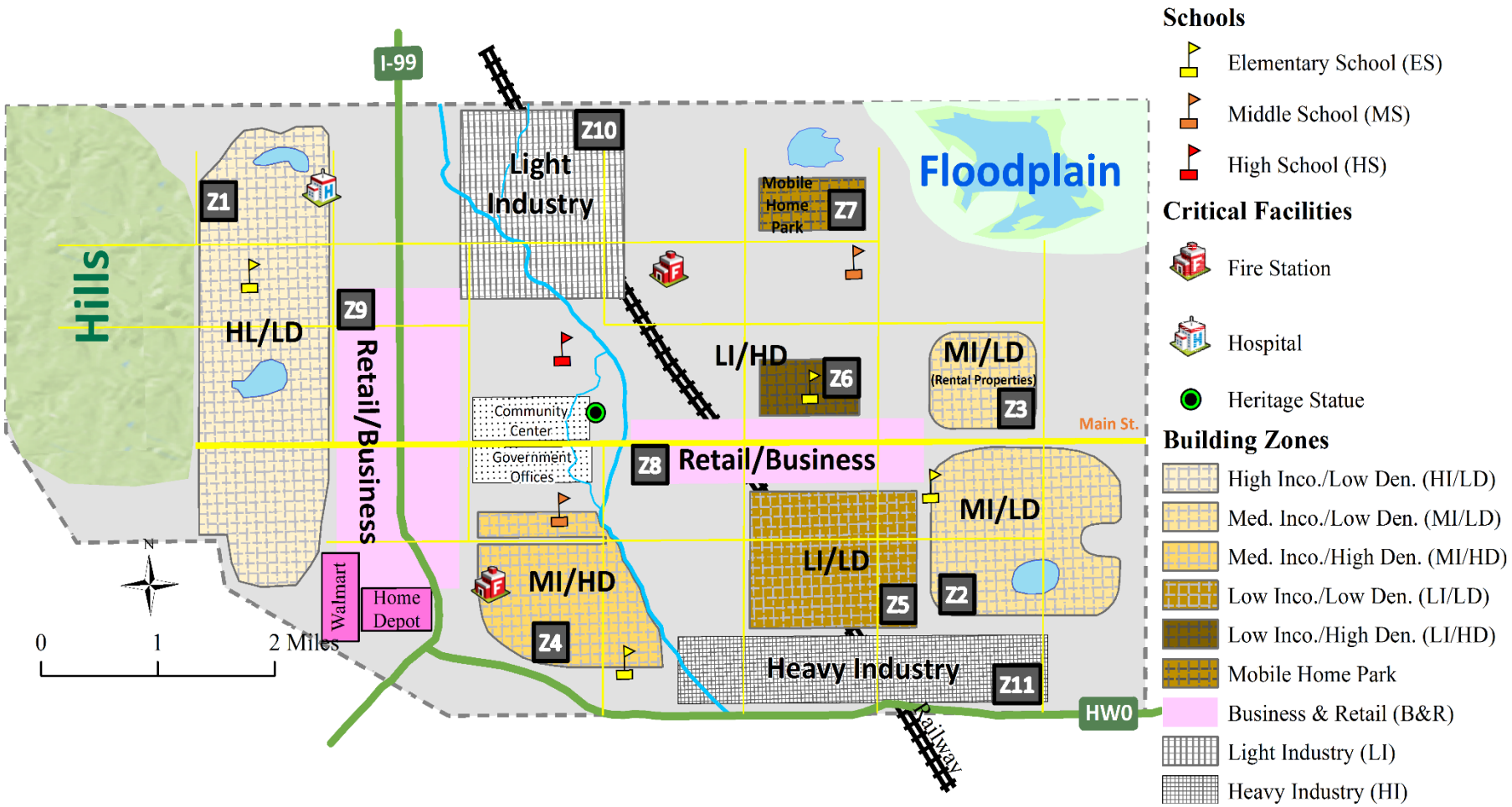
Weili Zhang
Charles D. Nicholson, Ph.D.
Naiyu Wang, Ph.D.
University of Oklahoma

Collaborators: Harvey Cutler, Ph.D., Walter Peacock, Ph.D., Sammy Zahran, Ph.D., Nathanael Rosenheim, Ph.D., Peihui Lin

Nov 4, 2015

COE Biannual Meeting, NIST

Centerville Building Zones



Mitigation Resource Allocation

Allocate limited resources to retrofit building types in each zone, to specific code levels w.r.t. competing objectives:

- minimize economic loss (i.e., structure, non-structure, and content loss, provided by the Building team)
- minimize total dislocation (OLS model provided by the Social Science team)

Requirement:

- overall disparity in the dislocation rates by socio-economic status does not increase

MRA Baseline

Total direct economic loss: **$856M**

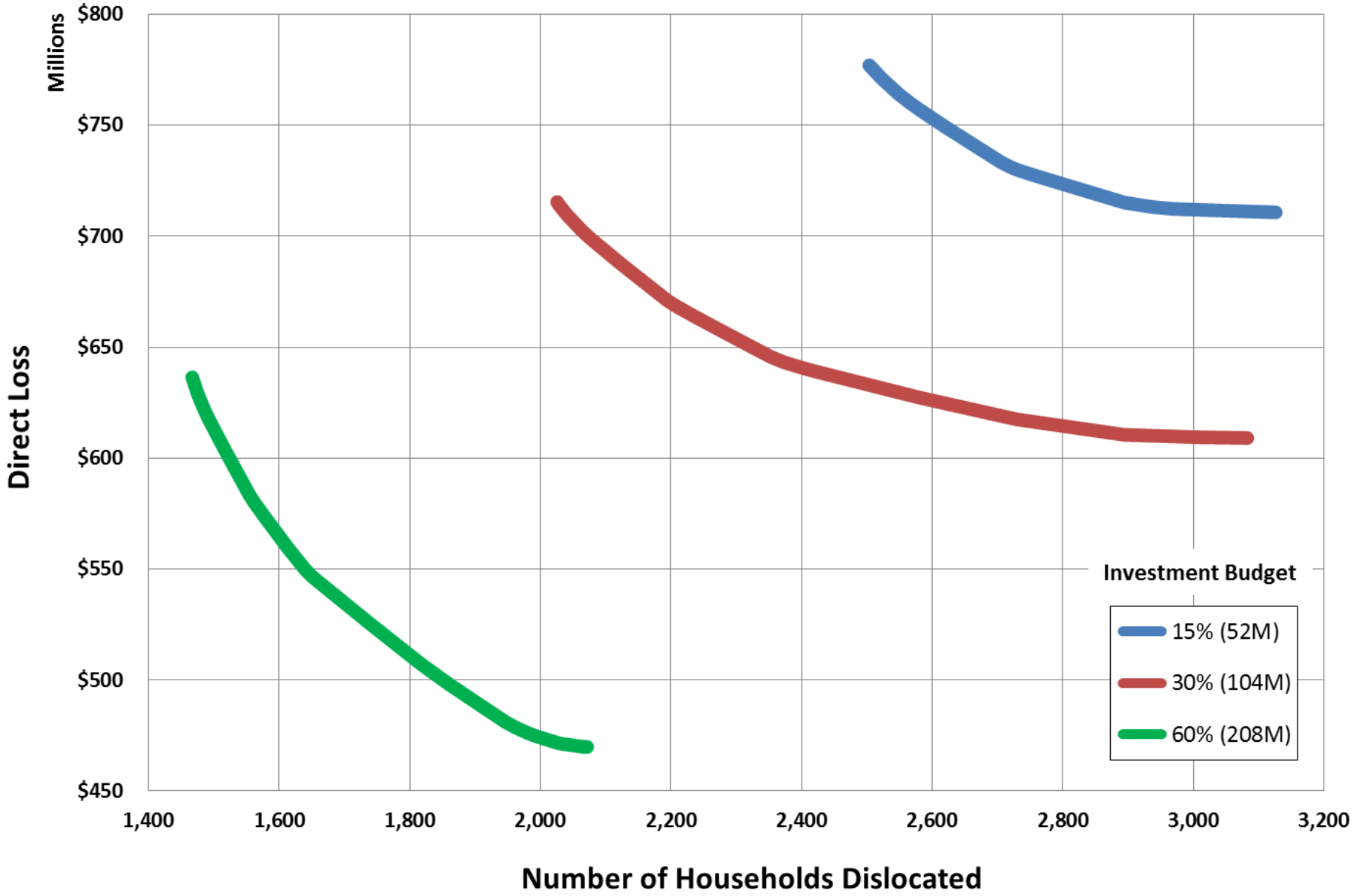
Total dislocation: **3,203 HH**

Evaluation optimization results under three budget levels:

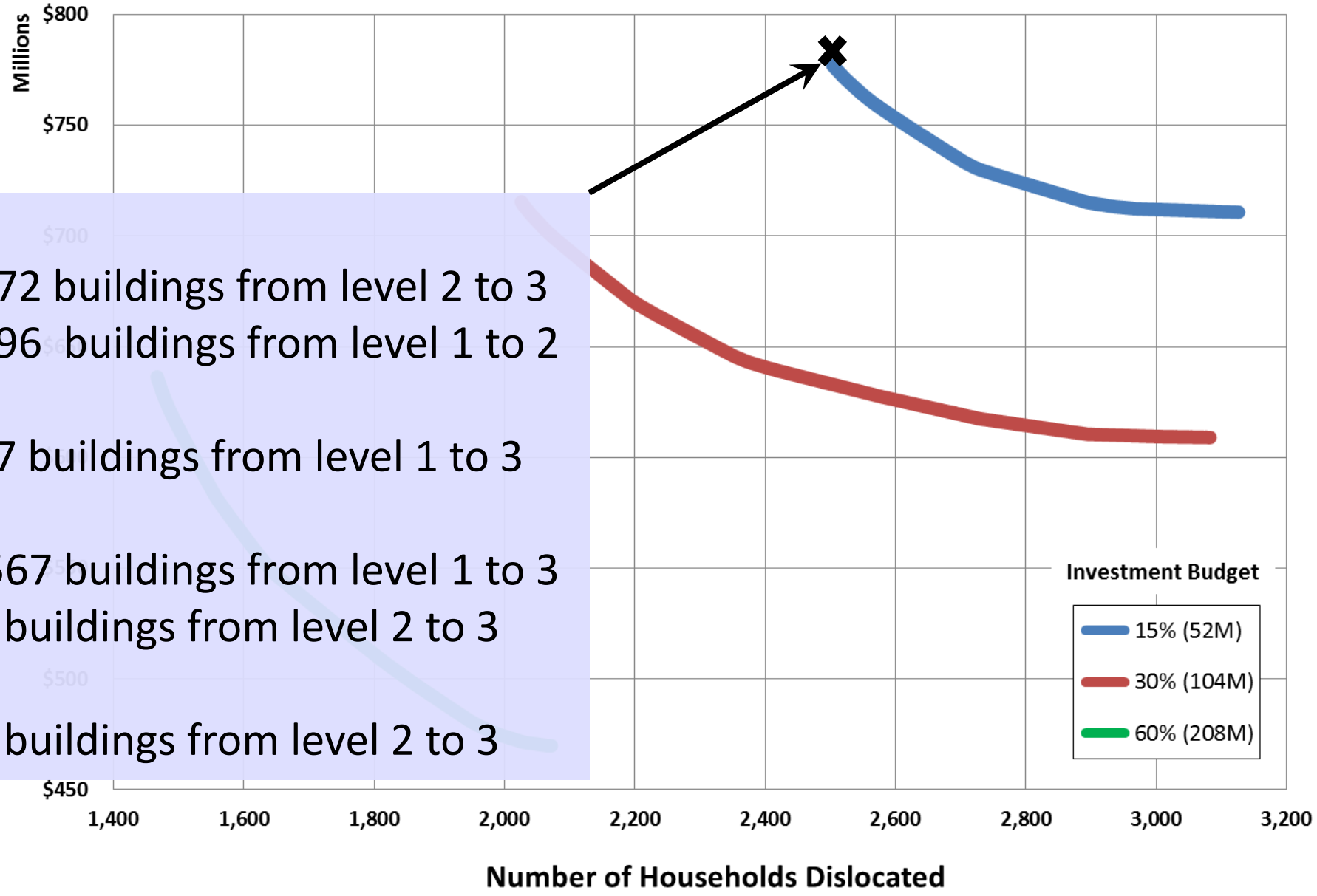
- **Low**: 15% of ideal budget
- **Medium**: 30% of ideal budget
- **High**: 60% of the ideal budget

Ideal budget ($346M) is sufficient to improve all buildings to highest code level.

Pareto Optimal Frontier: Capital Stock Loss and Dislocation



Pareto Optimal Frontier: Capital Stock Loss and Dislocation



Zone1:

W2: 1,072 buildings from level 2 to 3

W4: 2,196 buildings from level 1 to 2

Zone 2:

W1: 767 buildings from level 1 to 3

Zone 4:

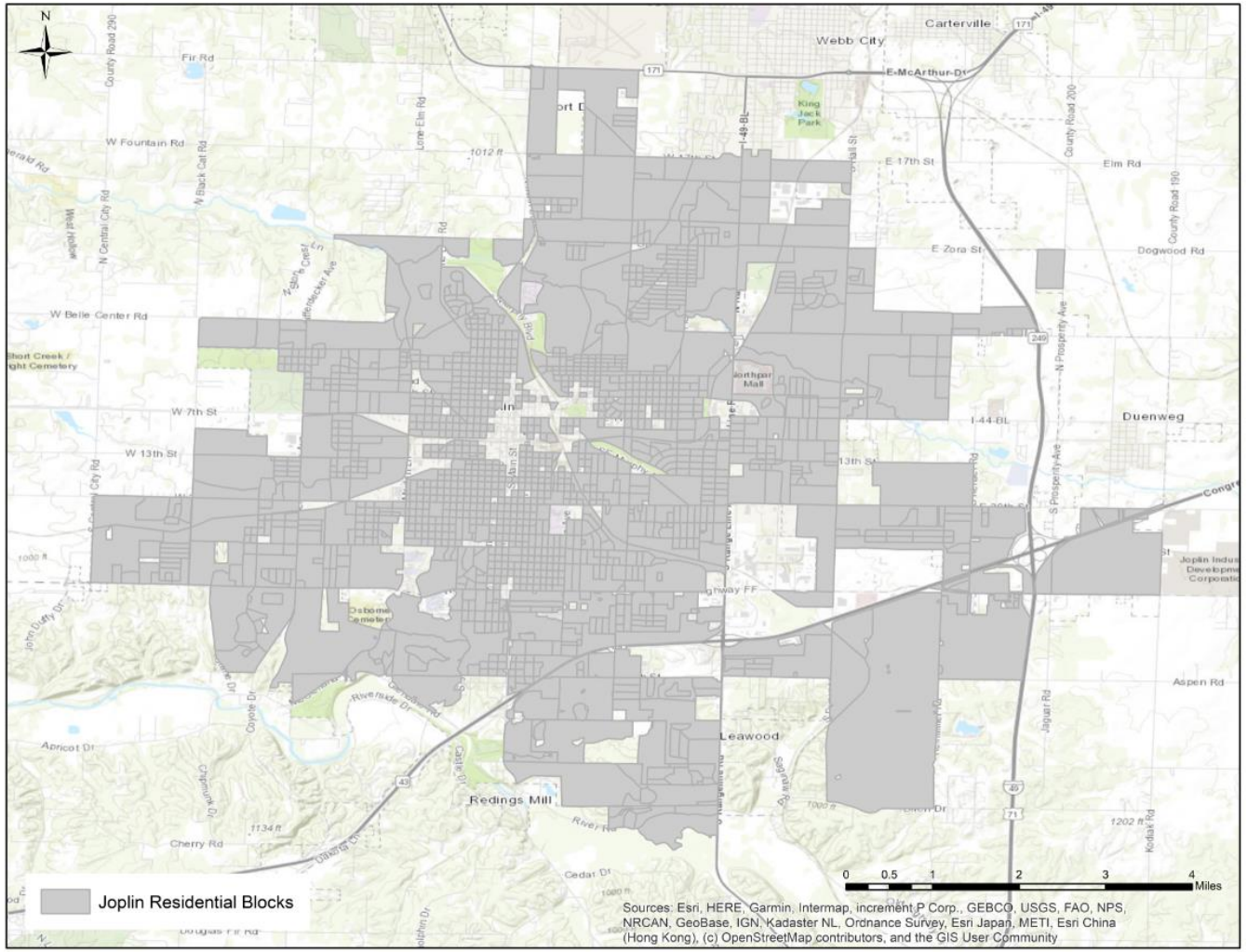
W1: 2,567 buildings from level 1 to 3

W5: 25 buildings from level 2 to 3

Zone 6:

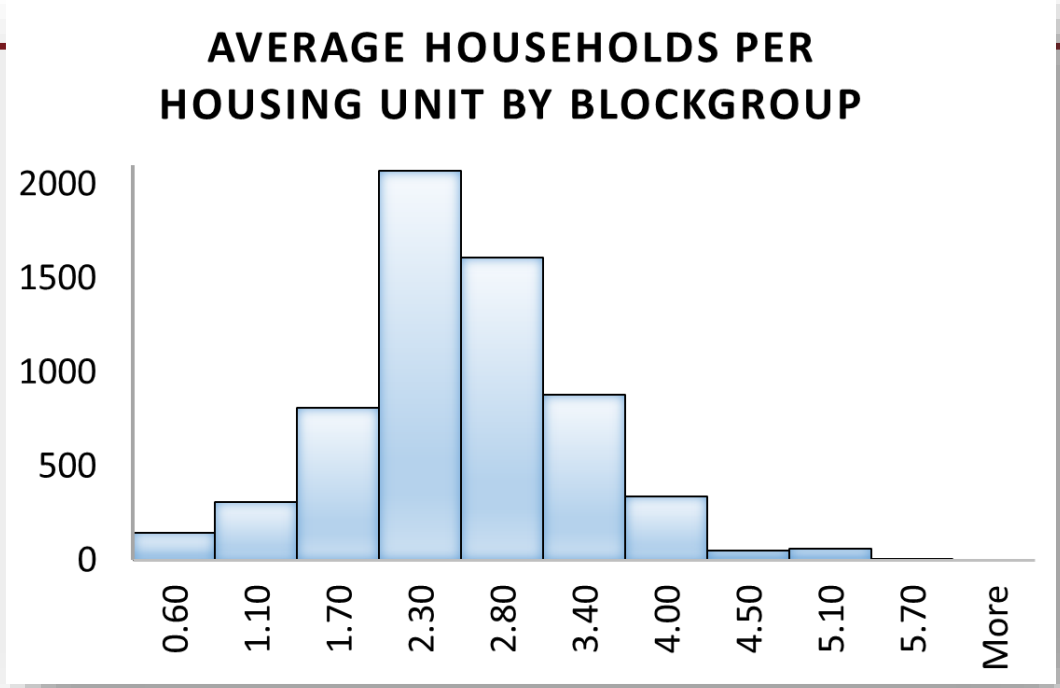
W5: 59 buildings from level 2 to 3

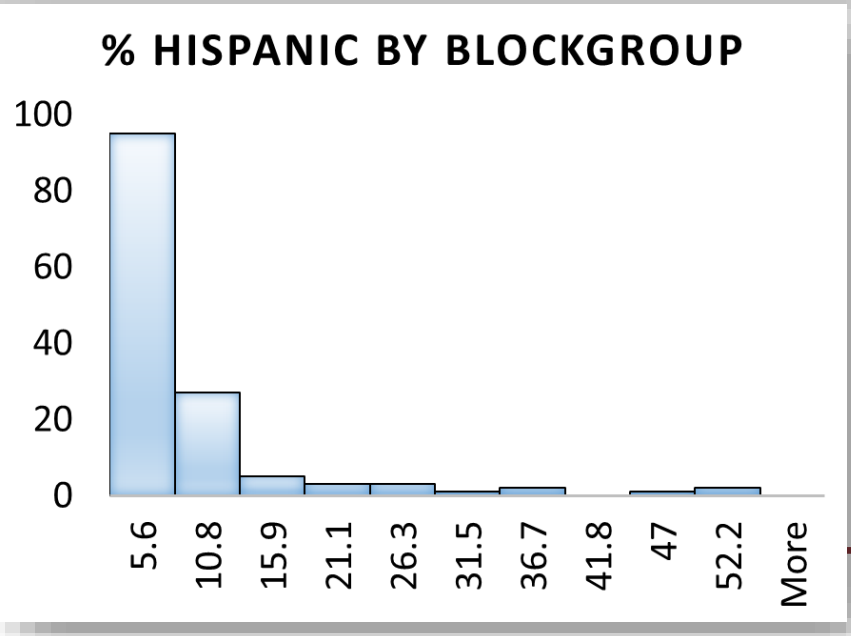
Joplin, MO Tornado Decision Support Case Study*



Joplin, MO

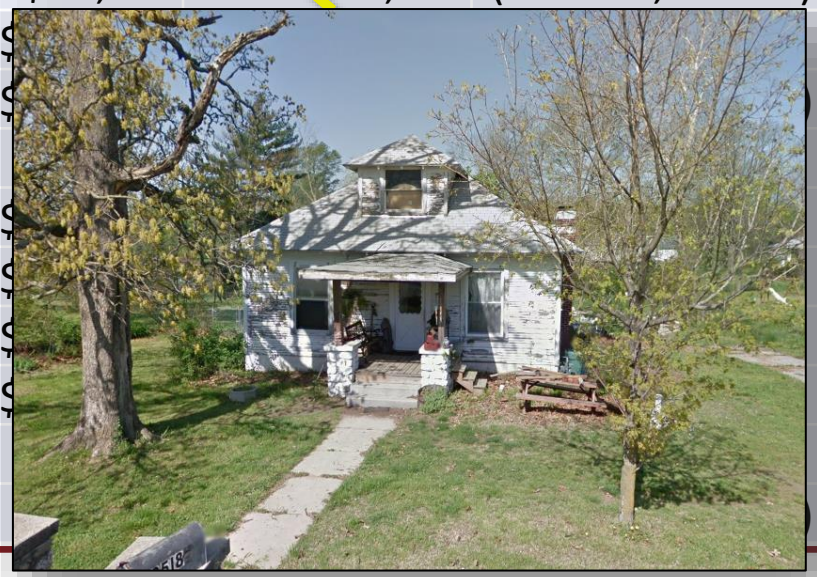
- Population: 52,000+ people
- 139 census blockgroups
  - 1545 disparate sub-blocks
    → *location component of DV*
- ~28,000 buildings
  - 16 building archetypes
    → *structure component of DV*
- Available race/ethnicity data by census blockgroup





Building Information

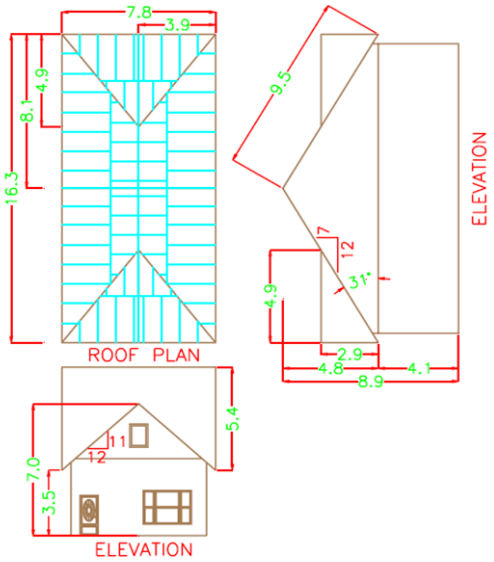
Structure ID	Block ID	Archetype	Year Built	Sqr footage	Appraised Building Value	Appraised Land Value	Location
S0000007	290970102003000	1	1947	2,283	$51,320	$14,630	(-94.477,37.11)
S0000008	290970102003000	1	1948	2,461	$63,110	$11,880	(-94.478,37.11)
S0000014	290970102003000	1	1900	1,377	$33,330	$16,130	(-94.478,37.112)
S0000016	290970102003000	1	1979	1,826	$89,690	$12,000	(-94.479,37.109)
S0000201	290970102002002	19	1996	13,258	$407,880	$20,290	(-94.49,37.113)
S0000216	290970102003000	1	1930	885	$37,120	$24,020	(-94.48,37.112)
S0000217	290970102003000	1	1950	1,532	$52,180	$17,870	(-94.479,37.112)
S0000219	290970102003000	19	1999	11,123	$733,370	$324,800	(-94.477,37.112)
S0000712	290970101003000	6	1984	3,424	$73,760	$72,600	(-94.485,37.099)
				15,644			
				20,407			
				1,753			
				1,865			
				5,271			
				1,588			
				3,623			
				1,739			
				739			



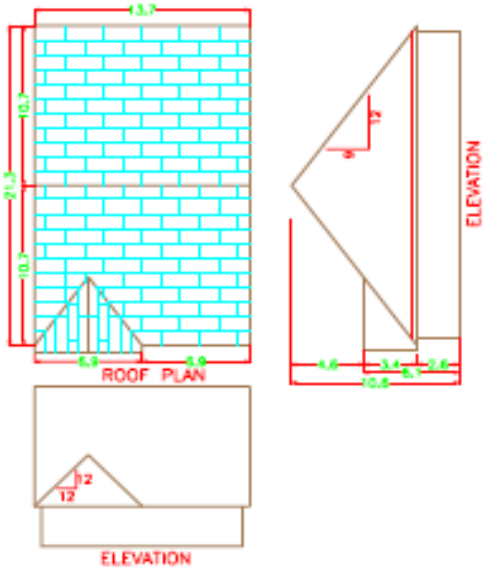
Retrofit Strategies

Retrofit strategies	*Roof covering*		*Roof sheathing nailing pattern*		*Roof-to-wall connection type*	
	Asphalt shingles	Clay tiles	8d C6/12	8d C6/6	Two 16d toe nails	Two H2.5 clips
1	**X**	**-**	**X**	**-**	**X**	**-**
2	**X**	**-**	**-**	**X**	**-**	**X**
3	**-**	**X**	**-**	**X**	**-**	**X**

- 3 tornado-specific mitigation strategies
- Estimated costs
- Fragility curves available
- ***Residential homes only…***



Structure Type 1

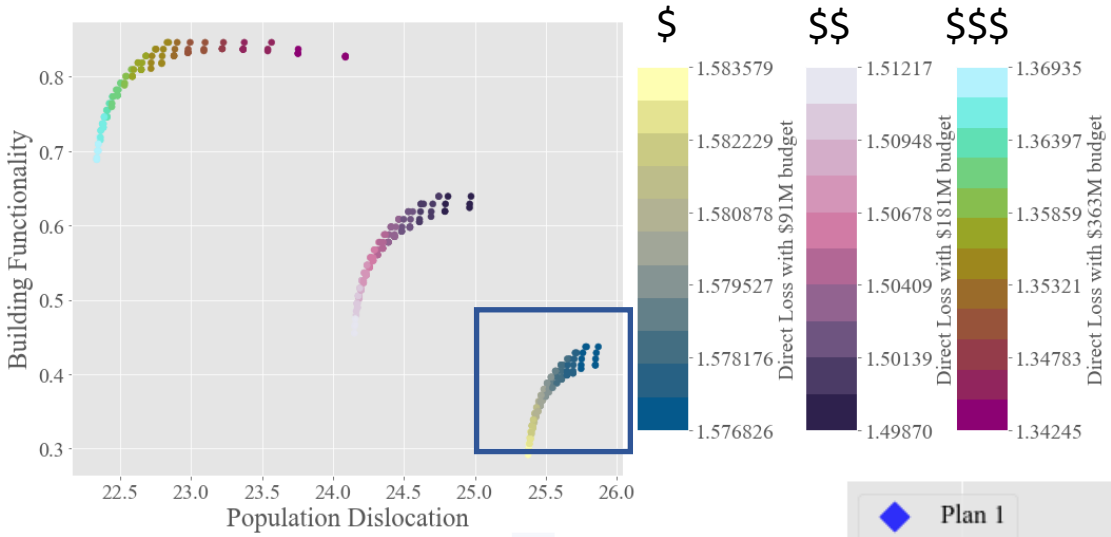


Structure Type 5

Masoomi, H., Ameri, M. R., & van de Lindt, J. W. (2018). Wind performance enhancement strategies for residential wood-frame buildings. Journal of Performance of Constructed Facilities, 32(3), 04018024.

Amini, M. O., & van de Lindt, J. W. (2013). Quantitative insight into rational tornado design wind speeds for residential wood-frame structures using fragility approach. Journal of Structural Engineering, 140(7), 04014033.

Tornado Decision-Support Analysis

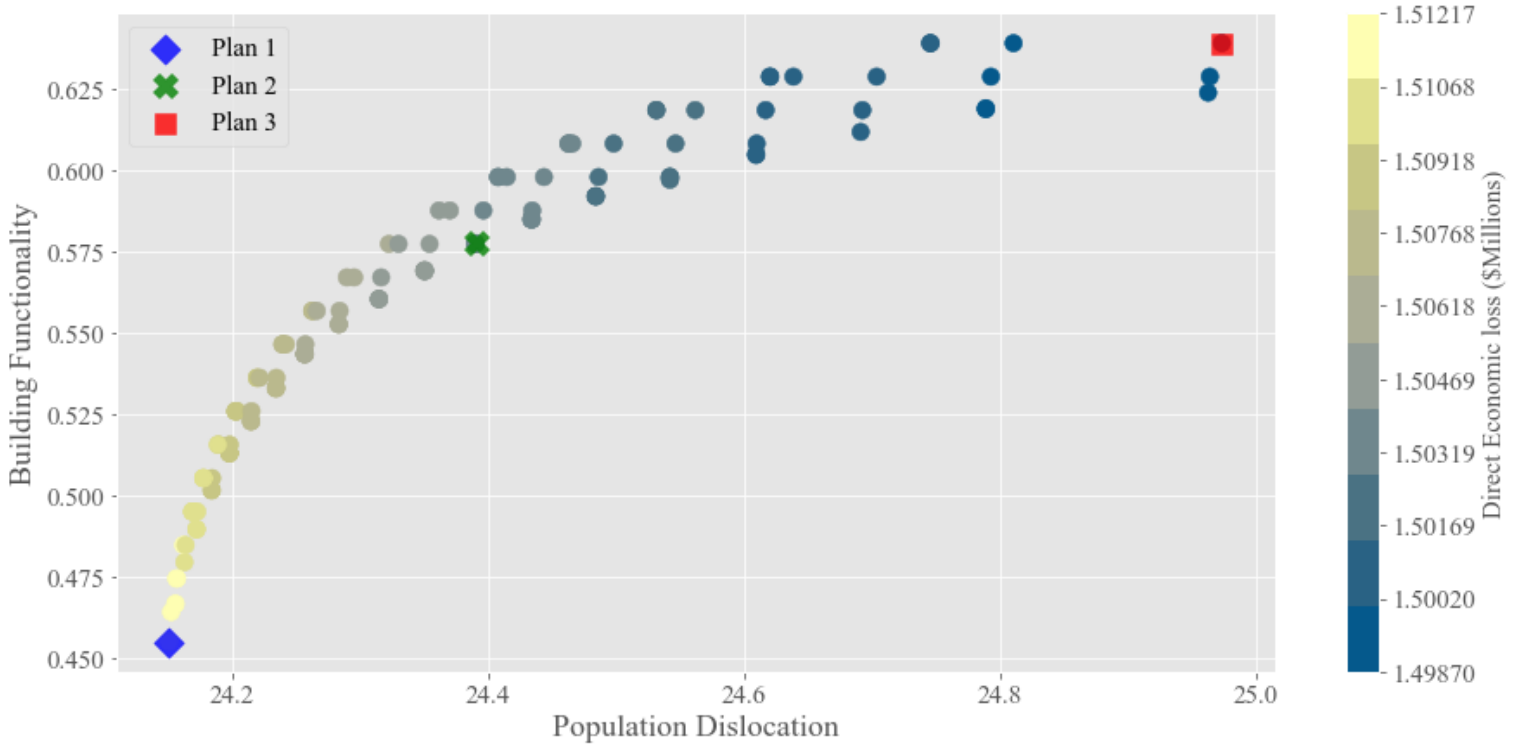


Objectives:

minimize direct economic loss
minimize population dislocation
maximize future building functionality

Benefits of tradeoff analysis

- Provide detail analysis on selected retrofit plans
- Allocate money according to the best interest of the community



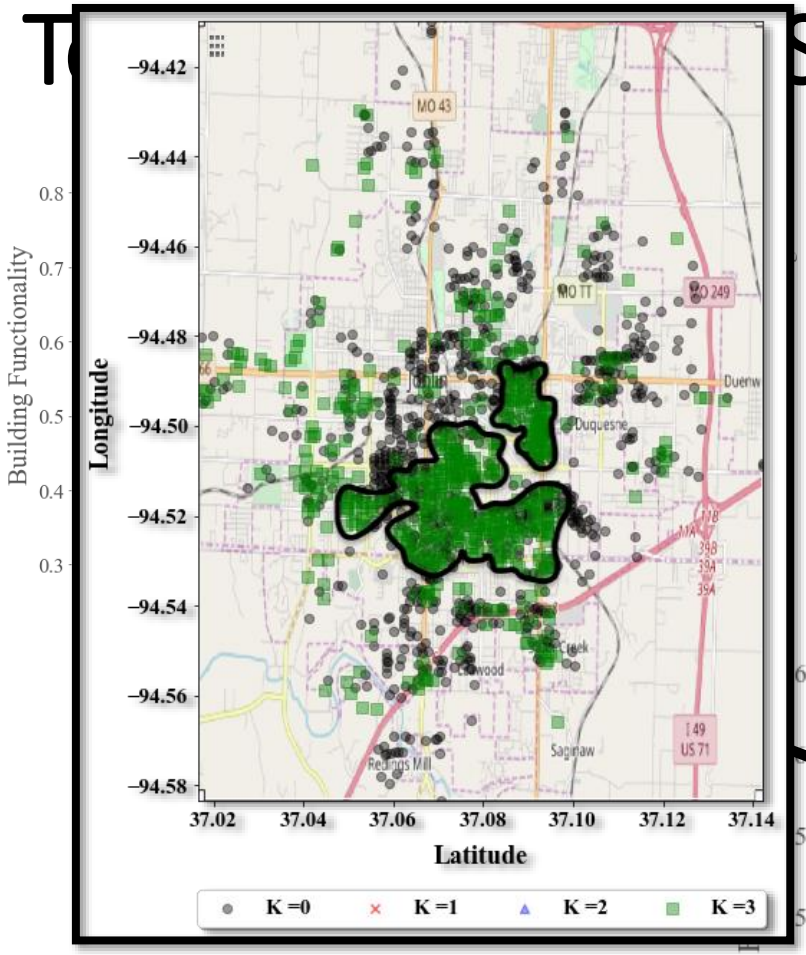
T Support Analysis

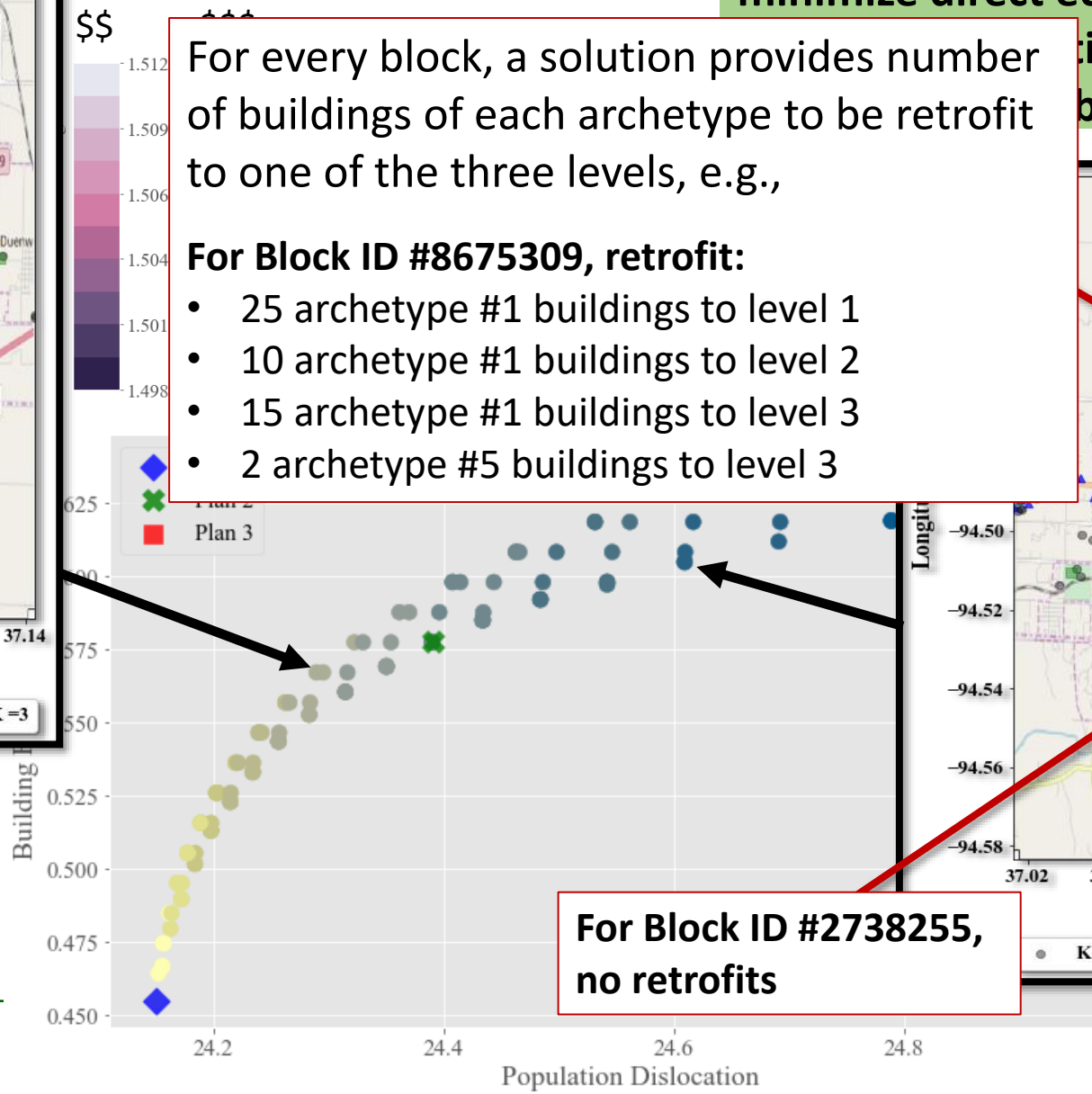
minimize direct economic loss
tion dislocation
building functionality

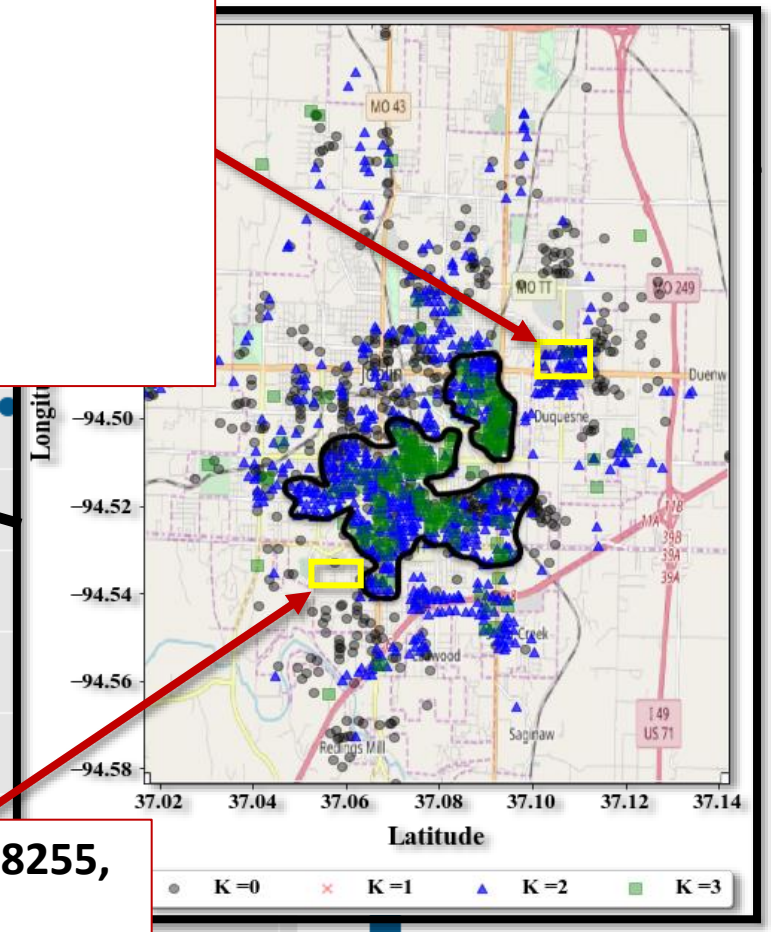
For every block, a solution provides number of buildings of each archetype to be retrofit to one of the three levels, e.g.,

For Block ID #8675309, retrofit:

- 25 archetype #1 buildings to level 1
- 10 archetype #1 buildings to level 2
- 15 archetype #1 buildings to level 3
- 2 archetype #5 buildings to level 3



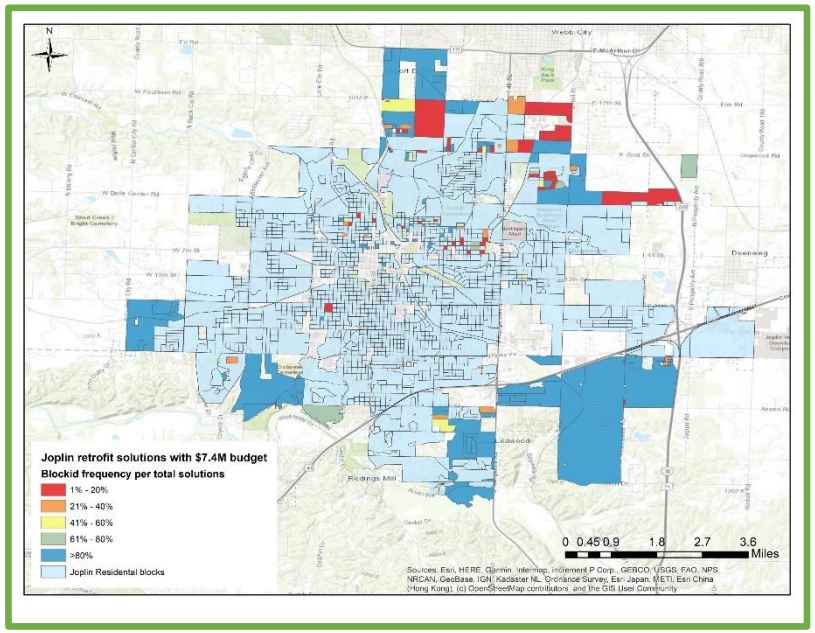




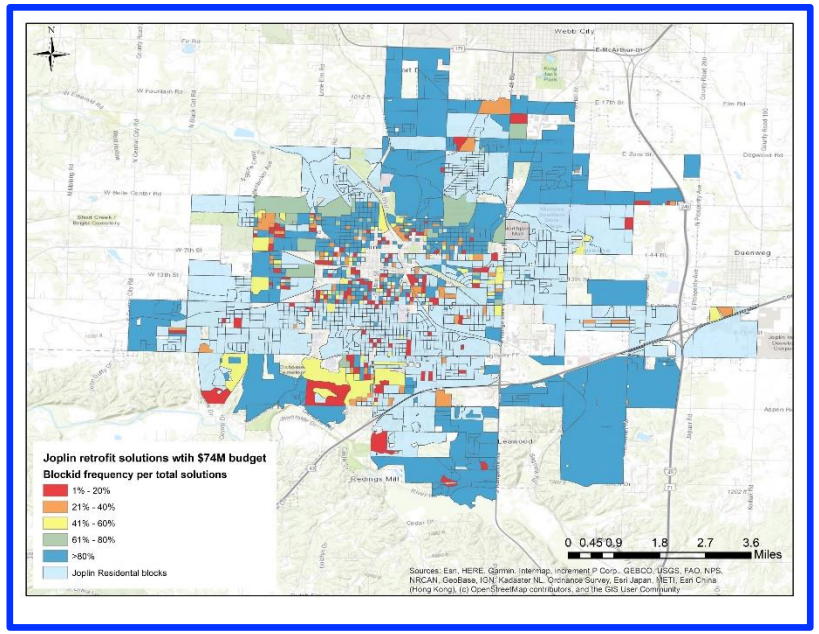
For Block ID #2738255, no retrofits

Priority Analysis

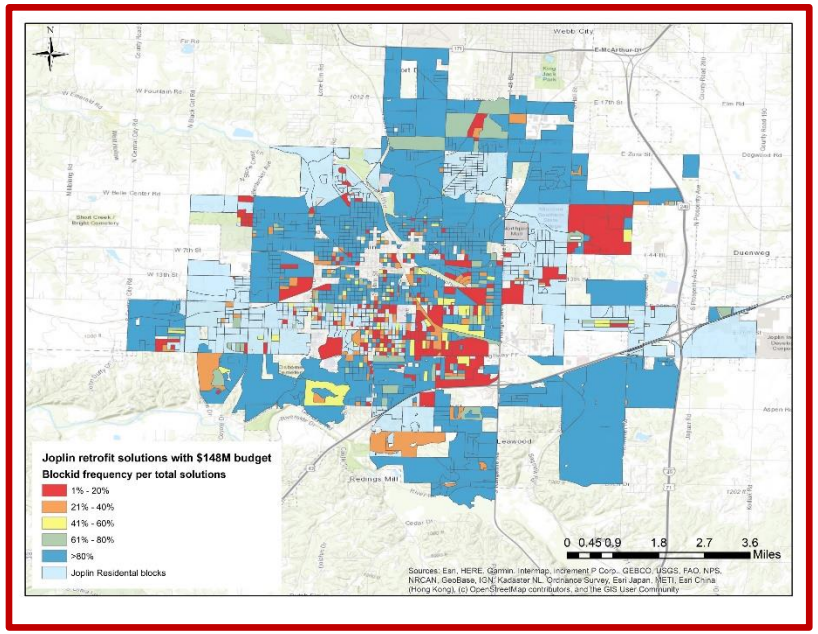
Budget $7.4M



Budget $74M

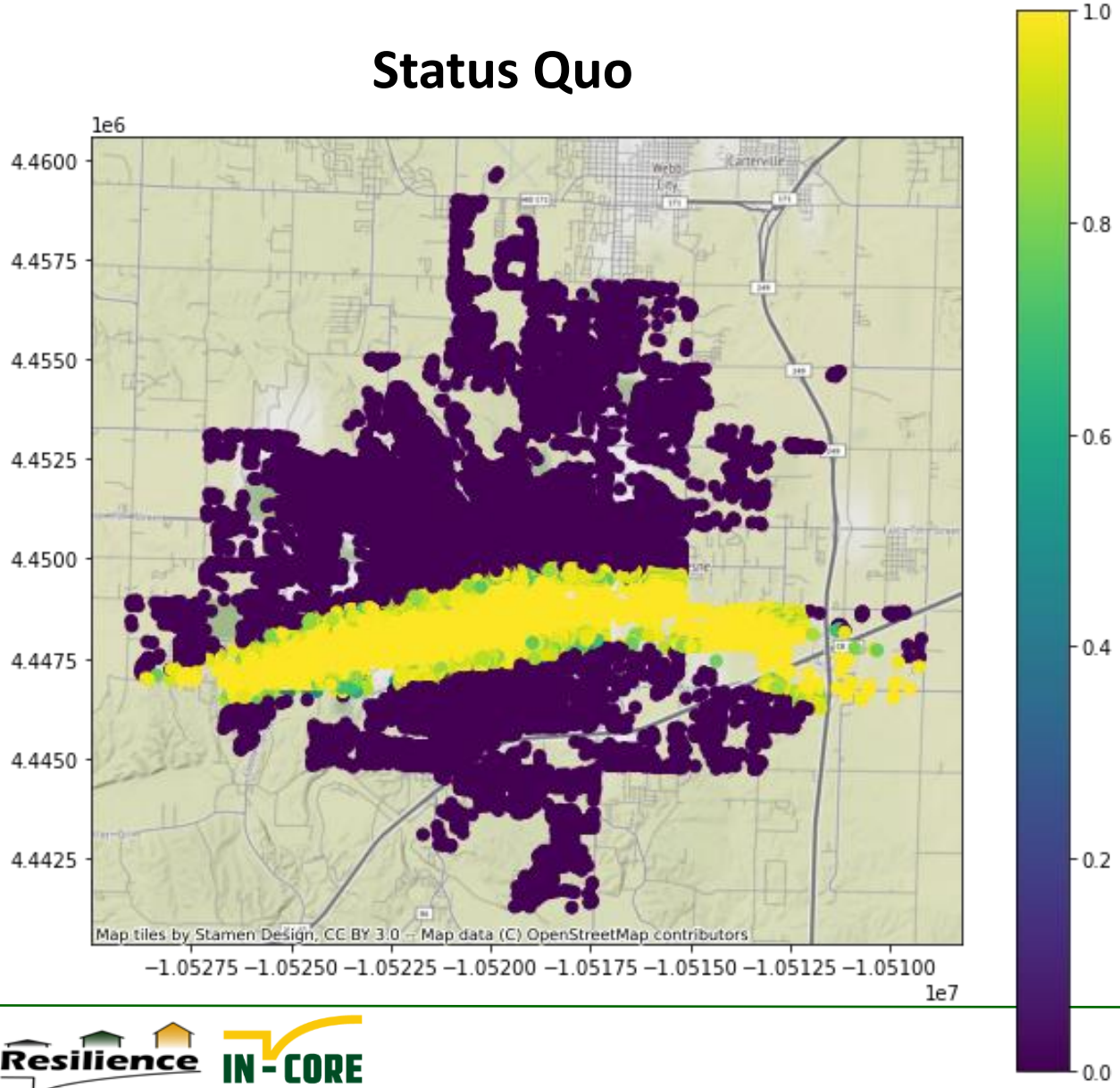


Budget $148M

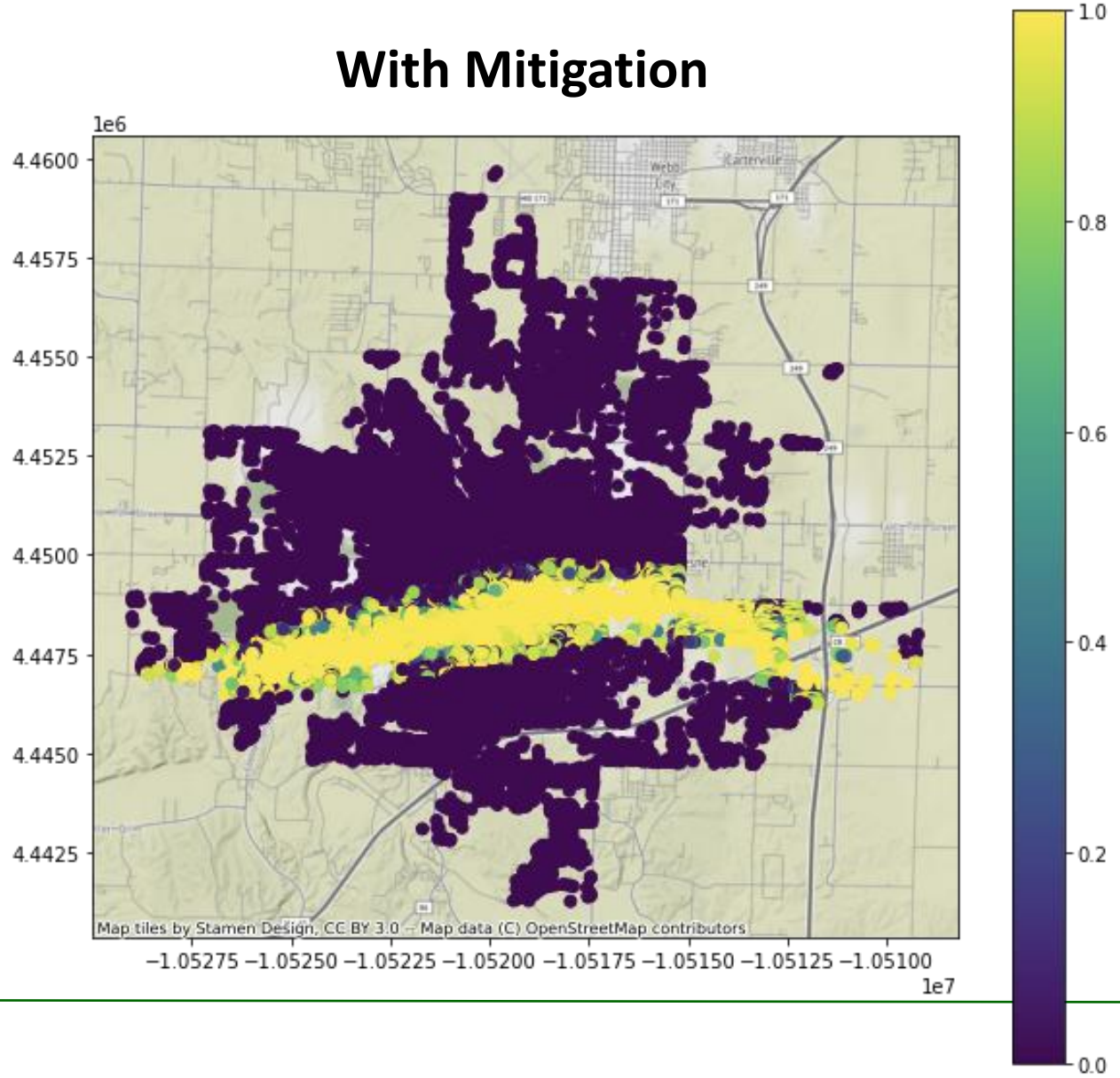


Evaluation

Status Quo

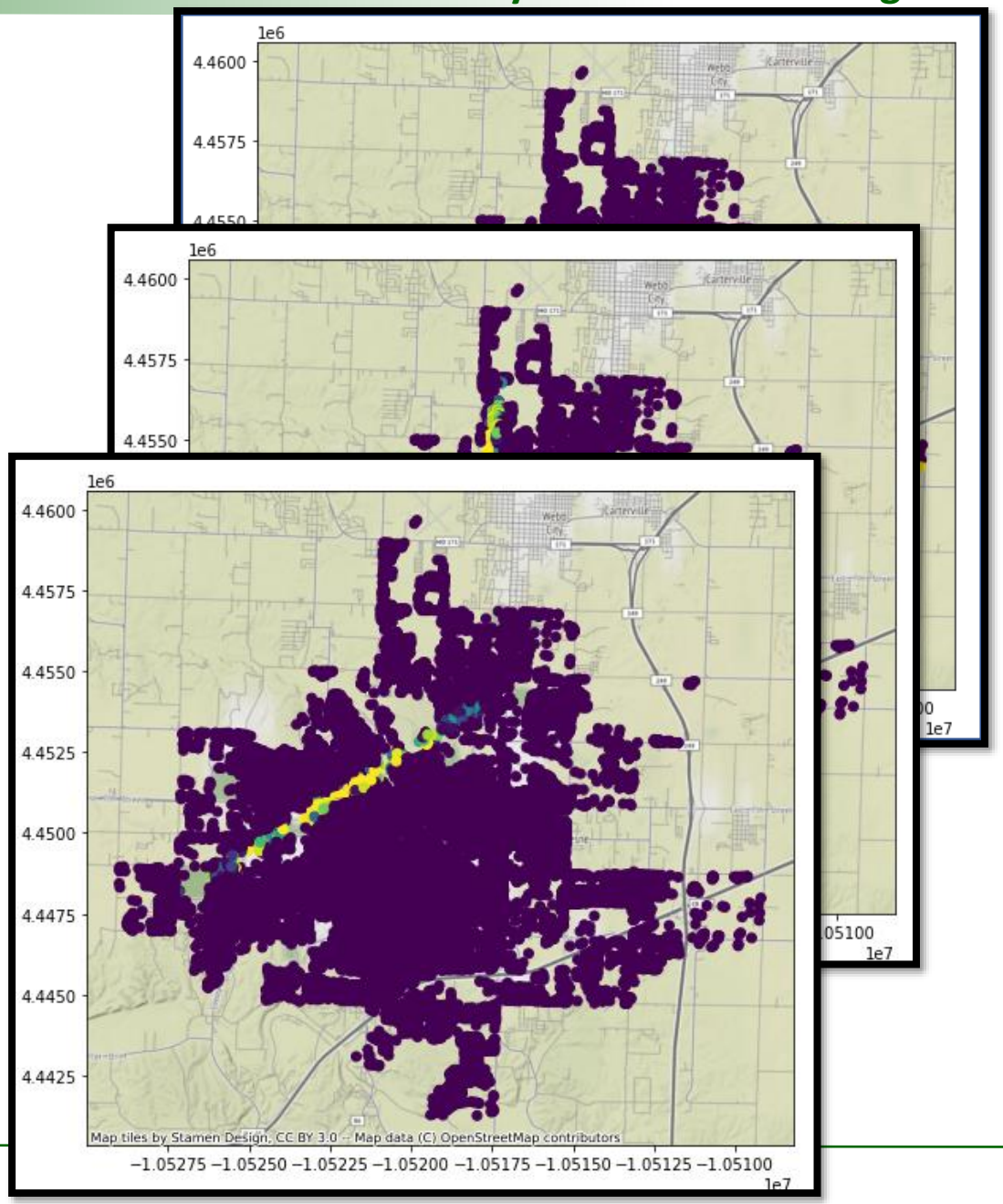


With Mitigation



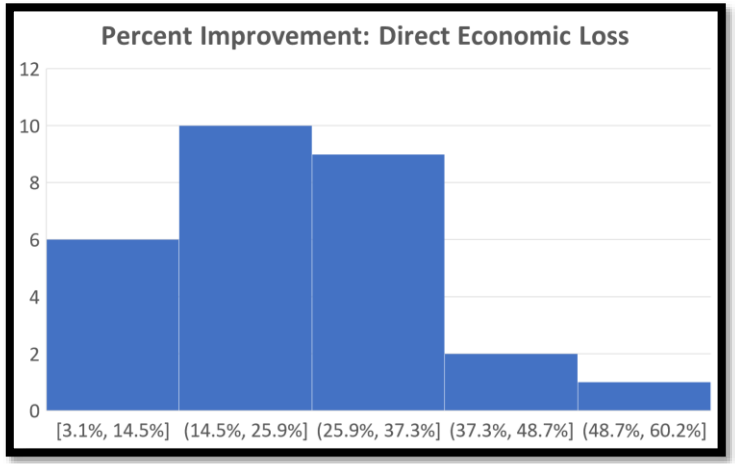
Evaluation

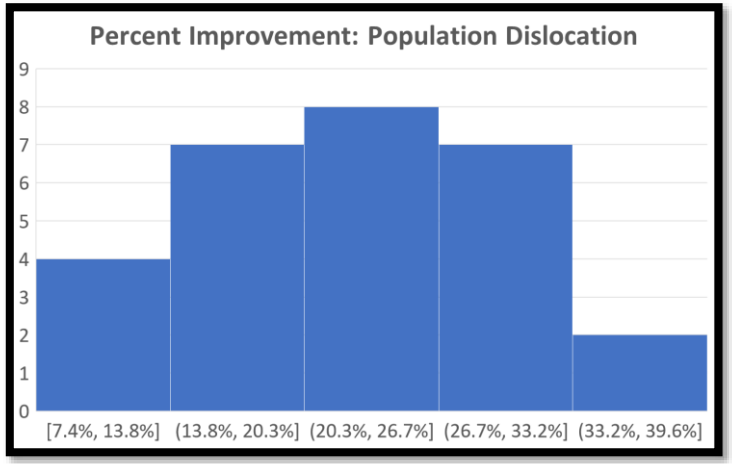
- 28 tornadoes simulated in IN-CORE, mostly EF3 with a few EF4/EF5
  - Joplin 2011 Tornado included
- Multiple optimization plans (blockgroup/archetype level) successfully evaluated
- Still working on chaining with the CGE model
- Evaluation on % improvement using optimization vs. no retrofits

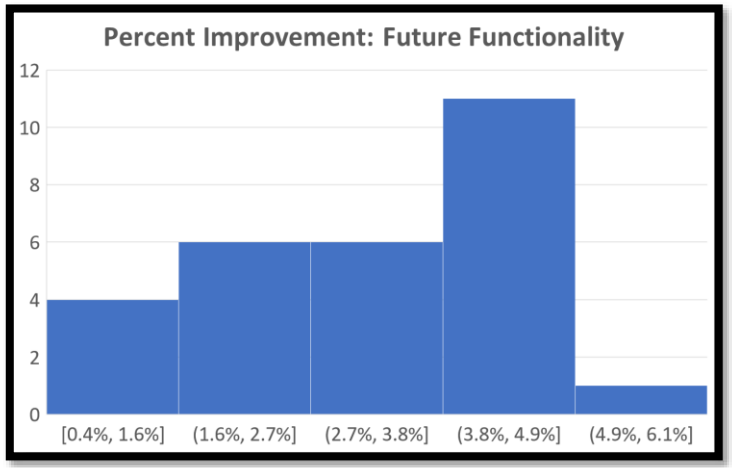


Evaluation

Tornado ID	Percent Improvments		
	Direct Loss	Dislocation	Functionality
1	43.6%	26.7%	3.89%
2	25.6%	25.0%	4.07%
3	35.1%	7.4%	0.43%
4	43.1%	39.6%	6.06%
5	18.8%	14.6%	2.10%
6	3.1%	8.9%	1.68%
7	28.5%	26.5%	4.02%
8	19.4%	17.8%	2.93%
9	26.4%	10.8%	2.24%
10	17.9%	25.4%	3.79%
11	28.7%	28.3%	4.79%
12	29.6%	31.8%	4.27%
13	25.8%	25.4%	4.09%
14	60.2%	21.4%	3.78%
15	29.8%	31.9%	4.50%
16	36.4%	22.0%	4.02%
17	21.0%	33.8%	4.25%
18	23.2%	29.3%	3.86%
19	27.6%	29.1%	3.82%
20	26.7%	15.8%	2.14%
21	9.0%	16.6%	1.47%
22	8.0%	7.8%	1.29%
23	21.7%	23.1%	3.69%
24	11.6%	19.6%	2.40%
25	16.2%	27.5%	3.78%
26	15.7%	24.3%	3.05%
27	9.6%	16.2%	2.53%
28	13.1%	18.4%	1.46%

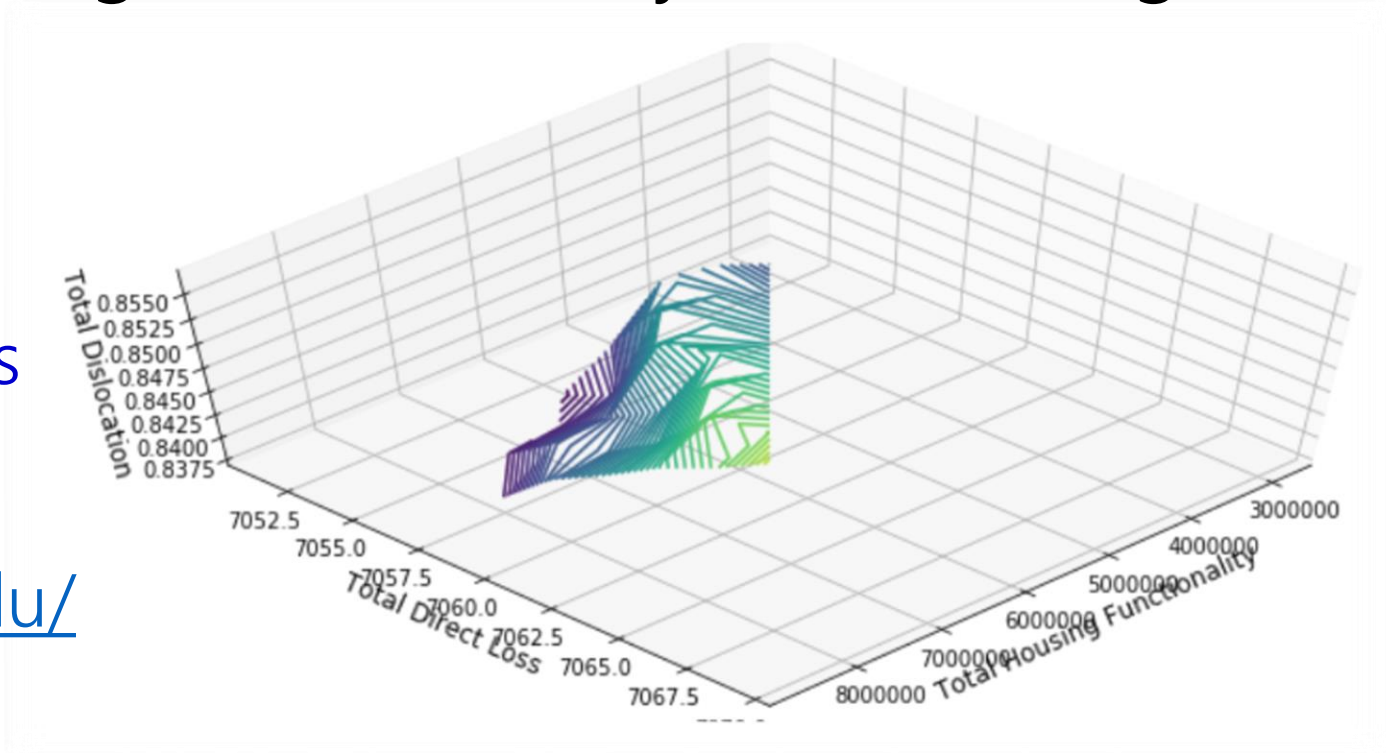






Summary

- Data collection, collation, cleaning, and coordinating is difficult
- Effective community resilience optimizations takes a tribe
- Practical optimization modeling is delimited by several things:
  - assumptions…
  - scope of interventions
  - quality, availability, granularity of data & models
- IN-CORE public release…
  - http://resilience.colostate.edu/



Examples and AMPL code

Multiple Objective Optimization

Galaxy Industries Production Problem (again)

Galaxy manufactures two toy guns:

- Space Rays, $8 dollar profit per unit
- Zappers, $5 dollar profit per unit

Management is seeking a weekly production schedule that will increase the company's profit subject to limited plastic, time, and marketing mix

Max: $8X_1 + 5X_2$ (Weekly profit)

subject to

$2X_1 + 1X_2 \leq 1000$ (Plastic)

$3X_1 + 4X_2 \leq 2400$ (Production Time)

$X_1 + X_2 \leq 700$ (Total production)

$X_1 - X_2 \leq 350$ (Mix)

$X_j >= 0,\ j = 1,2$ (Nonnegativity)

Galaxy Industries Production Problem (again)

What if the CEO said,

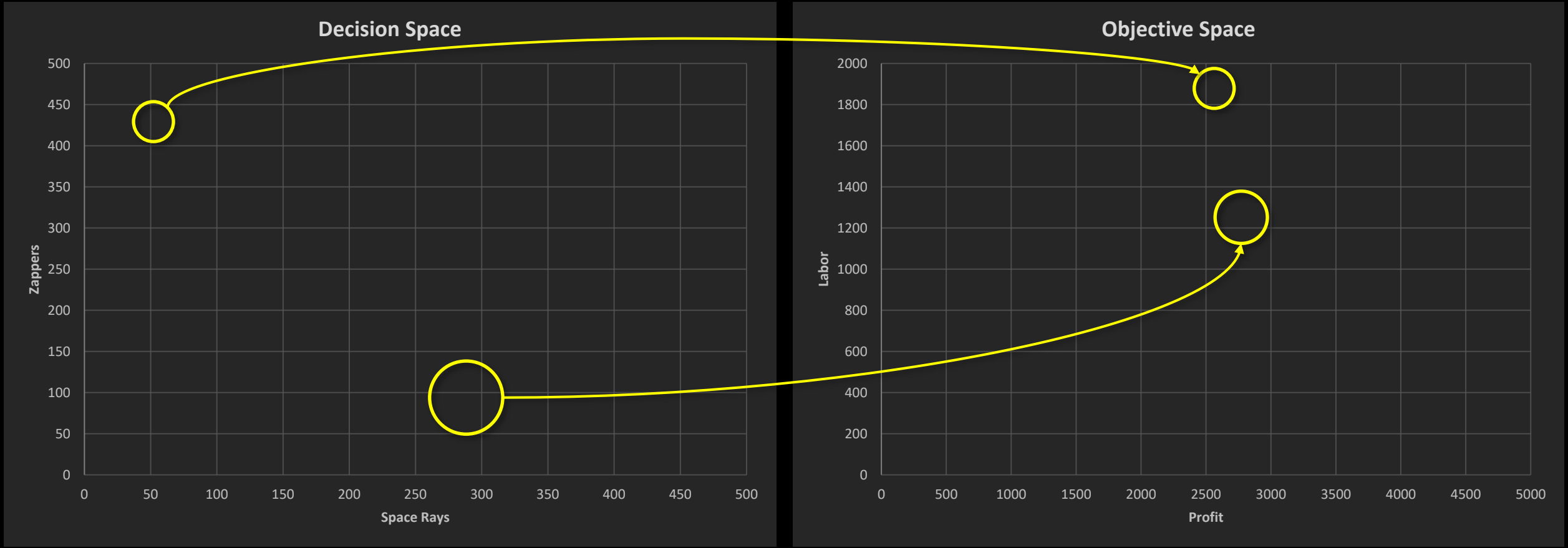
> "Maximizing profit is great, but we also want to minimize labor – let's do both!"

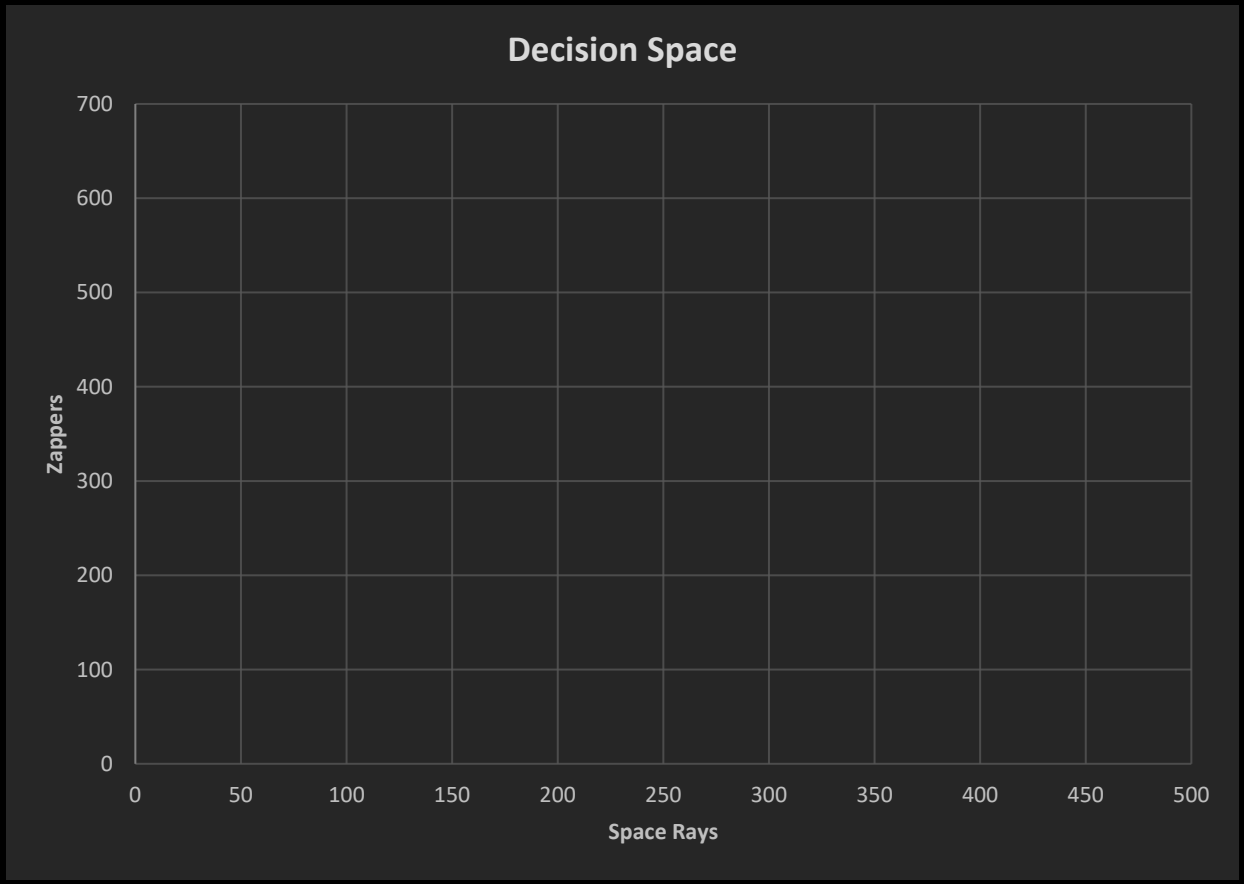
Galaxy Industries MOO (weighted sum)

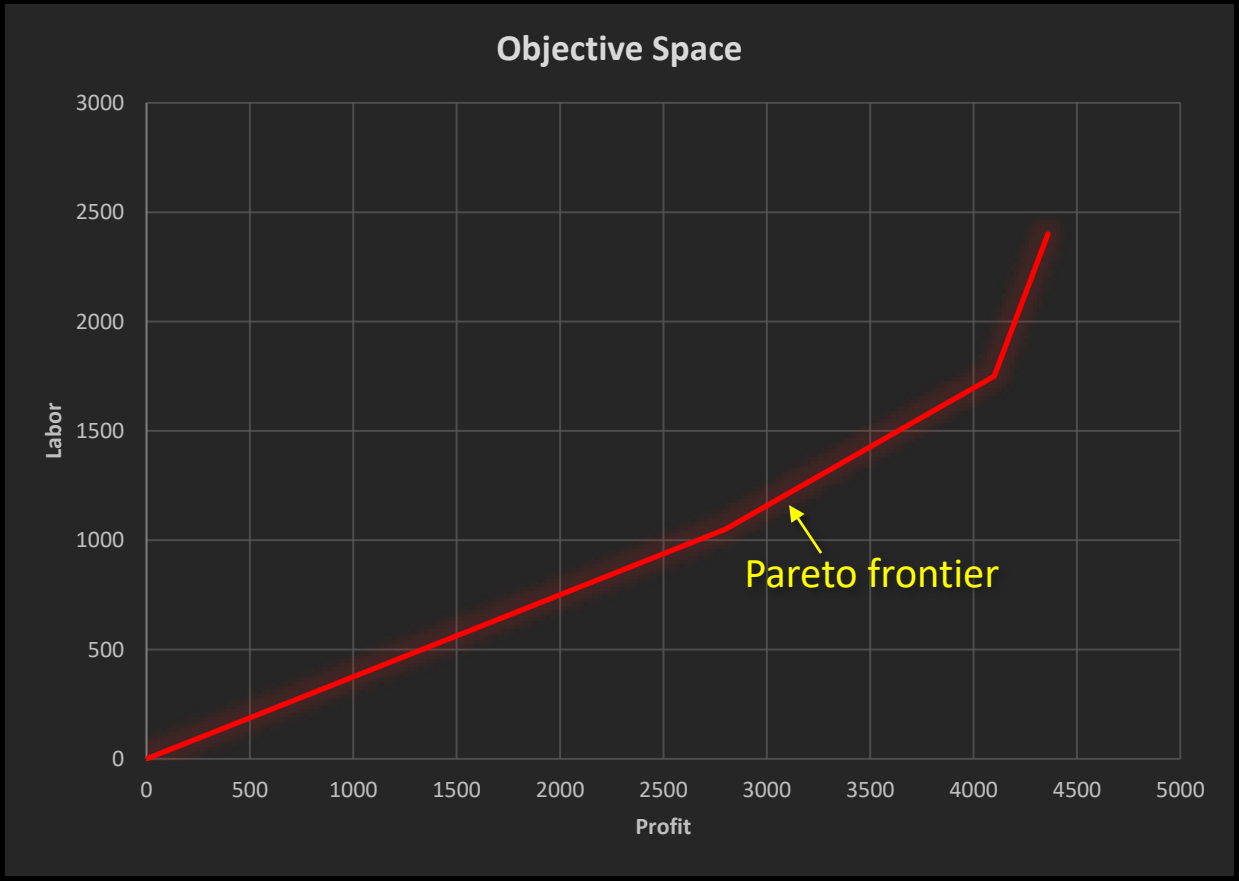
Maximize $\gamma_1(8X_1 + 5X_2) - \gamma_2(3X_1 + 4X_2)$

subject to

$2X_1 + 1X_2 \leq 1000$	(Plastic)
$3X_1 + 4X_2 \leq 2400$	(Production Time)
$X_1 + X_2 \leq 700$	(Total production)
$X_1 - X_2 \leq 350$	(Mix)
$X_j \geq 0,\ j = 1,2$	(Nonnegativity)


Decision Space
Zappers
Space Rays
0
50
100
150
200
250
300
350
400
450
500
Objective Space
Labor
Profit
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000


Decision Space
Zappers
700
600
500
400
300
200
100
0
0
50
100
150
200
250
300
350
400
450
500
Space Rays


Objective Space
Labor
3000
2500
2000
1500
1000
500
0
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
Profit
Pareto frontier

Multiple objective optimization AMPL Examples

Example problem:

- Multiple investment options – each with certain expected return and expected risk
- Budget of monies to be invested
- Some options have fixed costs
- Certain investment option mix are required, however, these are modeled as disjunctive constraints
- Competing objectives: maximize return; minimize risk

But first some more syntax…

- **Named problems** enable AMPL scripts to manage multiple problem instances within a single model and carry out iterative procedures that alternate between very different models
- The **let** command assigns values to parameters
- For-loops can be implemented, e.g.,

```
for {k in 0..100} {
  AMPL statements
}
```

- Redirecting output to files using **"> filename.txt"**

```
#SETS AND PARAMETERS ----------------------------------------
set INVESTMENTS;
set COMBINATIONS;

param return{INVESTMENTS} >=0;     #amount of returns as percent of investment
param risk{INVESTMENTS} >=0;       #amount of risk as percent of investment
param fixedCost{INVESTMENTS} >=0;

param upper{INVESTMENTS} >=0, <=1;      #upper bound on investment (percent of capital)
param lower{INVESTMENTS} >=0, <=1;      #lower bound on investment (percent of capital)

param capital >=0;

param combo{COMBINATIONS, INVESTMENTS};
param comboReq{COMBINATIONS} >=0;

param numConstraints >=0;
param bigM = capital;

#DATA ---------------------------------------------------
data riskReward.dat;
```

```
#DECISION VARIABLES ------------------------------------------
#dollars to invest in investment i \in INVESTMENTS
var x {i in INVESTMENTS} >=lower[i]*capital, <=upper[i]*capital;

#whether or not to invest in investment i
var y {i in INVESTMENTS} binary;

#to model the disjunctive constraints
var z {j in COMBINATIONS} binary;

#OBJECTIVE ----------------------------------------------------
maximize objRETURNS:   sum{i in INVESTMENTS} return[i] * x[i];
minimize objRISKS:     sum{i in INVESTMENTS} risk[i] * x[i];

#CONSTRAINTS --------------------------------------------------
s.t. budget: sum{i in INVESTMENTS} (x[i] + y[i]*fixedCost[i]) = capital;

s.t. mix {c in COMBINATIONS}: sum{i in INVESTMENTS} combo[c,i] * x[i] >= capital *
                              comboReq[c] - bigM*(1-z[c]);

s.t. disjunctive: sum{c in COMBINATIONS} z[c] >= numConstraints;

s.t. investBinary {i in INVESTMENTS}: x[i] <= y[i]*bigM;
```

Named problems

```
#PROBLEMS
problem maxReturn: objRETURNS, x, y, z, budget, mix, disjunctive, investBinary;
problem minRisk:   objRISKS,   x, y, z, budget, mix, disjunctive, investBinary;

printf "\nMaximize Return...........\n";
solve maxReturn;

printf "\nMinimze Risk...........\n";
solve minRisk;
```

Pre-emptive optimization for AMPL example

```
#PREMPTIVE OPTIMIZATION ------------------------------------------------
param optimalValue;

#using RETURN as the 1st priority

printf "\nOptimize first priority...\n";
solve maxReturn;
let optimalValue := objRETURNS;

#create a new constraint based on result of first optimization
s.t. stayOptReturns: sum{i in INVESTMENTS} return[i] * x[i] >= optimalValue;

#create new named problem with the new constraint and solve
printf "\nOptimize second priority subject to bounds on first...\n";
problem minRisk_2: objRISKS, x, y, z, budget, mix, disjunctive, investBinary, stayOptReturns;

solve minRisk_2;
```

Pre-emptive optimization for AMPL example

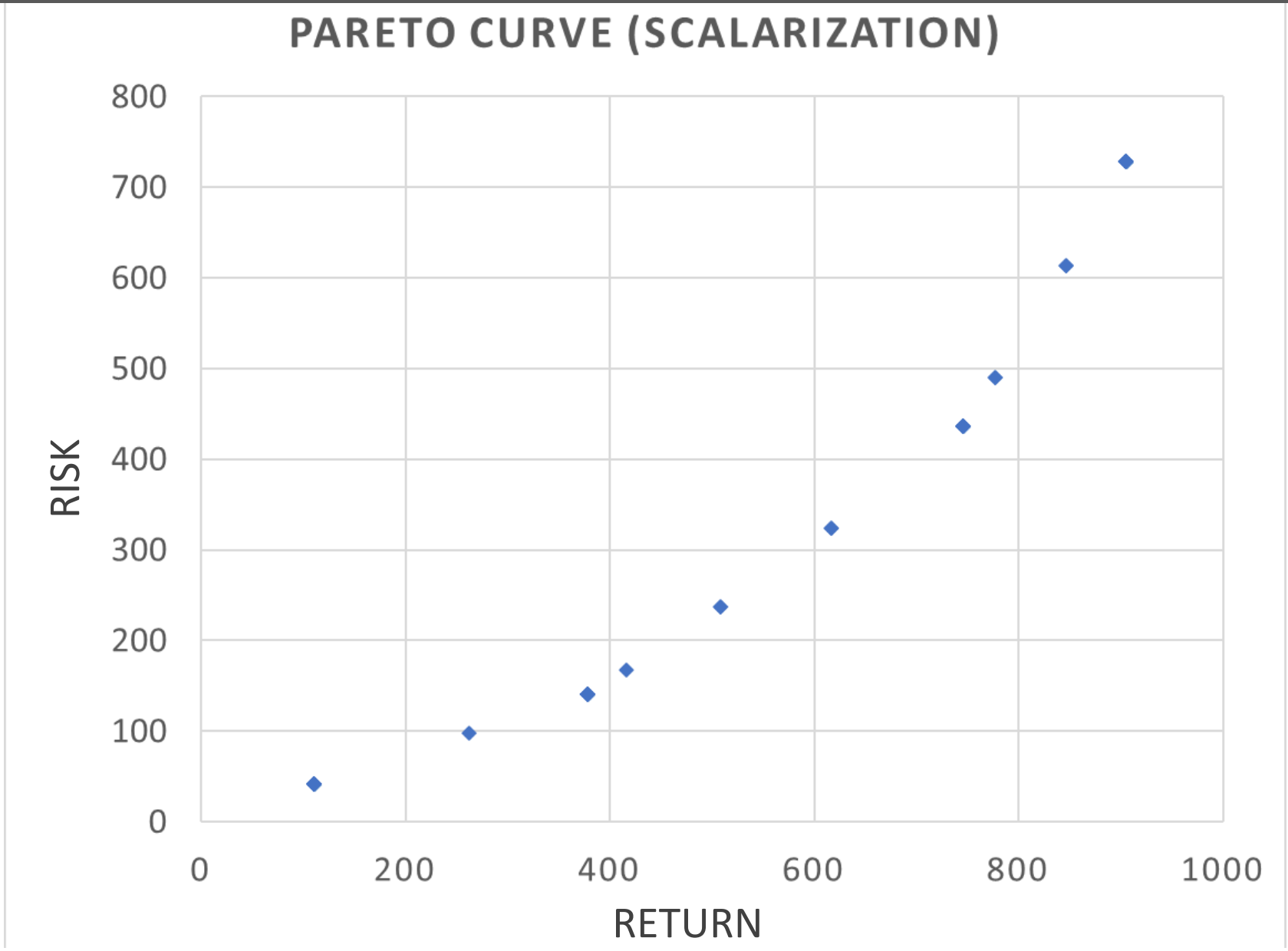
```
#PREMPTIVE OPTIMIZATION ------------------------------------------------
param optimalValue;

#or using RISK as the 1st priority

printf "\nOptimize first priority...\n";
solve minRisk;
let optimalValue := objRISKS;

#create a new constraint based on result of first optimization
s.t. stayOptRisks: sum{i in INVESTMENTS} risk[i] * x[i] <= optimalValue;

printf "\nOptimize second priority subject to bounds on first...\n";
problem maxReturn_2: objRETURNS, x, y, z, budget, mix, disjunctive, stayOptRisks, investBinary;

solve maxReturn_2;
```

Scalarization method for AMPL example

```
#SCALARIZATION ----------------------------------------------
printf "\n\nSCALARIZATION ----------------------------------------------\n";

param lambda1;
param lambda2;

let lambda1 := 0.25;
let lambda2 := 0.75;

maximize weighedRiskReturn:
     sum{i in INVESTMENTS} (lambda1 * (return[i] * x[i]) - lambda2 * (risk[i] * x[i]));

problem scalarized: weighedRiskReturn, x, y, z, mix, disjunctive, budget, investBinary;

solve scalarized;
```

Scalarization with looping for AMPL example

```
printf "\n\nMultiple values for SCALARIZATION ------------------------------------\n";

for {k in 0..100} {
  let lambda1 := k/100;
  let lambda2 := 1 - lambda1;

  solve scalarized;

  printf "%3.2f, %3.2f, %7.4f, %7.4f\n",
        lambda1, lambda2, objRETURNS, objRISKS > "paretoS.txt";

}
```

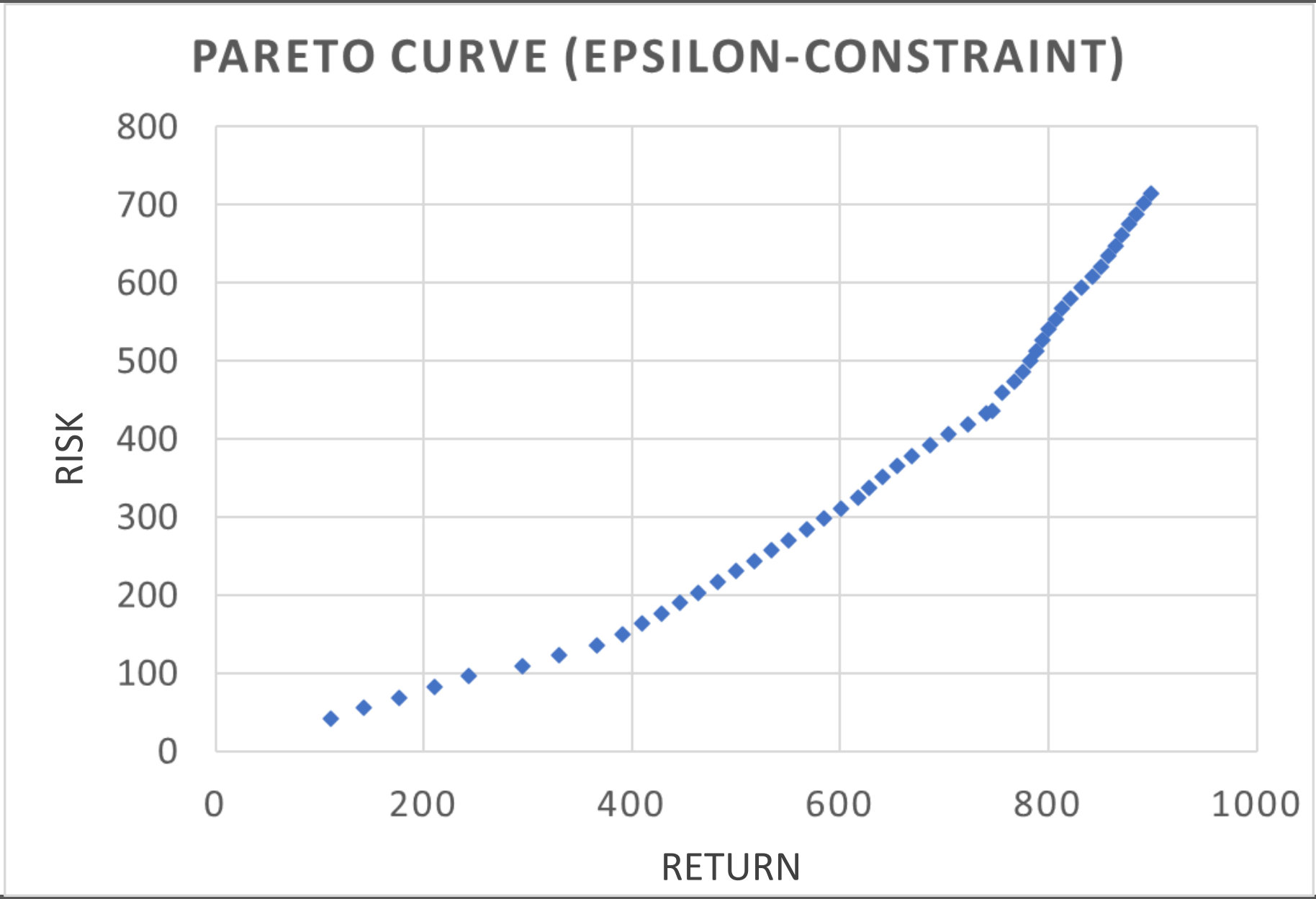

PARETO CURVE (SCALARIZATION)
800
700
600
500
400
300
200
100
0
RISK
0
200
400
600
800
1000
RETURN

ϵ-constraint method for AMPL example

```
#EPSILON-CONSTRAINT
printf "\n\EPSILON-CONSTRAINT ----------------------------------------------\n";

#get upper and lower bounds for objectives
param upperRisk;
param lowerRisk;

solve maxReturn;
let upperRisk:=objRISKS;

solve minRisk;
let lowerRisk:=objRISKS;

param epsilon;
let epsilon := lowerRisk;
```

ϵ-constraint method for AMPL example

```
#create the epsilon constraint
s.t. epsilsonRisks:  sum{i in INVESTMENTS} risk[i] * x[i] <= epsilon;

#create the new named problem
problem epsConst: objRETURNS, x, y, z, mix, disjunctive, budget, investBinary, epsilsonRisks;

param steps = 50;

printf "\n\nMultiple values for EPSILON-CONSTRAINT ------------------------------------\n";
for {eps in 0..steps} {

       let epsilon := lowerRisk + eps*(upperRisk - lowerRisk)/(steps+1);
       solve epsConst;

       printf "%d, %7.4f, %7.4f, %7.4f\n",
            eps, epsilon, objRETURNS, objRISKS > "paretoEps.txt";
}
```


PARETO CURVE (EPSILON-CONSTRAINT)
800
700
600
500
400
300
200
100
0
RISK
0
200
400
600
800
1000
RETURN

Goal programming (GP)

- Another version of multiple objective optimization is known as Goal Programming
  - This is closely related to the “scalarization” method already discussed
- The single objective function in GP is formed by minimizing deviations from one or more goals
- The deviations can be weighted to emphasize one goal over another

Have you used “deviation variables” before???

Galaxy Industries Production Problem (again)

Galaxy manufactures two toy guns:

- Space Rays, $8 dollar profit per unit
- Zappers, $5 dollar profit per unit

Management is seeking a weekly production schedule that will increase the company's profit subject to limited plastic, time, and marketing mix

Max: $8X_1 + 5X_2$ (Weekly profit)

subject to

$2X_1 + 1X_2 \leq 1000$ (Plastic)

$3X_1 + 4X_2 \leq 2400$ (Production Time)

$X_1 + X_2 \leq 700$ (Total production)

$X_1 - X_2 \leq 350$ (Mix)

$X_j > = 0,\ j = 1,2$ (Nonnegativity)

Galaxy Industries Production Problem (again)

What if the CEO said,

> "Optimization is great and all, but what I really care about is that our projections are *accurate* and that we hit our profit targets as discussed in our last shareholder meeting – that is, we need to make exactly $4000 per week."

Galaxy Industries Production Problem (again)

We can't simply constrain the problem using $8X_1 + 5X_2 = 4000$, since maybe it is impossible to make $4000!

$4000 is our aspiration level (i.e., a goal).

So, create new variables to represent deviations from $4000.

d^+: how much we are OVER the goal

d^- : how much we are UNDER the goal

And add the constraint: $8X_1 + 5X_2 = 4000 + d^+ - d^-$

Then, minimize any deviations: minimize $d^+ + d^-$

Galaxy Industries Goal Programming

minimize $\mathbf{d^+ + d^-}$	**(Deviations)**
subject to	
$\mathbf{8X_1 + 5X_2 = 4000 + d^+ - d^-}$	**(Goal constraint)**
$2X_1 + 1X_2 \leq 1000$	(Plastic)
$3X_1 + 4X_2 \leq 2400$	(Production Time)
$X_1 + X_2 \leq 700$	(Total production)
$X_1 - X_2 \leq 350$	(Mix)
$X_j \geq 0,\ j = 1,2$	(Nonnegativity)

```
#Galaxy Industries Goal Programming Problem

#decision variables
var SR >=0; #number of SpaceRay guns to manufacture
var  Z >=0; #number of Zapper guns to manufacture

#deviation variables
var d_positive >=0;  #deviation above goal
var d_negative >=0;  #deviation under goal

#goal: hit 4000 in profit exactly

#objective
minimize deviations: d_positive + d_negative;  #model objective as deviation from goal

#constraints
subject to goal: 8*SR + 5*Z = 4000 + d_positive - d_negative;

subject to plastic:   2*SR +   Z <= 1000;
subject to labor:     3*SR + 4*Z <= 2400;
subject to production:  SR +   Z <= 700;
subject to management:  SR -   Z <= 350;

solve;
printf "\nOptimal solution values:\n SR = %6.2f   Z = %6.2f \n", SR, Z;
printf "\nProfit generated: %6.2f\n",  8*SR + 5*Z;
printf "Deviation from goal: %6.2f \n", deviations;
```

Galaxy Industries Goal Programming

This is just a scalarized (i.e., weighted sum), multiple objective optimization problem…

- Objective 1: minimize positive deviations
- Objective 2: minimize negative deviations
- $\gamma_1 = \gamma_2 = 1$

And of course, we can change the weights if desired, e.g.,

- Underperforming is 10 times worse than exceeding the goal, so set $\gamma_1 = 1$ and $\gamma_2 = 10$
- or, if it is good to exceed the goal, set $\gamma_1 = -1$

Additionally, you can have multiple goals, each with differently weighted deviation variables