

# Definition

Genetic search algorithm  
natural selection  
natural genetics  
→ biology ~~is~~ borrowed

- Goldberg, 1989: “Genetic Algorithms are search algorithms based on the mechanics of natural selection and natural genetics.”
- A genetic algorithms is a directed random search procedure

**A genetic algorithm borrows ideas from biology to search a solution space for a target value.**

একটা solution  $\rightarrow$  বেটা search এর direction

এখানে একটা solution না নিয়ে multiple solution নিয়ে progress করছি।  
multiple direction

$$S_{i+1} = S_i + n_i \times d_i$$

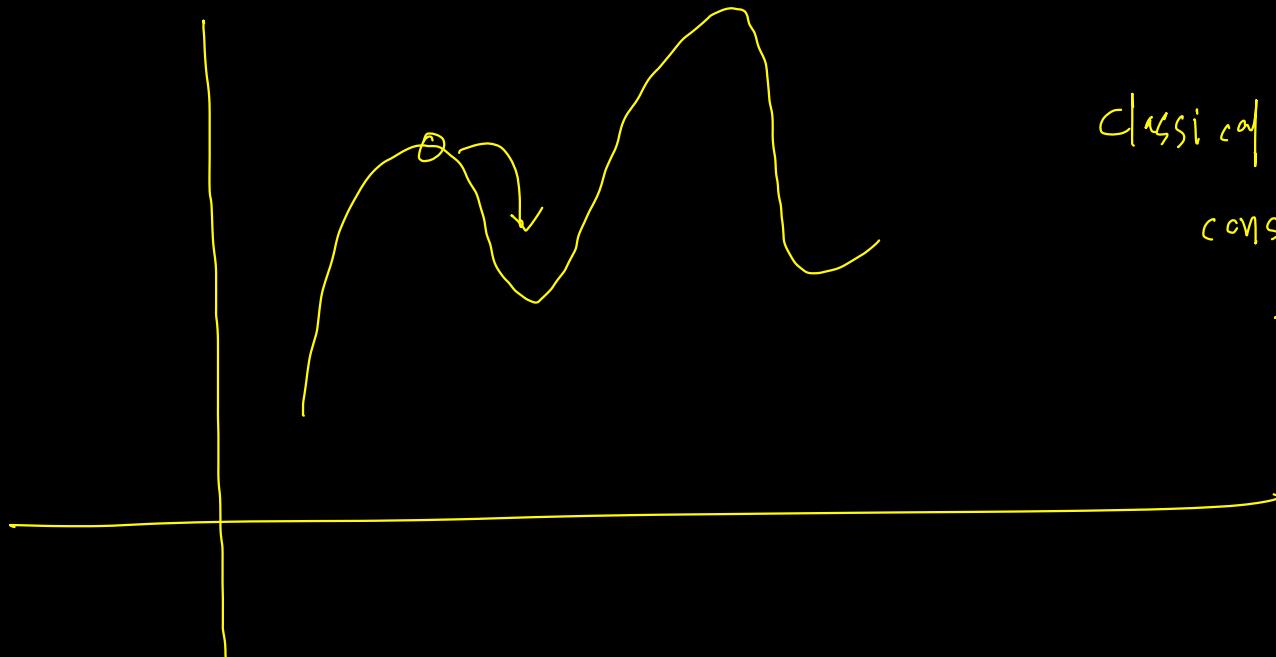
set of solution /  
Population of solution

classical algo এর মাঝে genetic এর diff  $\rightarrow$  multiple solution নিয়ে গত

একটা soln (out of the generated random solutions) luckily সমাধান প্রয়োজন requirements  
fulfil করে, তাহলে done.

2nd step evaluation  $\uparrow$

classical algo 70 soln কে modify করে নতুন soln generate করি  $\rightarrow$  deterministic  
(classical algo (4))  
natural selection and genetics randomness এরে ইউনিট deterministic না, unlike  
classical algo.

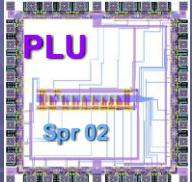


classical algo downwards  
 consider করেনা, genetic algo  
 তেওঁদের বিচু নই

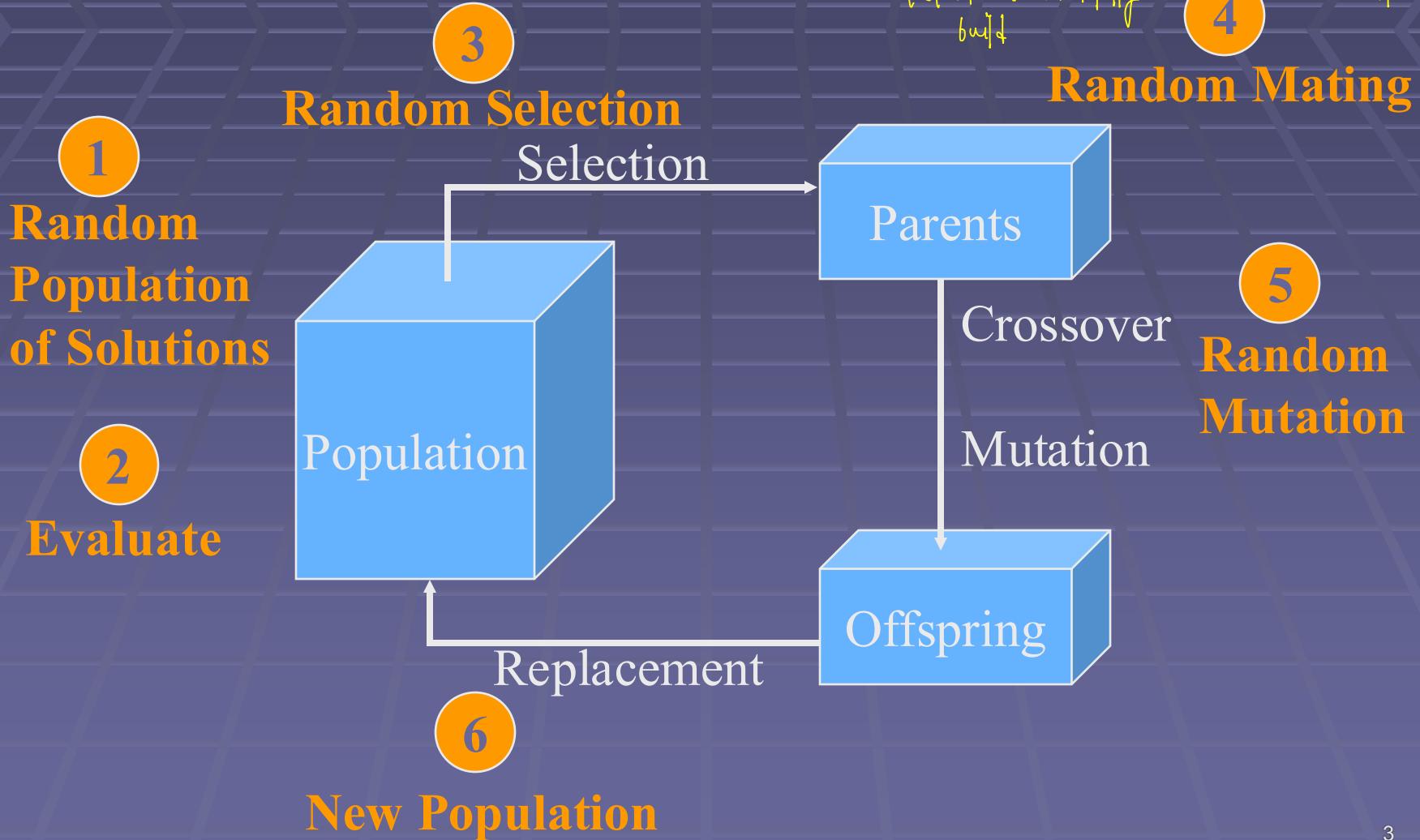
1. random population generate

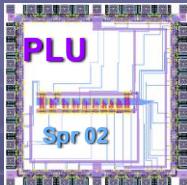
2. evaluate

→ fitness  
 function → solution কৃত কোটি/ একাধি measure করি

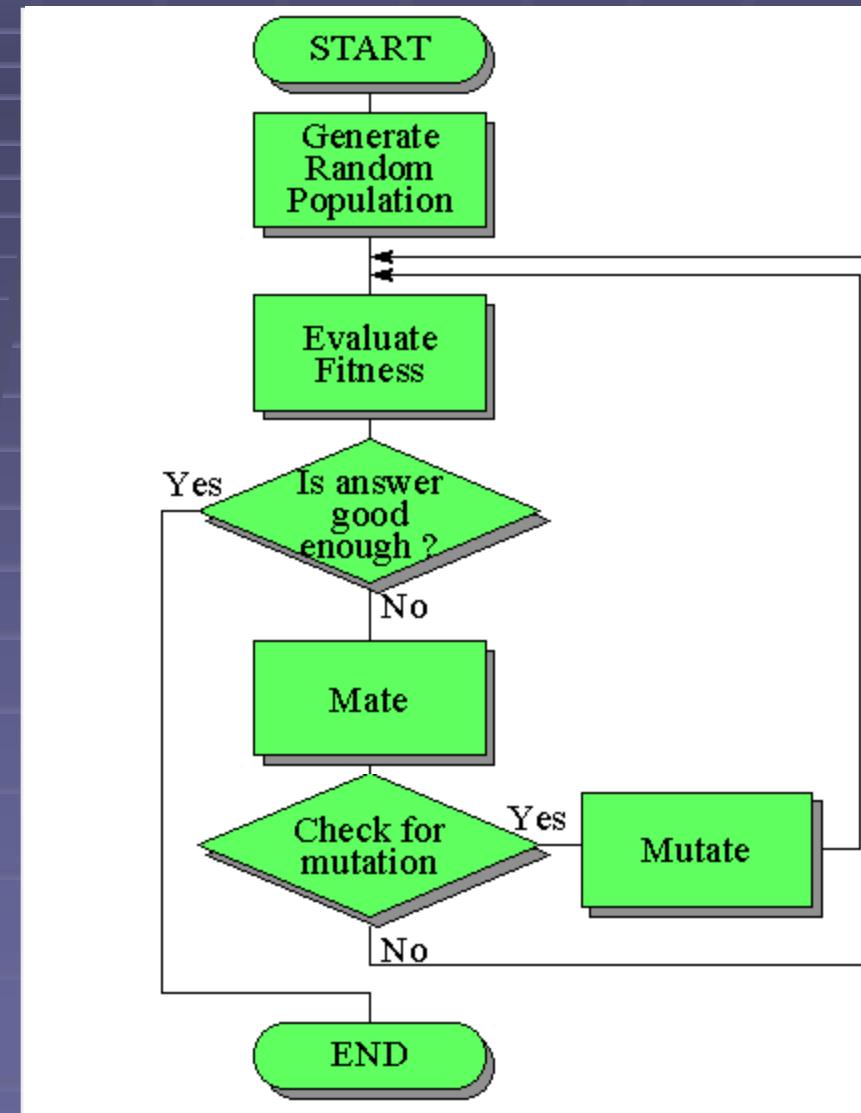


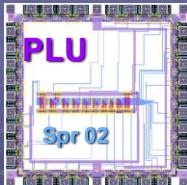
# General Approach





# GA Flow Chart

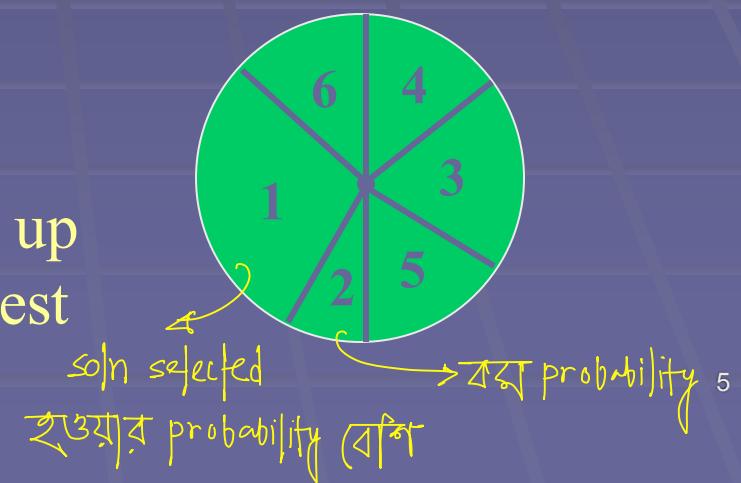


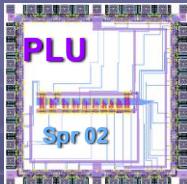


# Parent Selection

- The process in which individual strings in the population are selected to contribute to the next generation is called **parent selection**
  - based on fitness
  - strings with a high fitness have a higher probability of contributing one or more offspring to the next generation
- Biased Roulette Wheel Selection

When you spin the wheel, items 1 and 5 have the greatest chance of coming up while item 2 has the smallest





6 bit binary no. generate এর দশমিক মানের সর্বোচ্চ

let population size = 5  $\rightarrow$  5 bit এর সুব মানের

# Example

- Given the following population of chromosomes, select two parents:

Chromosome	Fitness	% fitness	cf
(1 0 1 0 0 1)	23	0.28	0.28
(1 1 1 0 0 1)	12	0.15	0.43
(0 1 1 0 1 1)	25	0.30	0.73
(0 1 0 1 1 0)	5	0.06	0.79
(0 1 1 0 1 0)	17	0.21	1.00
<hr/>		Total Fitness	82

→ normal binary to decimal converter এর

গুরুত্ব fitness score

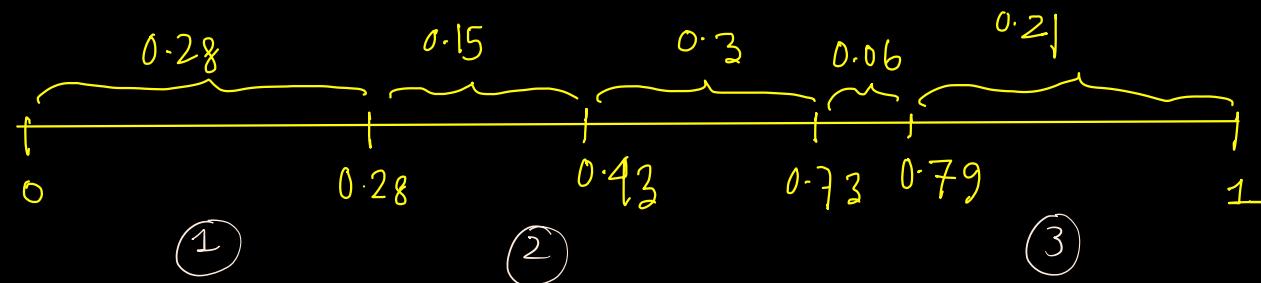
Find the total fitness of the population

Find the %fitness of each element

Find the cumulative fitness

Now, throw a random number between 0 and 1, if it is in the range 0 to 0.28 select element 1, between 0.28 and 0.43 select element 2,...

fitness func.  $\rightarrow$  binary to decimal converter  
decimal score calculate করা



0 to 1 random number generate করালো

①, ②, ③ এ range এ অন্যান্য probability কেন্দ্র।

Random solution মূলোর অব্যু মাত্রের fitness score ওপো তাদের child generate করতে দিব।

অব্যু natural selection এর পর natural genetic participation করব।

৫টি random number generate  $\rightarrow$  যেই range এ পড়বে যেই solution selected.

সো better solution মূলোর multiple copy করতে পারে, আবার ক্ষেত্রে solution বাদ পড়তে পারে,

process  $\frac{\text{random}}{\text{as random}} + \frac{\text{deterministic}}{\text{fitness score based}}$   
 number generate করালো

Initial Population



Fitness based selection (survival of the fittest)

natural selection

Mating pool → ২<sup>nd</sup> generation operator apply করা

Mating pool থেকে  $\geq 2$  parent randomly select.

যদি selected হলো

থেকে expected

সোন না নাই

algo random point থেকে start  
mutation

Natural genetics

Cross over — Involve two parents  
Mutation — Involve an offspring

Offspring এবং তেজস্ব mutation operation apply করবে,  $\rightarrow$  equivalent to the environmental factor

GA  $\rightarrow$  Uses both crossover and mutation operators

EPI  $\rightarrow$  Uses only mutation operator  
↳ evolutionary  
programming

Exploration  
 $\downarrow$   
long jump

vs  
Exploitation  
 $\downarrow$   
short jump

mutation দিয়ে

শুরুর দিয়ে যদি mutation operator ব্যবহার  
(একটি চোর না সেটি change করানো) শৈষণ  
ভিত্তে কোটি থারে  $\rightarrow$  that means শুধু  
exploration (কিন্তু exploitation).

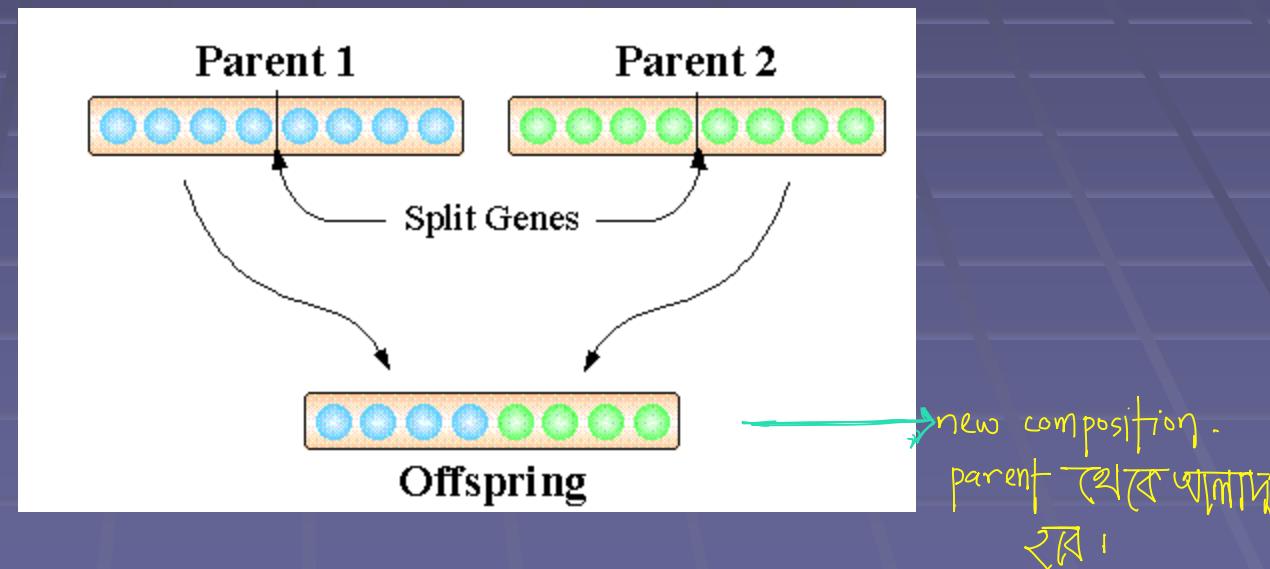
GA  $\geq$  CT operator.

In general crossover এর main operator রিমেক্স use করে  
mutation এ fine tuning রিমেক্স use করে

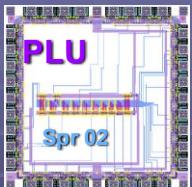
both parent এর বিতু বিতু  
attribute কে inherit করে।  
বিতুটা randomness আছে।

# Crossover

- Once two parents are selected, their chromosomes are mixed to create the children for the next generation

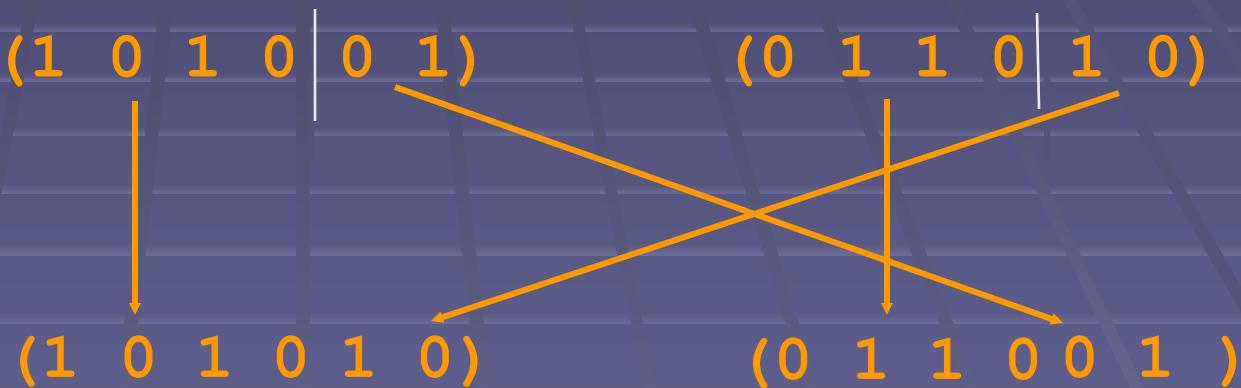


Called single point crossover



# Example

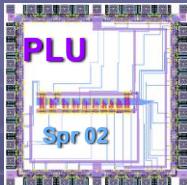
- Assume the parents selected from the previous example are: (1 0 1 0 | 0 1)



These are the two children which are now part of the next generation

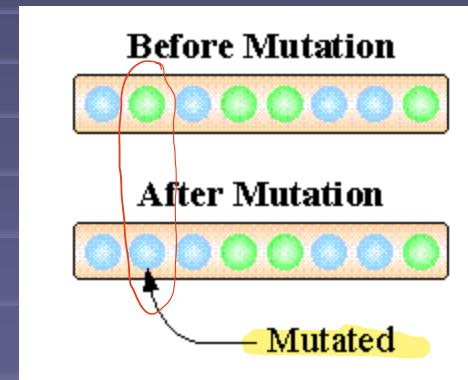
Find a random crossover point

Swap the bits after the crossover point



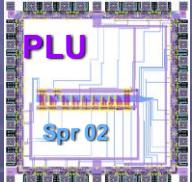
# Mutation

- A bit in a child is changed (from 1 to 0 or from 0 to 1) at random



This is a small probability event

The effect is to prevent a premature convergence to a local minimum or maximum



# GA Performance

- Increasing diversity by genetic operators
  - mutation
  - Recombination
- Decreasing diversity by selection
  - of parents
  - of survivors

## Three operators of GA

- Selection
- Crossover
- Mutation

যদি শুরু মাত্র fitness based selection নেই তাহলে শুরু best solution পুরো যাবে

→ এই column focus

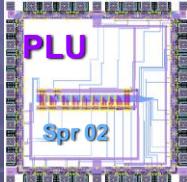
Chromosome	Fitness	% fitness	cf
(1 0 1 0 0 1)	23	0.28	0.28
(1 1 1 0 0 1)	12	0.15	0.43
(0 1 1 0 1 1)	25	0.30	0.73
(0 1 0 1 1 0)	5	0.06	0.79
(0 1 1 0 1 0)	17	0.21	1.00
<hr/> Total Fitness		82	

শুরু best soln পুরো pool নিনে → diversity ↓

→ একই রকম child তৈরি করে

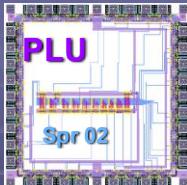
→ একই type পুরো similar type করে করে

better, different type  
এক soln পাওয়ার probability করে  
যাবে



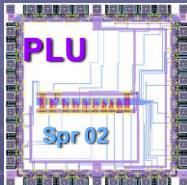
# Effects of the Genetic Operators

- Using selection alone will tend to fill the population with copies of the best individual from the initial population
- Using selection and crossover will tend to cause the algorithm to converge on a good but sub-optimal solution
- Using mutation alone induces a random walk through the search space
- Using selection and mutation creates a parallel, noise-tolerant, hill climbing algorithm



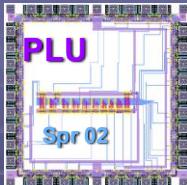
# The Algorithm

- randomly initialize population( $t$ )
- determine fitness of population( $t$ )
- repeat
  - select parents from population( $t$ )
  - perform crossover on parents creating population( $t+1$ )
  - perform mutation on population( $t+1$ )
  - determine fitness of population( $t+1$ )
- until best individual is good enough



# GA Applications

- GA's can be applied to several parts of the physical design problem
  - Partitioning
  - Placement
  - Other . . .
- Scope of the Partitioning problem
  - A standard layout benchmark suite has circuits ranging from 13,000 to 200,000 nodes.
  - The number of links range from 50,000 to 800,000

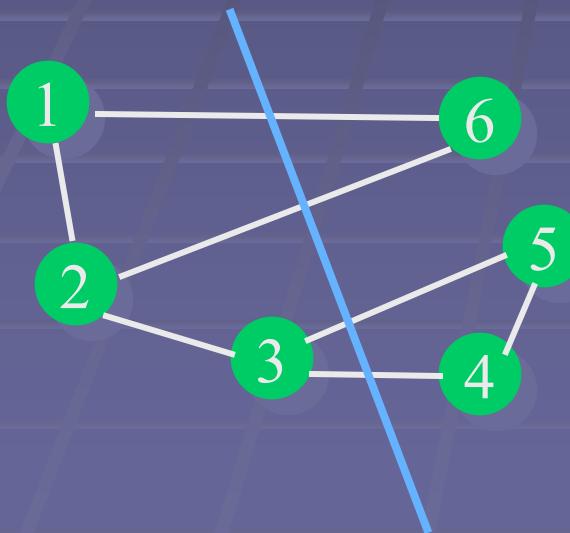


# Representation

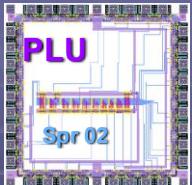
- A graph partition is represented by a binary string

Each node is represented by a bit

The 0 nodes are in one segment, the 1 nodes are in the other segment

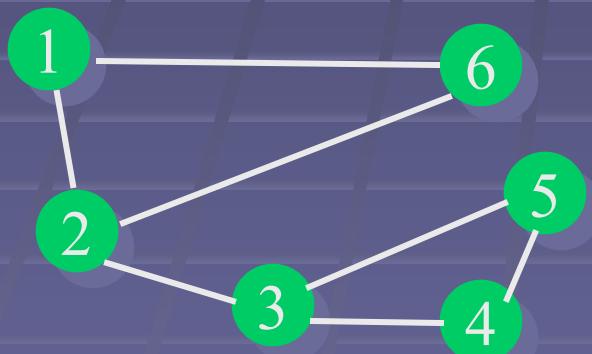


(0 0 0 1 1 1)



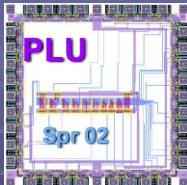
# Population

- A random population of binary strings is produced



(0 0 0 1 1 1)	4
(1 0 0 1 0 1)	4
(1 0 1 1 0 0)	5
(1 0 0 1 1 0)	4

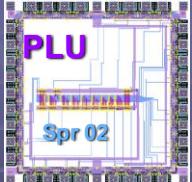
The fitness is the number of links



# Crossover

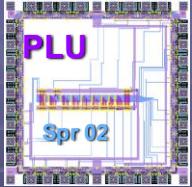
- Parents are randomly selected (with a bias to the better fit elements)
- The parents are combined to create two children (single point crossover)

$$\begin{array}{c|ccccc} (0 & 1 & 0 & 1 & 0 & 1) \\ \hline (1 & 0 & 0 & 1 & 0 & 1) \end{array}$$
$$\begin{array}{ccccc} (1 & 1 & 0 & 1 & 0 & 1) \\ (0 & 0 & 0 & 1 & 0 & 1) \end{array}$$



# Mutation

- For a small number of the new population elements perform a mutation operation
  - Randomly select two nodes and swap their positions



# Thank you