Warehouse Packaging Scheduling

ALC 21/22 - Project 2 Report - Group 5

Andreia Pereira 89414
Pedro Nunes 89525

Running Instructions

Our project was developed using Python and z3 (which must be installed), and z3's Optimize solver. Our code is in the directory /src/. To run the project from the main directory, simply type:

>>> python /src/project.py < instance_path.wps > res.out

Encoding

- X_{ioj} : Time at which Runner i places product j from order o in the conveyor belt.
- A_i : Time at which Runner i finishes working.
- P_{oj} : Time at which Product j from order o arrives at the packaging area.
- T : The total time of the problem.

Constraints

Some constraints are implemented implicitly by setting the variables' domains. This is the case for the following constraints:

- All products from all orders must arrive at the packaging area.
  - This is implicit since the domain for the P_{oj} variables is $]1, T]$.
- All runners start working at time 0.
  - This is implicit since the domain for the A_i variables is $]1, T]$.

Then, we explicitly implemented the following constraints:

1. A runner cannot spend less than 50% of the max timespan amongst other runners.
2. Runners start at an initial position and can go to any product shelf from there.
3. A runner can only carry one product at a time.
4. Only one product can arrive at the packaging area at a time.
5. A runner takes Tij time to go from product i to product j.
6. A runner can only carry a product if they're active
7. If a runner is at product i at time k, and arrives at product j at time k, then it does not carry any other product in times $]k, k+Tij[$.
8. A product j takes Cj time from the conveyor belt to the packaging area.
9. Each product j in order o is delivered by exactly one runner.

Breaking Symmetries

In order to achieve a better efficiency, we eliminate two types of symmetries in our encoding:

- If the same product appears in multiple orders, then there are various symmetric solutions. One solution that has $P_{oj} = k$ could be switched for one that has $P_{o'j} = k$. So, for any product j in that situation, we force a sequence by which the products j of all applicable orders arrive at the packaging station.
- If two runners have the same initial position, then it is irrelevant which runner does what sequence. So, again, we force a sequence by which the runners should stop working, using the variables A_i, in order to break that symmetry.

Optimal Time Search

After all constraints are encoded and symmetries are broken, we use the minimize() function available on the z3 library to minimize the variable T. This results in an optimal satisfiable solution in the minimum time possible.

Running Time Observations

The implementation using z3 and this encoding are clearly slower than the one in the first project. For this reason, our project cannot run some of the test instances provided by the faculty in a reasonable amount of time. This happens for tests t_2_3_10_2_2 and t_2_7_4_4_5.