CSP Problem Findings: Car Assembly

Report by Brent Mercado

Using the Car Assembly, we have generated this schedule report for the assembly:

```
Optimal Schedule:
WheelLF: 19 minutes - Worker 4
WheelRB: 20 minutes - Worker 1
WheelLB: 21 minutes - Worker 3
AxleF: 0 minutes - Worker 1
AxleB: 10 minutes - Worker 2
WheelRF: 10 minutes - Worker 3
NutLB: 23 minutes - Worker 1
NutRB: 23 minutes - Worker 2
NutLF: 21 minutes - Worker 4
NutRF: 20 minutes - Worker 2
CapLB: 25 minutes - Worker 3
CapRB: 25 minutes - Worker 4
CapRLF: 25 minutes - Worker 1
CapRF: 25 minutes - Worker 2
Inspection: 25 minutes - Worker 3
```

The constraint program language that we utilized to solve this CSP is the python-constraint library. While creating this program, it became evident that though the library was fully capable of solving high level CSP's, certain points of interest appeared to be naive. An example of this occurrence was found when defining the constraint in which axles must be assembled separately. By defining both variables as all different (using the allDifferent() function), the schedule returned their times a minute apart instead of being at least 10 minutes apart. Due to this, I had to predefine a precedence constraint for the back axle, so to force the algorithm to complete the 10 minutes of installation before starting the next one. Furthermore, to find the true minimum time of the algorithm, I had to experiment with different domains to find the absolute minimum amount of time that all the tasks had to be completed in. Initially, when domain was set to 30 minutes the program had forced the inspection out to the last possible minute, which violates our domain constraint. This is because our algorithm chooses to spread the tasks throughout the domain to implement them up to the last moment - skewing beginning and end times. The fix for this was to simply try different domains to find the minimal schedule before the problem object (as defined in the beginning of the program) returned a default value, or 'No solution'.

It is evident that constraint language utilized to create this program had certain limitations. This discrepancy can be observed through the inspection task, where in which the program schedules the third worker to perform the inspection. Because this worker is already preoccupied doing the CapLB task, it cannot perform the same operation to do the inspection afterwards. This is something that could not be resolved during the creation of this program.

Despite these limitations, the schedule appears to be functional in nature. Applying the given constraints yielded a positive resultant output. The syntax used to define constraints was concise and readable and effectively reinforced the Constraint-Problem work-flow to solve this category of problems.