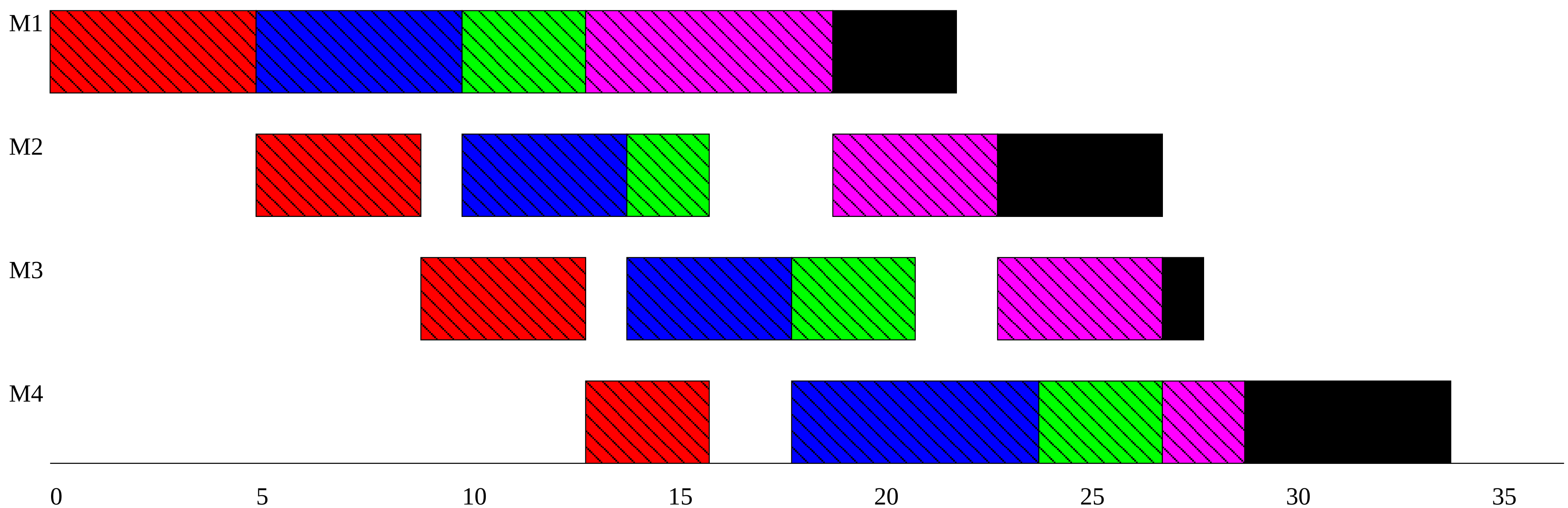
Permuatation scheduling for the flow shop problem.

4 machines, 5 jobs

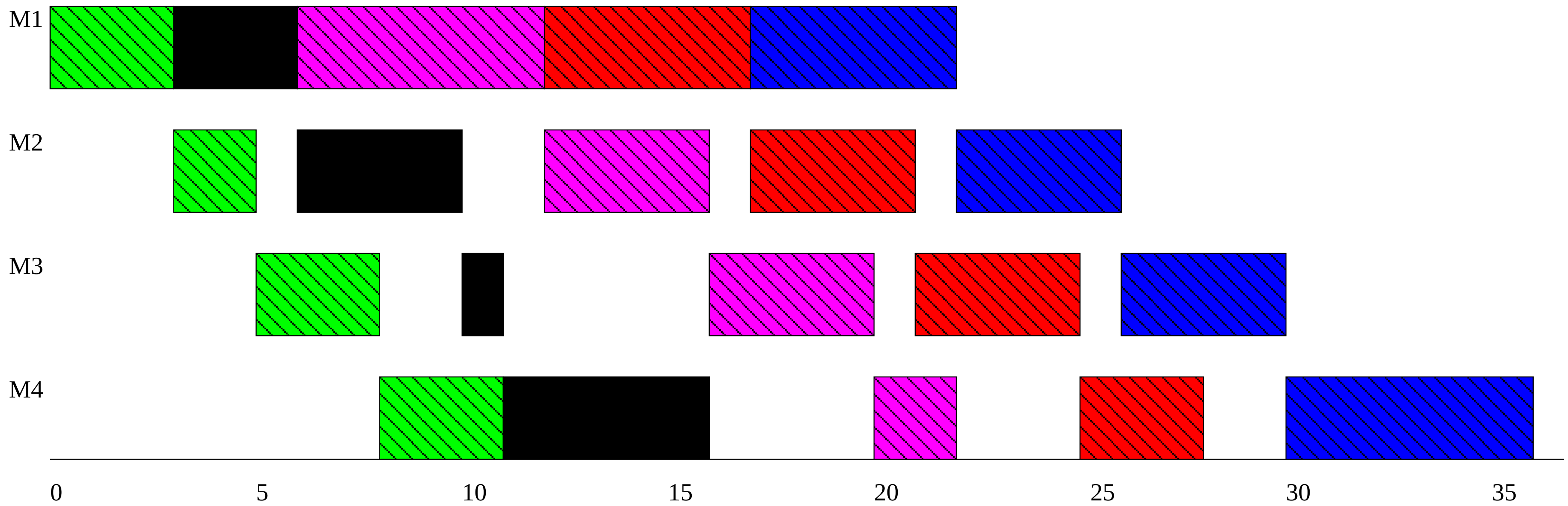
jobs	J_1	J_2	J_3	J_4	J_5
p_{1,j_i}	5	5	3	6	3
p_{2,j_i}	4	4	2	4	4
p_{3,j_i}	4	4	3	4	1
p_{4,j_i}	3	6	3	2	5

A permutation schedule in which the jobs run in the order 1,2,3,4,5



$$C_{\max} = 34$$

A permutation schedule in which the jobs run in the order 3,5,4,1,2



$$C_{\max} = 36$$

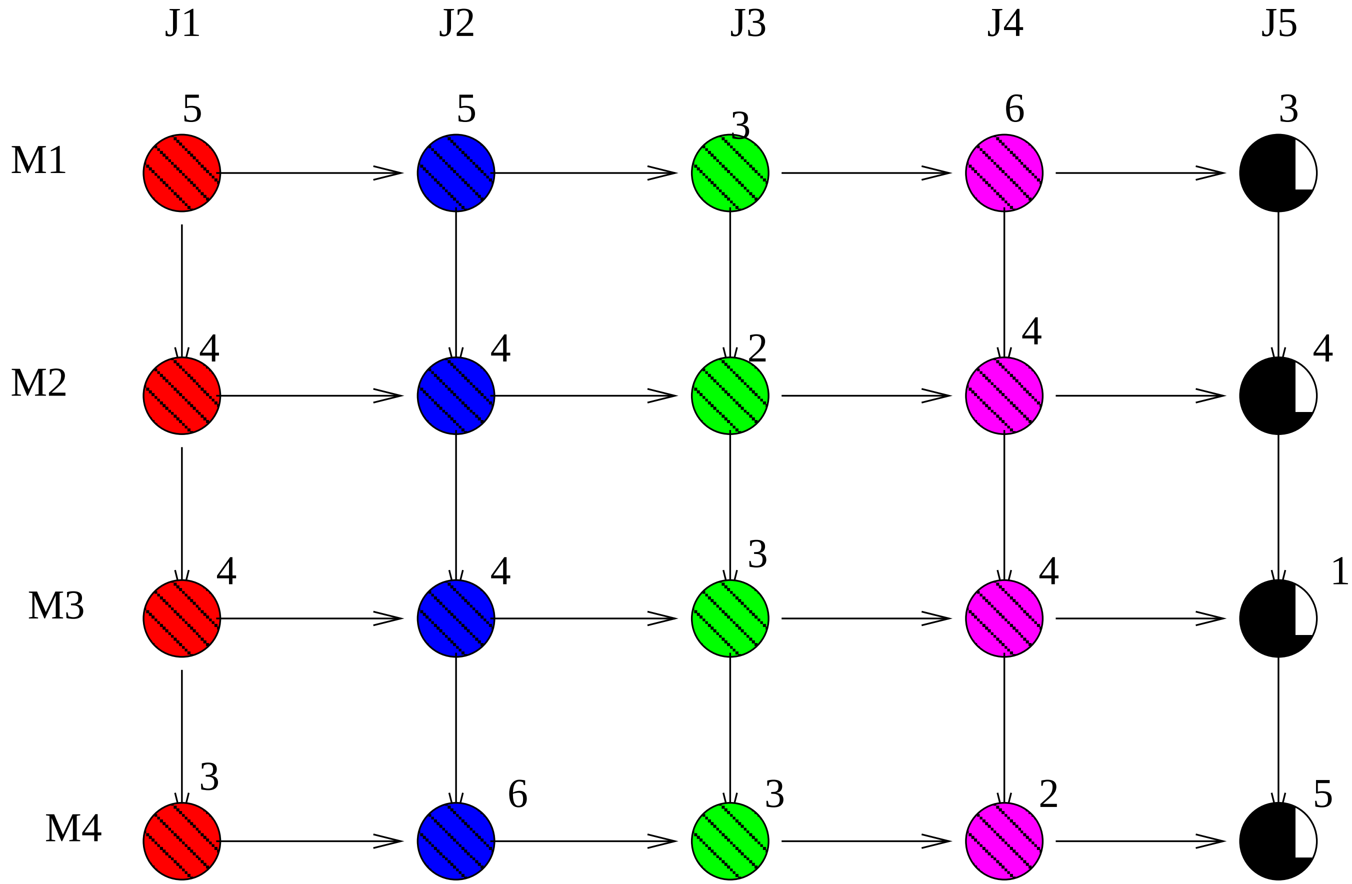
Computing the completion times in a permutation schedule

Let C_{i,j_k} be the completion time of the ith operation of job j_k. Assume that the jobs run in the order $j_1, \ldots, j_n$, and go through the machines in order $M_1, \ldots, M_m$.

$$C_{i,j_k} = \begin{cases} \Sigma_{\ell=1}^{i} p_{\ell,j_1} & \text{if } j_k = j_1 \\ \Sigma_{\ell=1}^{k} p_{1,j_\ell} & \text{if } i = 1 \\ \max(C_{i-1,j_k}, C_{i,j_{k-1}}) + p_{i,j_k} & \text{otherwise} \end{cases}$$

A disjunctive graph

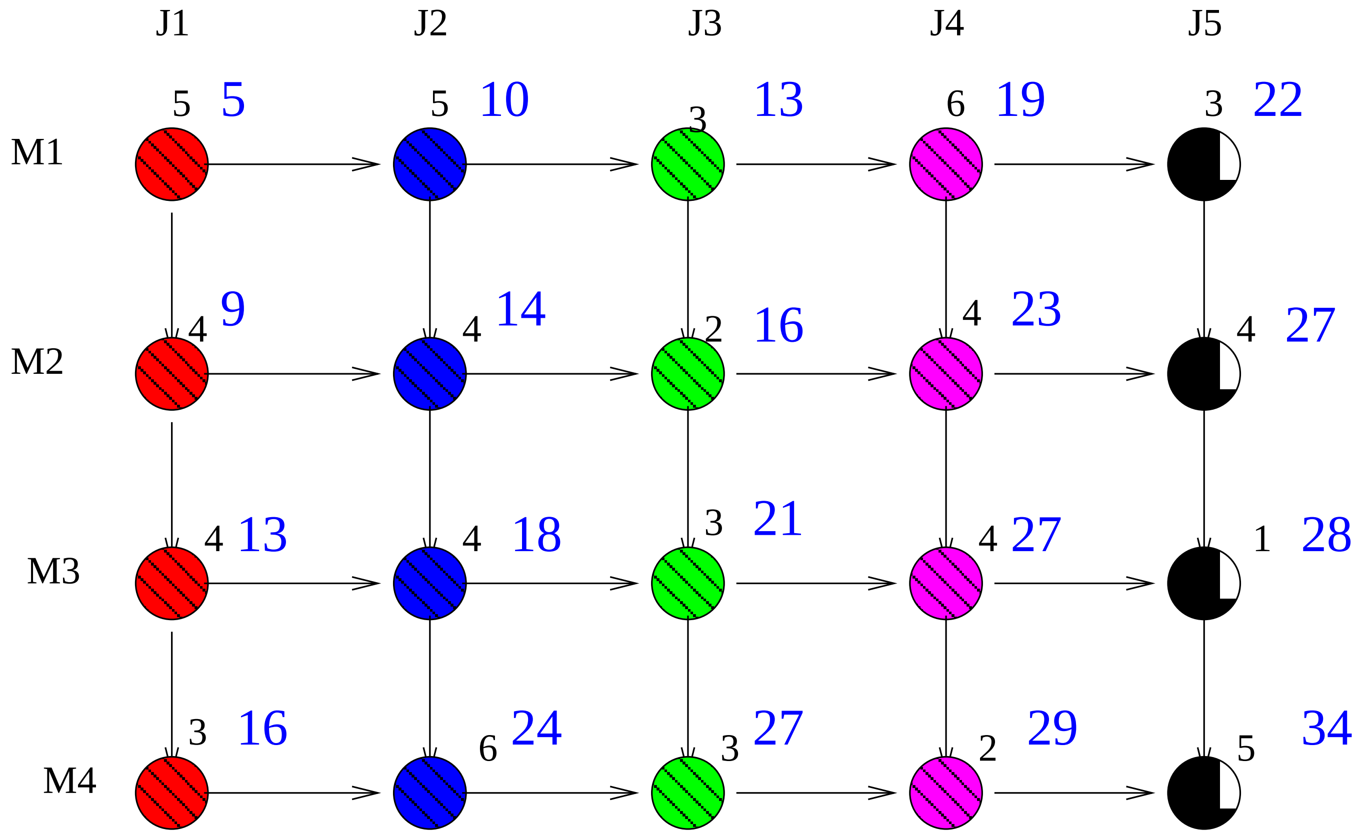
This is the graph associated with the permutation 1,2,3,4,5. The critical path in this graph is the makespan associated with scheduling the jobs in the given order. The critical path to any node is the completion time of that operation.



Computing the critical path

$$C_{i,j_k} = \begin{cases} \Sigma_{\ell=1}^{i} p_{\ell,j_1} & \text{if } j_k = j_1 \\ \Sigma_{\ell=1}^{k} p_{1,j_\ell} & \text{if } i = 1 \\ \max(C_{i-1,j_k}, C_{i,j_{k-1}}) + p_{i,j_k} & \text{otherwise} \end{cases}$$

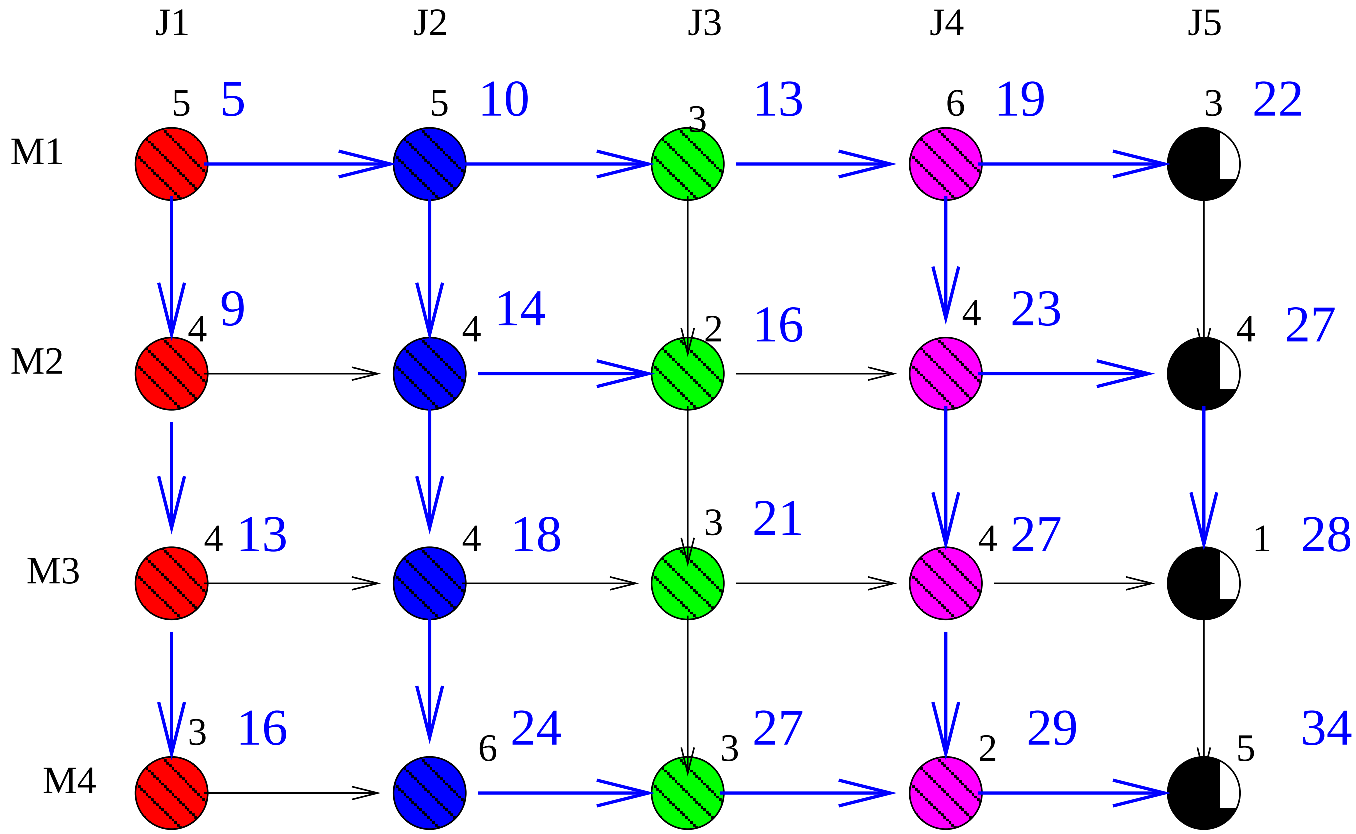
In the graph, each node's completion time is its processing time, plus the max completion time of its predecesors (immediately above and immediately to the left).



Here are the critical edges. The blue numbers are the C_{i,j_k} values. An edge between (i, j_k) and $(i', j_{k'})$ is critical if it was used to compute $C_{i',j_{k'}}$, i.e.

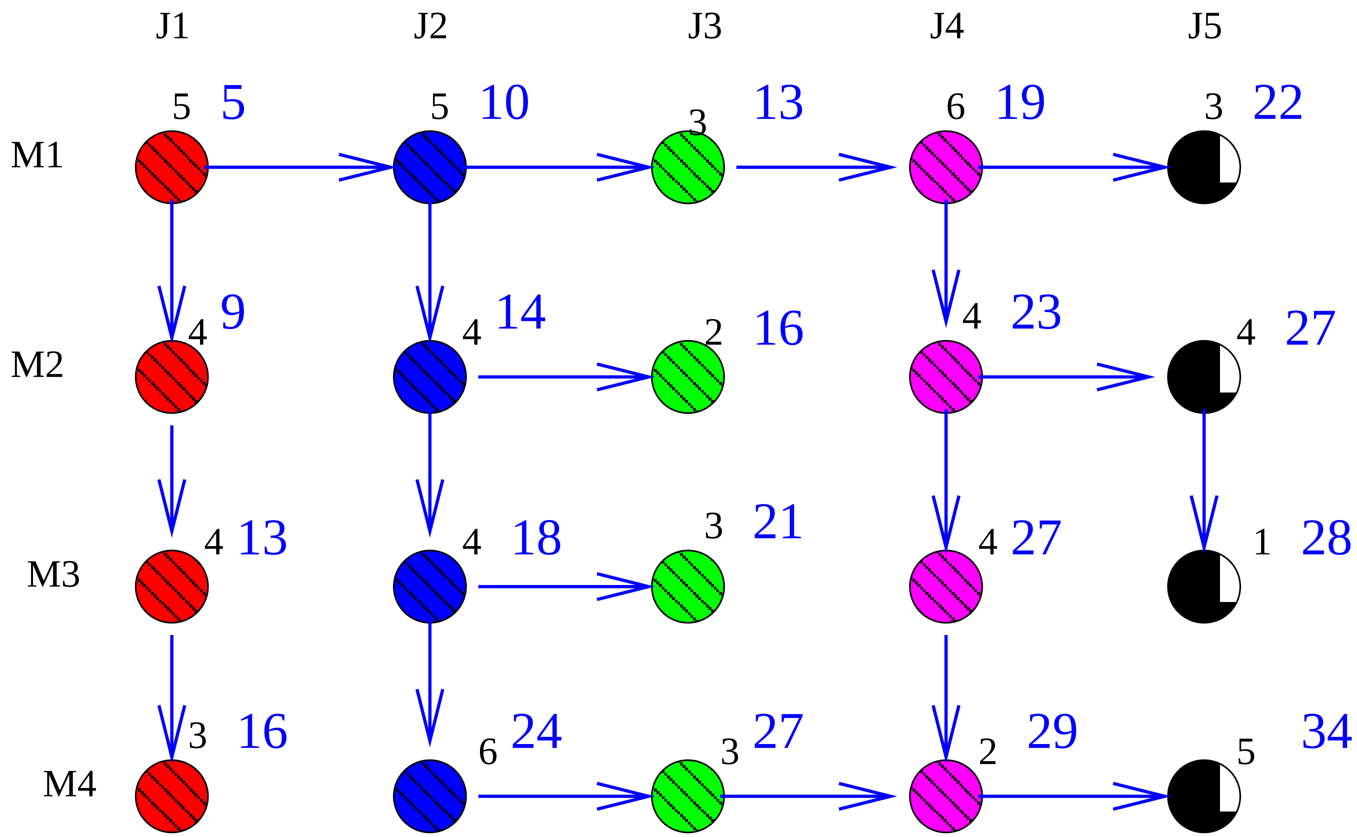
$$C_{i',j_{k'}} = C_{i,j_k} + p_{i',j_{k'}}$$

.



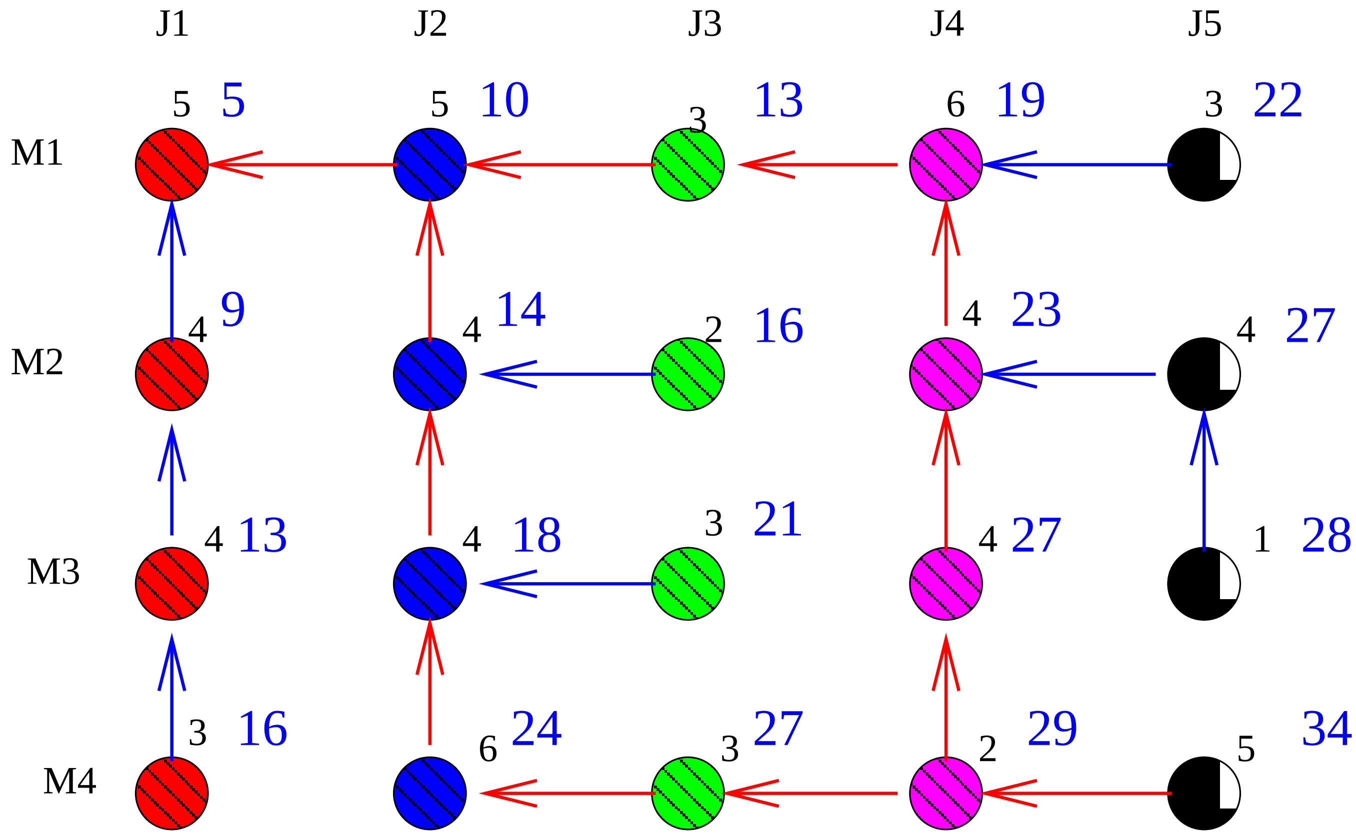
completion times

We can now consider the graph of only critical edges.



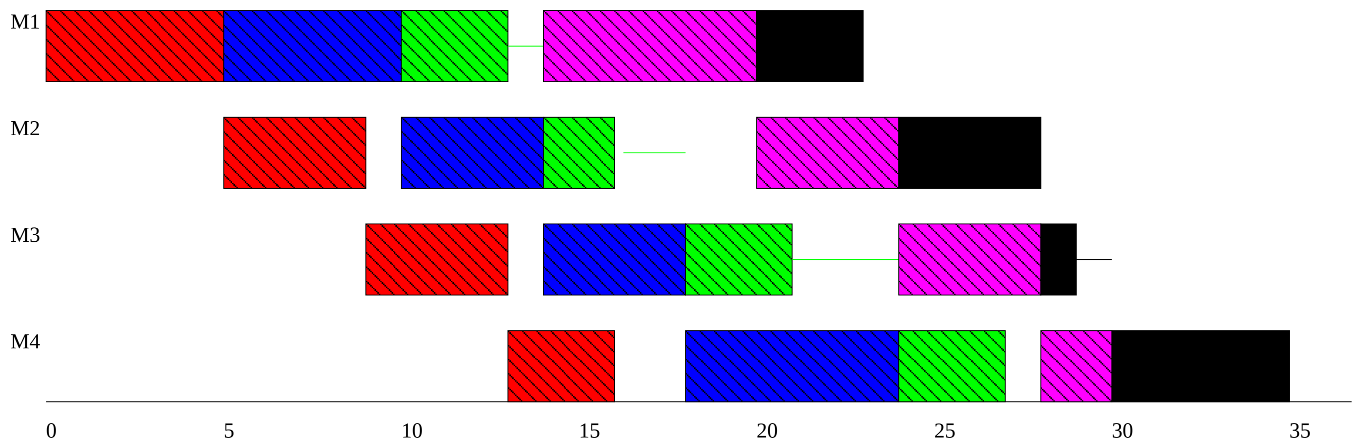
In this graph, we'd like to find the edges that are actually on a path from the source to the sink. To do so, we can reverse all the edges in the graph and do a breadth first search. Every edge traversed in this breadth first search is on a true critical path.

In this graph, we'd like to find the edges that are actually on a path from the source to the sink. To do so, we can reverse all the edges in the graph and do a breadth first search. Every edge traversed in this breadth first search is on a true critical path. These are colored red.



Blocking flow shop.

No intermediate storage. A job remains on its machine until the next machine is available. For our example, and the permutation $1, 2, 3, 4, 5$, we have the following schedule.



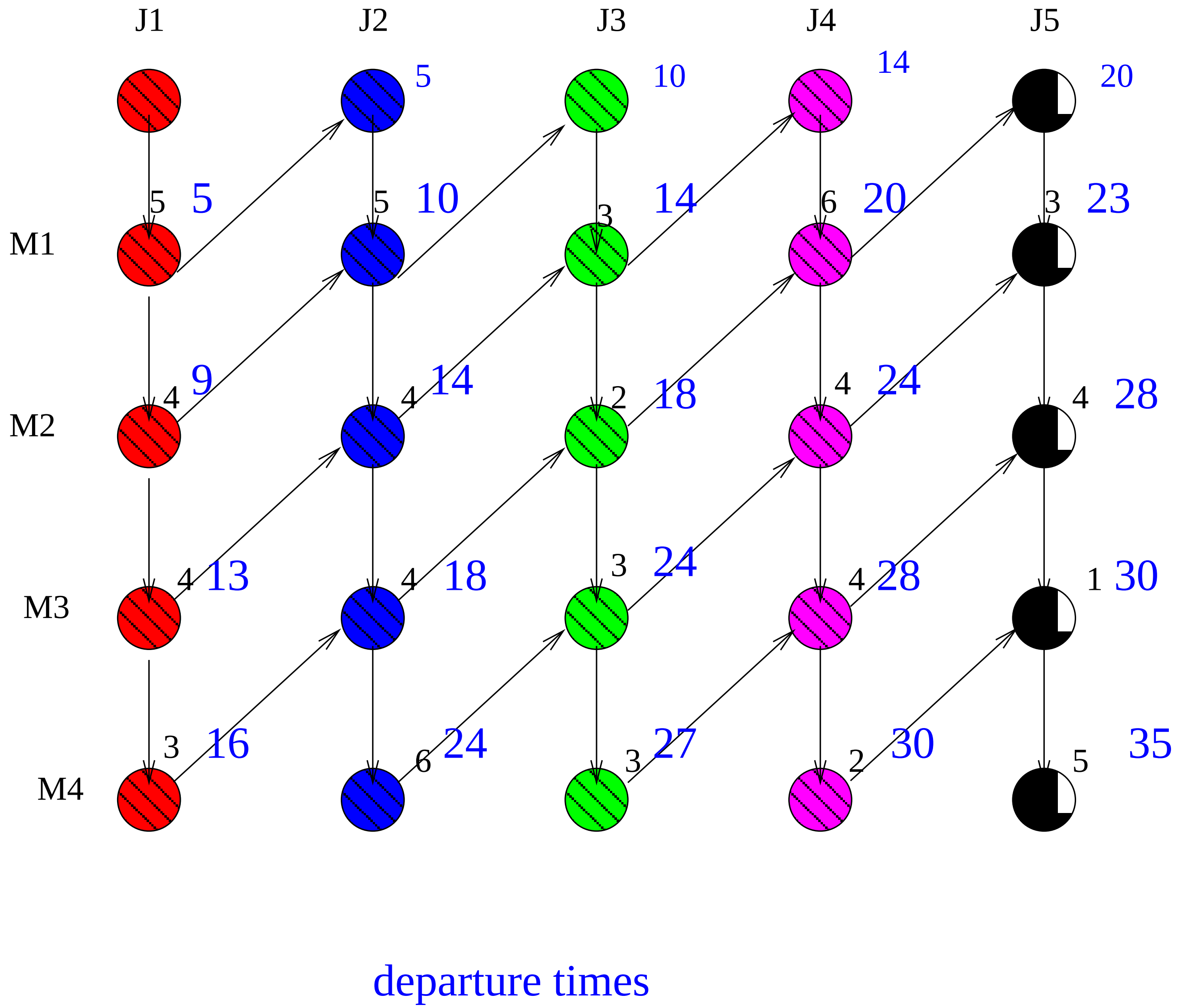
Makespan is 35. Lines represent blocking times.

Computing the completion times in a blocking permutation schedule

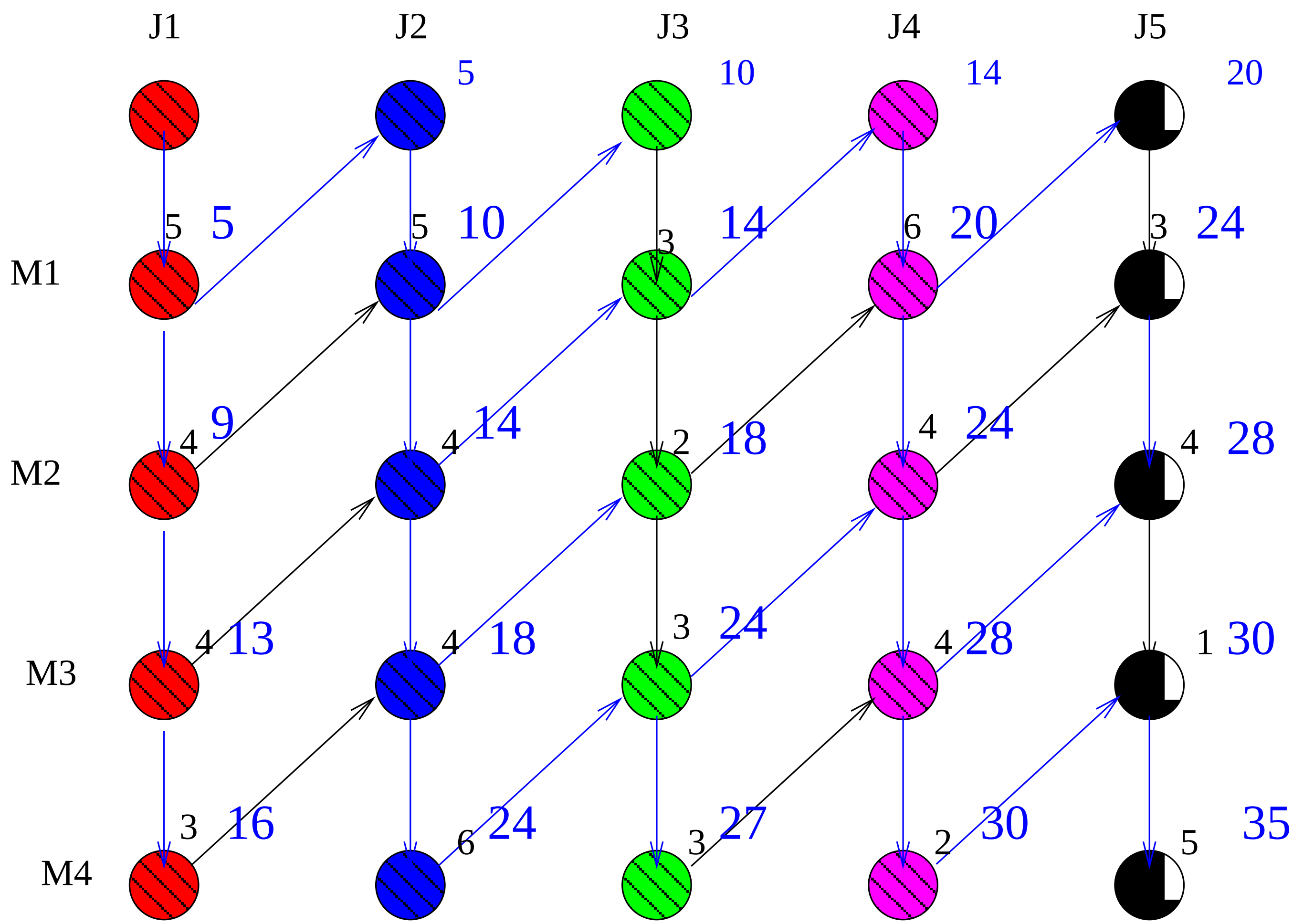
Let D_{i,j_k} be the time that job j_k departs machine i. (This is equal to the time that job j_k starts on machine $i+1$. Assume that the jobs run in the order $j_1, \ldots, j_n$, and go through the machines in order $M_1, \ldots, M_m$.

$$D_{i,j_k} = \begin{cases} \Sigma_{\ell=1}^{i} p_{\ell,j_1} & \text{if } j_k = j_1 \\ D_{m-1,j_k} + p_{m,j_k} & \text{if } i = m \\ \max(D_{i-1,j_k} + p_{i,j_k}, D_{i+1,j_{k-1}}) & \text{otherwise} \end{cases}$$

In the graph, each node's completion time can be computed by the formula above. The path computation is a non-standard. For the down edges, you add the processing time, but for the diagonal edges, you do not.



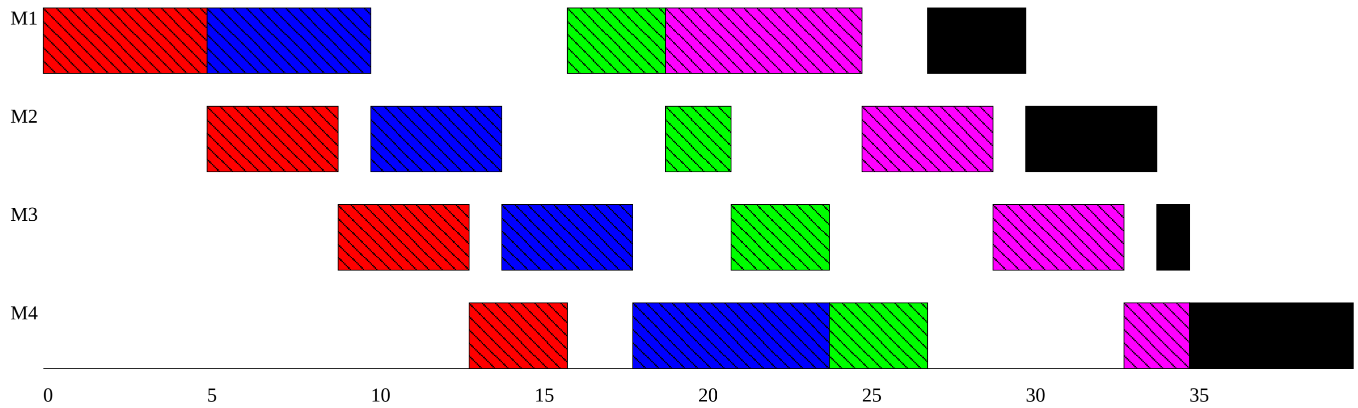
A critical path can be computed. The critical edges are shaded in blue. The are the edges from which the D values are actually computed. The computation of the critical path can now procede as before.



departure times

No wait flow shop (Hot potato).

Once a job starts processing, it must not incur any idle time.
For our example, and the permutation $1, 2, 3, 4, 5$, we have the following schedule.



Makespan is 40.