

A method and layout of serial-parallel scheduling problem

György Schuster, Tamás Sándor

H1084 Budapest, Tavaszmező 15-17
hal@k2.jozsef.kando.hu,
sandor.tamas@kvk.bmf.hu

Abstract: Some stations of production line have longer cycle time than the production period of the whole production. In these kind of cases parallel processing methods has to be applied. Scheduling of this method can have several problems, therefore it is very important that the geometrical layout and control meet all the requirements. Our suggestion shows a new method to solve the known problems of those kind of stations.

Keywords control, scheduling

1. The problem

Parallel processing [1] is necessary if a station cycle time is longer than the cycle time of the production line. If this method is applied you have to face to some problems, those are:

- Dead-lock, which means that the whole system can get into a state where its activity is impossible. The reason for this is the station have run out of resources.
- The resources of the station cannot be assigned optimally. As a consequence more resources have to be applied.

If the environment is fully determined the approach of produced units is known and cycle time is also well determined a quasi satisfactory solution can be found. Unfortunately these conditions are not fulfilled in every case.

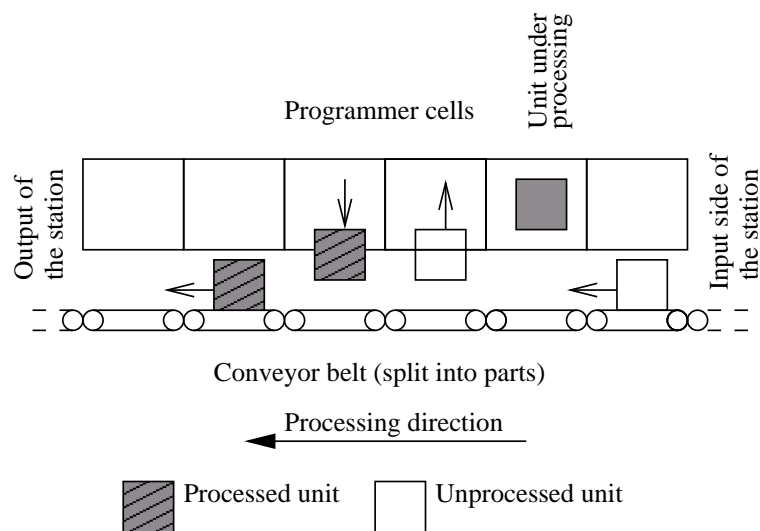
Some designers don't recognise the above mentioned problems.

As a good example we have chosen flash down loading (FDL) as the target of investigation. This operation is widespread in case of production of electronics.

FDL is a better example than any production of mechanical device. The reason for this its uncertainty a little bit higher than any phase of production of those devices.

The reason for the above mentioned uncertainty is the programming procedure can be disturbed by several other environmental effects. In these cases the unit must be reprogrammed partially or fully. Consequently programming time is not fully determined.

Let us consider the following situation. We have a production line whose cycle time is T_c and we have a phase of production FDL, which takes T_{FDL} . The problem is $T_{FDL} > T_c$. As a logical consequence we should apply more than one FDL station. The theoretical minimal number is $n_{FDL} = \lceil \frac{T_{FDL}}{T_c} \rceil$. We start our discussion with an existing FDL station you can see in 1. figure.

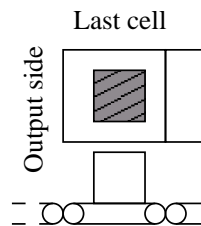


1. figure: Non-optimal layout

The conveyor belt carries the non-programmed units in front of the FDL cells. If there is an empty cell the units will be pushed into it to be processed. If the desired procedure is done the unit will be moved to the conveyor belt to be carried off the station. As it can be seen the last (n^{th}) cell can be used optimally.

The conveyor belt sometimes may be the bottle neck of the production. For instance the n^{th} cell contains a unit being processed and there is a non-processed unit at the n^{th} position on the conveyor belt. This situation is

a classical dead-lock. The station has run out of a resource, which is now the conveyor belt. This can be seen on 2. figure



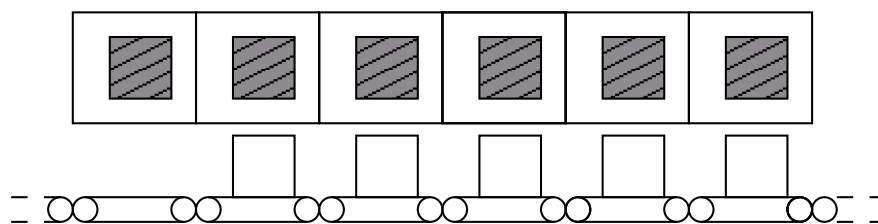
2. figure: A dead-lock situation

As it can be seen the non-processed unit at n^{th} position locks the n^{th} cell. The conveyor belt is not allowed to move the non-processed unit since the desired technological step of the station would be omitted. The most important rule is here — to avoid this situation!

Another problem is this kind of layout needs more cell than the optimal. Let us consider the following situation.

Each cell contains a unit, the conveyor belt has brought as many unit as it possible into the station — of course the last position is empty on the conveyor belt. In this case every cell must waiting for the following one. Depending on the process time, the speed of conveyor belt and the cell size this wait can be quite long.

The sketched situation holds the upper boundary of process time of the station. A logical solution would be increasing the speed of the conveyor belt.



3. figure: Full state of station

Thanks to the fact that the station conveyor belt can be full of units the speed of the conveyor belt speed doesn't need to be differ significantly from speed of the whole production line.

The situation on 3. figure can happen when production line starts. Under normal circumstances, when no errors emerge the distribution of processed and unprocessed units is quite complicated. Considering this problem and taking into consideration that used controllers are PLCs developer use the so-called "greedy" or "catch it if you can" algorithm.

"Greedy algorithm" means that a cell is empty and there is a unit in front of it the unit will be loaded into the cell to be programmed.

The problem is greedy method doesn't always match the optimal algorithm. We also have to take into consideration some cell could be out of order. Engineers count on this situation that's why they implement more cells than necessary.

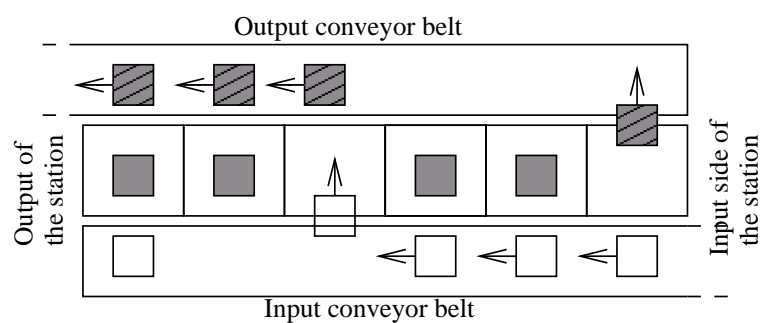
But there is another effect of cell break down. This may cause special situation ie. the dead-lock problem shifts backward.

To tell the truth it is very difficult to implement an optimal algorithm into PLC.

2. Suggested solutions

2.1. Two-belt solution

To avoid dead-lock problem the first solution is the so-called two-belt layout in which two conveyor belts are applied. The first belt is to bring units to cells the second belt carries off them. Speeds of belts are the same as the speed of production line.



4. figure: Two belt layout

The dead-lock situation is solved by this layout. Its main disadvantage is two conveyor belt is applied.

Scheduling is quite simple but sometimes collision can happen. That means a unit may have to wait for an empty place on the output conveyor belt in the actual cell.

As it can be seen this kind of situation won't cause any delaying in the production line since the unit in the cell is locked by a previously processed unit.

Let us consider the following situation. There are more than one cell empty. The input conveyor belt is full of non processed unit. In this case the cell distribution depends on the units position.

If we apply two-belt layout the production line won't be starving because any collision can happen on the output belt.

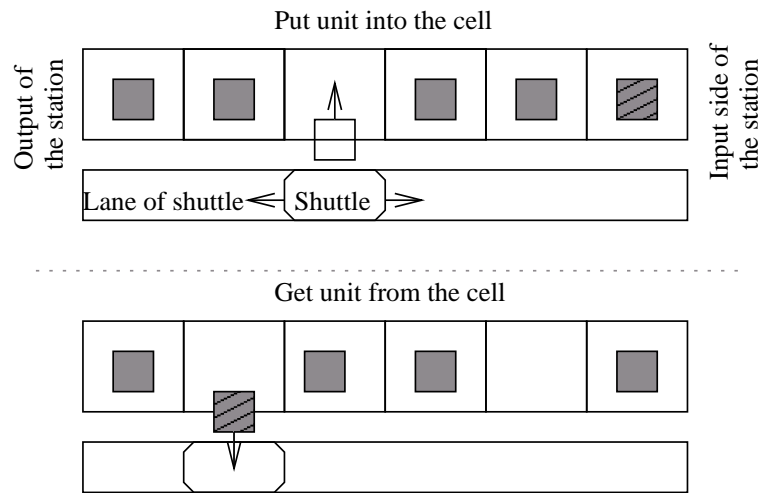
If there is a collision on output belt there is at least one programmed unit on it. This or these unit(s) will be the input of the remaining part of production line.

If the process time of each unit is the same this kind of distribution satisfactory. But in practice this condition is almost never granted. Therefore another layout and scheduling method should be found to control the station in optimal way.

At this point emerges the idea why two conveyor belts should be applied.

2.2. Shuttle solution

In this solution only one conveyor is applied to deliver processed and non-processed unit. The conveyor speed is at least 10-15 times higher than the average speed of production line. This layout can be seen on the 5. figure.



5. figure: Shuttle solution

The first question is the priority of bring in or carry out operation. Of course the carry out procedure has higher priority since it makes empty cells to be processed other units.

If there is a unit waiting for an empty cell but more than one cells are ready to accept it. The question is which cell should be chosen. There are two possibilities:

1. Decision about geometrical base. These are FIFO or LIFO solution [3].
2. Decision based on statistical features of cells. The first method has two possibilities the so called FIFO and LIFO way. These are almost equal because the sum of movement in both cases are the same. For instance we use FIFO way the bring in movement is short consequently carry off movement will be longer. The second method based on the statistical features of cell such average process time and standard deviance of process time implying delivery. If we have more than one empty cells the scheduling has the following steps:
 3. Choose the cell, which has the shortest average process time.
 4. If two or more cells have the same average process time choose the cell, which has smaller standard deviance. In the beginning we don't have any statistical values that's why one of the first methods should be applied to collect enough data to calculate those values.

Remark: It is important collected data should be recalculated. As for us the best method is to apply running average and standard deviation. This means not more than 20-30 sample should be used consequently the statistical features have to be recalculated time after time. The advantage of this way the cell won't suffer from an accidentally mistake.

Conclusion

According to simulation we obtained the following result.

- In case of original method the optimal control is almost impossible thanks to the often collisions and insufficient resources. Dead-lock can happen.

When we used only deterministic circumstances — without any uncertainty — the scheduling was sufficient. When we applied some disturbance — the variables of simulation were modified by 5% random value — the efficiency of the scheduling broke down.

- The two-belt solution is much better than the previous one. There are collisions but dead-lock doesn't occur under normal circumstances. To control this kind of stations is much easier. In case of disturbance the efficiency was getting worse but much better than in original case. The reason for this was the increasing number of collisions.

Its main disadvantage is two conveyor belts have to be applied.

- The shuttle solution seemed the best, easy to find the optimum scheduling without any collision consequently dead-lock is also impossible. If you apply the statistical method, the scheduling is much better than the two-belt and the original methods. When we apply disturbance the efficiency of the scheduling doesn't change remarkably.

The main disadvantage of this method is the quite fast movement of the shuttle.

Authors would like to express their special thanks to Ms. Gabriella Csomár and Mr. Róbert Grusz who gave some important viewpoints and ideas to solve the discussed problem and tested the simulation.

References

- [1] Peter Brucker: Scheduling Algorithms chapter 5.1.1.
- [2] Peter Brucker: Scheduling Algorithms chapter 10.2
- [3] Vízvári Béla: Bevezetés a termelésirányítás matematikai módszereibe 4.4.fejezet.