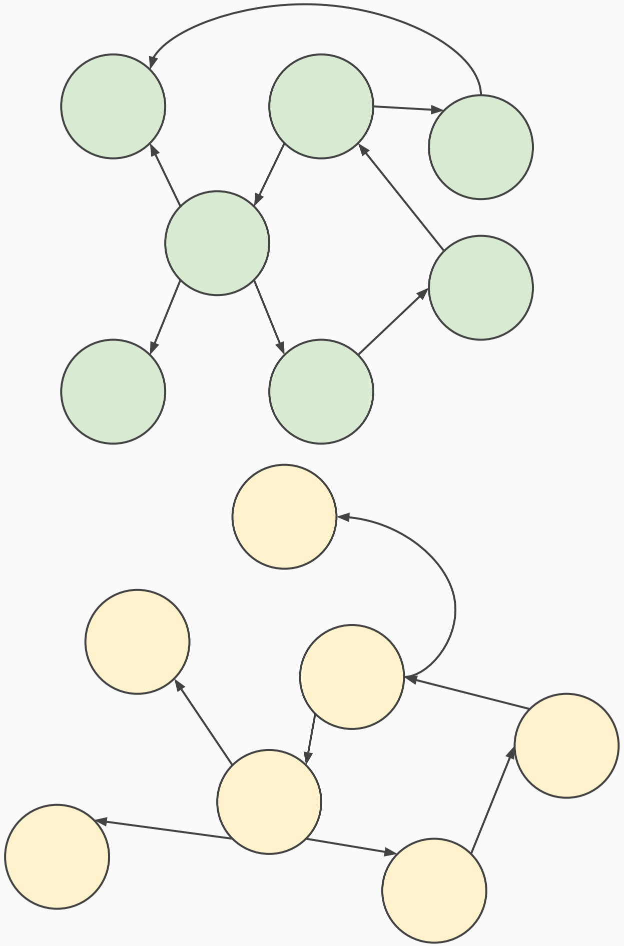
LEARNING BAYESIAN NETWORKS WITH EVOLUTIONARY PROGRAMMING

Tommy Moawad

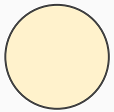
PROBLEM:

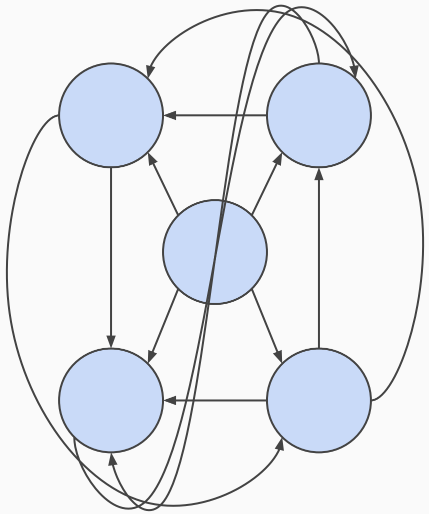
- WHICH NETWORK IS THE BEST?
- WHICH NODES ARE PARENTS OF OTHERS?

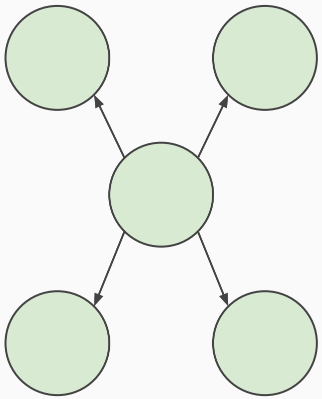


Fitness of a Bayesian Network

Suppose We Want To Determine The Relationship Between 5 Variables







FITNESS OF A BAYESIAN NETWORK: **MINIMUM DESCRIPTION LENGTH**

Let

- D = data
- h = hypothesis
- Huff(x) = Size of huffman encoding of x

MDL = Max [Probability that this is the correct hypothesis for the data]

$= \text{argmax}_h\ P(h|D)$

$= \text{argmax}_h$ Likelihood * Prior

$= \text{argmax}_h\ P(D|h) * P(h)$

$= \text{argmax}_h\ [\ \log_2 P(D|H) + \log_2(h)\ ]$

$= \text{argmin}_h\ [\ -\log_2 P(D|H) - \log_2(h)\ ]$

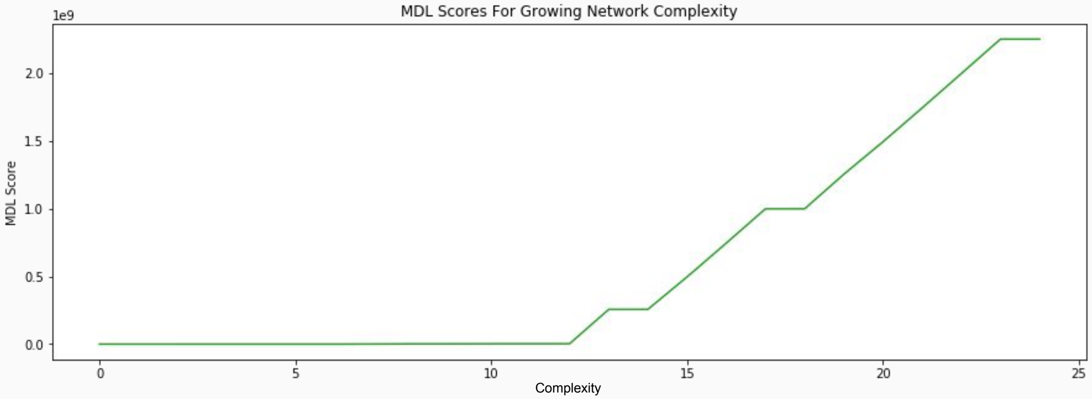
$= \text{argmin}_h[\ \text{length}(D|h) + \text{length}(h)\ ]$ (Information Theory: Optimal Code Length)

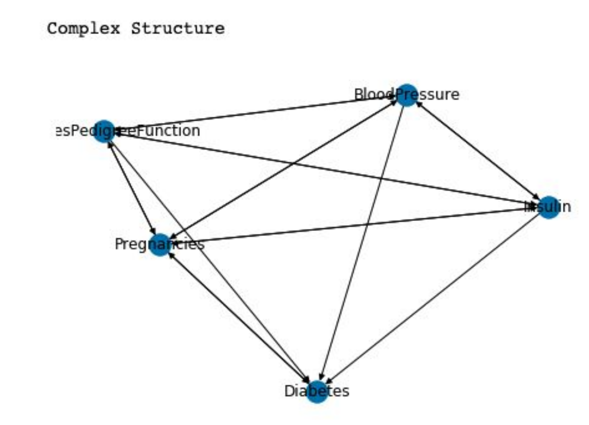
$= \text{argmin}_h$ [Huff(Data encoded with network probabilities) + Huff(network)]

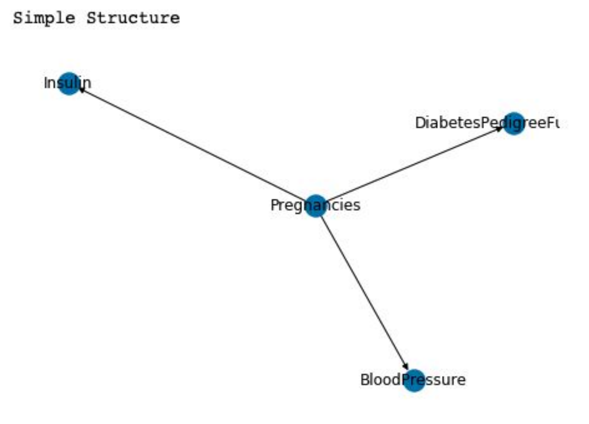
DIABETES DATASET

- Pregnancies
- Glucose
- Blood Pressure
- Skin Thickness
- Insulin
- BMI
- Diabetes Pedigree Function
- Age
- Diabetes

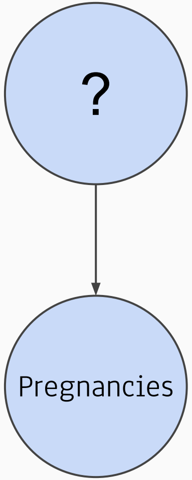
MDL PREFERS SIMPLE NETWORKS

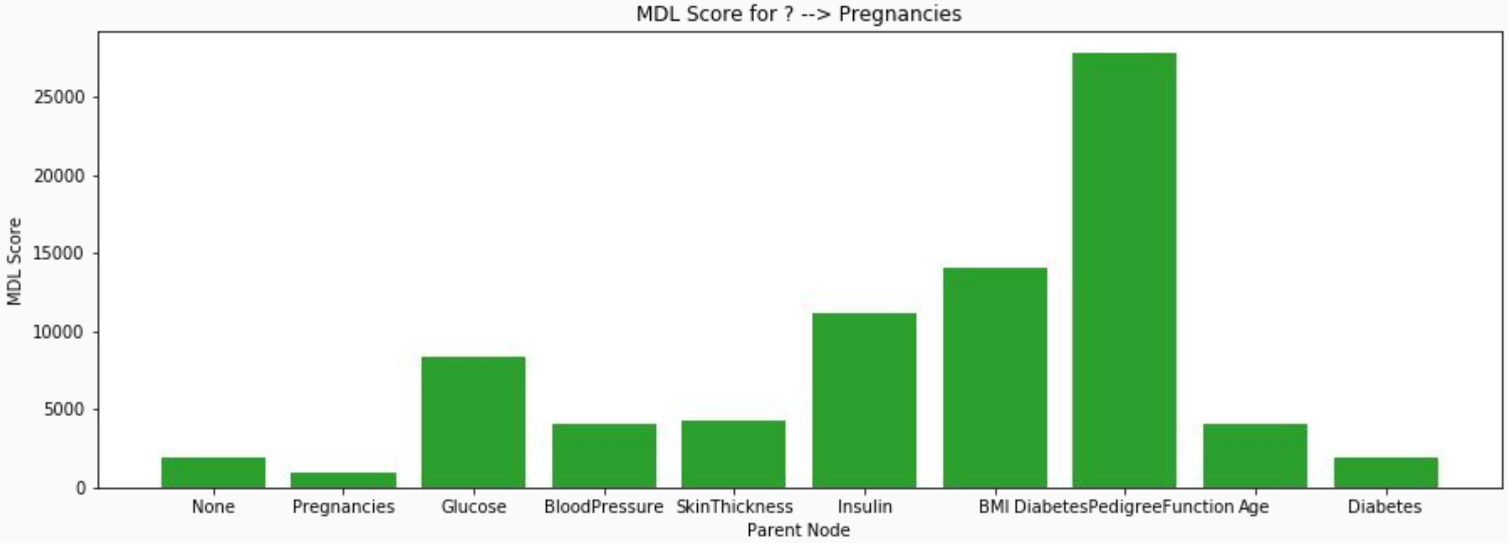




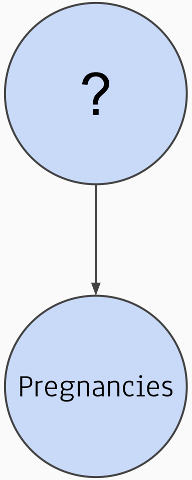


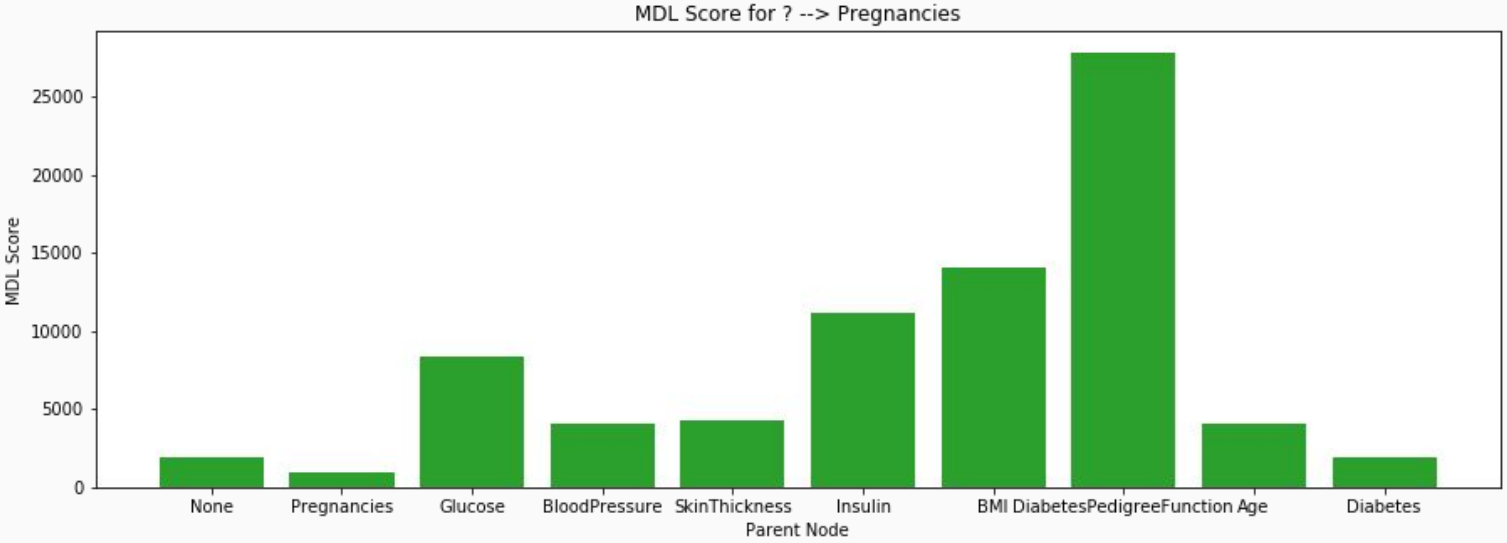
MDL FITS OUR INTUITION





MDL FITS OUR INTUITION

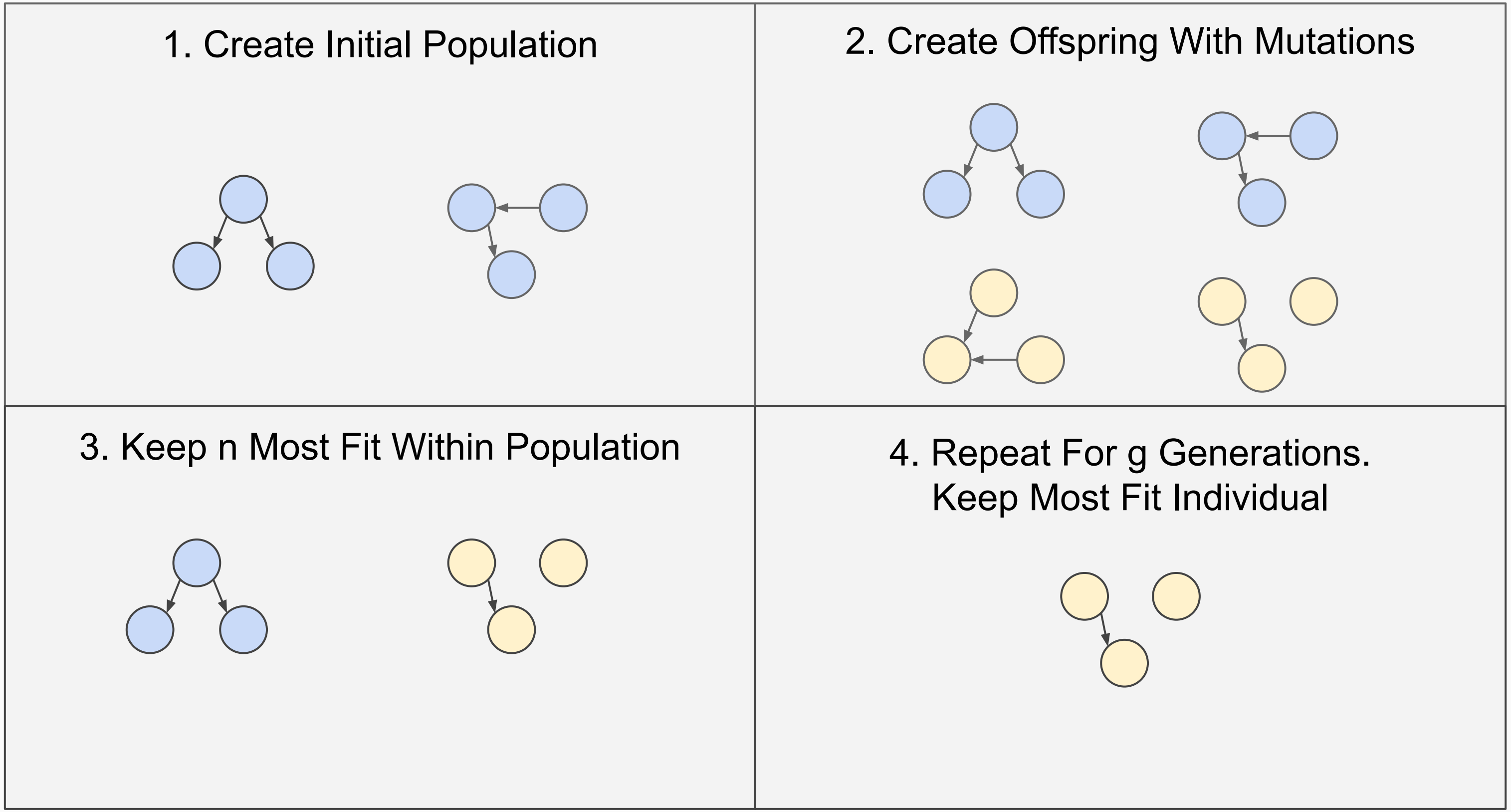




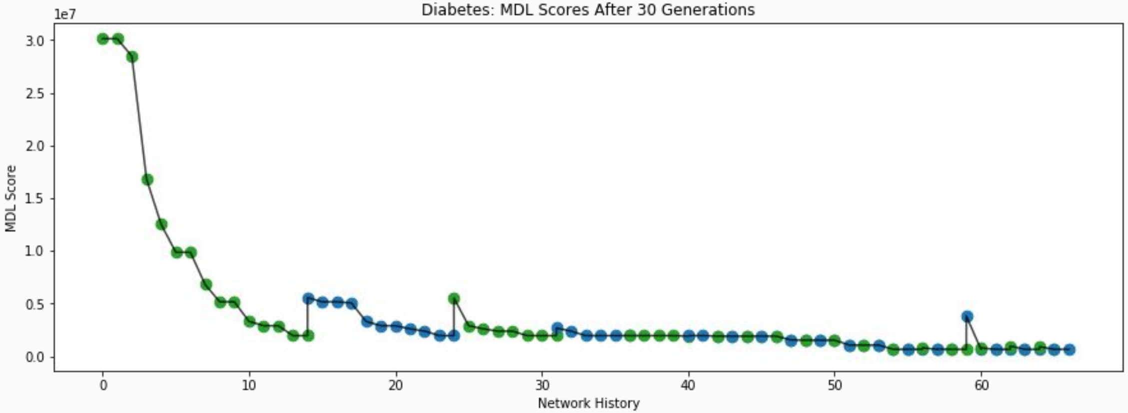
Possible sets of parents: $\binom{9}{0} + \binom{9}{1} + \ldots + \binom{9}{9}$

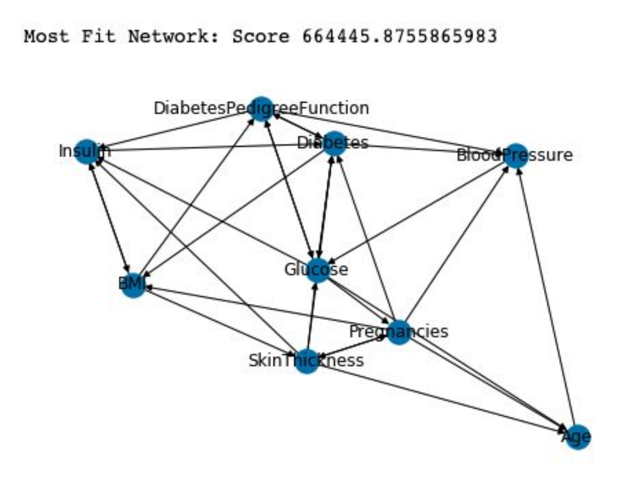
Possible Networks: $2^{9*9} = 2^{81}$

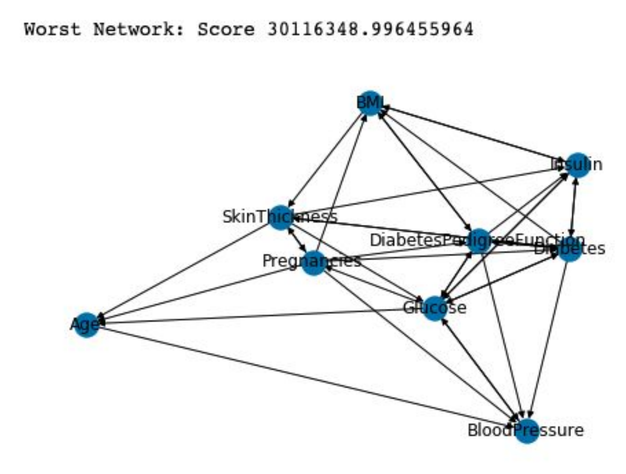
EVOLUTIONARY PROGRAMMING SIMULATES SURVIVAL OF THE FITTEST



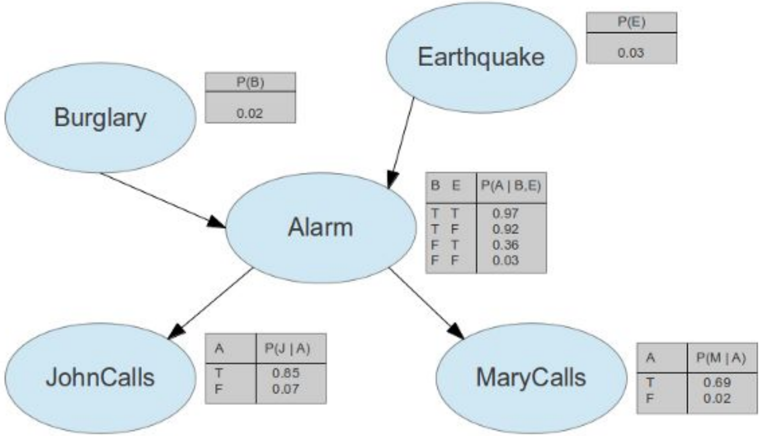
Diabetes Results







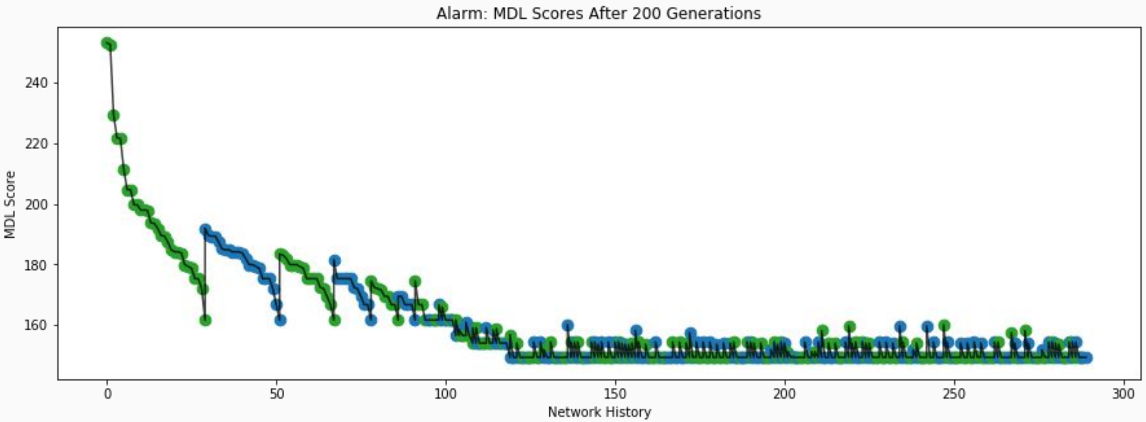
ALARM EXAMPLE

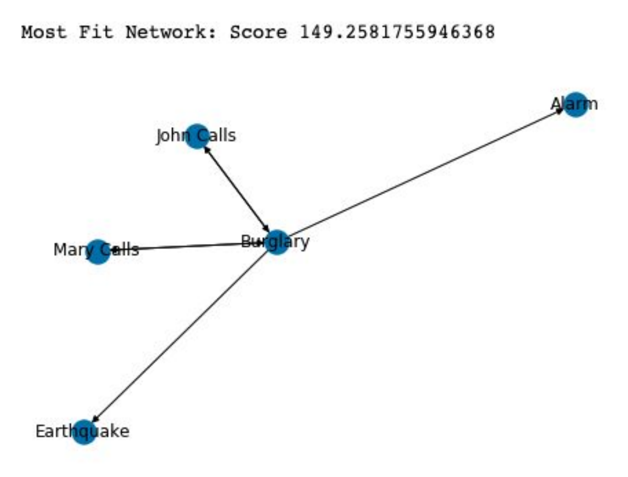


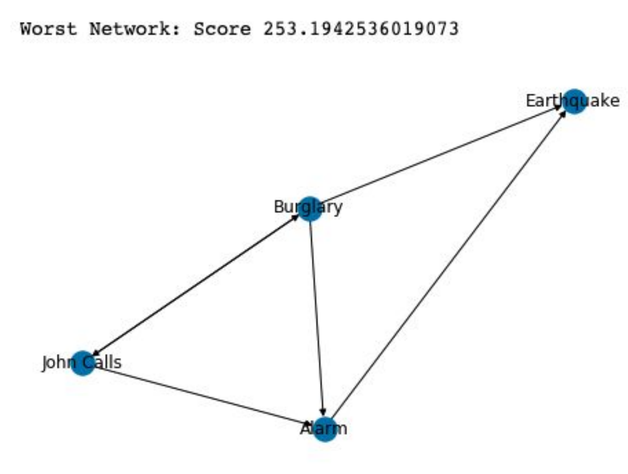

```
alarm.sample(10)
```

	Burglary	Earthquake	Alarm	John Calls	Mary Calls
51	True	False	False	True	True
146	False	False	False	False	False
62	False	False	False	False	False
52	True	False	True	True	False
188	False	False	False	False	False
159	False	False	False	False	False
57	False	False	False	False	False
119	False	False	False	False	False
97	False	False	False	False	False
20	False	False	False	False	False

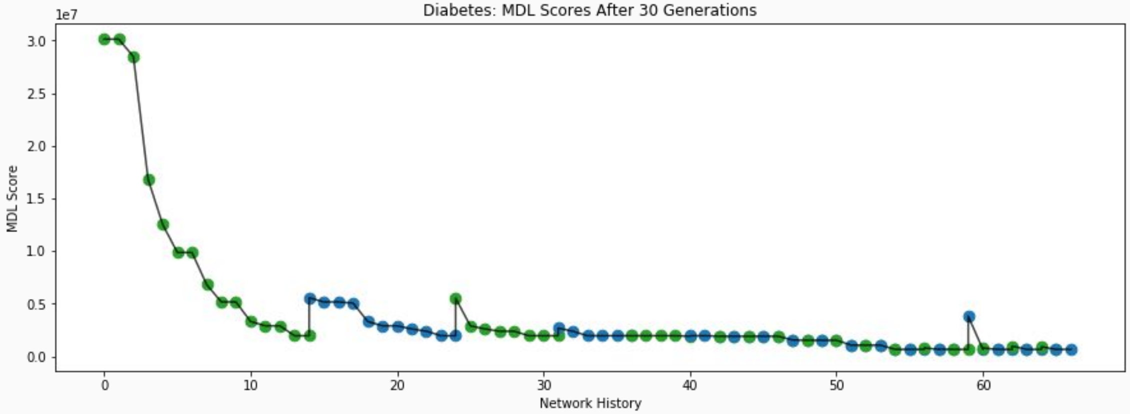
Alarm Results

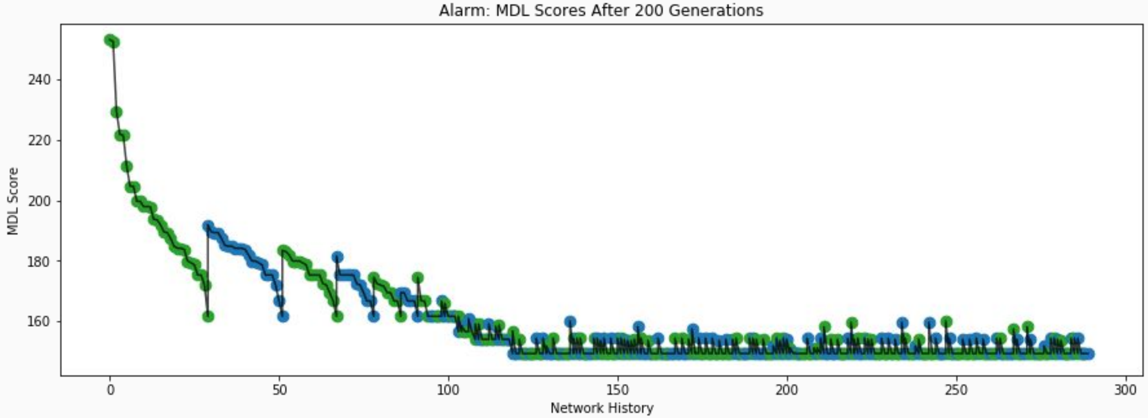






CONVERGES TO REDUNDANT NETWORKS

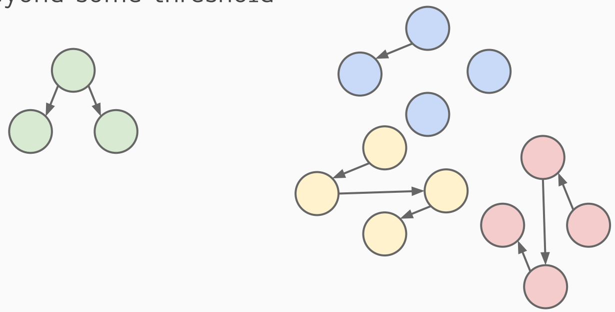




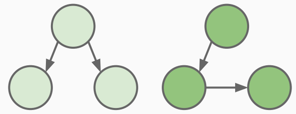
MODIFICATIONS

Favor Diversity with Increase Routine

If average distance between offspring_i and all other nodes is beyond some threshold



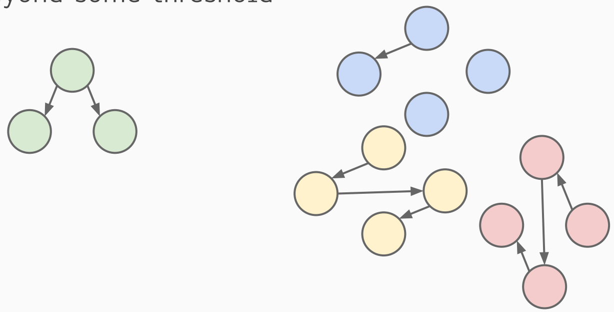
Then keep offspring_i and parent_i



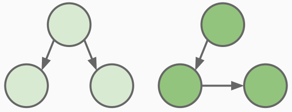
MODIFICATIONS

Favor Diversity with Increase Routine

If average distance between offspring$_i$ and all other nodes is beyond some threshold



Then keep offspring$_i$ and parent$_i$

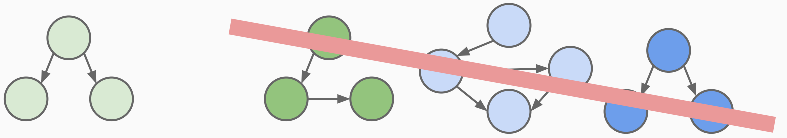


Avoid Redundancy with Decrease Routine

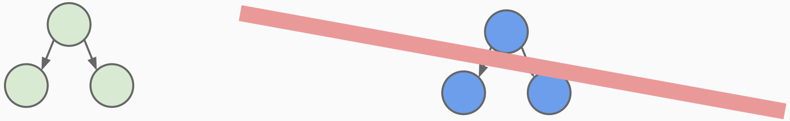
For every combination of network$_i$ and network

and their parents

If the children are fitter than their parents then keep fitter child



Else if the top two networks are identical remove one



Alarm Results