General GA Structure

- encoding
- objective function (*fitness*) - evaluation
- initial population of solutions
  - number of solutions
  - generating solutions
- evolution operators (move operators)
  - mutation – stochastic perturbation, usually to neighboring solution
  - recombination – selection of *parent* solutions and blending to form *child* solutions
  - population maintenance method – replacing old solutions with new solutions
- termination method

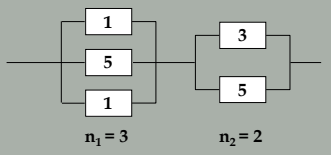
1

Solution Encoding – Example I

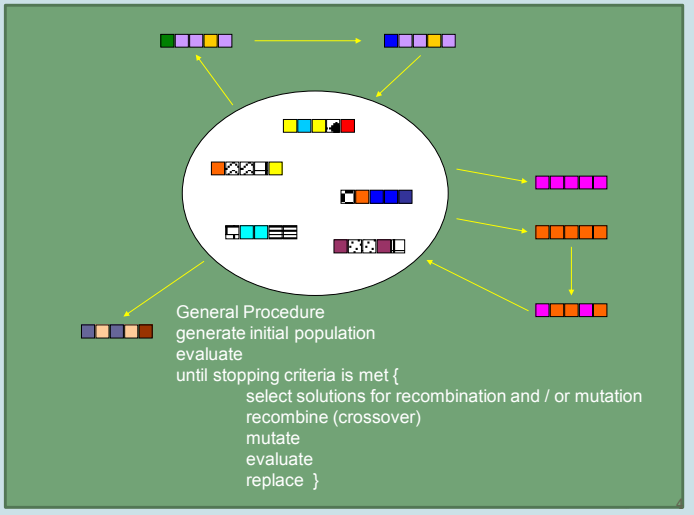
- A system has three subsystems (s=3) connected in series, where each subsystem consists of one hardware component and one software component.
- There are 4 available component types for HW and SW components : type 0, 1, 2, 3
- V = (1 3 |2 2 | 0 3) = (01 11| 10 10 | 00 11)

Solution Encoding – Example II

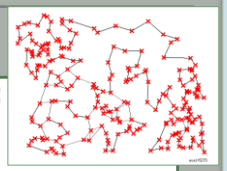
- A system has two subsystems (s=2) with 5 available parts for first (m_1=5) and 7 available parts for second (m_2=7) subsystem
- n_{max} is set to 4



- V = (1 1 5 6 | 3 5 8 8)



Combinatorial Example - TSP



Cities	A	B	C	D	E
A	-	10	5	2	8
B		-	6	5	4
C			-	9	2
D				-	7
E					-

objective: shortest "tour" that visits all cities only once and returns to starting city

search space = $\frac{(n-1)!}{2}$ = (4)(3)(2)(1)/2 = 12

5

Step One

Choose random initial solutions (population):

A B C D E
Fitness = 10 + 6 + 9 + 7 + 8 = 40

A D C B E
Fitness = 2 + 9 + 6 + 4 + 8 = 29

E D B C A
Fitness = 7 + 5 + 6 + 5 + 8 = 31

6

Step Two

Parent Selection:
- tournament selection
- roulette wheel

1. Randomly select 2 solutions, the ones with fitness = 31 and = 29. Fitness = 29 is better, so it is parent 1.
2. Randomly select a solution from the remaining population, let's say the one with fitness = 31. This is parent 2.

7

Step Three

Crossover the two parents to create one or more children.

parent 1	A	D	C	B	E	
parent 2	E	D	B	C	A	
Uniform crossover with repair	A	D	B	B	E	*initial*
Repair operator	A	D	B	C	E	child

8

Step Four

Evaluate child.

A D B C E

Fitness = 2 + 5 + 6 + 2 + 8 = 23

The child replaces the worse solution in the population (the one with fitness = 40).

9

Step Five

Mutate with small probability.

Randomly choose a solution, say the one with fitness = 31.

E D B C A

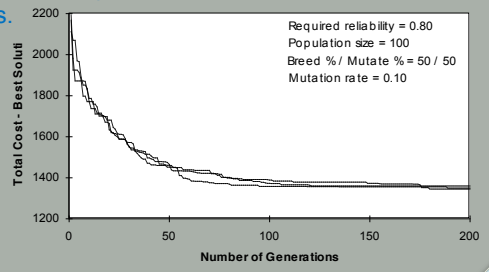
Swap two cities, randomly to create mutated solution;

E D B A C
Fitness = 7 + 5 + 10 + 5 + 2 = 29

10

Step Six

Repeat parent selection, child creation, population maintenance & mutation until either a maximum number of generations is reached or the best solution does not change for a specified number of generations.



11