SIMUL8 WHITEPAPER

SIMUL8 CORPORATION

The Balancing Act: An Example of Line Balancing

Simulation expert Brian Harrington explains the key steps every Industrial Engineer should take when considering Line Balancing, and simulation can take your analysis to the next level.

Written By: **Brian Harrington**

About the Author
Brian Harrington is a Six Sigma Black Belt with 20 years operations research and simulation experience at Ford Motor Company. He designs and implements manufacturing process improvements which incorporate many conflicting objectives such as robust, flexible, and Lean systems.

Simulation expert Brian Harrington discusses how simulation can play a key part in the successful completion of a manufacturing project when the conflicting objectives of cost, quality and time all need to be delivered on.

This paper outlines the key steps to take when starting out a Line Balancing project and is an ideal guide for an Industrial Engineer. The paper focuses on why simulation is a key tool to take the project to the next level.

Line Balancing is challenging, particularly when we are limited to deterministic calculations. When designing a new line with deterministic calculations we can only approximate behaviors rather than have exact data. With so many different and potentially conflicting requirements on the system, the outcomes of a new process design, or re-design, may be difficult to predict.

Simulation can create a well-balanced line that has the flexibility to hit targeted throughput consistently. With a simple simulation of the line assembly operations we can identify system bottlenecks, run different production schedules, and evaluate the impact of design and scheduling decisions, such as buffering requirements and product mix. This "what-if" analysis can be done quickly and accurately to evaluate all the conflicting decision criteria.

The key Line Balancing steps we will focus on are:

1. **The Core Essentials**
2. **Going Beyond with Simulation**

1

The Core Essentials

When designing and managing a mixed-model line-assembly, system engineers strive to satisfy objectives such as maximizing line throughput, minimizing the number of stations, maintaining a balance of work across stations, satisfying delivery rates, accommodating product mix changes, and more. Before we move on to the more complex steps it is important to understand how many stations are required and how we assign tasks to those stations.

Key learning points:

- ➲ Determining how many stations are needed
- ➲ Assigning tasks to stations

How many stations do I need?

One of the first questions when designing a new facility or line will be; "How many stations are required? The answer is a simple calculation derived from the "Takt Time" and the "Total Task Cycle Time". The takt time is a calculation for what is required to meet demand.

Takt time = Available working Time/ Customer Demand

In this example let's say that our target is to produce 500 units per day within an 8 hour shift. Therefore, the Takt Time would be as follows:
Takt Time = 480 minutes / 500 units = 0.96 minutes = 57.6 seconds

Each station should at least have a 57.6 second design cycle time to meet market demand of 500 units. In order to know how many stations are required we need to know some detailed insight into the underlying product, bill of material, and bill of process. This is how we can establish the required tasks to assemble the product. Let's assume that this new line has 12 required steps to complete the assembly.

The steps have been labeled [A-L] and each have a unique cycle time associated to that specific task. These cycle times could have been captured using MODAPTS or actual stop watch calculations. We now have the two key pieces of information to calculate the required number of stations. The number of stations is simply calculated by the below equation.

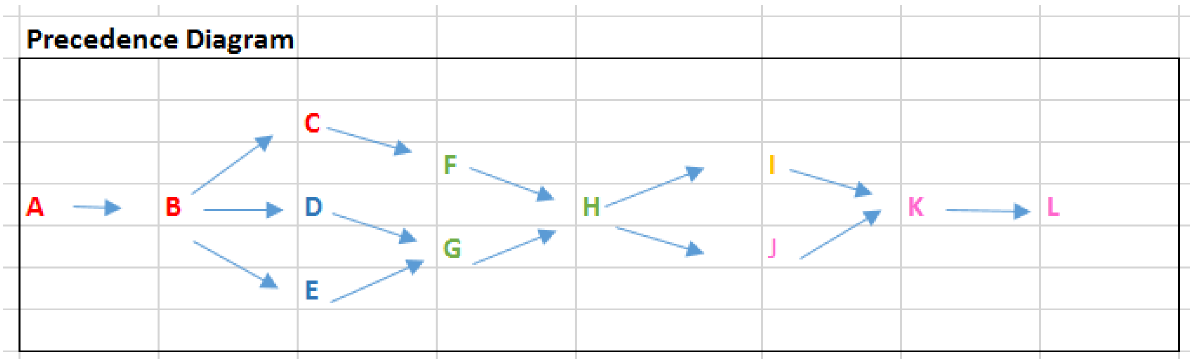
Task	Task Time (Seconds)
A	15
B	23
C	17
D	42
E	15
F	37
G	5
H	12
I	34
J	27
K	18
L	7
Total Time	**252**

Number of Stations = Total task Cycle Time / Takt Time

Assigning tasks to stations

Now we have 12 tasks that need to be accomplished within 5 stations it becomes a question of which tasks to include within a specific station. This is where the "Bill of Process" comes in; we need to know some information of the precedence or the order of the tasks. Certain tasks must be completed prior to taking action on other tasks. The "Bill of Process" is where each step or task is described to assemble the unit. It should clearly demonstrate the order of steps, including synchronous and simultaneous tasks. This is often captured in a "Precedence Diagram".

A Precedence Diagram is a lot like a process flow diagram; with shapes and arrows describing significant and critical steps within assembly of the product or service. In our example we will assume that we have been supplied with the following Precedence Diagram for our 12 tasks (A-L):



This clearly shows that task A must be completed before task B can be started. It also shows that that tasks C, D, and E can be started simultaneously after task B has been completed. Moreover, both tasks F and G must be completed before task H can start. We can now add a Precedence column to our initial Task Table:

Given the precedence we can now start assigning tasks to stations. One common approach is to use a "Task Assignment Table". This table will look at all eligible tasks to be included within a station, and keeping track of the accumulated cycle time within the station. A common scheme is to start in the order of the precedence diagram and seek the longest cycle time.

Task	Task Time (Seconds)	Precedence
A	15	
B	23	A
C	17	B
D	42	B
E	15	B
F	37	C
G	5	D, E
H	12	F, G
I	34	H
J	27	H
K	18	I, J
L	7	K
Total Time	**252**	

We start out with Station 1; the only eligible task is Task A. We then assign Task A using its 15 second cycle time. The remaining time left within the station is the Takt Time minus the assigned task; [57.6 seconds – 15 seconds] = 42.6 seconds. Therefore, we have 42.6 seconds remaining in Station 1 to assign additional tasks. The only qualified task is Task B, so we then assign Task B with its 23 second cycle time to Station 1. We now have a remaining time of 19.6 seconds.

Stn	Eligible Task	Assigned Task	Task Time	Cumulative Time	Remaining Time
1	A	A	15	15	42.6
	B	B	23	38	19.6
	C, E	C	17	**55**	2.6
2	D, E	D	42	42	15.6
	E	E	15	**57**	0.6
3	F, G	F	37	37	20.6
	G	G	5	42	15.6
	H	H	12	**54**	3.6
4	I, J	I	34	**34**	23.6
5	J	J	27	27	30.6
	K	K	18	45	12.6
	L	L	7	**52**	5.6

The next tasks are C, D, and E; but D does not qualify because its cycle time is greater than the remaining cycle time. Therefore, C and E become the next eligible tasks. Since we are using the longest cycle time rule; we will then select Task C using its 17 second cycle time. This now completes station 1 with a remaining time of 2.6 seconds; as there are no other identified tasks less than 2.6 seconds.

We are now ready to assign tasks to the 2nd station; the eligible tasks are D and E. Since D has a higher cycle time of 42 seconds it will be the first task assigned to Station 2. Task E can now close out the station with a remaining idle time of 0.6 seconds.

We then complete the remaining 3 stations using the same eligible task scheme to fill out the completed table. The newly designed line will then appear on the layout as depicted below. We can see that Station 4 is under-cycle with 23.6 seconds of idle time; but it could be the best available design according to the process precedence rules.

2

Going Beyond with Simulation

Our key steps now emphasize that with simulation line balancing can be carried out much more effectively. Adding to the previous steps we can really look further into the line balancing process and how we can help reach targeted throughputs.

Key learning points:

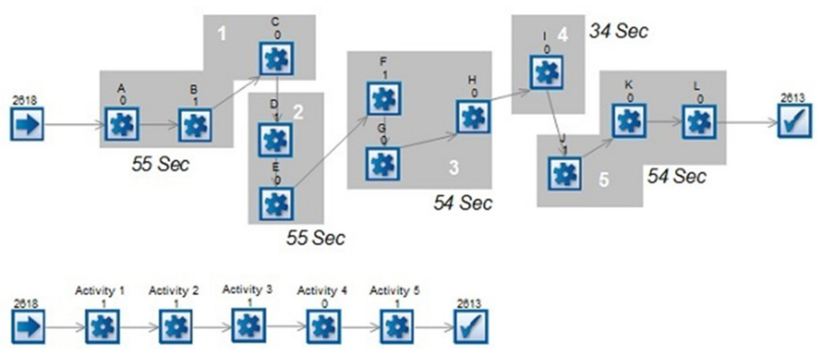
- ➲ Going Beyond Deterministic Calculations with Simulation
- ➲ Adding Operators to the Analysis
- ➲ Adding Additional Stochastic Behavior

Going Beyond Deterministic Calculations with Simulation

The deterministic Excel base tools can offer a lot of insight into designing and laying out your company's facility; but when we couple the initial designs with some simple simulation analysis we can vastly go beyond these deterministic calculations.

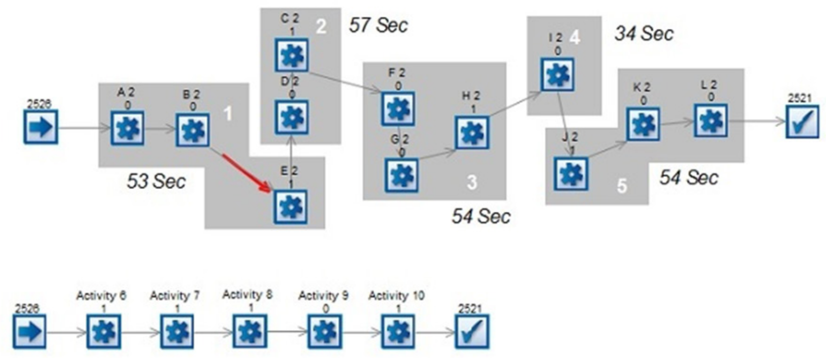
All of these charts and calculations assume no random downtime, over cycles, changeovers, etc. They also do not consider shared operators capable of working on various tasks. As soon as we bring this stochastic variation into the analysis; the calculations would become very complex. Although, with the use of simulation these stochastic parameters are handled with ease!

Precedence Diagram 1: Longest Eligible Task



Simulation can easily be applied at the start of your Precedence Diagrams. This is a natural starting point as it is a depiction of the process steps and routing links. Each line can be modeled, and all eligible routing variations can be compared by simply changing routing links. These small models are at the task level, once the best precedence diagram scheme model has been proven. They can be then rolled-up and modeled at the station level with the achieved cycle time.

Precedence Diagram 2: Exploring other Eligible Task Options



The above simulation shows two precedence diagrams that can be compared against throughput or number of units delivered. Remember our original target was to achieve a minimum of 500 units per shift. This model is set up to explore a week's worth of production; hence a target of 2500 units per week. We can clearly see that the first model has greater capability as it is achieving 2613 units versus the second feasible solution which is at 2521 units per week.

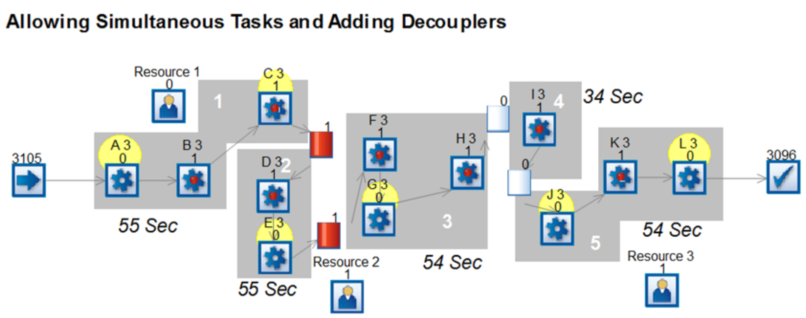
Adding Operators to the Analysis

Another typical consideration is adding manual operations or resources to the precedence diagram models. This is easily accomplished within a simulation model. Let's consider that the following tasks (A, C, E, G, J, and L) are manual operations which require an operator. All of the other operations are assumed to be automated.

Typical questions that arise are:

1. How many operators are required?
2. What impact do they have on throughput?
3. Can we meet our target of 500 units per shift with 3 operators?
4. Which operator is potentially causing losses?

All these questions can be answered within the simulation by testing various operator schemes including associated travel time between stations. By simply adding recourses to the above tasks we can examine the effects of manual operations within the precedence models.



In this simulation we placed 3 operators to cover the 6 manual operations within the Line. The first operator handles task A & C of station 1. The second operator covers task E & G, hence this operator may incur additional travel time to walk between stations. The third operator covers tasks J & L within station 5. The overall number of units produced has dropped from 2613 to 2521; which is slightly above the target.

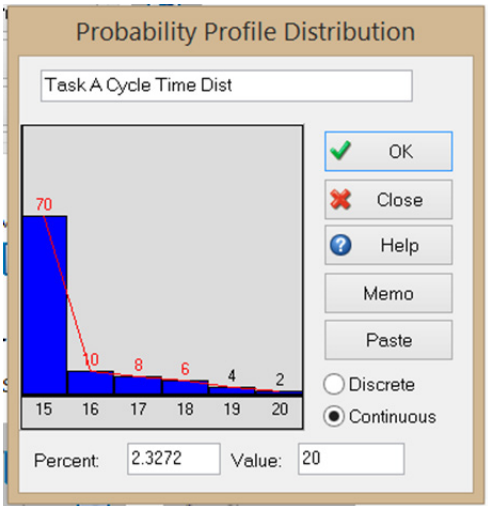
From the analysis we can determine that the shared resource between station 2 & 3 is causing the slight loss in throughput. It could be considered within the design specifications; but it also should be noted as a potential sensitive area within the line. Therefore, it would likely cause significant losses if it experiences over-cycles, changeovers, large downtime stoppages, etc.

These are very typical scenarios that exist; as most companies strive to keep their work force to a minimum and achieve high utilization per operator. The simulation will offer great insight into optimal placement of operators and achieving targeted resource utilization. Moreover, not just target utilization, but achieving a balanced utilization of the line side operator crew.

Adding Additional Stochastic Behavior

Over Cycles

We can now explore a scenario that takes over-cycles into account. For example, all the above manual operations can experience up to a 5 second over-cycle 30% of the time, so, instead of "Task A" using a fixed 15 seconds, we can use a user defined distribution that captures the random 30% over cycle condition.



This is known as a **Probability Profile Distribution**, in this case 70% of the time it will yield 15 seconds and the remaining skewed tail will account for over cycles all the way to 20 seconds. Hence, the cycle time will range from 15 to 20 seconds. We would then place this type of distribution on the remaining manual operations (C, E, G, J, and L) using their respective parameters. When we run this scenario with all of the over cycle distributions on the manual operations we fall below the targeted throughput by 28 units only achieving (2472 units / week).

Downtime

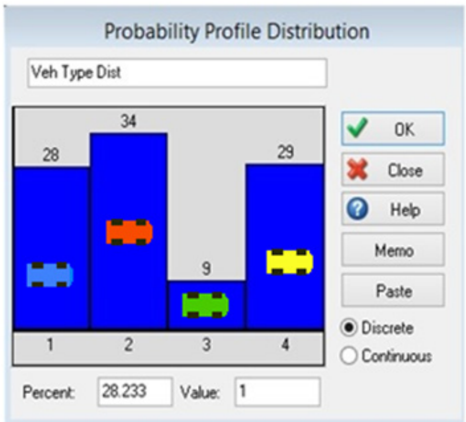
We could also add downtime to the simulation by applying "Mean Time Between Failures" (MTBF) and "Mean Time to Repair" (MTTR). For example, all of the automated tasks might be 95% efficient with a MTBF of 90 minutes and an average repair time of 5 minutes. When we run the previous scenario with the additional downtime on all of the automated tasks (B, D, F, H, I, K) the throughput significantly drops to 1777 units / week. This is where more advanced line balancing techniques are required; like allowing additional units into stations referred to as simultaneous tasks.

If we consider the first station with tasks (A, B, & C), where A and B require an operator and B is the automated task. Maybe the next task A can be started while task B is cycling. As you can imagine, there are many different types of schemes that could be built into the process steps of station 1. Station one could have an input buffer and an output buffer, commonly referred to a decouplers, these are techniques used to minimize the losses of sequential downtime on synchronous stations. All of these design scenarios would be extremely difficult without the use of simulation.

As you can see from the scenario below which allows for simultaneous tasks within stations and also has small decouplers (buffers) between stations, we can bring the throughput back to over target even with the additional downtime. Our next step would be to back off on some of the protective actions, and to hone in on which actions are feasible, affordable, and required to achieve targeted throughput.

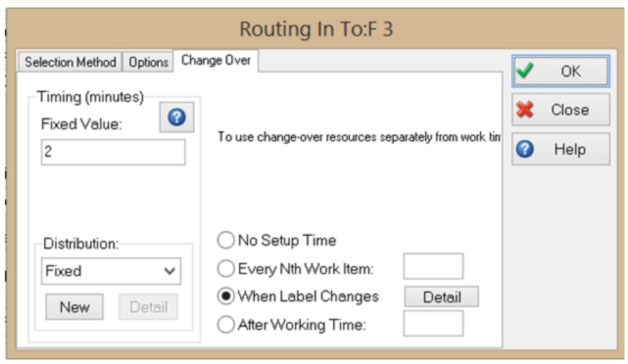
Changeovers

Lastly, let's examine the impacts of changeovers. Changeovers often occur when the product type changes. In other words, our facility might be producing 4 unique variations of the product. This is usually referred to as the "Product Mix". The facility might build these variations of the product in batches or completely random. Either way, this adds great complexity to line balancing; as each station may have unique cycle times per type. Furthermore, it can incur additional change over time within the station as a new product type enters the station.

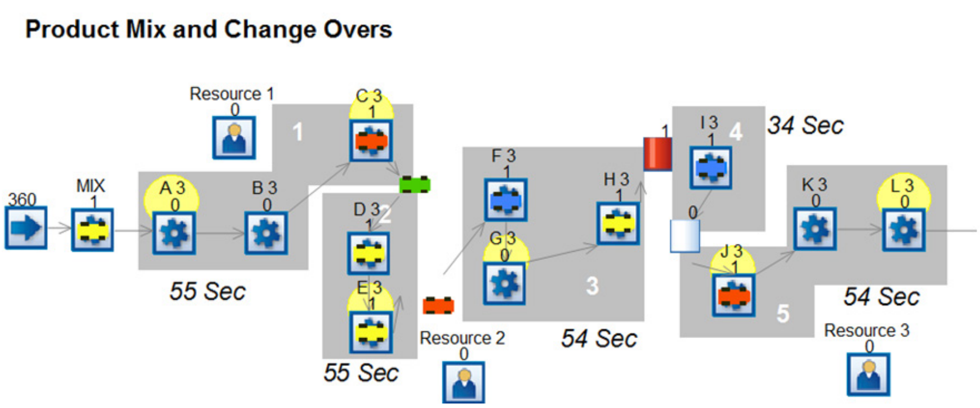


As soon as we add a product mix to the analysis the deterministic calculations even become more complex. With the use of simulation this can be made simple. It is very common to have product mixes within a model, unique cycle times according to type, and changeovers as necessary. In this example we see a product mix of 4 different types of vehicles (28% Blue, 34% Red, 9% Green, and 29% Yellow). We can easily run this product mix through the above simulation scenarios to see the impacts of unique cycle times according to vehicle type, and the impacts of changeovers.

In this example we can see the 4 vehicle types traveling through the model, where each type can have its own unique cycle time and can invoke a changeover. In this case a 2 minute changeover will occur when the vehicle type changes. This would more than likely require a batch build scheme to minimize the number of changeovers. One scheduling scheme would be to group the daily orders of 500 units in 4 unique groups.



Batch Build Schedule approximately as follows:
500 * 0.28 = **140 units**, 500 * 0.34 = **170 units**, 500 * 0.09 = **45 units**, 500* 0.29 = **145 units**
This would reduce the average number of changeovers to 4 per shift.



Concluding thoughts on Line Balancing

As we can see Line Balancing can become very complex especially when we start adding realistic stochastic parameters to the equation. The deterministic calculations are very useful for setting up the framework for designing a new line. They can be adjusted to test out worst case/best case, but can only approximate a steady case, such as a random product mix through a flexible manufacturing system. This is where simulation can add a world of realistic behavior to the analysis.

What is evident is that we can build small scale models that follow the deterministic line balancing paradigms, but with all the dynamic features of a complex model. We can easily add shared resources, simultaneous tasks, mix product variants, downtime, over-cycles, and changeovers to the analysis. **Thereby, achieving a well-balanced flexible manufacturing line that will hit their targeted throughput.**

Further Reading

More Than a Cycle

If you'd like some background reading on this topic please see our paper "More Than a Cycle" which helps understand everything you need to know about cycle times.

Line Balancing at Chrysler

Read about how SIMUL8 have developed a relationship with Chrysler over a number of years. Across various line balancing projects SIMUL8 helped the Chrysler simulation team save **millions of dollars.**