

两点混合环上的半在线算法

肖 满 李伟东

云南大学数学与统计学院 昆明 650504

(man1205@163.com)

摘 要 文中研究了两点混合环上负载均衡问题的两种半在线情形。给定一个两点混合环和若干流量需求,寻找合适的流量运输方式,使得环上的最大负载尽可能地小。当存在一个容量为 K 的缓冲区时,证明了该半在线情形的下界为 $4/3$ 。特别地,当 $K=1$ 时,证明了下界为 $3/2$,并给出了一个竞争比至多为 $8/5$ 的半在线算法。当所有流的需求之和已知时,设计了一个竞争比为 $3/2$ 的最优半在线算法。

关键词:混合环;半在线算法;缓冲区;竞争比;环负载

中图法分类号 TP301.6

Semi-online Algorithms for Mixed Ring with Two Nodes

XIAO Man and LI Wei-dong

School of Mathematics and Statistics, Yunnan University, Kunming 650504, China

Abstract Two semi-online scenarios of load balancing problem on a two-point mixed ring are studied. This paper gives a two-point mixed ring and several flow requests, finds a suitable flow transportation method to make the maximum load on the ring as small as possible. When there is a buffer with a capacity of K , the lower bound of $4/3$ is given. In particular, when $K=1$, a lower bound of $3/2$ is given, and a semi-online algorithm with a competitive ratio of at most $8/5$ is given. When the sum of all flow demands is known, an optimal semi-online algorithm with a competitive ratio of $3/2$ is designed.

Keywords Mixed ring, Online algorithm, Buffer, Competitive ratio, Ring loading

1 引言

环负载均衡问题在基于光纤的通信网络和多物流运输等领域有着广泛应用,近 20 年来,环负载均衡问题及其变种已得到深入的研究,在无向环和有向环上都取得了一些成果。环负载均衡问题即给定一个环网络 $R=(V, E)$, V 是顶点集, E 是无向边集, V 中的顶点之间有一些流,需要将这些流从起始顶点按照顺时针或者逆时针的方向通过 E 中的边运输到终点。 E 中每条边的负载为通过这条边的所有流的需求之和,目标为最小化 E 中边的最大负载。环负载均衡问题由 Cosares 等^[1]提出,文献[2-5]对其进行了大量研究。目前,环负载均衡问题研究主要有 3 种流模型:1)流的需求任意可分,即任意两个顶点之间的一个流可以任意分成两个部分,再分别沿着顺时针和逆时针两个方向运输;2)流的需求整数可分,即任意两个顶点之间的一个流可以任意分成两个整数部分,再分别沿着顺时针和逆时针两个方向运输;3)流的需求不可分,即任意两个顶点之间的一个流不能被拆分,只能沿着顺时针或逆时针方向运输。

对于无向环上的负载均衡问题, Khanna^[6]研究了流的需求不可分的情形,设计了一个多项式时间近似方案(PTAS)。 Schrijver 等^[3]研究了流的需求任意可分的情形,给出了一个 $O(n^2k)$ 时间复杂度的最优算法,其中 n 是无向环的顶点数

量, k 为流的数量。 Li 等^[6]研究了带惩罚费用的负载均衡问题,证明了该问题是 NP-hard 的,对流的需求整数可分情形给出了一个 1.58 -近似算法,对流的需求不可分情形给出了一个 $1.58+\epsilon$ -近似算法。 Nong 等^[7]考虑了流的需求整数可分情形下,带权重的环负载均衡问题。

对于有向环上的负载均衡问题, Wilfong 等^[8]考虑了流的需求任意可分的情形,设计了一种最优算法。 Myung 等^[9]考虑了流的需求整数可分的情形,给出了一个时间复杂度为 $O(nk)$ 的最优算法。 Myung 等^[10]提出了一个时间复杂度为 $O(\min\{nk, n^2\})$ 的最优算法,改进了这个结果。接着, Wang^[11]设计了一个线性时间的最优算法,再次改进了这个结果。 Becchetti 等^[12]考虑了流的需求不可分的情形,设计了一个 PTAS。 Guan 等^[13]考虑了带惩罚费用的情形,当流可分时,设计了一种 1.582 -近似算法;当流不可分时,设计了一种 3 -近似算法。

Havill 等^[14]首先提出了在线负载均衡问题,他们考虑了流的需求任意可分的情形,并给出了一个有向环上的最优在线算法。 Guan 等^[15]提出了混合环上的在线负载均衡问题,并考虑了两点混合环上的 3 种模型:1)流的需求任意可分,设计了一种最优在线算法;2)流的需求整数可分,设计了一种渐进最优在线算法;3)流的需求不可分,给出了一种在线算法。 两点环的 3 种环模型如图 1 所示。

基金项目:国家自然科学基金(12071417);云南省创新团队项目

This work was supported by the National Natural Science Foundation of China(12071417) and Yunnan Innovation Team Project.

通信作者:李伟东(weidongmath@126.com)

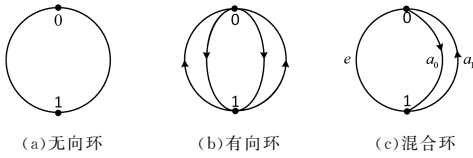


图1 3种环模型

Fig. 1 Three ring models

对于混合环上的在线负载均衡问题,目前仅知文献[15]中两点混合环上的成果,对于流不可分时,还未得到最优在线算法。受文献[15]的启发,结合文献[16],当流的需求不可分时,对于两点混合环上的在线负载均衡问题,我们分别考虑了带有一个容量为 $K \in \mathbb{N}^+$ 的缓冲区和流的需求之和 T 已知的情形。一个容量为 K 的缓冲区可以暂时储存至 K 个流时,当缓冲区中已经存有 K 个流时,若需要将新到达的一个流存入缓冲区,则需要先将缓冲区中的一个流分配到环上。

本文第2节描述了具体的问题,定义了文中使用的符号;第3节研究了带有一个容量为 K 的缓冲区的情形,对于 $\forall K \in \mathbb{N}^+$,给出了一个下界 $\frac{4}{3}$,特别是当 $K=1$ 时,给出了一个下界 $\frac{3}{2}$,并设计了一个竞争比为 $\frac{8}{5}$ 的半在线算法;第4节研究了提前获知流的需求之和 T 的情形,设计了一个竞争比为 $\frac{3}{2}$ 的最优半在线算法;最后总结全文。

2 预备知识

给定一个混合环 $R=(V, E \cup A)$,其中 $V=\{0,1\}$ 是环的顶点集合, $E=\{e(\in\{0,1\})\}$ 是环上的无向边集, $A=\{a_0=(0,1), a_1=(1,0)\}$ 是环上的有向边集。 V 中顶点之间有一些流需要运输,这些流都一个接一个地到达,且每一个流都不可分(即任意一个流不能被拆分为两部分,再沿着环上不同的边进行运输)。令第 j 个到达的流为 $f_j=(s_j, t_j, d_j)$,其中 s_j 是流 f_j 的源点, t_j 是流 f_j 的终点, d_j 是流 f_j 的需求,且有 $s_j \in \{0,1\}, t_j \in \{0,1\}, s_j \neq t_j$ 。 f_j 仅需通过有向边 a_{s_j} 从 s_j 运输到 t_j 或者仅通过无向边 e 从 s_j 运输到 t_j 。 V 中顶点之间的流被分为两种类型,若 $s_j=i \in \{0,1\}$,则称流 f_j 为 i 型流。且这些流都在线到达,即只有当前的流 f_j 被处理后,下一个流 $f_{j+1}=(s_{j+1}, t_{j+1}, d_{j+1})$ 才会到达,在 f_{j+1} 未到达之前, s_{j+1}, t_{j+1} 和 d_{j+1} 的值未知。目标为最小化混合环 R 上边的最大负载,环上每条边的负载即为通过这条边的流的需求之和。对于一个在线算法 A 和一个流序列 $\mathcal{F}=f_1, f_2, \dots, f_n$,令 L_a^j 为前 j 个流被处理后边 a 上的负载, $a \in \{a_0, e, a_1\}$ 。为简化符号,也令 $L_a^n = L_a$ 。因此,在线算法 A 对于流序列输出的目标函数值为:

$$C_A(\mathcal{F}) = \max\{L_{a_0}, L_e, L_{a_1}\}$$

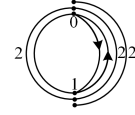
令 $C_{OPT}(\mathcal{F})$ (简记为 C_{OPT}) 为该问题离线情形的最优值,也将 $C_A(\mathcal{F})$ 简记为 C_A 。在线算法 A 的优劣用竞争比 ρ_A 来衡量,对于极小化问题,竞争比 ρ_A 定义为:

$$\rho_A = \inf\{\rho \geq 1 \mid C_A(I) \leq \rho C_{OPT}(I), \forall I\}$$

如果一个在线问题不存在竞争比小于 ρ 的在线算法,则称 ρ 是该问题的一个下界。如果在线算法 A 的竞争比与该在线问题的一个下界相等,则称算法 A 是最优的在线算法。

本文研究的问题在不考虑缓冲区和流的需求之和 T 未

知时,即为文献[17]中两类工件的在线排序问题,文献[17]给出了一个下界 $\frac{7}{4}$ 和一个竞争比为 $\frac{9}{5}$ 的在线算法。例如,考虑一个实例 (R, \mathcal{F}) ,流序列有4个流: $f_1=(0,1,1), f_2=(1,0,1), f_3=(0,1,2), f_4=(1,0,2)$ 。如图2所示,流不可分,对流序列的最优分配即将流 f_1 和 f_2 分配到无向边 e ;将流 f_3 和 f_4 分别分配到有向边 a_0 和 a_1 ,得到最优值 $C_{OPT}=2$ 。

图2 实例 (R, \mathcal{F}) 的最优分配Fig. 2 Optimal allocation of instance (R, \mathcal{F})

为方便叙述,对于 $\forall i \in \{0,1\}$,定义如下符号:

$T^j(i)$:前 j 个流中 i 型流的需求之和。

$L_e^j(i)$:前 j 个流被处理后,无向边 e 上 i 型流造成的负载。

$p_{\max}^j(i)$:前 j 个流中 i 型流的最大需求。

为简化符号,定义 $T^n(i)=T(i), L_e^n(i)=L_e(i), p_{\max}^n(i)=p_{\max}(i)$ 。

3 使用一个容量为 K 的缓冲区

本节考虑混合环 R 带有一个容量为 $K(K \in \mathbb{N}^+)$ 的缓冲区,当新的一个流到达后,可以直接分配到环上,也可以考虑暂时存入缓冲区。首先,对于 $\forall K \in \mathbb{N}^+$,证明了该问题下界至少为 $\frac{4}{3}$ 。接着特别考虑了 $K=1$ 时的情形,证明了下界至少为 $\frac{3}{2}$,并设计了一个竞争比至多为 $\frac{8}{5}$ 的半在线算法。

引理 1 令 $LB^j = \max\{\frac{T^j(0)+T^j(1)}{3}, \frac{T^j(0)}{2}, \frac{T^j(1)}{2}, p_{\max}^j(0), p_{\max}^j(1)\}$,该问题在前 j 个流被处理后(被分配到环上或者被存入缓冲区),其最优目标值 C_{OPT}^j 至少为 LB^j 。

证明:在前 j 个流被处理后,因为 C_{OPT}^j 不小于环 R 上3条边负载的平均值,所以 $C_{OPT}^j \geq \frac{T^j(0)+T^j(1)}{3}$ 。因为0型流只能通过有向边 a_0 和无向边 e 运输,所以 C_{OPT}^j 不小于0型流的需求之和在有向边 a_0 和无向边 e 上的平均值,即有 $C_{OPT}^j \geq \frac{T^j(0)}{2}$ 。同理,也有 $C_{OPT}^j \geq \frac{T^j(1)}{2}$ 。因为 C_{OPT}^j 不小于任意一个流的需求,所以 $C_{OPT}^j \geq \max\{p_{\max}^j(0), p_{\max}^j(1)\}$ 。因此,此引理成立。

为简化符号,令 $LB^n = LB$,则由引理1可得:

$$C_{OPT} \geq LB = \max\{\frac{T(0)+T(1)}{3}, \frac{T(0)}{2}, \frac{T(1)}{2}, p_{\max}(0), p_{\max}(1)\} \quad (1)$$

定理 1 对于 $\forall K \in \mathbb{N}^+$,不存在一个竞争比小于 $\frac{4}{3}$ 的在线算法 A 。

证明:令 N 是一个充分大的正整数, $\epsilon = \frac{1}{2NK}$,首先到达 $2NK$ 个大小为 ϵ 的0型流。

若 $L_{a_0}^{2NK-K} \geq \frac{2}{3}$,则最后一个流 $f_{2NK+1}=(1,0, \frac{1}{2})$ 到达。

在算法执行完毕之后,得到 $C_A \geq L_{a_0} \geq L_{a_0}^{2NK-K} \geq \frac{2}{3}$ 。此时,流

序列的最优分配即分别将 NK 个大小为 ϵ 的流分配到有向边 a_0 和无向边 e 上;将流 f_{2NK+1} 分配到有向边 a_1 上,得到 $C_{OPT} = \frac{1}{2}$ 。因此,

$$\frac{C_A}{C_{OPT}} \geq \frac{L_{a_0}}{C_{OPT}} \geq \frac{\frac{2}{3}}{\frac{1}{2}} = \frac{4}{3}$$

若 $L_{a_0}^{2NK-K} < \frac{2}{3}$, 则 $L_e^{2NK-K} > \frac{1}{3} - \frac{1}{2N}$ 。最后两个流 $f_{2NK+1} = (1, 0, 1)$ 和 $f_{2NK+2} = (1, 0, 1)$ 到达。因为流 f_{2NK+1} 和流 f_{2NK+2} 不能分配到有向边 a_0 , 在算法执行完毕之后, 可得 $C_A \geq \max\{L_e, L_{a_1}\} \geq 1 + \frac{1}{3} - \frac{1}{2N}$ 。此时, 流序列的最优分配策略即将 $2NK$ 个大小为 ϵ 的 0 型流分配到有向边 a_0 , 将流 $f_{2NK+1} = (1, 0, 1)$ 和 $f_{2NK+2} = (1, 0, 1)$ 分别分配到无向边 e 和有向边 a_1 上, 得到 $C_{OPT} = 1$ 。因此:

$$\frac{C_A}{C_{OPT}} \geq \frac{\max\{L_e, L_{a_1}\}}{C_{OPT}} \geq \frac{1 + \frac{1}{3} - \frac{1}{2N}}{1} = \frac{4}{3} - \frac{1}{2N} \rightarrow \frac{4}{3} \quad (N \rightarrow \infty)$$

证毕。

定理 2 当缓冲区容量 $K=1$ 时, 不存在一种竞争比小于 $\frac{3}{2}$ 的在线算法 A。

证明: 首先到达流 $f_1 = (0, 1, 1)$ 和 $f_2 = (1, 0, 1)$ 。因为缓冲区容量 $K=1$, 则必有一个流会被立即分配到环上, 不妨设流 f_1 被分配。

若算法 A 将流 f_1 分配到有向边 a_0 上, 则最后两个流 $f_3 = (0, 1, 2)$ 和 $f_4 = (1, 0, 2)$ 到达。在算法 A 执行完毕之后, 可得 $C_A \geq 3$ 。此时, 流序列的最优分配策略即将流 f_1 和 f_2 分配到无向边 e 上, 将流 f_3 分配到有向边 a_0 上, 将流 f_4 分配到有向边 a_1 上, 得到 $C_{OPT} = 2$ 。因此, $\frac{C_A}{C_{OPT}} \geq \frac{3}{2}$ 。

若算法 A 将流 f_1 分配到无向边 e 上, 最后一个流 $f_3 = (1, 0, 1)$ 到达。在算法 A 执行完毕之后, 得到 $C_A = \max\{L_e, L_{a_1}\} \geq 2$ 。此时, 流序列的最优分配策略即将流 f_1 分配到有向边 a_0 上, 将流 f_2 和 f_3 分别分配到无向边 e 和有向边 a_1 上, 得到 $C_{OPT} = 1$ 。因此, $\frac{C_A}{C_{OPT}} \geq 2$ 。证毕。

本文算法 H_1 在前面部分将尽可能多的流分配到有向边上, 最后将缓冲区中储存的流分配到无向边 e 上。算法 1 的具体步骤如下。

算法 1 H_1

Input: 混合环 R 上的一个流序列

Output: 流序列 F 在混合环 R 上的一个分配方案

1. 令 $j=1, L_{a_0}^{j-1} = L_e^{j-1} = L_{a_1}^{j-1} = 0$ 。
2. For 每一个流 $f_j = (s_j, t_j, d_j)$ 到达。
3. If 缓冲区中存有 s_j 型流, 比较流 f_j 和缓冲区中流的需求 (若缓冲区中没有流, 则假设缓冲区中存有一个需求为 0 的 s_j 型流)。将需求较大的流令为 B^j (也用 B^j 表示这个流的需求), 存入缓冲区, 将较小的流令为 f_j 。
4. End If
5. If $L_{a_{s_j}}^{j-1} + d_j \leq \frac{8}{5} LB^j$, 将流 f_j 分配到有向边 a_{s_j} 上。
6. Else $L_{a_{s_j}}^{j-1} + d_j > \frac{8}{5} LB^j$ 。
7. Else If $L_e^{j-1}(s_j) + d_j \leq \frac{1}{5} T^j(s_j)$, 将流 f_j 分到无向边 e 上。

8. Else 将缓冲区中 t_j 型流 B^j 分配到有向边 a_{t_j} 上, 并将流 f_j 存入缓冲区。
9. 令 $j=j+1$ 。
10. End For
11. 将缓冲区中的流分配到无向边 e 上。

算法 H_1 的流程框图如图 3 所示。

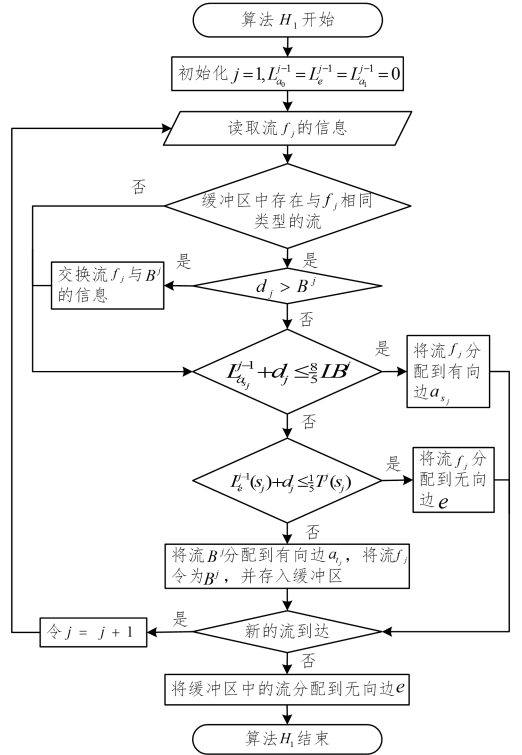


图 3 算法 H_1 的流程框图

Fig. 3 Flow chart of H_1 algorithm

定理 3 当缓冲区容量 $K=1$ 时, 算法 H_1 的竞争比至多为 $\frac{8}{5}$ 。

证明: 首先, 我们证明算法 H_1 执行步骤 8 时, 缓冲区中一定存有一个 t_j 型流 f_t , 且流 f_t 被分配到有向边 a_{t_j} 后, 必有 $L_{a_{t_j}}^{j-1} + d_t \leq \frac{8}{5} LB^j$ 。

当算法 H_1 执行步骤 8 时, 有:

$$L_e^{j-1}(s_j) + d_j > \frac{1}{5} T^j(s_j) \quad (2)$$

由步骤 3 可知, 若此时缓冲区中没有一个 t_j 型流, 则必有一个 s_j 型流 f_s 在步骤 3 被存入缓冲区, 且 $d_s \geq d_j$ 。又由步骤 6 有:

$$L_{a_{s_j}}^{j-1} + d_j > \frac{8}{5} LB^j \quad (3)$$

因此, 由不等式 (2)、不等式 (3) 和 $d_s \geq d_j$ 有:

$$\begin{aligned} L_e^{j-1}(s_j) + d_s + L_{a_{s_j}}^{j-1} + d_j &\geq L_e^{j-1}(s_j) + d_j + L_{a_{s_j}}^{j-1} + d_j \\ &> \frac{1}{5} T^j(s_j) + \frac{8}{5} LB^j \\ &\geq \frac{1}{5} T^j(s_j) + \frac{4}{5} T^j(s_j) \\ &= T^j(s_j) \end{aligned}$$

即有 $T^j(s_j) = L_e^{j-1}(s_j) + d_s + L_{a_{s_j}}^{j-1} + d_j > T^j(s_j)$, 矛盾, 因此证得算法 H_1 执行步骤 8 时, 缓冲区中一定存有一个 t_j 型流 f_t 。

接下来,证明流 f_i 被分配到有向边 a_{t_j} 后,有 $L_{a_{t_j}}^{j-1} + d_i \leq \frac{8}{5}LB^j$ 。

由步骤 6 和 LB^j 的定义可知:

$$L_{a_{t_j}}^{j-1} + d_j > \frac{8}{5}LB^j \geq \frac{8}{15}(T^j(0) + T^j(1)) \quad (4)$$

假设 $L_{a_{t_j}}^{j-1} + d_i > \frac{8}{5}LB^j$, 则可得:

$$L_{a_{t_j}}^{j-1} + d_i > \frac{8}{5}LB^j \geq \frac{8}{15}(T^j(0) + T^j(1)) \quad (5)$$

因此,由不等式(4)和不等式(5)可得:

$T^j(0) + T^j(1) = L_{a_{t_j}}^{j-1} + d_j + L_{a_{t_j}}^{j-1} + d_i > \frac{16}{15}(T^j(0) + T^j(1))$ 矛盾,因此流 f_i 被分配到有向边 a_{t_j} 后,必有 $L_{a_{t_j}}^{j-1} + d_i \leq \frac{8}{5}LB^j$ 。因此,对于 $k=0,1$,结合步骤 5 可知:

$$L_{a_k}^{n-1} \leq \frac{8}{5}LB^n = \frac{8}{5}LB \quad (6)$$

若 $C_{H_1} = L_{a_k}$,由步骤 11 知 $L_{a_k} = L_{a_k}^{n-1}$,则再由不等式(1)和不等式(6)得到:

$$\frac{C_{H_1}}{C_{OPT}} = \frac{L_{a_k}}{C_{OPT}} \leq \frac{\frac{8LB}{5}}{LB} = \frac{8}{5}$$

若 $C_{H_1} = L_e$,对于 $i \in \{0,1\}, 1 \leq j \leq n-1$,令 f_j 是最后一个被分配到无向边 e 上的 i 型流。则由算法 H_1 知 f_j 在步骤 7 被分配到无向边 e ,又由步骤 7 有:

$$L_e^j(i) = L_e^{j-1}(i) + d_j \leq \frac{1}{5}T^j(i) \quad (7)$$

因此,由 f_j 是 $1 \leq j \leq n-1$ 中最后一个被分配到无向边 e 上的 i 型流,和不等式(7)得到:

$$L_e^{n-1}(i) = L_e^j(i) \leq \frac{1}{5}T^j(i) \leq \frac{1}{5}T^{n-1}(i) \quad (8)$$

由不等式(8)得到 $L_e^{n-1} = L_e^{n-1}(0) + L_e^{n-1}(1) \leq \frac{1}{5}(T^{n-1}(0) + T^{n-1}(1))$,故由步骤 11 得到:

$$\begin{aligned} L_e &= L_e^{n-1} + B^n \\ &\leq \frac{1}{5}(T^{n-1}(0) + T^{n-1}(1)) + B^n \\ &\leq \frac{1}{5}(T^{n-1}(0) + T^{n-1}(1)) + \max\{p_{\max}(0), p_{\max}(1)\} \\ &\leq \frac{1}{5}(T(0) + T(1)) + \max\{p_{\max}(0), p_{\max}(1)\} \end{aligned} \quad (9)$$

因此,由不等式(1)和不等式(9)得到:

$$\begin{aligned} \frac{C_{H_1}}{C_{OPT}} &= \frac{L_e}{C_{OPT}} \\ &\leq \frac{\frac{1}{5}(T(0) + T(1)) + \max\{p_{\max}(0), p_{\max}(1)\}}{C_{OPT}} \\ &\leq \frac{\frac{1}{5}(T(0) + T(1)) + \max\{p_{\max}(0), p_{\max}(1)\}}{\frac{1}{3}(T(0) + T(1))} \\ &= \frac{8}{5} \end{aligned}$$

证毕。

4 已知流的需求之和 T

本节考虑在流达到之前已知所有流的需求之和为 T ,其

中 $T = T(0) + T(1)$ 。首先证明了该问题下界至少为 $\frac{3}{2}$,接着设计了一个竞争比至多为 $\frac{3}{2}$ 的半在线算法。

本节取 $LB^j = \max\{\frac{T}{3}, \frac{T^j(0)}{2}, \frac{T^j(1)}{2}, p_{\max}^j(0), p_{\max}^j(1)\}$, 则:

$$LB = \max\{\frac{T}{3}, \frac{T(0)}{2}, \frac{T(1)}{2}, p_{\max}(0), p_{\max}(1)\}$$

由不等式(1),同样有:

$$C_{OPT} \geq LB = \max\{\frac{T}{3}, \frac{T(0)}{2}, \frac{T(1)}{2}, p_{\max}(0), p_{\max}(1)\} \quad (10)$$

定理 4 当已知 0 型流和 1 型流的需求之和 T ,则不存在一种竞争比小于 $\frac{3}{2}$ 的在线算法 A 。

证明:令 $T=6$,首先流 $f_1=(0,1,1)$ 和 $f_2=(1,0,1)$ 到达。如果算法 A 将流 f_1 和 f_2 都分配给无向边 e ,则最后 3 个流 $f_3=(0,1,1), f_4=(1,0,1), f_5=(1,0,2)$ 到达。因为流 f_4 和 f_5 只能分配到无向边 e 和有向边 a_1 ,算法 A 结束后,可得 $C_A = \max\{L_e, L_{a_1}\} \geq 3$ 。此时流序列最优的一种分配即将流 f_1 和 f_3 分配到有向边 a_0 ,将流 f_2 和 f_4 分配到无向边 e ,将流 f_5 分配到有向边 a_1 ,得到 $C_{OPT}=2$ 。因此, $\frac{C_A}{C_{OPT}} \geq \frac{3}{2}$ 。

否则,到达最后 2 个流: $f_3=(0,1,2)$ 和 $f_4=(1,0,2)$ 。算法 A 结束后,易得 $C^A \geq 3$ 。此时流序列的最优分配即将流 f_1 和 f_2 分配到无向边 e ,将流 f_3 和 f_4 分别分配到有向边 a_0 和 a_1 ,得到 $C_{OPT}=2$ 。因此,得到 $\frac{C_A}{C_{OPT}} \geq \frac{3}{2}$ 。证毕。

本文算法 H_2 的主要策略是将尽可能多的流分配到有向边上,具体步骤如算法 2 所示。

算法 2 H_2

Input:混合环 R 上的一个流序列 F

Output:流序列 F 在混合环 R 上的一个分配方案

1. 令 $j=1, L_{a_0}^{j-1} = L_e^{j-1} = L_{a_1}^{j-1} = 0$ 。
2. For 每一个流 $f_j = (s_j, t_j, d_j)$ 到达。
3. If $L_{a_{s_j}}^{j-1} + d_j \leq \frac{3}{2}LB^j$, 将流 f_j 分配到有向边 a_{s_j} 上。
4. Else 将流 f_j 分配到无向边 e 上。
5. 令 $j=j+1$ 。
6. End For

算法 H_2 的流程图如图 4 所示。

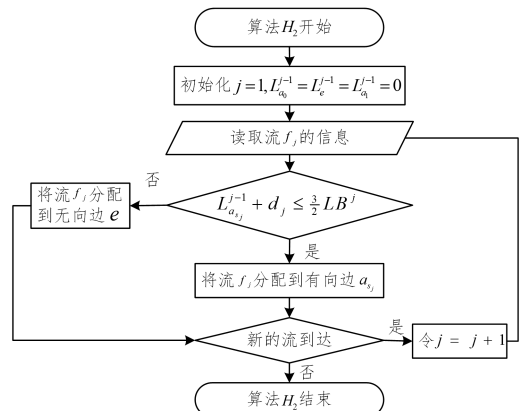


图 4 算法 H_2 的流程图

Fig. 4 Flow chart of H_2 algorithm

定理 5 当已知 0 型流和 1 型流的需求之和 T , 算法 H_2 的竞争比至多为 $\frac{3}{2}$.

证明: 首先证明无向边 e 上至多被分配了一种类型的流. 假设 e 上有 2 种类型的流(0 型流和 1 型流), 不妨令 e 上有 0 型流 f_x 和 1 型流 f_y , 即有 $s_x=0, s_y=1$. 因为流 f_x 和 f_y 都要被分配到无向边 e 上, 则由步骤 4 和 LB^l 的定义得到:

$$L_{a_x}^{x-1} + d_x > \frac{3}{2} LB^x \geq \frac{T}{2} \quad (11)$$

和

$$L_{a_y}^{y-1} + d_y > \frac{3}{2} LB^y \geq \frac{T}{2} \quad (12)$$

因此, 由不等式(11)和不等式(12)有:

$$T \geq L_{a_x}^{x-1} + d_x + L_{a_y}^{y-1} + d_y > T$$

矛盾, 故证得无向边 e 上至多只有一种类型的流.

对于 $k=0, 1$, 由步骤 3 可知 $L_{a_k} \leq \frac{3}{2} LB$. 若 $C_{H_2} = L_{a_k}$,

由不等式(10)得到:

$$\frac{L_{a_k}}{C_{OPT}} \leq \frac{\frac{3LB}{2}}{LB} = \frac{3}{2}$$

若 $C_{H_3} = L_e$, 当无向边 e 未被分配任意一个流, 则 $L_e = 0$. 当无向边 e 上有流, 则由前面的结论可知 e 上只有一种类型的流, 不妨令 e 上只有 0 型流, 且最后一个分配到 e 上的 0 型流为 $f_i (s_i=0)$. 因为 f_i 要被分配到 e 上, 则由步骤 4 得到:

$$L_{a_i}^{i-1} + d_i > \frac{3}{2} LB^i \quad (13)$$

又有 $d_i \leq p_{\max}^i(0) \leq LB^i$, 由不等式(13)得到 $L_{a_i}^{i-1} > \frac{LB^i}{2}$

$\geq \frac{T^i(0)}{4}$. 如果流 f_i 之后还有 0 型流到达, 将被分配到有向边 a_0 . 令流 f_i 之后的 0 型流的需求之和为 S_0 , 即有 $L_{a_0} = L_{a_i}^{i-1} + S_0 \geq \frac{T^i(0) + S_0}{4} = \frac{T(0)}{4}$, 因此得到 $L_e = T(0) - L_{a_0} \leq \frac{3}{4} T(0)$. 因此, 由不等式(10)得到:

$$\frac{L_e}{C_{OPT}} \leq \frac{\frac{3T(0)}{4}}{\frac{T(0)}{2}} = \frac{3}{2}$$

证毕.

结束语 对于两点混合环上的负载均衡问题, 本文研究了两种半在线情形. 当所有流的需求之和已知时, 本文设计了一种最优的半在线算法, 并严格证明了该算法的最优性. 当带有一个容量为 1 的缓冲区时, 给出了一个接近最优的半在线算法. 设计与缓冲区容量相关的最优半在线算法是未来值得研究的方向之一.

参 考 文 献

- [1] COSARES S, SANIEE I. An optimization problem related to balancing loads on sonet rings[J]. Telecommunication Systems, 1994, 3(2):165-181.
- [2] SCHRIJVER A, SEYMOUR P. The ring loading problem[J]. Siam Review, 1999, 41(4):777-791.
- [3] SCHRIJVER A, SEYMOUR P, WINKLER P. The ring loading problem[J]. Siam Journal on Discrete Mathematics, 1998, 11(1):1.

- [4] SKUTELLA M. A note on the ring loading problem [J]. Siam Journal on Discrete Mathematics, 2016, 30(1):327-342.
- [5] KHANNA S. A polynomial time approximation scheme for the sonet ring loading problem [J]. Bell Labs Technical Journal, 1997, 2(2):36-41.
- [6] LI W D, LIJ P, GUAN L. Approximation algorithms for the ring loading problem with penalty cost [J]. Information Processing Letters, 2014, 14(1/2):56-59.
- [7] NONG Q Q, YUAN J J, LIN Y X. The weighted link ring loading problem [J]. Journal of Combinatorial Optimization, 2009, 18(1):38-50.
- [8] WILFONG G, WINKLER P. Ring routing and wavelength translation [C] // Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, 1998:333-341.
- [9] MYUNG S Y, HU G K, DONG W T. Optimal load balancing on sonet bidirectional rings [J]. Operations Research, 1997, 45(1):148-152.
- [10] MYUNGS Y, HU G K. On the ring loading problem with demand splitting [J]. Operations Research Letters, 2004, 32(2):167-173.
- [11] WANG B F. Linear time algorithms for the ring loading problem with demand splitting [J]. Journal of Algorithms, 2005, 54(1):45-57.
- [12] BECCHETTI L, IANNI M D, MARCHETTI-SPACCAMELA A. Approximation algorithms for routing and call scheduling in all-optical chains and rings [J]. Theoretical Computer Science, 2002, 287(2):429-448.
- [13] GUAN L, LI J P, ZHANG X J, et al. The directed ring loading with penalty cost [C] // International Workshop on Algorithms and Computation, Springer, Cham, 2015:20-31.
- [14] HAVILL J T, HUTSON K R. Optimal online ring routing [J]. Networks, 2010, 57(2):187-197.
- [15] GUAN L, LI W D, XIAO M. Online algorithms for the mixed ring loading problem with two nodes [J]. Optimization Letters, 2021, 15(4):1229-1239.
- [16] CHEN X, DING N, DOSA G, et al. Online hierarchical scheduling on two machines with known total size of low-hierarchy jobs [J]. International Journal of Computer Mathematics, 2015, 92(5/6):873-881.
- [17] KARHI S, SHABTAY D. Online scheduling of two job types on a set of multipurpose machines [J]. International Journal of Production Economics, 2014, 150:155-162.



XIAO Man, born in 1995, master student. His main research interests include discrete optimization, computational complexity, and resource allocation.



LI Wei-dong, born in 1981, Ph.D, professor. His main research interests include discrete optimization, algorithmic game theory and cloud computing.