



Metaheuristic Optimization and Drug Design

Prof. Sanghamitra Bandyopadhyay

Machine Intelligence Unit

Indian Statistical Institute, Kolkata

sanghami@isical.ac.in

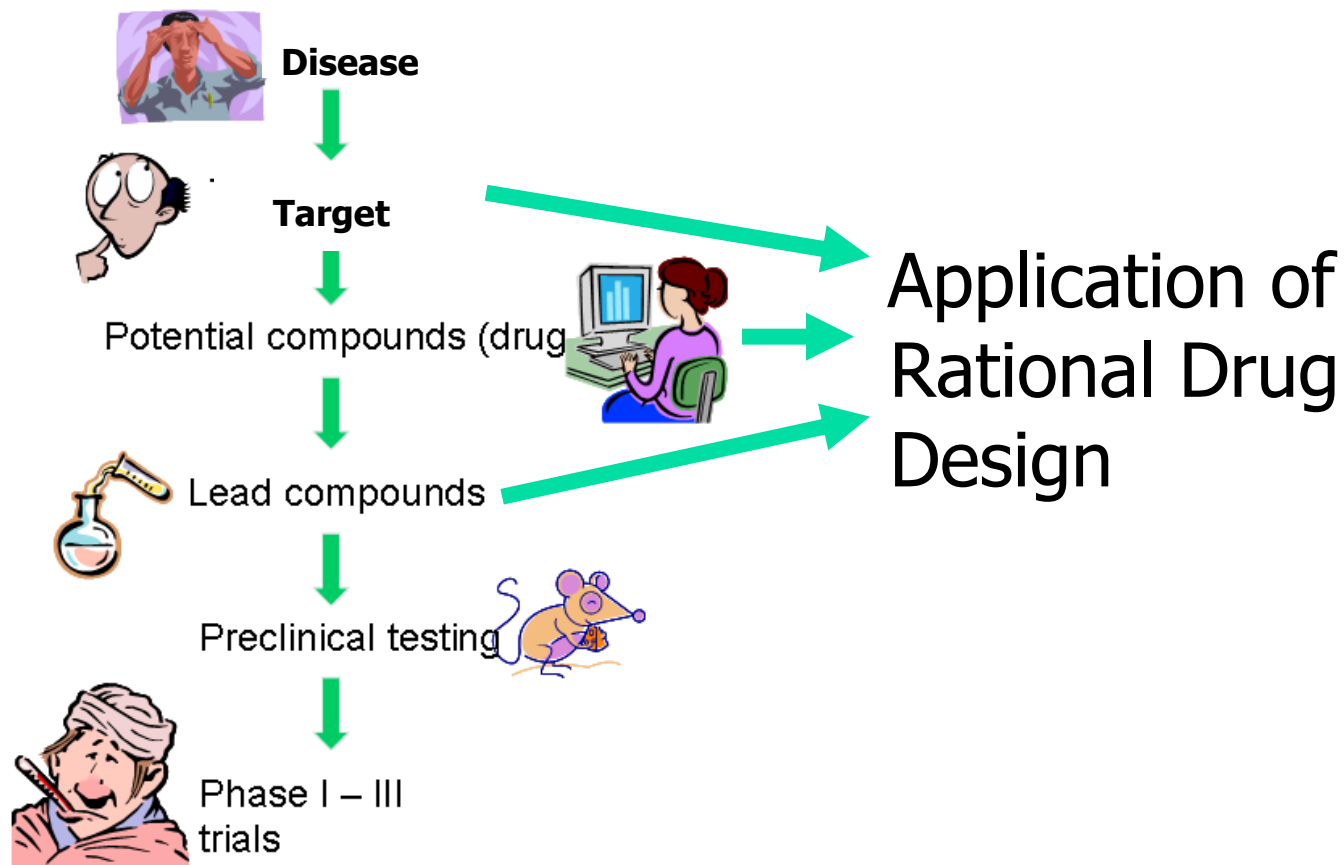
<http://www.isical.ac.in/~sanghami>



Outline of the Presentation

- What is rational drug design?
- Relevance of genetic algorithms
- Design Methodology
- Experimental Results
- Summary

Drug Discovery Process





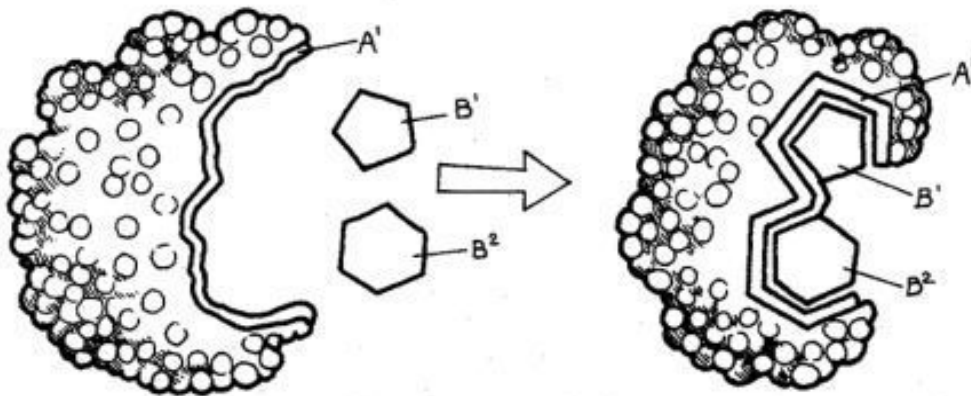
Rational Drug Design

- Traditionally drugs were discovered by chance observations
- Alternatively large scale screening was done to identify potential drugs
 - Expensive and time consuming.
- Rational Drug Design
 - design using the information about the 3D shape of proteins
 - To inhibit protein function
- Steps
 - Step 1: Looking for protein targets in the virus
 - Step 2: Identify the active site
 - Step 3: Design drug for blocking the active site
 - Step 4: Do further studies with the designed molecule

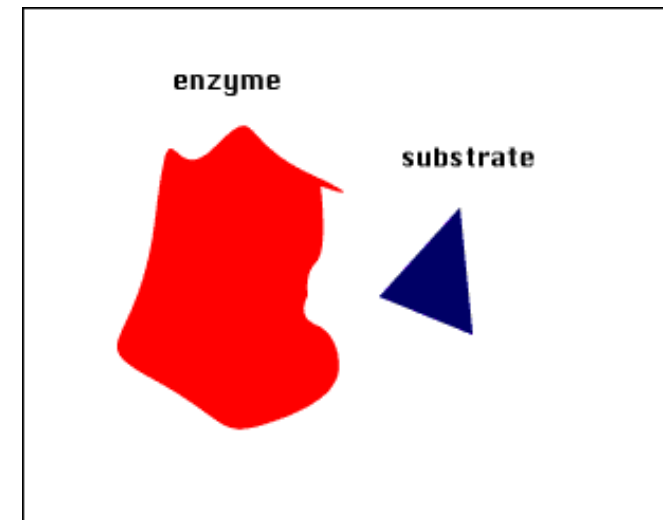
Active site and RDD

- Specific sites in proteins where all the action happens.
- Each protein has a specific shape so it will only perform a specific job.
- Example – An enzyme that increases the rate of a reaction

Joining things together

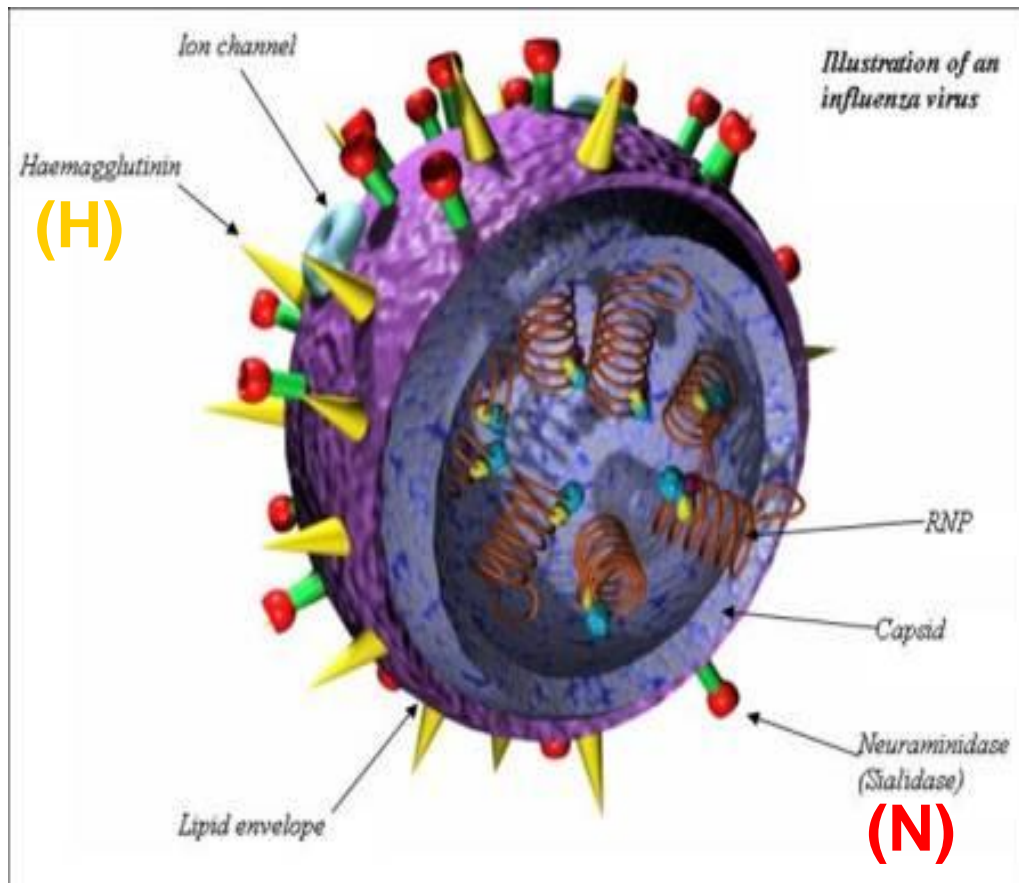


Ripping things apart



Designing a Flu Drug

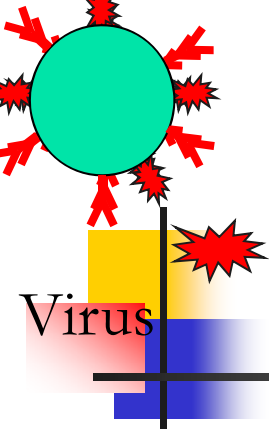
Step 1: looking for protein targets



Influenza viruses are named according to the proteins sticking out of their virus coat.

There are two types of protein = **N** and **H**.

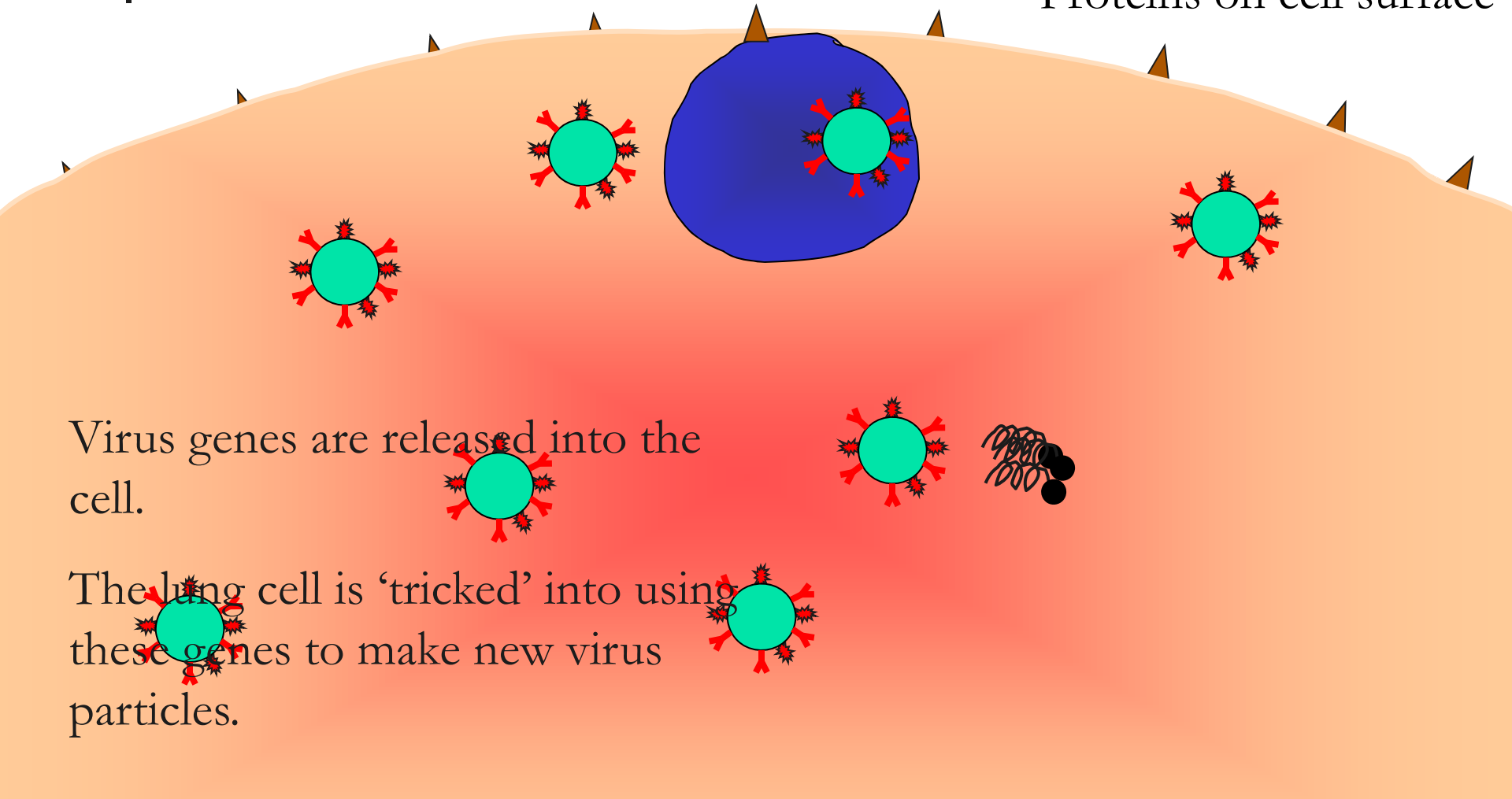
N and H have special shapes to perform specific jobs for the virus.



N cuts the links between the viruses and the cell surface so virus particles are free to go and infect more cells.

H attaches to cell surface proteins so virus can enter

Proteins on cell surface



Virus genes are released into the cell.

The lung cell is 'tricked' into using these genes to make new virus particles.

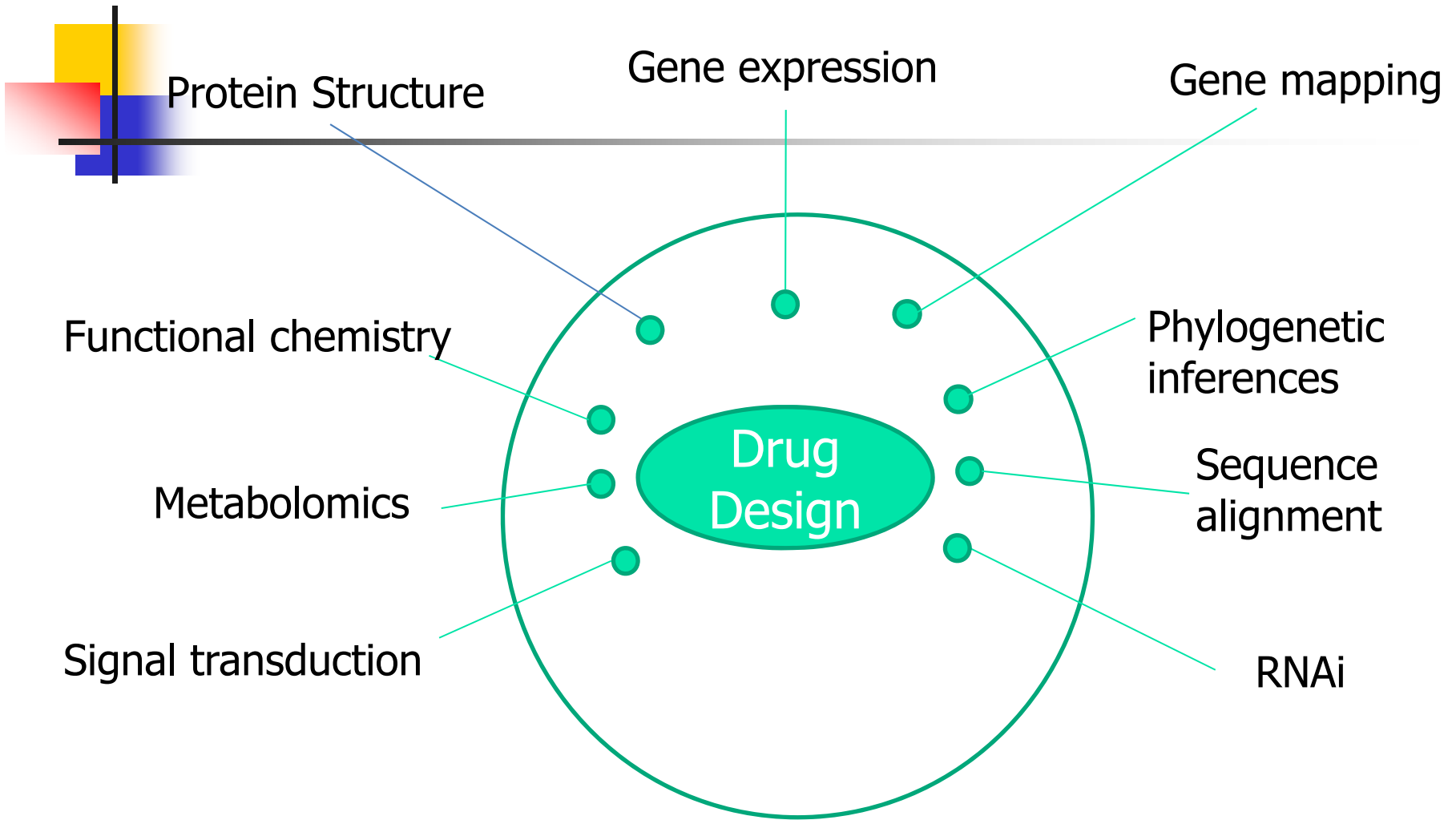
Design of Flu Drug



RELENZA

Australian team of scientists headed by Prof Peter Coleman. They designed the flu drug, Relenza

Bioinformatics in Computer Based Drug Design





GENETIC ALGORITHMS

- **DEFINITION**

Randomized search and optimization technique guided by the principle of natural genetic systems.

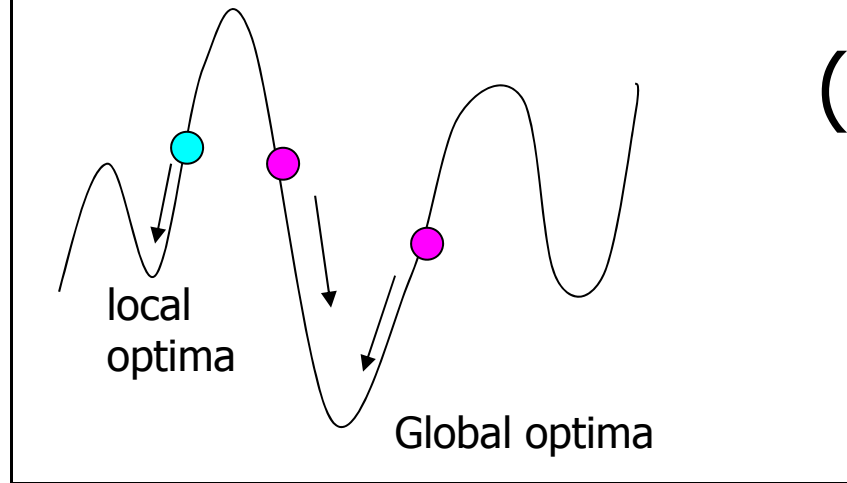
- **Why Genetic Algorithms (GAs) ?**

1. Most real life problems can not be solved in polynomial time using any deterministic algorithm
2. Sometimes near optimal solutions that can be generated quickly are more desirable than optimal solutions which require huge amount of time
3. When the prob. can be modeled as an optimization one.

Search Techniques

The traditional vs. the unconventional

- Calculus based techniques – gradient descent



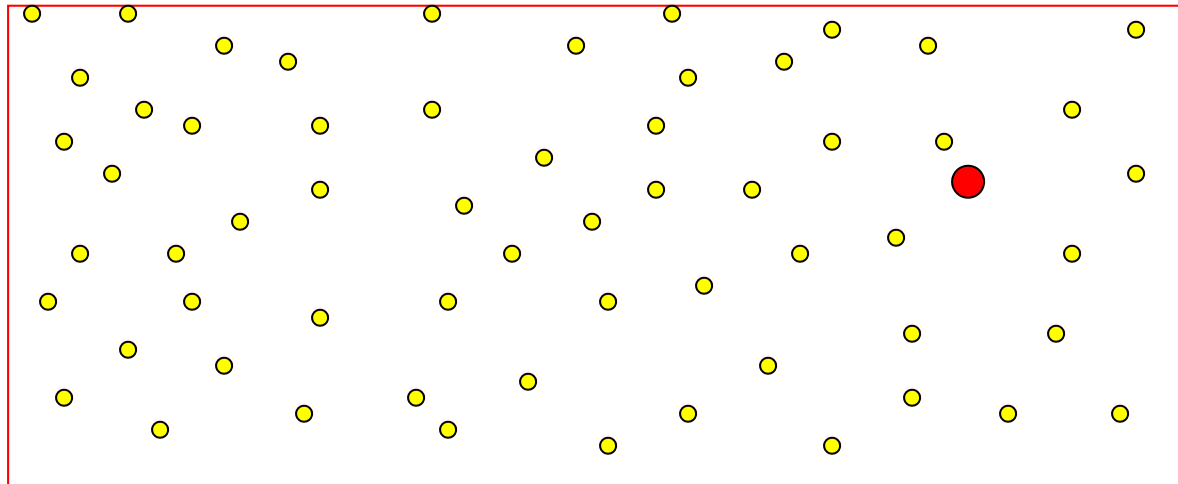
(hill climbing)

Continuous domain, quadratic optimization – best method

Search Techniques

The traditional vs. the unconventional

- Enumerative technique – dynamic programming



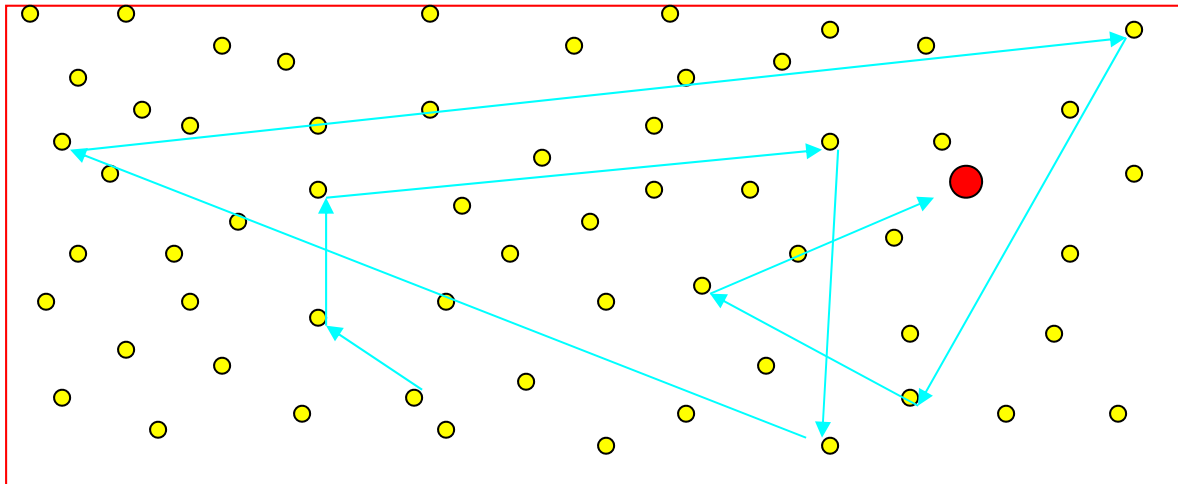
n points

What if n very very large? Quite likely in practice.

Search Techniques

The traditional vs. the unconventional

- Random technique – hoping to find the optimal



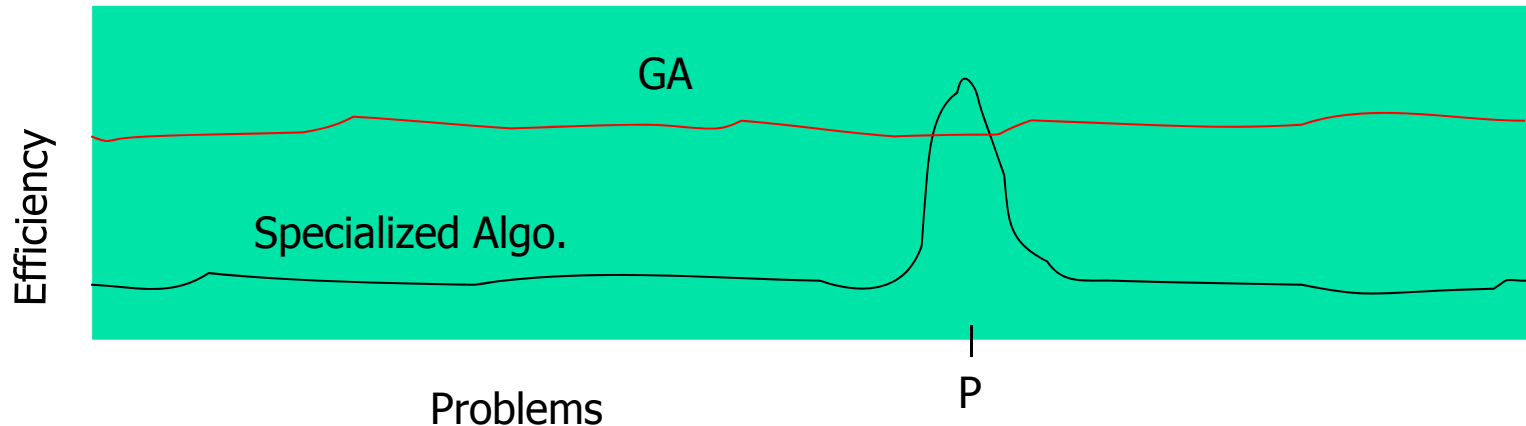
No better than enumerative in the long run

Randomized Algorithms

- Guided random search technique
- Uses the payoff function to guide search



Genetic Algorithms (GAs)



Specialized algorithms – best performance for special problems

Genetic algorithms – good performance over a wide range of problems

Genetic Algorithms - Features



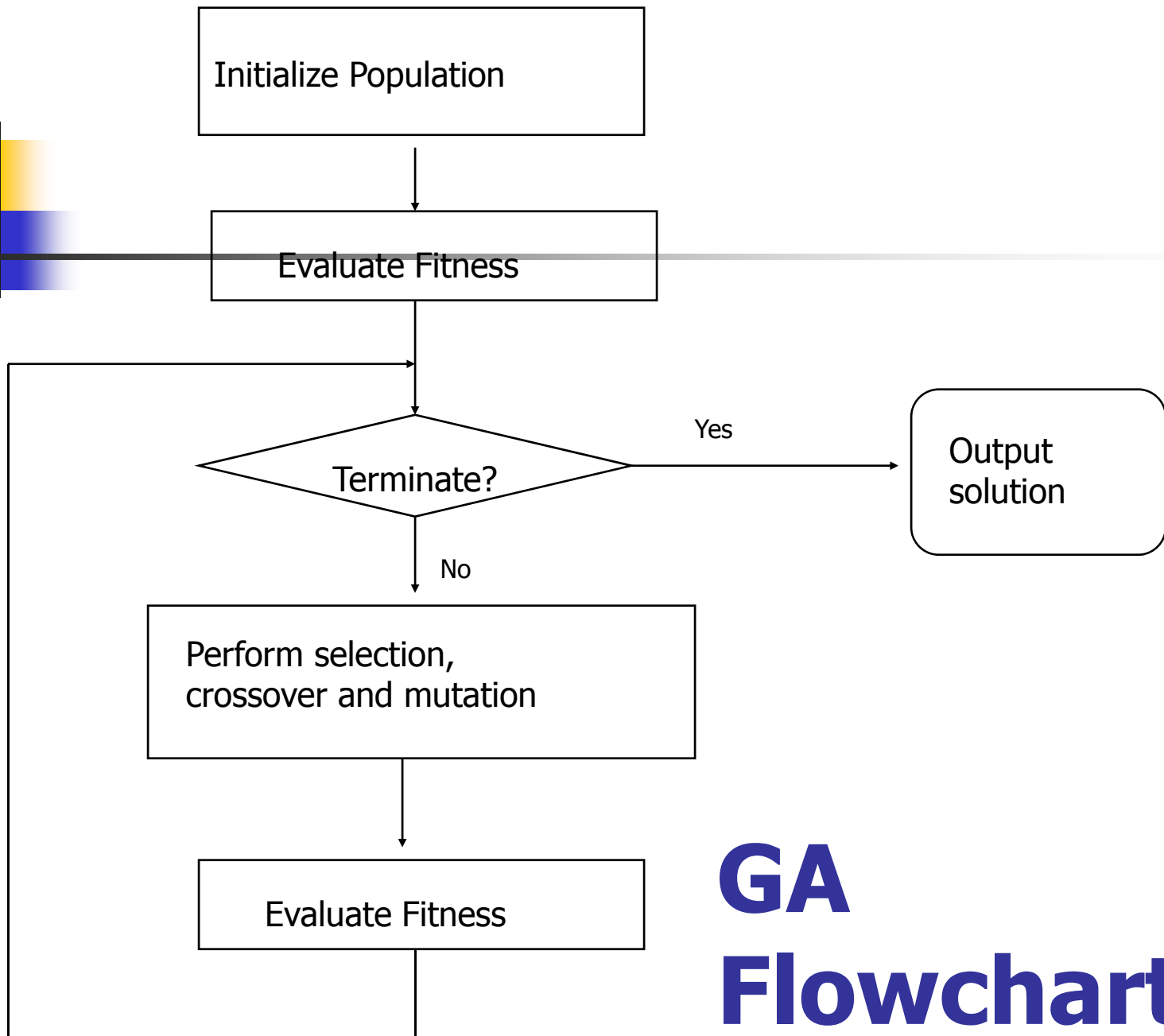
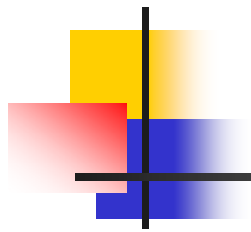
- Evolutionary Search and Optimization Technique
- Principles of Evolution (*survival of the fittest* and *inheritance*)
- Work with coding of the parameter set
- Searches from a population of points
- Uses probabilistic transition rules



Simple Generational GA

- 1▪ Randomly generate a population of chromosomes
- 2▪ Decode each chromosome to get an individual
- 3▪ Evaluate the fitness of each individual
- 4▪ Perform selection, crossover and mutation.
- 5▪ Repeat steps 2, 3 and 4 until a stop condition is true.

Note : There is no overlapping between generations.



GA Flowchart



Encoding Strategy and Population

- **Chromosome encodes a solution in the search space**

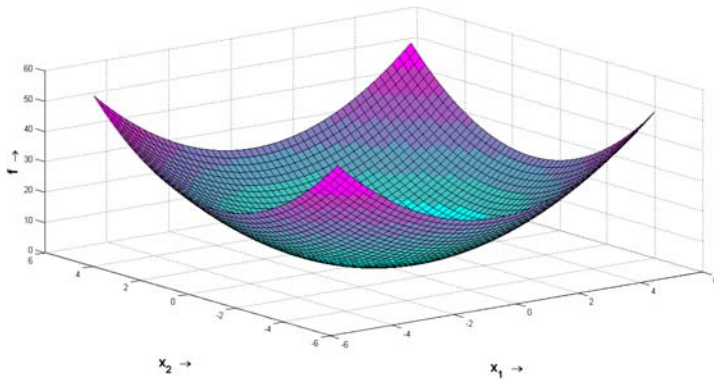
- Usually as strings of 0's and 1's
- If l is the string length, number of different chromosomes (or strings) is 2^l

- **Population**

- A set of chromosomes in a generation
- Population size is usually constant
- Common practice is to choose the initial population randomly.

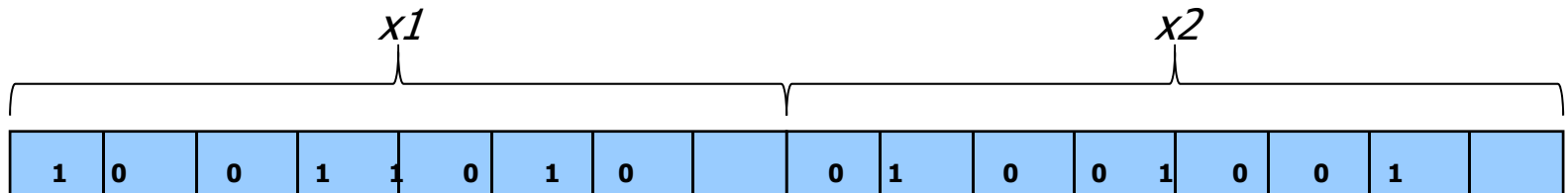
Encoding and Population - Example

Optimization Problem DeJong F1 - Sphere:
Minimize $fsphere(x) = \sum x_i^2$, $i=1,2,\dots,p$, $x_i \in [-5.12, 5.12]$,
Solution: $x^* = [0 \ 0 \dots 0]$, $fsphere(x^*) = 0$



Binary String of 8 bits used to represent each $x_i \rightarrow 0-255$, should map to -5.12 to 5.12

Example, $p=2$,
Chromosome encodes $x1$ and $x2$



Value of $x1 = 154 \longrightarrow (5.12*2)/255 * 154 + (-5.12) = 1.06415$

Value of $x2 = 73 \longrightarrow (5.12*2)/255 * 73 + (-5.12) = -2.1885$

Encoding and Population – Example

contd...



Population (size = 4)

Corresponding x

10011010 01001001 [154 73]

[1.06415 -2.1885]

01100111 11101001 [103 233]

[-0.98384 4.23654]

00010101 01011100 [21 92]

[-4.2767 -1.4255]

10111100 11000011 [188 195]

[2.42949 2.71058]



Fitness Evaluation

- Fitness/objective function associated with each chromosome
- Indicates the degree of goodness of the encoded solution
- Only problem specific information ([also known as the payoff information](#)) that GAs use

Fitness Evaluation - Example



Minimize $fsphere(x) = \sum x_i^2$

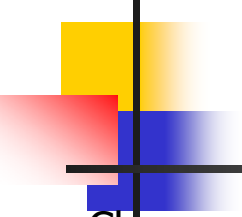
Population (size = 4)			Corresponding x	Objective fn.	Fitness fn.=1/Obj
10111100	11000011	[188 195]	[2.42949 2.71058]	13.2898	0.0752
10011010	01001001	[154 73]	[1.06415 -2.1885]	5.12194	0.16886
00010101	11011100	[21 220]	[-4.2767 3.7145]	32.0877	0.03116
01100111	11101001	[103 233]	[-0.98384 4.23654]	18.9162	0.05286



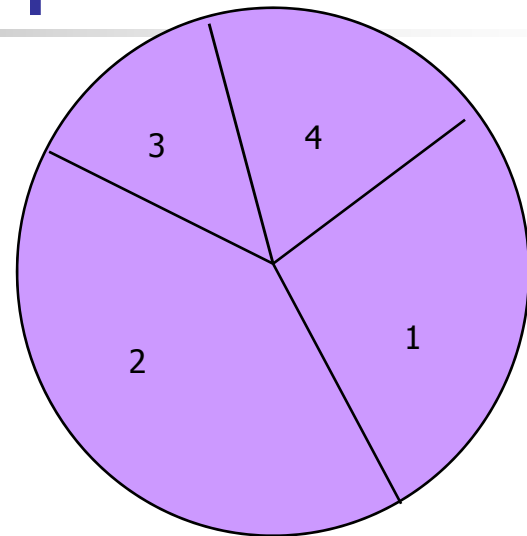
Selection

- More copies to good strings
- Fewer copies to bad string
- proportional selection scheme
 - Number of copies taken to be directly proportional to its fitness
 - mimics the natural selection procedure to some extent.
 - *Roulette wheel parent selection and stochastic universal selection* selection are two frequently used selection procedures.

Roulette Wheel Selection – Example



Chromosome #	Fitness
1	0.0752
2	0.16886
3	0.03116
4	0.05286



Spin 1	Chromosome 2 selected
Spin 2	Chromosome 1 selected
Spin 3	Chromosome 2 selected
Spin 4	Chromosome 4 selected

Mating
→
Pool

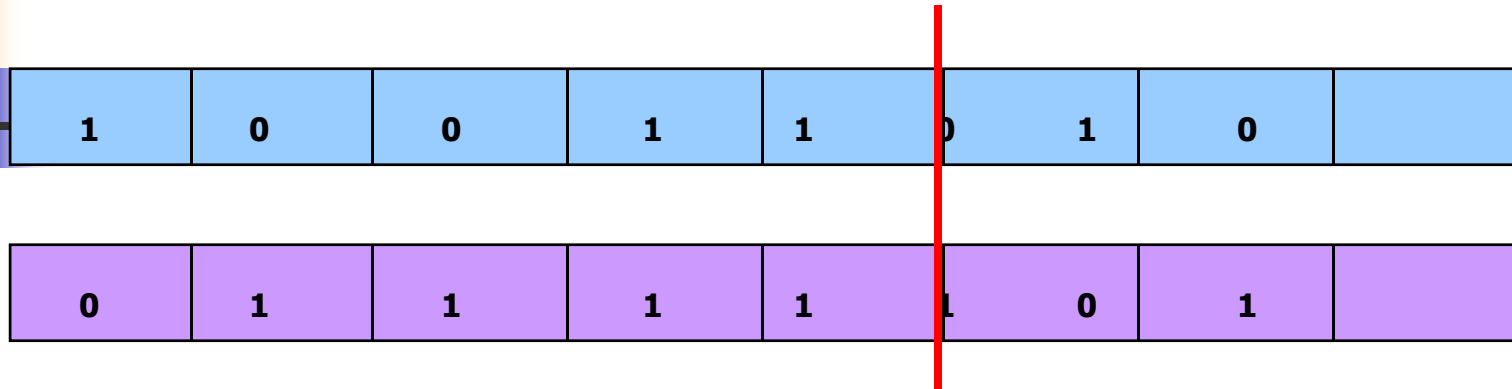
0	1	1	0	0	1	1	1
1	0	0	1	1	0	1	0
0	1	1	0	0	1	1	1
1	0	1	1	1	1	0	0



Crossover

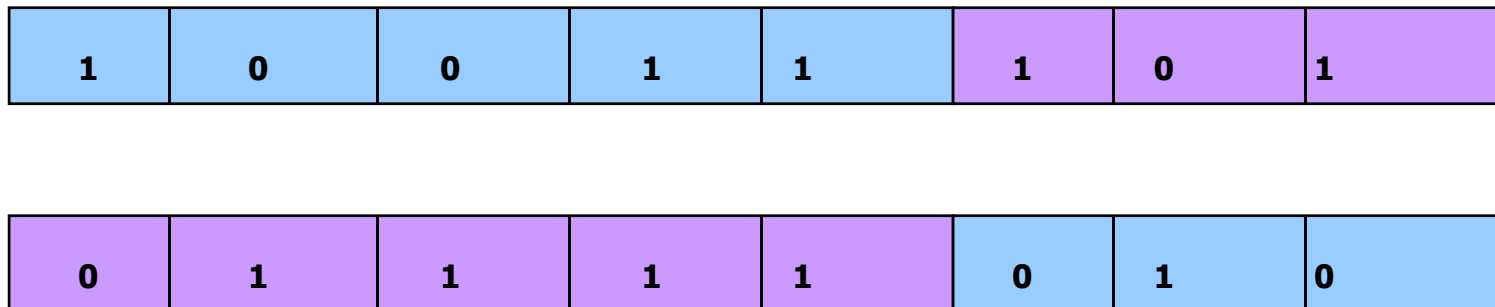
- **Exchange information**
 - between randomly selected parent chromosomes
 - Single point crossover is one of the most commonly used schemes.
 - probabilistic operation

Crossover – Example



Here l (string length) = 8. Let k (crossover point) = 5

Offspring formed :

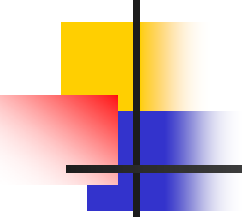




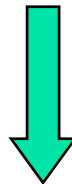
Mutation

- **random alteration** in the genetic structure
 - introduce **genetic diversity** into the population.
 - Exploration of new search areas
 - Mutating a binary gene involves simple **negation of the bit**,
 - Mutating a real coded gene defined in a variety of ways
 - probabilistic operation

Mutation- Example



1	0	0	1	1	0	1	0	
---	---	---	---	---	---	---	---	--



Mutations at positions 2 and 5

1	1	0	1	0	0	1	0	
---	---	---	---	---	---	---	---	--



Parameters ...

- **population size (usually fixed)**
- **string length (usually fixed)**
- **probabilities of performing crossover (μ_c) and mutation(μ_m),**
 - μ_c is kept high and μ_m is kept low
- **the termination criteria**
 - Generally a maximum number of iterations
- parameters are user determined
- parameters are problem dependent
- no firm guidelines
- parameters can be kept **variable and/or adaptive.**



Parameters – Example

For the example being considered,

$$P = 4, \quad I = 8.$$

But for most realistic cases

P is usually chosen in the range 50-100.

$$\mu_c = [0.6-0.9],$$

$$\mu_m = [0.01-0.1].$$

/usually depends on the required precision



Termination Criterion

The cycle of selection, crossover and mutation is repeated a number of times till one of the following occurs :

- **the average fitness value of a population becomes more or less constant over a specified number of generations,**
- **a desired objective function value is attained by at least one string in the population,**
- **the number of generations (or iterations) is greater than some threshold ---
-- most commonly used.**



Elitist Model of GAs

The best string (in terms of fitness) seen up to the current generation is preserved in a location either inside or outside the population.



Drug Design – Relevance of GAs

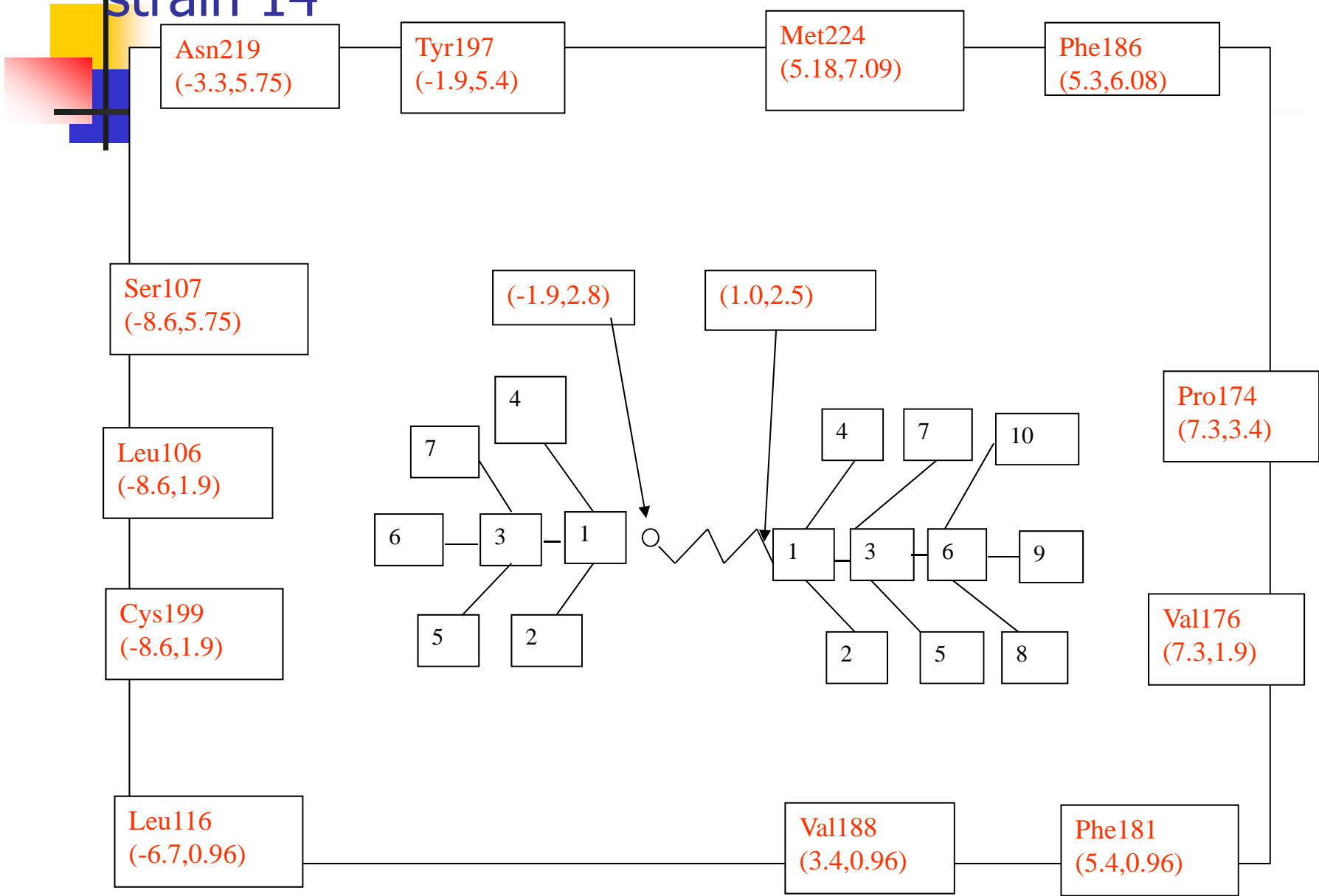
- Identify/design a suitable ligand which can bind to the active site of a protein to prevent its proliferation.
- Design the ligand using groups from a library of chemical groups
 - Such that interaction energy is minimized
- Drug design problem can be modeled as one of optimization
- Application of GAs becomes relevant.



Problem Objective

- Design of molecules that can bind to the active site of harmful protein (e.g., those crucial for the proliferation of microbial organisms, cancer cells or viruses).
- Such molecules can destroy the action of the target protein
 - thereby nullifying its activity which can be lethal to us.
- Accurate prediction of the structure of the potential inhibitors, while utilizing the knowledge about the structure of a target protein, is important in *drug design*.

Barrel shaped active site of human rhino virus strain 14





The Design Methodology

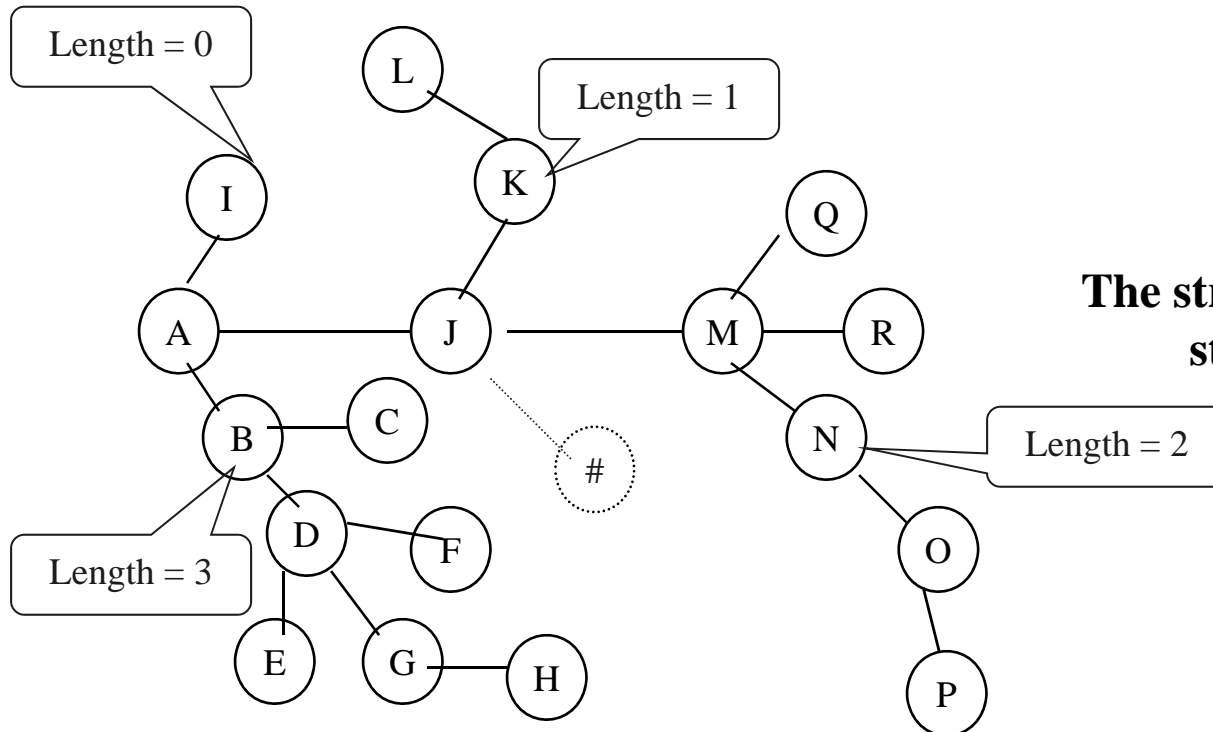
- Building ligands from a fragment library
- Optimizing the energy components
 - Van der Waals energy = $[(C_n / r^n) - (C_m / r^{12})]$
 - Electrostatic energy = $(q_1 q_2) / (4\pi\epsilon_0 r^2)$
 - $\epsilon_0 = 8.854185 \times 10^{-12}$ coulomb²/(N m²)
 - Bond stretching energy = $[k_l \times (l_{xy} - l_{xy,0})^2] / 2$
 - Angle bending energy = $[k_\theta \times (\theta - \theta_0)^2] / 2$
 - Torsional energy = $k_\phi \times (1 - \cos n \times (\phi - \phi_0))$
- GA is used for minimization

ENCODING STRATEGY

B	L	LC	LL	LU	LM	LML	LMU	U	UC	UL	UU	UM	UL	B	L
---	---	----	----	----	----	-----	-----	------	---	----	----	----	----	----	------	---	---

Lower tree structure

Upper tree structure



The string representation of this structure is as follows

AB3#CDEFG#H#I0J#0K1#L#MN2##0P##Q0R



Fragment Details

- Number of bonds a fragment can make is given below

Fragment Number	Number of connectivity		Fragment number	Number of connectivity
0	4		1	6
2	4		3	6
4	6		5	2
6	4		7	4
8	4		9	3
10	5		11	7
12	2		13	4
14	6		15	2
16	4		17	6
18	2		19	4
20	4		21	6
22	3		23	5
24	5		25	7
26	4		27	10
28	8		29	12
30	8		31	6
32	6		33	7
34	7		35	10
36	9		37	8
38	12		39	4
40	1			

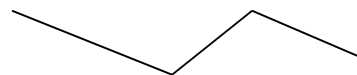


Some Fragments in the Library

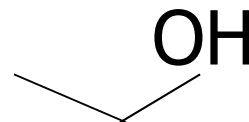
- Group 0 Alkyl 1C
 - Bond length ~ 0.65 along x-axis



- Group 1 Alkyl 3C
 - Bond length ~ 1.75 along x-axis



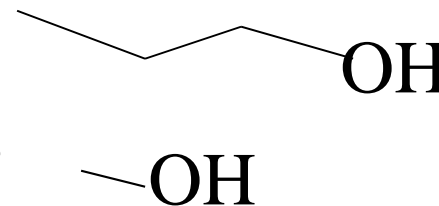
- Group 2 Alkyl 1C Polar
 - Bond length ~ 1.1 along x-axis



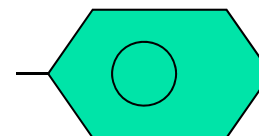


Groups Considered

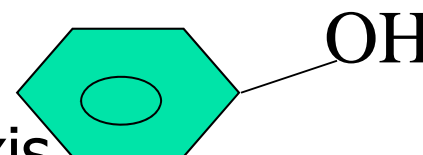
- Group 3 Alkyl 3C Polar
 - Bond length ~ 2.2 along x-axis
- Group 4 Polar



- Group 5 Aromatic
 - Bond length ~ 1.9 along x-axis



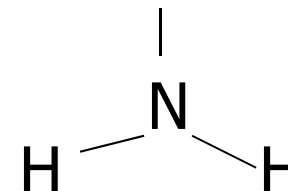
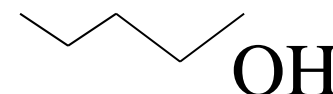
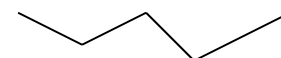
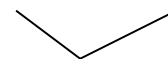
- Group 6 Aromatic polar
 - Bond length ~ 2.7 along x-axis





Groups Considered

- Group 7 Alkyl 2C
 - Bond length ~ 1.2 along x-axis
- Group 8 Alkyl 4C
 - Bond length ~ 2.5 along x-axis
- Group 9 Alkyl 4C Polar
 - Bond length ~ 2.9 along x-axis
- Group 10 Amine NH_2
 - Bond length ~ 0.5 along x-axis





Groups Considered

- Group 11 Alkyl 5C
 - Bond length ~ 3.1 along x-axis
- Group 12 Alkyl 2C Polar
 - Bond length ~ 1.68 along x-axis
- Group 13 Alkyl 5C Polar
 - Bond length ~ 3.58 along x-axis





Results

Energy Values (by InsightII in Kcal/mole)	HIV-1 Protease		HIV-1 Nef Protein	
	VGA	IVGA3 D	VGA	IVGA3D
Vander Waals Energy	4.724	-3.862	-2.616	-3.873
Coulombs Energy	-1.408	-4. 190	0.764	-3.691
Total Energy	3.316	-8.052	-1.852	-7.564



Results for similar CSD molecules

Name of the protein	Method used	CSD Ref code of the molecule	Energy (kcal)
HIV-1 Nef	VGA	IFEFOO	-11.43518
	IVGA3D	ADAPII	-18.39
HIV-1 Protease	VGA	VEHMUQ	-17.7638
	IVGA3D	SEWZOJ	-24. 76



Hydrogen Bond details

- Hydrogen Bonds For HIV-1 Nef Protein

- IVGA3D

Donor	Acceptor	Distance (Å)
Ligand:1:HH	Protein:83:N	1.92
Ligand:3:HH	Protein:120:O	1.87

- VGA

Donor	Acceptor	Distance (Å)
Ligand:2:HH	Protein:83:NH	2.32



Hydrogen Bond details

contd..

- Hydrogen Bonds For HIV-1 Protease

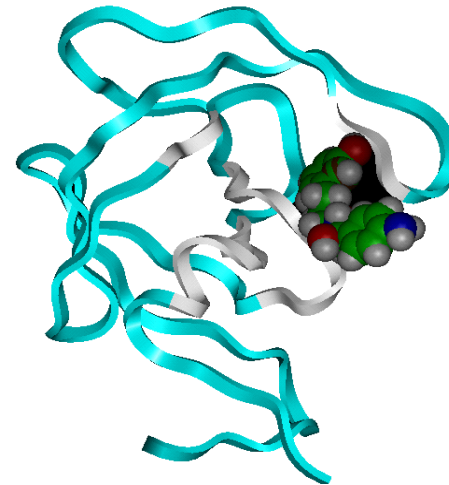
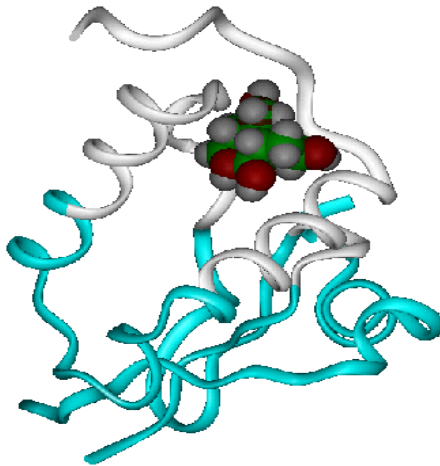
- IVGA3D

Donor (Å)	Acceptor	Distance
Ligand:2:HH	Protein:87:NH	2.14
Ligand:4:HH	Protein:27:O	0.25
Ligand:7:HH	Protein:48:O	2.48

- VGA

Donor (Å)	Acceptor	Distance
Protein:87:HH	Ligand:4:OH	2.13
Ligand:4:HH	Protein:87:NH	1.46

HIV-1 Nef protein and HIV-1 Protease docked with a molecule designed by IVGA3D



- Color code for Proteins: Cyan: protein, White: Active site
- Color code for ligands: White: Hydrogen, Red: Oxygen, Green: Carbon



Observations

- It is found that VGA designed molecule is associated with lower Van der Waals energy values as compared to the fixed string length GA based method.
- Moreover, it is found that the structure of the evolved molecule is such that it is amenable to stable configurations because of the presence of hydrogen bonds.
- Molecule designed using fixed length GA had heavier molecules
 - therefore, may be unstable.



Conclusions and Further Work

- An Improved VGA based technique for ligand design is proposed
 - no assumption regarding the size of the tree
 - Modified crossover and mutation operators are used.
- Proposed method found to provide solutions having characteristics amenable to stability
- Scope for further work
 - Need to analyze in 3 dimensions
 - Consider other optimizing criteria and multi-objective optimization algorithms
 - Consider structures other than tree



Publications

Books

- S. Bandyopadhyay and S. K. Pal, Classification and Learning Using Genetic Algorithms: Applications in Bioinformatics and Web Intelligence, Springer, Heidelberg, 2007.
- S. Bandyopadhyay, U. Maulik and J. T. L. Wang, (eds.), Analysis of Biological Data: A Soft Computing Approach, World Scientific, Singapore, 2007.

Articles

- S. Bandyopadhyay, A. Bagchi and U. Maulik, ``Active Site Driven Ligand Design: An Evolutionary Approach'', *Journal of Bioinformatics and Computational Biology*, vol. 3, No. 5, pp. 1053-1070, 2005.
- S. Santra and S. Bandyopadhyay, "Grid Count Tree Based Method For Efficient Outlier Detection", *Proceedings of the International Conference on Emerging Applications of IT*, February 10-11, Kolkata, India, pp. 309-312, 2006.
- S. Bandyopadhyay, A. Mukhopadhyay and U. Maulik, "An Improved Algorithm for Clustering Gene Expression Data", *Bioinformatics*, Oxford University Press, vol. 23, no. 21, pp. 2859-2865, 2007.
- S. Bandyopadhyay and S. Santra, "A Genetic Approach for Efficient Outlier Detection in Projected Space", *Pattern Recognition*, vol. 41, no. 4 pp. 1338-1349, 2008.
- S. Bandyopadhyay, S. Santra, U. Maulik and H. Muehlenbein, "In Silico Design of Ligands Using Properties of Target Active Sites", *Analysis of Biological Data: A Soft Computing Approach*, World Scientific, pp. 184-201, 2007.



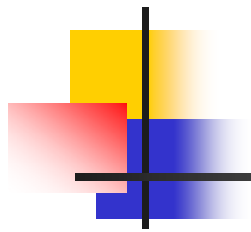
References

- N. M. Luscombe, D. Greenbaum, M. Gerstein, ``What is Bioinformatics: A Proposed definition and Overview of the field'', *Methods Inf. Med.*, vol. 40, pp. 346-358, 2001.
- J. Setubal and J. Meidanis, “Introduction to computational biology”, Brooks Cole, 1997.
- Jason T.L. Wang, Qi Cheng. Ma, Dennis Shasha, Cathy H. Wu, ``New Techniques for Extracting Features from Protein Sequences'', *IBM Systems Journal*, Special Issue on Deep Computing for the Life Sciences, vol-40, no-2, pp. 426-441, 2001.
- R. Chakraborty, S. Bandyopadhyay and U. Maulik, ``Extracting Features for Protein Sequence Classification", *Intl. Conf. on IT: Prospects and Challenges (ITPC)*, 2003.
- S. Bandyopadhyay, “An Efficient Technique for Superfamily Classification of Amino Acid Sequences: Feature Extraction, Fuzzy Clustering and Prototype Selection", *Fuzzy Sets & Systems*, vol. 152, pp. 5-16, 2005



References

- S. Bandyopadhyay, A. Bagchi and U. Maulik, “Active Site Driven Ligand Design: An Evolutionary Approach”, *J. of Bioinformatics and Computational Biology*, vol. 3, No. 5, pp. 1053-1070, 2005.
- S. Bandyopadhyay, “An Efficient Technique for Superfamily Classification of Amino Acid Sequences: Feature Extraction, Fuzzy Clustering and Prototype Selection”, *Fuzzy Sets & Systems*, vol. 152, pp. 5-16, 2005]
- S. S. Ray, S. Bandyopadhyay, and S. K. Pal, “Genetic Operators for Combinatorial Optimization in TSP and Microarray Gene Ordering”, *Applied Intelligence* (accepted).
- S. Bandyopadhyay, A. Mukhopadhyay and U. Maulik, “An Improved Algorithm for Clustering Gene Expression Data”, *Bioionformatics*, Oxford University Press, vol. 23, no. 21, pp. 2859-2865, 2007.



Thank you..

sanghami@isical.ac.in

<http://www.isical.ac.in/~sanghami>