EVOLVE WITH GENETIC ALGORITHMS

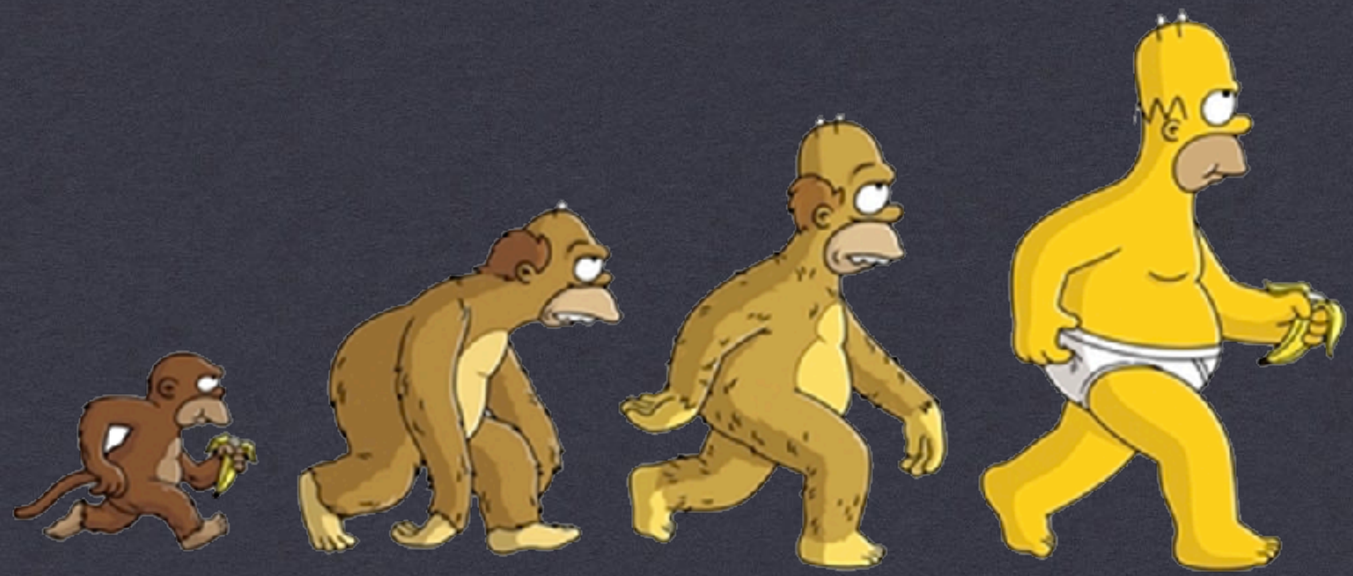
CHARLIE KOSTER

@CKOSTER22

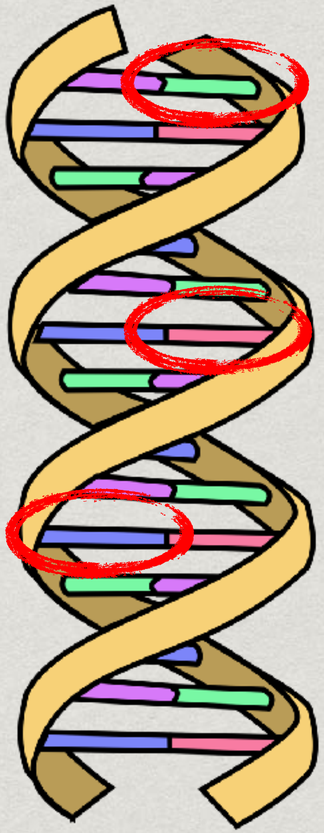
TITANIUM SPONSORS

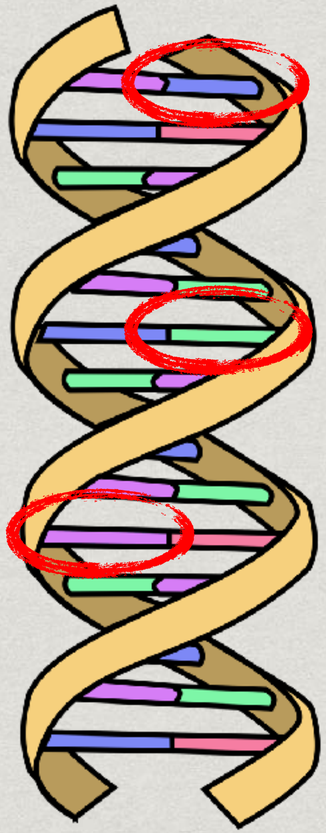
Platinum Sponsors

Gold Sponsors

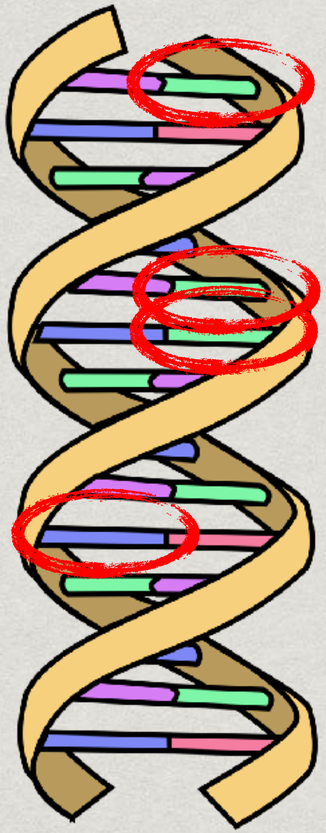


EVOLUTION IN 2 MINUTES

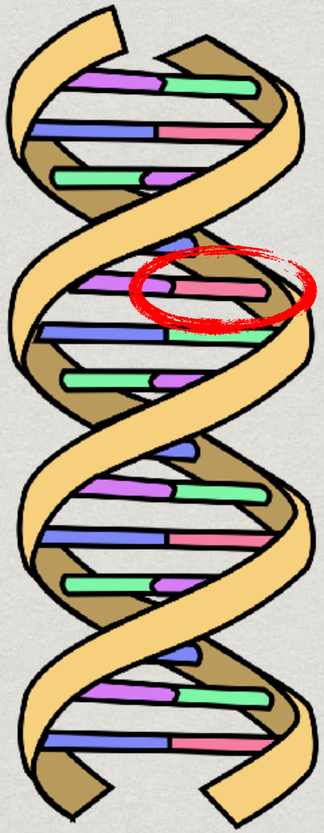




Crossover

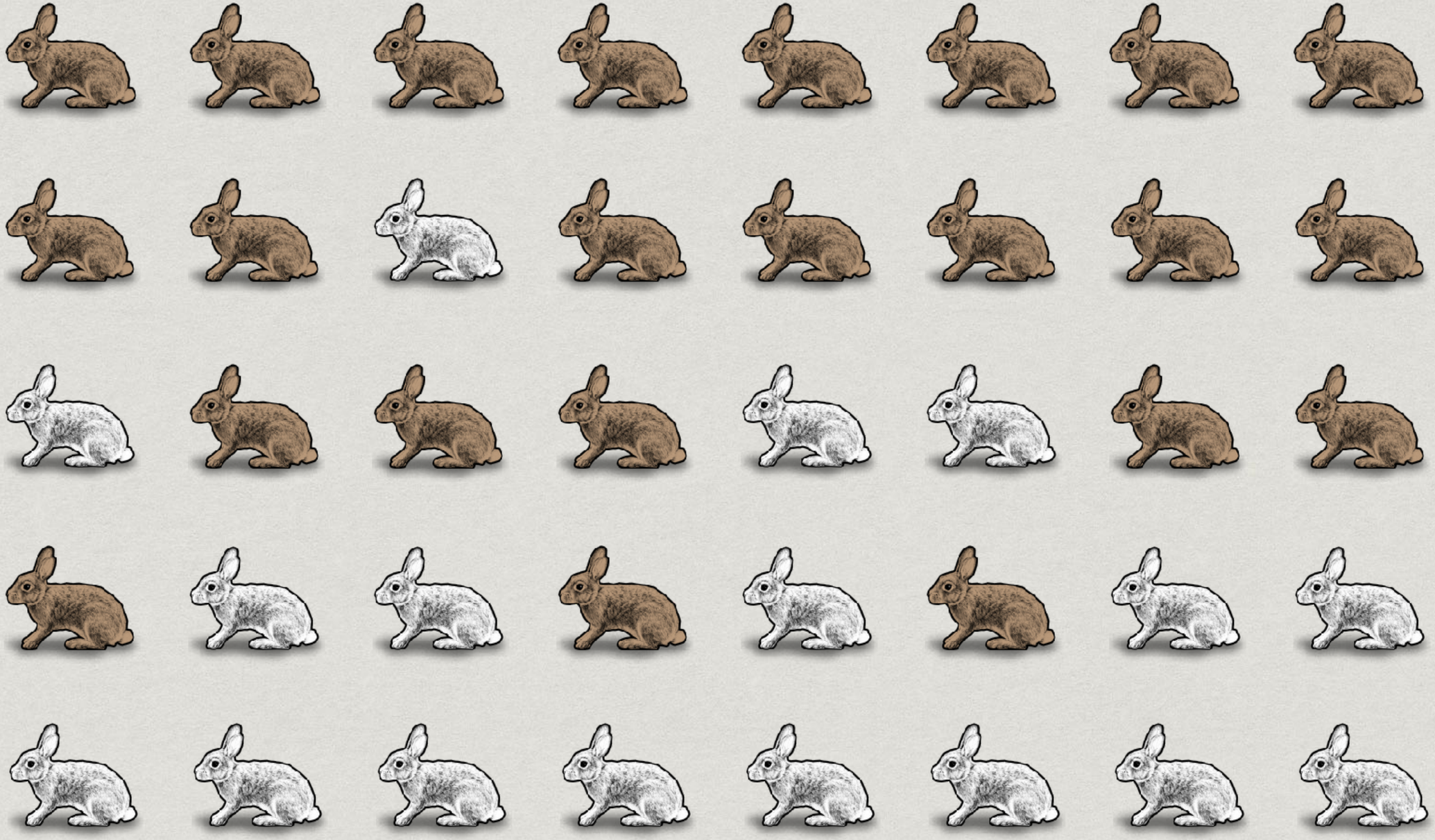


Mutation



https://commons.wikimedia.org/wiki/File:DNA_simple.svg

Time



BIOLOGY

Survive to reproduce

Organism

DNA

Natural selection

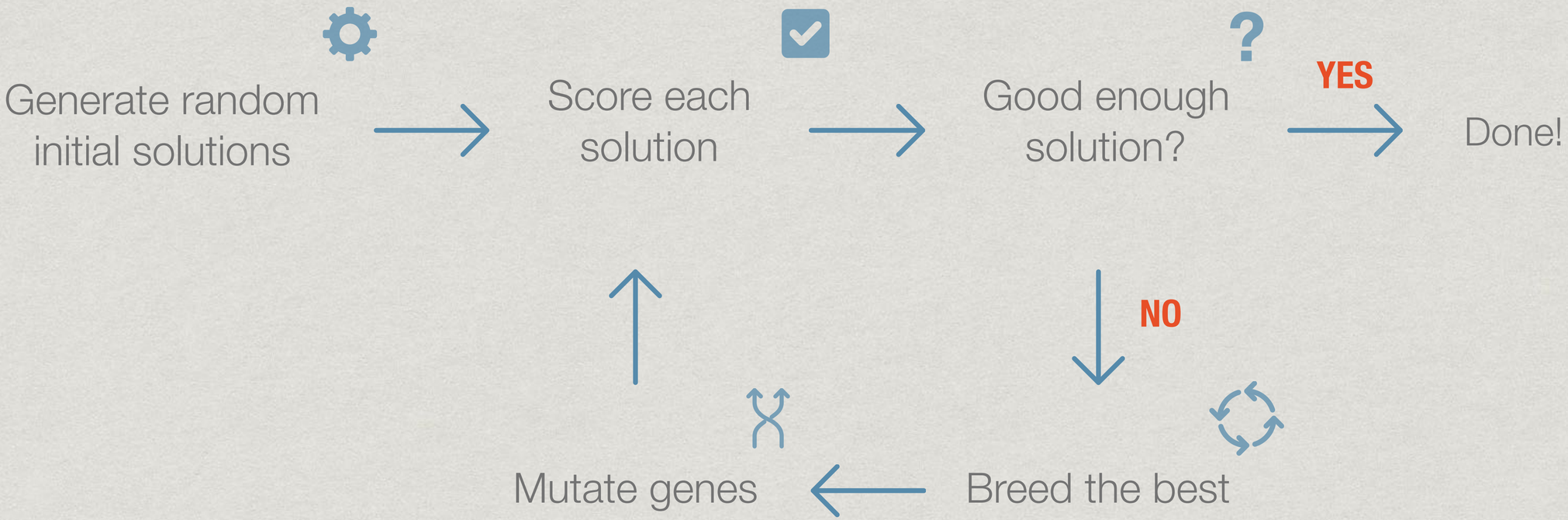
GENETIC
ALGORITHM

Optimization problem

Possible solution

Data record

Fitness function

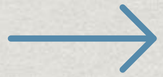

Generate random initial solutions
Score each solution
Good enough solution?
YES
Done!
NO
Breed the best
Mutate genes

EVOLVING SOLUTIONS BY EXAMPLE

Hello, World

Evolving Images

Meal Planner



1. Define the problem
2. Define a representation of the solution
3. Define a fitness function
4. Define how two solutions “breed”

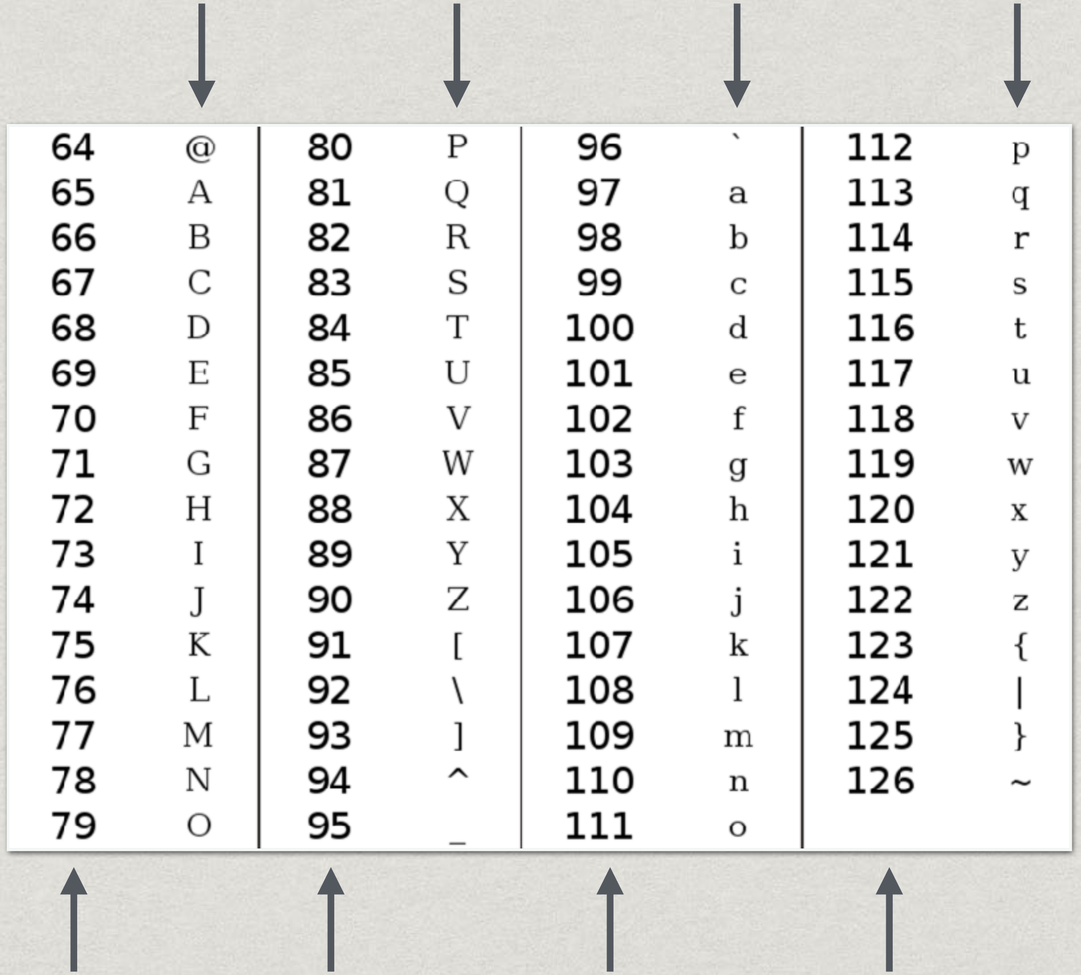
HELLO, WORLD

EXAMPLE 1

1. **Produce the phrase “Hello world”**
   Define the problem

2. **DNA is an array of ASCII codes**
   Model The Solution

3. **A function that sums the differences between ASCII codes**
   Define a fitness function



64	@	80	P	96	`	112	p
65	A	81	Q	97	a	113	q
66	B	82	R	98	b	114	r
67	C	83	S	99	c	115	s
68	D	84	T	100	d	116	t
69	E	85	U	101	e	117	u
70	F	86	V	102	f	118	v
71	G	87	W	103	g	119	w
72	H	88	X	104	h	120	x
73	I	89	Y	105	i	121	y
74	J	90	Z	106	j	122	z
75	K	91	[	107	k	123	{
76	L	92	\	108	l	124	|
77	M	93	]	109	m	125	}
78	N	94	^	110	n	126	~
79	O	95	_	111	o		

H	e	l	l	o		w	o	r	l	d
72	101	108	108	111	32	119	111	114	108	100

z	e	k	R	m	A	W	t	r	j	c
122	101	107	82	109	65	87	116	114	106	99

72 101 108 108 111 32 119 111 114 108 100

---

50 0 1 26 2 33 32 5 0 2 1

122 101 107 82 109 83 87 116 114 106 99

152

107	82	100	108	111	90	115	115	108	108	65
122	101	107	82	109	65	87	116	114	106	99

107	82	100	108	111	65	87	116	114	106	99
107	82	100	108	111	65	87	116	114	71	99

```
const crossoverDnas = (dna1: Dna, dna2: Dna): Dna => {
    let parent1DnaPart: Dna;
    let parent2DnaPart: Dna;

    if (Math.random() < 0.5) {
        parent1DnaPart = dna1.slice(0, CROSSOVER_SPLIT_INDEX);
        parent2DnaPart = dna2.slice(CROSSOVER_SPLIT_INDEX);
    } else {
        parent1DnaPart = dna2.slice(0, CROSSOVER_SPLIT_INDEX);
        parent2DnaPart = dna1.slice(CROSSOVER_SPLIT_INDEX);
    }

    return parent1DnaPart.concat(parent2DnaPart);
};
```

```
const mutateDna = (dna: Dna): Dna => {
    const randomIndex: number = getRandomIndex(dna.length);
    const randomAsciiCode: number = asciiForRandomCharacter();

    return dna
            .slice(0, randomIndex)
            .concat(randomAsciiCode)
            .concat(dna.slice(randomIndex + 1));
};
```

DEMO

GEN 0	176 FteAqKivhzs	218 krdliNvDato	188 SsJjZDmghoI	218 HMqkmlMeGJW
GEN 1	176 FteAqKivhzs	167 krdliAkyiOh	213 SsJjBKivhzs	268 doauWlMeGJW
GEN 10	93 HWkkmAbyivf	124 HWqkcAbyXvf	108 JWqkzAbwivh	91 HWkomAbyovh
GEN 50	45 HekkmAWtrjc	95 odkkmAQtnjc	61 HkkkmAQtnjc	83 HdkkmAxtnjc
GEN 500	3 Helko Wprkd	37 Hxlko fprkd	29 Hxlkh Wprkd	25 Hxlko Wprkg
GEN 762	0 **HELLO WORLD**			

NOTEWORTHY

- It isn’t random
- Mutation rate ~3% - 10%
- Computers are fast, but not for realtime systems

EVOLVING IMAGES

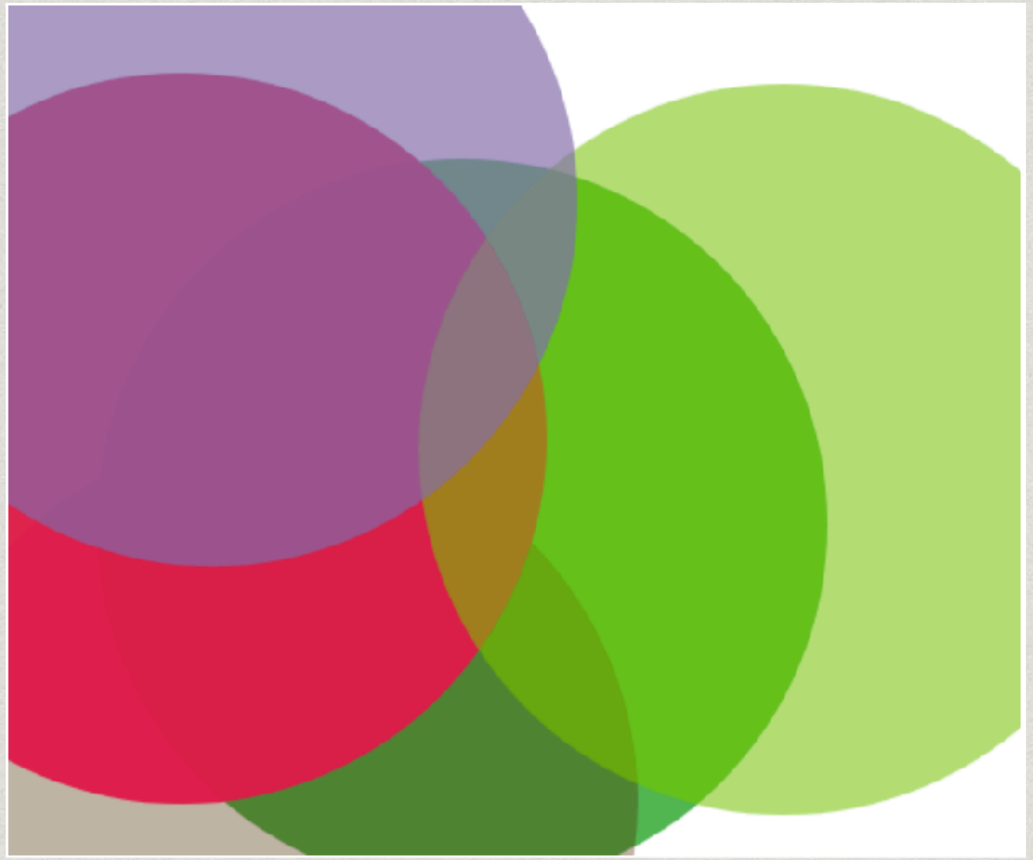
EXAMPLE 2

1. Replicate an Image With Shapes

Define the problem

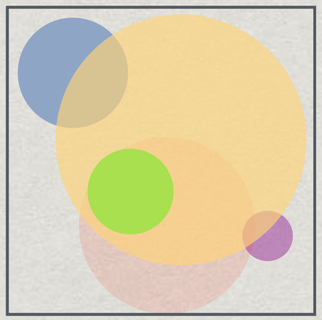
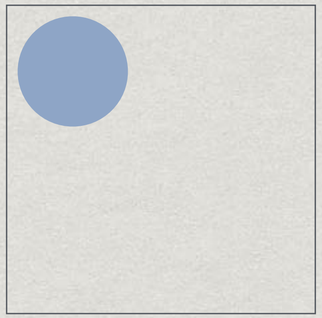
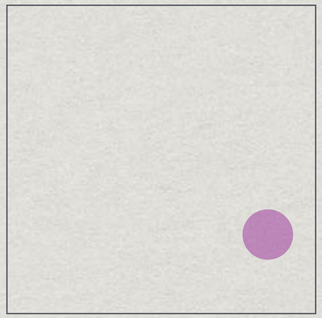
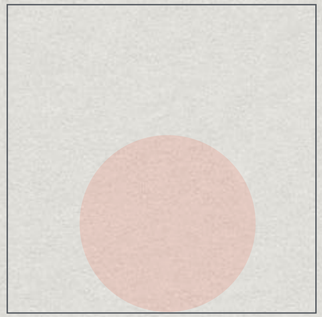
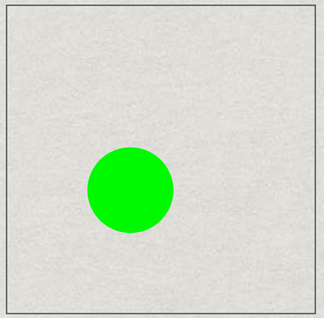
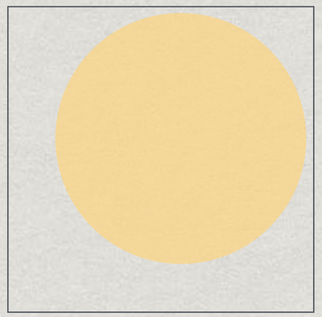
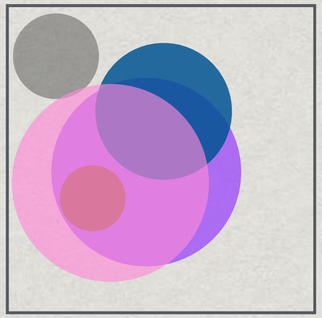
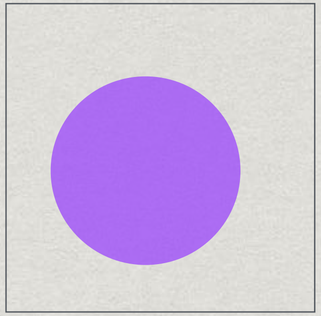
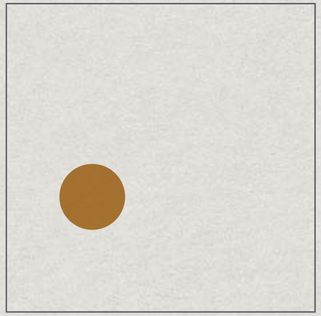
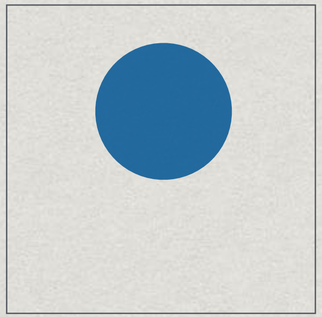
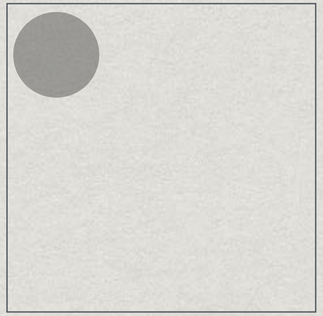
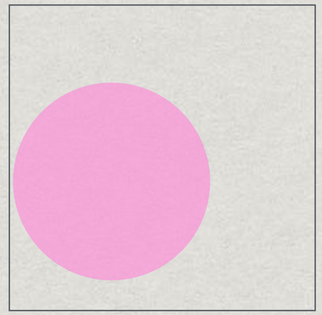
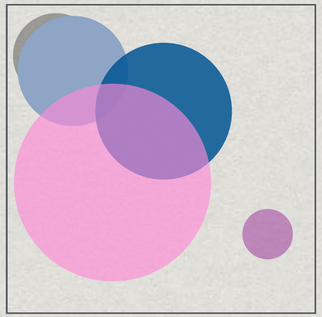
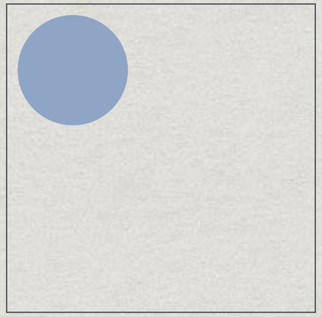
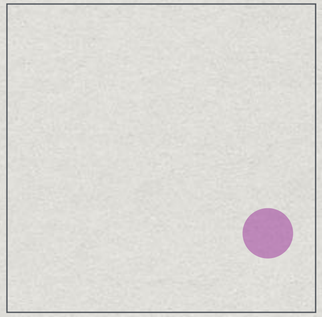
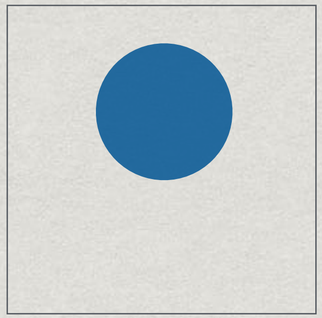
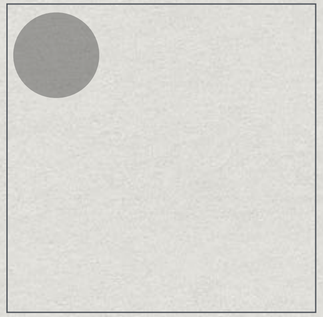
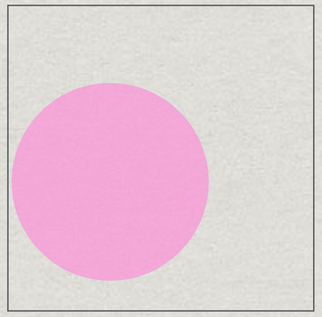
1. **Replicate an Image With Shapes**
   Define the problem

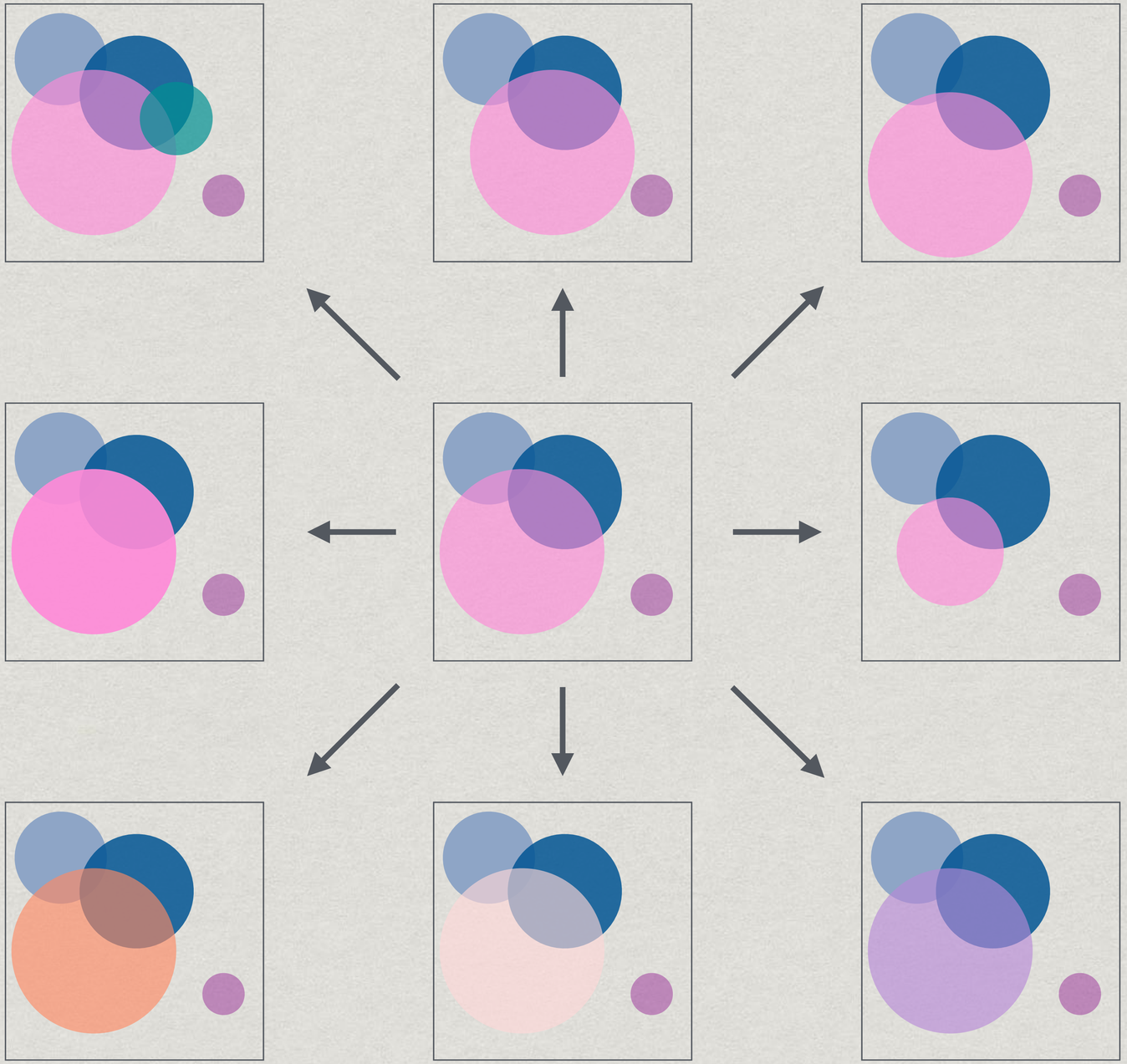
2. **Canvas with overlapping semi-transparent circles**
   Model The Solution



```
type Dna = Array<Circle>;

type Circle = {
    x: number,
    y: number,
    radius: number,
    red: number,
    green: number,
    blue: number,
    alpha: number
};
```

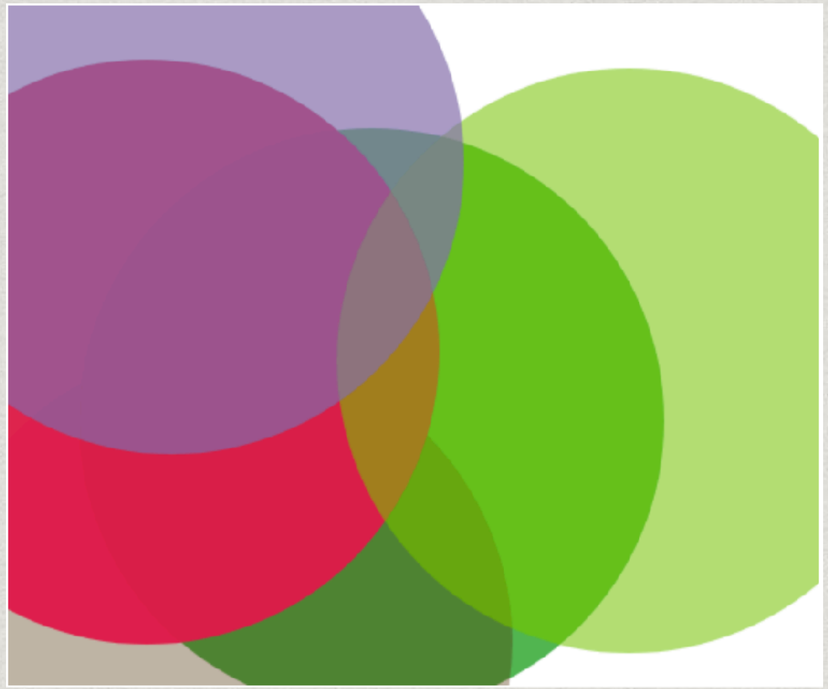


1. **Replicate an Image With Shapes**
   Define the problem

2. **DNA is an array of overlapping shapes**
   Model The Solution

3. **A function that inspects and diffs the canvas pixel by pixel**
   Define a fitness function

[(109,129,59,255),(113,129,60,255),(115,130,62,255),
(111,131,60,255),(111,131,61,255),(110,130,60,255),
(109,128,58,255),(113,127,61,255),(118,130,65,255),
(116,128,63,255),(109,128,59,255),(107,128,58,255),
(107,130,59,255),(109,129,59,255),(113,128,59,255),
(110,131,60,255),(106,133,60,255),(107,128,59,255),
(105,123,55,255),(109,128,58,255),(111,126,57,255),
(112,124,56,255),(106,122,55,255),(100,121,54,255),
(110,126,57,255),(112,128,57,255),(105,128,56,255),
(109,127,59,255),(101,86,42,255),(75,39,8,255),
(87,22,6,255),(96,22,10,255),(96,19,8,255),
(91,17,5,255),(93,18,6,255),(104,16,8,255),
(109,19,11,255),(130,39,29,255),(144,60,47,255),
(141,63,47,255),(143,61,47,255),(142,59,46,255),
(133,58,40,255),(138,64,46,255),(139,65,48,255),



[(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),…]

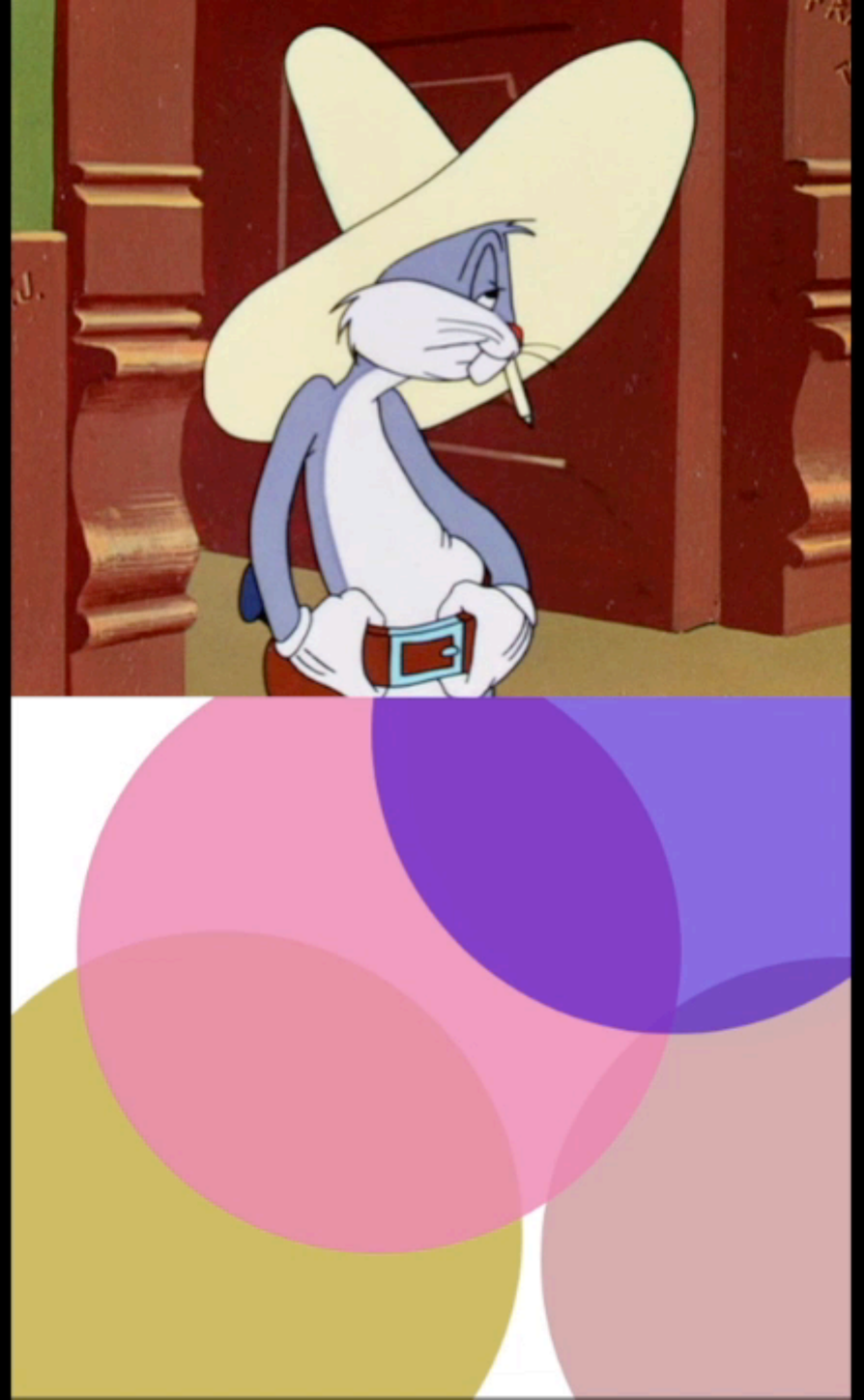
[(109,129,59,255),(113,129,60,255),(115,130,62,255),
(111,131,60,255),(111,131,61,255),(110,130,60,255),
(109,128,58,255),(113,127,61,255),(118,130,65,255),
(116,128,63,255),(109,128,59,255),(107,128,58,255),
(107,130,59,255),(109,129,59,255),(113,128,59,255),
(110,131,60,255),(106,133,60,255),(107,128,59,255),
(105,123,55,255),(109,128,58,255),(111,126,57,255),
(112,124,56,255),(106,122,55,255),(100,121,54,255),
(110,126,57,255),(112,128,57,255),(105,128,56,255),
(109,127,59,255),(101,86,42,255),(75,39,8,255),
(87,22,6,255),(96,22,10,255),(96,19,8,255),(91,17,5,255),
(93,18,6,255),(104,16,8,255),(109,19,11,255),
(130,39,29,255),(144,60,47,255),(141,63,47,255),
(143,61,47,255),(142,59,46,255),(133,58,40,255),
(138,64,46,255),(139,65,48,255),(140,63,47,255),
(140,59,45,255),…]

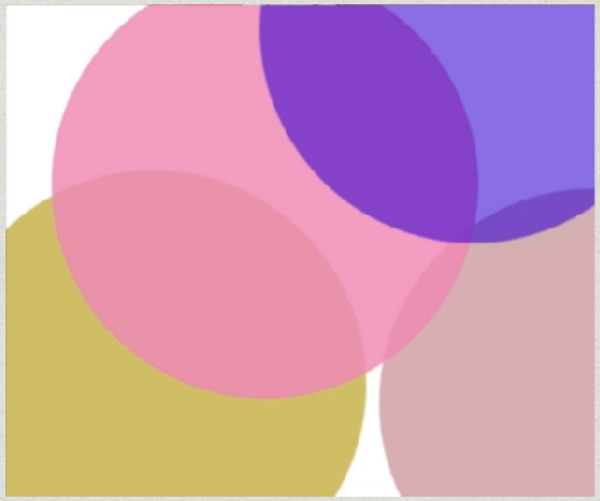
[(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),(128,110,170,171),(128,110,170,171),
(128,110,170,171),…]

~20 minutes

378 circles

10,900 generations





Gen: 0

Gen: 2200

Gen: 4400

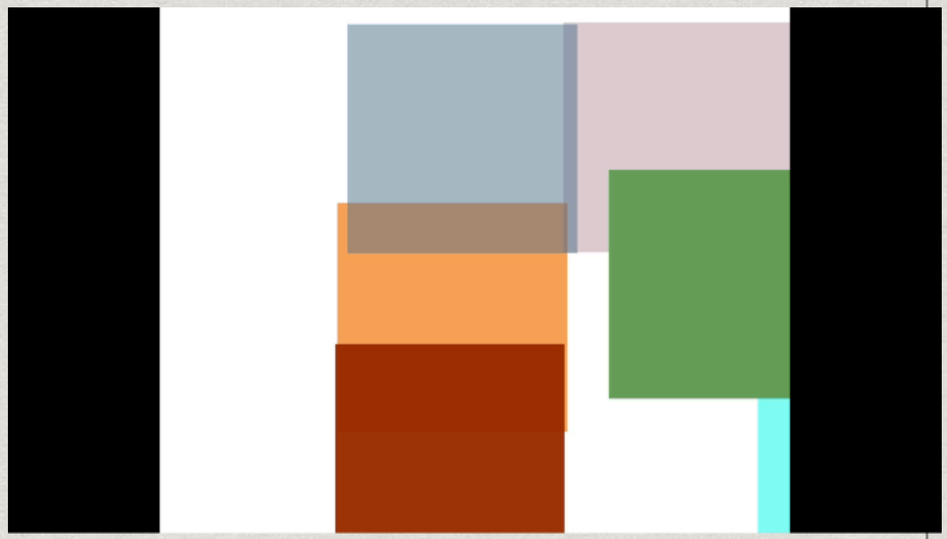
Gen: 6600

Gen: 8800

Gen: 10900

NOTEWORTHY

- Genetic Algorithms evolve **good enough solutions** for *optimization problems*
- There is always a bottleneck
- Performance tip - Know your solution space and traverse it efficiently
  - Lines instead of circles?
  - Squares instead of circles?
  - Other?



MEAL PLANNER

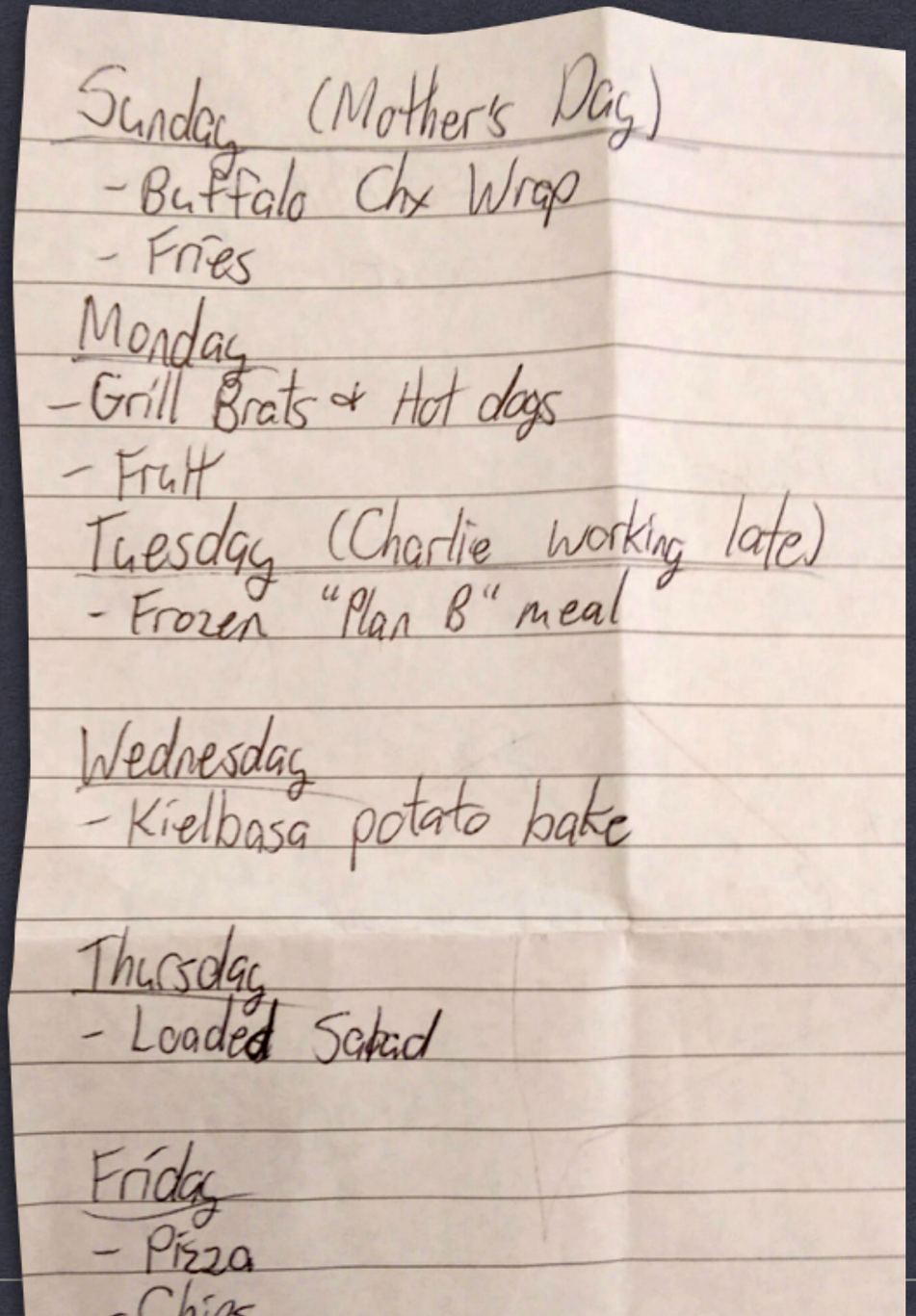
EXAMPLE 3

Define the Problem

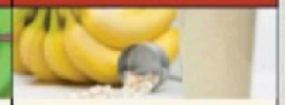
“Meal planning is tedious and I don’t like doing it”

-Me

- Schedule
- Holidays
- Weather
- Variety
- Leftovers
- Diet goals
- Prep time



Model the Solution

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Breakfast	 Protein Southwest Scramble	 Breakfast Pita Pizza	 Curry Cheddar Scrambled Eggs	 Brie cheese on bread	 Eggs with Hats on Top	 Fruit Ball	 Oatmeal banana protein shake
Lunch	 Merguez Sausage on a Bed of Quinoa	 Breaded Salmon	 Sweet Kale and Bean Mix	 Grilled Herbed Poussins	 Fillet of Sole with Leek Sauce	 Tarragon Crab Salad	 French Dip Sandwich
Dinner	 Maple Glazed Salmon	 Nut Burgers	 Romano Chicken	 Broccoli Almondine	 Fisherman's Quick Fish	 Black Bean and Salsa Soup	 Spinach Sautee With Brown Butter & Garlic

```
type alias MealPlan =
    { daysOfWeek : List Day }

type alias Day =
    { breakfast : Maybe Meal
    , lunch : Maybe Meal
    , dinner : Maybe Meal }
```

```
type alias Meal =
    { recipe : Recipe
    , numServings : Int }

type alias Recipe =
    { name : String
    , maxServings : Int
    , ingredients : List Ingredient }
```

DNA

Crossover step with lists

```
type alias MealPlan =
    { daysOfWeek : List Day }

type alias Day =
    { breakfast : Maybe Meal
    , lunch : Maybe Meal
    , dinner : Maybe Meal }
```

No mutation here!!

Mutable

No meal -> Random meal

Meal -> No meal

Meal -> Random meal

```
type alias Meal =
    { recipe : Recipe
    , numServings : Int }

type alias Recipe =
    { name : String
    , maxServings : Int
    , ingredients : List Ingredient }
```

Mutable

Recipe -> Random recipe

Servings -> Random servings

Immutable!

Financial

→ Cost
Food waste
Leftovers

Health

Health concerns
Diet type
Allergies
Perishability

Environment

Equipment
Cooking skill
Different people cooking

Food

Rating
Cuisine
→ Variety
Family recipes
Favorite foods
Ingredient availability
Backup meals

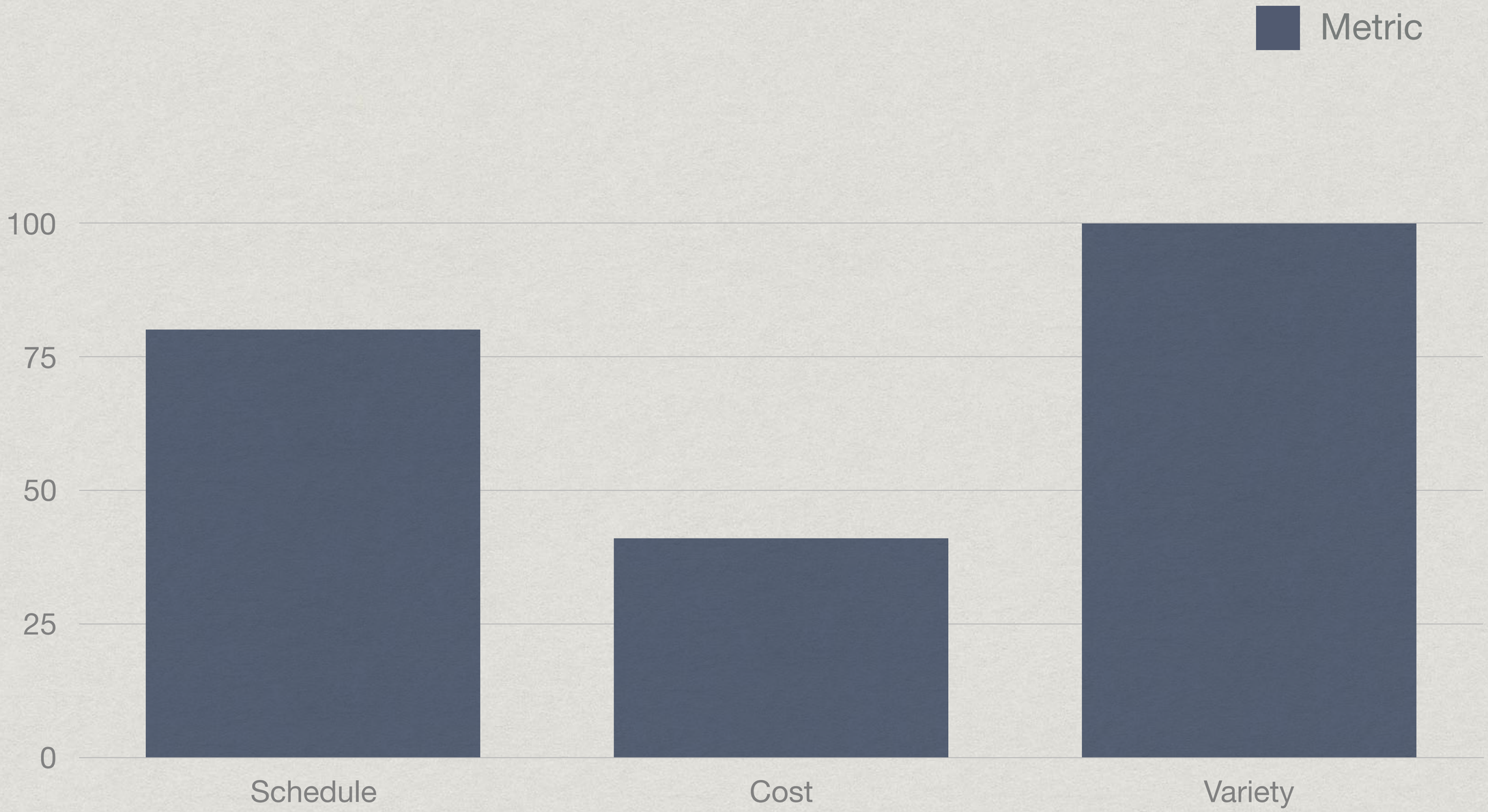
Scheduling

Prep time
→ Schedule
→ Restrictions on certain days
Seasonality

FITNESS FUNCTION

Metrics

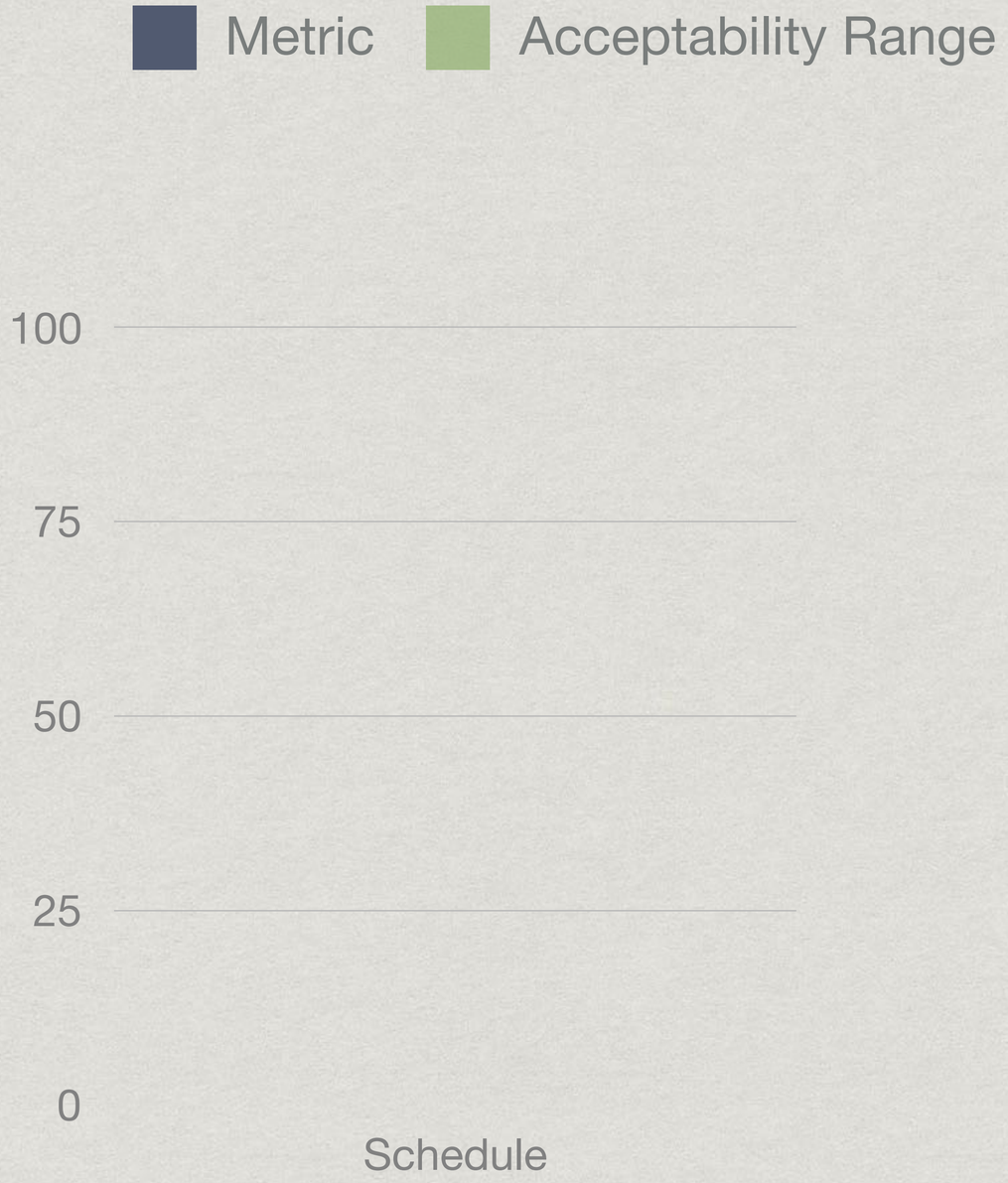
- Personal schedule — *No weekday breakfast, no Friday lunch, etc*
- Cost — *$70 - $80*
- Variety — *“Some variety”*


Metric
100
75
50
25
0
Schedule
Cost
Variety

FITNESS FUNCTION

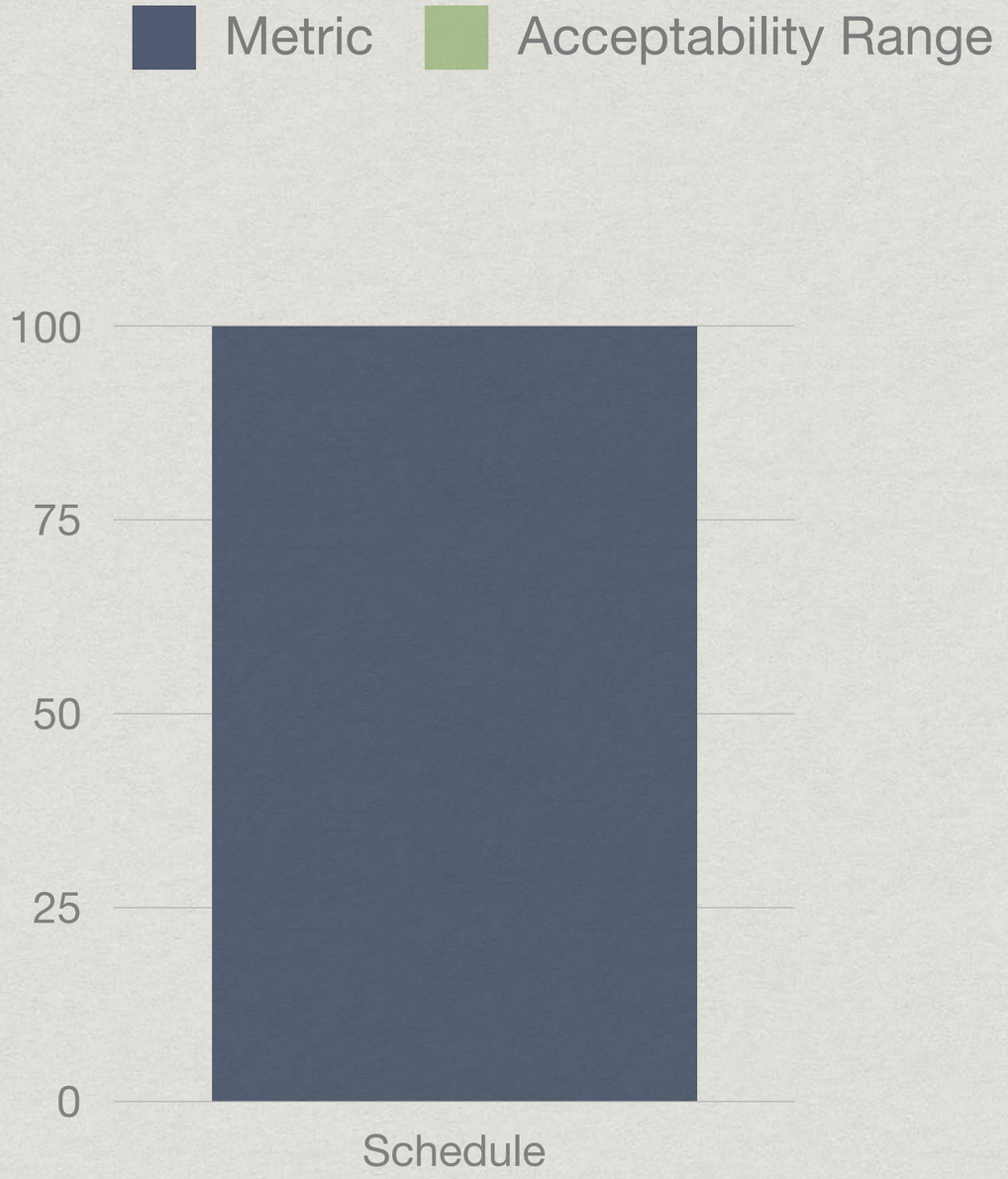
- What does 0 mean?
- What does 100 mean?
- What is an acceptable range?

1. **Every meal’s servings off by > 1**
   What does 0 mean?



1. **Every meal’s servings off by > 1**
   What does 0 mean?

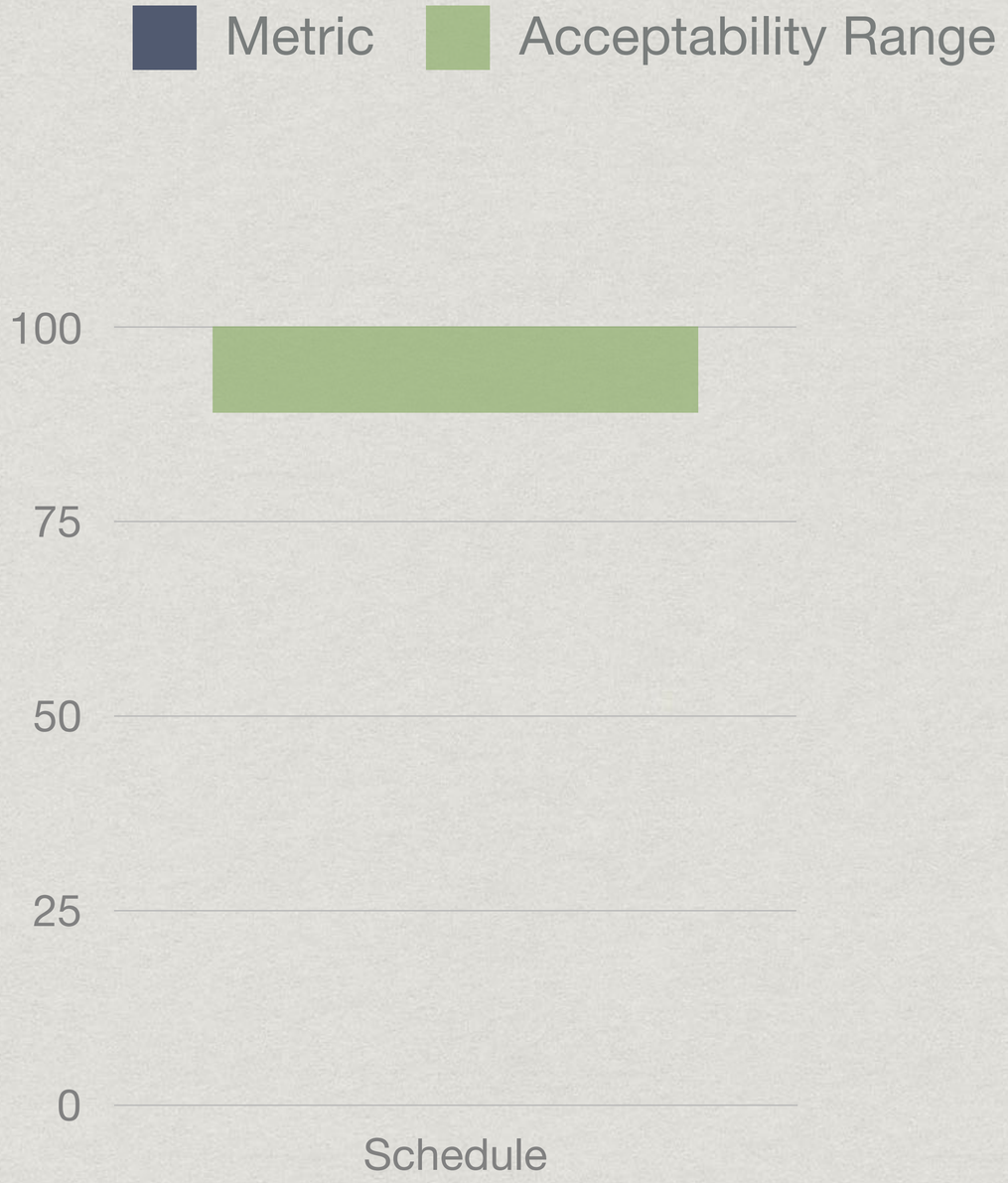
2. **# servings exactly matches our schedule**
   What does 100 mean?



1. **Every meal’s servings off by > 1**
   What does 0 mean?

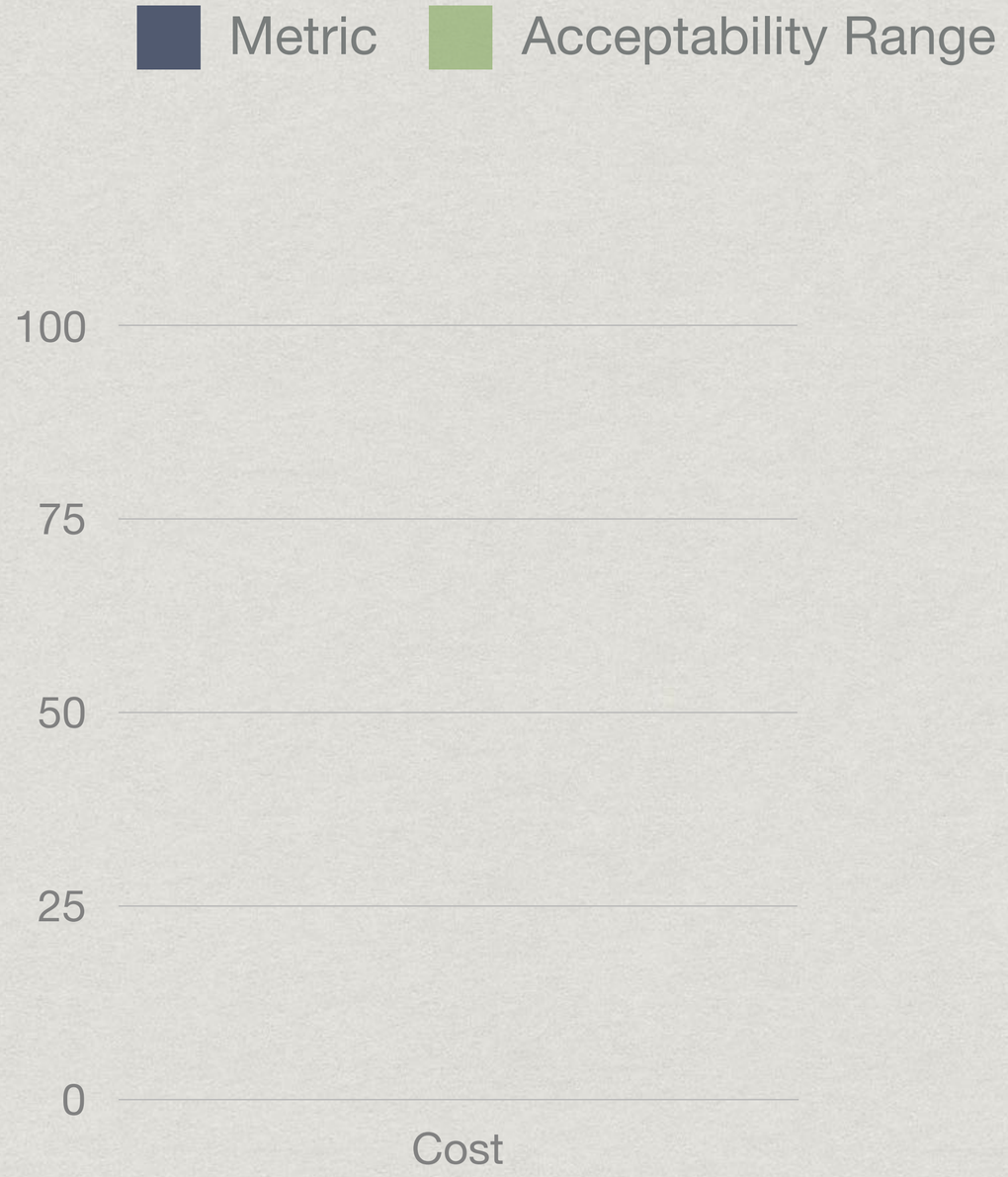
2. **# servings exactly matches our schedule**
   What does 100 mean?

3. **Ok with being off by a few meals 92 - 100**
   What is an acceptable range?



1. **More than I want to spend - $110+**

   What does 0 mean?

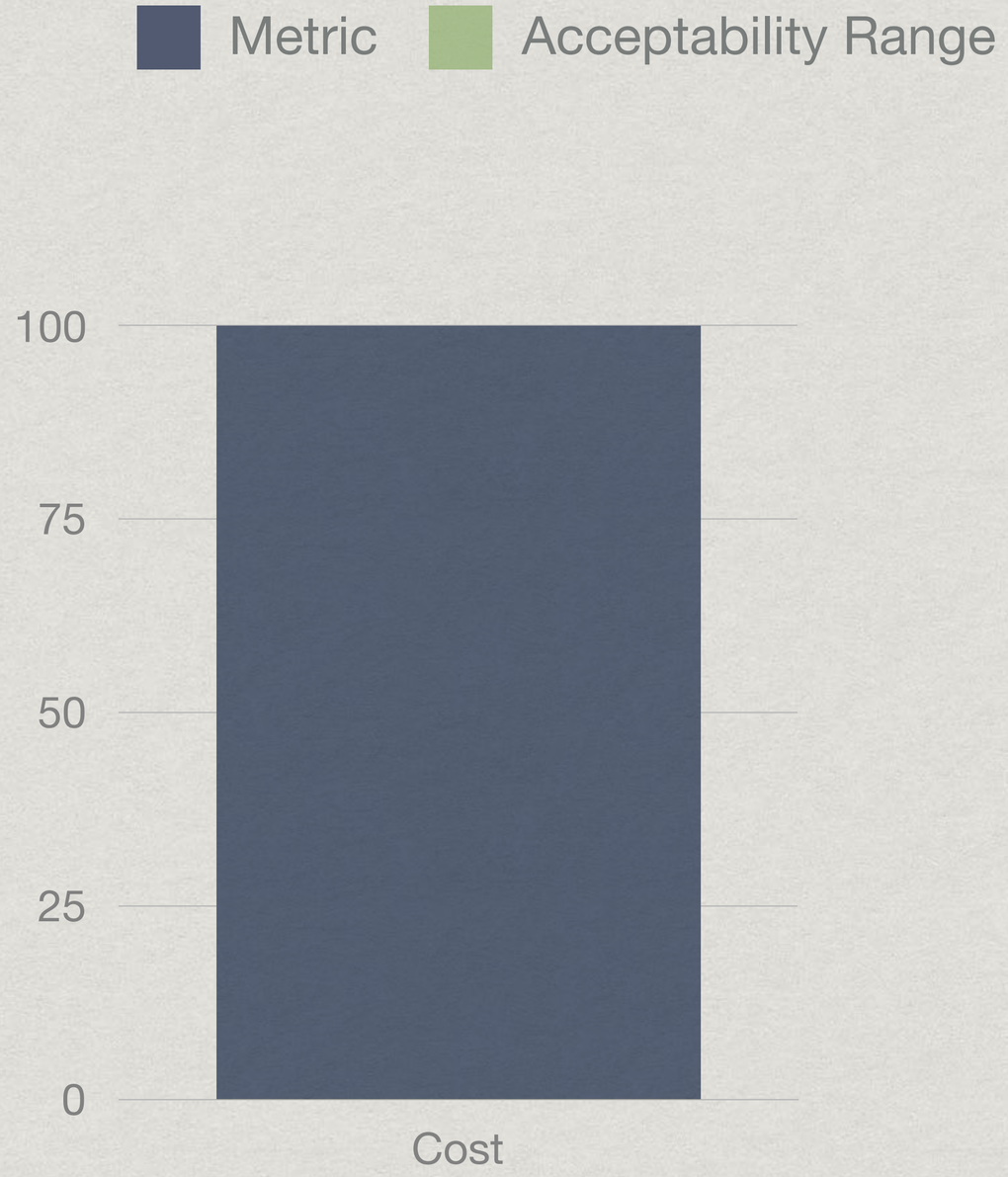


1. **More than I want to spend - $110+**

   What does 0 mean?

2. **Free food!**

   What does 100 mean?



1. **More than I want to spend - $110+**

   What does 0 mean?

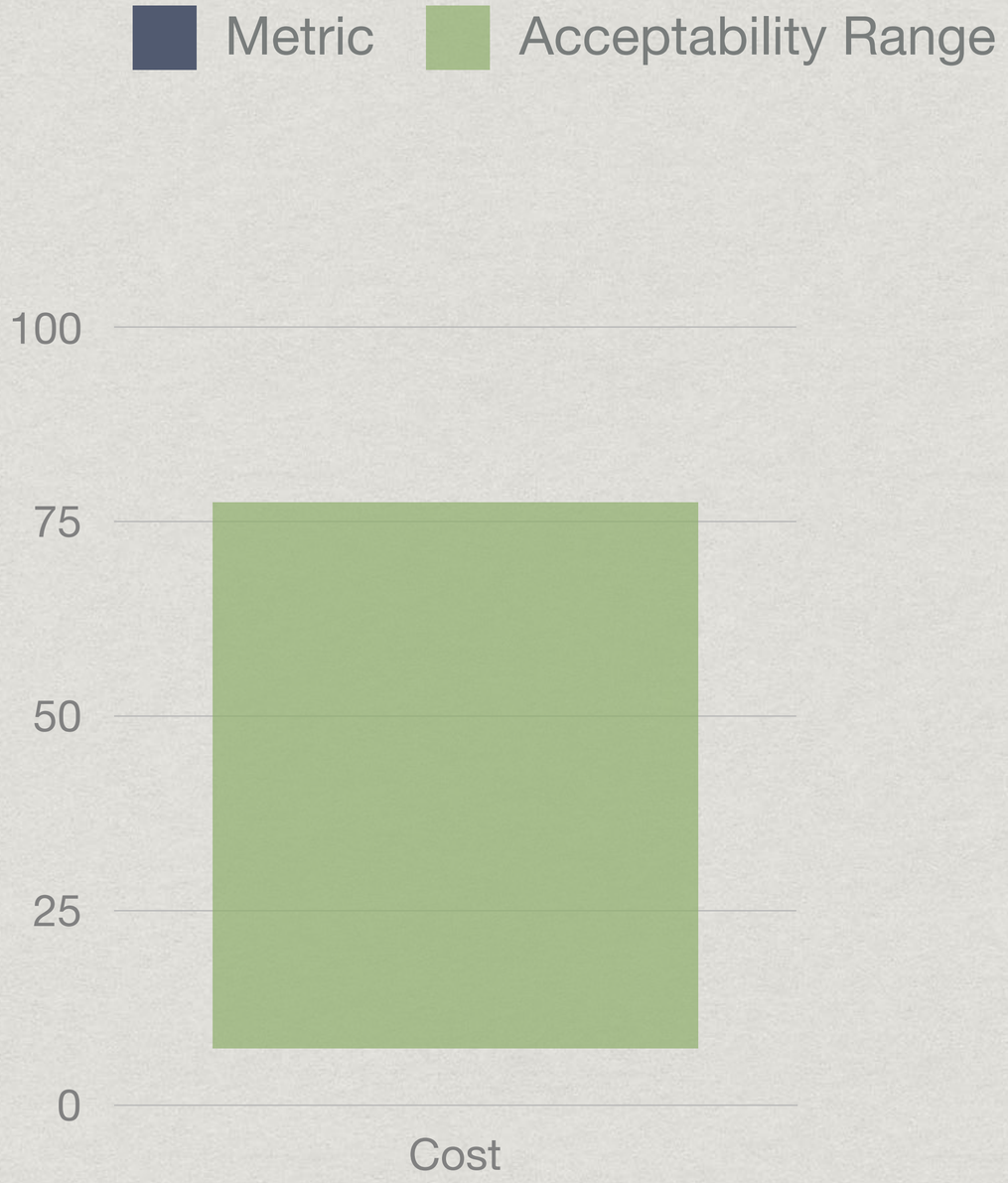
2. **Free food!**

   What does 100 mean?

3. **Less than $100, but not too cheap**

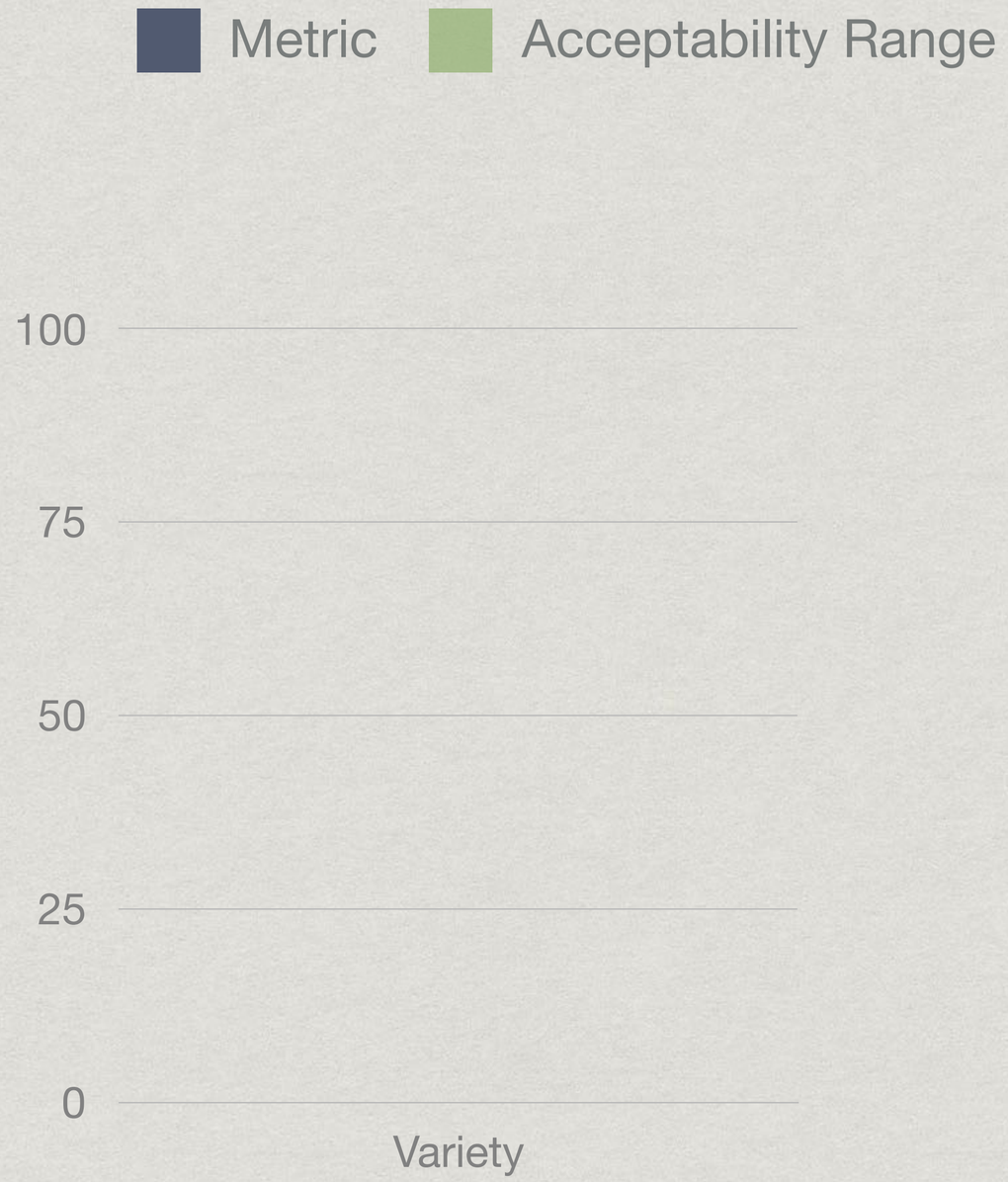
   9 - 80

   What is an acceptable range?



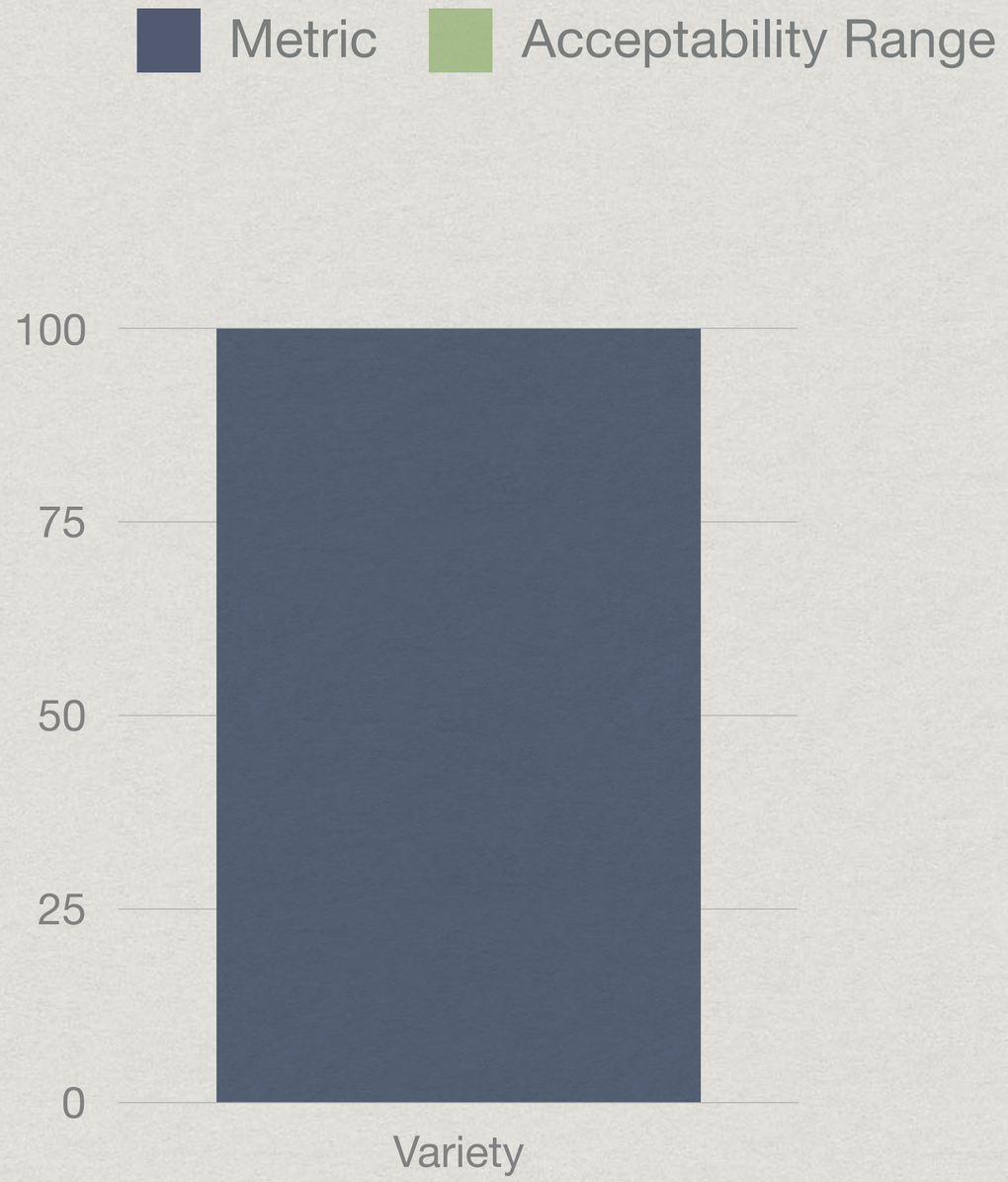
1. All meals are the same

What does 0 mean?



1. **All meals are the same**
   What does 0 mean?

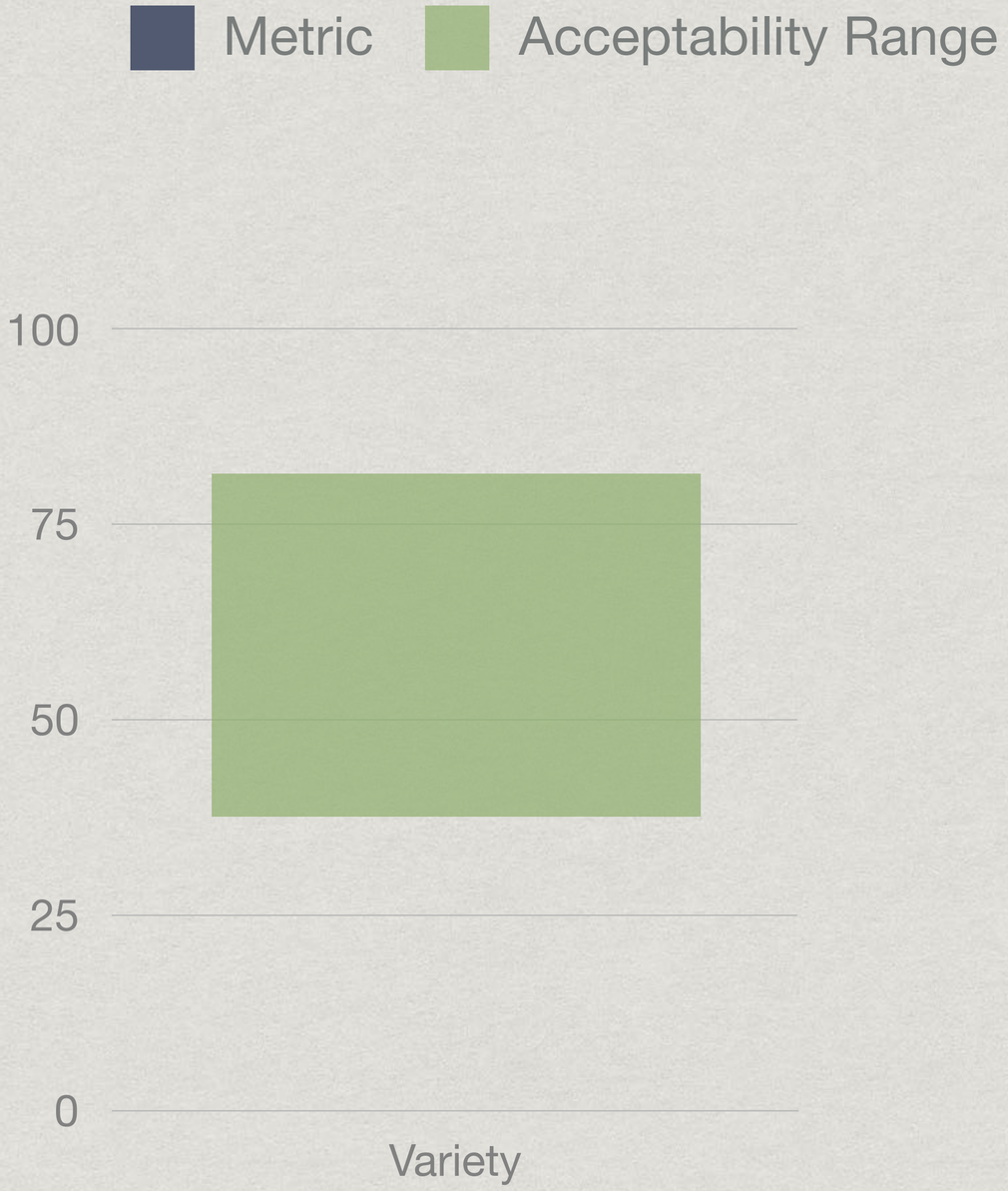
2. **No meals are duplicated**
   What does 100 mean?



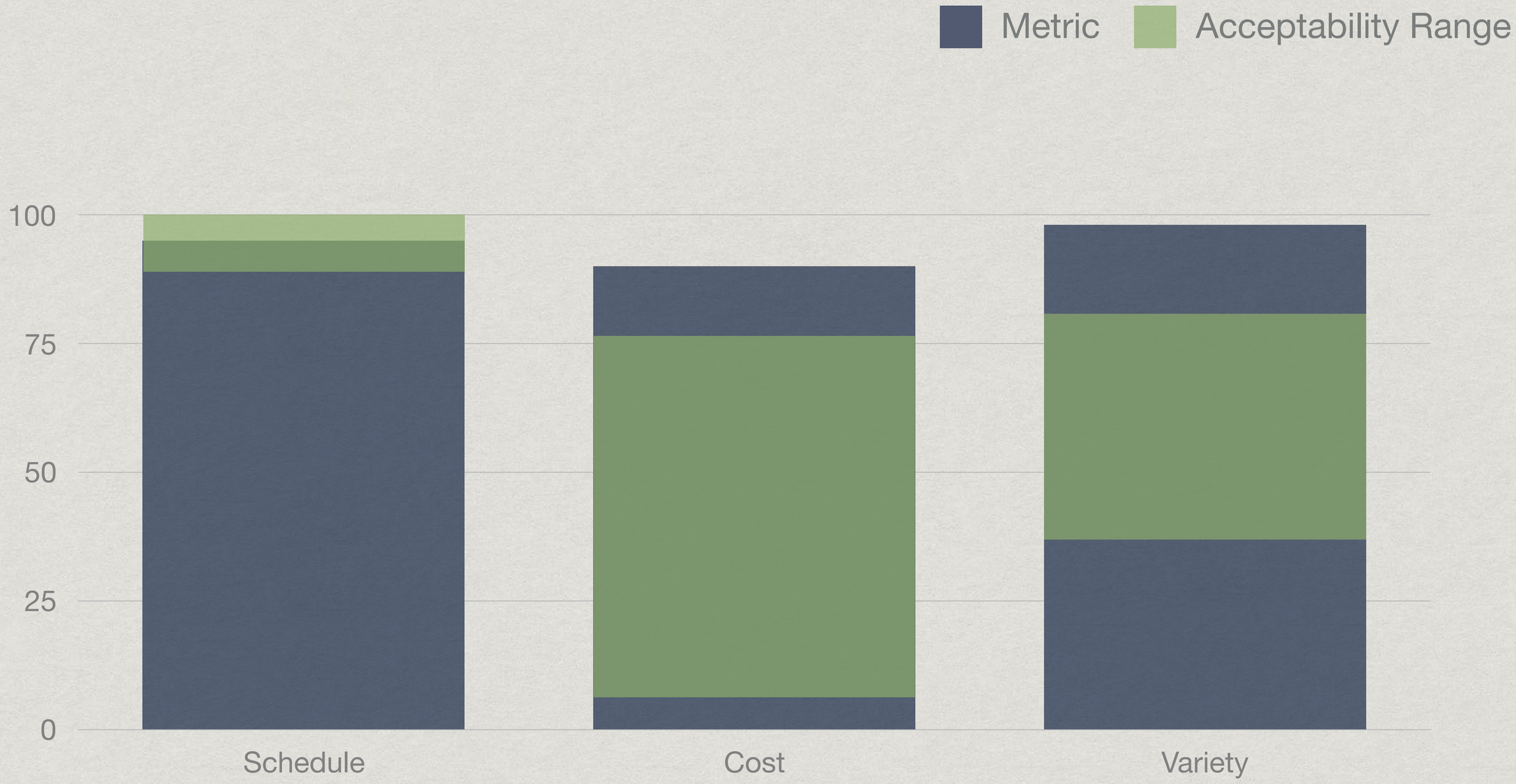
1. **All meals are the same**
   What does 0 mean?

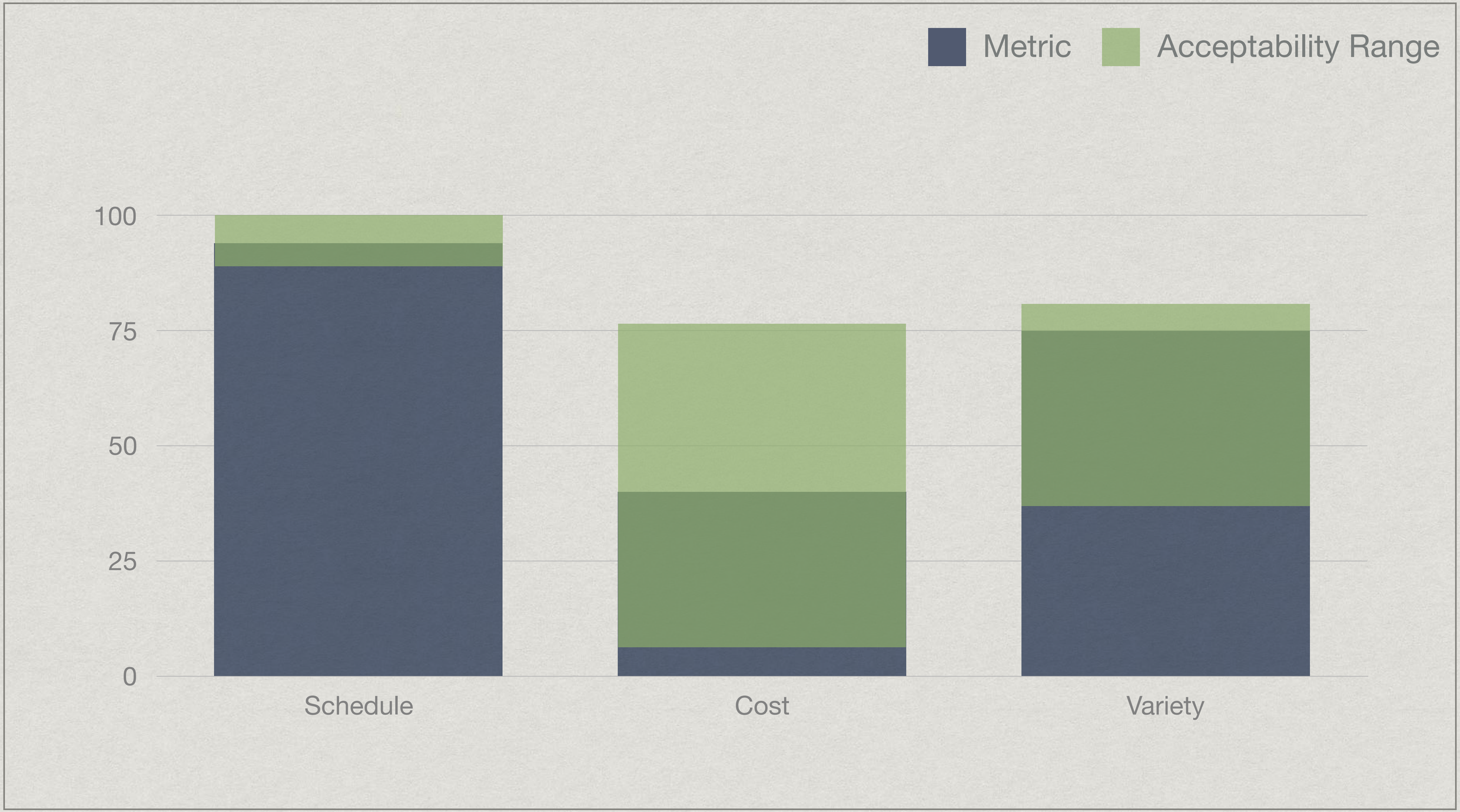
2. **No meals are duplicated**
   What does 100 mean?

3. **“Some variety”**
   37.5 - 80
   What is an acceptable range?




Metric
Acceptability Range
100
75
50
25
0
Schedule
Cost
Variety


Metric
Acceptability Range
100
75
50
25
0
Schedule
Cost
Variety


Metric
Acceptability Range
100
75
50
25
0
Schedule
Cost
Variety

DEMO

NOTEWORTHY

- Genetic Algorithms can solve multi-dimensional problems!
- Carefully modeling your data can be a time saver
- Build in a way to visualize progress

HOW CAN YOU GET STARTED?

- Find a problem that interests you
  - Improve the “Hello world” example (“HELLO WORLD” vs “Ifmmp Xfsme”)
  - Evolve the optimal path for mowing your lawn
  - Conference schedule organizer
- Resources
  - https://github.com/ckoster22/geneticAlgoKcdc2017
  - elm-genetic

THANK YOU!

CHARLIE KOSTER
@CKOSTER22