UNIVERSITÀ
DI TRENTO
Dipartimento di
Ingegneria Industriale

Final Project Group

The B pills

Group Members			
Stefano	*Tonini*	*EIT AUS*	*248413*
Lorenzo	*Sciotto*	*Management and Industrial System Engineering*	*248965*
Jacopo	*Dallafior*	*EIT AUS*	*249805*

1. Presentation

The B Pills is a leading company in the production of pills against certain degenerative diseases which it delivers to pharmacies and hospitals. After changing managers, the company decided to hire engineers to optimize the various processes, following certain production standards:

- in order to have maximum sterility of the bottles and blisters, they must be delivered one at a time
- the finished products should be divided already on the conveyor according to the final customer (hospital or pharmacy)
- The average daily demand is 66 orders divided into three 8-hour shifts, the request of the manager of The B pill is to go from a production per shift of 15 to an average production of 22 products per shift

It was possible to develop a digital twin of this real production system, analyse the performance and evaluate where bottle-neck and slowages appeared in the production system.
Furthermore, to improve the effectiveness, feasible simulations were built and evaluated.
The coordination office of Plan B receives orders from all Italian pharmacies and hospitals. Hospitals request larger product quantities, while the orders required from the pharmacies are smaller.

input

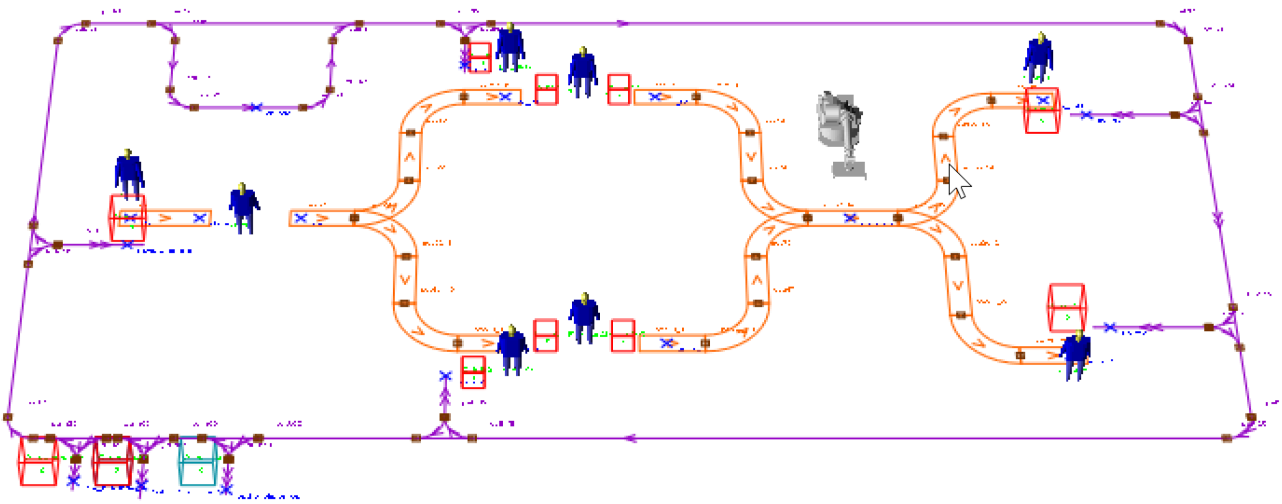
ID_order	Release_time(sec)	Prod_type	Order_dimens(pcs)	Destination_ID
101003	13	1	10	1
101033	142	2	10	1
101046	282	2	4	2
101085	419	1	9	1
101120	632	2	3	2
101160	1037	1	4	2
101198	1227	1	10	1
101213	1416	2	4	2
101224	1634	2	9	1
101252	1873	1	9	1
101293	2063	2	3	2

The orders are received once a week and scheduled in a csv file composed of 5 columns that provides necessary information to the manufacturing plant in order to fulfil the weekly order in shifts of 8 hours per day.
The first column represents the ID of each order, the second regards the time at which the raw powder should be picked up, the third one deals with product type, the fourth one states the order dimension and the last provides the destination of the specific order.

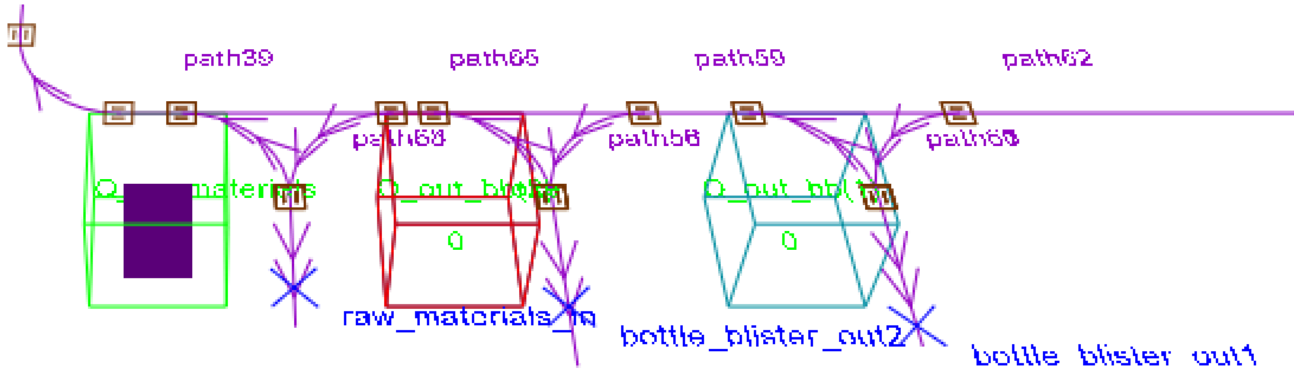
The company adopted a plant composed of two systems, and the figure below gives a clear understanding of the environment they created.

The first system is a conveyor, where orders enter and are worked. This system starts with the work of an operator that puts the correct quantity of powder brought by the forklift. At the conclusion of that operation, a stamping machine will produce the pills in batches of 10 pills per each stamp. Then the pills will be split in proportion to the orders that were received. The company produces two different packaging for the pills, one is the bottle confection, that is done by the bottle filling machine (it is the upper conveyor) and the other is the blister confection, that is done by a blister filling machine (it is the lower conveyor). Both types of confection contain 10 pills inside.
The confections after the elaboration will be boxed by a cartoning machine supervised by an operator. The box after those operations will move on the conveyor to an labelling machine that assigns a code to theS box. Then the box will go to a related outlet conveyor, that difference is link depending on the number of confections inside the box; One outlet conveyor is dedicated to the pharmacy and supermarket (it is the upper conveyor), and the other to the shop replenishment (it is the lower conveyor).

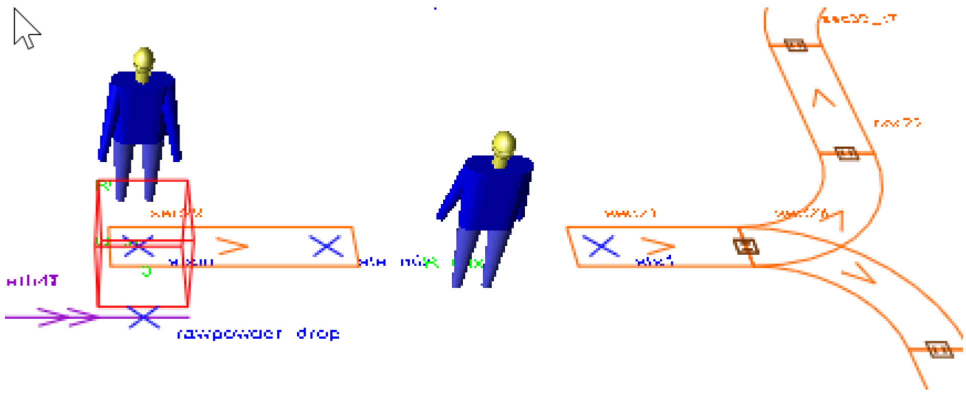
The second system of the company's plant is operated by a forklift, which refills some operations on the conveyor with raw materials or raw bottles/blisters and moves the final product from the outlet conveyor to the warehouse according to the types of product.



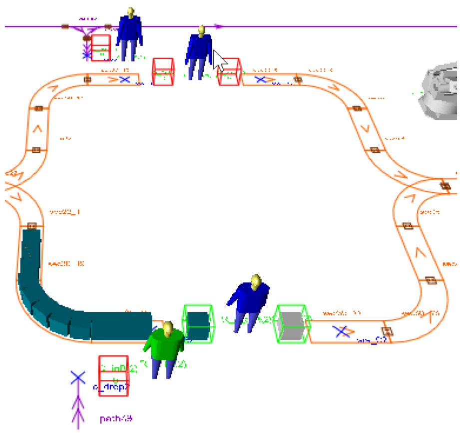
Analysing the company's plant in detail, from the bottom left corner we see, as shown in the figure below, three different queues. The first queue on the left is dedicated to the refillment of the conveyor with the raw powder and the row bootle and blister (***Q_rawmaterials***). The queue in the middle is dedicated to delivery products dedicated for the hospital (***Q_out_bb(1)***). The last queue on the right is used for the final product ready to be shipped for the pharmacy (***Q_out_bb(2)***).

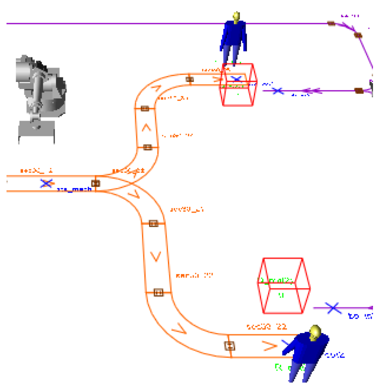


The forklift receives the order in the plant from the CSV, after this it will move from the ***cp_park*** (where it is at the beginning) to the ***Q_rawmaterials***. Once there, it brings a fixed quantity of powder from the ***Q_rawmaterials*** and it delivers this quantity to the ***Q_in***, where the quantity is fixed to produce an amount of 10 pills. If the quantity of raw material does not satisfy the request of the consumer, another forklift will start to operate to move more powder to ***Q_in***, until the request is achieved. Subsequently, the powder in ***Q_in*** is moved in ***sta_in*** by the operator ***Rin*** and after this, the raw powder is brought in ***sta_mix*** by the operator ***R1***. After the mixing procedure the mixed powder goes in ***sta1***, and in this station the powder is transformed into ten pills.



The upper plant is dedicated to producing a bottled product. A certain quantity of pills, based on the orders on the csv file, will arrive at ***Sta_B1***, where an operator ***R_b_fill(1)*** will refill the ordered number of bottles taken from ***Q_inB(1)***.
Through the use of the forklift, ***Q_inB(1)*** is refilled with raw bottles brought from the ***Q_rowmaterials***. The forklift will bring only a single raw bottle per each trip.
The required number of refilled bottles are moved from ***Q_B(1)*** to ***Q_cart(1)*** by the operator ***R_b_cart(1)*** into a box and then in the station ***sta_C1*** the box is cartoned.





The lower plant is dedicated to the production of blisters. The process is common to the previous description until ***Sta_B2***, where an operator ***R_b_fill(2)*** refills an empty blister taken from ***Q_inB(2)*** (where the forklift delivers one single blister at time).
The required amount of refilled blisters moves from ***Q_B(1)*** to ***Q_cart(1)***, thanks to the operator ***R_b_cart(2)***. As in the bottle process ***sta_C2*** is the station dedicated to the cartoning of a finished box of blisters.
The boxes obtained from ***sta_C1*** and ***sta_C2*** travel along the conveyor until they reach ***sta_mach***, where a robotic arm ***R_mach*** prints the box identification code. After completing the identification operation, the box will take the conveyor that brings it to the destination assigned in the CSV.

2. Simulation Entities

2.1. Load

- **L_dummy** (Number of Load Types: 1, First Process: P_read, Generation Limit: 1, Distribution: Constant, Mean: 5, First One At: Default) it's connected to the first process ***P_read*** so its functionality allows the first process to begin.
- **L_powder** (Number of Load Types: 1) it is recalled at the start of the process **P_powder** (with dimension = 1) to "*set this load type = L_powder"* and it is useful to
- **L_mixpowder** (Number of Load Types: 1) it is recalled at the start of the process **P_mixing** (with dimension =1)to "*set this load type = L_mixpowder"* and it is useful to
- **L_rawB** (Number of Load Types: 2) it is recalled at the start of the process **P_rawB** (with dimension = 2) to "*set this load type = L_rawB(procindex)*" and it is useful to
- **L_B** (Number of Load Types: 2) it is recalled at the start of the process **P_ship** (with dimension = 2) to "*set this load type = L_B(procindex)"* and it is useful to

2.2. Global Variables

- **V_pointer** (Type: FilePtr, Dimension: 1): it's used in the first process P_read to save results after reading the input file "input.csv" that has 5 columns and 106 rows.

- **V_count_cartoning** (Type: Integer; Dim:2): They are used in the process P_conv to keep count of finished bottles or blisters, to be stored in boxes.

- **V_count_pills** (Type: Integer; Dim:2): They are used in the process P_conv serves to ensure that each bottle or blister is filled with exactly the right number of pills (in our case ten pills).

- **V_dim_B** (Type: Integer; Dim:1): It is used in the process P_conv, is used in the if statement together with the variable V_count_pills. This variable is set to a value of ten and indicates the number of pills to be placed in the bottles or blisters.

- ***V_useless*** (Type: String, Dimension: 5): they are used in the first process *P_read* to read the results saved in *V_pointer* with the delimiter ";" and the dimension is 5 like the columns in "input.csv" saved in *V_pointer*.

- **V_mean_B** (Type: Real; Dim:1): It is used in P_ship to compute the average time between finished products.

- **Tot_B** (Type: Integer; Dim:1): It is used in P_ship to track all finished products.

2.3. Load Attribute

- ***V_pointer*** (Type: FilePtr, Dimension: 1): it is used in the first process ***P_read*** to save results after reading the input file “input.csv” that has 5 columns and 701 rows.
- ***V_useless*** (Type: String, Dimension: 5): they are used in the first process ***P_read*** to read the results saved in *V_pointer* with the delimiter “;” and the dimension is 5 like the columns in “input.csv” saved in *V_pointer*.
- **V_id** (Type: Integer; Dim: 1): it is used in ***P_read*** to read *V_useless(1)* from *V_pointer.* It reads the first column “ID_order” from the input file “input.csv”. It stores the ID_order numbers.
- **V_time** (type:Real; Dim: 1): it is used in ***P_read*** to read *V_useless(2)* from *V_pointer.* It reads the second column “Release_time(sec)” from the input file “input.csv”. It contains the time in which the powder arrives at the depot. Then always in *P_read,* while reading *V_pointer*, it waits until the time passes, making *V_time* minus the global counter “ac” and waiting for that number of seconds.
- ***V_type*** (type: Integer; Dim: 1):it is used in ***P_read*** to read *V_useless(3)* from *V_pointer.* It reads the third column “Prod_type” from the input file “input.csv”. It contains numbers that can be “1” (if *the order contains bottle*) or “2” (if *the order contains blositer*). In *P_read*, while reading *V_pointer*, if *V_type* is “1” it clones *V_dimension* to *P_conv(1)* otherwise if *V_type* is “2” it clones *V_dimension* to *P_conv(2). V_type* “1” or “2” classifies which part (up or down) the order goes in the first part of the conveyor (on the left).
- **V_dim** (type: Integer; Dim: 1): it is used in ***P_read*** to read *V_useless(4)* from *V_pointer.* It reads the fourth column “Order_dimension(pcs)” from the input file “input.csv”. It contains numbers of bottles or blisters ordered . In ***P_read***, while reading *V_pointer, V_dimension* is cloned to ***P_powder****.* In this way the loads graphically visible are the pieces of the order.
- **V_dest** (type: Integer; Dim: 1): it is used in ***P_read*** to read *V_useless(5)* from *V_pointer.* It reads the fifth column “Destination_ID” from the input file “input.csv”. It contains numbers from 1 to 4, they are “1” or “2” if *V_type* is “1” otherwise they are “3” or “4” if *V_type* is “2”. Graphically at the beginning of the path mover they are ordered from 1 to 4 from the top to the bottom, first two are for *V_type* “1” while last two are for *V_type* “2”. At the end of the path mover on the top, the two on the left are “4” and “2” from left to right, while the two on the right are “3” and “1” from left to right.

2.4. Order list

- ***OL_b*** (Dimension: **2**) It is a list of loads that are delayed in the simulation used to delay the filling of bottles and blisters until they reach the number of pills V_dim_B
- **OL_cartoning** (Dimension: **2**) It is a list of loads that are delayed in the simulation used to delay the filing of boxes of finished bottle or blister until they reach the number specified in the CSV.

2.5. Label

- ***LB*** *(Dim:2): it is used in the process* ***P_ship*** *to graphically show in LB(1) "A box of type:",procindex , "delivered to destination after time:", ac "* and then it shows the V_mean_B in seconds. *In LB(2) "Total number of boxes delivered:",Tot_B, "with average time of:", (ac/(Tot_B*60)), " min "*.

2.6. Resource

- **R1** (Dim: 1): It is used in process P_powder works with a normal distribution with a mean of 30 seconds and a standard deviation of 3 seconds. Graphically it is positioned at the start of the conveyor at the stain station, its job is to move the powder batches from the Q_onto the conveyor.
- **Rin** (Dim: 1): Used only for the first simulation in the P_powder process, this operator works following a normal distribution with a mean of 30 seconds with a standard deviation of 3 seconds. Graphically, it is positioned between station sta_mix and sta1. Its task is to move the incoming powder and place it in the mixing machine.
- **R_b_fill** (Dim: 2): These operators are used in the P_conv process and work according to a normal distribution with mean (10*V_type) and standard deviation of 3 seconds. They are positioned next to the relevant sta_fill station and their task is to place the empty bottles or blisters into the adding machine for filling.
- **R_b_cart** (Dim: 2): These operators are used in the P_conv process they work following a normal distribution with a mean of 30 seconds with standard deviation 3 seconds, they are positioned between the Q_B and Q_cart queues and their task is to place the completed bottles and blisters inside the final box that will be shipped.
- **R_mach** (Dim: 1): This machine is used in the P_ship process and works according to a normal distribution with a mean of 20 seconds and standard deviation of 3 seconds, it is positioned in the conveyor at the common point to the boxes of the different products, its task is to label the finished packages.
- **R_out** (Dim: 2): The operators for the P_ship process are only two in the first simulation, thereafter R_out has size 1. They work with a normal distribution with a mean of 20 seconds with standard deviation 3 seconds, the task performed is to make sure the boxes are closed and load them onto the forklift

2.7. Queue

- **Q_rawmaterials** (Dim: 1 *; Capacity: 1*): It is used in the process **P_powder** and it is used to contain raw powder, empty bottles and blisters
- **Q_in** (Dim: 1 *; Capacity: 1*): It is used in the process **P_powder** and it is used to contain the raw powder brought in by the forklift before being placed on the conveyor.
- **Q_inB** (Dim: 2 *; Capacity: Infinite*): they are used in the process **P_rawB** and they are the two queues containing respectively the raw bottles and raw blisters brought by the forklift.

- **Q_B** (Dim: 2 *; Capacity: Infinite*): they are used in the process **P_conv** and these contain bottles/blisters filled with ten pills each.
- **Q_cart** (Dim: 2 *; Capacity: Infinite*): they are used in the process **P_conv** and these contain the number of finished bottles or blisters that will go into the CSV order box.
- **Q_out** (Dim: 2 *; Capacity: 1*): they are used in the process **P_ship** and these queues contain the finished and labeled box ready to be collected by the forklift.
- **Q_out_bb** (Dim: 2 *; Capacity: Infinite*): they are used in the process **P_ship** and They are the last queues in the process and are used to collect the finished products brought by the forklift into the depot.

2.8 Processes

L_dummy ⇨ P_read ⇨ P_powder ⇨ P_rawB ⇨ P_mixing ⇨ P_conv ⇨ P_ship

P_read (Dim:1; Default Traffic Limit: Infinite)

```
begin

// Input file opening

open "dir/input.csv" for reading save result as V_pointer //Automod  v14  or greater


read V_useless(1),V_useless(2),V_useless(3),V_useless(4),V_useless(5) from V_pointer with delimiter ";"


// Reading each row of the input file (WHILE LOOP STRUCTURE)


while 1=1 do

        begin


        read V_id, V_time, V_type, V_dim, V_dest  from V_pointer with delimiter ";" at end


                begin

                        print " My input file is fully read" to message

                        send to die

                end
```

```
        wait for (V_time - ac) sec
        clone V_dim to P_powder
        end
end
```

P_powder (Dim:1; Default Traffic Limit: Infinite)

```
begin

        set load type = L_powder
        move into Q_rawmaterials
        move into forklift.raw_materials_in

        travel to forklift.rawpowder_drop
        move into Q_in
        wait for 10 sec
        move into conv.stain
        use Rin for normal 30,3 sec

        travel to conv.sta_mix
        use R_mix for normal 30,3 sec

        clone to P_rawB(V_type)

        clone 10 load to P_mixing

end
```

P_rawB (Dim:2; Default Traffic Limit: Infinite)

```
begin

    set load type = L_rawB(procindex)
    move into forklift.raw_materials_in
    travel to forklift.b_drop(procindex)
    move into Q_inB(procindex)

end
```

P_mixing (Dim:1; Default Traffic Limit: Infinite)

```
begin

    set this load type = L_mixpowder
    wait for 10 sec
    move into conv.sta1
     send to P_conv(V_type)

end
```

P_conv(Dim:2; Default Traffic Limit: Infinite)

```
begin

    move into conv.sta1
    travel to conv.sta_B(procindex)
    wait for 10 sec

    use R_b_fill(procindex) for normal (10*V_type),3 sec
    move into Q_B(procindex)
```

```
inc V_count_pills(procindex) by 1

if V_count_pills(procindex) < V_dim_B

        begin
                wait to be ordered on OL_b(procindex)
        end
else
        begin
                order (V_dim_B -1) loads from OL_b(procindex) to die
                wait for 5 sec
                set V_count_pills(procindex) to 0

        end

set this load type = L_rawB(procindex)
use R_b_cart(procindex) for normal 30,3 sec
move into Q_cart(procindex)
inc V_count_cartoning(procindex) by 1

if V_count_cartoning(procindex)  < V_dim

        begin
                wait to be ordered on OL_cartoning(procindex)
        end
else
        begin

                order (V_dim-1) loads from OL_cartoning(procindex) to die
                wait for 5 sec
```

```
                    set V_count_cartoning(procindex) to 0
                    move into conv.sta_C(procindex)

            end
 send to P_ship(V_dest)

end
```

P_ship(Dim:2; Default Traffic Limit: Infinite)

```
begin

    set this load type = L_B(procindex)
    travel to conv.sta_mach
    use R_mach for normal 20,3 sec
    travel to conv.sta_out(procindex)
    use R_out(procindex) for normal 20,3 sec

    move into Q_out(procindex)
    move into forklift.bb_in(procindex)

    travel to forklift.bottle_blister_out(procindex)

    move into Q_out_bb(procindex)
    inc Tot_B by 1
    print "A box of type:",procindex , "delivered to destination after time:", (ac/60), " min"  to LB(1)
    print "Total number of boxes delivered:",Tot_B, "with average time of:", (ac/(Tot_B*60)), " min "
     to LB(2)

end
```

2.9. Vehicles

➢ ***forklift***(Number of Vehicles:2, Vehicle start list: first_list): Used in the pathmover. It starts in the *first_list* which contains the parking control point *cp_park*. If it finds itself in some control points grouped in the *work_list* (as *cp_park,* the entry control point of the workstation *cp_work_in* or the exit control point *cp_out*) it does retrievals to the *pickup_list* (which contains the entry control points *cp_in* and the exit control point of the workstation *cp_work_out*) and then it does deliveries to the *cp_work_in* or *cp_out.* If it finds itself in some control points grouped in the *park_list* (which contains *cp_out* and the entry control point of the workstation *cp_work_in*) it goes to park in *cp_park.*

2.10. Lists

➢ ***NamedList*** which contains two lists:

○ ***first_list*** (List items: *cp_park*)
○ ***pickup_list*** (List items: *bb_in_1, bb_in_2, raw_materials_in_3*)

➢ ***WorkList*** which contains:

○ ***cp_park*** (Type: At, Name: pickup_list)
○ ***b_drop_1*** (Same As cp_park)
○ ***b_drop_2*** (Same As cp_park)
○ ***bottle_blister_out1*** (Same As cp_park)
○ ***bottle_blister_out2*** (Same As cp_park)
○ ***rawpowder_drop*** (Same As cp_park)

➢ ***ParkList*** which contains:

○ ***raw_materials_in*** (Type: At, Name: first_list
○ ***bottle_blister_out1*** (Same As raw_materials_in)
○ ***bottle_blister_out2*** (Same As raw_materials_in)
○ ***rawpowder_drop*** (Same As raw_materials_in)

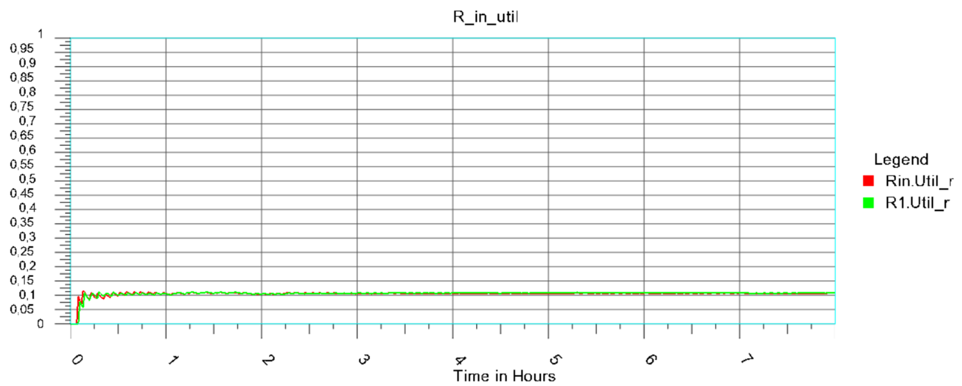
3. Simulation Results

First Simulation

Once the configuration phase of the first scenario was finished, we started to simulate the digital twin. For the simulation statistics, we considered the values at the end of an 8-hour work shift for all the scenarios performed. Our business graphs are updated with a frequency of 1 minute.

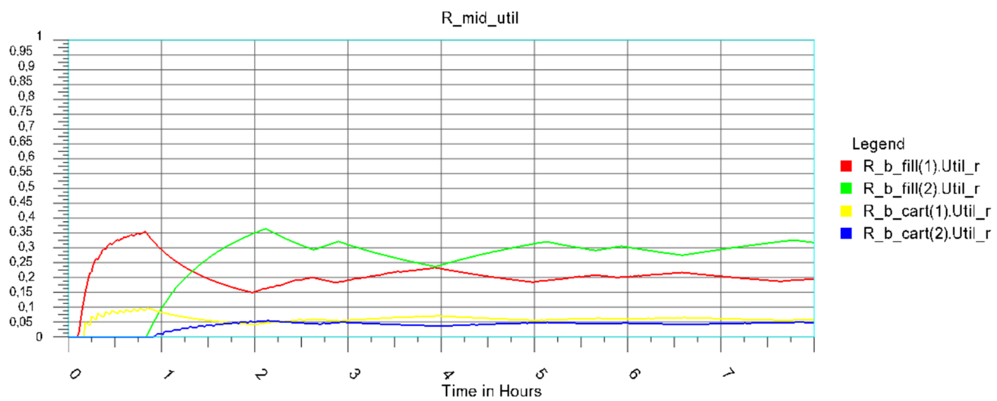
To analyse the performance of the model, as well as the feasibility of the improvements which could be made, we initially focused on the **resources utilization**, the business graphs below gives a clear understanding of the value. On the x-axis of these graphs there is the time in hours from 0 to 8 with a sign every 15 minutes, while on the y-axis there is a percentage from 0 to 1 that indicates the utilisation (Util_r) of the resources.

For the 2 operators Rin and R1 at the start of our conveyor, the business graph is the following:



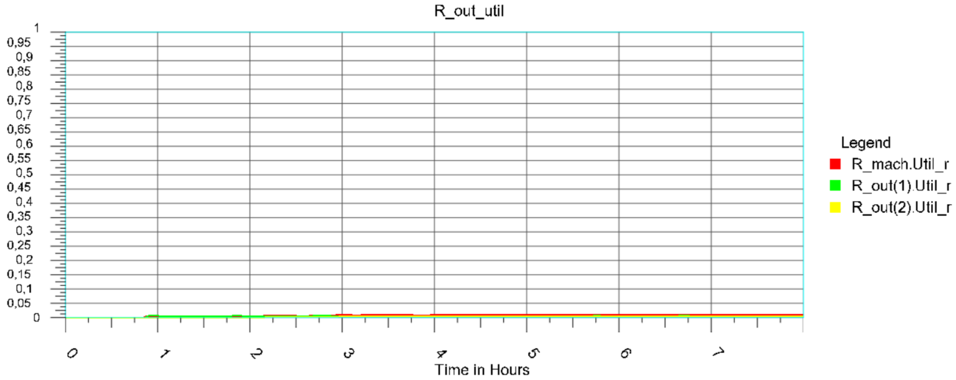
By reading this graph it is clearly visible that the two operators are balanced and their value of utilisation is constant throughout the 8 hours period.

For the 4 operators R_b_fill(1), R_b_fill(2), R_b_cart(1) and R_b_cart(2), located in the middle of the conveyor, the business graph is show in the figure underneath



Here it is visible that all the four resources are used almost in a balanced way for the types of work. In the first phase of the simulation, we can observe that only R_b_fill(1) and R_b_cart(1) are used and then the other two resources start to work. This happens because of the order list that we used for this simulation. After the start phase, the values of utilization are balanced again.

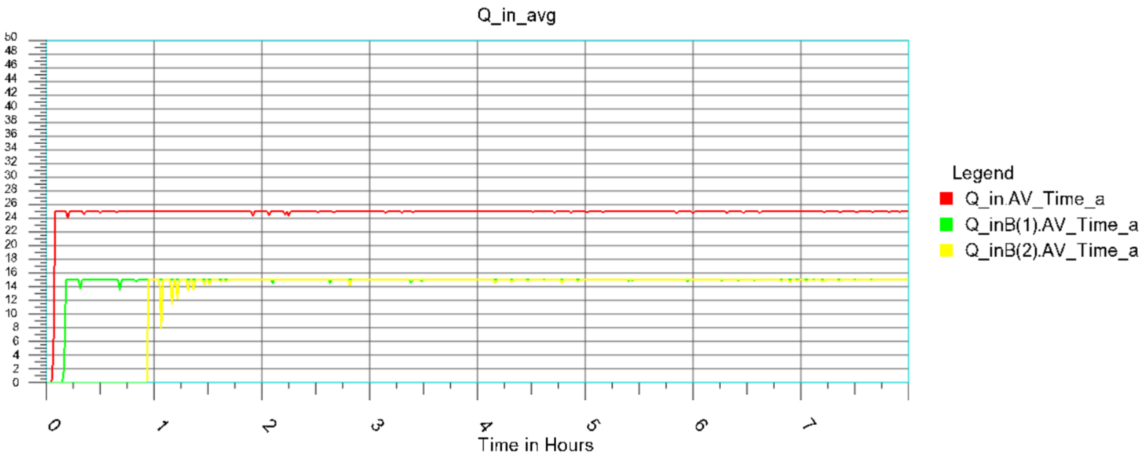
The final graph of the resource utilization shows the operators R_mach, R_out(1), and R_out(2) located at the end of the second part of the conveyor:



The graph shows a clear under-utilization of those resources; also, in the beginning, those resources don't work until the first box to be delivered is ready and that is linked to the order list that we have.

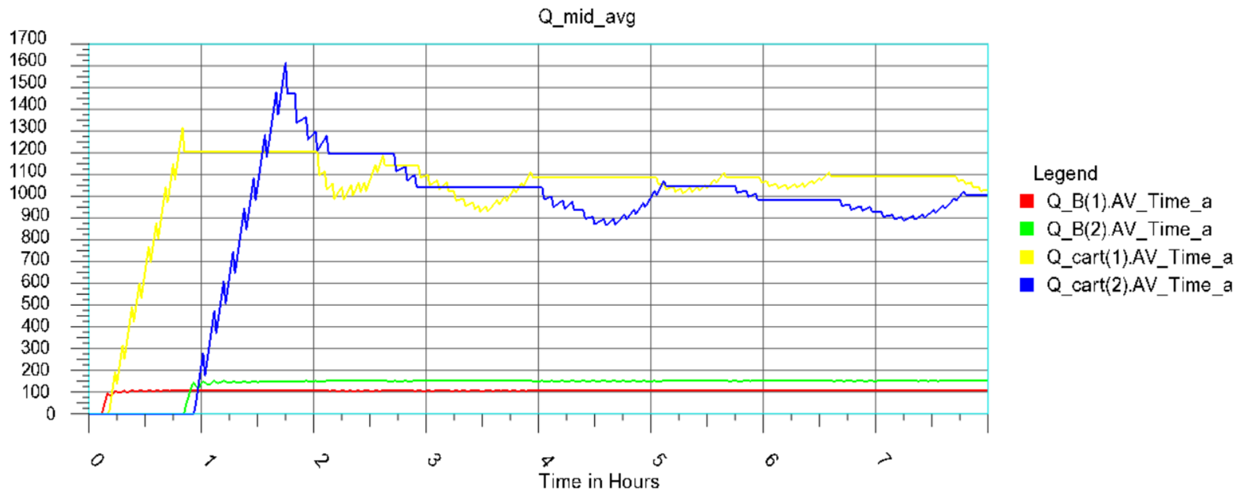
To proceed with the analysis, we focused on the **average time spent into the queue** using, as we did for resources, business graphs. On the x-axis of these graphs there is the time in hours from 0 to 8, with a sign every 15 minutes, while on the y-axis It is the average time spent in a specific queue, per each moment. The scale of value is in seconds and there is a frequency of update of 1 minute.

The first analysis is done for Q_in, Q_inB(1), and Q_inB(2), which are located in the first part of the system.



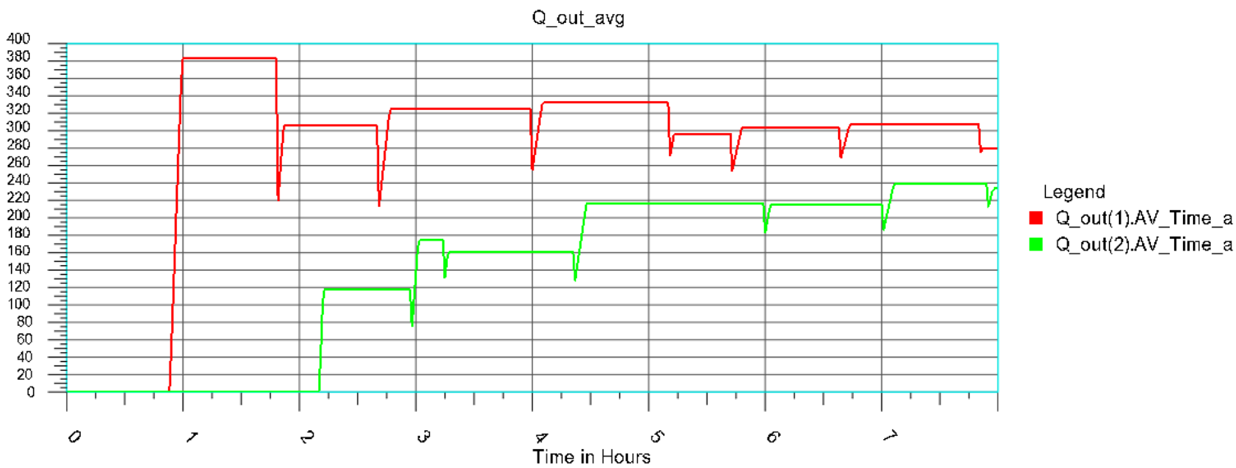
The average time spent in the queue is almost constant for all the time of the process, which happens because the dedicated operator promptly transfers the product to the next step as soon as it is finished.

The 4 queue under analyses are: Q_B(1), Q_B(2), Q_cart(1) and Q_cart(2) that are located in the middle of the conveyor, the business graph is show in the figure underneath:



The graph shows some interesting points. The Q_B(1) and Q_B(2) have a constant trend that happens because the frequency of work is constant. The Q_B(2) in the first 45 minutes does not have anything in the queue and that is attributable to the order list that we follow. The average time spent in the queue for the Q_cart(1) and Q_cart(2) is not constant because those work processes have to operate in relation to the order list request.

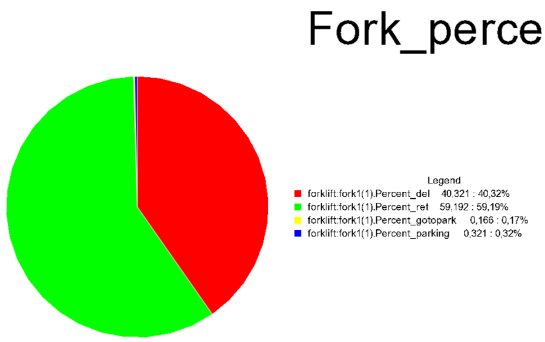
The final graph of the average time spent in the queue it shows the queue Q_out(1) and Q_out(2) located at the end of the second part of the conveyor:

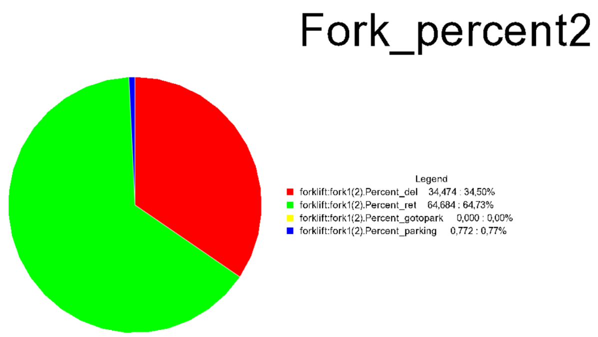


Based on the information provided by the graph, we observe that the queue Q_out_bb(1) began to be utilised after 50 minutes (the time required to complete the first order), after which Q_out(1) also commenced its use.The same reasoning is applied for Q_out_bb(2) and Q_out(2).

After examining the average time spent in the queue, we shifted our attention to the forklift usage and employed three pie charts for each analysis.

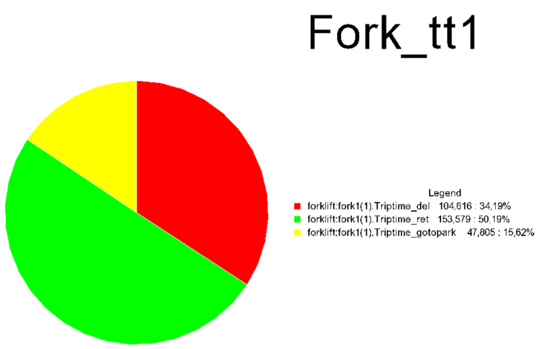
The first pie chart we looked at shows the percentage of time that the forklift spent on delivery (percent_del), retrieval (percent_ret), going to park (percent_gotopark) and parking (percent_parking). The business graph is displayed below:

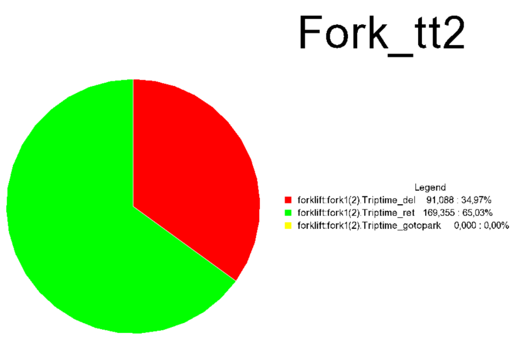




Delivery and retrieval are adding-value activities while gotopark and parking are not, and the first two together occupy more than 50% of the time. From this graph we can observe that, even if it's not optimised, it's balanced. We immediately noticed the extensive use of forklifts, which is due to the fact that the various product creation processes require the materials delivered by this.

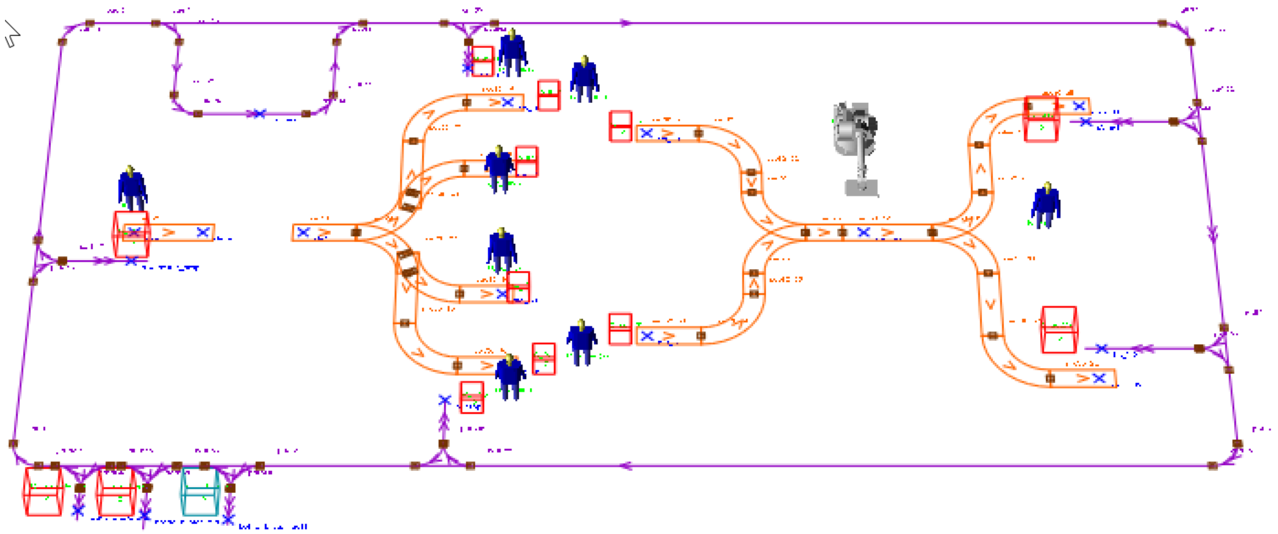
The second pie chart represents the **time spent by the forklift doing the trip on the path mover for the operation of *delivery, retrieval* and *gotopark***. The business graph is shown below:



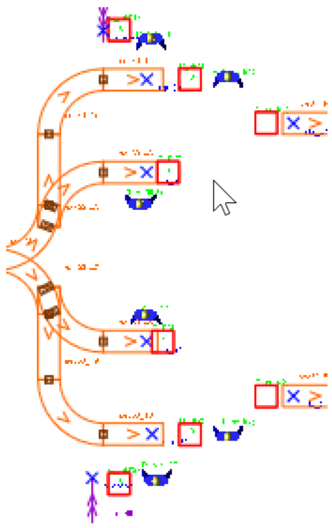


It is possible to see that the most heavy operation of forklift in terms of time is the delivery one.

Second Simulation



Analyzing the previous simulation it seems that resource utilization is unbalanced. The operators **R_b_fill** are the most utilized while operators **R_in R1** and both **R_out** are almost unutilized. Furthermore queues **Q_B** have the most average time in comparison with all queues. This average time can be seen as a WIP (Work in progress) cost therefore should be minimized. Following the company policy of not laying off employees, we thought of a way to redeploy human resources and at the same time reduce the clearly visible bottle neck formed immediately after ***sta1***. The pill hoarding is due to the fact that the operator in charge of the filling station **R_b_fill(1)**, cannot cope with customer demand.



In this simulation we decided to add two new conveyors in parallel, regarding the pill filling action. Considering the analysis carried out in the AS-IS scenario, we decided to reallocate the operator **R1** and one of the two operators to the shipping **R_out** and assigned them to the two new filling stations **sta_B3** and **sta_B4**.

The usage of vectorial processes, more specifically (procindex), ensures easier changes in the code to simulate the new configuration after rebuilding the plant.
In addition afterwards are presented all changes applied in the model:

- **Global variables**

➢ *V_count_pills* (Type: Integer, Dimension: **4**)

- **Resources**

➢ *R_b_fill* (Dimension: **4**)
➢ *R_out*(Dimension: **1**)

- **Queues**

➢ *Q_B* (Dimension: **4**, Capacity: Infinite)

- **Order list**

➢ *OL_b* (Dimension: **4**)

- **Processes**

➢ ***P_powder* (Dimension: 1, Default Traffic Limit: 1)**

```
[...]
if V_type == 1
        begin
                clone 10 load to P_mixing(nextof(1,3))
        end
if V_type == 2
        begin
                clone 10 load to P_mixing(nextof(2,4))
        end
[...]
```

○ ***P_mixing*(Dimension: 4, Default Traffic Limit: 1)**

○ ***P_conv*(Dimension: 4, Default Traffic Limit: 1)**

```
[...]
set this load type = L_rawB(V_type)
use R_b_cart(V_type) for normal 30,3 sec
```

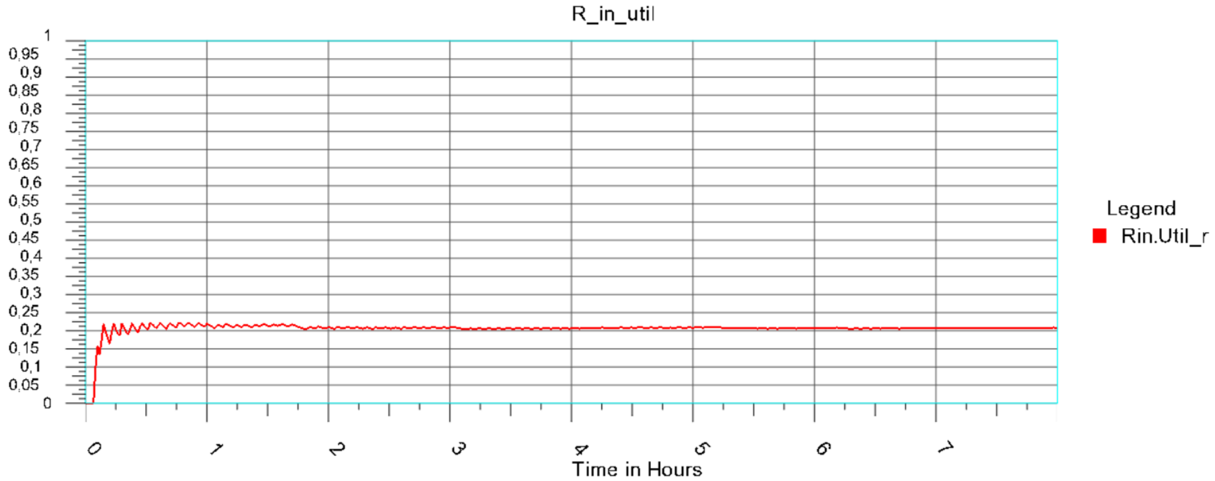
```
move into Q_cart(V_type)
inc V_count_cartoning(V_type) by 1
if V_count_cartoning(V_type)  < V_dim

    begin
    wait to be ordered on OL_cartoning(V_type)
    end
else
    begin
        order (V_dim-1) loads from OL_cartoning(V_type) to die
        wait for 5 sec

        set V_count_cartoning(V_type) to 0

    end
move into conv.sta_C(V_type)
[...]
```

To sum up the changes made in the process, the Automod function '*nextof(2,4)*' was used in **P_powder** which made possible the parallelization of the process **P_mixing** that in this configuration has dimensions 4. The '*nextof(.,.)*' whenever it is called keeps alternating the input value in the same order as they are written. This allows to uniformly clone the loads in **P_powder** between the parallelized conveyors.

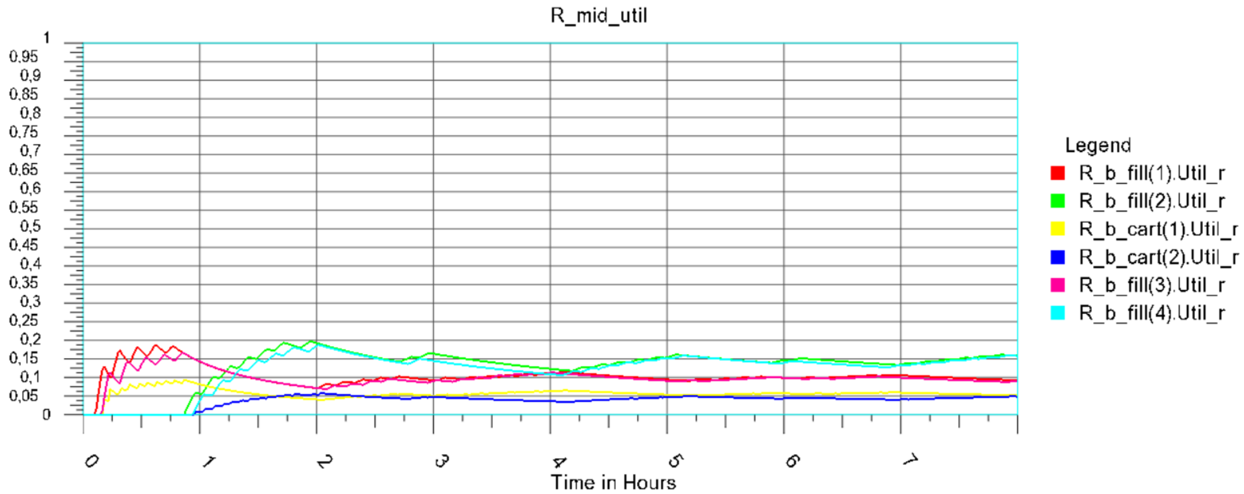
Furthermore all lines of code of **P_conv** in the first part remain equal to AS-IS configuration because they have dimension 4 and use the procindex that was sent from the previous process **P_mixing.** The second part of **P_conv** refers to entities that have dimension 2 and therefore instead of using **procindex** that ranged from 1 to 4 it was necessary to use **V_type**.
These small changes in the code are due to the decision to make the whole process **scalable**, so that there are no difficulties if future changes in the production are made.

The same business graphs as before were used to evaluate the improvement of the simulation

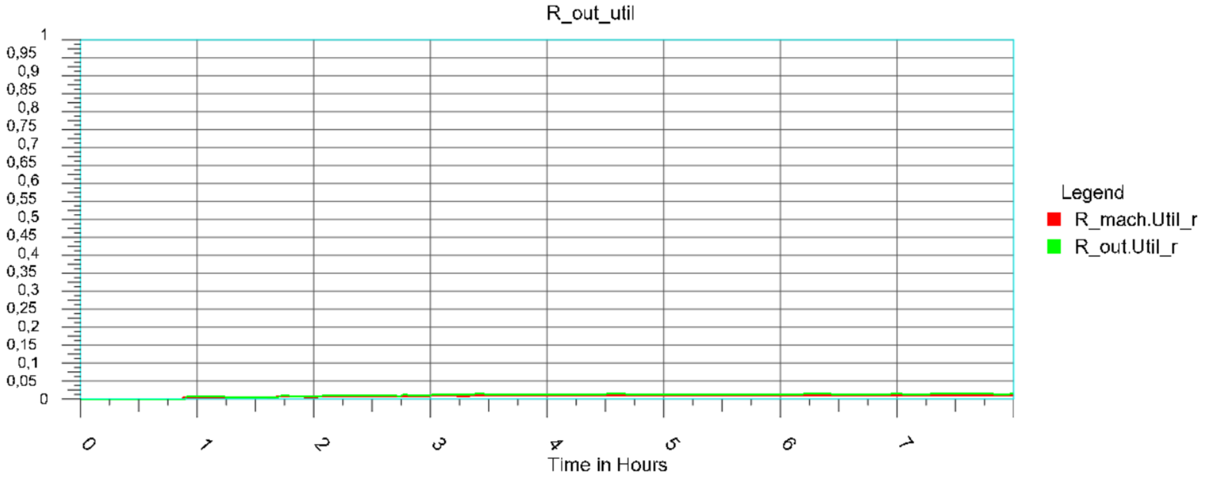
The first graph that we want to analyze is the Resource utilization. We note that having removed the **R1** operator, the **R_in** operator now works with a utilization of 20%.



The change of work station for the operator R1 has brought a significant reduction of utilization in the next phase of the process, which is appreciable as shown by the second graph.

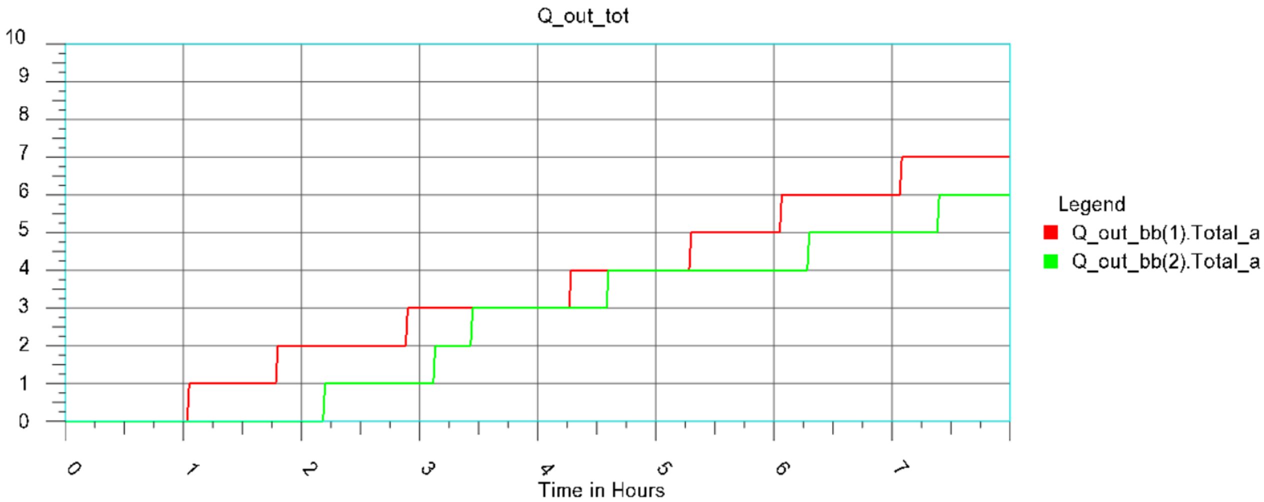


Looking at the operators working in the conveyor part in between, we can appreciate how the redeployment, rather than dismissal, of operators has balanced the workload. We no longer have operators employed at 35%, but they are all around 15%.

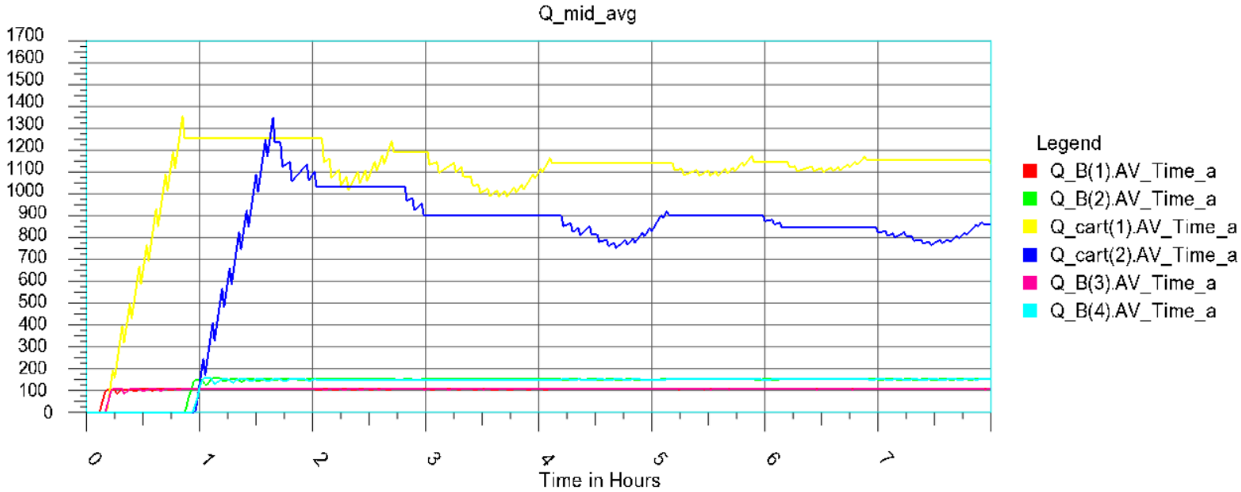


Furthermore, the removal of the **R_out(2)** operator, as can be seen from the graph, does not lead to any worsening, since the output products are sporadic, utilization still remains very low.

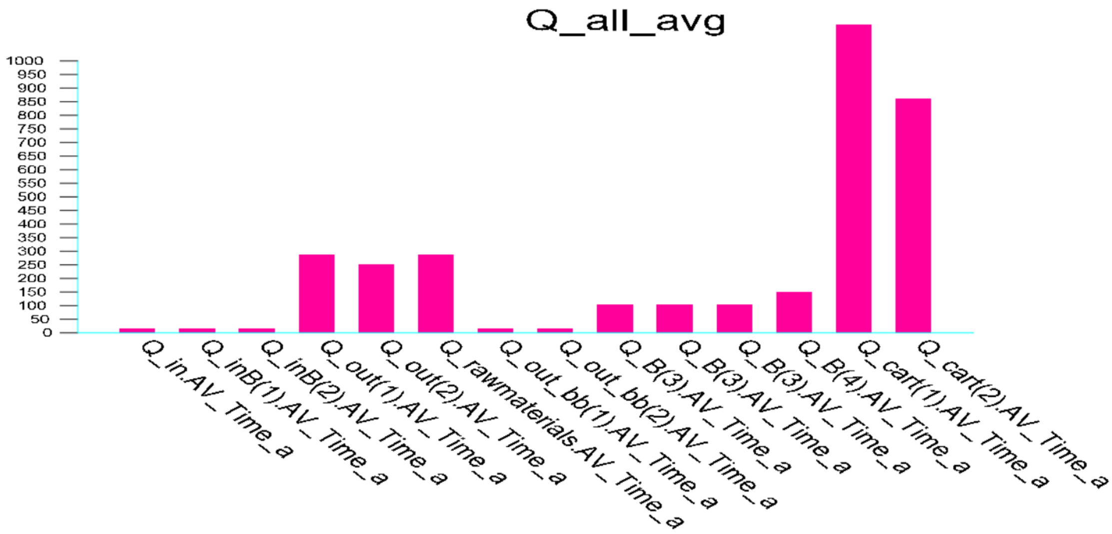
Despite a better utilization of resources, we note a deterioration in total output. The explanation can be found by considering that the streamlining of the bottleneck is not adequately supported by a delivery system for empty bottles and blisters. These arrive at too slow a pace compared to the pills. This deterioration can also be seen in the following graphs.



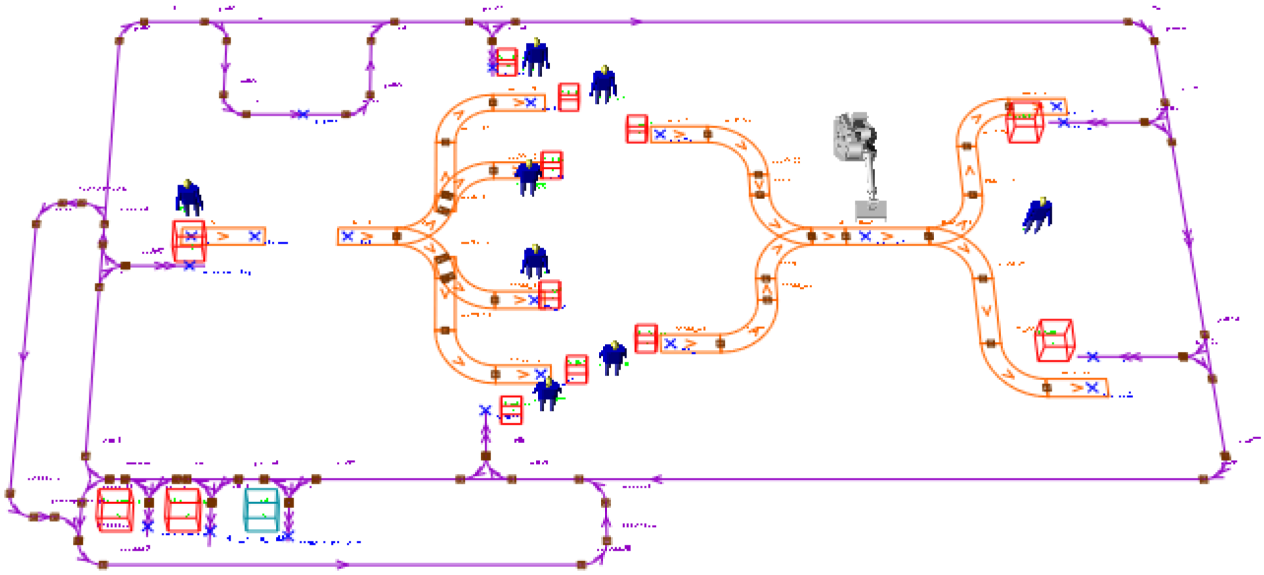
The increase in Queue average time for the cartoning process is precisely due to the inefficiency of forklifts in delivering the materials: as there are no empty bottles/blisters, the boxes take longer to be completed, thus waiting to be finished, slowing down production.



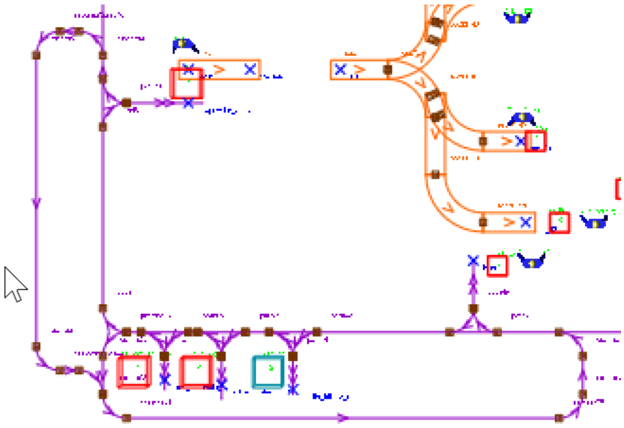
To have a complete analysis of all the queues in the system, histogram was used to understand the average time spent in the queue for all queues in the system. The x-axis represents all the queues in the system, while the y-axis represents the time (in seconds) spent inside each queue.



Third Simulation

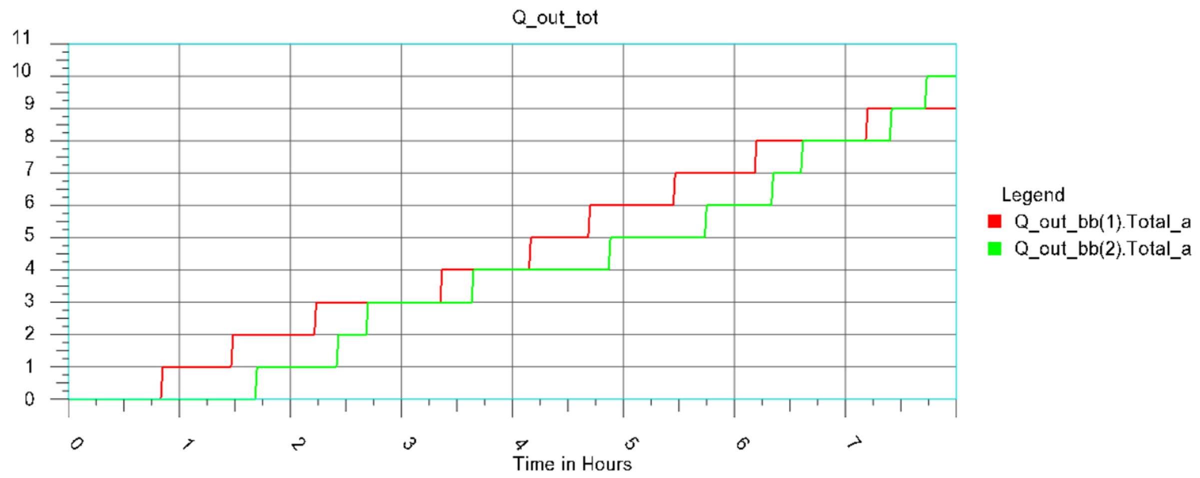


Noting the increase in the average queue time for cartoning operations, we realise that having paralysed stations allows production at a higher rate, but at the same time requires a higher rate of delivery of empty bottles and blisters, which for the current state of forklifts is unsustainable, we decided to check whether improving our system from the forklift point of view makes sense to keep the parallelisation.
This simulation was centralized to improve forklift utilization and therefore queues average time.



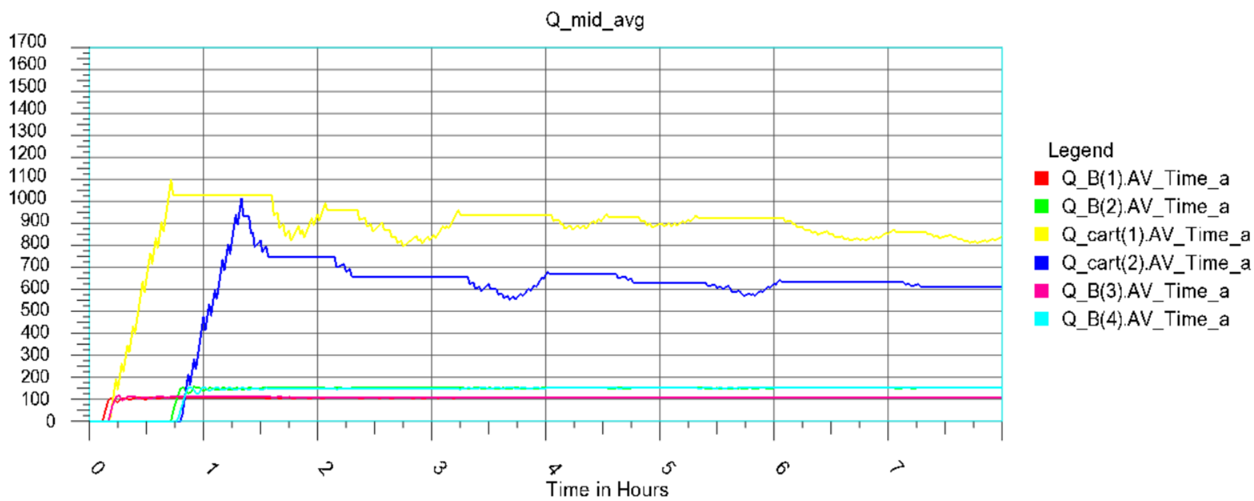
The decision was to introduce another forklift path that let them return back to the depot faster with shorter travel distance.
The route was shortened for two operations that are crucial to the efficiency of the company, the delivery of raw materials, in particular powder and blisters.

Similar to previous simulation the evaluation was made plotting business graphs.
The first improvement we want to highlight with this simulation is the increase in finished products, which is remarkable. We have improved from the previous situation where the values for Q_out_bb(1) and Q_out_bb(2) were 6 and 7, respectively. The current values for Q_out_bb(1) and Q_out_bb(2) are 9 and 10, respectively.

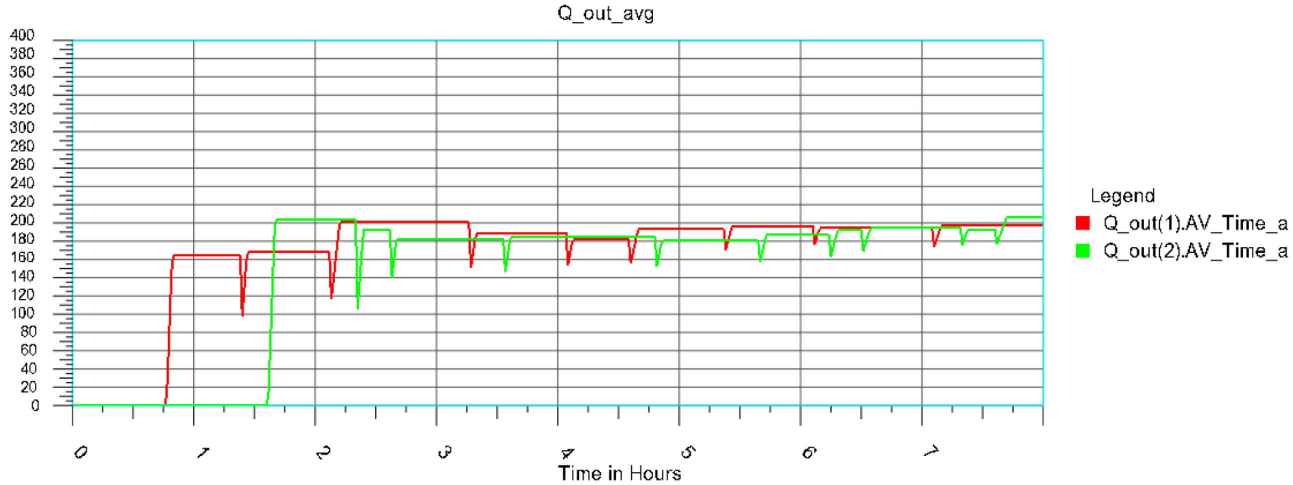


We also note that the problem highlighted in the previous simulation, i.e. the average queue time for the mid operations, also has an appreciable improvement here.

The graph shows the trend of the six queues under focus. It is evident that there is a reduction in the values of Q_cart(1) and Q_cart(2). This reduction is due to the better efficiency of the forklift, which has reduced the time taken to pick up raw materials

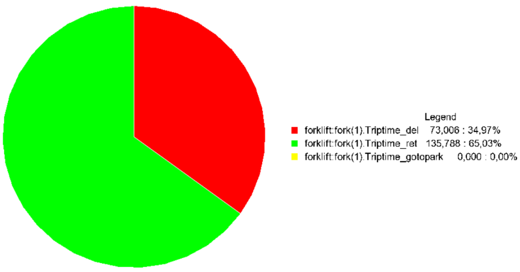


By taking less time to deliver raw materials, forklifts are freed up more quickly for the activity of delivering and retrieving the finished package, this fact is clearly highlighted by the reduction of the average queue time in the ending zone, where forklifts pick up boxes of finished products. We go from an average stay of about 300 seconds in the first simulation to about 200 seconds in this simulation

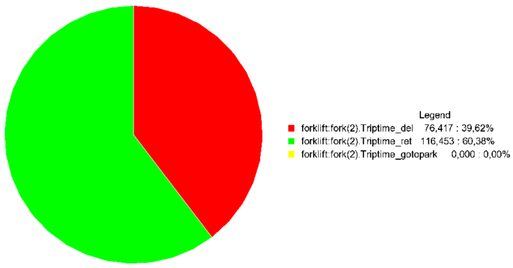


From consideration made, adding a shortcut path to the path mover will improve trips time results from the previous configuration.

Fork_tt1



Fork_tt2

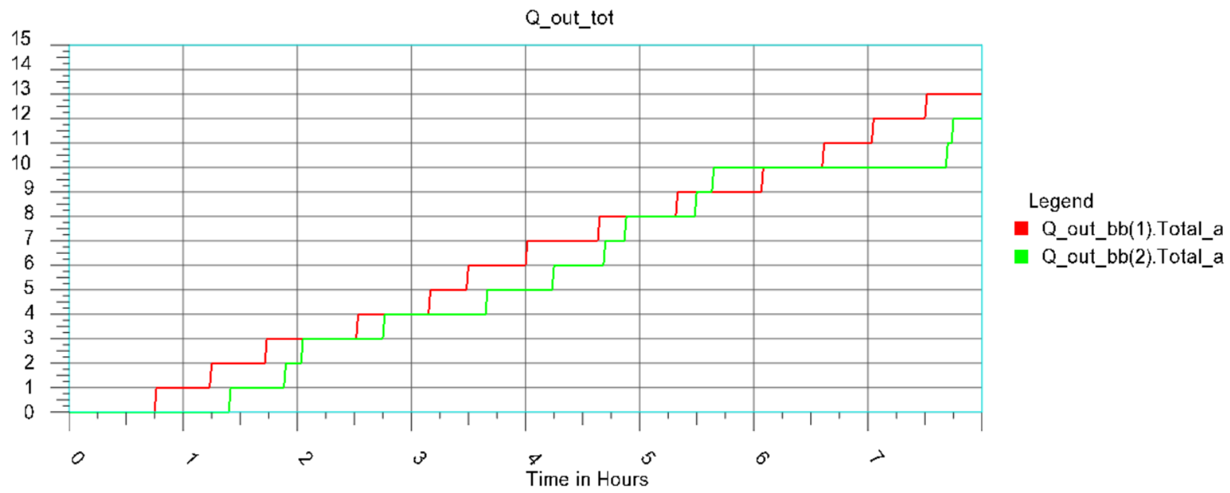


In fact this was confirmed from the pie charts of the total trips, by reducing the route for the retrieval of raw powder and raw blister, we achieved a significant reduction in the total time of retrievers.

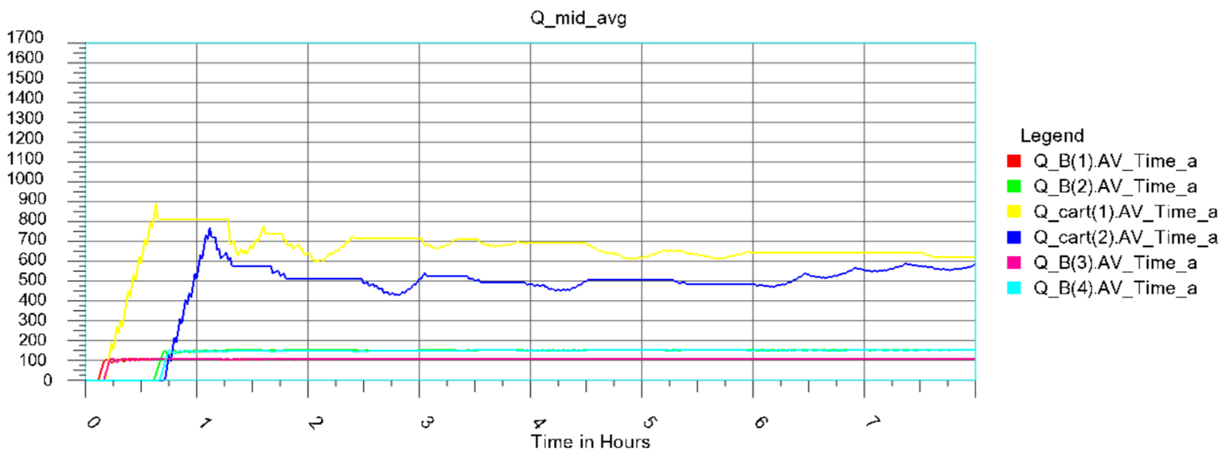
The last simulations are connected with the fundamental decision of finding the optimal number of forklift to use.

Fourth Simulation (3 Forklift)

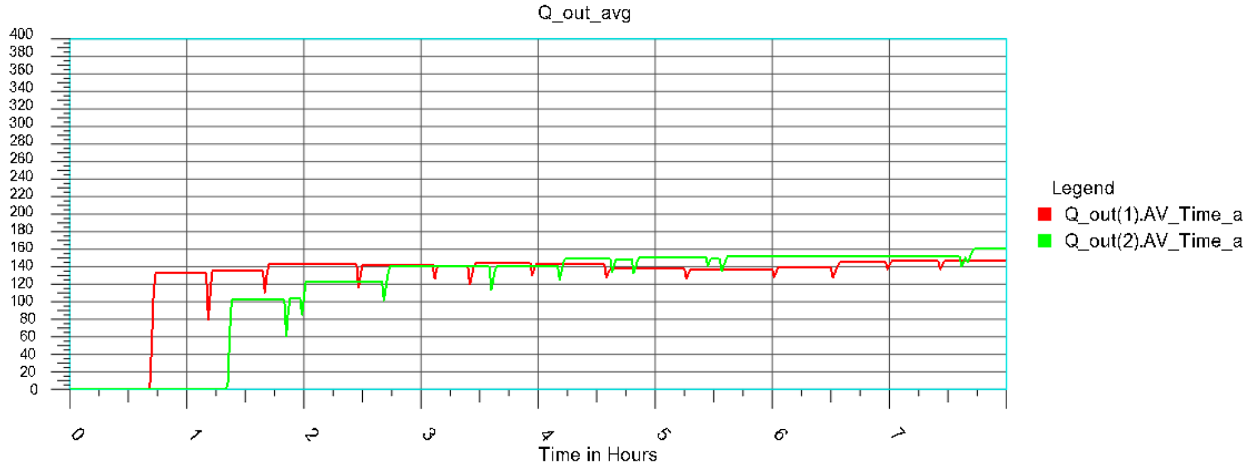
With the fourth simulation, noting a clear dependence of our system on forklifts, we focused our attention on the number of forklifts. Considering the continuous demand for raw materials, and also knowing from previous simulations that production has difficulty keeping up with deliveries of empty bottles and blisters it would not make sense to decrease the number of forklifts, so our choice was to add one.
This change involved only one modification at system level, no changes were made to the code.
As in the previous simulation, the first and very appreciable improvement was an increase in production of as many as six finished products in one shift, the total number of finished products now amounting to 25 that can also be seen in the following graph.



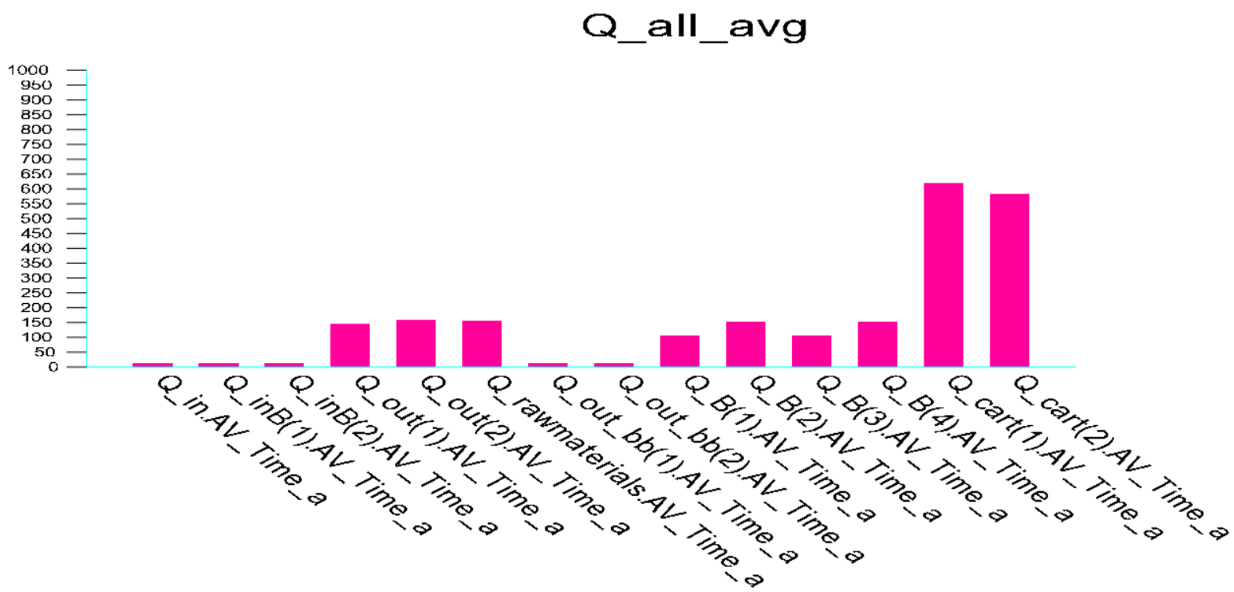
Another improvement we find by adding a forklift is the average time in the queues, this undergoes a further improvement (albeit less than between the second and third simulation).
The graph shows that we managed to reduce the average time in Q_cart(1) and Q_cart(2) by nearly 1000 seconds for the Q_cart(2) peak and 500 seconds for the Q_cart(1) peak. The difference is also substantial for the steady-state values in the hours when both Q_cart(1) and Q_cart(2) drop by around 500 seconds relative to the initial case and by around 200 seconds relative to the unoptimized forklift case.



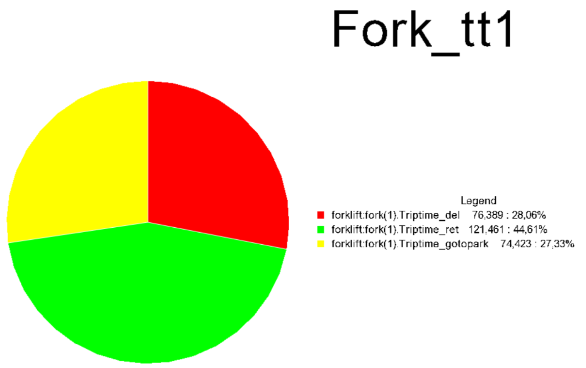
A similar consideration to that made in the previous simulation can also be made for this simulation: having more forklifts, they are freed more quickly by the retrieving of raw materials, the dwell time of the finished products in the queues is lowered even further reaching the excellent result of approximately 160 seconds.

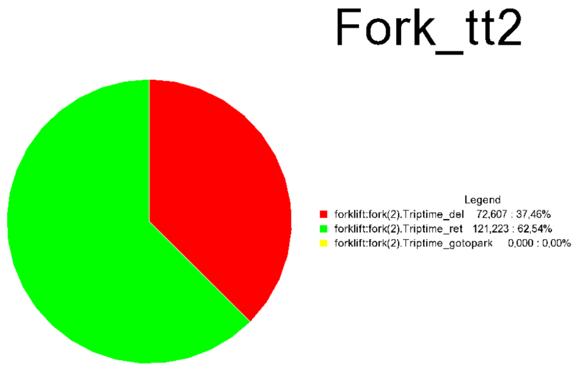


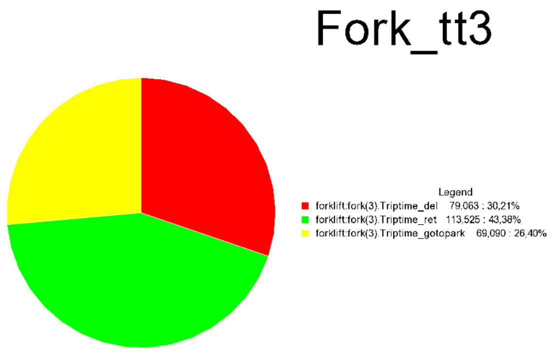
To obtain a comprehensive overview of the queues in the system, a histogram was utilized to determine the average duration spent in each queue. The horizontal axis of the histogram represents all the queues in the system, while the vertical axis represents the time (in seconds) spent inside each queue



In order to evaluate the utilization of the added forklift the following pie charts were used







In analyzing the data and charts from previous iterations, it is evident that there has been an increase in the percentage of 'go to parking.' This shift indicates a decrease in forklift utilization compared to the earlier simulation runs. However, the introduction of a third forklift has resulted in a higher number of finished product output.

Discussion

Following the sum of all modifications made in the previous simulation models and the adjustment of the optimal number of forklifts (also in relation to the investment cost) for the system, we were able to achieve a significant improvement of 66% more boxes compared to the starting configuration and a 30% improvement compared to a configuration with a non-optimal number of forklifts.

In order to visualize all simulation a table with the labels for every configuration was used

Simulation Number	Finished orders (Tot_B)	Avg Time between finished orders (V_mean_B) [min]	Configuration settings
1	15	31.94	AS - IS 2 Fork
2	13	34.10	Parallelo (2 Fork)
3	19	24.40	Parallelo + path (2 Fork)
4	25	18.59	Parallelo + path (3 Fork)
5	29	16.43	Parallelo + path (4 Fork)

The most valuable configuration in terms of all KPIs considered forklift cost and utilization, WIP cost and resource utilization we came to the conclusion that the results obtained with the fourth simulation allow for an excellent production rate with a number of forklifts taking into account the costs associated with the purchase of new forklifts.

This configuration implements the resources balance and the forklift path added and used three forklifts.

To sum up, the simulations have emphasized that it is essential to take a comprehensive view of the production system, moreover keeping the whole simulation scalable, it can be used for future improvements or in case of production changes. The redistribution of the operators can help improve resource utilization, but it is equally important to address inefficiencies in the supply chain, particularly with regard to material delivery. Optimizing forklift paths and relocate resources have been shown to be effective strategies for maximizing production and reducing wait times. These results underscore the significance of flexibility and continuous efforts to improve the production system, so that it can respond efficiently to dynamic market challenges.

In conclusion the manufacturing plant implementing this improvement will produce 25 orders in a 8 hours shift, which is actually a feasible amount of products in order to complete the daily order scheduled, in the csv file.