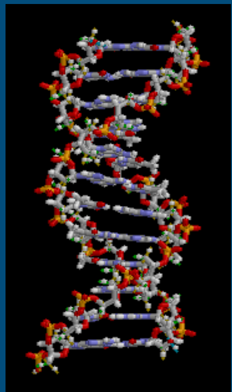
2017 Predictive Analytics Symposium

Session 8, Genetic Algorithms - Why and How to Use Them (workshop)

Moderator:
Stuart Klugman, FSA, CERA, Ph.D.

Presenters:
Brian Charles Grossmiller, FSA, FCA, MAAA
David L. Snell, ASA, MAAA



Session 8: Genetic Algorithms
- why and how to use them

SOA Predictive Modeling Symposium September 14, 2017 - Chicago

David Snell, FALU, FLMI, ASA, MAAA, CLU, ChFC, ARA, ACS, MCP

Brian Grossmiller, FSA, MAAA, FCA

NOTE: the training workbooks can be downloaded at www.GitHub.com/DaveSnell

SOCIETY OF ACTUARIES®

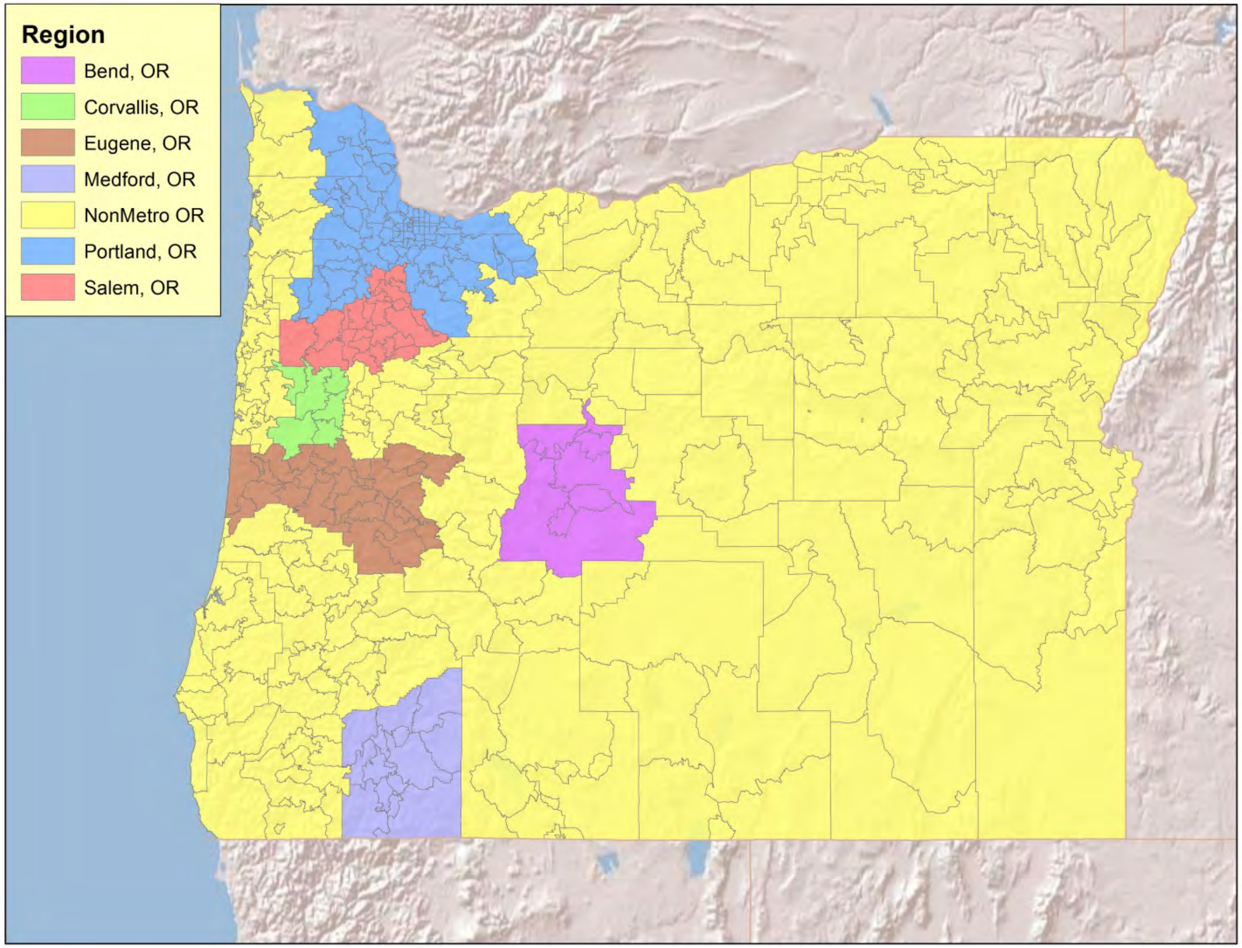
Background and History

- What is a Genetic Algorithm?
- Where did Genetic Algorithms come from?
- How are these being used in other industries?

Problem Solving with Genetic Algorithms

- Why should I use a Genetic Algorithm?
- How do I find a suitable problem?
- What do I need to do to implement a GA?

Example: Personnel Assignments



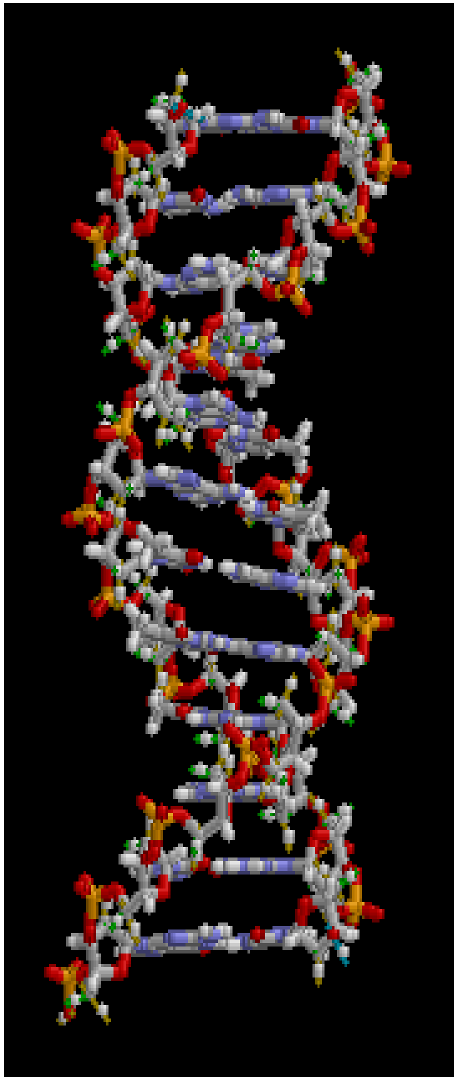
Example: Personnel Assignments

- We have 15 people to assign to 7 areas based on their preferences:

	Area Preference (Ranked 1-7)						
Employee	Portland	Salem	Eugene	Corvallis	Medford	Bend	NonMetro
Laura	2	5	4	1	3	6	7
Renee	6	2	7	4	3	1	5
Timothy	1	7	5	4	3	2	6
Robert	2	7	1	3	5	4	6
Adam	4	3	7	2	1	6	5
Jamie	6	1	5	7	2	3	4
Ana	6	4	5	7	1	3	2
Jessica	1	6	4	2	3	5	7
John	6	5	3	2	7	4	1
Anthony	3	5	1	7	2	4	6
Lisa	3	6	4	5	7	2	1
Donna	3	4	7	1	6	5	2
Claude	1	4	2	5	3	7	6
Deborah	3	6	7	4	1	2	5
David	1	6	4	7	2	3	5

A Genetic Algorithm can be useful for …

- Provider group selection
- Sales representatives and regions
- ERM … beyond CTE
- Stress tests when valuing a block of business
- Traveling Salesperson
- Non-linear equations



Genetic Algorithms

Why do we call these Genetic Algorithms?

They mimic our current knowledge of genetics.

We have trillions of cells.

DNA represents a blueprint for a cell.

It is used to generate copies.

The actual process involves proteins and lots of other biological terms …

and you don't have to know them to solve problems!

Yet another version of the Traveling Salesperson problem … with a few twists:

Distances vary between cities

Some cities are worth more than others to visit

			NE Road Trip						
City		1	2	3	4	5	6	7	8
Weight		1	1	4	2	1	2	2.5	1

City	City Num.	Albany, NY	Augusta, ME	Boston, MA	Bridgeport, CT	Concord, NH	Greewich, CT	Hartford, CT	Montpelier, VT		
Albany, NY	1	0	326	170	156	142	154	111	167		2nd trip to same city = 1/2
Augusta, ME	2	326	0	164	310	153	339	256	192		3rd trip to same city = 1/10
Boston, MA	3	170	164	0	153	68	182	99	179		4th+ trip to same city = 0
Bridgeport, CT	4	156	310	153	0	198	29	56	250		
Concord, NH	5	142	153	68	198	0	227	144	114		max. mileage
Greewich, CT	6	154	339	182	29	227	0	85	279		1000
Hartford, CT	7	111	256	99	56	144	85	0	194		
Montpelier, VT	8	167	192	179	250	114	279	194	0		

					mileage		visits	visit count	visit factor
starting point (evening arrival)	1	value	cost	adjusted value	row col		1	1	1
	1	1	0	1	6 C		1	2	0.75
	7	2.5	111	2.5	12 C		1	3	0.533333333
	6	2	85	1.5	11 I		2	4	0.4
	5	1	227	0.75	10 H		2	5	0.32
	6	2	227	1.5	11 G		2	6	0.266666667
	3	4	182	3	8 H		2	7	0.228571429
	4	2	153	2	9 E		1	8	0.2
	3	4	153	3	8 F		2	9	0.177777778
	5	1	68	0.75	10 E		2	10	0.16
	points	19.5	1206	16					
			mileage penalty	-206		Start Genetic Algorithm for this sample problem			
			total score	-190					

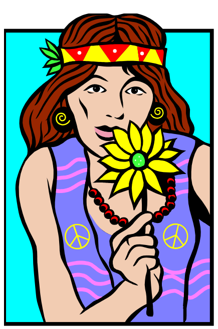
There is a mileage penalty in the scoring algorithm

You can visit a city more than once (but for less credit)

We don't want to exceed 1,000 miles

Criteria that make a problem suitable for a genetic algorithm

- The problem involves a lot of variables - to some extent, the more variables there are, the better this technique applies.
- Each variable can take on potential values to produce different solutions.
- We can substitute a value for each of the variables and that particular combination of individual values can be thought of as a solution set.
- The problem can be quantified in some manner so that any two solution sets can easily be compared to see which is better.



OMG

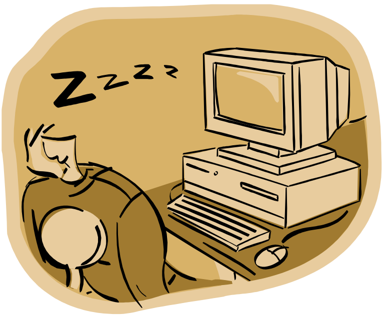
That is so simple!

Mitosis/meiosis
Single nucleotide polymorphisms
Alleles/ phenotypes
Adenine, Cytosine, Thymine, Guanine

Let’s see a real world actuarial example:

- Health Provider Network
- 500 potential providers for this example
- Each provider offers up to 36 specialties
- Each provider and specialty has a relative cost
- You have to provide a sufficient number of practitioners for each of the specialty choices
- You want to minimize overall cost
- More than 10 to the 150th possible solution sets

Try all possibilities?



$2^{500} > 10^{150}$ so trying all possibilities might take too long

… compare to
around 10^{17} seconds since Big Bang billions of years ago

Provider Network Cost Optimization

Each provider group is in (1) or out (0) of network.

5				Specialties:	Chiropractic	Pathology	Cardiovascular Disease	Family Practice	Obstetri
6				Cost	0.97	0.92	0.90	0.89	
7				Count Minimum	5	5	5	20	
8				Current Count	79	42	70	356	
9					125	62	82	597	
10		Total							
11					1	2	3	4	
12	Health System	In	Rela	Total Provider	Chiropractic	Pathology	Cardiovascular Disease	Family Practice	Obstetri
13	Provider # 1	1	0.77	M(G13:AP13)	0	23	24	64	
14	Provider # 2	1	0.90	355	1	12	26	83	
15	Provider # 3	1	0.79	287	0	0	9	66	
16	Provider # 4	0	1.13	228	0	0	0	65	
17	Provider # 5	0	0.89	216	0	0	11	67	
18	Provider # 6	0	1.36	137	0	0	0	0	
19	Provider # 7	0	1.50	129	3	17	0	10	
20	Provider # 8	0	1.32	85	0	0	0	18	
21	Provider # 9	0	1.33	38	0	0	0	0	
22	Provider # 10	0	1.08	37	0	1	0	0	
23	Provider # 11	1	1.04	35	0	0	0	0	
24	Provider # 12	0	0.73	35	0	0	0	0	
25	Provider # 13	0	1.16	34	0	0	0	0	
26	Provider # 14	0	1.32	28	0	0	0	0	
27	Provider # 15	0	1.12	27	0	0	0	4	
28	Provider # 16	1	1.03	27	0	0	0	18	
29	Provider # 17	1	0.78	26	0	0	0	0	
30	Provider # 18	0	1.22	26	0	0	0	0	
31	Provider # 19	1	1.55	25	0	0	0	1	
32	Provider # 20	1	0.84	21	0	0	0	13	

Each provider group can have multiple specialists; and has a relative cost.

500 Providers for this example; but could have thousands. Lots of specialties. Could have 2^500 (> 10^150) solution sets … might take a while by traditional methods. ☺

Provider Network Cost Optimization (continued)

Genetic Algorithm Presentation

Provider Network Fitness Function

Count of Contracts (Provider Groups) Used:	325
Included Providers (Specialists):	2,885
Relativity to Overall Network:	0.8966
Adequate Network:	Yes

Click Here to start genetic algorithm for solution set. You can modify parameters on the Parameters sheet.

Specialty	Available Providers	Required Providers	Selected Providers	Requirement Met	Relativity	Specialty Weight
Hospital	16	5	11	Yes	0.89	47.1%
Family Practice	597	20	438	Yes	0.90	7.7%
Physical Therapy	506	5	243	Yes	1.00	3.9%
Internal Medicine	376	20	296	Yes	0.89	3.8%
Obstetrics/Gynecology	277	5	195	Yes	0.88	3.8%
Pediatrics	351	5	249	Yes	0.95	3.4%
Orthopedic Surgery	147	5	100	Yes	0.88	3.2%
Hematology /Oncology	97	5	58	Yes	0.86	2.8%
Chiropractic	125	5	87	Yes	0.98	2.7%
Diagnostic Radiology	174	5	101	Yes	0.87	2.5%
Dermatology	61	5	47	Yes	0.81	2.1%
Ophthalmology	120	5	111	Yes	0.86	1.3%
Otolaryngology	52	5	45	Yes	0.82	1.3%
Gastroenterology	40	5	34	Yes	0.89	1.2%
Pathology	62	5	41	Yes	0.90	1.1%
Podiatry	44	5	32	Yes	1.01	1.0%
Acupuncturist	65	5	39	Yes	0.94	0.9%
Urology	44	5	32	Yes	0.93	0.9%
General Surgery	65	5	46	Yes	0.84	0.8%
Rheumatology	21	5	16	Yes	0.86	0.8%
Neurology	94	5	86	Yes	0.91	0.8%

Instructions | Summary | Provider_List | DNA | Parameters

Each specialty area must have adequate coverage.

Some problems just don't fit well into classical methods of solution:

Assume you have three equations:

- $y_1 = a * e * g + h + d^a$
- $y_2 = |h|! - |d|!$
- $y_3 = ((\sin(a)) + b) * \log(b + c))$
  $+ \cos(\min(c, d)) * (e - f + g * h)$

Find a combination of *a, b, c, d, e, f, g, h* such that the standard deviation of y_1, y_2, and y_3 is minimized.

Oh yeah!
We are math folks, so this might be too easy by itself!

Let's add some constraints to make it more interesting!

a has to be an integer from 1 to 10
b is a real number from 0 to 15
c is a real number from 1 to 3
d is a real number from 0.5 to 7
e is a real number from -10 to 50
f is an **even** integer from -20 to 40
g is a real number from 0 to 18
h is a real number from 3 to 12

How to attack a really monstrous problem (continued … expressed as an Excel sheet)

	A	Variable Value	ValueOK	Min	Max	Int?	Conditions
5	a	5.00	TRUE	1	10	Y	=IF(F5="Y",IF(AND(varA>=D5,varA<=E5,varA=INT(varA)),TRUE,FALSE),IF(AND(varA>=D5,varA<=E5),TRUE,FALSE))
6	b	7.50	TRUE	0	15	N	=IF(F6="Y",IF(AND(varB>=D6,varB<=E6,varB=INT(varB)),TRUE,FALSE),IF(AND(varB>=D6,varB<=E6),TRUE,FALSE))
7	c	2.00	TRUE	1	3	N	=IF(F7="Y",IF(AND(varC>=D7,varC<=E7,varC=INT(varC)),TRUE,FALSE),IF(AND(varC>=D7,varC<=E7),TRUE,FALSE))
8	d	3.00	TRUE	0.5	7	N	=IF(F8="Y",IF(AND(varD>=D8,varD<=E8,varD=INT(varD)),TRUE,FALSE),IF(AND(varD>=D8,varD<=E8),TRUE,FALSE))
9	e	20.00	TRUE	-10	50	N	=IF(F9="Y",IF(AND(varE>=D9,varE<=E9,varE=INT(varE)),TRUE,FALSE),IF(AND(varE>=D9,varE<=E9),TRUE,FALSE))
10	f	30.00	TRUE	-20	40	Y	=IF(F10="Y",IF(AND(varF>=D10,varF<=E10,varF=INT(varF),MOD(varF,2)=0),TRUE,FALSE),IF(AND(varF>=D10,varF<=E10),TRUE,FALSE))
11	g	9.00	TRUE	0	18	N	=IF(F11="Y",IF(AND(varG>=D11,varG<=E11,varG=INT(varG)),TRUE,FALSE),IF(AND(varG>=D11,varG<=E11),TRUE,FALSE))
12	h	8.00	TRUE	3	12	N	=IF(F12="Y",IF(AND(varH>=D12,varH<=E12,varH=INT(varH)),TRUE,FALSE),IF(AND(varH>=D12,varH<=E12),TRUE,FALSE))

use ctrl ~ to toggle formula view

	G	H
16	1,151.00	=varA*varE*varG+varH+varD^varA
17	40,314.00	=FACT(ABS(varH))-FACT(ABS(varD))
18	(19.41)	=((SIN(varA))+varB)*LOG(varB+varC)+COS(MIN(varC,varD))*(varE-varF+varG*varH)
19	22,956.10	<== we want to minimize the standard deviation of the three results above

Click here to try an extended form of the Genetic algorithm program

This is an example of a problem that the Excel Solver cannot handle. We have three equations above (in cells G16:G18) and we wish to minimize the standard deviation of them in cell G19.

The constraints on input variables appear in table range C5:E12 where each of the eight variables has an allowable minimum, maximum, and type (integer or real). Note that variable f must also satisfy the condition that it is an even integer. This type of constraint is difficult, at best, to work into the Solver.

samples to try manually:

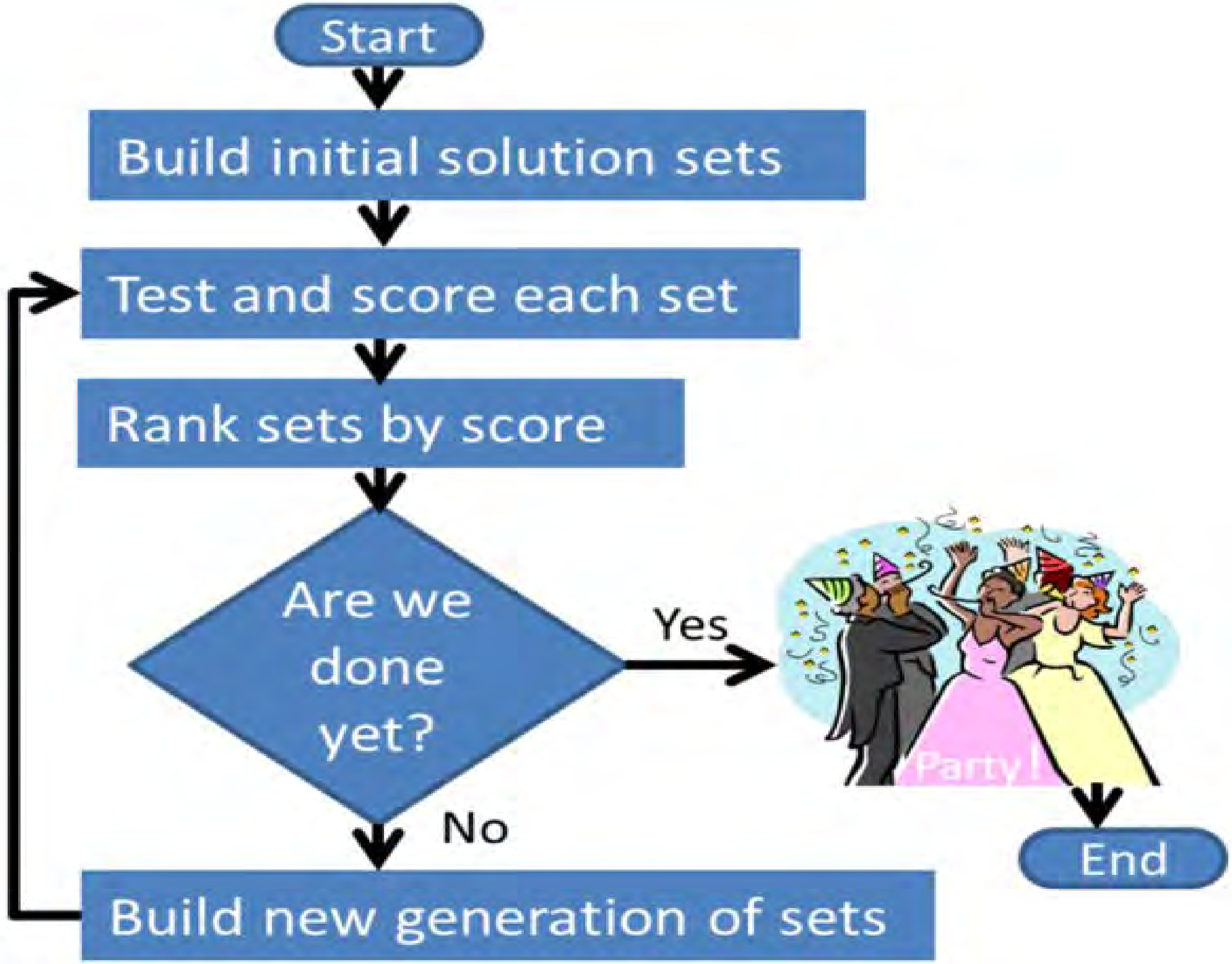
28	a	5.00
29	b	7.50
30	c	2.00

Region_Staffing | Instructions | Default_Parameters | DNA | SalesTrip | EquationSolver

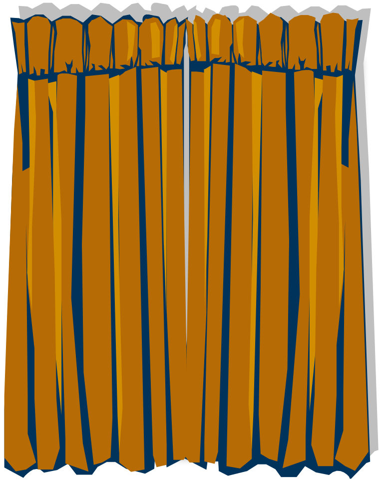
Looking Behind the Curtain

- Stepping through the code for our examples
- Learning to ~~fish~~ evolve

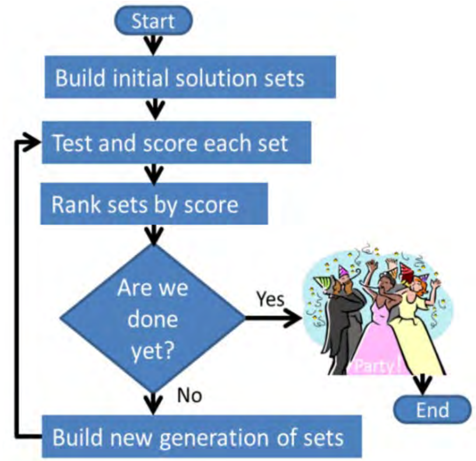
How to build a genetic algorithm



Understanding what is behind the curtain

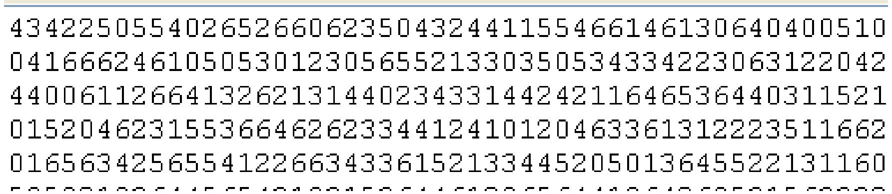


1. Populate a collection of possible solution sets.
2. Test each set of the collection and save the scores obtained.
3. Rank the scores.
4. Build the successive collection (generation) of solution sets.
5. Repeat steps 2 thru 4 until done.



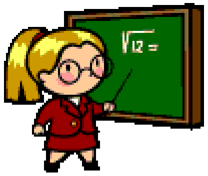
Genetic Algorithms (a simplistic recipe)

1. Randomly assign the gene string for N solution sets (Generation 1)



2. Test each solution and rank the scores.
3. Assign mating privileges to survivors per test scores (survival of the fittest).
4. Build a new generation and repeat the process.

In a kangaroo mob, only the highest ranking male gets mating privileges



Genetic Algorithms
(How twins are made)

… and some don't fare as well

Genetic Algorithms (simplistic crossover)

Pick parents, and pair them off to produce the next generation

Parent M (52.80 points): 54335351253404 315142153601520652551511513145155663
Parent F (45.60 points) :54335351525534 633242153604216362551511503145155625

arbitrary (pseudo-random) split point

Child A: $\underbrace{\text{54335351253404}}_{\text{genes from parent M}}$ $\underbrace{\text{633242153604216362551511503145155625}}_{\text{genes from parent F}}$

Child B: 54335351525534 315142153601520652551511513145155663

Genetic Algorithms are not limited to two parents (or even four grandparents)

Potential Parent	Gene 1	Gene 2	Gene 3	...	Gene ω
1	A	as	1		X
2	B	rt	0		Y
3	C	gh	0		Y
4	D	iu	1		Y
5	E	mn	1		X
6	F	iu	0		X
7	G	ew	1		Y
8	H	t6	1		X
9	I	u8	0		Y
...					
N	Z	9m	1		X
Child	C	t6	1	...	X
parent:	3	8	7		6

Your 'genes' can have vastly different characteristics and components.

A 'child' can have genes drawn from several different parents.

VBA example code

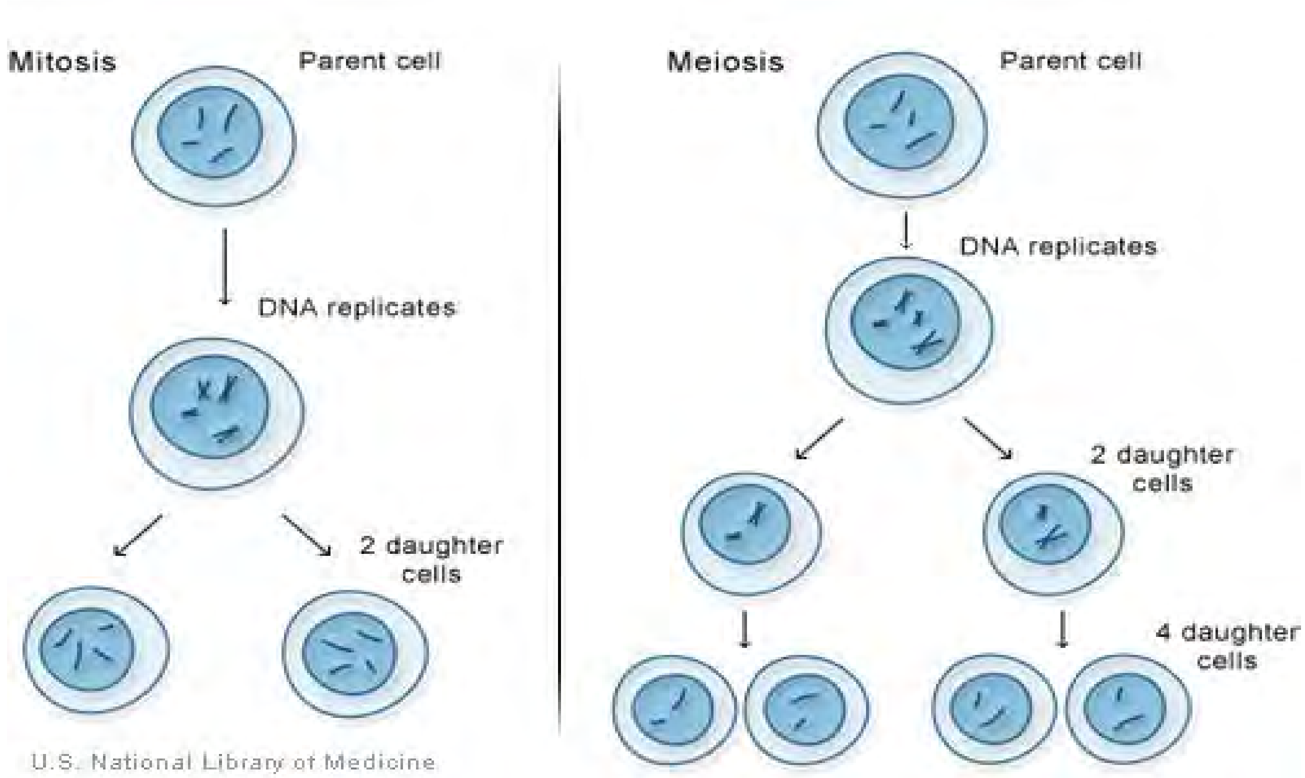
from elsewhere: elites = 20
setsPerGeneration = 100
parentPool = 40
solutionSets is a 2-dimensional array 30 by100

```
  Private Sub AddTheChildren()
  Dim parent As Integer, var As Long, child As Integer, children As
  Integer
1   children = setsPerGeneration - elites        (80 = 100 – 20)
2     For child = 1 To children                  (start with child set 1)
3       For var = 1 To setLength                 (1 to 30 if 30 variables per set)
4         parent = Int(parentPool * Rnd()) + 1   (e.g. parent 5 wins for variable 17 for child 1)
5         solutionSets(var, elites + child)
                       = solutionSets(var, parent)
6       Next var
7     Next child
  End Sub 'AddTheChildren
```

(so set variable 17 in new solution set 21 = 20 +1 to the value from variable 17 in old solution set 5)

Stuff we don't need to know!

Mitosis & Meiosis



Our Elites correspond to Mitosis

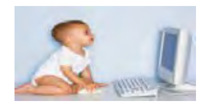
The rest of the new generation are analogous to Meiosis

Stuff we don't need to know!

Meiosis in action - Genetics 0.001








Gene	Parent 'M'		Parent 'F'		Child1		Child2		Child3		Child4	
1	Dad	Mom	Dad	Mom	Dad's Dad	Mom's Dad	Dad's Mom	Mom's Mom	Dad's Dad	Mom's Mom	Dad's Mom	Mom's Dad
2	Dad	Mom	Dad	Mom	Dad's Mom	Mom's Mom	Dad's Mom	Mom's Dad	Dad's Mom	Mom's Mom	Dad's Mom	Mom's Dad
3	Dad	Mom	Dad	Mom	Dad's Dad	Mom's Mom	Dad's Mom	Mom's Mom	Dad's Mom	Mom's Mom	Dad's Mom	Mom's Dad
4	Dad	Mom	Dad	Mom	Dad's Dad	Mom's Dad	Dad's Dad	Mom's Mom	Dad's Mom	Mom's Mom	Dad's Mom	Mom's Mom
...												
5,000	Dad	Mom	Dad	Mom	Dad's Mom	Mom's Mom	Dad's Dad	Mom's Mom	Dad's Dad	Mom's Mom	Dad's Mom	Mom's Mom
5001	Dad	Mom	Dad	Mom	Dad's Mom	Mom's Dad	Dad's Mom	Mom's Dad	Dad's Mom	Mom's Dad	Dad's Dad	Mom's Dad
...												

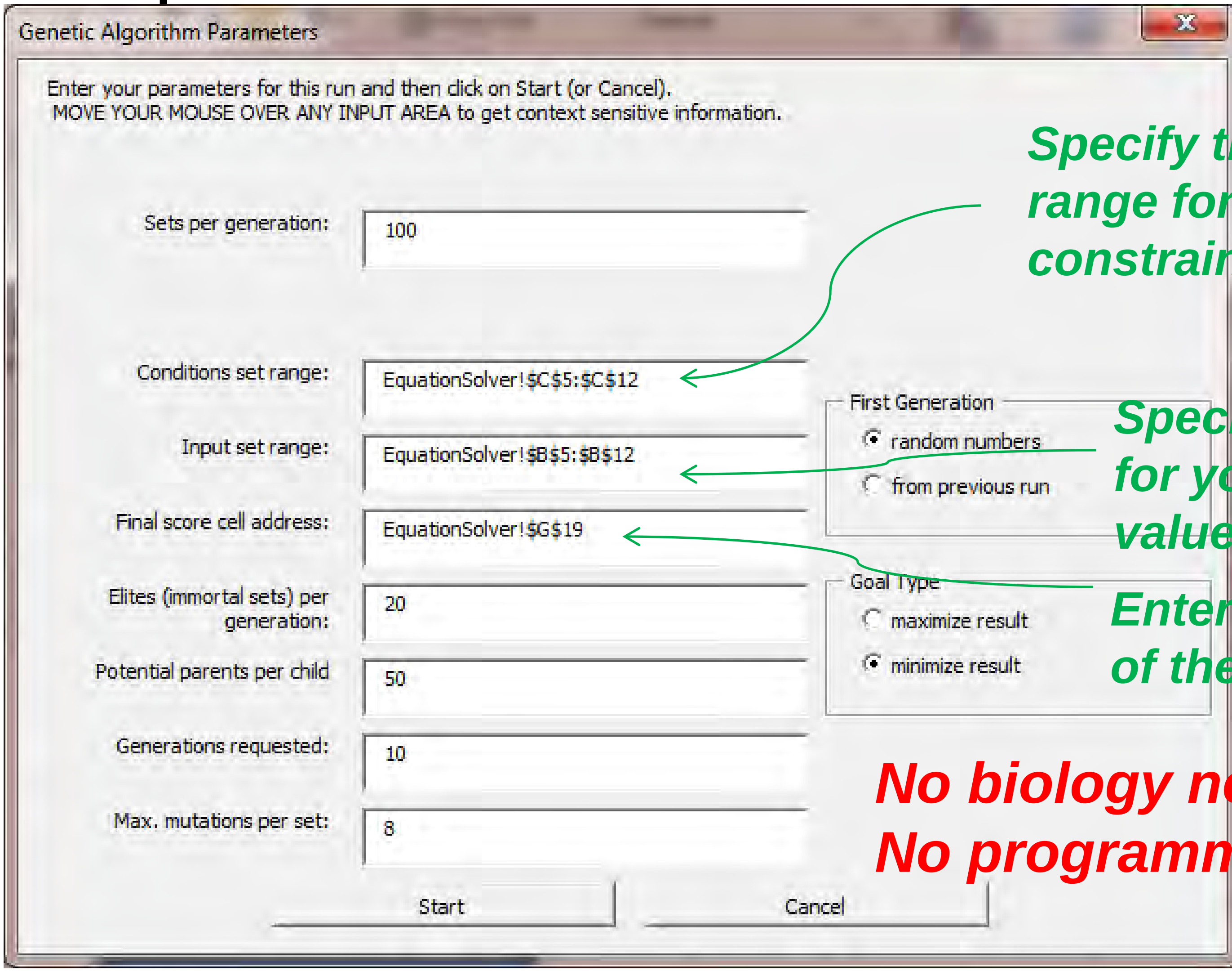
Note that each child gene is from two of the four grandparents: one on Mom's side, one on Dad's side (excluding mutations)

set of genes from Dad

set of genes from Mom

Parent genes prior to meiosis – each person gets one set from Dad and another set from Mom

Input Screen for *FREE* workbook



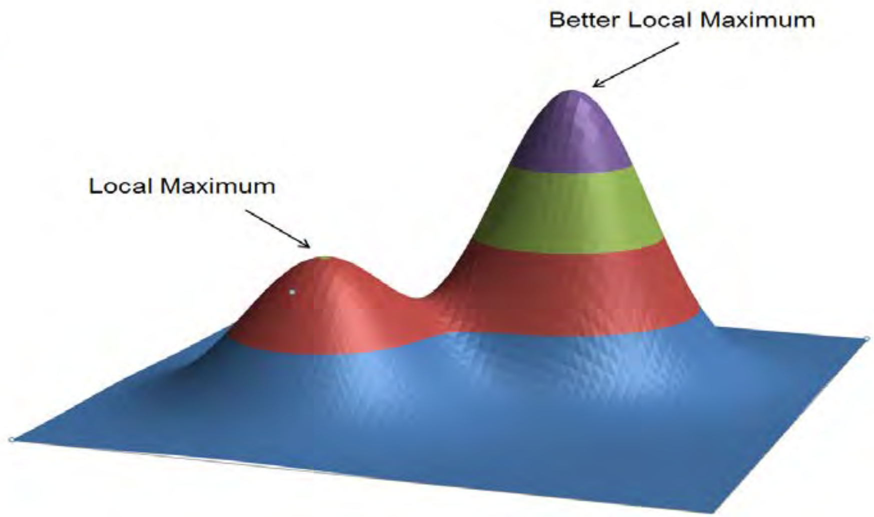
Specify the sheet and range for your constraint criteria.

Specify the range for your variable values

Enter the location of the final score

No biology needed!
No programming needed!

Hill Climbing … works really well when there is only one hill



Using an extended parent pool and increased mutation rate, a genetic algorithm can jump away from a local maximum

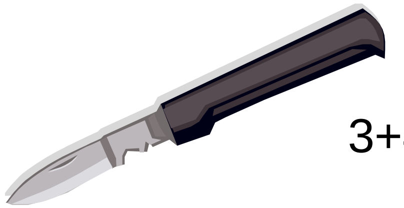
Our family in Tibet

Yak, yak, yak

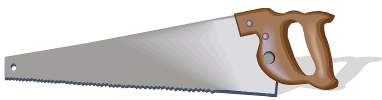
Recap – what did we learn?

- Genetic algorithms can be useful in many diverse types of situations.
- You don’t need to be a math, genetics, or stats wiz or a programmer to understand how to make one.
- You can use the free tool to do a lot of learning just with Excel.
- This is the tip of the iceberg. Join the Predictive Analytics & Futurism section and tap into a cornucopia of new tools and techniques.

New Tools Require New Skills

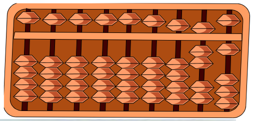


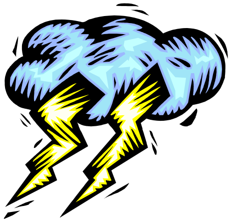
$3+4=4+3$



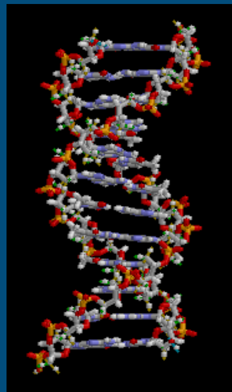
$3-4 \neq 4-3$

Find this video on YouTube via search term ChainsawAlton (one word).





Complexity Sciences



Session 8: Genetic Algorithms - why and how to use them

SOA Predictive Modeling Symposium September 14, 2017 - Chicago

David Snell, FALU, FLMI, ASA, MAAA, CLU, ChFC, ARA, ACS, MCP

Brian Grossmiller, FSA, MAAA, FCA

NOTE: the training workbooks can be downloaded at www.GitHub.com/DaveSnell

SOCIETY OF ACTUARIES®