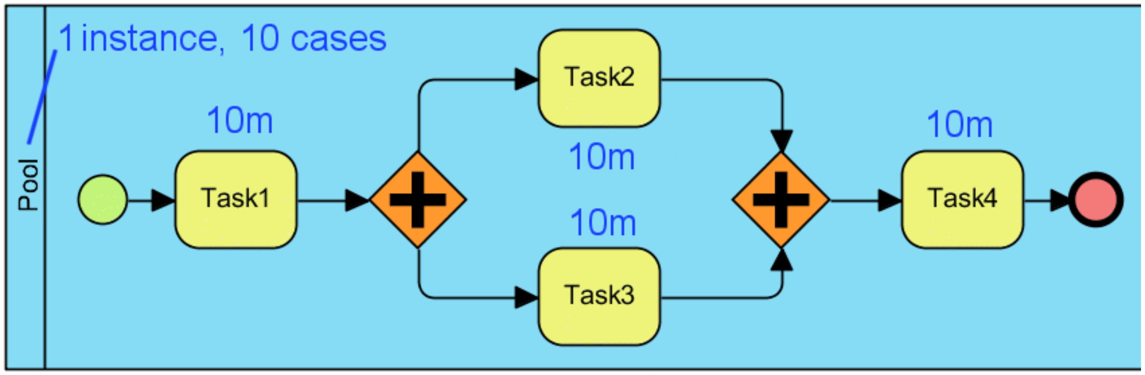
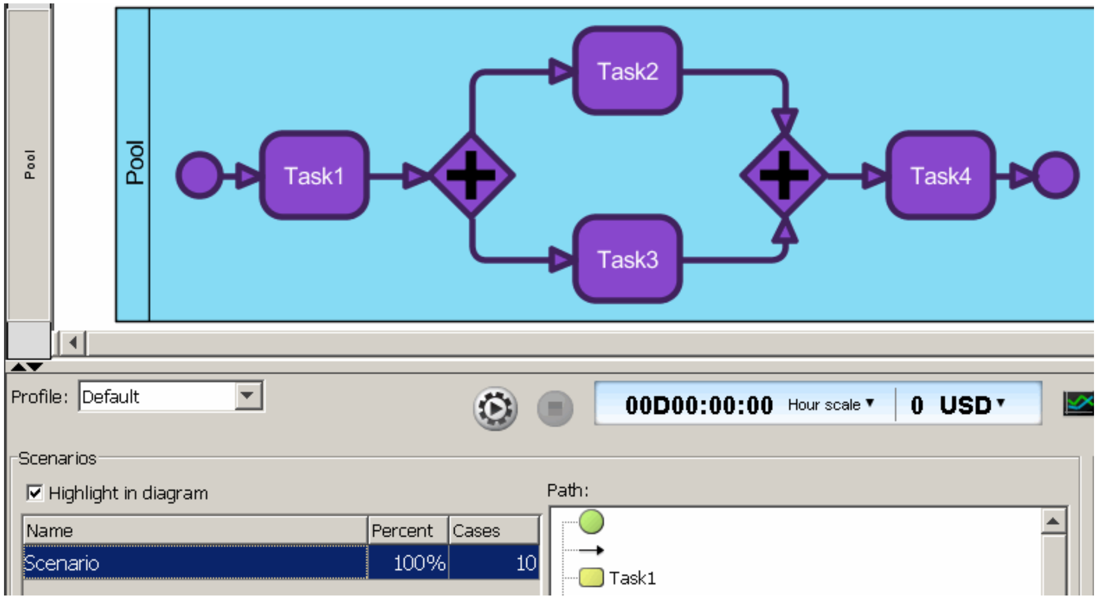
Simulation and Emulation

- Simulate the following simple diagram, with a unique scenario of 10 instances, so as to understand how the simulator works with parallel flows.



- Create a unique scenario by using parallel gateways only (fig.)



Profile: Default

00D00:00:00 Hour scale ▾ 0 USD ▾

Scenarios

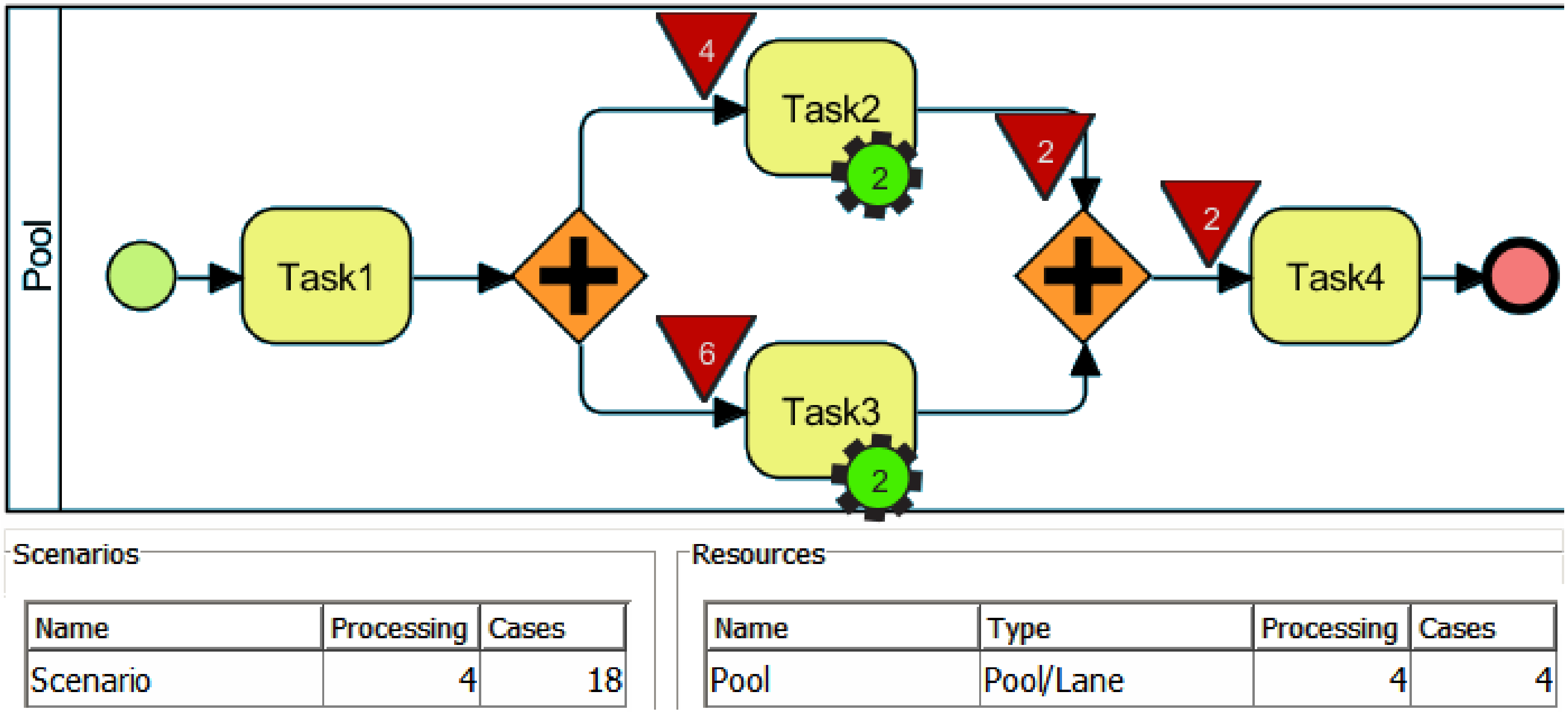
☑ Highlight in diagram

Name	Percent	Cases
Scenario	100%	10

Path:

Task1

- As expected, the parallel gateway duplicates the number of tokens. Indeed, the maximum number of tokens produced during the simulation is 20.
- To allow an actual parallel execution of Task2 and Task3, the number of instances of the Pool has to be increased, e.g. to 4. The total duration is 1h 40m.



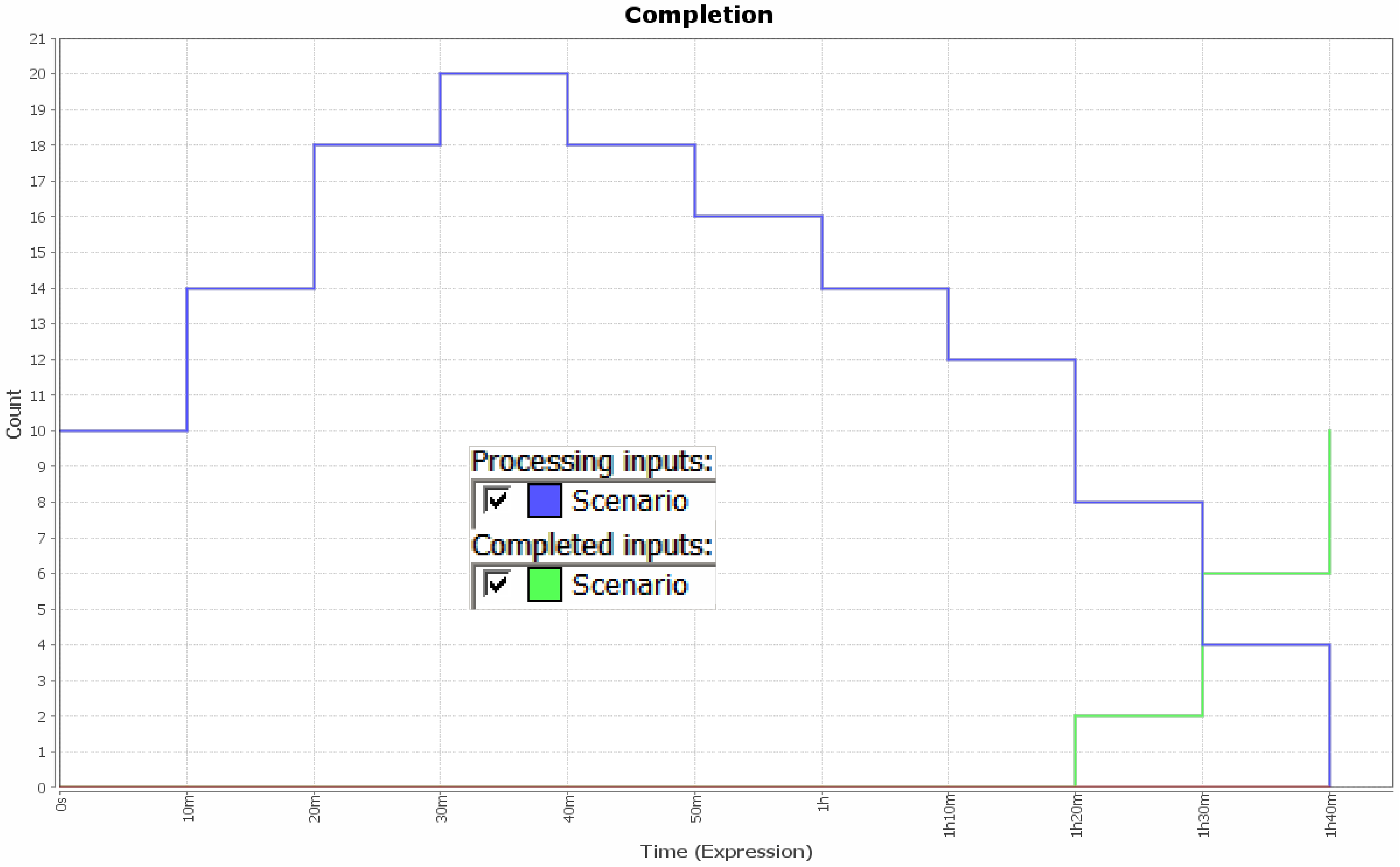
Scenarios

Name	Processing	Cases
Scenario	4	18

Resources

Name	Type	Processing	Cases
Pool	Pool/Lane	4	4

- Note that the simulator engine may perform an **unbalanced allocation** of Pool resources to tasks **when the number of tokens is low** (a few units). For example, in some of such trials we may discover that all Pools have been moved from Task2 to Task3 (and vice versa), leading to an increase of the completion time.
- To avoid these occurrences, perform the same trial varying the number of resources.



- For more complex layout, simulation of parallel paths in the same scenario is not allowed by the simulator. This means that, when creating a scenario, you are asked to choose a path even using the parallel gateway.

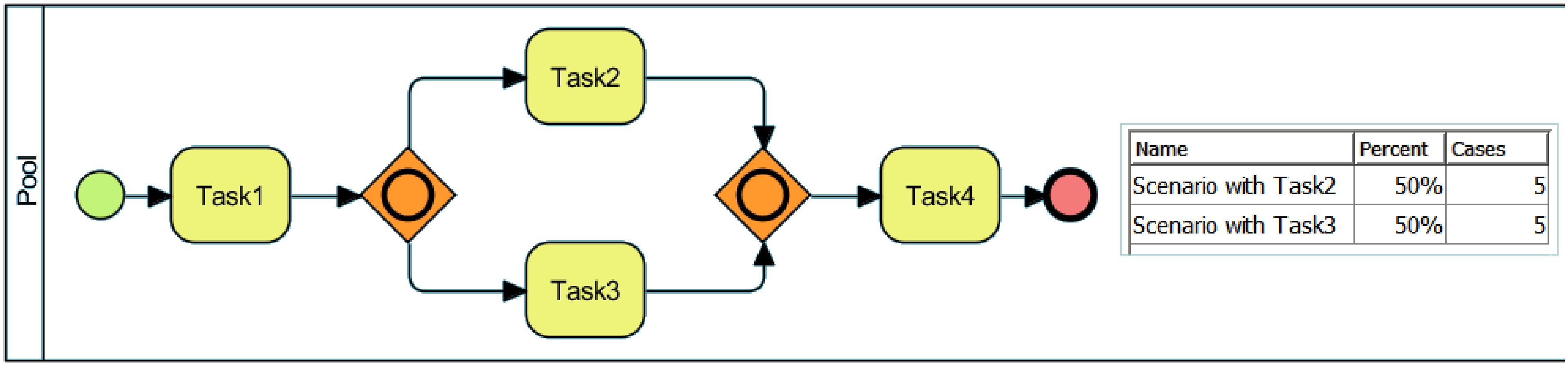
- For this reason, you have to create an **emulation diagram**, i.e., a diagram which is purposely created by considering the limitation of a simulation engine, so as to follow the actual model.

- **Simulation** involves modeling the underlying state of the system. Ideally, you should be able to look into the simulation and observe properties that you would also see if you looked into the original system.

- In practice, there may some shortcuts to the simulation for performance reasons, that is, some internal aspects of the simulation may actually be emulated.

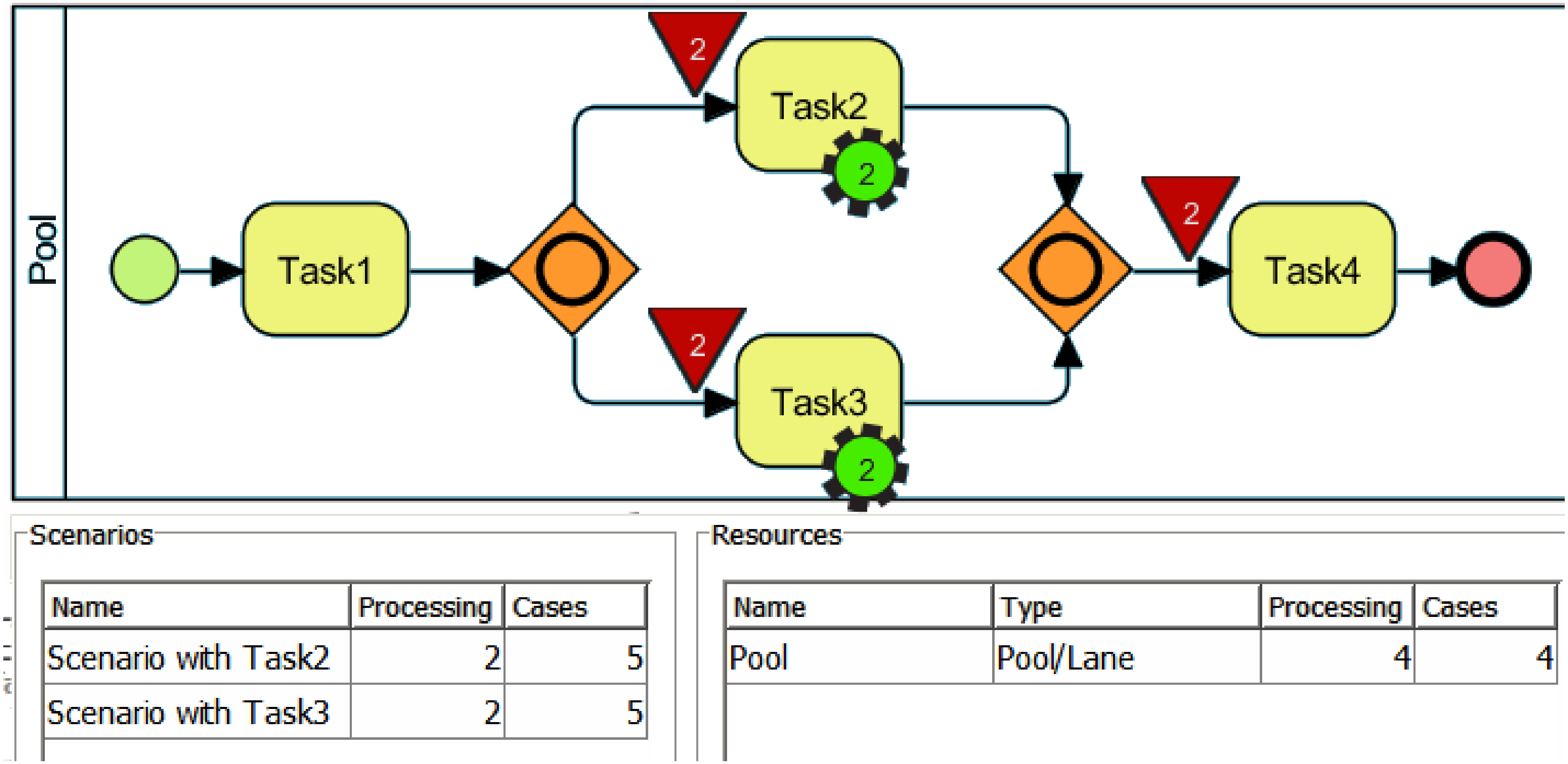
- **Emulation** is the process of mimicking the outwardly observable behavior to match an existing system. The internal state of the emulation mechanism does not have to accurately reflect the internal state of the system which it is emulating.

- Emulate the above example by using inclusive/exclusive gateways and by combining two scenarios.



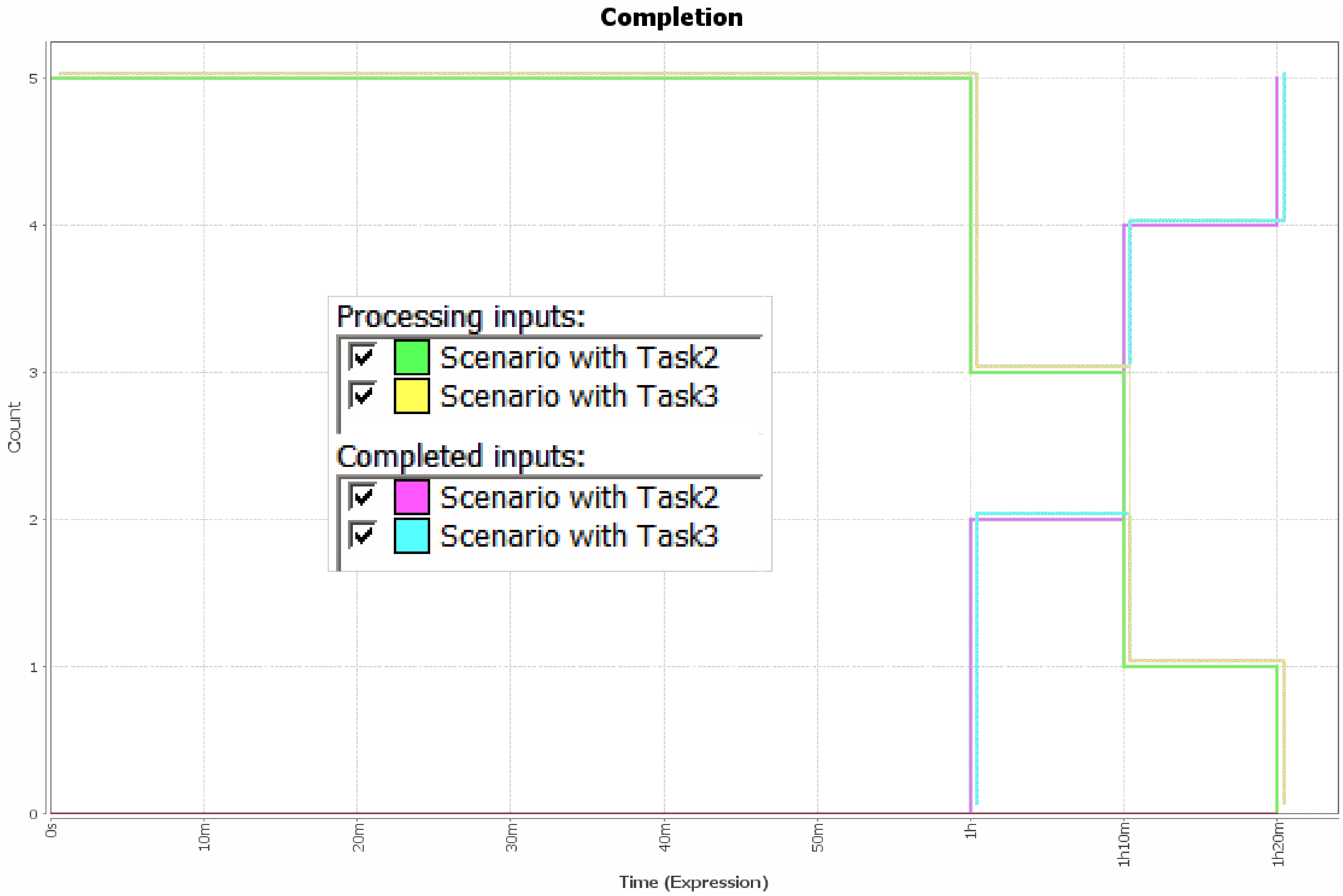
Name	Percent	Cases
Scenario with Task2	50%	5
Scenario with Task3	50%	5

- With the emulation, the total duration is shorter, i.e., 1h 20m, because there are no synchronization constraints between Task2 and Task3 at the merging node.

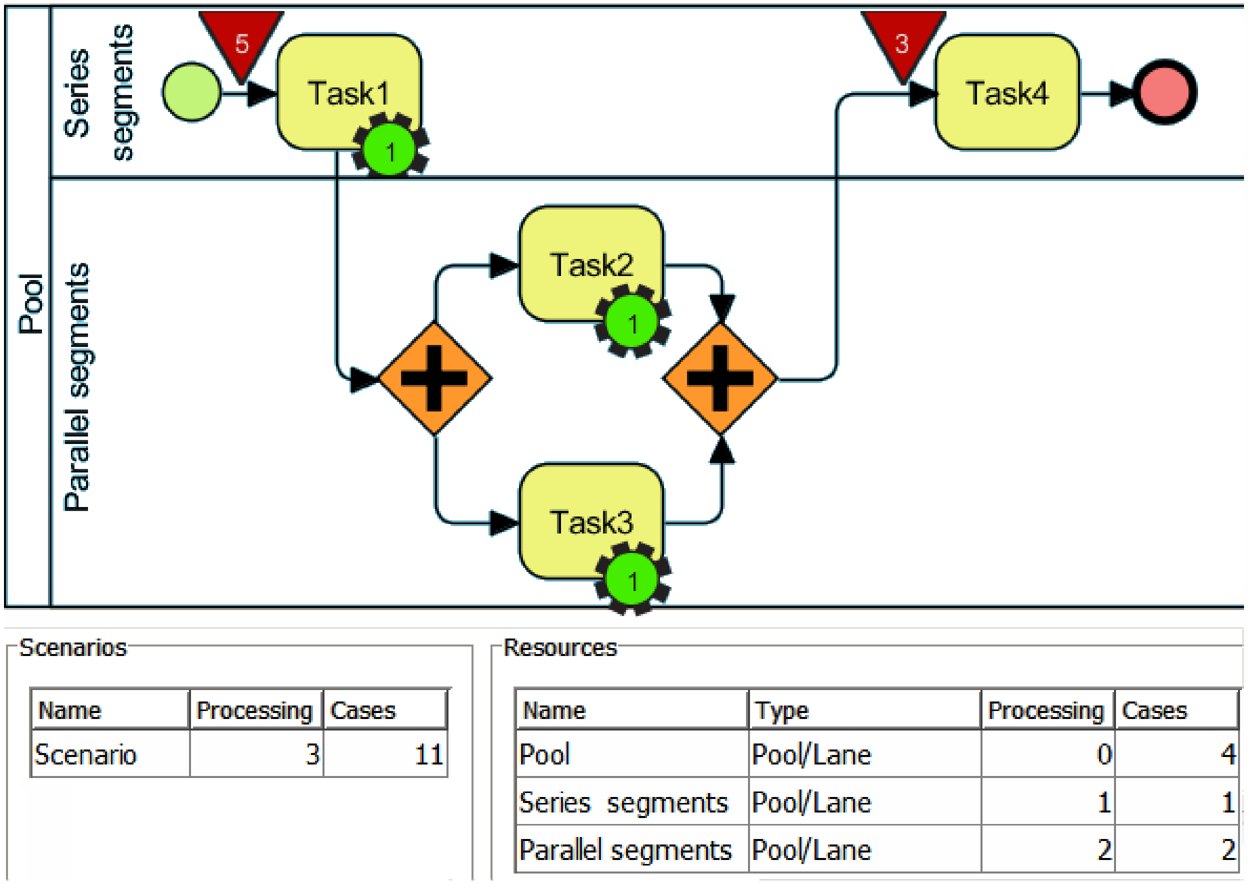


Name	Processing	Cases
Scenario with Task2	2	5
Scenario with Task3	2	5

Name	Type	Processing	Cases
Pool	Pool/Lane	4	4


Completion
Count
5
4
3
2
1
0
0s
10m
20m
30m
40m
50m
1h
1h10m
1h20m
Time (Expression)
Processing inputs:
☑ 🟩 Scenario with Task2
☑ 🟨 Scenario with Task3
Completed inputs:
☑ 🟪 Scenario with Task2
☑ 🟦 Scenario with Task3

- When assigning multiple instances of a resource (lane), the execution of single tasks (e.g., Task1 and Task4) is also parallelized, because the simulator allocates the available resource to all tasks.
- In order to decide which tasks have to be parallelized, we can define two levels of parallelism with a **mirror resource**, equipped with more instances, as the resource for parallel segments. A separate **base resource**, with a single instance, devoted to single task execution.



Scenarios

Name	Processing	Cases
Scenario	3	11

Resources

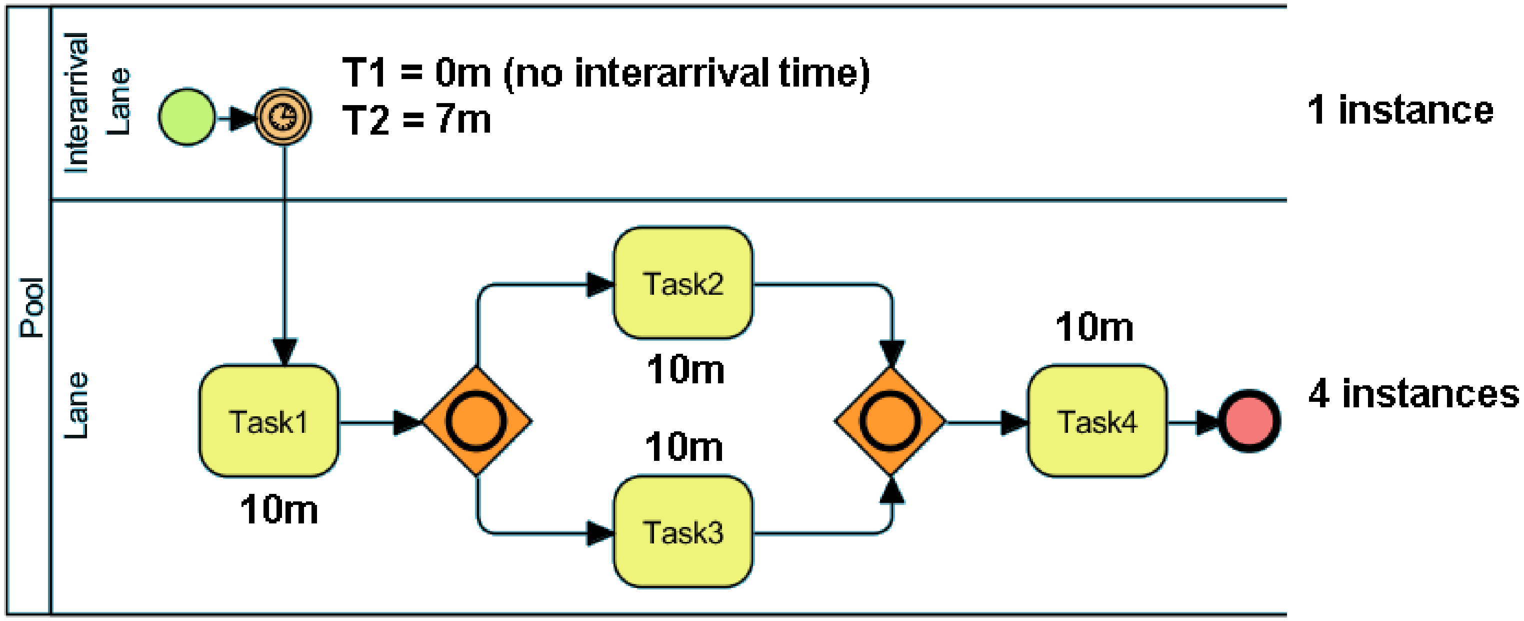
Name	Type	Processing	Cases
Pool	Pool/Lane	0	4
Series segments	Pool/Lane	1	1
Parallel segments	Pool/Lane	2	2

1 instance

2 instances

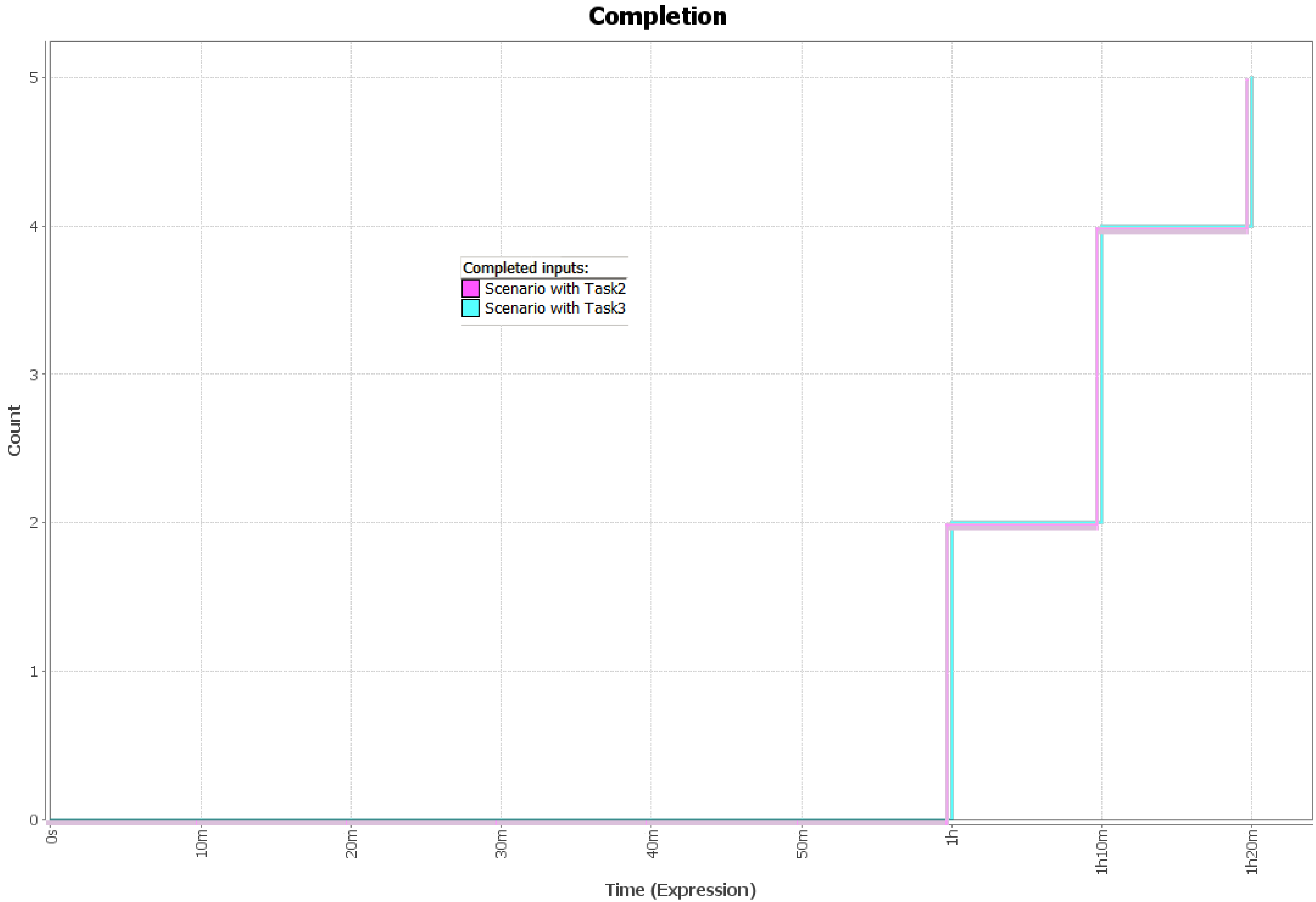
Emulating interarrival time

- When simulating no-peak scenarios, tokens (e.g. customers) arrive at time intervals. The average time between two consecutive customer is called **interarrival time**, and can be emulate with a 1-instance lane with a timer event.

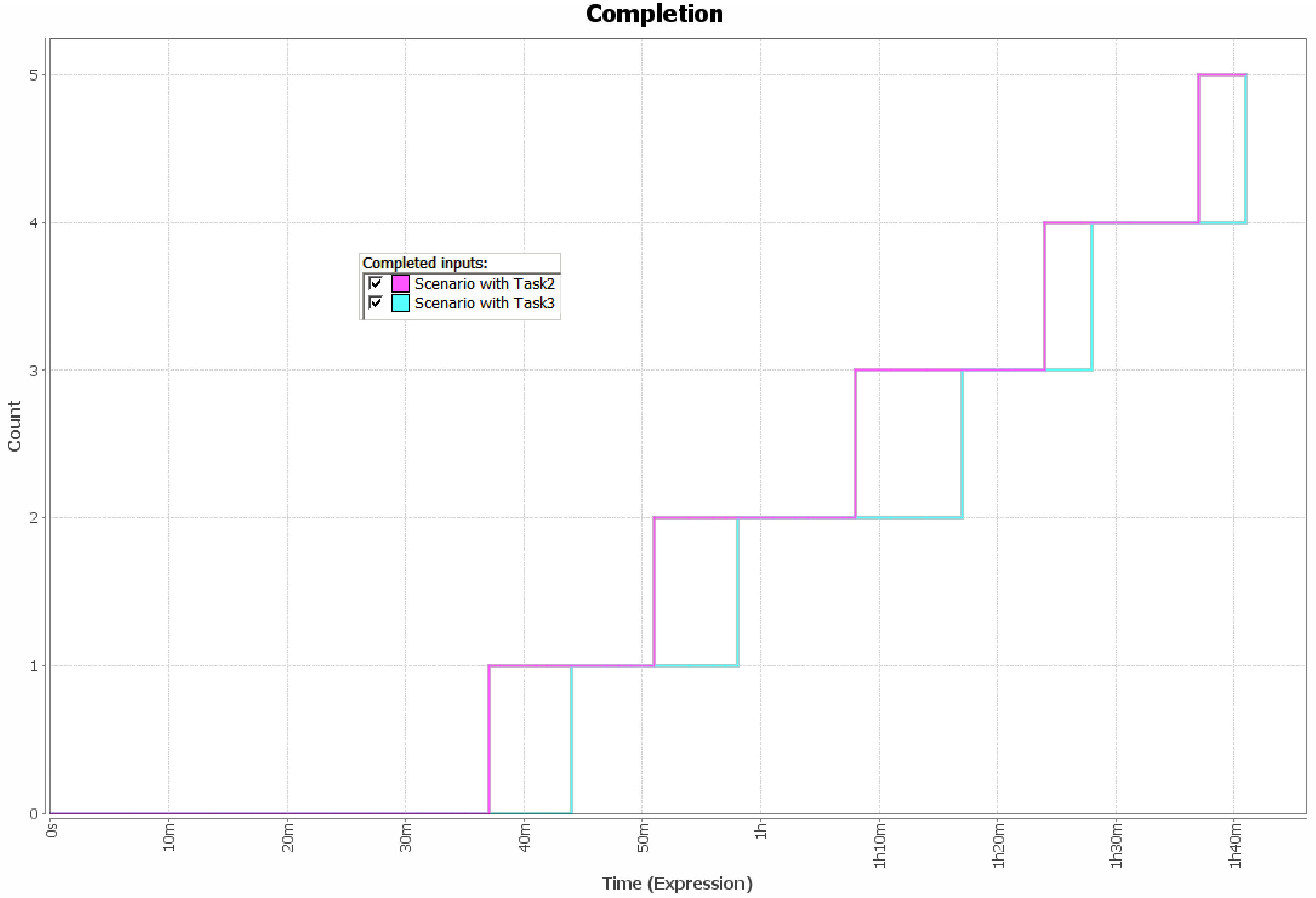


- Example1: with T1 the total duration is 1:20:00
- Example2: with T2 the total duration is 1:41:00

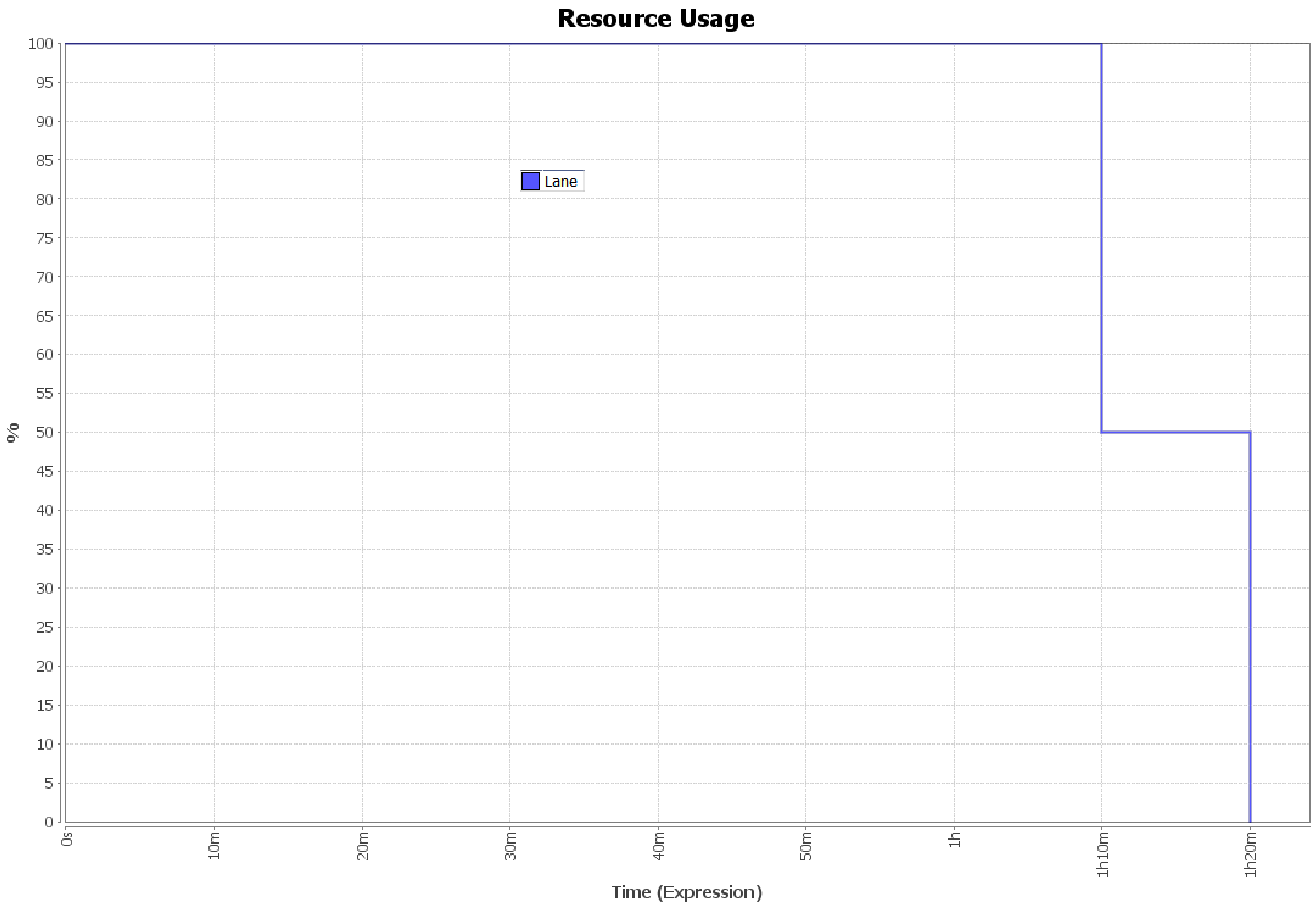
- **However, the interarrival time should be subtracted from the total duration**.


Completion
Completed inputs:
Scenario with Task2
Scenario with Task3
Count
0
1
2
3
4
5
Time (Expression)
0s
10m
20m
30m
40m
50m
1h
1h10m
1h20m

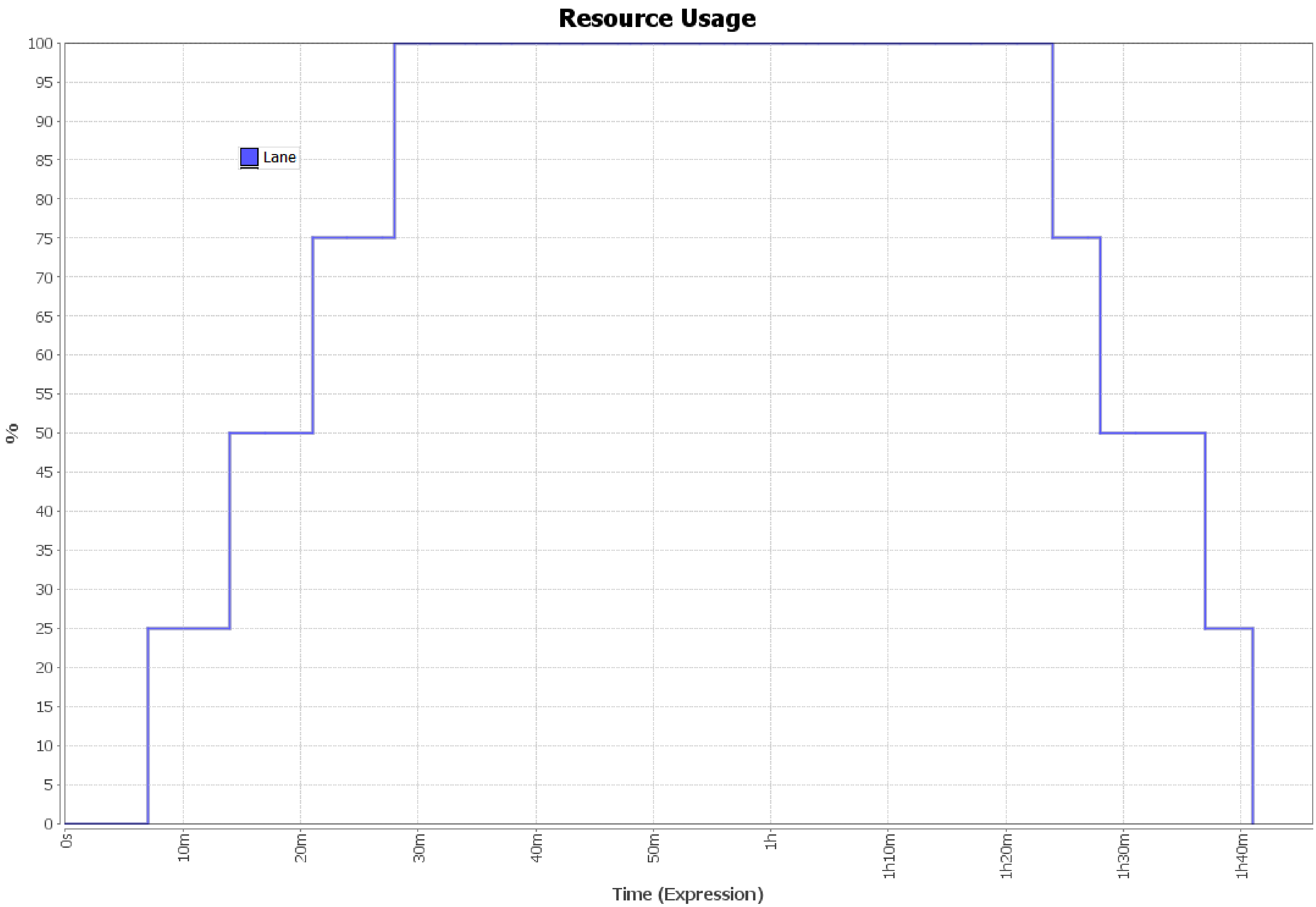
Example 1


Completion
Count
5
4
3
2
1
0
Completed inputs:
☑ 🟪 Scenario with Task2
☑ 🟦 Scenario with Task3
0s
10m
20m
30m
40m
50m
1h
1h10m
1h20m
1h30m
1h40m
Time (Expression)

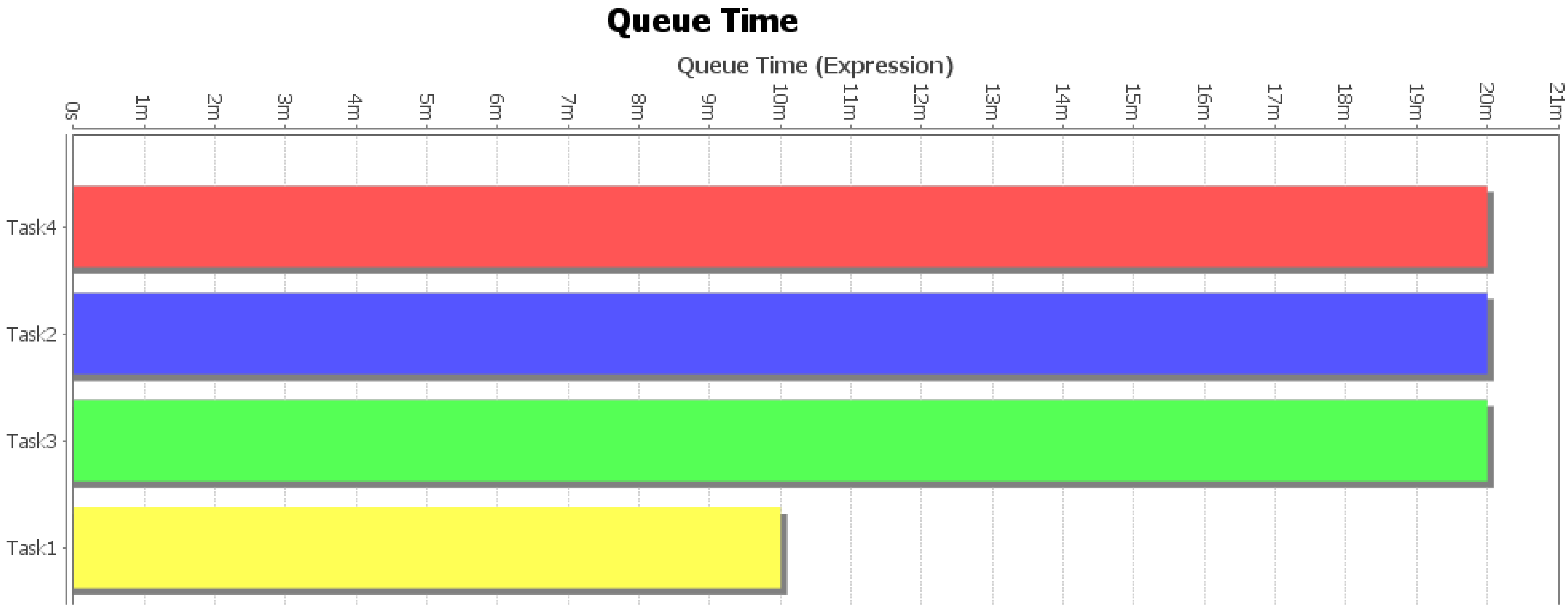
Example 2


Resource Usage
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
%
🟦 Lane
0s
10m
20m
30m
40m
50m
1h
1h10m
1h20m
Time (Expression)

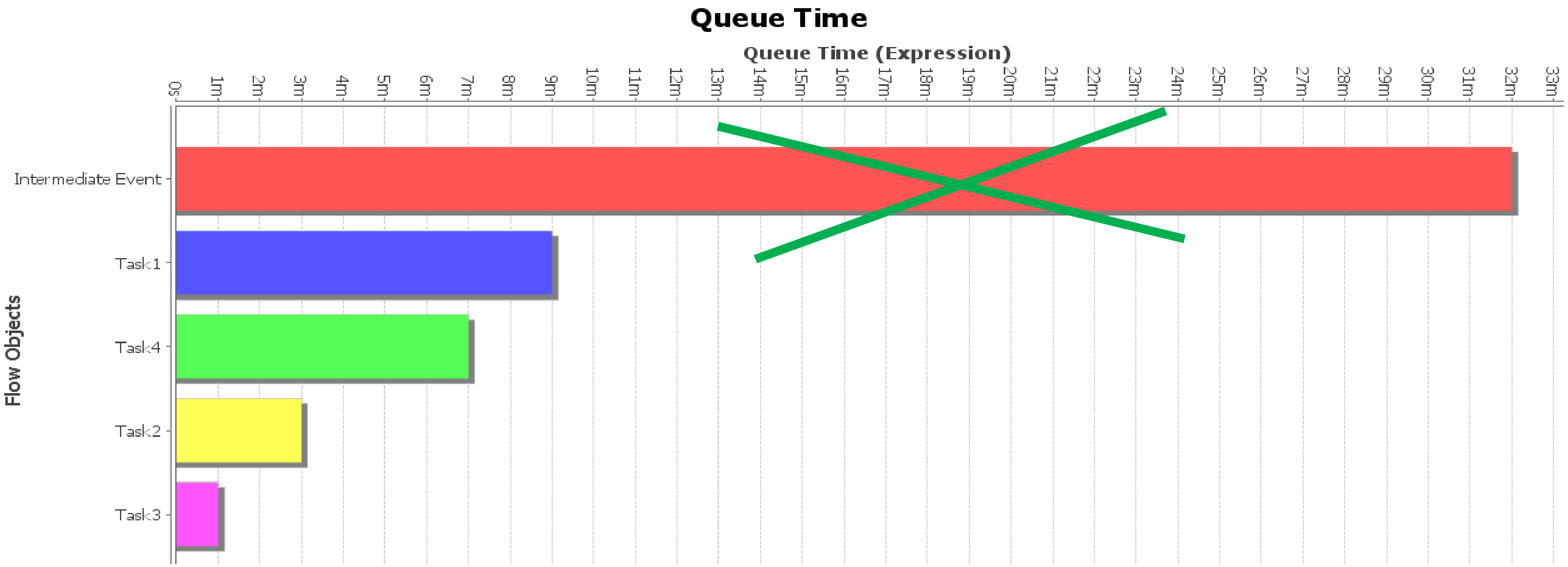
Example 1


Resource Usage
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
%
🟦 Lane
0s
10m
20m
30m
40m
50m
1h
1h10m
1h20m
1h30m
1h40m
Time (Expression)

Example 2


Queue Time
Queue Time (Expression)
0s
1m
2m
3m
4m
5m
6m
7m
8m
9m
10m
11m
12m
13m
14m
15m
16m
17m
18m
19m
20m
21m
Task4
Task2
Task3
Task1

Example 1


Queue Time
Queue Time (Expression)
0s
1m
2m
3m
4m
5m
6m
7m
8m
9m
10m
11m
12m
13m
14m
15m
16m
17m
18m
19m
20m
21m
22m
23m
24m
25m
26m
27m
28m
29m
30m
31m
32m
33m
Flow Objects
Intermediate Event
Task1
Task4
Task2
Task3

Example 2