中国科学: 数学 2020 年 第 50 卷 第 9 期: 1183～1200
SCIENTIA SINICA Mathematica
《中国科学》杂志社 SCIENCE CHINA PRESS
CrossMark click for updates

论 文

平行机在线排序综述

献给越民义教授 100 华诞

林凌[1], 谈之奕[2*]

1. 浙大城市学院计算机与计算科学学院, 杭州 310015;
2. 浙江大学数学科学学院, 杭州 310027
E-mail: linl@zucc.edu.cn, tanzy@zju.edu.cn

收稿日期: 2020-02-29; 接受日期: 2020-04-25; 网络出版日期: 2020-09-10; * 通信作者
国家自然科学基金 (批准号: 11801505 和 11671356) 和浙江省教育厅科研项目基金 (批准号: Y201533189) 资助项目

摘要 平行机排序是一类重要的组合优化问题, 列表在线排序是在线问题研究形成和发展的重要推手, 也是成果最为丰富的在线问题之一. 本文回顾以最大完工时间为目标的在线和半在线排序问题的最新进展, 总结工件可拒绝、机器可增加及目标为机器负载的 L^p 范数等 3 类复杂目标在线排序问题的主要结果, 介绍竞争比近似方案、带建议的在线算法和多样化算法性能指标等 3 个在线排序新课题.

关键词 排序 在线 竞争比

MSC (2010) 主题分类 90B35, 90C27

1 引言

在线问题 (online problem) 是 20 世纪 80 年代起逐渐形成的一个新的研究方向, 可用于求解在线问题的算法称为在线算法 (online algorithm). 在线的两个基本特征是实例信息随算法执行过程逐渐披露、以及算法决策不可因后续出现的信息而更改. 相应地, 实例信息在算法执行前全部已知的问题称为离线问题 (offline problem). 在线问题及算法可抽象为在算法设计者和对手 (adversary) 之间进行的一个需求 - 应答博弈 (request-answer game). 对手每给出一个需求, 算法根据此前双方的应对和当前需求决定如何应答, 而对手再根据至目前为止双方的应对给出下一个需求. 算法的目标是使其性能尽可能好, 而对手的目标则恰好相反.

为对一个在线算法求解在线问题的性能作出定量的评估, 最坏情况比被移植到在线问题的研究中, 并结合其特点称其为竞争比 (competitive ratio). 称

$$\rho=\inf\{r\geqslant 1 \mid C^A(I)\leqslant rC^*(I),\forall I\} \quad (\rho=\inf\{r\geqslant 1 \mid C^*(I)\leqslant rC^A(I),\forall I\})$$

英文引用格式: Lin L, Tan Z Y. Online scheduling on parallel machines: A survey (in Chinese). Sci Sin Math, 2020, 50: 1183–1200, doi: 10.1360/SSM-2020-0061

为一个极小化 (极大化) 在线问题的算法 A 的竞争比, 这里 $C^A(I)$ 和 $C^*(I)$ 分别为算法 A 求解实例 I 所得的目标值和实例 I 的最优目标值. 由于在线问题实例信息未在算法执行前全部提供, 可能出现任何在线算法, 不论其时间复杂度为何, 都不能求得最优值甚至竞争比无法小于某个数值的情况. 称在线问题下界 (lower bound) 为 ρ, 若任意在线算法求解该问题的竞争比至少为 ρ. 若一个问题下界为 ρ, 且 A 是求解该问题竞争比为 ρ 的算法, 则称 A 为最好 (optimal 或 best possible) 算法.

设计在线算法并证明其竞争比, 与证明问题下界一并构成了在线问题研究的基本方法, 称为竞争比分析 (competitive ratio analysis). 寻找一个在线问题的最好算法是研究的主要目标, 这需要通过设计算法和证明下界两方面的共同努力. 竞争比分析的主要框架在 Sleator 和 Tarjan 的在线问题奠基性文献 [1] 中已经给出, 并在此后不断丰富. 关于在线问题的专著和综述可参见文献 [2,3].

排序 (scheduling) 问题是组合优化中的主要问题之一. 其基本描述是将若干工件按一定规则安排在一台或多台机器上的某些时间段内加工, 形成一个可行排序 (feasible schedule), 并使该排序的某个目标函数达到最优. 在线问题一经提出, 便与排序理论深度融合. 在线排序成为排序的一个重要分支, 排序也为在线问题提供了丰富的研究内容. 在线排序模型除通常的列表在线 (online over list) 外, 还有凸显排序特色的实时在线 (online over time). 前者的特点是工件信息按某个列表顺序逐个给出, 算法在安排某个工件时不掌握列表中位于该工件之后的工件的信息. 后者的特点是每个工件有一个准备时间 (release time), 工件的信息在其准备时间到达后给出.

排序对在线问题发展的另一项贡献是孕育了半在线 (semi-online) 这一新的研究课题. 所谓半在线是介于在线与离线之间的一种过渡形态. 以排序而言, 或者已知后续工件的部分信息, 或者可使在线算法部分改变之前所做的决策. 半在线在现实中普遍存在, 如果不对其加以研究, 对相应的问题既不能用离线问题的算法求解, 用在线算法求解又可能失去额外信息或特殊机制带来的益处. 半在线研究也为从理论上揭示在线问题的性质、在线与离线之间的联系和界限提供了新的思路. 半在线起源于排序并不偶然, 这与排序丰富的模型和大量的研究积累是分不开的. 目前, 半在线研究已推广至其他组合优化问题, 催生了带建议的算法 (详见第 4 节) 等新的概念.

在三十多年的时间里, 平行机在线和半在线排序研究模型众多、成果丰硕. 文献 [4,5] 是在线排序发展初期的两篇重要综述. Tan 和 Zhang 撰写的综述文献 [6] 首次较为完整地介绍了当时的半在线研究成果. 近年来, 在线和半在线排序研究又取得了很多重要进展, 有必要进行增补. 限于篇幅, 本文只介绍平行机列表在线排序的相关结果. 对文献 [6] 中已涉及的内容, 主要介绍新的成果. 本文增加了 3 类未在文献 [6] 中介绍的问题, 并对在线排序 3 个新的方向予以简要介绍.

本文用 $\mathcal{J}=\{J_1,\dots,J_n\}$ 表示工件集, $\mathcal{M}=\{M_1,\dots,M_m\}$ 表示机器集. 工件 J_j 在 M_i 上的加工时间为 p_{ji}, $j=1,\dots,n$, $i=1,\dots,m$. 若存在 p_j $(j=1,\dots,n)$ 和 s_i $(i=1,\dots,m)$, 使得 $p_{ji}=\frac{p_j}{s_i}$, 则称相应的机器环境为同类机 (uniform machine), p_j 称为工件 J_j 的加工时间, s_i 称为机器 M_i 的速度. 若进一步有 $s_i=1, i=1,\dots,m$, 则称相应的机器环境为同型机 (identical machine). 不是同型机和同类机的机器环境称为不同类机 (unrelated machine). 有加工权限的平行机 [7–9] 是一类特殊的不同类机, 工件只能在部分机器上加工, 工件在可以加工的机器上加工时间相等或成比例, 相应地分别称为有加工权限的同型机和有加工权限的同类机. 对两台同类机, 用 s 表示两台机器的速度比. 称问题具有单位工件 (unit job) 特征, 若所有工件加工时间均为 1. 若不特别说明, 工件均相互独立, 从零时刻可开始加工. 加工可中断 (preemption) 是指工件可在一台机器上加工一段时间后中断加工, 并在之后继续加工其余部分. 加工可分割 (fractional) 是指工件可任意分割成多个部分, 且不同部分可同时在不同机器上加工. 对一个可行排序 σ, 用 $C_j(\sigma)$ 表示工件 J_j 在 σ 中的完工时间, $L_i(\sigma)$ 表示机器 M_i 在 σ 中的负载 (load). 列表在线排序中两个常见目标函数为工件最大完工时间

$C_{\max}(\sigma)=\max_{j=1,\dots,n}C_j(\sigma)=\max_{i=1,\dots,m}L_i(\sigma)$ 和机器最小负载 $C_{\min}(\sigma)=\min_{i=1,\dots,m}L_i(\sigma)$.

本文沿用文献 [10] 中提出的三参数排序法 (three-field notation) $\alpha|\beta|\gamma$ 和文献 [6] 中扩展的记号来表示一个排序问题. 本文中出现的各个域中的记号如下所示:

• α: P (同型机), Q (同类机), R (不同类机);

• β: pmpt (可中断), frac (可分割), $g=k$ (工件有 k 个不同等级), sum (已知工件总加工时间), opt (已知实例最优值), decr (已知工件加工时间递减), max (已知工件最大加工时间), UB&LB (已知工件加工时间上界与下界), buffer (缓冲), parallel (并行);

• γ: $C_{\max}$ (工件最大完工时间), $C_{\min}$ (机器最小负载), L^p (机器负载的 L^p 范数).

本节最后对涉及竞争比的几个问题给出说明. 第一, 在三参数表示法中, α 域中 Pm 表示 m 台同型机, 其中 m 为确定数, P 则表示任意台同型机. 若一个在线算法求解 $P|\beta|\gamma$ 的竞争比为 r, 则对任意固定的 m 台同型机相应问题 $Pm|\beta|\gamma$, 算法的竞争比不超过 r. 在线问题 $P|\beta|\gamma$ 的下界应理解为 $Pm|\beta|\gamma$ 下界 ρ_m 的上确界 $\sup_{m\geqslant 1}\rho_m$. 因此, 若 $P|\beta|\gamma$ 下界为 ρ, 则对某个固定的 m, $Pm|\beta|\gamma$ 可能存在竞争比小于 ρ 的算法. 第二, 部分排序问题带有一个或多个参数, 如同类机问题、已知工件加工时间上界与下界的半在线问题等, 问题下界和算法竞争比也可写为这些参数的函数. 设问题含有 k 个参数 $u_1,\dots,u_k$, 其下界为 $\rho(u_1,\dots,u_k)$, 一在线算法的参数竞争比为 $r(u_1,\dots,u_k)$. 若对任意的 $u_1,\dots,u_k$, 有 $\rho(u_1,\dots,u_k)=r(u_1,\dots,u_k)$, 则称算法为参数最好的. 分别称 $r=\sup_{u_1,\dots,u_k}r(u_1,\dots,u_k)$ 和 $\rho=\sup_{u_1,\dots,u_k}\rho(u_1,\dots,u_k)$ 为算法的常数竞争比和问题的常数下界. 若 $r=\rho$, 则称算法是常数最好的. 常数最好的算法未必是参数最好的. 第三, 若存在与实例 I 无关的常数 c, 使得 $C^A(I)\leqslant rC^*(I)+c$ 对任意的实例 I 均成立, 则称 r 为在线算法 A 的渐近竞争比. 若一在线问题不存在渐近竞争比小于 ρ 的在线算法, 则称 ρ 为渐近下界. 对单位工件等少数排序问题, 渐近竞争比 (渐近下界) 可能严格小于竞争比 (下界).

本文余下内容结构如下. 第 2 节主要介绍同型机、同类机和有等级平行机在线、半在线排序问题的最新进展. 第 3 节介绍工件加工可拒绝、机器可增加和目标为极小化机器负载的 L^p 范数 3 类在线排序问题. 第 4 节介绍竞争比近似方案、带建议的在线算法以及多样化算法性能指标等 3 个在线排序新的研究方向. 第 5 节总结全文并给出部分公开问题.

2　以最大完工时间为目标的在线排序问题

2.1　同型机和同类机经典在线排序问题

作为平行机在线排序中的基础性问题, $Pm||C_{\max}$ 一直受到研究者的重视. 1966 年, Graham [11] 提出了在线排序中有深远影响的 LS (list scheduling) 算法, 并证明了其竞争比为 $2-\frac{1}{m}$. LS 算法将工件安排在能使其最早完工的机器上加工. 直至二十多年后, 在线问题被系统提出, 竞争比分析理论渐趋成熟, 学界认识到 LS 算法是一个典型的在线算法, 且为 $m=2,3$ 时的最好算法 [12]. 至目前为止, LS 算法仍是 $Pm||C_{\max}$ 系列问题中唯一已知的最好算法. 自 1989 年以来, 算法和问题下界不断得到改进. 当 m 为任意值时, 目前已发表的最佳下界和最佳算法竞争比分别为 1.85358 [13] 和 1.9201 [14]. 尽管两者之间尚有不小的差距, 但二十年来几无进展. 下界改进困难无疑是因为所需构造的实例规模异常庞大、结构更加复杂, 而改进算法竞争比的困难并不仅在于算法设计本身. 多数组合优化问题最优值无法写为输入的显式函数. 因此在最坏情况分析中, 通常只能将算法所得排序的目标值与最优值的一个下界 (极大化问题则是上界) 加以比较. 如果最优值的下界与最优值本身存在较大的差距, 则能证得

的最坏情况比或竞争比会显著大于其真实值. Albers [15] 证明了若仅采用 $Pm||C_{\max}$ 目前已知的 3 个最优值的下界 $\frac{P}{m}$、$p_{(1)}$ 和 $\max_{k=1,\ldots,\lfloor\frac{n-1}{m}\rfloor}\sum_{j=km-k+1}^{km+1}p_{(j)}$, 则能证得的算法竞争比不小于 1.917. 这里 $P=\sum_{j=1}^{n}p_j$, $p_{(j)}$ 为加工时间按从大到小顺序居于第 j 位的工件的加工时间. 这一数值与目前最佳算法的竞争比非常接近, 说明长期未出现更好算法的原因更多地在于无法找到最优值更精确的估计.

文献 [16] 将这样的数值称为伪下界 (pseudo lower bound). 伪下界和下界同样是在线问题的固有属性, 但前者不仅体现了算法设计中的困难, 还反映了算法竞争比证明中的困难. 伪下界依赖于具体的最优值估计. Tan 和 Li [16] 给出了机器台数固定时基于前述 3 个最优值下界的伪下界. 当 $m=4,5,6$ 时, 其数值与 Chen 等 [17] 给出的最佳算法竞争比相同. 当然, 在机器台数较少时, 仍有可能采取更丰富的证明手段以突破伪下界的限制. 但直至目前, 尚未有这样的算法问世. 当机器台数较少时, 当前最佳下界、伪下界和最佳算法竞争比的数值如表 1 所示.

对 $Qm||C_{\max}$, 近年来, Ebenlendr 和 Sgall [19]、Jeż 等 [21] 分别给出了 m 任意和 m 较小时的改进常数下界. 这些下界的获得仍基于一个加工时间呈几何级数增长的工件集, 但在工件数量和机器速度设置上更为细致. $Qm||C_{\max}$ 的算法设计在近二十年内进展寥寥. 目前 m 任意时的最佳算法竞争比为 $3+2\sqrt{2}\approx 5.828$ [22], 而 m 较小时的最佳结果来自 1978 年 Cho 和 Sahni [20] 对 LS 算法竞争比的估计. 从表 1 中给出的数值可以看到, 问题下界和算法竞争比的差距明显大于同型机. 值得考虑的问题是可否通过给出伪下界以明确未来研究的重点方向.

一些文献研究了 m 较小时的参数下界及竞争比, 以及特殊机器速度下的算法设计. LS 算法是求解 $Q2||C_{\max}$ 的参数最好算法, 其竞争比为 $r_{Q2}(s)=\min\{\frac{s+2}{s+1},\frac{s+1}{s}\}$ [28], 这仍是目前 $Qm||C_{\max}$ 系列问题中唯一已知的参数最好算法. 在一项最近的研究中, Dolgui 等 [29] 考虑了 k 台机器速度为 s $(1<s\leqslant 2)$、$m-k$ 台机器速度为 1 的情形, 设计了竞争比不超过 2.618 的算法. 关于同型机和同类机经典在线排序的其他相关结果可参见文献 [6].

2.2 已知部分信息的半在线排序问题

已知部分信息的多台同型机半在线排序研究近年来取得了一些突破. 2015 年, Kellerer 等 [30] 给出了 $P|\mathrm{sum}|C_{\max}$ 竞争比为 $\alpha\approx 1.585$ 的算法, 其中 α 为方程 $4x^3-8x^2+2x+1=0$ 的正根. 这一个数值与先前 Albers 和 Hellwig [31] 给出的下界相等, 这使已知工件总加工时间成为同类问题中第一个得到任意台同型机最好算法的半在线模型. $P|\mathrm{opt}|C_{\max}$ 的算法则经历了多次改进 (参见文献 [32–34]), 目前最佳算法竞争比为 $\frac{3}{2}$. 另有一些文献对机器台数较少时的问题下界和算法进行了讨论, 详细结果见表 2. 需要特别指出的是, 普遍猜测但长期未有定论的 $Pm|\mathrm{opt}|C_{\max}$ 大于 $\frac{4}{3}$ 的下界, 近期在计算机辅

表 1 以最大完工时间为目标的在线排序问题相关结果

	m	任意	2	3	4	5	6	7
同型机	下界	1.85358 [13]	$\frac{3}{2}$ [12]	$\frac{5}{3}$ [12]	$\sqrt{3}$ [18]	1.746 [17]	1.773 [17]	1.791 [17]
	伪下界	1.917 [15]	–	–	$\frac{26}{15}$ [16]	$\frac{85}{48}$ [16]	$\frac{9}{5}$ [16]	1.815 [16]
	竞争比	1.9201 [14]	$\frac{3}{2}$ [11]	$\frac{5}{3}$ [11]	$\frac{26}{15}$ [17]	$\frac{85}{48}$ [17]	$\frac{9}{5}$ [17]	$\frac{125}{96}$ [17]
同类机	下界	2.564 [19]	$\frac{1+\sqrt{5}}{2}$ [20]	2 [20]	2.141 [21]	2.314 [21]	2.439 [21]	2.439 [21]
	竞争比	$3+2\sqrt{2}$ [22]	$\frac{1+\sqrt{5}}{2}$ [20]	2 [20]	2.225 [20]	2.414 [20]	2.581 [20]	2.732 [20]
等级设置 I 的	下界	e [23]	$\frac{5}{3}$ [24, 25]	2 [26]				
有等级同型机	竞争比	e + 1 [23]	$\frac{5}{3}$ [24, 25]	2 [26]	2.294 [27]	2.502 [27]	$\frac{25}{9}$ [26]	$\frac{1657}{586}$ [26]

表 2 多台同型机和有等级同型机以最大完工时间为目标的半在线排序问题相关结果

	m	sum		opt		max		decr		sum, max	
		下界	竞争比	下界	竞争比	下界	竞争比	下界	竞争比	下界	竞争比
同型机	任意	1.585 [31]	1.585 [30]		$\frac{3}{2}$ [34]				$\frac{5}{4}$ [35]		
	3	$\frac{\sqrt{129}-3}{6}$ [36]	$\frac{7}{5}$ [37]	$\frac{15}{11}$ [38]	$\frac{11}{8}$ [38]			$\frac{1+\sqrt{37}}{6}$ [39]	$\frac{1+\sqrt{37}}{6}$ [35]		
	4	$\frac{2\sqrt{10}-2}{3}$ [37]	$\frac{19}{13}$ [37]	$\frac{19}{14}$ [40]	1.462 [41]	$\frac{3+\sqrt{33}}{6}$ [37]	$\frac{1+\sqrt{5}}{2}$ [37]				
	5	$\frac{\sqrt{193}-5}{6}$ [37]	$\frac{3}{2}$ [37]		$\frac{3}{2}$ [41]	$\frac{3+\sqrt{33}}{6}$ [37]	$\frac{5}{3}$ [37]				
等级设置 I	3	$\frac{5}{3}$ [42]	$\frac{5}{3}$ [42]	$\frac{5}{3}$ [42]	$\frac{5}{3}$ [42]	$\frac{7}{4}$ [42]	$\frac{1+\sqrt{41}}{4}$ [42]			$\frac{5}{3}$ [42]	$\frac{5}{3}$ [42]
	4	$\frac{7}{4}$ [42]	$\frac{9}{5}$ [42]	$\frac{7}{4}$ [42]	$\frac{7}{4}$ [42]	1.8455 [42]	2 [42]			$\frac{7}{4}$ [42]	$\frac{3+\sqrt{17}}{4}$ [42]

助下获得了证实 (参见文献 [38,40]), 而 $P3|\text{decr}|C_{\max}$ 是目前 $m>2$ 且为固定数时唯一得到最好算法的问题. 两台同型机半在线问题成果更为丰富, 且绝大多数已得到最好算法, 相关结果可参见文献 [6].

对两台同类机半在线问题, 两台机器速度比 s 的值对问题性质有极大的影响. 因此给出问题的参数下界和算法的参数竞争比, 进而得到参数最好算法异常困难. Dósa 等 [43,44] 对 $Q2|\text{opt}|C_{\max}$ 部分 s 取值的下界和算法作出了改进, 尚未得到最好算法的 s 的区间长度, 竞争比与下界的差距均进一步缩小 (参见图 1). 从现有结果来看, 最好算法的参数竞争比非常复杂, 表现为分段多、分隔点无明显规律、函数形式多样, 研究难度不逊于多台同型机. Cao 和 Liu [59] 研究了 $Q2|\text{UB\&LB}|C_{\max}$, 该问题涉及两台机器速度比和工件加工时间上下界比两个参数. LS 算法仍是多数参数组合下的最好算法, 但在部分参数组合下, 存在优于 LS 的算法. 其他两台同类机半在线问题的研究结果可参见文献 [6].

2.3 特殊加工机制下的在线排序问题

工件按顺序安排加工且既有安排不可更改是列表在线的两个主要特征之一, 但现实中的系统对加工的限制可能并不如此严格. 充分利用特殊加工机制, 设计性能尽可能好的算法很有必要. 这一类型的半在线问题模型多样, 且多数带有参数. 参数的变化可诠释从在线到离线的演变过程. 参数的存在既为在算法性能和复杂度两者之间寻求平衡提供了可能, 也为研究带来了挑战. 根据加工机制的具体特点, 现有模型可分为并行、缓冲和重排三类, 但研究结果显示部分不同类别模型之间存在内在联系.

为行文方便, 约定以下术语, 将竞争比、问题下界和最好算法推广到带参数问题. 设模型带有一个参数 δ, δ 越小越接近在线问题, δ 越大越接近离线问题. 若当模型参数为 δ 时, 算法竞争比为 r, 则称算法的带参竞争比为 (r,δ). 若模型参数至多为 δ 时, 不存在竞争比小于 r 的在线算法, 则称问题的带参下界为 (r,δ). 若某算法带参竞争比为 (r,δ), 且对任意不依赖于实例的常数 $\delta'>\delta$, (r,δ') 均为问题的带参下界, 则称该算法为带参最好算法.

并行加工是最早提出的半在线模型之一 (参见文献 [60]). 在该模型中, 可将 $\delta\in\mathbb{Z}$ 个不同的在线算法作为子过程用于同一实例, 以其中最佳的结果作为算法输出. Kellerer 等 [60] 给出了两台同型机带参竞争比为 $(\frac{4}{3},2)$ 的带参最好算法, 此后多年来一直未有多台机的相关研究. 直至 2017 年, Albers 和 Hellwig [61] 分别给出了 $Pm||C_{\max}$ 带参竞争比为

$$\left(\frac{4}{3}+\varepsilon,\left(\frac{1}{\varepsilon}\right)^{O(\log\frac{1}{\varepsilon})}\right) \quad 和 \quad \left(1+\varepsilon,\left(\left\lfloor\frac{2m}{\varepsilon}\right\rfloor+1\right)^{\lceil\frac{\log\frac{2}{\varepsilon}}{\log(1+\frac{\varepsilon}{2})}\rceil}\right)$$

的在线算法. 同时还给出了问题的两个带参下界 $(\frac{4}{3},\lfloor\frac{m}{3}\rfloor)$ 和 $(1+\varepsilon,(\frac{\lfloor\frac{m}{2}\rfloor+\lfloor\frac{1}{4\varepsilon}\rfloor-1}{\lfloor\frac{1}{4\varepsilon}\rfloor-1})-1)$, 其中 $0<\varepsilon\leqslant\frac{1}{4}$.

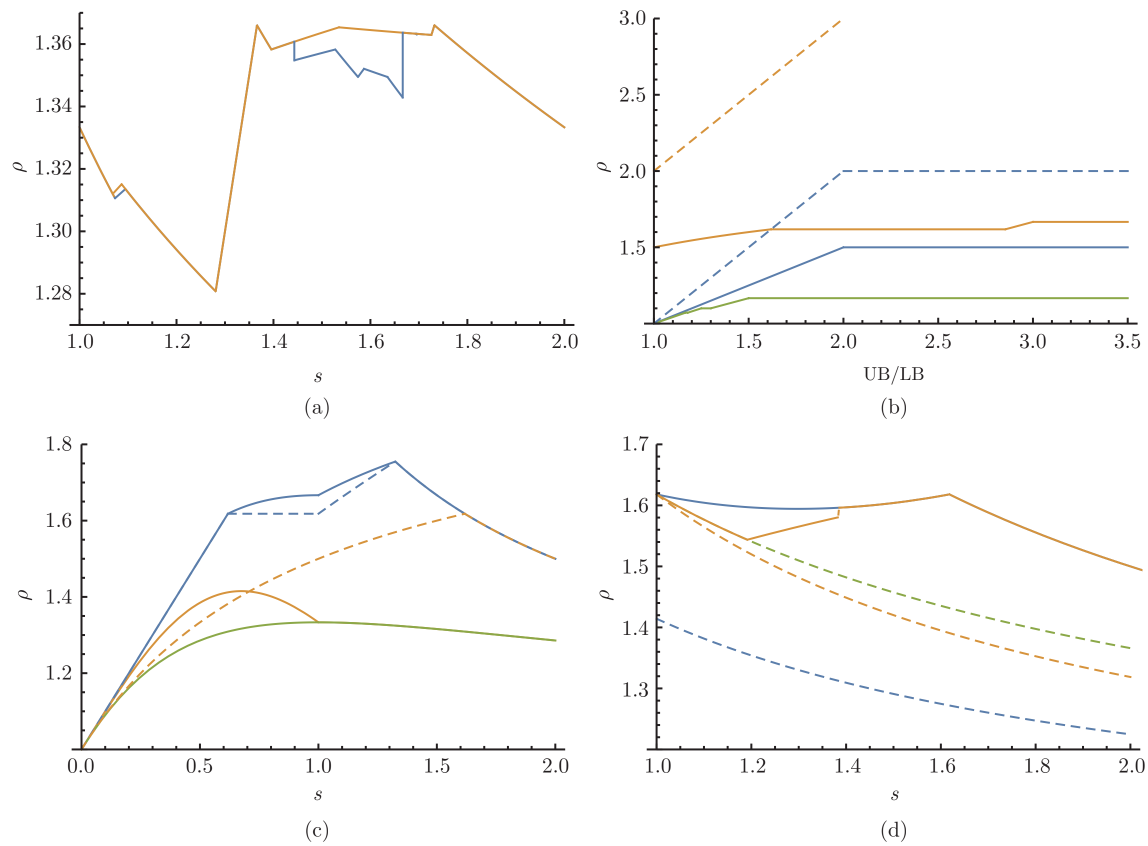

图 1 (网络版彩图) 部分问题参数下界和算法参数竞争比函数图像. **(a)** $Q2|\mathrm{sum}|C_{\max}$ 问题最佳参数下界与最佳算法参数竞争比 (参见文献 [43, 44]); **(b)** 已知工件加工时间上界和下界的半在线问题以加工时间上下界比为参数的参数最好算法竞争比 (实线: 自上至下, $P2|g=2$, UB&LB$|C_{\max}$ [45], $P2|$UB&LB$|C_{\max}$ [46] 与 $P2|g=2$, sum, UB&LB$|C_{\max}$ [47,48] (两问题竞争比相同), $P2|$decr, UB&LB$|C_{\max}$ [49]; 虚线: 自上至下, $P2|g=2$, UB&LB$|C_{\min}$ [50], $P2|$UB&LB$|C_{\min}$ [51] 与 $P2|g=2$, sum, UB&LB$|C_{\min}$ [50] (两问题竞争比相同)); **(c)** 两台有等级同类机在线与半在线问题最好算法竞争比 (实线: 自上至下, $Q2|g=2|C_{\max}$ [52], $Q2|g=2$, pmpt$|C_{\max}$ (参见文献 [53]), $Q2|g=2$, frac$|C_{\max}$ [54]; 虚线: 自上至下, $Q2|g=2$, max $|C_{\max}$ [55], $Q2|g=2$, opt$|C_{\max}$ 与 $Q2|g=2$, sum$|C_{\max}$ [55] (两问题竞争比相同)); **(d)** 两台同类机可拒绝在线与半在线问题下界与算法竞争比 (实线: 在线问题最佳参数下界与最佳算法参数竞争比 [56,57]; 虚线: 自上至下, 可中断 [56], 单位工件且可中断 [58], 单位工件并按罚值非增顺序到达且可中断 [58] 三问题最好算法竞争比)

缓冲 (buffer) 是指在系统中存在一个容量为 $\delta\in\mathbb{Z}$ 的缓冲区. 工件到达后可以直接安排加工, 也可以放在缓冲区内. 如果此时缓冲区内已有 δ 个工件, 则需立即加工缓冲区内的某个工件 (参见文献 [60,62]). 该模型也可作下面的解释. 现有长度为 $\delta+1$ 的特殊队列, 工件按原有顺序逐个进入队列, 随后队列中的工件可任意排列顺序. 当队列中工件数达到 $\delta+1$ 后, 需先将排在队列顶部的工件取出安排加工, 再接受新的工件进入队列. 记 γ_m 为方程

$$\left\lceil m-\frac{m}{\gamma_m}\right\rceil\frac{\gamma_m}{m}+(\gamma_m-1)\sum_{i=\lceil m-\frac{m}{\gamma_m}\rceil}^{m-1}\frac{1}{i}=1$$

的最小正根. 对固定的 m, γ_m 可通过数值计算得到, 如 $\gamma_3=\frac{15}{11}$, $\gamma_4=\frac{11}{8}$. 可以证明, $\lim_{m\to\infty}\gamma_m$ 存在

且其极限 $\gamma_\infty \approx 1.4659$ 为方程 $\gamma_\infty \mathrm{e}^{\gamma_\infty} = -\frac{1}{\mathrm{e}^2}$ 的最小正根. 现已得到的 $Pm|\text{buffer}|C_{\max}$ 的 3 个在线算法的带参竞争比分别为 $(\gamma_m, \lceil(5-\gamma_m)m\rceil+1)$、$(\frac{3}{2}, \lceil\frac{(2+\ln 3)m}{1+\ln 3}\rceil+1)$ 和 $(1+\frac{\gamma_m}{2}, m)$ [63,64]. Englert 等 [64] 还证明了若缓冲区长度为任意有限常数, 则不存在竞争比小于 γ_m 的在线算法, 因此, γ_m 可视作一个临界值. 该问题的另两个带参下界为 $(\frac{3}{2}, \lfloor\frac{m}{2}\rfloor-1)$ 和 $(1+\frac{1}{\sqrt{2}}, \lfloor\frac{m}{8}\rfloor-1)$, 其中后者仅当 $m \geqslant 8$ 时成立. 对同类机、可中断和以机器最小负载为目标的缓冲模型半在线排序问题相关结果可参见文献 [6].

重排是另一类特殊加工机制, 允许已安排加工工件按一定规则改变加工机器. 文献中提出了多种对算法设计有利的重排规则, 这些规则可由允许重排的时机和对重排工件的限制类型来界定.

- 逐步重排 - 比例限制 (reassignSP) [65]: 每个工件到达时可重排若干个总加工时间不超过当前已到达工件加工时间 δ 倍的工件;
- 逐步重排 - 个数限制 (reassignSQ) [66]: 每个工件到达时可重排不超过 δ 个工件;
- 最后重排 - 个数限制 (reassignFA) [67]: 所有工件到达后可重排不超过 δ 个工件;
- 最后重排 - 机器限制 (reassignFE) [67]: 所有工件到达后可重排每台机器 (或一台机器 [68]) 上最晚到达的 δ 个工件.

对 $Pm|\text{reassignSP}|C_{\max}$, Sanders 等 [65] 给出了带参竞争比为 $(1+\varepsilon, 2^{O(\frac{1}{\varepsilon}\log^2\frac{1}{\varepsilon})})$ 的在线算法. 同时也针对部分参数设计了多个在线算法, 它们的带参竞争比分别为 $(\frac{3}{2}, \frac{4}{3})$、$(\frac{3}{2}-\frac{1}{2m}, 2)$ 和 $(\frac{4}{3}, \frac{5}{2})$. 特别地, 对 $m=2$, 存在带参竞争比为 $(\frac{7}{6}, 1)$ 的带参最好算法. 对 $Pm|\text{reassignSP}|C_{\min}$, Galvez 等 [69] 给出了带参竞争比分别为 $(\frac{4m-2}{3m-1}+O(\varepsilon)$、$O(\frac{1}{\varepsilon^3}\log\frac{1}{\varepsilon}))$ 和 $(\frac{17}{10}+\varepsilon, O(\frac{1}{\varepsilon}))$ 的在线算法; 同时证明了若 δ 为有限常数, 则不存在竞争比小于 $\frac{17}{16}$ 的在线算法.

对 $Pm|\text{reassignFA}|C_{\max}$, Albers 和 Hellwig [70] 设计了带参竞争比为 $(\gamma_m, (\lceil\frac{2-\gamma_m}{(\gamma_m-1)^2}\rceil+4)m)$ 的在线算法, 并证明了 $(\gamma_m, o(n))$ 是一个带参下界, 这里 n 为工件数. 尽管这里的 γ_m 值与缓冲模型中的临界值相同, 但两个模型之间是否存在更深刻的联系尚不可知 (参见文献 [71]). 文献 [70] 还给出了带参竞争比分别为 $(\frac{5}{3}, 4m)$ 和 $(\frac{7}{4}, \frac{5}{2}m)$ 的在线算法. 但目前仅在 $m=2$ 时, 得到了带参竞争比为 $(\frac{4}{3}, 1)$ 的带参最好算法 [67]. Englert 等 [71] 进一步将研究推广到 $Qm|\text{reassignFA}|C_{\max}$, 给出了带参竞争比为 $(\gamma+\frac{1}{3}, O(m))$ 的在线算法, 这里 γ 是 m 台机器速度的某个函数, $1<\gamma\leqslant\gamma_\infty$. 另一方面, 存在常数 $\gamma_0=\Theta(1)$, 若仅重排 $o(n)$ 个工件, 则不存在竞争比为 $\gamma_\infty+\gamma_0$ 的在线算法. 对任意 $\gamma\in(1,2)$, $(\gamma, \lfloor\frac{m}{c^2}\rfloor-1)$ 是一个带参下界, 其中 $c=\lceil\frac{-\ln(2-\gamma)}{\ln\gamma}\rceil\leqslant\sqrt{m}$. 其他重排规则下的相关结果可参见文献 [6].

从以上对特殊加工机制研究的介绍可以看到, 对多个模型, 确实可以得到竞争比任意接近于 1 或某个临界值的在线算法, 但这样的算法往往有很大的复杂度, 对系统也有很高的要求, 未必适应于现实需求. 另一方面, 针对某个固定的机器数或某个特定参数的在线算法仍非常缺乏. 寻找某些参数下的带参最好算法, 乃至定量分析参数对算法设计的影响, 是富有挑战性的工作.

2.4 有等级的在线和半在线排序问题

有等级的平行机在线和半在线排序问题近年来研究较为活跃. 在该问题中, 每个工件和每台机器均有各自的等级 (hierarchy), 一般设为一正整数. 工件只能在等级不大于其等级的机器上加工. 服务行业中服务窗口和服务对象的分类设定是其典型的应用场景, 理论上它可视作同型 (类) 机与不同类机之间的一种过渡状态. 有等级平行机是有加工权限的平行机的一种特殊情形, 但无疑最具代表性, 成果也最丰富.

机器等级设置对算法设计和分析有重要影响. 对 m 台同型机, 典型的机器设置有以下两种.

等级设置 I $(g=m)$: 机器和工件共有 m 个等级, 每台机器等级均不相同;

等级设置 II ($g=2$): 机器和工件共有两个等级, 分别有 k ($1\leqslant k<m$) 和 $m-k$ 台机器等级相同. 当 $m=2$ 时, 两种设置是等价的. 对两台同类机, 由于两台机器等级不同, 此时两台机器速度比 s 可为任意正数. 等级设置 I 的在线问题相关结果见表 1. 对等级设置 II, 存在竞争比不超过

$$1+\frac{m^2-m}{m^2-km+k^2}<\frac{7}{3}$$

的在线算法 [72]. 部分文献 [73,74] 考虑了下面的模型: 有两种类型的工件, 每种类型的工件有一个固定的机器子集可供其加工, 加工两种类型工件的两个机器子集交可能非空. 尽管该模型属有加工权限的平行机排序, 等级设置 II 仅是其一种特殊情形, 但现有研究显示, 等级设置 II 恰是其中最困难的情形. 以下将该模型称为等级设置 $\widetilde{\mathrm{II}}$.

有等级的平行机半在线研究的内容更加丰富. 例如, 对两台有等级同型机, 目标为最大完工时间. 若分别已知等级为 1 和等级为 2 的工件的总加工时间, 则最好算法竞争比为 $\frac{4}{3}$. 若仅已知等级为 1 的工件的总加工时间, 最好算法竞争比为 $\frac{3}{2}$. 若仅已知等级为 2 的工件的总加工时间, 最好算法竞争比为 $\frac{20}{13}$ [75,76]. 当目标为机器最小负载时, 若已知工件最大加工时间, 不存在竞争比为常数的在线算法. 但若同时已知该工件 (加工时间相同时以最先到达计) 等级为 1, 最好算法竞争比为 $1+\frac{\sqrt{2}}{2}$; 若同时已知该工件等级为 2, 最好算法竞争比为 2.4814 [77].

Chen 等 [78] 给出了 $P2|g=2,\mathrm{buffer}|C_{\max}$ 和 $P2|g=2,\mathrm{reassignFA}|C_{\max}$ 竞争比均为 $\frac{3}{2}$ 的最好算法. 问题 $Q2|g=2$, UB&LB$|C_{\max}$ 涉及两台机器速度比和工件加工时间上下界比两个参数, Lu 和 Liu [79] 给出了参数最好算法. 这是目前少有的双参数竞争比分析结果. 图 1 给出了两台同类机在线和半在线问题的参数最好算法竞争比图像, 以及与 UB&LB 有关的同型机和有等级同型机参数最好算法竞争比图像. 有等级同型机的其他半在线结果参见表 2 和文献 [6].

有等级同型机在线和半在线问题尚有不少难点问题留待解决. 一是在两种等级设置下, 尚未找到当 m 为某个大于 3 的固定数时严格大于 2 的下界; 二是除 $Pm|\mathrm{opt},g=m|C_{\max}$ 存在竞争比为 $2-\frac{1}{m}$ 的算法 [42] 外, 其他半在线模型在任意台机时结果较少; 三是现有算法设计思想较为单一, 也没有像经典平行机问题那样发展出多样化的分析手段.

在机器和工件有等级的情况下, 一些非常特殊的问题, 即使其离线最优值可以在多项式时间内求得, 其在线算法的设计与分析也不像经典平行机那样简单. 第一个例子是单位工件. 当目标函数为工件最大完工时间时, 对等级设置 $\widetilde{\mathrm{II}}$, Chen 等 [80]、Shabtay 和 Karhi [81] 分别给出了竞争比和渐近竞争比意义下的最好算法. 上述算法可移植至等级设置 II 的相应问题, 其竞争比和渐近竞争比分别为 $\frac{3}{2}$ 和 $\theta(k,m)$, 这里 $\theta(k,m)=\frac{m^2}{m^2-km+k^2}$. Luo 等 [82] 研究了等级设置 II 下的两个带参半在线问题: 预知 (lookahead) 和缓冲. 所谓预知是指可获得之后到达的 $\delta\in\mathbb{Z}$ 个工件的信息. 文献 [82] 分别给出了竞争比为 $(\theta(k,m),\frac{m^2-1}{m-k}-k)$ 和 $(\theta(k,m),m-\frac{m}{m^2-km+k^2})$ 的带参最好算法. 当 $m=2$ 时, 算法带参竞争比均可改进为 $(\frac{4}{3},1)$, 且为带参最好的. 当目标函数为工件总完工时间时, 对两台同型机, Hu 等 [83]、Shabtay 和 Karhi [84] 分别给出了竞争比为 $\frac{16}{13}$ 和渐近竞争比为 1.1573 的最好算法. 对两台同类机, Zhou 等 [85] 给出了参数下界和常数最好算法.

另一个例子是可分割加工. 可分割加工对问题研究带来的一种便利是在同类机环境下, 等级相同的不同机器可以合并为一台具有更快速度的机器. 因此满足等级设置 II 的同型或同类机可转化为两台有等级同类机, 机器具有更多等级的情形也是类似的. 对等级设置 I 的同型机, Bar-Noy 等 [23] 给出了机器数任意时竞争比为 e 的最好算法. Tan 和 Zhang [26] 及 Zhang [86] 通过建立数学规划给出了 m 固定时的最好算法及其竞争比. Chassid 和 Epstein [54]、Lu 和 Liu [87]、Liu 和 Zhan [88] 相继研究了二、三、四台同类机的相应问题, 并给出了参数最好算法及其竞争比.

3　复杂目标函数下的在线排序问题

3.1　工件可拒绝在线排序问题

企业在安排生产计划时, 一般会根据所拥有的资源和条件, 综合考虑成本和收益, 有选择地接受订单安排生产. 2000 年, Bartal 等 [89] 以此为背景提出了工件加工可拒绝排序问题 (scheduling with rejection). 每个工件有相应的加工时间和罚值, 工件可以安排加工, 也可以被拒绝加工, 目标为所有被拒绝工件的总罚值与加工工件的一个可行排序的某个数量指标之和. 常选的指标有工件最大完工时间和工件总完工时间等. 极少数文献, 如文献 [90], 将指标取为机器最小负载, 此时问题由极小化问题变为极大化问题, 罚值也应相应理解为拒绝工件时的收益. 可拒绝排序问题覆盖离线、实时在线和列表在线等多种模型, 涉及多个排序常用目标. 以下若不特别说明, 介绍的均为以工件最大完工时间与总罚值之和为目标的平行机可拒绝列表在线排序问题.

Bartal 等 [89] 对多台同型机在线问题进行了研究, 对 m 台机, 给出了竞争比为 $\frac{3m-2+\sqrt{5m^2-8m+4}}{2m}$ 的在线算法; 对两台机, 给出了更好的竞争比为 $\frac{1+\sqrt{5}}{2}$ 的在线算法. 另一方面, 他们还证明了对 m 台机, 问题下界为 β_m, 其中 β_m 是方程 $x^m - x^{m-1} - x^{m-2} - \cdots - 1 = 0$ 的根. 而对任意台机, 问题下界为 $\frac{3+\sqrt{5}}{2}$. 这两个下界对单位工件这一特殊情形仍成立. 综合两方面的结果, 两台机和 m 任意时的最好算法均已得到, 但 m 为大于 2 的某个数乃至 3 台机时的最好算法至今仍是公开问题. 作为解决这一问题的努力, Epstein 和 Zebedat-Haider [91] 研究了 3 台机单位工件这一特殊情形, 设计了竞争比恰为 β_3 的最好算法. Epstein 和 Zebedat-Haider [92] 针对已知工件按罚值非增或非减顺序到达两种不同假定, 以及工件总数 n 已知或未知两种不同假定, 提出了 4 个半在线模型, 在 m 台机单位工件情况下, 均给出了最好算法. 设工件加工可以中断, Seiden [93] 证明了对任意台机, 问题下界为 2.12457, 且设计了竞争比不超过 $\frac{4+\sqrt{10}}{3}$ 的在线算法. 当 m 固定时, 可证得一较小的竞争比, 但同样未得到 $m > 2$ 时的最好算法.

对两台同类机可中断情形, Dósa 和 He [56] 给出了竞争比为 $\frac{1}{2} + \sqrt{\frac{s+4}{4s}}$ 的参数最好算法. 而对不可中断情形, 现有算法 [56, 57] 在 $s > 1.3852$ 时是参数最好的 (参见图 1). Epstein 和 Zebedat-Haider [58] 对两台同类机、单位工件和可中断情形给出了竞争比为 $\frac{\sqrt{9s^2+10s+1}+s+1}{4s}$ 的参数最好算法. 若进一步已知工件按罚值非增顺序到达, 则参数最好算法的竞争比可减小为 $\sqrt{\frac{s+1}{s}}$. 但若已知工件按罚值非减顺序到达, 参数最好算法竞争比不会减小.

一些文献研究了具有特殊加工机制的可拒绝排序问题, 并将关注点落在拒绝决策上. 现有可拒绝排序问题研究显示, 多数情况下, 工件拒绝与否的决策对算法性能的影响大于加工工件在机器上的安排所带来的影响. 这部分研究也证实了, 采用有限程度的特殊机制即可得到竞争比等于相应不可拒绝在线问题最好算法竞争比的在线算法, 也即抵消了可拒绝对算法设计带来的困难. Epstein 和 Zebedat-Haider [94] 提出了拒绝可撤回模型. 在所有工件加工完毕后, 可重新加工 $\delta \in \mathbb{Z}$ 个曾被拒绝的工件. 对两台同类机, 只需撤回至多 1 个工件, 即可得到竞争比为 $r_{Q2}(s)$ 的算法. 由于此即为 $Q2||C_{\max}$ 最好算法的竞争比, 因此即使允许撤回更多工件也不可能减小竞争比, 该算法也是该半在线问题的带参最好算法. 对 m 台同型机, 撤回至多 $m-1$ 个工件, 可得到竞争比为 $2-\frac{1}{m}$ 的算法. 尽管竞争比已与不可拒绝时的 LS 算法相同, 但尚未找到 $m \geqslant 4$ 时的带参最好算法. 撤回的作用更明显地体现在目标为机器最小负载与被拒绝工件总罚值之和的可拒绝排序问题上. 无论是 m 台同型机还是两台同类机, 如果不允许撤回, 则任意算法的竞争比不为有限常数. 但如果允许撤回, 即使只撤回至多一个工件, 也可得到与不可拒绝时最好算法相同竞争比的在线算法. Min 等 [95] 则提出了与之相对的模型, 即在所

有工件加工完毕后, 允许重新拒绝若干曾被接受的工件. 对两台同型机, 只需重新拒绝至多一个工件, 即可得到竞争比为 $\frac{3}{2}$ 的最好算法. 他们还证明了只需利用一个容量为 1 的缓冲区, 即可得到两台同型机竞争比为 $\frac{3}{2}$ 的最好算法. Epstein 和 Zebedat-Haider [90] 则研究了 m 台同型机单位工件的相应问题, 只需容量为 $m-1$ 的缓冲区, 无论目标为工件最大完工时间与被拒绝工件总罚值之和, 还是机器最小负载与被拒绝工件总罚值之和, 均能给出竞争比为 1 的最优算法, 但容量更小的缓冲区则无法做到这一点. Min 等 [96] 考虑了并行加工的可拒绝排序问题. 对两台同类机, 只需两个子过程并行即可得到竞争比为 $r_{Q2}(s)$ 的最好算法, 并且这两个子过程仅在拒绝策略上有所不同, 加工工件均采用 LS 算法安排加工.

Min 等 [97] 研究了两台有等级同类机可中断可拒绝在线排序, 给出的算法当 $s \in (0,1] \cup [\frac{1+\sqrt{5}}{2}, \infty)$ 时是参数最好的.

3.2 机器可增加的在线排序问题

在经典在线排序问题中, 机器数量均为事先给定的常数, 在安排和加工过程中不发生变化. 而更符合实际的情况是生产资料随着生产任务的增多逐渐添置. 为此, Imreh 和 Noga [98] 于 1999 年提出了机器可增加的排序问题. 机器数量从零开始逐渐增加, 且不设上限. 购买 i 台机器的费用为 c_i, 也即购买第 i 台机器的费用为 $c_i - c_{i-1}$, 其中 $c_0 = 0$. 目标函数为工件最大完工时间和购买机器费用的和. 此类问题也可称为机器有费用排序问题 (scheduling with machine cost). 在 Imreh 和 Noga 最初提出的模型中, $c_i = i$. 他们利用以若干个加工时间任意小的工件为工件集的实例, 证明了该问题下界至少为 $\frac{4}{3}$. 事实上, 该实例也适用于可中断、已知工件最大加工时间等多个问题, 并且对该模型的一些特殊情况或半在线问题, 如所有工件加工时间不超过单台机器费用, 已知工件按加工时间从大到小顺序到达, $\frac{4}{3}$ 确为最好算法竞争比. Dósa 和 Tan [99] 将一般情形的下界改进为 $\sqrt{2}$, 并证明了若仅采用

$$\begin{cases} p_{(1)} + \dfrac{P}{p_{(1)}}, & \text{若 } p_{(1)} \geqslant \sqrt{P}, \\ 2\sqrt{P}, & \text{其他} \end{cases}$$

作为最优值的估计, 则无法证得一算法的竞争比小于 $\frac{3}{2}$. 目前该问题最佳算法竞争比为 $\frac{2+\sqrt{7}}{3} \approx 1.5486$ (参见文献 [99]). 该问题可中断和部分半在线模型的研究结果参见表 3.

Imreh [104] 首先将该问题推广至任意机器费用的情形, 但仅给出了所有工件加工时间不超过任意单台机器费用这一特殊情况下竞争比为 2 的最好算法. 最近, Akaria 和 Epstein [105] 给出了一般情形下具有相同竞争比的最好算法. Jiang 等 [106] 研究了 c_i 为凹函数的特殊情形, 相关结果见表 3.

在 $c_i = i$ 的假设下, Csirik 等 [107] 将目标改为各台机器负载的平方和与机器数之和, 并给出了竞争比为 $\frac{4}{3}$ 的最好算法. Nagy-György 和 Imreh [108] 研究了工件可拒绝加工的机器可增加排序问题, 给出了竞争比为 $\frac{3+\sqrt{5}}{2}$ 的在线算法. Dósa 和 He [109] 考虑了上述问题所有工件加工时间小于 1 的特殊情形, 给出了竞争比为 2 的最好算法.

3.3 目标为机器负载的 L^p 范数的在线排序问题

设 $p > 1$, 称 $(\sum_{i=1}^{m} L_i^p(\sigma))^{\frac{1}{p}}$ 为机器负载的 L^p 范数. 易见工件最大完工时间即为 L^∞ 范数. 一般认为, p 较小时, L^p 范数在体现各台机器负载均衡性方面优于工件最大完工时间、数据存储和能量消耗等一些特定应用的目标函数, 也表现为 L^2 或其他范数 (参见文献 [110,111]). 贪婪 (greedy) 算法在目标为 L^p 范数的在线排序中有极其重要的地位. 在 L^p $(p < \infty)$ 范数在线排序中, 每安排一个工件,

表 3 机器可增加在线和半在线排序问题相关结果

		不可中断		可中断	
		下界	竞争比	下界	竞争比
$c_i = i$	在线	$\sqrt{2}$ [99]	$\frac{2+\sqrt{7}}{3}$ [99]	$\frac{4}{3}$ [98]	$\frac{2+2\sqrt{6}}{5}$ [100]
	在线, $p_j \leqslant 1$	$\frac{4}{3}$ [98]	$\frac{4}{3}$ [101]	$\frac{4}{3}$ [98]	$\frac{4}{3}$ [101]
	已知工件总加工时间	$\frac{6}{5}$ [102]	$\sqrt{2}$ [103]	1.0957 [102]	$\frac{5}{4}$ [102]
	已知工件最大加工时间	$\frac{4}{3}$ [98]	$\frac{3}{2}$ [101]	$\frac{4}{3}$ [98]	$\frac{4}{3}$ [100]
	已知工件加工时间递减	$\frac{4}{3}$ [98]	$\frac{4}{3}$ [100]	$\frac{4}{3}$ [98]	$\frac{4}{3}$ [100]
$c_i \geqslant c_{i-1}$	在线	2 [104]	2 [105]	2 [104]	2 [105]
	在线, $p_j \leqslant \min_i\{c_i - c_{i-1}\}$	2 [104]	2 [104]		
c_i 为凹函数	在线	$\frac{3}{2}$ [106]	$\frac{9+\sqrt{17}}{8}$ [106]	$\frac{3}{2}$ [106]	1.5654 [106]

目标值均会增加. 所谓贪婪算法是指将工件安排在可使目标值增加最少的机器上加工. 对同型机, 贪婪算法即为 LS 算法.

对 $Pm||L^p$, Avidor 等 [112] 证明了贪婪算法的竞争比为

$$\sup_{x\geqslant 1}\max_{l=1,\dots,m}\left(\frac{(m-l)(1+x)^p+l}{(m-l)x^p+l^{1-p}m^p}\right)^{\frac{1}{p}}.$$

按上式计算可得, 对任意机器数, 竞争比为 $2-\Theta(\frac{\ln p}{p})$. 将上述结论运用于两种特殊情况, 并通过给出问题下界, 可得到贪婪算法是求解 $P2||L^p$ 的竞争比为 $\sup_{x\geqslant 0}(\frac{1+(1+x)^p}{2^p+x^p})^{\frac{1}{p}}$ 的最好算法, 也是求解 $Pm||L^2$ ($m=2,3,4$) 时的最好算法, 竞争比分别为 $\frac{\sqrt{3+\sqrt{5}}}{2}$、$\frac{2\sqrt{3}}{3}$ 和 $\frac{\sqrt{3+\sqrt{5}}}{2}$. 从上述结果可以看到, $Pm||L^2$ 最好算法的竞争比并不是 m 的非减函数. 类似情况在其他排序问题中较为少见. 对 $m>4$, 可求得 $Pm||L^2$ 的一个下界为 $\sqrt{\sqrt{5}-1}$ [112]. 对 $Q||L^p$, Im 等 [113] 给出了一个竞争比为不依赖于 p 的有限常数的在线算法.

对 $R||L^p$ 的研究始于文献 [114]. Caragiannis [115] 最终证明了贪婪算法是求解 $R||L^p$ 竞争比为 $\frac{1}{2^{\frac{1}{p}}-1}$ 的最好算法, 其中下界对有加工权限的同型机仍成立. 文献 [116,117] 研究了有加工权限的平行机、单位工件和目标为 L^2 范数的在线排序问题. 其研究的初衷源自排序博弈, 显示了不同排序分支密切联系的一面. 对任意台有加工权限同类机, 贪婪算法竞争比恰为 $\frac{\sqrt{51}}{3}$. 对任意台有加工权限同型机, 贪婪算法竞争比至多为 $\sqrt{\frac{2}{3}\sqrt{21}+1}$, 至少为 2.

当 $m=2$ 时, Du 等 [118] 和 Shuai 等 [119] 相继给出了同型机和同类机可中断问题的最好在线算法, 其竞争比为二元参数方程的根. Lin 等 [120] 证明了贪婪算法是 $P2|\mathrm{decr}|L^p$ 的最好算法, 其竞争比为 $\max_{1\leqslant x\leqslant 2}(\frac{(x+2)^p+(x+1)^p}{(2x)^p+3^p})^{\frac{1}{p}}$. Qi 和 Yuan [121,122] 给出了两台有等级同型机 4 个半在线问题的最好算法. 若同时已知等级为 1 的工件和等级为 2 的工件的总加工时间, 则其竞争比为 $(\frac{2^p+4^p}{2\cdot 3^p})^{\frac{1}{p}}$. 对已知工件总加工时间、缓冲和最后重排 - 个数限制 3 个半在线模型, 其竞争比均为 $\max_{\frac{1}{2}\leqslant x\leqslant 1}(\frac{(1+x)^p+(1-x)^p}{2^p(x^p+(1-x)^p)})^{\frac{1}{p}}$.

4 在线排序问题的新研究

在线排序研究从 20 世纪 80 年代发端, 吸引了众多学者的兴趣, 产生了大量的成果, 在排序研究中占有重要的地位. 进入 21 世纪以来, 尽管又出现了排序博弈这样新的分支, 但在线排序研究方兴未艾, 并不断有新的理念、问题和技术出现. 以下列举部分近年来在线排序领域出现的新课题.

4.1 竞争比近似方案

2013 年, Gunther 等 [123] 在对实时在线排序问题的研究中, 提出了竞争比近似方案 (competitive ratio approximate scheme, CRAS) 的概念. 在线算法族 $\{A_\varepsilon\}$ 称为在线问题的竞争比近似方案, 若对任意的 $\varepsilon > 0$, 算法 A_ε 的竞争比至多为 $(1+\varepsilon)\rho^*$, 其中 ρ^* 为该在线问题最好算法的竞争比. 随即 Chen 等 [124]、Megow 和 Wiese [125] 独立给出了列表在线 $Pm||C_{\max}$ 的 CRAS, 并分别推广至 $Rm||C_{\max}$ 和 $Qm||C_{\max}$ 等问题.

从定义来看, 在线问题的 CRAS 与离线问题的多项式时间近似方案类似, 在最好在线算法尚未获得的情况下, 能得到竞争比与之任意接近的在线算法, 不失为一种好的选择. 但这些近似方案的时间复杂性普遍较高, 例如, $Pm||C_{\max}$ 的一个 CRAS 时间复杂度为 $m^{2^{O(\frac{1}{\varepsilon^2}\log^2\frac{1}{\varepsilon})}}$ (参见文献 [124]). 目前, 尚未有一个按 CRAS 的描述实现的可应用于实际的在线算法, 也没有据此得到一个最好算法未知的在线问题的最好算法竞争比的值. 因此, CRAS 为解决困难在线问题提供了新的思路, 但并未减少传统在线算法设计与分析方法的必要性.

4.2 带建议的在线算法

带建议的在线算法 (online algorithm with advice) 在 2010 年左右出现并逐渐发展. 所谓带建议是指在在线算法运行前或执行中, 可接受除实例信息外的额外信息. 建议的类型多种多样, 可以是正常情况下尚未披露的实例信息, 甚至是实例的最优解或最优值等. 建议可以多种形式提供给算法, 一般将建议的内容用二进制编码并以此长度作为建议复杂度的度量. 关于带建议的在线算法的一般介绍可参见文献 [126].

带建议的算法与半在线模型有相似之处. 很多半在线模型预知了实例的某些信息, 这些额外信息均可视作建议在算法执行前读取. 一些半在线模型并不提供额外信息, 如并行模型, 它与经典在线模型的区别在于可并行运行多个子过程, 并以它们产生的多个排序中的最佳者作为算法输出. 这一过程也可用带建议的在线算法实现. 此时只需将产生最佳排序的子过程的编号作为建议提供给算法, 算法仅运行该子过程即可获得相同的效果. 从文献 [61] 的结论可知, 对 $Pm||C_{\max}$ 的每个实例, 用长度为 $O(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon}\log\frac{m}{\varepsilon})$ 字节的建议可从 $(\frac{m}{\varepsilon})^{O(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon})}$ 个子过程中筛选出一个, 并以此加工工件, 即可得到带建议的竞争比为 $1+\varepsilon$ 的在线算法. 另有一些半在线模型, 如缓冲, 目前还未找到用带建议的算法实现的途径 (参见文献 [126]). 当然, 也有一些建议无法在现有半在线框架下解释.

带建议的在线算法提供了定量地探究算法性能与额外信息量之间关系的可能. 在半在线研究中, 也有鉴别何种信息有价值、多种信息的价值是否优于一种信息的构想 (参见文献 [6]), 但尚未对信息进行量化. 事实上, 如果不限制信息量的多少, 则可将最优解编码后作为建议提供给算法, 从而使问题归于平凡. 从另一角度来看, 单纯以信息量的多少而不是信息的内容作为标准, 使一些带建议的算法并不像半在线模型那样有切实的现实背景和应用价值.

对 $P||L^p$, Renault 等 [127] 给出了安排每个工件时仅需长度为 $O(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon})$ 的建议的竞争比 $1+\varepsilon$ 的在线算法. 而在 Boyar 等 [128] 设计的算法中, 为达到上述竞争比, 在算法运行过程中只需读取总长为 $O(\frac{1}{\varepsilon}\log^2 n)$ 的建议. 对 $Rm||C_{\max}$ 或 $Rm||L^p$, 建议的长度需增加到 $O(\frac{1}{\varepsilon^m}\log^{m+1} n)$. 另一方面, Renault 等 [127] 证明了对 $Pm||C_{\max}$、$Pm||C_{\min}$ 或 $Pm||L^p$, 为得到竞争比为 1 的算法, 任意算法至少需要总长为 $(n-2m)\log m$ 的建议.

4.3 多样化算法性能指标

竞争比是评判在线算法性能的主流指标, 但也存在一些明显的缺点. 例如, 相比较的两方, 在线算法与最优算法, 在实现环境上并不对等, 竞争比作为最坏情况度量无法反映算法的总体性能等. 为此, 不断有学者从不同角度出发, 提出了多样化的评判指标 (如文献 [129, 130]). 当然, 这些指标同样也都存在某些不足, 因而尚无法取代竞争比的地位, 但作为深入分析、客观比较算法性能的工具不乏其价值. 以下对部分与在线排序问题相关的结果择要予以介绍.

对列表在线排序问题, 算法给出的排序与工件安排的顺序有关. 设极小化问题实例 I 的工件集为 $\mathcal{J}=\{J_1,\ldots,J_n\}$, J_j 是算法第 j 个安排的工件. 记 S_n 为所有 n 阶置换的全体, 对任意 $\pi\in S_n$, 算法按 $J_{\pi(1)},\ldots,J_{\pi(n)}$ 的顺序安排工件的相应实例记为 $\pi(I)$. 由于实例最优解一般与工件顺序无关, 故 $C^*(\pi(I))=C^*(I)$.

随机序比 (random order ratio) 由 Kenyon [131] 在研究装箱问题时提出. 算法 A 的随机序比定义为

$$\sup_I \frac{1}{n!}\sum_{\pi\in S_n}\frac{C^A(\pi(I))}{C^*(I)}.$$

随机序比可视作一种平均情况分析. Osborn 和 Torng [132] 证明了 LS 算法求解 $Pm||C_{\max}$ 的随机序比仍将随 m 的增大而趋于 2.

相对最坏序比 (relative worst order ratio) 可用于比较两个在线算法的性能. 算法 A 对实例 I 的最坏值定义为 $\max_{\pi\in S_n}C^A(\pi(I))$. 对同一个实例, 两个算法的最坏值对应的置换 π 可能不同. 若对任意实例 I, $\max_{\pi\in S_n}C^A(\pi(I))\geqslant\max_{\pi\in S_n}C^B(\pi(I))$, 则称 A 和 B 可比, 此时

$$\sup_I\frac{\max_{\pi\in S_n}C^A(\pi(I))}{\max_{\pi\in S_n}C^B(\pi(I))}$$

称为 A 对 B 的相对最坏序比. 当然并不是任意一对算法都是可比的. 对 $Q2||C_{\max}$, 当 $s>\frac{1+\sqrt{5}}{2}$ 时, LS 算法的竞争比为 $\frac{s+1}{s}$, 但将所有工件安排在速度为 s 的机器上加工所得算法 Fast 的竞争比也为 $\frac{s+1}{s}$. 两个算法从竞争比角度并无差别, 但可以证明这两个算法可比且 Fast 对 LS 的相对最坏序比为 $\frac{s+1}{s}$ [133]. 因而 LS 算法优于 Fast 算法, 这是符合直观认识的.

在线有限比 (online-bounded ratio) 由 Boyar 等 [134] 提出, 它将竞争比定义中的最优解作了限制. 对实例 I 的任意可行排序 σ, 记 σ_j 为在 σ 中删去工件 $J_{j+1},\ldots,J_n$ 后所得的部分排序. 某排序或部分排序 σ 的目标值记为 $f(\sigma)$. 称 I 的一个可行排序 $\widetilde{\sigma}^A$ 为相应于算法 A 给出的排序 σ^A 的有限最优解, 若

$$f(\widetilde{\sigma}^A)=\min\{f(\sigma)\mid f(\sigma_j)\leqslant f(\sigma_j^A), j=1,\ldots,n\}.$$

算法 A 的在线有限比定义为 $\sup_I\frac{f(\sigma^A)}{f(\widetilde{\sigma}^A)}$. 有限最优解对原实例而言未必是最优的, 因此, 算法的在线有限比通常会小于竞争比. Boyar 等 [134] 证明了对 $Pm||C_{\max}$, LS 算法的在线有限比为 $2-\frac{1}{m-1}$, 且对 $m\geqslant 3$ 不存在在线有限比小于 $\frac{4}{3}$ 的在线算法. 对 $Q2||C_{\max}$ 和 $Pm||C_{\min}$, LS 算法的在线有限比均为 1.

5 结语

本文概述了在线和半在线排序的基本概念, 回顾了以最大完工时间为目标的在线和半在线排序问题的新进展, 总结了复杂目标函数在线排序的有关结果, 介绍了若干在线排序研究的新课题. 在之前各

节中, 已列出了一些具体的公开问题. 下面再给出若干各领域在线排序中的共性问题作为本文的结尾.

(1) 当前最优值的估计似乎成为很多在线问题研究的瓶颈所在. 如何构造新的可行的最优值估计, 能否证明更多的伪下界, 以及如何突破伪下界的限制证明算法的竞争比?

(2) 已有一些利用计算机辅助给出在线问题下界的成功案例. 计算机在下界乃至算法竞争比的证明中能否产生更大的作用, 计算机计算能力的提高和人工智能技术的发展将在多大程度上对在线问题研究产生积极的影响?

(3) 很多未解决的在线问题从描述或数学结构上来看并不复杂, 其突破有待引入何种数学工具? 目前很多问题算法竞争比的证明是通过复杂而又细致的分析得到的, 是否存在变革性的方法和技术能有效而又简明地加以解决?

致谢 衷心感谢审稿人对初稿提出的宝贵意见.

参考文献

1 Sleator D D, Tarjan R E. Amortized efficiency of list update and paging rules. Commun ACM, 1985, 28: 202–208
2 Borodin A, El-Yaniv R. Online Computation and Competitive Analysis. Cambridge: Cambridge University Press, 2005
3 Fiat A, Woeginger G J. Online Algorithms: The State of the Art. Heidelberg: Springer, 1998
4 Sgall J. On-line scheduling. In: Online Algorithms: The State of the Art. Lecture Notes in Computer Science, vol. 1442. Heidelberg: Springer, 1998, 196–231
5 Pruhs K, Sgall J, Torng E. Online scheduling. In: Handbook of Scheduling: Algorithms, Models, and Performance Analysis. Florida: CRC Press, 2004, Chapter 15, 1–44
6 Tan Z, Zhang A. Online and semi-online scheduling. In: Handbook of combinatorial optimization. Heidelberg: Springer, 2013, 2191–2252
7 Leung J Y T, Li C L. Scheduling with processing set restrictions: A survey. Int J Prod Econ, 2008, 116: 251–262
8 Leung J Y T, Li C L. Scheduling with processing set restrictions: A literature update. Int J Prod Econ, 2016, 175: 1–11
9 Lee K, Leung J Y T, Pinedo M L. Makespan minimization in online scheduling with machine eligibility. Ann Oper Res, 2013, 204: 189–222
10 Graham R L, Lawler E L, Lenstra J K, et al. Optimization and approximation in deterministic sequencing and scheduling: A survey. Ann Discrete Math, 1979, 5: 287–326
11 Graham R L. Bounds for certain multiprocessing anomalies. Bell Syst Tech J, 1966, 45: 1563–1581
12 Faigle U, Kern W, Turan G. On the performance of on-line algorithms for partition problems. Acta Cybernet, 1989, 9: 107–119
13 Gormley T, Reingold N, Torng E, et al. Generating adversaries for request-answer games. In: Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms. Philadelphia: SIAM, 2000, 564–565
14 Fleischer R, Wahl M. On-line scheduling revisited. J Sched, 2000, 3: 343–353
15 Albers S. On randomized online scheduling. In: Proceedings of the 34th Annual ACM Symposium on Theory of Computing. New York: Association for Computing Machinery, 2002, 134–143
16 Tan Z, Li R. Pseudo lower bounds for online parallel machine scheduling. Oper Res Lett, 2015, 43: 489–494
17 Chen B, van Vliet A, Woeginger G J. New lower and upper bounds for on-line scheduling. Oper Res Lett, 1994, 16: 221–230
18 Rudin III J F, Chandrasekaran R. Improved bounds for the online scheduling problem. SIAM J Comput, 2003, 32: 717–735
19 Ebenlendr T, Sgall J. A lower bound on deterministic online algorithms for scheduling on related machines without preemption. Theory Comput Syst, 2015, 56: 73–81
20 Cho Y, Sahni S. Bounds for list schedules on uniform processors. SIAM J Comput, 1980, 9: 91–103
21 Jeż L, Schwartz J, Sgall J, et al. Lower bounds for online makespan minimization on a small number of related machines. J Sched, 2013, 16: 539–547
22 Berman P, Charikar M, Karpinski M. On-line load balancing for related machines. J Algorithms, 2000, 35: 108–121
23 Bar-Noy A, Freund A, Naor J. On-line load balancing in a hierarchical server topology. SIAM J Comput, 2001, 31: 527–549

24 Park J, Chang S Y, Lee K. Online and semi-online scheduling of two machines under a grade of service provision. Oper Res Lett, 2006, 34: 692–696
25 Jiang Y, He Y, Tang C. Optimal online algorithms for scheduling on two identical machines under a grade of service. J Zhejiang Univ-Sci A, 2006, 7: 309–314
26 Tan Z, Zhang A. Online hierarchical scheduling: An approach using mathematical programming. Theor Comput Sci, 2011, 412: 246–256
27 Lim K, Lee K, Chang S Y. Improved bounds for online scheduling with eligibility constraints. Theor Comput Sci, 2011, 412: 5211–5224
28 Epstein L, Noga J, Seiden S, et al. Randomized on-line scheduling on two uniform machines. J Sched, 2001, 4: 71–92
29 Dolgui A, Kotov V, Nekrashevich A, et al. General parametric scheme for the online uniform machine scheduling problem with two different speeds. Inform Process Lett, 2018, 134: 18–23
30 Kellerer H, Kotov V, Gabay M. An efficient algorithm for semi-online multiprocessor scheduling with given total processing time. J Sched, 2015, 18: 623–630
31 Albers S, Hellwig M. Semi-online scheduling revisited. Theor Comput Sci, 2012, 443: 1–9
32 Kellerer H, Kotov V. An efficient algorithm for bin stretching. Oper Res Lett, 2013, 41: 343–346
33 Gabay M, Kotov V, Brauner N. Online bin stretching with bunch techniques. Theor Comput Sci, 2015, 602: 103–113
34 Böhm M, Sgall J, van Stee R, et al. A two-phase algorithm for bin stretching with stretching factor 1.5. J Comb Optim, 2017, 34: 810–828
35 Cheng T C E, Kellerer H, Kotov V. Algorithms better than LPT for semi-online scheduling with decreasing processing times. Oper Res Lett, 2012, 40: 349–352
36 Angelelli E, Speranza M G, Tuza Z. Semi on-line scheduling on three processors with known sum of the tasks. J Sched, 2007, 10: 263–269
37 Lee K, Lim K. Semi-online scheduling problems on a small number of machines. J Sched, 2013, 16: 461–477
38 Böhm M, Sgall J, van Stee R, et al. Online bin stretching with three bins. J Sched, 2017, 20: 601–621
39 Seiden S, Sgall J, Woeginger G. Semi-online scheduling with decreasing job sizes. Oper Res Lett, 2000, 27: 215–221
40 Gabay M, Brauner N, Kotov V. Improved lower bounds for the online bin stretching problem. 4OR, 2017, 15: 183–199
41 Azar Y, Regev O. On-line bin-stretching. Theor Comput Sci, 2001, 268: 17–41
42 Lee K, Hwang H C, Lim K. Semi-online scheduling with GoS eligibility constraints. Int J Prod Econ, 2014, 153: 204–214
43 Dósa G, Fügenschuh A, Tan Z, et al. Tight upper bounds for semi-online scheduling on two uniform machines with known optimum. CEJOR Cent Eur J Oper Res, 2018, 26: 161–180
44 Dósa G, Fügenschuh A, Tan Z, et al. Tight lower bounds for semi-online scheduling on two uniform machines with known optimum. CEJOR Cent Eur J Oper Res, 2019, 27: 1107–1130
45 Zhang A, Jiang Y, Fan L, et al. Optimal online algorithms on two hierarchical machines with tightly-grouped processing times. J Comb Optim, 2015, 29: 781–795
46 He Y, Zhang G. Semi on-line scheduling on two identical machines. Computing, 1999, 62: 179–187
47 Liu M, Chu C, Xu Y, et al. Semi-online scheduling on 2 machines under a grade of service provision with bounded processing times. J Comb Optim, 2011, 21: 138–149
48 Luo T, Xu Y. Optimal algorithm for semi-online scheduling on two machines under GoS levels. Optim Lett, 2016, 10: 207–213
49 Cao Q, Wan G. Semi-online scheduling with combined information on two identical machines in parallel. J Comb Optim, 2016, 31: 686–695
50 Luo T, Xu Y. Semi-online hierarchical load balancing problem with bounded processing times. Theor Comput Sci, 2015, 607: 75–82
51 He Y. The optimal on-line parallel machine scheduling. Comput Math Appl, 2000, 39: 117–121
52 Tan Z, Zhang A. A note on hierarchical scheduling on two uniform machines. J Comb Optim, 2010, 20: 85–95
53 Dósa G, Epstein L. Preemptive scheduling on a small number of hierarchical machines. Inform and Comput, 2008, 206: 602–619
54 Chassid O, Epstein L. The hierarchical model for load balancing on two machines. J Comb Optim, 2008, 15: 305–314
55 Lu X, Liu Z. Semi-online scheduling problems on two uniform machines under a grade of service provision. Theor Comput Sci, 2013, 489: 58–66
56 Dósa G, He Y. Preemptive and non-preemptive on-line algorithms for scheduling with rejection on two uniform machines. Computing, 2006, 76: 149–164
57 He Y, Min X. On-line uniform machine scheduling with rejection. Computing, 2000, 65: 1–12
58 Epstein L, Zebedat-Haider H. Preemptive online scheduling with rejection of unit jobs on two uniformly related

machines. J Sched, 2014, 17: 87–93

59 Cao Q, Liu Z. Semi-online scheduling with bounded job sizes on two uniform machines. Theor Comput Sci, 2016, 652: 1–17

60 Kellerer H, Kotov V, Speranza M G, et al. Semi on-line algorithms for the partition problem. Oper Res Lett, 1997, 21: 235–242

61 Albers S, Hellwig M. Online makespan minimization with parallel schedules. Algorithmica, 2017, 78: 492–520

62 Zhang G. A simple semi on-line algorithm for $P2||C_{\max}$ with a buffer. Inform Process Lett, 1997, 61: 145–148

63 Sun H, Fan R. Improved semi-online makespan scheduling with a reordering buffer. Inform Process Lett, 2013, 113: 434–439

64 Englert M, Özmen D, Westermann M. The power of reordering for online minimum makespan scheduling. SIAM J Comput, 2014, 43: 1220–1237

65 Sanders P, Sivadasan N, Skutella M. Online scheduling with bounded migration. Math Oper Res, 2009, 34: 481–498

66 Dósa G, Wang Y, Han X, et al. Online scheduling with rearrangement on two related machines. Theor Comput Sci, 2011, 412: 642–653

67 Tan Z, Yu S. Online scheduling with reassignment. Oper Res Lett, 2008, 36: 250–254

68 Min X, Liu J, Wang Y. Optimal semi-online algorithms for scheduling problems with reassignment on two identical machines. Inform Process Lett, 2011, 111: 423–428

69 Galvez W, Soto J, Verschae J. Symmetry exploitation for online machine covering with bounded migration. In: Proceeding of the 26th Annual European Symposium on Algorithms, Leibniz International Proceedings in Informatics, vol. 112. Philadelphia: SIAM, 2018, Article No. 32

70 Albers S, Hellwig M. On the value of job migration in online makespan minimization. Algorithmica, 2017, 79: 598–623

71 Englert M, Mezlaf D, Westermann M. Online makespan scheduling with job migration on uniform machines. In: Proceeding of the 26th Annual European Symposium on Algorithms. Leibniz International Proceeding of Informatics, vol. 112. Wadern: Dagstuhl Publishing, 2018, Article No. 26

72 Zhang A, Jiang Y, Tan Z. Online parallel machines scheduling with two hierarchies. Theor Comput Sci, 2009, 410: 3597–3605

73 Karhi S, Shabtay D. Online scheduling of two job types on a set of multipurpose machines. Int J Prod Econ, 2014, 150: 155–162

74 Karhi S. Semi-online scheduling of two job types on a set of multipurpose machines. J Oper Res Soc, 2018, 69: 1445–1455

75 Luo T, Xu Y. Semi-online scheduling on two machines with GoS levels and partial information of processing time. Sci World J, 2014, 2014: 576234

76 Chen X, Ding N, Dósa G, et al. Online hierarchical scheduling on two machines with known total size of low-hierarchy jobs. Int J Comput Math, 2015, 92: 873–881

77 Wu Y, Cheng T C E, Ji M. Optimal algorithms for semi-online machine covering on two hierarchical machines. Theor Comput Sci, 2014, 531: 37–46

78 Chen X, Xu Z, Dósa G, et al. Semi-online hierarchical scheduling problems with buffer or rearrangements. Inform Process Lett, 2013, 113: 127–131

79 Lu X, Liu Z. Online hierarchical scheduling on two uniform machines with bounded job sizes. Asia-Pac J Oper Res, 2015, 32: 1550032

80 Chen Y, Chen G T, Liu L C, et al. Online scheduling with unit processing times and processing set restrictions. J Oper Res Soc China, 2019, 7: 475–484

81 Shabtay D, Karhi S. Online scheduling of two job types on a set of multipurpose machines with unit processing times. Comput Oper Res, 2012, 39: 405–412

82 Luo T, Xu Y, Luo L, et al. Semi-online scheduling with two GoS levels and unit processing time. Theor Comput Sci, 2014, 521: 62–72

83 Hu J, Jiang Y, Zhou P, et al. Total completion time minimization in online hierarchical scheduling of unit-size jobs. J Comb Optim, 2017, 33: 866–881

84 Shabtay D, Karhi S. An asymptotically optimal online algorithm to minimize the total completion time on two multipurpose machines with unit processing times. Discrete Optim, 2012, 9: 241–248

85 Zhou H, Jiang Y, Zhou P, et al. Total completion time minimization scheduling on two hierarchical uniform machines. Theor Comput Sci, 2017, 702: 65–76

86 Zhang A. On the optimality of the LP-based algorithm for online scheduling with GoS eligibility constraints. Oper Res Lett, 2015, 43: 522–525

87 Lu X, Liu Z. An optimal online algorithm for fractional scheduling on uniform machines with three hierarchies. J

Syst Sci Complex, 2016, 29: 1650–1657
88 Liu Z, Zhan J. Online fractional hierarchical scheduling on uniformly related machines. Comput Oper Res, 2020, 113: 104778
89 Bartal Y, Leonardi S, Marchetti-Spaccamela A, et al. Multiprocessor scheduling with rejection. SIAM J Discrete Math, 2000, 13: 64–78
90 Epstein L, Zebedat-Haider H. Online scheduling with rejection and reordering: Exact algorithms for unit size jobs. J Comb Optim, 2014, 28: 875–892
91 Epstein L, Zebedat-Haider H. Online scheduling of unit jobs on three machines with rejection: A tight result. Inform Process Lett, 2016, 116: 252–255
92 Epstein L, Zebedat-Haider H. Rent or buy problems with a fixed time horizon. Theory Comput Syst, 2015, 56: 309–329
93 Seiden S S. Preemptive multiprocessor scheduling with rejection. Theor Comput Sci, 2001, 262: 437–458
94 Epstein L, Zebedat-Haider H. Online scheduling with rejection and withdrawal. Theor Comput Sci, 2011, 412: 6666–6674
95 Min X, Wang Y, Liu J, et al. Semi-online scheduling on two identical machines with rejection. J Comb Optim, 2013, 26: 472–479
96 Min X, Liu J, Wang Y. Optimal semi-online algorithm for scheduling with rejection on two uniform machines. J Comb Optim, 2011, 22: 674–683
97 Min X, Liu J, Dong Y, et al. Online preemptive hierarchical scheduling on two uniform machines with rejection. Asia-Pac J Oper Res, 2015, 32: 1550027
98 Imreh C, Noga J. Scheduling with machine cost. In: Proceeding of Randomization, Approximation, and Combinatorial Optimization. Algorithms and Techniques. Lecture Notes in Computer Science, vol. 1671. Heidelberg: Springer, 1999, 168–176
99 Dósa G, Tan Z. New upper and lower bounds for online scheduling with machine cost. Discrete Optim, 2010, 7: 125–135
100 Jiang Y, He Y. Preemptive online algorithms for scheduling with machine cost. Acta Inform, 2005, 41: 315–340
101 Dósa G, He Y. Better online algorithms for scheduling with machine cost. SIAM J Comput, 2004, 33: 1035–1051
102 Jiang Y W, He Y. Semi-online algorithms for scheduling with machine cost. J Comput Sci Tech, 2006, 21: 984–988
103 He Y, Cai S. Semi-online scheduling with machine cost. J Comput Sci Tech, 2002, 17: 781–787
104 Imreh C. Online scheduling with general machine cost functions. Discrete Appl Math, 2009, 157: 2070–2077
105 Akaria I, Epstein L. An optimal online algorithm for scheduling with general machine cost functions. J Sched, 2020, 23: 155–162
106 Jiang Y, Hu J, Liu L, et al. Competitive ratios for preemptive and non-preemptive online scheduling with nondecreasing concave machine cost. Inform Sci, 2014, 269: 128–141
107 Csirik J, Dósa G, Koszo D. Online scheduling with machine cost and a quadratic objective function. In: Proceeding of International Conference on Current Trends in Theory and Practice of Informatics. Lecture Notes in Computer Science, vol. 12011. Cham: Springer, 2020, 199–210
108 Nagy-György J, Imreh C. Online scheduling with machine cost and rejection. Discrete Appl Math, 2007, 155: 2546–2554
109 Dósa G, He Y. Scheduling with machine cost and rejection. J Comb Optim, 2006, 12: 337–350
110 Chandra A K, Wong C K. Worst-case analysis of a placement algorithm related to storage allocation. SIAM J Comput, 1975, 4: 249–263
111 Albers S. Energy-efficient algorithms. Commun ACM, 2010, 53: 86–96
112 Avidor A, Azar Y, Sgall J. Ancient and new algorithms for load balancing in the l_p norm. Algorithmica, 2001, 29: 422–441
113 Im S, Kell N, Panigrahi D, et al. Online load balancing on related machines. In: Proceedings of the 50th Annual ACM Symposium on Theory of Computing. New York: Association for Computing Machinery, 2018, 30–43
114 Awerbuch B, Azar Y, Grove E F, et al. Load balancing in the L_p norm. In: Proceedings of 36th IEEE Annual Foundations of Computer Science. New York: IEEE, 1995, 383–391
115 Caragiannis I. Better bounds for online load balancing on unrelated machines. In: Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms. Philadelphia: SIAM, 2008, 972–981
116 Caragiannis I, Flammini M, Kaklamanis C, et al. Tight bounds for selfish and greedy load balancing. Algorithmica, 2011, 61: 606–637
117 Suri S, Toth C D, Zhou Y. Selfish load balancing and atomic congestion games. Algorithmica, 2007, 47: 79–96
118 Du D, Jiang X, Zhang G. Optimal preemptive online scheduling to minimize l_p norm on two processors. J Ind Manag

Optim, 2005, 1: 345–351
119 Shuai T, Du D, Jiang X. On-line preemptive machine scheduling with ℓ_p norm on two uniform machines. J Sched, 2015, 18: 185–194
120 Lin L, Tan Z, He Y. Deterministic and randomized scheduling problems under the l_p norm on two identical machines. J Zhejiang Univ-Sci A, 2005, 6: 20–26
121 Qi X, Yuan J. Semi-online hierarchical scheduling on two machines for l_p-norm load balancing. Asia-Pac J Oper Res, 2019, 36: 1950002
122 Qi X, Yuan J. Semi-online hierarchical scheduling for l_p-norm load balancing with buffer or rearrangements. 4OR, 2017, 15: 265–276
123 Gunther E, Maurer O, Megow N, et al. A new approach to online scheduling: Approximating the optimal competitive ratio. In: Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms. Philadelphia: SIAM, 2013, 118–128
124 Chen L, Ye D, Zhang G. Approximating the optimal algorithm for online scheduling problems via dynamic programming. Asia-Pac J Oper Res, 2015, 32: 1540011
125 Megow N, Wiese A. Competitive-ratio approximation schemes for minimizing the makespan in the onlinelist model. arXiv:1303.1912, 2013
126 Boyar J, Favrholdt L M, Kudahl C, et al. Online algorithms with advice. ACM Comput Surv, 2017, 50: 19:1–19:34
127 Renault M P, Rosén A, van Stee R. Online algorithms with advice for bin packing and scheduling problems. Theor Comput Sci, 2015, 600: 155–170
128 Boyar J, Favrholdt L M, Kudahl C, et al. Weighted online problems with advice. Theory Comput Syst, 2018, 62: 1443–1469
129 Boyar J, Irani S, Larsen K S. A comparison of performance measures for online algorithms. Algorithmica, 2015, 72: 969–994
130 Boyar J, Larsen K S, Maiti A. A comparison of performance measures via online search. Theor Comput Sci, 2014, 532: 2–13
131 Kenyon C. Best-fit bin-packing with random order. In: Proceeding of 7th Annual ACM-SIAM Symposium on Discrete Algorithms, vol. 96. Philadelphia: SIAM, 1996, 359–364
132 Osborn C J, Torng E. List's worst-average-case or WAC ratio. J Sched, 2008, 11: 213–215
133 Epstein L, Favrholdt L M, Kohrt J S. Separating online scheduling algorithms with the relative worst order ratio. J Comb Optim, 2006, 12: 363–386
134 Boyar J, Epstein L, Favrholdt L M, et al. Online-bounded analysis. J Sched, 2018, 21: 429–441

Online scheduling on parallel machines: A survey

Ling Lin & Zhiyi Tan

Abstract Scheduling is one of the most fundamental combinatorial optimization problems. Online scheduling problems have received considerable research interest in the last three decades. In this paper, we summarize new results on online over list scheduling problems on parallel machines, including problems with an objective to minimize the makespan and the L^p norm of machine loads, semi-online scheduling problems on identical, uniform and hierarchial machines, scheduling with rejection and scheduling with machine costs. We also introduce several new directions of online scheduling.

Keywords scheduling, online, competitive ratio

MSC(2010) 90B35, 90C27

doi: 10.1360/SSM-2020-0061