

基于遗传算法的网络编码优化*

邓亮, 赵进, 王新⁺

(复旦大学 计算机科学技术学院, 上海 200433)

Genetic Algorithm Solution of Network Coding Optimization

DENG Liang, ZHAO Jin, WANG Xin⁺

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

+ Corresponding author: E-mail: xinw@fudan.edu.cn

Deng L, Zhao J, Wang X. Genetic algorithm solution of network coding optimization. Journal of Software, 2009,20(8):2269–2279. <http://www.jos.org.cn/1000-9825/3370.htm>

Abstract: After the best optimizing approach of network coding is being studied, some methods are proposed based on the characteristics of the network coding overhead optimization problem. First, two modifications are added to the preprocessing phase: 1) How to generate a fitness value to a network coding scheme under a certain network coding optimization request is presented. This makes different network coding optimization problems be solved with the same genetic algorithm. 2) An additional exam processing of the multi-in outgoing links is imported to reduce the solution space. Second, experimental results show that the random generated solution of network coding optimization problem can hardly achieve the multicast rate, three new steps are suggested be taken with the common genetic algorithm: 1) use more delicate member generating function to generate initial members; 2) add new members at the beginning of each round of the genetic algorithm to avoid localized optimization; 3) assign a fitness value based on each receiver's data rate rather than -1 to those network coding solutions which cannot achieve the max multicast rate. Experimental results show dramatic improvements in terms of both efficiency and result.

Key words: multicast; network coding; optimization; heuristic; genetic algorithm

摘要: 在前人优化研究方法的基础上,结合网络编码优化问题自身的特点提出了新的解决方案.首先是算法的预处理部分:1) 给出了统一的方法由不同的资源描述函数生成遗传算法所必须的适应值函数,使得各种不同的网络编码资源优化问题都能利用同样的遗传算法模型;2) 通过检验有多条输入链路的输出链路进一步缩小优化算法的搜索范围.其次,针对网络编码资源优化问题随机解几乎不能让所有接收者都达到组播速率的特点,在一般的遗传算法中加入以下新的处理:1) 在初始化阶段使用更为精细的算法产生更高质量的初始成员.2) 在遗传算法每次循环开始时额外调用初始成员生成算法,加入一定数量的新成员,从而避免了局部性问题.3) 对于不能达到最大组播

* Supported by the National Natural Science Foundation of China under Grant No.60702054 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2006AA01Z203 (国家高技术研究发展计划(863)); the Shanghai Rising-Star Program of China under Grant No.08QA14009 (上海市科委启明星计划); the Shanghai Educational Development Foundation of China under Grant No.2007CG07 (上海市教育发展基金会)

Received 2007-12-10; Accepted 2008-04-30

速率的网络编码方案,基于各个接收者各自的接收速率确定更为合适的适应值而不是统一设为-1,从而使这些方案也能参与算法的进一步处理而不是完全被淘汰.模拟实验结果显示,新的优化算法不仅运行得更快,而且输出的网络编码方案所消耗的资源也更少.

关键词: 组播;网络编码;优化;启发式;遗传算法

中图法分类号: TP393 文献标识码: A

网络编码^[1]是在 2000 年提出的一种新网络传输方法.在传统网络传输策略中,网络中间节点仅仅对上游链路的信息简单转发或者复制转发,而网络编码允许网络中间节点对来自不同输入链路的信息进行混合编码,以此产生新的信息进行转发.当向多个接收者同时发送数据时,使用网络编码能够达到仅使用转发或 IP 组播所达不到的组播速率.图 1 是一个网络编码下组播的简单例子.

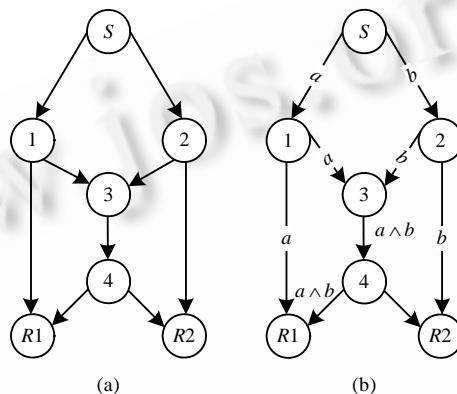


Fig.1 Network coding can make full use of the network multicast capacity

图 1 理论组播速率通过网络编码技术能够得到充分的利用

在如图 1(a)所示的网络拓扑下,所有链路带宽均相同,假定为 1. S 需要同时发送数据 a 和 b 到 $R1$ 和 $R2$,由于节点 3 与节点 4 之间的链路是瓶颈,无论是单纯的转发还是 IP 组播都无法同时将 a 和 b 发送给两个接收者.如图 1(b)所示,采用网络编码,在节点 3 将收到的 a 和 b 编码后发送, $R1$ 和 $R2$ 都可以通过解码,同时恢复出 a 和 b .

网络中,组播速率的理论最大值是所有接收者与发送者间最小割的最小值^[1].使用简单转发或者复制,不一定能达到这一最大速率.网络编码理论方面的研究已经证明,使用网络编码可以无限接近这种理论最大速率^[1],而线性网络编码在单源组播的情况下可以确实地达到这一理论最大速率^[2].

利用网络编码分发数据,需要为整个网络制订编码方案.目前常见的网络编码方案总的来说分为两类:确定性编码和非确定性编码.确定性编码需要人为预先指定参与编码的节点链路以及编码所需要的各种参数的值,非确定性编码则是通过运行特定的算法为网络制订编码方案以及相关参数的值.预先人为指定编码方案的方法只能适应网络规模不大、节点数量不多并且拓扑固定的情况,在简单的研究实验中往往采用这种方法.非确定性编码是网络编码研究的热点,在应用中,通常拓扑的规模比较大,拓扑的变化也不可避免,某些应用甚至要处理频繁变化的拓扑.目前,广泛应用的非确定性编码方案是一种基于随机线性网络编码^[3],在所有中间节点都进行网络编码的策略.这种策略的优点是每条链路可以单独地决定各自的编码方案,具有天然的分布式性质,而且在单源组播的场合下必然可以充分利用网络的最大组播速率^[3].其不足之处是在字母表大小有限的情况下会有极小的概率解码失败^[4]以及进行了冗余的网络编码操作.所谓冗余的网络编码是指在网络中,有一部分链路即使不进行网络编码,也可以使组播速率达到理论最大值,即并不是所有的中间节点都需要进行网络编码.因为网络编码是将不同来源的数据进行某种混合后传输,所以只有存在多条输入链路的节点才有可能需要对来自不同链路的信息进行编码之后发送.已经被广泛采用的一种简单的优化方法是只在有多条输入链路的节点进行编码.比起在所有的节点都进行编码,这种方法可以减小一部分冗余编码的开销.如图 1 中,只有节点 3 有多

条输入链路,这种情况下,节点3正好是为了达到最大组播速率必须要进行编码的节点.但是一般情况下,在所有具有多条输入链路的节点都进行编码同样会产生冗余的网络编码.其中一个原因是一个节点往往连接了多条输出链路,网络编码的主体应该是输出链路而不是节点;另一个原因是,并不是所有具有多输入链路的节点都需要网络编码.如图2(a)所示的网络:

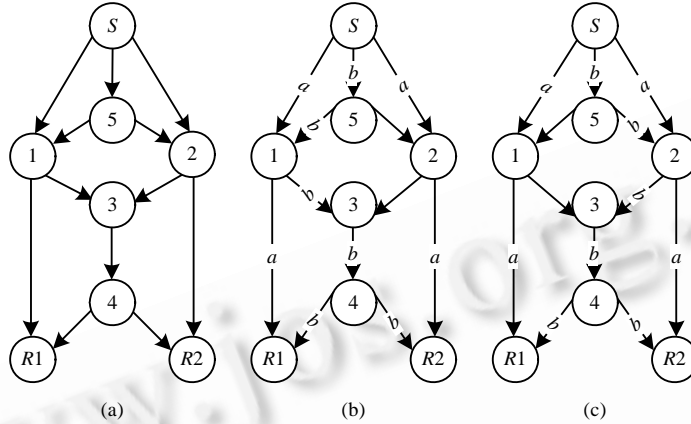


Fig.2 Some multi-in vertices do not need network coding

图2 存在不需要编码的多输入节点

节点3虽然有两条输入链路,但是如图2(b)和图2(c)所示的传输方法,节点3不必采用网络编码进行数据发送同样可以使两个接收者同时收到a和b.同样地,节点1和节点2有多条输入链路和多条输出链路,但也不必采用网络编码.

尽管随机线性网络编码方案具有完全分布式和实现简单的优点,但是在这一方案下,拓扑中所有的链路都要进行传输和网络编码.在实际应用中,如图2(b)所示,为了达到理论的组播速率,往往既不需要在所有的链路都进行网络编码也不需要使用所有的链路.因此,在利用网络编码带来的好处的同时,如何尽可能地减少各种开销就成为一个非常有意义的问题,称为网络编码优化问题.

网络编码优化问题是指,在给定的网络拓扑结构上,基于某个优化要求,在保证网络组播速率达到理论最大值的前提下,尽可能地只在必须的链路上网络编码.关于网络编码优化问题,已经有相关的工作取得了一些成果.Wu等人^[5]从理论上统一了Edmond的理论和Ahlswede等人的理论,并证明:只需在斯坦纳(Steiner)边上进行编码就可以达到理论上的最大组播速率.Cannons和Zeger^[6]研究了当只有一部分节点能够执行网络编码操作时的最大组播速率.Langberg等人^[7]证明了网络编码方案的存在与否与拓扑的大小无关并分别得出有环拓扑和无环拓扑的编码节点数量下限.Bhattad等人^[8]提出了新的网络组播模型,并提出了理论上能够解决网络编码优化问题的近似算法.但是,这个算法在文献[9]中被证明过于复杂而没有实际应用的可能.Kim等人^[9]证明了网络编码资源优化问题是NP难问题,并首先提出了应用遗传算法解决网络编码优化问题,然后又对算法作了部分改进并提出分布式算法^[10].该分布式算法基于文献[9]中的遗传算法,将优化由各个节点各自执行并增加了信息交换的部分,对于遗传算法的改进可以很简单地加入到分布式算法中.

对于NP难问题,目前认为较为有效的解决方法是启发式解法和逻辑的近似算法.在启发式算法与逻辑近似算法的对比中,遗传算法在解决特别复杂且研究不足的问题上有明显的优势^[11].经过对文献[9,10]所提出方法的研究和实践,我们发现,按照文献[9,10]所提出的方法,直接应用普通的遗传算法解决网络编码优化问题的性能仍然不令人满意.虽然采用了启发式算法,但是随着问题规模的增大,遗传算法单次循环需要的时间急剧增加,算法的运行时间也增加到惊人的程度^[10],得到的解的质量也随之大大幅度下降,并且这种方法并没有考虑尽可能地减少不必要的链路使用.针对以上不足,本文在算法中增加了对非编码链路的使用优化并且主要在以下两个方面提出了新的解决方案:首先是数据处理部分:1)拓展了遗传算法的使用范围,给出了由任意资源描述函

数产生遗传算法所必需的适应值函数的方法.2) 加入对于拥有多条输入链路的输出链路的预处理以收缩可能的网络编码方案的范围.其次,针对网络编码资源优化问题的随机解很难同时满足所有接受者的组播速率的特点在遗传算法中加入新的处理步骤:1) 细化初始化函数,采用 3 种不同的方案按比例生成遗传算法的初始成员.2) 每次循环伊始,额外调用初始化函数,加入一部分新成员以避免局部性.3) 给大多数不能让所有接收者都达到最大组播速率的网络编码方案赋予一个非负的适应值而不是统一指定适应值为-1.模拟实验结果表明,上述新加入的处理可以明显提高算法的效率和解的质量.

本文第 1 节介绍优化方法中数据处理部分和遗传算法的新方案.第 2 节给出模拟实验的详情和对结果的分析.第 3 节回顾本文的主要内容.

1 优化方法描述

网络编码优化所要解决的问题是,在组播网络中,基于某个资源优化要求,在保证网络组播速率达到理论最大值的前提下,尽可能地只保留必须的网络编码链路,寻求一种消耗网络资源最少的网络编码方案.文献[9]中已经证明了网络编码优化问题是 NP 难问题,并提出了基于一般遗传算法的解决方案.文献[9,10]所采用的遗传算法解决方案(genetic algorithm scheme,简称 GAS)由数据处理和遗传算法两部分构成.数据处理部分包括网络编码方案适应值(fitness value)计算和网络编码方案(network coding solution)下拓扑组播的速率验证;遗传算法主要通过模拟自然界进化现象对候选方案进行筛选,输出符合要求的编码方案,所需要的数据服务由数据处理部分完成.在一般的 NP 难问题中,随机候选方案是能够符合问题的要求的,只是由于相对于最佳方案所付出的代价较大,需要进一步优化,如旅行商(TSP)问题.而网络编码优化问题的随机解只有非常小的概率能够满足所有接收者组播速率的要求.因此现有的遗传算法解决方案(GAS)运行速度很慢,解的质量不高.另外,随着拓扑的增大,虽然 GAS 的循环次数不超过常数 N ,但是每次循环所需要的时间却急剧增大.当拓扑规模增大到大约 40 个节点时,每次循环就需要约 65.6 秒^[10].这样,为 40 节点的拓扑产生一个优化结果所需要的时间就常常高达几个小时,这在实际应用中是远远不够的.我们所提出的解决方案(improved genetic algorithm scheme 简称 IGAS)主要通过是在 GAS 各环节中加入各种针对网络编码优化问题的解决方案而来,其中特定方案下组播速率验证我们采用了与文献[10]相同的方法.下面我们将分两部分介绍数据处理和遗传算法中的新的解决方案.图 3 给出了 IGAS 的整体流程.经过一些简单的预处理(拓扑测量、最大组播速率计算、网络编码相关数据结构的生成),确认有必要进行网络编码方案的优化之后,将数据交由遗传算法进行处理并输出优化的方案.

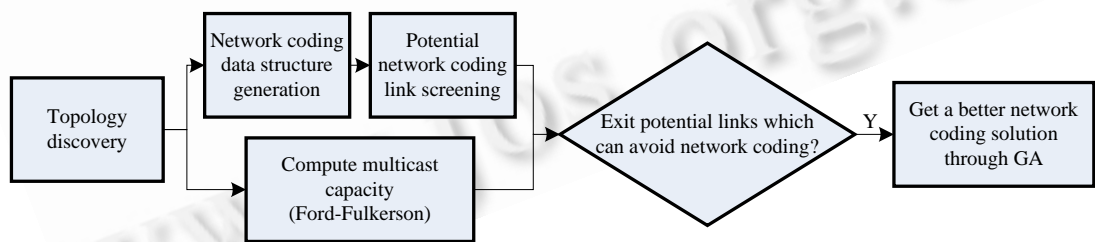


Fig.3 Flow chart of IGA

图 3 IGA 整体流程图

1.1 数据处理部分

在文献[10]中,GAS 优化的目标是 minimized 具有编码功能的链路数量,所给出的适应值函数也是基于这一要求.本文将优化的具体要求抽象为资源描述函数,并给出利用资源描述函数和网络编码方案产生适应值函数的一般方法.适应值是描述遗传算法中筛选对象优秀程度的值,越优秀的方案所对应的适应值越大,反之亦然.因此,首先为不能保证所有接收者都达到预定组播速率的方案赋予所有方案中最小的适应值.而对于能够保证所有接收者组播速率的方案,通过衡量它们所消耗资源的多少赋予恰当的适应值.对拓扑中的每个顶点(vertex)进

行编号,记第 v 个顶点为 $vertex(v)$;对输入这个顶点的所有链路进行编号,第 i 条输入链路记为 $vertex(v)_{lin}(i)$;对这个顶点的所有输出链路进行编号,第 j 条输出链路记为 $vertex(v)_{lout}(j)$;对顶点 v 使用一个系数矩阵表示在这一顶点的编码方案,在这一矩阵内所有元素的取值范围均为 $\{0,1\}$,位于 i 行 j 列的元素记作 $coe(v)(i)(j)$.三元数组 $coe(v)(i)(j)$ 称为网络编码方案数据结构,表示了整个拓扑的网络编码方案.对于某个节点 v ,第 j 个列向量表示所有输入链路在第 j 条输出链路上的混合情况,如果第 j 列所有元素均为 0,则输出链路 $vertex(v)_{lout}(j)$ 上没有任何信息被传送;如果该列除了一个元素为 1 以外,其他所有元素均为 0,则 $vertex(v)_{lout}(j)$ 仅仅转发 1 条输入链路的信息;如果该列有两个以上的元素为 1,则 $vertex(v)_{lout}(j)$ 上传送的是以上对应非 0 输入链路上的信息编码后的结果,即转发混合后的信息.根据某个特定的编码方案,按照资源优化的目标所给出的资源描述函数,可以获得一个编码方案的资源消耗总和.比如,发生编码的节点总数目(在应用中对应于具有编码功能的特殊节点的数量)或者网络编码计算的次数(在应用中对应编码的计算时间和消耗的能量)等,算法的目标是选择令这个值尽可能小的编码方案.编码的方案越优秀,资源消耗度量总和就越小.因此,需要将按照优秀程度递减的资源消耗总和函数转换为按优秀程度递增的适应值函数.解决的方法是简单地用一个足够大的常量减去资源消耗总和来产生需要的适应值函数.在应用中,只需保证这个常量大于最坏编码方案(即在所有可能的链路上都进行网络编码)的总资源消耗量,本文的实验中取这个常量为 65 535.

在文献[9,10]中,对于给定的拓扑,算法将所有拥有多条输入链路的节点都纳入网络编码优化的甄选范围,同时默认所有的单输入链路的输出链路均转发该输入链路的信息.这样可以简化算法的设计,因为只有拥有多条输入链路的输出链路才有可能需要进行网络编码,而设定其他链路都进行转发就可以把网络编码优化中不确定的因素尽可能地降低.不过,拥有多条输入链路只是需要进行网络编码的必要条件而不是充分条件,而设定所有其他链路都无条件进行转发无疑会多余地使用一些不必要使用的链路.针对这两点,我们相应地提出了新的解决方法.如前所述,网络编码的主体应该是链路而不是节点,称拥有多条输入链路的输出链路为候选链路.我们在算法中加入一段预处理,以检测某条候选链路是否可以在网络其他部分都进行网络编码的情况下单独进行不编码的传输(即传统的数据转发),对所有的 v 和 j ,依次考察候选链路 $vertex(v)_{lout}(j)$,令所有 $coe(z)(x)(y)=1(z \neq v)$;而当 $y=j$ 且 $z=v$,对所有 x , $coe(v)(x)(j)$ 只有一个取 1,其他元素均为 0,即该链路在网络其他部分都进行网络编码的情况下尝试只转发某条输入链路,在这种情况下进行组播速率检查.如果某种情况下所有接收者的组播速率能够得到满足,就认为该候选链路能够单独进行非网络编码的传输,从而纳入网络编码优化的甄选范围.极端情况下,如果所有的候选链路都被证明无法避免网络编码(即任意候选链路单独地限制为只能转发数据都会导致网络组播速率的下降),就没有任何进行网络编码优化的必要,在所有的候选链路都进行网络编码即为最优的方案.如图 1(a)所示的拓扑,在预处理阶段就可以发现整个网络中不存在可以不采用网络编码的候选链路,从而无须进行任何优化.由于整个网络编码方案解空间大小与甄选的候选链路数目成指数关系,即使在预处理阶段仅仅发现了一条候选链路不必考虑进行网络编码,也可以极大地减少可能的网络编码方案的数目,从而加快优化算法收敛的速度.预处理消耗的时间与遗传算法整体运行的时间相比微不足道.另外,我们同样加入了对于确定不必网络编码链路的优化(以算法的运行时间为代价),尽可能地发现能够不被使用的链路.如图 2(b)和图 2(c)所示,有些链路不必传送数据也可以达到最大组播速率.这一点在实际系统中具有重要意义.

由于遗传算法所生成编码方案的随机性,可能在优化过程中出现这样的方案,即一条输出链路可能将一条没有传输任何数据的输入链路纳入信息的网络编码过程.虽然这在实际中并不消耗多余的资源,但是在使用资源描述函数计算资源消耗量时,这样的混合会使资源描述函数计算的资源总和增大,从而影响算法的结果.因此,在遗传算法的每次循环中引入检查算法,对编码方案进行处理,清除这种不必要的信息混合.

1.2 遗传算法部分

为了便于对比介绍在遗传算法中引入的新的解决方案,首先我们来介绍一般遗传算法的流程,如图 4 所示.

最初,遗传算法调用新成员随机生成函数生成一定数量的初始成员,每个成员都是由二进制向量表示的网络编码方案,这些向量在遗传算法中称为基因.通过适应值函数的计算,每个成员的基因都有一个对应的适应值.基于适应值,可以进行基因的遗传、变异和成员之间的生存竞争.在遗传算法中,根据适应值,所有成员都有对

应的大小不同的被选择的概率.在基因遗传的环节,基于这一概率,选择一定数量的成员,通过两两之间交换基因的片段,生成一部分新成员,成员总数随之增加.在基因变异环节,遍历当前所有的成员基因的每个二进制位,每个位都有极小的概率被翻转,从而产生新成员.在主循环的最后,根据适应值,所有成员进行生存竞争,淘汰适应值较低的成员,将成员总数减少到预定的数值.遗传算法通常在满足一定的循环数(N)或者在一定的循环数($n(n < N)$)没有对最优解作出任何改进之后停止并输出当前的最佳方案.

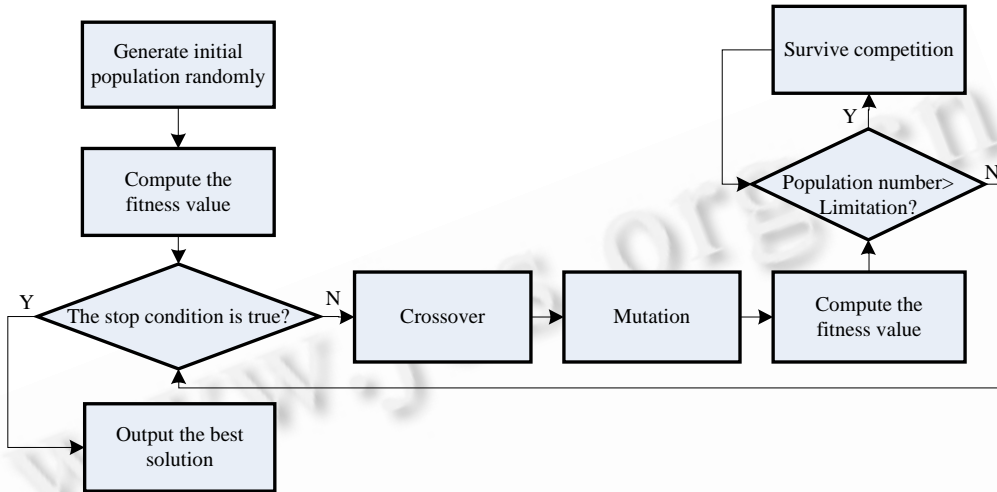


Fig.4 Flow chart of genetic algorithm

图4 一般遗传算法的流程图

网络编码资源优化问题存在着与一般的适合使用遗传算法解决的问题不同的特性,即随机解满足问题要求的可能性非常小.直接将一般的遗传算法应用到这一新问题上,其性能并不令人满意.比如,随着问题规模的增大,算法的运行时间剧增,解的质量降低很快,局部性^[11]的问题仍然比较严重.接下来将结合网络编码资源优化问题的特点介绍本文引入的新的解决方案.图5在一般性的遗传算法流程图的基础上,在各个环节上加入了新的解决方案,下面将逐一加以介绍.

在反复的实验中,我们注意到,初始成员的质量对整个算法的收敛速度有非常大的影响.当完全采用随机生成的初始成员时,在初始生成的网络编码方案质量较高的情况下,遗传算法的收敛速度比较快;而当随机生成的初始成员质量较差时,遗传算法往往在很长时间内都无法得到任何能够让所有接收者达到最大组播速率的网络编码方案.在文献[9]中这一现象已被注意到,其对策是加入一个在所有的中间节点都进行编码的初始成员.这一措施仅能保证初期的搜索不至于无法发现任何可用的编码方案,但对于算法的性能却并没有多少提高.为了尽可能的在最初阶段发现更优秀的编码方案,在尝试了多种网络编码生成方案之后,IGAS最终采用了3种方法共同生成初始成员.首先,与文献[9]一样,我们加入一个在所有的中间节点都进行编码的初始成员,以保证即使在最坏情况下也能输出可用的网络编码方案.第2种生成初始成员的方法是,每条候选链路都随机选择一条输入链路转发该链路上的信息.这种方法的根据是我们在随机拓扑上的多次实验,实验结果表明,在大量的拓扑中,为了达到理论的最大组播速率,只有少量的链路需要进行网络编码,其他大量的链路都是仅仅转发某条链路的信息.我们采用的第3种方法是完全随机生成网络编码方案.在不断的尝试之后,我们用第2种和第3种方案各生成50%的初始成员.实验结果显示,使用这种更为精细的初始成员生成方案可以在初始化阶段就发现一部分比在所有中间节点都进行编码更为优秀的网络编码方案(详见实验结果部分,图7中A点和B点的对比),可以大幅度减少整个算法的运行时间.

网络编码资源优化问题的特殊性在于,由遗传算法生成的编码方案由于本质上带有随机性,绝大部分都是无法让所有的接收者都达到最大的组播速率的.按照文献[9,10]的适应值计算方法,那些无法让所有接收者都达

到最大组播速率的方案(invalid network coding solution,简称 INCS)由于适应值为负,在当前循环被淘汰的几率接近 100%.在算法运行的初期,如果不采用精细的初始成员生成策略,则多数初始成员都将在短时间内被淘汰,整个算法在此期间就会在一定程度上退化到随机方案的筛选算法.即使采用精细的初始成员生成策略,也会由于基因遗传和变异,新生成的成员大多难免被淘汰而导致整个算法收敛速度很慢.鉴于此,我们希望更加细致的鉴别而不是统一舍弃所有的 INCS.在使用网络编码技术之后,各个接受者的网络流可以在同一链路上,在不消耗多余带宽的情况下共存^[1].因此,在组播中单独考虑各个接收者的速率情况就成为了可能.新的检查接收者组播速率的部分对各个接收者分别检查其传输速率.对于 INCS,不再统一赋予负的适应值,那些能够让更多接收者达到组播速率的方案将被赋予相对更大的适应值.只有那种令所有接收者都无法达到组播速率的方案才会被赋予负的适应值.

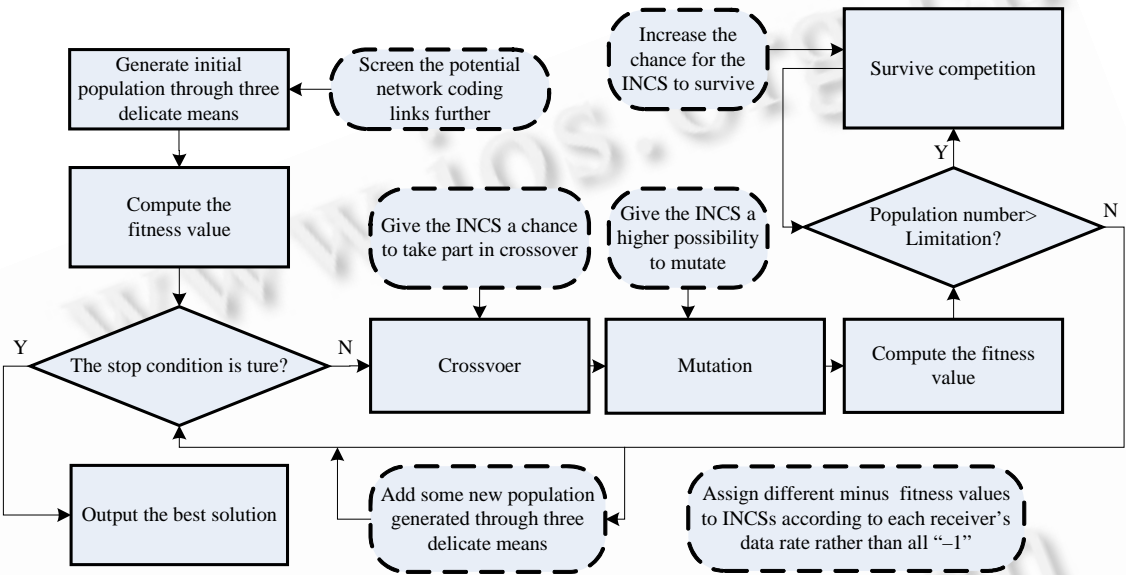


Fig.5 New solutions based on common genetic algorithm
图 5 在一般的遗传算法的基础上加入的新方案

遗传算法在实际应用中需要解决的一个重要问题是局部性^[11].局部性是指搜索算法会最终陷入解空间的一个局部空间进行搜索,所输出的最终解也只是该局部空间的最优解.在对文献[9,10]中算法的验证实验中存在这样的现象,随着程序的运行,所有的成员彼此之间越来越相像,出现了类似近亲繁殖的现象,这是典型的局部性现象.在网络编码优化问题中出现这一现象的主要原因是随机生成的新成员有很大的可能性被淘汰,保存下来的几乎都是某个优秀成员的轻微变异.这就导致算法主循环中的杂交遗传部分形同虚设,几乎无法产生可用的新方案.只有随机翻转二进制的变异环节才有很小的可能产生与这些近亲区别明显的满足所有接收者组播速率的网络编码方案.这就导致了搜索陷入局部性,向最优方案收敛的速度非常缓慢以至于直到最后也无法找到足够优秀的编码方案.上面提到的对 INCS 进一步筛选的方法在一定程度上缓解了局部性的问题,但是新方案的淘汰率仍然很高.为了从根本上解决局部性的问题,我们在遗传算法的主循环中加入一个新的部分.在每次循环开始的时候,同样调用之前提到的重新设计的精细的初始成员生成函数,补充一部分新成员参与之后的遗传环节和基因变异环节,同时修改完全基于适应值的遗传环节选择策略,固定一部分参与遗传的名额给新生成的成员.实验结果显示,这样的策略基本上解决了局部性的问题,也加快了整个优化算法的收敛过程.

综上所述,我们针对网络编码优化问题的特点,在遗传算法中加入了大量新的解决方案,并改变了遗传算法的流程,实际上提出了一种新的解决方案,称为 IGAS.实验结果显示,IGAS 的性能提升是十分明显的.接下来在第 2 节将对模拟实验的详情和结果作进一步的介绍.

2 仿真实验与结果分析

目前提出的网络编码优化问题的解决方案主要有 4 种:文献[7]、文献[8]、文献[9,10]和文献[12],其中文献[8]的方法已在文献[9]中被证明过于复杂而只有理论上的意义,文献[7]和文献[12]两种方法已经在文献[9]中被证明性能远不及 GAS,因此,下面的比较将主要集中在 GAS 和 IGAS 之间.

为了便于进行数据对比,在实验中 IGAS 采用了与文献[10]同样的资源描述函数,即优化的目标是使参与网络编码的链路数目总量最少(例如如图 1(b)中,参与网络编码链路的数量是 1,即连接点 3 和点 4 的链路).这样就可以在大致相同的拓扑规模上使用文献[9,10]中的数据与 IGAS 进行对比.

我们在包含不同节点数目的不同结构的网络拓扑上进行了模拟实验,其中一部分拓扑是由 BRITE(Boston University representative Internet topology generator,<http://www.cs.bu.edu/brite/>)模拟生成,另一部分拓扑是依照固定的模式由人工生成的.之所以采用了一部分人工生成的拓扑是因为,对于一个 NP 难问题,要获得随机拓扑上的最优解作为算法性能的参照几乎是不可能的.而无论拓扑的规模如何增加,按下面的方法所人工所生成的特殊拓扑上的最优化的网络编码方案却是固定的.我们希望通过随机拓扑测试收敛速度与各种情况下算法的性能,而通过固定拓扑测试算法对最优解的逼近程度.我们所构造的人工拓扑如图 6 所示.

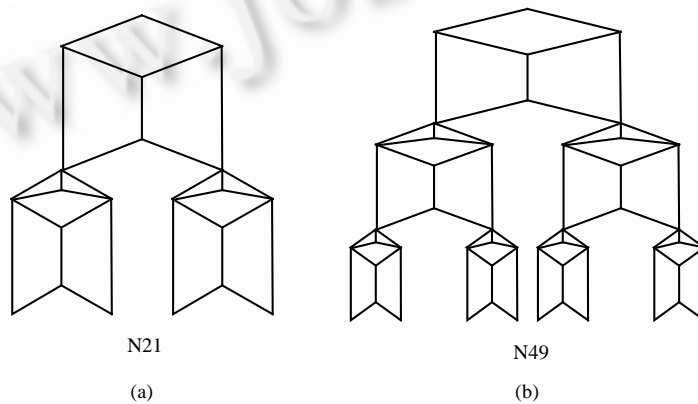


Fig.6 Two artificial topologies

图 6 两种主要的人工拓扑示意

通过将拓扑中的接收者不断地替换为图 1(a)或者图 2(a)的结构,可以获得任意规模的拓扑.而且只有图 1(a)中的一条链路需要进行网络编码.实验中主要使用了图中所示的两种拓扑,称为 N21 和 N49.

参照类似系统的经验,实验中其他参数的设置为:成员规模 100,每次循环加入 50 个新成员,基因突变的几率为 0.000 03.

作为对比,我们按照文献[10]中的非分布式算法的描述完成了一个遗传算法程序作为对比,主要的参数也按照文献[10]的描述进行了设置.在一部分实验中,为了测试算法的持续收敛性,特别设定了比较大的循环次数并且关闭了自动停止的设置.

实验中所采用的 8 组网络拓扑见表 1,拓扑一栏中列出的是该实验所使用的拓扑,N21 与 N49 是图 5 中人工生成的拓扑.四元组(节点数目,链路数目,接收者数目,组播速率)代表了 BRITE 随机生成的一个拓扑的主要参数.这里为了对比随机拓扑与人工拓扑下的不同表现,设定随机生成拓扑包含的节点数目同样包含 21 个节点与 49 个节点,链路的数量也控制在与人工拓扑大体相同的数量上.实验中第 5、6 组以及第 7、8 组实验采用的拓扑相同,区别在于后一组的实验关闭了自动停止的设置并且强制循环 1 000 次以观察持续收敛的性能.

表 2 给出了本文提出的 IGAS 与 GAS 在 8 组条件下各进行 10 次实验的平均结果.表格中最后改进列是最后一次对最佳网络编码方案进行更新的循环次数,平均结果列是优化网络编码方案资源占用量的平均值(即编码链路的数量),最佳结果列是所有实验中输出网络编码方案资源占用量的最小值.

Table 1 Experimental topologies

表 1 实验拓扑

Sequence number	Topology
1	N21
2	(21,40,4,2)
3	(21,57,4,5)
4	(21,76,5,4)
5	N49
6	N49
7	(49,190,6,6)
8	(49,190,6,6)

Table 2 Contrast between IGAS and GAS

表 2 网络编码优化结果对比

Sequence number	Algorithm	Best result last updated (number of cycles)	Average result (number of coding links)	Best result (number of coding links)
1	IGAS	9.8	1.2	1
	GAS	28.2	3.8	1
2	IGAS	7.6	0	0
	GAS	27.9	1.4	1
3	IGAS	18.4	2.8	0
	GAS	59.7	3.7	2
4	IGAS	21.3	1.5	1
	GAS	62.5	4.7	3
5	IGAS	58.5	2.6	1
	GAS	339.4	7.4	7
6	IGAS	218.5	1.4	1
	GAS	658.7	6.4	5
7	IGAS	71.7	4.3	4
	GAS	451.8	20.4	19
8	IGAS	325.2	3.5	3
	GAS	765.1	17.7	17

表 2 显示了在各种拓扑及条件下,IGAS 相对于 GAS 总能找到消耗资源明显更少的编码方案,这一差别在拓扑较大的情况下更为明显.关闭了自动停止的实验(6,8)显示,相同拓扑下,当优化算法运行到后期时,IGAS 仍能持续地对方案进行改进,直到非常接近最优的网络编码方案;而 GAS 由于陷入了局部性,即使强制运行更长的时间,对编码方案也难以有任何提高.另外,两种算法在同样规模的人工拓扑和随机拓扑上的性能基本一致,这也显示出之后用人工拓扑来衡量算法对于最佳结果的逼近程度是可信的.

表 2 的对比只是最终输出编码方案优劣的对比,虽然展示了 IGA 在编码方案优秀程度上带来的提升,却没有体现新加入的解决方案各自的作用.为了更直观地显示各种新方案带来的好处,我们额外地在人工拓扑 N49 上做了实验,完整地记录了两种算法的优化过程,并且以“代-编码边”图的形式表示在图 7 中.

在图 7 中,较低的一条线是 IGAS 的收敛过程,较高的一条线是 GAS 的收敛过程.在 N49 上,最佳的网络编码方案应该是只在一条链路上进行网络编码,即对应的资源消耗数量是 1.可以看出,IGAS 的曲线非常快地收敛于这一最优值.图中 A 点是在所有节点都进行编码的情况下编码链路的数量,一般的遗传算法的初始阶段通常只能得到这样的方案,而经过精细初始化的改进算法则可以在初始化阶段之后立刻得到位于 B 点附近的远远好于 A 点的方案.在算法运行的后期,显而易见,最终 GAS 陷入了局部性,收敛的速度变得非常缓慢;而 IGAS 则可以很快收敛到最优解附近(点 C,大约 120 次循环左右).另外,IGAS 点的分布更为密集,整个搜索过程更均匀,不像 GAS 随机性那么强.从图中可以直观地看出,IGAS 中所加入的新解决方案的好处明显地体现在遗传算法的各个阶段.

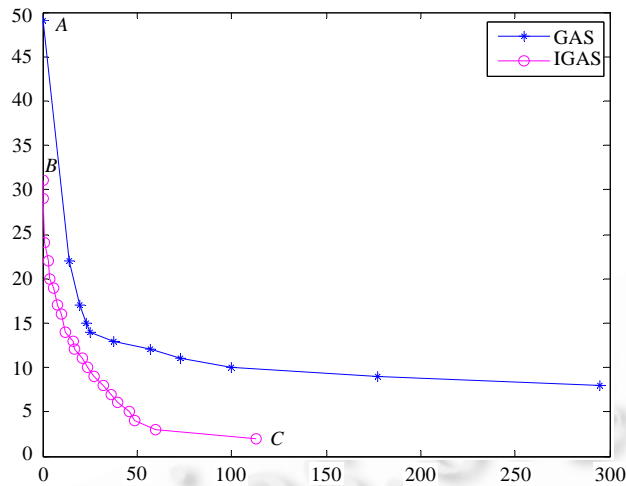


Fig.7 Contrast of two solutions on N49

图 7 在 N49 上对两种解决方案的对比实验

最后,我们在各种规模的随机拓扑上对比 IGAS 与 GAS 的总运行时间,两种方法平均输出网络编码优化方案的时间见表 3.

Table 3 Runtime contrast between IGAS and GAS**表 3** IGAS 与 GAS 运行时间对比

	Node number in topologies			
	15	30	60	120
IGAS	3.9s	14.8s	105.6s	24min.
GAS	6.4s	822.7s	5.5h	N/A

即使不考虑最后输出的网络编码优化结果的优劣(表 2 已经很清楚地展现了网络编码优化结果的对比),GAS 输出结果所需要的运行时间也远大于 IGAS;当拓扑规模持续增加时,GAS 的运行时间甚至增长到实际应用所无法容忍的程度.在含有 120 个节点的随机拓扑上,GAS 每循环一次就需要半小时左右,而这种规模的应用需要至少数百次循环才能产生结果.IGAS 在运行速度上的提高如此明显,原因主要有以下两点:首先,IGAS 所采用的预先甄选编码链路的方法在实验拓扑中能够发现相当数量的不必编码的候选链路,实际上极大地降低了问题的复杂度;其次,由于各种针对性解决方案的加入,IGAS 向最优网络编码方案收敛的速度比 GAS 快得多,能够在远小于 GAS 所需要的循环次数内输出优化的网络编码方案.

在实验中我们额外地观察到这样的现象:在多数的随机拓扑上,网络编码并不是达到最大组播速率所必需的;而即使是必须采用网络编码的拓扑,网络编码也仅仅发生在极少数链路上.这样的结果说明在所有链路都进行编码是巨大的浪费,本文对网络编码的优化工作是非常有意义的.

我们注意到,在这些不需要网络编码就可以达到最大组播速率的随机拓扑上,GAS 仍然要运行对应于拓扑规模的一定时间及循环次数,而 IGAS 只需要很少的循环次数以及很短的运行时间就可以输出不采用网络编码的传输方案(相对于 GAS),这是由于 IGAS 采用了新的初始成员生成算法,并且在每次循环开始的时候都调用这一生成算法.正如第 1.2 节中介绍的,新的初始成员生成算法中一部分传输方案是基于尽量尝试转发这一策略生成的,从而在不需要编码(即转发)的场景下对遗传算法的筛选过程起到了显著的加速作用.

3 总 结

本文围绕网络编码资源优化问题展开了相关研究.针对原有优化方法^[1,2]运行时间过长和优化结果不够令人满意的不足,分析了网络编码资源优化问题的随机解通常不能让所有接受者都达到组播速率的特点,在原有

的解决方案中加入新的部分.第一,给出了通过资源描述函数生成遗传算法中适应值函数的方法,使各种不同的网络编码资源优化问题都能在文中的解决方案下得到解决.第二,对拥有多条输入链路的输出链路进行检验,进一步缩小算法的筛选范围.第三,修改了遗传算法的流程,加入以下处理:1) 在遗传算法的初始化阶段使用更为精细的方法替代随机方法产生更高质量的初始成员;2) 在遗传算法每次循环开始的时候调用初始成员生成算法,加入一定数量的新成员,从根本上避免了局部性;3) 给不能让所有接收者都达到组播速率的编码方案赋予更为恰当的适应值而不是统一设为-1,使这些方案也能参与遗传产生后代.实验结果显示,IGA 优化方法运行更为快速,所产生的解的质量更高.

进一步的工作包括尝试利用非启发式的近似算法来求解网络编码资源优化问题,并与启发式算法的性能进行对比.

References:

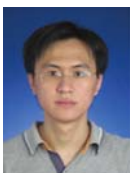
- [1] Ahlswede R, Cai N, Li SYR, Yeung RW. Network information flow. *IEEE Trans. on Information Theory*, 2000,46(4):1204–1206.
- [2] Li SYR, Yeung RW, Cai N. Linear network coding. *IEEE Trans. on Information Theory*, 2003,49(2):371–381.
- [3] Jaggi S, Sanders P, Chou PA, Effros M, Egnér S, Jain K, Tolhuizen LMGM. Polynomial time algorithms for multicast network code construction. *IEEE Trans. on Information Theory*, 2005,51(6):1973–1982.
- [4] Wang M, Li BC. How practical is network coding. In: *Proc. of the 14th IEEE Int'l Workshop on Quality of Service (IWQoS 2006)*. 2006. 274–278.
- [5] Wu YN, Jain K, Kung SY. A unification of network coding and tree-packing (routing) theorems. *IEEE Trans. on Information Theory*, 2006,52(6):2398–2409.
- [6] Cannons J, Zeger K. Network coding capacity with a constrained number of coding nodes. *IEEE Trans. on Information Theory*, 2008,54(3):1287–1290.
- [7] Langberg M, Sprintson A, Bruck J. The encoding complexity of network coding. *IEEE Trans. on Information Theory*, 2006,52(6):2386–2397.
- [8] Bhattad K, Ratnakar N, Koetter R, Narayanan KR. Minimal network coding for multicast. In: *Proc. of the IEEE ISIT. Adelaide: IEEE Computer Society*, 2005. 1730–1734.
- [9] Kim M, Ahn CW, Médard M, Effros M. On minimizing network coding resources: An evolutionary approach. In: *Proc. of the NetCod*. 2006.
- [10] Kim M, Médard M, Aggarwal V, O'Reilly UM, Kim W, Ahn CW, Effros M. Evolutionary approaches to minimizing network coding resources. In: *Proc. of the IEEE INFOCOM 2007*. Anchorage: IEEE Computer Society, 2007. 1991–1999.
- [11] Whitley D. A genetic algorithm tutorial. *Statistics and Computing*, 1994,4(2):65–85.
- [12] Fragouli C, Solijanin E. Information flow decomposition for network coding. *IEEE Trans. on Information Theory*, 2006,52(3):829–848.



邓亮(1983—),男,上海人,硕士,主要研究领域为网络编码.



王新(1973—),男,博士,副教授,CCF 高级会员,主要研究领域为自组织网络,多媒体传输.



赵进(1979—),男,博士,讲师,CCF 会员,主要研究领域为对等网络,网络编码.