

The multi-objective optimization for FJSP in urban flood emergency management is formulated as follows:  
Urban emergency response involves the coordinated efforts of multiple departments (e.g., emergency response teams, medical units, logistics), each with multiple response teams dispersed across different city regions. When a flood event occurs, multiple affected locations may require simultaneous intervention, each hosting different target groups (e.g., injured individuals, trapped residents, evacuees) that necessitate specialized departmental collaboration. Multiple response teams from different locations and departments have varying travel times to affected sites. Therefore, it is crucial to ensure that appropriate teams arrive at the appropriate locations as simultaneously as possible, and the latest arrival time is the shortest. This is to maximize coordination efficiency while minimizing overall response delays (e.g., rescue teams must first free trapped individuals before medical teams can provide treatment).  
This problem is abstracted as a multi-objective optimization challenge, balancing minimum Makespan (the longest arrival time among collaborative teams at all locations) and Synchronization Cost (The minimum sum of arrival time differences within teams at each location).  
The Makespan C\_max and Synchronization Cost SC\_max are defined as:  
 $C_{max} = \max(T_{i,j} \mid i \in R, j \in L)$   
 $SC_{max} = \sum_j \left[ \max T_{i,j} - \min T_{i,j} \right] \quad \{i \in R, j \in L\}$   
Where  $T_{i,j}$  is travel time for response team  $i$  to reach the affected location  $j$ .  $R$  is set of all response teams.  $L$  is set of all affected locations.

In this context, here is an example of a FJSP problem and the corresponding solution based on a multi-objective optimization genetic algorithm, and the same response teams will not be assigned to multiple locations at the same time.  
Please wait for me to make new requirements and optimize the pre-solution according to this new requirement and give me a solution that considers all locations globally.

Problem:  
Problem\_ID Num\_Locations Num\_Departments  
Num\_Target\_Groups Department Team\_ID Location\_Times  
Target\_Groups  
1 4 7 2 Dept\_1 1 [28, 26, 8, 2] Group\_1, Group\_2  
1 4 7 2 Dept\_1 2 [25, 14, 11, 8] Group\_1, Group\_2  
1 4 7 2 Dept\_1 3 [3, 29, 26, 28] Group\_1, Group\_2  
1 4 7 2 Dept\_1 4 [29, 10, 20, 11] Group\_1, Group\_2  
1 4 7 2 Dept\_1 5 [23, 5, 24, 14] Group\_1, Group\_2  
1 4 7 2 Dept\_2 1 [20, 2, 16, 9] Group\_1, Group\_2  
1 4 7 2 Dept\_2 2 [4, 19, 6, 10] Group\_1, Group\_2  
1 4 7 2 Dept\_2 3 [17, 19, 2, 14] Group\_1, Group\_2  
1 4 7 2 Dept\_2 4 [12, 16, 22, 17] Group\_1, Group\_2  
1 4 7 2 Dept\_2 5 [23, 18, 7, 4] Group\_1, Group\_2  
1 4 7 2 Dept\_3 1 [21, 28, 13, 14] Group\_1, Group\_2  
1 4 7 2 Dept\_3 2 [7, 20, 30, 9] Group\_1, Group\_2  
1 4 7 2 Dept\_3 3 [18, 18, 19, 10] Group\_1, Group\_2  
1 4 7 2 Dept\_3 4 [20, 27, 11, 23] Group\_1, Group\_2  
1 4 7 2 Dept\_3 5 [20, 12, 4, 30] Group\_1, Group\_2  
1 4 7 2 Dept\_4 1 [24, 25, 19, 8] Group\_1, Group\_2  
1 4 7 2 Dept\_4 2 [30, 11, 21, 3] Group\_1, Group\_2  
1 4 7 2 Dept\_4 3 [27, 28, 12, 12] Group\_1, Group\_2  
1 4 7 2 Dept\_4 4 [26, 20, 27, 28] Group\_1, Group\_2  
1 4 7 2 Dept\_4 5 [26, 19, 29, 20] Group\_1, Group\_2  
1 4 7 2 Dept\_5 1 [14, 18, 21, 3] Group\_1, Group\_2  
1 4 7 2 Dept\_5 2 [10, 2, 7, 13] Group\_1, Group\_2  
1 4 7 2 Dept\_5 3 [11, 14, 13, 16] Group\_1, Group\_2  
1 4 7 2 Dept\_5 4 [14, 25, 11, 20] Group\_1, Group\_2  
1 4 7 2 Dept\_5 5 [11, 20, 24, 2] Group\_1, Group\_2  
1 4 7 2 Dept\_6 1 [29, 28, 6, 18] Group\_1, Group\_2  
1 4 7 2 Dept\_6 2 [9, 24, 11, 20] Group\_1, Group\_2  
1 4 7 2 Dept\_6 3 [20, 5, 22, 20] Group\_1, Group\_2  
1 4 7 2 Dept\_6 4 [19, 29, 26, 22] Group\_1, Group\_2  
1 4 7 2 Dept\_6 5 [27, 15, 26, 27] Group\_1, Group\_2  
1 4 7 2 Dept\_7 1 [10, 21, 1, 25] Group\_1, Group\_2  
1 4 7 2 Dept\_7 2 [13, 29, 23, 18] Group\_1, Group\_2  
1 4 7 2 Dept\_7 3 [7, 25, 3, 22] Group\_1, Group\_2  
1 4 7 2 Dept\_7 4 [22, 18, 11, 10] Group\_1, Group\_2  
1 4 7 2 Dept\_7 5 [15, 12, 9, 22] Group\_1, Group\_2

Solution:  
Problem\_ID Location Department Assigned\_Team Multi  
Travel\_Time Multi Makespan Multi SyncCost Multi  
1 Location 1 Dept\_1 Team\_5 23 24 40  
1 Location 1 Dept\_2 Team\_5 23 24 40  
1 Location 1 Dept\_3 Team\_5 20 24 40  
1 Location 1 Dept\_4 Team\_1 24 24 40  
1 Location 1 Dept\_5 Team\_1 14 24 40  
1 Location 1 Dept\_6 Team\_4 19 24 40  
1 Location 1 Dept\_7 Team\_5 15 24 40  
1 Location 2 Dept\_1 Team\_2 14 24 40  
1 Location 2 Dept\_2 Team\_2 19 24 40  
1 Location 2 Dept\_3 Team\_2 20 24 40  
1 Location 2 Dept\_4 Team\_4 20 24 40  
1 Location 2 Dept\_5 Team\_5 20 24 40  
1 Location 2 Dept\_6 Team\_5 15 24 40  
1 Location 2 Dept\_7 Team\_1 21 24 40  
1 Location 3 Dept\_1 Team\_1 8 24 40  
1 Location 3 Dept\_2 Team\_4 22 24 40  
1 Location 3 Dept\_3 Team\_3 19 24 40  
1 Location 3 Dept\_4 Team\_2 21 24 40  
1 Location 3 Dept\_5 Team\_3 13 24 40  
1 Location 3 Dept\_6 Team\_3 22 24 40  
1 Location 3 Dept\_7 Team\_4 11 24 40  
1 Location 4 Dept\_1 Team\_4 11 24 40  
1 Location 4 Dept\_2 Team\_3 14 24 40  
1 Location 4 Dept\_3 Team\_1 14 24 40  
1 Location 4 Dept\_4 Team\_5 20 24 40  
1 Location 4 Dept\_5 Team\_4 20 24 40  
1 Location 4 Dept\_6 Team\_1 18 24 40  
1 Location 4 Dept\_7 Team\_2 18 24 40

Got it! You want to optimize the given Flood Job Shop Scheduling Problem (FJSP) by taking a global approach to assigning teams to locations, rather than handling each location individually. This way, you ensure a more balanced coordination across the entire problem.

Key Objectives:

- 1. Minimize Makespan (Longest arrival time across all collaborative teams at all locations).
- 2. Minimize Synchronization Cost (Sum of arrival time differences within teams at each location).
- 3. Ensure that the same response team is not assigned to multiple locations simultaneously.

Next Steps:

- 1. Formulate Global Optimization Strategy:
  - Use a multi-objective genetic algorithm (e.g., NSGA-II or SPEA2).
  - Incorporate a constraint check to prevent assigning the same team to multiple locations.
  - Define fitness functions for Makespan and Synchronization Cost.
- 2. Input Data Handling:
  - Parse the data to create a structured format (e.g., DataFrame).
  - Analyze travel times and group assignments.
- 3. Solution Generation:
  - Generate initial population based on random team assignments.
  - Use crossover and mutation strategies that maintain feasibility (no repeated team assignments).
- 4. Evaluation and Selection:
  - Evaluate Makespan and Synchronization Cost for each solution.
  - Use Pareto optimization to balance the two objectives.
- 5. Output:
  - Present optimal or near-optimal solutions.
  - Compare with the pre-solution given to assess improvements.

I'll start by loading the given problem data and processing it. Could you confirm whether you would like me to run a full optimization or just an improved solution based on the existing assignment?

Due to unexpected road conditions, Team 1 of Department 1's arrival time at Location 3 has increased to 30 minutes. reassigning teams to minimize Makespan or Synchronization Cost, please.

I will update the travel time for Team 1 of Department 1 at Location 3 to 30 minutes and then perform the optimization to minimize either Makespan or Synchronization Cost.

Steps:

- 1. Update the travel time in the given data.
- 2. Reassign teams to minimize Makespan or Synchronization Cost.
- 3. Present the optimized assignment.

Let me process the data and perform the optimization.

I've updated the travel time for Team 1 of Department 1 at Location 3 to 30 minutes and reassigned teams to minimize both Makespan and Synchronization Cost. You can FJSP the optimized team assignment in the displayed table.

Let me know if you need further adjustments or analysis! [↩]