Term Project – Phase 3 Announcement

Dear Students,

You have completed Phase 1 (Problem Definition) and Phase 2 (Example Scenario / Problem Instance). You will now begin **Phase 3 of the Term Project**, where you will design, implement, and test a metaheuristic algorithm adapted to the optimization problem you selected earlier. This stage represents the core algorithmic work of your project and directly contributes to your final report and presentation.

1. Selection of Metaheuristic Algorithm

Each student must choose **one** of the following metaheuristic algorithms:

- Simulated Annealing
- Tabu Search
- Advanced Hill Climbing (with intelligent improvements)
- Iterated Local Search
- Genetic Algorithm
- Particle Swarm Algorithm
- Ant Colony / Bee Colony Algorithm

Important Rules:

- If two students selected the same optimization problem, they may not choose the same metaheuristic unless their designs are significantly different.
- Students choosing Genetic Algorithm must use different crossover/mutation operators if more than one student picks GA.

2. Data Structure and Fitness Function Implementation

You must design and implement:

- A representation of an individual solution (chromosome, state, permutation, vector, etc.)
- A fitness/objective function for evaluating individuals

Examples:

- Graph Coloring: array of color assignments or array of vertices

- Knapsack: binary vector of selected items
- TSP: permutation of cities
- Bin Packing: list of bins and assigned items
- Job Shop Scheduling: operation sequences

3. Implementation of the Metaheuristic Algorithm (Final Report Section 5)

You may use any programming language (Python, C/C++, Java, MATLAB, etc.). Your implementation must include:

- Initialization method
- Neighborhood structure (if your algorithm is single solution-based) or operators (if your algorithm is population-based) such as crossover, mutation, perturbation, tabu rules, etc.
- Selection/acceptance criteria
- Update/evaluation loop
- Stopping criteria (iterations, time, stagnation)

4. Testing on the Example Scenario

You must run your algorithm on the small-scale problem instance defined in Phase 2. Your algorithm should produce the **expected correct result** for this small instance to confirm:

- Fitness function works
- Representation is correct
- Algorithm logic is correct

5. Experimental Setup (Final Report Section 6.1)

You must report:

- **System Information (Final Report Section 6.1.1):** CPU, RAM, operating system, programming language
- **Algorithm Parameters (Final Report Section 6.1.2):** iteration count, mutation rate (if applicable), temperature schedule, tabu tenure, number of particles/ants, time limit, population size etc.

- **Number of Runs:** Each instance must be run **5 times**

6. Experimental Results (Final Report Section 6.2)

Prepare tables summarizing the performance of your algorithm (Check the lecture slides about the experimental study for JSSP):

- Mean objective value
- Best objective value
- Standard deviation
- Best computation time
- Average computation time

For example:

Instance	Mean Obj.	Best Obj.	Std. Dev.	Mean Time (s)	Best Time (s)
Example	15.2	12	1.48	0.031	0.012

7. Final Report Sections 5–6–7

Fill in the following sections of the provided project template:

- Section 5 – Experimental Setup
- Section 6 – Experimental Results
- Section 7 – Discussion and Conclusion

Reports not compatible with the template will receive 0 points. Submit your report as a **PDF** file only.

8. Final Presentation

A presentation template will be provided shortly. Your slides must include:

- Outline
- Problem Definition
- General design of your algorithm
- Problem instance and individual representation
- Example run of your algorithm

- Experimental setup
- Experimental results

Presentation time: **8 minutes + 2 minutes Q&A**.

Grading

TP1	20%
TP2	10%
Final Report	40%
Presentation	30%

Submission Details

- Submit on YULearn as a zip file which includes (TP3_Name_Final.zip)
- Final Report (a single PDF): `TP3_Name_Problem.pdf`
- Source code + readme/manual

For any questions, you may contact: `gterci@cse.yeditepe.edu.tr`.