多约束应用层组播的算法研究

来卫国,侯惠峰,李 鸥
LAI Wei-guo,HOU Hui-feng,LI Ou

解放军信息工程大学 信息工程学院 通信工程系,郑州 450002
Communication Engineering Department,Information Engineering College,Information Engineering University of PLA,Zhengzhou 450002,China

LAI Wei-guo,HOU Hui-feng,LI Ou.Study of multi-constrained application layer multicast algorithms.Computer Engineering and Applications,2007,43(19):10-12.

Abstract: The paper studies the two layered application layer multicast networks based on Multicast Service Nodes(MSN),presents the degree-delay-constrained minimum spanning tree problem,and provides a heuristic algorithm named DD-Prim algorithm for this problem.In order to further improve the solution to this problem,the authors introduce a bias vector into the DD-Prim algorithm,rename it BDD-Prim algorithm.The authors use it as a decoder for genetic algorithm.Simulation results proof the genetic algorithm's efficiency.

Key words: multi-constrained;multicast;minimum spanning tree;genetic algorithm

摘 要:研究了基于组播服务节点(MSN)的两层应用层组播网络,提出了度和时延联合约束的最小生成树问题(DDCMST 问题),并给出了求解该问题的启发式算法——DD-Prim 算法。为了进一步提高求解的精度,在该算法中引入了偏置向量,得到了 BDD-PRIM 算法,并将其作为染色体编码的译码器应用到遗传算法中。仿真结果证明了遗传算法的有效性。

关键词:多约束;组播;最小生成树;遗传算法
文章编号:1002-8331(2007)19-0010-03 **文献标识码**:A **中图分类号**:TP301

1 引言

网络层组播协议未能在 Internet 获得广泛应用,应用层组播取而代之,得到长足发展并开始商业应用。应用层组播的系统结构分为基于主机的平面结构和基于组播服务节点(MSN)的分层结构。分层结构由 MSN 节点构成核心层,客户机通过连接到 MSN 节点来获取组播服务。MSN 核心层网络具有以下特点:

(1)由于任意两个 MSN 节点间均通过网络层路径物理相连,故任意两个 MSN 节点在应用层是虚拟连接的;

(2)由于接口带宽和处理能力有限,一个 MSN 节点只能连接有限个 MSN 节点,故 MSN 节点存在度约束条件。

度约束最小生成树问题(DCMST 问题)研究度约束条件下的应用层组播的优化问题。文献[1]在无约束最小生成树算法——普林(Prim)算法的基础上提出了求解 DCMST 问题的 D-Prim 算法。文献[2-4]研究了求解 DCMST 问题的遗传算法。

对于多数实时业务的应用层组播来说,还要考虑时延约束条件。因此本文提出了度和时延联合约束最小生成树问题。基于 D-Prim 算法,设计了求解 DDCMST 问题的贪婪算法——DD-Prim 算法。为了进一步提高求解的精度,本文创造性地在 DD-Prim 算法中引入偏置向量,并基于此提出了求解 DDCMST 问题的遗传算法。仿真结果证明了该算法的有效性。

2 网络模型和问题描述

MSN 网络用 $G(V,E)$表示,V 为节点集合,E 为边集合。由 MSN 网络特点(1)可知,$G(V,E)$为完全图(complete graph)。$e\in E$ 为图 G 的边,$cost(e)\in Z^{+}$为边 e 的代价,$delay(e)$为边 e 的时延。$d_T(v)$为树 T 上顶点 v 的度,$d_{\max}(v)\in Z^{+}$表示顶点 v 的度约束值。

定义 1 树的代价:树的代价 $cost(T)$为树 T 上所有边代价之和,即 $cost(T)=\sum\limits_{e\in T}cost(e)$。

定义 2 路径时延:路径时延 $D_T(v)$为从根节点 src 到节点 v 的路径上所有边的时延之和,即 $D_T(v)=\sum\limits_{l\in p}delay(l)$,其中 p 为树 T 上从 src 到 v 的路径。

定义 3 树的时延:树的时延 $D(T)$为树 T 上所有路径时延的最大值。即 $D(T)=\max\limits_{v\in T}(D_T(v))$。

定义 4 度和时延联合约束最小生成树问题:已知一个无向完全图 $G=(V,E)$,边代价 $cost(e)$和边时延 $delay(e)$,给定根节点 $root$,时延约束 D 和节点度约束 $d_{\max}(v)$,寻找图 G 的一个以 $root$ 为根的代价最小的生成树 T,且满足:

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.60472064)。
作者简介:来卫国(1972-),男,博士研究生,研究方向为 QoS 路由,组播技术,网络优化;侯惠峰,博士研究生,研究方向为无线传感器网;李鸥,教授,博士生导师,研究方向为无线通信。

$$\begin{cases} d_T(v)\leqslant d_{\max}(v) & \text{for all } v\in V \\ delay_T(v)\leqslant D & \text{for all } v\in V \end{cases} \tag{1}$$

定理 1 DDCMST 问题是 NP 完全问题。

证明 当时延限制 D 为无穷时，原问题成为 DCMST 问题。已知 DCMST 问题为 NP 完全问题[1]，故 DDCMST 问题为 NP 完全问题。

3　DD-Prim 算法

在该算法中，将节点分为 4 种颜色：没有找到前驱节点的树下(off-tree)节点为白色，找到前驱节点的树下节点为灰色，节点的当前度达到度约束的树上(on-tree)节点为红色，未达到度约束的树上节点为黑色。该算法从根节点开始，每次选择一个满足所有约束的节点加入到当前树中，直到当前树成为生成树。当某个树上节点变为红色时，就调用 Red_Node_Process 过程(图 2)进行节点状态的更新。DD-Prim 算法的伪码见图 1。其中，$C(v)$ 为连接节点 v 与其前驱节点的边的代价，$D(v)$ 为从根节点 $root$ 经过节点 v 的前驱节点到 v 的路径时延。

```
DD-Prim_Algorithm(G(V,E),cost(e),delay(e),root,d_max(v),DB)
输入：完全图 G(V,E)，边代价 cost(e)，边时延 delay(e)，根节点 root，度约束
      d_max(v)，时延约束 DB
输出：多度约束最小生成树 T=(W,L)，W 为节点集合，L 为边集合
1 foreach v∈V do
2   if delay(root,v)≤DB then
3     color(v)=GRAY；prev(v)=root；C(v)=cost(root,v)；D(v)=delay(root,v)
4   else
5     color(v)=WHITE；
6 color(root)=BLACK；
7 while (|L|<|V|-1) do
8   将所有灰色节点按照 C(v)的升序排列，并令 u 为该排列的首个节点
9   p=prev(u)；color(u)=BLACK；
10    L=L∪{{p,u}}；//将边{p,u}加入当前树
11  if d_T(p)==d_max(p) then
12    color(p)=RED；
13    Red_Node_Process(p)；//调用红色节点处理程序；见图 2。
14  foreach  树下节点 v do
15    if delay(u,v)+D(u)≤DB then
16      color(v)=GRAY；prev(v)=u；C(v)=cost(u,v)；D(v)=delay(u,v)+D(u)；
17  returen T
```

图 1　BD-Prim 算法的伪码

```
Red_Nolde_Process(p)
begin
foreach 灰色节点 v do
  if prev(v)==p then
    min=∞；
    foreach 黑色节点 w do
      if cost(w,v)<min and D(w)+delay(w,v)≤DB then
        min=cost(w,v)；C(v)=cost(w,v)；
        prev(v)=w；D(v)=D(w)+delay(w,v)；
end
```

图 2　红色节点处理程序

4　遗传算法

DD-Prim 算法是贪婪算法，不能保证结果的精确度。为了进一步提高算法精度，本文设计了求解 DDCMST 问题的遗传算法。首先改进了 DD-Prim 算法，得到了 BDD-Prim 算法，并以此作为染色体编码的译码器。

4.1　BDD-Prim 算法

BDD-Prim 算法中引入了一个 0-1 型偏置向量，在算法从排序好的灰色节点序列中作选择时(见图 1 第 8 行)，用当前偏置分量的值来决定选择的位置。如偏置分量为 0 则选择灰色节点序列的第一个节点，否则选择该序列中的第二个节点。

4.2　个体(染色体)编码

设网络节点数为 N，个体编码为长度为 N-1 的 0-1 型整数向量。个体每位服从参数为 p 的两点分布。

4.3　适应度函数

给定一个个体 i，以此个体编码作为偏置向量输入 BDD-Prim 算法，得到该个体对应的多约束生成树 T_i。个体 i 的适应度函数取为树 T_i 的代价函数，即 $fit(i)=c(T_i)$。

4.4　遗传算子

选择算子：使用轮盘赌算法，群体中任一个体 i 被选中的几率为

$$p(i)=\frac{fit_{\max}-fit(i)}{P^*fit_{\max}-\sum_{j=1}^{P}fit(j)}$$

其中 P 为种群规模，$fit_{\max}$ 为当前群体的最大适应度值，$fit(i)$ 为个体 i 的适应度值。

交叉算子：选用单点交叉算子。随机选择交叉位置，交叉概率为 p_c。

变异算子：随机单点变异。将变异位取反，变异概率为 p_m。

4.5　基本型和终止条件

使用 $\lambda+\mu$ 基本型。每进化代产生 μ 个新个体，与当前群体合并后，再剔除最差的 μ 个个体。采用固定代数的算法终止条件，即当进化代数达到设定值时，算法终止。

4.6　算法步骤

步骤 1 随机产生规模为 P 的群体。令临时群体规模为 0。

步骤 2 While 临时群体规模小于 μ do

用选择算子从当前群体中选择两个个体；

以概率 p_c 对上述两个个体进行交叉运算，得到两个新个体；

分别以概率 p_m 对两个新个体进行变异运算；

计算两个新个体的适应度函数，并将二者加入临时群体。

步骤 3 将临时群体与当前群体合并，剔除其中适应度函数值最大的 μ 个个体，得到新群体。

步骤 4 检查停止条件，如果满足条件，则算法终止，输出当前群体的最优个体及其适应度值；否则，清空临时群体，转到步骤 2。

5　仿真设计及结果分析

本文仿真程序用标准 C++语言编写，并利用了由麻省理工大学 Matthew Wall 开发的遗传算法库和印第安那大学开发的 Boost 图库。仿真主机的配置为 2.66 GHz P4 CPU，512 MB 内存。操作系统为中文 Windows XP SP2。

本文仿真比较了所提出的 DD-Prim 算法和遗传算法(GA

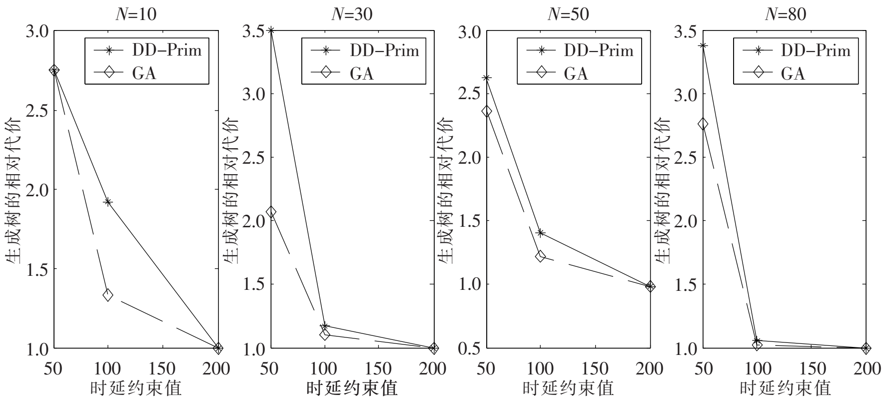

图 3 不同节点数量与时延约束下,DD-Prim 算法与 GA 算法所得组播树的相对代价

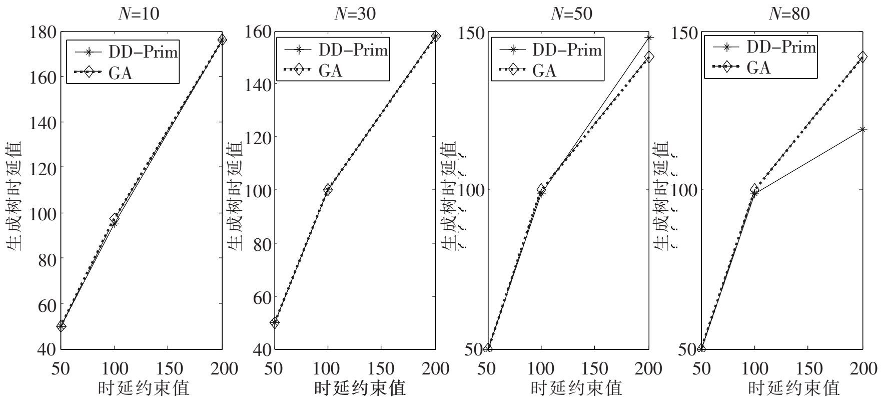

图 4 不同节点数量与时延约束下,DD-Prim 算法与 GA 算法所得组播树的时延值

算法)的性能。为便于比较,将 DD-Prim 算法和 GA 算法的结果除以 D-Prim算法的结果,得到生成树代价的相对值。

仿真中生成了 4 个完全图,每个图的节点数量分别为 10、30、50 和 80。在每个图中,边的代价服从[1,10]上均匀分布的随机变量,边的时延服从[10,50]上的均匀分布。对上述每一个图,分别令时延限制为 50、100 和 200 进行仿真。由于 D-Prim 算法、DD-Prim 算法是确定算法,故每次仿真只运行 1 次,而遗传算法是非确定算法,故每次仿真时运行 3 次,取最优值作为遗传算法的结果。算法的结果见图 3 和图 4。

由图 3 可知,当时延约束为 50 时,所有网络规模(节点数量)下,DD-Prim 算法与 GA 算法得到的组播树代价均远大于 D-Prim 算法得到的组播树的代价(生成树的相对代价大于 2)。这是因为为了满足苛刻的时延约束,前两种算法在生成树构造过程会优先选择时延值小的边, 而这些边的代价值可能很大。当时延约束为 200 时, 由于此时延约束过于宽松, 故对应的 DDCMST 问题就简化为不带时延约束的 DCMST 问题, 所以 DD-Prim 算法与 GA 算法得到的相对代价均为 1。在所有不同的网络规模下,GA 算法的结果均好于 DD-Prim 算法, 体现了并行算法的优势。不过遗传算法的运行时间要远远大于 DD-Prim 算法。

由图 4 可以看出,在所有网络规模下,DD-Prim 算法与 GA 算法得到的生成树的时延值均满足时延约束条件。

6 结论

针对应用层组播的实际需求,提出了度与时延联合约束下的最小生成树问题, 并给出了求解该问题的启发式算法——DD-Prim 算法。为了提高求解的精度,在 DD-Prim 算法中引入偏置向量,并应用到遗传算法中。仿真结果证明了遗传算法的有效性。在启发式算法中引入控制因素,然后应用到遗传算法中,这样就在启发式算法和遗传算法之间建立了桥梁,使得遗传算法可以充分利用算法领域的先进成果而不断发展进步。

(收稿日期:2007 年 3 月)

参考文献:

[1] Narula S C,Ho C A.Degree-constrained minimum spanning tree[J].Computers and Operations Research,1980,7(4):239-249.

[2] Raidl G R.An efficient evolutionary algorithm for the degree-constrained minimum panning tree problem[C]//Proc of the 2000 IEEE Congress on Evolutionary Computation.[S.l.]:IEEE Press,2000:104-111.

[3] Raidl G R.A weighted coding in a genetic algorithm for the degree-constrained minimum spanning tree problem[C]//Como:Proceedings of the 2000 ACM symposium on Applied computing,2000:440-445.

[4] Knowles J,Come D.A new evolutionary approach to the degree constrained minimum spanning tree problem [J].IEEE Transactions on Evolutionary Computation,2000,4(2):125-134.