What is the DAM Problem with Scheduling?

by Gray McQuarrie
Grayrock & Associates

"Never tell people how to do things. Tell them what to do, and they will surprise you with their ingenuity."
~General George S. Patton

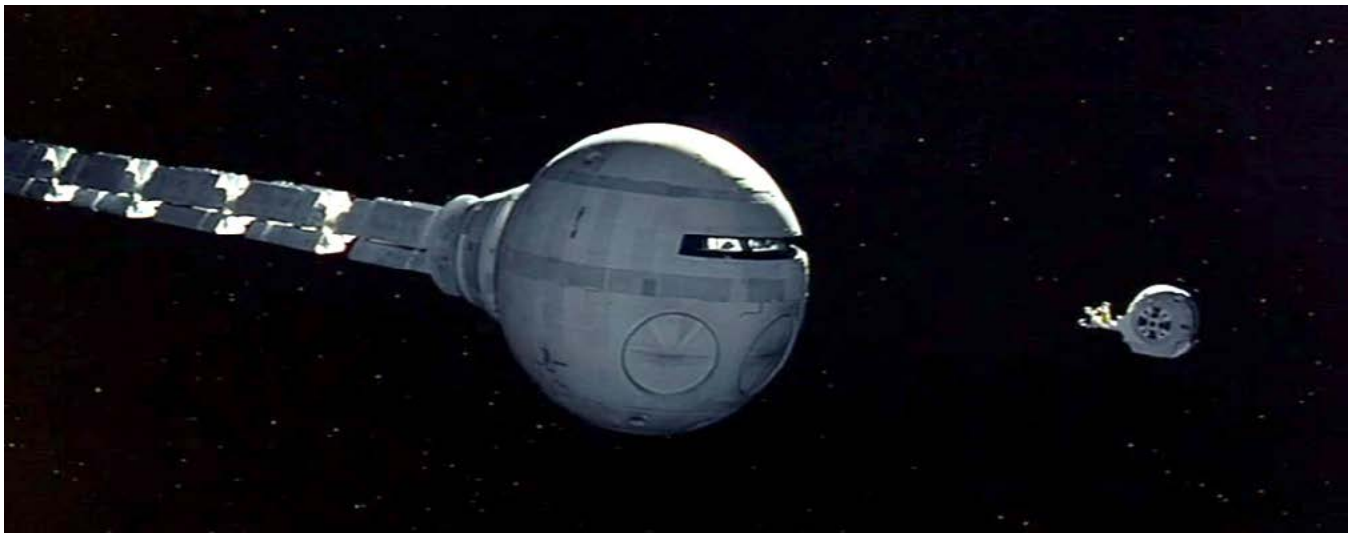
My good friend, retired U.S. Army Colonel Carl Schott, was once the commander in charge of the units that would have deployed tactical nuclear weapons if Russian tanks had ventured into West Germany. He once told me, "The very best plan never survives the first bullet." Part of his background was flying helicopters in Vietnam where, in one type of mission, he flew at night to draw North Vietnamese fire so that the gunships above him could identify the enemy and kill them.

Just like you, Carl had to deal with immediate reality. Rarely did anything go according to plan. Just as with the fog of war, in the fog of production, trying to figure out what is happening with the customer orders can be, at best, frustrating. We often find our master schedule is hardly that! It can cause any one of us to say, "What is the DAM problem with scheduling?"

Before I answer this question, I want to go off on one of my tangents. Recently, I was having morning coffee with Carl at Starbucks. He told me the German commanders told him that in World War II, they were extremely frustrated with the American infantry. For example, it was almost impossible to force the Americans into a defensive position and keep them there. The Americans were always improvising and deviating from the plan. The Germans couldn't predict how the Americans would respond in a given battle. If the Germans tried to deviate from their master plan, which often came from Hitler, they could get shot. The battle for Stalingrad exemplified the German war machine's rigidity to the plan and its inability to improvise.

What does this have to do with our master schedule and how we dispatch jobs in the shop? Production enterprise-level intelligent software systems, which tell us how material is to move on the factory floor, can create a rigidity that has the potential to sink us. I like to refer to these systems as HAL, the master computer system that controlled everything on the mothership in, "2001: A Space Odyssey." We are not to question HAL; we are simply to follow his orders: HAL, open the pod bay doors.

Let me illustrate why scheduling is so hard. In fact, in technical terms, scheduling is an NP-hard problem.



2001: A Space Odyssey, Metro-Goldwyn-Mayer 1968

First, there is a factorial relationship that appears simple. For a plant with a single machine and three jobs you would have 3! (3 factorial = 3 x 2 x 1 = 6) possible jobs sequenced such as 1, 2, 3 or 3, 1, 2, and so forth. How would you determine which sequence or schedule is best? It's rather easy in this case. For example, if job 1 were due first, job 3 next, and job 2 last, then with an earliest due date rule we have 1, 3, 2. If jobs 1 and 2 were volume orders and no setup was required going from job 1 to job 2 (or vice versa) and they were due much earlier than job 3, well, it's looking like job 3 should be last. In addition, if job 3 had a very long complicated setup going from either job 1 or job 2 and it was a small volume order, then the optimum schedule has to be 1, 2, 3 or 2, 1, 3. Simple logic. No need for HAL.

Record #	M1 Time (min)	M2 Time (min)	Order	End Time (min)
1	1.00	20.00	1	21.00
2	2.00	19.00	2	40.00
3	3.00	18.00	3	58.00
4	4.00	17.00	4	75.00
5	5.00	16.00	5	91.00
6	6.00	15.00	6	106.00
7	7.00	14.00	7	120.00
8	8.00	13.00	8	133.00
9	9.00	12.00	9	145.00
10	10.00	11.00	10	156.00
11	11.00	10.00	11	166.00
12	12.00	9.00	12	175.00
13	13.00	8.00	13	183.00
14	14.00	7.00	14	190.00
15	15.00	6.00	15	196.00
16	16.00	5.00	16	201.00
17	17.00	4.00	17	205.00
18	18.00	3.00	18	208.00
19	19.00	2.00	19	210.00
20	20.00	1.00	20	211.00

Table 1: Best case for start order for the model in Figure 1. The order column is the order the jobs are to be run. The "EndTime" is the time each job was completed. Note: Every job requires a total of 21 hours of production time. Record 20 shows the last job was completed after 211 hours.

But what happens to this problem with modest increases in size? Factory Physics, by Hopp and Spearman, provides a discussion on this subject in their chapter "Production Scheduling." For example, they illustrate the 10 x 10 problem popular in operations research: finding the optimum schedule for 10 jobs with 10 machines. Just for fun, write down as many different production schedules as you think are possible. For example, is a billion too many or too few? Through just one machine you would have 10! (10 x 9 x 8 x 7 … x 1) or more than 3.6 billion scheduling possibilities. To get the number of possibilities for all 10 machines it is 10! x 10! x 10! x … 10 times. This is approximately 4×10^{65} possible production schedules, which far exceeds the number of atoms on Earth! This takes many hours of computer time to solve. Since many U.S. plants are large, quick-turn shops dealing with hundreds of jobs running

Record #	M1 Time (min)	M2 Time (min)	Order	End Time (min)
1	1.00	20.00	17	231.00
2	2.00	19.00	19	266.00
3	3.00	18.00	9	136.00
4	4.00	17.00	10	153.00
5	5.00	16.00	18	247.00
6	6.00	15.00	2	39.00
7	7.00	14.00	13	189.00
8	8.00	13.00	3	52.00
9	9.00	12.00	12	175.00
10	10.00	11.00	16	211.00
11	11.00	10.00	11	163.00
12	12.00	9.00	20	275.00
13	13.00	8.00	7	103.00
14	14.00	7.00	15	200.00
15	15.00	6.00	6	88.00
16	16.00	5.00	5	72.00
17	17.00	4.00	14	193.00
18	18.00	3.00	1	21.00
19	19.00	2.00	4	54.00
20	20.00	1.00	8	116.00

Table 2: Random case. Note the jumbled run order. Record 12 shows the last job was completed after 275 hours.

through easily 10 or more machines, finding the optimum schedule could take HAL 70 or more years to figure out even if he could process 1 billion scheduling scenarios a second! In other words, "I am sorry Dave. I can't do that."

What can we mere humans do? Table 1 shows 20 jobs and their process times for two machines, M1 and M2. Notice that the total time to make any one job is always 21 hours. How many scheduling possibilities are there? Well 20! x 20!, or about 6 x 10^{36}. How can we figure out the best solution within the next minute or so? I put together a simple discrete event model to help us (Figure 1). Notice that either machine 1 or machine 2 could be a bottleneck, depending on what job is being run. You don't want machine 2 to be idle for long, so you might run the short job first on machine 1 so machine 2 gets going instead of sitting idle.

Does it matter in what order the jobs are released? Well, if we run the jobs in random order (Table 2), it took 275 hours. If we run the jobs starting with the fastest jobs in machine 1 first, and work our way down the list, then we have minimized the total process time, which is 211 hours (Table 1). If we reverse this order we get 320 hours (Table 3). So the answer is yes; the order in which the jobs are released does matter! Minimizing the makespan (total time it takes to complete all jobs) on two machines is called Johnson's rule. Unfortunately, as we add machines, it blows up into an NP-hard problem again.

Record #	M1 Time (min)	M2 Time (min)	Order	End Time (min)
1	1.00	20.00	20	320.00
2	2.00	19.00	19	300.00
3	3.00	18.00	18	281.00
4	4.00	17.00	17	263.00
5	5.00	16.00	16	245.00
6	6.00	15.00	15	230.00
7	7.00	14.00	14	215.00
8	8.00	13.00	13	201.00
9	9.00	12.00	12	188.00
10	10.00	11.00	11	176.00
11	11.00	10.00	10	165.00
12	12.00	9.00	9	153.00
13	13.00	8.00	8	140.00
14	14.00	7.00	7	126.00
15	15.00	6.00	6	111.00
16	16.00	5.00	5	95.00
17	17.00	4.00	4	78.00
18	18.00	3.00	3	60.00
19	19.00	2.00	2	41.00
20	20.00	1.00	1	21.00

Table 3: Worst case. Record 1 shows the last job was completed after 320 hours.

An obvious take-away is this: When you are confronted with a bottleneck, get the fast jobs out upstream to feed it first. When the bottleneck is running, start producing the jobs that take longer in the upstream operations. Of course, always pay attention to promised due dates. By doing this, you have reduced the number of machines to consider in your plant. And you have reduced the number of possible schedules to look at. In this way you have reduced the hard problem into a much simpler one. After all, we should be able to manually open the pod bay door.

Some years ago I walked into a PCB facility that was making very complex HDI product, but wasn't making any money. By focusing on constraining the WIP (CONWIP), identifying and focusing on the production bottlenecks to help develop a scheduling paradigm, working side by side with the operators to devise their own rules for dispatching jobs, and implementing a production pull system using Kanbans, profitability soared, WIP was reduced by two-thirds, and the standard production lead time decreased from 15 to four days with one-day quick-turns.

Let me run through some highlights.

I found three primary constraints: 1) lamination, 2) laser drill, and 3) electrical test. For a variety of reasons, the upstream constraint, lamination, was chosen as the primary constraint that would dictate the heart beat: the takt time for the factory. It was also used to determine the production batch size, the number of jobs to be started at once, and the frequency for production starts. Within each production release a set allocation for quickturns was also established. What this made clear to everyone in the plant was there was a finite capacity.

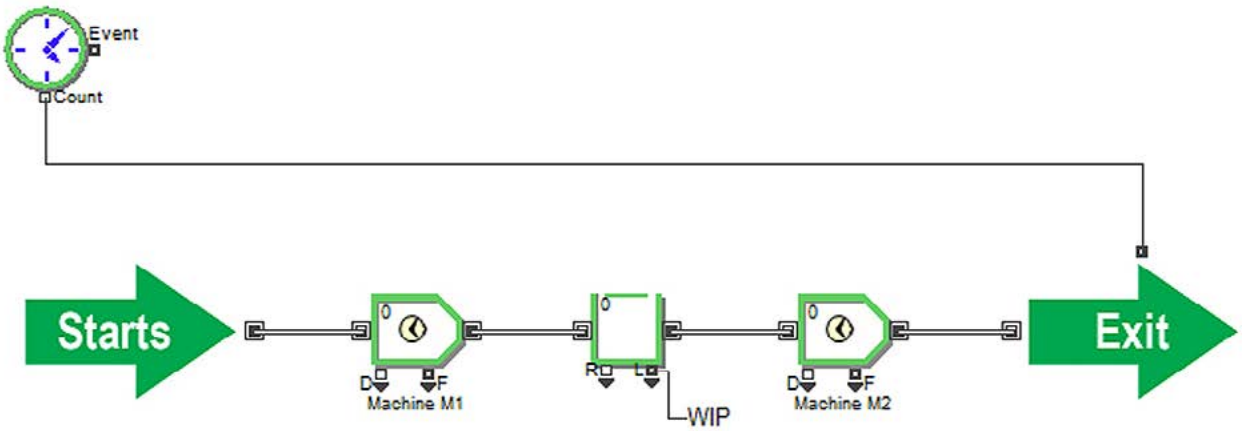


Figure 1: The Simple Discrete Event model. Tables 1, 2, and 3 show different sequences of jobs that come out of the "Starts" arrow, which have different processing times for M1 and M2. Note if the process time for M2 is long then it will be a bottleneck. Similarly if an M1 job is long, M2 will be waiting for work. How do we derive the best order or scheduling sequence? What will the WIP do between M1 and M2?

Overall profitability would be determined by how this capacity was managed. For example, if the quick-turn allocation wasn't full, everyone in the plant would know and begin to ask questions. This simple transparent scheduling system got everyone involved. Everyone understood what had to happen on a daily basis.

In order to manage the WIP level, the operators and supervisors created a Kanban system during their Kaizen. They created shelves to differentiate between sub-assemblies and repetitive processing. They created colored tags to go with the traveler to signify visually which jobs were due first. In order to figure out how to dispatch work from the Kanbans, they figured out ways the three bottlenecks would not starve. For the press area this meant organizing all material required for layup to be ready. The team was rather ingenious in how they organized this and made it work. Since most panels from the lamination department fed the laser drill department, the operators and leads, on their own, discovered that running the jobs with short lamination cycles at the beginning of the shift and abiding by the due dates was best.

I visit many factories that use computer dispatch, and I always find the operators have great ideas for improving the plant's flow, many of which are consistent with operations research theories even though they never had a class. Yet, despite their intelligent ideas, their voices weren't heard. Everyone must follow the computer dispatch list. Often when I look at the dispatch, I see jobs that have sat on the floor for 50 to 100 days or more! How can such a thing happen? It happens when you pay your employees not to think.

Months after the Kanbans were setup, I walked over to the press area where the WIP between lamination and laser drill was too high and violated the Kanban rule. I said to the lead, "Good grief, what are you doing?" The lead waved his hands as if he was a magician pretending to make it all disappear. Then he turned to me and said, "Don't worry about it."

He understood what was going on, and I did not. Let me explain. Figure 2 shows how the WIP changes for Table 2—the random case. Figure 3 shows how the WIP changes for the best case. Note that the maximum WIP for the best case, which got all of the jobs done the quickest, had 2x more WIP than the random case! Moreover, if the WIP between M1 and M2 is constrained to two units (Figure 4 shows the WIP result), then the time to make all of the jobs is much longer: 278 hours versus 211 hours. Not only that, the average cycle time was longer too: 153.5 hours versus 144.5 hours. This is an example where more WIP, not less WIP, makes the products move quicker! Why? Because we

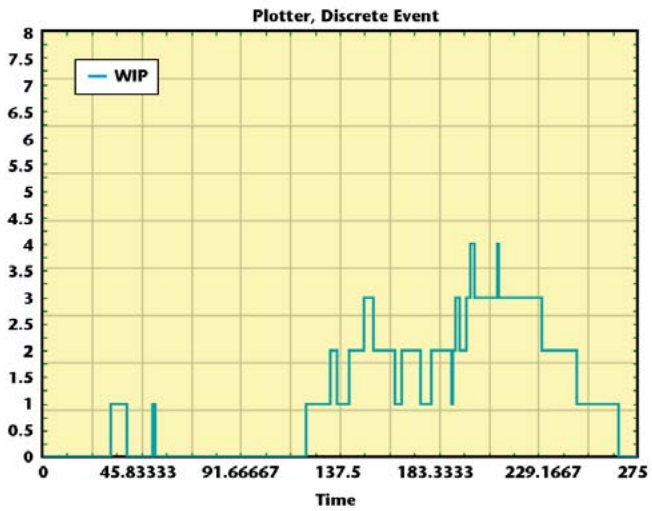


Figure 2: The WIP for the random case, which shows how WIP changed over the course of the jobs being run in random order (Table 2).

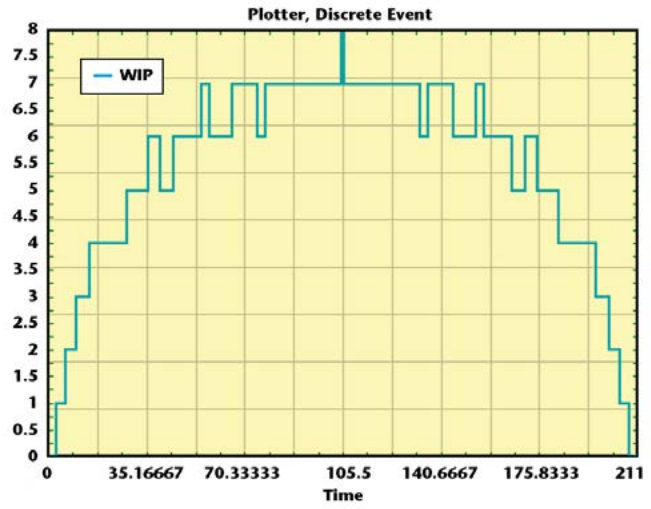


Figure 3: The WIP for the best case.

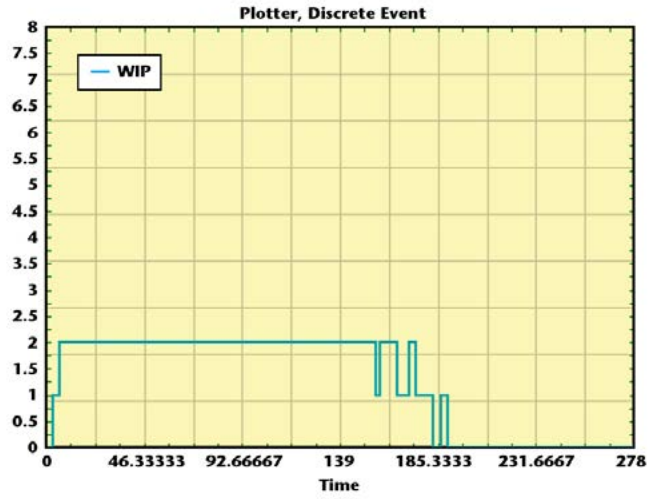


Figure 4: The constrained WIP case.

are dealing with jobs that have varying process times between two machines. Constraining the WIP causes blocking of machine 1, and as jobs cycle through in their sequence, eventually machine 2 will be starved. The leads and operators saw this, so they let the WIP and inventory float unconstrained; in this way they saw that both areas would always be producing.

How do you get everyone on the same page so this is understood? You won't do it with a spreadsheet. The only way you can see this type of dynamic behavior, without spending hours working on the plant floor, is building a simple model like the one in Figure 1. In this way, HAL serves us instead of the other way around. In this way HAL makes us all much smarter.

Figure 3 shows the relatively large amount of WIP that had the jobs going through the fastest with the shortest average cycle time when they were sequenced with the best case scenario (Table 1). This violates what we have come to understand about a low amount of WIP improving cycle time.

Figure 4 shows the WIP constrained to a maximum of two items, which if the jobs had the same process times, should be enough to keep both machines going. But this isn't the case. This produced a much worse result, though in Lean thinking and single piece flow thinking, this should have provided the best case and fastest cycle time, particularly in a scenario where only one item can be processed at a time. **PCB**

Gray McQuarrie is president of Grayrock & Associates, a team of experts dedicated to building collaborative team environments that make companies maximally effective. To read past columns, or to contact McQuarrie, click here.