中山大學
SUN YAT-SEN UNIVERSITY

Lecture 16
Optimization Algorithms (III)

Algorithm Design

zhangzizhen@gmail.com

Metaheuristics

- Single-Solution Based Metaheuristics
  - Simulated Annealing (模拟退火)
  - Tabu Search (禁忌搜索)
  - Iterated Local Search
  - Variable Neighborhood Search
- Population Based Metaheuristics
  - Genetic Algorithm (遗传算法)
  - Scatter Search
  - Ant Colony Optimization
  - Particle Swarm Optimization

Tabu Search

- TS was invented by Fred Glover (http://leeds-faculty.colorado.edu/glover/).
- TS is one of the most widespread Single-metaheuristics.
- The use of memory, which stores information related to the search process, represents the particular feature of TS.
- TS behaves like a steepest LS algorithm, but it accepts non-improving solutions to escape from local optima when all neighbors are non-improving solutions.
- TS works in a deterministic manner.

Tabu Search

- The best solution in the neighborhood is selected as the new incumbent （现任的）solution; this may generate cycles.
- TS discards some neighbors that have been visited previously to avoid cycles.
- TS manages a memory of the solutions or moves recently applied, which is called the tabu list, which constitutes the short-term memory.
- At each iteration of TS, the short-term memory is updated.

TS Components

- Tabu list（禁忌表）: The goal of using the short-term memory is to avoid cycles. Storing the list of all visited solutions is not practical for efficiency issues.
- Aspiration criterion（特赦原则）: selecting a tabu move if it generates a solution that is better than the best found solution so far.
- Intensification (medium-term memory): The medium-term memory stores the elite (e.g., best) solutions found during the search.
- Diversification (long-term memory): The long-term memory stores information on the visited solutions along the search. It explores the unvisited areas of the solution space.

TS Algorithm

Template of tabu search algorithm.

```
s = s_0 ; /* Initial solution */
Initialize the tabu list, medium-term and long-term memories ;
Repeat
  Find best admissible neighbor s' ; /* non tabu or aspiration criterion holds */
  s = s' ;
  Update tabu list, aspiration conditions, medium and long term memories ;
  If intensification_criterion holds  Then intensification ;
  If diversification_criterion holds  Then diversification ;
Until Stopping criteria satisfied
Output: Best solution found.
```

Tabu List

- Store the recent history of the search.
- Recording all visited solutions during the search high complexity of data storage and computational time.
- Recording the last k visited solutions.
- The most popular way to represent the tabu list is to record the move attributes.

Example—Tabu list based on move attributes

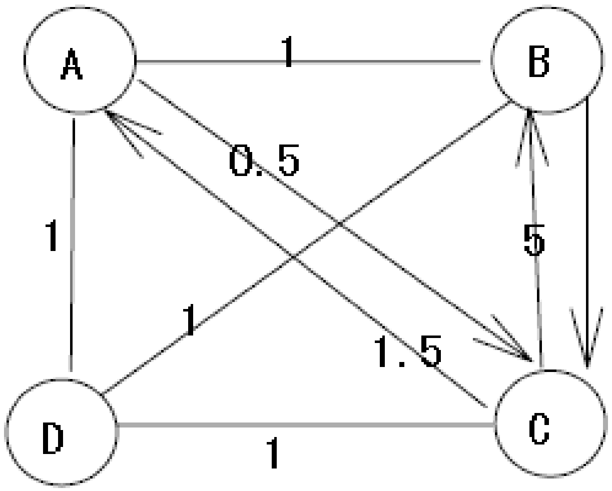
- Let us consider a permutation optimization problem, where the neighborhood is based on exchanging two elements of the permutation.
- Given a permutation π, a move is represented by two indices (i, j). This move generates the neighbor solution π' such that

$$\pi'(k) = \begin{cases} \pi(k) & \text{for } k \neq i \text{ and } k \neq j \\ \pi(j) & \text{for } k = i \\ \pi(i) & \text{for } k = j \end{cases}$$

- The inverse move (j, i) may be stored in the tabu list and is forbidden for a certain number of iterations, called tabu tenure.
- A stronger tabu representation may be related to the indices i and j. This will disallow any move involving the indices i and j.

Example -- TSP

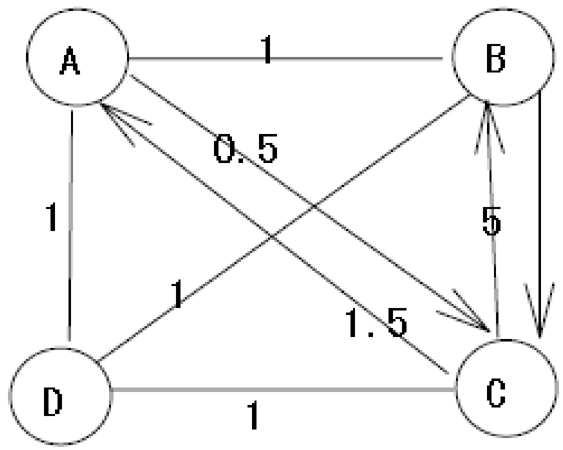
$$D=(d_{ij})=\begin{bmatrix} 0 & 1 & 0.5 & 1 \\ 1 & 0 & 1 & 1 \\ 1.5 & 5 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$



- Initial solution: x_0=(ABCD)， $f(x_0)=4$
- City A is the starting and ending vertex
- Neighborhood operator: 2-swap (swap a pair of cities)。

Example -- TSP

- Step 1:



Solution:

A	B	C	D

$f(x^0)=4$

Tabu list:

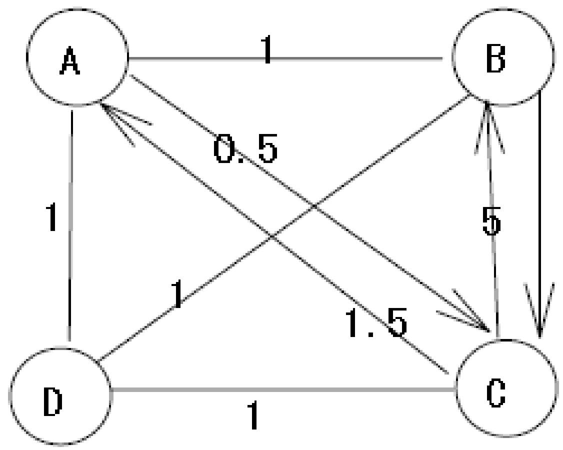
	B	C	D
A			
B			
C			

Candidate solution:

Swap	Fitness
CD	4.5
BC	7.5
BD	8

Example -- TSP

- Step 2:



Solution:

A	B	D	C

$f(x^1)=4.5$

Tabu list:

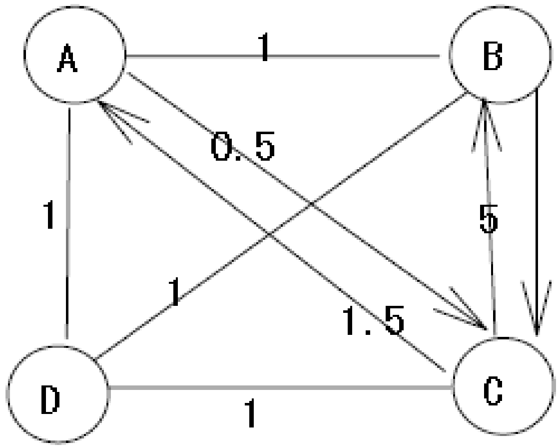
	B	C	D
A			
B			
C			3

Candidate solution:

Swap	Fitness
CD	4.5 T
BC	3.5 ☺
BD	4.5

Example -- TSP

- Step 3:



Solution:

A	C	D	B

$f(x^2)=3.5$

Tabu list:

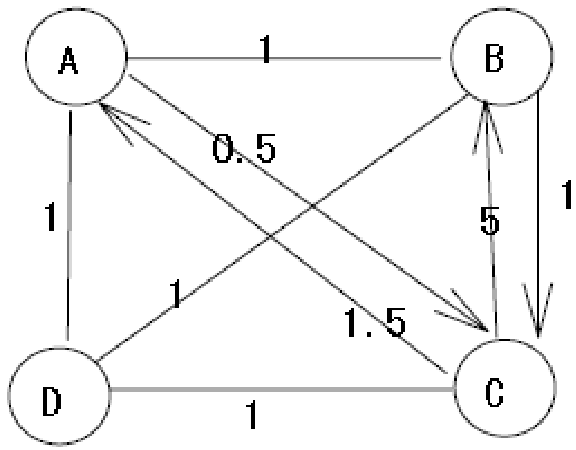
	B	C	D
A			
	B	3	
		C	2

Candidate solution:

Swap	Fitness
CD	8 T
BC	4.5 T
BD	3.5

Example -- TSP

- Step 4:



Note the length of the Tabu list

Solution:

A	C	B	D

$f(x^3)=7.5$

Tabu list:

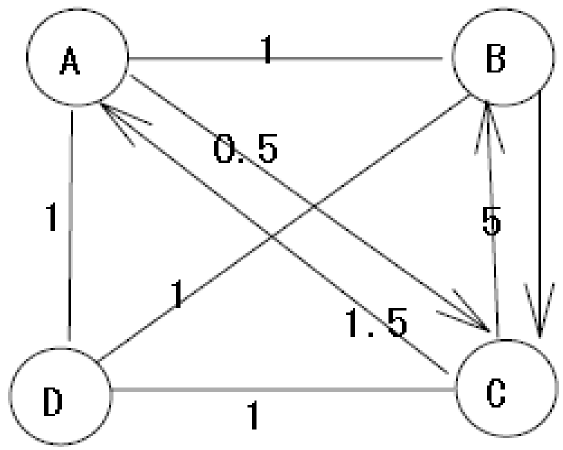
	B	C	D
A			
B		2	3
C			1

Candidate solution:

Swap	Fitness
CD	4.5 T
BC	4.5 T
BD	3.5 T

Example -- TSP

- Step 4:



Solution:

A	C	B	D

$f(x^3)=7.5$

Tabu list:

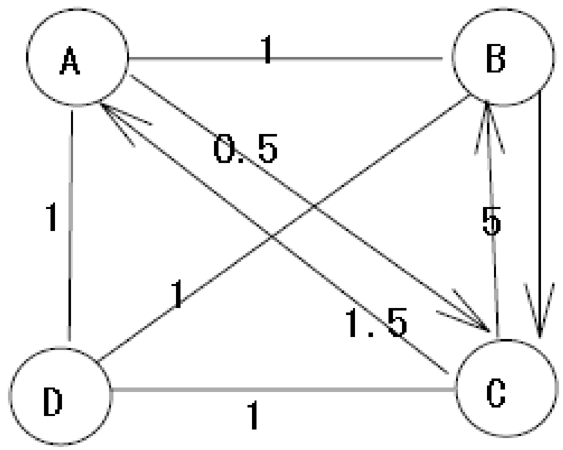
	B	C	D
A			
B		1	2
C			0

Candidate solution:

Swap	Fitness
CD	4.5 ☻
BC	4.5 T
BD	3.5 T

Example -- TSP

- Step 5:



Solution:

A	C	B	D

$f(x^4)=4.5$

Tabu list:

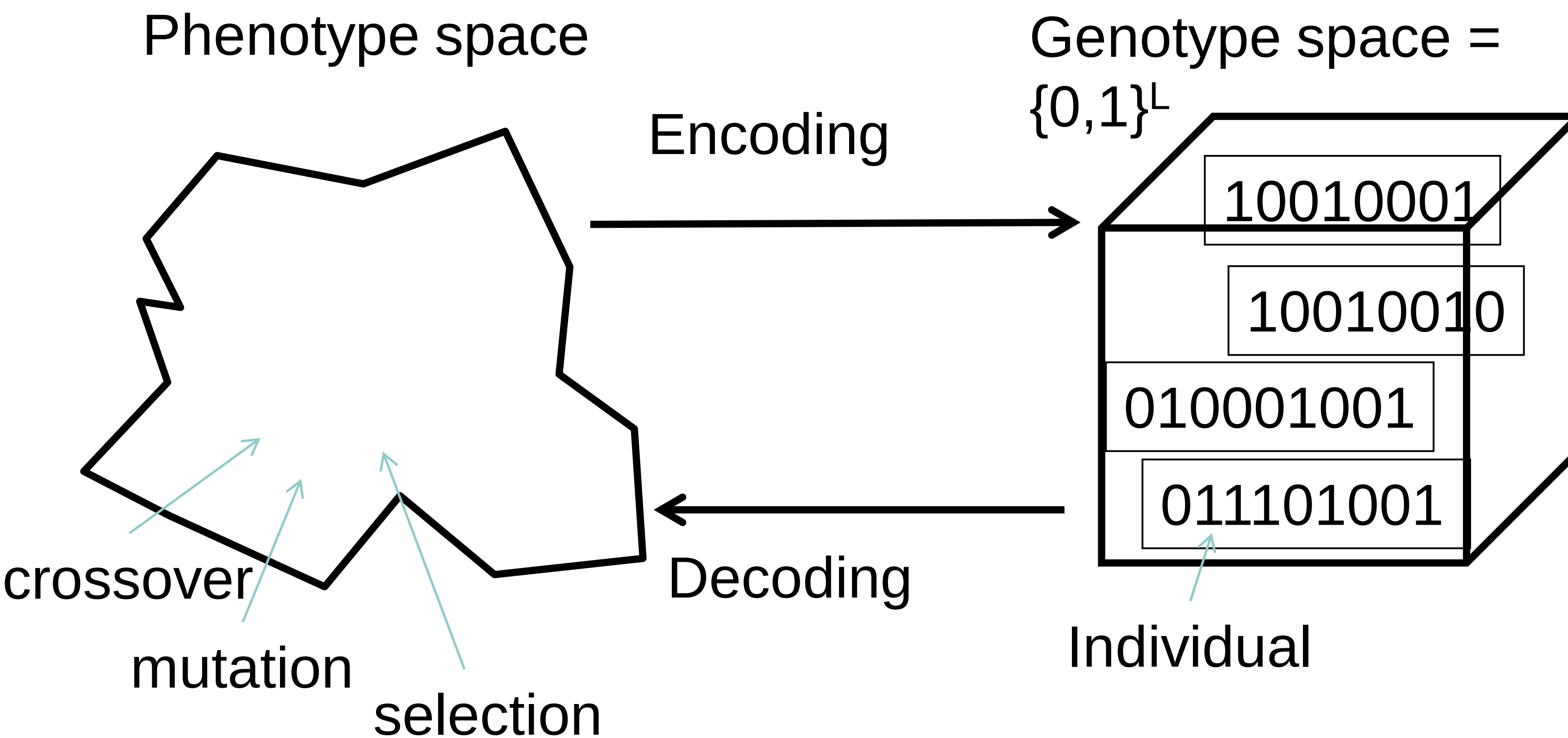
	B	C	D
A			
B		0	1
C			2

Candidate solution:

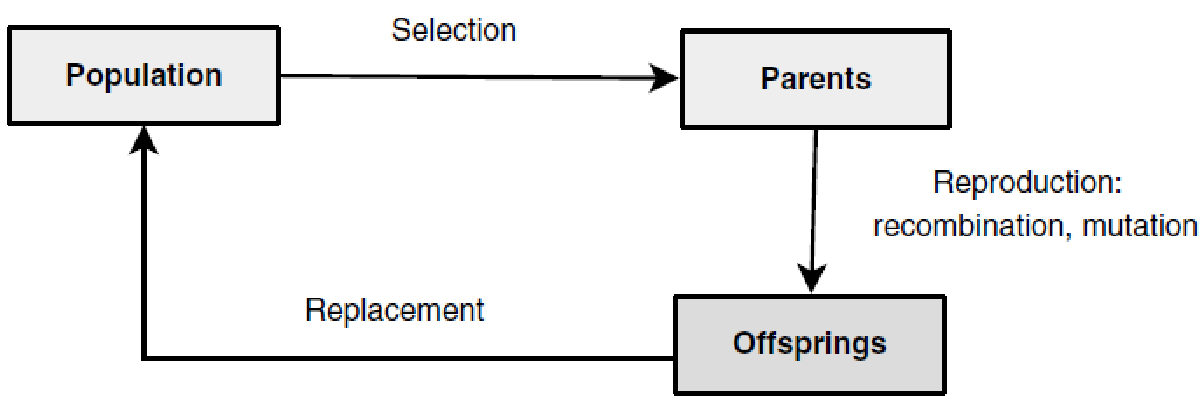
Swap	Fitness
CD	7.5 T
BC	8 ☻
BD	4.5 T

Genetic Algorithm (GA)

- A genetic algorithm (GA) is a search heuristic that mimics the process of natural selection.

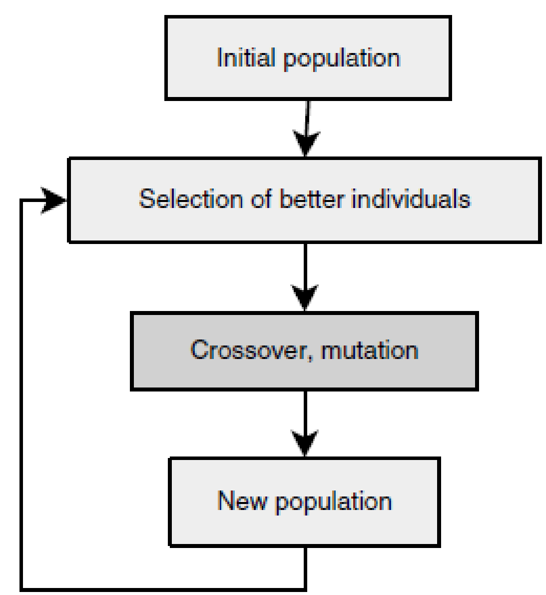


Genetic Algorithm (GA)



Template of an evolutionary algorithm.

Generate($P(0)$) ; /* Initial population */
$t = 0$;
While not Termination_Criterion($P(t)$) **Do**
    Evaluate($P(t)$) ;
    $P'(t) \quad = \text{Selection}(P(t))$;
    $P'(t) \quad = \text{Reproduction}(P'(t))$; Evaluate($P'(t)$) ;
    $P(t+1) = \text{Replace}(P(t), P'(t))$;
    $t = t + 1$;
End While
Output Best individual or best population found.



Main Components in Genetic Algorithm

- Representation: the encoded solution is referred as chromosome while the decision variables within a solution are genes.
- Population Initialization: generate a set of initial solutions.
- Objective Function: This is a common search component for all heuristics. In GA, the term fitness function refers to the objective function.
- Selection Strategy: The selection strategy addresses the following question: “Which parents for the next generation are chosen with a bias toward better fitness?”

Main Components in Genetic Algorithm

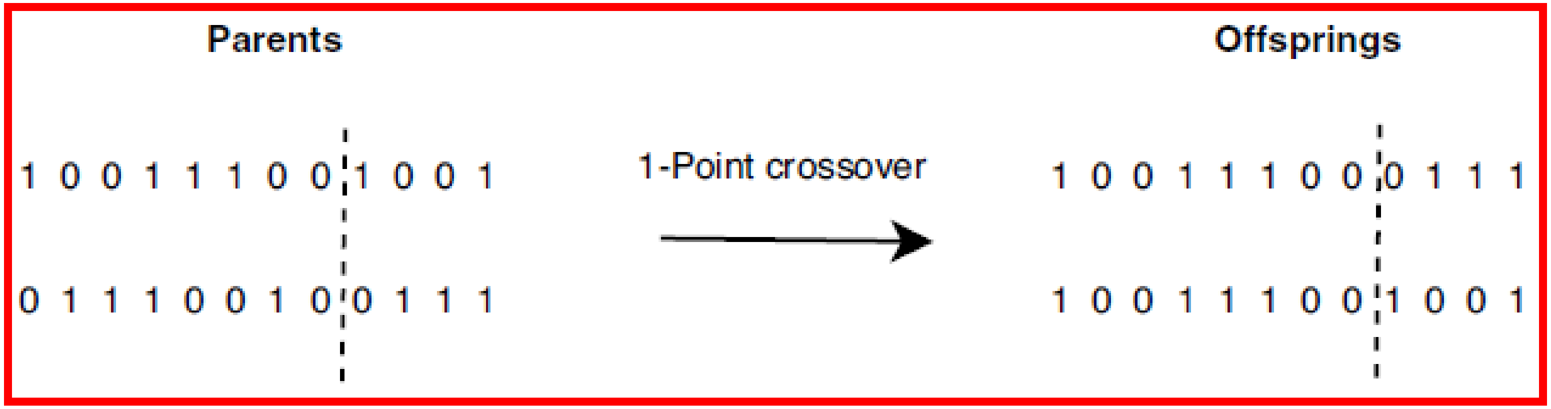
- Reproduction Strategy: The reproduction strategy consists in designing suitable mutation and crossover operators to generated new individuals (offspring).
- Replacement Strategy: The new offsprings compete with old individuals for their place in the next generation.
- Stopping Criteria: This is a common search component for all metaheuristics.

Crossover

- The role of crossover operators is to inherit some characteristics of the two parents to generate offsprings.
- The main characteristic of the crossover operator is heritability. The offsprings should inherit genetic materials from both parents.
- The crossover operator should produce valid solutions.
- The crossover rate p_c ($p_c \in [0, 1]$) represents the proportion of parents on which a crossover operator will act. The most commonly used rates are in the interval [0.45, 0.95].

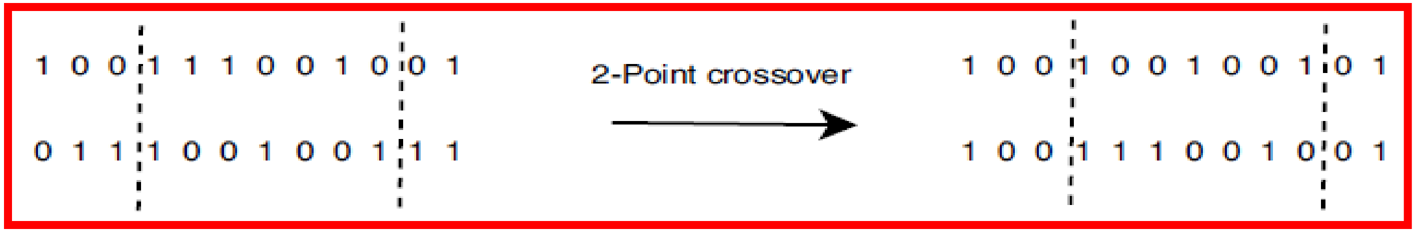
1-Point Crossover

- A crossover site k is randomly selected
- Two offsprings are created by interchanging the segments of the parents.

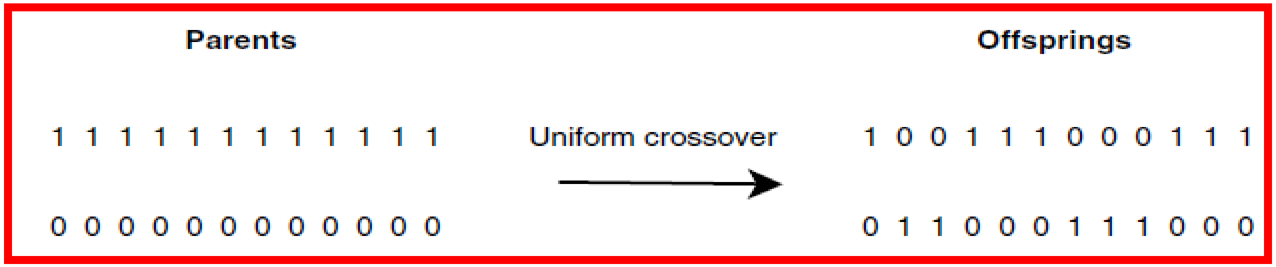


n-Point Crossover

- ***n*-point crossover**
  - *n* crossover sites are randomly selected.
  - The individuals *A*|*BCD*|*E* and *a*|*bcd*|*e* generate two offsprings *A*|*bcd*|*E* and *a*|*BCD*|*e*.

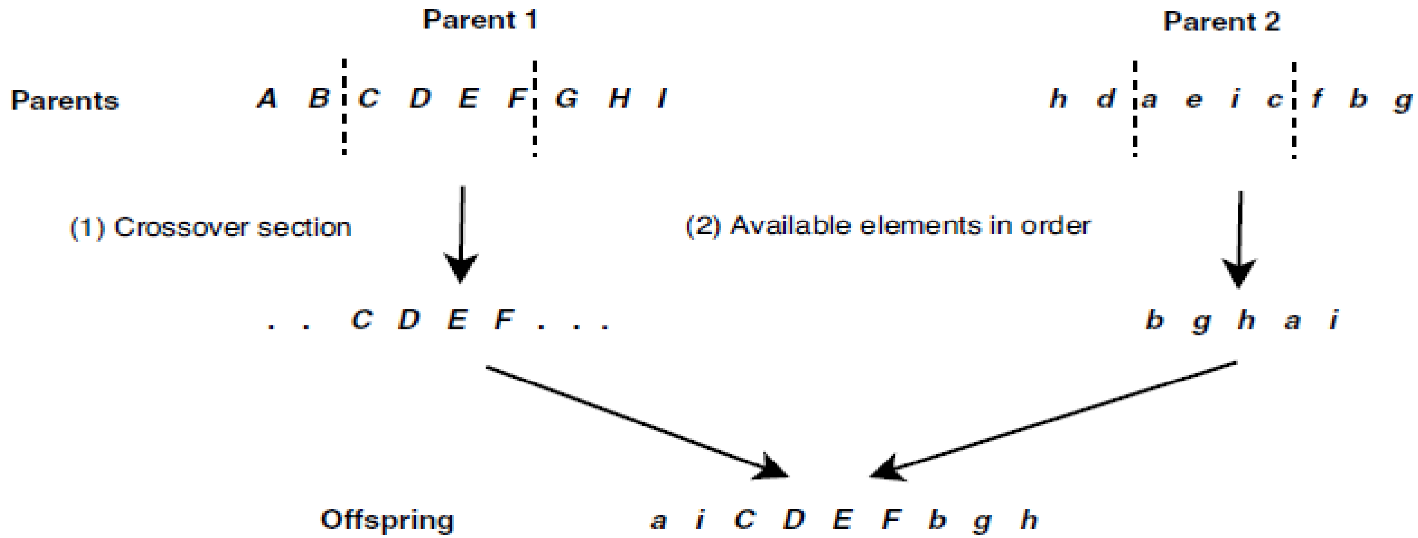


- **Uniform Crossover**
  - Each element of the offspring is selected randomly from either parent.
  - Each parent will contribute equally to generate the offsprings.



Crossover for Permutations

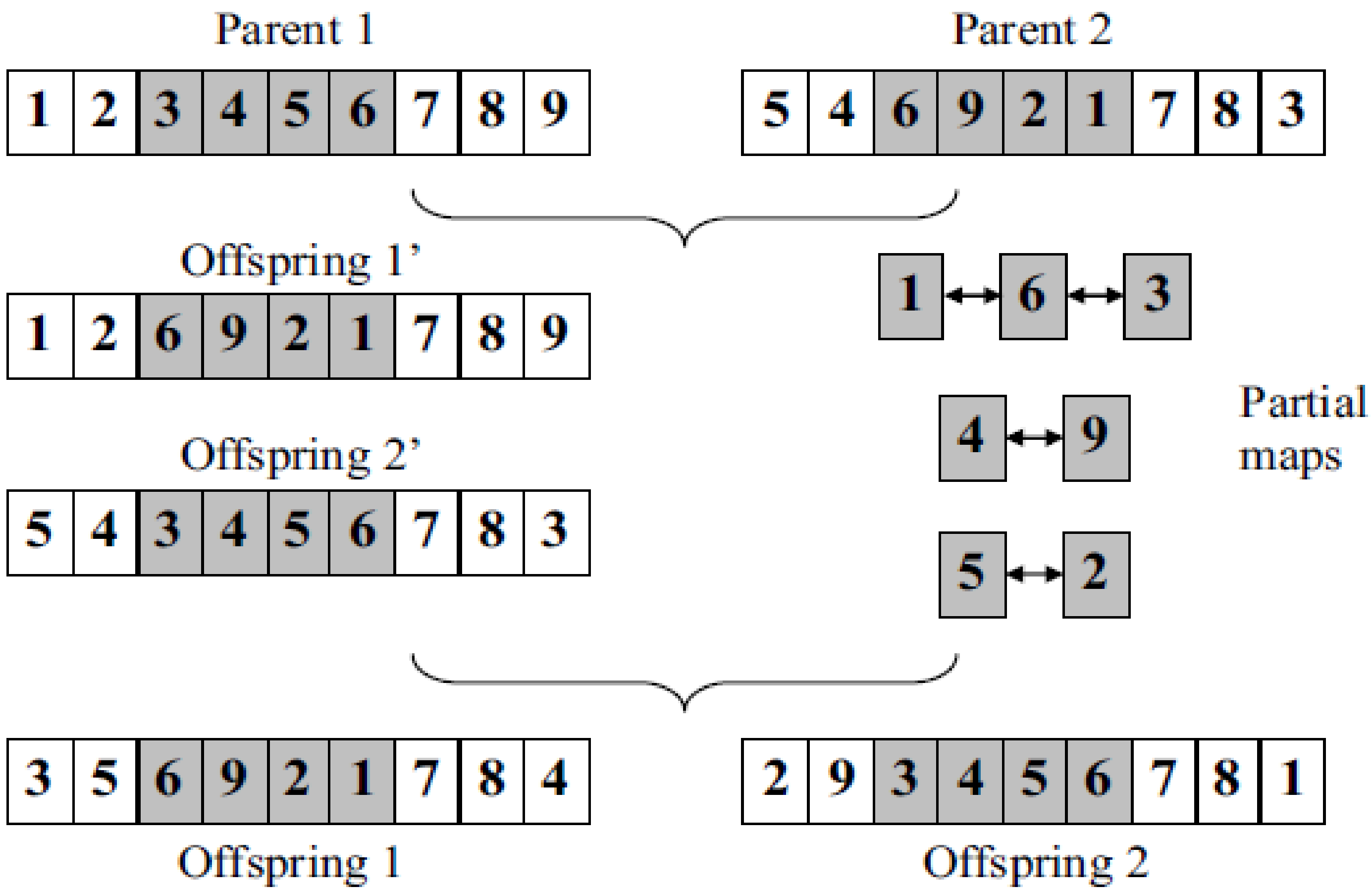
- Applying classical crossover operators to permutations will generate solutions that are not permutations (i.e., infeasible solutions).
- Hence, many permutation crossover operators have been designed as follows:
- **Order Crossover (OX)**



The order crossover for permutations.

Crossover for Permutations

- **Partially Mapped Crossover (PMX)**



An example of partially mapped crossover for permutation code

Mutation

- Mutations represent small changes of selected individuals of the population.
- The probability p_m defines the probability to mutate each element (gene) of the representation.
- In general, small values are recommended for this probability (e.g., $p_m \in [0.001, 0.01]$).
- Some strategies initialize mutation probability to $1/k$ where k is the number of decision variable

Selection Methods

- The better an individual is, the higher its chance of being parent.
- Worst individuals still have some chance to be selected.
- **Roulette (轮盘赌) Wheel Selection**
  - It will assign to each individual a selection probability that is proportional to its relative fitness.
  - Let f_i be the fitness of the individual *i* in the population *P*. Its probability to be selected is:

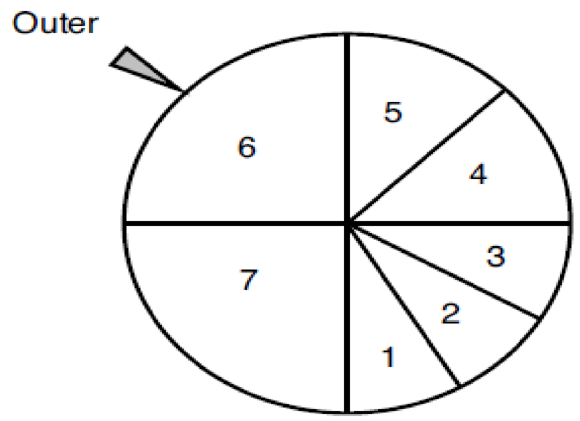
$$p_i = f_i \Big/ \left(\sum_{j=1}^{n} f_j \right)$$

  - A pie graph can be constructed where each individual is assigned a space on the graph that is proportional to its fitness.

Selection Methods

- An outer roulette wheel is placed around the pie.
- The selection of μ individuals is performed by μ independent spins (旋转) of the roulette wheel.
- Better individuals have more space and then more chance to be chosen.

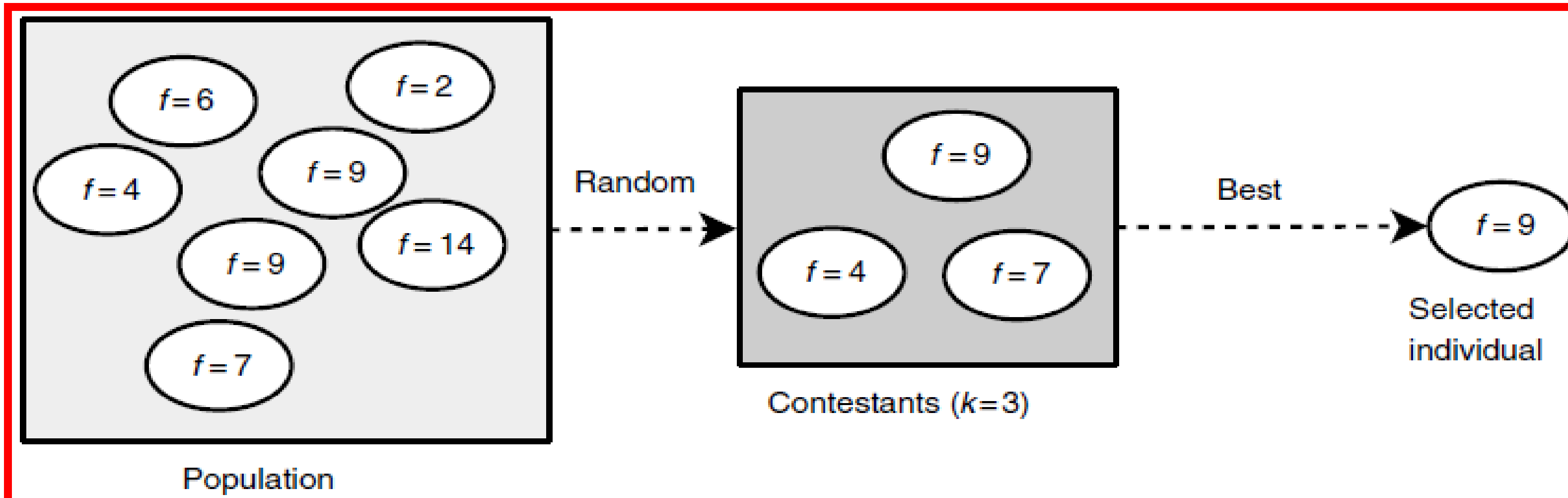
Individuals:	1	2	3	4	5	6	7
Fitness:	1	1	1	1.5	1.5	3	3



Roulette selection

Selection Methods

- **Tournament (锦标赛) Selection**
  - Randomly select *k* individuals; the parameter *k* is called the size of the tournament group
  - Then, select the best one from the selected *k* individuals.



Tournament selection strategy. For instance, a tournament of size 3 is performed. Three solutions are picked randomly from the population. The best solution from the picked individuals is then selected.

Summary

- Enhancing the Quality of the End Result
  - How to balance the Intensification and Diversification
    - Too much intensification -> local search
    - Too much diversification -> random search
  - Initialization Method
  - Hybrid Method
  - Operator Enhancement
- Reducing the Running Time
  - Parallel Computing
  - Employing Advanced Data Structures
  - Redesigning the procedure of Metaheuristics

Thank you!