

文章编号:1001-9081(2007)05-1180-03

多约束 QoS 移动 IP 最优路由的求解

朱翠涛,高 靛,汪汉新

(中南民族大学 电子信息工程学院,湖北 武汉 430073)

(zhucuitao@163.com)

摘要:提出了一种改进的分布式遗传算法(EDGA)和链路编码方法用于解决多约束服务质量(QoS)移动 IP 最优路由选择问题。将分布式遗传算法和 Cascade 模型相结合,在迁移算子部分设计一个中心监控器,观察每个子种群的进化,并对迁移个体的选择以及相应子种群的大小做出调整,使进化能力好的子种群得到更大的空间来搜索最优值。实验结果表明,该 EDGA 算法在求解速度和质量上取得了较大的改善。

关键词:分布式遗传算法;多约束路由选择;移动 IP

中图分类号: TN915.05 **文献标识码:** A

Multiple constrained QoS optimal routing selection of mobile IP

ZHU Cui-tao, GAO Liang, WANG Han-xin

(College of Electrical and Information Engineering, South-Central University for Nationalities, Wuhan Hubei 430074, China)

Abstract: This paper proposed a new Genetic Algorithm (GA), called Extended Distributed Genetic Algorithms (EDGA), and link coding for multiple constrained Quality of Service routing selection of Mobile IP. EDGA combines the advantages of Distributed Genetic Algorithms and Cascade Model, based on the separating operation, dynamically allotting the size of sub-populations according to their performance, directing the migration, and therefore making the searching of routes converge to the global optimization faster and better. The result of the experiments shows that the method is more effective than others.

Key words: distributed genetic algorithms; multiple constrained routing selection; mobile IP

0 引言

多约束最优路由选择的主要目标是为接入的业务选择满足其服务质量要求的传输路径,同时保证网络资源的有效利用。多约束最优路由选择分为:单播和多播,两类路由密切相关,很多情况下多播路由可以视为单播路由的推广。按照所求解的问题类型和求解方法,现有的单播多约束最优路由选择算法可归纳为:多项式非启发类、QoS 度量相关、探测法、扩展距离向量算法及花费函数法。

多约束 QoS 移动 IP 路由选择就是在网络中寻找一条既过定点,又能满足其带宽、时延和费用等要求的最佳通路。在现有的方法中,Dijkstra 方法,洪泛法,模拟退火法^[1]等用于求解多约束 QoS 路由选择问题,已取得了不少成果。这些算法的共同特点是通过一些变动规则,路由从搜索空间中的当前点到另一点,对于多峰分布常常会陷入局部的某个单峰的最优解。遗传算法(GA)相对于其他判决和寻优算法,具有并行搜索,快速评估的优点^[2,3]。然而,GA 最大缺陷在于其未成熟收敛,目前对其改进的方法分为集中型和分散型。本文从分布式遗传算法^[4]和 Cascade 模型^[5]入手,提出了一种改进分布式遗传算法(EDGA),并提出链路编码的方法,用于解决多约束 QoS 移动 IP 路由选择。

1 问题定义

定义 1 $G = (V, E)$ 称为网络模型的加权图,其中 V 是网络中所有节点组成的集合, E 是所有连接两个相邻节点的通

信链路(边)的集合。 S 为源节点, $D = \{d_1, d_2, \dots, d_n\}$ 为目的节点, M 为必过的中间节点。 $C = c_{ij}$ 表示费用矩阵(c_{ij} 为节点 i 到 j 的费用), $B = b_{ij}$ 为带宽矩阵(b_{ij} 为节点 i 到 j 的利用带宽), $\Delta = \Delta_{ij}$ 表示时延矩阵(Δ_{ij} 为节点 i 到 j 的通信的时延), $L = l_{ij}$ 表示丢包率矩阵(l_{ij} 为节点 i 到 j 丢包率)。其目标函数为:

$$v(L)^* = \min \left(\sum_{i=S}^M \sum_{j=S}^M f_{ij} x_{ij} + \sum_{i=M}^D \sum_{j=M}^D f_{ij} x_{ij} \right) \quad (1)$$

其中, $v(L)^*$ 为最小代价通路, x_{ij} 为节点 i 和节点 j 间通信代价,包括费用、时延和丢包率的综合。 f_{ij} 为连接指示函数:

$$f_{ij} = \begin{cases} 1, & i, j \text{ 间存在路由} \\ 0, & i, j \text{ 间不存在路由} \end{cases} \quad (2)$$

对于一个路由请求 R ,如果能够找到一条具有最小代价的路径 L ,同时满足以下条件,则 L 可被路由请求 R 接受。这些条件是:

$$(1) b_{ij} \leq B_s \quad (3)$$

$$(2) \sum_{i=S}^{d_t} \Delta_{ij} \leq \Delta_s, t = 1, 2, \dots, n \quad (4)$$

$$(3) 1 - \prod_{i=S}^{d_t} (1 - l_{ij}) \leq L_s, t = 1, 2, \dots, n \quad (5)$$

$$(4) \min \sum_{i=S}^{d_t} c_{ij} \leq C_s, t = 1, 2, \dots, n \quad (6)$$

其中 B_s, Δ_s, L_s, C_s 分别代表 QoS 要求的带宽、时延、丢包率和费用。为了阻止环路产生,约束条件为:

收稿日期:2006-11-15;修订日期:2007-01-17 基金项目:国家民委重点基金资助项目(07ZN02)

作者简介:朱翠涛(1967-),男,湖北十堰人,副教授,主要研究方向:IP 网、分布式计算;高靛(1982-),女,湖北仙桃人,硕士研究生,主要研究方向:路由算法;汪汉新(1966-),男,湖北武汉人,副教授,主要研究方向:软件无线电。

$$\sum_{j=8}^D f_{ij} - \sum_{j=8}^D f_{ji} = \begin{cases} \pm 1, & i = D \\ 0, & i \neq D \end{cases} \quad (7)$$

$$\sum_{i=8}^D f_{ij} \begin{cases} \leq 1, & i \neq D \\ = 0, & i = D \end{cases} \quad (8)$$

2 染色体及遗传算子的实现

遗传算法操作的是一群编码化的可行解,称为种群。它通过种群的更新与迭代来搜索全局最优解,而迭代是通过选择、交叉和变异等具有生物意义的遗传算子来实现。一般称包含选择、交叉和变异算子的遗传算法为传统遗传算法,记为SGA。

2.1 染色体

在遗传算法的应用中,首要工作是设计问题解的编码方式,即染色体表示,可以采用二进制、整数和实数变量等编码方式。

为了降低遗传算子操作复杂度,本文提出了链路编码的方法。在一个网络拓扑图中,两点间所有的链路有多少,编码的长度就有多长,当这个链路存在就用1表示,不存在就用0表示。链路编码方法示意网络拓扑如图1所示,假设路径为1-2-3-5,其编码为:

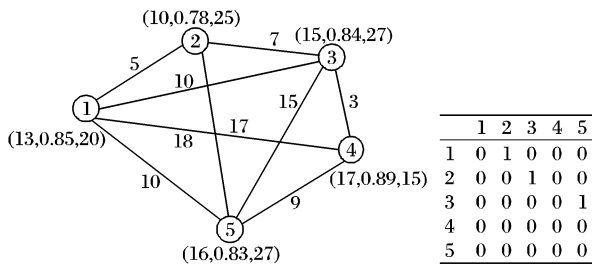


图1 链路编码示意图

与二进制编码过程中的初始化一样,产生一个行为种群数,列为基因位数的矩阵,然后以一个概率为0.5的随机函数为每一个体的基因位赋值为0或1。同时初始化中如果发现不符合约束条件的个体,要重新生成。

2.2 适应度评估函数

由于遗传算法在进化搜索中以适应度函数为依据,利用种群中每个个体的适应度值进行搜索,因此适应度函数直接影响到遗传算法的收敛速度以及能否找到全局最优解。本设计的适应度函数为:

$$f = \frac{1}{\sum c_{ij} \times \Delta_{ij}} \times \prod (i - l_{ij}) \quad (9)$$

c_{ij} 与 Δ_{ij} 之间的乘积越小及整条通路的丢包率越小,通路的适应度就越大。根据个体编码与适应函数解空间的对应关系来计算它的适应度。

2.3 遗传算子的实现

(1) 选择算子。通常人们利用一个随机函数决定一个个体的是否被保存,此方法的选择步骤简单,但是它带有盲目性,因为它的选择与个体优劣无关。本文采用排序法,用一个适应度线性相关的选择函数,把适应度高、性能好的个体保存下来。根据个体的适应度来决定自己被保留的可能性。这样我们能尽可能的保留最优的个体。

(2) 交叉算子。主要作用是全局寻优,它与编码方式密切相关。链路编码方式下,当随机函数产生概率小于变异概率时,随机生成小于编码长度大于0的两个数 i, j , 比较它们的大小,把两个选择用来交叉的个体 $\min(i, j)$ 和 $\max(i, j)$ 间的码交换。主要程序代码如下:

```
for R = 1: popsize
    if rand(1) < Pc
        crossj1 = floor((length( oldp1) - 1) * rand) + 1;
        crossj2 = floor((length( oldp1) - 1) * rand) + 1;
        minjcross = min( crossj1, crossj2);
        maxjcross = max( crossj1, crossj2);
        segment1 = oldp1( minjcross: maxjcross);
        segment2 = oldp2( maxjcross: - 1: minjcross);
        oldp12 = eliminate( oldp1, segment2);
        oldp21 = eliminate( oldp2, segment1);
        newp1 = [ segment2, oldp12];
        newp2 = [ segment1, oldp21];
        m = [ newp1; newp2];
```

(3) 变异算子。其目的:一是使在选择和交叉过程中某些丢失的遗传基因进行补充和恢复;二是避免算法陷入局部最优。

当随机函数产生概率小于变异概率,就要进行变异,随机生成小于编码长度大于0的数 m , 取 m 位基因的反码。当然我们也可以因为链路的特性,必须使整个路由是通路,我们换了一种变异方法,在链路间选择两点交换,来完成变异。主要程序代码如下:

```
for R = 1: popsize
    if rand(1) < Pm
        p = pop( R, :);
        a = floor( lchrom * rand(1)) + 1;
        b = floor( lchrom * rand(1)) + 1;
        minmutation = max( a, b);
        maxmutation = min( a, b);
        p([ minmutation maxmutation]) = p([ maxmutation minmutation]);
        t = pathlenfit( p, distmatrix);
```

3 改进的分布式遗传算法

为了提高遗传算法的求解速度和质量,在SGA算法基础上,本文提出了一种改进的分布式遗传算法(EDGA),算法流程如图2所示。

首先将整个种群分解成几个子种群,每个子种群单独地执行SGA,同时周期性地每隔 $Migint$ 代在每个子种群中选取特定的个体迁移到别的子种群中去。分布遗传算法可以使种群的进化跳出局部最优,使搜索更灵活,但是由于进行SGA的单个种群变小,个体的多样性减少,进化能力也随之削弱。为了克服此问题,本文在迁移算子部分设计了一个中心监控器,观察每个子种群的进化,并对迁移个体的选择以及相应子种群的大小做出调整,使进化能力好的子种群得到更大的空间来搜索最优值,同时尽可能把进化能力弱的子种群中的精英个体吸收到进化能力更好的子种群中去,满足其进化个体的多样性,使其更好的搜索出最优解。具体实现步骤为:

(1) 子群体空间再分配

迁移过程中,中心监控器对每个子种群的进化程度进行询问,整个子种群空间将依据各个子种群进化的优劣被重新瓜分。中心监控子程序计算出每个子种群中最优个体的适应度,并根据适应度大小对各个子种群进行排队。子种群下一次进化得到的空间大小与其当前的适应度成线性关系。

迁移过程中,中心监控子程序计算出每个子种群中最优个体的适应度,并根据适应度大小对各个子群进行排队。子群下一次进化得到的空间大小和其当前的适应度成线性关系:

$$P_i = \frac{f(P_i)}{F} P \quad (10)$$

其中, P_i 是下一次第 i 个子群被分配的个体个数, P 是整

个群体的总个数, $f(p_i)$ 是第 i 个子群的适应度函数, F 是总适应度函数 $F = \sum_{i=1}^n f(p_i)$ 。

进化过程中, 各个子种群在规模上会有很大不同。当一个子群生成条件很窄, 个体数很少, 那么它很难产生有竞争力的个体, 这样它在下次迁移时就很可能被继续缩小。如果一个子群进化已变得很迟钝, 而其他子种群继续发展时, 留给它的空间会越来越小, 形成恶性循环, 导致其最终消失。这样, 就又回到了 SGA。为平衡这一点, 本文选取一个加权值 $\alpha (0 \leq \alpha \leq 1)$ 。

$$P_i = \alpha \hat{P} + (1 - \alpha) \frac{f(p_i)}{F} P \quad (11)$$

其中, \hat{P} 是子种群的平均大小, $\hat{P} = p/n$, n 为子种群数。当 $\alpha \rightarrow 0$ 时, 加权几乎不起作用; 当 $\alpha \rightarrow 1$ 时子种群大小差别可以忽略, 优等种群得到奖励会很少。因此, α 取值必须适当。

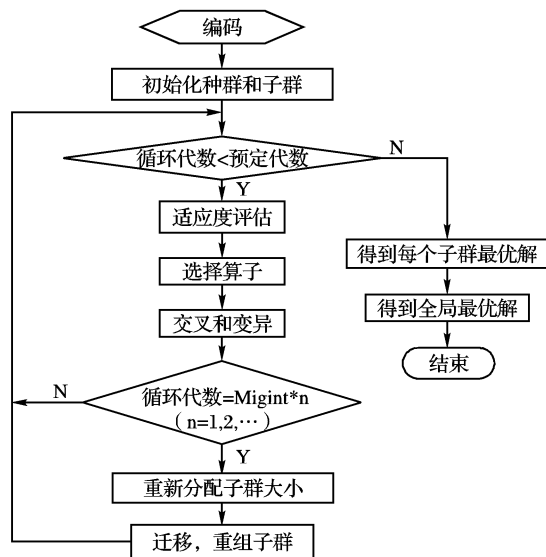


图2 EDGA 的流程图

(2) 个体迁移

适应度较低的子种群把自己的局部最优解迁移到适应度较高的子种群, 如式 (12) 所示, M_{p_i} 表示可以接受第 i 个子种群迁移个体的子种群集。

$$M_{p_i} = \{p_j \mid f(p_i) \leq f(p_j)\} \quad (12)$$

当子种群规模减小时, 其中适应度最小的那一部分个体被丢弃; 当子种群规模增大时, 除了吸收外部来的最优个体外, 还要随机复制自己来填充空间。适应度最高的子种群不会迁移任何个体到其他子种群中, 而适应度最低的子种群将没有个体迁移过来, 并且还要丢失一部分个体。

4 仿真实验与分析

利用 Matlab 仿真工具对本文提出的算法进行了实现。在 20 个节点的拓扑下, 分别对 SGA、EDGA 以及两种迁移策略进行了性能比较, 为了便于观察, 以适应度的倒数作为参照观察算法的收敛特性。把整个种群分成 4 个子种群 SUBP1、SUBP2、SUBP3 和 SUBP4, 同时执行本文实现的 SGA, 当执行了 q 代后, 比较各个子种群的最大适应度值, 重新分配空间, 然后再分别对种群 SUBP1, SUBP2, SUBP3, SUBP4 进行迁移操作。为了方便比较, SGA 运行了 500 代, 在这里每隔 10 代迁移一次。各子种群的收敛过程如图 3 所示。

为了说明 EDGA 优越性, 对 SGA、DGA (基于网络拓扑的迁移策略) 和 EDGA 进行了比较, 如图 4 所示。由图可以看

出, EDGA 在运行到第 18 代迁移时就收敛了, $shortpathlength4 = 3158$, $elapsed_time = 0.3440$ 。由于迁移间隔为 10 代, 那么达到总体收敛的时间是第 180 代, 而 SGA 收敛代数达到了 480 左右。同时, 最优路径综合代价改进前为 3803, 改进后是 3158。由此可以看出 EDGA 的优越性。

5 结语

本文将改进的分布式遗传算法用于解决移动 IP 的多约束 QoS 路由选择问题, 该算法在分布式遗传算法的基础上, 采用子种群来避免收敛到局部最优, 并且设计一个中心控制子程序控制子种群规模的大小和子种群之间个体的迁移方向, 使适应度高的子种群得到更好的个体, 更加容易收敛到全局最优。实验结果表明该算法大幅度提高了 QoS 路由选择的效率。当节点数更多时, 该算法的优势将会更加明显。

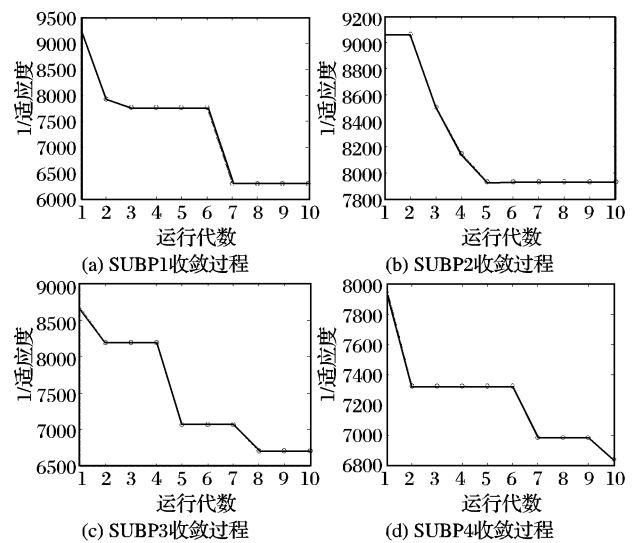


图3 各子种群的收敛过程

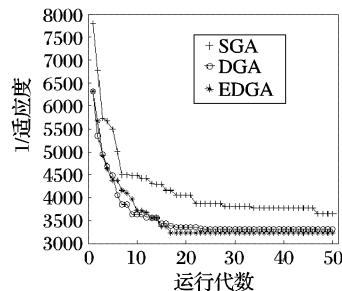


图4 SGA、DGA、EDGA 性能比较

参考文献:

[1] CUI Y, XU K, WU JP, et al. Multi-constrained routing based on simulated Annealing[J]. IEEE International Conference, 2003, (3): 102 - 106.

[2] ZOMAYAT AY. Observation on Using Genetic-Algorithms for Channel Allocation in Mobile Computing[J]. IEEE Transactions on Parallel And Distributed System, 2002, 13(9): 948 - 962.

[3] KARABUDAK D, HUANG CC, BING B. An Intelligent Admission Control Scheme for Next Generation Wireless Systems using Distributed Genetic Algorithms[J]. IEEE communications society, 2004, 13: 1748 - 1753.

[4] KARUNANITHI N, DAS R, WHITLEY D. Genetic Cascade Learning for Neural Networks[J]. IEEE Combinations of Genetic Algorithms and Neural Networks, 1992, (1): 134 - 145.

[5] VIGNAUX GA, MICHALEWICZ Z. A Genetic Algorithm for the Linear Transportation Problem[J]. IEEE Transactions on Systems, Man and Cybernetics, 1991, 21(2): 445 - 452.