To address Question 1, which is about identifying the combination of parameters that produces the best results, you need to analyze the performance of your evolutionary algorithm under different parameter settings. Here's an approach to implement and analyze this:

1. **Define Parameter Combinations:**
   * Create a set of parameter combinations to explore. For example, you might vary population size, tournament size, crossover operators, and mutation operators.
2. **Experiment Loop:**
   * Implement a loop to iterate through all the parameter combinations.
   * For each combination, run your evolutionary algorithm.
3. **Record Results:**
   * Record the best solution found for each parameter combination.
   * Store other relevant metrics, such as fitness values, convergence curves, and execution time.
4. **Identify the Best Combination:**
   * After all experiments, analyze the results to identify the combination that produced the best solution (minimized fitness).
5. **Present Results:**
   * Output or display a summary of the results, highlighting the best combination.

Now, let's modify the evolutionary algorithm code to include this experiment loop. Assuming you have a function **evolutionary\_algorithm** that takes parameters and returns results, here's a simplified example:

This is a basic structure, and you might need to adapt it based on your actual code structure and the parameters you want to explore. Additionally, you can extend this by saving and analyzing convergence curves and execution times for a more comprehensive evaluation.