Heuristic SAT solver for project scheduling with task preemption
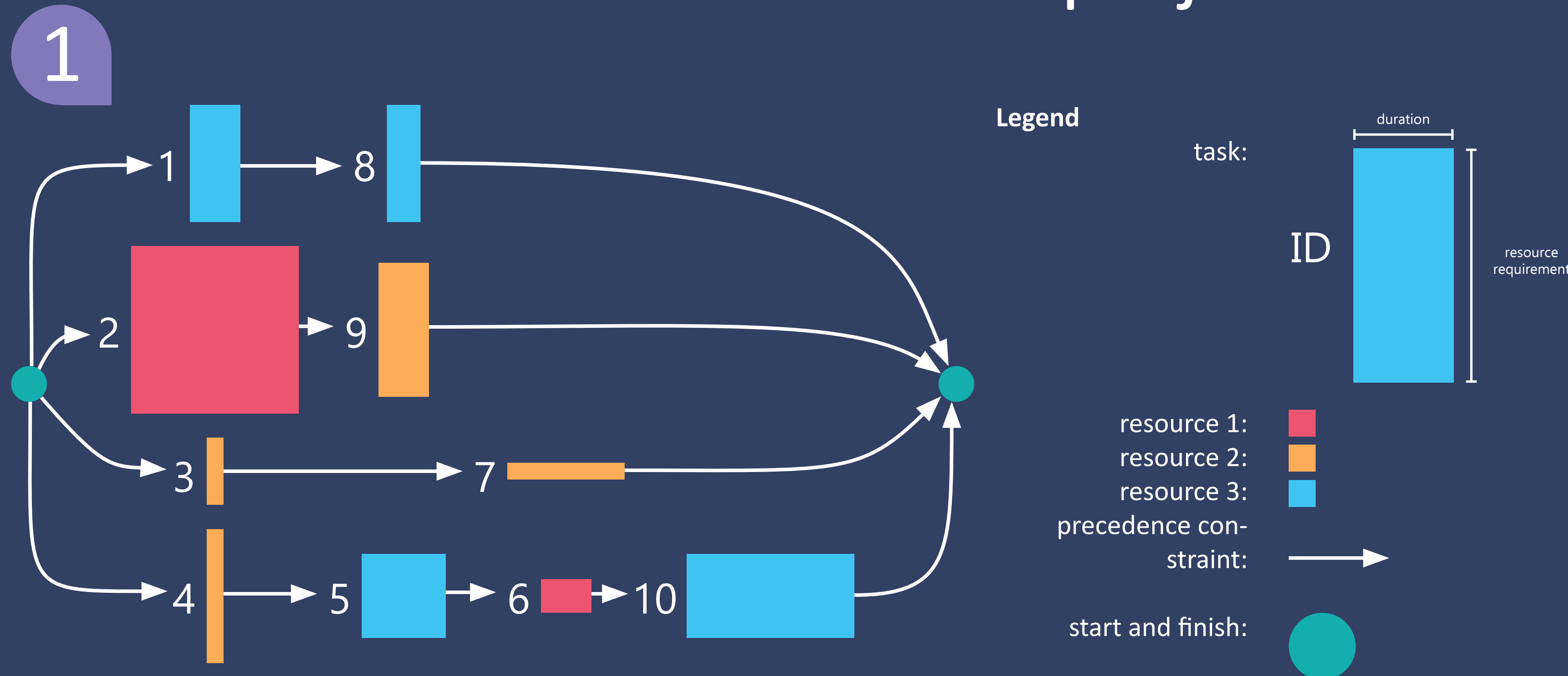
1

Legend

task:

duration

ID

resource requirement

resource 1:
resource 2:
resource 3:
precedence con-straint:

start and finish:

Figure 1: Example project task network
Source: Adapted from [1]

By: Jasper Vermeulen, j.vermeulen-1@student.tudelft.nl
Supervisor: Emir Demirović
At: 16/05/2022, Mekelweg 5, 2628 CD Delft

3

Background

Resource constraint project scheduling problem (RCPSP)
• a project has multiple tasks
• each task needs an amount of specific resource
• each task has a duration
• tasks have precedence constraints
• all tasks must be scheduled
• objective is to try and find the shortest makespan (makespan = total schedule time)

Preemption and setup times (PRCPSP-ST)
• a task can be put on hold before it finishes
• each preemption is penalized (setup time)
• preemption must result in a shorter schedule
Figure 2 and 3 show a reduction of the schedule time by allowing preemption

4

2

Motivation and research question

If production and logistics companies allocate their resources efficiently they can make more profit.
Algorithms can be made for a general version of the problem and take a long time return a result.
Alternatively a specialized algorithm can be used that is faster but very limited in application to custom requirements and constraints.
If the problem can be encoded for a very general problem solver (called SAT solvers) and then the solver can be taught a trick about the specific problem the best of both worlds might be combined.
So, the research question is:

"Can the addition of a simple heuristic to a SAT solver algorithm used to solve to PRCPSP-ST problems reduce the average makespan of the resulting schedule in an equal amount time? "

To answer the question three different algorithms are being made and compared on their result:
1. A heuristic algorithm
2. PRCPSP-ST encoding directly solved by a SAT solver
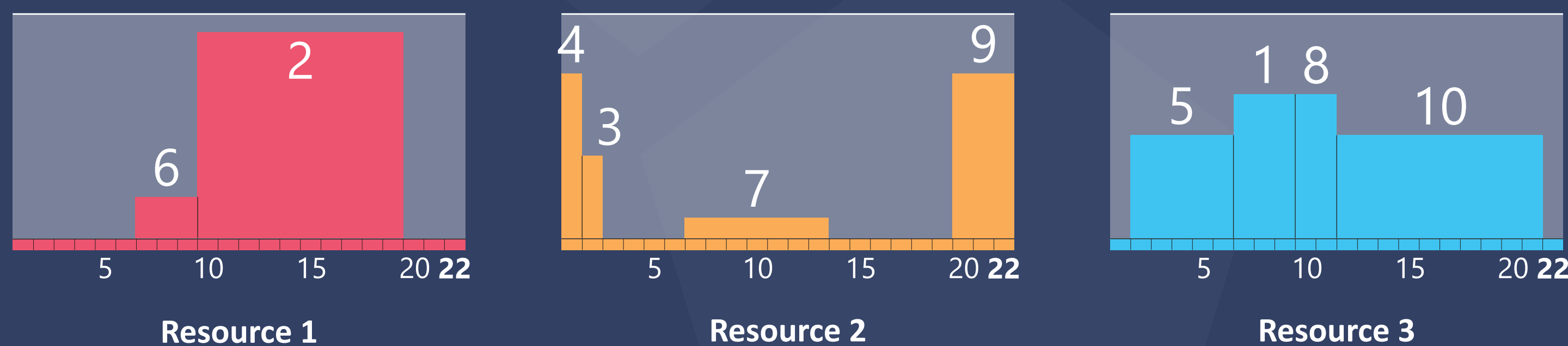3. Same encoding as in 2 but SAT solver is augmented with a heuristic

Work done by now

Done
• An adaption of the iterated greedy algorithm is implemented
• It solves simple instances (30 tasks) in a few seconds and can find the proven optimal solutions
• Most of the time no improvements to the schedule are found when preemption is allowed
• Hard to verify if the optimizations do not exist or are just not found

Work in progress
• Started work on the SAT encoding
• This encoding has to be done constraint per constraint
• For the second algorithm a MAX-SAT solver will be used
• It tries to make a feasible schedule while reducing an objective (eg. schedule time)

References
[1] Mario Vanhoucke and José Coelho. Resource-constrained project scheduling with activity splitting and setup times. *Computers Operations Research*, 109:230249, 2019.
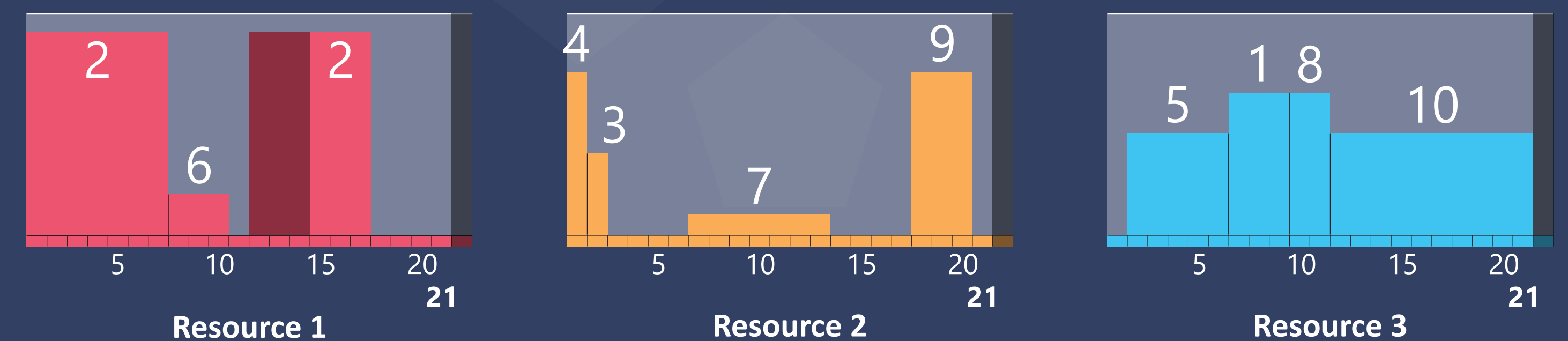
Resource 1   Resource 2   Resource 3

Figure 2: Shortest possible schedule of duration 22 without preemption
Source: Adapted from [1]

Resource 1   Resource 2   Resource 3

Figure 3: A shorter possible schedule of duration 21 with preemption of task 2
Source: Adapted from [1]