

静态最大可解网络编码的构造

黄 健, 梁松涛

(复旦大学计算机科学与技术学院, 上海 200433)

摘 要: 单源最大可解线性网络编码可以提高网络容量, 但没有考虑边失效或点失效的情况。为此, 利用贪心算法, 基于启发式规则, 提出一种静态最大可解网络编码的构造方法。该方法在每次迭代时, 通过向构造的编码向量中加入已有的向量, 并删除使用过的边, 提高网络传输速率。实验结果证明, 该方法在网络容量方面可以获得较大的增益。

关键词: 静态网络编码; 格局; 最大流; 最大可解网络编码; 启发式规则

Construction of Static Maximum Decodable Network Coding

HUANG Jian, LIANG Song-tao

(School of Computer Science, Fudan University, Shanghai 200433, China)

【Abstract】 Maximum decodable linear network code can improve the network capacity, but there are problems in the situation of edge false or node false. Based on greedy algorithm and a heuristic rule, this paper proposes an algorithm to construct the static maximum network coding. In each iteration process, it adds an existing vector, and deletes the edge used. Simulation result shows that the proposed algorithm can obtain great gain in network capacity.

【Key words】 static network coding; configuration; maximum flow; maximum decodable network coding; heuristic rule

DOI: 10.3969/j.issn.1000-3428.2012.04.035

1 概述

在传统网络传输中, 网络节点扮演着转发器的角色, 只是对其收到的信息进行存储和转发。在网络编码中, 网络节点起着编码器的作用, 可以对收到的信息进行编码处理后, 再转发出去。

文献[1]得出, 对于网络中的任一非源节点, 其接收信息的最大传输速率不超过网络最大流速率。文献[2]则证明了在单源多播网络中, 所有接收节点的最大流的最小值就是这些节点能够同时达到的信息传输的最大速率, 并且线性网络编码可以达到该最大速率。文献[3]构造了单源最大可解线性网络编码, 使得每个接收节点都以它们各自的最大流速率接收并解码出全部或部分原始信息。以上研究都没有考虑边失效的情况。本文在边失效的情况下, 给出了静态最大可解网络编码的构造方法。

2 静态最大可解网络编码

2.1 静态网络编码

静态网络编码是特殊的网络编码, 可以在边失效的情况下实现最优传输。本文用映射 $\lambda: E \rightarrow \{0,1\}$ 来描述网络的传输状态, 并称之为格局。 $\max f^\lambda(t)$ 表示源节点 s 到非源节点 t 的最大流, $\max f^\lambda(T)$ 表示 s 到非源节点集合 T 的最大流。

定义 1^[4] 令 λ 为网络的一个格局, 对于一个 ω 维的线性网络编码, 边 e 的 λ -全局编码向量 f_e^λ 是一个通过由上而下的递推关系得到的 ω 维列向量, 并且满足:

$$(1) f_e^\lambda = \lambda(e) \sum_{d \in \text{In}(t)} k_{d,e} f_d^\lambda$$

$$(2) \omega \text{ 条 } \lambda\text{-全局编码向量在 } \lambda \text{ 上线性无关。}$$

本文用行向量 x 来表示由源节点 s 发出的信息。在 λ 格局下, 对于点 t , 定义矩阵 $V_t^\lambda = \{f_e^\lambda : e \in \text{In}(t)\}$ 为传输矩阵; 而对于节点集合 T , 有 $V_T^\lambda = \langle \cup_{t \in T} V_t^\lambda \rangle$ 。

2.2 最大可解的基本概念

对于节点集合 T , 若 $\omega \leq \max f^\lambda(T)$, 所有的接收节点都可以解码得出源节点信息^[4]; 反之, T 中的接收节点没有完全利用网络的容量。

定义 2 对于图 G 上 ω 维的线性网络编码, s 为源节点, t 为接收节点, λ 为格局, 若 $\omega > \max f^\lambda(t)$, 则对于任一接收节点 t , 如果满足下列条件: (1) $\text{rank}(V_t^\lambda) = \max f^\lambda(t)$; (2) V_t^λ 恰好有 $\max(0, \omega - \max f^\lambda(t))$ 行零向量, 那么该网络编码称为静态最大可解网络编码。

3 基于贪心算法的构造方法

3.1 判定问题

定理 对于图 G , s 为源节点, t 为接收节点, λ 为网络的一个格局, 若 $\omega > \max f^\lambda(t)$, 则判定是否存在一个 ω 维的静态线性网络编码是 NP-困难的。

证明: 文献[3,5]证明了线性网络编码下的同类问题为 NP-困难的。因为静态网络编码也是线性网络编码, 通过对网络编码的代数研究^[6-7], ω 维静态线性网络编码的存在问题也是 NP-困难的。

本文将该问题简化为以下问题: 在 λ 格局中, 定义 $\text{msg}^\lambda(t)$ 为接收节点 t 接收一次消息后解码出的消息数, 如何构造网络编码可使 $\sum_{t \in T} \text{msg}^\lambda(t)$ 尽可能地大。当 $\sum_{t \in T} \text{msg}^\lambda(t) =$

基金项目: 国家自然科学基金资助项目(60772131); 教育部 2007 年新世纪优秀人才支持计划基金资助项目(NCET-07-0198); 霍英东基金特别资助项目(114401); 2011 年上海市曙光计划基金资助项目(10SG01)

作者简介: 黄 健(1985-), 男, 硕士, 主研方向: 网络编码, 密码学; 梁松涛, 硕士

收稿日期: 2011-07-12 **E-mail:** 082024035@fudan.edu.cn

$\sum_{i \in T} \max f^i(t)$ 时, 即是最大可解网络编码。

3.2 构造方法

本文使用贪心算法迭代出静态最大可解网络编码。由于迭代过程涉及不相交路径的选择, 因此采用启发式规则^[3]。在 λ 格局下, 对于每条边 e , 定义权重为 $w(e)$ ^[8]。由于源节点 s 发送的信息为 ω 维, 因此可以认为 s 的发送速率为 ω , 并将其拆分成 ω 个发送速率为 1 的源节点 $s_1, s_2, \dots, s_\omega$, 其中, s_i 发送信息 x_i 。定义 T_i 为以 s_i 为根的最小生成树, 并且 $w(T_i) = \sum_{e \in T_i} w(e)$ 。在每次迭代中, 若 $w(T_i) \leq w(T_j)$, 则 s_i 排在 s_j 的前面, 再根据权值, 先选择前 $\max f^i(t)$ 个拆分后的源节点, 这样就降低了构造过程中的随机性。

3.3 构造步骤

参照文献[3], 本文提出如下构造方法:

- (1)对源节点 s 做上述拆分, 记为源节点集合 S 。
- (2)以 S 中的每个源节点 s_i 为根, 分别构造 T_i 。
- (3)初始化 $flow = \omega$, 并且每条边的编码向量 $f_e = 0$ 。
- (4)选择前 ω_1 个源节点, 将此集合记为 S' , 其中, $\omega_1 = \min(flow, \max f^i(t))$ 。
- (5)在网络中添加新源节点 s' , 并且 s' 连接集合 S' 中的所有节点。
- (6)对于 s' , 构造边编码向量为 f'_e 的 ω_1 维线性网络编码, 并将 f_e 的数值改为 f'_e 对应位置的数值。
- (7)删除(6)中已用过的边, 再重新计算每个接收节点的最大流, 并删除集合 S' 中的所有节点。
- (8)如果接收节点全被删除或 $flow=0$, 则算法结束; 否则, 转步骤(4)。

3.4 与文献[3]方法的比较

与文献[3]相比, 静态最大可解线性网络编码的构造考虑了边失效的情况。随着时间的推移, 失效的边会有变化, 这时就需要重新迭代。

3.5 时间复杂度分析

因为迭代次数有限, 且每次迭代过程都可以在多项式时间内完成, 所以本文方法是多项式时间的。

4 实例分析

4.1 例证

下面针对图 1 的示例进行分析。其中, 图 1(a)表示边 mc 失效的网络; 图 1(b)表示边 bc 失效的网络; 源节点为 s ; 接收节点为 x, z 。

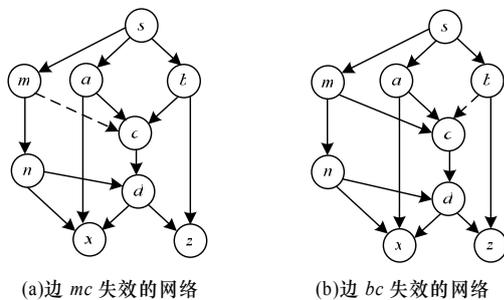


图 1 边失效网络示例

假设 mc 边失效, 即 $\lambda(mc) = 0$ 。根据图 1(a), $\max f^i(x) = 3$, $\max f^i(z) = 2$, $\omega_1 = 2$, 并初始化 $flow = 3$ 。

- (1)初始化每条边的编码向量为 $(0,0,0)^T$ 。
- (2)对源节点 s 进行拆分, 并利用启发式规则, 选择源为 s_{sa} 和 s_{sb} , 并修改编码向量如下:

$$f_{sa} = f_{ac} = f_{ax} = (1,0,0)^T$$

$$f_{sb} = f_{bc} = f_{bz} = (0,1,0)^T$$

(3)选取源为 s_{sm} , 进行第 2 次迭代, 得到编码向量:

$$f_{sm} = f_{mn} = f_{nx} = (0,0,1)^T$$

对于边 bc 失效的情况, 根据图 1(b), $\max f^i(x) = 2$, $\max f^i(z) = 2$, $\omega_1 = 2$, 并初始化 $flow = 3$ 。

按照构造方法, 选择源为 s_{sa} 和 s_{sb} , 此时, $\max f^i(x) = 1$, $\omega_1 > \max f^i(x)$ 。于是修改向量为:

$$f_{sa} = f_{ac} = f_{ax} = (1,0,0)^T$$

$$f_{sb} = f_{bz} = (0,1,0)^T$$

选取源为 s_{sm} , 进行第 2 次迭代, 得:

$$f_{sm} = f_{mc} = f_{cd} = f_{dx} = (0,0,1)^T$$

4.2 实验结果分析

首先设定以下参数, 以便进行定量分析: 极小最大流和极大最大流分别表示全部接收节点的最大流的最小值和最大值; 多播构造和近似构造则分别表示线性网络多播下和通过本文算法得出的 $\sum_{i \in T} \max f^i(t)$ 的值; 总最大流是 $\sum_{i \in T} \max f^i(t)$; 增益比 = (近似构造流 - 多播构造流) / 总最大流 $\times 100\%$ ^[3]。在此基础上用 C# 进行模拟实验。实验结果如表 1、表 2 所示(限于篇幅, 只给出部分结果)。表 1、表 2 表明本文的构造方法可以提高网络传输速率。其中, 表 1 是在随机生成的网络上进行实验的。可以看出, 增益比的数值不大, 而且差距也不是很大。表 2 则是在随机网络的基础上, 通过随机减少某些节点的最大流, 使得接收节点的最大流之间差异较大。通过比较表 1 与表 2 的实验结果可知, 本文算法在接收节点的最大流之间差异较大时, 能得到更大的增益。

表 1 完全随机网络下的实验结果

总节点数	接收节点	极小最大流	极大最大流	多播构造	近似构造	总最大流	增益比/(%)
100	5	39	44	195	211	215	7.44
100	15	31	37	465	501	594	6.06
500	10	231	246	2 310	2 391	2 495	3.25
500	50	181	227	9 050	9 779	11 099	6.57
1 000	10	455	508	4 550	4 697	4 915	2.99
1 000	50	433	468	21 650	22 015	23 008	2.02

表 2 有条件的随机网络下的实验结果

总节点数	接收节点	极小最大流	极大最大流	多播构造	近似构造	总最大流	增益比/(%)
100	5	15	46	75	145	181	38.70
100	15	14	49	210	243	421	7.84
500	10	72	245	720	1 090	1 611	23.00
500	50	80	237	4 000	4 315	8 066	3.91
1 000	10	164	497	1 640	2 362	3 336	21.60
1 000	50	162	477	8 100	8 515	16 085	2.58

对于静态网络编码, 本文引入了时间参数 t 。因为不同的时间下网络格局也不同, 1 h 后在同一个网络下进行实验, 得到的结果如表 3、表 4 所示。

表 3 1 h 后完全随机网络下的实验结果

总节点数	接收节点	极小最大流	极大最大流	多播构造	近似构造	总最大流	增益比/(%)
100	5	38	44	190	215	216	11.60
100	15	33	37	495	513	591	3.05
500	10	237	249	2 370	2 423	2 499	2.12
500	50	179	231	8 950	9 821	11 086	7.90
1 000	10	446	499	4 460	4 791	4 921	6.73
1 000	50	441	475	22 050	23 143	22 986	4.76

(下转第 115 页)