分类号：　　　　　　　　　　　　　　　　　密级：
UDC：　　　　　　　　　　　　　　　　　　学校代码：

青島大学

硕士学位论文

机会网暨蜂窝网算法与协议
—路由，数据分发与能耗优化

由　磊

指　导　教　师　魏长江教授，李建波副教授

学科专业名称　软件工程

论文提交日期　2015 年 6 月 16 日

论文答辩日期　2015 年 6 月 16 日

答辩委员会主席

关键词：机会网络　蜂窝网络　路由协议　数据分发　优化算法

摘　　要

Abstract

Keywords: opportunistic network; cellular network; routing protocol; data dissemination; optimization algorithm

目　　录

引　言

首段先从无线网络分类讲起，然后撇开较成熟的网络，而转到移动无线网上，此外再介绍 4G,5G 技术及亟待攻克的问题，然后，介绍一下引言的结构安排。

I. 无线网络分类概述

主要参考来源：Wikipedia，介绍无线网络分类，及其区别

II. 移动无线网研究历程——从 MANET 到 PCN

主要参考来源：Mobile Ad Hoc Networking [1]

III. 相辅相成——机会网与蜂窝网技术融合

其中一项，可以介绍一下 opportunistic offloading

IV. 核心技术简介：机会路由，数据分发与能耗优化

IV-A. 机会路由

IV-B. 数据分发

IV-C. 能耗优化

第一章　研究现状

1.1　机会路由协议

1.2　数据分发协议

1.3　蜂窝网能耗优化

第二章　论文工作总览

第一部分

最优化机会路由算法研究

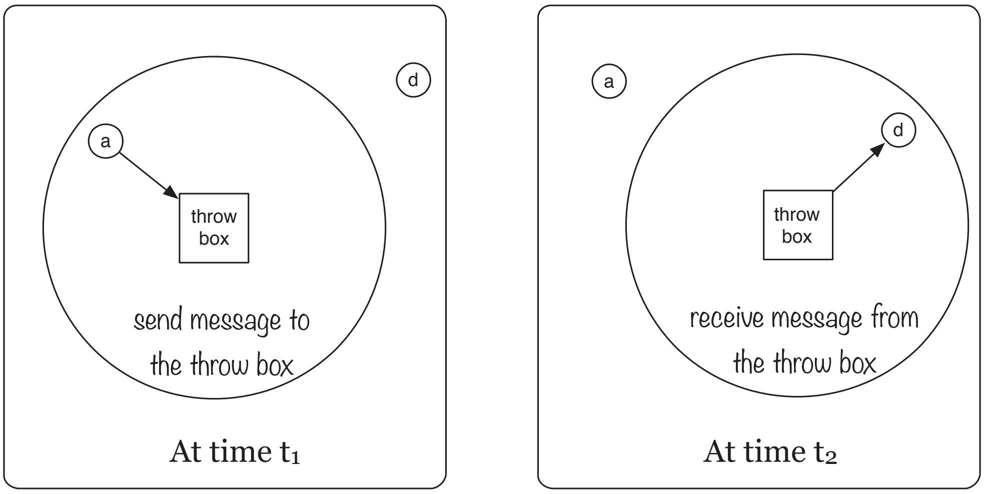
第三章　机会网路由优化概述

第四章　基于移动模式最优节点群组选取的路由算法

在许多节点具有一定社会属性的机会网络中，具有共同兴趣的移动用户往往访问一些与其兴趣相关的地点。研究表明，50% 的移动用户会在某一个特定的接入点（access point, AP）上花费约 74% 的时间 [2]。换言之，节点往往具有频繁访问某一或某一部分地点（简称为常访地点）的特点。这些常访地点可被看做 “连接” 这些节点的枢纽。可以通过在常访地点部署缓存设备，用以辅助消息传递，例如投掷盒（throw-box）[3] 等设备。缓存设备具有普通移动节点不具备的优势。首先，由于部署的缓存设备位置固定在常访地点，且节点往往在常访地点停留一段时间，所以节点与缓存设备之间具有比移动节点之间更加稳定的连接。其次，这些固定的缓存设备，没有如同移动节点那样便携性的限制和要求，故其存储容量往往比移动节点要大许多。由此，可以利用在常访地点部署缓存设备作为中继枢纽，从而提升机会网络中路由算法的性能表现。

本章研究多副本单播机会路由算法，基本假设如下。每条消息具有一对“源—目的”节点对（source-destination node pair），并且每条消息可在网络中具有多份拷贝。不同于以往基于社会网络分析求解社会关系的方法，或自定义社会属性的方法，本章利用节点收集的移动记录信息，从中提取出节点（群组）移动模式，并以此为依据选出最优中继节点群组。基本思想为：将持有某一条特定消息（或其拷贝）的任意一组节点看成一个整体，从记录信息中提取出移动模式，从而确定该节点群组的常访地点集合 A_1。目的节点的常访地点 A_2，也以相同方法获取。进而，取交集求得 $A = A_1 \cap A_2$，作为该节点群组与目的节点的共同常访地点集合。利用集合 A，可以求出消息的预测投递率。路由目标即为求得能使预测投递概率最大的节点群组。此外，超出有效期（deadline）的消息不再具有价值，在求解移动模式的过程中，消息的时效性也被考虑在内。本章基于节点的移动模式，提出构造最优节点群组作为中继节点群的移动模式相关最优路由算法（Movement Pattern-Aware optimal Routing, MPAR）。就目前掌握的资料来看，这是首次利用节点群组移动模式进行社会相关的路由算法研究。

本章组织如下：第4.1节引入系统模型及基本定义；第4.2节分析路由问题背后蕴含的两个关键属性，并给出最优路由问题的形式化定义；第4.3节证明了该最优问题的计算复杂性，并提出了启发式算法用以求解近似最优解；第4.4节给出路由算法的详细过程及描述；第4.5节对仿真实验结果进行分析；第4.6节总结本章内容。



(a) 节点 a 向投掷盒发送消息　　(b) 节点 d 从投掷盒接受消息

图 4–1 利用投掷盒完成消息从节点 a 到节点 d 的传递操作.

4.1 系统模型及基本定义

4.1.1 网络模型

网络节点集合记为 $\overline{N} \triangleq \{n_i|1 \leq i \leq n\}$. 节点在给定的地点集合间移动，地点集合记为 $\overline{A} \triangleq \{a_j|1 \leq j \leq m\}$. 任意一个节点 n_i 的常访地点集合 $A(n_i) \subseteq \overline{A}$. 该模型源于真实的移动网络，一个典型的实例是 Dartmouth College 的 Wi-Fi Campus Network [4]。另一符合该模型的一类网络是 VANET。在 VANET 中，大量的公交车，电车等在车站等固定地点间移动。我们假设节点 n_i 访问任意一个地点 a_j 的时间间隔服从指数分布。此外，在任意地点内都部署有一个投掷盒装置，用于接受，缓存及传递消息。在此模型中，投掷盒的缓存容量假设为足够大，即在投掷盒中不会因缓存满而产生消息丢弃。如图 4–1所示，节点 a 在 t_1 时刻将消息托管给投掷盒，在 t_2 时刻，该消息的目的节点 d 进入投掷盒的传输范围内，并从投掷盒接受该消息，至此，消息的投递操作完成。

我们假设每条消息 l 都具有一个值 τ_l，代表消息 l 在网络的剩余生存时间。随着时间消逝，τ_l 的值逐渐减少，当 $\tau_l = 0$ 时，该消息将从节点缓存中删除，不再存在于网络中。消息的生存时间属性，一定程度上避免了消息在网络中停留的时间过长。此外，应用程序所产生的消息因具有时效性，往往也会对消息设定一个生存时间。例如新闻发布或者广告发布服务，消息只在一定时间内有效。当消息超出某个时间即不再具有投递价值。

4.1.2 基本定义

由于作为手持设备的节点附属于具有社交属性的人群，节点的移动模式具有一定的周期性。举例而言，Smith 先生在周一到周五上班，故其常访地点可能包含家，办

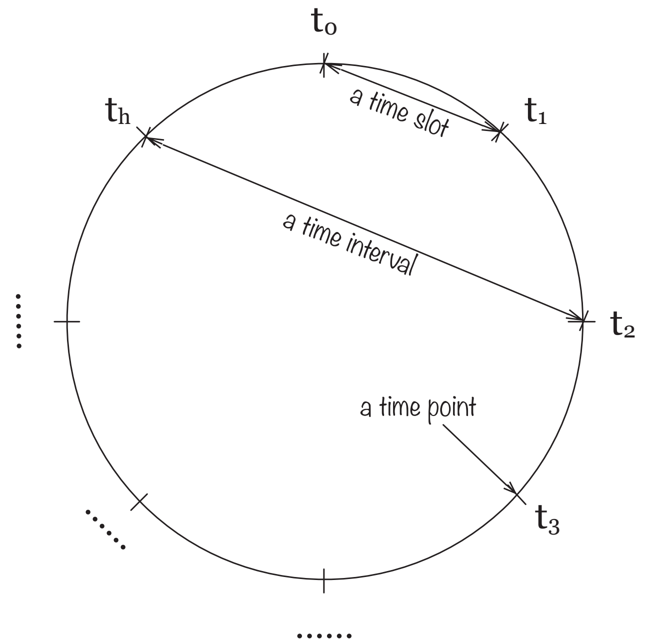


图 4–2 周期 T 划分为 h 个时间槽

公室，某些公交站点及一些快餐店等。周末，Smith 先生通常去健身俱乐部或咖啡馆，或者一些不同于工作时间去的地方。然而，他的行为通常以一星期为一个周期，在一定程度上重复，即其移动记录具有一定的周期性。假设周期为 T，若将 T 划分为 h 个时间槽，则每个时间槽的长度为 $\frac{T}{h}$。将这些划分好的时间槽的边界时间点，用序列 $< t_0, t_1, t_2, \ldots, t_h >$ 表示，则任意两个区间点 t_s 与 t_e（$t_s < t_e$）形成时间区间 $[t_s, t_e]$，如图图 4–2所示。针对时间区间 $[t_s, t_e]$，可以从节点的移动记录中，提取出对应的移动模式。节点的移动记录见定义1。

定义 1. 移动记录

节点 n_i 的移动记录定义为一个 $h \times m$ 矩阵，记为 $\mathbb{R}(i)$

$$\mathbb{R}(i) = \begin{bmatrix} \overbrace{\begin{matrix} {}^{1}\!/r^i_{1,1} & {}^{1}\!/r^i_{1,2} & \ldots & {}^{1}\!/r^i_{1,m} \\ {}^{1}\!/r^i_{2,1} & {}^{1}\!/r^i_{2,2} & \ldots & {}^{1}\!/r^i_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ {}^{1}\!/r^i_{h,1} & {}^{1}\!/r^i_{h,2} & \ldots & {}^{1}\!/r^i_{h,m} \end{matrix}}^{m\text{个地点}} \end{bmatrix} \left.\vphantom{\begin{matrix} a\\a\\a\\a\\a \end{matrix}}\right\} h\text{个时间槽}$$

其中第 k 行的向量 $\mathbb{R}(i,k)$ 代表时间槽 $[t_{k-1}, t_k]$ 内的移动记录，且有

$$\mathbb{R}(i,k) = [{}^{1}\!/r^i_{k,1}, {}^{1}\!/r^i_{k,2}, \ldots, {}^{1}\!/r^i_{k,m}]$$

其中 $r^i_{k,j}$ 代表节点 n_i 对地点 a_j 在时间槽 $[t_{k-1}, t_k]$ 内的平均访问时间间隔。

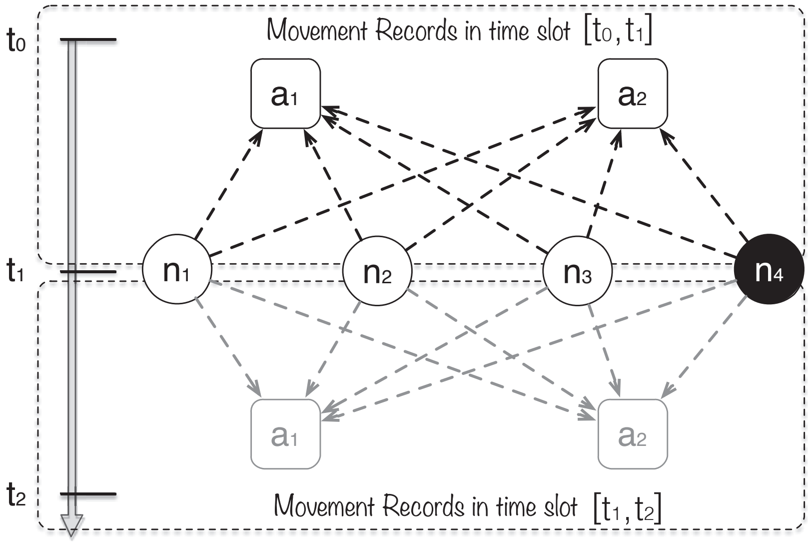


图 4–3 网络场景举例

从上述定义可知，$1/r^i_{k,j}$ 代表节点 n_i 对地点 a_j 的平均访问频率。特别的，若 n_i 在时间槽 $[t_{k-1}, t_k]$ 内从未到达过 a_j，则设定 $r^i_{k,j} = \infty$，于是有 $1/r^i_{k,j} = 0$. 对于所有的时间槽，可以求出节点 n_i 对于地点 a_j 的平均访问时间间隔 $M_{i,j}$，记为

$$M_{i,j} = \sum_{k=1}^{k=h} r^i_{k,j} \Big/ h \tag{4–1}$$

4.2 路由问题概览

在讨论路由相关细节之前，先对路由问题做一个总览。在4.2.1小节中，讨论如何从一组节点中提取对应的节点群组移动模式。随后在4.2.2小节中，分析路由相关的两个关键属性。最后，在4.2.3小节中，给出最优路由问题的形式化定义。

4.2.1 移动模式

移动模式从节点的移动记录中提取。首先定义函数 $\mathbb{E}$，用以将移动记录向量转化为对应的移动模式向量。

定义 2. 函数 $\mathbb{E}$

$$\mathbb{E}([x_1, x_2, \ldots, x_m]) = [\varkappa_1, \varkappa_2, \ldots, \varkappa_m]$$

其中

$$\varkappa_j = \begin{cases} 1 & x_j \geq \frac{\delta}{m} \sum_{i=1}^{i=m} x_i \\ 0 & otherwise \end{cases} \tag{4–2}$$

$0 < \delta < 1$ 是预设的系统参数。

函数 $\mathbb{E}$ 用于过滤节点—地点间的访问记录，不常访问的地点将被过滤掉。基于函数 $\mathbb{E}$ 可以直接定义移动模式，如定义3。

定义 3. 移动模式. 对于任意节点群组 N，其在时间区间 $[t_p, t_q]$ 内的移动模式，记为 $\mathcal{P}(V, [t_s, t_e])$, 定义为

$$\mathcal{P}(N, [t_s, t_e]) = \mathbb{E}(\sum_{n_x \in N} \sum_{i=s}^{i=e} \mathbb{R}(x, t_i)) \tag{4–3}$$

在定义3中，移动模式 $\mathcal{P}$ 从某节点（群组）在某个特定时间区间 $[t_s, t_e]$ 上的所有移动记录向量通过累加后，以函数 $\mathbb{E}$ 处理后得出。举例而言，假设如图 4–3所示的网络场景，网络中总共有四个节点 n_1, n_2, n_3, n_4 以及两个地点 a_1, a_2。其中 n_4 是目的节点，周期 T 被划分为两个时间槽，简记为 $[t_0, t_1]$ 与 $[t_1, t_2]$。

图 4–3中每条边的权值如表 4–1所示。权值代表“节点—地点”对访问平均间隔时间。针对每个节点的记录矩阵，表示为 $\mathbb{R}(1), \mathbb{R}(2), \mathbb{R}(3), \mathbb{R}(4)$。

$$\mathbb{R}(1) = \begin{bmatrix} 1/2.08 & 1/4.65 \\ 1/6.02 & 1/2.95 \end{bmatrix} \quad \mathbb{R}(2) = \begin{bmatrix} 1/4.4 & 1/4.3 \\ 1/4.0 & 1/4.5 \end{bmatrix}$$

$$\mathbb{R}(3) = \begin{bmatrix} 1/8.05 & 1/1.11 \\ 1/6.05 & 1/15.49 \end{bmatrix} \quad \mathbb{R}(4) = \begin{bmatrix} 1/2.6 & 1/3.3 \\ 1/3.5 & 1/3.5 \end{bmatrix}$$

表 4–1 网络场景图 4–3边权值

		a_1	a_2
$[t_0, t_1]$	n_1	2.08	4.65
	n_2	4.4	4.3
	n_3	8.05	1.11
	n_4	2.6	3.3
$[t_1, t_2]$	n_1	6.02	2.95
	n_2	4.0	4.5
	n_3	6.05	15.49
	n_4	3.5	3.5

节点集合 $\{n_1, n_2, n_3\}$ 的移动模式，可以利用公式4–3从对应的移动记录中提取出来。对应的移动记录累加计算如下：

$$\sum_{i=1}^{i=3} \sum_{j=1}^{j=2} \mathbb{R}(i, j) = \Big[\frac{1}{2.08} + \frac{1}{4.4} + \frac{1}{8.05} + \frac{1}{6.02} + \frac{1}{4.0} + \frac{1}{6.05}, \\ \frac{1}{4.65} + \frac{1}{4.3} + \frac{1}{1.11} + \frac{1}{2.95} + \frac{1}{4.5} + \frac{1}{15.49}\Big] = [1.414, 1.974]$$

针对公式4–2，设定参数 $\delta = 0.95$，从定义2得出

$$\frac{\delta}{m}\sum_{i=1}^{i=m} v_i = \frac{0.95}{2} \times (1.414 + 1.974) \approx 1.609$$

所得值 1.609 大于向量第一个元素值 1.414，且小于向量第二个元素值 1.974，故对应移动模式提取如下

$$\mathcal{P}(\{n_1, n_2, n_3\}, [t_0, t_2]) = [0, 1]$$

移动模式 $\mathcal{P}$ 指明了节点群组 $\{n_1, n_2, n_3\}$ 在时间区间 $[t_0, t_2]$ 内的的常访区域集合，即 $A(\{n_1, n_2, n_3\}, [t_0, t_2]) = \{a_2\}$。对于集合 $\overline{N}$
$\{n_d\}$，共有 $2^{n-1} - 1$ 个非空子集，每一个子集可被看做用于向目的节点 n_d 合作投递消息的一个整体。在本例中，基于移动记录，提取出非空节点集 $\{n_1, n_2, n_3\}$ 以及目的节点 n_4 的移动模式，结果如下

$$\begin{array}{ll}
\mathcal{P}(\{n_1\}, [t_0, t_2]) = [1, 0] & \mathcal{P}(\{n_1, n_2\}, [t_0, t_2]) = [1, 0] \\
\mathcal{P}(\{n_2\}, [t_0, t_2]) = [1, 1] & \mathcal{P}(\{n_1, n_3\}, [t_0, t_2]) = [0, 1] \\
\mathcal{P}(\{n_3\}, [t_0, t_2]) = [0, 1] & \mathcal{P}(\{n_2, n_3\}, [t_0, t_2]) = [0, 1] \\
\mathcal{P}(\{n_4\}, [t_0, t_2]) = [1, 1] & \mathcal{P}(\{n_1, n_2, n_3\}, [t_0, t_2]) = [0, 1]
\end{array}$$

对应的常访地点集合，如下

$$\begin{array}{ll}
A(\{n_1\}, [t_0, t_2]) = \{a_1\} & A(\{n_1, n_2\}, [t_0, t_2]) = \{a_1\} \\
A(\{n_2\}, [t_0, t_2]) = \{a_1, a_2\} & A(\{n_1, n_3\}, [t_0, t_2]) = \{a_2\} \\
A(\{n_3\}, [t_0, t_2]) = \{a_2\} & A(\{n_2, n_3\}, [t_0, t_2]) = \{a_2\} \\
A(\{n_4\}, [t_0, t_2]) = \{a_2\} & A(\{n_1, n_2, n_3\}, [t_0, t_2]) = \{a_1, a_2\}
\end{array}$$

4.2.2 路由相关的两个关键属性

4.2.2.1 投递概率

设节点集合 N 在时间区间 $[t_0, t_0 + t_l]$ 内的移动模式为 $P(N, [t_0, t_0 + t_l])$，目的节点 n_d 的移动模式为 $\mathcal{P}(n_d, [t_0, t_0 + \tau_l])$，共同常访地点集合记为 A，则有 $A = A(N, [t_0, t_0 + \tau_l]) \cap A(n_d, [t_0, t_0 + \tau_l])$。记任意地点 a_j 及任意节点 $n_i \in N$ 满足节点 n_i 和 n_d 都在某一时间访问 a_j。设随机变量 $T_{i,j}$ 及 $T_{d,j}$ 分别代表 n_i 和 n_d 到达 a_j 的时间，若要使消息能够成功传递，则须满足 $T_{i,j} < T_{d,j} < \tau_l$，换言之，$n_i$ 须先于 n_d 到达 a_j

托管消息，且 n_d 需在消息到期之前到达 a_j 取出消息。为了便于讨论，设起始时间点 $t_0 = 0$。此外，如4.1节所述，任意节点访问任意地点的时间间隔服从指数分布。从公式4–1可以计算出节点 n_i 访问地点 a_j 的指数分布参数 $\lambda_{i,j}$，即

$$\lambda_{i,j} = {}^{1}/_{M_{i,j}}$$

对于 A 为空集的情形，默认其预测投递概率为 0，到达其它地点的期望时延为 ∞。对于 A 非空的情形，可依公式 (4–4) 预测节点 n_i 对 n_d 的投递率。进一步的，节点集 N 对 n_d 的投递率可依公式 (4–5) 预测。

$$\begin{aligned}
P_{i,d} &= 1 - \prod_{a_j \in A} \left(1 - P(T_{i,j} < T_{d,j} < \tau_l)\right) \\
&= 1 - \prod_{a_j \in A} \left(1 - \int_0^{\tau_l}\int_{t_{i,j}}^{\tau_l} f_{T_{i,j},T_{d,j}}\left(t_{i,j}, t_{d,j}\right) dt_{d,j} dt_{i,j}\right) \\
&= 1 - \prod_{a_j \in A} \left(1 - \int_0^{\tau_l}\int_{t_{i,j}}^{\tau_l} f_{T_{i,j}}(t_{i,j}) f_{T_{d,j}}(t_{d,j}) dt_{d,j} dt_{i,j}\right) \\
&= 1 - \prod_{a_j \in A} \left(1 - \int_0^{\tau_l}\int_{t_{i,j}}^{\tau_l} \lambda_{d,j}\lambda_{i,j} e^{-\left(\lambda_{d,j}t_{d,j} + \lambda_{i,j}t_{i,j}\right)} dt_{d,j} dt_{i,j}\right) \\
&= 1 - \prod_{a_j \in A} \left(1 - \frac{\lambda_{i,j}\left(1 - e^{-\tau_l\left(\lambda_{d,j}+\lambda_{i,j}\right)}\right)}{\lambda_{d,j} + \lambda_{i,j}} - \left(e^{\tau_l\lambda_{i,j}} - 1\right) e^{-\tau_l\left(\lambda_{d,j}+\lambda_{i,j}\right)}\right)
\end{aligned} \tag{4–4}$$

$$\begin{aligned}
P_{N,d} &= 1 - \prod_{n_i \in N} \left(1 - P_{i,d}\right) \\
&= 1 - \prod_{n_i \in N}\prod_{a_j \in A} \left(1 - \frac{\lambda_{i,j}\left(1 - e^{-\tau_l\left(\lambda_{d,j}+\lambda_{i,j}\right)}\right)}{\lambda_{d,j} + \lambda_{i,j}} - \left(e^{\tau_l\lambda_{i,j}} - 1\right) e^{-\tau_l\left(\lambda_{d,j}+\lambda_{i,j}\right)}\right)
\end{aligned} \tag{4–5}$$

特别而言，当消息生存时间设为无穷时，即 $\tau_l = \infty$，有

$$\begin{aligned}
P_{i,d} &= 1 - \prod_{a_j \in A} \left(1 - \int_0^{\infty}\int_{t_{i,j}}^{\infty} \lambda_{d,j}\lambda_{i,j} e^{-\left(\lambda_{d,j}t_{d,j} + \lambda_{i,j}t_{i,j}\right)} dt_{d,j} dt_{i,j}\right) \\
&= 1 - \prod_{a_j \in A} \left(1 - \frac{\lambda_{i,j}}{\lambda_{d,j} + \lambda_{i,j}}\right)
\end{aligned} \tag{4–6}$$

对于 $P_{N,d}$，有

$$P_{N,d} = 1 - \prod_{n_i \in N}\prod_{a_j \in A} \left(1 - \frac{\lambda_{i,j}}{\lambda_{d,j} + \lambda_{i,j}}\right) = 1 - \prod_{n_i \in N}\prod_{a_j \in A} \frac{\lambda_{d,j}}{\lambda_{d,j} + \lambda_{i,j}} \tag{4–7}$$

对于图 4–3中的例子，为简便起见，讨论 $\tau_l = \infty$ 的情况 (形式化定义时，会给出一般情况下的定义)。对于节点集合 $\{n_1, n_2, n_3\}$ 而言，常访地点集合 $A(N, [t_0, t_3]) = \{a_2\}$；对于目的节点 n_d 而言，其常访地点集合 $A(n_4, [t_0, t_3]) = \{a_1, a_2\}$；由此可得出共同常访地点 $A = \{a_2\}$。每一个“节点—地点”对的平均访问时间间隔，可从公式 (4–1) 获得，即

$$\begin{array}{llll} M_{1,1} = 4.05 & M_{2,1} = 4.2 & M_{3,1} = 7.05 & M_{4,1} = 3.05 \\ M_{1,2} = 3.8 & M_{2,2} = 4.4 & M_{3,2} = 8.3 & M_{4,2} = 3.4 \end{array}$$

由以上计算结果，可预测节点集合 N 的投递概率，如下

$$\begin{aligned} P_{\{n_1,n_2,n_3\},4} &= 1 - \left(\frac{\lambda_{4,2}}{\lambda_{1,2}+\lambda_{4,2}}\right)\left(\frac{\lambda_{4,2}}{\lambda_{2,2}+\lambda_{4,2}}\right)\left(\frac{\lambda_{4,2}}{\lambda_{3,2}+\lambda_{4,2}}\right) \\ &= 1 - \left(\frac{M_{1,2}}{M_{1,2}+M_{4,2}}\right)\left(\frac{M_{2,2}}{M_{2,2}+M_{4,2}}\right)\left(\frac{M_{3,2}}{M_{3,2}+M_{4,2}}\right) = 0.789 \end{aligned}$$

4.2.2.2 期望时延

除了投递概率之外，另一个关键的属性即节点移动到其他地点的期望时延。设节点 n_i 的常访地点集合为 A_i，用随机变量 D_i 代表 n_i 移动到下一个地点 $a_j \in A_i$ 的期望时延。形式上，有 $D_i = \min\{T_{i,1}, T_{i,2}, \ldots, T_{i,|A_i|}\}$，由此可求得 D_i 的概率密度函数为

$$f_{D_i}(t) = \sum_{a_k \in A_i} \lambda_{i,k} \prod_{a_h \in A_i} e^{-\lambda_{i,h} t}$$

于是有

$$\begin{aligned} E[D_i] = \int_{-\infty}^{\infty} t f_{D_i}(t) dt &= \sum_{a_k \in A_i} \int_0^{\infty} t \lambda_{i,k} \prod_{a_h \in A_i} e^{-\lambda_{i,h} t} dt \\ &= \sum_{a_k \in A_i} \frac{\lambda_{i,k}}{\left(\sum_{a_k \in A_i} \lambda_{i,h}\right)^2} = \frac{1}{\sum_{a_k \in A_i} \lambda_{i,k}} \end{aligned} \tag{4–8}$$

对于图 4–3所示例子，可以分别对节点 n_1, n_2 及 n_3 根据公式 (4–8) 计算期望时延 $E[D_i]$。

$$\begin{aligned} E[D_1] &= {}^1/_{\lambda_{1,1}} = M_{1,1} = 4.05 \\ E[D_2] &= \tfrac{1}{\lambda_{2,1}+\lambda_{2,2}} = \tfrac{M_{2,1}M_{2,2}}{M_{2,1}+M_{2,2}} = 2.15 \\ E[D_3] &= {}^1/_{\lambda_{3,2}} = M_{3,2} = 8.3 \end{aligned}$$

对于任意一对节点 n_i 和 n_j，若 $E[D_i] < E[D_j]$，则从期望意义上讲，n_i 到达下一个新地点的时间要早于 n_j，这意味着 n_i 能够更早的将消息托管到新的投掷盒。在机会网络中多使用多拷贝策略进行路由，快速的向网络中传播足够数目的消息副本，对提升路由性能至关重要。期望时延 $E[D]$ 是一个很好的衡量指标，应尽可能的利用 $E[D]$

较小的节点分发消息。在本章提出的基于局部搜索的路由算法 Local-MPAR 中，$E[D]$ 被用于衡量节点作为网络中唯一的活跃节点是否合适。在基于禁忌搜索的路由算法 Tabu-MPAR 中，$E[D]$ 被用作向网络中非对称喷洒消息副本的衡量指标。在表 4–1所示例子中，节点 n_2 最适合作为活跃节点，因为其期望时延 $E[D]$ 最低，因此在期望意义上，n_2 可以较早的将消息拷贝托管给其它地点的投掷盒。

4.2.3 路由问题形式化定义

机会网络中的路由问题，其优化目标可有多种。衡量路由算法性能的最主要的两个指标，即是消息投递率和平均端到端时延。尽管端到端时延在路由优化中被看做一项非常重要的指标，但路由的性能表现仍需要以可被接受的投递概率为基础。换言之，较高的投递概率是保证网络正常工作的基础。本章意图最大化消息的投递概率，进而提高路由性能表现。即寻找子集 $N \subseteq \overline{N}$，从而使得 $P_{N,d} \geq P_{N',d}$ 对于 $\forall N' \subseteq \overline{N}$ 成立，其中 n_d 是消息的目的节点。

在图 4–3所示例子中，可依4.2.2小节中公式 (4–5)，预测集合 $\{n_1, n_2, n_3\}$ 所有非空子集的投递率，如下

$$
\begin{aligned}
&P_{\{n_1\},4} = 0.430 \qquad && P_{\{n_1,n_2\},4} = 0.670 \\
&P_{\{n_2\},4} = 0.673 \qquad && P_{\{n_2,n_3\},4} = 0.600 \\
&P_{\{n_3\},4} = 0.291 \qquad && P_{\{n_1,n_3\},4} = 0.626 \\
&P_{\{n_1,n_2,n_3\}} = 0.789 &&
\end{aligned}
$$

在本例中，最优集合即 $\{n_1, n_2, n_3\}$。注意计算结果为估计投递率，并不直接与持有该消息的节点数目成正比。投递率 $P_{N,d}$ 基于节点集合 N，N 在整个路由过程中被看做一个整体。对于某个 $N' \subset N$ 而言，概率 $P_{N,d} < P_{N',d}$ 是可能发生的。事实上，如图 4–3所示，对于集合 $\{n_2, n_3\}$ 而言，其预测投递率低于集合 $\{n_2\}$ 的预测投递率。如果忽略缓存限制和能耗资源，向整个网络中广播每条消息会获得最好的性能表现，因为在这种情况下增加中继节点的数目，可以提高消息的投递率（或者说至少不可能降低)。然而这种理想的情况在机会网络中往往并不存在。真实情况下，往往倾向于选择能够符合一定移动模式的最小节点群组，对消息进行投递。从例中可以看出，如果把持有该消息的所有节点视为一个整体，则其预测投递率被两项指标所影响：(1）节点集对应的移动模式，(2）节点集的大小。在正式定义路由问题之前，首先给出最优节点集合 N_{opt} 的定义。

定义 4. *The optimal set N_{opt}*
最优集合 N_{opt} 包含网络中的一组节点, 且有

$$N_{opt} = \min\{\arg\max_{N \subseteq \overline{N}} P_{N,d}\}$$

最优集合是能够取得最高预测投递率的最小集合，集合中的节点以合作的方式将消息传递给目的节点。基于此，可定义最优路由问题。

定义 5. N_{opt} 搜索问题.
令 $f(\mathbf{x})$ 代表目标函数，$g(\mathbf{x})$ 代表限制函数, 其中决策变量表示为 $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$，

有 $$f(\mathbf{x}) = \sum_{i=1}^{i=n} x_i \log\left(\prod_{a_j \in A}\left(1 - \frac{\lambda_{i,j}\left(1 - e^{-\tau_l\left(\lambda_{d,j}+\lambda_{i,j}\right)}\right)}{\lambda_{d,j}+\lambda_{i,j}} - \left(e^{\tau_l \lambda_{i,j}} - 1\right) e^{-\tau_l\left(\lambda_{d,j}+\lambda_{i,j}\right)}\right)\right)$$

$$g(\mathbf{x}) = \left|\sum_{i=1}^{i=n} x_i\right|$$

于是问题可以被形式化为如下

$$\begin{gathered}\min f(\mathbf{x}) \\ s.t. \forall \mathbf{x}',\ g(\mathbf{x}') - g(\mathbf{x}) \geq 0, \\ \forall i, x_i \in \{0, 1\}\end{gathered}$$

当求出最优节点集合 $\mathbf{x}_{opt}$ 时，有

$$N_{opt} = \{n_i | x_i > 0, 1 \leq i \leq n\}$$

在上述定义中，存在一个 $\boldsymbol{x}$ 与 N 之间的一对一映射。事实上，有 $f(\boldsymbol{x}) = \log(1 - P_{N,d})$，其中 $f(\boldsymbol{x})$ 与 $P_{N,d}$ 成负相关。由此，N_{opt} 搜索问题被形式化为最优化问题。

4.3 N_{opt} 搜索问题分析

本节讨论 N_{opt} 问题的计算复杂性，并提出了一种启发式方法，以近似计算最优解。首先证明对于在线算法 (online algorithm)，对于算法输入序列中新加入的元素，能够很大程度改变目标函数的值。对于离线算法 (offline algorithm)，给出了 $\mathcal{NP}$-hard 性质证明。最后，本章提出基于禁忌搜索的启发式算法。

4.3.1 计算复杂性证明

在线算法可以分片处理输入序列。换言之，算法的输入是以一个序列方式随时间交付由算法本身处理的，而不同于离线算法，需要一次接受整个算法输入。对于 N_{opt}，其在线算法的输入是网络中节点的移动记录，简单表示为 $\mathbb{R}_1, \mathbb{R}_2, \ldots, \mathbb{R}_q$ (其中序列中

的每个元素都是一个向量，例如对于在时间槽 t_i 的节点 n_x 可以有 $\mathbb{R}_j = \mathbb{R}(x, t_i)$)。假设当前只有序列的前 $1 - -k$ 个元素被输入算法。剩下的问题即探究当下一个新元素 $\mathbb{R}_{k+1}$ 被加入网络时，会在多大程度上影响目标函数的值, 见定理 1

定理 1. *对于 $\forall k \in [1, z)$，即使序列的前 1—k 部分已知，新加入的元素 $\mathbb{R}_{k+1}$ 仍可能将移动模式改变为至少有一个元素不为 0 的任意向量。*

证明. 将任意向量 R_i 表示为如下

$$R_i = [r_1^i, r_2^i, \ldots, r_m^i] \quad \forall j \in [1, m], r_j^i \geq 0$$

向量 $\sum_{i=1}^{i=k} \mathbb{R}_i$ 表示为

$$\sum_{i=1}^{i=k} \mathbb{R}_i = [S_1^k, S_2^k, \ldots, S_m^k]$$

于是对于 $\forall i \in [1, m]$ 及 $k \in [1, z)$ 有

$$S_i^{k+1} = S_i^k + r_i^{k+1}$$

相应地，用 $\mathcal{P}' = \mathbb{E}(\sum_{i=1}^{i=k+1} \mathbb{R}_i)$ 和 $\mathcal{P} = \mathbb{E}(\sum_{i=1}^{i=k+1} \mathbb{R}_i)$ 分别表示在加入新元素 $\mathbb{R}_{k+1}$ 之前和之后的移动模式向量。现证明对于任意的 $\mathcal{P}$, 都存在一个可能的新元素 $\mathbb{R}_{k+1}$，能使向量 $\mathcal{P}'$ 变为 $\mathcal{P}$. 当已知 $\mathcal{P} = [\varkappa_1, \varkappa_2, \ldots, \varkappa_m]$ 时，显然 $\mathbb{R}_{k+1}$ 存在，当且仅当如下线性不等式组成立

$$\forall i \in [1, m] \quad \begin{cases} S_i^{k+1} \geq \delta \sum_{j=1}^{j=m} \left(S_j^{k+1}/m \right) & \varkappa_i = 1 \\ S_i^{k+1} < \delta \sum_{j=1}^{j=m} \left(S_j^{k+1}/m \right) & \varkappa_i = 0 \end{cases}$$

即

$$\begin{cases} \sum_{j=1}^{j=m} r_j^{k+1} - \frac{m}{\delta} r_i^{k+1} \leq \frac{m}{\delta} S_i^k - \sum_{j=1}^{j=m} S_j^k & \varkappa_i = 1 \\ \sum_{j=1}^{j=m} r_j^{k+1} - \frac{m}{\delta} r_i^{k+1} > \frac{m}{\delta} S_i^k - \sum_{j=1}^{j=m} S_j^k & \varkappa_i = 0 \end{cases}$$

对于 $\forall i$, 将 $\frac{m}{\delta} S_i^k - \sum_{j=1}^{j=m} S_j^k$ 表示为 w_i, 并将 $\varkappa_i = 1$ 和 $\varkappa_i = 0$ 时的 w_i 分别表示为 $w_{i|\varkappa_i=1}$ 和 $w_{i|\varkappa_i=0}$。同理，用 r_i^{k+1} 表示 $\varkappa_i = 1$ 和 $\varkappa_i = 0$ 时的对应状态，有

$$\sum_{j=1}^{j=m} r_j^{k+1} - \frac{m}{\delta} r_{i|\varkappa_i=0}^{k+1} > w_{i|\varkappa_i=0} \tag{4–9}$$

且有

$$\sum_{j=1}^{j=m} r_j^{k+1} - \frac{m}{\delta} r_{i|\varkappa_i=1}^{k+1} \leq w_{i|\varkappa_i=1} \tag{4–10}$$

令 $r_{i|\varkappa_i=0}^{k+1} = 0$ 及 $\sum_{j=1}^{j=m} r_j^{k+1} = \max\{w_i\} + \epsilon \ (\epsilon > 0)$, 则等式（4–9）对于 $\forall \varkappa_i = 0$ 都成立。

接下来证明可以选择足够大的 ϵ 值，使得等式4–10成立，即只需让 $\sum_{j=1}^{j=m} r_j^{k+1} - \frac{m}{\delta} r_{i|\varkappa_i=1}^{k+1} \leq \min\{w_i\}$ 对于 $\forall \varkappa_i = 1$ 都成立，即

$$\max\{w_i\} + \epsilon - \frac{m}{\delta} r_{i|\varkappa_i=1}^{k+1} \leq \min\{w_i\}$$

从上述讨论, 只需让下述不等式同时成立.

$$r^{k+1}_{i|\varkappa_i=1} \geq \frac{\delta}{m}(\max\{w_i\} - \min\{w_i\} + 1)\ (0 < \delta < 1) \tag{4–11}$$

$$\sum_{j=1}^{j=m} r^{k+1}_{j|\varkappa_j=1} = \max\{w_i\} + \epsilon \tag{4–12}$$

假设 $\mathcal{P}$ 中 $\varkappa_i = 1$ 的数量是 z, 由于 $0 < z \leq m$, 有

$$z \cdot \frac{\delta}{m}(\max\{w_i\} - \min\{w_i\} + \epsilon) \leq \delta(\max\{w_i\} - \min\{w_i\} + \epsilon)$$

为使等式（4–11）和（4–12）同时成立, 只须有

$$\delta(\max\{w_i\} - \min\{w_i\} + \epsilon) < \max\{w_i\} + \epsilon = \sum_{j=1}^{j=m} r^{k+1}_{j|\varkappa_j=1}$$

即

$$\epsilon > \frac{\delta}{\delta - 1}\min\{w_i\} - \max\{w_i\}$$

证毕 □

定理 2 给出 N_{opt} 计算复杂性的证明。

定理 2. N_{opt} *问题属于* $\mathcal{NP}$*-hard 类.*

证明. 将子集和问题 (Subset Sum Problem, SSP) 归约为 N_{opt} 问题。假设已知 SSP 问题的一个解实例，用 $R = r_1, r_2, \ldots, r_n$，目标值用 v 表示。为清楚起见，将 R 变为如下形式

$$R = \{\log 2^{r_1}, \log 2^{r_2}, \ldots, \log 2^{r_n}\} = \{\log g_1, \log g_2, \ldots, \log g_n\}$$

其中有 $g_i = 2^{r_i}$ $(i \in [1, n])$. 将其所有非空子集表示为 $R_1, R_2 \ldots, R_{2^{|N|-1}}$。接下来，构造对应的 N_{opt} 实例。目标值为 $P = v$, 节点集合为 $N = n_1, n_2, \ldots, n_n$，对应上述提到的集合 R. 对于每一个 N 的非空子集 N_i, 都存在一个 R 的子集 R_i 与其一一对应 (所有的角标都是一一对应的), 将任意子集 N_j 表示为如下

$$N_j = \{n_{j_1}, n_{j_2}, \ldots, n_{j_k}\}$$

对于 $\forall j$, 令 $\lambda_{i,j} = 1, \tau_l = \sqrt{2}$, 对于 $\forall n_i \in N - d$, 令 $\lambda_{i,j} = \Lambda_i$. 此外, 令所有的节点都具有相同的移动记录 $\mathbb{R} = [0, 0, \ldots, 0]$. 于是，对于网络中所有的节点，都有

$$\mathcal{P} = [1, 1, \ldots, 1]$$

于是有

$$P_{N_j,d} = 1 - \prod_{n_i \in N}(1 - \Lambda_i e^{-\Lambda_i - 1})^n$$

令 $\kappa_i = \left(1 - \Lambda_i e^{-\Lambda_i - 1}\right)^n$, 则有

$$P_{N_j,d} = 1 - \kappa_{j_1}\kappa_{j_2}\cdots\kappa_{j_k}$$

由此，对于问题 SSP 的任意一个解实例，一定有一个问题 N_{opt} 的解实例与之对应。换言之

$$v = \log(g_{j_1}g_{j_2}\cdots g_{j_k}) \Longleftrightarrow P_{N_j,d} = 1 - \kappa_{j_1}\kappa_{j_2}\cdots\kappa_{j_n}$$

反之, 若 SSP 问题无解, 则 N_{opt} 问题一定也无解. 因为若 N_{opt} 有解, 则存在一个集合 $R_j = r_{j_1}, r_{j_2}, \ldots, r_{j_n}$ 使下式成立

$$P = P_{N_j}$$

与题设矛盾。

证毕 □

4.3.2　局部陷阱

对于 N_{opt}，一种直接的解法是局部搜索算法。然而，这有可能会陷入局部最优陷阱。对图 4–3中例子而言，假设节点 n_2 产生一条目的节点为 n_4 的消息，则其局部搜索过程如表 4–2所示。

表 4–2 图 4–3中的局部搜索过程示例

	current solution			next options		
	$\mathbf{x}^{now}$	N	$P_{N,4}$	$\mathbf{x}^{next}$	N'	$P_{N',4}$
Step 1	$[0,1,0]$	$\{n_2\}$	0.673	$[0,1,\underline{1}]$ $[0,\underline{0},0]$ $[\underline{1},1,0]$	$\{n_2,n_3\}$ ϕ $\{n_1,n_2\}$	0.600 0 0.670
Stop	$[0,1,0]$	$\{n_2\}$	0.673	—	—	—

如图 4–2，在开始时只有节点 n_2 持有该消息，对于 $N = \{n_2\}$，其预测投递率为 0.673，高于其周围 i 任何一个可行解所对应的目标值。算法会误认为当前解 $\boldsymbol{x}^{now}$ 即是最优解，然而真正的最优解是 $[1,1,1]$，对应预测投递率 0.789。在本例中，算法掉入了局部最优陷阱，而局部搜索算法无法跳出该陷阱。

4.3.3　基于禁忌搜索的求解方法

本小节基于禁忌搜索，提出解决 N_{opt} 的启发式算法，框架见算法 1所示。

禁忌搜索过程可被分为两步，即 Step 1 和 Step 2，其基本规则和数据结构见算法 1内的表格。其中数学符号的正式定义将在下文给出。整个搜索过程即是在解空间

Algorithm 1 Tabu Search Framework.

Require:

notation	**meaning**
p	Evaluation function
$\mathcal{N}$	Neighborhood of a solution
$\mathcal{S}$	Solution space
$\mathcal{E}$	Stop rules
$\mathcal{T}$	tabu table $\left\{\begin{array}{c\|c} \varphi & \text{Tabu element} \\ \hline \mathcal{L} & \text{Tabu length} \end{array}\right.$
$\mathcal{A}$	Aspiration criteria

Step 1:

choose an initial feasible solution $\mathbf{x}^{now}$ in $\mathcal{S}$
set the current best solution $\mathbf{x}^{best} \leftarrow \mathbf{x}^{now}$
set the tabu table $\mathcal{T} \leftarrow \emptyset$

Step 2:

if $\mathcal{E}(\mathbf{x}^{best}) = $ **true then**
    return $\mathbf{x}^{best}$
end if
get C by filtering $\mathcal{N}(\mathbf{x}^{now})$ according to $\mathcal{T}$ and $\mathcal{A}$
$\mathbf{x}^{now} \leftarrow \arg\max_{\mathbf{x} \in C} p(\mathbf{x})$
if $p(\mathbf{x}^{now}) < p(\mathbf{x}^{best})$ **then**
    $\mathbf{x}^{best} \leftarrow \mathbf{x}^{now}$
end if
update $\mathcal{T}$
goto Step 2

中一步一步更改当前可行解，同时评估该可行解对应的目标函数值，从而尝试获得最优解。Step 1 是禁忌搜索的初始化步骤，其中算法将在解空间 S 中选择一个可行解作为当前解，记为 $\boldsymbol{x}^{now}$。接着，最优解也被初始化为 $\boldsymbol{x}^{now}$，在此时，禁忌表 $\mathcal{T}$ 中并无记录，如 Step 1 中行 3 所示。

在 Step 2 中，若停止规则 $\mathcal{E}$ 被满足，则搜索过程终止并返回当前最优解 $\boldsymbol{x}^{best}$ 作为最终结果。若不满足停止规则，则从 $\boldsymbol{x}^{now}$ 的所有邻居 $\mathcal{N}(\boldsymbol{x}^{now})$ 中选择一个可行解更新 $\boldsymbol{x}^{now}$，该选择过程参照禁忌表 $\mathcal{T}$ 的状态，以及预先定义的特赦准则 $\mathcal{A}$。接着，算法将搜索子集 C，并选择能够使函数 p 取得最大值的 $\boldsymbol{x}$ 解向量。在 Step 2 的每一轮迭代中，都会尝试更新最优解 $\boldsymbol{x}^{best}$。最终，禁忌表 $\mathcal{T}$ 中对应的禁忌元素 ϕ 将会被更新为预先定义的禁忌步长 $\mathcal{L}(\phi)$。

下面给出算法1中所有数学符号的定义。

定义 6. 评估函数.
禁忌搜索中的评估函数表示为 $p(\boldsymbol{x})$, 有

$$p(\boldsymbol{x}) = -f(\boldsymbol{x})$$

在定义 6中，评估函数设为对定义 5中函数 $f(\boldsymbol{x})$ 的值取反。于是，搜索算法的目标即最大化评估函数的值。对于任一可行解向量 $\boldsymbol{x}$，规定其邻域如定义 7

定义 7. 邻域.
令 $\boldsymbol{x} = [x_1, x_2, \ldots, x_n](\forall i, x_i \in 0, 1)$ 为任一可行解向量, 则在算法中 $\boldsymbol{x}$ 的邻域 $\mathcal{N}(\boldsymbol{x})$ 定义如下

$$\mathcal{N}(\boldsymbol{x}) = \left\{ \boldsymbol{x'} \;\middle|\; \sum_{i=1}^{n} \left| x_i - x'_j \right| \leq 1 \right\} \tag{4–13}$$

如定义 7，某可行解的领域中的任一元素，与该可行解在向量上只有一个元素的差距。例如，如果当前可行解 $\boldsymbol{x}$ 为 $[0, 1, 0, 1]$，则其邻域

$$\mathcal{N}(\mathbf{x}) = \{[\underline{1}, 1, 0, 1], [0, \underline{0}, 0, 1], [0, 1, \underline{1}, 1], [0, 1, 0, \underline{0}]\}$$

其中每一个向量都是 $\boldsymbol{x}$ 的相邻解向量，下划线位置即是对原向量对应位置“取反”得到。事实上，整个解空间可被看做一个超立方体，其中每个顶点对应一个可行解向量，且解的邻域对应其在超立方体中的所有邻居节点，如图 4–4所示。

停止规则是禁忌搜索中最重要的准则之一，其决定搜索过程停止的条件。在本算法中，停止规则如定义 **??**所示。

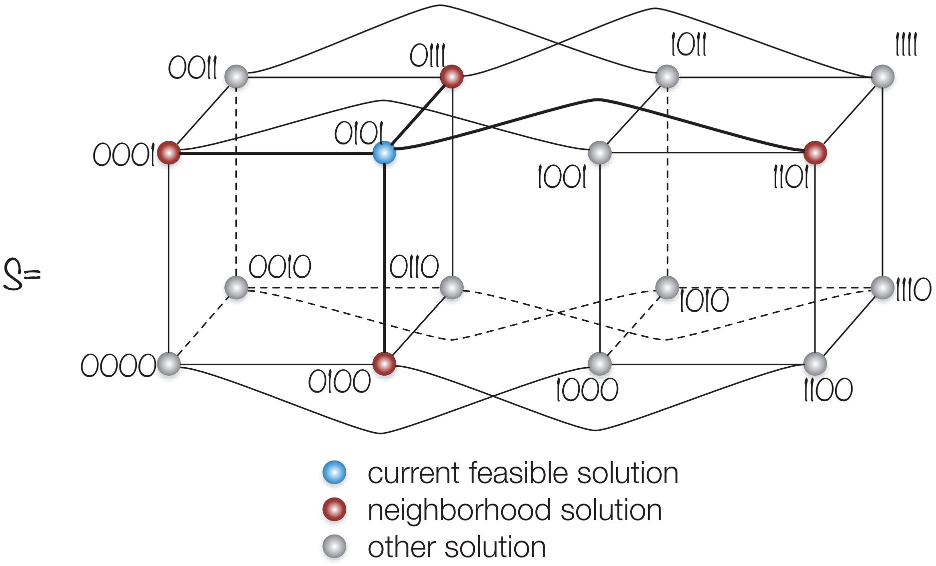


图 4–4 超立方体解空间

定义 8. 停止规则

停止规则 $\mathcal{E}$ 如下定义

$$\mathcal{E}(\boldsymbol{x}^{best}) = \begin{cases} true & \text{if } p(\mathbf{x}^{best}) \text{ not improved after } \theta \text{ steps} \\ false & \text{otherwise} \end{cases}$$

如之前所述，评估函数 p 的值在搜索过程的每一次迭代中都被更新。如果当前解向量 $\boldsymbol{x}^{now}$ 对应更好的评估函数值，即若有 $p(\boldsymbol{x}^{now}) > p(\boldsymbol{x}^{best})$，则用 $\boldsymbol{x}^{now}$ 的值更新 $\boldsymbol{x}^{best}$。然而，若当前记录的最优解对应的评估函数值 $p(\boldsymbol{x}^{best})$ 在一定步数之内都未被更新，则意味着算法可能无法找到更好的解（在算法本身不被修改的前提下）。在这种情况下，搜索过程停止，当前最优解向量将作为返回值从算法返回。

禁忌表 $\mathcal{T}$ 由两个主要元素组成，禁忌对象 ϕ 和禁忌长度 $\mathcal{L}(\phi)$。禁忌对象是禁忌搜索算法中实施禁忌的目标元素，可以定义为解向量的变化，或评估函数的变化。在本算法中，禁忌对象见定义 9。

定义 9. 禁忌对象

禁忌对象为解向量的变化，形式化定义如下

$$\varphi : \boldsymbol{x} \rightarrow \boldsymbol{x}'$$

有

$$\boldsymbol{x} = (x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n) \;\; (i \in [0, n])$$

且

$$\boldsymbol{x'} = (x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)\ \ (i \in [0, n], x_i \neq y_i)$$

事实上，禁忌对象是当前解向量在超立方体中对应的顶点向邻接顶点的迁移。正如图 4–4所示，从蓝色顶点（当前解）向红色顶点的迁移，即为禁忌对象。禁忌表中的禁忌长度如定义 10。

定义 10. 禁忌长度

禁忌长度由 $\mathcal{L}(\varphi)$ 表示, 有

$$\mathcal{L}(\varphi) = \lfloor T \rfloor$$

T 是服从正态分布的随机变量，有 $T \sim N(\mu, \sigma^2)$, 且令

$$\mu = \begin{cases} \sqrt{n}[1 + p(\boldsymbol{x'}) - p(\boldsymbol{x})] & p(\boldsymbol{x'}) > p(\boldsymbol{x}) \\ \sqrt{n} & \text{otherwise} \end{cases}$$

在定义 9中，禁忌长度设为对服从正态分布的随机变量 T 做下取整，其中正态分布的参数 μ 保证了对于能让目标函数值变的更好的解向量，在统计意义上具有一个较长的禁忌长度。将禁忌长度设为一个随机数，而非常量，其意义在于向搜索算法中加上一定的随机因素，如同舍伍德算法一样。随机性能够帮助算法跳出一些无休止的循环，且能在算法输入被蓄意设定的情况下表现的更好。对于禁忌长度，值 T 与评估函数的变化程度 $|p(\boldsymbol{x}) - p(\boldsymbol{x}')|$ 具有一定关系。评估函数值发生变化具有两种情况；第一种情况下 $p(\boldsymbol{x}') < p(\boldsymbol{x})$，意味着评估函数值到达了一个新的“山谷”；第二种情况下 $p(\boldsymbol{x}') > p(\boldsymbol{x})$，意味着评估函数值爬上了一个比以往更高的“峰顶”。在第一种情况下，应将禁忌长度设的更大，从而让算法能够跳出当前的局部陷阱；对于第二种情况，应将禁忌长度设的更小，以免解向量移动的太远，再次掉入另一个“山谷”。定义 9中的设定正好符合以上几点，此外对于正态分布参数 μ，有 $\mu \in [\sqrt{n}, 2\sqrt{n}]$。

从定义 9和定义 10出发，算法禁忌表见定义 11。

定义 11. 禁忌表

禁忌表的结构如下

$\mathcal{T}=$

1	2	$\cdots$	$\cdots$	$n-1$	n
t_1	t_2	$\cdots$	$\cdots$	t_{n-1}	t_n

假设禁忌对象为 $\varphi: \boldsymbol{x}->\boldsymbol{x}'$, 其更新规则如下

$$\forall i, t_i = \begin{cases} \mathcal{L}(\varphi) & \text{if } |x_i - x_i'| = 1 \\ 0 & \text{if } t_i = 0 \\ t_i - 1 & \text{otherwise} \end{cases}$$

禁忌表 $\mathcal{T}$ 设为一个 $2 \times n$ 的表，其中第一行代表禁忌对象中解向量的变化位置，第二行代表对于每个位置当前的禁忌步长。例如，设当前解向量发生变化，其对应的禁忌对象为 $\phi: [0,1,0] \to [1,1,0]$，则应当按如下更新当前的禁忌表（其中 $t_1, t_2, t_3 > 0$）

$$\begin{array}{|c|c|c|} \hline 1 & 2 & 3 \\ \hline t_1 & t_2 & t_3 \\ \hline \end{array} \longrightarrow \begin{array}{|c|c|c|} \hline 1 & 2 & 3 \\ \hline \mathcal{L}(\varphi) & t_2 - 1 & t_3 - 1 \\ \hline \end{array}$$

对于算法的每一次更新操作，向量变化所对应的位置，在禁忌表中都会以 $\mathcal{L}(\phi)$ 值更新，其它所有非零的表格元素都会做减 1 操作。任意位置，若其对应的禁忌步长不为 0，则仅当禁忌步长降为 0 时，当前解向量才可从该位置向对应的相邻解向量迁移。

禁忌搜索中另一个重要的法则，即为特赦准则。特赦准则可以对禁忌表中特定的禁忌对象进行赦免，从而使其能被算法重新选取。特赦准则如定义 12

定义 12. 特赦准则
特赦准则如下定义

对于 $\forall \boldsymbol{x}$, 若有 $p(\boldsymbol{x}) > p(\boldsymbol{x}^{best})$
则解向量 $\boldsymbol{x}$ 可以作为下一步选择，即使其在禁忌表 $\mathcal{T}$ 中被禁

当所有的位置都在表 $\mathcal{T}$ 中被禁时，当前解向量无法迁移到任何一个邻居向量。然而当某一个邻居向量所对应的评估函数值，优于当前所记录的最优值，则该向量会以特赦准则被解禁，即可以被算法选取。

图 4–3所示例子所对应的禁忌搜索过程，如图 4–3。如同表 4–2所示的局部搜索一样，消息源节点依然被选为 n_2，且初始解向量为 $[0,1,0]$。为了简化表述，在此，禁忌表长 $\mathcal{L}$ 设定为常数 3，而非定义中所述的随机量。此外，定义 8中的 θ 值设为 3.

在 Step 1 中，禁忌表 $\mathcal{T}$ 初始化为空，对于当前解向量，其所有的邻居解向量都可选。依算法1第 5 行，选择能使评估函数取最大值的解向量 $[1,1,0]$。在 Step 2 中，解向量第一个位置在表中被禁，故下一个解向量只能在 $[0,0,0]$ 及 $[1,1,1]$ 中选择。搜索过程一直进行到 Step 5，其中最优解在 $\theta = 3$ 步之后还未更新，触发停止规则，算法终止。纵览整个过程，与表 4–2所述局部搜索过程相比，算法可以成功跳出局部最优陷阱。

表 4–3 图 4–3例子对应的禁忌搜索过程

	current solution			current best solution			tabu table $\mathcal{T}$	next options			
	$\mathbf{x}^{now}$	N	$P_{N,4}$	$\mathbf{x}^{best}$	N^{best}	$P_{N^{best},4}$		$\mathbf{x}^{next}$	N’	$P_{N',4}$	status
Step 1	$[0,1,0]$	$\{n_2\}$	0.673	$[0,1,0]$	$\{n_2\}$	0.673	$\begin{array}{\|c\|c\|c\|}\hline 1&2&3\\\hline 0&0&0\\\hline\end{array}$	$[\underline{1},1,0]$	$\{n_1,n_2\}$	0.670	choosable
								$[0,\underline{0},0]$	ϕ	0	choosable
								$[0,1,\underline{1}]$	$\{n_2,n_3\}$	0.600	choosable
Step 2	$[1,1,0]$	$\{n_1,n_2\}$	0.670	$[0,1,0]$	$\{n_2\}$	0.673	$\begin{array}{\|c\|c\|c\|}\hline 1&2&3\\\hline 3&0&0\\\hline\end{array}$	$[\underline{0},1,0]$	$\{n_2\}$	0.673	tabu
								$[0,\underline{0},0]$	ϕ	0	choosable
								$[1,1,\underline{1}]$	$\{n_1,n_2,n_3\}$	0.789	choosable
Step 3	$[1,1,1]$	$\{n_1,n_2,n_3\}$	0.789	$[1,1,1]$	$\{n_1,n_2,n_3\}$	0.789	$\begin{array}{\|c\|c\|c\|}\hline 1&2&3\\\hline 2&0&3\\\hline\end{array}$	$[\underline{0},1,1]$	$\{n_2,n_3\}$	0.600	tabu
								$[1,\underline{0},1]$	$\{n_1,n_3\}$	0.626	choosable
								$[1,1,\underline{0}]$	$\{n_1,n_2\}$	0.670	tabu
Step 4	$[1,0,1]$	$\{n_1,n_3\}$	0.626	$[1,1,1]$	$\{n_1,n_2,n_3\}$	0.789	$\begin{array}{\|c\|c\|c\|}\hline 1&2&3\\\hline 1&3&2\\\hline\end{array}$	$[\underline{0},0,1]$	$\{n_3\}$	0.291	tabu
								$[1,\underline{1},1]$	$\{n_1,n_2,n_3\}$	0.789	tabu
								$[1,0,\underline{0}]$	$\{n_1\}$	0.430	tabu
Step 5	$[1,0,1]$	$\{n_1,n_3\}$	0.626	$[1,1,1]$	$\{n_1,n_2,n_3\}$	0.789	$\begin{array}{\|c\|c\|c\|}\hline 1&2&3\\\hline 0&2&1\\\hline\end{array}$	$[\underline{0},0,1]$	$\{n_3\}$	0.291	choosable
								$[1,\underline{1},1]$	$\{n_1,n_2,n_3\}$	0.789	tabu
								$[1,0,\underline{0}]$	$\{n_1\}$	0.430	tabu
Stop	$[1,1,1]$	$\{n_1,n_2,n_3\}$	0.789	—	—	—	—	—	—	—	—

4.4　最优化路由算法

本节给出有关路由过程的细节。在本节中，提出两个路由算法，基于局部搜索的 Local-MPAR 和基于禁忌搜索的 Tabu-MPAR。在 Local-MPAR 中，对任一节点而言，其它节点的 λ 值仅当该节点与其接触时获取；在 Tabu-MPAR 中，假设每个节点都知道其它所有节点对应的 λ 值。

在本章中，算法过程的解释是针对单一消息而言的（该消息在网络中可以具有一份或者多份拷贝）。为了进一步解释路由算法，本章参照文献 [**?**] 所提出的传染病路由算法的概念——将消息看做传播的病毒。由此，节点间一次成功的消息传输（复制）可被看做一次感染。上述讨论中，不包括消息的投递过程，即假设当 n_a 感染 n_b 时，n_b 不为消息的目的节点。以此出发，可以将网络中的节点状态划分为三类

- **感染节点**
  感染节点持有一条或者几条 gai’xiao

4.4.1 Local-MPAR: 基于局部搜索的路由算法

4.4.2 Tabu-MPAR: 基于禁忌搜索的路由算法

4.5 仿真实验

4.5.1 自变量：消息生存时间

4.5.2 自变量：节点缓存

4.6 本章总结

第五章　基于消息传输收益的最优队列调度算法

第六章　基于跳数的启发式距离向量算法

第二部分

机会网数据分发协议研究

第七章　机会网数据分发概述

第八章　基于一跳邻居位置的消息分发协议

第九章　基于非对称喷洒与授权转发的消息分发协议

第三部分

蜂窝网能耗优化研究

第十章　蜂窝网能耗优化概述

第十一章　交织多址及连续多址的能耗最优化

第十二章　异构网负载耦合模型下基于范围分配的能耗最优化

全文总结

这里是全文总结内容。

参考文献

[1] Conti M, Giordano S. Mobile ad hoc networking: milestones, challenges, and new research directions[J]. Communications Magazine, IEEE, 2014, 52(1):85–96.

[2] Henderson T, Kotz D, Abyzov I. The changing usage of a mature campus-wide wireless network[J]. 2004:187–201. http://dl.acm.org/citation.cfm?id=1023739.

[3] Ibrahim M, Nain P, Carreras I. Analysis of relay protocols for throwbox-equipped dtns[J]. … in Mobile, 2009. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5291625.

[4] Jedari B, Xia F. A Survey on Routing and Data Dissemination in Opportunistic Mobile Social Networks[J]. arXiv.org, 2013. http://arxiv.org/abs/1311.0347.

表格索引

插图索引

List of Algorithms

攻读学位期间发表的学术论文目录

[1] Chen H, Chan C T. Acoustic cloaking in three dimensions using acoustic metamaterials[J]. Applied Physics Letters, 2007, 91:183518.

[2] Chen H, Wu B I, Zhang B, et al. Electromagnetic Wave Interactions with a Metamaterial Cloak[J]. Physical Review Letters, 2007, 99(6):63903.

致　谢

感谢所有测试和使用交大硕士学位论文 LaTeX 模板的同学!
感谢那位最先制作出博士学位论文 LaTeX 模板的交大物理系同学!
感谢 William Wang 同学对模板移植做出的巨大贡献!

学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：____________

日　期：______年____月____日

学位论文版权使用授权书

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

保　密□，在__________年解密后适用本授权书。

本学位论文属于

不保密□。

(请在以上方框内打“✓”)

学位论文作者签名：__________　　　　指导教师签名：__________

日　期：______年____月____日　　　　日　期：______年____月____日