



Kasdi Merbah University – Ouargla
Department of Computer Science
Advanced Operations Research
Final Examination



| | | |
|---------------|---|--------------------|
| Lecturer name | : | Dr. Farouq Zitouni |
| Duration | : | Two (02) hours |
| Student name | : | Correction |

Problem 1 (Job Sequencing) (10 marks)

Consider the case of sequencing n jobs on a single machine. The processing time for job j is t_j and its due date is d_j . Completing job j ahead of its due date incurs a holding (storage) cost h_j per day. A tardy job j results in a penalty cost p_j per day. The following table provides the data for a 5-job problem.

Table 1 - Data for a single-machine 5-job sequencing problem.

| Job, j | Processing time in days, t_j | Due date, d_j | Holding cost, h_j | Penalty cost, p_j |
|----------|--------------------------------|-----------------|---------------------|---------------------|
| 1 | 10 | 15 | 3 | 10 |
| 2 | 8 | 20 | 2 | 22 |
| 3 | 6 | 10 | 5 | 10 |
| 4 | 7 | 30 | 4 | 8 |
| 5 | 4 | 12 | 6 | 15 |

We utilize the algorithm described below to solve the 5-job sequencing problem.

Step 0

- Encode chromosomes using a numeric representation¹.
- Generate a random population X of N feasible chromosomes ($N = 4$).
- For each chromosome P_i in the considered population, evaluate its fitness value (see the description below) - record P^* as the best available solution.

Step 1

- Select two parent chromosomes from population X .
- Crossover the parents' genes to create two children (see the description below).
- Mutate the children's genes randomly (see the description below).
- If the resulting solutions are infeasible, repeat **Step 1** until feasibility is achieved; else, replace the weakest two parents with the new children to form a new population X and update P^* . Go to **Step 2**.

¹ A chromosome is a sequence of 5 numbers selected from the set $\{1,2,3,4,5\}$.

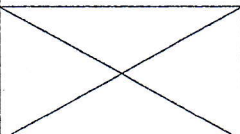
Step 2 If a termination condition has been reached, stop; return P^* as the best available solution; else, repeat **Step 1**.

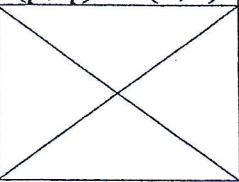
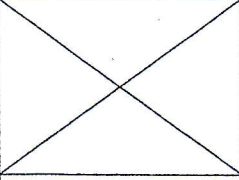
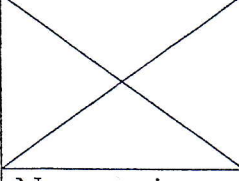
- To show how children are created, consider parent chromosomes $P_1 = 1 - 3 - 5 - 2 - 4$ and $P_2 = 5 - 4 - 2 - 3 - 1$. Suppose that a random 1-point crossover occurs at gene 3. The first two genes of children C_1 and C_2 are constructed by swapping the first two genes of P_1 and P_2 . The last three genes are the ones remaining from P_1 and P_2 after excluding the first two genes, e.g.,
 - First two genes of $C_1 = 5 - 4$.
 - First two genes of $C_2 = 1 - 3$.
 - Last three genes of $C_1 = \{1, 3, 5, 2, 4\} - \{5, 4\} = 1 - 3 - 2$.
 - Last three genes of $C_2 = \{5, 4, 2, 3, 1\} - \{1, 3\} = 5 - 4 - 2$.
 Thus, $C_1 = 5 - 4 - 1 - 3 - 2$ and $C_2 = 1 - 3 - 5 - 4 - 2$.
- Next, mutations of C_1 and C_2 are carried out in the following manner: If a random number $R < 0.1$, a child chromosome is subject to mutation. A mutation is then implemented for the child by swapping two randomly selected genes. For example, the random numbers $R = 0.8452$ and $R = 0.0342$ applied to C_1 and C_2 , respectively, indicate that only C_2 is mutated. Randomly choosing $p = 1$ and $q = 5$; thus, C_2 is mutated from $1 - 3 - 5 - 4 - 2$ to $2 - 3 - 5 - 4 - 1$.
- Finally, how the fitness value of a chromosome is estimated could be summarized below.

| Job | 1 | 2 | 3 | 4 | 5 |
|-------------------|----|----|-----|----|-----|
| Processing time | 10 | 8 | 6 | 7 | 4 |
| Due date | 15 | 20 | 10 | 30 | 12 |
| Completion time | 10 | 18 | 24 | 31 | 35 |
| Holding time | 5 | 2 | 0 | 0 | 0 |
| Delay time | 0 | 0 | 14 | 1 | 23 |
| Holding cost | 15 | 4 | 0 | 0 | 0 |
| Late penalty cost | 0 | 0 | 140 | 8 | 345 |

$$Z = (15 + 4 + 0 + 0 + 0) + (0 + 0 + 140 + 8 + 345) = 512.$$

Question: Iterate the previous algorithm three times, and then give the value of P^* with its fitness value.

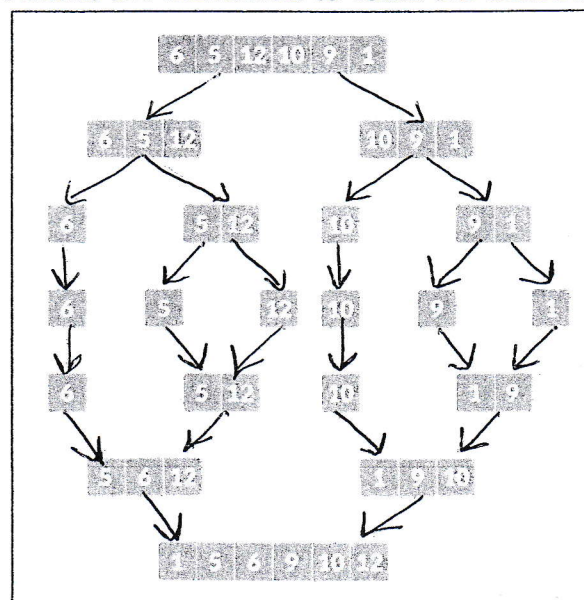
| Iteration | Population | $Z(P_i)$ | P^* | Crossover | Mutation |
|-----------|---------------------------|----------|-------|-----------------|---|
| 0 | $P_1 = 1 - 2 - 3 - 4 - 5$ | 512 | | P_4 and P_3 |  |
| | $P_2 = 2 - 3 - 4 - 1 - 5$ | 605 | | | |
| | $P_3 = 4 - 1 - 5 - 2 - 3$ | 695 | | | |

| | | | | | |
|---|-------------------|-----|-----------------|--|---|
| 1 | $P_4 = 3-2-1-4-5$ | 475 | P_3 and P_1 | | |
| | $C_1 = 3-5-4-1-2$ | 534 | | | $(p,q) = (2,5)$ |
| | $C_2 = 5-1-3-2-4$ | 367 | | | $(p,q) = (1,5)$ |
| | $P_1 = 1-2-3-4-5$ | 512 | | P_1 and P_4 |  |
| | $P_2 = 3-5-4-1-2$ | 534 | | | |
| | $P_3 = 5-1-3-2-4$ | 367 | | | |
| | $P_4 = 3-2-1-4-5$ | 475 | | | |
| | $C_1 = 5-3-1-2-4$ | 314 | | | $(p,q) = (2,3)$ |
| 2 | $C_2 = 1-5-3-2-4$ | 361 | P_4 and P_2 | | $(p,q) = (2,4)$ |
| | $P_1 = 5-3-1-2-4$ | 314 | | P_1 and P_4 |  |
| | $P_2 = 1-5-3-2-4$ | 361 | | | |
| | $P_3 = 5-1-3-2-4$ | 367 | | | |
| | $P_4 = 3-2-1-4-5$ | 475 | | | |
| | $C_1 = 2-3-5-1-4$ | 324 | | | $(p,q) = (1,2)$ |
| | $C_2 = 5-3-2-1-4$ | 222 | | | No mutation |
| 3 | $P_1 = 5-3-1-2-4$ | 314 | P_4 and P_2 |  | |
| | $P_2 = 1-5-3-2-4$ | 361 | | | |
| | $P_3 = 2-3-5-1-4$ | 324 | | | |
| | $P_4 = 5-3-2-1-4$ | 222 | | | |
| | $C_1 = 5-3-1-2-4$ | 314 | | | No mutation |
| | $C_2 = 1-5-3-2-4$ | 361 | | | No mutation |

$$P^* = 5-3-2-1-4 \quad Z(P^*) = 222$$

Problem 2 (Time complexity) (5 marks)

The merge sort is one of the most popular sorting algorithms based on the principle of divide and conquer. Here, a problem is divided into multiple sub-problems. Then, each one is solved individually. Finally, sub-problems are combined to form the final solution.



Using the divide and conquer technique, we divide a problem into subproblems. Then, when the solution to each subproblem is ready, we combine the results from the subproblems to solve the main problem. For example, suppose we had to sort an array A . Then, a subproblem would be to sort a sub-section of this array starting at index p and ending at index r , denoted as $A[p \dots r]$.

1. **Divide:** If q is the halfway point between p and r , then we can split the subarray $A[p \dots r]$ into two arrays $A[p \dots q]$ and $A[q + 1 \dots r]$.
2. **Conquer:** In the conquer step, we try to sort the subarrays $A[p \dots q]$ and $A[q + 1 \dots r]$. If we haven't yet reached the base case, we again divide these subarrays and try to sort them.
3. **Combine:** When the conquer step reaches the base stage, and we get two sorted subarrays $A[p \dots q]$ and $A[q + 1 \dots r]$ for array $A[p \dots r]$, we combine the results by creating a sorted array $A[p \dots r]$ from two sorted subarrays $A[p \dots q]$ and $A[q + 1 \dots r]$.

Question: compute the time complexity of the merge sort algorithm.

$$O(n \log n)$$

Problem 3 (Theoretical concepts on metaheuristics) (5 marks)

It is evident from the broader spectrum of successful applications that metaheuristic algorithms are potential solutions to challenging optimization problems. Among such algorithms are swarm-based methods like particle swarm optimization and ant colony optimization, which increasingly attract new researchers. However, despite their popularity, the core questions on performance issues are still partially answered due to limited insightful analyses. Exploration and exploitation are two fighting ingredients that all metaheuristics share. Due to the stochastic nature of the swarming process of individuals, it is often hard to mathematically quantify these components. Explain the relationship between the exploration and exploitation phases and their prominent roles according to your vision.

- 1) The relationship between exploration and exploitation is conflicting.
- 2) The role of exploration is to discover promising areas in the search space and avoid getting trapped in local optimums.
- 3) The role of exploitation is to focus the search process on promising areas so far found.