学校代码: 10286
分 类 号: TP393
密 级: 公开
UDC: 004.7
学 号: 109263

東南大學

SOUTHEAST UNIVERSITY

博士学位论文

能量自供给无线设备传输速率调度算法研究

研究生姓名：单冯
导 师 姓 名：罗军舟 教授

申请学位类别 工学博士 学位授予单位 东南大学
一级学科名称 计算机科学与技术 论文答辩日期 2015 年 12 月 28 日
二级学科名称 计算机应用技术 学位授予日期
答辩委员会主席 韦志辉 教授 评 阅 人

2015 年 12 月 29 日

东南大学

博士学位论文

能量自供给无线设备传输速率调度算法研究

专 业 名 称： 计算机应用技术

研究生姓名： 单冯

导 师 姓 名： 罗军舟 教授

WIRELESS TRANSIMISSION RATE SCHEDULING FOR ENERGY HARVESTING SYSTEMS

A Dissertation submitted to

Southeast University

For the Academic Degree of Doctor of Engineering

BY

Feng Shan

Supervised by:

Prof. Junzhou Luo

School of Computer Science and Engineering

Southeast University

December 29, 2015

东南大学学位论文独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得东南大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

研究生签名：_____ 日 期：_____

东南大学学位论文使用授权声明

东南大学、中国科学技术信息研究所、国家图书馆有权保留本人所送交学位论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权东南大学研究生院办理。

研究生签名：_____ 导师签名：_____ 日 期：_____

摘　要

随着各种无线设备不断渗透到人们日常生活和社会生产中的各个方面，其能耗问题日益突出，制约了无线设备的进一步发展与普及。能量自供给技术与无线传输速率调度技术分别从“开源”与“节流”两个角度解决无线设备能耗问题。能量自供给是指无线设备从环境中采集能量，是一种新兴的能源解决方案，代表了未来能源的发展方向。无线传输速率调度是指在数据传输的过程中调整传输速率，以慢速传输减小通信能耗、以高速传输减少延迟。将两者相结合，能量自供给无线设备速率调度算法设计的总体目标是低延迟、低能耗、高传输量。然而，低延迟意味着高传输速率，那么必然带来高能耗；反之亦然。所以上述目标不可能同时实现。如何为能量自供给无线设备在能耗、延迟、传输量三者之间寻求恰当的平衡是当前急需解决的问题。

已有工作针对能量自供给设备的速率调度研究，都单纯以最大化数据传输量为目标，存在如下不足。首先，在无线传输中，通信信道往往被多个应用共享，信道上的数据报文具有各自独立的传输延时约束，但最大化传输量的目标无法满足不同应用传输延时约束。其次，传统电池供电设备速率调度问题考虑到数据报文具有独立传输延时约束，但是已有的算法无法处理后到达的数据报文具有更为紧急的延时约束的情况。最后，在移动数据传输中，由于传输距离时刻变化，传输功率与传输速率之间的关系也时刻变化，一个颇具挑战的问题是在移动传输中最大化数据传输量；目前最好的研究结果是近似算法，是否存在直接计算最大数据传输量的最优算法尚无定论。

针对上述不足，本文分别研究了如下问题。首先，在能量自供给设备速率调度问题中引入传输延时约束，该问题本质是给定数据量和延时，最小化能耗。为此，本文提出截断方法、截除能耗高峰，并通过迭代计算最优速率调度。其次，研究电池供电设备节能速率调度问题，其本质是在无能量自供给时，在数据传输量和延时约束下最小化能耗。本文创立新型数据区间关键概念，通过反复寻找密度最大数据区间计算最优解。再次，研究移动传输中能量自供给设备数据量最大化问题，其本质是给定能耗约束，最大化数据传输量。本文设计水箱技术结合动态规划进行调度，在多项式时间内得到最优解。最后，本文开发了一套调度工具原型系统、封装支持标准函数调用的动态链接库。

本文通过“开源”与“节流”两种技术解决无线设备的能耗问题。在研究中为能量自供给无线设备设计的传输速率调度算法，具有低延迟、低能耗、高传输量三者平衡的特点。由于无线设备的广泛使用，该算法推广后，可以在人们日常生活和社会生产的各个方面发挥重要作用。

关键词： 能量自供给，无线传输，速率调度，传输延时约束，移动汇聚节点

Abstract

Since wireless devices are becoming more and more popular, their energy shortage problem becomes more and more serious, and thus attracts a lot attentions. Energy harvesting technique and transmission rate scheduling are thus proposed to solve this problem from 'earning energy' and 'saving energy' aspects respectively. Energy harvesting technique collects energy from surrounding natural environments which provides extra energy, and is a promising solution to the energy shortage problem. Wireless transmission rate scheduling changes the rate during transmission, either to reduce energy consumption or to reduce transmission latency. For energy harvesting wireless devices, the general goals of rate scheduling are short transmission delay, low energy consumption and high throughput. However, short transmission delay means high transmission rate, which results in high energy consumption, vice versa. Therefore, the general goals can not be achieved simultaneously. How to balance the three goals on an energy harvesting wireless device becomes an urgent problem.

Existing research works on rate scheduling of energy harvesting wireless devices mainly focus on maximizing throughput, which have several drawbacks. First, a wireless communication channel is usually shared by multiple applications, hence packets have individual transmission delay constraints. However, the goal of maximizing throughput can not guarantee individual delay constraints to be satisfied. Second, although individual delay constraints are already considered by rate scheduling problem on battery powered wireless devices. No existing algorithm can handle the case when a latter arrived packet carries a more urgent delay constraint. Finally, in mobile transmission, the transmission distance changes over time, hence it is rather challenging to design a rate schedule that maximize throughput. An existing work gives an approximation algorithm, whether or not an optimal algorithm exists remains unknown.

To deal with the above drawbacks, this thesis proposes and solves the following three problems. First, in the rate scheduling for energy harvesting wireless devices, the transmission delay constraints are introduced. The essence behind is to minimize energy consumption while given transmission amount and allowed delay. This thesis propose the Truncation method that truncates high rate and computes the optimal rate schedule by iteration. Second, for the rate scheduling problem of battery powered wireless devices, whose essence is to minimize energy consumption while given transmission data amount and delay constraints, but without energy harvesting, this thesis presents the Densest Interval First (DIF) policy which repeatedly locates

the densest data interval and determines its optimal transmission rate. Third, in throughput maximization problem for mobile energy harvesting wireless devices, the essence is to maximize throughput while given available energy. This thesis designs the WaterTanks method together with dynamic programming to determine slot assignments and power controls, and obtain the optimal solution. Finally, this thesis develops a wireless transmission rate scheduling tool prototype. It implements all proposed algorithms, and provides users unified interface as well as standard dynamic link library.

This thesis solves the energy shortage problem by the 'earning' and 'saving' methods. The proposed rate scheduling algorithms for energy harvesting wireless devices are balanced amongst the general goals of short transmission delay, low energy consumption and high throughput. Because of the popularity of wireless devices, our proposed algorithms can play an important role in people's lives.

Keywords: Energy harvesting, wireless communications, rate scheduling, transmission delay constraint, mobile sink

目 录

摘要 . . . . . **I**

Abstract . . . . . **III**

插图目录 . . . . . **IX**

表格目录 . . . . . **XI**

第 1 章 绪论 **1**

1.1 研究背景 1

1.1.1 无线传输速率调度技术 2

1.1.2 能量自供给技术 2

1.2 研究挑战、问题与意义 3

1.2.1 研究挑战 3

1.2.2 研究问题与意义 4

1.3 相关工作和研究现状 6

1.3.1 处理器动态电压调整 (DVS) 技术 6

1.3.2 静止传输时电池供电设备节能速率调度 6

1.3.3 静止传输时能量自供给设备能量高效调度 7

1.3.4 移动传输中能量自供给设备数据量最大化 8

1.3.5 研究现状总结与研究思路 9

1.4 研究目标和内容 10

1.4.1 研究目标 10

1.4.2 研究内容 11

1.5 论文组织结构 12

第 2 章 静止传输时电池供电设备节能速率调度 **13**

2.1 引言 13

2.2 系统模型和问题定义 14

2.2.1 数据集及其延迟约束 14

2.2.2 系统模型 15

2.2.3 问题定义 16

2.3 密度最大区间优先 (DIF) 策略 16

2.3.1 最优速率策略的基本性质 16

2.3.2 数据区间和密度最大区间 18

2.3.3 密度最大区间优先（DIF）策略 20

2.4 具体数据报文最优调度 28

2.5 在线策略和仿真结果 . . . 31
2.5.1 已有的在线策略 . . . 31
2.5.2 密度指导的降温策略 . . . 32
2.5.3 仿真结果 . . . 35
2.6 本章小结 . . . 37

第 3 章 静止传输时能量自供给设备能量高效调度 . . . 39
3.1 引言 . . . 39
3.2 问题定义 . . . 40
3.3 带有公共截止时间的数据报文最优速率调度 . . . 43
3.3.1 最优速率调度的基本性质 . . . 43
3.3.2 “截断”方法 . . . 44
3.4 带有独立截止时间的数据报文最优速率调度 . . . 47
3.4.1 最优性质 . . . 47
3.4.2 截断方法的一般化 . . . 48
3.5 离散速率调度的框架 . . . 52
3.6 传输吞吐量最大化 . . . 54
3.6.1 公共截止时间的情况 . . . 54
3.6.2 独立截止时间的情况 . . . 55
3.7 动态联机算法和仿真实验 . . . 55
3.7.1 动态联机算法 . . . 56
3.7.2 仿真实验设置 . . . 57
3.7.3 仿真结果 . . . 57
3.8 详细证明和补充信息 . . . 58
3.8.1 “ZM 速率调度算法”的简短介绍 . . . 58
3.8.2 引理 3.3 的证明 . . . 60
3.8.3 定理 3.3 的证明 . . . 60
3.8.4 定理 3.4 的证明 . . . 61
3.9 本章小结 . . . 61

第 4 章 移动传输中能量自供给设备数据量最大化 . . . 63
4.1 引言 . . . 63
4.2 问题定义 . . . 64
4.3 静态离线最优算法 . . . 66
4.3.1 单传感器 DCMM 问题和凸规划建模 . . . 66
4.3.2 水箱技术 . . . 69
4.3.3 计算水位的算法 . . . 70
4.3.4 一般化的 DCMM 问题的最优解 . . . 72
4.4 动态联机分布式算法 . . . 74
4.5 算法性能评估 . . . 75
4.5.1 仿真设置 . . . 75
4.5.2 仿真结果 . . . 76
4.6 本章小结 . . . 78

第 5 章 能量自供给无线设备传输速率调度工具原型系统 79
5.1 系统简介 79
5.1.1 系统需求 79
5.1.2 系统目标与组成 80
5.2 系统设计与实现 80
5.2.1 系统总框架 80
5.2.2 系统总体流程图 81
5.2.3 静止电池调度模块设计 82
5.2.4 静止自供给调度模块设计 83
5.2.5 移动自供给调度模块设计 83
5.2.6 执行与结果展示模块设计 84
5.3 系统应用示例 85
5.3.1 系统组成与逻辑关系 85
5.3.2 系统主界面 85
5.3.3 静止场景构建 85
5.3.4 移动场景构建 86
5.3.5 参数配置 87
5.3.6 调度运算 88
5.3.7 结果展示 88
5.4 本章小结 91

第 6 章 总结与展望 93
6.1 论文总结 93
6.1.1 静止传输时电池供电设备节能速率调度（第 2 章） 93
6.1.2 静止传输时能量自供给设备能量高效调度（第 3 章） 93
6.1.3 移动传输中能量自供给设备数据量最大化（第 4 章） 93
6.1.4 传输功率调度工具原型系统（第 5 章） 94
6.2 下一步研究的方向 94

参考文献 97

致谢 101

攻读博士期间论文发表情况 103

攻读博士期间参与的科研项目 105

作者简介 107

插图目录

1.1 香农容量定理决定 $p = g(r)$ 是凹函数 . . . 2
1.2 静止传输时电池供电设备节能速率调度 . . . 4
1.3 静止传输时能量自供给设备能量高效调度 . . . 4
1.4 移动传输中能量自供给设备数据量最大化 . . . 5
1.5 研究内容交互关系 . . . 11

2.1 $n = 4$ 的数据集的例子 . . . 15
2.2 应用 DIF 策略的一个示例 . . . 23
2.3 区间 $[e_k, e_{k+1})$ 和区间 $[e_u, e_v)$ 之间的关系 . . . 27
2.4 BA-OF 策略，最优策略 OPT，以及 DGC 策略的比较 . . . 32
2.5 速率函数满足指数递减 . . . 33
2.6 最优 DIF 策略、BA-OF 策略和在线策略 DGC 的比较（第一部分） . . . 37
2.6 最优 DIF 策略、BA-OF 策略和在线策略 DGC 的比较（第二部分） . . . 38

3.1 具有四个数据报文和四个能量采集的例子 . . . 41
3.2 离散速率模型 . . . 41
3.3 “截断”方法的示例 . . . 44
3.4 一般化的截断法的示例 . . . 51
3.5 新“速率 -功率”函数 $p = G(r)$ . . . 53
3.6 离散速率模型下的截断法示例 . . . 54
3.7 能耗最小化问题的动态联机算法与静态离线算法能量消耗比较 . . . 58
3.8 吞吐量最大化问题的动态联机算法与静态离线算法数据传输量比较 . . . 59
3.9 $A(t)$ 曲线，$D_{min}(t)$ 曲线以及 $D^{zm}(t)$ 曲线 . . . 60

4.1 DCMM-1 问题示例 . . . 67
4.2 水箱技术的示例 . . . 70
4.3 排序水箱技术的示例 . . . 71
4.4 WTS,WOL 和 RLX 三个算法的性能比较 . . . 77

5.1 系统总体框架图 . . . 81
5.2 系统总体流程图 . . . 82
5.3 静止电池调度模块流程图 . . . 82
5.4 静止自供给调度模块流程图 . . . 83
5.5 水箱 (WaterTanks) 算法流程图 . . . 84

5.6 非阻塞模块调用多线程交互图 84
5.7 系统文件逻辑关系图 85
5.8 系统主界面 86
5.9 静止场景构建界面 87
5.10 静止场景构建：随机生成界面 87
5.11 移动场景构建界面 88
5.12 移动场景构建：随机生成界面 88
5.13 静止场景参数配置界面 89
5.14 移动场景参数配置界面 89
5.15 系统调度运算界面 90
5.16 静止场景调度结果展示界面 90
5.17 移动场景调度结果展示界面 91

表格目录

2.1 图 2.1 中数据集的数据区间参数表（第一部分） 19
2.2 图 2.1 中数据集的数据区间参数表（第二部分） 19
2.3 本章主要符号及其释意 21
2.4 第一次迭代后有效区间更新参数（第一部分） 23
2.5 第一次迭代后有效区间更新参数（第二部分） 25
2.6 DIF,BA-OF 和 DGC 策略能耗比较 36

4.1 拉格朗日乘子的取值范围组合表 68

第一章　绪论

1.1 研究背景

近年来，随着人类科学进步和社会经济的发展，各式各样的无线设备已经渗透到人们日常生活和社会生产中的方方面面，并呈迅猛发展的态势。在日常生活中，以智能手机作为典型代表的无线智能终端设备近年来得到相当地普及、成为生活中不可或缺的重要工具[1–3]，甚至可以说现代人完全离不开智能手机；而在社会生产中，以无线传感器作为典型代表的无线设备同样成为人们不可或缺的助手，其广泛应用于物联网、工业、农业生产等领域[4]，是提高生产效率、增加生产安全的重要保障之一。可以说，技术的发展已经使得无线设备在人们生产、生活中起到了举足轻重的作用，并且其重要性还在日益增加。

无线设备通常由电池供给能量，出于可移动性、便携性以及工艺、成本的原因，电池的物理大小受到限制，从而导致单次可以供给的电量也受到限制。因此，包括无线智能终端和无线传感器在内无线设备的能耗问题日益突出，进而影响到设备正常使用和进一步推广。全球市场资讯公司 J. D. Power and Associates 对手机消费者满意度调查发现，消费者对移动电话的电池使用时间短感到最不满意，在那些对手机电池使用时间不满意的用户中，仅有 13% 的人表示将来还会购买该手机制造商的设备[8]。在电池技术短期内不到突破的情况下，降低系统能耗成为唯一选择。事实上，电池供电能力与设备系统能耗之间的显著矛盾不仅仅表现在手机为代表的无线智能终端上，在无线传感器网络中同样存在，甚至更甚。这是因为，无线传感器网络往往部署在不易或者根本无法置换电池的环境之中，却要承担繁重的数据采集和数据传输任务，而且还必须保持尽量长时间的工作时间。做不到进一步降低无线设备的能耗，将阻碍包括无线智能终端和无线传感器在内的无线设备的良好发展势头、浪费能源、增加废气排放，影响社会经济发展。

在无线设备各个部件中，无线射频模块的能耗占比非常大。根据 2010 年经典研究工作的分析和测试，在智能移动电话中，“绝大多数能量都被消耗在 GSM 无线通信模块和显示模块上”[7]。而最新的研究和测量[5, 6] 表明，在智能移动电话上，当进行无线发射（移动网络数据收发或者 2G/3G 语音通话）时，其射频模块的能耗往往超过显示模块和处理器模块的能耗。而在无线传感器上，由于不存在显示模块，绝大多是能量都消耗在数据采集和无线通信上[4]。数据采集是无线传感器的基本使命，其能耗属于必要消耗，从而降低无线传输能耗成了研究关注的重点。综上，降低无线射频模块的能耗是进一步降低无线设备的能耗的关键环节，研究无线通信能耗优化和节能技术可以降低无

线射频模块的能耗，进而增加无线设备的工作时长、提高系统性能，还有利于节约能源成本、保护环境。

下面介绍两个技术，分别从“节流”和“开源”两个角度，解决无线设备能耗的问题。

1.1.1 无线传输速率调度技术

无线传输速率调度 (Rate Scheduling) 技术，也称无线数据传输速率自适应技术，是指在数据传输的过程中动态调整传输速率的技术。根据 IEEE802.11 系列协议规定，常见的无线局域网 (WLAN) 的物理层配置多种编码调制技术，可提供多种数据传输速率。例如，802.11b 物理层支持 4 种传输速率，802.11g 物理层支持 12 种传输速率，802.11n 物理层支持 32 种速率。此外，4G LTE 也具有可以调节的传输功率（具体的硬件功率/速率调节方法不在本文的研究范畴内）。事实上，大多数无线射频模块的速率都是可以调整和调度的。利用速率调度技术进行无线通信节能的依据是无线通信理论中的基本原理：香农容量定理 (Shannon-Hartley Theorem)。根据香农容量定理，在静止放置的一对无线收发设备之间进行数据通信时，以较低的无线传输速率传输可以节省能量，以较高的无线传输速率传输则可以缩短传输延时；在相对移动的收发设备之间通信时，近距离传输更节能，而长距离传输延迟更低。传输距离、传输能耗和传输延时之间的权衡与折衷一直是无线数据传输领域理论研究的热点[13–24, 47]。其理论背景之一是相同传输距离上数据传输速率 r 与传输功耗 p 之间具有凹函数的关系，$p = g(r)$，如图1.1所示：数据传输速率 r 与传输功耗 p 之间关系 $p = g(r)$ 是凹函数。

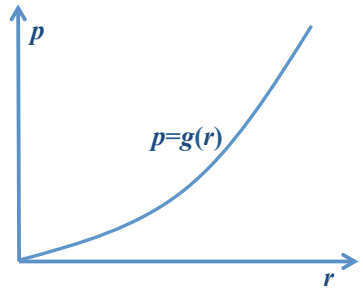


图 1.1 香农容量定理决定 $p = g(r)$ 是凹函数

1.1.2 能量自供给技术

可以看到，无线传输速率调度技术本质是传输距离、传输延迟和传输能耗之间的折衷，出发点是降低能耗，属于“节流”，为了真正解决无线设备通信能耗问题，还需要“开源”。能量自供给技术就是一项“开源”技术。

能量自供给 (Energy Harvesting) 是一种通过能量采集器，利用光伏、热电、压电、电磁等各种物理或化学效应把器件周围环境中广泛存在的太阳能、热能、机械能、风能甚至无线电信号[10] 等能量转换成可以使用的电能的技术。因为不需要消耗任何燃料或物

质，理论上可以供取之不尽的能量，因此工业界近年来对其产生了浓厚的兴趣，催生出许多日渐成熟的商业化产品。比如，许多手表生产商（西铁城、卡西欧、精工等）都推出了光动能手表；许多手机生产商 (三星、中兴、诺基亚等）都曾发布太阳能手机以减少用户的充电频率；具有能量自供给功能的无线传感器网络正开始进入爆发点，适应了供电线缆及人力不易到达的环境，将广泛用于安全监测系统、智能建筑、现代农业、嵌入式控制系统、医疗监护系统等领域[12]。在学术领域，越来越多的学者开始研究能量自供给下的通讯。比如 2013 年顶级会议 SIGCOMM 的最佳论文奖授予华盛顿大学研究团队的论文[10]，研究采集电视塔发射信号中的能量进行无电池通讯。2015 年 5 月，该团队宣布他们设计并开发出世界上第一套 WiFi 无线供电系统[11]，该系统使用商用无线发射芯片，提供正常 WiFi 信号的同时可以为无电池摄像头供电。顶级期刊 IEEE Journal on Selected Areas in Communications (JSAC) 在 2015 年 -2016 年至少有 4 期专刊专门发表能量自供给等无线通讯节能技术相关研究成果。

因此，具有能量自供给能力的无线设备在可预见的将来会成为一种潮流，针对其的无线通信能耗优化技术研究具有重要的现实意义和一定的前瞻性。

本文将关注能量自供给无线设备的传输速率调度算法设计与性能分析，而不是这两个技术本身。

1.2 研究挑战、问题与意义

1.2.1 研究挑战

无线速率调度技术和能量自供给技术的出现，提供了新的角度来研究无线通信能耗优化问题，同时也带来了新的挑战。

根据上述无线速率调度技术的介绍，其本质是传输距离、传输能耗和传输延时三者之间的折衷，本文为了简化讨论，根据传输距离是否在传输过程中变化，将研究分为静止传输和移动传输两部分。

首先，静止放置的能量自供给无线设备上的速率调度问题，不再是简单的传输延时与传输能耗之间的折衷。这是因为，能量是从环境中获取的，具有动态到达的特点，当前能量不足将无法支持低延迟传输，数据传输将只能进行高延迟低能耗的传输。只有电池电量得到大量补给，传输延时与传输能耗之间的折衷才可以顺利进行。然而，由于目前技术限制，能量自供给无线设备每次从自然界中采集的能量往往是有限的并且补给时间间隔长。在此背景下，速率调度与能量自供给高度耦合，必须相互结合，综合考虑。

其次，移动中的无线设备往往与沿途多个固定无线设备进行通信，移动设备与这些固定设备之间的距离具有先消后长、此消彼长的关系，为了更好的利用能量，必须要在多个固定设备之间适时切换。而各个固定设备的能量为自供给，其电量具有动态变化的特点，切换时要为能量较少的设备选择短距离传输、而能量较多的设备可以进行长距离传输。这两层动态因素决定了这样的切换时机的选择并不简单。

1.2.2 研究问题与意义

在静止数据传输中，为了更好地设计能量自供给无线设备的传输速率调度算法，本文首先分析和研究一个更为基础的问题：电池供电的无线设备传输速率的调度。从模型上看，电池供电是能量自供给的特殊情况。

(1) 静止传输时电池供电设备节能速率调度

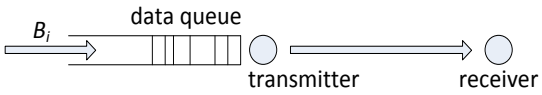


图 1.2 静止传输时电池供电设备节能速率调度

不失一般地，本文考虑一个简单、通用、基础的通信模型，如图1.2所示，一组动态到达的数据报文需要从一台设备传输到另一台设备上。由于这些数据报文来自不同的上层应用，而不同的应用对服务质量（Quality-of-Service, QoS）的要求不尽相同，因而数据报文可能具有相对独立的传输延时约束。如何进行无线传输速率调度、安排数据报文的发送顺序，使得所有数据报文都在自己的传输延时约束内完成，并且消耗的通信能量最少。这是一个基础性的问题，早在 2001 年，研究人员就在计算机网络顶级会议 INFOCOM 和次年的顶级期刊 IEEE/ACM Transactions on Networking 上提出该问题。那时，他们解决了该问题的一个特例情况：所有数据报文共享相同的截止时间。在随后的十数年里，研究人员不断冲击个难题，但一直没有完整地解决它。目前最好的结果是在 2009 年，再次发表在本领域顶级期刊 IEEE/ACM Transactions on Networking 上，他们的结果允许数据报文具有独立的传输延时约束，但是要求数据报文的截止时间顺序与到达顺序保持一致。虽然该结果更具一般性，涵盖了之前所有的结果，但是它解决的问题仍然只是原始问题的一个特例情况，无法处理后到达的数据报文具有更加紧急截止时间的情况。攻克这个开放了长达十数年之久的难题，突破思维定势、创新思考、提出最优算法是本文第一方面的研究。

(2) 静止传输时能量自供给设备能量高效调度

由于目前技术限制，能量自供给无线设备从自然界中采集的能量往往是有限的。相对于无线设备的能耗，自供给的能量较小而且补给间歇时间长，因此，在设计调度算法时，必须仍然关注能耗。

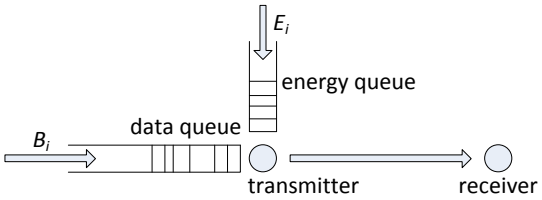


图 1.3 静止传输时能量自供给设备能量高效调度

在如图1.3所示的模型中，一台静止放置的无线能量自供给设备发送一组数据报文

给另一台无线设备。由于来自不同应用的数据报文具有不同的传输延时约束，本文的目标是设计一个传输速率调度算法，在保证数据报文传输延时约束的要求下，以最少的能量消耗完所有数据报文的发送。与已有工作相比，本文的不同之处在于，1) 允许数据报文具有独立的传输延时约束，该设定是保证应用相关服务质量 (QoS) 的迫切需要。2) 本文限制传输速率取值在一组离散数值中，因为许多现实应用中存在相应需求。该模型是上一个研究模型的扩展，是能量自供给设备上的基础性问题。然而，解决该问题具有重大意义的同时也存在巨大挑战。首先，现有的针对无线能量自供给设备的研究，多专著于最大化吞吐量，因此其中设计算法无法满足数据报文独立传输延时约束，不能提供面向应用的服务质量 (QoS) 保障。其次，在已有工作中，多数研究允许传输速率调度时连续取值，即假设连续速率模型。连续速率模型虽然可以简化分析、降低算法设计复杂度，但是很多现实的无线设备可选的调节速率是离散的，即离散速率模型。这是本文第二方面的研究。

(3) 移动传输中能量自供给设备数据量最大化

前面两个研究方面都是考虑静止放置的无线设备上的速率调度。而在现实中，大量的无线设备处于移动状态，它们的数据收发都是在移动状态下进行的。因此，研究移动的能量自供给无线设备之间进行数据传输是本文的第三个研究方面。

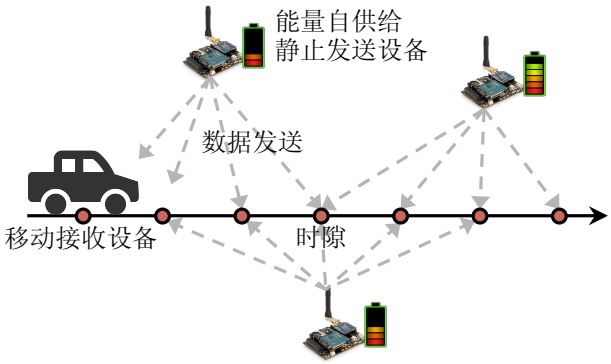


图 1.4 移动传输中能量自供给设备数据量最大化

一个典型的应用场景是在无线传感器网络中，移动汇聚节点定期在预设路径上往返移动并收集部署在路径旁边的传感器上的采集数据。由于传感器具备能量自供给能力，因此，本文的目标是为每次移动收集的过程设计时隙分配和速率调度算法，最大化移动汇聚数据收集总量。该问题的示意图如图1.4所示。相关工作中，虽然已有许多分别讨论汇聚节点移动技术[38–47] 和传感器节点能量自供给技术[48–57]，但很少同时考虑这两项技术。一个同时考虑了这两项技术的相关工作[59] 证明该问题是 NP 完全问题，进而提出了一个两步骤的近似算法。本文探寻的问题是，这个问题确实是 NP 完全问题吗？是否可以通过仔细分析和详细推演，找到一个多项式时间内完成单步骤的方案，直接得到最优解？

1.3 相关工作和研究现状

本文基于无线传输速率调度和能量自供给两种技术，研究无线通信能耗优化与节能算法。其中，利用无线传输速率调度技术进行输速率调度节能的部分内容，与利用处理器动态电压调整 (DVS) 技术进行处理器节能调度在抽象模型上具有一定的一致性。因此，本节除了从三个研究方面（静止传输时电池供电设备节能速率调度、能量自供给设备能量高效调度，和移动传输中能量自供给设备数据量最大化）对相关工作进行调研和总结外，还对利用处理器动态电压调整技术进行处理器调度节能进行了调研和总结。

1.3.1 处理器动态电压调整 (DVS) 技术

与无线传输速率调度技术相似，处理器动态电压调整 (Dynamic Voltage Scaling, DVS) 技术是在处理器运行时动态改变运行电压/频率的一种节能技术。大部分厂家的处理器都允许动态调整运行电压或者频率，例如 Intel 公司的 StrongARM 和 XScale 处理器、AMD 公司的 Athlon 处理器和 Motorola 公司的 Motorola 处理器等。DVS 技术节能的基本原理是通过降低处理器运行时的供电电压/工作频率来减少系统能量消耗。处理器的功率函数可表示为处理器速度 (电压/频率) 的凹函数。不难看出，无线传输速率调度和处理器动态电压调整两种节能技术对应的抽象理论模型是一致的：都是基于速率/速度和功率的凹函数关系。所以，本文基于速率调度技术提出的无线通信能耗优化和节能技术同样适用于基于 DVS 技术的处理器任务调度研究。从这个角度看，本文研究的意义还包括解决大规模数据处理、云计算在企业应用中能源成本巨大的问题和进一步降低智能移动终端处理器的能耗。

基于 DVS 技术的处理器任务调度节能方面的研究工作开展得比较早。早在 1995 年，Yao、Demers 和 Shenker[66] 就提出在满足所有任务在截止时间的约束下，最小化能量消耗。他们提出最优调度算法 YDS，在 $O(n^3)$ 时间内计算出最优调度，其中 n 是需要执行的任务数。Ishihara 和 Yasuura[67] 最早开始研究离散速率模型下基于 DVS 技术的任务调度。他们提出的算法解决了 CPU 只允许有两个速度的最优调度问题。Kwon 和 Kim[68] 扩展该模型，他们的算法允许处理器可以在一组离散速度上工作。之后，Li 和 Yao[69] 改进离散速率模型下的计算时间复杂性到 $O(d \log d)$，其中 d 是允许的速度个数。他们还将理想模型下最优算法的计算时间改进为 $O(n^2 \log n)$。

利用无线传输速率调度技术进行传输速率调度节能与利用处理器动态电压调整 (DVS) 技术进行处理器节能调度，在抽象模型上具有一定的一致性，但是目前尚未有学者观察到其中的联系，指出它们之间的关系。我们发现该联系，并且将处理器动态电压调整领域中的经典算法消化转接到无线传输速率调度领域。

1.3.2 静止传输时电池供电设备节能速率调度

在无线通信领域，能量自供给技术出现之前，研究人员就开始研究在仅有电池供电的无线设备上进行速率调度，以高效地使用有限的能量。那时，本文研究的静止传输时

电池供电设备节能速率调度问题，被称为带传输延时约束的节能数据报文调度问题。早在 2001 年，学者 Prabhakar, Uysal-Biyikoglu 和 El Gamal 第一次形式化地定义了带传输延时约束的节能数据报文调度问题[13]。从此开始，该问题引起了无线通信领域研究人员的广泛关注。在最初的文献[13, 14] 中，他们考虑一组动态到达的数据报文，且所有数据报文具有相同的传输截止时间。在静态离线场景下，所有数据报文的到达时间和该公共截止时间在调度之前已知，作者提出了一个全局最优调度算法，可以在保证所有数据报文在截止时间之前完成的前提下，最小化能量消耗。作者在证明过程中分析得到几个重要的最优解性质，事实表明这几个最优解性质在多个相似问题上都适用，这有力地推动了该问题的扩展。Uysal-Biyikoglu 和 El Gamal[15] 扩展了该问题，考虑多元接入信道和信道衰落。他们提出的 FlowRight 算法可以直接计算理论问题的最优解，但是前提是所有数据报文具有共同的截止时间。但是，Chen 等人[16, 17] 指出，公共截止时间的满足并不代表每个具体数据报文都具有较短的传输延时。

因此，在 2004 年，Khojastepour 和 Sabharwal[18] 开始讨论带各自传输延时约束的节能数据报文调度问题。他们提出的 water-filling 方法可以直接计算最优的速率调度。然而，该方法是适用范围有限，只能对已经在缓冲区中等待的数据报文进行调度。显然这是一个简单的情况，但却是研究带独立传输延时约束的数据报文的良好开端。

允许数据报文具有独立传输延时约束的能量高效速率调度问题也吸引了其他学者。2006 年前后，Chen 等人[16, 17] 设计并提出了处理独立传输延时的离线最优速率调度算法。该算法要求所有的数据报文必须具有相同的传输延迟约束，即所有数据报文的达到时间和截止时间的之间的间隔都一样。之后，作者进一步扩展该算法，使之应用到动态联机场景[19] 和衰减信道场景[20, 21]。虽然，Chen 等人宣称该研究结果可以推广到允许不相等的传输延迟约束的场景下，这不是一个显而易见的简单工作。

在 2009 年，Zafer 和 Modiano[23] 设计并提出一个最优算法，可以允许数据报文具有任意大小、任意到达时间和任意截止时间。其它研究工作使用的数据报文模型都是该模型的特例，因此，该研究工作涵盖了其它研究工作，是它们的统一与扩展。该算法利用累积曲线来刻画数据报文的到达和发送。其最核心观察为：代表可行解的数据发送曲线必须是在数据到达曲线和最小发送曲线之间。如果将这些曲线使用“累积数据 -时间”图来刻画，那么其思路直观、易于理解。后来，该算法被扩展到衰减信道上[24]。Zafer 和 Modiano 的该研究成果是带传输延时约束的节能数据报文调度问题的一个重大进展、被广泛认可，可以说其是该领域的里程碑。然而，他们需要做一个不得以的假设：任何较早到达的数据报文必须具有较早的截止时间。换句话讲，累积曲线的方法无法处理后到的数据报文具有更为紧急的截止时间的情况。带传输延时约束的节能数据报文调度问题仍然没有得到完整解决，是一个开放了十数年的难题。

1.3.3 静止传输时能量自供给设备能量高效调度

随着能量自供给技术的日益成熟和商业化，能量自供给无线设备的速率调度问题吸引了极大的研究兴趣[48–56]。现有研究工作按照研究目标可以分为两类：最小化传输延

时和最大化数据传输量。

Yang 和 Ulukus[48, 49] 为能量自供给无线设备设计了静态离线的速率调度算法，该算法的目标是最小化数据报文的传输延时。Tutuncuoglu 和 Yener 在电池容量受限的约束下研究进一步优化该速率调度算法。Ozel 等人[52, 53] 在衰减信道中研究传输数据块的速率调度算法，其目标是最小化其传输延时。Uysal-Biyikoglu 小组和 Ulukus 小组[54, 55] 都研究在广播通信系统中将不同的数据块同时发送给不同用户（接收方），并最小化传输延时的问题。Uysal-Biyikoglu 的小组进一步扩展该问题，研究将不同的数据队列中的数据报文同时广播给不同的接收方，研究目标也是最小化传输延时。显然，所有这些研究工作都没有考虑数据报文独立传输延时的约束。

此外，几乎所有的已有工作都假设传输速率是连续可控的。虽然这种连续速率模型可以简化算法分析、降低设计难度，然而在多数现实系统中，传输速率或者发送功率常常是离散取值的。例如，在 GSM 网络中，上行和下行传输功率的取值都必须在从 5dBm 开始以 2dBm 为步长变化到 33dBm 的离散数值中选取。在 CDMA 方案 IS-95[32] 中，传输功率的取值以 0.5dB 为一阶，上行功率在一个长 85dB 的动态区间中取，下行功率在一个长 12dB 的动态区间中取。无线局域网（WLAN）行业标准 IEEE 802.11b/g/n 分别允许数据发送和接收数据在 4/12/32 个离散速率上进行[33]。而在学术界，离散速率模型使用越来越多[34–37]。最近，Bacinoglu 和 Uysal-Biyikoglu[63] 考虑了离散速率模型，他们提出最大化吞吐量的在线调度策略。Bodin 和 Gunduz[64] 也考虑离散速率调度问题。根据他们的假设，连续速率模型的最优解可以转换为离散速率调度。

除了上述和本工作直接相关的研究成果外，本文也调研了其他的能量自供给的研究结果。Gatzianas 等人[27] 从统计学的角度研究系统效用最大化的问题。Vaze 等人[28, 29] 提出一个具有近似度的在线算法，最大化衰减信道上的速率。Zhang 等人[30] 以最大化吞吐量为目标，考虑数据报文的传递问题。Gurakan 等人[31] 在采集的能量可传输的假设下研究吞吐量最大化的问题。

1.3.4 移动传输中能量自供给设备数据量最大化

在无线传感器网络的相关研究中，许多已有研究工作讨论汇聚节点的移动性[38–47]和能量自供给无线设备上的速率调度[48–57]。但同时讨论两者的研究工作则较少[58, 59]。能量自供给无线传感器网络中移动汇聚节点数据收集最大化问题首先由 Ren, Liang 和 Xu 提出[59]，该工作和本文工作直接相关。他们设计了一个两步骤的近似算法（称之为 RLX 算法）。在第一个步骤中，他们为每一个传感器的每一个时隙静态的决定一个传输速率。然后，在第二个步骤中，这些传输速率成为优化算法的输入，为每一个时隙确定一个允许在其中进行数据传输的传感器。第二个步骤中的优化算法被映射到通用分配问题 (Generalized Assignment Problem, GAP) 上，其中一组物品（时隙）需要被分配给一组容器（传感器），每个分配都关联了一个利润（在时隙中传输的数据）和一个物品大小（消耗的能量）。因为容器有容量限制（传感器有限的可用电量），目标是找到一种分配方案可以最大化利润（传输的数据量）。因为，GAP 问题是 NP 完全问题，所以他们套

用 GAP 问题的一个已有算法，给出一个近似算法，可以保证 $2+\epsilon$ 的近似比，其中 ϵ 是 0 和 1 之间的一个任意数，$0<\epsilon<1$。下面，讨论其他的相关工作。

许多研究表明，汇聚节点的移动性可以显著提高网络寿命和增加吞吐量[38–47]。然而，因为这样的移动性，无线传感器节点和该汇聚节点的距离不断变化，影响传输功耗和传输速率的设定。在移动汇聚节点的移动数据传输中，有三类常见的信道模型：简单模型[38–41]，比例模型[42–46] 和 AWGN 模型[47]。简单模型假设传输能耗与数据传输量（或者接收量）成正比，不考虑传输速率、传输功率、收发端的距离等因素。在该模型下，Ma 和 Yang[38] 首次提出移动汇聚节点数据采集机制。他们设计了一个启发式算法，规划移动路径、均衡传感器之间流量，藉此增加网络寿命。Wang 等人[39] 在将汇聚节点移动性整合到路由算法中时，同样假设简单信道模型，其算法使网络寿命接近上界。Xing 等人使用简单模型[40] 设计并提出基于集结地的数据收集方法，将受控移动和内网数据缓冲综合考虑，增加网络吞吐量。

根据 Tse 等人的通信教科书[65]，第二个信道模型为：若传输方设定固定传输功率，则接收方接收到的功率与两者之间距离成指数关系、数据接收速率和接收功率成正比。该模型称为比例模型，同样被很多研究采用[42–46]。比如，Zhao 和 Yang[42] 在该模型下形式化定义了网络效用最大化问题。Shi 和 Hou[43, 44] 在其可保证网络寿命 $(1-\epsilon)$ 的解决方案中也假设了该模型。再一个例子是，Yun 等人[45] 在该模型下考虑以牺牲数据传输延时为代价，最大化网络寿命。文献[46] 在该模型下进一步扩展了[45] 中的研究成果。

第三个模型是经典的高斯白噪音模型 (Additive White Gaussian Noise, AWGN)，该模型更复杂，但也更广泛使用。根据该模型，数据传输速率是与接收功率成凸函数关系的，称为香农公式 (Shannon-Hartley Formula)[65]。一个重要的相关工作为 Chakrabarti 等人[47] 完成的。其中采用该模型，并假设传输功率固定，因此，其研究目标为传输速率和通信距离之间的折衷与权衡。本文采用 AWGN 信道模型，因为该模型是三中模型中最接近现实的。

1.3.5 研究现状总结与研究思路

综上所述，基于传输速率调度和能量自供给的无线能耗优化算法研究已经分别取得了一定的成果，对于本文研究工作的开展具有启发和借鉴意义，但是仍然存在一些不足。下面列举已有研究中存在的不足。

(1) 静止传输时电池供电设备节能速率调度

已有工作尚未完整地解决该问题，当数据报文具有任意截止时间时尚未设计出最优算法。目前最好的研究成果允许数据报文具有独立截止时间，但要求其与到达时间具有一致的顺序。该研究成果的核心基于数据“累积曲线”的方法，然而该方法无法处理后到的数据报文具有更加紧急的截止时间的一般性问题。为了解决该一般性问题，必须设计全新的算法。

针对以上问题，本文创立新型数据区间关键概念，提出密度最大数据区间优先策略，通过反复计算密度最大的数据区间和该区间上的传输速率，解决了该问题最具一般性的

模型。

(2) 静止传输时能量自供给设备能量高效调度

虽然能量自供给技术正在逐渐被普及到无线设备上，然而相关的研究还处于起步阶段。目前已有的研究主要关注于在给定时间内最大化数据传输量，所提出的算法无法保证具体数据报文的传输延时、并忽略能量高效利用。另外，多数研究允许传输速率任意连续取值，但是在很多实际无线设备上，可选的速率常常是离散的。因此，已有的算法无法直接应用到这些无线设备上。

针对以上问题，本文设计了“截断”方法，截除能耗高峰，并通过迭代计算最优速率调度。该截断方法得到的速率调度是连续取值的，对于离散速率的约束，本文进一步提出通用框架将使用连续速率模型的任何算法转变为使用离散速率模型的算法，并且保持原算法的最优性。

(3) 移动传输中能量自供给设备数据量最大化

在无线传感器网络中，汇聚节点的移动技术和传感器节点的能量自供给技术分别可以增加网络寿命、提高网络吞吐量。然而，现有研究工作分别研究这两项技术中其中一种，很少有研究工作将这两种技术结合起来考虑。第一个结合考虑的工作[59] 提出了能量自供给无线传感器网络中移动汇聚节点数据收集最大化问题，他们证明这是 NP 完全问题，并设计了近似算法。但是他们将原问题分为两个部分分别考虑，这本身增加了问题的复杂性，最终导致无法找到算法的最优解。

针对以上问题，本文通过深入理论分析，发现不同时隙传输功率之间的重要联系，基于此设计水箱技术，并通过与动态规划联合调度，确定时隙分配与功率控制方案，在多项式时间内得到最优解。

1.4 研究目标和内容

1.4.1 研究目标

本文利用无线传输速率调度技术和能量自供给技术，为无线设备设计能耗优化算法，在保证无线传输服务质量的前提下，减小通信能量消耗、延长设备工作时间、提高数据传输量。首先，在静止数据传输中，为电池供电无线设备设计传输速率调度算法，最小化带独立传输延时约束的数据报文的传输总能耗。其次，在静止数据传输中，为能量自供给的无线设备设计传输速率调度算法，最小化带独立传输延时约束的数据报文的传输总能耗，并且传输速率的取值限制在一组离散的数值中。若采集的能量不足以完成所有数据报文，则设计最大化数据传输量的速率调度。最后，在移动数据传输中，考虑能量自供给无线传感器网络，移动汇聚节点定期在预设路径上往返移动并收集部署在路径旁边的传感器上的采集数据，设计一套时隙分配和速率控制方案，最大化汇聚节点的数据收集量。

在系统方面，本文的目标是设计、开发一套无线设备传输速率调度工具原型系统，实现和测试三个研究问题中设计、提出的算法。

1.4.2 研究内容

为了实现上述研究目标，本博士论文拟从以下四个方面展开研究，各研究点之间的交互关系如图1.5所示。研究点①静止传输时电池供电设备节能速率调度作为一个基础研究点，其为后续两个研究点的展开奠定了基础。研究点②静止传输时能量自供给设备能量高效调度研究的模型是研究点①的直接扩展：增加了能量自供给并且保留了其他假设。研究点③移动传输中能量自供给设备数据量最大化是研究点①和②从静止传输到移动传输的演进和扩展。最后，研究点④将前面三个研究点①②③的内容整合成一个整体，开发一套工具原型系统。

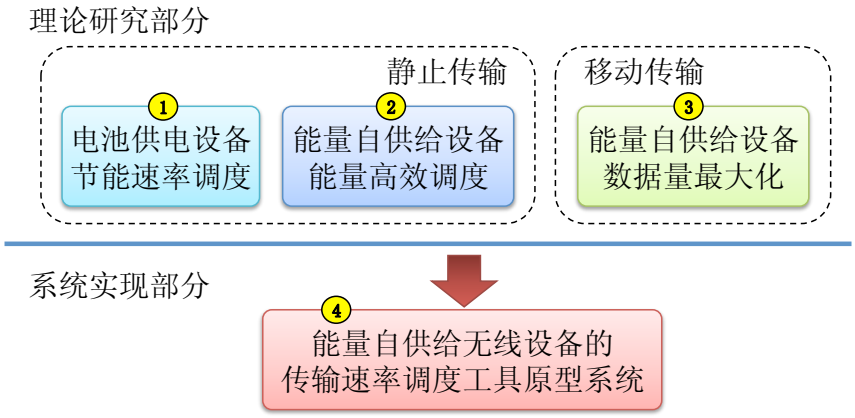


图 1.5 研究内容交互关系

① 静止传输时电池供电设备节能速率调度

给定一个传输速率可控制的无线设备和一组动态到达且具有独立传输延时约束的数据报文，研究如何设计传输速率调度算法将所有数据报文在截止时间之前传输给指定的无线接收设备并且消耗最少的能量。设计的算法要能够允许数据报文具有任意的传输截止时间，因而所有已解决的问题都是该问题的特例。

② 静止传输时能量自供给设备能量高效调度

给定一个静止放置的能量自供给无线数据传输设备和一组动态到达且具有独立传输延时约束的数据报文，研究如何设计传输速率调度算法将所有数据报文在截止时间之前传输给指定的固定放置的无线接收设备。传输速率的取值被限制在一组离散的数值中。如果从环境中采集的能量足够发送完所有数据报文，则研究最小化能耗的速率调度算法；如果能量不足够，则研究利用所有采集到的能量，最大化数据传输量。

③ 移动传输中能量自供给设备数据量最大化

研究无线传感器网络中移动汇聚节点的数据采集最大化问题。设移动汇聚节点在预设路径上移动，并且从部署在路径旁边的能量自供给无线传感器节点上接收数据。由于时间被划分为时隙，研究设计一个多项式时间内的最优算法分配这些时隙、调节传输速率以最大化数据采集量。

④ 能量自供给无线设备的传输速率调度工具原型系统

在理论研究的基础上，论文设计并开发了一套能量自供给无线设备的传输速率调度

工具原型系统，包括五大模块：输入与类型判断模块、结果展示模块，以及三大功能模块：静止电池调度模块、静止自供给调度模块与移动自供给调度模块。通过输入与类型判断模块处理输入参数和调用相应的功能模块；通过结果展示模块向用户展示算法调度结果；三大功能模块分别实现三个研究点中提出的算法。

1.5 论文组织结构

本论文共 6 章，第1章为绪论，第2章、第3章和第4章三章属于理论研究部分, 第5章为系统实现部分，第6章总结全文。具体的内容安排如下: 本章主要介绍了研究工作的背景，包括无线数据传输速率调度技术与能量自供给技术，并详细介绍了研究挑战以及重要的研究意义。在给出相关研究工作之后，本章阐述了本论文的研究目标和研究内容。最后，给出本文的组织结构安排。

第2章首先形式化地给出系统模型，并正式定义了静止传输时电池供电设备节能速率调度问题。然后提出全新概念“数据区间”，并介绍“密度最大区间优先策略”，随后给出具体数据报文的最优调度方案：使用最早截止时间优先策略具体调度数据报文，并给出该方案的正确性证明。最后设计动态联机策略，并用仿真验证其性能。

第3章首先形式化地定义了静止传输时能量自供给设备能量高效调度中的两个相关问题：能量最小化问题 (min-E 问题) 和吞吐量最大化问题 (max-T 问题)。然后，首先分两步解决了 min-E 问题：先解决带有公共截止时间的数据报文最优速率调度，再解决带有独立截止时间的数据报文最优速率调度。随后，提出了离散速率调度的框架。之后，通过凸规划解决了 max-T 问题。最后提出动态联机算法并通过仿真验证其性能。

第4章首先正式定义了移动传输中能量自供给设备数据量最大化问题，称为 DCMM 问题。然后，设计并提出静态离线 DCMM 问题的最优算法。在设计过程中，首先解决了一个简化的 DCMM 问题，进而提出了“水箱技术”；一般性的 DCMM 问题使用动态规划解决。最后，提出动态联机算法，并使用仿真验证其性能。

第5章设计并开发了一套能量自供给无线设备的传输速率调度工具原型系统。首先给出系统简介，包括系统需求、系统目标与系统组成；然后介绍系统的设计与实现，包括系统总框架图、系统总流程图、各个模块的设计；最后给出系统部署与应用示例。

第6章对本论文的研究工作进行了总结，并结合当前的研究现状和能量自供给技术发展趋势对未来无线传输速率调度领域的研究工作进行了展望。

第二章　静止传输时电池供电设备节能速率调度

给定静止放置的无线传输设备和无线接收设备，一个颇具挑战性的问题是如何设计传输速率调度算法将一组动态到达的数据报文在它们的截止时间之前传输完毕并且消耗的能量最少。十数年以来，研究人员不断冲击该问题，目前最新最好的结果可以解决允许数据报文具有任意的截止时间的问题，但是要求其与报文到达时间保持一致。本章提出密度最大区间优先策略，通过反复计算密度最大的区间和该区间上的传输速率，首次解决了该问题最具一般性的模型，涵盖了所有已有算法。本章随后证明最早截止时间优先算法可以具体调度每一个数据报文，并保证每一个数据报文都在其截止时间之前被发送完毕。最后，本章提出新颖的基于牛顿冷却定理的动态联机策略。仿真结果显示该策略的平均能耗与 DIF 策略的能耗保持在 110% 的范围内。

2.1 引言

许多无线网络基于电池工作，比如无线传感器网络、无线 Ad Hoc 网络和移动通信网络。在这些网络中，有效地使用有限的能量至关重要，因为这关系到无线设备工作时长以及无线网络的寿命。已有大量研究工作对此展开研究，设计能量高效的路由策略、数据聚集算法等等。

在无线通信中，一个通信信道常常被不同应用共享，这些应用常常具有不同实时性要求。因此，其产生的数据报文具有不同的达到时间和不同的传输延时要求。一个颇具挑战的研究任务是设计一套传输速率控制策略和调度算法，以最小的能耗将所有数据报文在截止时间之前发送完毕。根据第1.3.2节中介绍的相关工作，该问题是开放了十数年之久的难题。Zafer 和 Modiano[23] 提出的“累积曲线”方法无法解决这个更具一般性的问题。因而需要全新的方法。本文的贡献概况如下。

1. 本文解决了这个长期开放的问题。提出了“密度最大区间优先 (DIF)”策略，直接计算最优速率调度算法。
2. 密度最大区间优先策略的提出开启了解决其它相似速率调度问题的新渠道。在未来，该方法可以推广到衰减信道、能量自供给设备和多信道系统中。
3. 本文证明，一旦密度最大区间优先策略确定了时间区间上的传输速率，那么“最早截止时间优先 (EDF)”策略可以得到最优具体报文调度，该调度可以保证使用最少的能量将数据报文在截止时间之前传输完毕。

4. 受牛顿冷却定理启迪，本文也设计了一个的动态联机“密度指导冷却 (DGC)”策略。仿真显示该策略的平均能耗总是可以控制在静态离线最优策略密度最大区间优先策略的 110% 范围内。

2.2 系统模型和问题定义

本文的系统考虑单用户、点对点传输信道，在该信道上有一组 n 个数据报文需要从发送节点传输到接收节点。首先对数据集和传输延迟进行建模。

2.2.1 数据集及其延迟约束

假设 $P=\{P_1,P_2,\ldots,P_n\}$ 是一组 n 个待传输数据报文。其中，数据报文 P_i 具有任意到达时间 $a_i\geq 0$，任意截止时间 $d_i(>a_i)$，和任意大小 $B_i>0$。如果用三元组可以标记为 $P_i=(B_i,a_i,d_i)$,$1\leq i\leq n$。时间区间 $[a_i,d_i)$ 定义为 P_i 的“等待区间”。注，本章将会使用左闭右开的区间代表一个时间区间。本章允许多个数据报文在同一个时间到达，或者多个数据报文具有相同的截止时间。数据报文的大小是通过比特数（bit）衡量的，而且可以以任何速度传输。数据报文 $P_i,1\leq i\leq n$ 的传输只能在其到达时间之后开始，必须在其截止时间 d_i 之前完成。这被称之为“时序约束”。一个报文可以被分为多个部分传输，但是必须要在截止时间之前完成传输。对于静态离线问题，我们假设所有 P_i 的信息，包括 B_i,a_i,d_i，$1\leq i\leq n$，都在调度之前已知。为了简便起见，我们假设报文的达到时间已经被排序过，所以 $0\leq a_1\leq a_2\leq\ldots\leq a_n$。在调度之前，我们也将所有的截止时间排序，所以 $d_{h_1}\leq d_{h_2}\leq\ldots\leq d_{h_n}$，其中 $h_1,h_2,\ldots,h_n$ 是 $1,2,\ldots,n$ 的一个排列。令 $T=d_{h_n}$ 表示这个序列中最大的截止时间。

给定序列 $0\leq a_1\leq a_2\leq\ldots\leq a_n$ 和序列 $0\leq d_{h_1}\leq d_{h_2}\leq\ldots\leq d_{h_n}=T$，以及相应的 $B_i,1\leq i\leq n$，便完整地定义了待发送的数据集。如果一个数据报文在 t 时刻到达，那么我说一个“到达事件”在 t 时刻发生，t 时刻被称为“（事件）到达点”。同理，如果一个截止时间在 t 时刻发生，那么我们说一个“截止时间事件”发生在 t 时刻，t 时刻被称为一个“（事件）截止时间点”。因此，从 $t=0$ 到 T，总共有 $2n$ 个事件。一个到达点可能也是一个截止时间点。如果多个事件在同一时间发生，那么不同事件点的数量 m 可能会比 $2n$ 小，$m\leq 2n$。我们假设 m 个不同的事件点 $e_k,1\leq k\leq m$，已经被按照发生的先后顺序排序过，$0\leq e_1<e_2<\ldots<e_m$。显然，$e_1=a_1$，$e_m=d_{h_n}=T$。两个相邻的事件点之间的时间区间称为“时元”，并且时元 $E_k=[e_k,e_{k+1}),1\leq k\leq m-1$ 被称为第 k 个时元。图2.1显示了一个 $n=4$ 的数据集例子，其中 $P_1=(10K,2,6),P_2=(8K,3,12),P_3=(20K,5,9),P_4=(7K,7,11)$，$T=12$，$m=8$。7 个时元分别是 $E_1=[2,3),E_2=[3,5),E_3=[5,6),E_4=[6,7),E_5=[7,9),E_6=[9,11),E_7=[11,12)$。

我们用函数 ξ 将到达点或者截止时间点映射到它们在事件序列中的排序。因此，如果 $a_i=e_k$，那么 $\xi(a_i)=k$。相似的，如果 $d_{h_i}=e_k$，那么 $\xi(d_{h_i})=k$。函数 ξ 很容易获

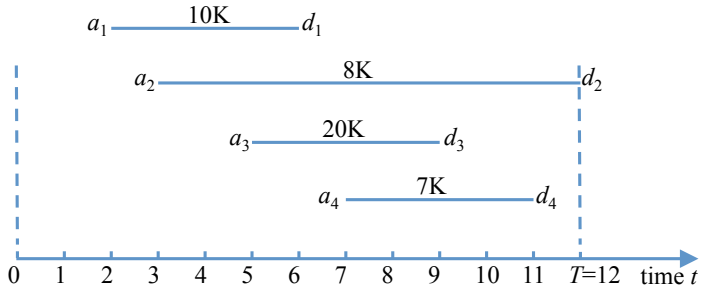


图 2.1 $n = 4$ 的数据集的例子

得并且在调度之前就已知。

显然，我们关于传输延迟限制的模型不对事件序列和数据报文大小做任何约束，所以是最一般化的模型。所有前人的数据报文模型都可以被概括成该模型的特例情况。

2.2.2 系统模型

与已有工作[13–17, 19, 20, 22, 23, 25, 49] 中的模型假设一样，我们也考虑单用户点对点传输信道并且同样假设传输节点能够自适应的改变传输速率 r。r 通过函数 $r = g(p)$ 和发送功率 p 联系起来。函数 $g(p)$ 称为“功率 -速率”函数并且满足凸函数性质[1]。

定义 2.1. “功率 -速率”函数 $r = g(p)$ 如果满足“凸性质”，那么它一定满足下列 3 个条件：

(i) $g(0) = 0$，并且当 $p \to \infty$ 时 $g(p) \to \infty$；

(ii) $g(p)$ 单调递增，并且关于 p 严格上凸；

(iii) $r = g(p)$ 连续可导。

许多实际编码解码系统都满足这样的凸性质，比如，在单用户的加性高斯白噪声信道 (additive White Gaussian Noise, AWGN) 中的最优随机编码。该编码中有 $g(p) = \frac{1}{2}\log(1 + p/N)$，其中 N 是热噪声的等级，常常被设为 $N = 1$[20, 49]。

定义 2.2. 在 t 时刻，用于传输数据报文 P_i 的速率 $r_i(t)$: $R^+ \to R^+$ 被定义成数据报文 P_i 的“报文传输速率函数”，其中 $0 \le t < T, 1 \le i \le n$。

根据时序约束，该函数必须满足下列公式。

$$\int_0^T r_i(t)\,\mathrm{d}t = \int_{a_i}^{d_i} r_i(t)\,\mathrm{d}t = B_i, \quad 1 \le i \le n. \tag{2.1}$$

定义 2.3. 给定 n 个报文传输速率函数，$r_i(t)$，$0 \le t < T, 1 \le i \le n$，所有报文传输速率函数之和 $r(t) = \sum_{i=1}^{n} r_i(t), 0 \le t < T$，被定义成“总速率函数”。

[1]本论文规定凸函数与凹函数都具有凸函数性质。基于此，在不产生原则错误的情况下，我们混用关键词“凸函数”与“凹函数”。

根据传输模型，我们有 $r(t) = g(p(r)), 0 \leq t < T$。显然，一个总速率函数 $r(t), 0 \leq t < T$，唯一定义了时间区间 $[0, T)$ 上的传输速率策略。给定一个总速率函数 $r(t)$，对应的总能量消耗可以通过下面的积分计算[23]。

$$E = \int_0^T g^{-1}(r(t))\,\mathrm{d}t \tag{2.2}$$

2.2.3 问题定义

给定一个上述数据集，我们的目标是寻找满足时序约束的最优解。我们首先定义可行解。

定义 2.4 (可行解)**.** 给定包括 n 个数据报文的数据集 $P_i = (B_i, a_i, d_i)$，$1 \leq i \leq n$ 和上述系统模型，报文传输速率函数集合 $S\{r_i(t), 0 \leq t < T | 1 \leq i \leq n\}$ 被称为可行解，如果其中每一个 $r_i(t), 1 \leq i \leq n$ 都满足公式 (2.1)。

定义 2.5 (最优解)**.** 给定包括 n 个数据报文的数据集 $P_i = (B_i, a_i, d_i)$，$1 \leq i \leq n$ 和上述系统模型，可行解 S 对应的总速率 $r(t), 0 \leq t < T$ 被称为最优解，如果其使得公式 (2.2) 中定义的能量消耗最小。对应的总速率函数 $r(t)$ 称为“最优总速率函数”，标记为 $r^{opt}(t), 0 \leq t < T$。

现在，我们要研究的问题可以定义为如下形式。

定义 2.6. 静止传输时电池供电设备节能速率调度问题，或者离线能量高效的数据报文传输调度问题，是为任意 n 个数据报文 $P_i = (B_i, a_i, d_i)$，$1 \leq i \leq n$ 寻找最优解 $S = \{r_i(t), 0 \leq t < T | 1 \leq i \leq n\}$。

2.3 密度最大区间优先 (DIF) 策略

在我们介绍“密度最大区间优先 (DIF)”策略之前，我们首先需要讨论最优传输速率策略 $r^{opt}(t)$ 必须具有的一些基本性质。在如下给出的最优性质中，一部分是相关工作中已有的，另一些是新的。

2.3.1 最优速率策略的基本性质

显而易见[49]，鉴于“功率 -速率”函数的凸性质，任何一个时元 $[e_k, e_{k+1})$ 中都只可能有一个传输速率，$1 \leq k \leq m-1$。假设存在两个不同的速率 $r_1 < r_2$，我们总能在 r_1 与 r_1 之间找到一个新速率 r，使得传输相同的数据量却消耗更少的能量。该方法被称为“均衡”。因此，计算最优速率策略其实就是为每一个时元找一个速率使得总能耗最小而且所有的截止时间都得到满足。“均衡”是设计最优速率控制策略中的最重要的概念。建议读者参照相关工作[49] 熟悉该概念。

由于存在时序约束，从 $t=0$ 到任意时间 $t>0$，一方面总数据传输量不能够超过已到达数据总量，另一方面又不能小于已过截止时间的数据总量。因此，我们有下面的不等式。

$$\sum_{d_i \le t} B_i \le \int_0^t r^{opt}(x)\,\mathrm{d}x \le \sum_{a_i < t} B_i, \quad \forall t \in [0,T). \tag{2.3}$$

现在，下列两个引理介绍了其它最优速率策略的性质。

引理 2.1. 在最优总速率函数 $r^{opt}(t)$ 中，速率的增加只可能在到达点上，速率的减小只可能在截止时间点上。

证明. 我们已经解释了在最优速率策略 $r^{opt}(t)$ 中，速率变化只可能在事件点上。现在，为了引出反证法中的矛盾，我们假设最优速率策略在事件点 e_j 上增加了速率，但是却没有报文在 e_j 上到达。显然，$j>1$，因为 $e_1=a_1$ 就是一个到达点。假设该策略在时元 $[e_{j-1},e_j)$ 上使用速率 r_1，在时元 $[e_j,e_{j+1})$ 上使用速率 r_2，并且 $r_2>r_1$。那么，我们可以均衡它们，而使用速率 $r, r_1<r<r_2$ 在时间区间 $[e_{j-1},e_{j+1})$ 上传输相同的数据量却使用更少的能量。也就是说，一些原先在时元 $[e_j,e_{j+1})$ 上传输的数据量被挪动到时元 $[e_{j-1},e_j)$ 上传输了。这样做，没有错过任何截止时间，因为我们提前传输了一些数据量；也没有数据在到达之前传输，因为在 e_j 上没有报文到达，在 (e_{j-1},e_j) 上也没有。这样的改变没有违反时序约束，但是却减少了能量消耗。这和 $r^{opt}(t)$ 是最优解矛盾。

相似的，为了引出反证法中的矛盾，假设该速率策略在时元 $[e_{j-1},e_j)$ 上使用速率 r_1，但是在时元 $[e_j,e_{j+1})$ 上减少速率到 $r_2(<r_1)$，并且没有截止时间事件在 e_i 上发生，那么我们可以均衡它们使之用速率 $r, r_1<r<r_2$ 在时间区间 $[e_{j-1},e_{j+1})$ 上传输相同的数据量而使用更少的能量。也就是说，一些原先在时元 $[e_{j-1},e_j)$ 上传输的数据量被挪动到时元 $[e_j,e_{j+1})$ 上传输了。这样做，既没有数据在到达之前传输，也没有数据错过截止时间，因为它们的截止时间都在 e_{j+1} 或者之后。这样的改变没有违反时序约束，但是却减少了能量消耗。这也是一个矛盾。 □

这里我们要感谢已有工作[23] 在相对简单的报文模型上给出了和该引理相似的引理（未证明）。但是，其正确性在更一般性的模型下并不是显而易见的，所以我们给出正式的证明帮助读者理解。

引理 2.2. 设 $r_k(t)$ 是最优速率策略 $r^{opt}(t)$ 中用于传输 $P_i=(B_k,a_k,d_k)$ 的部分。令 H 是时间区间 $[a_k,d_k)$ 包含的所有时元，令 $H'\subseteq H$ 是 $r_k(t)>0$ 的时元的集合。则我们有下面两条：

(1) H' 中的任何时元上的最优速率策略 $r^{opt(t)}$ 都相等，等于速率 r。

(2) $H-H'$ 中任何时元上的最优速率策略 $r^{opt(t)}$ 都大约或者等于 r。

证明. 我们用反证法证明 (1)。假设 H' 中的时元 1 和时元 2 分别使用两个不同的最优速率，$r_1<r_2$。那么，我们使用“均衡”的方法将在时元 2 中传输的一些 P_k 的数据量挪到时元 1 中传输。通过这样做，我们使两个时元中的速率一样并且降低了总的能量消

耗，这和速率 $r^{opt}(t)$ 是最优解矛盾。接着，我们证明 (2)。假设，有一个在集合 $H-H'$ 中的时元 x 其中的速率 $r^{opt}(t)<r$，那么我们可以将时元 $y\in H'$ 上传输的一部分 P_k 的数据量挪动到时元 x 上传输。通过这样做，我们降低了时元 y 上的速率，增加了时元 x 上的速率，还减小了总的能量消耗。这和速率 $r^{opt}(t)$ 是最优解矛盾。因此，(2) 也是正确的。 □

2.3.2 数据区间和密度最大区间

在本节中，我们分别介绍两个关键概念：“数据区间”和“密度最大区间”。为了更好地理解它们，让我们首先简要的介绍 DIF 策略。DIF 策略的核心是迭代的，每一次迭代，要做下列三件事：

(1) 从未分配的数据报文中选择一个子集合 S，同时从可用时元中选择一个子集合 E。

(2) 为 E 中所有时元指定一个统一的传输速率 r，S 中的数据报文都将在 E 中将以该速率传输。

(3) 将 S 中的数据报文标记为“已分配”，将 E 中的时元标记为（对未分配数据报文）“不可用”。

DIF 策略的细节将在下一节讨论。我们现在定义“数据区间”和“密度最大（数据）区间”。

定义 2.7. 给定数据集 $P_k=(B_k,a_k,d_k),1\le k\le n$，“数据区间 $[i,j)$”是指从到达时间点 a_i 到截止时间点 d_{h_j} 的时间区间。也就是说当 $a_i<d_{h_j},1\le i,j\le n$ 时，$I[i,j)=[a_i,d_{h_j})$。

多个数据报文可能共享相同的到达点或者共同的截止时间点，所以我们可能会有相同的数据区间，即 $I[i,j)=I[u,v),i\neq u$ 并且/或者 $j\neq v$。比如，如果 $a_1=5,a_2=7,a_3=7,d_1=9,d_2=9,d_3=12$，那么我们有 $d_{h_1}=9,d_{h_2}=9,d_{h_3}=12$。所有的时间区间分别是：

$I[1,1)=[5,9),I[1,2)=[5,9),I[1,1)=[5,12),$
$I[2,1)=[7,9),I[2,2)=[7,9),I[2,1)=[7,12),$
$I[3,1)=[7,9),I[3,2)=[7,9),I[3,1)=[7,12).$

这些相同区间并不会影响到我们的算法，因为一旦任何一个区间被在某个迭代选中，那么其它的相同区间在更新后数据报文集合是空的。一旦数据报文集合变成空的，该数据区间就变失效了。细节在下一小节讨论。

定义 2.8. 我们为每一个数据区间 $I[i,j)$ 定义如下四个变量参数，它们会在迭代过程中动态变化。

(1) “数据集合 $S[i,j]$”是一个数据报文的集合，这些数据报文的等待区间包含在 $I[i,j)$ 中，并且目前没有被分配时元。初始时，$S[i,j]=\{P_k|[a_k,d_k)\subseteq I[i,j)\}$.

(2) “数据负载 $B[i,j]$” 是所有被包含在 $S[i,j]$ 中的所有数据总和，即 $B[i,j] = \sum_{P_k \in S[i,j]} B_k$。

(3) “可用时间长度 $L[i,j]$” 是 $I[i,j)$ 中所有可用时元的总时间长度。初始时 $L[i,j] = (d_{h_j} - a_i)$。

(4) “密度 $D[i,j]$” 在 $L[i,j] > 0$ 时定义成 $D[i,j] = \frac{B[i,j]}{L[i,j]}$，否则 $D[i,j] = 0$。

注，左闭右开区间只表示时间区间，而不是参数的形式。让我们看一个例子。对于图2.1中给出的数据集，表格2.1 和表格2.2中给出了各数据区间的参数初始值。

表 2.1 图2.1中数据集的数据区间参数表（第一部分）

$S[i,j]$	$d_{h_1} = d_1$ 6	$d_{h_2} = d_3$ 9	$d_{h_3} = d_4$ 11	$d_{h_4} = d_2$ 12
$a_2 = 3$	P_1(10k,2,6)	P_1(10k,2,6) P_3(20k,5,9)	P_1(10k,2,6) P_3(20k,5,9) P_4(7k,7,11)	P_1(10k,2,6) P_3(20k,5,9) P_4(7k,7,11) P_2(8k,3,12)
$a_2 = 3$	$\varnothing$	P_3(20k,5,9)	P_3(20k,5,9) P_4(7k,7,11)	P_3(20k,5,9) P_4(7k,7,11) P_2(8k,3,12)
$a_3 = 5$	$\varnothing$	P_3(20k,5,9)	P_3(20k,5,9) P_4(7k,7,11)	P_3(20k,5,9) P_4(7k,7,11)
$a_4 = 7$	Undefined	$\varnothing$	P_4(7k,7,11)	P_4(7k,7,11)

表 2.2 图2.1中数据集的数据区间参数表（第二部分）

$B[i,j]$ $L[i,j]$ $D[i,j]$	$d_{h_1} = d_1$ 6	$d_{h_2} = d_3$ 9	$d_{h_3} = d_4$ 11	$d_{h_4} = d_2$ 12
$a_1 = 2$	B=10k L=6-2=4 D=10/4=2.5	B=30k L=9-2=7 D=30/7=4.286	B=37k L=11-2=9 D=37/9=4.111	B=45k L=12-2=10 D=45/10=4.5
$a_2 = 3$	B=0k L=6-3=3 D=0	B=20k L=9-3=6 D=20/6=3.333	B=27k L=11-3=8 D=27/8=3.375	B=35k L=12-3=9 D=35/9=3.889
$a_3 = 5$	B=0k L=6-5=1 D=0	B=20k L=9-5=4 D=20/4=5	B=27k L=11-5=6 D=27/6=4.5	B=35k L=12-5=7 D=35/7=3.857
$a_4 = 7$	Undefined	B=0k L=9-7=2 D=0	B=7k L=11-7=4 D=7/4=1.75	B=7k L=12-7=5 D=7/5=1.4

定理 2.1 (基本最大密度区间定理)**.** 给定一个数据集 $P = \{P_i(B_i, a_i, d_i)|1 \leq i \leq n\}$，如果数据区间 $I[i,j)$ 上的密度 $D[i,j]$ 是所有数据区间中最大的，那么 $I[i,j)$ 被称作密度最大区间，并且有下列陈述。

(1) 任何最优速率策略都必须为时间区间 $[a_i, d_{h_j})$ 中的每一个时元设定速率 $r = D[i,j]$ 。

(2) 任何最优解都必须在时间区间 $[a_i, d_{h_j})$ 中将数据报文集合 $S[i,j]$ 排他地传输完。

证明. 我们用反证法证明 (1)。假设最优速率策略在时间区间 $[a_i, d_{h_j})$ 上使用的速率是 $r^{opt}(t)$，而在某个时元中，$r^{opt}(t) \neq D[i,j] = r$。我们断言至少存在在一个时元 $[e_k, e_{k+1}) \subseteq [a_i, d_{h_j})$，其中有 $r^{opt}(t)$ 等于 r_{ek} 并且 $r_{ek} > r$。否则的话，在整个时间区间 $[a_i, d_{h_j})$ 上都有 $r^{opt}(t) \leq r$，但某些时元上有 $r^{opt}(t) < r$，意味着 $\int_{a_i}^{d_{h_j}} r^{opt}(x)\,\mathrm{d}x < r \times (d_{h_j} - a_i) = B[i,j]$，某些数据报文会错过他们的截止时间。所以，我们可以假设时元 $[e_k, e_{k+1}) \subseteq [a_i, d_{h_j})$ 中有 $r^{opt}(t) = r_{ek} > r$。现在扩展 $[e_k, e_{k+1})$ 为更大的区间 $[e_u, e_v)$。令 $[e_u, e_v) \supset [e_k, e_{k+1})$ 是最大的时间区间，其中每一个时元上都有速率 $r^{opt}(t) \geq r$。注，$[e_u, e_v)$ 并不一定包含 $[a_i, d_{h_j})$，反之亦然。

显然，$r^{opt}(t)$ 在 e_u 上增大了速率而在 e_v 上降低了速率，要不然的话，我们可以将该时间区间扩得更大。根据引理2.1，e_u 必定是一个到达点而 e_v 必定是一个截止时间点。所以，$[e_u, e_v)$ 是一个数据区间而且它的密度不会比 r 更大，因为 r 是所有区间上最大的密度。我们有 $r^{opt}(t) \geq r$ 对于整个区间 $[e_u, e_v)$ 成立并且 $r^{opt}(t) > r$ 对某些时元 $[e_k, e_{k+1}) \subseteq [a_i, d_{h_j})$ 成立，因此 $\int_{e_u}^{e_v} r^{opt}(x)\,\mathrm{d}x > \int_{e_u}^{e_v} r\,\mathrm{d}x < B[u,v] = \sum_{[a_k,d_k)\subseteq[e_u,e_u)} B_k$，最优策略必须要在时间区间 $[e_u, e_v)$ 中传输数据量超过 $B[u,v]$。显然，$[e_u, e_v) \neq [0,T)$，否则的话最优策略就会传输比所有数据报文总和更多的数据量。因此，最优策略必定传输了至少一个在 e_u 之前到达或者在 e_v 之后截止的数据报文 P_x。该就和引理 2.2矛盾。(1) 得证。

陈述（2）直接可以从陈述（1）得到，因为速率 $r = D[i,j]$ 刚好够将集合 $S[i,j]$ 中的所有数据报文完成。 □

如果 $I[i,j) = [a_1, T)$，那么我们的任务业已完成。否则的话，我们需要继续寻找一个最优速率策略在余下的区间 $([0,T) - I[i,j)) = [0, a_i) \cup [d_{h_j}, T)$ 上传输剩余数据报文集合 $P' = P - S[i,j]$。可以看出，只要我们把剩余的数据报文集合看成是新的数据报文集合，同样的问题又重现了。唯一的不同在于，不允许使用 $I[i,j)$ 中时元，因为他们都已经被分配给 $S[i,j]$ 了。

2.3.3 密度最大区间优先（DIF）策略

正如上一节中指出的，DIF 策略通过迭代计算传输速率。在每一次迭代中，具体完成如下工作：

(1) 为每一个区间 $I[i,j)$ 重新计算数据集 $S[i,j]$，$1 \leq i,j \leq n$。数据集由所有等待区间被包含在 $[a_i, d_{h_j})$，并且当前仍为“未分配”的数据报文组成。

(2) 为每一个区间 $I[i,j)$ 重新计算数据负载 $B[i,j]$，可用时间长度 $L[i,j]$，和密度 $D[i,j]$，$1 \le i,j \le n$。

(3) 寻找密度最大的区间 $I[i,j)$，并且为包含在 $I[i,j)$ 中的每一个可用时元设定统一速率 $r = D[i,j]$，这些可用时元将被用来发送 $S[i,j]$。

(4) 标记所有在 $I[i,j)$ 中的时元为（对下轮迭代）“不可用”。标记所有 $S[i,j]$ 中的数据报文为“已分配”。

假设时间区间 $[0,a_1)$ 中 $r = 0$。

定义 2.9. 如果一个数据区间 $I[i,j)$ 具有非空的数据集 $S[i,j]$，那么它被定义为“有效”，反之为“无效”。

在 DIF 策略执行期间，数据报文总是从“未分配”变成“已分配”；时元总是从“可用”变成“不可用”；数据区间总是从“有效”变成“无效”。DIF 算法在所有数据区间都变成“无效”时停止。

我们用 $M[i,j] = 1$ 和 $M[i,j] = 0$ 分别标记数据区间 $I[i,j)$ 是有效或者无效。我们用 $R[i] = 0$，标记数据报文 P_i 已经被分配，$R[i] = 0$ 标记未分配，$1 \le i \le n$。我们用 $E[k] = 0$ 标记第 k 个时元已经被分配给某个数据报文而变得不可用，用 $E[k] = 1$ 标记它还对余下的数据报文可用，$1 \le k \le m-1$。$R[i]$ 和 $E[k]$ 在每一次迭代之后都需要更新。初始时，$R[i] = 1, 1 \le i \le n, E[k] = 1, 1 \le k \le m-1$。本章中使用的主要符号在表2.3中做了总结，方便读者参考。

表 2.3 本章主要符号及其释意

符号	释意
P_k	第 k 个数据报文，$P_k = (B_k, a_k, d_k)$，具有大小 B_k，到达时间 a_k 和截止时间 d_k
E_k	第 k 个时元，$E_k = [e_k, e_{k+1})$，在事件点 e_k 和 e_{k+1} 之间的时间区间
$R[k]$	=0, 数据报文 P_k“已分配”时元，或者 =1，数据报文 P_k“未分配”任何时元
$E[k]$	=0, 时元 E_k“不可用”，或者 =1，时元 E_k 对余下数据报文“可用”
$I[i,j)$	$= [a_i, d_{h_j})$，从到达时间 a_i 到截止时间 d_{h_j} 的时间区间
$S[i,j]$	等待区间包含在 $I[i,j)$ 中“未分配”的数据报文集合
$B[i,j]$	$S[i,j]$ 中的总数据量
$L[i,j]$	$I[i,j)$ 中的所有当前可用时元的总长度
$D[i,j]$	$= B[i,j]/L[i,j]$，$I[i,j)$ 的数据密度
$M[i,j]$	=0, 数据区间 $I[i,j)$ 为“无效”，或者 =1，数据区间 $I[i,j)$ 为“有效”
$T[i,j]$	$I[i,j)$ 中的“可用”时元的集合

如果时间区间 $I[i,j)$ 在某次迭代中被确认为密度最大区间，那么 Availability-Update 中的伪代码可以更新 $R[i], 1 \le i \le n$ 和 $E[k], 1 \le k \le m-1$。基本上讲，我们会将那些

等待区间包含在 $[a_i, d_{h_j})$ 中的数据报文标记为“已分配”，将 $[a_i, d_{h_j})$ 中的时元都标记为“不可用”。

Algorithm 1 *Availability-Update*($E[], R[], I[i,j]$)

```
1: for k ← 1 to m − 1 do
2:    if [e_k, e_{k+1}) ⊆ [a_i, d_{h_j}) then
3:       E[k] ← 0                          // 有可能本来就有 E[k] = 0
4:    end if
5: end for
6: for i ← 1 to n do
7:    if [a_i, d_j) ⊆ [a_i, d_{h_j}) then
8:       R[i] ← 0                          // 有可能本来就有 R[k] = 0
9:    end if
10: end for
```

显然，Availability-Update 的复杂度是 $O(n)$。让我们继续看图 2.1中的例子。从表2.1和表2.2中，我们发现密度最大的区间是 $I[3,2) = [5,9)$，其密度是 5。因此，我们指定数据报文 P_3 用速率 $r = 5$ 传输，同时我们将时元 $E_3 = [5,6), E_4 = [6,7)$ 和 $E_5 = [7,9)$ 分配给数据报文 P_3。在完成分配之后，我们设置 $R[3] = 0$ 即意味着 P_3 已经被指定了速率，然后，继续为余下数据报文集合执行。我们也设置 $E[3] = E[4] = E[5] = 0$，即意味着时元 3、时元 4 和时元 5 对余下的数据报文不可用。最后，我们设置 $M[3,2] = 0$，即意味着区间 $I[3,2)$ 变成无效了。一旦我们发现一个区间的数据报文集合是空的或者无定义，我们就将该区间设置为无效。如图2.2所示，有三个数据报文余下。因为它们不可以使用阴影区域，所以 P_1 必须在时间 5 之前完成，尽管它的截止时间是 6；P_4 要在时间 9 之后开始，虽然它在 7 时刻已经到达；P_2 只能使用时间区间 $[3,5)$ 和 $[9,12)$，以至于它被分割成两段分别传输。

图2.2表示在图2.1中的数据报文集合上应用 DIF 策略，第一轮迭代之后的剩余数据报文和可用时元的一个示例。阴影里的时元对剩余数据报文不可用。从图2.2中，我们可以得到的总结是，每经过一次迭代，我们就需要为每一个有效的数据区间重新计算它们的参数，这是因为每轮之后剩余未分配的数据报文集合和可用时元集合都发生了改变。我们按照这样的顺序做更新，$I[i,1), I[i,2), \ldots, I[i,n), i = 1,2,\ldots,n$。这样的顺序允许我们使用一个有效的贪心策略从 $I[i,j)$ 变化到 $I[i,j+1)$。

因为 $I[i,j) = [a_i, d_{h_j})$ 以及 $I[i,j+1) = [a_i, d_{h_{j+1}})$，我们有 $I[i,j) \subseteq I[i,j+1)$。因此，我们有下列两个序列的“包含”关系：

$$I[i,1) \subseteq I[i,2) \subseteq \ldots \subseteq I[i,j) \subseteq I[i,j+1) \subseteq \ldots \subseteq I[i,n) \tag{2.4}$$

和

$$S[i,1) \subseteq S[i,2) \subseteq \ldots \subseteq S[i,j) \subseteq S[i,j+1) \subseteq \ldots \subseteq S[i,n) \tag{2.5}$$

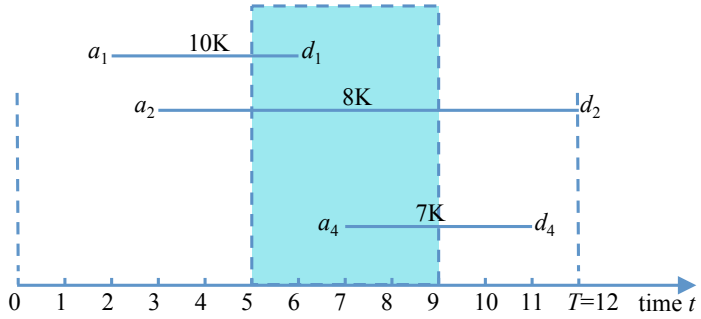


图 2.2 应用 DIF 策略的一个示例

此外，我们也有下面的一个序列的“小于等于”关系：

$$L[i,1) \leq L[i,2) \leq \ldots \leq L[i,j) \leq L[i,j+1) \leq \ldots \leq L[i,n) \tag{2.6}$$

基于如上性质，所有区间的参数都可以按照程序 Density-Update 快速有效的计算得到。如何集合 $S[i,j]$ 不是空集，那么程序就将标记 $M[i,j]=1$，否则标记为 $M[i,j]=0$。然而，即使标记是 $M[i,j]=0$，仍然计算区间 $I[i,j)$ 的参数 $S[i,j]$ 和参数 $B[i,j]$，因为我需要它们扩展整个序列，从 $S[i,1]$ 到 $S[i,n]$，从 $B[i,1]$ 到 $B[i,n]$。显而易见，未定义的区间总是出现在序列 (2.4) 的前端。当 $I[i,j)$ 是未定义的时候，我们设定 $L[i,j]=0$ 和 $D[i,j]=0$。

显然，程序“Density-Update”可以自然地处理有重复区间的情况。让我们继续看图2.1中的例子。对第一轮迭代之后余下的数据报文集合，应用 Density-Update，所得的所有有效区间的的密度都显示在表 2.4和表2.5中。Density-Update 也可以用来计算第一轮迭代之前的数据区间的参数。

表 2.4 第一次迭代后有效区间更新参数（第一部分）

$S[i,j]$	$d_{h_1}=d_1$ 6	$d_{h_2}=d_3$ 9	$d_{h_3}=d_4$ 11	$d_{h_4}=d_2$ 12
$a_2=3$	P_1(10k,2,6)	P_1(10k,2,6)	P_1(10k,2,6) P_4(7k,7,11)	P_1(10k,2,6) P_4(7k,7,11) P_2(8k,3,12)
$a_2=3$	$\varnothing$	$\varnothing$	P_4(7k,7,11)	P_4(7k,7,11) P_2(8k,3,12)
$a_3=5$	$\varnothing$	$\varnothing$	P_4(7k,7,11)	P_4(7k,7,11)
$a_4=7$	Undefined	$\varnothing$	P_4(7k,7,11)	P_4(7k,7,11)

不难看出 Density-Update 的时间复杂度是 $O(n^2)$。第一，Density-Update 的初始化需要 $O(n^2)$ 步骤。第二，在第9行中循环中，针对每一个 i，索引 j 都从 1 变化到 n。此外，对于每一个索引 j，第20行中的 **for** 会检查 $d_{h_{j-1}}$ 和 d_{h_j} 之间的时元。所以，对取遍所有 j 值的整个循环来说，每一个时元最多只会被检查一次。因为一共有 $m-1<2n$ 个时元，针对每一个 i，只需要 $O(n)$ 的时间检查。因此，Density-Update 的时间复杂度是 $O(n^2)$。

Algorithm 2 *Density-Update*($E[], R[], n$)

for $i \leftarrow 1$ **to** n **do** // 初始化循环
  for $j \leftarrow 0$ **to** n **do** // j 从 0 开始是为了循环目的
    $M[i,j] \leftarrow 0$ // 初始化，每个 $I[i,j)$ 都是“无效”的
    $S[i,j] \leftarrow \varnothing$ // $\varnothing$= 空集
    $B[i,j] \leftarrow 0$
    $L[i,j] \leftarrow 0$
  end for
end for
for $i \leftarrow 1$ **to** n **do**
  $d_{h_0} \leftarrow a_i$ // 设置 d_{h_0} 是为了循环的便利
  for $j \leftarrow 1$ **to** n **do**
    if $R[h_j] = 1$ **and** $a_{h_j} \geq a_i$ **then** // P_{h_j} 未分配并且 $[a_{h_j}, d_{h_j}) \subseteq I[i,j)$
      $S[i,j] \leftarrow S[i,j-1] \cup \{P_{h_j}\}$
      $B[i,j] \leftarrow B[i,j-1] + B_{h_j}$
    else
      $S[i,j] \leftarrow S[i,j-1]$
      $B[i,j] \leftarrow B[i,j-1]$
    end if
    $L[i,j] \leftarrow L[i,j-1]$ // 开始加入新的时元
    for $k \leftarrow \max\{\xi(d_{h_{j-1}}), \xi(a_i)\}$ **to** $\xi(d_{h_j}) - 1$ **do**
      if $E[k] = 1$ **then** // 时元 k 可用
        $L[i,j] \leftarrow L[i,j] + (e_{k+1} - e_k)$
      end if
    end for
    if $S[i,j] \neq \varnothing$ **then**
      $M[i,j] \leftarrow 1$ // $I[i,j)$ 为有效
      $D[i,j] = B[i,j]/L[i,j]$
    end if
  end for
end for

基于 Availability-Update 和 Density-Update，DIF 策略的算法展示在下面。我们假设算法的输入包括：

(1) $P_i = (B_i, a_i, d_i), 1 \leq i \leq n.$

(2) $0 \leq a_1 \leq a_2 \leq \ldots \leq a_n, 0 \leq d_{h_1} \leq d_{h_2} \leq \ldots \leq d_{h_n} = T.$

(3) $0 \leq e_1 \leq e_2 \leq \ldots \leq e_m.$

表 2.5 第一次迭代后有效区间更新参数（第二部分）

$B[i,j]$ $L[i,j]$ $D[i,j]$	$d_{h_1}=d_1$ 6	$d_{h_2}=d_3$ 9	$d_{h_3}=d_4$ 11	$d_{h_4}=d_2$ 12
$a_1=2$	B=10k L=1+2=3 D=10/3=3.333	B=10k L=3 D=10/3=3.333	B=17k L=3+2=5 D=17/5=3.4	B=25k L=5+1=6 D=25/6=4.167
$a_2=3$	M=0 inactive	M=0 inactive	B=7k L=2+2=4 D=7/4=1.75	B=15k L=4+1=5 D=15/5=3
$a_3=5$	M=0 inactive	M=0 inactive	B=7k L=2 D=7/2=3.5	B=7k L=2+1=3 D=7/3=2.333
$a_4=7$	M=0 inactive	M=0 inactive	B=7k L=2 D=7/2=3.5	B=7k L=2+1=3 D=7/3=2.333

(4) 函数 ξ 可以将每一个到达时间 a_i 和截止时间 d_{h_j} 映射到它在时元序列的顺序值。

对于图2.1中的例子，我们从表格2.4 和表格2.5中观察得到 $I[1,4)$ 在第二次迭代中是密度最大的区间。因为 $S[1,4]=\{P_1,P_4,P_2\}$，我们将时元 $E_1=[2,3), E_2=[3,5), E_6=[9,11)$ 和 $E_7=[11,12)$ 分配给数据报文集合 $\{P_1,P_4,P_2\}$，并且使用传输速率 $r=25/6=4.167$ 。显然，对于该例子，DIF 策略在两轮迭代之后就结束了。

因为总共有 n 个数据报文并且在第9行的 **while** 循环中每一次迭代都会至少有一个数据报文变成“已分配”，所以 DIF 策略最多需要 n 轮迭代就可以结束。因为程序 Availability-Update 和 Density-Update 各自需要 $O(n^2)$ 的步骤，所以 DIF 策略的时间复杂度是 $O(n^3)$。

在我们证明其正确性之前，我们首先给出几个性质。

性质 2.1. 在 while 循环第16行中指定给 $T[i,j]$ 的速率不大于在之前的迭代中指定的速率，因为在每一次循环中，密度最大的区间被 Availabiliy-Update 移出了。

性质 2.2. DIF 事实上把 n 个数据报文分成若干组。每当密度最大的区间 $I[i,j)$ 在一次迭代中被确定时，所有在 $S[i,j]$ 中当前“未分配”的数据报文都被分离出来。所有在 $S[i,j]$ 中的数据报文将被以速率 $r=D[i,j]$ 在 $I[i,j)$ 中的“可用”时元中传输。

性质 2.3. $S[i,j]$ 中的数据报文在 $T[i,j]$ 中的可用时元上以速率 $r=D[i,j]$ 传输，根据性质2.1，r 是整个区间 $I[i,j)$ 中所有时元中最小的速率。换句话说，对于任意 $S[i,j]$，它的 $T[i,j]$ 由 $I[i,j)$ 上速率最小的所有时元组成。

性质 2.4. 当 $I[i,j)$ 被确认为密度最大的区间时，数据区间 $I[u,v)\subset I[i,j)$ 的密度必定小于等于 $I[i,j)$ 的密度，也就是说，$D[u,v]\leq D[i,j]$，因为否则 $I[i,j)$ 就不会是密度最大的区间。

Algorithm 3 *DIF-Policy*($P[], n, m, \xi$)

1: **for** $k \leftarrow 1$ **to** $m-1$ **do**
2: $E[k] \leftarrow 1$ // 标记所有的时元为“可用”
3: **end for**
4: **for** $i \leftarrow 1$ **to** n **do**
5: $R[i] \leftarrow 1$ // 标记所有的数据报文为“未分配”
6: **end for**
7: Desity-Update($E[], R[], n$)
8: 寻找密度最大的区间 $I[i,j)$ // $D[i,j] = \max\{D[u,v]|M[u,v]=1\}$
9: **while** $D[i,j] > 0$ **do**
10: $T[i,j] = \varnothing$ // 分配给 $S[i,j]$ 的时元集合
11: **for** $u \leftarrow \xi(a_i)$ **to** $\xi(d_{h_j}) - 1$ **do**
12: **if** $E[u] = 1$ **then**
13: $T[i,j] \leftarrow T[i,j] \cup$ {epoch E_u}
14: **end if**
15: **end for**
16: 分配速率 $r = D[i,j]$ 给所有 $T[i,j]$ 中的时元
17: Availability-Update($E[], R[], I[i,j]$)
18: Density-Update($E[], R[], n$)
19: 寻找密度最大的有效区间 $I[i,j)$ // $D[i,j] = \max\{D[u,v]|M[u,v]=1\}$
20: **end while**
21: **for** $k \leftarrow 1$ **to** $m-1$ **do**
22: **if** $E[k] = 1$ **then** // 所有余下的可用时元
23: 为时元 E_k 分配速率 $r = 0$
24: **end if**
25: **end for**

只要我们能够证明在 while 循环的第16行中，任何 $T[i,j]$ 上“可用”时元上的最优总速率都必定等于速率 $r = D[i,j]$，DIF 策略的正确性就可以得到证明。我们用定理2.2阐述该结论，该定理可以看成是定理2.1的扩展。

定理 2.2 (一般化的最大密度区间定理)**.** 在 DIF 策略的任何一次迭代（while 循环）中，令 $I[i,j)$ 满足 $D[i,j] = \max\{D[u,v]|M[u,v]=1\}$。那么，下列陈述成立：

(1) 任何最优速率策略都必须分配速率 $r = D[i,j]$ 给 $T[i,j]$ 中的每一个可用时元。

(2) 任何最优解都必须将数据报文集 $S[i,j]$ 排他地在 $T[i,j]$ 中传输完毕。

证明. 我们用归纳法证明，该方法针对迭代进行归纳。根据定理2.1，在第一次迭代中，定理2.2的结论是成立的，这将作为归纳法的基底。假设定理2.2对前 k 轮迭代都是正确的，我们的目标是证明它对第 $k+1$ 次迭代也是正确的，其中 $k \geq 1$。

在第 $k+1$ 轮迭代的开始，令数据区间 $I[i,j)$ 满足 $D[i,j]=\max\{D[u,v]|M[u,v]=1\}$。如果 $D[i,j]=0$，那么所有数据报文必定已经在之前的迭代中被分配了传输速率了，而且在第23行中指定速率 $r=0$ 给所有可用时元是任何最优策略唯一的选择。所以，我们假设 $D[i,j]>0$。令 $T[i,j]$ 是在第 $k+1$ 轮迭代开始的时候，时间区间 $I[i,j)=[a_i,d_{h_j})$ 中的所有可用时元的集合。假设 $ropt(t), a_i \le t < d_{h_j}$，是最优策略中使用的速率。根据归纳法，$r^{opt}(t)$ 等于在第 $k+1$ 轮迭代开始之前 DIF 策略为不可用时元设定的速率。

如果在某个 $T[i,j]$ 中的时元中有 $r^{opt}(t)\neq D[i,j]=r$，那么我们断言一定有一个时元 $[e_k,e_{k+1})\subseteq T[i,j]$ 使得 $r^{opt}>r$。否则，我们会有 $r^{opt}(t)\le r$ 对 $T[i,j]$ 中的时元都成立，而且在某个时元中有 $r^{opt}(t)<r$，这就暗示了 $\sum_{[e_k,e_{k+1})\in T[i,j]}\int_{e_k}^{e_{k+1}} r^{opt}(t)\,\mathrm{d}t < r\times L[i,j]=B[i,j]$，一些数据会错过它们的截止时间。假设 $r^{opt}(t)=r_{ek}>r$ 在时元 $[e_k,e_{k+1})\subseteq T[i,j]$ 中被使用。我们将时间区间 $[e_k,e_{k+1})$ 扩展成更大的区间。令 $[e_u,e_v)$ 是使得 $[e_k,e_{k+1})\subseteq[e_u,e_v)$ 以及在 $[e_u,e_v)$ 中 $r^{opt}(t)\ge r$ 同时成立的最长数据区间。注，区间 $[e_u,e_v)$ 可以同时包括可用和不可用的时元。$r^{opt}(t)$ 在 e_u 上增加速度，而在 e_v 上减少速率。

根据引理2.1，e_u 必须是一个到达点 a_x 而 e_v 必须是一个截止时间点 d_{h_y}，因此，$[e_u,e_v)$ 是一个数据区间 $I[x,y)$ 并且其密度 $D[x,y]$ 在第 $k+1$ 轮迭代开始的时候不会大于 r，因为 r 是最大的密度。显然 $I[x,y)$ 在第 $k+1$ 轮迭代开始的时候是有效的因为它包括了 $[e_k,e_{k+1})$。此外，如果时间区间 $[e_u,e_v)$ 包括一个在之前的第 k' 轮循环（$k'\le k$）中被密度最大区间 $I[x',y')$ 覆盖而标记为不可用的时元，那么时间区间 $I[x',y')$ 必定被整体的包含在 $I[x,y)$ 中，也就是说 $I[x',y')\subseteq I[x,y)$，因为根据性质2.1所有被标记了的时元具有更高的速率 $r'\ge r$。

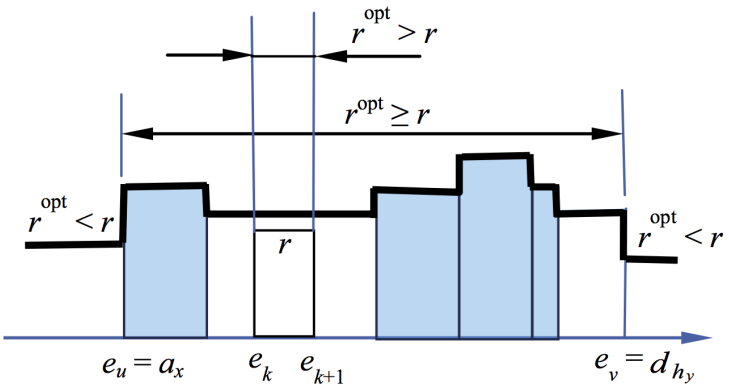


图 2.3 区间 $[e_k,e_{k+1})$ 和区间 $[e_u,e_v)$ 之间的关系

图2.3展示了区间 $[e_k,e_{k+1})$ 和区间 $[e_u,e_v)$ 之间的关系，也展示了在区间 $[e_u,e_v)$ 上 $r^{opt}(t)$ 和 r 之间的关系：在区间 $[e_u,e_v)$ 上，灰色区域在之前的迭代中已经被分配了的时元，所以它们在本次第 $k+1$ 轮迭代中不再可用。

我们将等待区间在 $I[x,y)$ 中的所有数据报文分成两组。第一组包括了在 DIF 策略中第 $k+1$ 轮迭代之前就被分配时元和指定传输速率的数据报文，而第二组包括了余下的在第 $k+1$ 轮迭代开始的时候未分配的数据报文。令 $B^{(1)}$ 和 $B^{(2)}$ 分配代表这个两组中的

数据负载。那么，在第 $k+1$ 轮迭代开始的时候，$I[x,y)$ 的密度 $D[x,y] = B^{(2)}/L[x,y]$，其中 $L[x,y]$ 是所有 $T[x,y]$ 中的可用时间的总长度。因为在整个区间 $[e_u, e_v)$ 上有 $r^{opt}(t) \geq r$ 并且在 $[e_k, e_{k+1})$ 上有 $r^{opt}(t)$ 严格大于 r，那么在区间 $[e_u, e_v)$ 上最优速率传输的数据量是：

$$B = B^{(1)} + \sum_{[e_k,e_{k+1}) \in T[x,y]} \int_{e_k}^{e_{k+1}} r^{opt}(t) u dt > B^{(1)} + r \times L[x,y].$$

密度 $D[i,y] = r$ 在第 $k+1$ 轮迭代开始的时候是最大的，我们有 $r \times L[x,y] \geq D[x,y] \times L[x,y] = B^{(2)}$。因此，我们有 $B > B^{(1)} + B^{(2)}$，这暗示着最优策略传输了某个数据报文 P_z 而其到达时间 a_z 比 a_x 小或者其截止时间 d_z 比 d_{h_j} 大。两种情况都与引理2.2矛盾。所有陈述（1）得证。

陈述（2）直接可以从（1）得到，因为速率 $r = D[i,j]$ 刚刚够传输完所有 $S[i,j]$ 中的数据报文。 □

2.4 具体数据报文最优调度

一旦确定最高密度区间 $I[i,j)$ 上面的速率，那么我们就需要在该区间上调度 $S[i,j]$ 中的数据报文。不合理的调度可能会违反时序约束。让我们继续看上一节中图2.1中的例子。在 DIF 策略已经确定区间 $[2,5)$ 和区间 $[9,12)$ 要使用速率 4.167 传输 P_1, P_2 和 P_4 之后，我们应该如何调度它们呢？如果我们就以 P_1, P_2 和 P_4 的顺序一个一个地调度的话，P_4 中的某些数据会错过截止时间。当 $S[i,j]$ 中有许多数据报文时，这样的问题变得非常复杂。DIF 策略仅仅确定每一个时元上面最优的传输速率，我们需要安排好具体每个数据报文的真正传输。除非我们能够找到一种调度方案可以使得每一个数据报文都能够在各自的截止时间之前传输完毕，否则我们没有完整解决了定理2.5中给出的最优问题。幸运的是，我们发现最早截止时间优先（EDF）算法可以完美的解决该问题。我们可以在每一次迭代当中，对 $S[i,j]$ 中的数据报文使用 EDF 算法，但是最好是只使用一次 EDF，从 $t = 0$ 到 $t = T$，按照 DIF 策略确定的每个时元上的速率传输。其伪代码如 EDF-Schedule 所示。

让我们继续讨论图2.1中的例子。根据 DIF 策略，最优速率如下：$r[2,3) = r[3,5) = 4.167$，$r[5,6) = r[6,7) = r[7,9) = 5$，$r[9,11) = r[11,12) = 4.167$。EDF 采取下列步骤：

1. 在 $t = 2$ 时，因为 P_1 已经达到，在 $[2,3)$ 上以速率 4.167 传输 P_1，剩余的大小 =10-4.167=5.833。

2. 在 $t = 3$ 时，插入 P_2。P_1 仍然在 H 的堆顶。$f_t = 3 + 5.833/4.167 = 3 + 1.4 = 4.4 < 5$。$P_1$ 在 4.4 时结束。删除 P_1。

3. 在 $t = 4.4$ 时，P_2 开始以速率 4.167 传输。$f_t = 4.4 + 8/4.167 = 4.4 + 1.92 = 6.32$。因为 6.32>5，所以 P_2 的传输在 $t = 5$ 时中止。剩下的大小 =8-4.167(5-4.4)=8-2.5=5.5。

Algorithm 4 *EDF-Schedule*(P, m)

build a min-hepa H using deadlines as the keys. // 必要时用达到时间区分先后
for $i \leftarrow 1$ **to** $m-1$ **do**
  if e_i is an arrival point **then**
    insert those packets that arrive at e_i into H
  end if
  $r \leftarrow r(E_i)$ // DIF 策略分配给 E_i 的速率
  $s_t \leftarrow e_i$ // 发送下一个数据报文的开始时间
  while $s_t < e_{i+1}$ **and** $H \neq \varnothing$ **do**
    $P_k \leftarrow$ packet at the root of H // 获取 H 堆顶上的数据报文而不把它取下来
    $f_t \leftarrow s_t + B_k/r$ // 预期的结束时间
    if $f_t > e_{i+1}$ **then**
      $f_t \leftarrow e_{i+1}$
    end if
    transmit P_k at rate r in time interval $[s_t, f_t)$
    if $r \times (f_t - s_t) < B_k$ **then**
      $B_k \leftarrow B_k - r \times (f_t - s_t)$ // P_k 的剩余大小
    else
      extract P_k from H
    end if
    $s_t \leftarrow f_t$ // 下一个开始时间
  end while
end for

4. 在 $t = 5$ 时，插入 P_3。P_3 位于堆顶。$f_t = 5 + 20/5 = 9 > 6$，以速率 5 传输 P_3 直到 $t = 6$。剩余的大小 =20-5=15。

5. 在 $t = 6$ 时，没有插入。继续以速率 5 传输 P_3。$f_t = 6 + 15/5 = 9 > 7$，传输 P_3 直到 $t = 7$ 。剩余的大小 =15-5=10。

6. 在 $t = 7$ 时，插入 P_4，但是 P_3 仍然在堆顶。继续以速率 5 传输 P_3。$f_t = 7 + 10/5 = 9$ 。P_3 在 $t = 9$ 时完成。删除 P_3。

7. 在 $t = 9$ 时，没有插入。因为 P_4 在堆顶，以速率 4.167 传输 P_4。$f_t = 9 + 7/4.167 = 9 + 1.68 = 10.68 < 11$。$P_4$ 在 $t = 10.68$ 完成。删除 P_4。

8. 在 $t = 10.68$ 时，P_2 在堆顶。$f_t = 10.68 + 5.5/4.167 = 10.68 + 1.32 = 12 > 11$，传输 P_2 直到 11。剩余的大小 =5.5-4.167(11-10.68)=5.5-1.333=4.167。

9. 在 $t = 11$ 时，没有插入。继续以速率 4.167 传输 P_2 直到 12。删除 P_2。

在这一节剩下的部分里，我们讨论 EDF 调度的正确性。

引理 2.3. 对于 DIF 策略确定的区间 $I[i,j)$，如果所有 $T[i,j]$ 中包含的未分配时元都以速率 $D[i,j]$ 排他地传输 $S[i,j]$，那么 EDF 调度可以保证 $S[i,j]$ 中的所有数据报文都在截止时间之前完成。

证明. 我们使用反证法。为了到矛盾，我们假设数据报文 $P_k = (B_k, a_k, d_k)$ 是第一个错过其截止时间 d_k 的报文。根据 EDF 调度，只有那些在 $S[i,j]$ 中，并且其截止时间比 d_k 早的数据报文在 $T[a_k, d_k]$ 中传输了，其中 $T[a_k, d_k] \subset T[i,j]$ 是时间区间 $[a_k, d_k)$ 中包含的所有可用时元的集合。此外，$T[a_k, d_k]$ 必定是连续没有中断的，因为在 d_k 时刻仍然有 P_k 的没有完成的数据。现在，通过寻找最早的时间 $t \leq a_k$ 使在 $[t, d_k)$ 的可用时元中只有 $S[i,j]$ 中截止时间比 d_k 早的数据报文传输，我们将区间 $[a_k, d_k)$ 扩展成为 $[t, d_k)$。该时间 t 必定是一个到达时间，因为如果 t 是截止时间，那么我们一定可以找到一个更早的时间 t。令 $t = a_u \leq a_k$。另外，我们可以看到在数据区间 $[a_u, d_k)$ 上，只有在 a_u 或者之后到达的数据报文可以被传输，否则，我们可以进一步扩展 a_u 到一个更早的时间上。这就是说，在 $[a_u, d_k)$ 中的可用时元上，EDF 调度连续不停的以速率 $D[i,j]$ 传输在 a_u 之后到达、在 d_k 之前截止时间的数据报文，但是仍然导致 P_k 错过了自己的截止时间。这就暗示了区间 $[a_u, d_k)$ 的密度比 $D[i,j]$ 更高，这和性质2.4矛盾。 □

推论 2.1. 对于 DIF 策略确定的区间 $I[i,j)$，如果所有 $T[i,j]$ 中包含的未分配时元都被排他地用来传输 $S[i,j]$ 中的数据报文，那么 $S[i,j]$ 中总是有足够的数据可以被按照速率 $D[i,j]$ 传输。

该观察是正确的，因为 $T[i,j]$ 中的可用时间以及速率 $D[i,j]$，刚刚足够发送完数据负载 $B[i,j]$。因为，根据引理2.3，没有数据报文会错过截止时间，所以 $T[i,j]$ 中的传输必须连续不停地传输。这就暗示了 $S[i,j]$ 中总是有足够的数据可以被按照速率 $D[i,j]$ 传输。

定理 2.3. 给定 n 个数据报文 $P_i = (B_i, a_i, d_i), 1 \leq i \leq n$，如果传输速率使用 DIF 策略确定的速率，那么 EDF 调度可以保证所有数据报文都可以在其截止时间之前完成。

证明. 根据引理2.3，我们只需要说明对于任何一个 DIF 策略确定的区间 $I[i,j)$，所有 $T[i,j]$ 中包含的未分配时元都被排他地用来传输 $S[i,j]$ 中的数据报文。

根据反证法，令时元 $E_h = [e_h, e_{h+1}) \subset I[u,v)$ 是第一个分配给区间 $I[u,v)$ 的 $S[u,v]$，但是却被 EDF 用来传输属于另一个区间 $I[i,j)$ 的 $S[i,j]$ 的数据报文 $P_k = (B_k, a_k, d_k)$。显然，DIF 确定 $I[u,v)$ 比确定 $I[i,j)$ 早。因此，我们有下列结论。

首先，我们说 $d_k \leq d_v$，否则的话，EDF 调度不会在 $I[u,v)$ 中传输 P_k，因为根据推论，$S[u,v]$ 中总是有足够的数据在 $I[u,v)$ 中传输，而且 $S[u,v]$ 中数据的截止时间比 d_k 还要早。在 $I[u,v)$ 中传输 P_k 是不可能的。其次，因为 P_k 属于 $S[i,j]$，我们必定有 $a_k < a_u$，否则的话，P_k 就属于 $S[u,v]$ 了。第三，我们必定有 $d_k > a_u$，因为 P_k 在 $E_h \subset I[u,v)$ 中传输。综上，我们有 $[a_k, d_k) - I[u,v) = [a_k, a_u)$。而在 $I[u,v)$ 的整个时间区间上，没有对 $S[i,j]$（包括 P_k）可用的时元，P_k 必定早已在 a_u 的时候就已经错过了自己的截止时间了。这和引理2.3 矛盾了。 □

定理2.3显示了 DIF 策略和 EDF 调度一起最优地解决了在定义 2.6中给出的能量高效的数据报文调度问题。注，我们在定理 2.2中证明的每个时元上的最优速率是唯一的。然而，在 EDF 调度中确定的具体数据报文传输调度却不是唯一的。举个例子，如果我们有 3 个数据报文，$P_1 = (2K, 0, 2), P_2 = (3K, 0, 3), P_3 = (1K, 2, 3)$。最优速率是 2K 每个时元。EDF 调度会从时刻 $t = 0$ 到 $t = 1$ 传输 P_1，然后从时刻 $t = 1$ 到 $t = 2.5$ 传输 P_2，最后从时刻 $t = 2.5$ 到 $t = 3$ 传输 P_3。另外一个可行的方法是从时刻 $t = 0$ 到 $t = 1$ 传输 P_2，剩余 1K 数据，然后从时刻 $t = 1$ 到 $t = 2$ 传输 P_1，从时刻 $t = 2$ 到 $t = 2.5$ 传输 P_3，最后从时刻 $t = 2.5$ 到 $t = 3$ 传输剩余的 P_2。EDF 调度只是一种方案安排具体数据报文的传输，使其满足约束而且消耗最小的能量。显然，在传输速率确定之后，EDF 调度是最有效而且最方便的方法安排具体数据报文的发送调度。

2.5 在线策略和仿真结果

在之前的小节中，我们已经设计了一个离线最优速率控制策略和一个可以保证截止时间的具体数据报文传输调度算法用以解决能量高效的数据报文传输问题。基于该离线策略，在本节中，我们设计一个不需要任何数据报文到达信息（包括到达时间、截止时间、报文大小）的在线速率控制策略和数据报文调度算法。

2.5.1 已有的在线策略

一个直观的在线策略是每当一个新的数据报文到达的时候，采用离线的 DIF 策略和 EDF 调度对该新数据报文以及剩余的未完成积压报文一起处理，然后按照新的调度传输。积压适配（Backog Adaptive, BA）策略[23] 和在线齐平（Online Flush, OF）调度[17] 就是采用该方法，我们称之为 BA-OF 策略。

BA-OF 策略的基本思想是维护一个（积压）队列缓存所有未完成的数据报文。它们根据截止时间排序。根据这些积压数据报文信息，BA-OF 策略使用离线算法计算最优速率。每当一个新的数据报文到达时，它就被插入到积压队列中，并且传输速率也会被相应地重新计算。一旦速率计算出来，队列中的数据报文就被按序传输。

BA-OF 策略计算速率的方法如下。假设当前时间是 t_0，并且有 k 个数据报文被缓存，$P_i = (B_i, a_i, d_i), i = 1, 2, \ldots, k$。那么，BA-OF 策略使用下列公式计算速率 r_0 和索引 j：

$$r_0 = \max_{0 \le d_j \le d_k} \frac{\sum_{i=1}^{j} B_i}{d_j - t_0}, \qquad d_j = \arg \max_{0 \le d_j \le d_k} \frac{\sum_{i=1}^{j} B_i}{d_j - t_0} \tag{2.7}$$

一旦 (r_0, d_j) 对确定，那么传输速率 r_0 将会被使用直到 P_j 被传输完毕或者一个新的数据报文到达。那时，公式 (2.7) 会被再次用来计算新的传输速率。根据文献[23]，如果不再有新的数据报文到达的话，那么该策略是最优的。

公式 (2.7) 事实上从当前时刻 t_0 开始的所有区间中寻找密度最大的区间。此外，该在线策略使用 EDF 为当前密度最大区间调度。因此，该在线策略和我们的离线最优策

略不谋而合。

2.5.2 密度指导的降温策略

尽管 BA-OF 策略具有一定的效果，但是它略显保守。如果下一个数据报文具有较大的数据量和较紧急的截止时间，BA-OF 策略会使用一个很高的速率，消耗很多能量。假如通过预测可以预判这样的情况，那么可以节省这部分能量。图2.4(a) 和 (e) 分别展示了数据报文 P_1 和 P_2 的两种相交方式。在 $t = a_1$ 时，只有数据报文 P_1 在队列中，所以 BA-OF 决定使用低的速率，如 (b) 和 (f) 中的时元 1 所示。在 $t = a_2$ 时，一个大数据报文 P_2 到达。有两种情况，$d_2 < d_1$ 和 $d_2 \geq d_1$。在两种情况中，BA-OF 都会被迫使用较高的速率传输 P_2 和 P_1 余下的部分，正如 (b) 和 (f) 中时元 2 和 3 所示。(c) 和 (g) 中给出了传输两个数据报文的最优速率。比较 (c) 和 (b)，(g) 和 (f)，我们总结是如果能够预测出有大而且紧急的数据报文的到来，便可使用较高的速率预传输当前队列中的数据报文。然而，困难的是我们不知道下一个数据报文什么时候到来，它有多大。为了更好的决定多少数据量需要用高的速率提前发送，做一个好的预测极其重要。

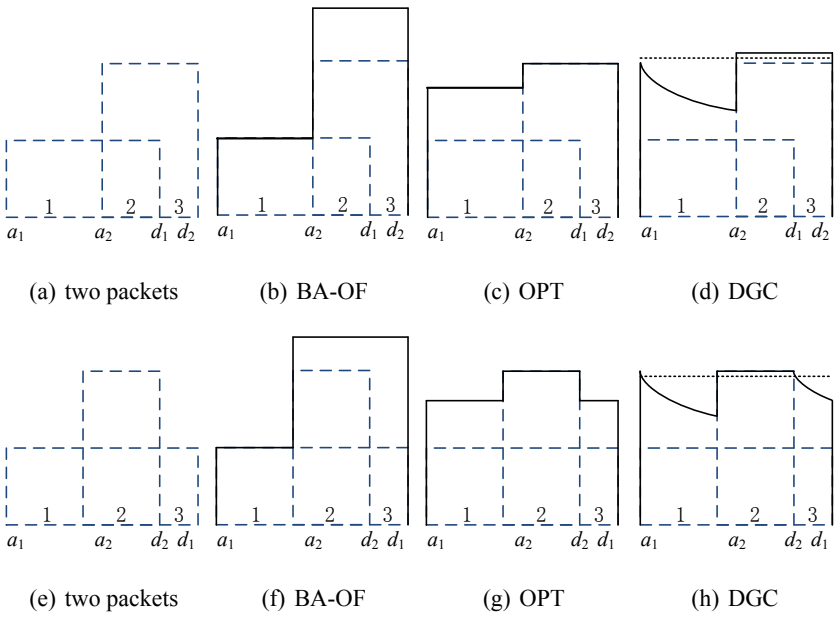


图 2.4 BA-OF 策略，最优策略 OPT，以及 DGC 策略的比较

下面，我们解释图2.4(d) 和 (h) 中所显示的在线策略传输速率。我们使用历史平均密度作为分界线来区分大数据报文和小数据报文。具体地，我们首先应用公式 (2.7) 来为当前剩余未完成的挤压数据报文计算速率 r_0 和结束时间 d_j。如果 r_0 大于历史平均密度（速率），亦即当前的数据报文是属于高密度的区间，那么在这些时元上我们的策略在采用和 BA-OF 相同的速率。但是，如果 r_0 小于历史平均密度，我们预计大数据报文会在不久后到达，因此我们按照一个精心设计的速率函数来提前发送更多的数据，而不是像 BA-OF 那样用一个恒定的速率来发。该速率函数以历史平均密度 (比 r 更大) 开始，然后逐渐减小到一个小的速率，就像当一个高温物体放在低温环境里它的温度变化一样。

图2.5给出了根据牛顿冷却定律，物品的温度下降曲线，也就是 $r = (a-b)e^{-\lambda(t-t_0)} + b$，其中 a 是物体的原始温度，b 是环境的温度，λ 是冷却因子。在我们的实验中，我们重新定义了这三个参数的含义但使用了原来的公式。因此，我们称我们的在线策略是密度指导的冷却 (Density Guided Cooling, DGC) 策略。与 BA-OF 策略相比，DGC 策略对待大的数据报文和小的数据报文更灵活，而且也能保证所有数据报文满足截止时间要求。我们下面介绍细节。

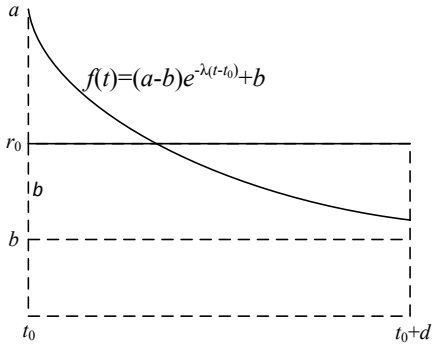


图 2.5 速率函数满足指数递减

我们一个一个的讨论 a, b 和 λ 三个参数的定义。首先，令 a 是历史平均密度。

定义 2.10 (历史平均密度)**.** 假设 $r(t)$ 是历史传输速率，而（事件达到点）$t_p(>0)$ 是上一次计算平均速率 a_{pre} 的时间。那么，在时间点 $t_0(>t_p)$，新的平均速率 a_{new} 按照这样计算。

$$a_{pre} = \frac{\sum_{e_k<t_p} r(e_k)(e_{k+1}-e_k)}{t_p}, \qquad a_{new} = \frac{\sum_{t_p\le e_k<t_0} r(e_k)(e_{k+1}-e_k)}{t_p - t_0}$$

设 $\alpha = \frac{t_p}{t_0}$，历史平均密度 a 这样计算：

$$a = \alpha a_{pre} + (1-\alpha)a_{new} \tag{2.8}$$

显而易见，公式 (2.8) 准确的计算了在有限时间区间 $[0, t_0)$ 上的平均密度。但是，在现实环境中 t_0 会变得非常大或者趋近无穷，我们用常数 α 替代变量 $\frac{t_p}{t_0}$。在计算 TCP 的环回时延时，也用到了相似的思想。因此，历史平均密度可以同时在有限时间和无限时间上得到计算。

历史平均密度是物体开始下降的“高温度”a，我们现在定义温度下降最终停止的环境“低温度”b。

如果 $r_0 < a$，我们定义“最小保证速率”b 作为速率函数下降的下届。注，速率 b 总是比速率 r_0 小。为了确定速率 b，我们引入公式 (2.9) 中定义的可变常数参数“侵率”$\beta(0<\beta<1)$。我们尝试了不同的 β 值并且观察到了相似的结果。在本节所报告的仿真实验结果中，我们设定 $\beta = 0.5$。

$$\beta = \frac{r_0 - b}{a - b}, \quad 0 \le b < r_0 < a \tag{2.9}$$

根据公式 (2.9)，“最小保证速率”b 可以用公式 (2.10) 来计算。

$$b = \begin{cases} \frac{r_0 - \beta\alpha}{1-\beta}, & \text{if} \quad \beta\alpha \leq r_0 < a \\ 0, & \text{if} \quad r_0 < \beta\alpha \end{cases} \tag{2.10}$$

给定“高温度”a 和“环境低温度”b，我们按照下述公式来确定“冷却参数”λ。

$$\lambda = \lambda(d) = \frac{A}{d} \tag{2.11}$$

其中 A 满足下列等式：

$$1 - e^{-A} = \beta A \tag{2.12}$$

在给定 β 的情况下，A 的值可以很容易预先计算得到。d 是当前速率函数定义域的长度，并且 d 大于 $d_j - t_0$。如何确定 d 会在下一子节中讨论。

定义 2.11 (速率预测函数)**.** 在 $t = t_0$ 时刻，给定历史平均密度 a，速率 r_0 及其结束时间 d_j，最小保证速率 b，冷却参数 λ，那么在区间 $[t_0, d_j)$ 上的速率预测函数如下定义：

$$f(t) = \begin{cases} (a-b)e^{-\lambda(t-t_0)} + b, & \text{if} \quad r_0 < a \\ r_0, & \text{if} \quad r_0 \geq a \end{cases} \tag{2.13}$$

该公式中使用了指数递减，而指数递减在自然科学中有许多应用，比如流体动力学，核放射理论和热力传导等。

提出并使用这样的速率预测函数的几点理由：（1）长远来看，未来数据报文的密度预期应该是在历史平均 a 左右的。因此，我们的预测函数以速率 a 作为开始。（2）在许多系统中，数据报文的到达都满足泊松分布，在泊松分布中，到达间隔时间满足指数分布。（3）随着时间的推移，剩余的数据量越来越少，无需提前传输更多数据。相反，预测函数需要降低速率以节省能量。（4）预测速率必须在给定区间中大于一个最小速率。该最小速率 b，根据情况可能会等于 0。

我们的 DGC 在线策略包括 4 步：

(1) 根据公式 (2.8) 计算历史评价密度 a，使用公式 (2.7) 计算 (r_0, d_j)，再用公式 (2.10)。

(2) 根据公式 (2.13) 设置速率预测函数 $f(t)$，其中 $\lambda = \lambda(d)$ 是根据公式 (2.11) 计算得到的。

(3) 从 t_0 开始安装速率 $f(t)$ 传输数据报文直到 d_j。

如果所有数据报文在 d_j 之前都发送完成，那么就暂停发送直到 d_j。

如果一个新数据报文在 d_j 之前的某个时刻到达，就截止时间将其插入到队列中。

(4) 转（1）。

定理 2.4. DGC 在线算法保证所有数据报文满足各自的截止时间要求。

证明. 公式 (2.7) 中的 BA-OF 策略保证所有数据报文都满足各自的截止时间要求[17, 23]。所以，我们只需要证明 DGC 策略保证任何一个数据报文的完成时间都和用 BA-OF 策略一样或者更早。对 $r_0 \geq a$ 的情况，这显然成立，我们只需要证明 $r_0 < a$ 的情况。

因为在 DGC 在线策略中，$\lambda = \frac{A}{d}$，所以 $A = \lambda \times d$。根据公式 (2.11) 和 (2.12)，我们有

$$1 - e^{-\lambda d} = \beta \lambda d$$

根据公式 (2.13) 中定义的速率函数，我们进一步计算在其定义域 $[t_0, t_0 + d)$ 上的数据传输总量，如下：

$$\begin{aligned}\int_{t_0}^{t_0+d} f(t)\,\mathrm{d}t &= \int_{t_0}^{t_0+d} ((a-b)e^{-\lambda(t-t_0)} + b)\,\mathrm{d}t \\ &= bd + (a-b)(\frac{1}{\lambda} - \frac{e^{-\lambda d}}{\lambda}) = bd + (a-b)d \\ &= bd + (r_0 - b)d = r_0 d\end{aligned}$$

这意味着公式 (2.13) 中定义的速率函数在区间 $[t_0, t_0 + d)$ 上的数据传输总量恰好等于如果使用常数速率 r_0 传输的数据量。因为 $[t_0, d_j) \subseteq [t_0, t_0 + d)$ 并且公式 (2.13) 中定义的速率函数是一个递减函数，那么，在任何一点 $t_x \in [t_0, d_j)$ 上，我们必定有

$$\int_{t_0}^{t_x} f(t)\,\mathrm{d}t > \int_{t_0}^{t_x} r_0 t\,\mathrm{d}t$$

也就是说，在区间 $[t_0, t_x)$ 上，DGC 在线策略传输的数据量超过了 BA-OF 策略，因此，DGC 策略保证每一个数据报文能比在 BA-OF 策略中提早完成。

□

2.5.3 仿真结果

在我们的仿真中，我们设定侵率 $\beta = 0.5$（如前述）。这样，我们就有 $\frac{r_0 - b}{a - b} = 0.5$ 和 $A = 1.59$。根据 (2.11)，$\lambda = \frac{A}{d} = \frac{1.59}{d}$，所以，要计算 λ，我们必须先计算 d。令 $c = d_j - t_0$。我们如下设置 d：

$$d = \begin{cases} 2c, & \text{if } c > avg(delay) \\ 2avg(delay), & \text{otherwise} \end{cases} \tag{2.14}$$

其中 $avg(delay)$ 是所有已到达的数据报文的平均延时要求，可以用 (2.8) 中类似的方法计算得到。注，每当一个新的速率预测函数被计算得到后，d 也要相应的重新计算。由于 c 和 $avg(delay)$ 都在不断更新，因此 d 也会随时间不断变化。综上所述，在我们的仿真中用到的特定速率函数是：

$$f(t) = \begin{cases} 2(a-b)e^{\frac{-1.59(t-t_0)}{d}} + 2r_0 - a, & \text{if } \ r_0 < a \\ r_0, & \text{if } \ r_0 \geq a \end{cases} \tag{2.15}$$

从 (2.14)，我们能看出 $d > c$，因此定理2.4成立：所有的数据报文能满足他们各自的延时要求。

在我们的仿真实验中，我们也仔细的设定数据报文相关的参数，包括报文到达，报文大小和报文延时要求。(i) 根据大多数前人的工作，我们假设报文到达是一个泊松过程。(ii) 我们假设报文大小满足正态分布 $\mathcal{N}(s, 0.1s)$，其中 s 是平均报文大小。(iii) 我们用三种分布的混合分布来模拟任意报文延时要求，即平均分布 $\mathcal{U}(0.1q, 0.9q$，正态分布 $\mathcal{N}(q, 0.3q)$ 和修改的指数分布 $EXP(0.9q) + 0.1q$，其中 q 是平均报文延时要求。所有报文的延时要求都大于 $0.1q$。我们这样假设是因为数据报文必须有一个最小的区间 $[0, 0.1q)$ 来传输。

本节中所有的图和表中展示的数值都是 40 个随机仿真实例的平均计算结果。每一个随机仿真实例中，都产生了 300 个满足上述设定的数据报文。

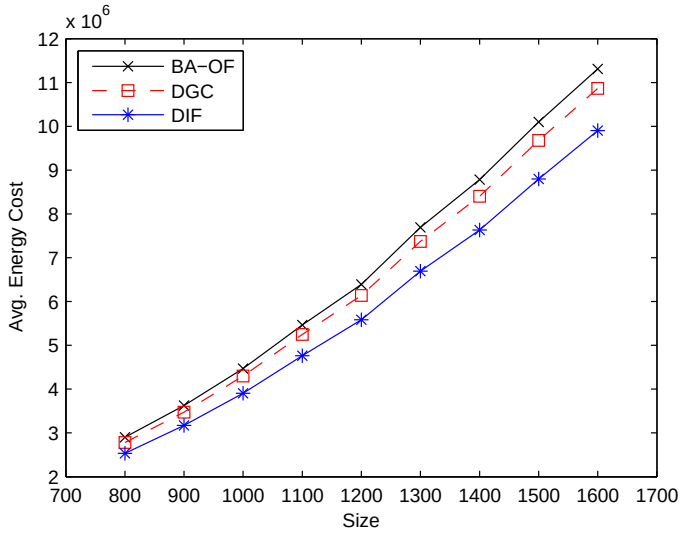
我们首先比较 DGC 策略和 BA-OF 策略的性能。

表 2.6 DIF,BA-OF 和 DGC 策略能耗比较

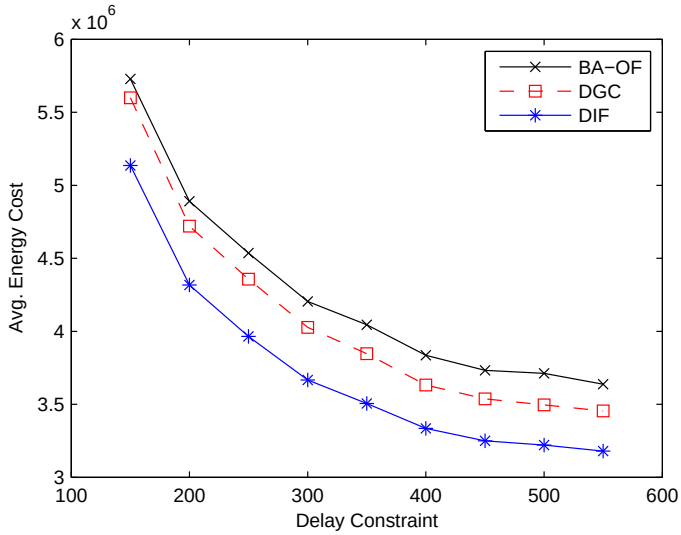
	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6
DIF(10^6)	6.59	3.93	3.20	2.84	2.67	2.52	2.48	2.35
BA-OF(10^6)	7.58	4.51	3.59	3.12	2.90	2.70	2.63	2.48
DGC(10^6)	7.18	4.33	3.50	3.07	2.87	2.68	2.62	2.48
BA-OF/DIF(%)	115.0	114.8	112.0	110.0	108.5	107.2	106.3	105.7
DGC/DIF(%)	108.9	110.4	109.2	108.1	107.3	106.5	105.9	105.5

从表2.6中，我们能看出当“报文平均间隔时间 - 平均报文时延要求”的比值变大时，BA-OF 策略和 DGC 策略两者也都更趋向于接近离线最优解 (DIF) 的结果。这是因为，当该比值变大超过 1 的时候，报文间隔时间比延时要求还要大，即，报文的截止时间在下一个报文到达时间之前。所以，很少甚至没有调度的需要并且所有调度算法都能得到接近最优的结果。然而，当该比值比较小的时候，BA-OF 策略无法得到和离线最优解接近的结果。而在这种情况下，我们的 DGC 策略性能超出 BA-OF 策略，产生的结果也在离线最优解的常数倍以内。

在图2.6中，默认设置是平均数据报文大小 1000 个单位，平均延时要求是 250 个单位时间和平均到达间隔是 100 个时间单位。为了研究这三个参数对算法性能的影响，它们被一一单独改变进行分析比较。从图2.6(a)-(c) 中可以看出，在所有情况下，DGC 策略性能稳定的超出 BA-OF 策略。在几乎所有的情况下，DGC 策略输出的结果都在离线最优解的 110% 范围内。在图2.6(a) 中，DGC 的平均能耗随着平均数据报文大小的增加而增加，但是几乎稳定在离线最优解的 109% 范围内。在图2.6(b) 中，平均能耗随着数据报文延时要求的增长而降低。这是因为数据报文延时要求越长，这些数据报文的紧急程度越低，发送它们的速率越小从而消耗更少的能量。在图2.6(c) 中，平均能耗随着到达间隙的增加而减少。这是因为到达间隙越大，到达率越低，单位时间内到达的数据报文越少，因此消耗的能量越少。在 (b) 和 (c) 中，DGC 的曲线几乎和离线最优解的曲线平行，这说明 DGC 的性能是稳定的。



(a) 报文大小对算法的影响



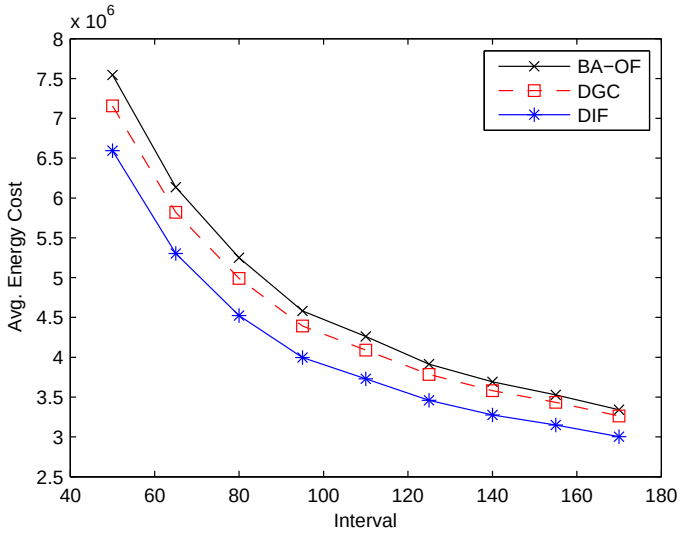
(b) 报文延时要求对算法的影响

图 2.6 最优 DIF 策略、BA-OF 策略和在线策略 DGC 的比较（第一部分）

我们进行了大量的实验分析，得到了丰富的实验结果远超过图2.6所展示。所有的实验结果都显示 DGC 策略比现有的 BA-OF 算法更加适合处理数据报文各项参数动态变化的场景，并且效果更优。

2.6 本章小结

本章解决了允许数据报文具有任意截止时间的能量高效的数据报文传输调度问题，。为了把握这一难题的本质，本章引入了“数据区间”的概念，提出了“密度最大区间优先”（DIF）的新技术。我们随后证明了最早截止时间优先（EDF）的策略是最优的，而且可以在 DIF 策略确定传输速率之后有效的进行具体数据报文的传输调度。最后，我们



(c) 报文间隔时间对算法的影响

图 2.6 最优 DIF 策略、BA-OF 策略和在线策略 DGC 的比较（第二部分）

提出了一个在线的调度策略 DGC 算法。仿真实验显示，通过更好的预测和预判，DGC 策略稳定的输出最优解 110% 内的结果。DIF 策略和 EDF 策略的整合可能为其他能量高效问题的研究，比如能量采集问题、多信道系统和衰弱信道，提供一个通用的方法。

第三章　静止传输时能量自供给设备能量高效调度

本章设计了截断算法，将一组动态到达的数据报文从静止放置的能量自供给无线设备上传输到同样静止放置的无线接收设备上，并最小化传输能耗。与已有工作的显著不同在于，本章允许数据报文具有独立的传输延时约束，该设定是所有相关工作中最具一般性和最通用的，而且也是保证应用相关服务质量 (QoS) 的迫切需要。本章还限制速率在一组离散数中取值，该设定更具实用意义，并且许多现实应用都存在该需求。做为第一个成果，在速率连续模型下，本章设计了“截断”最优算法。随后提出通用框架将使用连续速率模型的任何算法转变为使用离散速率模型的算法，并且保持原算法的最优性。在采集的能量不足以保证所有数据报文满足各自截止时间的情况下，目标就变为利用有限的能量最大化吞吐量。截断算法可以直接判定该情况，并且当数据报文具有共同截止时间时，计算出最优速率调度。最后，根据最优静态离线算法，本章设计了动态联机算法，仿真表明该算法可以达到近似最优的性能。本章是第2章的深化和推进。

3.1 引言

能量自供给 (Energy Harvesting) 是一种通过能量采集器，利用光伏、热电、压电、电磁等各种物理或化学效应把器件周围环境中广泛存在的太阳能、热能、机械能、风能甚至无线电信号[10] 等能量转换成可以使用的电能的技术。能量自供给技术使得无线设备可以工作更长时间，因此该技术在众多领域都具有被看好的商业前景。虽然在理论上该技术可以提供无限的能量供给，但是因为每次能量采集的量不多，而且不稳定，所以本章仍然要设计算法控制能量消耗。无线通信中，根据著名香农定理 (Shannon-Hartley Theorem)，以较低的无线传输速率传输可以节省能量，以较高的无线传输速率传输则可以缩短传输延时。大量的相关研究工作据此设计电池供电设备的能量高效的速率调度算法[13, 14, 16, 17, 22, 23, 57] 或者能量自供给设备的能量高效的速率调度算法[48–56]。

本章尝试在离散速率模型下解决该速率调度问题。无论是在连续速率模型下，还是在离散速率模型下，允许数据报文具有独立传输延时，该设定是所有已有工作中最据一般性的。独立传输延时与离散速率调度两者的结合决定了本章研究内容与能量自供给已设备设计速率调度的其它相关工作具有很大不同。

作为第一个也是最重要的关键步骤，本章首先设计了一个静态离线算法计算最优速率调度算法。在静态离线问题中，所有关于数据报文和能量采集的信息都已知，此外，可用的离散速率也已经给定。该步骤扩展并改进了一个已有速率调度算法，但即使这

样，仍然存在很大的算法设计挑战：第一，在任意时刻，传输速率调度消耗的能量不能超过采集的能量，剩余未用的能量可以存储在电池中以备将来使用。第二，在任意时刻，速率调度传输的数据量不能多于到达系统的数据总量，未传输的数据量留在系统中等待传输。第三，传输速率调度要保证每一个数据报文都在其截止时间之前传输完毕。第四，传输速率只能在有限的离散数值中取值。第五，传输速率调度只能以指定的可用速率（连续取值范围或者离散数值集合）传输。此外，额外还需要考虑一个特殊情况，即当采集的能量不足以保证所有数据报文的在截止时间之前发生完毕的情况。那么，最小化能耗的目标就变得没有意义，相反，应该考虑利用所有可用能量最大化数据传输量。

3.2 问题定义

令 $P = \{P_1, P_2, \ldots, P_n\}$ 是一组 n 个待传输的数据报文。每一个数据报文 $P_i = (B_i, a_i, d_i)$ 都有一个大小 B_i，一个到达时间 a_i 和一个独立的截止时间 d_i ($> a_i$)。数据报文已经被排序过，所以 $a_1 \leq a_2 \leq \ldots \leq a_n$。假设数据报文截止时间与报文到达时间保持一致的顺序，也就是 $d_1 \leq d_2 \leq \ldots \leq d_n$。这是一个合理的假设，因为在大多数实际的系统中，数据报文一般先存储在队列中，等待按照“先进先出 (First In First Out, FIFO)”的原则传输，所以数据报文的完成时间必然和他们的到达时间一致。因此，在本章中最早截止时间优先 (Earliest Deadline First, EDF) 调度和先进先出调度是一样的。令 $T = d_n$ 是最后一个截止时间。数据报文 P_i 的传输只能在其到达时间 a_i 之后开始，必须在其截止时间 d_i 之前完成。这被称之为“时序约束”[49]。

与已有工作[28–31, 48–56] 一致，令 $H = \{H_1, H_2, \ldots, H_m\}$ 是一组 m 个能量采集的实例（简称“采集”）。一个采集 $H_i = (E_i, c_i), 1 \leq i \leq m$ 表示在时间点 $t = c_i$ 上，有 E_i 的能量被系统获得。假设 $0 < c_1 < c_2 < \ldots < c_m \leq T$。无线设备上电池中的初始能量被当做发生在时刻 $t = 0$ 上的采集 H_0。因为每次采集的能量非常有限，和电池的容量相比非常小，为了简单起见，假设电池足够大，足已存储所有采集的能量。因此，假设从 0 时刻到任意时刻 t，消耗了的能量不能超过所有采集的到的能量。这被称之为“能量约束”。

对于采集 $H_i = (E_i, c_i), 1 \leq i \leq m$，我们说一个“采集事件”发生在时刻 $t = c_i$，c_i 被称为一个“采集（事件）点”。类似的，到达时间 a_i 和截止时间 $d_i, 1 \leq i \leq n$，被分别称为“到达事件/点”和“截止事件/点”。因此，从时刻 0 到 T，一共有 $(m + 2n)$ 个事件点，$e_i, i = 1, 2, \ldots, m + 2n$，其中 $e_1 \leq e_2 \leq \ldots \leq e_{m+2n} = d_n = T$。两个相邻事件点之间的时间区间被称为“时元”。如果两个事件点发生在同一时刻的话，时元的长度可能会是零。图3.1显示了一个具有四个数据报文和四个能量采集的例子，$P_1(240kb, 0s, 3s)$, $P_2(450kb, 2s, 5s)$, $P_3(230kb, 4s, 7s)$ 和 $P_4(720kb, 5s, 8s)$, 一个四个采集，$H_0(2.85mJ, 0s)$, $H_1(1.09mJ, 3s)$, $H_2(3.78mJ, 4s)$ 和 $H_3(4.80mJ, 6s)$ (初始能量被以发生在时刻 $0s$ 的采集 H_0 对待)。

我们考虑单用户点对点传输信道[13, 14, 16, 17, 22, 23, 48–52]，与已有工作一致，无线设备也假设可以自主的改变传输速率 r，并且 r 和传输功率 p 存在函数 $p = g(r)$ 的关系。该函数 $p = g(r)$ 被称作“速率 -功率”函数，而且具有凸函数性质[13, 14, 16, 17, 22, 23, 48–52]。

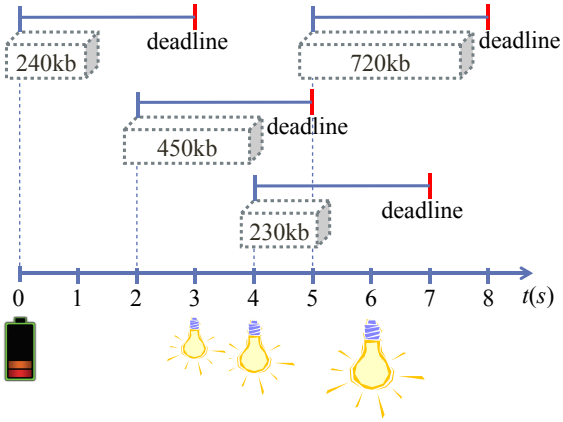


图 3.1 具有四个数据报文和四个能量采集的例子

事实上，在许多现实编码/解码方案下，速率 -功率函数都具有凸性质，比如在高斯白噪声信道 (Additive White Gaussian Noise, AWGN) 下单用户的最优随机编码中，$r = g^{-1}(p) = \frac{1}{2}\log(1 + \frac{p}{N})$，其中 N 是热噪声的等级，常常被设定为 $N = 1$[49]。

早先的研究者都假设传输速率是连续可控的。但是这种连续速率模型并不总是实际可行的。我们因此引入“离散速率模型”。令 $R = \{\gamma_0 = 0, \gamma_1, \ldots, \gamma_s = r_{max}\}$ 是一组可允许的传输速率集合，$\{\rho_0 = 0, \rho_1, \ldots, \rho_s\}$ 是一组对应的传输功率，并且有 $\rho_i = g(\gamma_i), i = 1, 2, \ldots, s$。这样一个离散速率模型如图3.2所示。相比连续速率模型，离散速率函数更难处理，因为它有更多的约束。

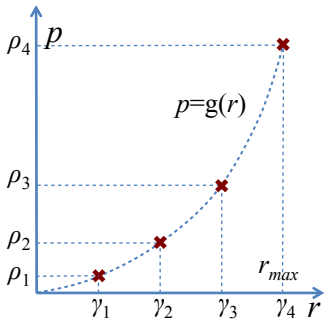


图 3.2 离散速率模型

定义 3.1. 在 t 时刻，用于传输数据报文 P_i 的速率 $r_{P_i}(t) : [0, T) \to \{\gamma_0, \gamma_1, \ldots, \gamma_s\}$ 被定义成数据报文 P_i 的“报文传输速率函数”，其中 $0 \le t < T, 1 \le i \le n$。

因此，时序约束可以被如下表示

$$\int_0^T r_{P_i}(t)\,\mathrm{d}t = \int_{a_i}^{d_i} r_{P_i}(t)\,\mathrm{d}t = B_i, \quad i = 1, 2, \ldots, n. \tag{3.1}$$

定义 3.2. 给定一组 n 个报文传输速率函数，$r_{P_i}(t)$，$0 \le t < T, 1 \le i \le n$，所有报文传输速率函数之和 $r(t) = \sum_{i=1}^{n} r_{P_i}(t), 0 \le t < T$，被定义成“总速率函数”。

总速率调度 $r(t)$ 唯一地确定了任意时间 t 上的传输速率，再以按先进先出的顺序发送数据报文。给定一个总速率调度 $r(t)$，在 0 到 t 之间的能耗可用公式 $E(t) = \int_0^t g(r(x))\,\mathrm{d}x$ 计算。所以 t 时刻的“能量约束”可以表达为

$$E(t) = \int_0^t g(r(x))\,\mathrm{d}x \le \sum_{i:c_i<t} E_i,\ \ t \in [0,T). \tag{3.2}$$

定义 3.3. 给定一组数据报文 P，一组采集 H，和一组允许的速率集合 R，如果一组 n 个速率调度 $r_{P_1}(t), r_{P_2}(t), \ldots, r_{P_n}(t)$ 同时满足“时序约束”公式 (3.1) 和“能量约束”公式 (3.2)，那么它们就称作“可行解”。对应的的总速率调度被称为“可行速率调度”。

现在，我们的 min-E 问题定义如下。

定义 3.4 (min-E 问题)**.** “能量最小化问题”是寻找一个可行解，其对应的总速率调度 $r(t)$ 最小化总的能量消耗 $E(T)$。这样的总速率调度被称作“最优速率调度”，标记为 $r^{opt}(t)$（或者 r^{opt}，如果不产生歧义的话）。

如果采集的能量不足以传输所有数据报文的话，min-E 问题可能没有可行解，时序约束公式 (3.1) 因此无法得到满足。在这种情况下，我们就利用有限的可用能量最大化吞吐量，即 max-T 问题。在 max-T 问题中，不需要公式 (3.1)，而是将该等式松弛到下面的不等式。

$$\int_0^T r_{P_i}(t)\,\mathrm{d}t = \int_{a_i}^{d_i} r_{P_i}(t)\,\mathrm{d}t \le B_i, i = 1,2,\ldots,n \tag{3.3}$$

我们假设数据传输以比特作为单位，所以数据可以被以任意大小传输和丢弃。在 $[0,T)$ 中，实际传输的数据量可以用下面的积分公式计算。

$$B = \int_0^T r(t)\,\mathrm{d}t. \tag{3.4}$$

现在，我们的 max-T 问题，如下定义。

定义 3.5 (max-T 问题)**.** “吞吐量最大化问题”是寻找一个总速率调度 $r(t)$ 使由公式 (3.4) 确定的实际数据传输量最大，同时保证约束 (3.2) 和 (3.3) 得到满足。

min-E 问题与 max-T 问题统称“静止传输时能量自供给设备能量高效调度问题”，我们设计了“截断”算法解决该问题。在给出我们的“截断”算法之前，我们需要介绍一个非常相关的已知问题，它假设连续的速率模型、不考虑能量采集。

定义 3.6 (文献[17, 23])**.** 在连续速率模型下，数据报文模型如上所示，“电池供电设备的数据报文节能调度问题”是寻找一个总速率调度 $r(t)$，使能量消耗 $E(T)$ 最小，并且满足公式 (3.1) 中的约束。

在该电池供电设备的数据报文节能调度问题中，没有考虑能量采集，它的首要目标是最小化传输中消耗的能量。而且他们的目标暗示无线设备的初始能量无线大。我们称该问题的最优总速率调度为“ZM 速率调度”并表示为 $r^{zm}(t)$（或者 r^{zm}，如果没有歧义的话）。r^{zm} 可以用 Zafer 和 Modiano 提出的算法[23] 计算得到。第3.8.1节中给出了该算法的一个简明介绍。

3.3 带有公共截止时间的数据报文最优速率调度

我们以一个简化了的 min-E 问题开始：假设所有的数据报文都共享一个公共的截止时间 ($d_1 = d_2 = \ldots = d_n = T$) 并且采用连续速率模型 ($0 \leq r \leq r_{max}$)。我们称该简化了的问题为“公共截止时间能耗最小化问题”。在我介绍“截断”方法之前，我们首先研究最优速率调度必须具有哪样的性质。

3.3.1 最优速率调度的基本性质

参考文献[49] 已经表明，在任意时元 $[e_k, e_{k+1}), k = 1, 2, \ldots, m + 2n - 1$ 中，只有一个恒定速率应该被使用。因为如果两个速率 $r_p \neq r_q$ 被使用，那么我们总是可以找到一个速率 r 在这两个速率之间，$r_p < r < r_q$，或者 $r_p > r > r_q$，用来传输同样的数据量却消耗更少的能量，或者使用同样的能量传输更多的数据量。这样的办法被称为“均衡”，相应的，两个速率被称为可以被“均衡”[49]。

因此，最优速率调度 $r^{opt}(t)$ 是一个阶梯函数，在每个时元内保持恒定的速率。在时元和时元之间，r^{opt} 可能会变化，这样的变化满足下面性质。

引理 3.1. r^{opt} 的增加只可能在到达点或者采集点上。

引理3.1很容易用反证法证明，证明的关键是在速率变化点前后的两个时元上使用“均衡”方法。我们省略细节。因此“公共截止时间能耗最小化问题”的 r^{opt} 在截止时间 T 之前不会减小。

引理 3.2. 如果 r^{opt} 在 t 上增加了，那么，要么所有 t 之前到达的数据报文都已经被传输完毕了，要么所有 t 之前采集的能量都已经被用尽了。点 t 被相应地称为 r^{opt} 的“数据临界点”或者“能量临界点”。

证明. 我们用反证法证明。假设 r^{opt} 在 t 上增加，但数据和能量在 t 上都有剩余，那么 t 前后紧邻的两个时元可以被均衡，这和 r^{opt} 是最优的矛盾。所有，该引理得证。 □

文献[49] 已有相似的结果，不过它是针对一个简单模型的。

根据引理3.1和引理3.2，数据临界点 $t < T$ 显然（必定）是一个到达点，而一个能量临界点 $t < T$ 必定是一个采集点。

注，我们只讨论传输用的能量。总有一份最小能量保留在系统只，以确保设备在能量临界点之后仍然能够工作。这部分能量在我们的研究中是不考虑的。

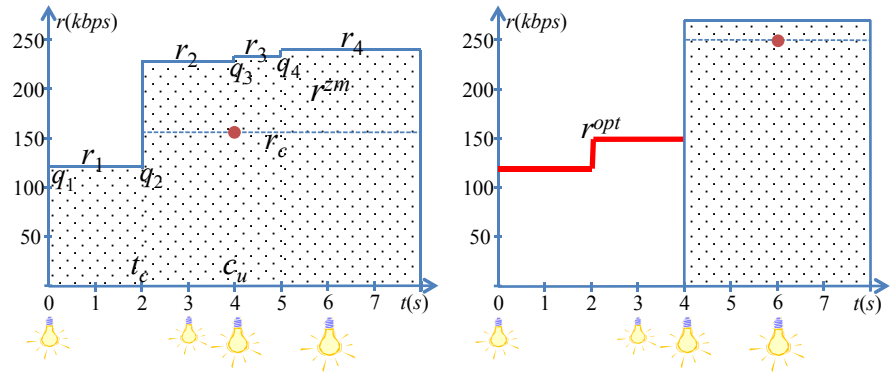
引理 3.3. 如果 t 是 r^{opt} 的第一个能量临界点，在区间 $[0, t)$ 中，任何最大化的数据传输量的速率调度必定和最优速率调度是一致。换句话说，最优速率调度是唯一的，它最大化了区间 $[0, t)$ 上的数据传输量（吞吐量）。

证明. 见第3.8.2节。 □

因此，计算 r^{opt} 的关键就是找到它的第一个能量临界点。

3.3.2 “截断”方法

“截断”方法的总架构思想如下。我们首先假设初始能量足够大而不需要使用采集的能量，计算 r^{zm} 以最小化总传输能耗[23]。如果，在任意时刻 $t \in [0, T)$，采集的能量都足以支持 r^{zm}，那么 r^{zm} 已经最小化了能耗，所以我们有 $r^{opt} = r^{zm}$。否则，“采集”不足够支持 r^{zm}。因此，我们把 r^{zm} 截断成为 $\min\{r^{zm}, r_c\}$，其中水平线 r_c 切下了 r^{zm}，如图3.3所示。显然，只要 r_c 足够小，“采集”肯定足够支持截断后的速率。但是，如果 r_c 太小的话，那么很少的数据量可以在临界点之前传输，结果更多的数据要在之后以更加高的速率传输，会消耗更多的能量。所以，我们希望找到最大的速率 r_c 使得“采集”足够支持速率 $\min\{r^{zm}, r_c\}$（称为 r^{zm} 被 r_c 截断的速率）。显而易见，在这样的速率调度中，会存在这样一个点，所以采集的能量都被在该点上耗尽，但是一个新的“采集”刚好发生在该点上。该点将被证明是第一个能量临界点，我们还会声明在该点之前 r^{zm} 被 r_c 截断的速率是最优的。从该新点开始，一模一样的问题重复了。



(a) r^{zm} 被 r_c 截断　　(b) 下一轮迭代

图 3.3 “截断”方法的示例

我们现在讨论“截断”方法的细节。

假设 $r_1, r_2, \ldots, r_s$ 是被 r^{zm} 使用的 s 个不同的速率。它们按顺序排列，其中速率 r_i 从 q_i 开始，到 q_{i+1} 结束。两个已知结果[14, 23] 是：(1) $r_1 < r_2 < \ldots < r_s$；(2) 在点 q_i 上，已到达的数据报文都必定已全部发送完毕。更多细节可以参考第3.8.1节。

因为得到 r^{zm} 的前提是有非常大的初始能量，而在我们的模型中，能量来源是能量采集，所以我们需要一个程序 energy-constraint-check 来检查 r^{zm} 是否在每时每刻都可以被采集的能量支持。为此，我们需要按照时间顺序一一检查每一个时元。在初始化时，我们设定变量 $e = h = 0$。对当前时元，如果时元的起始点是采集事件，那么将采集的能量加到变量 h 中。在该时元中消耗的能量被加到变量 e 中。如果 $e < h$，那么至少在该时元结束的时候，“能量约束”还是满足的，这样程序继续检查下一个时元。如果没有一个时元违反了“能量约束”，我们说 r^{zm} 通过了 energy-constraint-check 检测。否则，检查失败。

如果 r^{zm} 通过了 energy-constraint-check 检查，那么它已经是“公共截止时间能耗最

小化问题”的最优解了，因为速率调度 r^{zm} 最小化了传输能耗。如果 r^{zm} 没有通过检测，那么我们提出算法 runcation 计算速率 r_c 和第一个能量临界点 c_u。图3.3展示了“截断”方法。

Algorithm 5 Truncation(P, H, r^{zm}, t)

let $r_1, r_2, \ldots, r_s$ be rates of r^{zm} in $[t, T)$
let q_i be the start point of the interval with r_i
let $r_0 \leftarrow 0, H_{m+1} \leftarrow (0, d_n)$ for loop propose
for $i \leftarrow 1$ to s **do**
    let $r(t)$ be the rate r^{zm} truncated by r_i
    if $r(t)$ fails the *energy-constraint-check* **then**
        break   //exit the **for** loop
    end if
end for
$t_c \leftarrow q_i$
let $E(t_c)$ be the energy consumed by r^{zm} before t_c
$u \leftarrow arg\min_{j:c_j>t_c}\{\frac{\sum_{i=0}^{j-1} E_i - E(t_c)}{c_j - t_c}\}$
$r_c \leftarrow g^{-1}(\frac{\sum_{j=0}^{u-1} E_j - E(t_c)}{c_u - t_c})$
let $r(t) \leftarrow \min\{r^{zm}, r_c\}$
return $(r(t), c_u)$

其中 **for** 循环确定的索引 i 满足 $r_{i-1} \le r_c < r_i$。令 $t_c = q_i$，$E(t_c)$ 是 r^{zm} 在时间的 t_c 之前的能耗。对每一个采集事件点 $c_j > t_c$，令 $H(c_j)$ 是时间区间 $[0, c_j)$ 中的总能量采集(不包括点 c_j)，那么 $E = H(c_j) - E(t_c)$ 是我们可以在 $[t_c, c_j)$ 中使用的最大的能量。如果我们在 $[t_c, c_j)$ 中使用恒定速率，那么我们能够使用的最大速率是 $r_{c_j} = g^{-1}(E/(c_j - t_c))$。我们寻找一个采集点使得 r_{c_j} 最小。我们用该速率作为 r_c“截断”。该 r_c 是最大的，因为任何更大的速率都将导致所有的能量在采集点之前的某点耗尽。公式 (3.5) 直接计算这样的点 c_u 和速率 r_c。

$$u = \arg\min_{j:c_j>t_c}\{\frac{\sum_{i=0}^{j-1} E_i - E(t_c)}{c_j - t_c}\}, r_c = g^{-1}(\frac{\sum_{i=1}^{u-1} E_i - E(t_c)}{c_u - t_c}) \tag{3.5}$$

定理 3.1. 算法 Truncation 准确地计算出 r^{opt} 的第一个能量临界点 c_u，如果它存在的话；r^{zm} 被 r_c 截断的速率在 $[0, c_u)$ 上是最优的。

证明. 我们首先通过反证法证明 c_u 是 r^{opt} 的第一个能量临界点。假设 $c_v \neq c_u$ 是最优调度的第一个能量临界点。根据公式 (3.5)，$\frac{\sum_{j=1}^{v-1} E_j - E(t_c)}{c_v - t_c} > \frac{\sum_{j=1}^{u-1} E_j - E(t_c)}{c_u - t_c}$，并且我们有 $g^{-1}(\frac{\sum_{j=1}^{v-1} E_j - E(t_c)}{c_v - t_c}) \ge r_c$。如果 $v < u$，那么在点 c_v 之前必定存在一个时元 p 其中 $r^{opt} > r_c$，因为根据假设，速率调度 r^{opt} 会在 c_v 之前使用完所有的能量。所以，在 $[c_v, c_u)$ 中必定存在一个时元 q 其中 $r^{opt} < r_c$。这样一来，在时元 p 和 q 上，我们可以应用“均衡”减少 r^{opt} 的能耗，这显然是一个矛盾。如果 $v > u$，那么在 c_u 之前必定存在一个时

元 p 其中 $r^{opt} < r_c$，因为使用速率调度 r^{opt} 会使得在 c_u 上仍有能量剩余。这部分的能量余量同时也暗示在 $[c_u, c_v)$ 中必定存在一个时元 q 其中 $r^{opt} > r_c$。均衡因此可以被应用到时元 p 和 q 上，这也是一个矛盾。因此，c_u 是最优解的第一个能量临界点。

令 $r(t)$ 是 r^{zm} 被 r_c 截断的速率。根据第3.8.1节，所有在 $[0, t_c)$ 中到达的数据报文都必须在 $[0, t_c)$ 中被 $r(t)$ 传输完成。任何调度都不可能在 $[0, t_c)$ 中传输更多数据量，并且速率 $r(t)$ 以最小的能耗做到了这一点[14, 23]。因此，速率 $r(t)$ 在 $[t_c, c_u)$ 中具有最大的可用能量。因为 $r(t)$ 在 $[t_c, c_u)$ 中使用了恒定的速率，该恒定速率最大化了吞吐量，所以 $r(t)$ 在 $[t_c, c_u)$ 中传输了最多的数据量。根据引理3.3，$r(t)$ 在 $[0, c_u)$ 中与 r^{opt} 相同。 □

我们在算法 Common-Truncation 中，通过反复调用算法 Truncation，计算最优速率调度。注，我们假设了最高速率是 r_{max}。所以，在每一轮迭代中，我们要保证每一个时元上的速率都不超过 r_{max}（第14行）。在最后一轮迭代中，如果 r^{zm} 通过了 energy-constraint-check 能量约束检查，我们还要检查是否在所有时元中都有 $r^{zm} < r_{max}$。如果通过了，那么，我们设置 r^{opt} 为 r^{zm}，同时所有数据可以得到传输；否则，我们设置 r^{opt} 为 r_{max}，当然，此时无论有多少能量可用，数据都不可能全部得到传输。

Algorithm 6 Common-Truncation(P, H, r_{max})

```
1:  let r^opt ← 0 in [0, T); t ← 0
2:  while t < T do
3:    let r^zm be the optimal ZM rate schedule
4:    if r^zm passes the energy-constraint-check then
5:      if r^zm < r_max in all epochs then
6:        let r^opt ← r^zm in [t, T)
7:        return (r^opt, all-sent)
8:      else
9:        let r^opt ← min{r^zm, r_max} in [t, T)
10:       return (r^opt, partially-sent)
11:     end if
12:   else
13:     (r(t), c_u) ←Truncation(P, H, r^zm, t)
14:     r^opt(t) ← min{r(t), r_max}, in [t, c_u)
15:     schedule transmission according to r^opt until c_u
16:     update packets set P and harvesting set H
17:     t ← c_u
18:   end if
19: end while
20: return (r^opt, partially-sent) //handle the case when T = c_u
```

定理 3.2. 当返回值是 all-sent 时，算法 Common-Truncation 正确地计算出“公共截止时间能耗最小化问题”的最优解 r^{opt}。

证明. 定理3.1证明了算法 Truncation 返回的调度具有部分正确性。Truncation 方法保证了最多的数据在每一个能量临界点之前发送，因此，在能量临界点上剩余数据量最小。在最后一轮迭代中，只在最优 ZM 速率调度通过了 energy-constraint-check 能量约束检查并且所有速率都比 r_{max} 低才会被返回 all-sent。因此，最后一轮迭代中的能耗也是最小的。这保证了算法的最优性。 □

3.4 带有独立截止时间的数据报文最优速率调度

在这一节中，我们仍然考虑连续速率模型，但是允许数据报文具有独立截止时间。我们称该问题是“独立截止时间能耗最小化问题”。图3.1是该问题一个例子。

3.4.1 最优性质

在“独立截止时间能耗最小化问题”中，引理3.1，引理3.2和引理3.3仍然成立。因为截止事件点可能在区间 $[0,T)$ 中任何时刻发生，我们引入下面的引理作为引理3.1和引理3.2的补充。

引理 3.4. r^{opt} 的下降只能在截止点上。如果 r^{opt} 在截止点 d_k 下降，那么数据报文集合 $\{P_1,P_2,\ldots,P_k\}$ 在 d_k 之前被排他地完成了。该点 d_k 被称为“截止临界点”。

通过使用“均衡”法，该引理很容易用反证法证明。我们因此省略细节。事实上，关于“均衡”法，我们有一个更强的定理3.3。

定理 3.3. 一个速率调度是最优的，当且仅当该速率调度中任何两个时元都不能被“均衡”。

证明. 见第3.8.3节。 □

我们现在定义一个子问题，讨论它的最优性质，因为在后面会使用到。

定义 3.7. 一个 i-最优速率调度 $r^{opt(i)}(t)$（或者简称 $r^{opt(i)}$）是仅考虑数据报文 $\{P_1,P_2,\ldots,P_i\}$ 的最优传输速率调度，其中 $1\le i\le n$。

显然，我们有 $r^{opt(n)}=r^{opt}$。不失一般地，我们假设在区间 $[d_i,T)$ 中有 $r^{opt(i)}=0$，其中 d_i 是数据报文集合 $\{P_1,P_2,\ldots,P_i\}$ 中最后一个截止时间。

定理 3.4. $(i+1)$-最优速率调度上的速率在任何一个时元中都不低于 i-最优速率调度上的速率，即，$r^{opt(i+1)}\ge r^{opt(i)}, i=1,2,\ldots,n-1$。

证明. 见第3.8.4节。 □

基于上述定理，我们有下面的结果。

引理 3.5. 在任何数据/能量临界点之前，速率调度 $r^{opt(i)}$ 与 $r^{opt(i+1)}$ 完全一致，$i=1,2,\ldots,n-1$。

证明. 如果 t 是能量临界点，所有 t 之前采集的能量都会被 $r^{opt(i)}$ 在 t 时刻用完，而 $r^{opt(i+1)} \geq r^{opt(i)}$，因此，这两个速率在时间 t 之前必定相同。相似的，如果 t 是数据临界点，那么所有 t 之前到达的数据都会被速率调度 $r^{opt(i)}$ 在 t 时刻传输完毕，而 $r^{opt(i+1)} \geq r^{opt(i)}$，因此，这两个速率调度在时间 t 之前必定相同。 □

根据以上引理，直接可以得到如下定理，将原始问题和刚定义的子问题联系起来。

定理 3.5. 在任何数据/能量临界点之前，速率调度 $r^{opt(i)}$ 与 r^{opt} 完全一致，$i = 1, 2, \ldots, n$。

我们现在定义另一个相似的子问题，然后之间给出它的最优性质。

定义 3.8. 一个 i-ZM 最优速率调度 $r^{zm(i)}(t)$（或者简称 $r^{zm(i)}$）是仅考虑数据报文 $\{P_1, P_2, \ldots, P_i\}$ 的 ZM 最优传输速率调度，其中 $1 \leq i \leq n$。

定理 3.6. $r^{zm(i+1)} \geq r^{zm(i)}, i = 1, 2, \ldots, n-1.$

这是因为“i-ZM 最优速率调度”问题是我们的能耗最小化问题的特例。显然，我们有 $r^{zm(n)} = r^{zm}$。

3.4.2 截断方法的一般化

回忆“截断”的核心是计算最大的速率 r_c 用以截断 r^{zm}，并使得截断后的速率可以被采集支撑。一般化后的“截断”方法也是这么做的，但是不是那么直接。

首选，我们调用 energy-constraint-check 检查 $r^{zm(i)}, i = 1, 2, \ldots, n$。如果所有 $r^{zm(i)}$ 都通过检查，那么 $r^{opt} = r^{zm(n)}$，问题解决。否则，我们寻找使得 $r^{zm(k)}$ 不通过的最小索引 k。这样，采集不足以支撑 $r^{zm(k)}$，所以截断是必须的。令程序 Smallest-K 返回这样的 k 值。

一方面，$r^{zm(k)}$ 没有通过 energy-constraint-check 检查，需要被截断。另一方面，$r^{zm(k-1)}$ 通过了 energy-constraint-check 检查，它因此是传输数据报文集合 $\{P_1, P_2, \ldots, P_{k-1}\}$ 的最优速率调度，即 $r^{opt(k-1)} = r^{zm(k-1)}$。根据定理3.4，我们有 $r^{opt(k)} \geq r^{opt(k-1)} = r^{zm(k-1)}$。这就是说，截断后的 $r^{zm(k)}$ 应该不能比 $r^{zm(k-1)}$ 更加小。因此，速率调度 $r^{zm(k-1)}$ 被称为“基础 (base)”。“基础”和 $r^{zm(k)}$ 之间的差别部分被称为“扩展 (extra)”。“基础”在任何情况下都不应该被截断，只有“扩展”部分才应该被截断。

其次，我们确定第一个使得 $r^{zm(k-1)}(t) < r^{zm(k)}(t)$ 成立的点 t。这可以通过一个一个检查时元完成。我们声明，在时间点 t 之后，**(1)** 速率 $r^{zm(k)}$ 不会下降，除了在 P_k 的截止点 d_k 上；**(2)** 速率 $r^{zm(k-1)}$ 不会上升，除了在 P_{k-1} 的截止点 d_{k-1} 上。

声明 (1) 可以通过反证法证明。假设 $r^{zm(k)}$ 首选下降的点是 $t' > t$ 并且 $t' < d_k$。根据第3.8.1节，t' 是一个截止点，并且在该点之前，所有具有更早截止时间的数据报文都已经被 $r^{zm(k)}$ 在 t' 之前发送完毕了，而且其他任何数据报文都还没有开始发送。因此，

包括 $r^{zm(k-1)}$ 在内的任何可行调度都必须将这些已经发送了的数据报文发送完。这和 $r^{zm(k-1)}(t) < r^{zm(k)}(t)$ 这一事实矛盾。

声明 (2) 也可以通过反证法证明。假设 $r^{zm(k-1)}$ 首次上升的时间点是 $t' > t$ 并且 $t' < d_{k-1}$。根据第3.8.1节，t' 是一个到达点，并且在该点之前，所有具有更早到达时间的数据报文都已经被 $r^{zm(k-1)}$ 在 t' 之前发送完毕了，意味着没有可行解可以比 $r^{zm(k-1)}$ 发送更多的数据量。这和 $r^{zm(k-1)}(t) < r^{zm(k)}(t)$ 这一事实矛盾。

这两个声明可以看出，速率函数 $r^{zm(k-1)}$ 在 t 之后的形状看上去像一个下楼的楼梯；函数 $r^{zm(k)}$ 在 t 之后的形状看上去像一个上楼的楼梯，如图3.4(c) 所示。

再次，我们计算“截断”的位置。如图3.4所示，我们会使用速率为 r_c 的直线截断“扩展”。我们的目标是寻找最大的 r_c，使得截断后的速率在 d_k 之前都可以被采集支撑。根据以上两个声明，我们假设 $r^{zm(k-1)}(t) = r_d > r_{d-1} > \ldots > r_1$ 是 $r^{zm(k-1)}$ 从 t 到 d_{k-1} 上使用的速率。令 $r^{zm(k)}(t) = r_{d+1} < r_{d+2} < \ldots < r_s$ 是 $r^{zm(k)}$ 从点 t 到 d_k 使用的速率。因为 $r^{zm(k-1)}(t) < r^{zm(k)}(t)$，我们有 $r_1 < r_2 < \ldots < r_d < r_{d+1} < r_{d+2} < \ldots < r_s$。令每个速率 r_i 的起始时间是 $q_i, 1 \le i \le s$。通过一个一个检查 $\{r_1, r_2, \ldots, r_s\}$ 中的每一个速率，找到第一个无法通过 energy-constraint-check 检查的 r_i，我们可以确定 r_c 到底在哪两个速率之间，即 $r_{i-1} < r_c \le r_i$。

如果 $r_i \le r_d$，那么 r_c 从点 q_{i-1} 开始，否则的话，$r_d < r_i$，r_c 从点 q_i 开始。我们使用 t_c 表示 r_c 的其实点。那么，速率 r_c 的具体值可以通过在3.3小节中的方法计算。算法 General-Truncation 给出了具体的伪代码。

示例 我们使用图3.1中的例子展示一般化的截断法，如图3.4所示。“速率 -功率”函数是 $r = g^{-1}(p) = 10^3 \log(1 + 0.1p)$，其中 p 是传输功率（毫瓦），r 是传输速率（kbps）。可以验证 $r^{zm(3)}$ 通过了 energy-constraint-check 检查，但是 $r^{zm(4)}$ 没有通过。所以 $k = 3$。“基础”和“扩展”的形状如 (c) 所示。“扩展”使用的速率分别是从 $q_1 = 5$ 开始的 $r_1 = 115$，从 $q_2 = 2$ 开始的 $r_2 = 150$，从 $q_3 = 2$ 开始的 $r_3 = 225$，从 $q_4 = 4$ 开始的 $r_4 = 230$，和从 $q_5 = 5$ 开始的 $r_5 = 240$。“扩展”被 r_1 截断后和“基础”的组合体的速率分别是 120 在 $[0, 2)$ 上, 150 在 $[2, 5)$ 上和 115 在 $[5, 8)$ 上，该速率通过了检查；被 r_2 截断后的速率分别是 120 在 $[0, 2)$ 上和 150 在 $[2, 8)$ 上，该速率也通过了检查；被 r_3 截断后的速率分别是 120 在 $[0, 2)$ 上和 225 在 $[2, 8)$ 上，该速率没有通过了检查。所以，$r_2 \le r_c < r_3$ 并且 $t_c = 2$。在 $t_c = 2$ 之前的能耗是 $E(t_c) = 2g(120) = 1.73(mJ)$。我们检查每一个能量采集点：$c_1 : \frac{E_0 - 1.73}{c_1 - 2} = 1.12$, $c_2 : \frac{E_0 + E_1 - 1.73}{c_2 - 2} = 1.10$, $c_3 : \frac{E_0 + E_1 + E_2 - 1.73}{c_3 - 2} = 1.50$, 和 $c_4 : \frac{E_0 + E_1 + E_2 + E_3 - 22.3}{c_4 - 2} = 1.80$。注，为了便于 **for** 循环的计算，$H_4 = (0, 8)$ 是一个虚拟的采集。显然，我们有 $u = 2$，$r_c = g^{-1}(1.10) = 151$ 以及 $c_u = c_2 = 4$。正如图3.4(d) 所示，最优速率是 120 在 $[0, 2)$ 上和 151 在 $[2, 4)$ 上。同样的问题在更新数据报文集合后再现了，这次以 $c_2 = 4$ 为起点。

定理 3.7. 算法 General-Truncation 正确计算了 r^{opt} 的第一个能量临界点 c_u，如果它存在的话；该算法确定的速率在区间 $[0, c_u)$ 上与 r^{opt} 一致。

证明. 在计算 c_u 时使用了公式 (3.5)。根据定理3.1的证明，c_u 是第一个能量临界点，

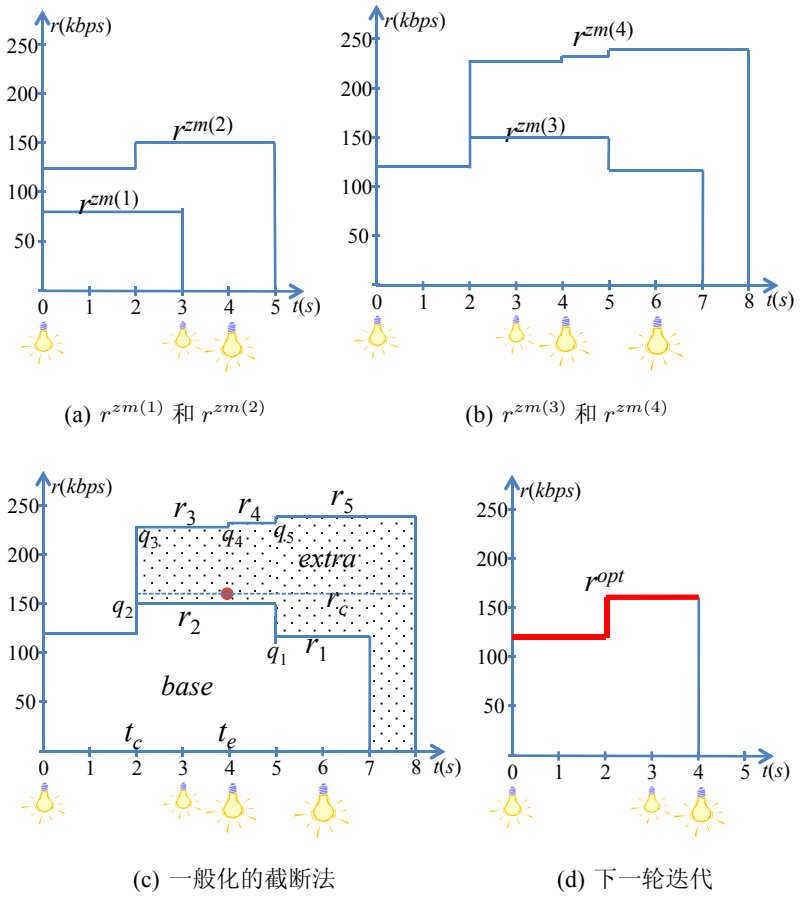
Algorithm 7 General-Truncation(P,H,$r^{zm(k-1)}$,$r^{zm(k)}$)

let t be the first point such that $r^{zm(k-1)}(t) < r^{zm(k)}(t)$; $r_d > r_{d-1} > \ldots > r_1$ be the rates used by $r^{zm(k-1)}$ from point t to d_{k-1}; $r_{d+1} < r_{d+2} < \ldots < r_s$ be the rates used by $r^{zm(k)}$ from point t to d_k
let the starting time for each rate r_i be q_i, $1 \leq i \leq s$
let $r_0 \leftarrow 0, q_0 \leftarrow d_k, H_{m+1} \leftarrow (0, d_n)$ for loop propose
for $i \leftarrow 1$ to s **do**
    let $r(t)$ be the rate of *base* plus the *extra* truncated by r_i
    if $r(t)$ fails the *energy-constraint-check* **then**
        break //exit the for loop
    end if
end for
if $r_i \leq r_d$ **then**
    $t_c \leftarrow q_{i-1}$
    $E(t_c) \leftarrow$ energy consumed by $r^{zm(k-1)}$ before t_c
else
    $t_c \leftarrow q_i$
    $E(t_c) \leftarrow$ energy consumed by $r^{zm(k)}$ before t_c
end if
$u \leftarrow arg\min_{j:c_j>t_c}\{\frac{\sum_{i=0}^{j-1} E_i - E(t_c)}{c_j - t_c}\}$
$r_c \leftarrow g^{-1}(\frac{\sum_{i=0}^{u-1} E_i - E(t_c)}{c_u - t_c})$
let $r(t)$ be the rate r^{zm} truncated by r_c
return $(r(t), c_u)$

意味着所有采集的能量在 c_u 点会被截断后的速率耗尽并且新的采集正好发生在该点 c_u。此外，常数速率 r_c 是使得采集在 $[0, c_u)$ 中足够支撑截断速率的最大值。现在，我们证明在 $[0, c_u)$ 中截断后的速率与 r^{opt} 一致。

令 $r(t)$ 是由“基础”和被 r_c 截断后的“扩展”组成的截断速率。根据定理3.3，我们只需要证明在 $[0, c_u)$ 中没有任何两个时元可以被“均衡”。我们现在证明这是正确的。

令 $p < q$ 是 $[0, c_u)$ 中的两个时元，速率分别是 r_p 和 r_q。如果 $r_p = r_q$，那么 p 和 q 无法被“均衡”。如果 $r_p < r_q$，那么令 x 是时元 p 之后第一个速率增加的点。点 x 必定发生在 $r^{zm(k)}$ 上，这是因为根据前面两个声明，$r^{zm(k)} \neq r^{zm(k-1)}$ 是从点 t 开始的，而且 $r^{zm(k-1)}$ 在 t 之后不会上升。根据第3.8.1节，在 x 上数据发送完了，所以 r_p 不可能被“均衡”增加。如果 $r_p > r_q$，令 x 是时元 p 之后第一个速率下降的点。相似的，我们可以说明点 x 必定发生在 $r^{zm(k-1)}$ 上。根据第3.8.1节，x 必定是 $r^{zm(k-1)}$ 的一个截止临界点。所以，r_p 不可能被“均衡”降低，因为否则的话一些数据就会超过截止时间。所以，我们的结论是在 c_u 之前，$r(t)$ 是最优的。 □



(a) $r^{zm(1)}$ 和 $r^{zm(2)}$　(b) $r^{zm(3)}$ 和 $r^{zm(4)}$

(c) 一般化的截断法　(d) 下一轮迭代

图 3.4　一般化的截断法的示例

在截断完成、能量临界点 c_u 计算出来后，队列中的数据报文按照 FIFO 的顺序使用该调度速率发送，直到临界点 c_u。然后，我们将该临界点 c_u 视为一个新的初始点，计算剩下的数据报文。那么，截断的方法就可以一直重复直到不再有数据报文剩余。通过这样的方法，独立截止时间能耗最小化问题得到了解决。算法 Individual-Truncation 给出了伪代码。

定理 3.8. 算法 Individual-Truncation 在 $O(mn(n^2+m))$ 时间内计算出“独立截止时间能耗最小化问题”的最优解。

证明. 算法的正确性是因为定理3.7保证了算法 Individual-Truncation 做出的每一次截断都是正确的。复杂度分析如下。**while** 循环重复最多 m 次，因为一次循环确定至少一个能量临界点，而总共只有 m 个能量临界点。在每一次 **while** 循环中，最耗时的是程序 Smallest-K。Smallest-K 需要计算 r^{zm}，还需要最多 n 次对程序 energy-constraint-check 的调用。使用文献[23] 中的方法计算 r^{zm} 的时间复杂度是 $O(n^2)$。energy-constraint-check 要检查每一个时元，可以在 $O(m+2n)$ 步骤内完成。因此，Smallest-K 在 $O(n(n^2+m))$ 步骤内完成，总的时间复杂度是 $O(mn(n^2+m))$。 □

Algorithm 8 Individual-Truncation(P, H, r_{max})

let $r^{opt} \leftarrow 0$ in $[0, T)$
$t \leftarrow 0$
while $t < T$ **do**
  let r^{zm} be the optimal *ZM rate schedule* for P
  if r^{zm} passes the *energy-constraint-check* **then**
    if $r^{zm} < r_{max}$ in every epoch **then**
      $r^{opt} \leftarrow r^{zm}$ in $[t, T)$
      return $r(t)$
    else
      return '*non-exist*'
    end if
  else
    $k \leftarrow$ Smallest-K(P, H)
    ($r(t)$,c_u)←General-Truncation(H,$r^{zm(k-1)}$,$r^{zm(k)}$)
    $r^{opt} \leftarrow \min\{r(t), r_{max}\}$, in $[t, c_u)$
    schedule the transmission according to r^{opt} until c_u
    update packets set P and harvesting set H
    $t \leftarrow c_u$
  end if
end while
return '*non-exist*' //handle the case when $T = c_u$

3.5 离散速率调度的框架

在这一节中，我们提出一个通用的框架用以将连续速率模型下的调度算法转换成离散速率模型下的调度算法，只要“速率 -功率”是凸函数就可以保持原调度算法所具有的最优性。

相关领域内的绝大多数已有速率调度算法（包括我们的算法）都是基于“速率 -功率”是函数的凸性质并且考虑连续速率模型[13, 14, 16, 17, 22, 23, 48–51]。事实上，所有在传输延时和能量消耗之间做折衷的工作[13, 14, 16, 17, 22, 23]，或者是在传输延时和吞吐量之间做折衷的工作[48–51] 都是基于“速率 -功率”函数的凸性质。

我们框架的核心思想是设计一个新的凸“速率 -功率”函数 $p = G(r)$ 将离散速率的信息包含在其中。令 $R = \{\gamma_0, \gamma_1, \gamma_2, \ldots, \gamma_s\}$ 是离散速率模型中允许的一组速率，并且有 $0 = \gamma_0 < \gamma_1 < \gamma_2 < \ldots < \gamma_s$；$\{\rho_0 = 0, \rho_1, \ldots, \rho_s\}$ 是一组对应的功率，满足 $\rho_i = g(\gamma_i), i = 1, 2, \ldots, s$。如图3.5所示，允许的速率 -功率是一组离散点 $(\gamma_i, g(\gamma_i)), i = 0, 1, 2, \ldots, s$。我们将这些离散点连接起来，形成一个连续的凹曲线，该曲线可以用新的“速率 -功率”函数 $p = G(r)$ 刻画，如图3.5所示。显然，$p = G(r)$ 是连续

的凹函数，公式如下。

$$p = G(r) = \frac{\gamma_{u+1} - r}{\gamma_{u+1} - \gamma_u} g(\gamma_u) + \frac{r - \gamma_u}{\gamma_{u+1} - \gamma_u} g(\gamma_{u+1}), \text{ if } \gamma_u \leq r < \gamma_{u+1}. \tag{3.6}$$

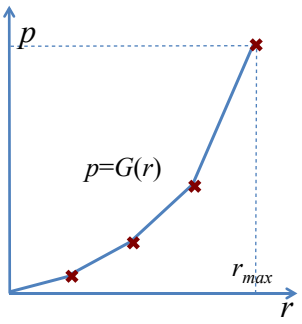


图 3.5 新“速率 -功率”函数 $p = G(r)$

我们现在为上述相关工作中提出的算法设计一个通用两步骤框架。在“替换步骤 (replacing step)”，将 $p = g(r)$ 替换成 $p = G(r)$ 再调用原来的算法产生一个速率调度 $r(t)$。在“转化步骤 (converting step)”，将 $r(t)$ 转化成离散速率。在时元 $[e_i, e_{i+1})$ 中，令 r 是该时元上的速率 $r(t)$，其中 $\gamma_u \leq r < \gamma_{u+1}$。我们将其一分为二，$[e_i, \frac{r-\gamma_u}{\gamma_{u+1}-\gamma_u} e_i + \frac{\gamma_{u+1}-r}{\gamma_{u+1}-\gamma_u} e_{i+1})$ 和 $[\frac{r-\gamma_u}{\gamma_{u+1}-\gamma_u} e_i + \frac{\gamma_{u+1}-r}{\gamma_{u+1}-\gamma_u} e_{i+1}, e_{i+1})$。令前者中的速率是 γ_u，后者中的速率是 γ_{u+1}。该两步骤产生了一个离散速率模型下的可行解，且其保持了原来的最优性。

定理 3.9. 本文提出的框架产生的是最优的离散速率调度。

证明. 该正确性是由于：**(1)**“替换步骤”中产生的速率调度 $r(t)$ 对于连续函数 $p = G(r)$ 是最优的。**(2)** 最优离散速率调度的结果不会比速率调度 $r(t)$ 更好。**(3)**“转化步骤”中产生的离散速率调度的结果和 $r(t)$ 一样。

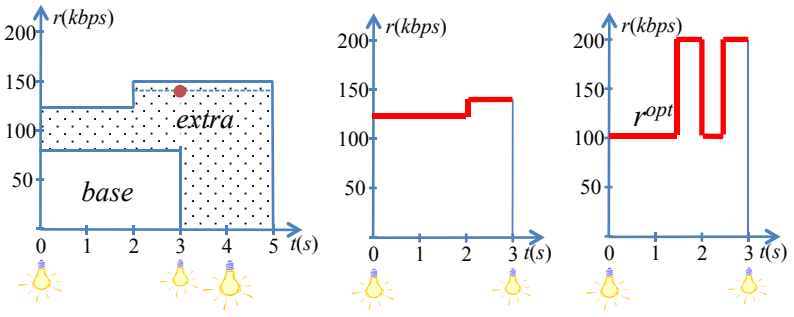
(1) 是正确的，因为原算法的正确性是基于 $p = g(r)$ 的连续性和凸性质，因为 $p = G(r)$ 同样是连续的凸函数，所以“替换步骤”产生的速率调度 $r(t)$ 对 $p = G(r)$ 说仍然是最优的。

(2) 是正确的，因为离散速率调度的解空间是离散的点，而计算 $r(t)$ 使用的解空间是包含了这些离散点的分段线性函数。换句话说，离散速率调度的解空间是计算 $r(t)$ 的解空间的子集合。所以，(2) 是正确的。

(3) 的正确性可以如下证明。时元 $[e_i, e_{i+1})$ 中，在“转化步骤”之后，数据传输量是 $\gamma_u(\frac{r-\gamma_u}{\gamma_{u+1}-\gamma_u} e_i + \frac{\gamma_{u+1}-r}{\gamma_{u+1}-\gamma_u} e_{i+1} - e_i) + \gamma_{u+1}(e_{i+1} - \frac{r-\gamma_u}{\gamma_{u+1}-\gamma_u} e_i - \frac{\gamma_{u+1}-r}{\gamma_{u+1}-\gamma_u} e_{i+1}) = r(e_{i+1} - e_i)$；能量消耗是 $g(\gamma_u)(\frac{r-\gamma_u}{\gamma_{u+1}-\gamma_u} e_i + \frac{\gamma_{u+1}-r}{\gamma_{u+1}-\gamma_u} e_{i+1} - e_i) + g(\gamma_{u+1})(e_{i+1} - \frac{r-\gamma_u}{\gamma_{u+1}-\gamma_u} e_i - \frac{\gamma_{u+1}-r}{\gamma_{u+1}-\gamma_u} e_{i+1}) = (\frac{\gamma_{u+1}-r}{\gamma_{u+1}-\gamma_u} g(\gamma_u) + \frac{r-\gamma_u}{\gamma_{u+1}-\gamma_u} g(\gamma_{u+1}))(e_{i+1} - e_i) = G(r)(e_{i+1} - e_i)$。两者都没有改变。 □

我们仍然使用图3.1中的例子来展示如何使用该框架在离散速率模型下解决“能耗最小化问题”，如图3.6所示。

示例 我们仍然使用图3.1中的例子，并且突出使用新函数 $r = G(p)$ 后的不同。令 $\gamma_1 = 100, \gamma_2 = 200, \gamma_3 = 300$。所以分段线性函数 $p = G(r)$ 是 $p = 0.0072r$ 当 $0 \leq r < 100$



(a) $r^{zm(1)}$ 和 $r^{zm(2)}$ (b) G 调度 (c) 离散调度

图 3.6 离散速率模型下的截断法示例

时, $p = 0.0077r - 0.0515$ 当 $100 \leq r < 200$ 时, $p = 0.0082r - 0.1619$ 当 $200 \leq r < 300$ 时。$r^{zm(1)}$ 和 $r^{zm(2)}$ 在图3.6(a) 中。可以检验，$r^{zm(1)}$ 通过了 energy-constraint-check 检查，而 $r^{zm(2)}$ 没有通过。所以对新函数，$k = 1$，注，对原函数，$k = 3$。组成“扩展”的速率是从 $q_1 = 0$ 开始的 $r_1 = 80$，从 $q_2 = 0$ 开始的 $r_2 = 120$，和从 $q_3 = 2$ 开始的 $r_3 = 150$。我们有 $120 < r_c < 150$ 和 $t_c = \max\{q_2, q_3\} = 2$。对于新函数我们有 $E(t_c) = 2G(120) = 1.74(mJ)$，而对于原函数，$E(t_c) = 1.73(mJ)$。根据公式 (3.5)，$c_u = c_1 = 3$ 和 $r_c = G^{-1}(E_0 - E(t_c)) = 149$，而原来 $c_u = 4$。所以“替换步骤”返回的速率调度是 120 在 $[0, 2)$ 中和 149 在 $[2, 3)$ 中，如图 (b) 所示。在“转化步骤”之后，最优离散速率是 $100(kbps)$ 在 $[0s, 1.6s)$ 和 $[2s, 2.51s)$ 中, $200(kbps)$ 在 $[1.6s, 2s)$ 和 $[2.51s, 3s)$ 中，如图 (c) 所示。

3.6 传输吞吐量最大化

如果能耗最小化问题不存在可行解，那么我们转而研究吞吐量最大化问题，该问题的目标是寻找一个速率调度使得 P 中最多的数据量被传输。我们考虑连续速率模型，因为根据前面讨论的方法，连续速率模型下的最优算法可以转化成离散速率模型下的最优算法。我们首先讨论所有数据报文共享一个截止时间的情况，然后讨论一般化的吞吐量最大化问题。我们假设吞吐量包括部分发送的数据量：如果截止时间到了，剩余的没有发送完毕的数据量就被从队列中丢弃。

3.6.1 公共截止时间的情况

“公共截止时间吞吐量最大化问题”假设所有数据报文共享一个公共的截止时间。在第3.3节中，“公共截止时间能耗最小化问题”可用算法 Common-Truncation 来解决。事实上，该算法可以确定能耗最小化问题没有可行解的情况，并且还可以直接计算吞吐量最大化问题的最优调度。

定理 3.10. 如果算法 Common-Truncation 在结束时 partially-sent 是返回结果，那么

它输出的结果是“公共截止时间吞吐量最大化问题”的最优速率调度。

证明. 该算法返回 partially-sent 值的情况有两种：一、**while** 循环在 $t = T$ 时终止，二、r^{zm} 通过了 energy-constraint-check 检查但是某些时元中的速率比 r_{max} 高。第一种情况意味着在最后一轮迭代中，截断发生后 $t = c_u = T$ 是一个能量临界点。根据引理3.3，在 T 之前所有的采集能量都被耗尽以支持传输最多的数据量。第二种情况中，当 $r^{zm} > r_{max}$ 时我们设 $r^{opt} = r_{max}$，否则我们设 $r^{opt} = r^{zm}$。根据第3.8.1节中关于 r^{zm} 的两个性质，不可能传输再多的数据量了。 □

3.6.2 独立截止时间的情况

在一般化的吞吐量最大化问题中，数据报文是允许有独立的截止时间的。根据我们的初步分析，该问题具有非常复杂的结构，不存在解析解。我们因此提出一个数值解的方法：将该问题建模成凸规划，使用标准凸规划技术在一定误差范围内进行求解。

总共有 $m + 2n$ 个事件点，$e_j, 1 \le j \le m + 2n$，其中包括 m 个采集事件点，n 个到达事件点和 n 个截止事件点。将它们排序，$e_1 \le e_2 \le \ldots \le e_{m+2n}$。我们使用函数 ξ 将每一个事件点映射到该点在事件序列中的序号上。如果 $a_i = e_k$，那么 $\xi(a_i) = k$。函数 ξ 在调度之前可以轻易得到。令时元 $[e_j, e_{j+1})$ 的长度为 $l_j, 1 \le j \le n + 2m - 1$。

那么，吞吐量最大化问题可以如下重新定义。

$$\text{max. } B = \sum_{j=1}^{m+2n-1} r_j l_j, \tag{3.7}$$

$$\text{s.t. } r_j = \sum_{i=1}^{n} r_{ij} \le r_{max}, \;\; j = 1, 2, \ldots, m + 2n - 1, \tag{3.8}$$

$$\sum_{j=1}^{m+2n-1} r_{ij} l_j = \sum_{j=\xi(a_i)}^{\xi(d_i)-1} r_{ij} l_j \le B_i, \;\; i = 1, 2, \ldots, n, \tag{3.9}$$

$$\sum_{j=1}^{\xi(c_k)-1} g(r_j) l_j \le \sum_{i=1}^{k-1} E_i, \;\; k = 2, 3, \ldots, m. \tag{3.10}$$

公式 (3.7) 是实际传输数据量，(3.9) 是松弛后的“时序约束”，(3.10) 是“能量约束”。因为 $p = g(r)$ 是凹函数，该问题显然是一个凸规划。标准凸规划技术可以用来在允许的误差范围内解决该问题。必须指出的是，在该数值解法中，要求的精度越高，所需要的迭代次数（运行时间）也就越多。

3.7 动态联机算法和仿真实验

在本节中，基于截断法，我们设计了一个动态联机速率调度算法。该算法不需要预知到达时间分布、截止时间、数据报文大小、采集时间和采集量等任何信息。我们通过与静态离线最优解进行比较，评估该动态联机算法的性能。

3.7.1 动态联机算法

Algorithm 5 Online-Truncation

步骤 1: 每一个仍然在 FIFO 队列中的数据报文都被当做全新的数据报文对待，更改它们的到达时间为 t（当前时间），保持截止时间保持不变。如果一个数据报文被部分传输，其未传输的部分构成其新的大小。// 注：在 FIFO 队列中，一次只有一个这样的数据报文，其他报文都保持原样。

步骤 2: 对上一步中得到的新数据报文集合 P，在假设当前 t 的初始能量 E 足够大的前提下，计算最优速率调度 r^{zm}。// 因为数据报文具有公共的到达时间 t，而根据第3.8.1节，r^{zm} 的速率是非递增阶梯函数。

步骤 3: 如果 E 不足够支撑 r^{zm}，那么寻找 r_c 将 r^{zm} 截断，使得 $r(t) = \min\{r^{zm}, r_c\}$ 刚够被 E 支持。// 根据第3.4小节中的讨论，在连续速率模型中，如果不再有更多数据报文到达，那么 $r(t)$ 已经最大化了的数据传输量。

步骤 4: 将每一个时元 $[e_i, e_{i+1})$ 都分切成一系列等长的子区间，除了最后一个子区间，长度都是 w，其中 w 是可以调整的参数。// 进行该切分的目的是防止以某一恒定速率传输太长的时间，避免报文达到或者能量采集的到来导致措手不及。

步骤 5: 使用第3.5节中的技术，将每一个子区间一分为二，每一段上使用临近的两个离散速率。令转化后的离散速率调度为 r^{local}。

步骤 6: 按照 r^{local} 来传输队列中的数据报文。

我们的动态联机算法使用贪心的策略：按照现有的信息，计算出最优静态离线调度，按照该调度传输，直到新的事件发生（数据报文到达或者采集发生）。我们的动态联机算法的主要目标的是最大化吞吐量，然而，我们也尽可能的减小能量消耗，因为这两个目标是相关关联的。当一个新的事件在 t 时刻发生的时候，动态联机算法按照算法 Online-Truncation 中给出的步骤重新计算新的调度。

当新的到达事件/采集事件发生的时候，我们的动态联机算法重复上述步骤。为了评估该算法的性能，我们设计进行了多组仿真实验，我们将在下面两节中讨论。

3.7.2 仿真实验设置

鉴于目前尚未有其它算法在允许独立传输延时约束的情况下研究能耗最小化问题和吞吐量最大化问题，所以我们将上一节设计的动态联机算法与静态离线最优算法进行性能比较。

采用 Yang 等人[49] 的假设，我们考虑一个带宽受限的加性高斯白噪声 (AWGN) 信道，带宽 W = 1MHz，噪声强度是 $N_0 = 10^{-19}$W/Hz。设收发节点之间的距离是 1km，信道路径衰减率 h 是 110dB。那么，我们有 $r = g^{-1}(p) = W\log_2(1+\frac{ph}{N_0W}) = 10^3\log(1+0.1p)$，其中 p 的单位是毫瓦，r 的单位是 kbps。传输速率是离散的，可选的值集合是 $\{0, 50, 100, \ldots, 600\}$kbps。在我们的仿真中，我们设置算法参数 w 是 0.2s。

与已有的工作一致，我们也假设数据报文到达时间满足泊松分布，并且设数据报文平均间隔时间是 14s。数据报文大小满足均匀分布 $\mathcal{U}(0.01z, 1.99z)$，其中 z 是平均报文大小，在本仿真中，该值会以 100kb 为步长，从 400kb 变化到 1000kb。数据报文延时约束的值是从 $(0.2q, 1.8q)$ 中随机选取的，其中 q 是平均延时约束，并且设置成 20s。我们假设能量采集的发生也满足泊松随机过程，在本仿真中，其平均发生间隔设置成不同的值，从 12s 到 18s 不等。能量采集量满足均匀分布 $\mathcal{U}(0, 2h)$，其中 h 是平均采集量，本仿真中，以 1mJ 为步长，从 2mJ 变化到 8mJ。

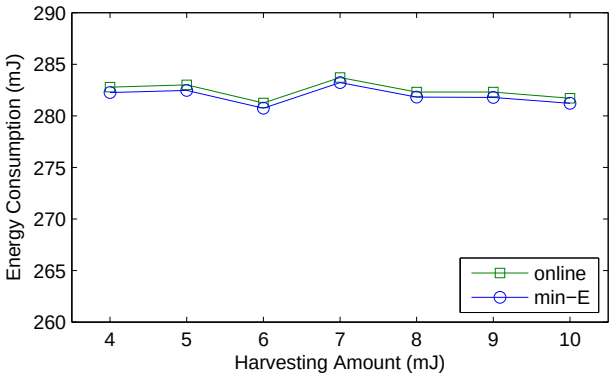
本节中用到的所有数值（图片上的点和表述中的值）都是 150 个随机实例仿真结果的平均值。在每一个实例中，100 个数据报文和 100 个能量采集按照上述模型随机生成。注：所有随机生成的截止时间都按照早到的数据报文具有早的截止时间进行了排序。

3.7.3 仿真结果

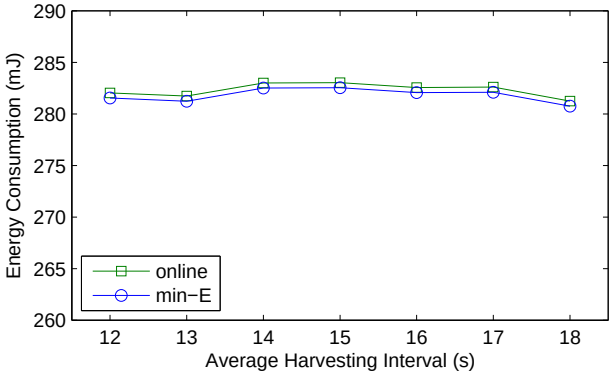
图3.7展示了联机动态算法的能量消耗与静止离线最优算法的能量消耗的比较。为了公平起见，只有当联机动态算法和静止离线最优算法同时都完成了所有数据报文传输时，我们才比较它们的能量消耗。设置默认平均数据报文大小 z=400kb；默认平均采集间隔是 12s；默认平均采集量 h=8.0mJ，然后分别改变这三个参数来研究它们对算法性能的影响，其结果如图3.7(a)(b)(c) 所示。我观察到，动态联机算法在所有三幅图中都仅比静态离线最优算法略多消耗一点能量。在 (a) 和 (b) 中，因为传输的数据总量未变化，所以能量消耗曲线变化不大。在 (c) 中，两条曲线随着平均数据报文大小的增加而增加。

在图3.8中，我们将动态联机算法的数据传输量与静态离线算法的最优解（最大吞吐量）进行比较。在默认设置中，平均数据报文大小 z=500kb；平均采集间隔是 15s；平均采集量 h=5.0mJ。这三个参数对算法性能的影响分别展示在 (a)、(b) 和 (c) 中。我们观察到在所有图中，我们动态联机算法的曲线与最大吞吐量的曲线非常接近。(a) 显示了吞吐量随着平均采集量的增加而增加。(b) 显示了吞吐量随着采集间隔的加大而缓慢减小，这是因为采集间隔越大，采集到的总能量越少，能支撑的传输量也越小。(c) 显示了数据传输量随着平均数据报文大小的增加而增长。

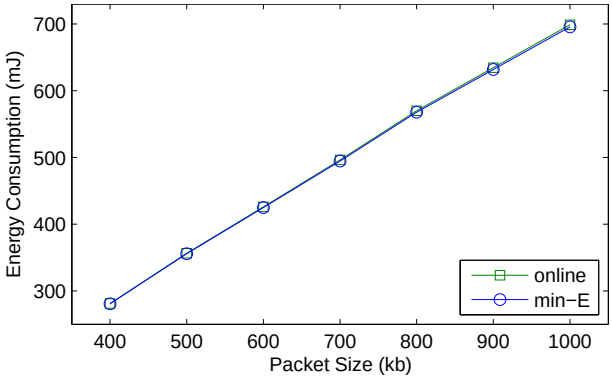
所有上述图片显示，我们的动态联机算法平均可以达到最优吞吐量的 93%。在能量消耗方面，我们的算法平均可以达到的比值为 $E^*/E > 93\%$，其中 E^* 表示最优解中的



(a)



(b)



(c)

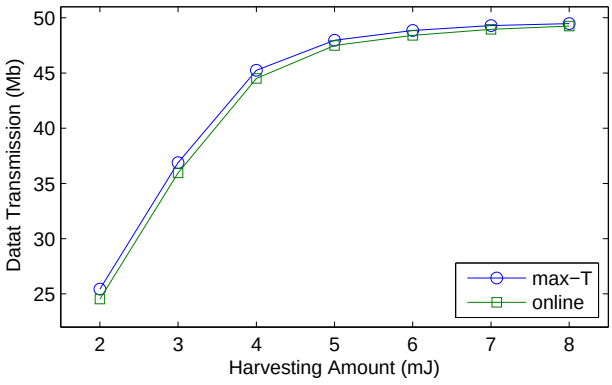
图 3.7 能耗最小化问题的动态联机算法与静态离线算法能量消耗比较

最小能量消耗。

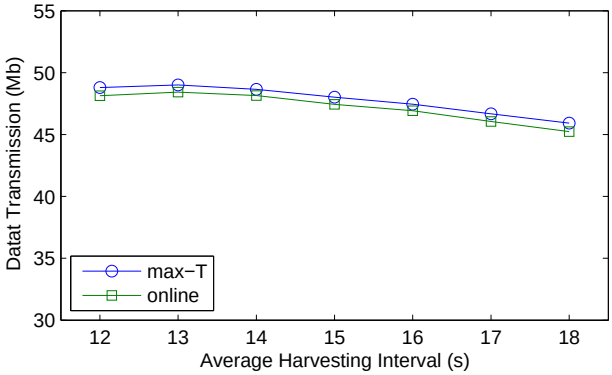
3.8 详细证明和补充信息

3.8.1 “ZM 速率调度算法”的简短介绍

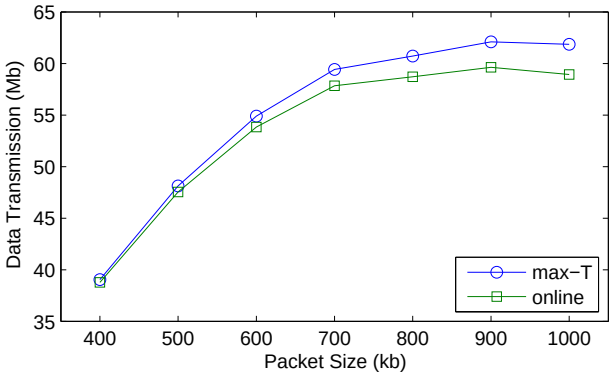
令 $A(t) = \sum_{i:a_i \le t} B_i$ 表示在 $[0, t)$ 中到达的数据累计总量；令 $D_{min}(t) = \sum_{i:d_i \le t} B_i$ 表示那些不在 t 时刻之前发送，就将无法满足其 QoS 要求的数据的累计总量。图3.9展



(a)



(b)



(c)

图 3.8 吞吐量最大化问题的动态联机算法与静态离线算法数据传输量比较

示了累积“数据 -时间”图。$A(t)$ 的曲线称为到达曲线；$D_{min}(t)$ 的曲线称为最少发送曲线。一个颇为关键的观察是可行解的发送曲线一定在到达曲线和最小发送曲线之间。现在，将发送曲线想象成一根皮筋，将其一端栓在原点，另一端经过 $D_{min}(T)(=A(T))$，如图3.9所示，如果我们将皮筋拉直，那么它构成的形状就是最优发送曲线 $D^{zm}(t)$。

$D^{zm}(t)$ 曲线的斜率就是最优传输速率 r^{zm}。r^{zm} 有两个重要的性质[14, 23]，如下。

- r^{zm} 的下降只可能发生在截止时间点上。如果 r^{zm} 在 t 上降低，那么所有具有更早截止时间（包括 t）的数据报文都在 t 之前被发送完，而且其他任何数据报文都没有开

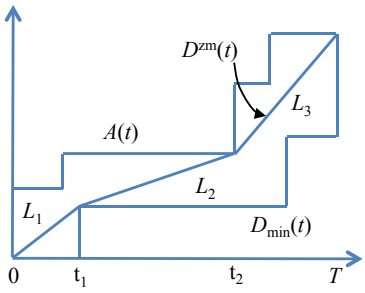


图 3.9 $A(t)$ 曲线，$D_{min}(t)$ 曲线以及 $D^{zm}(t)$ 曲线

始传输。

- r^{zm} 的增加只可能发生在到达点上。如果 r^{zm} 在 t 上增加，那么所有在 t 之前到达的数据报文在 t 之前全部传输完毕。

3.8.2 引理3.3的证明

我们用反证法证明该引理。假设速率调度 $r(t)$ 最大化了 $[0,t)$ 上发送的数据量，但是在某些时元中却有 $r \neq r^{opt}$。那么，必定存在一个时元 p 其中 $r > r^{opt}$，因为否则的话 $r(t)$ 传输的数据量会比 r^{opt} 少。令 r_p 和 r_p^{opt} 分别表示在时元 p 中 $r(t)$ 和 r^{opt} 的速率。那么，一定存在另外一个时元 q 其中 $r < r^{opt}$，因为 t 是 r^{opt} 的一个能量临界点，而 $r(t)$ 不能在 $[0,t)$ 中比 r^{opt} 消耗更多能量。相似的定义 r_q 和 r_q^{opt}。根据引理3.1，我们有 r^{opt} 在 T 以外不会下降。

如果 $p < q$，那么我们必定有 $r_p^{opt} = r_q^{opt}$，否则的话，根据引理3.2，两者之间一定存在一个数据临界点，而 $r(t)$ 在该点之前比 r^{opt} 传输了更多的数据量，这与数据临界点的定义矛盾。所有，$r_p > r_p^{opt} = r_q^{opt} > r_q$。显然，$r_p$ 和 r_q 可以使用“均衡”的方法传输更加多的数据量。这和 $r(t)$ 传输了最多的数据量的假设矛盾。如果 $q < p$，那么我们有 $r_q < r_q^{opt} \leq r_p^{opt} < r_p$。因为 $r(t)$ 在时元 q 结束时比 r^{opt} 传输的数据量更少，所以在时元 q 结束的时候，系统里既有剩余能量也有剩余数据量。那么，r_p 和 r_q 就必定可以用“均衡”的方法在 $[0,t)$ 中传输更多的数据量。这也和 $r(t)$ 是最优矛盾。

3.8.3 定理3.3的证明

显然如果速率 $r(t)$ 最优，那么其不能被“均衡”。

我们现在证明，如果速率调度 $r(t)$ 在不违反“时序约束”和“能量约束”的前提下，无法被“均衡”，那么这样的速率调度是最优的。假设，$r(t)$ 却不是最优的。那么，与最优调度 r^{opt} 相比，令时元 p 是第一个使得 $r(t) \neq r^{opt}$ 的时元，将该时元中的两个速率分别记为 r_p 和 r_p^{opt}。共有两种情况：$r_p^{opt} > r_p$ 和 $r_p > r_p^{opt}$。我们证明第一种情况，因为第二种情况的证明相似。由于 $r_p^{opt} > r_p$ 并且两个调度都发送完了所有数据量，那么必定存在至少一个后续时元 q 其中 $r_q^{opt} < r_q$。令 q 是第一个这样的时元。

在时元 p 和 q 之间区间中，没有 r^{opt} 的截止临界点，因为否则的话 $r(t)$ 不能在该截止临界点之前比 r^{opt} 传输更少的数据量。因此，根据引理3.4有 $r_p^{opt} \leq r_q^{opt}$。我们有下面两个关于 $r(t)$ 的事实：(1) $r_p < r_p^{opt} \leq r_q^{opt} < r_q$；(2) 在时元 p 结束的时候，数据和能量的余量都不是零。所有 r_p 和 r_q 可以被“均衡”。这是一个矛盾，所以 $r(t)$ 是最优的。

3.8.4 定理3.4的证明

显然，至少存在一个时元其有 $r^{opt(i+1)} > r^{opt(i)}$，这是因为 $r^{opt(i+1)}$ 多完成了一个数据报文的发送。令时元 p 是第一个这样的时元。我们首先证明在 p 之前有 $r^{opt(i+1)} = r^{opt(i)}$，再证明在 p 之后有 $r^{opt(i+1)} > r^{opt(i)}$。

在时元 p 之前，有 $r^{opt(i+1)} = r^{opt(i)}$。

假设相反，在 p 之前某几个时元上 $r^{opt(i+1)} < r^{opt(i)}$，令 q 是其中最后一个。令 t 是时元 q 结束的时间点。那么，在时间点 t 上，要么 $r^{opt(i)}$ 下降，要么 $r^{opt(i+1)}$ 上升。如果 $r^{opt(i)}$ 下降，那么根据引理3.4，t 是一个截止临界点。然而速率调度 $r^{opt(i+1)}$ 在该截止临界点之前比 $r^{opt(i)}$ 传输的数据量更少，因此某些数据报文必定错过了它们的截止时间，这和 $r^{opt(i+1)}$ 的可行性矛盾。如果 $r^{opt(i+1)}$ 在 t 点上升，那么根据引理3.2，t 要么是一个能量临界点，要么是一个数据临界点。如果 t 是一个能量（数据）临界点，那么速率调度 $r^{opt(i+1)}$ 在 t 点将能量耗尽（数据发完）。这就暗示了能量（数据）不足以在 t 之前支撑传输速率 $r^{opt(i)}$，这是一个矛盾。

在 p 之后的时元中，$r^{opt(i+1)} > r^{opt(i)}$。

假设相反，某些时元上有 $r^{opt(i+1)} \leq r^{opt(i)}$，令 q 是其中第一个。所以，在时元 q 的开始点，要么 $r^{opt(i+1)}$ 下降，要么 $r^{opt(i)}$ 上升。两者都不可能。证明和上述相似。

3.9 本章小结

在本节中，我们研究了无线能量自供给设备上发送一组具有独立传输延时约束的数据报文的离散速率调度问题，我们分别研究了能量最小化和吞吐量最大化两个子问题。基于在数据报文传输延时与能量消耗之间基础性的折衷，我们提出“截断法”来计算能量最小化的连续速率调度。然后，提出一般化的框架可以将任何连续速率调度算法转换为离散速率调度算法，只要原算法的最优性是基于“速率 -功率”函数的凸性质的，那么该框架就能保证新算法具有与原算法相同的最优性。如果采集的能量不足以将所有数据报文发送完毕，那么我们研究吞吐量最大化的问题。当所有数据报文共享公共截止时间时，我们的“截断法”可以直接计算最大传输量，否则的话，我们利用凸规划求解数值解。在本节中，我们也设计了一个动态联机算法最大化数据传输量。仿真结果显示，我们的动态联机算法非常高效，平均与最优解的近似度很高。

第四章　移动传输中能量自供给设备数据量最大化

本章考虑一个移动汇聚节点定期沿着预定义的路径移动，并从部署在路径旁的能量自供给无线传感器上收集数据。时间被等分成 m 个时隙，一个时隙只允许最多一个传感器传输数据。一个基本问题是：某一个传感器应该在哪一个时隙中传输、应该使用多大的传输功率，可以使得移动汇聚节点在 m 个时隙中收集的数据量最大化？已有工作证明该问题是的 NP 完全问题，并且设计了近似算法。本章发现该问题可以在多项式时间内找到最优解，前提是不预设每一个传感器的传输功率，而是将其视为算法输出的一部分。本章设计了“水箱”新技术，与动态规划一起，成功的解决了该问题。本章还设计了动态联机算法，仿真实验明确显示该算法的性能远远优于已有算法。本章与前两章是一脉相承的整体，是问题从静止到移动的顺承与扩张。

4.1 引言

无线传感器网络正成为许多重要应用领域的关键技术，包括环境监测、地域监控等。因为传感器节点的能量供应有限，延长无线传感器网络的寿命是该领域的一项及其重要的研究目标。为此，出现了两项新技术：移动汇聚节点技术[38–47] 和能量自供应技术[48–57] 本章同时使用这两项新技术，研究目标不再是最大化网络工作时长，而是最大化数据收集量，虽然这两个目标是相互关联的。

本章考虑一个移动汇聚节点以恒定速率沿一条预定义的路径移动，并且从部署在路径旁边的传感器上直接收集数据。假设传感器都具备能量自供应功能，并且可以自行控制传输功率进而控制传输速率。从路径的一端移动到另一端称的时间被等分成 m 个时隙，编号从 1 到 m。相应的，路径也被等分成 m 个路段，与时隙一一对应。在移动过程中，如果移动汇聚节点在某时隙中其处于传感器 s 的传输范围内，那么，这该时隙称为对传感器 s 可用，所有这样的时隙记做 $A(s)$。只有在集合 $A(s)$ 中，移动汇聚节点可以从 s 上收集数据。移动汇聚节点可能会同时进入到多个传感器的传输范围内，某个时隙可能同时对多个传感器可用。但是，移动汇聚节点在一个时隙中只能接收一个传感器的传输的数据，所以需要一个选择策略，从这些传感器中选一个进行传输。由于传感器采集的可用能量有限和可用时隙也有限，本章研究的问题是：如何安排传感器的传输时隙、如何设置传输功率，才能最大化总数据传输量？本章将该问题称为移动汇聚节点数据采集最大化问题 (Data Collection Maximization by Mobile sink, DCMM)，也称移动传输中能量自供给设备数据量最大化。

许多潜在的应用场景都可以被建模成 DCMM 问题。比如，在交通违章监控中，监控传感器部署在高速公路两旁，它们相互协作确定超速的汽车并拍照。装备了数据收集设备的警车每天从公路上驶过进行数据收集[62]，称为移动汇聚节点。其他应用场景还包括天然气管道、石油管道或者是水管的监控，桥梁建筑结构健康监控[61] 等线形监控。对于地域监控，所有的监控数据可以预传输到路径旁边的传感器上，再由移动汇聚节点在一次移动中收集。因此，DCMM 问题是一个基础的理论问题，具有广泛的应用背景。

4.2 问题定义

本节给出移动传输中能量自供给设备数据量最大化问题，或者称为 DCMM 问题的正式定义。假设 n 个能量自供给无线传感器，$s_1, s_2, \ldots, s_n$，沿着路径 P 部署。一个移动汇聚节点 S 以速度 γ 沿着 P 匀速移动，当进入传感器的传输范围时，可以收集传感器上采集的数据。从 P 的一端到另一端的移动称作“周游”。移动汇聚节点周期性地在两个端点之间做周游。显然，只需要考虑一次周游上的 DCMM 问题即可，因为，设计的算法可以在每一次周游上应用。和文献[59] 一样，本章设定如下符号和假设：

(1) 汇聚节点 S 需要 m 个时隙完成一次周游，其中每一个时隙都具有固定的长度 τ，所以，一次周游的总时间是 $m\tau$。

(2) m 个时隙按顺序标记为 $1, 2, \ldots, m$。其中时隙 k 占据了时间区间 $(t_{k-1}, t_k]$，$t_k = t_{k-1} + \tau = k\tau, t_0 = 0, 1 \leq k \leq m$。为了简单起见，用 $(k-1, k]$ 代表时隙 k 或者时间区间 $(t_{k-1}, t_k]$。相似的，如果 $0 \leq i < j \leq m$，使用 $(i, j]$ 代表时间区间 $(t_i, t_j]$，即从时隙 $i+1$ 到时隙 j 的连续时隙；如果 $i = j$，则 $(i, j] = \varnothing$。

(3) 相应的，路径 P 的总长度是 $L = \gamma m\tau$，它被切分为长度为 $\gamma\tau$ 的 m 个小段，称为“路段”。路段 k 代表 S 在时隙 k 中移动的路径。

(4) 能量自供给无线传感器将采集到的能量存储在电池中。令 $E_i, 1 \leq i \leq n$ 表示在当前的周游开始时，传感器 s_i 中的可用电量。和文献[59] 一样，本章假设传感器上采集了足够的数据等待传输给汇聚节点。注，本章使用的新方法可以容易地对传感器上待传输的数据总量进行限制。我们将这部分的工作留给读者，以便专心解决新问题。

(5) 路径 P 相对平滑而且足够长，从任意一个传感器的角度看，路径近似是直线。所以，本章假设传感器 s_i 的可用时隙集合 $A(s_i)$ 是连续的时隙，即 $A(s_i) = (b_i, e_i] =$ {slot $b_i + 1$, slot $b_i + 2, \ldots,$ slot e_i}，$b_i \leq e_i$；如果 $b_i = e_i$，则 $A(s_i) = \varnothing$。

(6) 虽然 s_i 和 S 之间的距离不断在变化，但是在时隙 j 中（或者路段 j 中），这样的变化非常小，因而假设它是一个常数 d_{ij}，而且可以提前预知。

所有上述假设都与 Ren，Liang 和 Xu 在文献 [59] 中所做的假设一样。为了将功率控制引入到解决方法中，本章还要介绍几个新的符号和假设，如下。

(7) 令 p_{ij} 和 r_{ij} 分别表示传感器 s_i 在时隙 j 中使用的传输功率和数据速率，其中 $j \in A(s_i)$，$1 \le i \le n$。r_{ij} 和 p_{ij} 都是待计算的变量。p_{ij} 的取值可以是允许范围中的任意值：

$$0 \le p_{ij} \le p_i^{max}, \tag{4.1}$$

其中，p_i^{max} 是传感器 s_i 的最大传输速率，由其硬件决定。

(8) 本文采用 AWGN 信道模型，其中 p_{ij} 和 r_{ij} 的关系满足“香农容量定理 (Shannon-Hartley Theorem)”[65]。

$$r_{ij} = \frac{1}{2}\log(1 + \frac{p_{ij}}{d_{ij}^{\alpha}}) \tag{4.2}$$

其中 α 被称为“路径衰减指数”，其取值通常在 2 和 6 之间。

说明 1 有一些研究[42–46, 59]假设 $r \propto \frac{p}{d^{\alpha}}$，这是公式 (4.2) 的简化模型。另有一些研究[48–51, 57]采用模型 $r = \frac{1}{2}\log(1+p)$，这也是公式 (4.2) 的一个特例情况。Chakrabarti 等人[47]采用了公式 (4.2) 中的模型，但是却假设 p 是固定的而不是可以调整的。本章的模型是最通用的模型，涵盖了其他所有模型。

说明 2 根据大多数研究[42–56, 59]，本章允许速率/功率可以在取值范围内取任意实数值。尽管有些无线设备只允许取某些离散的数值，但是根据第3.5节中提出的通用框架，连续速率模型下设计的算法可以很容易的转化为离散速率模型下的算法。所以，本节允许速率/功率连续取值。

定义 4.1 (时隙分配). 给定一条路径 P，一个移动汇聚节点 S，一组能量自供给的无线传感器 s_i，$1 \le i \le n$ 和 m 个时隙，“时隙分配”是指将一组连续时隙 $(\beta_i, \epsilon_i]$ 分配给传感器 s_i，$1 \le i \le n$，并且使得：

(a) $(\beta_i, \epsilon_i]$ 是可用集合 $A(s_i)$ 的一个子集，$(\beta_i, \epsilon_i] \subset A(s_i) = (b_i, e_i]$。$(\beta_i, \epsilon_i]$ 可以是空集，如果 $\beta_i = \epsilon_i$ 的话。

(b) n 个“分配时隙”$(\beta_i, \epsilon_i]$，$1 \le i \le n$ 互不相交。不失一般的，假设沿路径 P 方向进行标记：

$$0 \le \beta_1 \le \epsilon_1 \le \beta_2 \le \epsilon_2 \le \ldots \le \beta_n \le \epsilon_n \le m. \tag{4.3}$$

定义 4.2 (功率控制). 给定时隙分配，将 $(\beta_i, \epsilon_i]$ 分配给传感器 s_i，$1 \le i \le n$，“功率控制”是指在公式 (4.1) 允许的范围内为传感器 s_i 确定时隙 $j, j = \beta_i + 1, \beta_i + 2, \ldots \epsilon_i$ 中使用的传输功率 p_{ij}。要求总能耗满足：

$$\sum_{j=1}^{m} \tau p_{ij} = \sum_{\beta_i+1}^{\epsilon_i} \tau p_{ij} \le E_i. \tag{4.4}$$

当 $j \notin (\beta_i, \epsilon_i]$ 时，$p_{ij} = 0$。

说明 3 为了叙述方便，当时隙 j 对 s_i 不可用时，我们设定 $p_{ij} = r_{ij} = 0$ 。这样做的好处是每一个时隙对每一个传感器都是可用的。

显然，根据香农容量公式，从所有 n 个传感器传输到移动汇聚节点的数据总量 B 是

$$B = \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{\tau}{2}\log(1+\frac{p_{ij}}{d_{ij}^{\alpha}}). \tag{4.5}$$

定义 4.3. 给定一条路径 P，一个移动汇聚节点 S，一组能量自供给的无线传感器 s_i，m 个时隙，s_i 与路段 j 的距离 d_{ij}，以及可用能量 E_i，$1 \le i \le n, 1 \le j \le m$，DCMM 问题是确定“时隙分配”和“功率控制”使得由公式 (4.5) 确定的总数据传输量最大。

4.3 静态离线最优算法

在本节中，我们解决静态离线的 DCMM 问题。在该问题中，移动汇聚节点了解全局信息，包括所有传感器的位置、可用时隙、电池总能量等。我们提出一个全新的“水箱法”，与动态规划一起，成为解决静态离线 DCMM 问题的关键。

4.3.1 单传感器 DCMM 问题和凸规划建模

我们首先研究简化的 DCMM 问题，即路径上只部署一个传感器。该问题简称为 DCMM-1 问题。DCMM-1 问题非常简单，以至于可以用凸规划建模。但是，我们并不试图用标准的凸规划的方法解决该问题，而是通过使用凸规划分析该问题，发现一些性质。在该过程中，我们发现一个非常有趣的最优性质，基于该性质，我们提出一个有效而且有意思的新方法。

首先，让我们简要地介绍一下解决凸规划常用的工具 KKT 条件[60]。

假设一个普通的凸规划

$$\min f_0(x) \text{ s.t. } f_i(x) \le 0, i = 1, 2, \ldots, n. \tag{4.6}$$

其中函数 $f_i, i = 0, 1, \ldots, n$ 都是凸函数。我们假设该规划有解而且函数 $f_i, i = 0, 1, \ldots, n$ 都可导。令 λ_i 是 $f_i(x), i = 1, 2, \ldots, n$ 对应的拉格朗日乘子。那么，x 和 $\{\lambda_i | 1 \le i \le n\}$ 是可行的对偶解的充分必要条件（KKT 条件）[60] 是

$$f_i(x) \le 0, \quad i = 1, 2, \ldots, n, \tag{4.7}$$

$$\lambda_i \ge 0, \quad i = 1, 2, \ldots, n, \tag{4.8}$$

$$\lambda_i f_i(x) = 0, \quad i = 1, 2, \ldots, n, \tag{4.9}$$

$$\nabla f_0(x) + \sum_{i=1}^{n}\lambda_i \nabla f_i(x) = 0. \tag{4.10}$$

其次，我们看 DCMM-1 问题，该问题如图4.1所示。其中可用时隙是 (0,7]。每一个圆圈代表汇聚节点在每个时隙的位置，与圆圈相连的虚线表示该位置与传感器之间的距

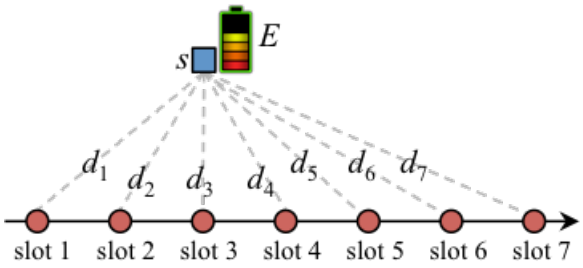


图 4.1 DCMM-1 问题示例

离 s。问题的目标是为每一个时隙决定一个传输功率 p，使得在电池可用能量 E 的限制下，传感器 s 传输的数据量（汇聚节点 S 的数据采集量）最大。令传感器 s 部署在路径 P 的旁边，s 和路段 j 之间的距离是 $d_j, 1 \leq j \leq m$。令 s 上可用的能量为 E，可用的时隙为 $A(s) = (b, e]$。那么，DCMM-1 可以建模成如下优化问题。

$$\text{maximize} \sum_{j=b+1}^{e} \frac{\tau}{2} \log(1 + \frac{p_j}{d_j^{\alpha}}), \tag{4.11}$$

$$\text{subject to} \sum_{j=b+1}^{e} \tau p_j \leq E, \tag{4.12}$$

$$p_j \geq 0, j \in (b, e], \tag{4.13}$$

$$p_j \leq p^{max}, j \in (b, e]. \tag{4.14}$$

上述公式中需要最终计算的变量是 p_j，$b+1 \leq j \leq e$。

再次，我们为不等式 (4.12) 关联拉格朗日乘子 μ，为不等式 (4.13) 关联拉格朗日乘子 $\lambda_i, j = b+1, b+2, \ldots e$，为不等式 (4.14) 关联拉格朗日乘子 $\omega_i, j = b+1, b+2, \ldots e$。那么，对应的拉格朗日函数 [60] 定义如下：

$$L(p, \mu, \lambda, \omega) = -\sum_{j=b+1}^{e} \frac{\tau}{2} \log(1 + \frac{p_j}{d_j^{\alpha}}) + \mu(\sum_{j=b+1}^{e} \tau p_j - E) - \sum_{j=b+1}^{e} \lambda_j p_j + \sum_{j=b+1}^{e} \omega_j (p_j - p^{max}), \tag{4.15}$$

其中 p, λ, ω 分别表示 $\{p_j\}$，$\{\lambda_j\}$ 和 $\{\omega_j\}$，$b+1 \leq j \leq e$。

现在，充分必要的 KKT 条件，除了公式 (4.12)(4.13)(4.14) 还包括：

$$\lambda_j p_j = 0, \quad \forall j \in (b, e], \tag{4.16}$$

$$\lambda_j \geq 0, \quad \forall j \in (b, e], \tag{4.17}$$

$$\omega_j (p_j - p^{max}) = 0, \quad \forall j \in (b, e], \tag{4.18}$$

$$\omega_j \geq 0, \quad \forall j \in [b, e], \tag{4.19}$$

$$\mu(\sum_{j=b+1}^{e} \tau p_j - E) \geq 0, \tag{4.20}$$

$$\mu \geq 0, \tag{4.21}$$

$$-\frac{\tau}{2(\frac{1}{h_j} + p_j)} + \mu\tau - \lambda_j + \omega_j = 0, \quad \forall j \in (b, e]. \tag{4.22}$$

接下来，我们要分析该 KKT 条件。

根据 (4.22)，我们有

$$p_j + d_j^{\alpha} = \frac{1}{2(\mu + \frac{\omega_j - \lambda_j}{\tau}) \ln 2}, \ \forall j \in (b, e]. \tag{4.23}$$

根据 (4.13)(4.16)(4.17)，对 λ_j 和 p_j 的取值总共有三种可能的组合，表4.1(a) 列出了这三种情况。相似的，根据 (4.14)(4.18)(4.19)，对 ω_j 和 p_j 的取值也有三种可能的组合，表4.1(b) 列出了这三种情况。联合表4.1(a) 和表4.1(b)，我们可以获得 λ_j，ω_j 和 p_j 的所有可能的取值组合，表4.1(c) 列出了所有的情况。

表 4.1 拉格朗日乘子的取值范围组合表

(a) λ_j 和 p_j

Case	λ_j	p_i
1	0	>0
2	>0	0
3	0	0

(b) ω_j 和 p_j

Case	ω_j	p_i
1	>0	p^{max}
2	0	$<p^{max}$
3	0	p^{max}

(c) p_j，ω_j 和 λ_j

Case	p_i	λ_j	ω_j
1	$=0$	≥ 0	0
2	$\in (0, p^{max})$	0	0
3	$=p^{max}$	0	≥ 0

从表4.1(c)，我们可以看出，p_j, ω_j 和 λ_j 的取值总共有三种可能的情况。让我们仔细检查公式 (4.23)，逐一考虑这三种情况，如下。其中令 $u = \frac{1}{2\mu \ln 2}$。

情况 1, $p_j = 0$，$\lambda_j \geq 0$ 和 $\omega_j \geq 0$。我们有 $p_j + d_j^{\alpha} = d_j^{\alpha}$。因为 $\lambda_j \geq 0$ 和 $\omega_j = 0$，我们有 $d_j^{\alpha} = \frac{1}{2(\mu + \frac{\omega_j - \lambda_j}{\tau}) \ln 2} \geq \frac{1}{2\mu \ln 2} = u$；

情况 2, $p_j \in (0, p^{max})$，$\lambda_j = \omega_j = 0$。我们有 $p_j + d_j^{\alpha} = \frac{1}{2\mu \ln 2} = u$。因为 $0 \leq p_j \leq p^{max}$，我们 $d_j^{\alpha} < u < p^{max} + d_j^{\alpha}$。

情况 3，$p_j = p^{max}$，$\lambda_j = 0$，$\omega_j \geq 0$。我们有 $p_j + d_j^{\alpha} = p^{max} + d_j^{\alpha}$。因为 $\lambda_j = 0$ 和 $\omega_j \geq 0$，我们有 $p^{max} + d_j^{\alpha} = \frac{1}{2(\mu + \frac{\omega_j - \lambda_j}{\tau}) \ln 2} \leq \frac{1}{2\mu \ln 2} = u$。

作为总结，我们为 $p_j + d_j^\alpha$ 的赋值构建如下一个公式。

$$p_j + d_j^\alpha = \begin{cases} d_j^\alpha & \text{if } u \le d_j^\alpha, \\ u & \text{if } d_j^\alpha < u < p^{max} + d_j^\alpha, \quad \forall j = [b, e]. \\ p^{max} + d_j^\alpha & \text{if } u \ge p^{max} + d_j^\alpha. \end{cases} \tag{4.24}$$

从公式 (4.24) 中，我们总结 DCMM-1 问题的最优性质为引理 4.1。

引理 4.1. 在 DCMM-1 问题的最优解 $\mathbf{p} = \{p_j | j = b+1, b+2, \ldots, e\}$ 中，必然存在一个唯一的变量 u 满足公式 (4.24)。

证明. 从上面的讨论中可以明确，如果我们知道最优解的 $u(>0)$ 值，那么根据公式 (4.24) 我们可以唯一地确定最优解 $\mathbf{p} = \{p_j | j = b+1, b+2, \ldots, e\}$，因为 d_j^α 和 p^{max} 是已知常数。具体地，当 u 在 d_j^α 和 $p^{max} + d_j^\alpha$ 之间时，$p_j + d_j^\alpha$ 等于 u；否则，$p_j + d_j^\alpha$ 等于边界值，当 $u \le d_j^\alpha$ 时等于 d_j^α，当 $u \ge p^{max} + d_j^\alpha$ 时等于 $p^{max} + d_j^\alpha$。此外，u 的值也是可以唯一确定的，因为根据公式 (4.12)，$p_j, j = b+1, b+2, \ldots, e$ 的和必须满足 $\sum_{j=b+1}^{e} p_j = \frac{E}{\tau}$。根据公式 (4.24)，$p_j, j = b+1, b+2, \ldots, e$ 的值随着 u 的增加而单调递增，所以 u 的取值是唯一的。 □

要解决 DCMM-1 问题，关键就是要找到该唯一的 u 值。

4.3.2 水箱技术

为了寻找该唯一的 u 值，我们在本小结中设计并介绍“水箱技术”。

我们为每一个时隙 j 构造一个圆柱形水箱 j，如图4.2所示。所有水箱完全相同：其高度是 p^{max}，其底面积是单位面积，因此容积都是 p^{max}。这些水箱被放置在不同的高度上：水箱 j 的地面放置的高度是 d_j^α，显然，其顶高度是 $p^{max} + d_j^\alpha$。如图 4.2所示，一根水平放置在地面（高度 0）的细长水管将每一个水箱连接起来，各个水箱之间的水因此可以自由流动，假设未有水溢出和泄露。水管的一端封闭，另一端有一个水阀，可以控制外部的高压水流进入水箱系统。

定理 4.1. 假设上述水箱系统中注入体积为 E/τ 的水，忽略水阀和水管中的水量，如果此时系统中的水位是 u 并且水箱 j 中的水量为 p_j，$b+1 \le j \le e$，那么 u 和 $\mathbf{p} = \{p_j | j = b+1, b+2, \ldots, e\}$ 必定满足公式 (4.24)。即，$\mathbf{p}$ 是 DCMM-1 问题的最优解。

证明. 我们分析当水箱系统中注入 E/τ 体积水时水位 u 和 p_j 的关系。如图 4.2所示，因为水箱底为单位面积，所以水箱 j 中的水位高度（相对于水箱底）和总水量都是 p_j。因此，当 $u \le d_j^\alpha$ 时我们有 $p_j = 0$；当 $d_j^\alpha < u < p^{max} + d_j^\alpha$ 时我们有 $p_j + d_j^\alpha = u$；当 $u \ge p^{max} + d_j^\alpha$ 时我们有 $p_j = p^{max}$。这正好与公式 (4.24) 完全匹配。此外，总注入水量是 E/τ，所以我们有 $\sum_{j=b+1}^{e} p_j = \frac{E}{\tau}$。$\mathbf{p}$ 就是 DCMM-1 问题的最优解。 □

因此，接下来，我们设计一个快速有效的算法计算该水位 u。

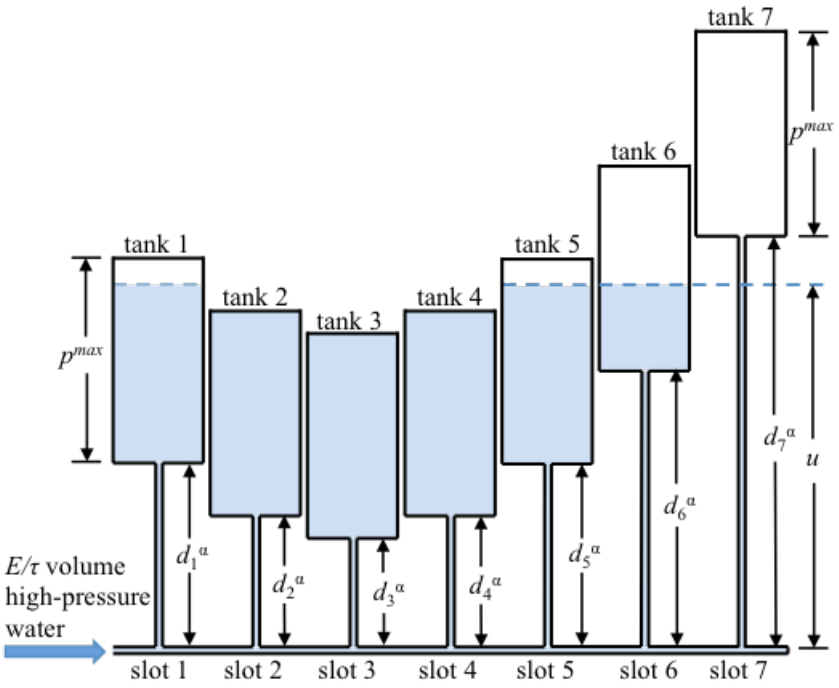


图 4.2 水箱技术的示例

4.3.3 计算水位的算法

在本节中，我们设计一个快速计算水位 u 的算法。在水箱系统中，由于水是可以在水箱之间自由流动的，所以重新安排水箱顺序不会影响 u 的值。我们按照水箱底面高度递增的顺序重新安排水箱，如图 4.3所示。令 $floor(j) = d_j^{\alpha}$，$ceiling(j) = floor(j)+p_i^{max}$，$b+1 \leq j \leq e$。不失一般地，我们假设 $floor(b+1) \leq floor(b+2) \leq \ldots \leq floor(e)$。算法的总体思想如下。

起初，所有的水箱都是空的。想象我们缓缓地将 E/τ 的水量注入这些空水箱中，水一定会一个一个地、从左到有流入各个水箱中。令 u 是当前的水位，初始时 $u = floor(b+1)$，令 remain_water 表示剩余的待注入水箱的水量，初始时 $remain_water = E/\tau$。在当前水位为 u 时，算法记录了最右一个满足 $floor(f) \leq u$ 的水箱 f，初始时 $f = b+1$。此时，最左边几个水箱可能已经满了，令水箱 c 是最右边一个已满的水箱，$ceiling(c) \leq u$，初始时 $c = b$。因此，水箱 $c+1, c+2, \ldots, f$ 都有水但是都未满。此时，当更多的水注入到水箱系统时，水首先流入水箱 $c+1, c+2, \ldots, f$ 中，最后才流入水箱 $f+1$。在水注入的时候，两个事件其一一定会发生：水开始流入水箱 $f+1$，水箱 $c+1$ 满了。很容易计算哪个事件会早发生，以及需要注入多少水可以使该事件发生。那么，当这么多水注入后，要么 c 更新为 $c+1$，要么 f 更新为 $f+1$。之后，算法又会重复该过程，直到剩余的水不足以使得下一个新事件发生。当此之时，剩余的水流入到水箱 $c+1, c+2, \ldots, f$ 中，并且形成了最终的水位 u。该算法的细节和伪代码在算法 WaterLevel 中。

因为水箱的数目是 m，因此算法 WaterLevel 的复杂度是 $O(m)$。在算法 WaterLevel 计算出变量 u 之后，所有 $p_j \geq 0, j \in (b, e]$，都可以通过公式 (4.24) 计算得到。使用算法

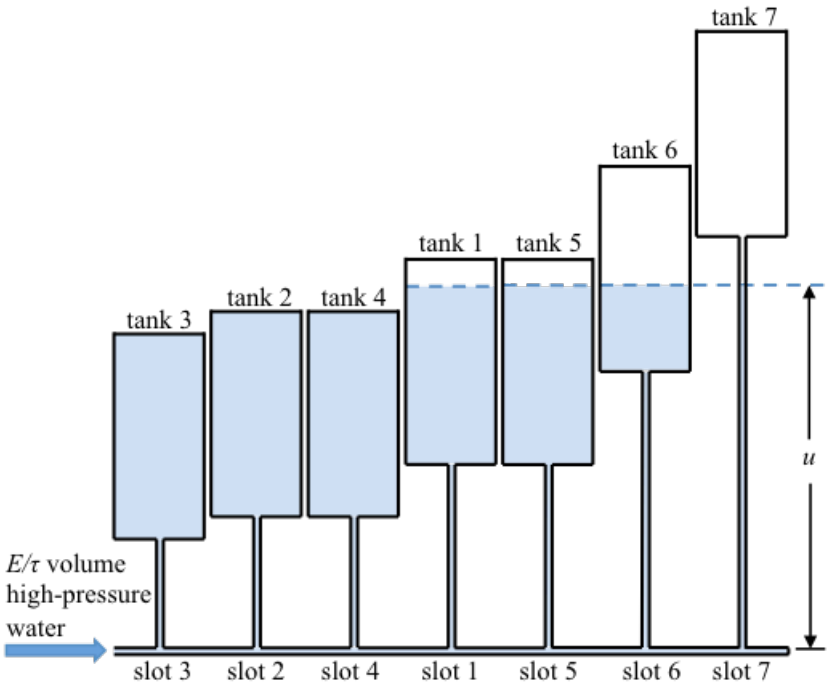


图 4.3 排序水箱技术的示例

Algorithm 9 WaterLevel(E, $[b, e]$, $floor$, $ceiling$)

if b=e **then**
  return
end if
$c = b, u = floor(b+1), f = b+1, remain_water = E/\tau$
let $floor(e+1) = \infty$ for loop purpose
while $c \leq e$ **do**
  $u_{new} = u + remain_water/(f-c)$
  if $u_{new} \leq ceiling(c+1)$ **and** $u_{new} \leq floor(f+1)$ **then**
    return u_{new}
  else
    $u_{new} = \min\{ceiling(c+1), floor(f+1)\}$
    $remain_water = remain_water - (u_{new} - u)(f-c)$
    $u = u_{new}$
    if $ceiling(c+1) \leq floor(f+1)$ **then**
      $c = c+1$
    else
      $f = f+1$
    end if
  end if
end while
return $ceiling(e)$

WaterLevel 作为子程序，算法 WaterTank 解决了 DCMM-1 问题。显然，算法 WaterTanks 在 $O(m)$ 的时间内执行完毕。

Algorithm 10 WaterTank($s_i, E, \{d_j\}, (b, e], p^{max}$)

1: sort tanks by their floor heights
2: let $floor(j) = d_j^{\alpha}, ceiling(j) = floor(j) + p^{max}$, for each $j \in (b, e]$
3: compute p_j by Eq.(4.24), for $j \in (b, e]$
4: **return** $\sum_{j=b+1}^{e} \frac{\tau}{2} \log(1 + \frac{p_j}{d_j^{\alpha}})$ and p_j

性质 4.1. 我们观察到距离 d_j 较小的水箱在最优解中具有较大的 p_j 值。这意味着当移动汇聚节点与传感器距离较近时进行数据传输更节能。

说明 4. 如果传感器 s 在利用能量 E 传输数据时，只允许使用其中 k 个时隙时，那么，根据性质4.1，我们必须选择与该传感器距离最近的 k 个“路段”对应的时隙。而且这 k 个时隙必须是连续的，用 $(x, y]$ 表示，其中 $(x, y] \subset (b, e]$。相对的，假设给定一组连续时隙 $(x, y] \subset (b, e]$，要求为传感器 s 计算最优功率控制策略最大化 $(x, y]$ 中传输的数据量，那么我们只需要调用算法 WaterTank 并使用 $(x, y]$ 代替 $(b, e]$ 作为输入。

4.3.4 一般化的 DCMM 问题的最优解

在本节中，我们考虑一般化的 DCMM 问题，与 DCMM-1 的区别是其具有 n 个传感器的。根据定义 4.3，算法不但要为每个传感器分配一组连续时隙，而且还要为每个时隙确定传输速率，以达到最大化采集数据量的目标。因为一个时隙最多只能分配给一个传感器，所以算法的目标是为每一个传感器 $s_i, 1 \leq i \leq n$ 确定一组连续时隙 $(\beta_i, \epsilon_i]$，使得 $0 \leq \beta_1 \leq \epsilon_1 \leq \beta_2 \leq \epsilon_2 \leq \ldots \leq \beta_n \leq \epsilon_n \leq m$，并且最大化数据采集量。一旦传输区间（即连续时隙）$\{(\beta_i, \epsilon_i]\}$ 确定，那么，根据说明 4，传输功率以及相应的传输速率可以很容易通过调度算法 WaterTank 计算确定。现在的问题是如何确定 $\{(\beta_i, \epsilon_i]\}$。

我们将每一个区间 $(\beta_i, \epsilon_i]$ 扩大，使得开始点与前一个区间的结束点 ϵ_{i-1} 相等。现在，区间 $(\beta_i, \epsilon_i]$ 扩大变为 $(\epsilon_{i-1}, \epsilon_i]$。由于 $(\epsilon_{i-1}, \epsilon_i]$ 可能会与 $A(s_i)$ 不一致；当不一致发生时，只有 $A(s_i)$ 中的时隙是可用的。因此，区间 $(x, y] = (\epsilon_{i-1}, \epsilon_i] \cap A(s_i)$ 被真正用来作为算法 WaterTank 的输入。根据说明 4，算法 WaterTank 可以为传感器 s_i 计算给定区间 $(x, y]$ 上的最优功率控制。之后，所有其他的时隙 $(\epsilon_{i-1}, \epsilon_i] - (x, y]$ 都被设置为传输功率 0。所以，DCMM 问题变为找 $n-1$ 个整数，$0 \leq \epsilon_1 \leq \epsilon_2 \leq \ldots \leq \epsilon_{n-1} \leq m$，使得时隙 $(0, \epsilon_1]$ 分配给传感器 s_1，时隙 $(\epsilon_1, \epsilon_2]$ 分配给传感器 s_2，...，时隙 $(\epsilon_{n-1}, m]$ 分配给传感器 s_n。显然，只要确定这 $n-1$ 个整数，我们便可以调用 WaterTank 计算传输功率。所以，关键的任务是确定最优的这 $n-1$ 个整数，$\epsilon_1, \epsilon_2, \ldots, \epsilon_{n-1}$。这些整数被称为“分离点”，因为它们将整个时间分割成 n 个互不相交的区间。

给定 n 个传感器 $s_i, 1 \leq i \leq n$，假设 s_i 具有可用能量 E_i 和可用时间区间 $(b_i, e_i]$。我们将它们按照 b_i 的值进行排序，使其满足 $b_1 \leq b_2 \leq \ldots \leq b_{n-1} \leq b_n$。对于任意两个传

感器 s_i 和 s_j，如果 $b_i = b_j$，那么，我们参考他们的结束时间 e_i 和 e_j。具有较早结束时间的被安排在前面。如果 $b_i = b_j$ 并且 $e_i = e_j$，那么任意顺序都可以。现在，我们用动态规划最优地确定这 $n-1$ 个“分离点”。

我们首先设计一个子问题：为前 i 个传感器最优地分配前 j 个时隙。令 $B(i,j)$ 表示移动汇聚节点 S 在时隙 {1, 2, ..., j} 中从传感器 $\{s_1, s_2, \ldots, s_i\}$ 上收集的最多的数据量。我们的最终目标是计算 $B(n,m)$。令时隙 $(k,j]$ 是在子问题 $B(i,j)$ 的最优解中分配给最后一个传感器 i 的时隙集合，那么我们有下面的关系 $B(i,j) = B(i-1,k) +$ WaterTanks$(E_i, \{d_{ih}\}, (k,j], p_i^{max})$，其中 $h \in (k,j]$。因为 $0 \le k \le j$，所以动态规划可以找到 k 使得 $B(i,j)$ 最大。下面的公式是动态规划的递推公式：

$$B(i,j) = \max_{0 \le k \le j} \{B(i-1,k) + \text{WaterTanks}(E_i, \{d_{ih}\}, (k,j], p_i^{max})\}, \text{ if } i > 0, j > 0, \quad (4.25)$$

$$B(i,j) = 0, \text{ if } i = 0 \text{ or } j = 0, \quad (4.26)$$

完整的动态规划的伪代码在算法 MultSensors 中给出。因为算法 WaterTank 的运行时间是 $O(m)$，所以算法 MultSensor 需要 $O(nm^3)$ 的时间。

Algorithm 11 MultipleSensors($\{s_i(E_i, p_i^{max}\}, \{d_{ij}\}, \{(b_i, e_i)\}$)

```
for i = 1 to n do
    B(i,0) = Epsilon(i,0) = 0
end for
for j = 1 to m do
    B(0,j) = 0
end for
for i = 1 to n do
    for j = 1 to m do
        for k = 0 to j do
            x(k) = B(i-1,k) + WaterTanks(E_i, {d_ih}, (k,j], p_i^max)
        end for
        B(i,j) = max(x);
        Epsilon(i,j) = arg max(x).
    end for
end for
ε_n = m
for i = n to 1 step -1 do
    ε_{i-1} = Epsilon(i, ε_i).
end for
return B(n,m), {ε_i}
```

基于算法 MultipleSensors，算法 WTS 可以在多项式时间内解决一般化的 DCMM 问题。其输入包括传感器集合 $\{s_i(E_i, p_i^{max}\}$，移动汇聚节点速度 γ，时隙长度 τ，和路径

总长度 L，其输出是最优时隙分配方案和功率控制策略以最大化数据采集量。显然，算法 WTS 在 $O(nm^3)$ 的时间内完成。

Algorithm 12 WTS($\{s_i(E_i, p_i^{max}\}, n, \gamma, \tau, L\}$)

1: $m = ceiling\frac{L}{\tau\gamma}$
2: **for** every sensor s_i **do**
3: compute its availabe time slot $(b_i, e_i]$
4: compute d_{ij} for every time slot $j \in (b_i, e_i]$
5: **end for**
6: MultSensors($\{s_i(E_i, p_i^{max}\}, \{d_{ij}\}, \{(b_i, e_i)\}$)
7: Notify the solution to all sensors $\{s_i\}$

4.4 动态联机分布式算法

在前面的小节中，我们设计了最优算法 WTS，其在多项式时间内解决静态离线的 DCMM 问题。然而，WTS 算法需要用到全局信息，包括所有传感器的位置、可用时隙、可用能量等。在实践中，尤其是在大规模网络中，获取全局信息的代价非常高。仅当移动汇聚节点进入到传感器的传输范围内的时候，该传感器的当前信息才能通知给汇聚节点。在本节中，我们提出一个具有高可扩展性的联机在线分布式算法 WOL 解决动态联机 DCMM 问题。

WOL 算法要求移动汇聚节点在周游的过程中实时跟踪在当前传输范围内的传感器信息。基于这些已知信息，使用静态离线最优算法计算当前最优的时隙分配方案和功率控制策略。具体地，在每一个时隙的开始，移动汇聚节点广播一个“请求”信息，所有收到该信息的传感器返回一个“回复”信息。显然，移动汇聚节点收到的每一个“回复”信息都代表一个传感器与之互在传输范围内。在该“回复”信息中，还包含了发送传感器的 ID，位置，以及电池电量等信息。一旦收到该“回复”信息，移动汇聚节点立即可以判断该传感器是否第一次回复。所有第一次回复的传感器构成了传感器集合 S_{new}。显然，一个传感器只会在 S_{new} 中出现一次，即在移动汇聚节点首次进入其传输范围时。而且集合 S_{new} 也可能是空的，表示移动汇聚节点在当前时隙未进入任何新的传感器的传输范围。在每一个时隙开始，子程序 WOL 都被调用一次。该子程序维护了在当前传输范围中的传感器结合 S，并且调用静态离线最优算法进行求解。

在 WOL 子程序的第 2-5 行中，对于 S_{new} 中的每一个 s_i，其可用时隙 $A(s_i)$ 和距离 d_{ij} 是根据位置计算得到。然后，集合 S 按照如下方法更新：a) 添加新的传感器（第 6 行）; b）删除不再在传输范围中的传感器（第 9-11 行）；c）对 S 中的每一个传感器，调整它的可用时隙，从当前时隙开始（第 8 行）。

在必要的更新之后，静态离线最优算法 MultSensors 被用来为 S 中的传感器计算新的时隙分配方案和功率调度策略。然后，这些调度结果会被通知到对应的传感器。当接收到这些调度信息之后，传感器存储并按照其进行调度，直到收到新的调度信息。

Algorithm 13 WOL(S_{new})

Let t be the current time
for every new sensor $s_i \in S_{new}$ **do**
  compute its abailabe slots $(b_i, e_i]$
  compute d_{ij} for every time slot in $(b_i, e_i]$
end for
$S = S \cup S_{new}$
for every sensor $s_i \in S_{new}$ **do**
  update available slots to $(t, e_i]$
  if $t \geq e_i$ **then**
    $S = S - s_i$
  end if
end for
return MultSensors($\{s_i(E_i, p_i^{max}\}, \{d_{ij}\}, \{(b_i, e_i)\}$)

移动汇聚节点和传感器的联机动态分布式框架在参考文献 [59] 中的第 6.1 节也有讨论，但是其使用了一个不同的静态离线算法。

4.5 算法性能评估

在本节中，通过与已有算法的比较，我们对提出的算法的有效性进行评估。

4.5.1 仿真设置

在我们的仿真中，我们考虑一个 10000m 长的直线道路。在不同的仿真实例中，部署 $40-350$ 不等的传感器在道路的两旁。任何传感器与道路的最大距离是 180m。在实验中，我们设置不同的汇聚节点移动速率，从 7m/s 到 22m/s 不等。任意两个相邻的“周游”之间，每一个传感器可以采集一个随机的能量，该能量大小满足均匀分布 $U(1, 3000)$mJ。在一个“周游”中，仅当移动汇聚节进入到 200m 的传输范围内的时候，传感器才可以将次数据发送给汇聚节点。在传输过程中，传输速率 r 与传输功率 p 之间满足关系公式 $r = \frac{1}{2}\log(1 + \frac{p}{d^2})$，其中 d 表示汇聚节点与传感器节点之间的距离。每一个传感器的最大传输距离 p^{max} 都是 330mW。时间被等量分解成时隙，时隙的长度 τ 在不同的设定下从 1s 到 7s 不等。我们使用“路段”的中点代表该路段计算移动汇聚节点与传感器节点的距离。

为了研究网络中的传感器个数 n，移动汇聚节点的速度 γ，以及时隙长度 τ 三者对算法性能的影响，我们令他们一次变化一个。我们使用 Ren, Liang 和 Xu [59] 提出的算法 RLX 作为基准，与我们的算法进行性能比照。对于给定的一种参数设定，我们随机产生 20 组实例，并运行 RLX,WTS 和 WOL 三个算法。

因为 RLX 算法使用距离预决定传输速率和传输功率，而不考虑两者在 AWGN 信道

中的关系公式，为了做公平有意义的比较，我们采用他们原始的参数设置，但是根据给定的传输距离和传输功率根据香农容量定理的关系公式修改传输速率。我们发现，在保持距离和传输功率不变的前提下，修改后的传输速率相比原始的传输速率稍有增加，意味着 RXL 的算法性能在修改后稍有提高。具体的，我们将实验参数设置罗列如下：

(1) 当距离是 0 到 20 米时，我们使用传输速率 0.2555Mbps，传输功率为 170mW，因为 $0.2555 = \frac{1}{2}\log(1+\frac{170}{20^2})$；

(2) 当距离是 20 到 50 米时，我们使用传输速率 0.0608，传输功率为 220mW，因为 $0.0608 = \frac{1}{2}\log(1+\frac{220}{50^2})$；

(3) 当距离是 60 到 120 米时，我们使用传输速率 0.0149bps，传输功率为 300mW，因为 $0.0149 = \frac{1}{2}\log(1+\frac{300}{120^2})$；

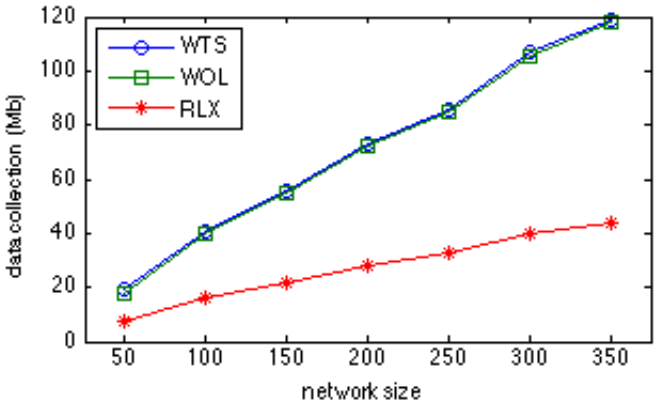
(4) 当距离是 120 到 200 米时，我们使用传输速率 0.0059Mbps，传输功率为 330mW，因为 $0.0059 = \frac{1}{2}\log(1+\frac{330}{200^2})$。

RLX 算法有两个版本，联机在线版本和静态离线版本。我们只与静态离线版本的算法进行比较，这是因为 1) 两个版本具有相似的算法性能[59]，2) 我们与较高性能的算法进行比较。

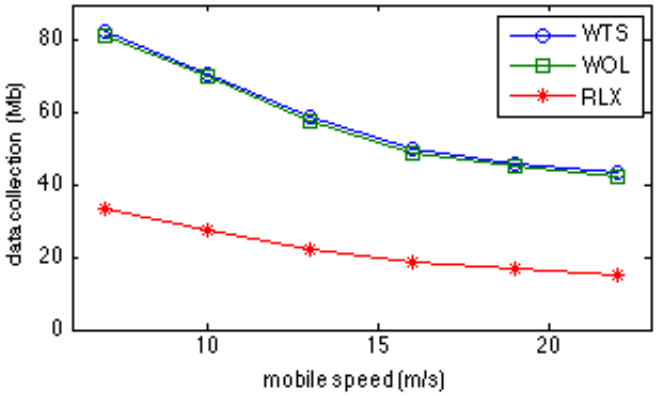
4.5.2 仿真结果

图4.4展示了 WTS,WOL 和 RLX 三个算法的性能比较，其中性能的衡量标准是一次“周游”中移动汇聚节点采集的数据总量。其中默认设置是：网络中传感器数目 200，汇聚节点的移动速度 10m/s，时隙长度 1s。图 4.4(a)(b)(c) 分别显示了算法性能受三个参数的影响：网络中传感器的个数 n，移动汇聚节点的速度 γ，以及时隙的长度 τ。图 4.4(a) 展示了网络中的传感器个数 n 以 50 为步长从 50 变化到 350 的仿真结果。我们观察到 WTS 算法的性能始终超越了 RLX 算法，大约超过 300% 之多。而且，两者之间的差距随着部署的传感器数量的增加而增加。图4.4(b) 显示了在以 3m/s 为步长，从 7m/s 到 22m/s 改变移动汇聚节点移动速度时的仿真结果。在不同的移动速度下，WTS 的算法能性能是 RLX 的 2-3 倍。同时，汇聚节点移动速度越慢，从传感器上采集到的数据量越大。从图4.4(c) 上，我们看出当我们以 1s 为步长，从 1s 到 7s 改变时隙长度时，WTS 算法性能总是大约 3-4 倍优于 RLX 算法。与图4.4(a) 和 (b) 相比较，我们可以看到时隙长度对算法性能的影响不及网络中传感器个数以及汇聚节点的移动速率来得大。作为总结，图4.4清楚的显示了 WTS 算法的性能从数据采集量上看显著优越于算法 RLX。这样结果并不意外，与我们的预期相符，因为 RLX 算法是 DCMM 问题的近似算法，它的近似度只有 $2+\epsilon$。此外，即使 RLX 算法能在第二步中计算出最优解，该最优解也局限于第一步中预设的传输功率。第一步中预设的传输功率鲜有最优的，从根本上导致 RLX 算法调度执行之后，传感器中还剩余相当的可用能量。相对地，WTS 算法可以利用所

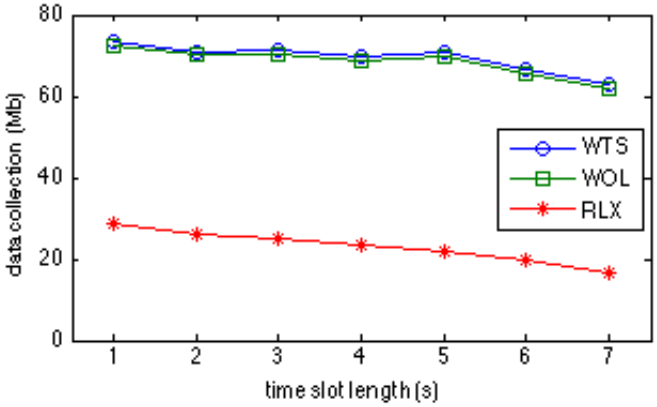
有可用能量以到达最大化数据传输量的目的。仿真实验也从侧面验证了 WTS 算法的最优性。



(a) 网络中传感器数目对结果的影响



(b) 汇聚节点移动速度对结果的影响



(c) 时隙长度对结果的影响

图 4.4 WTS,WOL 和 RLX 三个算法的性能比较

从仿真实验中，我们观察到一个有趣的现象：动态联机在线算法 WOL 性能与静态离线算法 WTS 的性能相当，这不难解释。移动汇聚节点只有进入到下一个新传感器的传输范围时，才得以能更新已知传感器的信息，并重新计算时隙分配和功率控制。因为

传感器的传输范围相比“路段”长度较大，所以导致已知传感器集合也较大。仅在前进方向远处的传感器才会对当前计算出来的调度产生影响，因为遥远前方未知传感器对当前调度的影响较小，所以我们的当前调度相对较优注，根据文献[59]中的仿真实验结果，RLX 算法的静态离线和联机动态版本的性能也十分接近。

所以，我们的最终结论是算法 WTS 和 WOL 的性能在各种实验设置下都远超算法 RLX。

4.6 本章小结

在本章中，我们研究了能量自供给无线传感器网络中数据采集最大化问题。具体地，我们为部署在道路两边的传感器网络设计时隙分配和功率控制算法，以最大化采集的数据量。我们设计了一个高效的多项式时间算法，称为 WTS，可计算该问题的最优解，虽然该问题之前曾经被证明是 NP 完全问题。该问题解决的关键是引入了新颖的水箱方法，并且联合使用动态规划方法。我们同时也设计了动态联机算法 WOL。根据仿真实验，在所有的实验设置下，我们提出的 WTS 算法产生的最优调度（以数据采集量为衡量标注）显著地优于已有算法。仿真实验结果显示，动态联机算法 WOL 的性能与静态离线算法 WTS 十分接近。

第五章　能量自供给无线设备传输速率调度工具原型系统

本章设计开发一套无线设备传输速率调度工具原型系统将前面三章有机的联系在一起。该系统实现了前三章中提出的算法并集成于一体。一方面，本系统通过统一的用户交互界面，根据用户的输入信息，自动判断问题类型，调用对应的算法；另一方面，本系统还提供底层函数调用接口，提供外部程序动态联机调用。本章主要包括：系统简介（目标、需求与组成），系统的设计与实现（总框架与各模块），系统的应用示例（界面展示）。

5.1 系统简介

能量自供给无线设备传输速率调度工具原型系统 (SEU-RateSchedule) 分别实现了第2章提出的密度最大区间优先 (DIF) 策略、第3章提出的截断 (Truncation) 方法以及第4章提出的水箱 (WaterTanks) 技术。

前三章提出的这三个速率调度算法的调度结果全都体现为在数据传输过程中对传输速率进行控制。SEU-RateSchedule 作为本文的原型系统，并不试图在射频模块物理硬件上开发硬件驱动程序，控制发送速率。而是编制了一套具有标准函数调用接口的库函数，供外部的硬件驱动程序调用，辅助硬件驱动程序决策传输速率。此外，本原型系统还开发了用户友好界面，直观展示率调度结果。为速率调度算法设计人员分析算法性能、检验算法正确性提供直观依据；也可以为无线设备传输能耗分析人员进行能耗瓶颈分析提供参考。

概况地讲，本章开发原型系统的总体目标包括如下。其一，编制了具有标准函数调用接口的库函数，方便将本文提出的算法部署到真实系统中。其二，将算法调度结果通过用户友好界面直观展示，方便算法设计人员和系统能耗分析人员参考。

5.1.1 系统需求

前三章提出的三个问题都存在静态离线调度问题和动态联机问题两个版本。静态离线问题是指在调度之前，就已经知道所有信息，比如每个数据报文的到达时间以及大小，能量采集的时间以及大小，所有传感器节点的位置以及电池电量。动态联机问题是指这些信息实时地通知给调度器。比如，只有数据报文到达后，调度器才知道其大小，只有能量采集后才知道其能量，只有进入其传输半径才知道传感器的位置和能量信息。前三章分别设计了相应的静态离线问题的最优解算法与动态联机问题的启发式算

法。SEU-RateSchedule 系统需要分别实现这三个问题的静态离散算法和动态联机算法，并且可以展示结果。

具体来讲，对于离线问题，SEU-RateSchedule 系统需要有一套用户界面，可以用来为三个问题独立地进行输入的设定，该输入可以是随机生成的，也可以是从预存的文件中读取的。在系统输入模块中，三个问题的输入既要相互区别，可以独立进行，又要相互联系，共享共同的输入组件。此外，该系统应该有统一的界面进行三个静态离线算法的调度运算和结果展示，便于研究人员查看调度结果。

对于动态联机问题，SEU-RateSchedule 系统不需要提供任何用户界面。这是因为，在动态联机问题中数据报文与能量采集都是实时产生的，而且输出结果也必须实时反馈给调用者，提供实时调度参考。因此，整个过程都是动态的，不需要用户界面。对于这类问题，SEU-RateSchedule 系统提供标准化的函数调度用，封装成函数库，支持标准函数调用，实时接收新的输入信息，调度运算，输出更新的速度调度信息，辅助硬件进行传输功率调整。

5.1.2 系统目标与组成

通过上一小节的系统需求分析，系统的目标与组成如下。

第一，设计简单易用的用户界面。输入部分的设计目标是三个问题的输入既要可以独立进行，又要尽量共享组件。调度运行与结果展示部分的设计目标是三个问题可以统一界面。

第二，提供标准化的函数调用并封装成函数库。所提供的标准函数调用的设计目标是可以实时接收新的输入信息，并且调度运输之后，输出更新的速率调度信息。

在本系统中，使用的封装库为“动态链接库”(Dynamic Link Library, DLL)。动态链接库是微软公司在微软视窗操作系统（Windows）中，实现共享函数库概念的一种方式。作为视窗操作系统的基础和核心，Windows API 就是通过动态链接库的形式提供的。也就是说视窗操作系统的运行，就是一个个动态链接库的函数调用。因此，本章编写并封装的动态链接库在视窗操作系统中具有非常高的通用性，可以被任何语言编写的程序，甚至是硬件驱动程序所调用。

5.2 系统设计与实现

根据上述的系统需求、目标与组成，对系统进行总体设计，主要包括：系统总框架、系统各模块总体流程图、以及各模块设计等。

5.2.1 系统总框架

SEU-RateSchedule 系统分为两个主要部分。其一称为用户界面，主要功能包括处理用户输入、判断调用模块类型、调用动态链接库中封装的标准函数、将调用结果进行绘图展示。其二称为动态链接库，封装了三类问题的静态离线最优解算法和动态联机启发

式算法，并对外提供标准函数调用接口。SEU-RateSchedule 系统的总体框架以及用户界面与动态链接库之间的关系如图5.1所示。

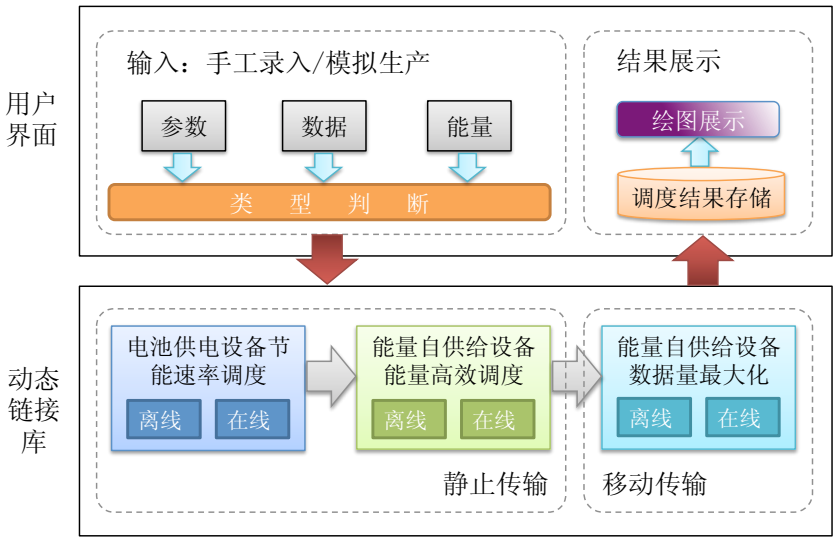


图 5.1　系统总体框架图

用户界面主要包括两个部分：输入与结果展示。输入又包括数据、能量以及参数，可以是手工录入，也可以是模拟生产。根据输入信息，用户界面需要进行判断，并调用动态链接库中对应的函数。动态链接库中函数调用的结果首先被存储到磁盘文件中，结果展示模块随后读取这些文件，并绘制可视化的调度结果，给用户展示直观简单的图形。

动态链接库对外提供的标准函数调用接口分为三种类型，分别对应前三章中介绍的算法，它们分别是密度最大区间优先策略、截断方法以及水箱技术。基于这三种静态离线最优解算法，前三章还提出了对应的动态联机在线算法，动态链接库予以一并实现。

动态链接库中在线算法的实现，理论上可以联机动态地为任何外部程序提供速率调度指导，比如硬件驱动程序、实时调度程序等，具有重要的意义。但是，在线算法的封装，仅仅是对前三章提出算法的实现，不涉及任何其他内容。因此，本章作为调度工具原型系统的介绍，主要介绍用户界面与动态链接库中静态离线算法之间的关系。

5.2.2 系统总体流程图

用户界面中的输入模块、结果展示模块、动态链接库中的密度最大区间优先策略、截断方法以及水箱技术之间的关系，组成了系统的总体流程图。该总体流程图如图5.1所示，其中静止电池调度模块对应密度最大区间优先策略，静止自供应调度模块对应截断方法，移动自供给调度模块对应水箱技术。

当用户进行输入设置之后，SEU-RateSchedule 系统首先根据输入信息判断待计算的场景适用哪一种算法。判定的依据是用户设定的场景中传输节点是否“可移动”、是否“能量自供给”。如果可移动且带自供给那么就使用水箱技术求解，称为移动自供给调度模块；如果不可移动且不带自供给那么就使用密度最大区间优先策略求解，称为静止电

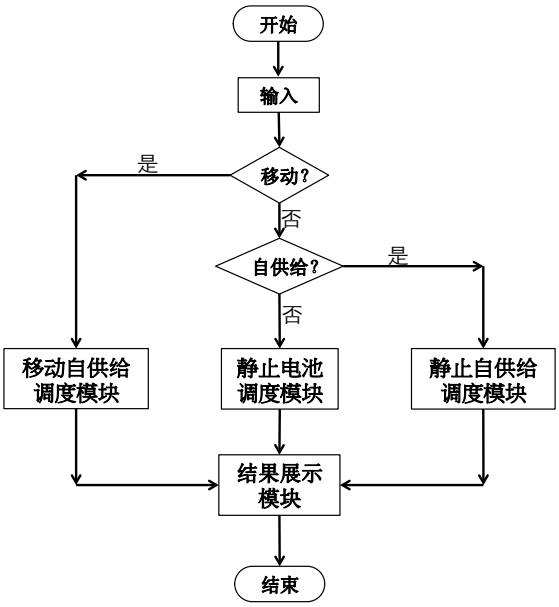


图 5.2 系统总体流程图

池调度模块；如果不可移动且而带自供给那么采用截断方法进行求解，称为静止自供给调度模块。确定用户设定的场景以及调用的算法之后，进行动态链接库函数调用，调度结果存储在文件中。SEU-RateSchedule 系统的结果展示模块对存储在文件的结果进行绘图和展示。在该过程中需要系统各个模块之间紧密互动。

从图5.1中可以看出，SEU-RateSchedule 系统由静止电池调度模块、静止自供给调度模块、移动自供给调度模块以及执行与结果展示模块等四个主要的模块组成。本节余下部分介绍这个四个模块的设计。

5.2.3 静止电池调度模块设计

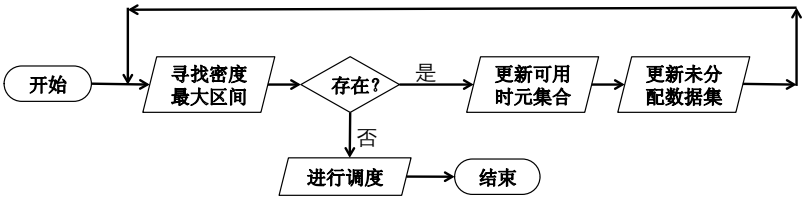


图 5.3 静止电池调度模块流程图

本模块设计实际上是密度最大区间优先 (DIF) 策略的实现。本模块的输入包括一组待传输的数据报文，输出是速率调度。对于输入的数据报文，首先提取所有到达时间与截止时间信息。任何一个到达时间与任意一个截止时间构成一个时间区间。时间区间包含的数据量除以区间长度成为密度。所以，本流程中第一步是寻找密度最大的区间，如果不存在这样的区间，或者说密度最大的区间的密度是零，则直接进行调度，并且算法终止。如果存在这样的区间，则区间密度就是该区间上的最优速率调度。接下来更新可用时元，将该区间的上时元标记为不可用；更新未分配数据报文集合，将该区间中包含

的数据报文标记为已分配。循环进行查找密度最大的区间。本模块的流程图如图5.3所示。

5.2.4 静止自供给调度模块设计

本模块实际上是截断 (Truncation) 方法的实现。本模块的输入包括一组待传输的数据报文和一组能量采集，输出是速率调度。由于数据报文与能量采集都是在时间轴上的事件，因此，本设计采用按照时间推进的方法进行调度。假设时间轴上发生最晚的时间为 T。如果当前时间大于等于 T，那么算法结束。否则的话，计算最小的 k 使得 $(k+1)$-ZM 调度 $r^{rm(k+1)}$ 不能被能量采集支持，即采集的能量不足够按照 $r^{rm(k+1)}$ 传输。如果找不到这样的 k，表明采集的能量足够按照 r^{rm} 将所有数据报文传输完。否则的话，采用“截断”方法将速率调度 $r^{rm(k)}$ 与速率调度 $r^{rm(k+1)}$ 之间的部分进行截断，并按照截断后的速率进行调度。最后更新当前时间，循环判断当前时间是否大于等于 T。本模块的流程图如图5.4所示。

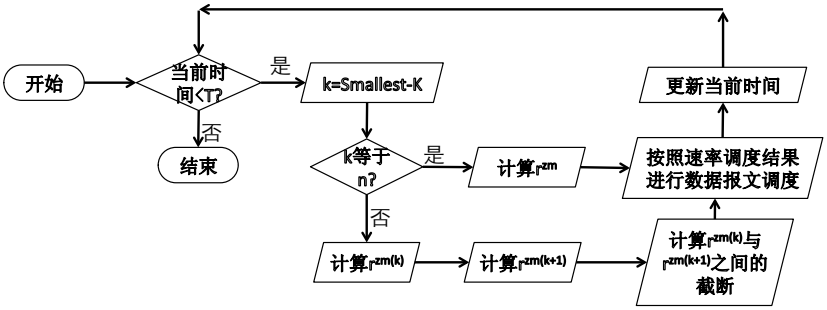


图 5.4 静止自供给调度模块流程图

5.2.5 移动自供给调度模块设计

本模块实际上是水箱 (WaterTanks) 技术的实现。本模块的输入包括各个传感器节点的位置信息、电池电量信息，移动汇聚节点的移动速率、方向、传输半径，以及系统时隙长度等信息。输出包括时隙调度和功率控制。目标是最大化移动汇聚节点的数据采集量。令 $B(i,j)$ 表示移动汇聚节点 S 在时隙 $\{1,2,\ldots,j\}$ 中从传感器 $\{s_1,s_2,\ldots,s_i\}$ 上收集的最多的数据量。通过动态规划的递推公式 (4.25) 与 (4.26)，很容易能够计算出最终目标 $B(n,m)$。所以，本设计的目标是算法 WaterTanks。该算法的输入是一系列水箱以及一定量的水，目标是计算将水注入水箱后的自由水面的高度。图5.5给出了该算法的流程。首先假设所有水箱一开始是空的，然后将水缓慢注入水箱中。该算法将注水的过程分为许多步骤，在每一步骤中计算水位上升量以及剩余水量。在下一步注水之前检查水箱是否已经全满，如果是，那么最高水箱顶便是要计算的值。如果水箱未全满，那么判断剩余水量是否足够使水面上升到最矮的水箱顶或者最矮的 (新) 水箱低。如果足够，那么就记录新的水位，并计算剩余水量；如果不足够，那么就直接计算新水位，并返回结果。

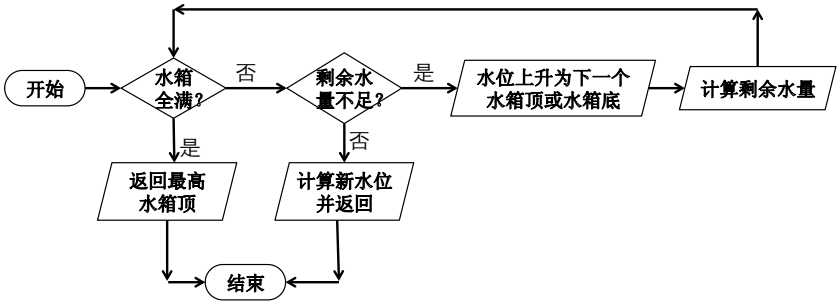


图 5.5 水箱 (WaterTanks) 算法流程图

5.2.6 执行与结果展示模块设计

由于静止电池调度模块、静止自供给调度模块以及移动自供给调度模块分别实现了密度最大区间优先 (DIF) 策略、截断 (Truncation) 方法以及水箱 (WaterTanks) 技术，因此，模块运行时间是三种算法的运行时间中最长的那个。而随着输入规模的增加，模块运行时间随之不断增加。具体表现在算法函数的返回时间增加。由于这些函数调用属于阻塞式调用，如果不加处理的话，在函数调用期间（从函数调用开始到到函数返回结束）主调用进程，即用户界面，会处于“僵死”状态，严重影响用户体验。因此，本模块设计采用多线程协助的方式，实现了非阻塞式函数调用。

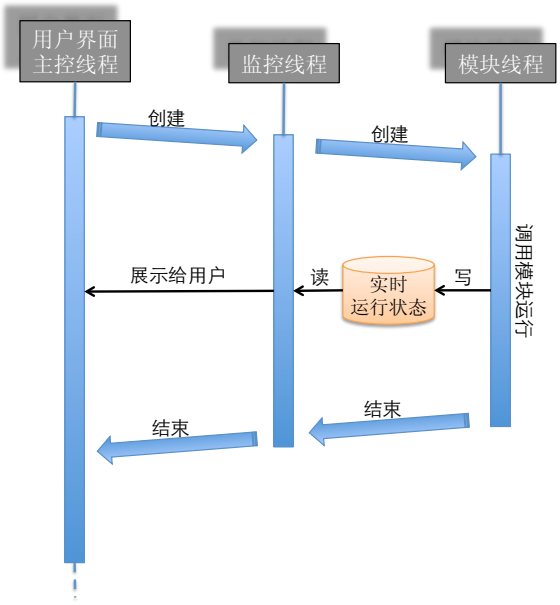


图 5.6 非阻塞模块调用多线程交互图

用户界面线程为主控线程，自始至终都处于活动状态。当需要调用某调度模块时，主控线程首先创建一个监控线程，监控线程再创建一个线程调用调度模块，该线程称为模块线程。这样，调度模块调用阻塞在模块线程上，不影响其他主控线程和监控线程。在调度模块运行的过程中产生实时运行状态信息，被写入本地文件中。监控线程再将该实时运行状态信息读取，并交给用户界面主控线程在用户界面上进行显示，给用户实时

信息。调度模块将调度结果存储在本地文中。模块线程在调度模块调用完成后结束，而监控线程在模块线程结束后结束。

调度模块将调度结果存储在本地文件中，结果展示模块读取这些调度结果，进行绘图，并展示给用户。

5.3 系统应用示例

SEU-RateSchedule 系统以 Microsoft Visual Studio 2010 集成开发环境作为开发工具，使用到.NET Framework 4.1 以及 Matlab R2015a。具体的，用户界面使用 Microsoft Visual Basic 2010 编程实现，动态链接库是用 Microsoft Visual C++ 2010 开发而得，结果展示部分使用 Matlab R2015a 进行编程绘图。

5.3.1 系统组成与逻辑关系

SEU-RateSchedule 系统的组成相对简单：主要包括一个应用程序文件 (exe)、三个动态链接库文件 (dll) 以及三个 Matlab 程序文件 (m)，其它还包括若干中间结果存储文件 (txt)。这些文件之间的逻辑关系如图5.7所示。

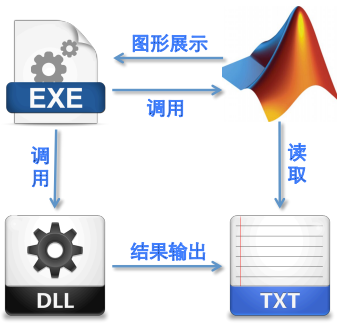


图 5.7 系统文件逻辑关系图

5.3.2 系统主界面

SEU-RateSchedule 系统的主界面如图5.8所示，包括模式选择栏和工作区两个部分。左边的模式选择栏中共七个按钮，分别是开始界面、静止场景、移动场景、参数配置、调度运算、结果展示、系统退出。右边的工作区在系统开始时是系统的欢迎界面，也称作开始界面。在不同的功能模式下，工作区会根据用户输入、参数配置、结果显示的需要进行控件的布置。

5.3.3 静止场景构建

在静止场景模式下，工作区包括数据报文的输入与编辑、能量采集的输入与编辑两个主要部分。其中，用户可以根据需要启用或者禁止能量采集的输入与编辑，该操作可

图 5.8 系统主界面

以通过点击能量采集标题出的复选框实现。该复选框选中表示该静止场景允许能量自供给，否则不允许。不论是数据报文还是能量采集，都可以逐条进行添加或者删除，也可以批量从文件导入，还可以随机生成。界面如图5.9所示。

图5.10是数据报文和能量采集随机生成的参数设置界面。数据报文可设置的参数包括：总数、平均大小、平均延时、平均间隔，能量采集可设置的参数包括：个数、平均大小、平均间隔。用户可以自行输入，也可以使用默认参数。根据用户在图5.9中是否选择了能量自供给复选框，本窗口中能量自供给部分的参数会自动启用或者禁止。

5.3.4 移动场景构建

在移动场景构建模式下，工作区可以输入与编辑无线传感器信息。这些信息具体包括：传感器编号、横坐标、纵坐标、电池电量、最高速率。用户可以逐条添加或者删除，也可以从文件批量导入，还可以随机生成。界面如图5.11所示。

图5.10是传感器信息随机生成参数设置界面。可以设置的参数包括：传感器数量、电池总容量、最高传输速率。用户可以自行输入，也可以使用默认参数。

可以看到，无论是静止场景构建界面 (图5.9) 还是移动场景构建界面 (图5.11) 都存在一个“下一步”按钮，点击该按钮可以将用户带入到参数配置界面。

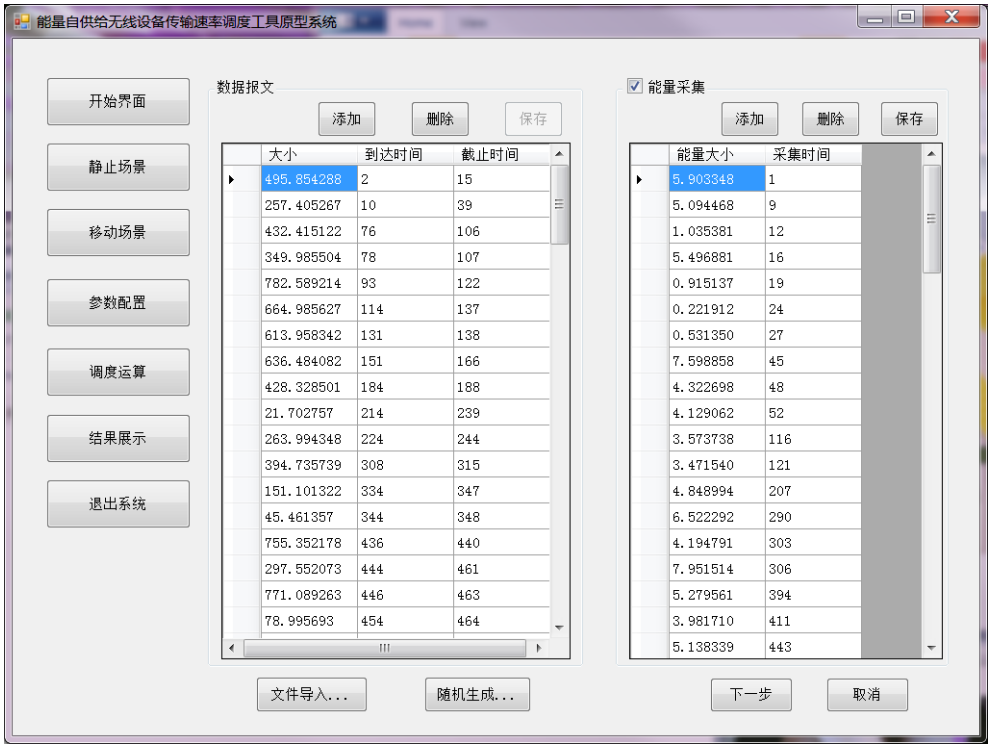


图 5.9 静止场景构建界面

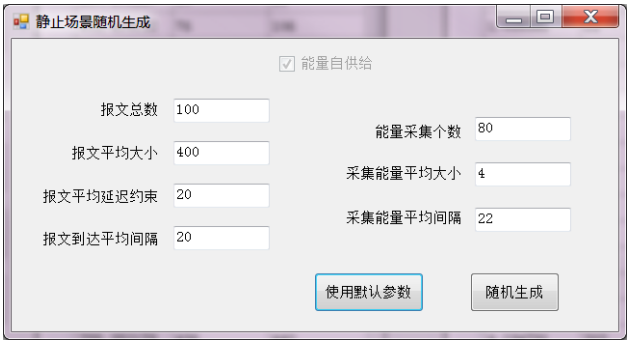


图 5.10 静止场景构建：随机生成界面

5.3.5 参数配置

参数配置界面中的工作区分为静止场景参数配置界面和移动场景参数配置界面两种，系统会根据用户之前的输入自动选择展示。静止场景参数配置界面如图5.13所示。可以配置的参数包括：收发节点之间的距离、温度、信道类型或者自定义带宽、频率、硬件最高速率三个参数。

移动场景参数配置界面如图5.14所示。可以配置的参数包括：汇聚节点的移动速度、时隙长度、最大传输距离、路径总长度。

通过点击“下一步”按钮进入到调度运算界面。

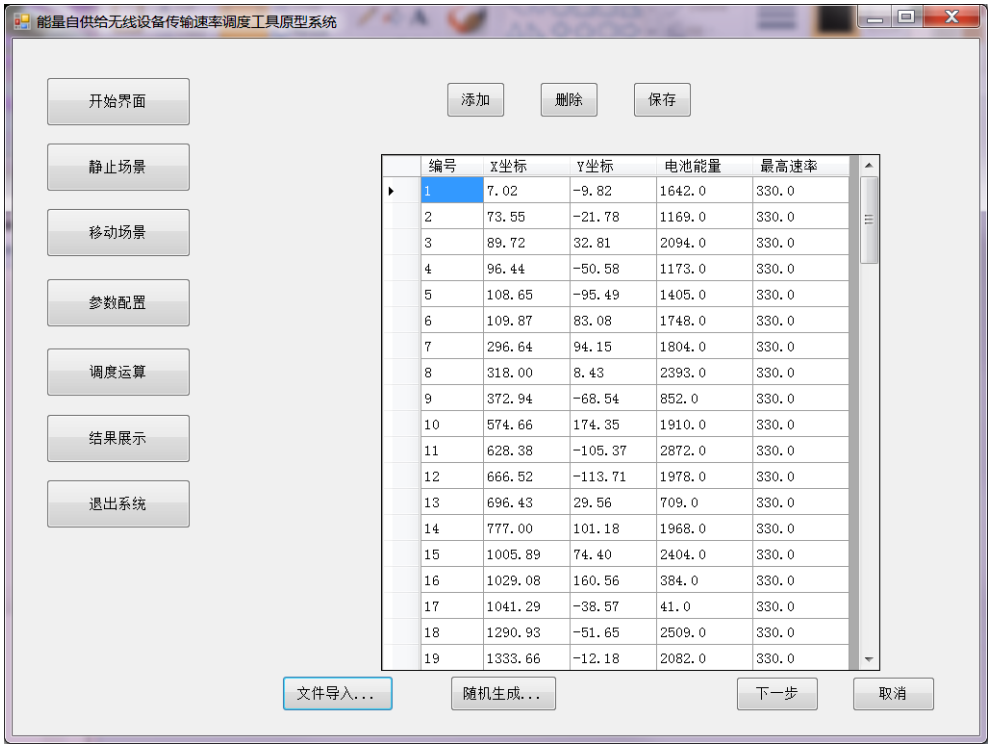


图 5.11 移动场景构建界面

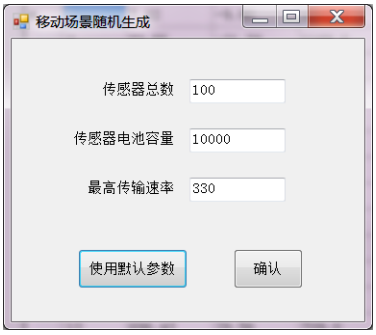


图 5.12 移动场景构建：随机生成界面

5.3.6 调度运算

调度运算是通过调用动态链接库中的封装的函数实现的。调度包括计算静态离线问题的最优解和动态在线问题的启发式算法。根据用户在之前步骤中的输入，系统自动判断调用静止电池调度模块、静止自供给调度模块还是移动自供给调度模块。调度运算界面实时展示调度运算的状态，如图5.15所示。

调度运算结束之后，可以点击“下一步”进入到结果展示界面。

5.3.7 结果展示

在结果展示模式下，工作区有两种状态，静态调度结果展示与移动调度结果展示。静止调度待展示的结果是速率调度。本系统既计算了静态离线的最优速率调度，也计算

图 5.13 静止场景参数配置界面

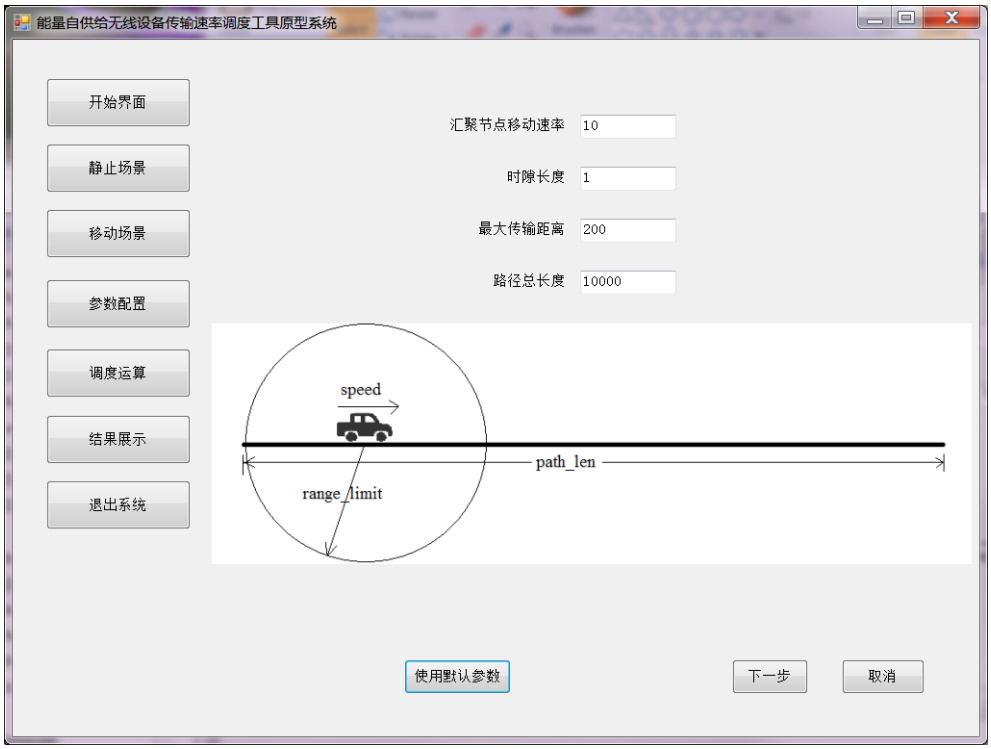


图 5.14 移动场景参数配置界面

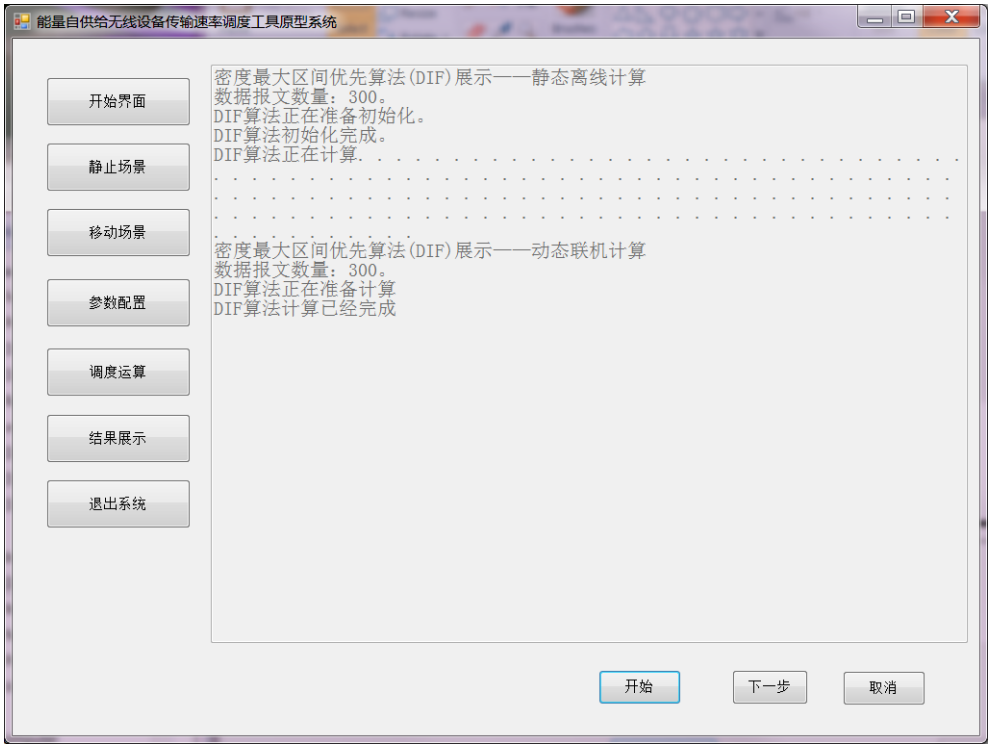


图 5.15 系统调度运算界面

了动态联机的速率调度。因此在本模式下，工作区有一个选项卡控件，其中包含三个选项页，分别展示了静态离线速率调度结果、动态联机调度结果以及两者混合比较。

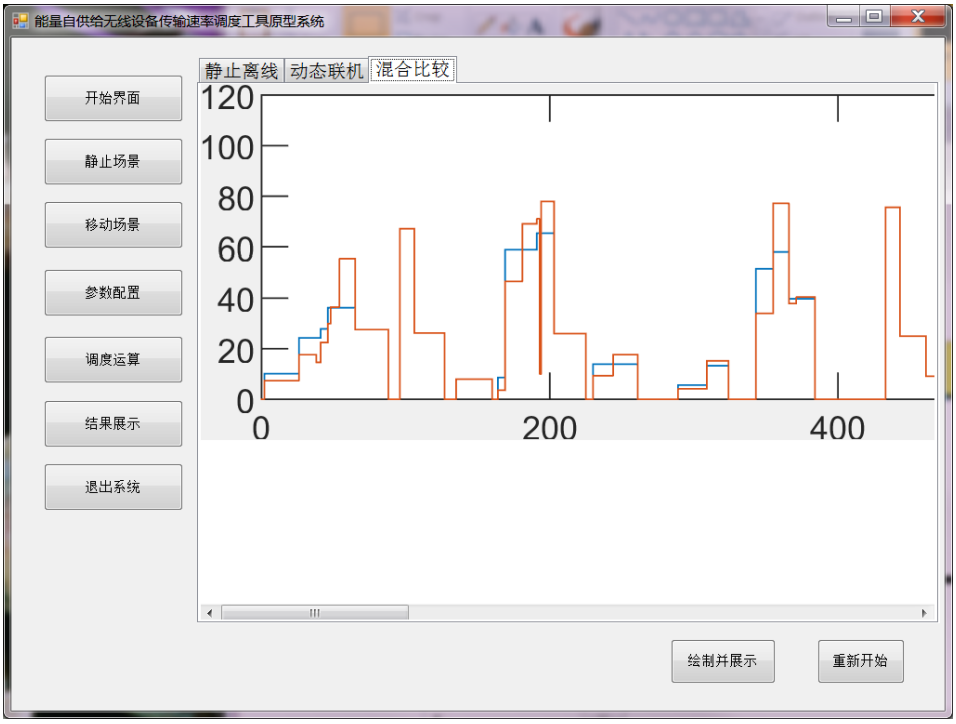


图 5.16 静止场景调度结果展示界面

移动调度待展示的结果既有时隙分配，也有功率控制。同样，工作区有一个选项卡

控件，总共有两个选项也，分别展示了静态离线速率调度结果、动态联机调度结果。

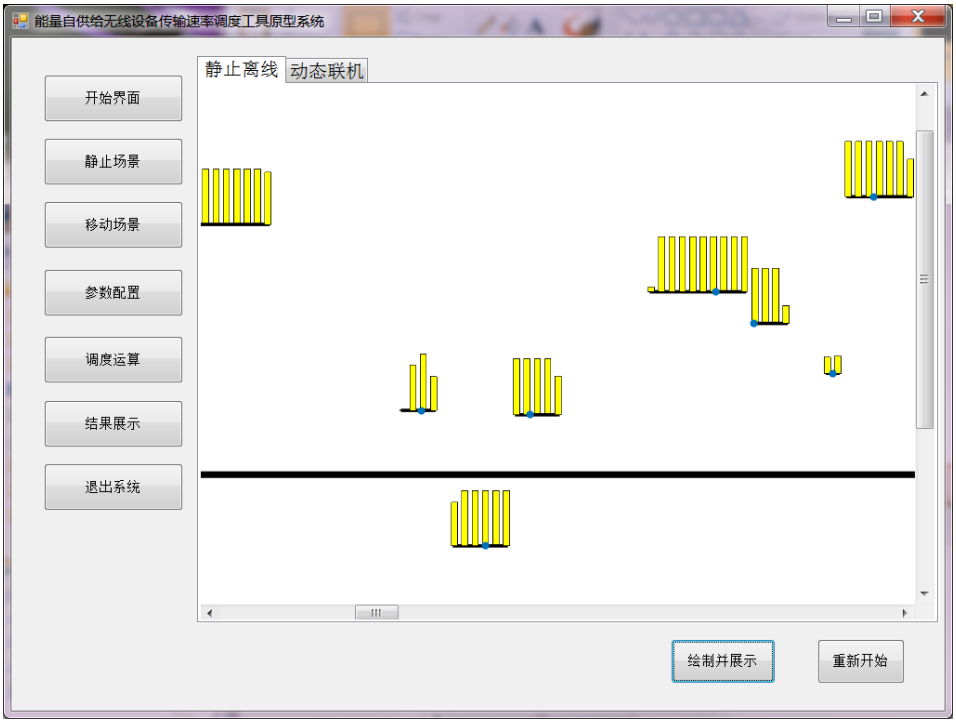


图 5.17 移动场景调度结果展示界面

5.4 本章小结

本章在本文理论研究的基础上，设计并开发了一套能量自供给无线设备传输速率调度工具原型系统，称为 SEU-RateSchedule 系统。该系统包括用户界面输入模块、静止电池调度模块、静止自动机调度模块、移动自供给调度模块以及执行与结果展示模块等主要模块。SEU-RateSchedule 系统除了提供简单、易用的用户界面外，还提供封装好的标准程序调用动态链接库，供任外部函数调度，极大地拓展了本系统的应用范围。本章分别介绍了 SEU-RateSchedule 系统的总架构图、总流程图、各模块的设计、系统组成与逻辑关系等内容。最后，给出系统的一个应用实例，对系统的各项功能进行测试，验证了系统的可用性。

第六章　总结与展望

作为本博士论文正文的最后一部分，本章对论文工作和取得的相关成果进行简单的总结，并展望后续研究工作的一些新的方向。

6.1 论文总结

本论文所取得的研究成果可以概括为以下四个部分，其中前三部分是理论研究，为本博士论文的主体，最后一部分为原型系统实现工作，用以展示并验证本博士论文的相关理论工作。

6.1.1 静止传输时电池供电设备节能速率调度（第2章）

第2章提出最优算法解决了电池供电设备节能速率调度问题，该问题目标是将一组允许任意截止时间的数据报文使用最小能耗发送完毕。本章提出基于数据区间的新技术，称为密度最大区间优先策略。该策略击中问题的核心难点，所以得以将问题解决。本章还证明最早截止时间优先调度策略调度数据报文，可以并保证数据报文在其截止时间之前传输完毕。最后，本章提出一个在线策略“密度指导冷却”策略。仿真实验显示，该在线策略通过优化预测可将调度结果提高到最优解的 110% 范围内。

6.1.2 静止传输时能量自供给设备能量高效调度（第3章）

第3章研究了能量自供给设备能量高效调度问题，该问题可以描述为以最小的能耗将一组允许传输延时约束的数据报文从能量自供给设备上传输的到指定接收设备上，约束是传输速率只能取离散数值。基于传输延时与传输能耗之间的基础折衷，本章提出“截断”方法计算该问题的最优解。然后，本章提出一个通用框架，可以将任意基于连续速率模型的调度算法（包括本文的“截断”方法）转变为基于离散速率模型的调度算法。本章还提出一个能量高效的在线算法为动态联机问题进行速率调度。仿真实验结果显示，该在线算法的平均性能与最优算法存在近似关系。

6.1.3 移动传输中能量自供给设备数据量最大化（第4章）

第4章研究了能量自供给无线传感器网络中的数据采集最大化问题。具体地，本章考虑将时隙分配给部署在移动路径旁边传感器，并控制每一个传感器的传输功率，使用

有限的能量最大化传输数据量。本章提出的高效多项式算法，称为“水箱”技术，可以直接计算该问题的最优解，而该问题曾经被证明是 NP 完全问题。“水箱”技术将原本两步骤的方法改为单步骤，该单步骤方法中使用到动态规划技术。本章还设计了在线算法，解决动态联机问题。仿真结果表明本章设计的最优解算法性能超过了前人提出的算法；本章提出的在线算法性能几乎与最优算法一致。

6.1.4 传输功率调度工具原型系统（第5章）

第5章设计了能量自供给无线设备传输速率调度工具原型系统，称为 SEU-RateSchedule 系统。该系统将第2章提出的密度最大区间优先 (DIF) 策略、第3章提出的截断 (Truncation) 方法以及第4章提出的水箱 (WaterTanks) 技术融为一体，建立统一的标准程序调用接口以及用户界面，计算静态离线速率调度和动态联机速率调度。本章首先对 SEU-RateSchedule 系统做了简要介绍，包括系统目标、系统需求、系统组成。接着，介绍了系统设计与实现，包括系统总框架、系统总流程图、各个模块的设计等。最后给出了系统部属与应用示例，主要是系统运行界面截图。

6.2 下一步研究的方向

论文的研究工作涉及到能量自供给技术与无线传输速率调度技术，取得了一定的成果，但仍然存在许多不足之处，部分内容有待继续研究，具体如下：

本博士论文第2章提出密度最大区间优先 (DIF) 策略解决静止传输时电池供电无线设备的节能调度问题。密度最大区间优先策略解决了开放了十多年之久的难题，它在解决思路上的突破，方法论上的创新，开启一个全新领域，开辟了解决其它相似速率调度问题的新渠道。在未来，下一步研究方向包括：将该方法推广到 1) 衰弱信道上，即当无线通信信道非稳定，其信道增益是时间的函数，该场景较本文考虑的稳定信道场景更为常见和通用；2) 能量自供给设备的无线数据传输，即当无线传输设备的能量供给主要从环境中获取，该场景与第3章相似，但是允许任意数据报文截止时间；3) 多信道系统中，即当无线数据发送节点向多个接收设备同时传输数据报文，该场景是实际系统的一种，较为常见，其研究具有重要意义。

本博士论文第3章研究了能量自供给设备能量高效调度问题。下一步工作计划在三个方向展开，分别是：设计新的算法，降低时间复杂度；为无线数据传输节点设计具有近似度的在线算法；将 Truncation 算法推广到云计算研究中数据中心的任务调度。1）算法 Truncation 计算静态离线最优解，时间复杂度是 $O(mn(n^2+m))$，其中 n 是数据报文，m 是能量采集的个数。该算法的时间复杂度尚较高，下一步计划设计更高效算法解决这一问题。2）算法 Truncation 解决静态离线问题，该问题虽然具有重要的意义，但是实际问题往往是动态联机的在线问题，下一步计划设计一个具有近似度的在线算法。3）无线数据传输的能耗模型与处理器的任务处理能耗模型一致，所以，下一步计划将本章提出的 Truncation 方法推广到数据中心任务处理问题上。

本博士论文第4章研究了能量自供给无线传感器网络中的数据采集最大化问题。在本章的讨论中，传感器节点的假设可以增强；无线传输模型的假定没有考虑传输路径上的障碍物；传感器网络被默认是密集部署，没有考虑稀疏部署的情况。首先，本章可以为每一个传感器增加两个约束条件，一个是数据量存储容量限制 D，另一个是可用数据量限制 $(B < D)$。本章只需要在为 DCMM-1 问题定义的凸规划中增加一两个不等式约束条件即可。water-tank 技术只需要做一些简单修改就可以解决这个新问题。其次，在现实中，像树木和碎石之类的障碍物可能会阻碍无线信号的传播，减弱信号，以至于基于距离的信号递减模型不再成立。然而，这样的“路径衰减”参数 $\{h_j\}$ 是可以通过现场测量获得的，因此，引入该参数后该问题仍然可以解决。通过将 d_j^α 替换成 $1/h_j$，水箱法仍然可以用来直接计算最大数据采集量。最后，在稀疏部署的传感器网络中，每一个传感器都有机会传输自己采集的数据是非常重要的。公式 (4.25) 与本章所提算法可以修改来满足这样的要求，并且得到最优解。

参考文献

[1] IDC's Worldwide Smart Connected Device Tracker Forecast Data. February 28, 2013

[2] U.S. Wireless Smartphone Customer Satisfaction Study. J.D. Power and Associates, 2014.

[3] 中国互联网信息中心. 第 35 次中国互联网络发展状况统计报告. 2015 年 2 月.

[4] J. Yick, B. Mukherjee, and D. Ghosal. "Wireless sensor network survey." Computer networks, vol. 52, no. 12 (2008): 2292-2330.

[5] R. Murmuria, J. Medsger, A. Stavrou, and J. M. Voas. "Mobile application and device power usage measurements." Proc. IEEE SERE 2012, pp. 147-156.

[6] L. Ardito, G. Procaccianti, M. Torchiano, and G. Migliore. "Profiling Power Consumption on Mobile Devices." Proc.IARIA ENERGY 2013, pp. 101-106.

[7] A. Carroll, and G. Heiser. "An Analysis of Power Consumption in a Smartphone." Proc. USENIX 2010, pp. 1-14.

[8] IDC's Worldwide Smart Connected Device Tracker Forecast Data. February 28, 2013

[9] 国家工业和信息化部. 通信业"十二五"发展规划.2012 年 5 月.

[10] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. R. Smith. "Ambient backscatter: wireless communication out of thin air." Proc. ACM SIGCOMM 2013, pp. 39-50.

[11] V. Talla, B. Kellogg, B. Ransford, S. Naderiparizi, S. Gollakota, J. Smith. "Powering the Next Billion Devices with Wi-Fi." arXiv:1505.06815, 2015.

[12] A. Pang, and R. Bannatyne. "The Energy Harvesting Tipping Point for Wireless Sensor Applications." Instrumentation, Measurement, Circuits and Systems, vol. 127 (2012): 387-391.

[13] B. Prabhakar, E. Uysal Biyikoglu, and A. El Gamal. "Energy-Efficient Transmission Over a Wireless Link via Lazy packet Scheduling." Proc. IEEE INFOCOM 2001, pp. 386-394.

[14] E. Uysal-Biyikoglu, B. Prabhakar, and A. El Gamal. "Energy-Efficient Packet Transmission Over a Wireless Link." IEEE/ACM Trans. on Netw., vol. 10, no. 4 (2002): 478-499.

[15] E. Uysal-Biyikoglu, A. El Gamal. "On adaptive transmission for energy efficiency in wireless data networks." IEEE Trans. Inf. Theory, vol. 50, no. 12 (2004): 3081-3094.

[16] W. Chen and U. Mitra. "Energy efficient scheduling with individual packet delay constraints." Proc. IEEE INFOCOM 2006.

[17] W. Chen, M. J. Neely, and U. Mitra. "Energy-Efficient Transmissions With Individual Packet Delay Constraints." IEEE Trans. Inf. Theory, vol. 54, no. 5 (2008): 2090-2109.

[18] M.A. Khojastepour, A. Sabharwal. "Delay-constrained scheduling: power efficiency, filter design, and bounds." Proc. IEEE INFOCOM 2004, pp. 1938-1949.

[19] W. Chen, M.J. Neely, U. Mitra. "Energy efficient scheduling with individual packet delay constraints: offline and online results." Proc. IEEE INFOCOM 2007, pp. 1136-1144.

[20] W. Chen, U. Mitra, M. Neely. “Energy-efficient scheduling with individual delay constraints over a fading channel.” Proc. WiOpt 2007.

[21] W. Chen, U. Mitra, M. Neely. “Energy-efficient scheduling with individual packet delay constraints over a fading channel.” Wireless Netw. vol. 15, no. 5 (2009): 601-618.

[22] M. Zafer and E. Modiano. “A Calculus approach to minimum energy transmission policies with quality of service guarantees.” Proc. IEEE INFOCOM 2005, pp. 548-559.

[23] M. Zafer and E. Modiano. “A Calculus Approach to Energy-Efficient Data Transmission With Quality-of-Service Constraints.” IEEE/ACM Trans. on Netw., vol. 17, no. 3 (2009): 898-911.

[24] M. Zafer, E. Modiano. “Optimal rate control for delay-constrained data transmission over a wireless channel.” IEEE Trans. Inf. Theory, vol. 54 no. 9 (2008): 4020-4039.

[25] J. Yang, S. Ulukus. “Optimal packet scheduling in a multiple access channel with rechargeable nodes.” Proc. IEEE ICC 2011.

[26] K. Tutuncuoglu, and A. Yener. “Sum-rate optimal power policies for energy harvesting transmitters in an interference channel.” J. Commun. Netw., vol. 14, no. 2 (2012): 151-161.

[27] M. Gatzianas, L. Georgiadis, and L. Tassiulas. “Control of wireless networks with rechargeable batteries.” IEEE Trans. on Wireless Comm., 9, no. 2 (2010): 581-593.

[28] R. Vaze. “Competitive ratio analysis of online algorithms to minimize packet transmission time in energy harvesting communication system.” Proc. IEEE INFOCOM 2013, pp. 115-1123.

[29] R. Vaze, R. Garg, and N. Pathak. “Dynamic power allocation for maximizing throughput in energy-harvesting communication system.” IEEE/ACM Trans. on Netw., vol. 22, no. 5 (2014): 1621-1630.

[30] C. Huang, R. Zhang, and S. Cui. “Throughput Maximization for the Gaussian Relay Channel with Energy Harvesting Constraints.” IEEE Journal on Sel. Areas in Comm., vol. 31, no. 8 (2013): 1469-1479.

[31] B. Gurakan, O. Ozel, J. Yang, and S. Ulukus. “Energy cooperation in energy harvesting two-way communications.” Proc. IEEE ICC 2013.

[32] Qualcomm Inc. “An overview of the application of code division multiple access (CDMA) to digital cellular systems and personal cellular networks.” Doc. EX60-10010, 1992.

[33] IEEE Wireless Local Area Networks, IEEE Standard 802.11, 2012.

[34] E. Altman, K. Avrachenkov, G. Miller, and B. Prabhu. “Discrete power control: cooperative and non-cooperative optimization.” Proc. IEEE INFOCOM 2007, pp. 37-45.

[35] Y. Xing, and R. Chandramouli. “Stochastic learning solution for distributed discrete power control game in wireless data networks.” IEEE/ACM Trans. on Netw., vol. 16, no. 4 (2008): 932-944.

[36] T. Kim, H. Lim, and J. C. Hou. “Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks.” Proc. ACM MobiCom, 2006.

[37] T. Kim, H. Lim, and J. C. Hou. “Understanding and improving the spatial reuse in multihop wireless networks.” IEEE Trans. on Mobile Computing, vol. 7, no. 10 (2008): 1200-1212.

[38] M. Ma, and Y. Yang. “SenCar: an energy-efficient data gathering mechanism for large-scale multihop sensor networks.” IEEE Trans. on Parallel and Dist. Sys., vol. 18, no. 10 (2007): 1476-1488.

[39] W. Wang, V. Srinivasan, and K.-C. Chua. “Extending the lifetime of wireless sensor networks through mobile relays.” IEEE/ACM Trans. on Netw. vol. 16, no. 5 (2008): 1108-1120.

[40] G. Xing, M. Li, T. Wang, W. Jia, and J. Huang. “Efficient rendezvous algorithms for mobility-enabled wireless sensor networks.” IEEE Trans. on Mobile Comp., vol. 11, no. 1 (2012): 47-60.

[41] S. Gao, H. Zhang, and S. K. Das. “Efficient data collection in wireless sensor networks with path-constrained mobile sinks.” IEEE Trans. on Mobile Comp., vol. 10, no. 4 (2011): 592-608.

[42] M. Zhao, and Y. Yang. “Optimization-based distributed algorithms for mobile data gathering in wireless sensor networks.” IEEE Trans. on Mobile Comp., vol. 11, no. 10 (2012): 1464-1477.

[43] Y. Shi, and Y. T. Hou. “Theoretical results on base station movement problem for sensor network.” Proc. IEEE INFOCOM 2008.

[44] Y. Shi, and Y. Thomas Hou. “Some fundamental results on base station movement problem for wireless sensor networks.” IEEE/ACM Trans. on Netw., vol. 20, no. 4 (2012): 1054-1067.

[45] Y. Yun, and Y. Xia. “Maximizing the lifetime of wireless sensor networks with mobile sink in delay-tolerant applications.” IEEE Trans. on Mobile Comp., vol. 9, no. 9 (2010): 1308-1318.

[46] Y. Yun, Y. Xia, B. Behdani, and J. C. Smith. “Distributed algorithm for lifetime maximization in a delay-tolerant wireless sensor network with a mobile sink.” IEEE Trans. on Mobile Comp., vol. 12, no. 10 (2013): 1920-1930.

[47] A. Chakrabarti, A. Sabharwal, and B. Aazhang. “Communication power optimization in a sensor network with a path-constrained mobile observer.” ACM Trans. on Sensor Netw., vol 2, no. 3 (2006): 297-324.

[48] J. Yang and S. Ulukus. “Transmission completion time minimization in an energy harvesting system.” Proc. IEEE CISS 2010.

[49] J. Yang and S. Ulukus. “Optimal Packet Scheduling in an Energy Harvesting Communication System.” IEEE Trans. On Comm., vol. 60, no 1 (2012): 220-230.

[50] K. Tutuncuoglu, and A. Yener. “Short-term throughput maximization for battery limited energy harvesting nodes.” Proc. IEEE ICC 2011, pp. 1-5.

[51] K. Tutuncuoglu, and A. Yener. “Optimum transmission policies for battery limited energy harvesting nodes.” IEEE Trans. on Wireless Commu., vol. 11, no. 3 (2012): 1180-1189.

[52] O. Ozel, K. Tutuncuoglu, J. Yang, S. Ulukus, and A. Yener. “Resource management for fading wireless channels with energy harvesting nodes.” Proc. IEEE INFOCOM 2011, pp. 456-460.

[53] O. Ozel, K. Tutuncuoglu, J. Yang, S. Ulukus, and A. Yener. “Transmission with energy harvesting nodes in fading wireless channels: Optimal policies.” IEEE J. on Sel. Areas in Comm., vol. 29, no. 8 (2011): 1732-1743.

[54] M. A. Antepli, E. Uysal-Biyikoglu, and H. Erkal. “Optimal packet scheduling on an energy harvesting broadcast link.” IEEE Journal on Sel. Areas in Comm., vol. 29, no. 8 (2011): 1721-1731.

[55] O. Ozel, J. Yang, and S. Ulukus. “Optimal broadcast scheduling for an energy harvesting rechargeable transmitter with a finite capacity battery.” IEEE Trans. on Wireless Comm., vol. 11, no. 6 (2012): 2193-2203.

[56] H. Erkal, F. Ozcelik and E. Uysal-Biyikoglu. “Optimal offline broadcast scheduling with an energy harvesting transmitter.” EURASIP Journal on Wireless Communications and Networking, no. 1 (2013): 1-20.

[57] F. Shan, J. Luo, and X. Shen. “Optimal energy efficient packet scheduling with arbitrary individual deadline guarantee.” Computer Networks, vol. 75 (2014): 351-366.

[58] M. Zhao, J. Li, and Y. Yang. “A Framework of Joint Mobile Energy Replenishment and Data Gathering in Wireless Rechargeable Sensor Networks.” IEEE Trans. on Mobile Comp., vol. 13, no. 12 (2014):

2689-2705

[59] X. Ren, W. Liang, and W. Xu. “Data Collection Maximization in Renewable Sensor Networks via Time-Slot Scheduling.” IEEE Trans. on Compers, vol. 64, no. 7 (2015): 1870-1883.

[60] Stephen Boyd, and Lieven Vandenberghe. Convex Optimization. Cambridge University Press, 2004

[61] G.-D. Zhou, and T.-H. Yi. “Recent developments on wireless sensor networks technology for bridge health monitoring.” Mathematical Problems in Engineering, vol. 2013 (2013).

[62] L. Song, and D. Hatzinakos. “Architecture of wireless sensor networks with mobile sinks: Sparsely deployed sensors.” IEEE Trans. on Vehicular Tech., vol. 56, no. 4 (2007): 1826-1836.

[63] B. T. Bacinoglu, E. Uysal-Biyikoglu. “Finite-Horizon Online Transmission Scheduling on an Energy Harvesting Communication Link with a Discrete Set of Rates.” Journal of Communications and Networks, vol. 16, no. 3 (2014): 393-300.

[64] A. Bodin, D.Gunduz. “Energy Harvesting Communication System with a Finite Set of Transmission Rates.” Proc. VDE European Wireless, 2014.

[65] David Tse, and Pramod Viswanath. Fundamentals of Wireless Communication. Cambridge University Press, 2005.

[66] F. Yao, A. Demers, and S. Shenker. “A scheduling model for reduced CPU energy.” Symp. IEEE Foundations of Computer Science, 1995.

[67] T. Ishihara, and H. Yasuura. “Voltage scheduling problem for dynamically variable voltage processors.” Proc. ACM International Symposium on Low Power Electronics and Design, 1998.

[68] W.-C. Kwon, and T. Kim. “Optimal voltage allocation techniques for dynamically variable voltage processors.” Proc. ACM Design Automation Conference 2003.

[69] M. Li, and F. F. Yao. “An Efficient Algorithm for Computing Optimal Discrete Voltage Schedules.” SIAM J. Comput., vol. 35, no. 3 (2005): 658-671.

致　谢

在不知不觉中我写到了致谢部分，亦如在不知不觉中我在东南大学业已七年，我的硕博学习即将结束。在感慨光阴似箭、日月如梭的同时，我要借此机会向一路陪伴我的老师、同学、朋友和家人致以深深的感谢。

首先，我要感谢我的导师罗军舟教授。在我攻读博士学位期间，罗老师对我学习、科研、生活等各方面都照顾有加。在本文攥写过程中，他悉心指导我确立大纲，在整体上把握、在细节上拿捏，倾注了大量心血。感谢罗老师给我提供了宝贵的公派联合培养的机会，支持我选择自己感兴趣的科研方向，是我的科研指路人。感谢他引导我建立崇高的科研精神与正确的科研目标，这是我一辈子的财富：立大志、做大事；为中国的计算机事业发展做贡献。罗老师治学严谨、工作刻苦、诲人不倦，是我终身学习的榜样。

其次，我要感谢我在美国密苏里大学堪萨斯城分校两年联合培养期间的导师沈孝钧教授。我刚到美国时，由于年级比较小，科研基础薄弱，他毫无保留地传授令我受益终身。沈老师善于思考并且勤于思考的行动，一丝不苟的治学态度，不畏艰难冲击难题的精神，这些都深远地影响着我。感谢沈老师与沈师母在生活上对我无微不至的照顾，令我在异国他乡倍感温暖。

再次，我要感谢吴巍炜副教授。吴巍炜老师思维敏锐、洞察力强、算法理论基础深厚，他给我带来全新的科研方法与思维方式。感谢吴老师在许多具体科研问题上给我指点迷津，与他富有成效的讨论才使得我能够在科研上继续成长。吴老师平易近人，能够与学生打成一片。

我还要感谢实验室的曹玖新教授、刘波副教授、宋爱波副教授、杨明副教授、李伟副教授、东方副教授、方效林老师、吴文甲老师、张竞慧老师、凌振老师等。还有与我一起奋斗过的师兄、师弟与同学们，他们是熊润群、王宇翔、金嘉晖、龚奇源、蒋健、夏怒、陆悠、李辉、吴保川、陈忠文、沈典、顾梁、史济源、丁玎等。以及我在公派期间结识的其他公派人员：陶蕾、刘雁孝、杨翩翩、王业静、蒋忠辉，以及中国留学生蒋玮霖、丁璐、马尼拉等。

衷心感谢我的父母、家人以及我的妻子张文燕及其家人，感谢他们在我漫长的求学生涯中所给予的理解、支持、关心、照顾。最后还要特别感谢我尚在襁褓中的女儿单君和，她给我带来无限的幸福。

谨以此文献给所有关心我、帮助我、陪伴我的人们！

单冯

2015 年 6 月于东南大学九龙湖校区

攻读博士期间论文发表情况

(1) **Feng Shan**, Junzhou Luo, Weiwei Wu and Xiaojun Shen. “Optimal Wireless Power Transfer Scheduling for Delay Minimization.” IEEE INFOCOM 2016, San Francisco, CA, USA, April 10-15, 2016. (CCF 推荐 A 类会议, EI 收录)。

(2) **Feng Shan**, Junzhou Luo, Weiwei Wu, Minming Li, Xiaojun Shen. “Discrete Rate Scheduling for Packets with Individual Deadlines in Energy Harvesting Systems.” IEEE Journal on Selected Areas in Communications (JSAC), vol. 33, no. 3 (2015): 438-451. (CCF 推荐 A 类期刊，SCI 收录：影响因子 4.138，EI 收录)。

(3) **Feng Shan**, Junzhou Luo, and Xiaojun Shen. “Optimal energy efficient packet scheduling with arbitrary individual deadline guarantee.” Elsevier Computer Networks (COMNET), vol. 75 (2014): 351-366. (CCF 推荐 B 类期刊，SCI 收录：影响因子 1.256，EI 收录)。

(4) **Feng Shan**, Weifa Liang, Jun Luo, and Xiaojun Shen. “Network lifetime maximization for time-sensitive data gathering in wireless sensor networks.” Elsevier Computer Networks (COMNET), vol. 57, no. 5 (2013): 1063-1077. (CCF 推荐 B 类期刊，SCI 收录：影响因子 1.256，EI 收录)。

(5) Xili Wan, **Feng Shan**, Xiaojun Shen. “An Optimal Algorithm for Time-Slot Assignment in SS/TDMA Satellite Systems.” Proc. ICCCN 2013, International Conference on Computer Communications and Networks. (CCF 推荐 C 类会议，EI 收录)。

(6) Bo Liu, Junzhou Luo, **Feng Shan**, Wei Li, Jiahui Jin, and Xiaojun Shen. “On Finding Maximum Disjoint Paths for Many-to-One Routing in Wireless Multi-Hop Network.” IEICE Transactions on Information and Systems, vol. 97, no. 10 (2014): 2632-2640. (SCI 收录，EI 收录)

攻读博士期间参与的科研项目

(1) 国家自然科学基金重点项目“AMS 数据计算环境应用示范”(90912002)，2009 年 7 月至 2010 年 12 月。参与基于 gLite 的 AMS 应用示范平台的搭建与开发，主要负责 d-Cache 存储部分。

(2) 国家自然科学基金项目“带能量供给与变速延迟约束的电压调整节能调度算法研究”(61300024)，2014 年 1 月至 2016 年 12 月。参与设计节能调度算法，解决带能量供给与变速延迟约束的电压调整问题，并对算法的最优性进行理论分析与证明。

(3) 国家自然科学基金项目“面向能耗优化的无线 Mesh 网络节点部署与节能调度机制研究”(61402104)，2015 年 1 月至 2017 年 12 月。从事无线 Mesh 网络节点部署与节能调度的理论研究，设计能耗优化的调度机制，并进行理论分析。

(4) 江苏省自然科学基金项目“带电压调整延迟与动态能量约束的能耗最优化算法研究”(BK20130634)，2013 年 7 月至 2016 年 6 月。参与设计能耗优化算法，解决带电压调整延迟与动态能量约束的电压调整问题，并进行理论分析与证明。

(5) 江苏省自然科学基金项目“基于拓扑规划和节能调度的无线 Mesh 网络能耗优化研究”(BK20140648)，2014 年 7 月至 2017 年 6 月。从事无线 Mesh 网络能耗优化的理论研究，设计基于拓扑规划的节能调度算法，并对该算法进行理论分析。

作者简介

姓 名	单 冯	性 别	男	出生年月	1985 年 5 月
籍 贯	江苏南通			电子邮件	shanfeng@seu.edu.cn
通信地址	南京市江宁区东南大学路 2 号东南大学计算机科学与工程学院 江苏省网络与信息安全重点实验室，211189				
联系电话	13914487644				

教育与经历

2008.9-今	东南大学	计算机科学与工程学院	攻读工学博士学位 (硕博连读)
2010.9-2012.8	美国密苏里大学 堪萨斯城分校	计算与工程学院	国家公派 -联合培养
2004.9-2008.6	河海大学	计算机与信息工程学院	获得工学学士学位

研究方向

1) 算法设计与分析
2) 无线数据传输节能调度

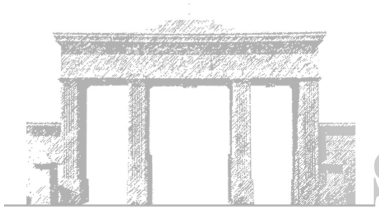
攻读博士学位期间参与的科研项目

1) 国家自然科学基金重点项目“AMS 数据计算环境应用示范”(90912002)，2009 年 7 月至 2010 年 12 月。
2) 国家自然科学基金项目“带能量供给与变速延迟约束的电压调整节能调度算法研究”(61300024)，2014 年 1 月至 2016 年 12 月。
3) 国家自然科学基金项目“面向能耗优化的无线 Mesh 网络节点部署与节能调度机制研究”(61402104)，2015 年 1 月至 2017 年 12 月。
4) 江苏省自然科学基金项目“带电压调整延迟与动态能量约束的能耗最优化算法研究”(BK20130634)，2013 年 7 月至 2016 年 6 月。
5) 江苏省自然科学基金项目“基于拓扑规划和节能调度的无线 Mesh 网络能耗优化研究”(BK20140648)，2014 年 7 月至 2017 年 6 月。

获奖情况

2008-2009 学年　东南大学研究生二等奖学金
2009-2010 学年　东南大学研究生二等奖学金
2010　获得国家留学基金委资助，到美国高校进行两年联合培养

心求至善



SOUTHEAST UNIVERSITY