Optimization Techniques - Final Report

João Martins, José Carvalho and Maria Lopes

Abstract

The paradigm of Industry 4.0 introduces a shift on traditional factory floors with the introduction of robotics, optimization, or artificial intelligence. In shop-floor or warehouse management, task allocation is one of the main challenges that a robot can face. This can include where, when, and what the robot should do at each time. This document describes the development of optimization techniques that allow optimal task scheduling for a robot that must manage a shop floor with input and output warehouses and machines. The robot must find the optimal solution for doing the necessary operations and delivering the final parts to the output. In this paper, Mixed Integer Programming (MIP) and Constraint Programming (CP) models are developed to minimize the distance covered by the robot. The results show that both approaches are successful at describing the scheduling problem and, when paired with a solver, can deliver optimal solutions for the actions of the robot. The execution times indicate that the MIP model does not scale well with an increased number of operations, as the number of constraints and variables consequently enlarges. This is ultimately a limitation of linear programming, which is overcome to some degree with constraint programming. To better understand the advantages of using each model, a thorough comparison between models will be conducted. All the files used in this project are available in this GitHub repository: https://github.com/mariaslopes/ratf_optimizer.

1 Introduction

This document addresses the development of optimization solutions based on MIP and CP to address the optimization challenge encountered in the RobotAtFactory 4.0 Competition[1], hosted at the Festival Nacional de Robótica. This competition is designed to simulate real-world scenarios involving the deployment of autonomous robots within factory environments. The primary objective is to leverage robotic capabilities for the efficient warehouse management and transportation of objects across the factory shop floor. The shop floor used in the competition can be visualized in Figure 1.

The shop floor map is described by a graph $\mathcal{G}$, where each node is denoted by an ID/index and attributed to an ArUco code. All adjacent nodes are connected to create edges. In the context of the competition, the objective entails transporting boxes from the input warehouses, denoted by indexes 50 to 53 in the figure, to the output warehouse at nodes 60 to 63. Each box, distinguished by its color, requires a specific sequence of operations that can be performed at designated machines. Boxes of blue color can be directly transferred to the output warehouse. Green boxes necessitate modifications achievable at either machine A (nodes 57, 58, 60, 61) or machine B (nodes 54, 55, 58, 59). The inputs for machines A and B are in the nodes 57, 61, 54, and 58, respectively, while the outputs are in the nodes 56 and 60 for machine A and 55 and 59 for machine B. To initiate the transformation process, a box must be placed at the input of the respective machine. The box is assumed to be immediately available for collection at the corresponding output. There are also red boxes that require two operations, sequentially passing through machine A and then machine B.

The focus is obtaining a sequence of movements for the robot to fulfill the objective while minimizing the total distance traveled. The robot in this scenario is omnidirectional, capable of translating and rotating simultaneously. It also can carry up to 4 boxes at the same time. In the initial simplified model, the robot's location is assumed to be a 2D point on the map, disregarding the orientation required for picking up or depositing the box.

Additionally, the competition is divided into three different stages. In stage one, there can only be blue boxes at the input warehouse. There can be blue or green boxes in stage two, and the operation done at the green boxes can be done on both machines. At the final stage, all three colors can be

[1] https://www.festivalnacionalrobotica.pt/2024-1/?page_id=1228

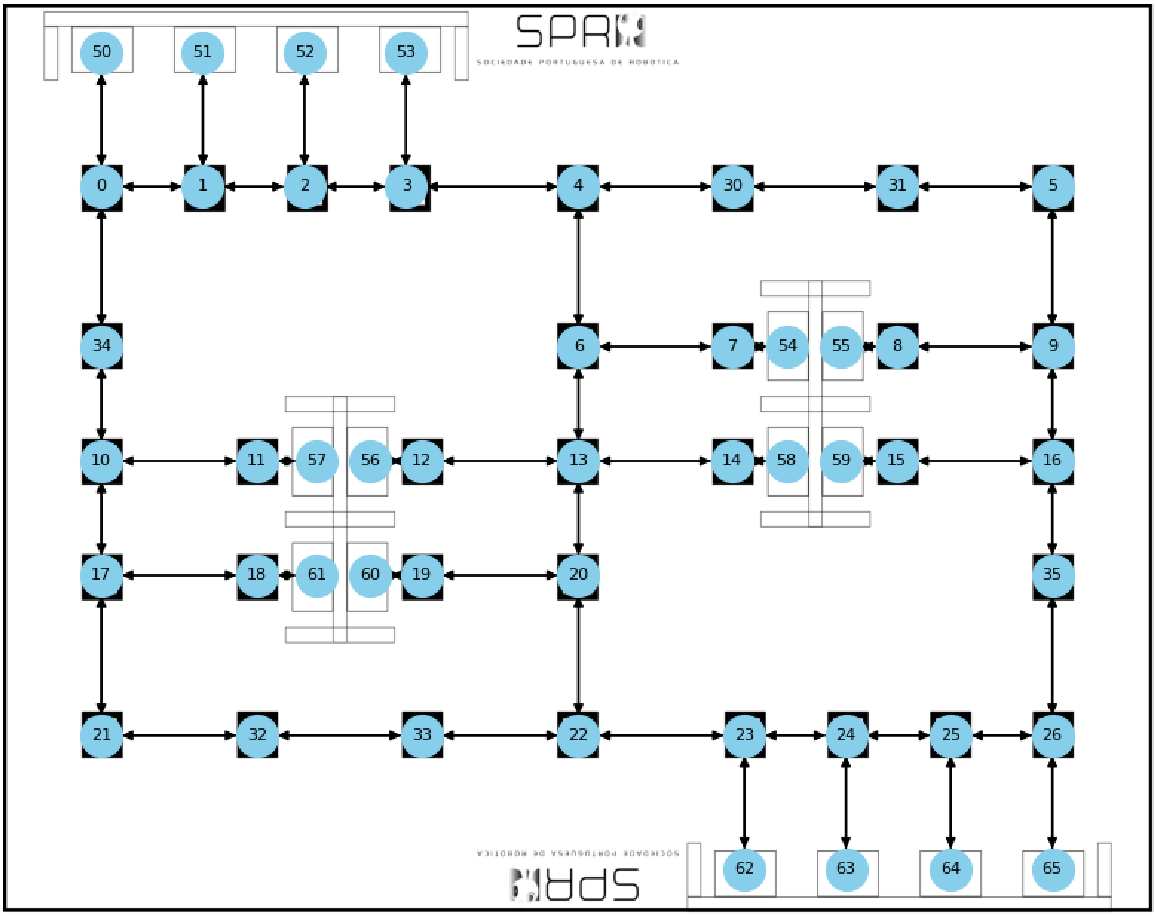


Figure 1: Shop floor map of the Robot@Factory 4.0 competition

present. In this case, the green boxes must only be worked in machine B. A video of a robot performing stage 3 of the competition can be visualized here.

In the official competition, the number of boxes of each color is known, but the exact order in which they appear is unknown until the stage starts. For the purpose of this work, it was considered that the number of boxes of each color is unknown. For this reason, the tests were divided into three parts, considering all the possible hypotheses for each stage.

Besides this introduction, the remainder of this paper is structured as follows. Section 2 showcases the mixed integer programming model developed. Section 3 shows the tests and results obtained by solving the problem using MIP model with Cplex Studio. Then, Sections 4 and 5 show the model and results, respectively, using the constraint programming model. Section 6 compares the two models. Finally, Section 7 summarizes the conclusions that can be driven from this work.

2 Mixed integer programming model

Sets:

$$\mathbb{L} = \{18, 19, 11, 12, 50, 51, 52, 53, 7, 8, 14, 15, 62, 63, 64, 65\} : \text{Possible robot locations indexes}$$

$$\mathbb{I} = \{50, 51, 52, 53\}, \mathbb{I} \subset \mathbb{L} : \text{Subset representing the locations of the input warehouses}$$

$$\mathbb{E} = \{62, 63, 64, 65\}, \mathbb{E} \subset \mathbb{L} : \text{Subset representing the locations of the output/exit warehouses}$$

$$\mathbb{P} = \{1, 2, 3, 4\} : \text{Number of boxes}$$

$$\mathbb{M} = \{7, 8, 11, 12, 18, 19, 14, 15\}, \mathbb{M} \subset \mathbb{L} : \text{Subset containning the machine locations}$$

$$\mathbb{M}^1 = \{7, 11, 14, 18\}, \mathbb{M}^1 \subset \mathbb{M} : \text{Subset containning the locations of the machine inputs}$$

$\mathbb{A} = \{11, 12, 18, 19\}, \mathbb{A} \subset \mathbb{M}$: Subset that contains all the Machine A locations.
$\mathbb{A}^{\mathbf{i}} = \{11, 18\}, \mathbb{A}^{\mathbf{i}} \subset \mathbb{A}$: Subset that contains all the Machine A input locations.
$\mathbb{A}^{\mathbf{o}} = \{12, 19\}, \mathbb{A}^{\mathbf{o}} \subset \mathbb{A}$: Subset that contains all the Machine A output locations.
$\mathbb{B} = \{7, 8, 14, 15\}, \mathbb{B} \subset \mathbb{M}$: Subset that contains all the Machine B locations.
$\mathbb{B}^{\mathbf{i}} = \{7, 14\}, \mathbb{B}^{\mathbf{i}} \subset \mathbb{B}$: Subset that contains all the Machine B input locations.
$\mathbb{B}^{\mathbf{o}} = \{8, 15\}, \mathbb{B}^{\mathbf{o}} \subset \mathbb{B}$ Subset that contains all the Machine B output locations.
$\mathbb{T} = \{0, \ldots, t_f\}$: Number of iterations.
$\mathbb{T}_1 = \{0, \ldots, t_f - 1\}$: Number of iterations minus the last.

The number of iterations needed to complete the task will change depending on the combination of boxes at the input warehouse. For each combination, that number can be quantified depending on the number of operations that each box (c), η_c, by the expression:

$$t_f = 2 \cdot \sum_{c \in \mathbb{P}} (\eta_c + 1) + 1$$

Data:

$d_{i,j}$: Length of the shortest path between location i and location j.
η_c : Number of operations that must be done to box c
o_i : Index offset from the input $i \in \mathbb{M}^{\mathbf{i}}$ to the output $k \in \mathbb{M}^{\mathbf{o}}$ of the Machine, $\quad k = i + o_i$
If $i \notin \mathbb{M}^{\mathbf{o}}$ then $o_i = 0$
L_c : Initial location of box c, $L_c \in \mathbb{I}$
R : Initial location of the robot, $R \in \mathbb{L}$
C : Robot carry capacity
M : Large constant

Indexes:

t : Timestamp, $t \in \mathbb{T}$
i and j : Locations, $i, j \in \mathbb{L}$
c : Boxes, $c \in \mathbb{P}$

In this case, t indicates the order in which the movements must be followed; it is the iteration index. As mentioned before, the competition has three different stages. So, first, the model was obtained for stage one and subsequently improved for the other stages.

2.1 Stage 1

Decision Variables:

$$x_{i,j}^t = \begin{cases} 1, & \text{if the robot moved from location } i \text{ to location } j \text{, at timestamp } t. \\ 0, & \text{otherwise.} \end{cases}$$

$$p_c^t = \begin{cases} 1, & \text{if the robot is carrying the box } c \text{ at timestamp } t. \\ 0, & \text{otherwise.} \end{cases}$$

$$b_{c,i}^t = \begin{cases} 1, & \text{if the box } c \text{ is at location } i \text{ at timestamp } t. \\ 0, & \text{otherwise.} \end{cases}$$

$$\sigma_{c,i}^t = \begin{cases} 1, & \text{if the robot picked the box } c \text{ at location } i \text{ at timestamp } t. \\ 0, & \text{otherwise.} \end{cases}$$

$$\delta_{c,i}^t = \begin{cases} 1, & \text{if the robot dropped the box } c \text{ at location } i \text{ at timestamp } t. \\ 0, & \text{otherwise.} \end{cases}$$

Objective:

$$\text{minimize} \quad z = \sum_{t \in \mathbb{T}} \sum_{i \in \mathbb{L}} \sum_{j \in \mathbb{L}} x_{i,j}^t d_{i,j}$$

Subject to:

Each box's initial location must be the position with index L_c:

$$b_{c,L_c}^0 = 1, \quad \forall c \in \mathbb{P} \tag{1}$$

The robot's initial location must be the position with index R:

$$x_{R,R}^0 = 1 \tag{2}$$

In the beginning, the robot is not carrying boxes:

$$p_c^0 = 0, \quad \forall c \in \mathbb{P} \tag{3}$$

All boxes must be in the output warehouse at the final timestamp:

$$b_{c,i}^{t_f} \geq ||\mathbb{P}||, \quad \forall c \in \mathbb{P} \tag{4}$$

The robot can carry up to C boxes at each timestamp:

$$\sum_{c \in \mathbb{P}} p_c^t \leq C, \quad \forall c \in \mathbb{P} \tag{5}$$

At each timestamp, the robot can only move once:

$$\sum_{i \in \mathbb{L}} \sum_{j \in \mathbb{L}} x_{i,j}^t = 1, \quad \forall t \in \mathbb{T} \tag{6}$$

The box has two possible locations, or the robot is carrying it or is in a location on the map:

$$\sum_{i \in \mathbb{L}} b_{c,i}^t + p_c^t = 1, \quad \forall c \in \mathbb{P}, \quad \forall t \in \mathbb{T} \tag{7}$$

At each location, there can only be one box a each timestamp:

$$\sum_{c \in \mathbb{P}} b_{c,i}^t \leq 1, \quad \forall i \in \mathbb{L}, \quad \forall t \in \mathbb{T} \tag{8}$$

Robot motion continuity flow:

$$\sum_{k \in \mathbb{L}} x_{k,i}^t \geq \sum_{j \in \mathbb{L}} x_{i,j}^{t+1}, \quad \forall i \in \mathbb{L}, \quad \forall t \in \mathbb{T}_1 \tag{9}$$

The robot can only pick a box from one location if it is at that same location:

$$\sigma_{c,j}^t \leq \sum_{i \in \mathbb{L}} x_{i,j}^t, \quad \forall c \in \mathbb{P}, \quad \forall j \in \mathbb{L}, \quad \forall t \in \mathbb{T} \tag{10}$$

The robot can only pick the box in one location if the box is at that location:

$$\sigma_{c,i}^t \leq b_{c,i}^t, \quad \forall c \in \mathbb{P}, \quad \forall i \in \mathbb{L}, \quad \forall t \in \mathbb{T} \tag{11}$$

If the robot picks the box, then the box is not in the previous location anymore:

$$b_{c,i}^{t+1} \leq b_{c,i}^{t} - \sigma_{c,i}^{t} + M\delta_{c,j}^{t}, \quad \forall c \in \mathbb{P}, \quad \forall i \in \mathbb{L}, \quad \forall t \in \mathbb{T}_1 \tag{12}$$

After picking one box, the robot is carrying one more than in the previous timestamp:

$$p_c^{t+1} \leq p_c^t + \sum_{i \in \mathbb{L}} (\sigma_{c,i}^t + M\delta_{c,i}^t), \quad \forall c \in \mathbb{P}, \quad \forall t \in \mathbb{T}_1 \tag{13}$$

The robot can only drop the box at one location if it is in that exact location:

$$\delta_{c,j}^t \leq \sum_{i \in \mathbb{L}} x_{i,j}^t, \quad \forall c \in \mathbb{P}, \quad \forall j \in \mathbb{L}, \quad \forall t \in \mathbb{T} \tag{14}$$

The robot can only drop the box at one location if there is no other box in that location:

$$\delta_{c,i}^t \leq 1 - \sum_{k \in \mathbb{P}} b_{k,i+o_i}^t, \quad \forall c \in \mathbb{P}, \quad \forall i \in \mathbb{L}, \quad \forall t \in \mathbb{T} \tag{15}$$

The robot can only drop the box if it is carrying it:

$$\sum_{i \in \mathbb{L}} \delta_{c,i}^t \leq p_c^t, \quad \forall c \in \mathbb{P}, \quad \forall t \in \mathbb{T} \tag{16}$$

If the robot drops the box in one location, then the box remains in that location:

$$b_{c,i+o_i}^{t+1} \geq \delta_{c,i}^t - M\sigma_{c,i}^t, \quad \forall c \in \mathbb{P}, \quad \forall i \in \mathbb{L}, \quad \forall t \in \mathbb{T}_1 \tag{17}$$

After dropping one box, the robot is carrying one less than in the previous timestamp:

$$p_c^{t+1} \geq p_c^t - \sum_{i \in \mathbb{L}} (\delta_{c,i}^t - M\sigma_{c,i}^t), \quad \forall c \in \mathbb{P}, \quad \forall t \in \mathbb{T}_1 \tag{18}$$

2.2 Stage 2

To address the complexities brought by the existence of a green box. The following decision variable and constraints were added to the model.

Decision variables:

n_c^t : The number of operations already done to box c at timestamp t

Constraints:

At the beginning, the boxes do not have operations performed:

$$n_c^0 = 0, \quad \forall c \in \mathbb{P} \tag{19}$$

At the final timestamp, each box must have been submitted to the correct number of operations:

$$n_c^{t_f} \geq \eta_c, \quad \forall c \in \mathbb{P} \tag{20}$$

When a box is dropped at the input of a machine, then an operation is done on it:

$$n_c^{t+1} \leq n_c^t + \sum_{i \in \mathbb{M}^{\mathbf{i}}} \delta_{c,i}^t, \quad \forall c \in \mathbb{P} \tag{21}$$

2.3 Stage 3

Finally, to adapt the model to deal with the red boxes and force the green boxes to pass through machine B, the following decision variables and constraints were included:

Decision variables:

$$\alpha_c^t = \begin{cases} 1, & \text{if the box } c \text{ must be operated at Machine A.} \\ 0, & \text{otherwise.} \end{cases}$$

$$\beta_c^t = \begin{cases} 1, & \text{f the box } c \text{ must be operated at Machine B.} \\ 0, & \text{otherwise.} \end{cases}$$

Constraints:

If the operation must be done in Machine A, the box must be dropped off at the inputs of that machine:

$$\delta_{c,i}^t \leq \alpha_c^t, \quad \forall c \in \mathbb{P}, \quad \forall i \in \mathbb{A}^{\mathbf{i}}, \quad \forall t \in \mathbb{T} \tag{22}$$

If the operation must be done in the Machine A, the box cannot be dropped on any other place besides the Machine A inputs:

$$\delta_{c,i}^t \leq 1 - \alpha_c^t, \quad \forall c \in \mathbb{P}, \quad \forall i \in \mathbb{L} \setminus \mathbb{A}^{\mathbf{i}}, \quad \forall t \in \mathbb{T} \tag{23}$$

If the box has operations to do in Machines A and B, it must drop the box in Machine A first. Otherwise, the box must be dropped in the inputs of Machine B:

$$\delta_{c,i}^t \leq \beta_c^t - \alpha_c^t, \quad \forall c \in \mathbb{P}, \quad \forall i \in \mathbb{B}^{\mathbf{i}}, \quad \forall t \in \mathbb{T} \tag{24}$$

$$\delta_{c,i}^t \leq 1 - \beta_c^t + \alpha_c^t, \quad \forall c \in \mathbb{P}, \quad \forall i \in \mathbb{L} \setminus \mathbb{B}^{\mathbf{i}}, \quad \forall t \in \mathbb{T} \tag{25}$$

If the operations needed in a box are greater or equal to two, one must be done in machine A and the anoother in B. If there is only one operation left, it must be done on Machine B:

$$M \cdot \beta_c^t \geq \eta_c - n_c^t, \quad \forall c \in \mathbb{P}, \quad \forall t \in \mathbb{T} \tag{26}$$

$$\alpha_c^t + \beta_c^t \leq \eta_c - n_c^t, \quad \forall c \in \mathbb{P}, \quad \forall t \in \mathbb{T} \tag{27}$$

3 Mixed integer programming: Tests and Results

To test the developed models, they were submitted to multiple instances. Initially, the idea was to run all possible test cases. However, there were more than 300 tests, and for more complex cases, the solver took more than 5 hours to find a feasible solution. For this reason, the tests started with the instances directly applicable to the competition. Then, other interesting cases that had not been included were addressed for testing and comparison purposes.

The Python scripts used to generate the instances and posterior data evaluation files can be found on our GitHub page. All instances were run in CPLEX Studio on a computer with an Intel i9-10900K @ 3.7GHz CPU and 64GB of RAM DDR4 @ 2933Mhz.

3.1 Stage 1

In stage 1, all the boxes are blue, and no operations are needed (they must only be transported from the input to the output warehouse). Table 1 show the results of this stage solved with each of the three models to understand how the variables and constraints added to the other models affect the performance when solving the simpler case. Additionally, the robot's capacity also varied between 1 and 4 boxes. For this stage, the number of movements the robot must do is fixed and can be obtained from Section 2, being equal to 9.

Table 1: Results from stage 1 executed with the three models

Boxes	Capacity	Objective (m)	Model 1		Model 2		Model 3	
			Time (min)	Iterations (M[2])	Time (min)	Iterations (M)	Time (min)	Iterations (M)
BBBB	1	14.62	00:14	1.7	00:16	2.0	00:14	1.9
BBBB	2	8.53	00:21	1.7	00:30	2.5	00:26	2.9
BBBB	3	7.93	00:20	1.9	00:23	1.7	00:20	1.4
BBBB	4	5.49	00:19	2.0	00:14	1.1	00:12	1.2
Constraints			4198		4242		5602	
Variables			4520		4560		4640	

The results show that model 1 was faster to solve for capacities 1 and 2. For capacities 3 and 4, model 3 could solve with fewer iterations, translating into similar or less time to solve. As it is possible to see, the number of variables and constraints increased across the models; however, this did not seem to impact its performance. Moreover, model 3, which has more variables and constraints, solved the problem with a time similar to model 1, for capacity equal to 1 and 3 or better than, with capacity equal to 4.

Model 2 showed comparatively poor results, except for capacity 4. It was expected to perform worse than model 3, as model 2 has more locations to process the green boxes, and it would take time to explore those branches. However, it was not expected that model 1 would be worse than any of the others since it does not consider the operations of the machines. On the other side, model 1 is less restricted, and therefore, it could consider picking and dropping a box from and to each one of the locations instead of a subset of $\mathbb{L}$, possibly explaining the delays.

3.2 Stage 2

Compared to stage 1, the stage 2 model includes processing the boxes in machines before taking them to the output warehouse. The tests of this stage consider two blue boxes and two green boxes, so the number of movements of the robot is always 13.

For these instances, the time to solve with this model was, in the worst case, 14:32 min, which limits its deployment in real-time scenarios. However, it only needs to run once. Therefore, running all cases in only a few hours is possible.

On the table of Appendix A.1, the results of running the testing scenarios of stage 2 with models 2 and 3 are illustrated. The difference is that model 3 limits green boxes to be processed only in machine B, while model 2 does not restrict where the operations must be performed. While model 3 has 30% more constraints and 2% more variables compared to model 2, it solved the problem 2.53 times faster on average, as seen in Figure 2.

Furthermore, the variance of model 3 is significantly lower compared with the one in model 2. Considering Figure 3, for all robot carry capacity values, the Inter Quartile Range (IQR) is smaller for all capacities using model 3. Moreover, the first quartile of model 2 is almost always worse than the third quartile of model 3, perfectly representing this difference in time efficiency. The main reason for this counter-intuitive speed-up appears to be the reduced feasible solution space.

Moreover, both models perform better for carry capacities $C > 2$ and significantly worse at $C = 2$, which was already observed in stage 1. This might be related to the degeneracy of the feasible solution space, where certain values of C reduce the number of similar solutions.

On a final note, due to the fact that by adding constraints to a model, the objective will be, in the best case equal, it is notorious in some cases that model 2 resulted in lower objective values than model 3, saving up 0.6 meters in the robot's trajectory, which is relevant for the competition. Hence, since neither model can be used in real-time to calculate the desired trajectory for the robot, model 2 will be used since it returns a smaller path.

[2] Millions

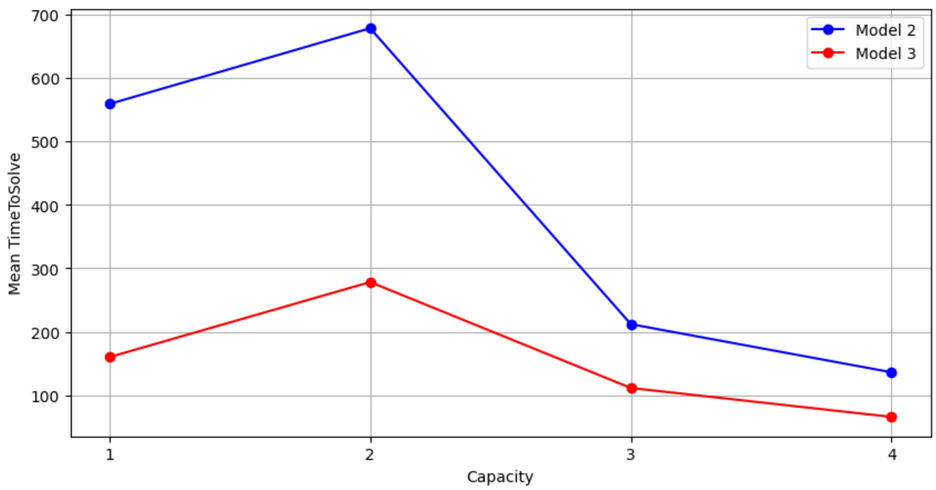


Figure 2: Comparison of the time needed to solve with the two models.

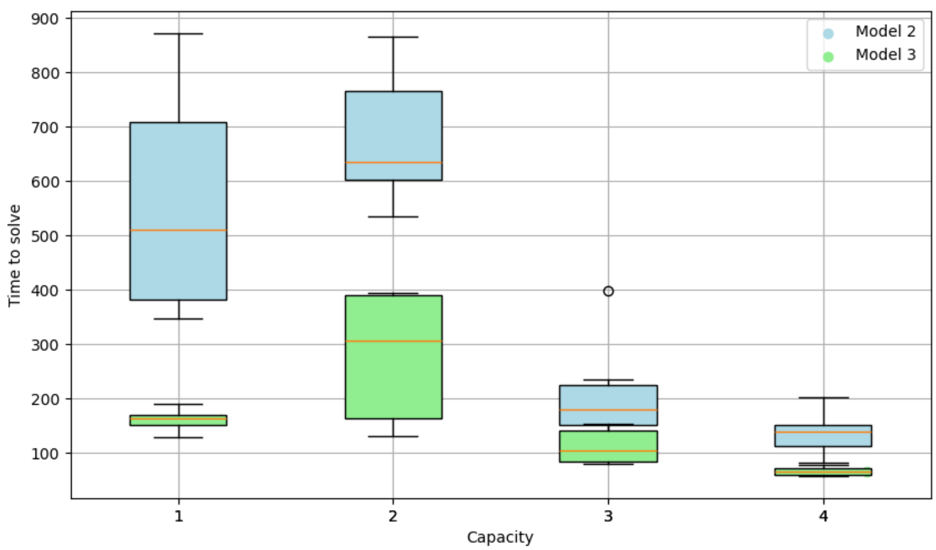


Figure 3: Box plots for time to solve in stage 2.

3.3 Stage 3

In all the stage 3 instances considered in this work, there will be a red box (2 operations), two green (1 operation), and one blue (no operations), which results in 17 movements for the robot. Since these tests were computationally expensive to the solver, as the number of variables and constraints increased significantly, a 1h timeout was defined for each instance. With this modification, it is important to include the gap and the lower bound of each instance to understand if the existing solution is close to the optimal and, therefore, acceptable. The table of Appendix A.2 includes the results obtained for this stage by varying the robot's capacity between 1 and 4 for each sequence.

As it is possible to observe, stage 3 is a much more complex problem than stages 1 and 2, requiring much more iterations and time to find the optimal solution. Furthermore, 1 hour was insufficient to find the optimal solution for almost all cases.

Figure 4 depicts the number of instances where the optimal solution was not found for all capacities. The worst case was when the robot's capacity was equal to 2, where the solver did not reach the optimal solution in 12 out of 12 instances. This is closely followed by $C = 3$ with 11 out of 12 instances with no optimal solution, whereas $C = 1$ and $C = 4$ achieved much better results with 0 and 3 truncated solutions, respectively.

However, looking only at the number of truncated solutions does not contain all the relevant

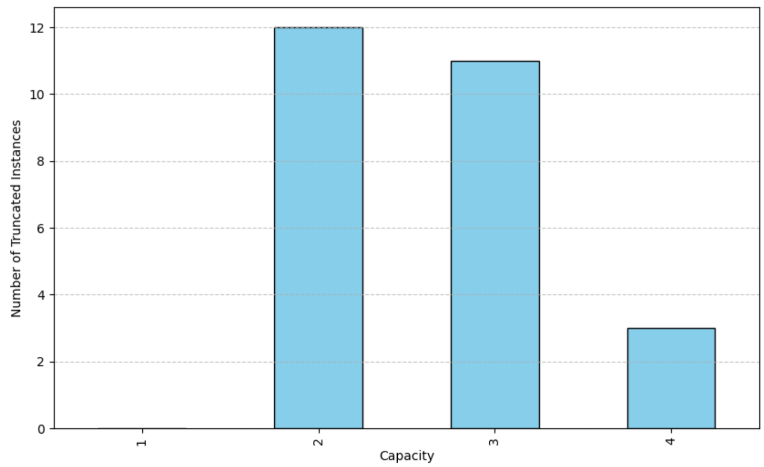


Figure 4: Number of truncated instances by capacity.

information. With this in mind, consider Figure 5, which contains two box plots with the relative gap and time to solve for all capacities.

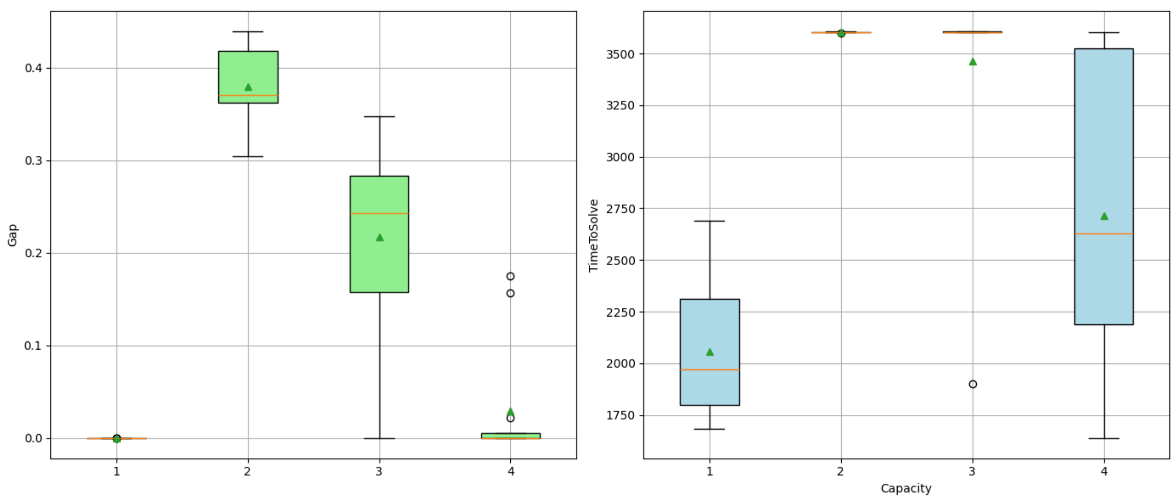


Figure 5: Box plots for time to solve in stage 3.

It is clearly visible that in the cases of $C = 1$ and $C = 4$, the median gap is very close to zero, and the median time to solve is well within the time limit. In the case of the worst scenarios, $C = 2$ and $C = 3$, while by only looking at the number of truncated instances, one could assume that they perform equally badly, this assumption would not be accurate. In the case of $C = 3$, its Q3 gap is lower than the median value of $C = 2$. Furthermore, the mean gap of $C = 3$ is 56% smaller. Finally, the mean of the gap values obtained was 0% for capacity 1, 38% for capacity 2, 22% for capacity 3, and 3% for capacity 4. These results are mostly consistent with the complexity analysis done for stage 2, where $C = 2$ is the harder scenario; however, $C = 1$ and $C = 3$ switch places, the latter more complex in stage 3.

3.4 Extra cases

To finalize the studies of the model, some other interesting cases were addressed; for capacities 1 and 2 (which are the guaranteed values for the competition), what would be the impact of adding more green boxes in stage 2? To answer this question, instances with 3 and 4 green boxes were executed. Table 2 contains the results of running the extra cases of stage 2 with model 2 for the boxes and capacities referred.

Table 2: Results from extra instances of stage 2

Boxes	Capacity	Time (h)	Objective (m)	Iterations (M)	Constraints	Variables	Movements	Gap (%)	Lower bound (m)
GGGB	1	00:54:13	14.52	151.8	6870	7296	15	0	14.52
GGGB	2	01:00:00	11.20	171.1				14	9.64
GGGG	1	01:00:09	17.95	109.5	7746	8208	17	87	2.39
GGGG	2	01:00:11	14.25	91.9				68	4.52

As it is possible to see, only the first case could be solved within the 1-hour time limit, while the others have been stopped before finding the optimal solution. The cases with 4 green boxes could deliver an integer solution with a gap of 87% and 68%, showing how far they were from exploring all solutions and reaching an optimal solution. In fact, this behavior is expected since there are 4 machine slots to put each one of the boxes, and there are many combinations, resulting in a large solution space to explore. Furthermore, it is well represented that capacity 2 significantly reduces the objective function, corroborating the results of the other test runs.

For stage 3, the same assumption was made, so cases with more red boxes were also performed but just for capacity 1, given the increased delay brought by capacity 2. Since the complexity is much greater than in the previous cases, the timeout was increased to 5 hours instead of 1 hour. The results can be found in Table 3.

Table 3: Results from extra instances performed for stage 3

Boxes	Time (h)	Objective (m)	Iterations (M)	Constraints	Variables	Movements	Gap (%)	Lower bound (m)
RRBB	00:25:55	15.53	61.5	10194	8352	17	0	15.53
RRGB	01:46:22	17.77	213.2	11342	9280	19	0	17.77
RRGG	05:00:10	19.86	416.4	12490	10208	21	55	8.88
RRRB	05:00:02	20.16	490.6	12490	10208	21	62	7.73
RRRG	05:00:00	No solution found yet						
RRRR	05:00:11	25.89	301.2	14786	12064	25	84	4.24

As one can see from the results, the solver efficiently identifies the optimal solution within the allotted time frame when dealing with up to two red boxes and one green box.

However, when the number of red boxes exceeds three, the solver fails to find the optimal solution within the specified 5-hour window. In nearly all cases, except for one instance where no feasible solution was found within the time available, a solution was obtained. Yet, this solution has a significant gap, exceeding 50%, indicating that the result might still be substantially reduced.

4 Constraint Programming Model

This formulation was based on common scheduling problems, using the following concepts:

- **Tasks** - These represent all phases a box is submitted to. There are four types of tasks:
  - *Entry* - the box is waiting in the input warehouse;
  - *Picking X* - the box is being picked for the Xth time;
  - *Droping X* - the box is being dropped for the Xth time;
  - *Machine Y* - the box needs to be processed by the machine:
    - A for $Y = A$;
    - B for $Y = B$;
    - Either one for Y empty;
  - *Output* - the box is complete and in the output warehouse.

- **Workers** - These indicate the places that may perform tasks, which include:
  - Input and output warehouses: $E1, E2, E3, E4, O1, O2, O3, O4$;
  - Robot: R ;
  - Machine entries: $A1, A2, B1, B2$;
- **Skills** - These relate the tasks to the workers and specify the locations in the set $\mathbb{L}$ where they must be performed;
  - The robot can perform the picking and dropping tasks that do not have a specific location;
  - Tasks named *Entry*, *Machine*, and *Output* are performed by works E, A or B, and O, respectively, and each worker has an associated location of set $\mathbb{L}$. For example $E1$ is location 50.
- **Precedences** - These dictate the order in which the tasks must be performed and their temporal relationship. For instance, the *Picking* must follow an *Entry*, and both must be finished same time.

4.1 Mathematical formulation

For this model, the sets and data variables introduced in Section 2 are again utilized, and the following were added:

Definitions[3]:

Interval: A variable that defines the relationship between the start time t_s and end time t_e of an operation. Note that $t_s, t_e \in \mathbb{T}$ and $t_s \leq t_e$;

- `startOf`(I)=t_s;
- `endOf`(I)=t_e.

Precendece: A set of two intervals I_1, I_2 that are related by a temporal constraint;

- `startAtEnd`(I_1, I_2): A constraint that enforces that $t_s^1 = t_e^2$;
- `endBeforeStart`(I_1, I_2): A constraint that enforces that $t_s^1 = t_e^2$;
- `endAtEnd`(I_1, I_2): A constraint that enforces that $t_e^1 = t_e^2$.

Skill: A variable that includes a worker w, the operation o, performed by that worker, and the location l where o can be performed;

`cumulFunction`: A variable that models a quantity changing over time based on the decision variables, using the following functions:

- `stepAtStart`(I, v): A function that returns v at the start of the interval I;
- `stepAtEnd`(I, v): A function that returns v at the end of the interval I.

`noOverlap`$(\{I_1, \ldots, I_n\})$: A constraint that enforces that there are no overlaps in a given set of intervals;

`alternative`$(I,\{I_1, \ldots, I_n\})$: A constraint that ensures only one interval from the set $\{I_1, \ldots, I_n\}$, is active during the interval I ;

`alwaysIn`(x, t_1, t_2, v_1, v_2): A constraint that ensures the variable x stays within the range v_1 to v_2 during the time interval from t_1 and t_2.

[3]More details can be found here.

Data:

$\mathbb{O}$: Set of all the operations to perform in the current instance
$\mathbb{O}_p$: Set of all picking operations $\in \mathbb{O}$
$\mathbb{O}_d$: Set of all dropping operations $\in \mathbb{O}$
$\mathbb{O}_m$: Set of all machine-related operations $\in \mathbb{O}$
$\mathbb{O}^c$: Set of mandatory operations $\subset \mathbb{O}$ that a part $c \in \mathbb{P}$ must undergo
$\mathbb{S}$: Set of all skills
$\boldsymbol{\rho}^c$: Set of the precedences for the operations of each box $c \in \mathbb{P}$
$\boldsymbol{\rho}^c_{se}$: Set of the precedences $\subset \boldsymbol{\rho}^c$ where the temporal relation is '*startAtEnd*'
$\boldsymbol{\rho}^c_{ee}$: Set of the precedences $\subset \boldsymbol{\rho}^c$ where the temporal relation is '*endAtEnd*'
$\boldsymbol{\rho}^c_{es}$: Set of the precedences $\subset \boldsymbol{\rho}^c$ where the temporal relation is '*endBeforeStart*'
$\mathbb{W}$: Set of all workers
$\mathbb{W}^O$: Set of all Output Warehouses
D^o : Deadline of each operation $o \in \mathbb{O}$, contained in $[0, t_f[$

Decision Variables:

X^t : Location of the robot at each timestamp , $t \in \mathbb{T}_1$

$\texttt{tasks}^c_o$: Set of intervals where each task must be addressed, $c \in \mathbb{P}, o \in \mathbb{O}$

- Optional if $o \notin \mathbb{O}^c$
- Duration can be within an interval $\subset D^o$

$\texttt{wtasks}^c_s$: Set of intervals where a worker must address a task, $c \in \mathbb{P}, s \in \mathbb{S}$

- Optional for any values
- Duration can be within an interval $\subset D^o$

Cumulative Function:
The robot's capacity changes only with the picking and dropping operations

$$\begin{aligned}\texttt{current\_capacity} = &\sum_{c\in\mathbb{P}} \sum_{p\in\mathbb{O}_p\cap\mathbb{O}^c} \texttt{stepAtStart}(\texttt{tasks}^c_p, 1) \\ &- \sum_{c\in\mathbb{P}} \sum_{d\in\mathbb{O}_d\cap\mathbb{O}^c} \texttt{stepAtEnd}(\texttt{tasks}^c_d, 1)\end{aligned}$$

Objective:

$$\text{minimize} z = \sum_{t\in\mathbb{T}_1} d_{X^t, X^{t+1}}$$

Constraints:
At all timestamps, the robot's capacity can not be surpassed:

$$\texttt{alwaysIn}(\texttt{current\_capacity}, 0, t_f, 0, C) \tag{1}$$

The initial position of the robot is location 18:

$$X^0 = 18 \tag{2}$$

All the locations of the robot must belong to set $\mathbb{L}$:

$$X^t \in \mathbb{L}, \quad \forall t \in \mathbb{T}_1 \tag{3}$$

In the beginning ($t = 0$), all the boxes are in the input warehouse:

$$\texttt{startOf}(\texttt{tasks}^c_{Entry}) = 0, \quad \forall c \in \mathbb{P} \tag{4}$$

An operation of picking and dropping can not occur simultaneously:

$$\texttt{noOverlap}(\{\texttt{tasks}^c_o | \forall o \in \mathbb{O}_p \cap \mathbb{O}_d\}), \quad \forall c \in \mathbb{P} \tag{5}$$

Apply the temporal relations to the operations:

$$\texttt{startAtEnd}(\texttt{tasks}^c_{o_1}, \texttt{tasks}^c_{o_2}), \quad \forall\{o_1, o_2\} \in \boldsymbol{\rho}^c_{se}, \quad \forall c \in \mathbb{P} \tag{6}$$

$$\texttt{endAtEnd}(\texttt{tasks}^c_{o_1}, \texttt{tasks}^c_{o_2}), \quad \forall\{o_1, o_2\} \in \boldsymbol{\rho}^c_{ee}, \quad \forall c \in \mathbb{P} \tag{7}$$

$$\texttt{endBeforeStart}(\texttt{tasks}^c_{o_1}, \texttt{tasks}^c_{o_2}), \quad \forall\{o_1, o_2\} \in \boldsymbol{\rho}^c_{es}, \quad \forall c \in \mathbb{P} \tag{8}$$

An operation can only be assigned to one worker:

$$\texttt{alternative}(\texttt{tasks}^c_o, \{\texttt{wtasks}^c_s | \forall s \in \mathbb{S}, s_o = o\}), \quad \forall o \in \mathbb{O}, \quad \forall c \in \mathbb{P} \tag{9}$$

The Entry task must be assigned to the worker in the initial condition L_c:

$$\texttt{alternative}(\texttt{tasks}^c_{Entry}, \{\texttt{wtasks}^c_s | \forall s \in \mathbb{S}, s_l = L_c\}), \quad \forall c \in \mathbb{P} \tag{10}$$

All the boxes must be in the exit warehouse in the final timestamp:

$$\texttt{endOf}(\texttt{tasks}^c_{Output}) = t_f, \quad \forall c \in \mathbb{P} \tag{11}$$

The operations must be performed by only one of the eligible workers:

$$\texttt{noOverlap}(\{\texttt{wtasks}^c_s | \forall s \in \mathbb{S}, s_w = w, \forall c \in \mathbb{P}\}), \quad \forall w \in \mathbb{W}, \quad \forall c \in \mathbb{P}, \tag{12}$$

When a box leaves the input warehouse, the robot is in that location:

$$\texttt{endOf}(\texttt{wtasks}^c_s) = t \Rightarrow X^t = s_l, \quad \forall s, s_o = \mathit{Entry}, \quad \forall c \in \mathbb{P}, \quad \forall t \in \mathbb{T}_1 \tag{13}$$

When a box arrives in the output warehouse, the robot is in that location:

$$\texttt{startOf}(\texttt{wtasks}^c_s) = t \Rightarrow X^t = s_l, \quad \forall s, s_o = \mathit{Output}, \quad \forall c \in \mathbb{P}, \quad \forall t \in \mathbb{T}_1 \tag{14}$$

When a box arrives in a machine, the robot in that machine input location:

$$\texttt{startOf}(\texttt{wtasks}^c_s) = t \Rightarrow X^t = s_l, \quad \forall s, s_o \in \mathbb{O}_m, \quad \forall c \in \mathbb{P}, \quad \forall t \in \mathbb{T}_1 \tag{15}$$

When a box leaves in a machine, the robot must be in that machine output location:

$$\texttt{endOf}(\texttt{wtasks}^c_s) = t \Rightarrow X^t = s_l + 1, \quad \forall s, s_o \in \mathbb{O}_m, \quad \forall c \in \mathbb{P}, \quad \forall t \in \mathbb{T}_1 \tag{16}$$

In order to reduce the solution space, some constraints were added, which made the solver faster. For instance, some impossible precedences were pointed out so the model does not have to consider them. Hence, the following constraint was added to guarantee that the impossible precedences do not occur.

$$\texttt{noOverlap}(\texttt{tasks}^c_o | \forall o \in \mathbb{O} \setminus \boldsymbol{\rho}^c), \quad \forall c \in \mathbb{P} \tag{17}$$

Furthermore, all optional tasks $o_o \in \mathbb{O} \setminus \mathbb{O}_m$ are contained in the set of precedences $\boldsymbol{\rho}^c_{es}$, in the form of $\{\texttt{Output}, o_o\}$ for the constraint (8). This way, since constraint (11) enforces that the `Output` task must finish in the final timestep, all the optional tasks do not have time to start, greatly reducing the solution space, as otherwise, they could be scheduled anywhere, without any impact on the optimal solution.

5 Contraint Programming: Tests and Results

To evaluate the Constraint Programming model, two sets of tests were executed. The first consisted of performing a Grid Search on the CP search-phase hyperparameters in order to find the best-performing ones and evaluate how they evolve with problem complexity. The final set of tests was to subject the CP model to the same tests done on the MIP model in order to understand the correctness and efficiency of the provided solutions.

5.1 Hyperparameter selection

To find the best set of hyperparameters for the search phase, a grid search method was employed. The search methods considered were Iterative Diving and Restart, as they were by far the best-performing ones. As for the variable and value selection possibilities, all the available functions in CPLEX were tested, resulting in 252 possible combinations. These possible combinations of parameters were tested in a Round 2 scenario, "GGBB", using third-round rules, which means that the green box can be operated by either one of the machines. The capacities $C \in \{1, 2, 3\}$ were considered. Unfortunately, the case with $C = 4$ could not be evaluated due to time constraints. For each scenario, the top 10 parameter combinations are presented in Table 4, Table 5, and Table 6 respectively. Consult the Appendix B.1 for the whole results list.

Table 4: Top 10 parameters combination for capacity (C) equal to 1.

Method	Variable Selection	Value Selection	Time (s)
IterativeDiving	selectSmallest(regretOnMin())	selectSmallest(value())	1.192
IterativeDiving	selectSmallest(domainMin())	selectSmallest(valueSuccessRate())	1.193
IterativeDiving	selectSmallest(regretOnMin())	selectRandomValue()	1.198
IterativeDiving	selectSmallest(domainMin())	selectSmallest(value())	1.211
IterativeDiving	selectSmallest(regretOnMin())	selectLargest(value())	1.212
IterativeDiving	selectSmallest(domainMin())	selectRandomValue()	1.212
IterativeDiving	selectSmallest(domainMax())	selectSmallest(valueSuccessRate())	1.221
IterativeDiving	selectSmallest(domainMin())	selectLargest(valueSuccessRate())	1.221
IterativeDiving	selectSmallest(regretOnMin())	selectSmallest(valueSuccessRate())	1.222
IterativeDiving	selectSmallest(domainMin())	selectLargest(value())	1.230

Table 5: Top 10 parameters combination for capacity (C) equal to 2.

Method	Variable Selection	Value Selection	Time (s)
Restart	selectSmallest(domainMin())	selectSmallest(value())	20.735
Restart	selectSmallest(domainMin())	selectSmallest(valueSuccessRate())	20.863
Restart	selectSmallest(domainMin())	selectLargest(value())	20.890
Restart	selectSmallest(domainMin())	selectSmallest(valueImpact())	20.898
Restart	selectSmallest(domainMin())	selectLargest(valueImpact())	20.898
Restart	selectSmallest(regretOnMin())	selectLargest(value())	20.931
Restart	selectSmallest(domainMax())	selectLargest(valueSuccessRate())	20.941
Restart	selectSmallest(domainMin())	selectLargest(valueSuccessRate())	20.956
Restart	selectSmallest(regretOnMin())	selectSmallest(valueImpact())	20.968
Restart	selectSmallest(regretOnMin())	selectLargest(valueImpact())	20.995

Table 6: Top 10 parameters combination for capacity (C) equal to 3.

Method	Variable Selection	Value Selection	Time (s)
IterativeDiving	selectLargest(impact())	selectSmallest(value())	74.12
IterativeDiving	selectLargest(domainMin())	selectSmallest(value())	74.205
IterativeDiving	selectSmallest(successRate())	selectSmallest(value())	76.753
IterativeDiving	selectLargest(impact())	selectSmallest(valueImpact())	80.333
IterativeDiving	selectLargest(domainMin())	selectSmallest(valueSuccessRate())	80.689
IterativeDiving	selectSmallest(successRate())	selectSmallest(valueSuccessRate())	80.785
IterativeDiving	selectLargest(domainMin())	selectSmallest(valueImpact())	80.852
IterativeDiving	selectSmallest(successRate())	selectSmallest(valueImpact())	80.862
IterativeDiving	selectLargest(impact())	selectSmallest(valueSuccessRate())	81.046
IterativeDiving	selectLargest(domainMin())	selectLargest(value())	81.32

From the results, it is possible to conclude that Iterative Diving is the best search-phase method for $C = 1$ and $C = 3$, but puzzlingly worse for $C = 2$. One reason for this occurrence might be related to the solution space having more degeneracy in the problem when $C = 2$, and the Restart method suits this situation better. For the tests, Restart was chosen for $C = 2$ and Iterative Diving for the rest.

Furthermore, on the value selection side, the criteria of selecting the smallest parameter are well suited for all sub-problems, and selecting the consistently ranks well, therefore this option was chosen for value selection on all cases.

Finally, for the variable selection, with $C = 1$ and $C = 2$, selecting the variable with the smallest domain minimum value was very well suited, consistently ranking in the top 10. Therefore, this was the chosen variable selector in those scenarios. For $C = 3$, the choice of variable selector is more difficult, as there were parameter choices that did not appear well ranked in the other scenarios. For this case, the chosen selector was selecting the smallest Success Rate, as it appears three times in the top 10 and goes well with the value selector.

It is important to note that, due to the lack of time, the assumption that $C = 4$ would be similar to $C = 3$ was made, and all the hyperparameters are the same for both scenarios.

5.2 Results from instance execution

To test the efficacy and correctness of the CP model, it was subjected to the same test dataset as the MIP. The results made it possible to conclude that the model outputs correct solutions. Whenever the MIP obtained the optimal solution, the CP model matched it, and all solutions better than the MIP were also valid. Furthermore, it is possible to see a leap in performance, which will be analyzed more in-depth in Section 6.

The results show that the model is quick in solving instances where the robot capacity is $C = 1$, which results are present in Table 7. This is visible because all instances with this condition were finished in less than 5 seconds. and even when the MIP did not find a feasible solution after 5 hours, CP found the optimal one in less than 15 minutes. However, the model struggles with $C > 1$ due to the parallelism that it introduces. However, it is important to note that the model always finds the optimal solution quickly (usually in less than 10 seconds), but then the engine takes a lot of time to explore the whole solution space and guarantee that the solution is actually optimal. This fact was exploited in Round 3, as the engine timeout was set to 2 minutes, and the optimal solution was found for all cases[4]. The results of stages 2 and 3 can be found in Appendices B.2 and B.3, respectively.

Considering now Figure 6, which represents graphically the results of using the CP model for solving the Round 2 problem, where machines are involved. As done for MIP, two cases were considered; in

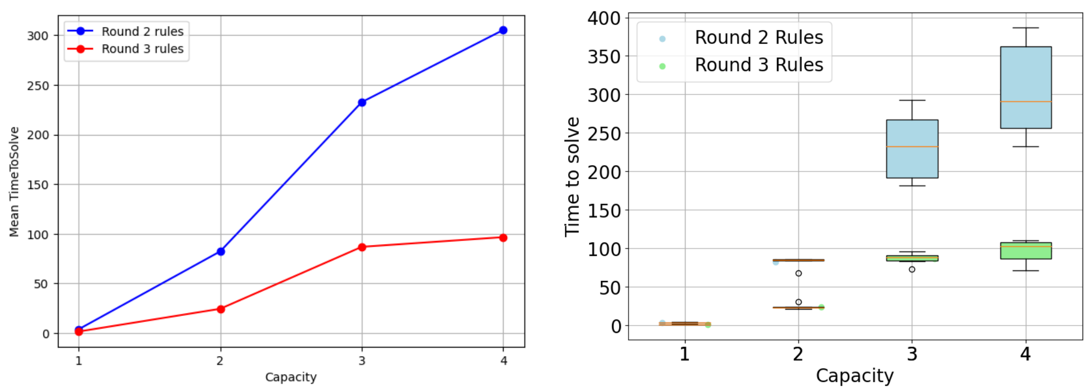
[4]The time limit had to be set to 2 minutes because the OPL engine always crashed after running for 50 minutes, and therefore it was infeasible to run large testbeds. A similar issue was found online; however, a solution was not found until the deadline of the project; hence the time was truncated.

Table 7: Results from Stage 1

Boxes	**Capacity**	**Time (s)**	**Objective (m)**
BBBB	1	0.669	14.62
BBBB	2	1.112	8.53
BBBB	3	2.354	7.93
BBBB	4	2.404	5.49

one, the box can go to any machine, which is the round 2 rule, and in another, where the box can only go to Machine B, which is the round 3 rule. Unlike in the MIP, no different models exist in CP; only the data changes.

Going over the analysis, it is possible to see that solving with round 2 rules takes, in general, more time to reach the solution. This is because the search space includes selecting which machine the box should be operated on. On the other hand, rules from round 3 consider that that operation can only be executed in Machine B, which greatly reduces the search space and allows the solver to reach a solution faster. These results are consistent with the ones presented for the MIP model; however, in this case, the effects are more noticeable as the capacity grows.



(a) (b)

Figure 6: Comparison of time to solve for Round 2 with the rulesets from round 2 and 3

As far as the extra rounds, which are not expected in the competition but represent the more complex cases for the model, it is possible to see that CP can scale even to the most complex cases when $C = 1$. Note that once again, even in the case "GGGG" with $C = 2$ where the algorithm timed out, it still found a solution in less than 30 seconds, a solution which is most likely optimal. The results are presented in Table 8 for stage 2 and Table 9 for stage 3.

Table 8: Results of the Round 2 extra cases

Boxes	**Capacity**	**Time (s)**	**Objective (m)**
GGGB	1	14.592	14.518
GGGG	1	64.627	14.932
GGGB	2	1261.62	11.058
GGGG	2	3000.034	11.652

Table 9: Results of the Round 3 extra cases

Boxes	Capacity	Time (s)	Objective (m)
RRBB	1	6.146	15.532
RRGB	1	44.971	17.774
RRGG	1	275.548	19.561
RRRB	1	86.894	20.018
RRRG	1	480.979	21.685
RRRR	1	816.299	23.929

6 Performance comparison - MIP vs CP

A direct comparison between the times and objective function is now addressed. In Table 10, a resume of the results obtained from the CP and MIP for stage 1 is shown. This is the simplest case, which requires only the robot to transport the boxes from the input to the output warehouses, and the difference in time to reach the same objective function is already significant, with CP being almost 10 times faster.

Table 10: Comparasion between MIP and CP for stage 1.

Boxes	Capacity	Time (s)		Objective (m)
		MIP	CP	
BBBB	1	13.69	0.67	14.62
BBBB	2	21.48	1.11	8.53
BBBB	3	19.56	2.35	7.93
BBBB	4	19.13	2.40	5.49

The tables from Appendice C have a summary of the side-by-side results of MIP and CP. Appendix C.1 has the results from running the stage 2 instances having both machines as hypotheses, while in Appendix C.2, the round 3 rules are applied instead. For both cases, the two models converged for the same optimal value, with the CP being notably faster at reaching the solution. For round 3, C.3, CP was able to reach an optimal solution within the 2 minutes to which it was limited, while the MIP had a limit time of 1 hour that was reached for most part of the scenarios.

Consider now Figure Figure 7, which visually depicts the number of times the CP model solved faster or slower than the MIP model. It is visible that while most of the time CP is faster, there are some cases where MIP solves it quicker. These cases are mostly in Round 2 with Capacity > 3 because while CP scales poorly with capacity, MIP is mostly unaffected, being that the impact is mostly due to the number of operations. However, it is worth noting that in this case, CP actually finds the solution in less than 10 seconds, and the rest of the time is spent exploring the rest of the solution space.

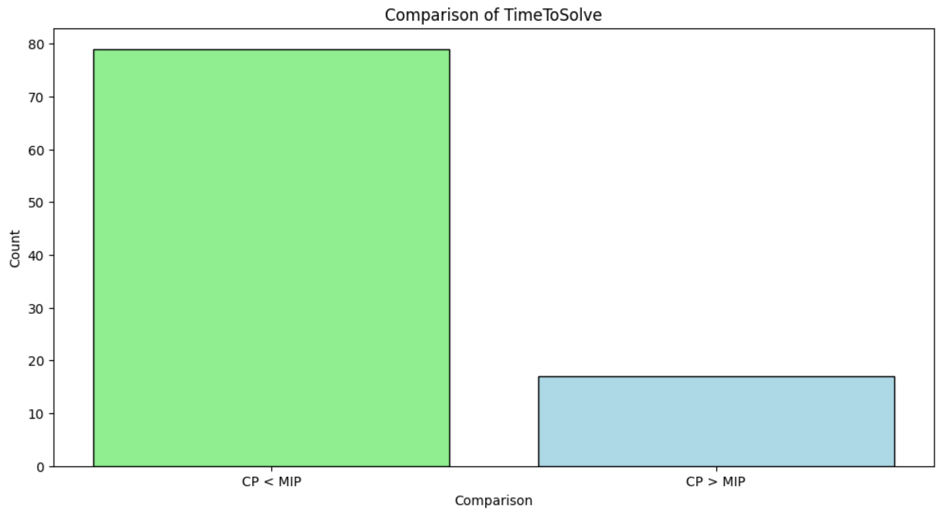


Figure 7: Comparison of Time to Solve between MIP vs CP. Number of events in which CP > MIP or CP < MIP

7 Conclusions

This document described the formulation and development of Mixed Integer Programming and Constraint Programming models for the optimal scheduling of a robot managing a shop floor. The problem proposal is based on the Robot@Factory 4.0 Competition at the Portuguese Robotic Open. The goal was to find the sequence of actions (move to a position, pick up box a box, drop off box) that satisfied all the operations of the boxes in the input warehouse while minimizing the distance covered by the robot.

With this purpose in mind, first, the mathematical model for the MIP was formulated with three different sub-models. The models are designed incrementally, as the problem is part of a competition that increases in difficulty depending on the stage. A base model is described for stage 1, and more constraints and variables are introduced for stages 2 and 3. These models take into account all restrictions and rules defined by the competition rulebook.

The models were implemented in IBM ILOG CPLEX Optimization Studio, and tested in multiple instances for the different models, stages, and robot carry capacities. The results show that: The model was well formulated and designed and can deliver optimal solutions. For stages 1 and 2 and any capacity, the time it takes to solve is feasible with MIP (< 1h), but it is not adequate to use during the competition, where the stages must be completed in less than 10 minutes. The model does not scale well with the number of operations, as the time to solve starts being unfeasible (> 5h) in some situations with more than two red boxes. Generally, the hardest scenarios are when the carrying capacity equals 2, suggesting more degeneracy in the solution space with this parameter.

During the development of the MIP model, it was also possible to see the effects of the constraints on the solution space and how it can affect the solver efficiency, proved by the case study in stage 1. Some extra instances were added to address the impact of more complex cases on the model. The results show that 1 hour is insufficient to solve these cases as the resulting gap is large or no solution was found for the MIP model. Furthermore, the number of iterations was significantly higher within 1 hour, explained by the resulting large solution space, as many combinations could address the problem and be used by the robot to transport the boxes between machines and warehouses.

The second part of the document described the development and deployment of the Constraint Programming model. This model was much faster throughout the tests, reaching the optimal solution within seconds. However, in some instances, it takes a long time to finish searching the remaining solution space. With this model, it was possible to reach a solution almost 200 times faster than the MIP model. Increasing the number of operations was not the main complexity factor for the CP model. Instead, increasing the robot carry capacity was the main factor that increased the solution time, which is a different behavior when compared to the MIP. One interesting aspect to explore in future work is the issue of taking too long to explore the whole solution space. One possible solution would be to introduce some global constraints and prune the feasible solutions.

In conclusion, while the model formulation and implementation were successful for both MIP and CP, the limitations of both techniques were also evident. Nonetheless, with the employed optimization techniques, the Robot@Factory 4.0 Scheduling problem is effectively solved, as optimal solutions were achieved for all possible competition scenarios.

A Mixed Integer Programming

A.1 Results from stage 2

Table 11: Results from stage 2 performed with models 2 and 3.

Boxes	Capacity	Model 2			Model 3		
		Time (min)	**Objective (m)**	**Iterations (M)**	**Time (min)**	**Objective (m)**	**Iterations (M)**
BBGG	1	06:20	14.70	24.4	02:51	14.70	11.3
BBGG	2	10:36	10.61	37.7	06:32	10.61	23.3
BBGG	3	02:12	9.37	8.6	02:34	9.41	10.8
BBGG	4	01:42	8.48	7.6	01:08	8.48	4.1
BGBG	1	14:32	14.55	51.8	02:08	14.70	7.6
BGBG	2	09:52	10.61	41.3	02:20	10.61	8.5
BGBG	3	02:46	9.07	9.8	01:35	9.41	6.1
BGBG	4	02:19	8.48	8.7	00:58	8.48	3.9
BGGB	1	06:28	14.55	26.7	02:43	14.70	11.4
BGGB	2	10:32	10.47	36.3	06:22	10.61	24.7
BGGB	3	06:39	9.07	21.7	01:19	9.41	4.0
BGGB	4	01:21	8.48	4.5	01:11	8.48	5.1
GBBG	1	10:34	14.25	32.4	02:28	14.70	9.6
GBBG	2	13:29	10.31	48.8	02:10	10.31	8.4
GBBG	3	02:27	8.77	8.4	01:53	9.41	5.7
GBBG	4	02:35	8.48	9.5	00:58	8.48	4.2
GBGB	1	12:13	14.25	35.3	03:10	14.70	13.1
GBGB	2	14:26	10.31	60.6	06:34	10.31	30.2
GBGB	3	03:14	8.77	11.6	01:19	9.41	5.3
GBGB	4	03:22	8.48	12.2	01:18	8.48	5.0
GGBB	1	05:47	14.25	18.3	02:42	14.70	11.1
GGBB	2	08:54	10.01	28.1	03:51	10.01	13.7
GGBB	3	03:55	8.77	13.8	02:30	9.41	9.2
GGBB	4	02:19	8.48	7.4	01:03	8.48	4.2
Constraints		5994			7898		
Variables		6384			6496		

A.2 Results from stage 3

Table 12: Results from stage 3.

Boxes	Capacity	Time (h)	Objective (m)	Gap (%)	Lower bound (m)	Iterations (M)
BGGR	1	00:30:03	17.66	0	17.66	75.5
BGGR	2	01:00:05	15.14	37	9.55	121.9
BGGR	3	01:00:05	13.81	31	9.52	132.3
BGGR	4	00:47:34	12.85	0	12.85	120.6
BGRG	1	00:28:02	17.36	0	17.36	68.2
BGRG	2	01:00:03	15.11	31	10.38	143.5
BGRG	3	01:00:05	13.66	24	10.34	134.7
BGRG	4	00:46:21	12.85	0	12.85	118.1
BRGG	1	00:38:20	17.36	0	17.36	91.0
BRGG	2	01:00:06	14.69	37	9.25	126.7
BRGG	3	01:00:03	13.58	22	10.54	143.3
BRGG	4	01:00:01	12.85	2	12.57	142.8
GBGR	1	00:31:07	17.66	0	17.66	73.6
GBGR	2	01:00:05	14.99	44	8.41	144.4
GBGR	3	01:00:08	14.23	35	9.29	114.7
GBGR	4	00:35:06	12.85	0	12.85	93.8
GBRG	1	00:39:38	17.36	0	17.36	90.8
GBRG	2	01:00:05	14.97	37	9.43	128.4
GBRG	3	01:00:07	13.51	25	10.08	109.0
GBRG	4	00:27:18	12.85	0	12.85	71.4
GGBR	1	00:44:49	17.66	0	17.66	97.8
GGBR	2	01:00:05	14.98	37	9.42	150.2
GGBR	3	01:00:01	13.81	14	11.91	159.8
GGBR	4	00:41:16	12.85	0	12.85	102.7
GGRB	1	00:33:39	17.36	0	17.36	75.7
GGRB	2	01:00:01	14.69	30	10.22	150.1
GGRB	3	01:00:05	13.66	27	9.92	138.0
GGRB	4	00:58:19	12.85	0	12.85	145.7
GRBG	1	00:32:04	17.06	0	17.06	70.7
GRBG	2	01:00:05	14.68	40	8.82	132.7
GRBG	3	01:00:02	13.51	16	11.29	137.2
GRBG	4	01:00:02	12.85	18	10.60	133.9
GRGB	1	00:39:01	17.06	0	17.06	97.9
GRGB	2	01:00:04	14.81	42	8.64	145.8
GRGB	3	01:00:07	13.51	33	9.08	111.9
GRGB	4	01:00:01	12.85	16	10.84	144.8
RBGG	1	00:28:46	17.06	0	17.06	69.9
RBGG	2	01:00:05	14.69	44	8.26	130.5
RBGG	3	00:31:40	13.28	0	13.28	86.7
RBGG	4	00:36:56	12.85	0	12.85	101.0
RGBG	1	00:35:51	16.76	0	16.76	92.7

RGBG	2	01:00:02	14.39	34	9.46	151.8
RGBG	3	01:00:02	13.28	7	12.30	143.4
RGBG	4	00:31:53	12.85	0	12.85	82.5
RGGB	1	00:29:38	16.76	0	16.76	61.7
RGGB	2	01:00:05	14.09	42	8.15	127.8
RGGB	3	01:00:03	13.51	24	10.23	146.0
RGGB	4	00:37:50	12.85	0	12.85	92.9

B Constraint Programming

B.1 Hyperparameter Selection

B.1.1 Capacity = 1

Table 13: All hyperparameter combinations for capacity one ordered by time to solve

Method	Variable Selection	Value Selection	Value (s)
IterativeDiving	selectSmallest(regretOnMin())	selectSmallest(value())	1.192
IterativeDiving	selectSmallest(domainMin())	selectSmallest(valueSuccessRate())	1.193
IterativeDiving	selectSmallest(regretOnMin())	selectRandomValue()	1.198
IterativeDiving	selectSmallest(domainMin())	selectSmallest(value())	1.211
IterativeDiving	selectSmallest(domainMin())	selectRandomValue()	1.212
IterativeDiving	selectSmallest(regretOnMin())	selectLargest(value())	1.212
IterativeDiving	selectSmallest(domainMin())	selectLargest(valueSuccessRate())	1.221
IterativeDiving	selectSmallest(domainMax())	selectSmallest(valueSuccessRate())	1.221
IterativeDiving	selectSmallest(regretOnMin())	selectSmallest(valueSuccessRate())	1.222
IterativeDiving	selectSmallest(domainMin())	selectLargest(value())	1.23
IterativeDiving	selectSmallest(domainMax())	selectSmallest(value())	1.23
IterativeDiving	selectSmallest(domainMin())	selectLargest(valueImpact())	1.231
IterativeDiving	selectSmallest(successRate())	selectRandomValue()	1.239
IterativeDiving	selectSmallest(successRate())	selectLargest(value())	1.24
IterativeDiving	selectSmallest(domainMax())	selectLargest(valueSuccessRate())	1.243
IterativeDiving	selectSmallest(domainMax())	selectLargest(value())	1.248
IterativeDiving	selectSmallest(impact())	selectSmallest(valueSuccessRate())	1.248
IterativeDiving	selectSmallest(domainMax())	selectRandomValue()	1.249
IterativeDiving	selectSmallest(regretOnMax())	selectLargest(valueImpact())	1.249
IterativeDiving	selectSmallest(regretOnMax())	selectLargest(valueSuccessRate())	1.253
IterativeDiving	selectSmallest(impact())	selectLargest(valueSuccessRate())	1.254
IterativeDiving	selectSmallest(regretOnMin())	selectLargest(valueSuccessRate())	1.254
IterativeDiving	selectSmallest(regretOnMax())	selectSmallest(valueSuccessRate())	1.26
IterativeDiving	selectSmallest(localImpact())	selectSmallest(valueImpact())	1.265
IterativeDiving	selectSmallest(domainSize())	selectSmallest(valueSuccessRate())	1.266
IterativeDiving	selectSmallest(regretOnMax())	selectSmallest(value())	1.267
IterativeDiving	selectSmallest(impact())	selectLargest(value())	1.27
IterativeDiving	selectSmallest(domainSize())	selectSmallest(value())	1.27
IterativeDiving	selectSmallest(ImpactOfLastBranch())	selectSmallest(value())	1.278
IterativeDiving	selectSmallest(ImpactOfLastBranch())	selectLargest(value())	1.279
IterativeDiving	selectSmallest(regretOnMax())	selectSmallest(valueImpact())	1.28
IterativeDiving	selectSmallest(impact())	selectSmallest(valueImpact())	1.28
IterativeDiving	selectSmallest(ImpactOfLastBranch())	selectSmallest(valueImpact())	1.287
IterativeDiving	selectSmallest(localImpact())	selectSmallest(value())	1.287
IterativeDiving	selectSmallest(regretOnMax())	selectRandomValue()	1.289
IterativeDiving	selectSmallest(regretOnMin())	selectLargest(valueImpact())	1.29
IterativeDiving	selectSmallest(domainMin())	selectSmallest(valueImpact())	1.293
IterativeDiving	selectSmallest(ImpactOfLastBranch())	selectLargest(valueSuccessRate())	1.293

IterativeDiving	selectSmallest(domainMax())	selectLargest(valueImpact())	1.294
IterativeDiving	selectSmallest(domainSize())	selectSmallest(valueImpact())	1.294
IterativeDiving	selectSmallest(successRate())	selectSmallest(value())	1.297
IterativeDiving	selectSmallest(impact())	selectLargest(valueImpact())	1.299
IterativeDiving	selectSmallest(successRate())	selectSmallest(valueImpact())	1.302
IterativeDiving	selectSmallest(regretOnMax())	selectLargest(value())	1.303
IterativeDiving	selectSmallest(successRate())	selectSmallest(valueSuccessRate())	1.308
IterativeDiving	selectSmallest(localImpact())	selectRandomValue()	1.311
IterativeDiving	selectLargest(domainMax())	selectRandomValue()	1.311
IterativeDiving	selectLargest(domainMin())	selectRandomValue()	1.311
IterativeDiving	selectLargest(domainMin())	selectLargest(value())	1.312
IterativeDiving	selectSmallest(localImpact())	selectLargest(valueImpact())	1.315
IterativeDiving	selectLargest(domainMin())	selectSmallest(valueSuccessRate())	1.315
IterativeDiving	selectLargest(regretOnMax())	selectSmallest(valueImpact())	1.316
IterativeDiving	selectLargest(regretOnMax())	selectRandomValue()	1.316
IterativeDiving	selectSmallest(localImpact())	selectSmallest(valueSuccessRate())	1.32
IterativeDiving	selectLargest(domainMin())	selectSmallest(value())	1.323
IterativeDiving	selectLargest(domainMin())	selectSmallest(valueImpact())	1.323
IterativeDiving	selectLargest(regretOnMin())	selectLargest(value())	1.327
IterativeDiving	selectLargest(regretOnMin())	selectLargest(valueImpact())	1.328
IterativeDiving	selectLargest(regretOnMin())	selectSmallest(valueSuccessRate())	1.328
IterativeDiving	selectSmallest(successRate())	selectLargest(valueImpact())	1.329
IterativeDiving	selectSmallest(impact())	selectRandomValue()	1.329
IterativeDiving	selectSmallest(domainSize())	selectLargest(valueSuccessRate())	1.33
IterativeDiving	selectLargest(regretOnMax())	selectSmallest(value())	1.333
IterativeDiving	selectLargest(regretOnMin())	selectSmallest(value())	1.333
IterativeDiving	selectSmallest(regretOnMin())	selectSmallest(valueImpact())	1.334
IterativeDiving	selectSmallest(ImpactOfLastBranch())	selectRandomValue()	1.336
IterativeDiving	selectSmallest(impact())	selectSmallest(value())	1.337
IterativeDiving	selectLargest(domainMax())	selectLargest(value())	1.339
IterativeDiving	selectLargest(domainMax())	selectLargest(valueSuccessRate())	1.339
IterativeDiving	selectSmallest(domainSize())	selectLargest(value())	1.343
IterativeDiving	selectLargest(domainMax())	selectSmallest(valueImpact())	1.343
IterativeDiving	selectLargest(domainMin())	selectLargest(valueSuccessRate())	1.343
IterativeDiving	selectSmallest(localImpact())	selectLargest(value())	1.343
IterativeDiving	selectLargest(impact())	selectRandomValue()	1.345
IterativeDiving	selectLargest(domainMin())	selectLargest(valueImpact())	1.346
IterativeDiving	selectRandomVar()	selectSmallest(valueSuccessRate())	1.347
IterativeDiving	selectLargest(regretOnMin())	selectRandomValue()	1.347
IterativeDiving	selectLargest(domainMax())	selectSmallest(valueSuccessRate())	1.348
IterativeDiving	selectLargest(ImpactOfLastBranch())	selectSmallest(value())	1.348
IterativeDiving	selectLargest(ImpactOfLastBranch())	selectSmallest(valueSuccessRate())	1.349
IterativeDiving	selectLargest(ImpactOfLastBranch())	selectLargest(valueSuccessRate())	1.352
IterativeDiving	selectLargest(regretOnMax())	selectSmallest(valueSuccessRate())	1.352
IterativeDiving	selectLargest(localImpact())	selectLargest(valueImpact())	1.355

IterativeDiving	selectLargest(impact())	selectLargest(valueSuccessRate())	1.355
IterativeDiving	selectSmallest(domainSize())	selectRandomValue()	1.355
IterativeDiving	selectLargest(successRate())	selectLargest(value())	1.361
IterativeDiving	selectSmallest(ImpactOfLastBranch())	selectLargest(valueImpact())	1.361
IterativeDiving	selectLargest(impact())	selectSmallest(value())	1.362
IterativeDiving	selectSmallest(domainMax())	selectSmallest(valueImpact())	1.362
IterativeDiving	selectLargest(regretOnMax())	selectLargest(valueImpact())	1.363
IterativeDiving	selectLargest(successRate())	selectSmallest(valueSuccessRate())	1.365
IterativeDiving	selectSmallest(successRate())	selectLargest(valueSuccessRate())	1.367
IterativeDiving	selectLargest(localImpact())	selectLargest(valueSuccessRate())	1.37
IterativeDiving	selectLargest(localImpact())	selectLargest(value())	1.371
IterativeDiving	selectSmallest(localImpact())	selectLargest(valueSuccessRate())	1.371
IterativeDiving	selectLargest(regretOnMax())	selectLargest(valueSuccessRate())	1.375
IterativeDiving	selectRandomVar()	selectRandomValue()	1.376
IterativeDiving	selectLargest(successRate())	selectSmallest(value())	1.378
IterativeDiving	selectRandomVar()	selectSmallest(valueImpact())	1.378
IterativeDiving	selectLargest(ImpactOfLastBranch())	selectSmallest(valueImpact())	1.378
IterativeDiving	selectSmallest(domainSize())	selectLargest(valueImpact())	1.383
IterativeDiving	selectLargest(successRate())	selectSmallest(valueImpact())	1.384
IterativeDiving	selectLargest(localImpact())	selectRandomValue()	1.386
IterativeDiving	selectLargest(localImpact())	selectSmallest(valueImpact())	1.386
IterativeDiving	selectLargest(regretOnMin())	selectSmallest(valueImpact())	1.387
IterativeDiving	selectRandomVar()	selectSmallest(value())	1.396
IterativeDiving	selectLargest(impact())	selectSmallest(valueSuccessRate())	1.399
IterativeDiving	selectRandomVar()	selectLargest(valueSuccessRate())	1.4
IterativeDiving	selectLargest(successRate())	selectRandomValue()	1.408
IterativeDiving	selectRandomVar()	selectLargest(value())	1.408
IterativeDiving	selectSmallest(ImpactOfLastBranch())	selectSmallest(valueSuccessRate())	1.416
IterativeDiving	selectLargest(localImpact())	selectSmallest(value())	1.419
IterativeDiving	selectLargest(successRate())	selectLargest(valueImpact())	1.421
IterativeDiving	selectLargest(ImpactOfLastBranch())	selectLargest(value())	1.423
IterativeDiving	selectLargest(domainMax())	selectSmallest(value())	1.425
IterativeDiving	selectLargest(impact())	selectLargest(valueImpact())	1.426
IterativeDiving	selectRandomVar()	selectLargest(valueImpact())	1.427
IterativeDiving	selectLargest(ImpactOfLastBranch())	selectRandomValue()	1.43
IterativeDiving	selectLargest(domainMax())	selectLargest(valueImpact())	1.431
IterativeDiving	selectLargest(localImpact())	selectSmallest(valueSuccessRate())	1.438
IterativeDiving	selectLargest(impact())	selectSmallest(valueImpact())	1.44
IterativeDiving	selectLargest(regretOnMax())	selectLargest(value())	1.446
IterativeDiving	selectLargest(impact())	selectLargest(value())	1.446
IterativeDiving	selectLargest(ImpactOfLastBranch())	selectLargest(valueImpact())	1.451
IterativeDiving	selectLargest(regretOnMin())	selectLargest(valueSuccessRate())	1.465
IterativeDiving	selectLargest(successRate())	selectLargest(valueSuccessRate())	1.493
Restart	selectSmallest(impact())	selectLargest(valueImpact())	1.617
Restart	selectSmallest(domainSize())	selectLargest(value())	1.617

Restart	selectSmallest(regretOnMin())	selectLargest(valueSuccessRate())	1.618
Restart	selectLargest(successRate())	selectLargest(valueSuccessRate())	1.618
Restart	selectRandomVar()	selectLargest(value())	1.625
Restart	selectSmallest(regretOnMax())	selectLargest(value())	1.627
Restart	selectLargest(localImpact())	selectLargest(value())	1.628
Restart	selectSmallest(localImpact())	selectSmallest(valueImpact())	1.631
Restart	selectSmallest(domainMax())	selectSmallest(valueSuccessRate())	1.633
Restart	selectLargest(successRate())	selectLargest(value())	1.633
Restart	selectLargest(domainMin())	selectLargest(valueSuccessRate())	1.635
Restart	selectLargest(regretOnMax())	selectSmallest(valueSuccessRate())	1.635
Restart	selectSmallest(domainMin())	selectLargest(valueSuccessRate())	1.637
Restart	selectSmallest(impact())	selectRandomValue()	1.638
Restart	selectLargest(regretOnMin())	selectLargest(valueSuccessRate())	1.639
Restart	selectLargest(regretOnMax())	selectLargest(value())	1.641
Restart	selectSmallest(domainSize())	selectSmallest(valueImpact())	1.641
Restart	selectLargest(localImpact())	selectSmallest(value())	1.643
Restart	selectLargest(impact())	selectLargest(valueSuccessRate())	1.644
Restart	selectLargest(localImpact())	selectLargest(valueSuccessRate())	1.645
Restart	selectLargest(domainMax())	selectSmallest(valueSuccessRate())	1.646
Restart	selectSmallest(domainMax())	selectLargest(valueImpact())	1.648
Restart	selectSmallest(successRate())	selectSmallest(valueSuccessRate())	1.649
Restart	selectLargest(impact())	selectSmallest(value())	1.65
Restart	selectSmallest(domainMin())	selectSmallest(value())	1.651
Restart	selectLargest(domainMin())	selectSmallest(value())	1.651
Restart	selectSmallest(regretOnMin())	selectSmallest(valueSuccessRate())	1.651
Restart	selectLargest(impact())	selectRandomValue()	1.651
Restart	selectLargest(regretOnMin())	selectLargest(valueImpact())	1.654
Restart	selectSmallest(successRate())	selectRandomValue()	1.655
Restart	selectSmallest(regretOnMin())	selectSmallest(value())	1.658
Restart	selectRandomVar()	selectSmallest(valueImpact())	1.659
Restart	selectLargest(successRate())	selectSmallest(value())	1.659
Restart	selectLargest(domainMin())	selectSmallest(valueSuccessRate())	1.66
Restart	selectSmallest(localImpact())	selectSmallest(valueSuccessRate())	1.662
Restart	selectSmallest(impact())	selectLargest(value())	1.664
Restart	selectLargest(impact())	selectSmallest(valueImpact())	1.664
Restart	selectSmallest(impact())	selectSmallest(value())	1.668
Restart	selectLargest(impact())	selectLargest(value())	1.668
Restart	selectSmallest(domainMax())	selectSmallest(valueImpact())	1.669
Restart	selectLargest(domainMax())	selectSmallest(valueImpact())	1.671
Restart	selectLargest(regretOnMax())	selectLargest(valueImpact())	1.672
Restart	selectLargest(regretOnMin())	selectSmallest(valueImpact())	1.674
Restart	selectLargest(regretOnMax())	selectRandomValue()	1.675
Restart	selectSmallest(ImpactOfLastBranch())	selectLargest(value())	1.675
Restart	selectSmallest(impact())	selectSmallest(valueSuccessRate())	1.675
Restart	selectLargest(domainMax())	selectSmallest(value())	1.676

Restart	selectLargest(domainMin())	selectLargest(value())	1.676
Restart	selectSmallest(domainSize())	selectLargest(valueSuccessRate())	1.677
Restart	selectRandomVar()	selectRandomValue()	1.679
Restart	selectLargest(regretOnMin())	selectSmallest(value())	1.681
Restart	selectSmallest(regretOnMin())	selectLargest(valueImpact())	1.684
Restart	selectSmallest(domainSize())	selectSmallest(value())	1.685
Restart	selectSmallest(localImpact())	selectLargest(valueImpact())	1.686
Restart	selectSmallest(ImpactOfLastBranch())	selectRandomValue()	1.687
Restart	selectSmallest(domainSize())	selectSmallest(valueSuccessRate())	1.689
Restart	selectSmallest(impact())	selectLargest(valueSuccessRate())	1.689
Restart	selectLargest(domainMax())	selectLargest(valueSuccessRate())	1.69
Restart	selectSmallest(ImpactOfLastBranch())	selectSmallest(valueImpact())	1.691
Restart	selectRandomVar()	selectLargest(valueImpact())	1.691
Restart	selectSmallest(domainMin())	selectSmallest(valueSuccessRate())	1.692
Restart	selectLargest(ImpactOfLastBranch())	selectSmallest(value())	1.693
Restart	selectRandomVar()	selectSmallest(value())	1.694
Restart	selectSmallest(ImpactOfLastBranch())	selectSmallest(valueSuccessRate())	1.695
Restart	selectLargest(domainMax())	selectRandomValue()	1.696
Restart	selectLargest(regretOnMax())	selectSmallest(value())	1.696
Restart	selectSmallest(domainMin())	selectLargest(valueImpact())	1.696
Restart	selectSmallest(domainMax())	selectLargest(valueSuccessRate())	1.697
Restart	selectLargest(regretOnMin())	selectLargest(value())	1.698
Restart	selectLargest(ImpactOfLastBranch())	selectSmallest(valueImpact())	1.699
Restart	selectSmallest(ImpactOfLastBranch())	selectLargest(valueImpact())	1.701
Restart	selectSmallest(regretOnMax())	selectLargest(valueSuccessRate())	1.701
Restart	selectSmallest(localImpact())	selectLargest(value())	1.702
Restart	selectLargest(domainMax())	selectLargest(value())	1.703
Restart	selectLargest(ImpactOfLastBranch())	selectLargest(valueSuccessRate())	1.705
Restart	selectLargest(localImpact())	selectSmallest(valueImpact())	1.707
Restart	selectLargest(ImpactOfLastBranch())	selectLargest(valueImpact())	1.707
Restart	selectLargest(ImpactOfLastBranch())	selectLargest(value())	1.708
Restart	selectRandomVar()	selectSmallest(valueSuccessRate())	1.709
Restart	selectLargest(impact())	selectSmallest(valueSuccessRate())	1.711
Restart	selectLargest(localImpact())	selectSmallest(valueSuccessRate())	1.713
Restart	selectSmallest(impact())	selectSmallest(valueImpact())	1.716
Restart	selectLargest(domainMax())	selectLargest(valueImpact())	1.718
Restart	selectLargest(regretOnMax())	selectLargest(valueSuccessRate())	1.718
Restart	selectSmallest(domainSize())	selectRandomValue()	1.721
Restart	selectRandomVar()	selectLargest(valueSuccessRate())	1.721
Restart	selectSmallest(successRate())	selectSmallest(value())	1.723
Restart	selectLargest(successRate())	selectLargest(valueImpact())	1.725
Restart	selectSmallest(domainMin())	selectSmallest(valueImpact())	1.725
Restart	selectLargest(successRate())	selectSmallest(valueSuccessRate())	1.726
Restart	selectSmallest(regretOnMax())	selectSmallest(valueImpact())	1.726
Restart	selectLargest(regretOnMin())	selectSmallest(valueSuccessRate())	1.729

Restart	selectLargest(domainMin())	selectLargest(valueImpact())	1.735
Restart	selectLargest(domainMin())	selectSmallest(valueImpact())	1.736
Restart	selectLargest(localImpact())	selectLargest(valueImpact())	1.739
Restart	selectLargest(domainMin())	selectRandomValue()	1.739
Restart	selectLargest(ImpactOfLastBranch())	selectRandomValue()	1.74
Restart	selectSmallest(localImpact())	selectLargest(valueSuccessRate())	1.741
Restart	selectLargest(impact())	selectLargest(valueImpact())	1.742
Restart	selectLargest(localImpact())	selectRandomValue()	1.742
Restart	selectLargest(ImpactOfLastBranch())	selectSmallest(valueSuccessRate())	1.744
Restart	selectLargest(successRate())	selectRandomValue()	1.748
Restart	selectSmallest(domainMax())	selectSmallest(value())	1.748
Restart	selectSmallest(regretOnMax())	selectSmallest(value())	1.749
Restart	selectSmallest(domainSize())	selectLargest(valueImpact())	1.75
Restart	selectLargest(regretOnMax())	selectSmallest(valueImpact())	1.752
Restart	selectLargest(successRate())	selectSmallest(valueImpact())	1.754
Restart	selectSmallest(domainMax())	selectLargest(value())	1.758
Restart	selectSmallest(successRate())	selectLargest(valueSuccessRate())	1.766
Restart	selectSmallest(localImpact())	selectSmallest(value())	1.768
Restart	selectSmallest(domainMax())	selectRandomValue()	1.771
Restart	selectSmallest(regretOnMin())	selectRandomValue()	1.774
Restart	selectSmallest(domainMin())	selectLargest(value())	1.777
Restart	selectLargest(regretOnMin())	selectRandomValue()	1.782
Restart	selectSmallest(regretOnMin())	selectSmallest(valueImpact())	1.786
Restart	selectSmallest(successRate())	selectSmallest(valueImpact())	1.79
Restart	selectSmallest(regretOnMax())	selectRandomValue()	1.796
Restart	selectSmallest(regretOnMin())	selectLargest(value())	1.797
Restart	selectSmallest(localImpact())	selectRandomValue()	1.813
Restart	selectSmallest(domainMin())	selectRandomValue()	1.817
Restart	selectSmallest(regretOnMax())	selectLargest(valueImpact())	1.818
Restart	selectSmallest(ImpactOfLastBranch())	selectSmallest(value())	1.822
Restart	selectSmallest(ImpactOfLastBranch())	selectLargest(valueSuccessRate())	1.828
Restart	selectSmallest(successRate())	selectLargest(valueImpact())	1.835
Restart	selectSmallest(successRate())	selectLargest(value())	1.84
Restart	selectSmallest(regretOnMax())	selectSmallest(valueSuccessRate())	1.87

B.1.2 Iterative Diving (C=1)

Table 14: All iterations of capacity 1 executed with Iterative Diving method

	Largest Value	**Largest Value Impact**	**Largest Value Success Rate**	**Random Value**	**Smallest Value**	**Smallest Value Impact**	**Smallest Value Success Rate**
Largest Impact Of Last Branch	1.423	1.451	1.352	1.43	1.348	1.378	1.349
Largest Domain Max	1.339	1.431	1.339	1.311	1.425	1.343	1.348
Largest Domain Min	1.312	1.346	1.343	1.311	1.323	1.323	1.315
Largest Impact	1.446	1.426	1.355	1.345	1.362	1.44	1.399
Largest Local Impact	1.371	1.355	1.37	1.386	1.419	1.386	1.438
Largest Regret On Max	1.446	1.363	1.375	1.316	1.333	1.316	1.352
Largest Regret On Min	1.327	1.328	1.465	1.347	1.333	1.387	1.328
Largest Success Rate	1.361	1.421	1.493	1.408	1.378	1.384	1.365
Random Var	1.408	1.427	1.4	1.376	1.396	1.378	1.347
Smallest Impact Of Last Branch	1.279	1.361	1.293	1.336	1.278	1.287	1.416
Smallest Domain Max	1.248	1.294	1.243	1.249	1.23	1.362	1.221
Smallest Domain Min	1.23	1.231	1.221	1.212	1.211	1.293	1.193
Smallest Domain Size	1.343	1.383	1.33	1.355	1.27	1.294	1.266
Smallest Impact	1.27	1.299	1.254	1.329	1.337	1.28	1.248
Smallest Local Impact	1.343	1.315	1.371	1.311	1.287	1.265	1.32
Smallest Regret On Max	1.303	1.249	1.253	1.289	1.267	1.28	1.26
Smallest Regret On Min	1.212	1.29	1.254	1.198	1.192	1.334	1.222
Smallest Success Rate	1.24	1.329	1.367	1.239	1.297	1.302	1.308

B.1.3 Restart (C=1)

Table 15: All iterations of capacity 1 executed with Restart method

	Largest Value	Largest Value Impact	Largest Value Success Rate	Random Value	Smallest Value	Smallest Value Impact	Smallest Value Success Rate
Largest Impact Of Last Branch	1.708	1.707	1.705	1.74	1.693	1.699	1.744
Largest Domain Max	1.703	1.718	1.69	1.696	1.676	1.671	1.646
Largest Domain Min	1.676	1.735	1.635	1.739	1.651	1.736	1.66
Largest Impact	1.668	1.742	1.644	1.651	1.65	1.664	1.711
Largest Local Impact	1.628	1.739	1.645	1.742	1.643	1.707	1.713
Largest Regret On Max	1.641	1.672	1.718	1.675	1.696	1.752	1.635
Largest Regret On Min	1.698	1.654	1.639	1.782	1.681	1.674	1.729
Largest Success Rate	1.633	1.725	1.618	1.748	1.659	1.754	1.726
Random Var	1.625	1.691	1.721	1.679	1.694	1.659	1.709
Smallest Impact Of Last Branch	1.675	1.701	1.828	1.687	1.822	1.691	1.695
Smallest Domain Max	1.758	1.648	1.697	1.771	1.748	1.669	1.633
Smallest Domain Min	1.777	1.696	1.637	1.817	1.651	1.725	1.692
Smallest Domain Size	1.617	1.75	1.677	1.721	1.685	1.641	1.689
Smallest Impact	1.664	1.617	1.689	1.638	1.668	1.716	1.675
Smallest Local Impact	1.702	1.686	1.741	1.813	1.768	1.631	1.662
Smallest Regret On Max	1.627	1.818	1.701	1.796	1.749	1.726	1.87
Smallest Regret On Min	1.797	1.684	1.618	1.774	1.658	1.786	1.651
Smallest Success Rate	1.84	1.835	1.766	1.655	1.723	1.79	1.649

B.1.4 Capacity = 2

Table 16: All hyperparameter combinations for capacity two

Method	Variable Selection	Value Selection	Time (s)
Restart	selectSmallest(domainMin())	selectSmallest(value())	20.735
Restart	selectSmallest(domainMin())	selectSmallest(valueSuccessRate())	20.863
Restart	selectSmallest(domainMin())	selectLargest(value())	20.89
Restart	selectSmallest(domainMin())	selectLargest(valueImpact())	20.898
Restart	selectSmallest(domainMin())	selectSmallest(valueImpact())	20.898
Restart	selectSmallest(regretOnMin())	selectLargest(value())	20.931
Restart	selectSmallest(domainMax())	selectLargest(valueSuccessRate())	20.941
Restart	selectSmallest(domainMin())	selectLargest(valueSuccessRate())	20.956
Restart	selectSmallest(regretOnMin())	selectSmallest(valueImpact())	20.968
Restart	selectSmallest(regretOnMin())	selectLargest(valueImpact())	20.995
Restart	selectSmallest(domainMax())	selectLargest(valueImpact())	20.997
Restart	selectSmallest(domainMax())	selectLargest(value())	21.024
Restart	selectSmallest(regretOnMin())	selectSmallest(valueSuccessRate())	21.025
Restart	selectSmallest(regretOnMin())	selectSmallest(value())	21.028
Restart	selectSmallest(domainMax())	selectSmallest(valueImpact())	21.058
Restart	selectSmallest(regretOnMax())	selectSmallest(valueSuccessRate())	21.08
Restart	selectSmallest(domainMax())	selectSmallest(valueSuccessRate())	21.084
Restart	selectSmallest(regretOnMin())	selectLargest(valueSuccessRate())	21.088
Restart	selectSmallest(regretOnMax())	selectLargest(value())	21.132
Restart	selectSmallest(regretOnMax())	selectSmallest(valueImpact())	21.135
Restart	selectSmallest(regretOnMax())	selectLargest(valueSuccessRate())	21.137
Restart	selectSmallest(regretOnMax())	selectSmallest(value())	21.152
Restart	selectSmallest(domainMax())	selectSmallest(value())	21.153
Restart	selectSmallest(successRate())	selectSmallest(valueSuccessRate())	21.187
Restart	selectSmallest(successRate())	selectSmallest(value())	21.216
Restart	selectSmallest(regretOnMax())	selectLargest(valueImpact())	21.252
Restart	selectSmallest(successRate())	selectSmallest(valueImpact())	21.278
Restart	selectSmallest(successRate())	selectLargest(valueSuccessRate())	21.332
Restart	selectSmallest(impact())	selectSmallest(valueImpact())	21.359
Restart	selectSmallest(successRate())	selectLargest(value())	21.376
Restart	selectSmallest(successRate())	selectLargest(valueImpact())	21.423
Restart	selectSmallest(ImpactOfLastBranch())	selectSmallest(valueSuccessRate())	21.427
Restart	selectSmallest(impact())	selectLargest(valueImpact())	21.433
Restart	selectSmallest(impact())	selectLargest(value())	21.456
Restart	selectSmallest(impact())	selectSmallest(valueSuccessRate())	21.459
Restart	selectSmallest(impact())	selectSmallest(value())	21.466
Restart	selectSmallest(localImpact())	selectSmallest(valueSuccessRate())	21.536
Restart	selectSmallest(impact())	selectLargest(valueSuccessRate())	21.539
Restart	selectSmallest(ImpactOfLastBranch())	selectSmallest(value())	21.556
Restart	selectSmallest(ImpactOfLastBranch())	selectLargest(value())	21.566
Restart	selectSmallest(localImpact())	selectSmallest(value())	21.571

Restart	selectSmallest(ImpactOfLastBranch())	selectSmallest(valueImpact())	21.586
Restart	selectSmallest(localImpact())	selectSmallest(valueImpact())	21.592
Restart	selectSmallest(localImpact())	selectLargest(valueSuccessRate())	21.647
Restart	selectSmallest(localImpact())	selectLargest(valueImpact())	21.652
Restart	selectSmallest(localImpact())	selectLargest(value())	21.78
Restart	selectSmallest(ImpactOfLastBranch())	selectLargest(valueSuccessRate())	21.792
Restart	selectSmallest(domainSize())	selectSmallest(value())	21.809
Restart	selectSmallest(ImpactOfLastBranch())	selectLargest(valueImpact())	21.827
Restart	selectSmallest(domainSize())	selectLargest(valueSuccessRate())	21.874
Restart	selectSmallest(domainSize())	selectSmallest(valueImpact())	21.892
Restart	selectSmallest(domainSize())	selectLargest(value())	21.901
Restart	selectSmallest(domainSize())	selectSmallest(valueSuccessRate())	21.909
IterativeDiving	selectSmallest(domainMin())	selectSmallest(value())	21.921
Restart	selectLargest(domainMin())	selectSmallest(value())	21.923
Restart	selectSmallest(domainSize())	selectLargest(valueImpact())	22.009
Restart	selectLargest(domainMin())	selectSmallest(valueSuccessRate())	22.044
Restart	selectLargest(domainMin())	selectLargest(valueImpact())	22.093
IterativeDiving	selectSmallest(domainMax())	selectSmallest(valueSuccessRate())	22.097
IterativeDiving	selectSmallest(domainMin())	selectLargest(value())	22.1
Restart	selectLargest(domainMin())	selectLargest(valueSuccessRate())	22.101
Restart	selectLargest(domainMin())	selectLargest(value())	22.117
IterativeDiving	selectSmallest(domainMax())	selectSmallest(value())	22.132
IterativeDiving	selectSmallest(regretOnMin())	selectRandomValue()	22.219
IterativeDiving	selectSmallest(domainMax())	selectLargest(value())	22.232
IterativeDiving	selectSmallest(domainMin())	selectLargest(valueImpact())	22.244
IterativeDiving	selectSmallest(domainMax())	selectLargest(valueImpact())	22.255
IterativeDiving	selectSmallest(regretOnMin())	selectLargest(valueSuccessRate())	22.264
IterativeDiving	selectSmallest(regretOnMin())	selectSmallest(valueSuccessRate())	22.307
IterativeDiving	selectSmallest(regretOnMax())	selectLargest(value())	22.319
Restart	selectLargest(domainMin())	selectSmallest(valueImpact())	22.32
Restart	selectLargest(domainMax())	selectLargest(value())	22.321
Restart	selectLargest(domainMax())	selectSmallest(valueSuccessRate())	22.377
Restart	selectLargest(domainMax())	selectSmallest(value())	22.377
IterativeDiving	selectSmallest(regretOnMin())	selectLargest(value())	22.377
IterativeDiving	selectSmallest(domainMin())	selectSmallest(valueSuccessRate())	22.381
IterativeDiving	selectSmallest(domainMax())	selectSmallest(valueImpact())	22.387
IterativeDiving	selectSmallest(regretOnMax())	selectSmallest(valueImpact())	22.401
IterativeDiving	selectSmallest(regretOnMax())	selectSmallest(value())	22.405
IterativeDiving	selectSmallest(domainMax())	selectLargest(valueSuccessRate())	22.412
IterativeDiving	selectSmallest(successRate())	selectSmallest(value())	22.414
IterativeDiving	selectSmallest(regretOnMax())	selectSmallest(valueSuccessRate())	22.42
IterativeDiving	selectSmallest(regretOnMax())	selectRandomValue()	22.428
Restart	selectLargest(domainMax())	selectLargest(valueSuccessRate())	22.454
Restart	selectLargest(domainMax())	selectLargest(valueImpact())	22.458
Restart	selectLargest(regretOnMin())	selectSmallest(valueSuccessRate())	22.464

IterativeDiving	selectSmallest(domainMax())	selectRandomValue()	22.486
IterativeDiving	selectSmallest(successRate())	selectLargest(value())	22.49
IterativeDiving	selectSmallest(regretOnMax())	selectLargest(valueImpact())	22.494
IterativeDiving	selectSmallest(regretOnMin())	selectLargest(valueImpact())	22.494
Restart	selectLargest(domainMax())	selectSmallest(valueImpact())	22.504
Restart	selectLargest(regretOnMin())	selectLargest(valueSuccessRate())	22.507
Restart	selectLargest(regretOnMin())	selectSmallest(value())	22.507
IterativeDiving	selectSmallest(regretOnMin())	selectSmallest(value())	22.515
Restart	selectLargest(regretOnMin())	selectSmallest(valueImpact())	22.525
IterativeDiving	selectSmallest(successRate())	selectRandomValue()	22.531
IterativeDiving	selectSmallest(successRate())	selectLargest(valueImpact())	22.532
Restart	selectLargest(regretOnMin())	selectLargest(value())	22.542
IterativeDiving	selectSmallest(regretOnMin())	selectSmallest(valueImpact())	22.585
IterativeDiving	selectSmallest(regretOnMax())	selectLargest(valueSuccessRate())	22.593
IterativeDiving	selectSmallest(successRate())	selectLargest(valueSuccessRate())	22.595
IterativeDiving	selectSmallest(successRate())	selectSmallest(valueImpact())	22.618
Restart	selectLargest(regretOnMin())	selectLargest(valueImpact())	22.639
IterativeDiving	selectSmallest(impact())	selectSmallest(valueImpact())	22.66
IterativeDiving	selectSmallest(domainMin())	selectRandomValue()	22.661
IterativeDiving	selectSmallest(successRate())	selectSmallest(valueSuccessRate())	22.681
IterativeDiving	selectSmallest(localImpact())	selectSmallest(valueImpact())	22.704
Restart	selectLargest(regretOnMax())	selectSmallest(valueImpact())	22.716
IterativeDiving	selectSmallest(impact())	selectLargest(value())	22.721
Restart	selectLargest(regretOnMax())	selectSmallest(valueSuccessRate())	22.722
Restart	selectLargest(regretOnMax())	selectSmallest(value())	22.723
IterativeDiving	selectSmallest(domainMin())	selectLargest(valueSuccessRate())	22.773
IterativeDiving	selectSmallest(impact())	selectLargest(valueSuccessRate())	22.775
IterativeDiving	selectSmallest(impact())	selectSmallest(valueSuccessRate())	22.793
IterativeDiving	selectSmallest(impact())	selectLargest(valueImpact())	22.825
IterativeDiving	selectSmallest(localImpact())	selectLargest(value())	22.832
IterativeDiving	selectSmallest(impact())	selectRandomValue()	22.856
IterativeDiving	selectSmallest(localImpact())	selectSmallest(valueSuccessRate())	22.861
IterativeDiving	selectSmallest(ImpactOfLastBranch())	selectSmallest(valueImpact())	22.884
IterativeDiving	selectSmallest(domainMin())	selectSmallest(valueImpact())	22.901
IterativeDiving	selectSmallest(localImpact())	selectSmallest(value())	22.903
IterativeDiving	selectSmallest(localImpact())	selectLargest(valueImpact())	22.906
IterativeDiving	selectSmallest(impact())	selectSmallest(value())	22.954
IterativeDiving	selectSmallest(ImpactOfLastBranch())	selectSmallest(value())	22.968
IterativeDiving	selectSmallest(ImpactOfLastBranch())	selectLargest(valueImpact())	23.008
IterativeDiving	selectSmallest(domainSize())	selectSmallest(valueSuccessRate())	23.061
IterativeDiving	selectSmallest(localImpact())	selectLargest(valueSuccessRate())	23.088
IterativeDiving	selectSmallest(ImpactOfLastBranch())	selectLargest(value())	23.13
Restart	selectLargest(regretOnMax())	selectLargest(value())	23.156
IterativeDiving	selectSmallest(ImpactOfLastBranch())	selectSmallest(valueSuccessRate())	23.158
IterativeDiving	selectSmallest(domainSize())	selectSmallest(valueImpact())	23.244

IterativeDiving	selectSmallest(domainSize())	selectLargest(value())	23.284
IterativeDiving	selectSmallest(domainSize())	selectLargest(valueImpact())	23.326
IterativeDiving	selectSmallest(ImpactOfLastBranch())	selectLargest(valueSuccessRate())	23.329
IterativeDiving	selectSmallest(localImpact())	selectRandomValue()	23.35
Restart	selectLargest(successRate())	selectSmallest(value())	23.38
IterativeDiving	selectSmallest(domainSize())	selectSmallest(value())	23.392
IterativeDiving	selectLargest(domainMin())	selectSmallest(value())	23.483
Restart	selectLargest(regretOnMax())	selectLargest(valueSuccessRate())	23.494
Restart	selectLargest(regretOnMax())	selectLargest(valueImpact())	23.496
IterativeDiving	selectLargest(domainMin())	selectSmallest(valueImpact())	23.507
Restart	selectLargest(successRate())	selectSmallest(valueImpact())	23.532
IterativeDiving	selectSmallest(domainSize())	selectLargest(valueSuccessRate())	23.547
Restart	selectLargest(successRate())	selectLargest(value())	23.549
Restart	selectLargest(impact())	selectSmallest(value())	23.598
IterativeDiving	selectSmallest(ImpactOfLastBranch())	selectRandomValue()	23.611
Restart	selectLargest(impact())	selectSmallest(valueImpact())	23.613
Restart	selectLargest(successRate())	selectLargest(valueImpact())	23.622
Restart	selectLargest(impact())	selectLargest(valueSuccessRate())	23.665
Restart	selectLargest(successRate())	selectLargest(valueSuccessRate())	23.673
Restart	selectLargest(impact())	selectLargest(value())	23.682
IterativeDiving	selectLargest(domainMin())	selectSmallest(valueSuccessRate())	23.683
IterativeDiving	selectLargest(domainMin())	selectLargest(valueSuccessRate())	23.686
Restart	selectLargest(successRate())	selectSmallest(valueSuccessRate())	23.723
IterativeDiving	selectLargest(domainMax())	selectSmallest(value())	23.739
Restart	selectLargest(localImpact())	selectSmallest(value())	23.759
IterativeDiving	selectLargest(regretOnMin())	selectSmallest(value())	23.791
Restart	selectLargest(impact())	selectLargest(valueImpact())	23.798
Restart	selectLargest(impact())	selectSmallest(valueSuccessRate())	23.81
Restart	selectLargest(ImpactOfLastBranch())	selectSmallest(value())	23.846
IterativeDiving	selectLargest(domainMin())	selectLargest(valueImpact())	23.847
Restart	selectLargest(localImpact())	selectSmallest(valueSuccessRate())	23.857
IterativeDiving	selectLargest(domainMax())	selectLargest(value())	23.887
Restart	selectLargest(localImpact())	selectLargest(value())	23.9
IterativeDiving	selectLargest(domainMax())	selectSmallest(valueSuccessRate())	23.91
IterativeDiving	selectSmallest(domainSize())	selectRandomValue()	23.942
IterativeDiving	selectLargest(domainMin())	selectLargest(value())	23.943
Restart	selectLargest(localImpact())	selectSmallest(valueImpact())	23.943
IterativeDiving	selectLargest(domainMax())	selectSmallest(valueImpact())	23.947
Restart	selectLargest(localImpact())	selectLargest(valueImpact())	23.998
Restart	selectLargest(ImpactOfLastBranch())	selectLargest(value())	24.007
Restart	selectLargest(ImpactOfLastBranch())	selectSmallest(valueSuccessRate())	24.058
IterativeDiving	selectLargest(regretOnMin())	selectLargest(value())	24.065
IterativeDiving	selectLargest(regretOnMin())	selectSmallest(valueSuccessRate())	24.081
Restart	selectLargest(ImpactOfLastBranch())	selectLargest(valueImpact())	24.089
Restart	selectLargest(localImpact())	selectLargest(valueSuccessRate())	24.108

Restart	selectLargest(ImpactOfLastBranch())	selectSmallest(valueImpact())	24.129
IterativeDiving	selectLargest(domainMax())	selectLargest(valueImpact())	24.133
IterativeDiving	selectLargest(domainMax())	selectLargest(valueSuccessRate())	24.142
Restart	selectLargest(ImpactOfLastBranch())	selectLargest(valueSuccessRate())	24.186
IterativeDiving	selectLargest(domainMin())	selectRandomValue()	24.404
IterativeDiving	selectLargest(regretOnMin())	selectSmallest(valueImpact())	24.492
IterativeDiving	selectLargest(domainMax())	selectRandomValue()	24.775
IterativeDiving	selectLargest(regretOnMin())	selectLargest(valueImpact())	24.861
IterativeDiving	selectLargest(regretOnMax())	selectLargest(valueImpact())	24.873
IterativeDiving	selectLargest(regretOnMin())	selectLargest(valueSuccessRate())	24.954
IterativeDiving	selectLargest(regretOnMax())	selectLargest(value())	25.012
IterativeDiving	selectLargest(successRate())	selectSmallest(value())	25.046
IterativeDiving	selectLargest(regretOnMax())	selectLargest(valueSuccessRate())	25.049
IterativeDiving	selectLargest(regretOnMax())	selectSmallest(valueImpact())	25.22
IterativeDiving	selectLargest(regretOnMax())	selectSmallest(value())	25.429
IterativeDiving	selectLargest(regretOnMax())	selectSmallest(valueSuccessRate())	25.499
IterativeDiving	selectLargest(successRate())	selectLargest(valueImpact())	25.696
IterativeDiving	selectLargest(successRate())	selectSmallest(valueSuccessRate())	25.781
IterativeDiving	selectLargest(successRate())	selectLargest(value())	25.848
IterativeDiving	selectLargest(successRate())	selectLargest(valueSuccessRate())	25.93
IterativeDiving	selectLargest(impact())	selectLargest(value())	25.994
IterativeDiving	selectLargest(successRate())	selectSmallest(valueImpact())	26.002
IterativeDiving	selectLargest(impact())	selectSmallest(value())	26.02
IterativeDiving	selectLargest(localImpact())	selectLargest(value())	26.112
IterativeDiving	selectLargest(localImpact())	selectSmallest(value())	26.174
IterativeDiving	selectLargest(impact())	selectLargest(valueSuccessRate())	26.182
IterativeDiving	selectLargest(impact())	selectSmallest(valueSuccessRate())	26.23
IterativeDiving	selectLargest(ImpactOfLastBranch())	selectSmallest(valueSuccessRate())	26.252
IterativeDiving	selectLargest(ImpactOfLastBranch())	selectSmallest(value())	26.278
IterativeDiving	selectLargest(impact())	selectSmallest(valueImpact())	26.315
IterativeDiving	selectLargest(localImpact())	selectSmallest(valueImpact())	26.329
IterativeDiving	selectLargest(ImpactOfLastBranch())	selectLargest(valueImpact())	26.346
IterativeDiving	selectLargest(impact())	selectLargest(valueImpact())	26.359
IterativeDiving	selectLargest(localImpact())	selectSmallest(valueSuccessRate())	26.377
IterativeDiving	selectLargest(ImpactOfLastBranch())	selectLargest(valueSuccessRate())	26.474
IterativeDiving	selectLargest(regretOnMin())	selectRandomValue()	26.523
IterativeDiving	selectLargest(ImpactOfLastBranch())	selectLargest(value())	26.538
IterativeDiving	selectLargest(localImpact())	selectLargest(valueImpact())	26.591
IterativeDiving	selectLargest(localImpact())	selectLargest(valueSuccessRate())	26.776
IterativeDiving	selectLargest(ImpactOfLastBranch())	selectSmallest(valueImpact())	27.228
IterativeDiving	selectLargest(regretOnMax())	selectRandomValue()	28.226
IterativeDiving	selectLargest(successRate())	selectRandomValue()	28.762
IterativeDiving	selectLargest(impact())	selectRandomValue()	29.352
IterativeDiving	selectRandomVar()	selectSmallest(value())	30.508
IterativeDiving	selectLargest(ImpactOfLastBranch())	selectRandomValue()	30.637

IterativeDiving	selectLargest(localImpact())	selectRandomValue()	30.769
IterativeDiving	selectRandomVar()	selectSmallest(valueImpact())	30.858
IterativeDiving	selectRandomVar()	selectLargest(value())	32.157
IterativeDiving	selectRandomVar()	selectSmallest(valueSuccessRate())	33.441

B.1.5 Iterative Diving (C=2)

Table 17: All iterations of capacity 2 executed with Iterative Diving method

	Largest Value	Largest Value Impact	Largest Value Success Rate	Random Value	Smallest Value	Smallest Value Impact	Smallest Value Success Rate
Largest Impact Of Last Branch	26.54	26.35	26.47	30.64	26.28	27.23	26.25
Largest Domain Max	23.89	24.13	24.14	24.78	23.74	23.95	23.91
Largest Domain Min	23.94	23.85	23.69	24.4	23.48	23.51	23.68
Largest Impact	25.99	26.36	26.18	29.35	26.02	26.32	26.23
Largest Local Impact	26.11	26.59	26.78	30.77	26.17	26.33	26.38
Largest Regret On Max	25.01	24.87	25.05	28.23	25.43	25.22	25.5
Largest Regret On Min	24.07	24.86	24.95	26.52	23.79	24.49	24.08
Largest Success Rate	25.85	25.7	25.93	28.76	25.05	26	25.78
Random Var	32.16	30.51	30.86	33.44	1.694	1.659	1.709
Smallest Impact Of Last Branch	23.13	23.01	23.33	23.61	22.97	22.88	23.16
Smallest Domain Max	22.23	22.26	22.41	22.49	22.13	22.39	22.1
Smallest Domain Min	22.1	22.24	22.77	22.66	21.92	22.9	22.38
Smallest Domain Size	23.28	23.33	23.55	23.94	23.39	23.24	23.06
Smallest Impact	22.72	22.83	22.78	22.86	22.95	22.66	22.79
Smallest Local Impact	22.83	22.91	23.09	23.35	22.9	22.7	22.86
Smallest Regret On Max	22.32	22.49	22.59	22.43	22.41	22.4	22.42
Smallest Regret On Min	22.38	22.49	22.26	22.22	22.52	22.59	22.31
Smallest Success Rate	22.49	22.53	22.6	22.53	22.41	22.62	22.68

B.1.6 Restart (C=2)

Table 18: All iterations of capacity 2 executed with Restart method

	Largest Value	Largest Value Impact	Largest Value Success Rate	Smallest Value	Smallest Value Impact	Smallest Value Success Rate
Largest Impact Of Last Branch	24.01	24.09	24.19	23.85	24.13	24.06
Largest Domain Max	22.32	22.46	22.45	22.38	22.5	22.38
Largest Domain Min	22.12	22.09	22.1	21.92	22.32	22.04
Largest Impact	23.68	23.8	23.67	23.6	23.61	23.81
Largest Local Impact	23.9	24	24.11	23.76	23.94	23.86
Largest Regret On Max	23.16	23.5	23.49	22.72	22.72	22.72
Largest Regret On Min	22.54	22.64	22.51	22.51	22.53	22.46
Largest Success Rate	23.55	23.62	23.67	23.38	23.53	23.72
Smallest Impact Of Last Branch	21.57	21.83	21.79	21.56	21.59	21.43
Smallest Domain Max	21.02	21	20.94	21.15	21.06	21.08
Smallest Domain Min	20.89	20.9	20.96	20.74	20.9	20.86
Smallest Domain Size	21.9	22.01	21.87	21.81	21.89	21.91
Smallest Impact	21.46	21.43	21.54	21.47	21.36	21.46
Smallest Local Impact	21.78	21.65	21.65	21.57	21.59	21.54
Smallest Regret On Max	21.13	21.25	21.14	21.15	21.14	21.08
Smallest Regret On Min	20.93	21	21.09	21.03	20.97	21.03
Smallest Success Rate	21.38	21.42	21.33	21.22	21.28	21.19

Table 19: All iterations of capacity 2 executed with Iterative Diving method

	Largest Value	**Largest Value Impact**	**Largest Value Success Rate**	**Random Value**	**Smallest Value**	**Smallest Value Impact**	**Smallest Value Success Rate**
Largest Impact Of Last Branch	26.54	26.35	26.47	30.64	26.28	27.23	26.25
Largest Domain Max	23.89	24.13	24.14	24.78	23.74	23.95	23.91
Largest Domain Min	23.94	23.85	23.69	24.4	23.48	23.51	23.68
Largest Impact	25.99	26.36	26.18	29.35	26.02	26.32	26.23
Largest Local Impact	26.11	26.59	26.78	30.77	26.17	26.33	26.38
Largest Regret On Max	25.01	24.87	25.05	28.23	25.43	25.22	25.5
Largest Regret On Min	24.07	24.86	24.95	26.52	23.79	24.49	24.08
Largest Success Rate	25.85	25.7	25.93	28.76	25.05	26	25.78
Random Var	32.16	30.51	30.86	33.44	1.396	1.378	1.347
Smallest Impact Of Last Branch	23.13	23.01	23.33	23.61	22.97	22.88	23.16
Smallest Domain Max	22.23	22.26	22.41	22.49	22.13	22.39	22.1
Smallest Domain Min	22.1	22.24	22.77	22.66	21.92	22.9	22.38
Smallest Domain Size	23.28	23.33	23.55	23.94	23.39	23.24	23.06
Smallest Impact	22.72	22.83	22.78	22.86	22.95	22.66	22.79
Smallest Local Impact	22.83	22.91	23.09	23.35	22.9	22.7	22.86
Smallest Regret On Max	22.32	22.49	22.59	22.43	22.41	22.4	22.42
Smallest Regret On Min	22.38	22.49	22.26	22.22	22.52	22.59	22.31
Smallest Success Rate	22.49	22.53	22.6	22.53	22.41	22.62	22.68

B.1.7 Capacity = 3

Table 20: All hyperparameter combinations for capacity three

Method	Variable Selection	Value Selection	Time (s)
IterativeDiving	selectLargest(impact())	selectSmallest(value())	74.12
IterativeDiving	selectLargest(domainMin())	selectSmallest(value())	74.205
IterativeDiving	selectSmallest(successRate())	selectSmallest(value())	76.753
IterativeDiving	selectLargest(impact())	selectSmallest(valueImpact())	80.333
IterativeDiving	selectLargest(domainMin())	selectSmallest(valueSuccessRate())	80.689
IterativeDiving	selectSmallest(successRate())	selectSmallest(valueSuccessRate())	80.785
IterativeDiving	selectLargest(domainMin())	selectSmallest(valueImpact())	80.852
IterativeDiving	selectSmallest(successRate())	selectSmallest(valueImpact())	80.862
IterativeDiving	selectLargest(impact())	selectSmallest(valueSuccessRate())	81.046
IterativeDiving	selectLargest(domainMin())	selectLargest(value())	81.32
IterativeDiving	selectLargest(impact())	selectLargest(value())	81.344
IterativeDiving	selectSmallest(successRate())	selectLargest(value())	81.418
IterativeDiving	selectLargest(impact())	selectLargest(valueImpact())	81.597
IterativeDiving	selectLargest(impact())	selectLargest(valueSuccessRate())	81.789
IterativeDiving	selectLargest(domainMin())	selectLargest(valueSuccessRate())	81.81
IterativeDiving	selectLargest(domainMin())	selectLargest(valueImpact())	81.961
IterativeDiving	selectSmallest(domainMin())	selectSmallest(value())	81.987
IterativeDiving	selectLargest(localImpact())	selectSmallest(value())	82.262
IterativeDiving	selectSmallest(successRate())	selectLargest(valueImpact())	82.321
IterativeDiving	selectSmallest(successRate())	selectLargest(valueSuccessRate())	82.346
IterativeDiving	selectLargest(domainMax())	selectSmallest(value())	82.464
IterativeDiving	selectSmallest(impact())	selectSmallest(value())	82.527
IterativeDiving	selectLargest(localImpact())	selectSmallest(valueImpact())	82.698
IterativeDiving	selectSmallest(domainMin())	selectSmallest(valueImpact())	82.727
IterativeDiving	selectSmallest(domainMin())	selectSmallest(valueSuccessRate())	82.772
IterativeDiving	selectLargest(localImpact())	selectSmallest(valueSuccessRate())	83.034
IterativeDiving	selectLargest(domainMax())	selectSmallest(valueImpact())	83.107
IterativeDiving	selectSmallest(impact())	selectSmallest(valueImpact())	83.259
IterativeDiving	selectLargest(domainMax())	selectSmallest(valueSuccessRate())	83.321
IterativeDiving	selectSmallest(impact())	selectSmallest(valueSuccessRate())	83.416
IterativeDiving	selectLargest(domainMax())	selectLargest(value())	83.487
IterativeDiving	selectLargest(localImpact())	selectLargest(value())	83.579
IterativeDiving	selectSmallest(impact())	selectLargest(value())	83.645
IterativeDiving	selectLargest(localImpact())	selectLargest(valueImpact())	83.774
IterativeDiving	selectSmallest(impact())	selectLargest(valueImpact())	84.038
IterativeDiving	selectLargest(ImpactOfLastBranch())	selectSmallest(value())	84.074
Restart	selectSmallest(successRate())	selectSmallest(value())	84.11
IterativeDiving	selectLargest(domainMax())	selectLargest(valueImpact())	84.199
IterativeDiving	selectLargest(localImpact())	selectLargest(valueSuccessRate())	84.231
IterativeDiving	selectLargest(domainMax())	selectLargest(valueSuccessRate())	84.312
Restart	selectLargest(domainMin())	selectSmallest(value())	84.335

IterativeDiving	selectSmallest(domainMin())	selectLargest(valueImpact())	84.343
IterativeDiving	selectSmallest(domainMin())	selectLargest(valueSuccessRate())	84.375
IterativeDiving	selectSmallest(domainMin())	selectLargest(value())	84.444
IterativeDiving	selectLargest(regretOnMin())	selectSmallest(value())	84.455
IterativeDiving	selectSmallest(impact())	selectLargest(valueSuccessRate())	84.613
IterativeDiving	selectSmallest(localImpact())	selectSmallest(value())	84.697
IterativeDiving	selectSmallest(localImpact())	selectSmallest(valueImpact())	84.831
IterativeDiving	selectLargest(ImpactOfLastBranch())	selectSmallest(valueSuccessRate())	84.998
IterativeDiving	selectLargest(regretOnMin())	selectSmallest(valueImpact())	85.146
IterativeDiving	selectLargest(ImpactOfLastBranch())	selectSmallest(valueImpact())	85.207
Restart	selectLargest(impact())	selectSmallest(value())	85.311
IterativeDiving	selectLargest(regretOnMin())	selectSmallest(valueSuccessRate())	85.397
IterativeDiving	selectSmallest(domainMax())	selectSmallest(value())	85.494
IterativeDiving	selectSmallest(localImpact())	selectSmallest(valueSuccessRate())	85.601
IterativeDiving	selectSmallest(domainMax())	selectSmallest(valueSuccessRate())	85.74
IterativeDiving	selectLargest(ImpactOfLastBranch())	selectLargest(value())	85.847
IterativeDiving	selectSmallest(domainMax())	selectSmallest(valueImpact())	86.053
IterativeDiving	selectLargest(regretOnMin())	selectLargest(value())	86.057
IterativeDiving	selectSmallest(localImpact())	selectLargest(value())	86.079
Restart	selectSmallest(regretOnMax())	selectLargest(valueSuccessRate())	86.474
Restart	selectLargest(successRate())	selectLargest(valueSuccessRate())	86.841
IterativeDiving	selectLargest(regretOnMin())	selectLargest(valueSuccessRate())	87.191
IterativeDiving	selectLargest(ImpactOfLastBranch())	selectLargest(valueImpact())	87.193
IterativeDiving	selectSmallest(domainMax())	selectLargest(value())	87.357
IterativeDiving	selectSmallest(localImpact())	selectLargest(valueImpact())	87.384
IterativeDiving	selectLargest(regretOnMin())	selectLargest(valueImpact())	87.499
IterativeDiving	selectSmallest(localImpact())	selectLargest(valueSuccessRate())	87.858
IterativeDiving	selectLargest(ImpactOfLastBranch())	selectLargest(valueSuccessRate())	87.93
IterativeDiving	selectLargest(regretOnMax())	selectSmallest(value())	87.989
IterativeDiving	selectSmallest(ImpactOfLastBranch())	selectSmallest(value())	88.298
IterativeDiving	selectSmallest(domainMax())	selectLargest(valueImpact())	88.371
IterativeDiving	selectSmallest(domainMax())	selectLargest(valueSuccessRate())	88.562
IterativeDiving	selectLargest(regretOnMax())	selectSmallest(valueImpact())	88.608
IterativeDiving	selectSmallest(ImpactOfLastBranch())	selectSmallest(valueImpact())	88.781
IterativeDiving	selectSmallest(regretOnMin())	selectSmallest(value())	89.351
IterativeDiving	selectSmallest(regretOnMin())	selectSmallest(valueImpact())	90.603
Restart	selectLargest(impact())	selectSmallest(valueImpact())	90.982
Restart	selectLargest(domainMin())	selectSmallest(valueImpact())	91.056
Restart	selectSmallest(successRate())	selectSmallest(valueImpact())	91.103
Restart	selectSmallest(successRate())	selectSmallest(valueSuccessRate())	91.342
Restart	selectLargest(impact())	selectSmallest(valueSuccessRate())	91.472
Restart	selectLargest(domainMin())	selectLargest(value())	91.479
IterativeDiving	selectLargest(regretOnMax())	selectSmallest(valueSuccessRate())	91.571
Restart	selectLargest(domainMin())	selectSmallest(valueSuccessRate())	91.681
Restart	selectLargest(impact())	selectLargest(value())	91.776

IterativeDiving	selectSmallest(ImpactOfLastBranch())	selectSmallest(valueSuccessRate())	91.797
Restart	selectSmallest(successRate())	selectLargest(value())	92.131
Restart	selectSmallest(domainSize())	selectLargest(valueSuccessRate())	92.139
Restart	selectLargest(impact())	selectLargest(valueImpact())	92.15
IterativeDiving	selectSmallest(ImpactOfLastBranch())	selectLargest(value())	92.205
IterativeDiving	selectLargest(regretOnMax())	selectLargest(value())	92.332
Restart	selectLargest(impact())	selectLargest(valueSuccessRate())	92.344
Restart	selectSmallest(successRate())	selectLargest(valueSuccessRate())	92.614
Restart	selectLargest(domainMin())	selectLargest(valueImpact())	92.637
Restart	selectSmallest(domainMin())	selectSmallest(value())	92.771
Restart	selectSmallest(successRate())	selectLargest(valueImpact())	92.902
Restart	selectLargest(domainMin())	selectLargest(valueSuccessRate())	92.934
IterativeDiving	selectLargest(regretOnMax())	selectLargest(valueImpact())	92.935
Restart	selectSmallest(domainMin())	selectSmallest(valueSuccessRate())	93.08
Restart	selectLargest(localImpact())	selectSmallest(value())	93.091
Restart	selectLargest(localImpact())	selectSmallest(valueImpact())	93.304
Restart	selectSmallest(impact())	selectSmallest(value())	93.408
Restart	selectSmallest(domainMin())	selectSmallest(valueImpact())	93.438
Restart	selectLargest(domainMax())	selectSmallest(value())	93.481
IterativeDiving	selectSmallest(regretOnMin())	selectLargest(value())	93.66
IterativeDiving	selectLargest(regretOnMax())	selectLargest(valueSuccessRate())	93.725
Restart	selectSmallest(domainMin())	selectLargest(value())	93.823
Restart	selectLargest(domainMax())	selectSmallest(valueSuccessRate())	93.833
IterativeDiving	selectLargest(successRate())	selectSmallest(value())	93.837
Restart	selectLargest(localImpact())	selectSmallest(valueSuccessRate())	93.877
Restart	selectSmallest(impact())	selectSmallest(valueSuccessRate())	93.952
Restart	selectSmallest(impact())	selectSmallest(valueImpact())	93.961
Restart	selectLargest(domainMax())	selectSmallest(valueImpact())	94.014
IterativeDiving	selectSmallest(regretOnMin())	selectSmallest(valueSuccessRate())	94.262
Restart	selectLargest(localImpact())	selectLargest(value())	94.307
IterativeDiving	selectSmallest(ImpactOfLastBranch())	selectLargest(valueSuccessRate())	94.338
IterativeDiving	selectSmallest(domainSize())	selectSmallest(value())	94.448
IterativeDiving	selectSmallest(ImpactOfLastBranch())	selectLargest(valueImpact())	94.451
Restart	selectLargest(domainMax())	selectLargest(value())	94.474
Restart	selectSmallest(impact())	selectLargest(value())	94.488
Restart	selectSmallest(domainMin())	selectLargest(valueImpact())	94.523
Restart	selectSmallest(domainMin())	selectLargest(valueSuccessRate())	94.694
Restart	selectLargest(localImpact())	selectLargest(valueImpact())	94.816
Restart	selectSmallest(impact())	selectLargest(valueImpact())	94.89
IterativeDiving	selectLargest(successRate())	selectSmallest(valueImpact())	94.898
IterativeDiving	selectSmallest(regretOnMin())	selectLargest(valueImpact())	95.162
Restart	selectLargest(localImpact())	selectLargest(valueSuccessRate())	95.188
IterativeDiving	selectLargest(successRate())	selectSmallest(valueSuccessRate())	95.286
Restart	selectSmallest(impact())	selectLargest(valueSuccessRate())	95.3
Restart	selectSmallest(domainMax())	selectSmallest(value())	95.349

Restart	selectLargest(domainMax())	selectLargest(valueImpact())	95.518
IterativeDiving	selectSmallest(regretOnMax())	selectSmallest(value())	95.573
IterativeDiving	selectSmallest(domainSize())	selectSmallest(valueSuccessRate())	95.631
Restart	selectSmallest(localImpact())	selectSmallest(value())	95.746
Restart	selectLargest(domainMax())	selectLargest(valueSuccessRate())	95.762
Restart	selectLargest(ImpactOfLastBranch())	selectSmallest(value())	95.763
IterativeDiving	selectLargest(successRate())	selectLargest(value())	95.939
IterativeDiving	selectSmallest(regretOnMin())	selectLargest(valueSuccessRate())	96.072
Restart	selectLargest(regretOnMin())	selectSmallest(value())	96.09
Restart	selectLargest(ImpactOfLastBranch())	selectSmallest(valueImpact())	96.104
Restart	selectSmallest(domainMax())	selectSmallest(valueImpact())	96.116
IterativeDiving	selectSmallest(domainSize())	selectSmallest(valueImpact())	96.253
Restart	selectLargest(regretOnMin())	selectSmallest(valueImpact())	96.591
Restart	selectSmallest(localImpact())	selectSmallest(valueImpact())	96.707
IterativeDiving	selectSmallest(regretOnMax())	selectSmallest(valueImpact())	97.248
Restart	selectLargest(ImpactOfLastBranch())	selectSmallest(valueSuccessRate())	97.253
IterativeDiving	selectLargest(successRate())	selectLargest(valueImpact())	97.28
Restart	selectSmallest(domainMax())	selectSmallest(valueSuccessRate())	97.34
Restart	selectLargest(regretOnMin())	selectSmallest(valueSuccessRate())	97.588
Restart	selectSmallest(localImpact())	selectSmallest(valueSuccessRate())	97.767
IterativeDiving	selectLargest(successRate())	selectLargest(valueSuccessRate())	98.033
Restart	selectSmallest(domainMax())	selectLargest(value())	98.097
Restart	selectLargest(ImpactOfLastBranch())	selectLargest(value())	98.228
IterativeDiving	selectSmallest(domainSize())	selectLargest(value())	98.412
IterativeDiving	selectSmallest(domainSize())	selectLargest(valueSuccessRate())	98.425
IterativeDiving	selectSmallest(domainSize())	selectLargest(valueImpact())	98.571
Restart	selectLargest(regretOnMin())	selectLargest(value())	98.581
Restart	selectSmallest(localImpact())	selectLargest(value())	98.609
IterativeDiving	selectSmallest(regretOnMax())	selectLargest(value())	98.618
IterativeDiving	selectSmallest(regretOnMax())	selectSmallest(valueSuccessRate())	98.714
Restart	selectSmallest(domainMax())	selectLargest(valueImpact())	98.932
Restart	selectLargest(ImpactOfLastBranch())	selectLargest(valueImpact())	99.129
Restart	selectSmallest(localImpact())	selectLargest(valueImpact())	99.466
IterativeDiving	selectSmallest(regretOnMax())	selectLargest(valueImpact())	99.692
Restart	selectLargest(regretOnMin())	selectLargest(valueImpact())	99.996
Restart	selectSmallest(domainMax())	selectLargest(valueSuccessRate())	100.61
Restart	selectLargest(ImpactOfLastBranch())	selectLargest(valueSuccessRate())	100.616
Restart	selectLargest(regretOnMin())	selectLargest(valueSuccessRate())	100.883
Restart	selectSmallest(localImpact())	selectLargest(valueSuccessRate())	100.903
IterativeDiving	selectSmallest(regretOnMax())	selectLargest(valueSuccessRate())	101.557
Restart	selectSmallest(regretOnMin())	selectSmallest(value())	102.823
Restart	selectSmallest(ImpactOfLastBranch())	selectSmallest(value())	102.957
Restart	selectSmallest(regretOnMin())	selectSmallest(valueImpact())	103.821
Restart	selectSmallest(ImpactOfLastBranch())	selectSmallest(valueImpact())	104.13
Restart	selectSmallest(regretOnMin())	selectSmallest(valueSuccessRate())	104.235

Restart	selectSmallest(regretOnMin())	selectLargest(value())	104.326
Restart	selectSmallest(ImpactOfLastBranch())	selectSmallest(valueSuccessRate())	104.987
Restart	selectLargest(regretOnMax())	selectSmallest(valueImpact())	105.022
Restart	selectLargest(regretOnMax())	selectSmallest(valueSuccessRate())	105.757
Restart	selectLargest(regretOnMax())	selectSmallest(value())	105.766
Restart	selectLargest(regretOnMax())	selectLargest(value())	105.909
Restart	selectSmallest(ImpactOfLastBranch())	selectLargest(value())	105.961
Restart	selectSmallest(regretOnMin())	selectLargest(valueImpact())	106.191
Restart	selectSmallest(regretOnMin())	selectLargest(valueSuccessRate())	106.799
Restart	selectLargest(regretOnMax())	selectLargest(valueImpact())	107.153
Restart	selectSmallest(ImpactOfLastBranch())	selectLargest(valueImpact())	107.277
Restart	selectLargest(regretOnMax())	selectLargest(valueSuccessRate())	107.567
Restart	selectSmallest(ImpactOfLastBranch())	selectLargest(valueSuccessRate())	107.626
Restart	selectLargest(successRate())	selectSmallest(value())	107.87
Restart	selectSmallest(regretOnMax())	selectSmallest(value())	108.388
Restart	selectSmallest(domainSize())	selectSmallest(value())	108.956
Restart	selectSmallest(regretOnMax())	selectSmallest(valueImpact())	109.836
Restart	selectSmallest(regretOnMax())	selectSmallest(valueSuccessRate())	110.095
Restart	selectSmallest(domainSize())	selectSmallest(valueImpact())	110.287
Restart	selectLargest(successRate())	selectSmallest(valueImpact())	110.809
Restart	selectLargest(successRate())	selectSmallest(valueSuccessRate())	111.247
Restart	selectSmallest(regretOnMax())	selectLargest(valueImpact())	112.006
Restart	selectSmallest(domainSize())	selectSmallest(valueSuccessRate())	112.36
Restart	selectLargest(successRate())	selectLargest(value())	112.405
Restart	selectSmallest(regretOnMax())	selectLargest(value())	112.569
Restart	selectSmallest(domainSize())	selectLargest(value())	113.153
Restart	selectLargest(successRate())	selectLargest(valueImpact())	114.888
Restart	selectSmallest(domainSize())	selectLargest(valueImpact())	115.015

B.1.8 Iterative Diving (C=3)

Table 21: All iterations of capacity 3 executed with Iterative Diving method

	Largest Value	**Largest Value Impact**	**Largest Value Success Rate**	**Smallest Value**	**Smallest Value Impact**	**Smallest Value Success Rate**
Largest Impact Of Last Branch	85.85	87.19	87.93	84.07	85.21	85
Largest Domain Max	83.49	84.2	84.31	82.46	83.11	83.32
Largest Domain Min	81.32	81.96	81.81	74.21	80.85	80.69
Largest Impact	81.34	81.6	81.79	74.12	80.33	81.05
Largest Local Impact	83.58	83.77	84.23	82.26	82.7	83.03
Largest Regret On Max	92.33	92.94	93.73	87.99	88.61	91.57
Largest Regret On Min	86.06	87.5	87.19	84.46	85.15	85.4
Largest Success Rate	95.94	97.28	98.03	93.84	94.9	95.29
Smallest Impact Of Last Branch	92.21	94.45	94.34	88.3	88.78	91.8
Smallest Domain Max	87.36	88.37	88.56	85.49	86.05	85.74
Smallest Domain Min	84.44	84.34	84.38	81.99	82.73	82.77
Smallest Domain Size	98.41	98.57	98.43	94.45	96.25	95.63
Smallest Impact	83.65	84.04	84.61	82.53	83.26	83.42
Smallest Local Impact	86.08	87.38	87.86	84.7	84.83	85.6
Smallest Regret On Max	98.62	99.69	101.56	95.57	97.25	98.71
Smallest Regret On Min	93.66	95.16	96.07	89.35	90.6	94.26
Smallest Success Rate	81.42	82.32	82.35	76.75	80.86	80.79

B.1.9 Restart (C=3)

Table 22: All iterations of capacity 3 executed with Restart method

	Largest Value	**Largest Value Impact**	**Largest Value Success Rate**	**Smallest Value**	**Smallest Value Impact**	**Smallest Value Success Rate**
Largest Impact Of Last Branch	98.23	99.13	100.62	95.76	96.10	97.25
Largest Domain Max	94.47	95.52	95.76	93.48	94.01	93.83
Largest Domain Min	91.48	92.64	92.93	84.34	91.06	91.68
Largest Impact	91.78	92.15	92.34	85.31	90.98	91.47
Largest Local Impact	94.31	94.82	95.19	93.09	93.30	93.88
Largest Regret On Max	105.91	107.15	107.57	105.77	105.02	105.76
Largest Regret On Min	98.58	100.00	100.88	96.09	96.59	97.59
Largest Success Rate	112.41	114.89	86.84	107.87	110.81	111.25
Smallest Impact Of Last Branch	105.96	107.28	107.63	102.96	104.13	104.99
Smallest Domain Max	98.10	98.93	100.61	95.35	96.12	97.34
Smallest Domain Min	93.82	94.52	94.69	92.77	93.44	93.08
Smallest Domain Size	113.15	115.02	92.14	108.96	110.29	112.36
Smallest Impact	94.49	94.89	95.30	93.41	93.96	93.95
Smallest Local Impact	98.61	99.47	100.90	95.75	96.71	97.77
Smallest Regret On Max	112.57	112.01	86.47	108.39	109.84	110.10
Smallest Regret On Min	104.33	106.19	106.80	102.82	103.82	104.24
Smallest Success Rate	92.13	92.90	92.61	84.11	91.10	91.34

Table 23: All iterations of capacity 3 executed with Iterative Diving method

	Largest Value	**Largest Value Impact**	**Largest Value Success Rate**	**Random Value**	**Smallest Value**	**Smallest Value Impact**	**Smallest Value Success Rate**
Largest Impact Of Last Branch	26.54	26.35	26.47	30.64	26.28	27.23	26.25
Largest Domain Max	23.89	24.13	24.14	24.78	23.74	23.95	23.91
Largest Domain Min	23.94	23.85	23.69	24.4	23.48	23.51	23.68
Largest Impact	25.99	26.36	26.18	29.35	26.02	26.32	26.23
Largest Local Impact	26.11	26.59	26.78	30.77	26.17	26.33	26.38
Largest Regret On Max	25.01	24.87	25.05	28.23	25.43	25.22	25.5
Largest Regret On Min	24.07	24.86	24.95	26.52	23.79	24.49	24.08
Largest Success Rate	25.85	25.7	25.93	28.76	25.05	26	25.78
Random Var	32.16	30.51	30.86	33.44	1.396	1.378	1.347
Smallest Impact Of Last Branch	23.13	23.01	23.33	23.61	22.97	22.88	23.16
Smallest Domain Max	22.23	22.26	22.41	22.49	22.13	22.39	22.1
Smallest Domain Min	22.1	22.24	22.77	22.66	21.92	22.9	22.38
Smallest Domain Size	23.28	23.33	23.55	23.94	23.39	23.24	23.06
Smallest Impact	22.72	22.83	22.78	22.86	22.95	22.66	22.79
Smallest Local Impact	22.83	22.91	23.09	23.35	22.9	22.7	22.86
Smallest Regret On Max	22.32	22.49	22.59	22.43	22.41	22.4	22.42
Smallest Regret On Min	22.38	22.49	22.26	22.22	22.52	22.59	22.31
Smallest Success Rate	22.49	22.53	22.6	22.53	22.41	22.62	22.68

B.2 Results from stage 2

Table 24: Results from stage 2 performed with rules from stage 2 and 3.

Scenario		Model 2		Model 3	
Boxes	Capacity	Time (s)	Objective (m)	Time (sec)	Objective (m)
BBGG	1	3.351	14.704	1.315	14.704
BBGG	2	84.515	10.614	24.25	10.314
BBGG	3	292.815	9.366	95.753	9.414
BBGG	4	232.317	8.484	108.722	8.484
GBBG	1	3.613	14.246	1.902	14.704
GBBG	2	67.786	10.314	23.722	10.614
GBBG	3	217.314	8.766	91.669	9.414
GBBG	4	267.924	8.484	110.203	8.484
GGBB	1	3.658	14.246	1.248	14.704
GGBB	2	86.308	10.014	21.519	10.014
GGBB	3	273.74	8.766	83.473	9.414
GGBB	4	252.866	8.484	106.48	8.484
GBGB	1	3.579	14.246	1.696	14.704
GBGB	2	85.664	10.314	23.358	10.614
GBGB	3	183.52	8.766	88.905	9.414
GBGB	4	386.891	8.484	82.679	8.484
BGBG	1	4.393	14.546	1.273	14.704
BGBG	2	85.245	10.614	22.822	10.614
BGBG	3	181.574	9.066	73.52	9.414
BGBG	4	378.503	8.484	71.962	8.484
BGGB	1	4.143	14.546	1.479	14.704
BGGB	2	84.123	10.468	30.752	10.314
BGGB	3	247.059	9.066	87.626	9.414
BGGB	4	313.775	8.484	99.517	8.484

B.3 Results from Stage 3

Table 25: Results from Stage 3

Boxes	Capacity	Time (s)	Objective (m)
BRGG	1	20.716	16.132
BRGG	2	120.024	11.952
BRGG	3	120.039	11.952
BRGG	4	120.011	11.957
RGGB	1	19.844	15.232
RGGB	2	120.033	11.952
RGGB	3	120.049	11.652
RGGB	4	120.009	11.652
GBGR	1	17.122	15.832
GBGR	2	120.025	12.252
GBGR	3	120.016	12.252
GBGR	4	120.01	11.957
GGBR	1	17.954	15.832
GGBR	2	120.032	12.252
GGBR	3	120.04	12.252
GGBR	4	120.026	11.957
GRGB	1	16.747	15.232
GRGB	2	120.023	11.652
GRGB	3	120.033	11.652
GRGB	4	120.008	11.652
GBRG	1	17.306	15.832
GBRG	2	120.012	12.252
GBRG	3	120.038	11.952
GBRG	4	120.032	11.957
GGRB	1	17.470	15.532
GGRB	2	120.035	11.952
GGRB	3	120.013	11.952
GGRB	4	120.151	11.957
RBGG	1	22.054	15.832
RBGG	2	120.033	11.952
RBGG	3	120.045	11.952
RBGG	4	120.042	11.957
BGGR	1	20.772	16.132
BGGR	2	120.008	12.252
BGGR	3	120.007	12.252
BGGR	4	120.056	11.957
RGBG	1	19.425	15.532
RGBG	2	120.035	11.652
RGBG	3	120.019	11.652
RGBG	4	120.034	11.652
GRBG	1	17.200	15.532
GRBG	2	120.031	12.252
GRBG	3	120.009	11.652
GRBG	4	120.023	11.652
BGRG	1	20.130	16.132
BGRG	2	120.020	11.952
BGRG	3	120.055	11.952
BGRG	4	120.056	11.957

C Models comparison

C.1 Stage 2

Table 26: Comparasion between MIP and CP for stage 2 with stage 2 rules

Boxes	Capacity	Time (s)		Objective (m)
		MIP	CP	
BBGG	1	379.969	3.351	14.704
GBBG	1	633.562	3.613	14.246
GGBB	1	347.000	3.658	14.246
GBGB	1	733.172	3.579	14.246
BGBG	1	871.75	4.393	14.546
BGGB	1	387.516	4.143	14.546
BBGG	2	636.141	84.515	10.614
GBBG	2	808.86	67.786	10.314
GGBB	2	533.937	86.308	10.014
GBGB	2	865.672	85.664	10.314
BGBG	2	591.859	85.245	10.614
BGGB	2	632.187	84.123	10.468
BBGG	3	132.344	292.815	9.366
GBBG	3	146.515	217.314	8.766
GGBB	3	234.922	273.74	8.766
GBGB	3	193.937	183.52	8.766
BGBG	3	165.75	181.574	9.066
BGGB	3	398.531	247.059	9.066
BBGG	4	102.391	232.317	8.484
GBBG	4	154.64	267.924	8.484
GGBB	4	139.14	252.866	8.484
GBGB	4	202.047	386.891	8.484
BGBG	4	138.532	378.503	8.484
BGGB	4	80.579	313.775	8.484

C.2 Stage 2 with stage 3 rules

Table 27: Comparasion between MIP and CP for stage 2 with stage 3 rules

Boxes	**Capacity**	**Time (s)**		**Objective (m)**
		MIP	**CP**	
GBBG	1	147.797	1.315	14.704
BGGB	1	162.704	1.902	14.704
GGBB	1	161.86	1.248	14.704
BGBG	1	127.844	1.696	14.704
BBGG	1	171.203	1.273	14.704
GBGB	1	189.64	1.479	14.704
GBBG	2	130.015	24.25	10.314
BGGB	2	382.484	23.722	10.614
GGBB	2	230.922	21.519	10.014
BGBG	2	140.343	23.358	10.614
BBGG	2	392.375	22.822	10.614
GBGB	2	393.719	30.752	10.314
GBBG	3	112.656	95.753	9.414
BGGB	3	79.281	91.669	9.414
GGBB	3	149.969	83.473	9.414
BGBG	3	94.75	88.905	9.414
BBGG	3	153.547	73.52	9.414
GBGB	3	78.797	87.626	9.414
GBBG	4	57.703	108.722	8.484
BGGB	4	71.203	110.203	8.484
GGBB	4	63.312	106.48	8.484
BGBG	4	57.515	82.679	8.484
BBGG	4	67.828	71.962	8.484
GBGB	4	78.141	99.517	8.484

C.3 Stage 3

Scenario		Time (s)	
Boxes	Capacity	MIP	CP
BGGR	1	1802.875	20.772
BGRG	1	1681.734	20.13
BRGG	1	2300.047	20.716
GBGR	1	1867.297	17.122
GBRG	1	2378.25	17.306
GGBR	1	2689.39	17.954
GGRB	1	2019.313	17.47
GRBG	1	1924.094	17.2
GRGB	1	2340.781	16.747
RBGG	1	1725.891	22.054
RGBG	1	2150.641	19.425
RGGB	1	1778.016	19.844
BGGR	2	3605.078	120.008
BGRG	2	3603.406	120.02
BRGG	2	3605.625	120.024
GBGR	2	3605.453	120.025
GBRG	2	3604.547	120.012
GGBR	2	3604.562	120.032
GGRB	2	3601	120.035
GRBG	2	3605.204	120.031
GRGB	2	3603.969	120.023
RBGG	2	3604.984	120.033
RGBG	2	3602.25	120.035
RGGB	2	3604.954	120.033
BGGR	3	3604.625	120.007
BGRG	3	3604.766	120.055
BRGG	3	3603.219	120.039
GBGR	3	3607.547	120.016
GBRG	3	3607.281	120.038
GGBR	3	3601.485	120.04
GGRB	3	3604.797	120.013
GRBG	3	3601.906	120.009
GRGB	3	3606.625	120.033
RBGG	3	1900.282	120.045
RGBG	3	3601.734	120.019
RGGB	3	3603.422	120.049
BGGR	4	2854.281	120.056
BGRG	4	2780.656	120.056
BRGG	4	3600.891	120.011
GBGR	4	2106.031	120.01
GBRG	4	1638.188	120.032
GGBR	4	2476.406	120.026
GGRB	4	3498.781	120.151
GRBG	4	3602.031	120.023
GRGB	4	3601.375	120.008
RBGG	4	2216.125	120.042
RGBG	4	1913.016	120.034
RGGB	4	2270.328	120.009