

分类号 TP301
学校代码 10487

学号 M202073296
密级 公开

华中科技大学

硕士学位论文

(学术型 ☒ 专业型 ☐)

论文标题
论文标题第二行

学位申请人： XXX

学科专业： XXX

指导教师： XX 教授

答辩日期： 2023 年 5 月 12 日

**A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Master Degree in Engineering**

**Randomized Algorithms for Related Problems of
Minimum Cost Perfect Matching With Delay**

Candidate : XXX

Major : Computing software and theory

Supervisor : Prof. XX

Huazhong University of Science and Technology

Wuhan 430074, P. R. China

May, 2023

独创性声明

本人声明所呈交的学位论文是我个人在导师的指导下进行的研究工作及取得的成果。尽我所知，除文中已标明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本论文属于 ☐ 保 密 ☐，在 ____ 年解密后适用本授权书。
☐ 不保密 ☐。

（请在以上方框内打“√”）

学位论文作者签名：

指导教师签名：

日期： 年 月 日

日期： 年 月 日

华中科技大学硕士学位论文

摘 要

关键词:

Abstract

Keywords:

目 录

| | |
|----------------------------|------|
| 摘 要 | I |
| Abstract | II |
| 1 绪论 | |
| 1.1 研究的背景与意义 | (1) |
| 1.2 主要研究内容 | (1) |
| 1.3 组织结构 | (1) |
| 2 第二章标题 | |
| 2.1 相关概念 | (2) |
| 2.2 问题模型的简化分析 | (2) |
| 2.3 本章小结 | (2) |
| 3 两源带延迟最小代价完美匹配的随机近似 | |
| 3.1 最优解的结构分析 | (3) |
| 3.2 2-竞争的状态化随机延迟算法 | (7) |
| 3.3 随机近似下界的归纳证明 | (11) |
| 3.4 随机近似下界的演绎推导 | (16) |
| 3.5 本章小结 | (20) |
| 4 乘车带延迟最小代价完美匹配问题及随机算法 | |
| 5 总结与展望 | |
| 致 谢 | (23) |
| 参考文献 | (24) |
| 附录 1 攻读硕士学位期间取得的研究成果 | (25) |

1 绪论

1.1 研究的背景与意义

1.2 主要研究内容

1.3 组织结构

2 第二章标题

2.1 相关概念

2.2 问题模型的简化分析

2.3 本章小结

3 两源带延迟最小代价完美匹配的随机近似

3.1 最优解的结构分析

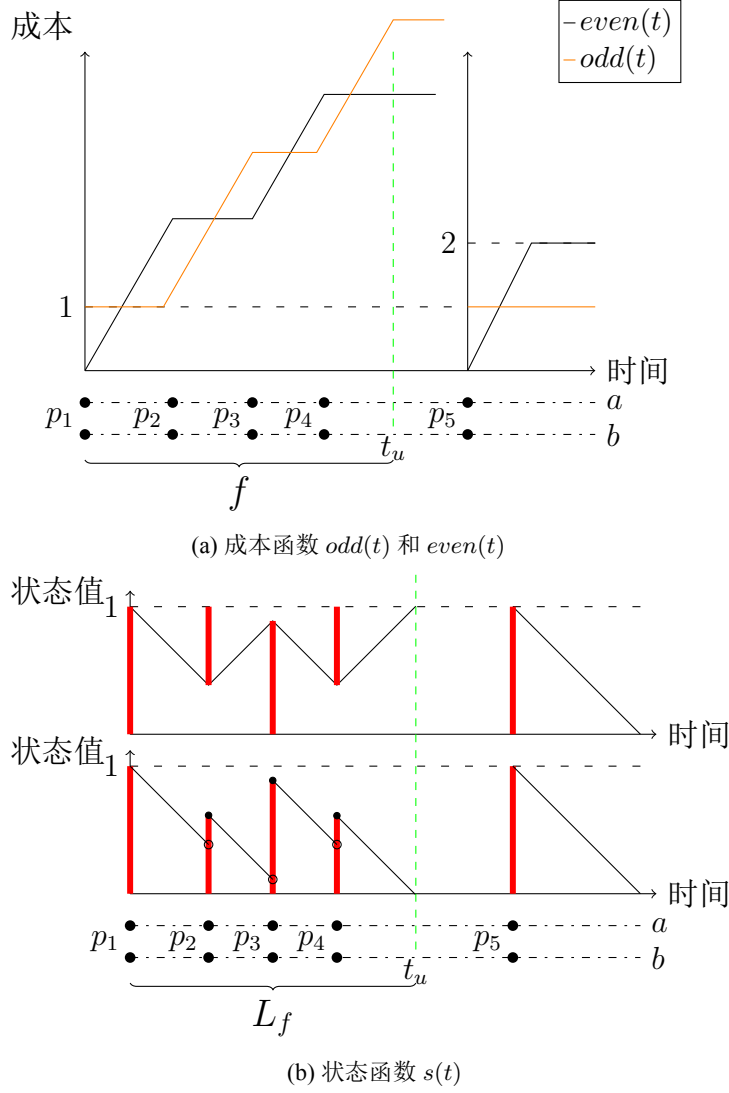


图 3.1 成本函数 $odd(t)$, $even(t)$ 和状态函数 $s(t)$

接下来，对每个独立片段都可以定义其状态函数：

$$s(t) = \begin{cases} \frac{odd(t) - even(t) + 1}{2} & \text{若 } d \text{ 为奇,} \\ \frac{even(t) - odd(t) + 1}{2} & \text{若 } d \text{ 为偶} \end{cases}$$

假设片段开始为 0 时刻, d 是到 t 时刻为止 (包括 t 时刻) 所到达的请求对数量。在图 3.2 中展示了一个样例, 七个请求对的序列被划分为 3 个独立片段。在每个独立片段的每个请求对中, 红色线段的长度等于该请求对的状态值。

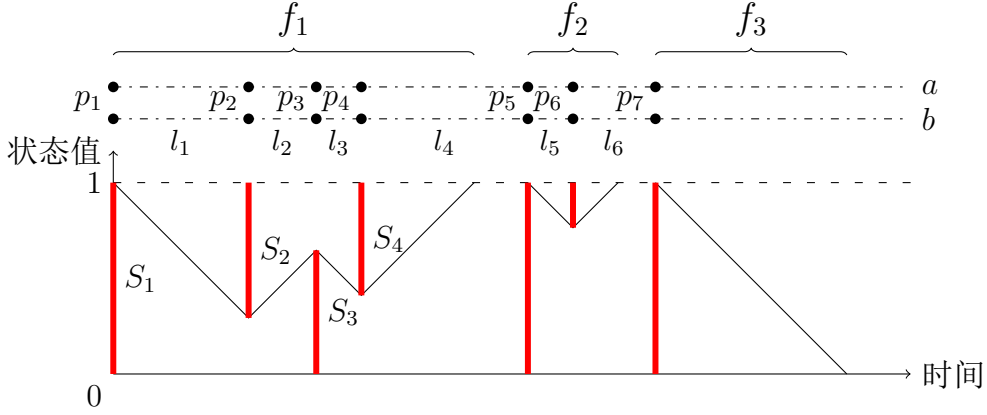


图 3.2 独立片段识别样例

注意到算法 ?? 中使用请求对的状态值来识别独立片段, 下面将状态值和状态函数之间建立联系, 即证明 $s(t_i) = S_i$ 。

引理 3.1: 对任意独立片段 $f = (f_1, \dots, f_k)$ 和其中的任一请求对 $f_1 \leq i \leq f_k$, 有 $s(t_i) = S_i$ 。

证明: 考虑 $i \geq 2$ 。如果 $i - f_1$ 是奇数, 在时间区间 $(t_{i-1}, t_i]$ 内, $odd(\cdot)$ 不变, 而 $even(\cdot)$ 增加了 $2l_{i-1}$ 。因此可推出,

$$\begin{aligned} s(t_i) &= (even(t_i) - odd(t_i) + 1)/2 = (even(t_{i-1}) + 2l_{i-1} - odd(t_{i-1}) + 1)/2 \\ &= 1 - (odd(t_{i-1}) - even(t_{i-1}) + 1)/2 + l_{i-1} = 1 - s(t_{i-1}) + l_{i-1}. \end{aligned}$$

对于每个独立片段 $f = (f_1, \dots, f_k)$, 状态函数在时间间隔 $[t_{i-1}, t_i]$ 中以单位速率减少。在最后一对 f_k 独立片段之后, 令 t_u 为状态函数最早变为 0 的时刻, 即 $s(t_u) = 0$ 。显然, $t_u = t_{f_k} + s(f_k)$ 。为了完善定义, 在 t_u 之后, 让 $s(\cdot)$ 保持 0, 直到下一个独立片段开始。称 t_u 为独立片段 f 的结束时间。由此可以定义几何量 $L_f := t_u - t_{f_1}$ 为独立片段 f 的长度。证毕。 ■

可以发现, 对于每个独立片段 $f = (f_1, \dots, f_k)$, 状态函数在时间间隔 $[t_{i-1}, t_i]$ 中以单位速率减少。在最后一对 f_k 独立片段之后, 令 t_u 为状态函数最早变为 0 的时刻, 也就是 $s(t_u) = 0$ 。显然, $t_u = t_{f_k} + s(f_k)$ 。为了完善定义, 在 t_u 之后, 让 $s(\cdot)$ 保持 0, 直到下一个独立片段开始。称 t_u 为独立片段 f 的结束时间。由此可以定义几何量 $L_f := t_u - t_{f_1}$ 为独立片段 f 的长度。

在前文的基础上, 定义了一个离线算法 TOPI (独立片段识别算法)。该算法的输入是一个序列, 首先使用算法??获得独立片段。对于每个独立片段, 如果它是一个独立偶片段, 则使用 $\text{Match}_{\text{even}}$ 进行处理; 否则采用 $\text{Match}_{\text{odd}}$ 处理。以下定理证明算法 TOPI 是最优的。

定理 3.1: 独立片段识别算法 TOPI 为 2-MPMD 问题的最优离线算法。

证明: 考虑一个序列 $R_p = (p_1, \dots, p_n)$ 。定义一个切片序列 $R[i]$ 为包含前 i 个请求对的子序列, 即 $R[i] := (p_1, p_2, \dots, p_i)$ 。令 $R[0]$ 表示空序列。对于任意切片序列 $R[i]$, 如果 TOPI 能在其上产生最优解, 则称该切片序列是可验证的。同时, 令 $\text{TOPI}(R[i])$ 表示 TOPI 在 $R[i]$ 上的解所产生的总成本。接下来归纳法作进行证明。

基础情形: $\text{TOPI}(R[0]) = 0$, 且 $\text{TOPI}(R[1]) = \text{Cost}(\{\text{Internal}(p_1)\})$, $R[0]$ 和 $R[1]$ 是可验证的。

归纳假设: 假设 $R[i-1]$ 和 $R[i-2]$ 都是可验证的。

归纳证明: 现在验证 $R[i]$ 。考虑序列 $R[i]$ 中最后一个请求对 p_i 。令 $f = (f_1, \dots, f_k)$ 为 TOPI 算法在解 $R[i]$ 中包含 p_i 的独立片段。由于 $R[i]$ 的最优解必须匹配请求对 q_j , 因此存在两种可能的匹配情况:

(1) OPT 外部匹配 p_{i-1} 和 p_i , 之后 $\text{OPT}(R[i]) = \text{TOPI}(R[i-2]) + 2l_{i-1}$ 。

(2) OPT 内部匹配 p_i , 之后 $\text{OPT}(R[i]) = \text{TOPI}(R[i-1]) + 1$ 。

注意算法??中, 当一个新的请求对到达时, 有两个不同的选项:

(a) 将其添加到当前片段中, 然后通过子程序执行 $\text{External}(p_{i-1}, p_i)$ 。这对应于情况 (1), 即 $\text{TOPI}(R[i]) = \text{TOPI}(R[i-2]) + 2l_{i-1}$ 。

(b) 开始一个新的片段，然后通过子程序执行 $\text{Internal}(p_i)$ 。这对应于情况 (2)，即 $\text{TOPI}(R[i]) = \text{TOPI}(R[i-1]) + 1$ 。

考虑在独立片段 f 中的最后一对点 p_i 。不失一般性，假设 $i - f_1$ 是奇数。需要证明 TOPI 对于 p_i 的决策是最优的。假设对于 p_i 选择了选项 (1)。根据子程序 ??，条件 $l_{i-1} < s(t_{i-1})$ 成立。

因此可推出，

$$\begin{aligned} l_{i-1} &< s(t_{i-1}) \\ \Rightarrow \quad \text{even}_f(t_{i-1}) - \text{odd}_f(t_{i-1}) &< 1 - 2l_{i-1} \\ \Rightarrow \quad \text{even}_f(t_{i-2}) - \text{odd}_f(t_{i-1}) &< 1 - 2l_{i-1} \end{aligned} \quad (3.1)$$

依据假设 $i - f_1$ 为奇数，因此 $R[i-2]$ 为独立偶片段, $R[i-1]$ 为独立奇片段。此前定义有 TOPI 在独立片段的成本函数，对应为 $\text{even}_f(t_{i-2})$ 和 $\text{odd}_f(t_{i-1})$ ，因此可推出：

$$\text{even}_f(t_{i-2}) - \text{odd}_f(t_{i-1}) = \text{TOPI}(R[i-2]) - \text{TOPI}(R[i-1]) \quad (3.2)$$

结合不等式3.1和等式3.2，有：

$$\text{TOPI}(R[i-2]) + 2l_{i-1} < \text{TOPI}(R[i-1]) + 1 \quad (3.3)$$

即选项 (1) 比选项 (2) 更好。

因此，算法 TOPI 对每一个请求对始终选择最优匹配，最终产生最优解。证毕。■

注意到从算法??中得到的片段是与成本无关的。全局最优解等于每个独立片段的局部最优解的组合。这也是“独立片段”这个术语的起源。对于单个独立片段，下面的引理表明其长度等于其最优解的成本。这个结论在后续的随机算法竞争分析中非常有用。

引理 3.2: 任一独立片段 f 均满足其长度等于其最优解的成本，即 $L_f = \text{Cost}_{\text{OPT}}(f)$ 。

证明： 假设有一个独立片段 $f = (f_1, f_2, \dots, f_k)$ ，其中应用了两个子程序 $\text{Match}_{\text{odd}}$ 和 $\text{Match}_{\text{even}}$ 。两个程序分别计算 f 的偶成本函数和奇成本函数，表示为 $\text{even}_f(t)$ 和 $\text{odd}_f(t)$ 。注意到 $\text{odd}_f(t_{f_1}) = 1$ 和 $\text{even}_f(t_{f_1}) = 0$ 。在时间 t_{f_1} 到 t_{f_k} 之间，其中一个函数按照速度 2 增长，而另一个保持不变。

$\text{Match}_{\text{odd}}$ 和 $\text{Match}_{\text{even}}$ 在 k 为奇数和偶数时分别产生一个 f 的解。如果 k 是奇数， $\text{odd}_f(t_{f_k})$ 等于 $\text{Cost}_{\text{OPT}}(f)$ 。因此可推出：

$$\begin{aligned} \text{odd}_f(t_{f_k}) + \text{even}_f(t_{f_k}) &= 1 + 2 * (t_{f_k} - t_{f_1}) \\ \Rightarrow s(t_{f_k}) + (t_{f_k} - t_{f_1}) &= \text{odd}_f(t_{f_k}) \\ \Rightarrow L_f &= \text{odd}_f(t_{f_k}) \end{aligned}$$

如果 k 是偶数，可以通过交换上面的 even 和 odd 符号作等价证明。证毕。 ■

3.2 2-竞争的状态化随机延迟算法

本节介绍了状态化随机延迟算法（Randomized Delaying State-based Algorithm, RDSA），通过基于状态值的随机延迟决策，该算法可高效处理请求对序列，其竞争比被证明为 2。同时提出了 RDSA 的广义版本，适用于原始输入序列，并保证了竞争比为 2。

3.2.1 状态化随机延迟算法

算法 ?? 在接收到每个请求对 p_i 时会在线计算其状态值 S_i 。若 RDSA 在空闲时接收请求对 p_i ，从 0 到 S_i 中均匀采样值 X_i ，然后将请求对 p_i 延迟 X_i 秒。如果下一对 p_{i+1} 在不晚于 $t(p_i) + X_i$ 时刻到达，则 RDSA 会在 $t(p_{i+1})$ 时刻执行 $\text{External}(p_i, p_{i+1})$ 。否则，RDSA 将在 $t(p_i) + X$ 时刻执行 $\text{Internal}(p_i)$ 。

由于随机性，无法确定算法 RDSA 在新请求对到达时处于等待或空闲状态。尽管如此，在一个包含 k 个请求对的单个独立片段 $f = (f_1, \dots, f_k)$ 并且 $f_1 \leq i \leq f_k$ 的情况下，将 RDSA 在请求对 p_i 到达时处于空闲状态的概率，定义为 P_i ，以下引理证明该概率等于状态值 $s(t(p_i))$ 。

引理 3.3： 对于任意独立片段 $f = (f_1, \dots, f_k)$ 且满足 $f_1 \leq i \leq f_k$ 中，有 $P_i = s(t(p_i))$ 。

算法 3-1: 状态化随机延迟算法 RDSA

输入: 在线输入请求对

输出: 匹配解 M

```

1: 初始化  $S_1 \leftarrow 1$ .
2: while 算法空闲时, 在时刻  $t(p_i)$  接收请求对  $p_i$  do
3:    $S_i \leftarrow \min\{1 - S_{i-1} + l_{i-1}, 1\}$ 
4:   采样  $X_i \sim \mathbf{U}(0, S_i)$ 
5:   if 在时间段  $(t(p_i), t(p_i) + X_i]$  接收请求对  $p_{i+1}$  then
6:      $S_{i+1} \leftarrow \min\{1 - S_i + l_i, 1\}$ 
7:      $M$  add  $\text{External}(p_i, p_{i+1})$  at time  $t(p_{i+1})$ 
8:   else
9:      $M$  add  $\text{Internal}(p_i)$  at time  $t(p_i) + X$ 
10: return  $M$ .

```

证明: 使用归纳法证明该引理。对第一请求对 p_{f_1} , $P_1 = s(t(p_{f_1})) = 1$ 。假设 $f_1 < i \leq f_k$ 并且 $P_{i-1} = s(t(p_{f_{i-1}}))$ 。

考虑两个连续的请求对 p_{i-1} 和 p_i 。当 p_i 到达时, RDSA 可能处于空闲状态的情况有两种可能, 如下所示。

(1) 当 p_{i-1} 到达时, RDSA 执行 $\text{External}(p_{i-2}, p_{i-1})$ 并等待。这种情况发生的概率为 $1 - P_{i-1}$ 。

(2) 在时间间隔 $[t(p_{i-1}), t(p_i))$ 内, RDSA 执行 $\text{Internal}(p_{i-1})$, 其中 X_{i-1} 的样本小于 l_{i-1} 。这种情况的概率为 $P_{i-1} \cdot \frac{l_{i-1}}{S_{i-1}} = l_{i-1}$ 。

总之, 当 p_i 到达时, RDSA 处于空闲状态的概率等于 $P_i = 1 - P_{i-1} + l_{i-1} = s(t(p_i))$ 。证毕。 ■

现在, 证明随机在线算法 RDSA 在每个独立片段的解上竞争比都为 2。

定理 3.2: 对于任意独立片段 $f = (f_1, f_2, \dots, f_k)$, 算法 RDSA 在 f 上的竞争比为 2。

证明: 设 L_f 为某个独立片段 f 的长度, t_u 为该独立片段的结束时间。令 $rdsa(t)$ 表示 RDSA 在时间 t 之前所产生的总期望成本 (包括空间和时间成本)。根据引理 3.2, 该独立片段上最优解的成本等于 L_f 。使用归纳法验证当 $f_1 \leq i \leq f_k$ 时, 有

$$rdsa(t(p_i)) = 2(t(p_i) - t(p_{f_1})) + S_i \cdot (1 - S_i) \quad (3.4)$$

基础情形: 当 $i = 1$ 时, $rdsa(t_{f_1}) = 0$ 且 $S_1 = 1$, 因此直接得到等式 3.4。

归纳假设: 当 $f_1 \leq i < f_k$, 假设等式 3.4 对于任意 i 成立, 现在证明其对于 $i + 1$ 也成立。

归纳证明: 考虑时间区间 $(t(p_i), t(p_{i+1}))$ 。相较于 $rdsa(t(p_i))$, 如果 RDSA 在 p_i 到达时处于空闲状态, 则 $rdsa(t(p_{i+1}))$ 会包含一些额外的成本, 这种情况的概率是 P_i 。

现在讨论 RDSA 为 p_i 选择的延迟时间 X_i 。如果 $X_i < l_i$, 则 RDSA 会在时间 $t(p_i) + X_i$ 执行 $\text{Internal}(p_i)$, 额外成本为 $2X_i + 1$ 。否则, $X_i \geq l_i$, 并且 RDSA 会执行 $\text{External}(p_i, p_{i+1})$, 额外成本为 $2l_i$ 。总之, 时间区间 $(t(p_i), t(p_{i+1}))$ 内产生的额外成本期望为

$$P_i \cdot \left(\int_0^{l_i} \frac{2X_i + 1}{S_i} d(X_i) + \int_{l_i}^{S_i} \frac{2l_i}{S_i} d(X_i) \right) = l_i \cdot (1 + 2S_i - l_i)$$

因此,

$$\begin{aligned} rdsa(t(p_{i+1})) &= rdsa(t(p_i)) + l_i \cdot (1 + 2S_i - l_i) \\ &= 2(t(p_i) - t(p_{f_1})) + S_i \cdot (1 - S_i) + l_i \cdot (1 + 2S_i - l_i) \\ &= 2(t(p_{i+1}) - t(p_{f_1})) + S_{i+1} \cdot (1 - S_{i+1}) \end{aligned}$$

由上可知, 等式 3.4 对于任何 $f_1 \leq i \leq f_k$ 都成立。

最后计算 $(t(p_{f_k}), t_u]$ 的期间成本, 其中只包含 $t(p_{f_k}) + X_{f_k}$ 时刻的 $\text{Internal}(p_{f_k})$ 的成本为 $P_{f_k} \cdot (2\mathbf{E}[X_{f_k}] + 1) = S_{f_k} \cdot (S_{f_k} + 1)$ 。因此, $rdsa(t_u) = rdsa(t(p_{f_k})) + S_{f_k} \cdot (S_{f_k} + 1) = 2L_f$ 。证毕。 ■

在定理 3.1 中证明了最优解的成本是所有独立片段成本之和，因此每个独立片段对于最优算法来说是独立的。同样地，在状态化随机延迟算法中，由于该算法是基于状态值 S_i 选择选项，每个独立片段都以初始状态值 1 开始。因此，上一个独立片段的解不会影响当前独立片段的解的选择。对于 RDSA 来说，所有独立片段也是独立的。

综上所述，状态化随机延迟算法的竞争比为

$$\frac{Cost_{RDSA}(R_p)}{Cost_{OPT}(R_p)} = \frac{\sum_f Cost_{RDSA}(f)}{\sum_f Cost_{OPT}(f)} = \frac{2 \sum_f Cost_{OPT}(f)}{\sum_f Cost_{OPT}(f)} = 2 \quad (3.5)$$

推论 3.1: 对任意请求对序列 R ，状态化随机延迟算法 RDSA 在 R 上的竞争比为 2。

3.2.2 广义版本的状态化随机延迟算法 RDSA

本节介绍了 RDSA 的广义版本，适用于原始序列。假设给定请求序列 R 由 n 个请求组成， r_1, \dots, r_n ，其中 $t(r_1) \leq \dots \leq t(r_n)$ 。对于任何有效算法 ALG ，每次最多有两个开放的请求。通过这种方式，可以根据开放的请求数量的奇偶性将时间轴分为“奇数区”和“偶数区”。更准确地说，对于 $1 \leq i \leq n$ ，当 i 为奇数时，区间 $[t(r_i), t(r_{i+1}))$ 被称为奇数区；当 i 为偶数时，该区间被称为偶数区^①。请注意，对于每个有效算法，包括最优算法，在每个奇数区间内都只有一个开放的请求，这导致不可避免的等待成本。在广义版本中仍然将某些请求成对考虑，并为每对请求定义到达时间。接下来按以下方式呈现 RDSA 的广义版本。

开始时，没有请求，算法处于空闲状态。

当算法处于空闲状态时，检查接下来到达的两个请求 r_i 和 r_{i+1} （其中 i 为奇数）。时间区间 $[t(r_i), t(r_{i+1}))$ 是奇数区间。在时间 $t(r_{i+1})$ ，如果 $x(r_i) = x(r_{i+1})$ ，则算法匹配这两个请求并保持空闲状态。否则，算法会将这两个请求视为到达时间为 $t(r_{i+1})$ 的一对请求，并进入等待状态。该对请求也被分配了一个类似于算法 ?? 中的状态值 S 。如果它是第一对，则其状态值为 1。否则，它的状态值等于 $\min\{1 - S' + l, 1\}$ ，其中 S' 是前一对的状态值， l 是从上一对到 $t(r_{i+1})$ 到达时间之间偶数区间的总长度。同样从 $\mathbf{U}(0, S)$ 中抽样一个延迟时间 X 。

^① 此处定义 $t(r_{n+1}) = +\infty$ 。

当算法处于等待状态时, 设 R_1 和 R_2 是具有状态值 S 的一对开放请求, 其中 $x(R_1) = a$, $x(R_2) = b$ 。类似地, 检查接下来到达的两个请求 r_i, r_{i+1} , 并且时间区间 $[t(r_i), t(r_{i+1}))$ 是奇数区间。不失一般性, 假设 $x(r_i) = a$ 。在时间 t_i , 算法匹配请求 R_1 和 r_i 。如果 $x(r_{i+1}) = a$, 则请求 R_2 和 r_{i+1} 被视为一对具有相同状态值和到达时间的请求。否则, 当 $x(r_{i+1}) = b$ 时, 算法匹配 R_2 和 r_{i+1} 并变为空闲状态。同时, 请求 r_i 和 r_{i+1} 被视为一对, 其状态值为 $\min\{1 - S' + l, 1\}$, 而到达时间为 $t(r_{i+1})$, 这与之前类似。请注意, 算法不会等待太久。更确切地说, 一旦从时间 $\min\{t(R_1), t(R_2)\}$ 开始, 偶数区间的总长度达到 X , 则请求 R_1 和 R_2 立即匹配, 并且算法变为空闲状态。

以下定理显示 RDSA 的广义版本也具有竞争比 2。

定理 3.3: 算法 RDSA 的广义版本对于在线 2-MPMD 问题是 2 竞争比的。

证明: 考虑一个奇数区间 $[t(p_i), t(p_{i+1}))$, 其中 i 是奇数。设 $\Delta t = t(p_{i+1}) - t(p_i)$ 。对于任何有效算法, 包括 RDSA 的广义版本, 在此时间间隔内恰好存在一个开放的请求。对于所有不早于 $t(p_{i+1})$ 到达的请求, 让它们提前 Δt 到达。这样, 最优解的成本减少了 Δt 。与此同时, RDSA 广义版本产生的解的成本也减少了 Δt , 即其竞争比不会增加。

现在, 任何奇数区间的长度都为零, 即输入序列已被改变为成对序列。广义版的 RDSA 与原始版本相同。根据推论 3.1, 该算法是 2-竞争的。证毕。 ■

3.3 随机近似下界的归纳证明

本节采用姚氏最小最大原则 (Yao's principle)^[1] 推导 2-MPMD 问题的随机近似下界, 并表明状态化随机延迟算法是理论最优的。通过构建一种随机序列的分布, 使得任何在线确定性算法的期望成本比至少为 2。

3.3.1 随机序列构建

构造过程中使用状态函数 $s(t)$ 来指示构造^②, 注意到状态函数 $s(t)$ 可以由折线描述, 其中每个转折点对应于序列中的一个请求对, 如图 3.1b 所示。随机匹配序列 \tilde{R} 构造过程如下:

② 这仍然是一个状态化过程, 构造的当前步骤会基于上一个步骤的状态

(1) 将第一对放置在时间 0，并将其状态值设置为 1。

(2) 对于上一次放置的请求对，假设其拥有到达时间 t 和状态值 $S = s(t)$ ，采样一个非负时间间隔 $X \sim \text{Exp}(1)$ ^③。

(3) 如果 $X < S$ ，则在时间 $t + X$ 放置一请求对，并将其状态值设为 $1 - S + X$ ；否则终止过程。

(4) 重复步骤 2 和 3，直到过程结束。

由此得到的序列实际上是一个独立片段。假设存在一个请求对到达时间为 t 、状态值为 $S = s(t)$ ，令 $P_t(x)$ 是在间隔 $(t, t + x)$ 内未进行任何放置的概率。因此，当 $x < s(t)$ 时， $P_t(x) = \int_0^x e^{-y} dy = e^{-x}$ ，与 t 无关。为简单起见，在 $x \leq s(t)$ 时使用 $P(x)$ 代替记号 $P_t(x)$ ，且 $P(x) = e^{-x}$ 。

3.3.2 竞争比分析

现在能够计算 OPT 和任何在线确定性算法 ALG 产生的期望成本。需要注意的是，构造过程是状态化的，也就是对状态值为 S 的任意时间点后构造的序列的分布只与 S 有关。首先讨论 OPT 产生的期望成本，由引理 3.2 可知这等于相应独立片段长度的期望值。

引理 3.4： 最优算法在 \tilde{R} 的期望成本至多为 1。

证明： 对于 $1 \leq i$ 且 $0 \leq s \leq 1$ ，假设第 i 请求对在时间 t 以状态值 s 放置，则 $T(i, s)$ 表示第 i 对之后的独立片段期望长度。注意到 $T(1, 1)$ 刚好是所希望的。第 $(i + 1)$ 请求对有以下两种可能情况：

(1) 第 $(i + 1)$ 请求对不存在，即序列的结束时间为 $t + s$ 。该情况的概率为 $P(s) = e^{-s}$ ，额外成本为 s ，如图 3.3a 所示。

(2) 第 $(i + 1)$ 请求对的到达时间为 $t + x$ ，其中 $0 \leq x < s$ ，其概率密度为 $P(x) = e^{-x}$ ，额外成本为 $x + T(i + 1, 1 - s + x)$ ，如图 3.3b 所示。

^③ 回想一下 $\text{Exp}(1)$ 是指数分布，其概率密度函数为 $f(x) = e^{-x}$ ，其中 $x \geq 0$ 。

因此, 对于 $1 \leq i$ 和 $0 \leq s \leq 1$,

$$T(i, s) = s \cdot e^{-s} + \int_0^s (x + T(i+1, 1-s+x)) \cdot e^{-x} dx$$

假设 $L > 0$ 是足够大的常数, 定义 $T'(i, s)$ 为辅助函数, 使得当 $i \geq L$ 时, $T'(i, s)$ 满足与 $T'(i, s) = -s^2 + 2s$ 相同的方程。那么, 对于 $i = L-1, \dots, 1$, 使用归纳法如下:

$$\begin{aligned} T'(i, s) &= s \cdot e^{-s} + \int_0^s (x + T'(i+1, 1-s+x)) \cdot e^{-x} dx \\ &= s \cdot e^{-s} + \int_0^s (x - (1-s+x)^2 + (1-s+x)) \cdot e^{-x} dx \quad (\text{归纳假设}) \\ &= -s^2 + 2s \end{aligned}$$

因此, 对于所有 $i \geq 1$ 和 $0 \leq s \leq 1$, 有 $T'(i, s) = -s^2 + 2s$ 。具体来说, $T'(1, 1) = 1$ 。现在考虑 $T(1, 1)$ 和 $T'(1, 1)$ 之间的差距。对于 $\ell \geq L$, 独立片段包含至少 ℓ 对的概率最多为 $(1 - P(1))^\ell = (1 - e^{-1})^\ell$ 。对于具有 ℓ 对且第 L 对状态值为 s 的独立片段, 在第 L 对之后实际发生的成本最多为 $\ell - L + 1$ ^④, 其中在 T' 中将其视为 $-s^2 + 2s \geq 0$ 。因此,

$$T(1, 1) \leq T'(1, 1) + \sum_{\ell=L}^{+\infty} (\ell - L + 1) \cdot (1 - e^{-1})^\ell = 1 + e^2 \cdot (1 - e^{-1})^L.$$

当 L 可以任意大时, $T(1, 1) \leq 1$ 。证毕。 ■

现在讨论任何在线确定性算法产生的期望成本。

引理 3.5: 对于任何在线确定性算法 ALG , ALG 所产生的期望成本至少为 2。

证明: 假设对于 $1 \leq i$ 和 $0 \leq s \leq 1$, 第 i 对数 p_i 在时间 t 以状态值 s 放置。令 $ALG(i, s, 0)$ 和 $ALG(i, s, 1)$ 分别表示 ALG 在接收 p_i 后处于空闲或等待状态时产生的期望成本。目标是 $ALG(1, 1, 1) \geq 2$ 。

假设当接收到配对 p_i 时, ALG 处于空闲状态。如果 p_i 是该独立片段的最后一对, 则不会产生额外成本。否则, 假设在 p_i 之后的第一对在时间 $t+x$ 放置, 其概率

④ 简单地内部匹配所有对, 这不会引起比最优算法更少的成本

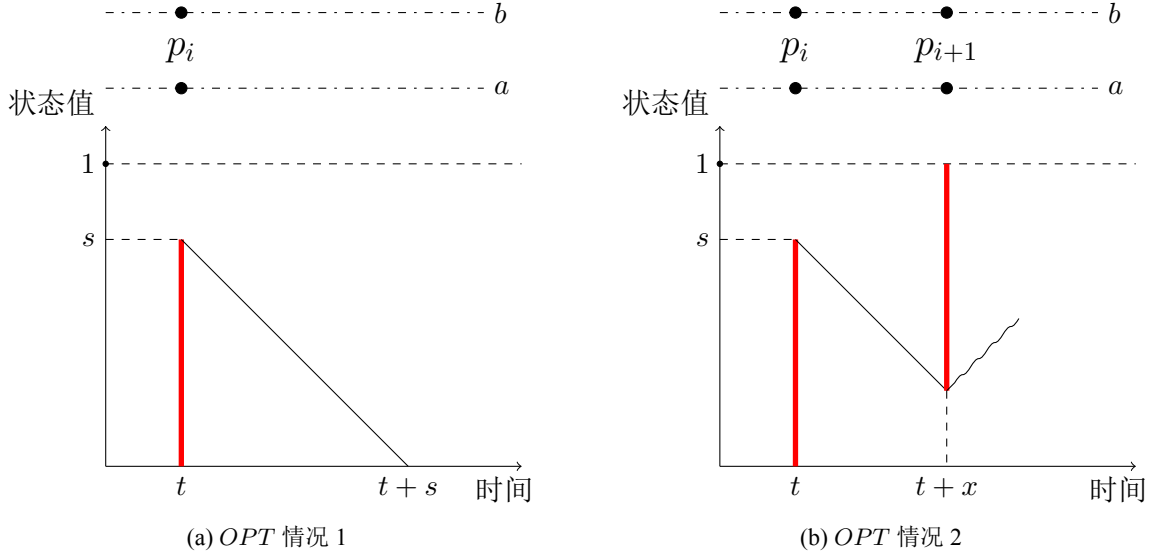


图 3.3 最优算法 *OPT* 在 \tilde{R} 上的期望成本

密度为 e^{-x} 。按照这种方式, *ALG* 在时间 $t+x$ 开始等待, 这表明

$$ALG(i, s, 0) = \int_0^s ALG(i+1, 1-s+x, 1) \cdot e^{-x} dx$$

该情况如图 3.4d 所示。

现在考虑当 *ALG* 收到对 (p_i) 时处于等待状态的情况。假设 *ALG* 将对 p_i 延迟 m 秒, 其中 $m \leq s$ 。有以下三种可能情况。

(1) 第 $(i+1)$ 对不存在, 即序列的结束时间是 $t+s$ 。此情况的概率为 $P(s) = e^{-s}$, *ALG* 在时间 $t+m$ 执行 *Internal*(p_i), 额外成本为 $2m+1$, 如图 3.4a 所示。

(2) 第 $(i+1)$ 对 p_{i+1} 的到达时间为 $t+x$, 其中 $x \leq m$, 其概率密度为 e^{-x} 。*ALG* 在时间 $t+x$ 执行 *External*(p_i, p_{i+1}) 并转为空闲状态, 额外成本为 $2x + ALG(i+1, 1-s+x, 0)$ 如图 3.4b 所示。

(3) 第 $(i+1)$ 对 p_{i+1} 的到达时间为 $t+x$, 其中 $x > m$, 其概率密度为 e^{-x} 。*ALG* 在时间 $t+m$ 执行 *Internal*(p_i) 并转为空闲状态。在时间 $t+x$, 它再次转为等待状态。此情况下的额外成本为 $2m+1 + ALG(i+1, 1-s+x, 1)$ 如图 3.4c 所示。

因此, 对于 $1 \leq i$ 且 $0 \leq s \leq 1$, 有:

$$\begin{aligned} ALG(i, s, 1) &= (2m+1) \cdot e^{-s} + \int_0^m (2x + ALG(i+1, 1-s+x, 0)) \cdot e^{-x} dx \\ &\quad + \int_m^s (2m+1 + ALG(i+1, 1-s+x, 1)) \cdot e^{-x} dx \end{aligned}$$

与引理 3.4 类似, 令 $L > 0$ 为足够大的常数, 并定义 $ALG'(i, s, 0)$ 和 $ALG'(i, s, 1)$ 作为两个辅助函数, 使得相同的方程成立且在 $i \geq L$ 和 $j \in \{0, 1\}$ 时, $ALG'(i, s, j) = j - s^2 + 2s$ 。然后, 对于 $i = L-1, \dots, 1$, 使用归纳方法如下:

$$\begin{aligned} ALG'(i, s, 0) &= \int_0^s ALG'(i+1, 1-s+x, 1) \cdot e^{-x} dx \\ &= \int_0^s (1 - (1-s+x)^2 + 2(1-s+x)) \cdot e^{-x} dx \\ &= -s^2 + 2s \\ ALG'(i, s, 1) &= (2m+1) \cdot e^{-s} + \int_0^m (2x + ALG'(i+1, 1-s+x, 0)) \cdot e^{-x} dx \\ &\quad + \int_m^s (2m+1 + ALG'(i+1, 1-s+x, 1)) \cdot e^{-x} dx \\ &= (2m+1) \cdot e^{-s} + \int_0^m (2x - (1-s+x)^2 + 2(1-s+x)) \cdot e^{-x} dx \\ &\quad + \int_m^s (2m+2 - (1-s+x)^2 + 2(1-s+x)) \cdot e^{-x} dx \\ &= 1 - s^2 + 2s \end{aligned}$$

与引理 3.4 类似, 设 $L > 0$ 为一个足够大的常数, 并定义 $ALG'(i, s, 0)$ 和 $ALG'(i, s, 1)$ 为两个辅助函数, 使得相同的方程成立, 当 $i \geq L$ 且 $j \in \{0, 1\}$ 时, 有 $ALG'(i, s, j) = j - s^2 + 2s$ 。然后, 对于 $i = L-1, \dots, 1$, 使用归纳法如下:

$$\begin{aligned} ALG'(i, s, 0) &= \int_0^s ALG'(i+1, 1-s+x, 1) \cdot e^{-x} dx \\ &= \int_0^s (1 - (1-s+x)^2 + 2(1-s+x)) \cdot e^{-x} dx \\ &= -s^2 + 2s \\ ALG'(i, s, 1) &= (2m+1) \cdot e^{-s} + \int_0^m (2x + ALG'(i+1, 1-s+x, 0)) \cdot e^{-x} dx \end{aligned}$$

$$\begin{aligned}
 & + \int_m^s (2m+1 + ALG'(i+1, 1-s+x, 1)) \cdot e^{-x} dx \\
 & = (2m+1) \cdot e^{-s} + \int_0^m (2x - (1-s+x)^2 + 2(1-s+x)) \cdot e^{-x} dx \\
 & + \int_m^s (2m+2 - (1-s+x)^2 + 2(1-s+x)) \cdot e^{-x} dx \\
 & = 1 - s^2 + 2s
 \end{aligned}$$

因此, 当 $i \geq L$ 且 $j \in \{0, 1\}$ 时, $ALG'(i, s, j) = j - s^2 + 2s$ 。特别地, $ALG'(1, 1, 1) = 2$ 。现在, 让考虑 $ALG(1, 1, 1)$ 和 $ALG'(1, 1, 1)$ 之间的差距。对于 $\ell \geq L$, 独立片段包含至少 ℓ 对的概率最多为 $(1 - P(1))^\ell = (1 - e^{-1})^\ell$ 。对于一个具有 ℓ 个请求对, 并且第 L 对具有状态值 s 的独立片段, 在第 L 对之后实际产生的成本不小于 0, 其中在 T' 中被认为是至多 $1 - s^2 + 2s \geq 2$ 。因此,

$$ALG(1, 1, 1) \geq ALG'(1, 1, 1) - \sum_{\ell=L}^{+\infty} 2(1 - e^{-1})^\ell = 2 - 2e(1 - e^{-1})^L$$

由于 L 可以任意大, $ALG(1, 1, 1) \geq 2$ 。证毕。 ■

定理 3.4: 任何随机化在线算法在 2-MPMD 上的竞争比至少为 2。

证明: 依据引理 3.4 和引理 3.5, $ALG(1, 1, 1) \geq 2$ 且 $T(1, 1) \leq 1$, 即对于 \tilde{R} , 任何在线确定性算法的期望成本比率至少为 2。基于姚氏最小最大原则^[1], 任何随机化在线算法的竞争比至少为 2。证毕。 ■

3.4 随机近似下界的演绎推导

本节将介绍解决随机理论下界中使用到的技术。

通过直觉可知需要构造具有某种分布的独立片段, 并且这个过程是一个状态化的过程。但是这个分布是未知的。由第 3.1 节可知, 最优解的产生完全基于状态值, 因此可根据生成片段时当前的状态值来决定之后的片段如何生成。即在某一个时刻, 假设该时刻的状态值为 s , 令其之后 Δt 时间内生成一个请求对的概率应该为 $f(s)\Delta t$, 也就是构造序列的生成概率密度为 $f(s)$ 。

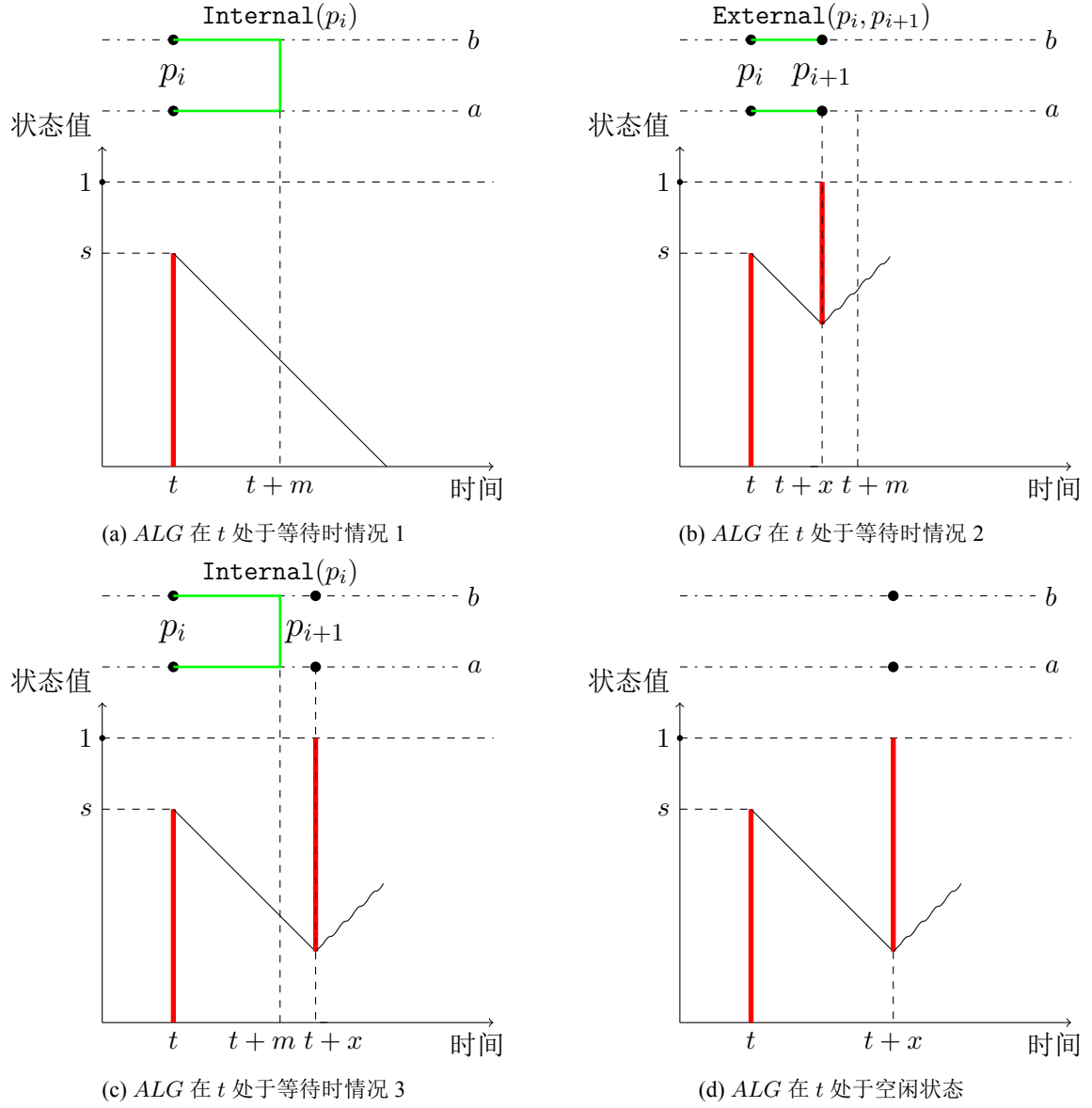


图 3.4 确定性算法 ALG 在 \tilde{R} 上的期望成本

接下来计算一个重要的条件概率 $g_s(y)$ ，该条件概率指示在一个状态值为 s 的时间点之后，一直持续到状态值为 y 之间都没有新的请求对生成的概率。

$$\begin{aligned} g_s(y - \varepsilon) &= g_s(y)(1 - f(y)\varepsilon) \text{ 且 } g_s(s) = 1 \\ \Rightarrow g'_s(y) &= \frac{g_s(y) - g_s(y - \varepsilon)}{\varepsilon} = g_s(y) * f(y) \\ \Rightarrow g'_s(y) &= g_s(y)f(y) \Rightarrow g_s(y) = e^{F(y)-F(s)} \end{aligned}$$

现在需要计算 OPT 和任何在线确定性算法 ALG 产生的期望成本。需要注意的是，构造过程是状态化的，也就是说，对于状态值为 s 的任意时间点后构造的序列的分布，只与 s 有关。因此，假设某一时间点其状态值为 s ，计算算法在这一点之后构造序列的期望成本，该值仅与状态值 s 有关。首先讨论 OPT 产生的期望成本，这等于相应独立片段长度的期望值。 $T(s)$ 考虑两种情况：

(1) 构造过程在之后没有生成请求对，独立片段增加长度 s 。这种情况的概率为 $g_s(0) = e^{F(0)-F(s)}$ 。

(2) 构造过程在之后状态 y 生成请求对，独立片段增加长度 $(s - y) + T(1 - y)$ 。这种情况的概率密度为 $g_s(y) = e^{F(y)-F(s)}$ 。

因此可得递推式：

$$T(s) = se^{F(0)-F(s)} + \int_0^s [(s - y) + T(1 - y)]e^{F(y)-F(s)}f(y)dy \quad (3.6)$$

接下来计算 ALG 的期望成本，对任意 ALG 参数化。由于 ALG 在任一时间点要么等待，要么空闲，并且在这一时间点后可以延迟等待，将延迟等待的时间参数化为 m ^⑤。可获得 ALG 对构造序列的成本计算^⑥，类似3.3节中的定义，将 ALG 等待状态的版本定义为 $A(1, s, m)$ ，空闲状态的版本定义为 $A(0, s)$ ，两者均可展开为积分递推式， $A(1, s, m)$ 考虑两种情况：

⑤ 注意到 $m > s$ 的算法可以等价转化为 $m = s$ 的算法

⑥ 注意在 $A(0, s)$ 中的 m^* 的值并不能确定

(1) 构造过程在 m 时间内没有生成请求对, ALG 会在 m 时间内部匹配, 其成本为 $1 + 2m$ 。此后 ALG 转换为空闲状态。这种情况的概率为 $g_s(s - m) = e^{-m}$ 。

(2) 构造过程在 $(0, m)$ 时间内生成请求对, 其 ALG 外部匹配成本为 $2(s - y)$, 此后 ALG 转换为空闲状态。这种情况的概率密度为 $f(y)e^{y-s}$ 。

$A(0, s)$ 只会考虑构造算法生成某对请求点之后, 算法便转换为等待状态, 其概率密度为 $e^{F(y)-F(s)}$, 因此可得递推式:

$$\begin{aligned} A(1, s, m) &= [(1 + 2m) + A(0, s - m)]e^{-m} + \int_{s-m}^s [2(s - y) + A(0, 1 - y)]f(y)e^{y-s}dy \\ A(0, s) &= \int_0^s A(1, 1 - y, m^*)e^{F(y)-F(s)}f(y)dy \end{aligned} \quad (3.7)$$

此时目标是 $\max_{f(x)} \min_m A(1, 1, m)/T(1)$, 即找到一种分布, 能够攻击所有确定性算法, 并找到其中成本比最大的一种分布。这类似于博弈论中扮演敌手, 设计一个分布去攻击所有的确定性策略。因此可假设最优分布满足混合策略纳什均衡, 即所有确定性算法都具有相同的成本。

假设 ALG 延迟等待任意值在 \tilde{R} 上的期望成本都相同, 则假设以下等式成立:

$$\forall m, \quad A(1, s, m) = a(s) \Rightarrow \frac{d}{dm}a(s) = 0 \quad (3.8)$$

$$A(0, s) = b(s) \quad (3.9)$$

让 $A(1, s, m)$ 对 m 求导可得:

$$\begin{aligned} \frac{d}{dm}A(1, s, m) &= 2me^{-m}f(s - m) - 2me^{-m} - b(-m + s)e^{-m} \\ &\quad + b(m - s + 1)e^{-m}f(s - m) - e^{-m}b'(s - m) + e^{-m} \end{aligned} \quad (3.10)$$

重整化, 令 $s - m = t$, 联立 $a'(s) = 0$ 的假设可得方程:

$$2mf(t) - 2m - b(t) + b(1 - t)f(t) - b'(t) + 1 = 0 \quad (3.11)$$

同时, 让 $A(0, s)$ 对 s 求导可得:

$$\frac{d}{dm}A(0, s) =$$

$$\begin{aligned}
 b'(s) &= a(1-s)f(s) - f(s) \int_0^s a(1-y)e^{-F(s)+F(y)}f(y) dy \\
 b'(s) &= a(1-s)f(s) - f(s)b(s)
 \end{aligned} \tag{3.12}$$

同时，可得：

$$A(1, s, 0) = a(s) = 1 + b(s) \tag{3.13}$$

联立方程 3.11、3.12和3.13可得：

$$\begin{aligned}
 b'(t) &= 2mf(t) - 2m - b(t) + b(1-t)f(t) + 1 \\
 &= 1 + b(1-t)f(t) - b(t)f(t) \\
 \Rightarrow \quad f(t)(2m + b(t)) &= 2m + b(t) \\
 \Rightarrow \quad f(t) &= 1
 \end{aligned}$$

因此可得 $f(s) = 1$ 。将其代入 $T(s), A(1, s, m)$ 可推出：

$$T(s) = -s^2 + 2s \tag{3.14}$$

$$A(1, s, m) = -s^2 + 2s + 1 \tag{3.15}$$

取 $s = 1$ ，可以得到目标值 $T(1) = 1, A(1, 1, m) = 2$ ，与上一节归纳证明的结果一致，并且发现算法 $A(1, 1, m)$ 的值与 m 无关，满足假设的等式3.8。

3.5 本章小结

4 乘车带延迟最小代价完美匹配问题及随机算法

5 总结与展望

致 谢

参考文献

- [1] Andrew C. Yao. Probabilistic computations: Toward a unified measure of complexity.
In: Proceedings of the 18th Annual Symposium on Foundations of Computer Science
(SFCS 1977), 1977: 222–227

附录 1 攻读硕士学位期间取得的研究成果

专 利

[1] 全部作者的姓名全称，本人的名字加粗. 专利题名. 专利国别，专利文献种类，专利号或申请号

标 准

[1] 全部作者的姓名全称，本人的名字加粗. 标准题名. 哪种层次的标准，发表年

科技奖励

[1] 全部作者的姓名全称，本人的名字加粗. 题目. 国家级/省部级科技类奖，获奖年

[2] 全部作者的姓名全称，本人的名字加粗. 题目. 国际/国内竞赛类奖，获奖年