Lecture 6:

Genetic Algorithms

Week of January 30, 2023

University California, Berkeley

Machine Learning Algorithms

MSSE 277B, 3 Units

Spring 2023

Prof. Teresa Head-Gordon

Departments of Chemistry, Bioengineering, Chemical and Biomolecular Engineering

Review of Previous Lecture: Simulated Annealing

Program Simulated Annealing

Initialize search variable(s) → $f(x_1, x_2 \ldots x_D) = const - \sum_i^D x_i\,(sin(\sqrt{x_i}))$
$x_i \in [-500,500]$ for $i = 1, \ldots, D$
must decide on stopping criteria N_{temp}, define number of steps, N_{equil}

Do I=1,N_{temp}

SA uses cooling schedule → explore linear, geometric, logarithmic;

Do J=1,Nequil

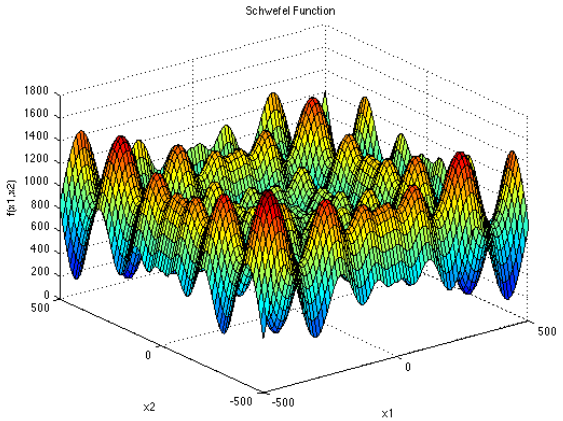
Gaussian Random Walk → The visitation function $x_i = x_i + (2 * URN - 1) \times \Delta$, Δ=0.5 for $i = 1, \ldots, D$

Metropolis MC NVT → $acc(i \to j) = \min[1, exp(-\beta(E(\vec{x}_j^D) - E(\vec{x}_i^D))]$

End do

Global criteria (global opt?) → $E(\vec{x}_{new}^D) < E(\vec{x}_{so\ far}^D)$

End Do



Schwefel function for D=100 in order to find its global minimum using SA

Today's Lecture

Lecture Purpose: We have spent time developing an optimization technique called simulated annealing that is based on concepts borrowed from chemistry, physics, and materials

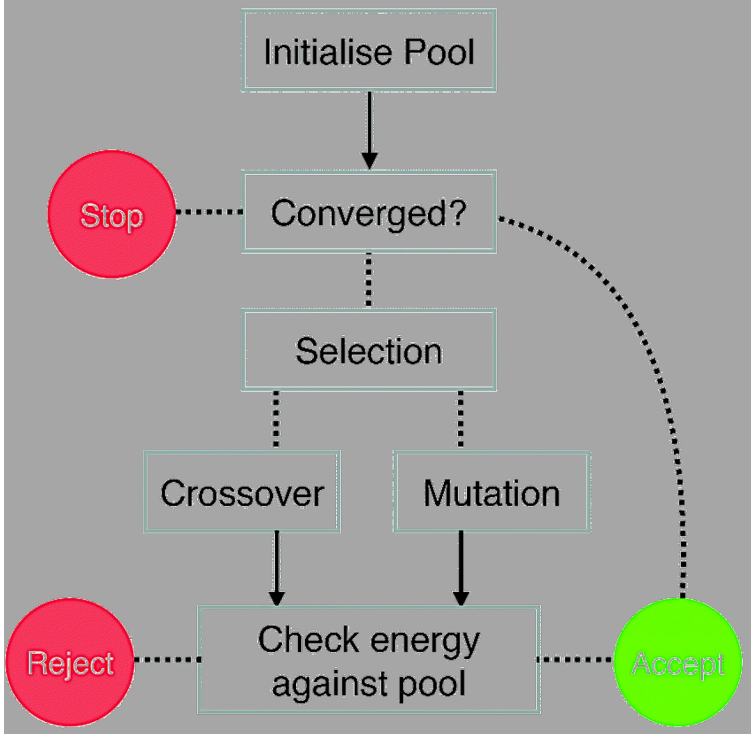
- Today we will cover the biology inspired optimization procedure of genetic algorithms (GA) and next time neural networks (NN). GA and NN are well established optimization techniques that are best for combinatorial optimization (GA) or pattern recognition problems (NN). DNA computing is clever but not ultimately practical.

- Lets begin with GA's – which is a search technique that relies on biological concepts of natural selection and survival of the fittest.

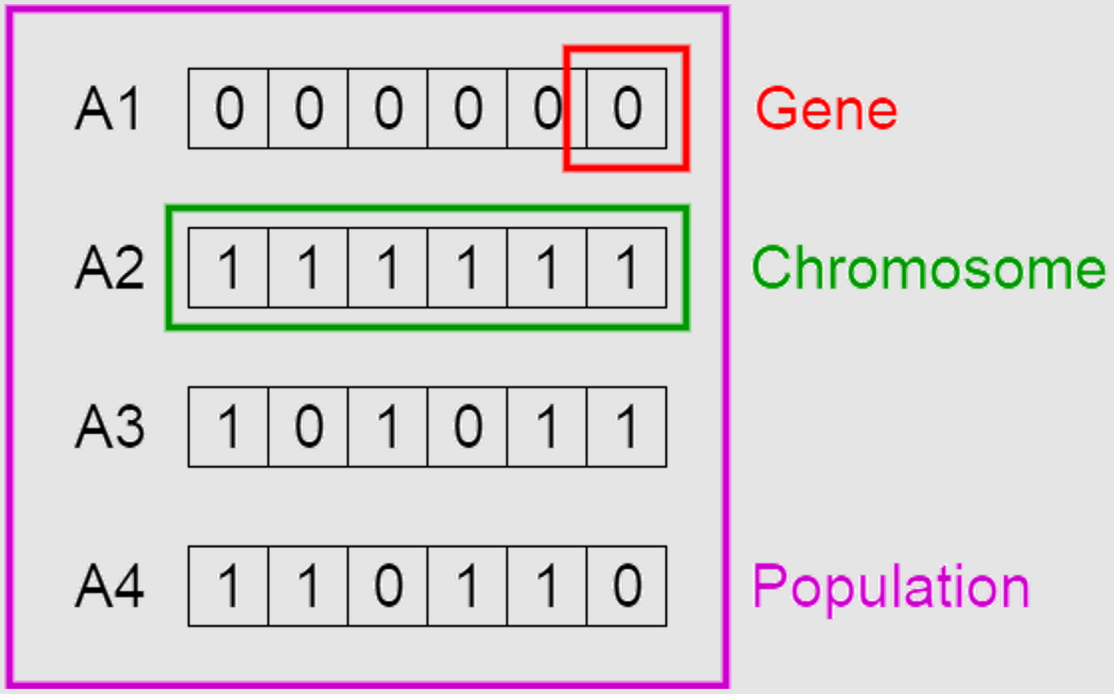
John Holland, ~1970

Genetic Algorithms: Biological Concepts

Typically used for optimization of discrete combinatorial problems, some applications to continuous variable optimization such as molecular biochemistry.



- Encoding function: biological function is encoded in DNA. In optimization applications, how to best encode your problem: variables and objective function values (theoretical foundations rely on binary string encoding)
- Interbreeding populations: "Genetic Pools" of solutions comprised of sub-optimal to optimal global minimum estimates
- Genetic operators: Population undergoes changes by operations of reproduction, cross-over, mutation, maybe others
- Survival of the fittest: as measured by some cost function; generational score, etc.



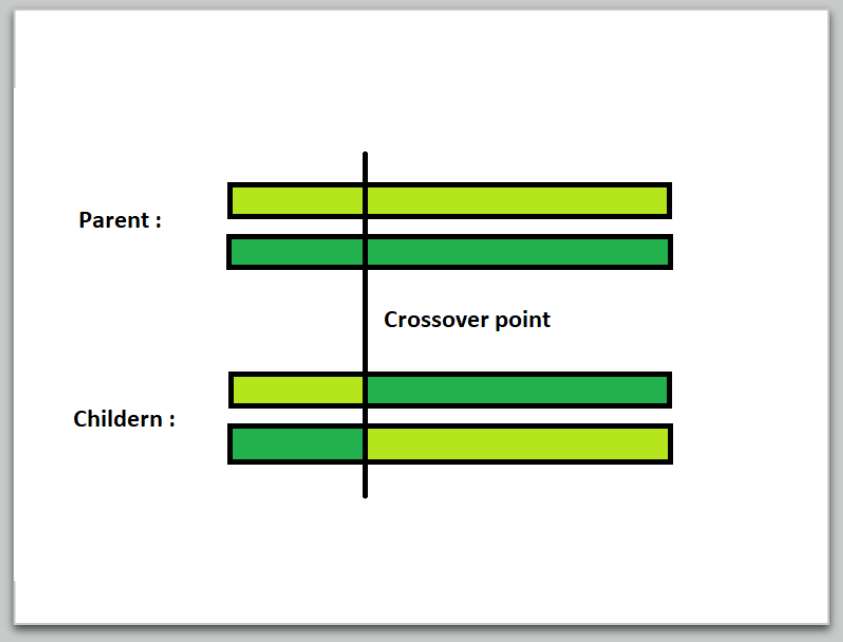
Genetic Pools, Chromosomes and Genes

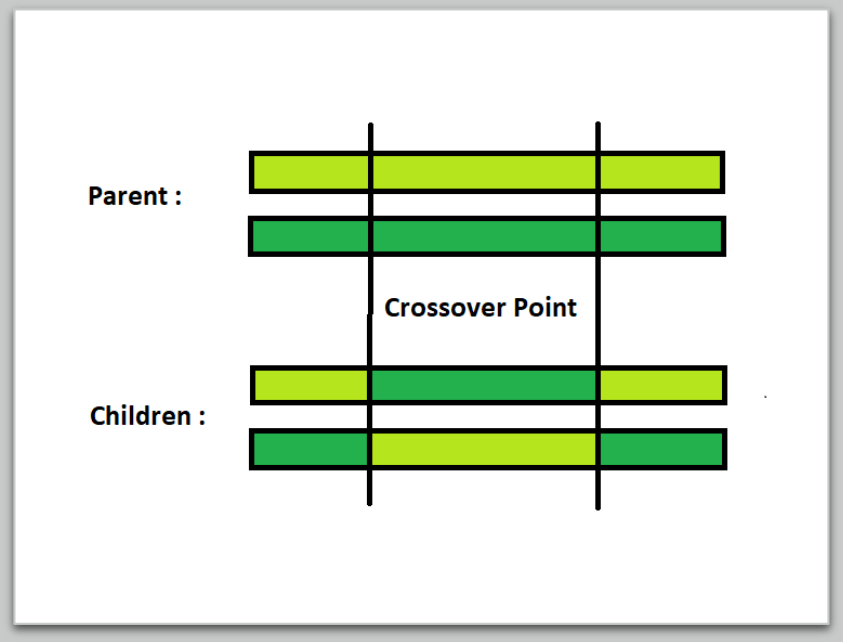
Gene: amino acids with some structural attribute: charge, hydrophobicity, backbone dihedral angle values

Chromosome: protein sequence

A genetic "pool" or population is the space of possible solutions that is created. These can be viewed as chromosomes with an organized gene structure of attributes.

Genetic Operators: Cross-over

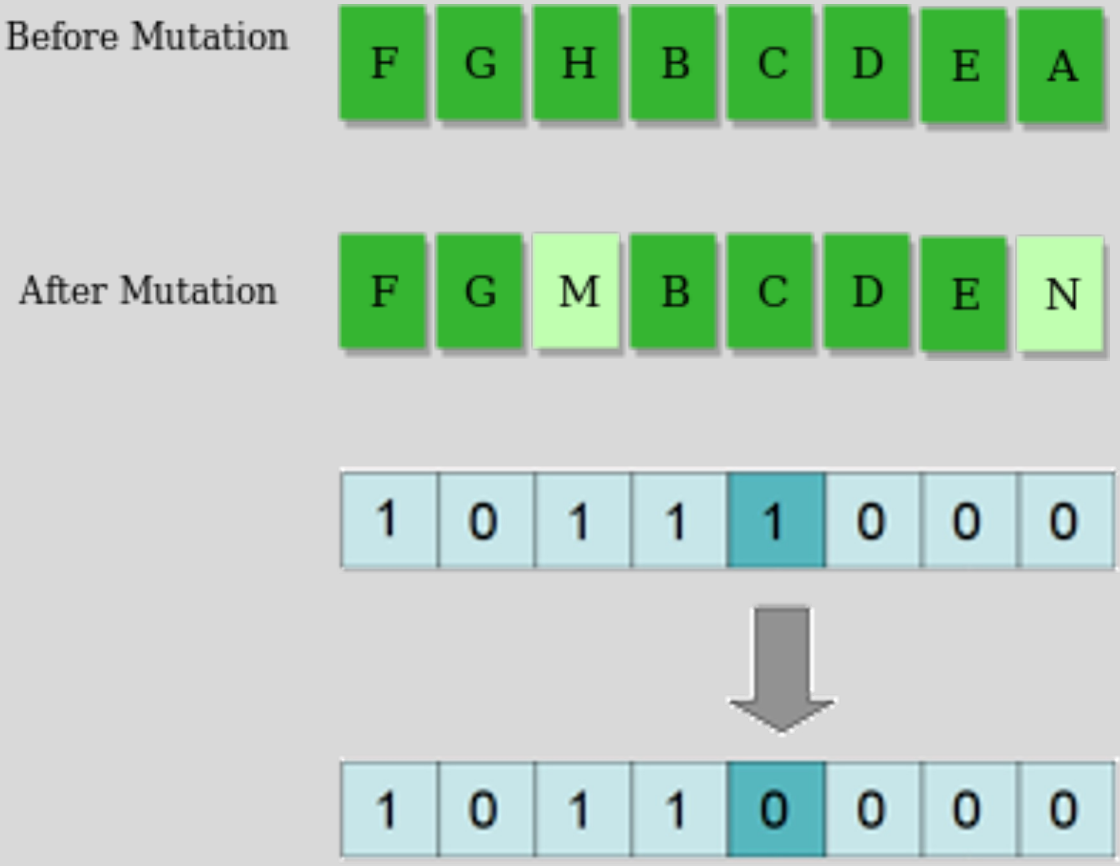




- Cross-over or recombination: 2 strings are selected from the population (could be randomly, or according to fitness, etc) and based on a cross-over probability, randomly create a splice point at the same position for the strings and swap a section of string:

- Two point cross-over: Randomly select two splice points for a pair of strings, and swap sub-sequence between splice points of the 2 strings

- Splice points need to be defined depending on application; sometimes strings can be composed of multiple chromosomes, and maybe do not splice at the gene level

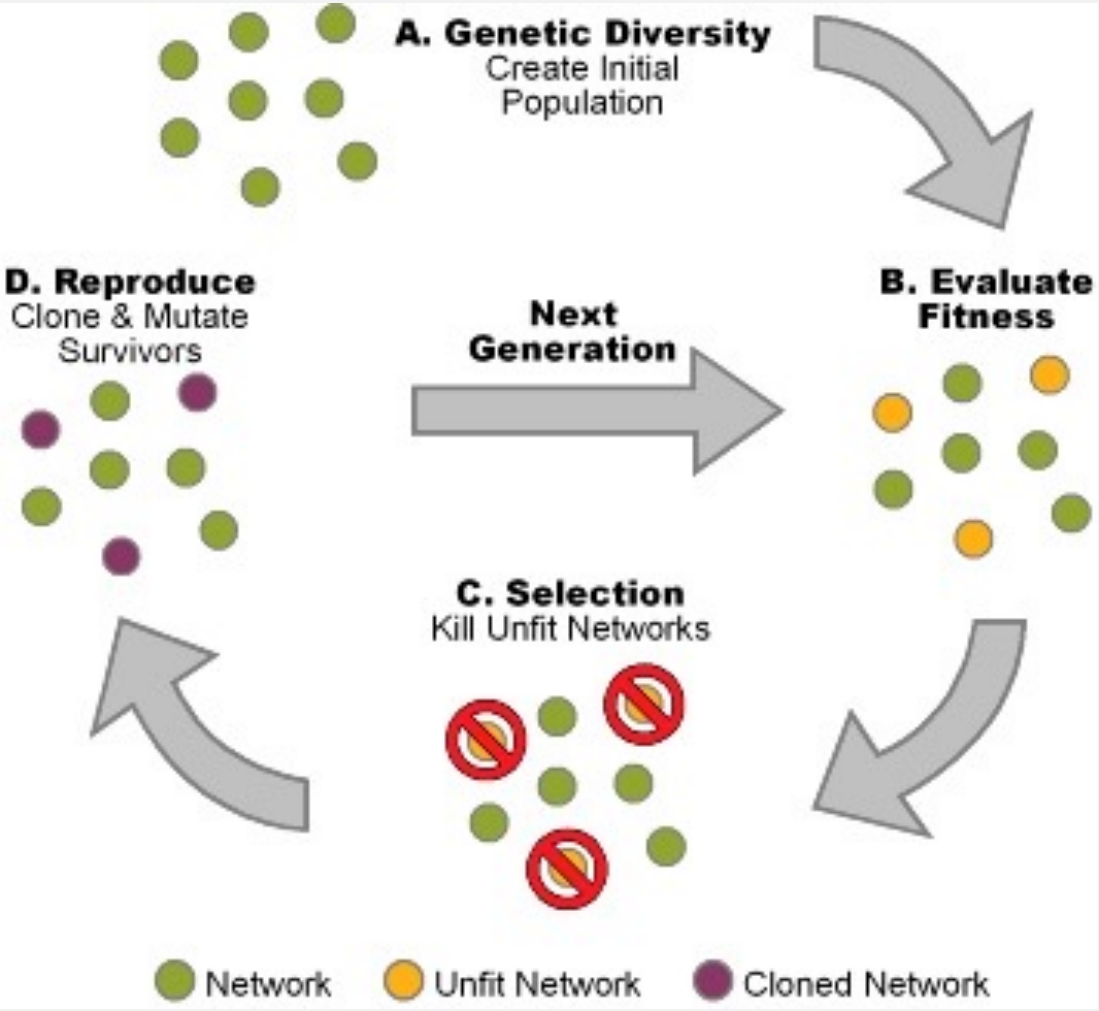


Genetic Operators: Mutation

Mutation: population that is created by series of reproduction or cross-over is not necessarily complete or diverse enough. Therefore for each element for each string of entire mating pool, there is a small probability that it can be mutated at random.

Note the before and after "encodings". We will come back to this later…..

Genetic Operators: Reproduction



Reproduction: a population of strings survives from one generation to the next based on a probability that is a function of their measure of fitness

Fitness is defined based on a cost function. Reproduction can be decided on "survival of the fittest" but also can include population level decisions of birth and death of children vs. adults.

Genetic Algorithms: example

(1) Encode x-values as binary string that covers 8 discrete x-values over [0,7] interval. $2^3=8$

0: 000	3: 011	6: 100
1: 001	4: 110	7: 111
2: 010	5: 101	

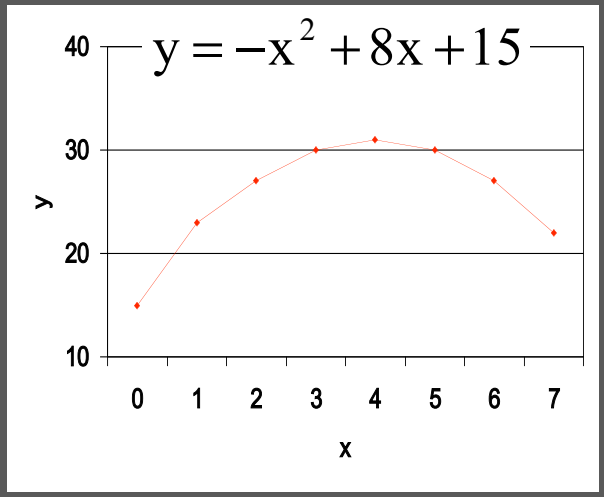
(2) Define a fitness function

$$f(x_i) = \frac{y(x_i)}{\sum_j y(x_j)}$$

(3) Define random population of solutions with binary encoding

$$\begin{matrix} 000 \\ 010 \\ 011 \end{matrix} \Rightarrow \begin{pmatrix} 0 \\ 2 \\ 3 \end{pmatrix}$$

Optimization of a simple quadratic function:



Over range [0,7] find maximum in y.

Of course solution can be read off: $Y_{max}=31$ at $X_{max}=4$

but lets illustrate how a meta-heuristic method solves it

Genetic Algorithms: example

4) Perform selection based on fitness to get mating population

string	fitness	# selections
000	0.0003	0
010	0.123	2
011	0.134	2

(5) Based on probability (URN), it is decided to do a string cross-over. Pick pair of sequences from pool, randomly pick a splice point, and swap strings

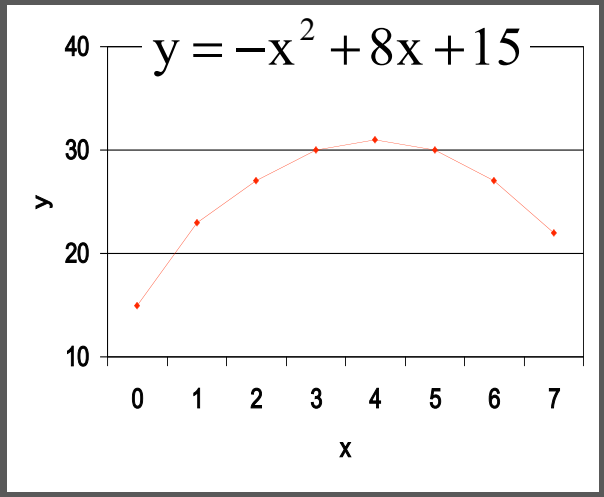
000 010

011 001

(6) Based on probability (URN), we decide to mutate. Pick string at random, pick site at random, and mutate.

011---->011---->001

Optimization of a simple quadratic function:



Over range [0,7] find maximum in y.

Of course solution can be read off: $Y_{max}=31$ at $X_{max}=4$

but lets illustrate how a meta-heuristic method solves it

Genetic Algorithms: example

(7) Now based on some fitness function and other selection rules, advance the population to the next generation.

011 011

010 001

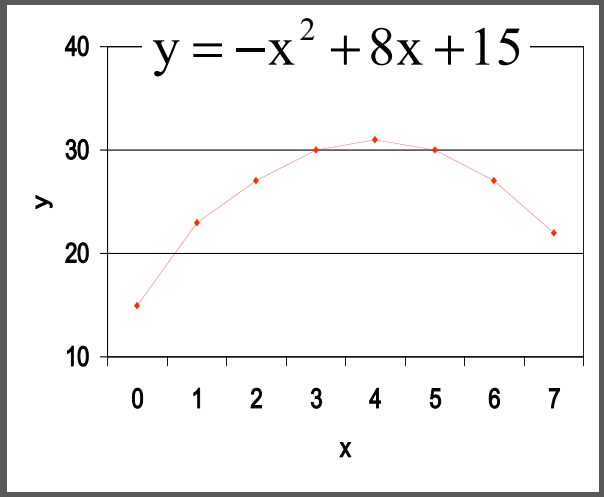
And repeat until find "optimal" solution.

- GA uses past knowledge (previous generations) to direct the stochastic search.

- Genetic algorithms use a population of points to conduct a search, not just a single point advancement through tree

GA's that use binary encodings typically have stronger theoretical foundations

Optimization of a simple quadratic function:



Over range [0,7] find maximum in y.

Of course solution can be read off: Y_{max}=31 at X_{max}=4

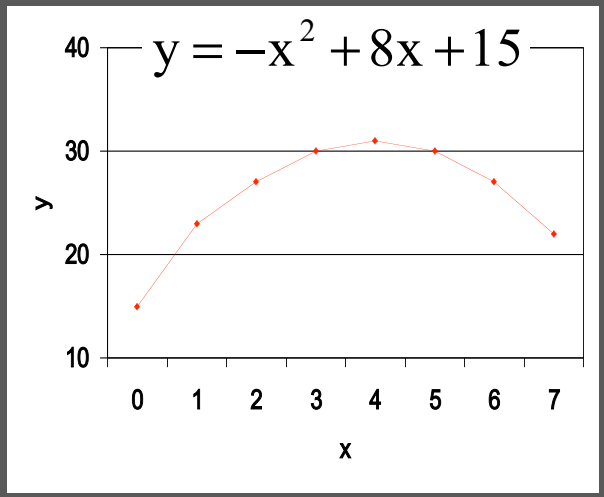
but lets illustrate how a meta-heuristic method solves it

Genetic Algorithms: Foundations

string	decoded value	fitness
00000	0	0.0003
00001	1	
00010	2	0.123
00100	3	0.134
01000	4	
10000	5	
…	…	…
11111	31	etc

Let's reconsider this quadratic optimization example by assuming we can't "bracket" the maximum so well, i.e. over range [0,7]. Our best estimate is that it is in a region [0-31]

Now we define a binary string for solution space that is $2^5=32$



Genetic Algorithms: Schema

schema	Matching strings	Order	Length
000**	00000, 00001, 00010, 00011	3	3-1=2
00*00	00100, 00000	4	5-1=4
0	Many!	1	3-3=0
	etc		

Now consider strings in terms of blocks known as "schemata". We will define a wildcard variable * that can be either 0 or 1.

For example:
*1111 schemata has 2 matching strings, 01111, 11111

Classification of schema:
Define length between last and first specified string positions. Define order has the number of specified sites.

If we encoded our problem such that the fittest of the population was represented by

0***0 (length = 4, order=2)

It should reproduce very often because of its fitness score!

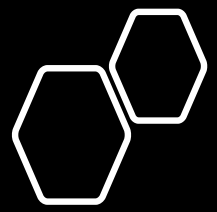
However this is a non-optimal encoding since genetic operators such as cross-over will disrupt this schemata,

In fact any schemata of long length, and/or high order, will therefore *reduce* its representation in subsequent generations.

Mutation always disrupts schemata's (but good for moving through solution space)

0 * * * 0 1 * * * 0

Genetic Algorithms: Schema



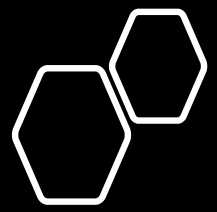
Instead we want very fit schema to have very short length and small order since genetic operators will be less likely to disrupt the advancement of those fit solutions to the next generation.

0**** (short length, low order)

In fact Holland showed that these well-encoded schema (short length and low order) grow exponentially from generation to generation.

So with proper encoding and low mutation rates, the final population will be dominated by the fittest members!

Genetic Algorithms: Schema



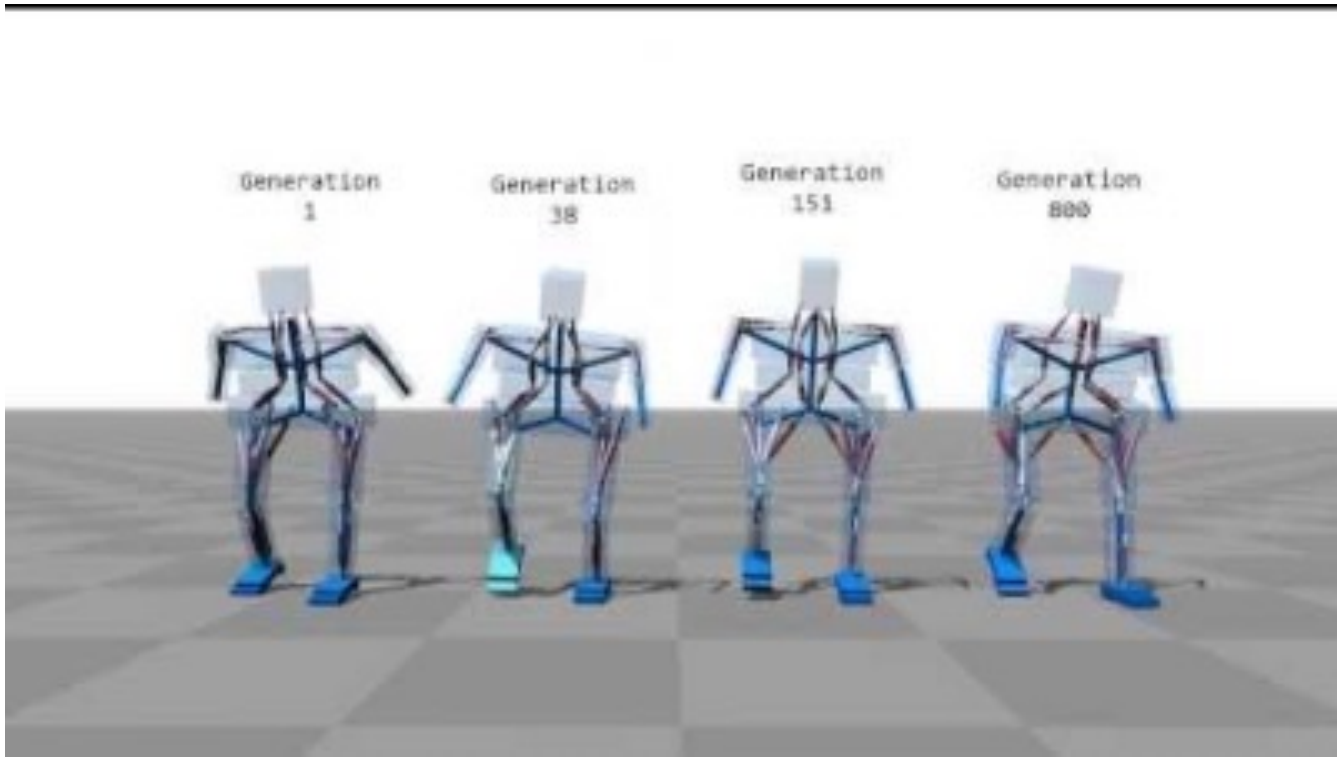
We will find additional biological concepts and operators useful:

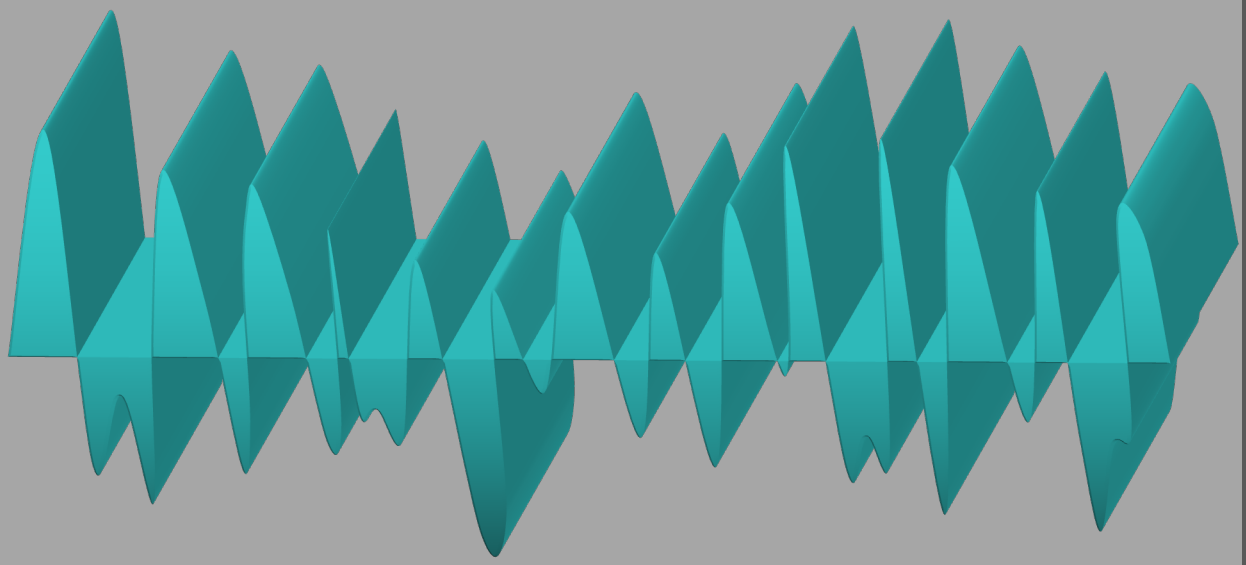
(1) How to advance population from generation to generation:

- Total replacement: only newly created offspring advance and their parent population is completely discarded. Disadvantage: fit parents are replaced by useless children!
- Elitist replacement: all parents and offspring are ordered according to fittest, and best n-members advance.
- etc

(2) The probability of applying any given operator can change dynamically during run time of the optimization.

Additional GA Concepts





Native State: Global Free Energy

Application Area: Structure Prediction

Given amino acid sequence, an objective function, and a search method one can predict the protein's tertiary structure.

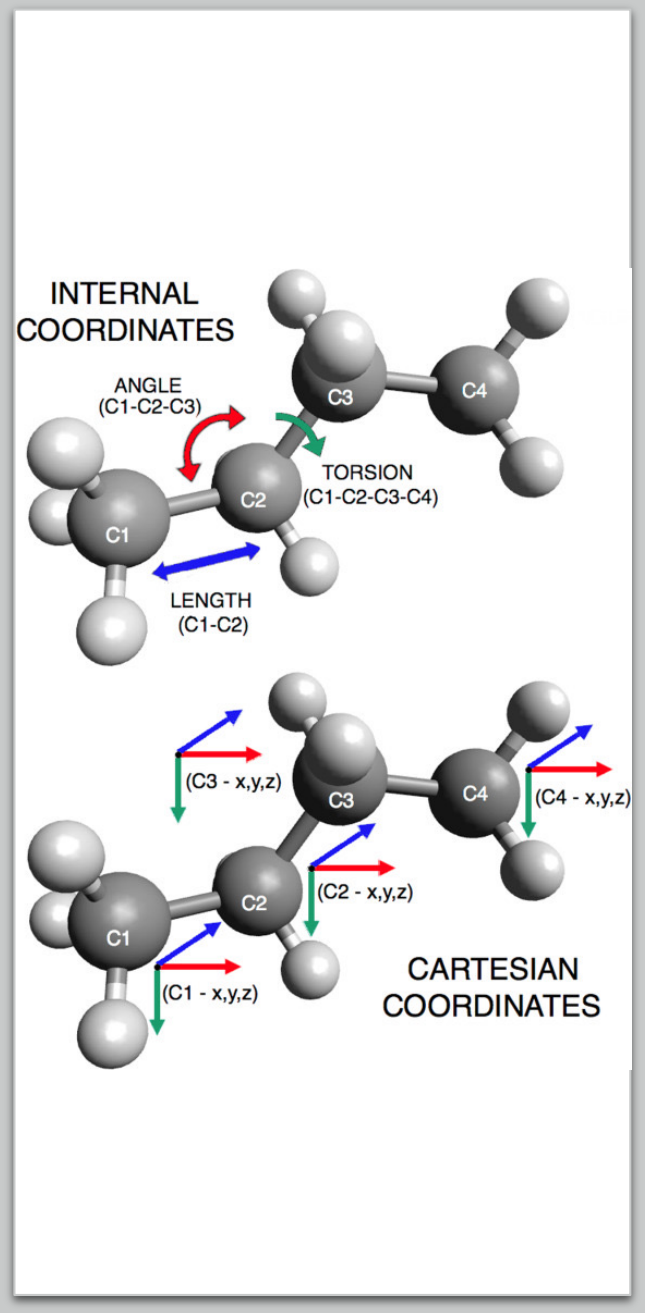
Based on the hypothesis that the native state corresponds to the global free energy minimum

G=H-TS-->minimum

and that at the minimum entropy is zero, and that H~PES

One strategy is to use meta-heuristic techniques like SA or GA to perform this search for the "fittest" protein structure.

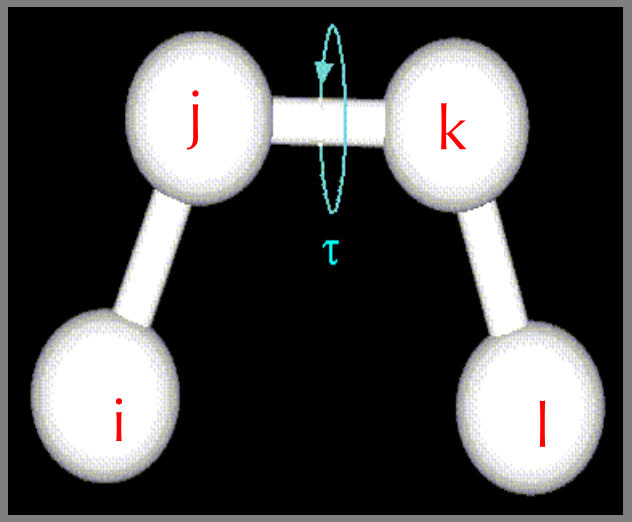
Protein Representation

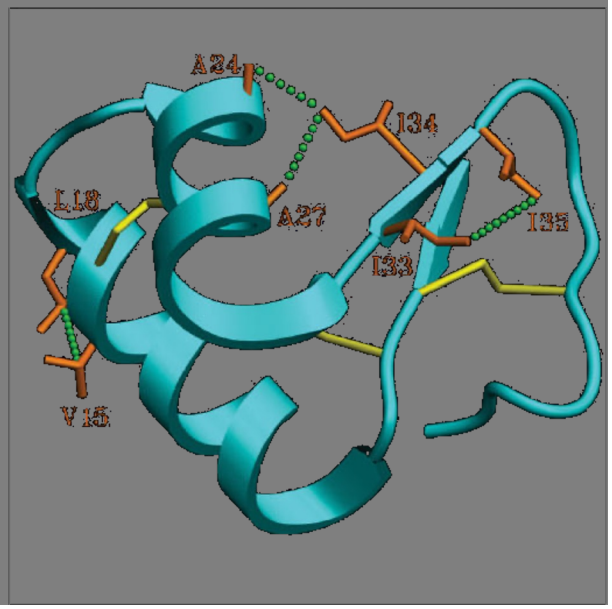


Cartesian coordinates. Visitation functions such as Gaussian Random Walks in Cartesian space for chain molecules creates untenable protein conformations. In addition, GA operators create highly unfit conformations and slow progress of global search.

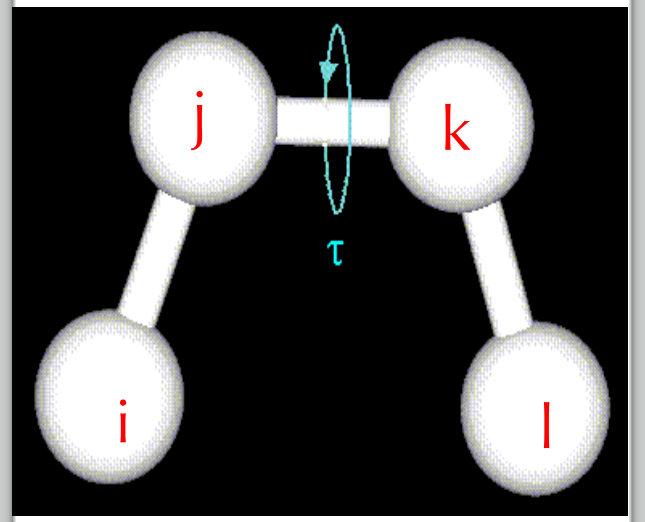
Torsion angles. GA operators and SA random moves create displacements about rotatable bonds: more likely to generate sound protein conformations.

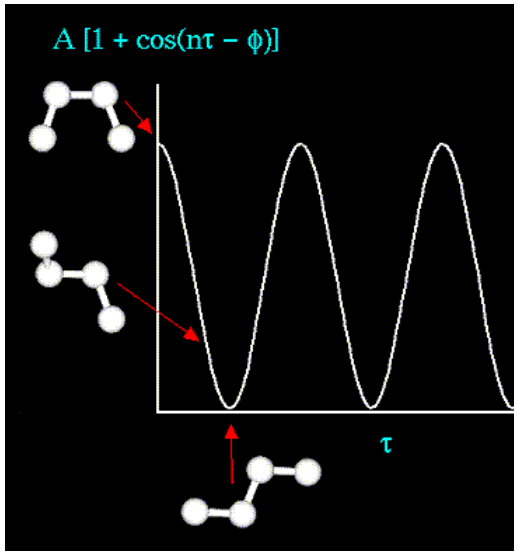
$$V_{dihedral} = \sum_{n=1}^{\#\,dihedrals} k_{\tau n}\left[1+\cos\left(m\tau_n - \tau_{n0}\right)\right]$$





Binary Encoding of Solution Space





$$\{\tau\}_l \Rightarrow \phi_1 = 106^\circ, \psi_1 = 90^\circ \ldots \phi_n = 46^\circ, \psi_n = -90^\circ \ldots \phi_N = -56^\circ, \psi_N = 57^\circ$$

000: 0-50°
001: 50-80°
010: 80-120°
100: 120-160°
110: 160-200°
101: 200-250°
011: 250-300°
111: 300-360°

Would need to know native state to encode solution space so that low order and small length are commensurate with global energy state!

But you don't necessarily know enough to make a good encoding!

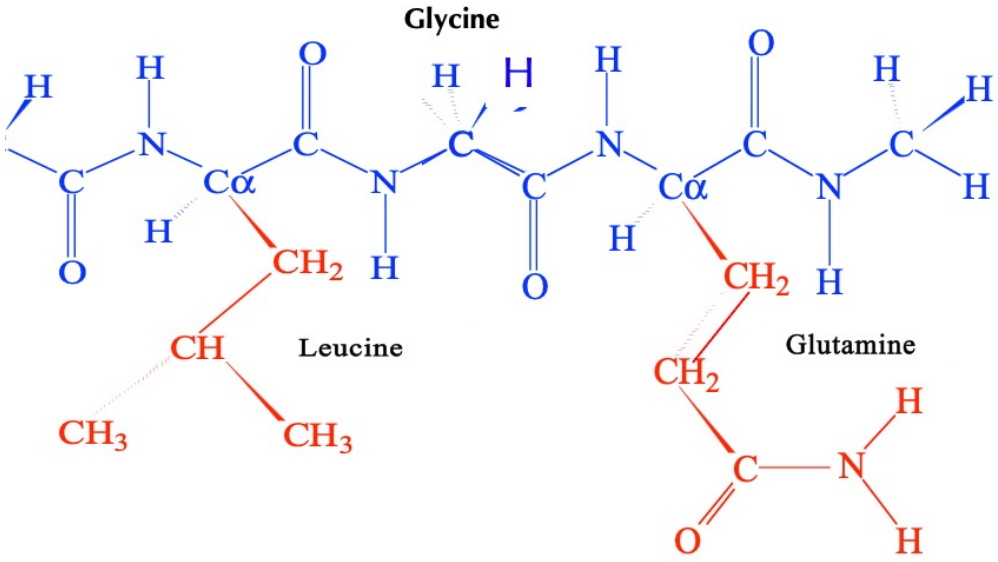
LeGrand & Merz, Schulze-Kremer did not attempt to encode solution space for protein structure prediction to exploit schemata!

Instead, they used continuous torsion values, that allowed them to define GA operators that are more specific to this application area.

Fitness of configuration member depends on torsion and non-bonded energy (bonds and angles held fixed, ignored):

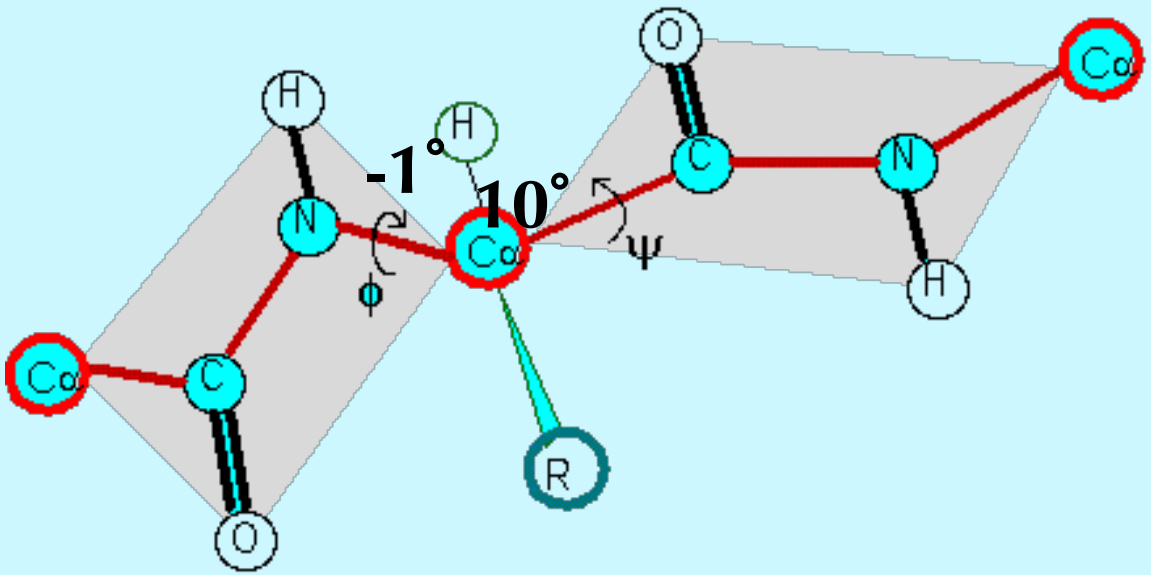
$$V_{Protein} = \sum_{i}^{\#dihedrals} k_{\chi}\left[1+\cos(n\chi+\delta)\right] + \sum_{i}^{\#atoms}\sum_{i<j}^{\#atoms}\left\{\frac{q_i q_j}{r_{ij}} + \varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right]\right\}$$

"Encoding" of Solution Space for GA



$$\{\tau\} = \phi_1, \psi_1, \chi_{11}, \chi_{12}, \cdots \chi_{1n}, \cdots \phi_5, \psi_5, \chi_{N1}, \chi_{N2}, \cdots \chi_{Nn}$$

Genetic Operators: Variate



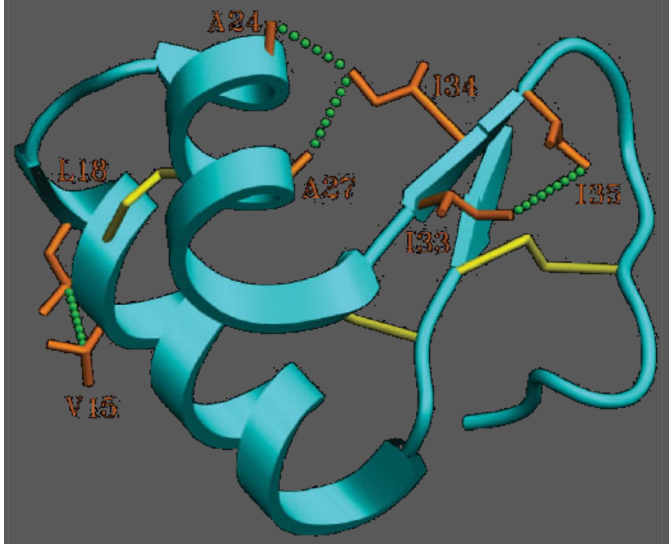
http://bioinformatics.org/molvis/phipsi/

- Variate operator increments or decrements (with or without equal probability) the torsion angle by 1°, 5° or 10° (each chosen randomly).
- Uniform cross-over: Each amino acid torsion of one string can be swapped with corresponding torsion in another string
- Two point cross-over: Randomly select two splice points for a pair of strings, and swap sub-sequence structure between splice points between the 2 strings

Genetic Algorithm: Dynamics

GA operator parameters	Value	
Number individuals/generation	100	
Number generations	5000	
Mutation rate (start/end)	80%	20%
Variate (start/end)	80%	20%
Variate10° (start/end)	60%	0%
Variate 5° (start/end)	30%	5%
Variate1° (start/end)	10%	80%
Cross-over (start/end)	70%	10%
Cross-over uniform (start/end)	90%	10%
Cross-over 2-point (start/end)	10%	90%

The following GA algorithm was applied to crambin, a 40 amino acid protein (unusually hydrophobic) to predict its known native structure

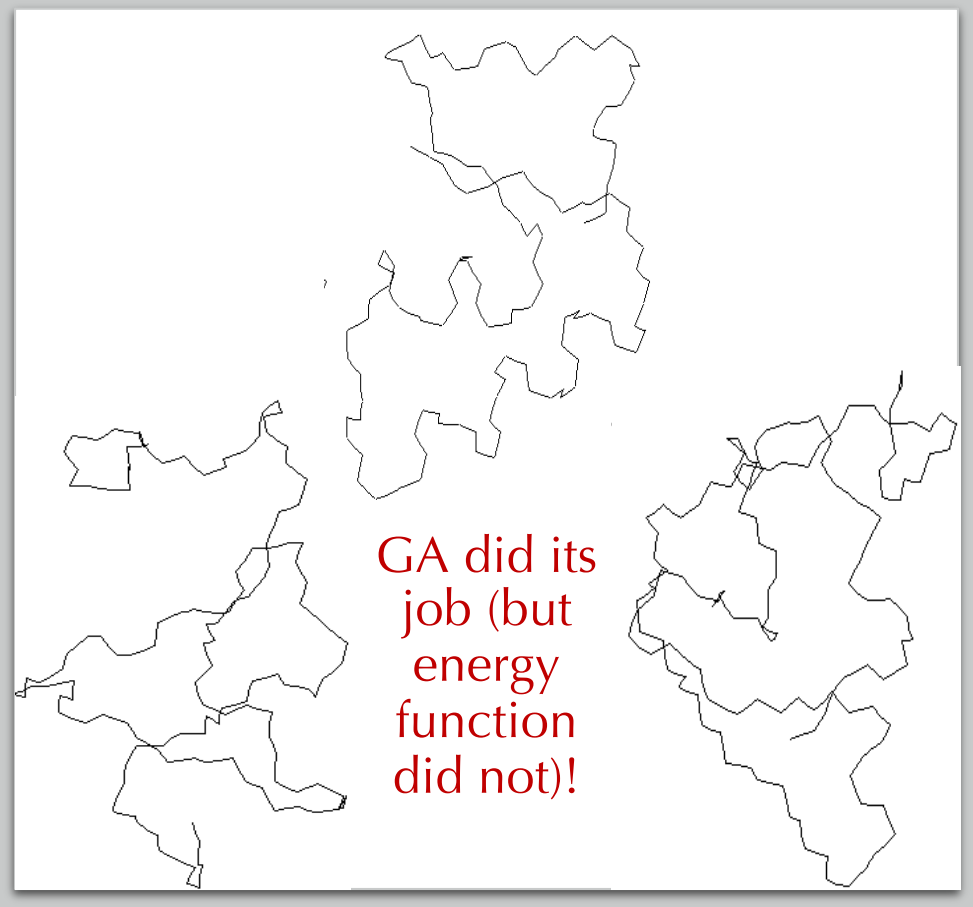


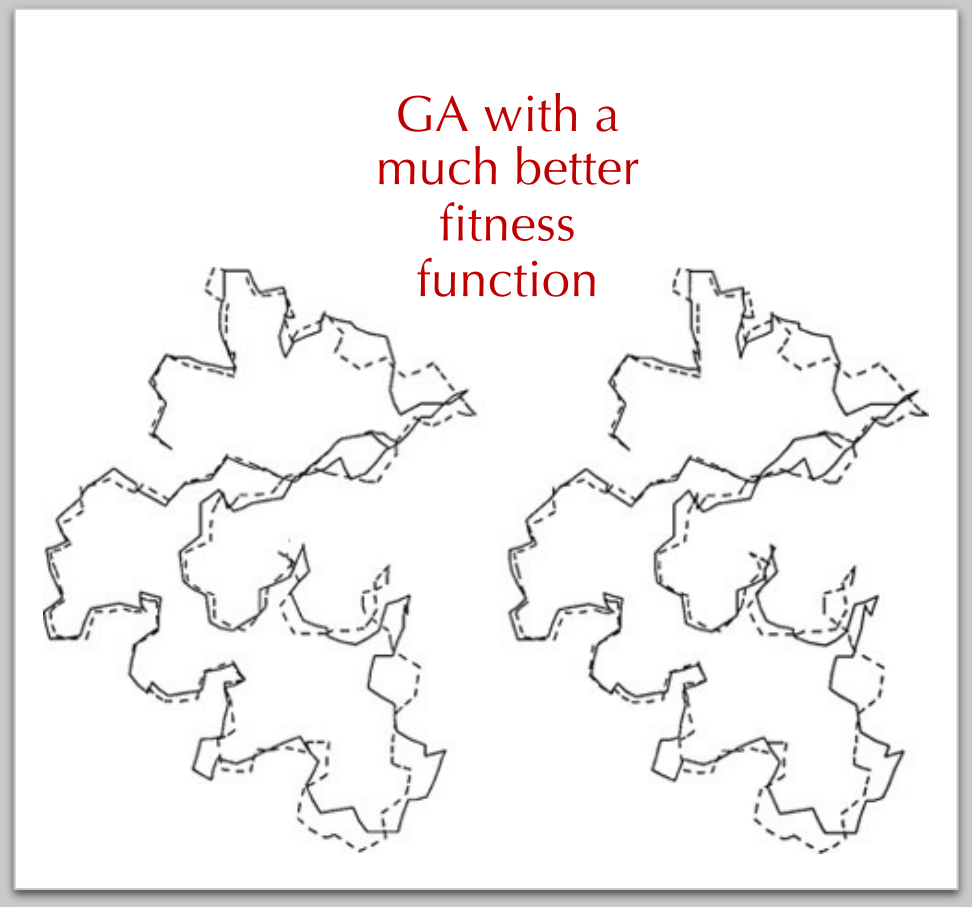
Note that the algorithm adapted the evolutionary operators over "time". Equivalent to "cold" temperatures in SA

Defined GA Applied to Crambin

(1) Best structures (for fitness): ~8-10A rmsd. Essentially random prediction

(2) GA with better fitness function- much better!





http://www.techfak.uni-bielefeld.de/bcd/Curric/ProtEn/1215.html