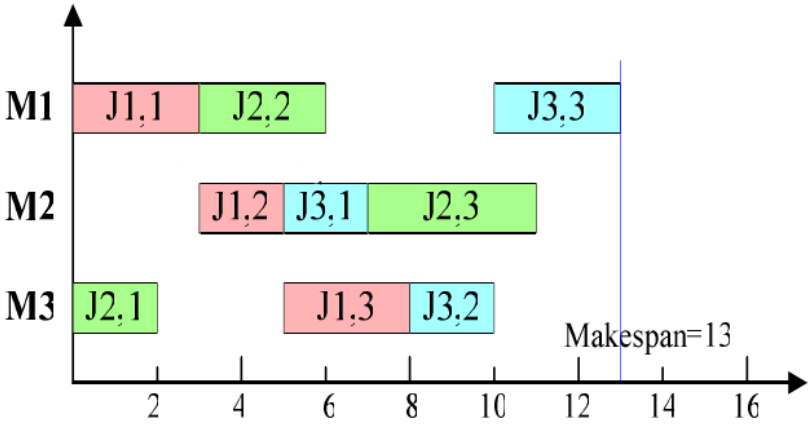
Job Shop Scheduling (JSP)
车间作业调度问题

吕志鹏

人工智能与优化研究所 所长
华中科技大学 计算机学院

问题描述

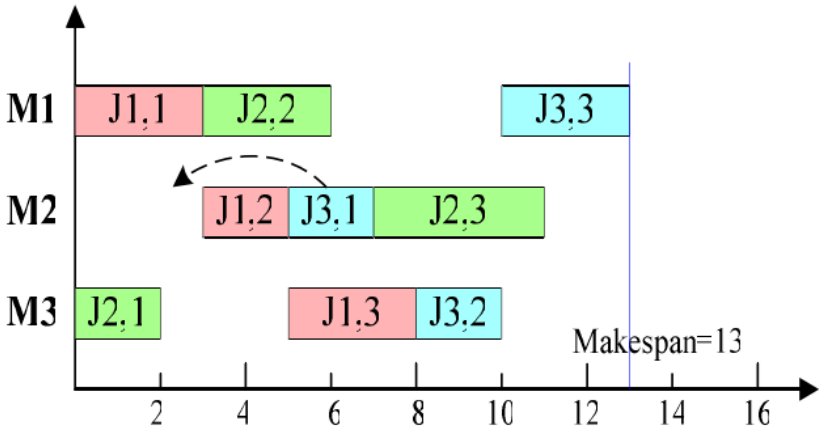
- 已知：有n 个工件{ J1，J2, …, Jn }在 m 台机器{ M1, M2, …, Mm}上加工，每个工件以一定的次序在所有的机器上轮流加工，每个工件分成 m 个工序，而每个工序对应相应的加工机器。其中，工序的加工时间给定。
- 工件上的约束：每个工件上的工序只能在上一个工序执行结束之后，才能开始执行下一个工序。
- 机器上的约束：每个机器某一个时刻最多只能执行一个工件，而且执行过程是非抢占的。
- 目标：给出调度方案，使调度总结束时间最小。（Makespan）

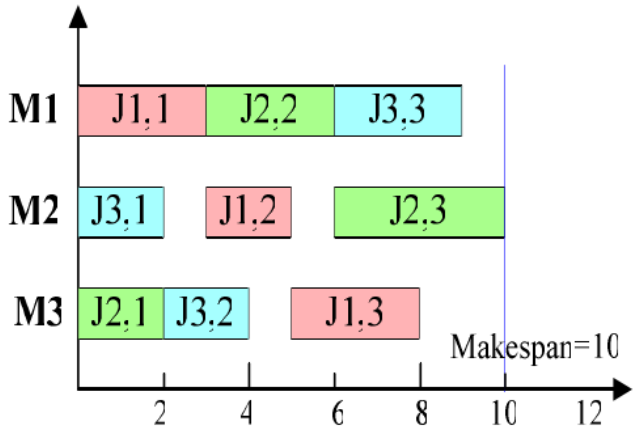


	工序1	工序2	工序3
J1 ：	M1	M2	M3
J2 ：	M3	M1	M2
J3 ：	M2	M3	M1

问题描述

- 已知：有n 个工件{ J1，J2, …, Jn }在 m 台机器{ M1, M2, …, Mm}上加工，每个工件以一定的次序在所有的机器上轮流加工，每个工件分成 m 个工序，而每个工序对应相应的加工机器。其中，工序的加工时间给定。
- 工件上的约束：每个工件上的工序只能在上一个工序执行结束之后，才能开始执行下一个工序。
- 机器上的约束：每个机器某一个时刻最多只能执行一个工件，而且执行过程是非抢占的。
- 目标：给出调度方案，使调度总结束时间最小。（Makespan）

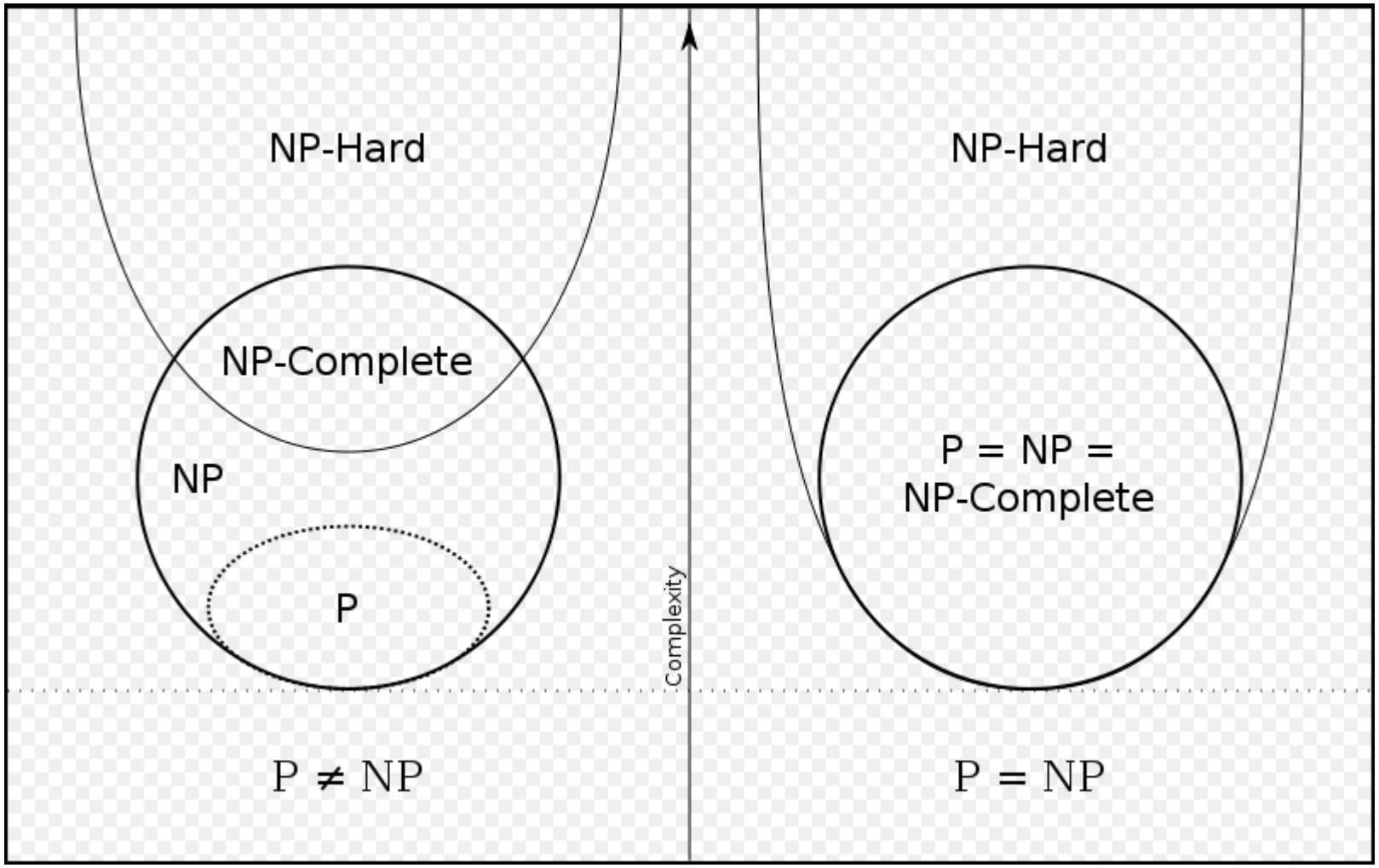




特点

- 典型性：JSP是最基本的调度问题，同时，和TSP问题一样，也是最经典的几个NP-hard 问题之一。
- 复杂性：Garey在1976年第一次给出了此问题的复杂度分析，而后在1979年，Lenstra证明JSP是NP-hard问题并具有挑战性的难度。Fisher和Thompson在1963年提出的FT10*10算例在20多年之后才被找到最优解。目前最先进的算法很难求得较小规模问题的最优解。
- 实用性：应用领域极其广泛，涉及航母调度，机场飞机调度，港口码头货船调度，汽车加工流水线等。

NP-Hard问题


NP-Hard
NP-Complete
NP
P
Complexity
P ≠ NP
NP-Hard
P = NP =
NP-Complete
P = NP

求解方法

- 精确算法：分支限界算法(Branch and Bound)、整数规划算法(Integer Programming)和回溯算法。

- 启发式算法: 优先指派算法(先来先服务FCFS，最短加工时间优先规则)、模拟退火算法(Simulated Annealing)、禁忌搜索算法(Tabu Search)以及遗传算法(Genetic Algorithm)等。

- 近似算法：能给出解的优度与最优解差值的一类算法，但是找到的解的质量往往不高。

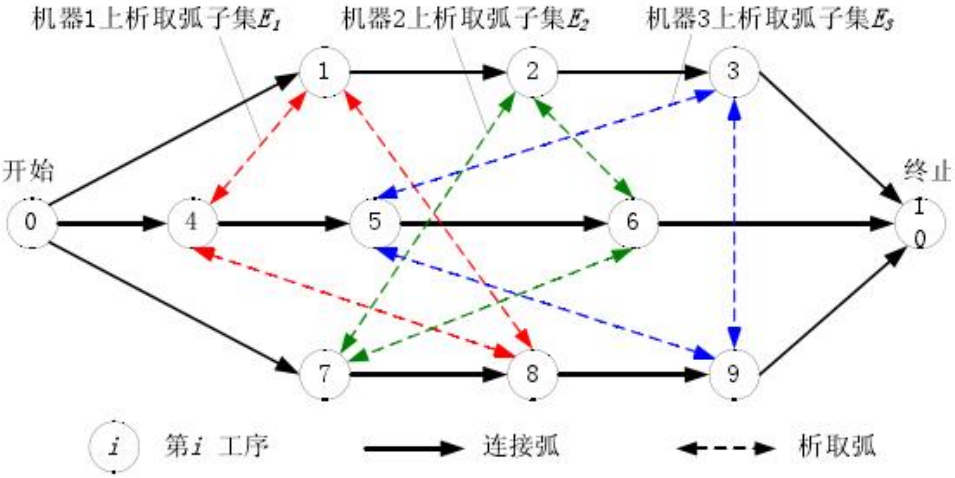
研究背景

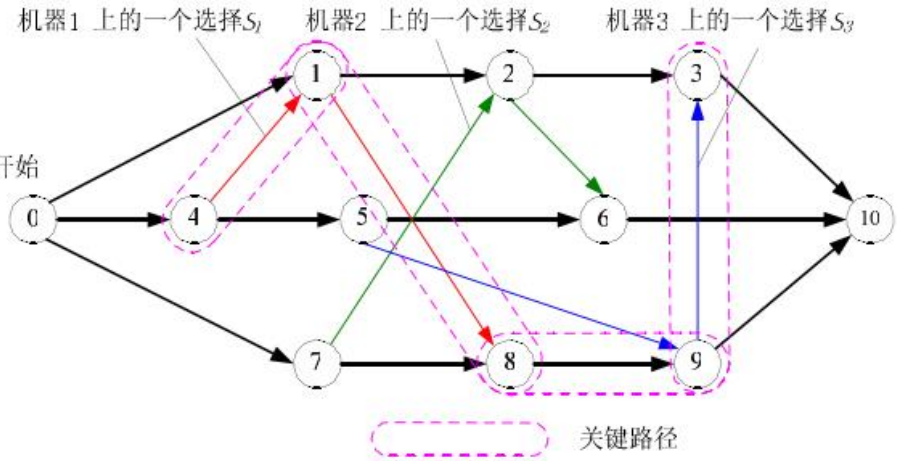
- **i-TSAB** by Nowicki and Smutnicki (2005)
  Journal of Scheduling
- **GES** by Pardalos and Shylo (2006)
  Computational Management Science
- **TS** and **TS/SA** by Zhang et al. (2007, 2008)
  Computers & Operations Research
- **AlgFix** by Pardalos et al. (2010)
  Computational Optimization and Applications
- **CP/LS** by Beck et al. (2011)
  Informs Journal on Computing
- **GES/TS** by Nasiri and Kianfar (2012)
  Computers & Industrial Engineering
- **BRKGA** by Goncalves and Resende (2013)
  International Transactions in Operational Research
- **HEA** by Cheng et al. (2013)
  Annals of Operations Research

基本理论

- 关键路径：从起点到终点的最长路径，其长度就是Makespan。
- 关键工序：在关键路径上的工序。

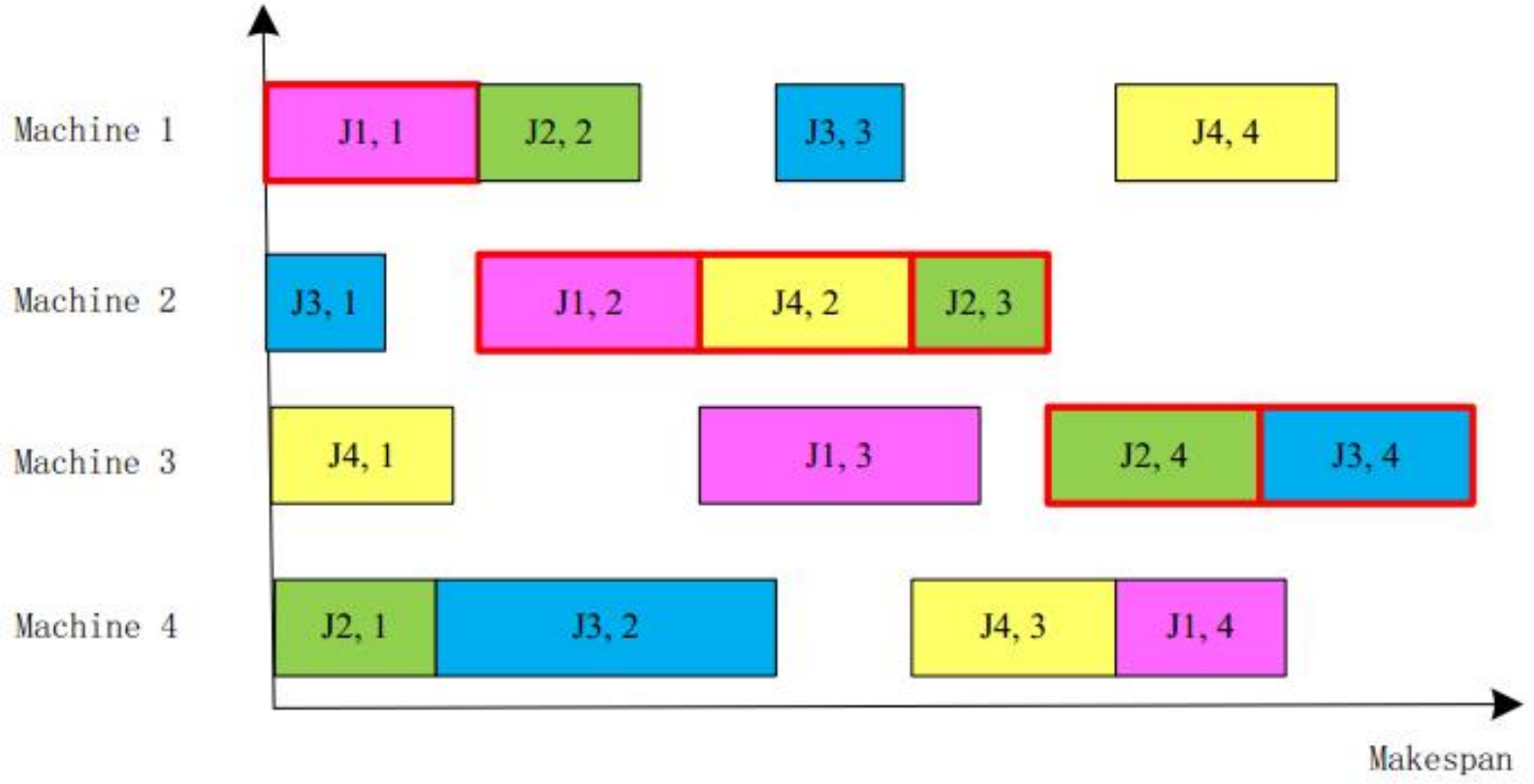
工件	加工顺序（机器序列，加工时间）		
J1	(1, 3)	(2, 2)	(3, 5)
J2	(1, 3)	(3, 5)	(2, 1)
J3	(2, 2)	(1, 5)	(3, 3)





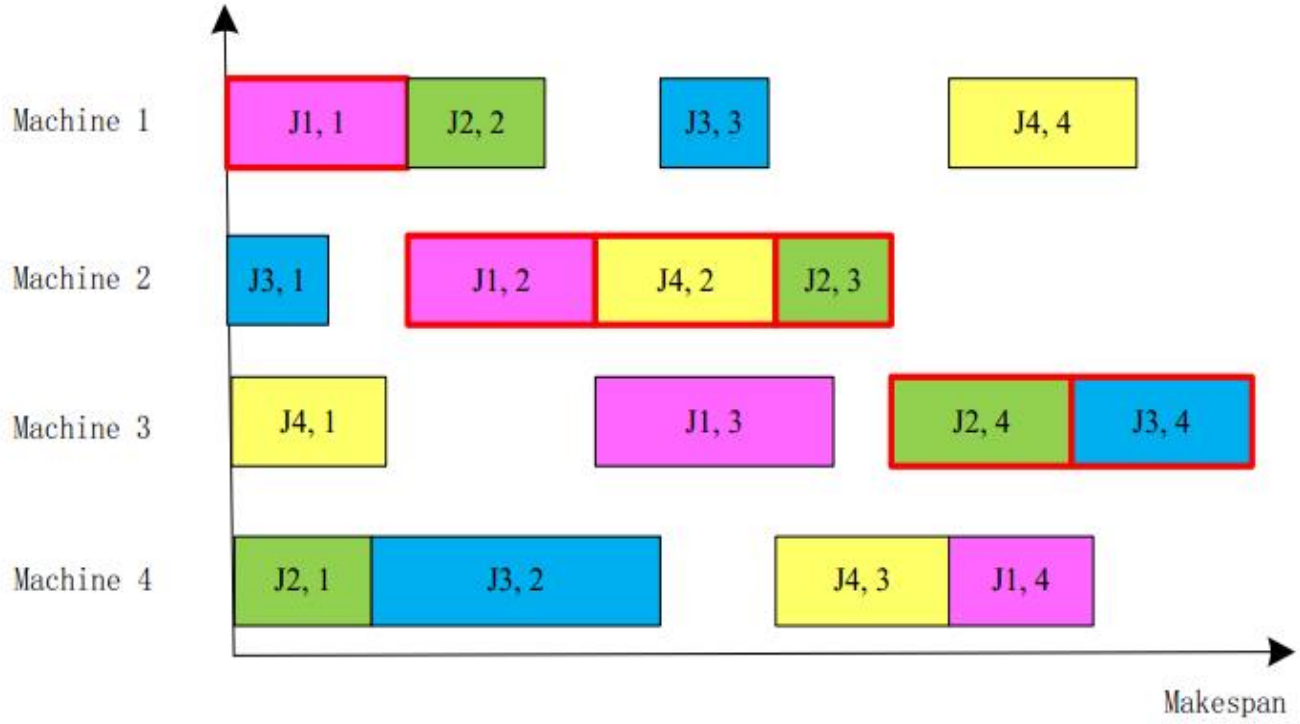
概念定义

- 关键块：同一台机器上最大序列的连续关键工序。



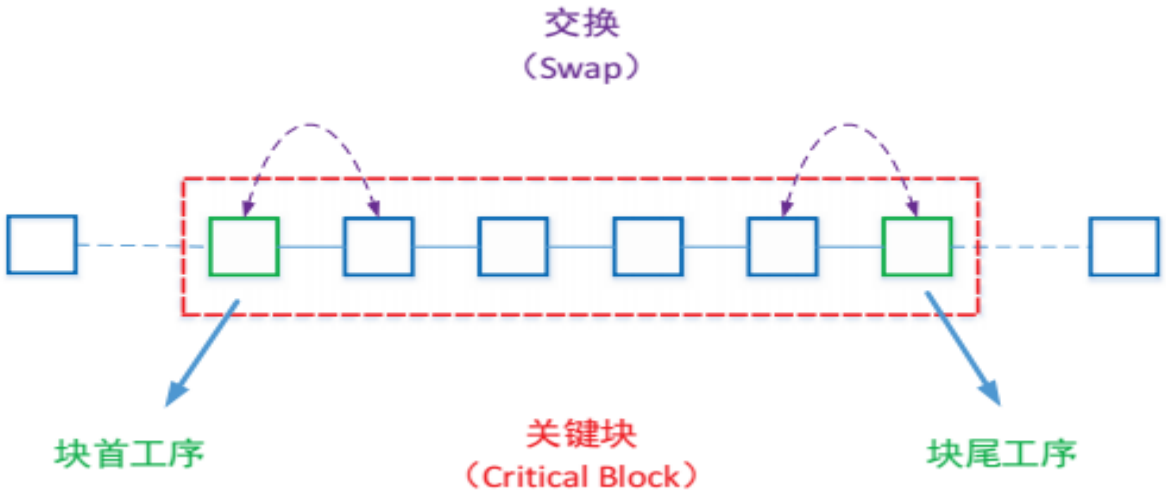
邻域结构

- 邻域动作（N1-N7）都是基于关键路径的操作。
- Matsuo在1988提出，交换关键块内的工序不能减小Makespan。
- 现今最有效的，Nowicki和Smutnicki提出的N5邻域结构，Balas和Vazacopoulos的N6，以及Zhang的N7。



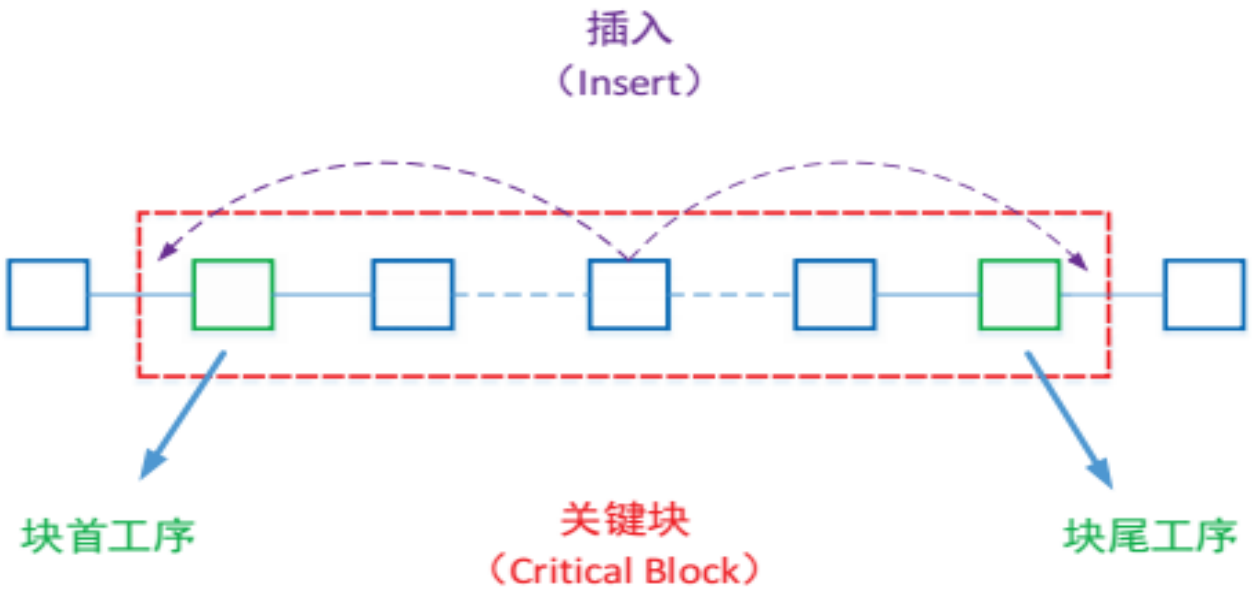
邻域动作N5

- N5邻域结构是Nowicki和Smutnicki在1992年提出并发表在Management Science上的。
- 仅交换块首和块尾两相邻工件以避免交换关键块内的工件。
- 优点：邻域结构小，速度快；缺点：过度限制无效移动，使领域连通能力和搜索效率受到削弱。



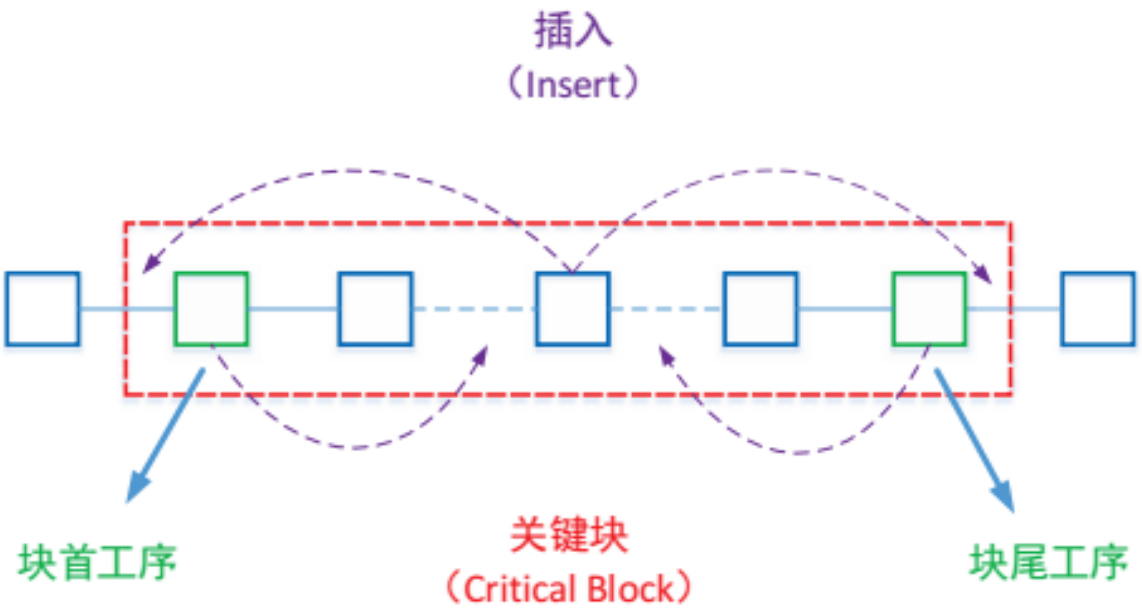
邻域动作N6

- N6邻域结构是Balas和Vazacopoulos在1988年提出并发表在Management Science。
- 把内部工件移动到块首之前和块尾之后。一直沿用至今，仍是最有效的邻域结构之一。



邻域动作N7

- N7邻域结构是张超勇在2007年提出并发表在COR上。
- 在N6的基础上增加了一种邻域动作即把关键块的首尾工件移动至关键块内部。一直沿用至今，仍是最有效的邻域结构之一。



邻域动作比较

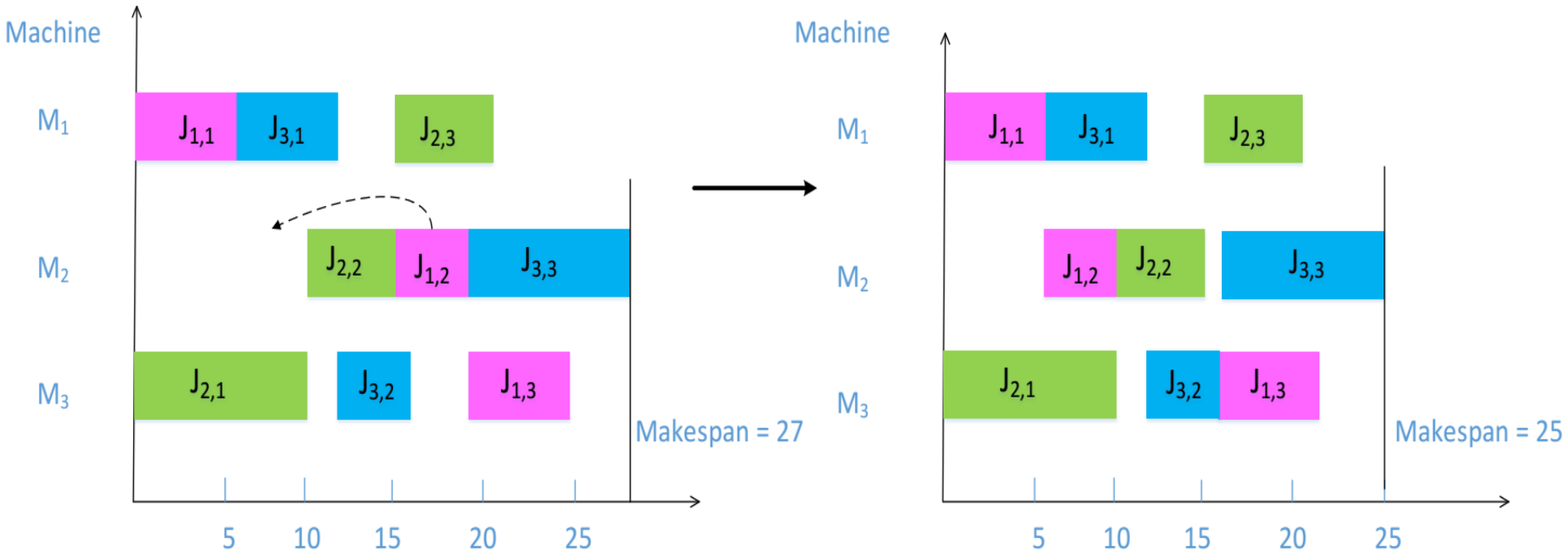
	N7	N6	N5
MC_{max}	**1416**	1420	1462
MEN	106163	70216	**48973**
CPU-time	298	205	**121**

- MCmax : Average makespan.
- MEN: The mean number of evaluated neighbors.
- CPU-time: The total CPU time performed in all (72) instances.

邻域评估策略

(1)精确评估策略

(2)近似评估策略



近似邻域评估策略

定义：

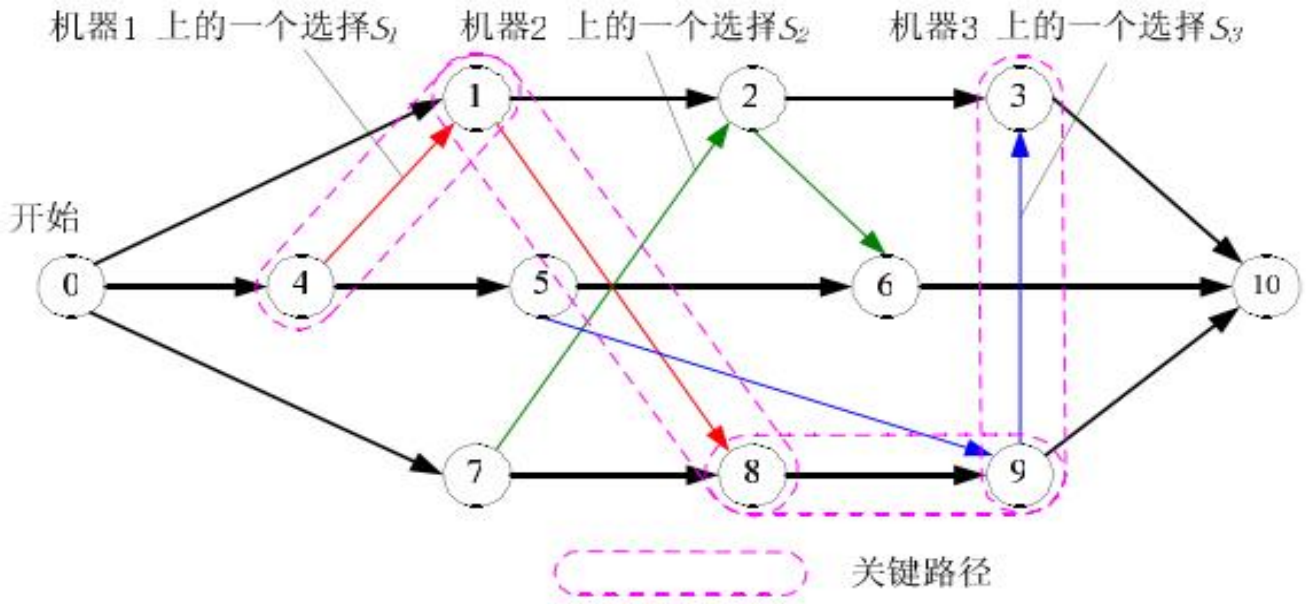
R[i]: 表示从起始工序到工序 i 的最长路径长度。

Q[i]: 表示从工序 i 到终止工序的最长路径长度。

求法：

R[i] = Max{R[j] + T[j], R[k] + T[k]}; j,k为i的前序工序。

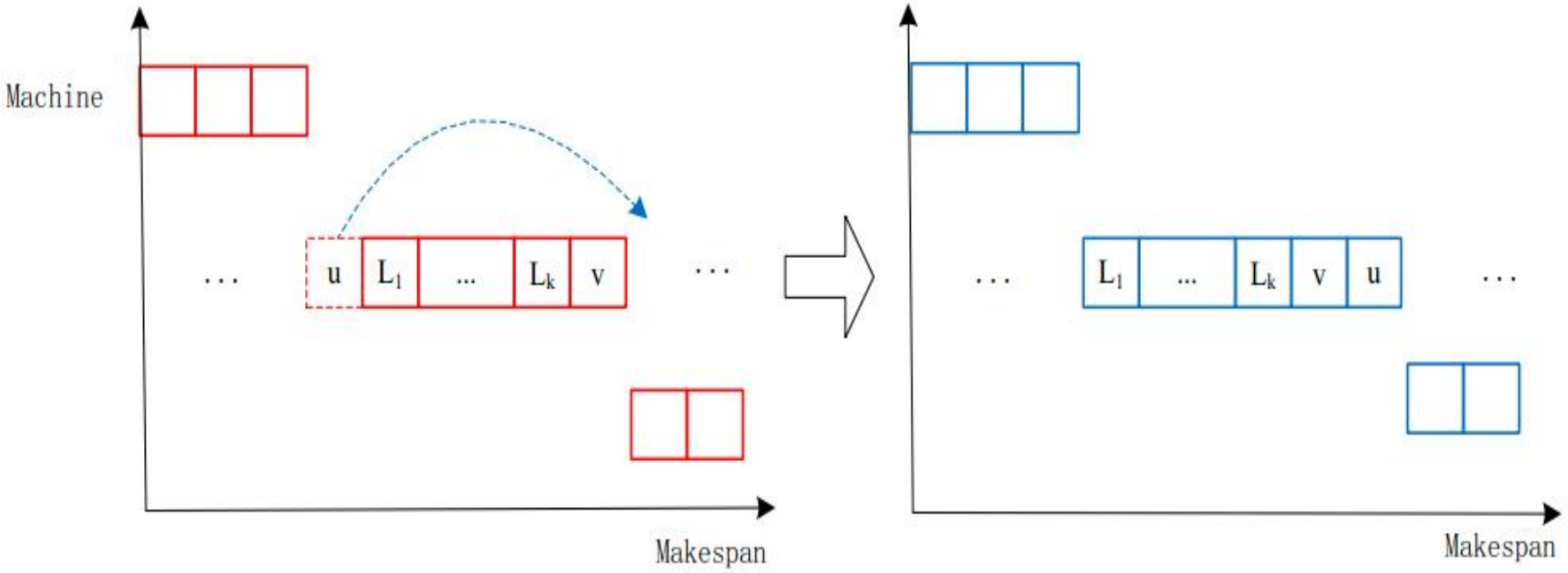
Q[i] = Max{Q[j], Q[k]} + T[i];　　j,k为i的后序工序。



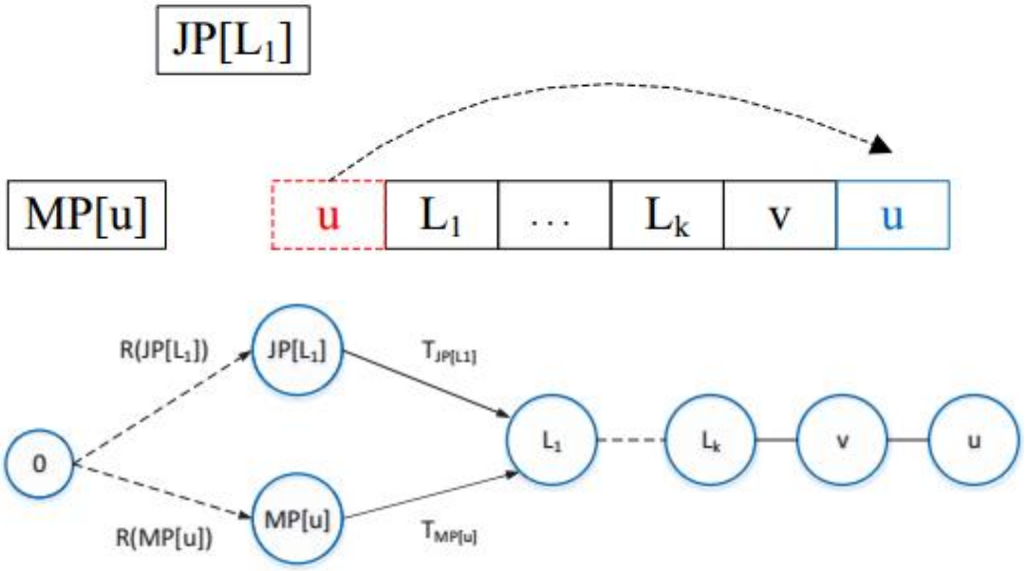
近似邻域评估策略

近似评价值：

$$Makespan^{u,v} = Max\{ R^{u,v}(w) + Q^{u,v}(w) \}; \qquad w \in \{ u, L_1, \cdots, L_k, v \}.$$



近似邻域评估策略



$R^{u,v}$ 的求法：

$$R^{u,v}(L_1)=\begin{cases} R(JP[L_1])+T_{JP[L_1]}; & \text{若u是机器上的第一个工序} \\ Max\{R(JP[L_1])+T_{JP[L_1]},R(MP[u])+T_{MP[u]}\}; & \text{否则} \end{cases}$$

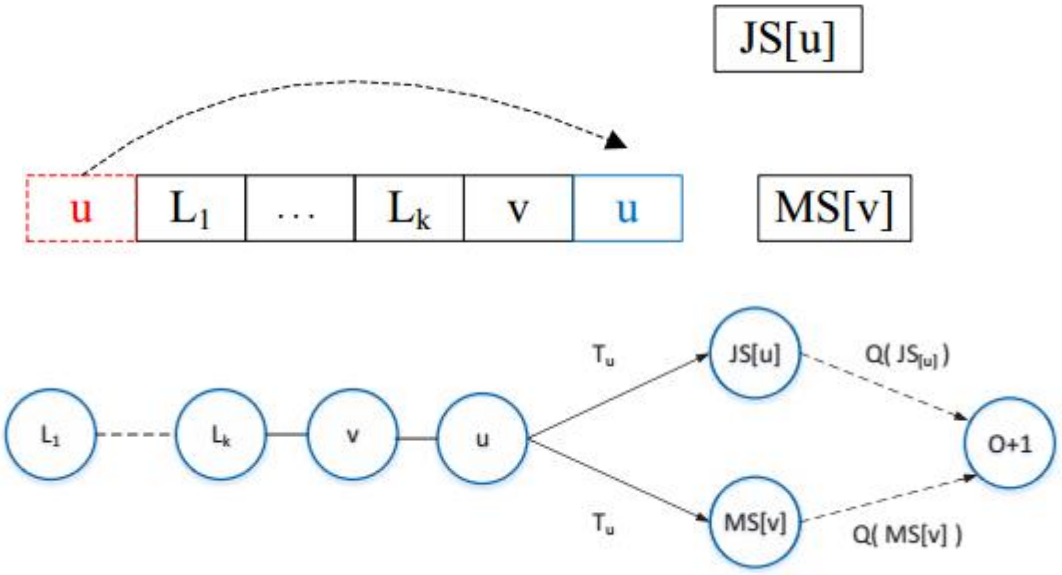
对于$w \in L_2,\ldots,L_k,v$,

$$R^{u,v}(w)=Max\{R(JP[w])+T_{JP[w]},\ R^{u,v}(MP[w])+T_{MP[w]}\}.$$

$$R^{u,v}(u)=Max\{R(JP[u])+T_{JP[u]},\ R^{u,v}(v)+T_v\}.$$

其中，$JP[i]$表示与工序i的同工件的先序工序，$MP[i]$表示与工序i同机器的先序工序。

近似邻域评估策略



$Q^{u,v}$的求法：

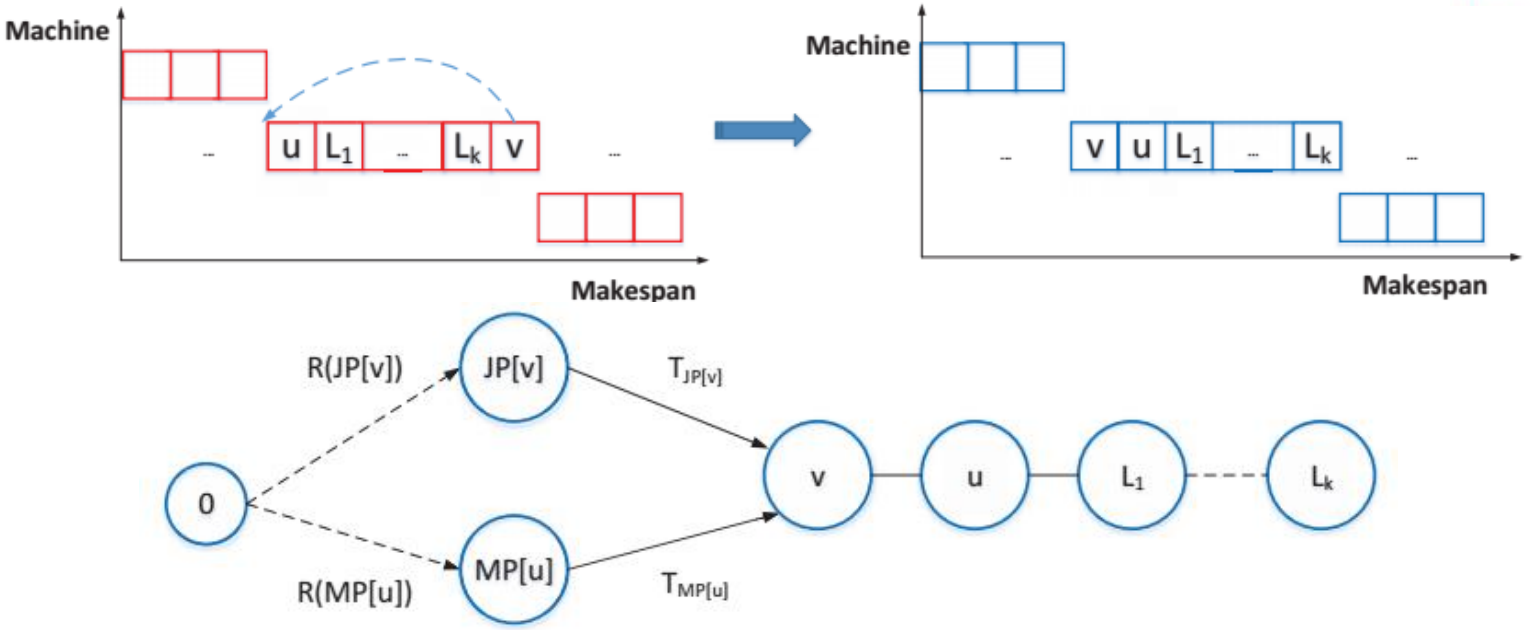
$$Q^{u,v}(u) = \begin{cases} Q(JS[u]) + T_u; & \text{若v是机器上的最后一个工序} \\ Max\{Q(JS[u]), Q(MS[v])\} + T_u; & \text{否则} \end{cases}$$

$$Q^{u,v}(v) = Max\{Q(JS[v], Q^{u,v}(u))\} + T_v;$$

对于$w \in L_1, \ldots, L_k$,

$$Q^{u,v}(w) = Max\{Q(JS[w]), Q^{u,v}(MS[w])\} + T_w$$

近似邻域评估策略



$R^{u,v}$ 的求法：

$$R^{u,v}(v) = \begin{cases} R(JP[v]) + T_{JP[v]}; & \text{若u是机器上的第一个工序} \\ Max\{R(JP[v] + T_{JP[v]}, R(MP[u]) + T_{MP[u]}\}; & \text{否则} \end{cases}$$

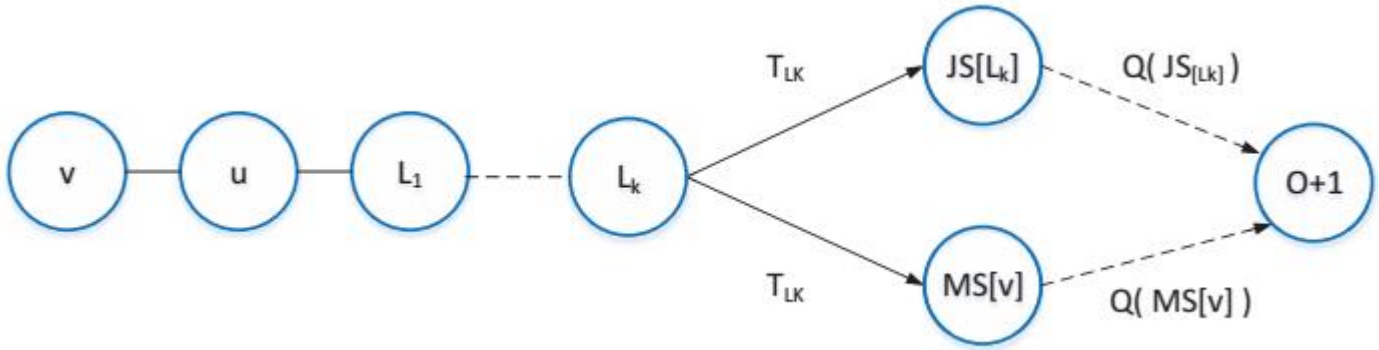
$$R^{u,v}(u) = Max\{R(JP[u]) + T_{JP[u]},\ R^{u,v}(v) + T_v\}$$

对于$w \in L_1, \ldots, L_k$,

$$R^{u,v}(w) = Max\{R(JP[w] + T_{JP[w]}),\ R^{u,v}(MP[w]) + T_{MP[w]}\}$$

其中，$JP[i]$表示与工序i的同工件的先序工序，$MP[i]$表示与工序i同机器的先序工序。

近似邻域评估策略



$Q^{u,v}$的求法：

$$Q^{u,v}(l_k)=\begin{cases} Q(JS[l_k])+T_{l_k}; & \text{若v是机器上的最后一个工序} \\ Max\{Q(JS[l_k],Q(MS[v]))\}+T_{l_k}; & \text{否则} \end{cases}$$

对于$w \in u, L_2, \ldots, L_{k-1}$，

$$Q^{u,v}(w)=Max\{Q(JS[w]),\ Q^{u,v}(MS[w])\}+T_w$$

$$Q^{u,v}(v)=Max\{Q(JS[v]),\ Q^{u,v}(u)\}+T_v$$

其中，$JS[i]$表示与工序i的同工件的后序工序，$MS[i]$表示与工序i同机器的后序工序。

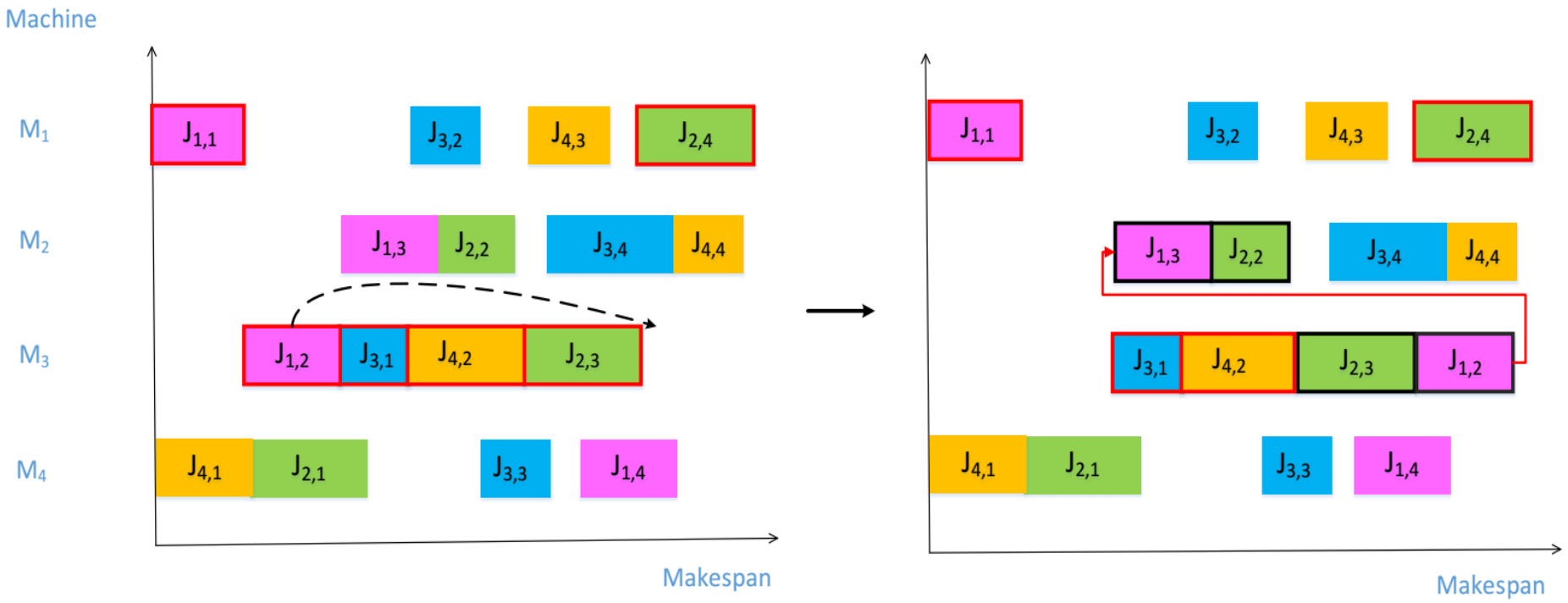
近似邻域评估策略

实验结果显示，近似方法花费的计算时间一般为精确方法的10-20%，并且问题规模越大，近似方法的执行效率也越高。一般直觉认为的精确方法在解的质量方面优于近似方法，但是实验结果没有证实这个观点。

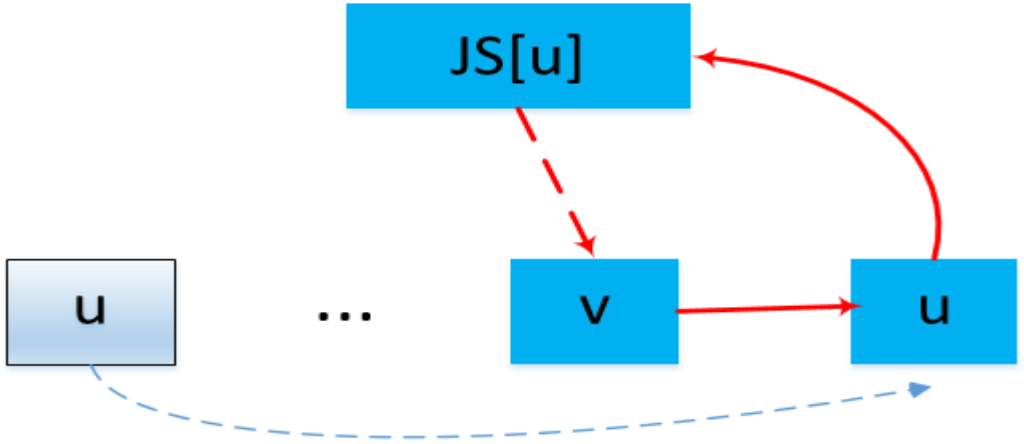
表 3.3　近似方法与精确方法测试结果比较

	MC_{max}	MRE	NBE	MEN	MNI	CPU-time	Tabu list size
近似方法	3338	4.42	40	215598	10445	367.2	12
精确方法	3339	4.44	41	234598	10785	2206.8	12

动作合法性


Machine
M1
M2
M3
M4
J1,1
J3,2
J4,3
J2,4
J1,3
J2,2
J3,4
J4,4
J1,2
J3,1
J4,2
J2,3
J4,1
J2,1
J3,3
J1,4
Makespan
J1,1
J3,2
J4,3
J2,4
J1,3
J2,2
J3,4
J4,4
J3,1
J4,2
J2,3
J1,2
J4,1
J2,1
J3,3
J1,4
Makespan

合法性定理



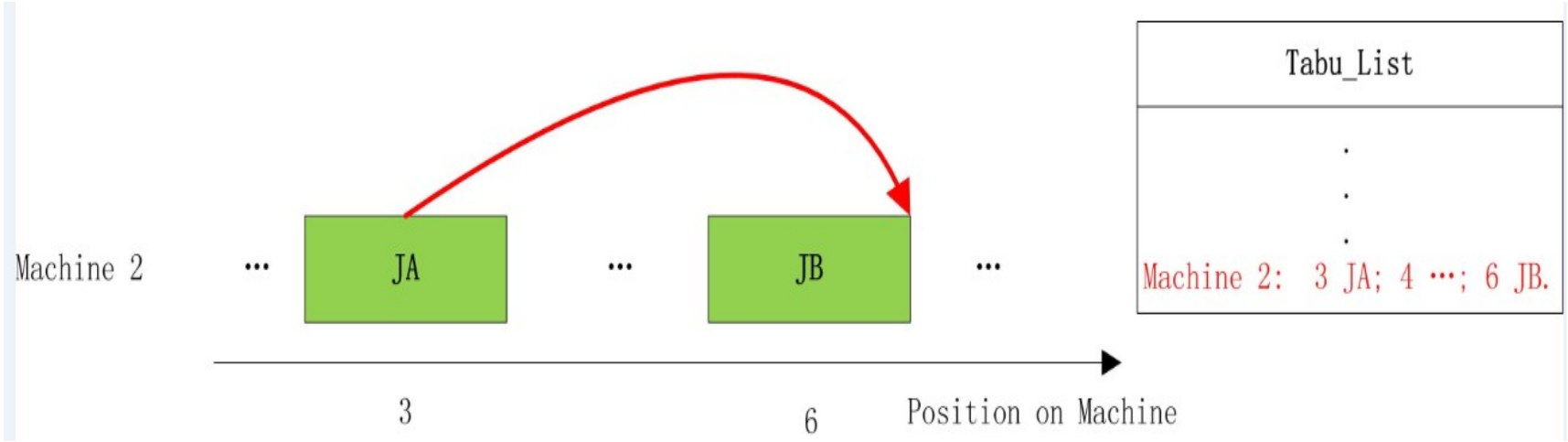
定理：设一个可行解 s，如果其两个关键工序 u 和 v 在同一台机器上，且 Q(V) ≥ Q(JS[u])，那么移动工序 u 至工序 v 之后将产生非循环的完全选择。

证明:
命题A: 如果存在从JS[u]到V的路径 , 那么 Q(JS[u]) ≥ Q(v) + L(JS[u], v) > Q(v)。

命题A的逆否命题： 如果Q(v) ≥ Q(JS[u]), 那么 JS[u] 不存在到V的路径。

禁忌对象

禁忌部分解（格局），而非动作。



拓展阅读

(1)论文题目：
Guided Local Search with Shifting Bottleneck for Job Shop Scheduling
作者：E Balas，A Vazacopoulos

(2)论文题目：
基于自然启发式算法的作业车间调度问题理论与应用研究
作者：张超勇

谢谢！