Work Group 3 (Group 13)

Manufacture of metallic chairs

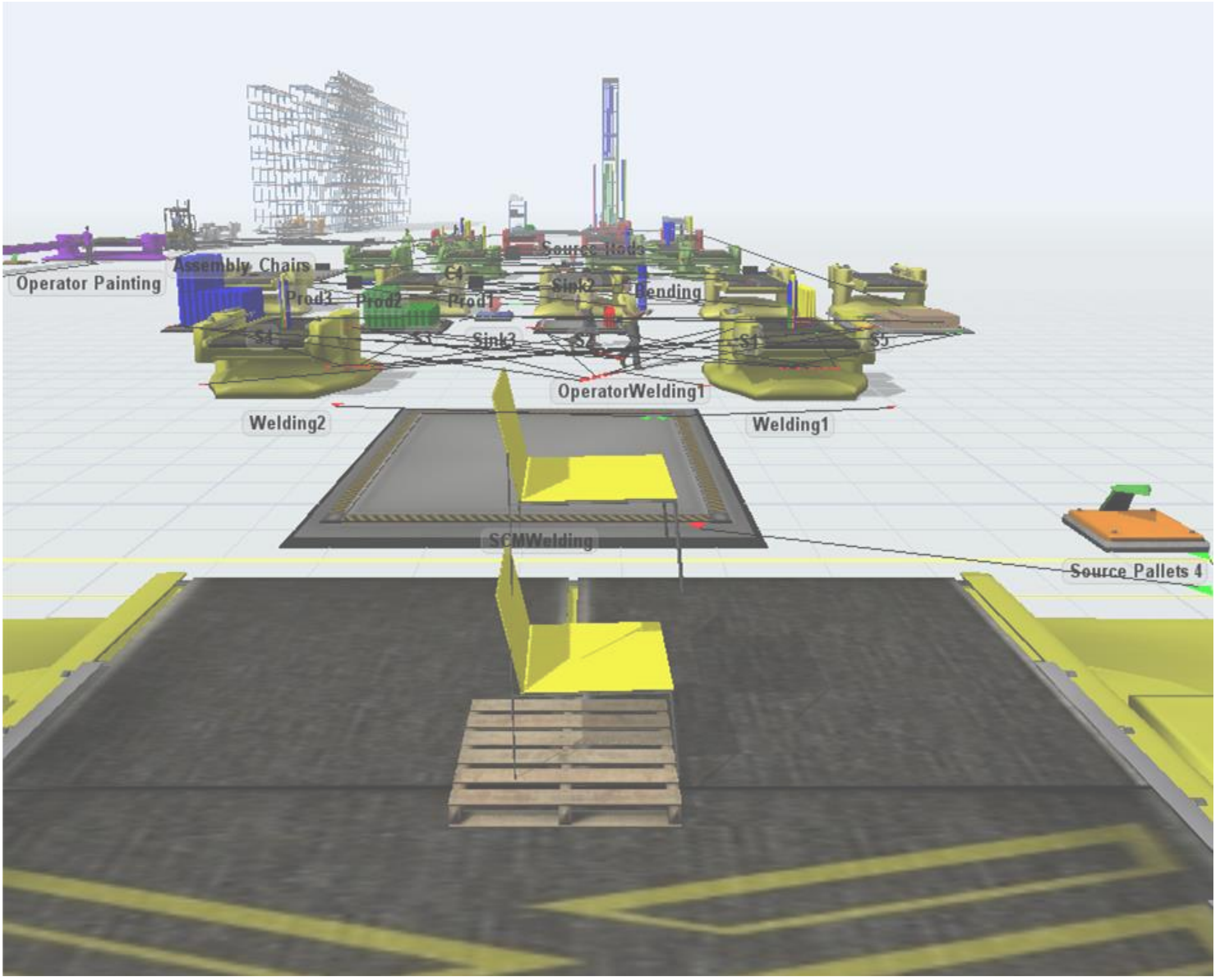
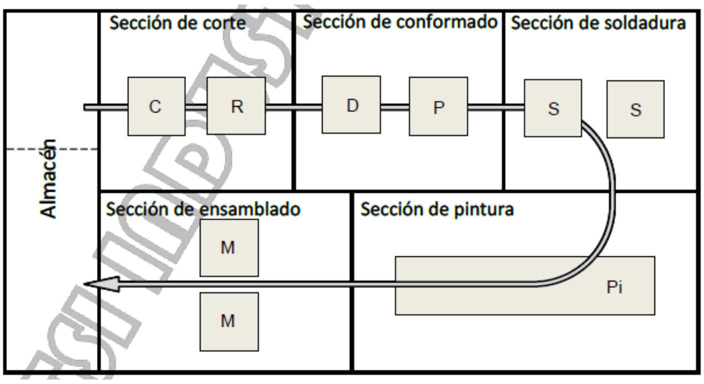


TABLE OF CONTENTS

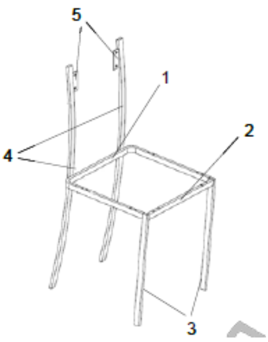
1. CASE DESCRIPTION

Our team has been assigned with the manufacturing of metal chairs (case description 6.2). A layout of our processing line and the different stages is shown in the image below:



The whole process can be explained in two different sections:

- Through the assembly, forming, welding and painting stages, the frame of the chair is made. This structure consists of 5 different elements: seatU, crossbar, frontlegs, backlegs and wings. The different elements and their section and lengths are shown in the image below.



Mercurio				
Pieza	Denominación	Cant/silla	Sección	Longitud
1	Asiento U	1	20X10	940
2	Travesaño	1	20X10	360
3	Pata	2	18X18	440
4	Pata-respaldo	2	18X18	840
5	Alas	2	30X2	70

- Later, at the assembly section, the frame of the chair is combined with the seats and backrests to finish the product.

The times of these processes are listed in the following table.

Tiempos de producción				
Componente	Asiento U	Travesaño	Pata	Pata-respaldo
Cortado				
Tiemp prepa.	30	30	30	30
Carga	6	6	6	6
Cortado	4	4	4	4
Descarga	1	1	1	1
Repasado				
Carga	4	4	4	4
Repasado	6	5	5	6
Descarga	3	3	3	3
Doblado				
Tiemp prepa.	70	70	70	70
Carga	2			2
Doblado	7			4
Descarga	2			2

Punzonado				
Tiemp prepa.	50	50	50	50
Carga	5	5		
Punzonado	7	4		
Descarga	2	2		
Soldadura				
Tiemp prepa.	40			
Carga	6	4	6	8
Soldadura	45			
Descarga	3			
Pintado				
Tiemp cambio	600			
Carga	4			
Pintado	120			
Descarga	3			
Montaje				
Carga	7			
Montaje	50			
Descarga	3			

1.1. Parts needed to complete the required production levels.

To assess the demand for the production of chairs, our objective is to make 600 units: 300 grey chairs, 200 silver chairs and 100 white chairs.

For each chair, we have to use different elements. The wings, seats and backrests are outsourced and arrive to our processing line ready to be assembled. On the other hand, the rest of the elements have to be manufactured from scratch using 6m long rods. Using the table below, we calculate the rods that we are going to need to finish the process.

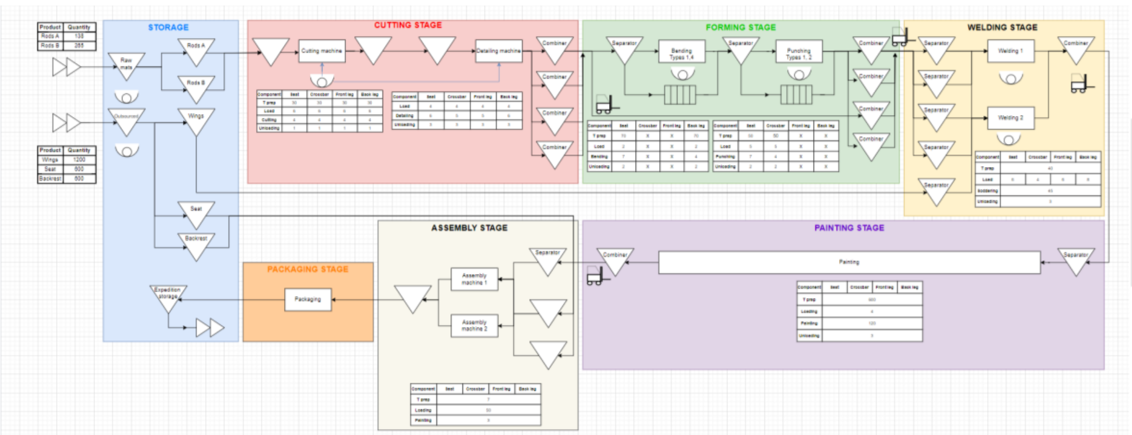
Element	Quantity/chair	S (mm^2)	L (m)	Lrod (m)	Lrod/L	Items/rod	Total rods
Seat U	1	20x10	0.940	6	6.38	6	100
Crossbar	1	20x10	0.360	6	16.67	16	38
Front Legs	2	18x18	0.440	6	13.64	13	93
Back Legs	2	18x18	0.840	6	7.14	7	172
Wings	2	OUTSOURCED					
Seat	1						
Backrest	1						

Therefore, the products sourced to our manufacturing line are going to be:

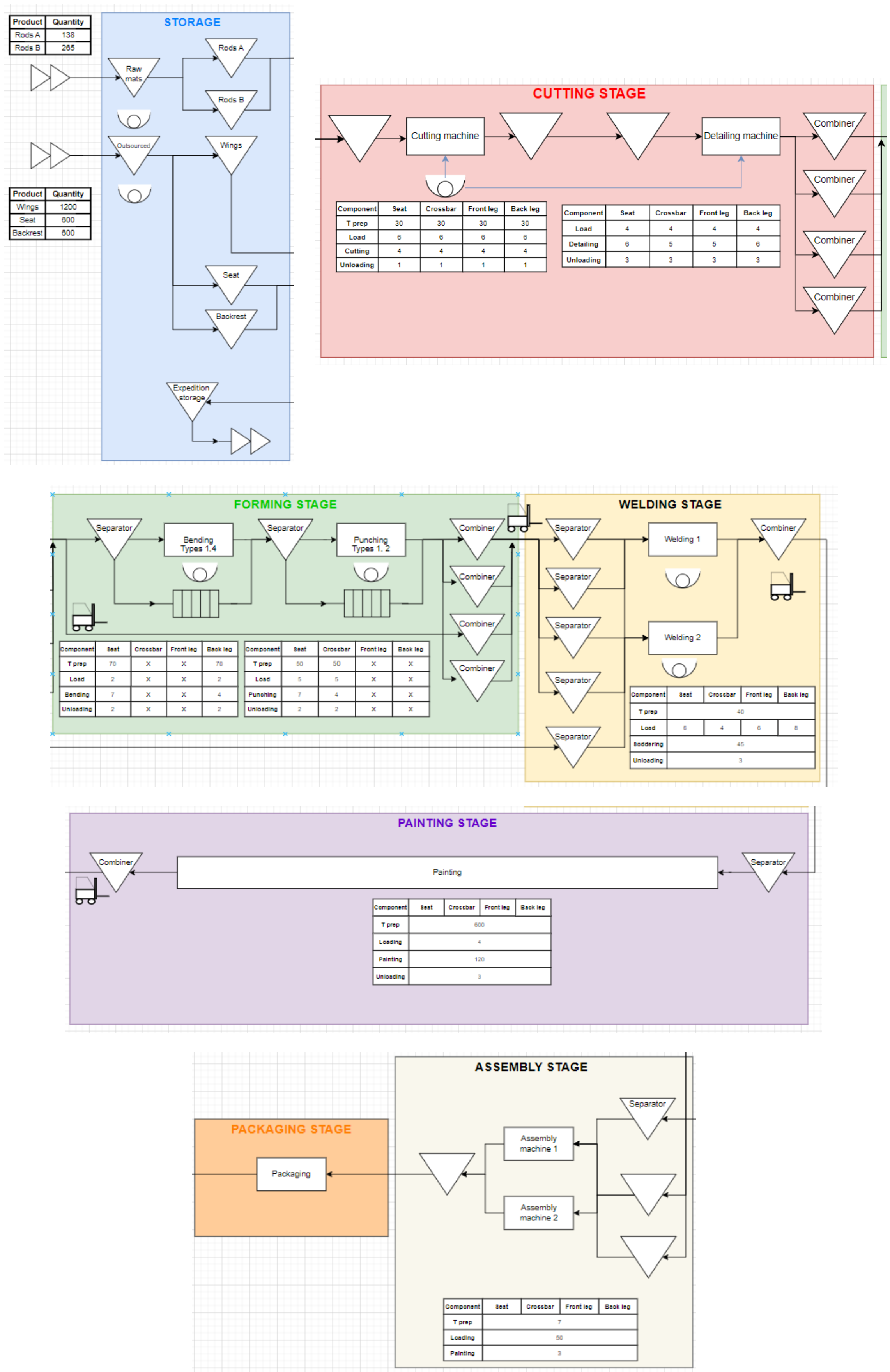
Element	Quantity needed for total production
Rods A (20x10)	138
Rods B (18x18)	265
Wings	1200
Seats	600
Backrests	600

2. VSM

Once we have a better understanding of the case description, we can design a VSM that we will later use to implement the process in FlexSim.



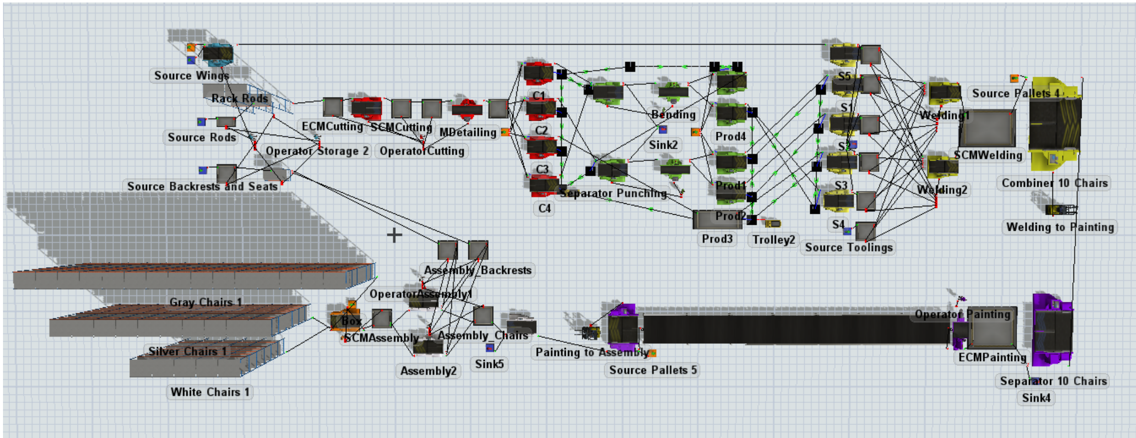
The VSM is color coded, each color represents a different stage of the manufacturing line. We can take a closer look at the different stages:



It is important to take into account that the times given in the case description and shown in the VSM have a variation of 10%, which will be implemented into the FlexSim model.

3. FLEXSIM MODEL: EXPLANATION OF CONFIGURATION

The processing line designed in the VSM is implemented in FlexSim. The image below shows a screenshot of said processing line:

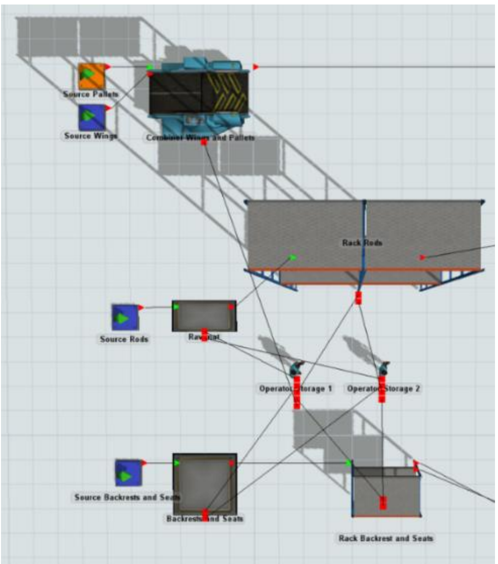


As we can see, the machines, combiners and separators in the different stages follow the same color code as in the VSM, allowing for a better understanding of the process. A trajectory is set for the trolleys in between the cutting and forming sections so that the trolley doesn't run in the way of the operators handling the bending and punching machines. As for the forklift, it wasn't necessary to implement any trajectories as they don't come close to the working area of any operator.

All of the information in the case description regarding times is transferred to a Global Table called "Production", which will be used to set the times of the different machines. However, whenever the times are the same for the different types of products, the table won't be used and we will use a uniform distribution to take into account the 10% variation stablished for the times.

	SetUpCutting	LCutting	Cutting	UCutting	LDetailing	Detailing	UDetailing	SetUpBendin	LBending	Bending	UBending	SetUpPunchi	LPunching	Punching	UPunching	SetUpWeldin	LWelding	Welding	UWelding
SeatU	30	6	4	1	4	6	3	70	2	7	2	50	5	7	2	40	6	45	3
Crossbar	30	6	4	1	4	5	3	70	0	0	0	50	5	4	2	40	4	45	3
Frontlegs	30	6	4	1	4	5	3	70	0	0	0	0	0	0	0	40	6	45	3
Backlegs	30	6	4	1	4	6	3	70	2	4	2	0	0	0	0	40	8	45	3

3.1. Storage stage



The first stage of the processing line is the storage section, which is run by two operators which are in charge of handling, organizing and transporting all of the products coming to the factory. There are three different source blocks delivering items to the storage section. A series of labels are created in each of the source blocks to later estimate processing and operators' times.

The first one, called Source Wings, delivers 1200 wings and marks the production objective for the line: whenever we run out of wings, we have manufactured 600 chairs. These wings are combined using pallets and transported by one of the operators directly to the welding stage, where they will first be needed.

Arrivals - Source Wings

Arrivals 1 Labels 18

	ItemName	Quantity	Type	LLCut	ECut	SCut	SALCut	LLDet	EDet	SDet	SALDet	LLBend	EBend	SBend	LLPunch	EPunch	SPunch	SALForming	LLContAsser	SALContAss
Arrival1	Wings	1200	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The other outsourced products are the seats and backrests, and the quantities needed to make a chair are one of each.

Arrivals - Source Backrests and Seats

Arrivals 2 Labels 9

	ItemName	Quantity	Type	Level	SWeld	SALWeld	LLPaint	EPaint	SPaint	SALPaint	LLAssembly
Arrival1	Backrests	1	6	2	0	0	0	0	0	0	0
Arrival2	Seats	1	7	3	0	0	0	0	0	0	0

Finally, the last source block represents the rods. As it has been mentioned before, there are two different types of rods, and each of them allows for the manufacturing of two different types of items. The quantities arriving to the line are set so that the process is optimized and so that not all of the first rods are processed first and then have to wait for the other parts to make the first chair. This way, the body of the chair can be welded earlier on in the process.

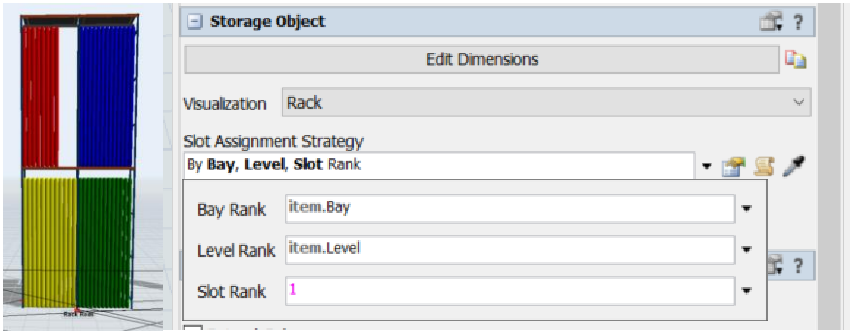
Arrivals - Source Rods

Arrivals 4 Labels 20

	ItemName	Quantity	Type	Bay	Level	LLCut	ECut	SCut	SALCut	LLDet	EDet	SDet	SALDet	LLBend	EBend	SBend	LLPunch	EPunch	SPunch	SALForming	LLContAsser	SALContAss
A1	SeatU	3	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
A2	Crossbar	1	2	1	2	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0
B1	Frontlegs	3	3	2	1	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
B2	Backlegs	6	4	2	2	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0

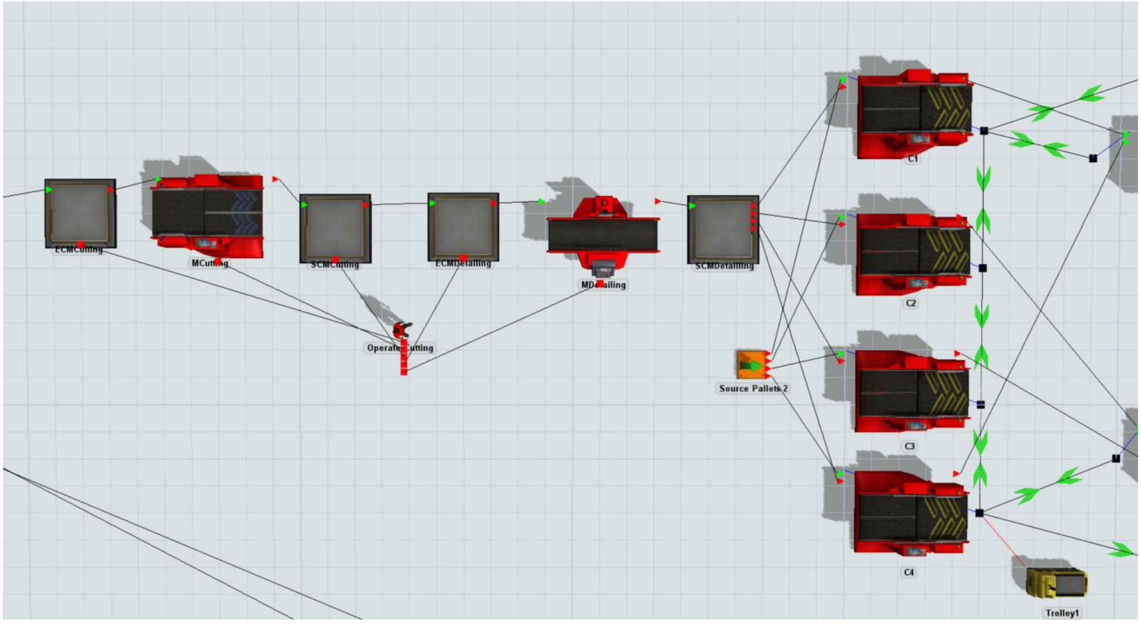
The size off each type of product is set using a visual trigger on exit in the source block. Additionally, each item type is set to have a different color. The last two labels in the arrival table are used to classify the items in the open vertical racks. We can do this by using the slot assignment strategy option in the rack properties.

```
{ // ************* PickOption Start ************* //
/***popup:SetSizeLocRot*/
/***tag:name*//**Set Size*/
Object involved = /** \nObject: *//***tag:object*//**/item/**/;
int value=item.Type;
double x; double y; double z;
switch (value) {
    case 1: x=0.2; y=0.1; z=6; break;
    case 2: x=0.2; y=0.1; z=6; break;
    case 3: x=0.18; y=0.18; z=6; break;
    case 4: x=0.18; y=0.18; z=6; break;
}
/** \nCommand: *//***tag:command*//**/setsize/**list:setloc~setrot~setsize*/(involved, x, y, z);
} // ******* PickOption End ******* //
```

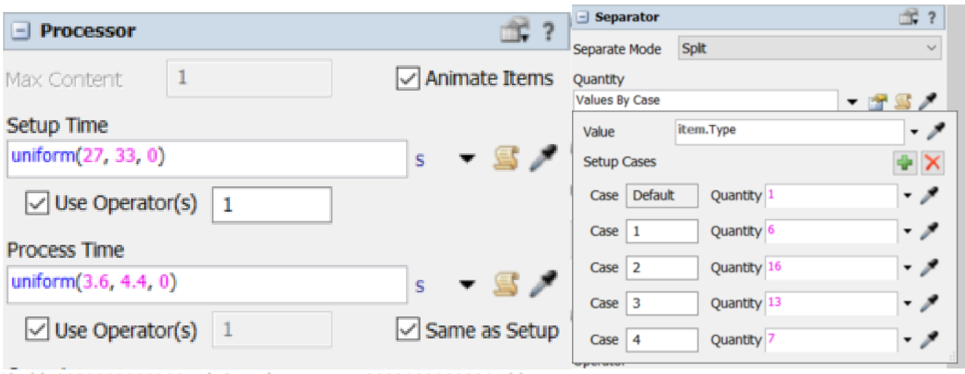


3.2. Cutting stage

For this section of the line, a single operator is in charge of both the cutting and the detailing machines.

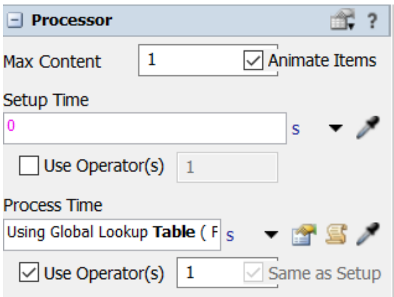


The cutting machine is actually a combiner that splits the rods into different quantities depending on the item type, as seen below. The size of these smaller parts also depends on the type of the item, and is accordingly set using a visual trigger on exit. The times for the cutting process are set using a uniform distribution, to take into account the 10% variation.




```
{ // ************* PickOption Start ************* //
/***popup:SetSizeLocRot*/
/***tag:name*//**Set Size*/
Object involved = /** \nObject: *//***tag:object*//**/item/**/;
int value=item.Type;
double x; double y; double z;
switch (value) {
    case 1: x=0.2; y=0.1; z=0.94; break;
    case 2: x=0.2; y=0.1; z=0.36; break;
    case 3: x=0.18; y=0.18; z=0.44; break;
    case 4: x=0.18; y=0.18; z=0.84; break;
}
/** \nCommand: *//***tag:command*//**/setsize/**list:setloc~setrot~setsize*/(involved, x, y, z);
} // ******* PickOption End ******* //
```

As per the detailing machine, the processing times are different depending on the type of the item, so we use the Global Table to set this time and the function Global Table Lookup.



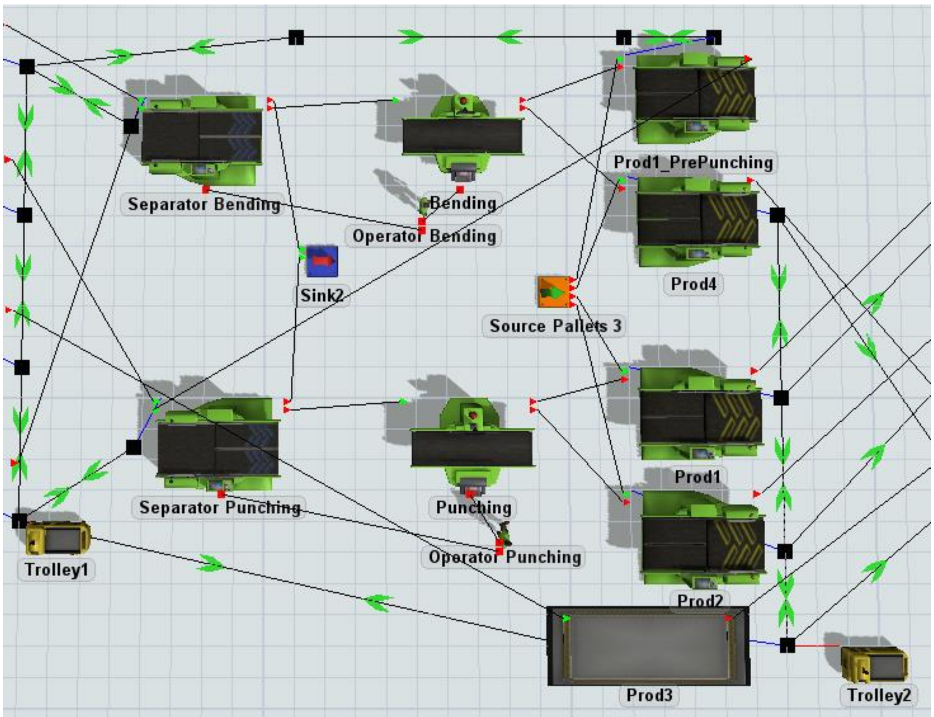
The last important coding aspect of this section are the operator's loading and unloading times, which are different for cutting and detailing. We set these times using a trigger in the queues so that when the operator picks up an item, the loading and unloading times are set correctly.

ECMCutting - On Exit

```
/**Custom Code*/
Object current = ownerobject(c);
Object item = param(1);
int port = param(2);
int load = 6;
treenode loadtimeop=Model.find("OperatorCutting>variables/loadtime");
setnodenum(loadtimeop,load);
int unload = 1;
treenode unloadtimeop=Model.find("OperatorCutting>variables/unloadtime");
setnodenum(unloadtimeop,unload);
```

Transport between the cutting stage and the next section of the line is done by using a trolley, which transports batches of 20, 30, 30 and 20 items for types 1,2,3 and 4 respectively. We use four combiners to merge all of these batches together, as the previous machines do not work with batches but for individual items or rods.

3.3. Forming stage



At this stage there are two different machines for bending and punching, run by two different operators. Depending on the type, items go through one, two or none of these machines:

- Type 1 goes through both bending and punching.
- Type 2 only goes through punching.
- Type 3 skips both machines.
- Type 4 goes through bending.

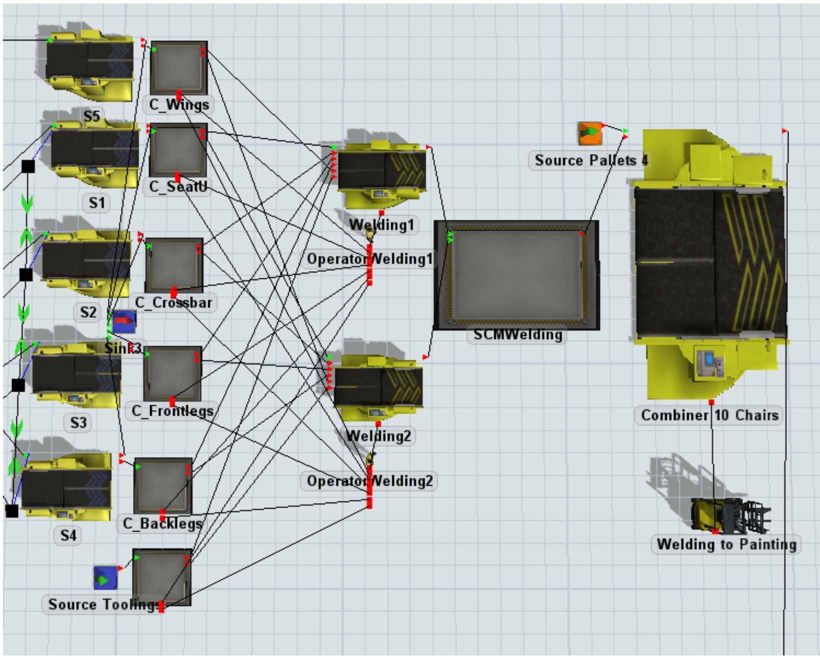
Both machines do batch processing, and their times are set using the Global Table.

Transport between this section and the next is done by using a trolley with the same characteristics as the one explained for the previous stage.

3.4. Welding stage

In this section, the 4 different types of products that we have made out of the rods, as well as the wings, are all combined together to make the frame of the chair. This is done by using two combiners in parallel that represent two welding stations, each of them run by a different operator.

Additionally, another source block is added to the line to deliver the tooling needed for the welding of the chairs, which goes into the combiner each time a chair is welded. The tooling is different for each type of chair, therefore we assign them the types 11, 12 and 13 for toolings for the gray, silver and white chairs, respectively.



The labels created in the source for the tooling are the following.

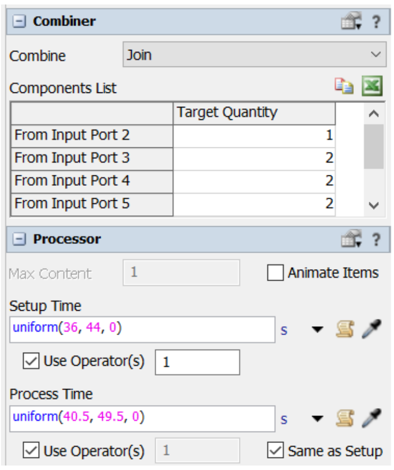
Arrivals - Source Toolings

Arrivals 3 Labels 18

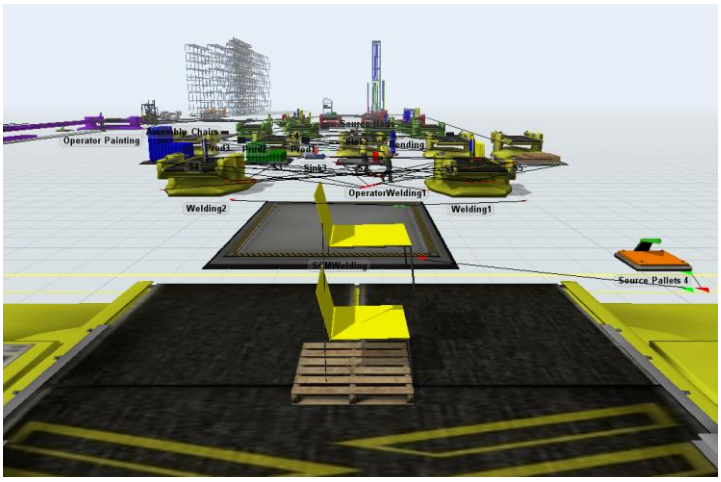
	ItemName	Quantity	Type	LLCut	ECut	SCut	SALCut	LLDet	EDet	SDet	SALDet	LLBend	EBend	SBend	LLPunch	EPunch	SPunch	SALForming	LLContAsser	SALContAss
Arrival1	GrayTooling	300	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Arrival2	SilverTooling	200	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Arrival3	WhiteToolinç	100	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

We realize that we would not need a different tooling for each individual chair and that the tooling would be recycled and used for each type of chairs. However, we could not implement this in the FlexSim model as we were not able to set two different outputs for the combiners, and that´s why we have decided to use this strategy to implement the tooling into our model.

The welding stations are exactly the same and use the configuration that follows:

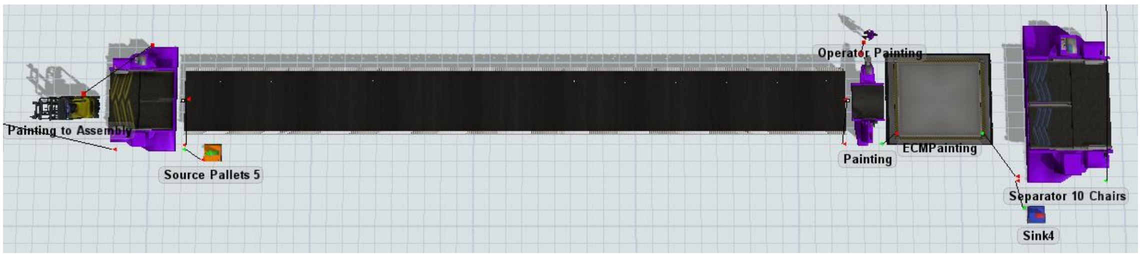


At the exit of the welding stations, a visual trigger is set to change the 3d shape to that of a chair. The chairs are then combined into batches of 10 for transport via forklift to the next stage of the manufacturing line: the painting stage.



3.5. Painting stage

When chairs arrive to the painting stage, the first thing that they find is a separator. This is done to separate the batches of 10 that arrive to this section and so that they individually go through the painting machine. Then, they go through a conveyor belt as seen in the image below:



The painting machine is 42 meters long in total, following the case description, where the actual painting processor is 2 meters long and the conveyor belt is 40 meters long.

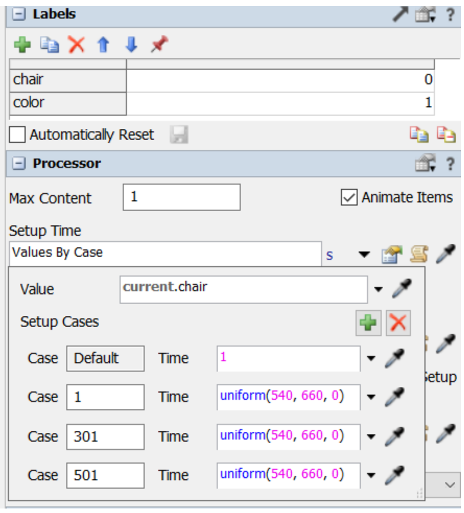
On the entry of the painting machine, a label is created that counts the number of chairs that enter the machine, which is very important to then change the color for the 301st and 501st chairs to silver and white respectively.

Painting - On Entry

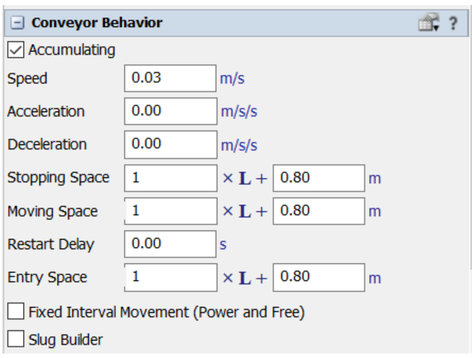
```
treenode item = parnode(1);
treenode current = ownerobject(c);
int port = parval(2);

{ //************* PickOption Start *************\\
int numero=getlabelnum(current,"chair");
/***popup:SetCombo*/
/***tag:title*//**Set Label Value*/
numero++;
treenode involved = /***tag:item*//**/current/**/;
/***tag:command*//**/setlabelnum/**/(involved, /***tag:var*//**/"chair"/**/ /***tag:value*//**/,numero/**/);

if (numero>600){
    treenode involved = /***tag:item*//**/current/**/;
    /***tag:command*//**/setlabelnum/**/(involved, /***tag:var*//**/"chair"/**/ /***tag:value*//**/,1/**/);
    numero=1;
}

if (numero>500){
    setlabelnum(current,"color",3);
} else {
    if (numero>300){
    setlabelnum(current,"color",2);
    }
    else {
    setlabelnum(current,"color",1);
    }
}
} //******* PickOption End *******\\
```

The label chairs is equal to 0 on reset, and color is equal to 1, which represents the color gray. Color 2 represents the color silver and finally color 3 represents the color white.

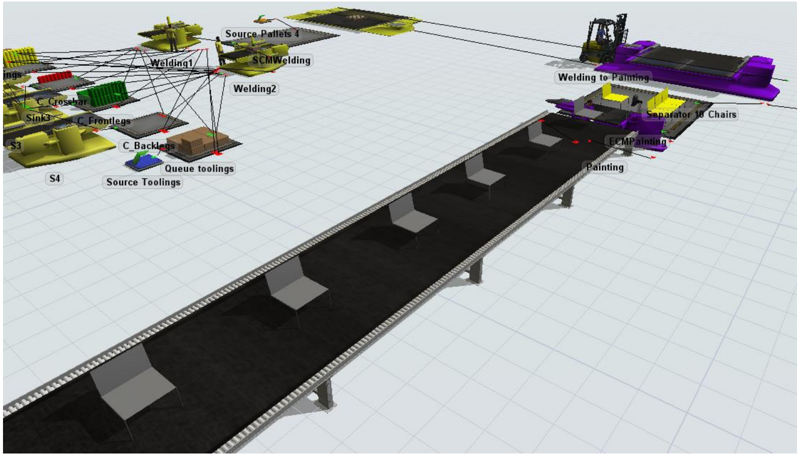
Therefore, when the 301st chair comes into the machine, the label color turns to 2, and then a set up time starts that resembles the changing color time. The same thing happens for the 501st chair.



Then, the chairs go through the conveyor belt, which has the following characteristics:



The configuration of the conveyor would not let us use a uniform distribution for its speed, so we decided to establish the average speed between the minimum and maximum values given in the case description as the actual continuous speed of the conveyor.



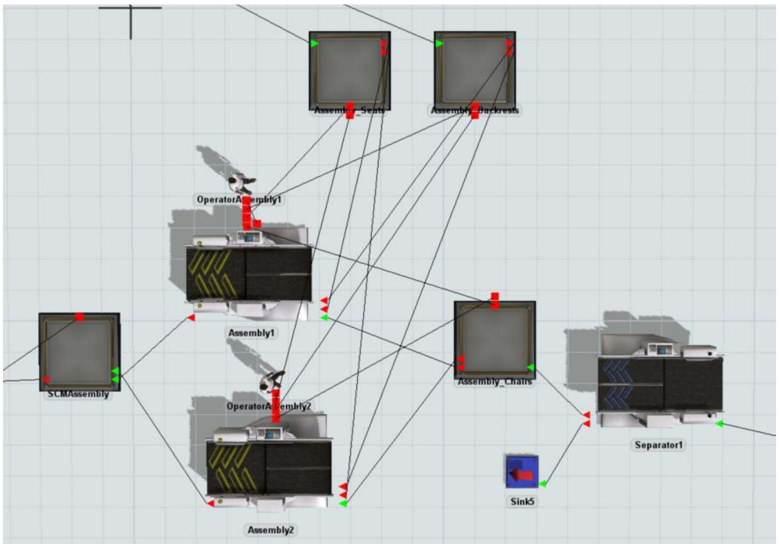
The color of the chairs is set using a trigger on the exit of the painting machine which uses the color label we had created before, assigning the gray color to chairs with the label chair 1, color silver to the label chair 2 and white to the label chair 3.

```
/**Set Color By Case*/
int value = /** \nValue: *//***tag:item*//**/getlabelnum(current,"color")/**/;
/** \nCases:\n*/
switch (value) {
    /***tagex:data*//**\nCase: */case /**/2/**/: /** Color: */ /**/colorsilver(item)/**/;break;
/**\nCase: */case /**/3/**/: /** Color: */ /**/colorwhite(item)/**/;break;
/***//**/ /**/
    default: /** \nDefault Value: *//***tag:default_value*//**/colorgray(item)/**/;break;
}
```

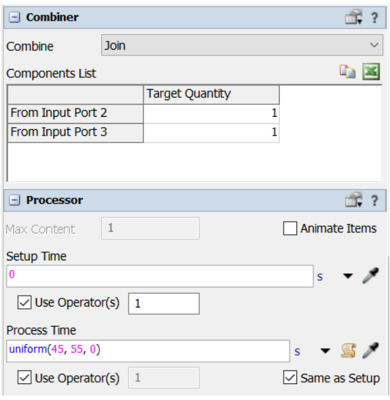
The transport between the painting stage and the next section is done by using a forklift, stacking the chairs in batches of 10.

3.6. Assembly stage

For the penultimate stage of the processing line, the painted frames of the chairs are combined with the outsourced seats and backrests. There are two assembly machines working in parallel, each one run by a different operator.

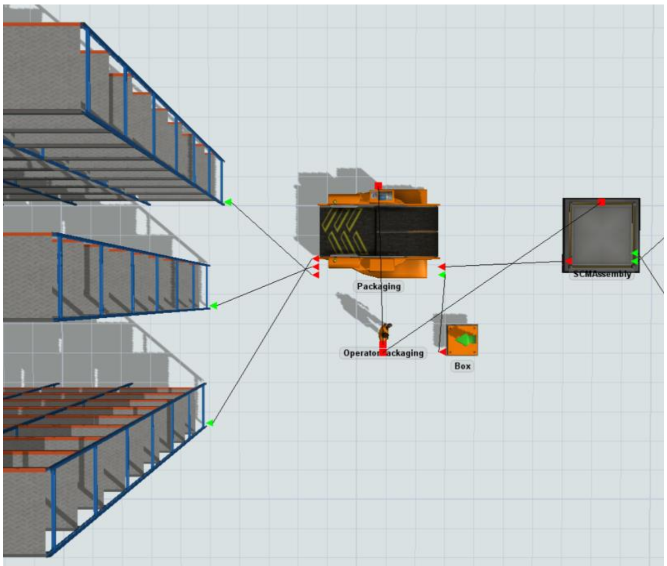


The configuration for the assembly machines is shown below, and is exactly the same for the two machines:

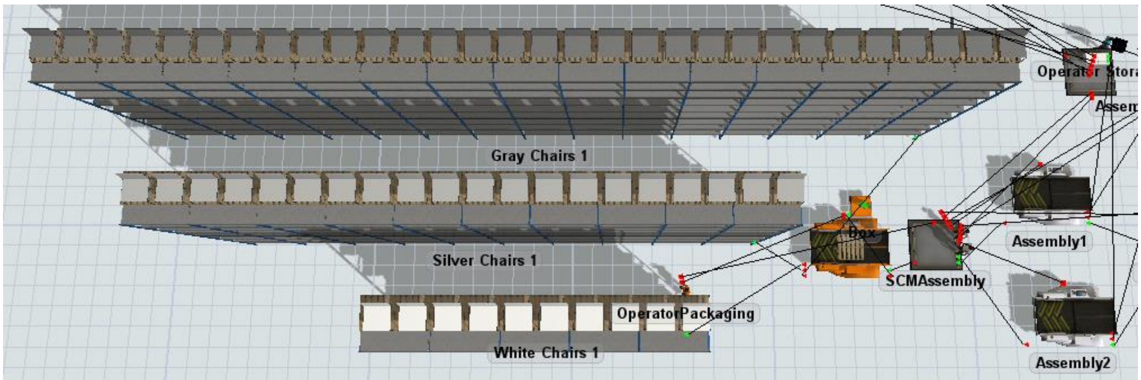


3.7. Packaging stage

The last step of the process is the packaging. The chairs are combined with a pallet that serves as a box for the chairs and they are then stored in open vertical racks, separated between colors.



As we can see in the image below, the production plan is completed successfully, and the 600 chairs are manufactured.

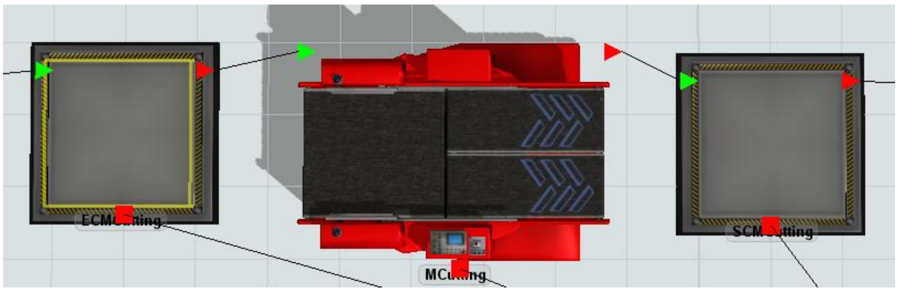


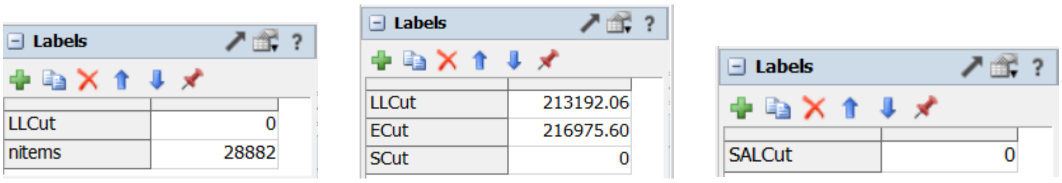
This storage section marks the end of the manufacturing line.

4. FLEXSIM MODEL: IMPLEMENTING TIMES INTO THE MODEL

As it has been previously stated, we have created a series of labels in the source blocks that will be used to save the entry and exit times into the different processors and queues in the processing line. This is done by triggers on entry/exit to set the label values to the time of the model.

For example, for the very first machine of the line, the cutting machine, we establish the following labels and triggers:

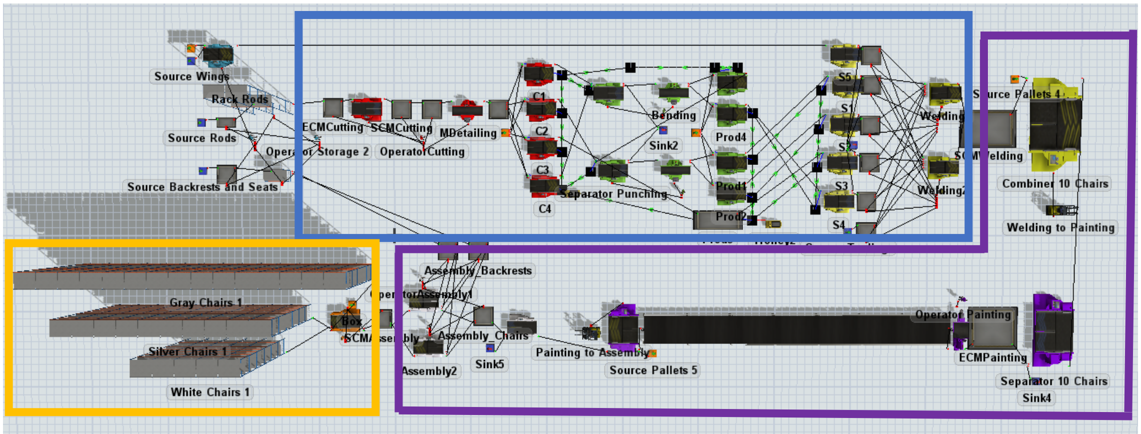




- LLCut is the time at which items enter the container prior to the machine, and is obtained by using an on entry trigger in the queue.
- ECut is the time at which items enter the cutting machine, and is obtained by using an entry trigger in the machine.
- SCut is the time at which items exit the cutting machine, and is obtained by using an on exit trigger in the machine.
- SALCut is the time at which items exit the container following the cutting machine, and is obtained by using an on exit trigger in the queue.

This process is used accordingly for all of the different machines in the processing line.

To gather all of these times together, we will divide the processing line into three different sections.



This has to be done because the items going through the line are different: in the beginning the items are rods, but when they exit the welding machines and create a chair, another item is created and we can not keep assigning times to the label of the rods. The same happens for the packaging stage: when the chairs are combined with a box the item changes and we cannot retrieve those times in the same labels as before.

For the first section of the processing line (blue rectangle), times are gathered in two tables called TimesRods1 and TimesRods2 in the welding machines.

Welding1 - On Entry

```
/**Custom Code*/
Object current = ownerobject(c);
Object item = param(1);
int port = param(2);
double type=item.Type;
double LLCut=item.LLCut;
double ECut=item.ECut;
double SCut=item.SCut;
double SALCut=item.SALCut;
double LLDet=item.LLDet;
double EDet=item.EDet;
double SDet=item.SDet;
double SALDet=item.SALDet;
double LLBend=item.LLBend;
double EBend=item.EBend;
double SBend=item.SBend;
double LLPunch=item.LLPunch;
double EPunch=item.EPunch;
double SPunch=item.SPunch;
double SALForming=item.SALForming;
double LLContAssembly=item.LLContAssembly;
double SALContAssembly=item.SALContAssembly;

Table tabla=reftable("TimesRods1");
int filas=current.stats.input.value;

tabla.setSize(filas,18);

tabla[filas][1]=type;
tabla[filas][2]=LLCut;
tabla[filas][3]=ECut;
tabla[filas][4]=SCut;
tabla[filas][5]=SALCut;
tabla[filas][6]=LLDet;
tabla[filas][7]=EDet;
tabla[filas][8]=SDet;
tabla[filas][9]=SALDet;
tabla[filas][10]=LLBend;
tabla[filas][11]=EBend;
tabla[filas][12]=SBend;
tabla[filas][13]=LLPunch;
tabla[filas][14]=EPunch;
tabla[filas][15]=SPunch;
tabla[filas][16]=SALForming;
tabla[filas][17]=LLContAssembly;
tabla[filas][18]=SALContAssembly;
```

Welding2 - On Entry

```
/**Custom Code*/
Object current = ownerobject(c);
Object item = param(1);
int port = param(2);
double type=item.Type;
double LLCut=item.LLCut;
double ECut=item.ECut;
double SCut=item.SCut;
double SALCut=item.SALCut;
double LLDet=item.LLDet;
double EDet=item.EDet;
double SDet=item.SDet;
double SALDet=item.SALDet;
double LLBend=item.LLBend;
double EBend=item.EBend;
double SBend=item.SBend;
double LLPunch=item.LLPunch;
double EPunch=item.EPunch;
double SPunch=item.SPunch;
double SALForming=item.SALForming;
double LLContAssembly=item.LLContAssembly;
double SALContAssembly=item.SALContAssembly;

Table tabla=reftable("TimesRods2");
int filas=current.stats.input.value;

tabla.setSize(filas,18);

tabla[filas][1]=type;
tabla[filas][2]=LLCut;
tabla[filas][3]=ECut;
tabla[filas][4]=SCut;
tabla[filas][5]=SALCut;
tabla[filas][6]=LLDet;
tabla[filas][7]=EDet;
tabla[filas][8]=SDet;
tabla[filas][9]=SALDet;
tabla[filas][10]=LLBend;
tabla[filas][11]=EBend;
tabla[filas][12]=SBend;
tabla[filas][13]=LLPunch;
tabla[filas][14]=EPunch;
tabla[filas][15]=SPunch;
tabla[filas][16]=SALForming;
tabla[filas][17]=LLContAssembly;
tabla[filas][18]=SALContAssembly;
```

As we can see, we have 9 different items per chair: one seatU (type 1), one crossbar (type2), two front (type 3) and back legs (type 4), two wings (type 5) and the tooling (types 11, 12, 13, depending on the type of chair). The tables look as follows:

	type	LLCut	ECut	SCut	SALCut	LLDet	EDet	SDet	SALDet	LLBend	EBend	SBend	LLPunch	EPunch	SPunch	SALForming	LLContAssem	SALContAsse
Row 1	1	461.97	3429.09	3546.31	4444.17	4444.17	5202.93	5216.74	5222.40	6330.46	6425.41	6513.78	9154.90	10983.42	11073.87	11104.50	11104.50	11109.69
	2	141.37	274.41	338.19	338.19	338.19	1048.25	1061.06	1067.26	0	0	0	6364.69	9100.28	9192.35	9219.75	9219.75	11121.25
	3	188.95	1097.03	1282.93	2089.12	2089.12	2973.41	2986.22	2992.41	0	0	0	0	0	0	3029.54	3029.54	11134.53
	3	188.95	1097.03	1249.67	2055.87	2055.87	2940.16	2952.97	2959.16	0	0	0	0	0	0	3029.54	3029.54	11149.09
	4	204.76	1602.70	1684.42	2494.15	2494.15	3379.29	3393.10	3399.29	4356.73	6308.80	6408.28	0	0	0	6440.12	6440.12	11165.72
	4	204.76	1602.70	1717.66	2599.00	2599.00	3413.54	3496.52	3500.73	4356.73	6205.95	6291.36	0	0	0	6440.12	6440.12	11184.37
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.33	93.33	11200.86
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.33	93.33	11213.02
	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11231.29
	1	42.57	188.48	246.61	246.61	246.61	979.73	993.54	999.74	6330.46	6739.88	6827.94	9154.90	10702.06	10776.63	11104.50	11104.50	11368.16
	2	141.37	274.41	338.19	338.19	338.19	1221.32	1234.13	1240.33	0	0	0	6364.69	8817.35	8891.08	9219.75	9219.75	11399.64
	3	173.14	580.59	1009.08	1817.40	1817.40	2703.66	2716.47	2722.67	0	0	0	0	0	0	3029.54	3029.54	11413.25
	3	173.14	580.59	974.82	1784.15	1784.15	2670.41	2683.23	2689.42	0	0	0	0	0	0	3029.54	3029.54	11427.82
	4	204.76	1602.70	1883.90	2765.24	2765.24	3653.97	3667.78	3673.97	4356.73	5693.86	5779.35	0	0	0	6440.12	6440.12	11444.45
	4	220.57	1904.34	1989.38	2798.49	2798.49	3688.22	3773.70	3777.91	4356.73	5591.09	5676.49	0	0	0	6440.12	6440.12	11463.09
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.33	93.33	11479.58
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.33	93.33	11491.74
	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11510.01
	1	42.57	188.48	246.61	246.61	246.61	911.22	925.02	931.22	6330.46	6949.29	7037.55	9154.90	10517.04	10590.16	11104.50	11104.50	11641.59
	2	141.37	274.41	338.19	457.37	457.37	1287.84	1300.65	1306.85	0	0	0	6364.69	8630.38	8703.75	9219.75	9219.75	11653.14
	3	173.14	580.59	872.04	1684.42	1684.42	2499.06	2583.46	2587.67	0	0	0	0	0	0	3029.54	3029.54	11666.42
	3	173.14	580.59	837.78	1582.25	1582.25	2465.82	2478.63	2484.82	0	0	0	0	0	0	3029.54	3029.54	11680.99
	4	220.57	1904.34	2089.12	2968.49	2968.49	3862.65	3876.47	3882.66	4356.73	5283.94	5369.10	0	0	0	6440.12	6440.12	11697.62
	4	220.57	1904.34	2122.37	3001.74	3001.74	3896.90	3910.72	3916.91	4356.73	5181.88	5266.97	0	0	0	6440.12	6440.12	11716.26
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.33	93.33	11732.75
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.33	93.33	11744.91
	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11763.18
	1	42.57	188.48	246.61	246.61	246.61	842.70	856.51	862.70	6330.46	7159.60	7248.05	9154.90	10329.01	10403.69	11104.50	11104.50	11901.34
	2	141.37	274.41	338.19	525.89	525.89	1354.36	1367.17	1373.37	0	0	0	6364.69	8443.29	8516.95	9219.75	9219.75	11912.89
	3	173.14	580.59	735.00	1482.48	1482.48	2366.07	2378.88	2385.08	0	0	0	0	0	0	3029.54	3029.54	11926.17
	3	173.14	580.59	700.75	1449.22	1449.22	2332.82	2345.64	2351.83	0	0	0	0	0	0	3029.54	3029.54	11940.74
	4	236.38	2209.31	2294.66	3101.48	3101.48	4069.07	4082.88	4089.07	4356.73	4874.31	4959.62	0	0	0	6440.12	6440.12	11957.37
	4	236.38	2209.31	2327.91	3208.13	3208.13	4103.31	4117.13	4123.32	4356.73	4772.01	4857.23	0	0	0	6440.12	6440.12	11976.01
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.33	93.33	11992.51
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.33	93.33	12004.66
	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12022.94

For the next section of the line (purple rectangle), times are gathered in the tables called TimesPaintAndAssembly1 and TimesPaintAndAssembly2 at the two assembly machines.

Assembly1 - On Entry

```
/**Custom Code*/
Object current = ownerobject(c);
Object item = param(1);
int port = param(2);

double SWeld=item.SWeld;
double SALWeld=item.SALWeld;
double LLPaint=item.LLPaint;
double EPaint=item.EPaint;
double SPaint=item.SPaint;
double SALPaint=item.SALPaint;
double LLAssembly=item.LLAssembly;
double EAssembly=time();


Table tabla=reftable("TimesPaintAndAssembly1");
int filas=current.stats.input.value;

tabla.setSize(filas,8);

tabla[filas][1]=SWeld;
tabla[filas][2]=SALWeld;
tabla[filas][3]=LLPaint;
tabla[filas][4]=EPaint;
tabla[filas][5]=SPaint;
tabla[filas][6]=SALPaint;
tabla[filas][7]=LLAssembly;
tabla[filas][8]=EAssembly;
```

Assembly2 - On Entry

```
/**Custom Code*/
Object current = ownerobject(c);
Object item = param(1);
int port = param(2);

double SWeld=item.SWeld;
double SALWeld=item.SALWeld;
double LLPaint=item.LLPaint;
double EPaint=item.EPaint;
double SPaint=item.SPaint;
double SALPaint=item.SALPaint;
double LLAssembly=item.LLAssembly;
double EAssembly=time();


Table tabla=reftable("TimesPaintAndAssembly2");
int filas=current.stats.input.value;

tabla.setSize(filas,8);

tabla[filas][1]=SWeld;
tabla[filas][2]=SALWeld;
tabla[filas][3]=LLPaint;
tabla[filas][4]=EPaint;
tabla[filas][5]=SPaint;
tabla[filas][6]=SALPaint;
tabla[filas][7]=LLAssembly;
tabla[filas][8]=EAssembly;
```

There are three items that form a finished chair: the frame of the chair obtained in the previous section of the line, and the seats and backrests, which don´t go through the painting stage. The tables look as follows:

	SWeld	SALWeld	LLPaint	EPaint	SPaint	SALPaint	LLAssembly	EAssembly
Row 1	11343.22	12405.02	12422.86	14135.18	14270.92	15607.25	15635.70	15653.85
	0	0	0	0	0	0	477.55	15670.48
	0	0	0	0	0	0	353.87	15688.41
	11548.33	12405.02	12422.86	13863.97	13998.12	15334.46	15635.70	15773.78
	0	0	0	0	0	0	15681.81	15790.61
	0	0	0	0	0	0	15738.34	15809.53
	11752.13	12405.02	12422.86	13600.95	13732.23	15068.56	15635.70	15890.22
	0	0	0	0	0	0	15821.15	15906.89
	0	0	0	0	0	0	15869.42	15925.17
	11964.13	12405.02	12422.86	13353.12	13469.31	14805.65	15635.70	16011.79
	0	0	0	0	0	0	15951.21	16028.70
	0	0	0	0	0	0	15976.27	16047.58
	12164.28	12405.02	12422.86	13086.16	13220.94	14557.27	15635.70	16134.59
	0	0	0	0	0	0	16081.22	16150.72
	0	0	0	0	0	0	16119.49	16169.12
	12415.45	13468.43	13491.49	15404.63	15529.49	16865.82	16897.29	16917.39
	0	0	0	0	0	0	16276.25	16934.55
	0	0	0	0	0	0	16309.57	16952.77
	12691.78	13468.43	13491.49	15143.55	15278.06	16614.40	16897.29	17035.84
	0	0	0	0	0	0	16946.19	17052.35
	0	0	0	0	0	0	17002.71	17070.52
	12945.89	13468.43	13491.49	14893.28	15018.77	16355.11	16897.29	17152.09
	0	0	0	0	0	0	17076.19	17169.65
	0	0	0	0	0	0	17124.46	17187.64
	13204.79	13468.43	13491.49	14643.37	14765.66	16102.00	16897.29	17275.35
	0	0	0	0	0	0	17206.25	17291.69
	0	0	0	0	0	0	17231.31	17309.66
	13455.87	13468.43	13491.49	14392.19	14519.90	15856.23	16897.29	17391.58
	0	0	0	0	0	0	17341.64	17407.81
	0	0	0	0	0	0	17374.96	17425.41
	14901.57	15960.50	15983.56	17130.53	17254.19	18590.53	18621.99	18641.68
	0	0	0	0	0	0	17473.31	18658.71
	0	0	0	0	0	0	17498.37	18677.20

Finally, packaging times and the final time at which chairs arrive to the expedition storage racks are stored in three different tables, one for each type of chair.

Gray Chairs 1 - On Entry

```
/**Custom Code*/
Object current = ownerobject(c);
Object item = param(1);
int port = param(2);
int bay = param(3);
int level = param(4);


double SALPackaging=item.SALPackaging;
double LLFinalStorage=Model.time;

Table tabla=reftable("GrayChairs");
int filas=current.stats.input.value;

tabla.setSize(filas,2);

tabla[filas][1]=SALPackaging;
tabla[filas][2]=LLFinalStorage;

```

Silver Chairs 1 - On Entry

```
/**Custom Code*/
Object current = ownerobject(c);
Object item = param(1);
int port = param(2);
int bay = param(3);
int level = param(4);
double SALPackaging=item.SALPackaging;
double LLFinalStorage=Model.time;

Table tabla=reftable("SilverChairs");
int filas=current.stats.input.value;

tabla.setSize(filas,2);

tabla[filas][1]=SALPackaging;
tabla[filas][2]=LLFinalStorage;

```

White Chairs 1 - On Entry

```
/**Custom Code*/
Object current = ownerobject(c);
Object item = param(1);
int port = param(2);
int bay = param(3);
int level = param(4);
double SALPackaging=item.SALPackaging;
double LLFinalStorage=Model.time;

Table tabla=reftable("WhiteChairs");
int filas=current.stats.input.value;

tabla.setSize(filas,2);

tabla[filas][1]=SALPackaging;
tabla[filas][2]=LLFinalStorage;

```

These tables look as follows:

	SALPackagin	LLFinalStorag
Row 1	15756.43	15756.43
	15766.76	15766.76
	15870.21	15870.21
	15889.81	15889.81
	15989.18	15989.18
	16017.67	16017.67
	16112.59	16112.59
	16149.45	16149.45
	16232.97	16232.97
	16273.30	16273.30
	17016.62	17016.62
	17027.37	17027.37
	17132.73	17132.73
	17153.48	17153.48
	17255.98	17255.98
	17279.81	17279.81
	17371.10	17371.10
	17405.53	17405.53
	17486.01	17486.01
	17530.24	17530.24
	18741.46	18741.46
	18752.08	18752.08
	18864.15	18864.15
	18883.57	18883.57
	18980.50	18980.50
	19015.96	19015.96
	19097.29	19097.29
	19142.79	19142.79
	19212.92	19212.92
	19268.61	19268.61

	SALPackagin	LLFinalStorag
Row 1	117183.81	117188.20
	117200.43	117204.82
	117306.67	117311.06
	117327.89	117332.28
	117430.95	117435.34
	117453.72	117458.11
	117554.84	117559.23
	117581.31	117585.70
	117674.09	117678.48
	117706.66	117711.05
	118740.54	118744.93
	118751.39	118755.78
	118862.55	118866.94
	118874.99	118879.38
	118979.12	118983.51
	118998.11	119002.50
	119096.36	119100.75
	119128.40	119132.79
	119214.29	119218.68
	119258.19	119262.58
	122830.19	122834.58
	122841.04	122845.43
	122949.03	122953.41
	122962.02	122966.41
	123065.89	123070.27
	123095.90	123100.29
	123187.09	123191.47
	123227.31	123231.70
	123300.89	123305.28
	123359.97	123364.36
	127031.21	127035.59
	127042.95	127047.33

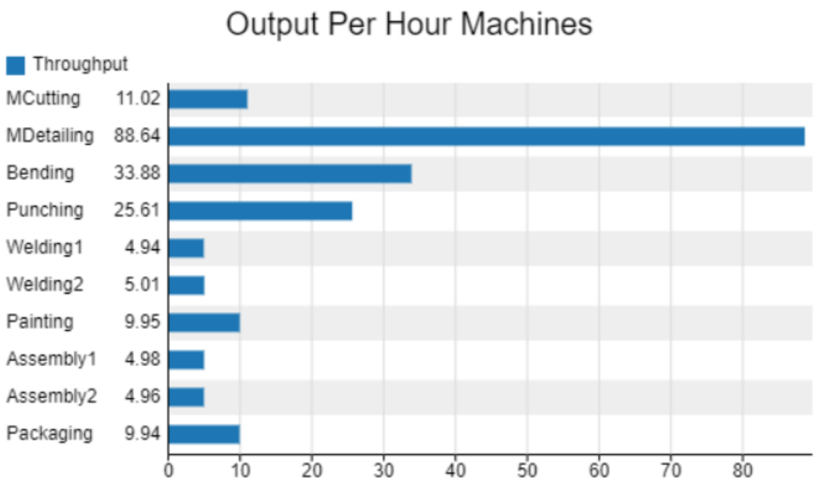
	SALPackagin	LLFinalStorag
Row 1	187977.20	187983.51
	187991.88	187998.17
	188099.06	188105.35
	188113.73	188120.02
	188220.52	188226.81
	188236.52	188242.81
	188340.42	188346.71
	188364.10	188370.39
	188461.33	188467.62
	188497.05	188503.34
	191623.92	191630.21
	191638.59	191644.88
	191741.86	191748.15
	191761.24	191767.53
	191866.30	191872.59
	191888.69	191894.98
	191983.15	191989.44
	192022.34	192028.63
	192104.55	192110.84
	192148.26	192154.55
	193657.42	193663.71
	193672.09	193678.38
	193771.19	193777.48
	193798.13	193804.42
	193886.40	193892.69
	193923.49	193929.78
	194004.39	194010.68
	194056.75	194063.04
	194122.64	194128.93
	194190.47	194196.76
	195752.16	195758.45
	195766.83	195773.12

5. DASHBOARD DATA

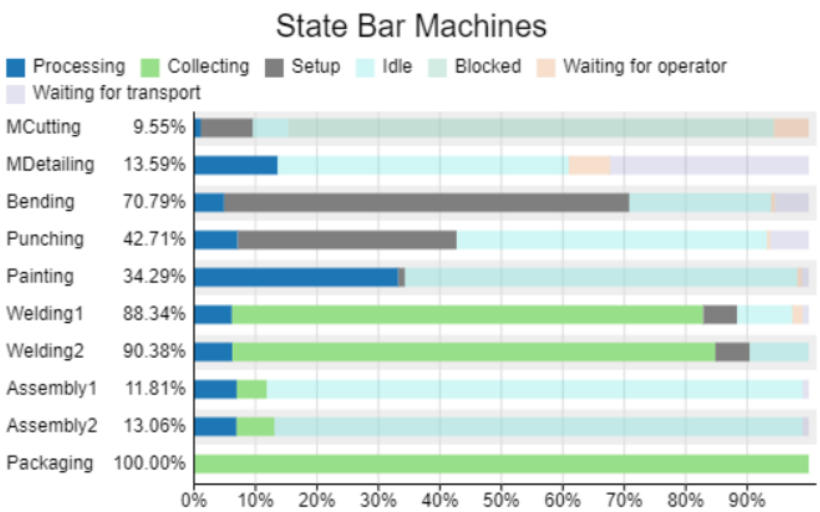
For the dashboard data, we are going to be focusing on the productivity of the machines using the output data per hour as well as the state bar information. We are also going to take a look at operators and transporters.

5.1. Dashboard data for the Machines

In the graph below we can see the output per hour of every machine in the processing line. We can see how the machines that are doubled and work in parallel have almost identical productivity levels.



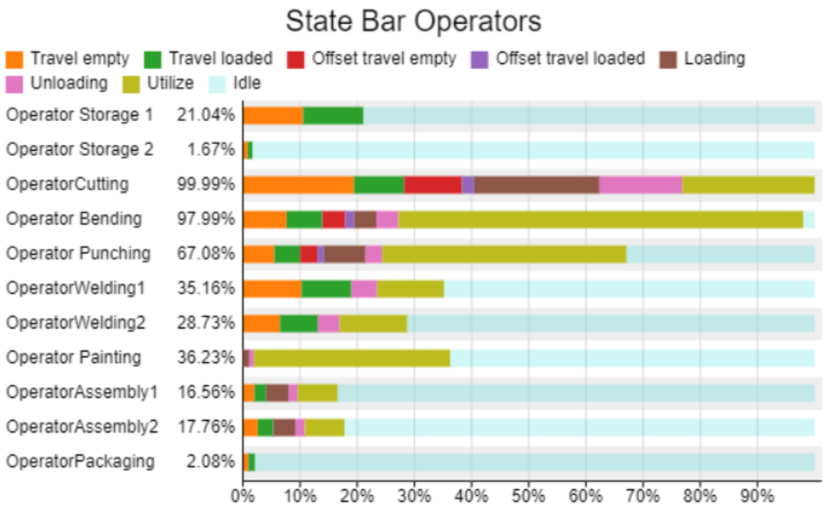
This information can also be complemented with the state bar, which is shown below.



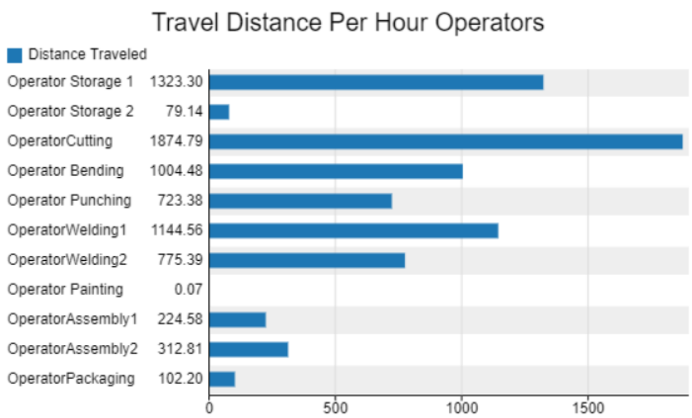
Some of the conclusions that we get from these graphs are that the machine with the lowest processing rate is the cutting machine and that the bending machine has a very long set up time compared to the rest of the machines.

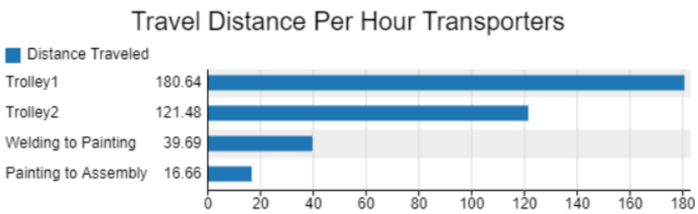
5.2. Dashboard data for the Operators and Transporters

To address the productivity of the operators, we are going to analize their state bar, shown below. The most relevant points that we can take from this graph are that the operators in charge of the cutting stage (both the cutting and detailing machines) and the operator in charge of the bending machine are working at almost a 100% of their capacity. This means that we have found two bottle necks in the line, and in order to improve productivity we will have to double these operators.



Additionally, information regarding the distance traveled per hour by the operators and transporters in shown in the graphs below.

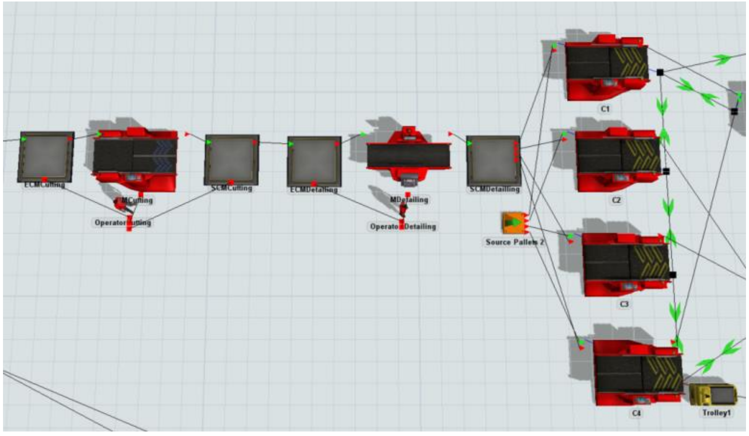




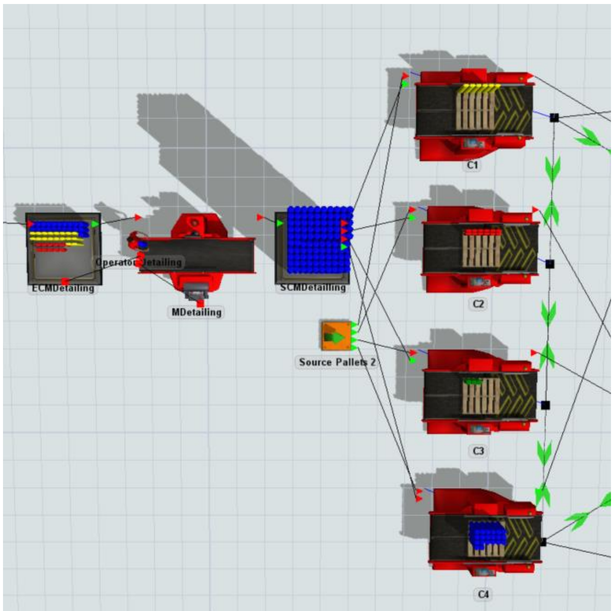
6. CHANGES AND IMPROVEMENTS TO THE MODEL

6.1. Adding another operator to the cutting stage (implemented)

The first thing we decided to modify in the processing line is the cutting stage. Because there is only operator working in this section of the line, the productivity is very low. Therefore, we will add another worker to this stage so that each one operates a separate machine: one operator for the cutting machine and another operator for the detailing machine.



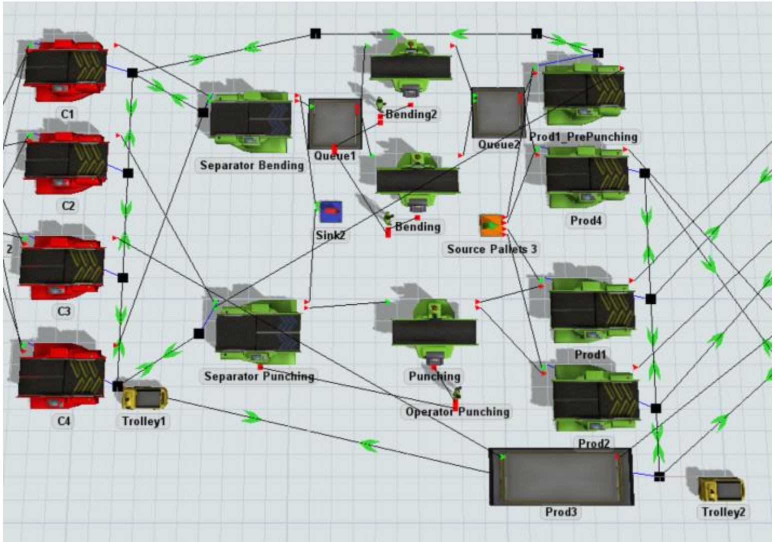
By doing so, we started to see a bottle neck forming after the detailing section, caused by the low productivity of the bending machine as this last machine could not handle the growing flow of items arriving to it.



This is mainly due to the bending machine having such a big setup time compared to some of the other machines, as we could see from the dashboard data obtained before. Therefore, we decided to add another bending machine in parallel with the one already installed in the manufacturing line.

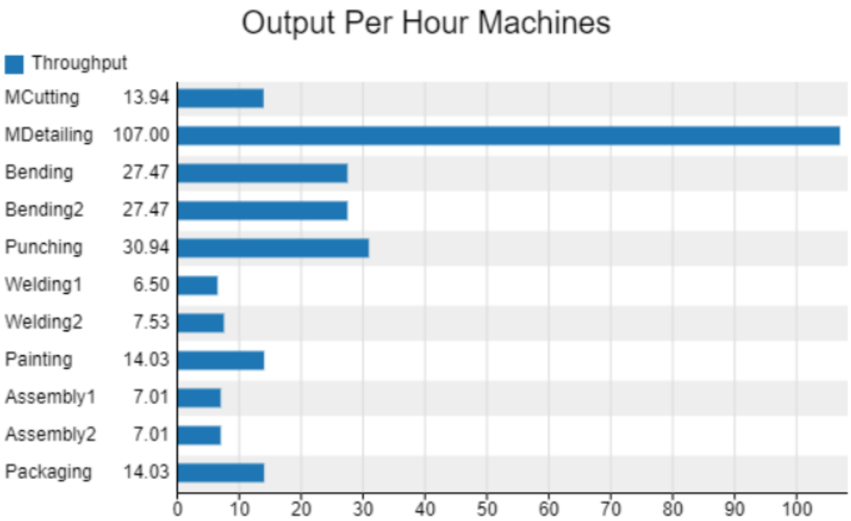
6.2. Adding another bending machine in parallel (implemented)

To solve the mentioned bottle neck, we added another bending machine to the line, run by a new operator.



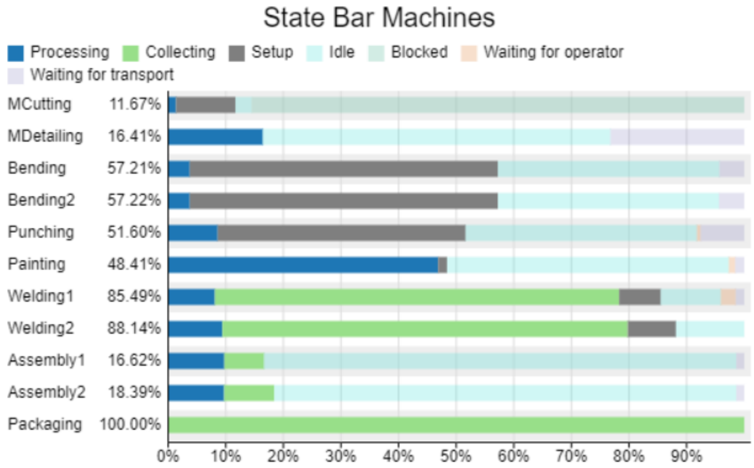
This way, we have managed to improve the productivity of the line without causing bottle neck issues and we have improved the times for the process.

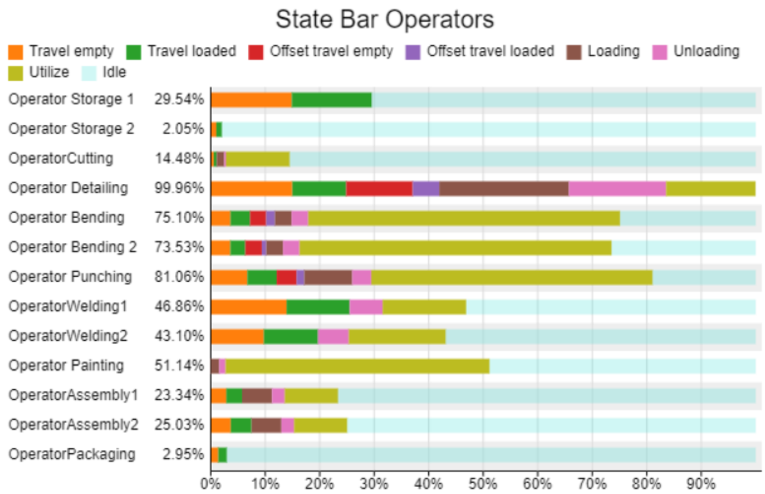
We can now see how the dashboard data has changed after applying these changes:

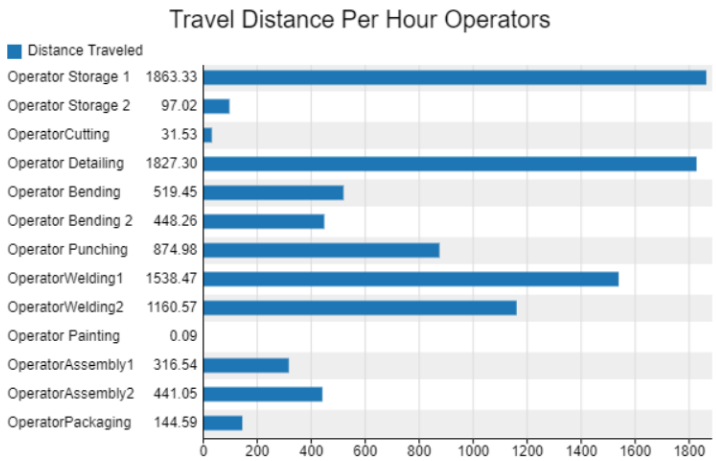


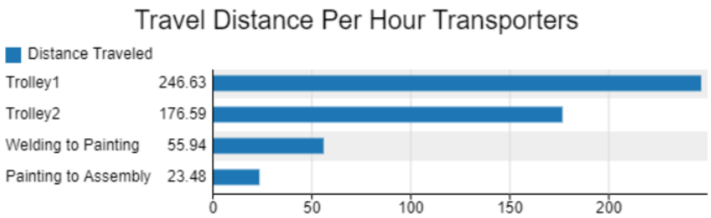
We can see from this graph how the changes have helped increase the productivity of all the machines. The only exception are the bending machines, which have dropped in productivity from 33 output per hour to 27.5, which means they are overlapping a little bit. However, this overlapping is acceptable and we can accept the results.

For contrast with the dashboard data obtained for the basic model, the same information for the improved model is also provided below:


State Bar Machines
Processing
Collecting
Setup
Idle
Blocked
Waiting for operator
Waiting for transport
MCutting 11.67%
MDetailing 16.41%
Bending 57.21%
Bending2 57.22%
Punching 51.60%
Painting 48.41%
Welding1 85.49%
Welding2 88.14%
Assembly1 16.62%
Assembly2 18.39%
Packaging 100.00%
0% 10% 20% 30% 40% 50% 60% 70% 80% 90%


State Bar Operators
Travel empty
Travel loaded
Offset travel empty
Offset travel loaded
Loading
Unloading
Utilize
Idle
Operator Storage 1 29.54%
Operator Storage 2 2.05%
OperatorCutting 14.48%
Operator Detailing 99.96%
Operator Bending 75.10%
Operator Bending 2 73.53%
Operator Punching 81.06%
OperatorWelding1 46.86%
OperatorWelding2 43.10%
Operator Painting 51.14%
OperatorAssembly1 23.34%
OperatorAssembly2 25.03%
OperatorPackaging 2.95%
0% 10% 20% 30% 40% 50% 60% 70% 80% 90%

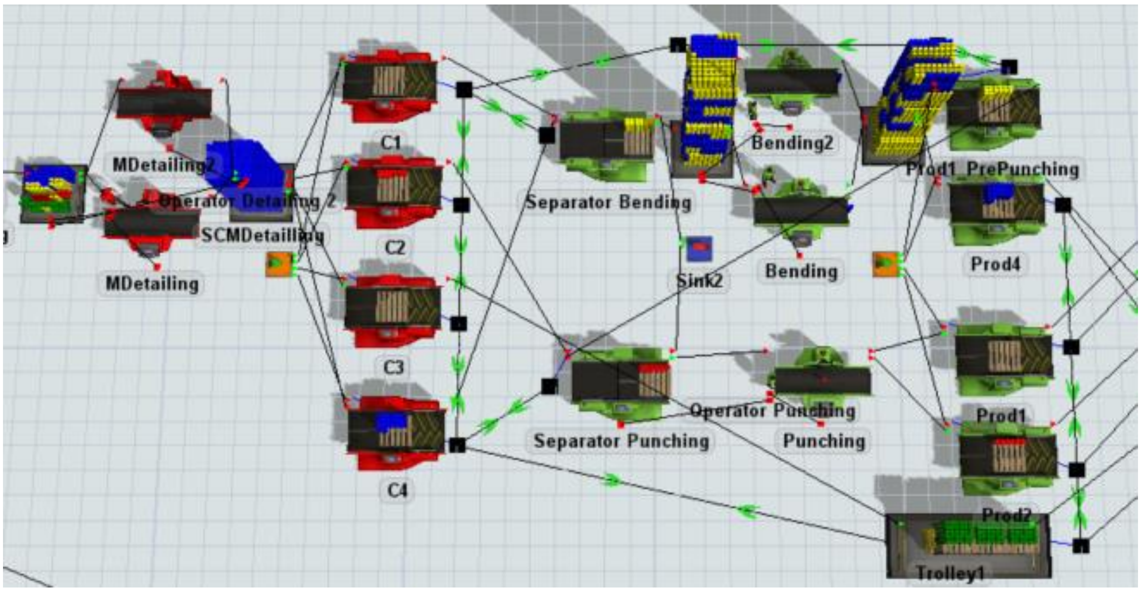

Travel Distance Per Hour Operators
Distance Traveled
Operator Storage 1 1863.33
Operator Storage 2 97.02
OperatorCutting 31.53
Operator Detailing 1827.30
Operator Bending 519.45
Operator Bending 2 448.26
Operator Punching 874.98
OperatorWelding1 1538.47
OperatorWelding2 1160.57
Operator Painting 0.09
OperatorAssembly1 316.54
OperatorAssembly2 441.05
OperatorPackaging 144.59
0 200 400 600 800 1000 1200 1400 1600 1800


Travel Distance Per Hour Transporters
Distance Traveled
Trolley1 246.63
Trolley2 176.59
Welding to Painting 55.94
Painting to Assembly 23.48
0 50 100 150 200

6.3. Adding another cutting machine in parallel (not implemented)

We thought about adding another cutting machine to the line, as this is the machine with the lowest productivity of the factory. However, adding another cutting machine would mean an increasing flow of items through the detailing machine, whose operator was already working at almost a 100% of its capacity. Therefore, adding another cutting machine would mean that we would also have to add another detailing machine to the station.

If we did so, we would again be in the same situation as before, where a bottle neck would start to form at the exit of the detailing line, caused by the bending machines.



To solve this we would have to add another bending machine in parallel, and we would start a never ending loophole of improvements, also adding to our costs of machines and operators. Also, we would run out of space in FlexSim if we were to add more objects like machines and operators. Therefore, we finally decided not to add another cutting machine to the line.

7. TIME AND COST ANALYSIS: BASIC VS IMPROVED MODEL

7.1. Cycle time comparison

Apart from the results on the dashboards, where we can compare the basic model and the improved and the effect on each section with the changes made, we can look at the times and therefore the costs of the production.

All of the times reflected, and the estimations of costs are calculated for the production of 600 chairs or a week´s work. To fulfil the production mentioned the basic model needed around 60 hours of work. On the other hand, the improved model only needed around 43 hours. Looking at the cycle time we can see the improvement made to the production line.

$$Total\ Cycle\ Time = 217044s = \ 2\ days\ 12\ hours\ 17\ minutes$$

$$Total\ Cycle\ Time\ after\ improvements = 153961s = 1\ day\ 18\ hours\ 46\ minutes$$

The new cycle time is 153961s, which means that we have managed to improve our times by 29,07%.

To achieve these results, we added additional operators and machines. It is not enough to make such a comparison just based on time, because we have to take into account the added costs of these operators and machines, and if reducing time means increasing costs, such an improvement should not be considered.

To evaluate whether such improvements have positive results, we will need to compare the costs involved in each model.

7.2. Cost comparison

To obtain good method of comparison of each production model we are going to calculate the costs involved in the manufacture of a single chair for both. To calculate those costs, we divided them into costs of operators, costs of machines and costs of transportation. To be as accurate as possible we extract the times that we needed from the models, but we also had to estimate some data that were not specified.

7.2.1. Cost of operators

First, we searched for an estimation for the annual salary of the operator depending on the preparation needed to carry out the work. Then if we assume that the operators work 5 days a week and 8 hours each day, we can estimate the hourly wage. To know how long the operators work to complete the production of a week or 600 chairs we need the cycle time, that we know for both models. With the cycle time in hours and the hourly wage we can calculate the operator´s cost for the production cycle mentioned. If we divide that cost by the number of chairs, we obtain the operator´s cost per chair. So the cost of each process will be divided between the operators´ costs and the machines´ costs.

	Annual Salary	Hourly Salary	Hours of production	Operator´s cost	Operator´s cost/chair
Storage Operator 1	16000	8.333333333	60.28994167	502.4161806	0.837360301
Storage Operator 2	16000	8.333333333	60.28994167	502.4161806	0.837360301
Cutting Operator	18000	9.375	60.28994167	565.2182032	0.942030339
Bending Operator	18000	9.375	60.28994167	565.2182032	0.942030339
Punching Operator	18000	9.375	60.28994167	565.2182032	0.942030339
Welding Operator 1	24000	12.5	60.28994167	753.6242709	1.256040451
Welding Operator 2	24000	12.5	60.28994167	753.6242709	1.256040451
Painting Operator	20000	10.41666667	60.28994167	628.0202257	1.046700376
Assembly Operator 1	18000	9.375	60.28994167	565.2182032	0.942030339
Assembly Operator 2	18000	9.375	60.28994167	565.2182032	0.942030339
Packaging Operator	16000	8.333333333	60.28994167	502.4161806	0.837360301
Forklift Driver 1	18000	9.375	60.28994167	565.2182032	0.942030339
Forklift Driver 2	18000	9.375	60.28994167	565.2182032	0.942030339
				TOTAL	**12.66507455**

	Annual Salary	Hourly Salary	Hours of production	Operator´s cost	Operator´s cost/chair
Storage Operator 1	16000	8.333333333	42.76671667	356.3893056	0.593982176
Storage Operator 2	16000	8.333333333	42.76671667	356.3893056	0.593982176
Cutting Operator	18000	9.375	42.76671667	400.9379688	0.668229948
Detailing Operator	18000	9.375	42.76671667	400.9379688	0.668229948
Bending Operator 1	18000	9.375	42.76671667	400.9379688	0.668229948
Bending Operator 2	18000	9.375	42.76671667	400.9379688	0.668229948
Punching Operator	18000	9.375	42.76671667	400.9379688	0.668229948
Welding Operator 1	24000	12.5	42.76671667	534.5839584	0.890973264
Welding Operator 2	24000	12.5	42.76671667	534.5839584	0.890973264
Painting Operator	20000	10.41666667	42.76671667	445.486632	0.74247772
Assembly Operator 1	18000	9.375	42.76671667	400.9379688	0.668229948
Assembly Operator 2	18000	9.375	42.76671667	400.9379688	0.668229948
Packaging Operator	16000	8.333333333	42.76671667	356.3893056	0.593982176
Forklift Driver 1	18000	9.375	42.76671667	400.9379688	0.668229948
Forklift Driver 2	18000	9.375	42.76671667	400.9379688	0.668229948
				TOTAL	**10.32044031**

7.2.2. Cost of machines

Like we did with the operators, first we searched for an estimation for the prices of the machines needed to achieve the process. We decided that for the assembly and packaging process there is no need of buying a specific machine, so the cost of that process will be estimated only by the operator´s cost. We also searched for the useful life for each machine, with that information and like we did with the operators, only considering that the machines are used 5 days a week 8 hours, we can calculate an approximation for the hourly cost. But, in this case, unlike the operators´ calculation we are going to use the process´ times not the cycle time. To have more precision we extract that information from the corresponding model. Modifying the code and adding triggers on entry and exit on the different processors we can have different timetables of the sections of each model. With these times calculated we obtain the machines´ cost for a week of production or 600 chairs. If we divide that cost by the number of chairs, we obtain the machine´s cost per chair. We can see that this number is lower than the operator's cost because the machines needed are not very expensive and there are supposed to be use for a long time. And their deterioration is only considered when each process is being done.

	Price	Useful Life (years)	Hourly Cost	Hours of process	Machine´s cost	Machine´s cost/chair
Cutting	1000	5	0.10416667	56.66898	5.90301875	0.009838365
Detailing	500	10	0.02604167	35.61287	0.92741849	0.001545697
Bending	4500	5	0.46875	40.71139	19.08346406	0.031805773
Punching	9000	5	0.9375	20.45248889	19.17420833	0.031957014
Welding 1	2500	14	0.09300595	22.25061	2.069439174	0.003449065
Welding 2	2500	14	0.09300595	19.92305	1.85296224	0.00308827
Painting	1000	18	0.02893519	21.79126944	0.630534417	0.001050891
					TOTAL	**0.082735076**

	Price	Useful Life (years)	Hourly Cost	Hours of process	Machine´s cost	Machine´s cost/chair
Cutting	1000	5	0.10416667	41.73598	4.347497917	0.00724583
Detailing	500	10	0.02604167	17.58178	0.457858854	0.000763098
Bending 1	4500	5	0.46875	31.61913	14.82146719	0.024702445
Bending 2	4500	5	0.46875	25.57057	11.98620469	0.019977008
Punching	9000	5	0.9375	29.38679	27.55011563	0.045916859
Welding 1	2500	14	0.09300595	12.71665	1.182724144	0.001971207
Welding 2	2500	14	0.09300595	11.44412	1.06437128	0.001773952
Painting	1000	18	0.02893519	21.85220278	0.632297534	0.001053829
					TOTAL	**0.103404229**

7.2.3. Costs of transportation

At the end we followed the same steps that we did on the costs of machines. Because at the end they are machines specifically for the transportation between process. First, we extract like we did with the machines' times from the model, but at the end the model does not have the dimensions and distances of a real production, so we calculate the average between both forklifts. For the trolleys we decided that they are activate all the cycle time

	Price	Useful Life (years)	Hourly Cost	Hours Used	Cost	Cost/chair
Forklift 1	20000	10	1.04166667	52.5458431	54.7352532	0.09122542
Forklift 2	20000	10	1.04166667	52.5458431	54.7352532	0.09122542
Trolley 1	7000	10	0.36458333	60.2899417	21.9807079	0.03663451
Trolley 2	7000	10	0.36458333	60.2899417	21.9807079	0.03663451
					TOTAL	0.25571987

	Price	Useful Life (years)	Hourly Cost	Hours Used	Cost	Cost/chair
Forklift 1	20000	10	1.04166667	37.2734343	38.8264941	0.06471082
Forklift 2	20000	10	1.04166667	37.2734343	38.8264941	0.06471082
Trolley 1	7000	10	0.36458333	42.7667167	15.5920321	0.02598672
Trolley 2	7000	10	0.36458333	42.7667167	15.5920321	0.02598672
					TOTAL	0.18139509

7.2.4. Total costs

Taking into account all of the above, we have enough information to compare and have a good estimation of the costs of the production of a chair. It is difficult to estimate the raw materials cost because normally it is cheaper to buy bigger quantities, or the providers have discounts. At the end we decided not add it because we don't have enough information and it doesn't affect on the comparison.

	Costs
Transportation	0.25571987
Operators	12.6650746
Machines	0.08273508
TOTAL	13.0035295

	Costs
Transportation	0.18139509
Operators	10.3204403
Machines	0.10340423
TOTAL	10.6052396

We can see that with the improved model we not only reduce the cycle time nearly a 30 % but also we reduce the costs of production 19,45%, which allows us to confirm that the modifications made to the production line are effective.