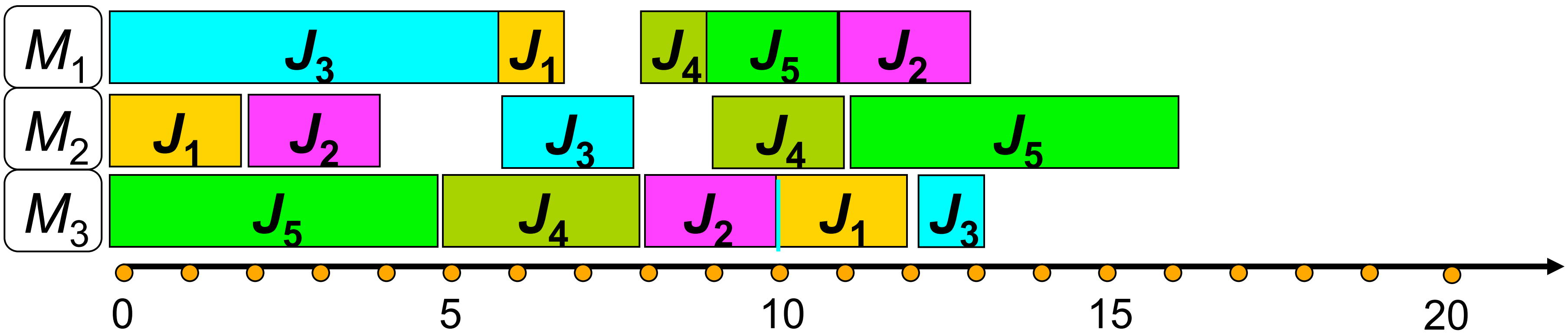
Approximation Schemes

Open Shop Problem

$O||C_{\max}$ and $Om||C_{\max}$

- $\{J_1,\ldots, J_n\}$ is set of jobs.
- $\{M_1,\ldots, M_m\}$ is set of machines.
- $J_i : \{O_{i1},\ldots, O_{im}\}$ is set of operations of job J_i ,
- O_{ik} must be processed on machine M_k,
- $O_{ik} : p_{ik} \geq 0$ $(i=1,\ldots, n; k=1,\ldots, m)$,

Instance and Schedule: $C_{\max} = 16$



J_1	$p_{11} = 1$	$p_{12} = 2$	$p_{13} = 2$
J_2	$p_{21} = 2$	$p_{22} = 2$	$p_{23} = 2$
J_3	$p_{31} = 6$	$p_{32} = 2$	$p_{33} = 1$
J_4	$p_{41} = 1$	$p_{42} = 2$	$p_{43} = 3$
J_5	$p_{51} = 2$	$p_{52} = 5$	$p_{53} = 5$

Lower bounds

- Let P_j be the length of job J_j, $P_j = \sum_{i=1}^{m} p_{ij}$.

- $P_{\max} = \max_j P_j$ is the maximum job length.
- Let L_k be the load of machine M_k.
- $L_{\max} = \max_i L_i$ is the maximum machine load.

$$\text{OPT} \geq \max\{P_{\max}, L_{\max}\}$$

Greedy Algorithm for $O||C_{\max}$

- Whenever a machine becomes idle, if there is an unscheduled operation available to be scheduled on that machine, schedule it.
- An operation is available if it belongs to a job which is currently not undergoing any processing.

2-approximation algorithm

Theorem 8.1 (Racsmany [1982])

Greedy algorithm is a 2-approximation algorithm for $O||C_{\max}$.

Proof

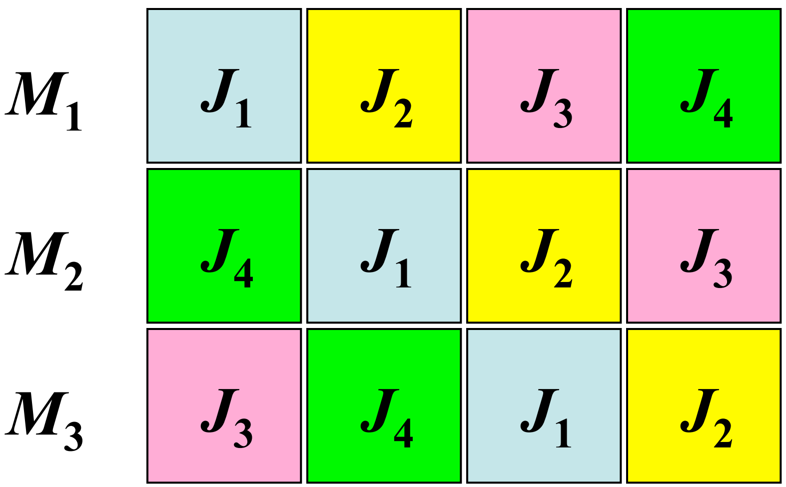
- Consider the machine M_i that finishes last, and let J_j be the job whose operation O_k completes last on that machine.
- We claim that, at every point in time during the schedule, either job J_j is undergoing processing on some machine, or machine M_i is busy or both.
- If such were not the case, operation O_k would have been scheduled earlier, since if M_i and J_j were both available earlier, clearly O_k would have been processed at that time.
- Thus the schedule's makespan is less than

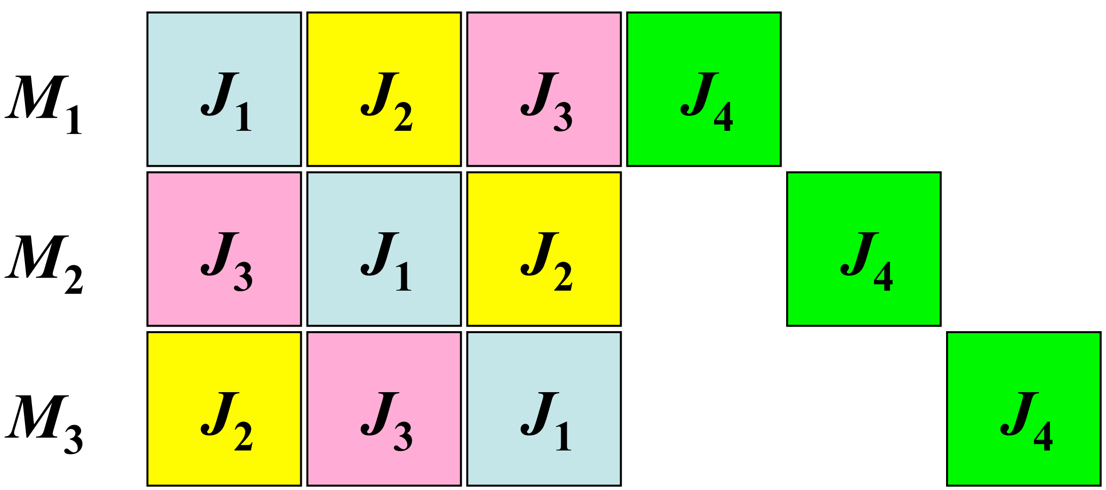
$$P_{\max} + L_{\max} \leq 2\mathrm{OPT}\,.$$

Tight instance

- A simple example shows that the greedy algorithm does not do better than $2 - 1/m$ in the worst case.
- Given m machines and $m + 1$ jobs, each having a unit-length operation on each machine.
- It is not hard to see that the optimal schedule has length $m + 1$, since the jobs can simply be rotated through the machines.
- Suppose that the greedy algorithm assign m jobs in the interval $(0, m]$. Then the operation J_{m+1} must undergo processing sequentially, and the overall schedule produced has length $2m$,

Tight instance





Inapproximability of $O||C_{\max}$

- **Theorem 8.2 (Williamson et al.)**

  The problem of deciding if there is an open shop schedule of length at most 4 is *NP*-complete.

- **Corollary 8.3 (Williamson et al.)**

  For any $\rho < 5/4$, there does not exist a polynomial-time ρ -approximation algorithm for the open shop problem, unless $P = NP$.

Monotonne-Not-All-Equal-3Sat

- **Instance:** Set *U* of variables, collection *Z* of clauses over *U* such that each clause has size 3 and contains only unnegated variables.
- **Question:** Is the truth assignment for *U* such that each clause in *C* has at least one true variable and at least one false variable?

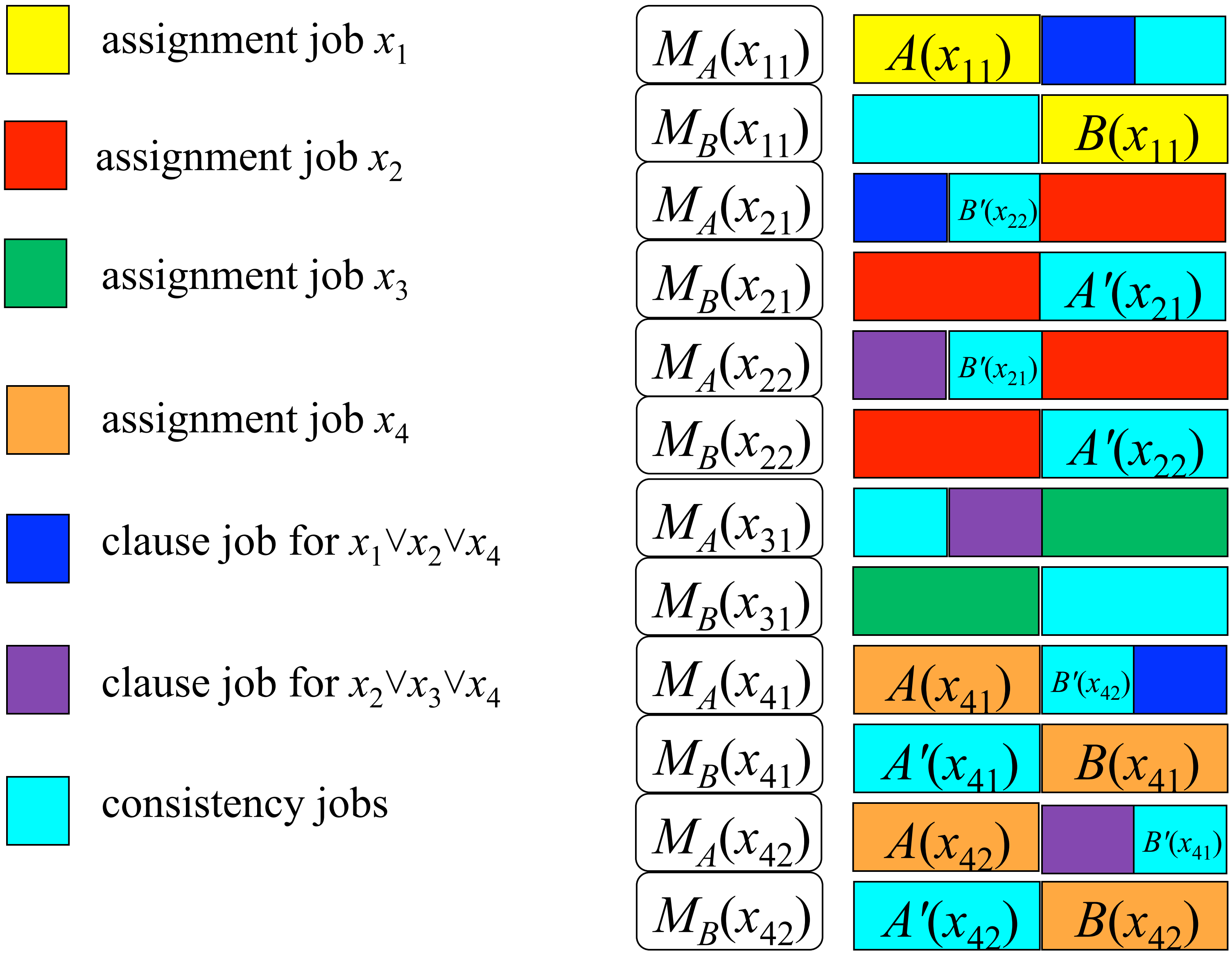
Instance I of SAT

- $U = \{x_1, \ldots, x_u\}$, $Z = \{z_1, \ldots, z_v\}$
- Suppose each variable x_i appears t_i times in Z.
- For notational convenience, we view the k-th occurrence of x_i as the variable x_{ik}.
- Let $\sigma(x_{ik})$ denote the next occurrence of x_i , cyclically ordered; that is $\sigma(x_{ik}) = x_{il}$, where $l = k \bmod t_i + 1$.
- transform an instance I into an inWe stance I_O of the $O||C_{\max}$ problem.

Instance I_O

- $2u$ machines and $2u + v$ jobs
- For each variable x_{ik} we construct two machines $M_A(x_{ik})$ and $M_B(x_{ik})$ and 3 types of jobs:
- An *assignment* job J_{ik} with operations $A(x_{ik})$ and $B(x_{ik})$ of length 2, which are to be processed by $M_A(x_{ik})$ and $M_B(x_{ik})$, respectively.
- A *consistency job* J'_{ik} has operations $A'(x_{ik})$ and $B'(x_{ik})$ of length 2 and 1, respectively, which must be processed by $M_B(x_{ik})$ and $M_A(\sigma(x_{ik}))$.
- For each clause $c = (x \vee y \vee z)$, we construct a clause job J_c with three unit-length operations $T(x)$, $T(y)$, and $T(z)$, to be processed on $M_A(x)$, $M_A(y)$, and $M_A(z)$, respectively.

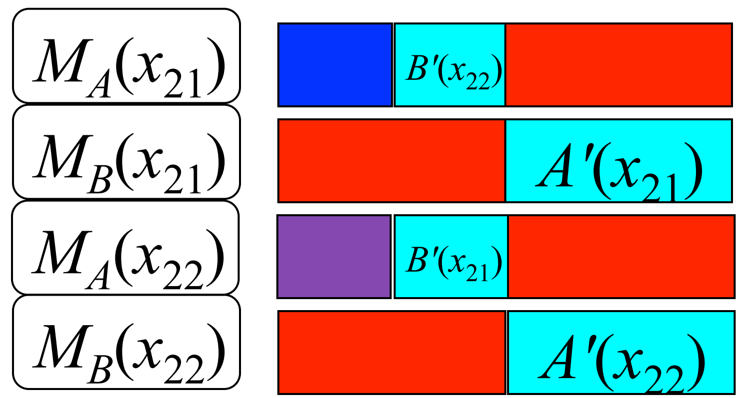
$$(x_1 \vee x_2 \vee x_4) \wedge (x_2 \vee x_3 \vee x_4)$$



Consistency jobs

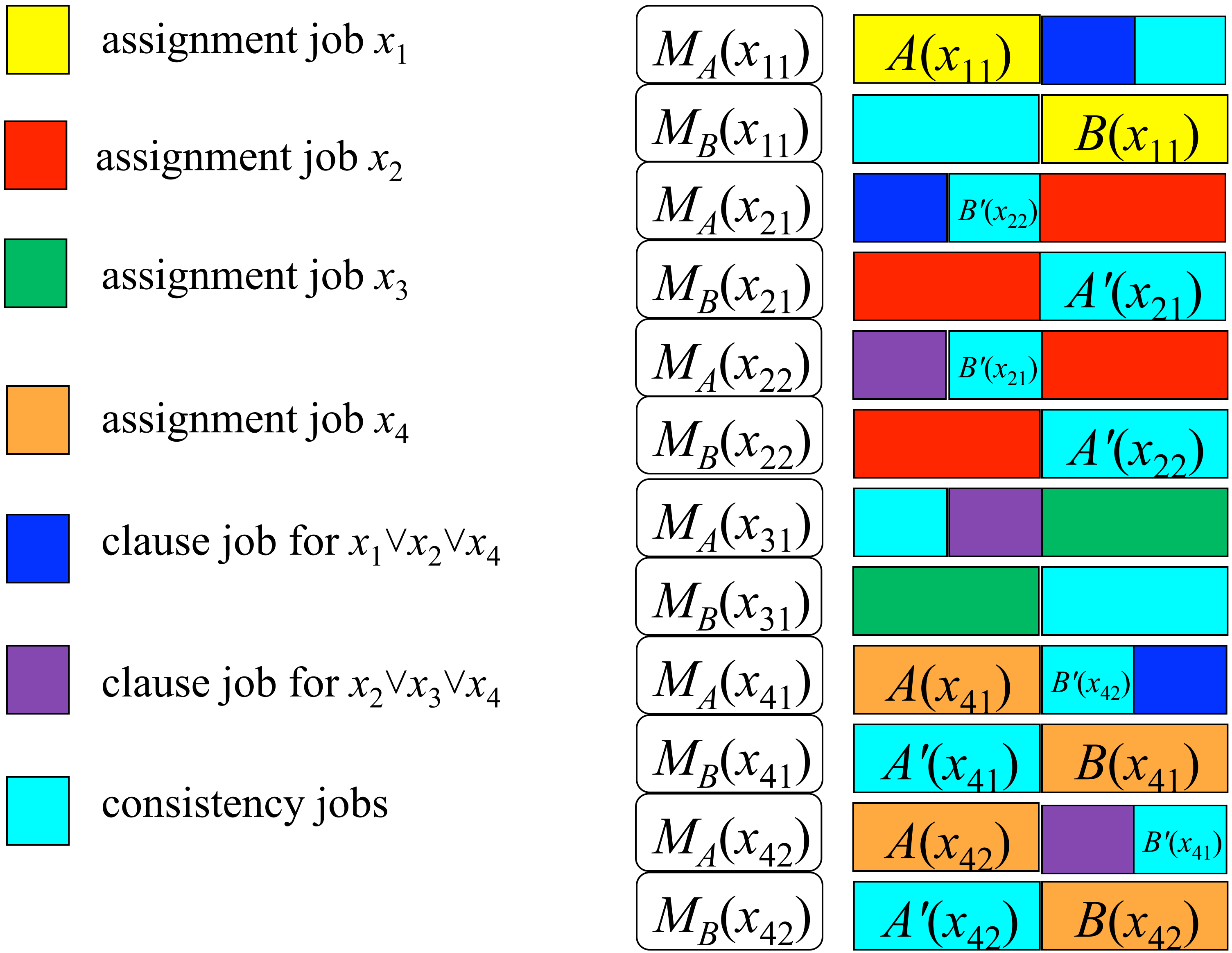
assignment job x_2

consistency jobs



A consistency job ensures that the value of variable x_{ik} is equal to the value of its next occurrence.

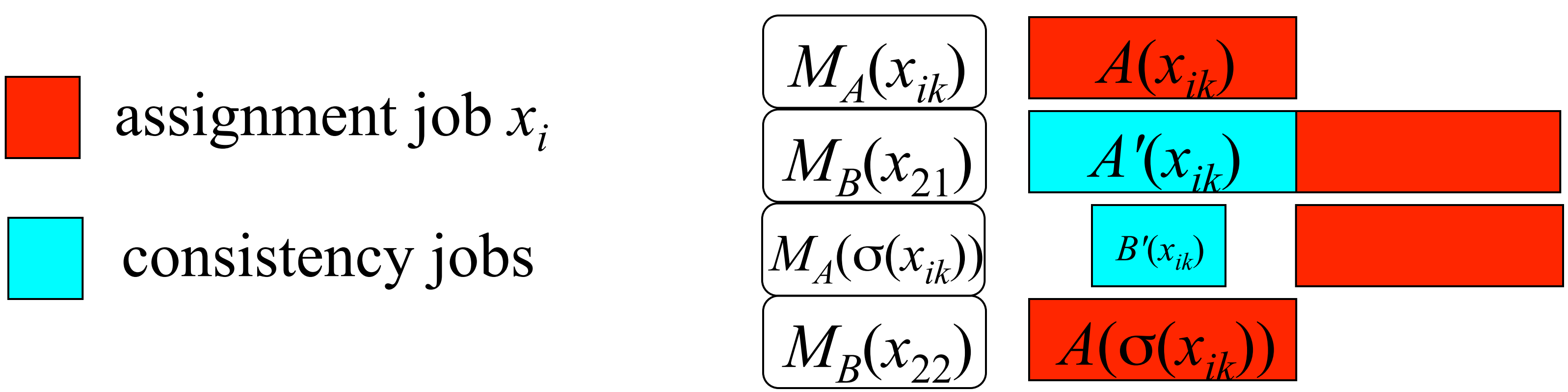
$$(x_1 \vee x_2 \vee x_4) \wedge (x_2 \vee x_3 \vee x_4)$$



Properties of instance I_O

Suppose that there is a schedule of length 4.

- In any such schedule, either every machine $M_A(x_{ik})$ ($k = 1,\ldots, t_i$) process its assignment operation from time 0 to 2, or every machine $M_A(x_{ik})$ process its assignment operation from time 2 to 4.
- If this is not a case, then the exists i and k such that $M_A(x_{ik})$ processes its assignment operation from time 0 to 2, and $M_A(\sigma(x_{ik}))$ processes its assignment operation from time 2 to 4. But $M_B(x_{ik})$ processes its assignment operation from time 2 to 4. The consistency job for x_{ik} must be processed on both $M_B(x_{ik})$ and $M_A(\sigma(x_{ik}))$ from time 0 to 2, which is a contradiction.

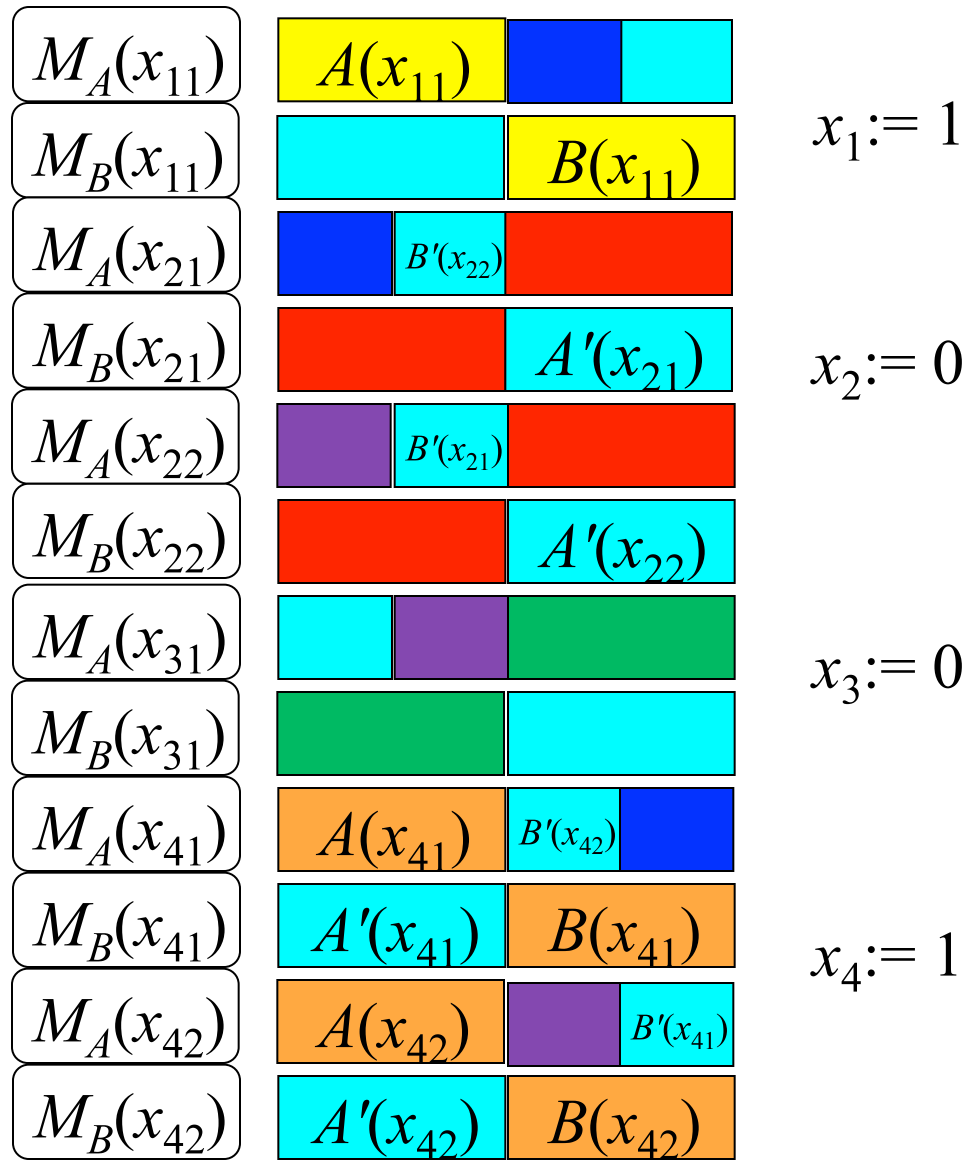


Satisfying assignment

- For each variable x_i, set x_i to be ***true*** if the assignment operation for $M_A(x_{ik})$ runs from 0 to 2, and ***false*** otherwise.
- We know that a clause operation has been scheduled between time 2 and 4 in case the variable corresponding to that operation has been set true and sometime between time 0 and 2 in case the variable has been set false.
- Because each clause job has three unit length operations wich have been scheduled in nonoverlapping time periods, not all of its operations can correspond to ***true*** variables and not all of its operations can correspond to ***false*** variables. Hence at least one variable of each clause must be ***true*** and at least one variable must be ***false***.

$$(x_1 \vee x_2 \vee x_4) \wedge (x_2 \vee x_3 \vee x_4)$$

- assignment job x_1
- assignment job x_2
- assignment job x_3
- assignment job x_4
- clause job for $x_1 \vee x_2 \vee x_4$
- clause job for $x_2 \vee x_3 \vee x_4$
- consistency jobs



$Om||C_{\max}$

Let $0 < \varepsilon < 1$ be some small number such that $1/\varepsilon$ is an integer.

Let $m \geq 3$ be an integer

$LB = \max\{P_{\max},\ L_{\max}\}$

$LB \leq OPT(C_{\max}) \leq 2LB$

Partition of jobs
$(0 < \varepsilon' < \varepsilon/(m^2+1) < 1)$

We define three set of jobs.

For a rational number α with $\varepsilon^{m/\varepsilon} \leq \alpha \leq \varepsilon$ set

Big $= \{J_j \in J \mid P_j \geq \alpha LB\}$,

Small $= \{J_j \in J \mid \alpha\varepsilon' LB < P_j < \alpha LB\}$,

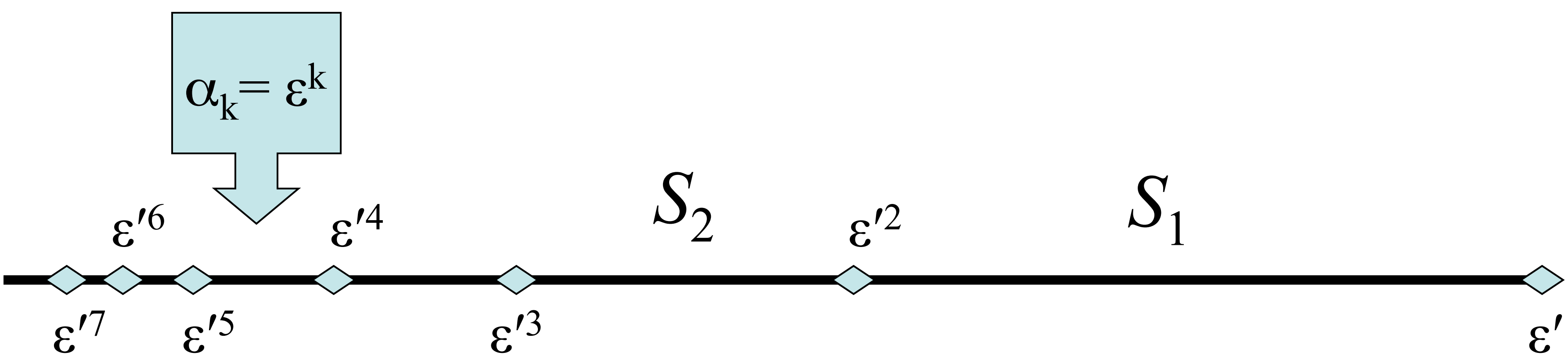
Tiny $= \{J_j \in J \mid P_j \leq \alpha\varepsilon' LB\}$.

- The number of big jobs is bounded by $m/\alpha \leq m\varepsilon'^{-m/\varepsilon'}$.
- The total length of the small jobs is at most $\varepsilon' LB$.

How to choose such α

$$\textbf{Small} = \{J_j \in J \mid \alpha\varepsilon' LB < P_j < \alpha LB\}$$

- Define a sequence of real numbers $\alpha_l = (\varepsilon')^l$, $l \geq 0$.
- Consider the sets S_l of small operations with respect to α_l.
- For $i \neq j$ the sets S_i and are S_j disjoint

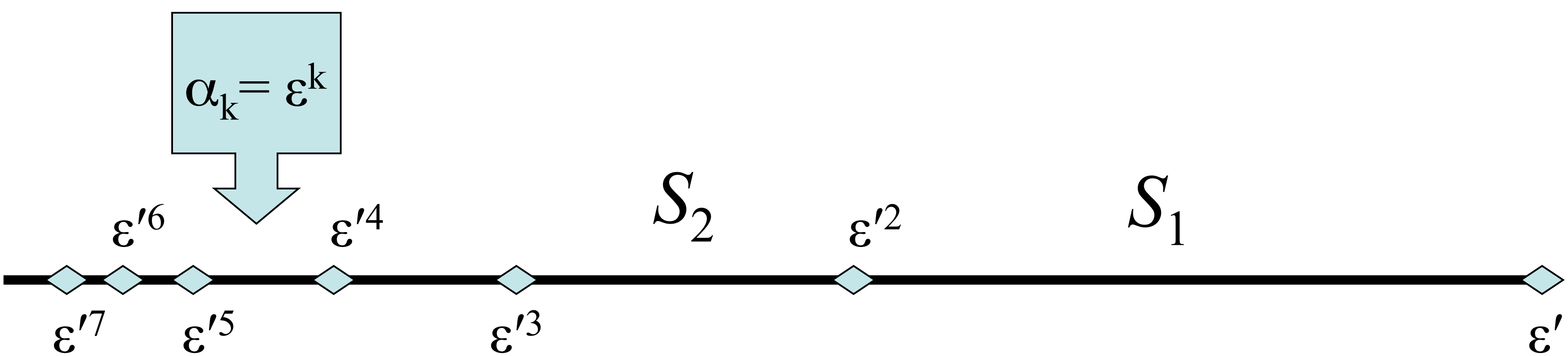


How to choose such α

- Since the total length of all operations is at most *mLB,* the exists a number $k \le m/\varepsilon'$ for which S_k is as desired.

$$\sum_{j \in S_k} P_j \le \varepsilon' LB.$$

- We set $\alpha = \alpha_k$. Note that the value of α depends on the input, but it is bounded by constants independent on the input.

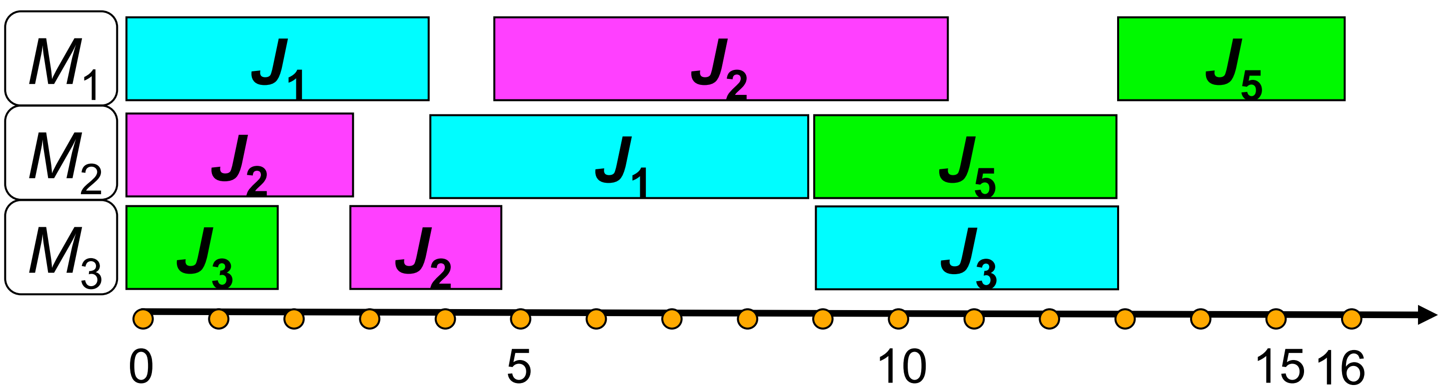


Algorithm OpenShop

1. Find an optimal schedule σ_1 for big jobs.
   – We fix an order of the big jobs for each machine and fix an order of the operations for each big job. On any machine M_i, the schedule big jobs induces a sequence of gaps.
2. Schedule tiny operations into gaps of σ_1. Denote the obtained schedule as σ_2.
3. Add the small jobs at the end of σ_2 in a greedy way.

Step 1

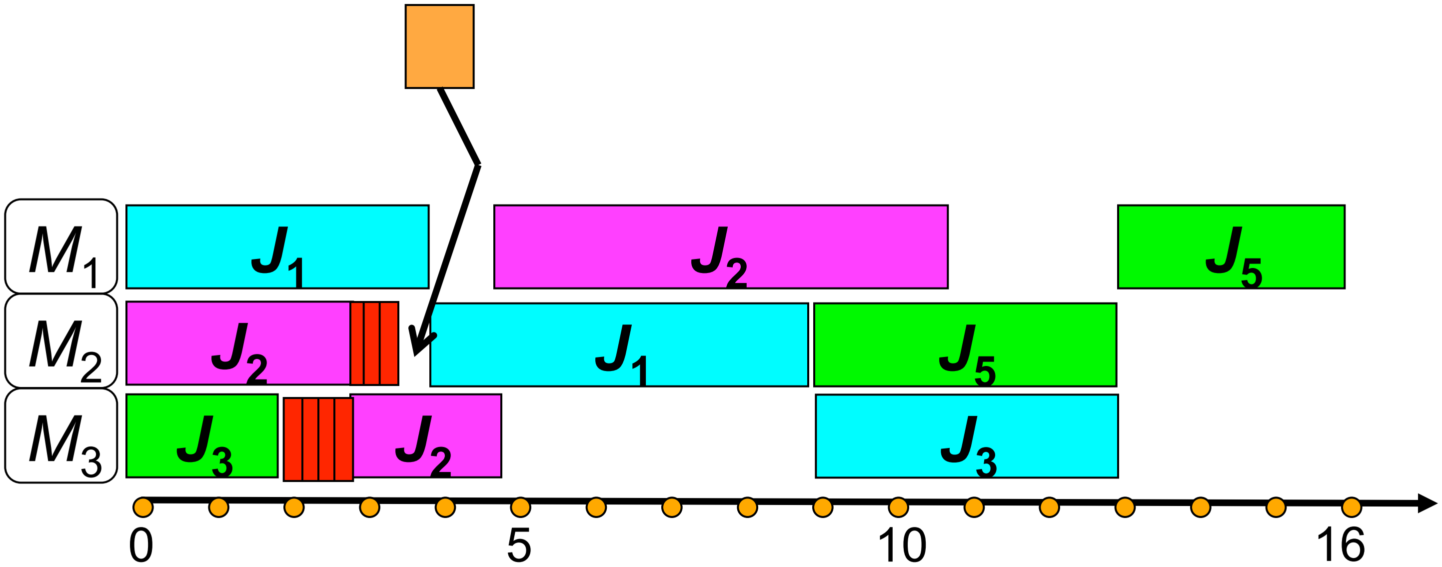
- The number of big jobs is bounded by $\frac{m}{\alpha} \leq \frac{m}{\varepsilon'^{\left(m/\varepsilon'\right)}}$.
- The number of big jobs is bounded by a constant that only depends on ε and m.
- We enumerate all schedules of big jobs and take the best one. We note that OPT(Big) ≤ OPT.
- There are at most m^2/α gaps in the schedule σ_1.



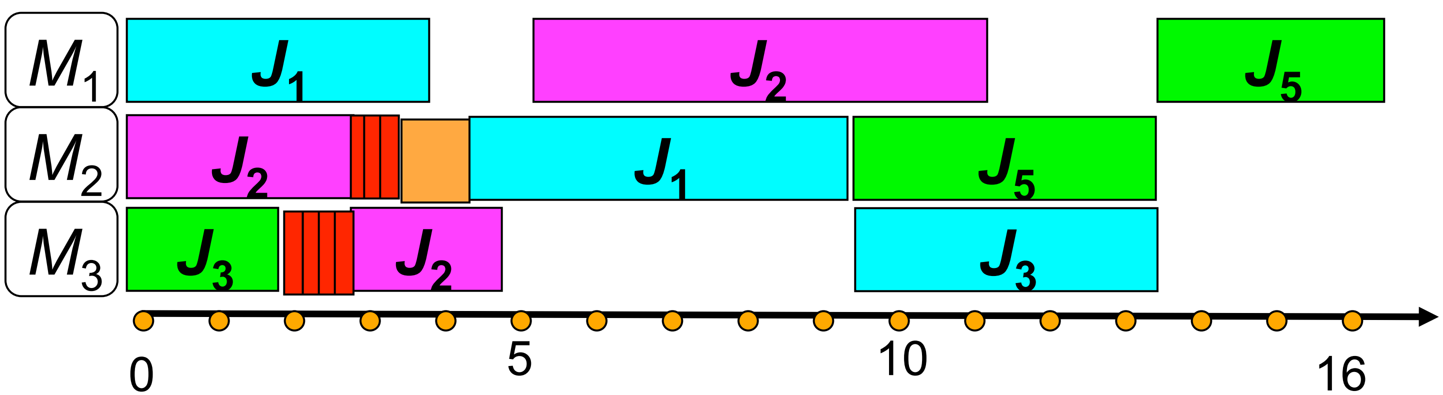
Steps 2 and 3

- Starting at time $t = 0$, the algorithm tries to schedule one of the available unscheduled tiny operations at every time t where one of the machines M_i becomes idle.
- Let at time t an available operation O_{ik} be considered and the remaining part of the gap is less than the length of operation O_{ik}. The reason is that some big operation O' starts at time τ and $p_{ik} > \tau - t$. In this case shift operation O' and every operation, which starts after the completion of O', to the right by $p_{ik} - \tau + t$ time units.
- Let $C_{\max}(\sigma_2)$ be the length of the obtained schedule.
- Starting at time $C_{\max}(\sigma_2)$ schedule the small jobs in a greedy way.

Step 2



M1
M2
M3
J1
J2
J5
J2
J1
J5
J3
J2
J3
0
5
10
16



M1
M2
M3
J1
J2
J5
J2
J1
J5
J3
J2
J3
0
5
10
16

Analysis of the algorithm

- Let $C_{\max}(\sigma)$ be the makespan of the schedule by Algorithm OpenShop. Since the total length of small jobs is at most $\varepsilon' LB$, it follows that $C_{\max}(\sigma) \leq C_{\max}(\sigma_2) + \varepsilon' LB$.
- Let us estimate $C_{\max}(\sigma_2)$.
- Let O_{jk} be an operation, which completes last in σ_2, i.e. $C_{jk} = C_{\max}(\sigma_2)$.
- Let λ be the sum of the lengths of all shifts produced by the algorithm.
- Let μ be the total idle times on machine M_k.

O_{jk} is an operation of a big job

- $C_{\max}(\sigma_2) \le C_{\max}(\sigma_1) + \lambda$
- $\lambda \le (m^2/\alpha)\cdot\alpha\varepsilon' LB = m^2\varepsilon' LB$
- $C_{\max}(\sigma_2) \le C_{\max}(\sigma_1) + m^2\varepsilon' LB \le OPT + m^2\varepsilon' LB$

O_{jk} is an operation of a tiny job

- $C_{\max}(\sigma_2) \le L_k + \mu$
- $\mu \le p_k \le \alpha\varepsilon' LB$
- $C_{\max}(\sigma_2) \le OPT + \alpha\varepsilon' LB \le$
  $< OPT + m^2\varepsilon' LB$

$$C_{\max}(\sigma) \le C_{\max}(\sigma_2) + \varepsilon' LB \le$$
$$\le OPT + (m^2+1)\varepsilon' LB \le (1+\varepsilon)\, OPT.$$

PTAS

Theorem 8.4 (Sevastianov, Woeginger 1996)

For every fixed ε > 0 and any fixed m ≥ 2, there exists a polynomial-time (1+ ε)-approximation algorithm for the $Om||C_{\max}$ problem.

Exercise

- Let σ be the schedule obtained by the greedy algorithm. Let H be the set of intervals in σ such that no machine is idle during these intervals. Let W be the total length of the intervals from H. Suppose that $W \geq 3P_{\max}$.
- Obtain a good estimate of the ratio of $C_{\max}(\sigma)$ to OPT.