

基于人工鱼群算法的多播树演化寻优

马炫, 刘庆

(西安理工大学 自动化与信息工程学院, 陕西 西安 710048)

摘要:提出了一种双域模型人工鱼群算法。算法采用前驱节点指向的编码方法形成多播树表示人工鱼, 将搜索空间分为可行域和非可行域。分别赋予可行域和非可行域的人工鱼不同的游动目标, 设计行为算子自适应地执行 4 种人工鱼行为。数值实验结果表明, 提出的算法可以有效利用非可行个体, 具有较好的求解时延约束最小代价多播树的性能。

关键词:多播树; 时延约束; 人工鱼群算法; 双域模型

中图分类号: TP18

文献标识码: A

文章编号: 1000-436X(2012)09-0001-07

Evolving multicast tree based artificial fish swarm algorithm

MA Xuan, LIU Qing

(School of Automation and Information Engineering, Xi'an University of Technology, Xi'an 710048, China)

Abstract: An artificial fish swarm algorithm with two regions model was proposed. The algorithm used a multicast tree formed by the predecessor orientation method to represent an artificial fish, and divided the search space into feasible region and infeasible region. The artificial fishes in the feasible region and the infeasible region were given different moving objectives, respectively. The behavior operator was designed to adaptively implement four behaviors of artificial fish. Experimental results show that the proposed algorithm can effectively make use of the infeasible individuals and performs better in searching least-cost multicast tree with delay constraint.

Key words: multicast tree; delay constraint; artificial fish swarm algorithm; two regions model

1 引言

多播(multicast)是一种由源节点同时向多个目的节点发送信息的通信方式, 被广泛应用于多媒体信息传输, 如视频会议、远程教育等。多播路由技术是实现多播通信方式的一个关键技术, 可以归结为图论中的求解受限 Steiner 树问题, 该问题是一个 NP 完全问题。近年来, 启发式算法和基于进化算法的求解方法受到许多学者的关注, 分别提出了基于 Dijkstra 的 CDMPH 算法^[1]、遗传算法、蚁群优化算法、粒子群优化算法及混合进化算法等^[2~11]。这些算法分别设计了不同的寻优机制搜索最优多

播树, 使求解质量和求解速度得到大幅提高, 详细算法可参考上述文献。

然而, 在基于进化算法的求解多播树的方法中, 产生不满足约束条件的非可行解是难以避免的。实验表明, 当约束条件比较苛刻时, 大量非可行个体的产生对算法性能影响比较大。目前, 处理非可行个体主要有以下 3 种方法, 一是将非可行个体直接从种群中剔除^[2], 二是对非可行个体进行修复使其变为可行解^[12, 13], 三是在种群中对非可行个体在评价时采用惩罚的方法^[14]。将非可行个体从种群中剔除会损失个体中经过若干代的进化已有的一些优秀的进化信息, 而且易使算法性能劣化。修

收稿日期: 2011-09-08; 修回日期: 2012-03-20

基金项目: 陕西省教育厅专项科研计划基金资助项目(09JK641)

Foundation Item: The Department of Education of Shaanxi Province (09JK641)

复的方法是利用求解问题的性质对非可行解予以修复,但由于是取决于问题的,难以找到通用的修复方法。对于时延受限多播树,多播树的修复比较复杂且效率不高。惩罚的方法是通过一个预先设定的惩罚系数对目标函数值予以惩罚,这种方法会带来另外一个困难的问题,即惩罚参数值的选择。基于此,本文研究在进化算法中如何将设计寻优机制和非可行解处理有机结合,从而提高算法性能。

人工鱼群算法是受自然鱼的游动行为启发提出的一种群智能进化算法^[15],具有较好的自组织性和自适应性,在求解大规模复杂优化问题上取得了较好效果^[16]。本文提出一种基于人工鱼群算法的双域模型,通过利用非可行个体提高求解时延约束最小代价多播树算法的性能。算法将人工鱼的搜索空间分为可行域和非可行域,并为 2 个区域的人工鱼赋予不同的行为目标,通过自适应地实施各种游动行为,实现最优解的搜索。与 CDMPH 算法、遗传算法和粒子群算法进行数值实验比较,验证算法的有效性。

2 多播路由问题描述

给定一个无向赋权图 $G=(V, E)$, V 和 E 分别表示 G 中所有节点的集合和所有边的集合。 G 中存在一个源节点 s 和一个目的节点集合 $M \in \{V - \{s\}\}$, 设 $p_T(s, d_j)$ 是从源节点 s 到目的节点的 d_j 的路径。对于任意 2 个节点之间的边 $e \in E$, 时延函数 $delay(e): E \rightarrow R^+$, 代价函数 $cost(e): E \rightarrow R^+$, 带宽函数 $bandwidth(e): E \rightarrow R^+$ 。以源节点 s 为根的多播树 $T(s, M)$ 有如下关系

$$C(T(s, M)) = \sum_{e \in T(s, M)} cost(e) \quad (1)$$

$$B(p_T(s, d_j)) = \min\{bandwidth(e), e \in p_T(s, d_j)\} \quad (2)$$

$$D(P_T(s, d)) = \sum_{e \in P_T(s, d)} delay(e) \quad (3)$$

多播路由问题可描述为

$$\begin{cases} \min \sum_{e \in T} C(T(s, M)) \\ D(p_T(s, d_j)) & D \\ B(p_T(s, d_j)) & B \end{cases} \quad (4)$$

其中, D 和 B 分别表示时延约束和带宽约束。

3 双域模型人工鱼群算法

3.1 人工鱼行为

鱼类一般具有 4 种典型行为^[15]: 1) 随机游动行为, 鱼在水中漫无目的地随意游动; 2) 觅食行为, 鱼在水中觅食, 一旦在其视野内发现食物, 鱼会迅速向食物的方向游动, 与随机游动相比, 觅食行为具有目的性; 3) 追尾行为, 当鱼群中的一条或几条鱼发现食物时, 其邻近的伙伴会尾随其快速游向食物; 4) 聚群行为, 鱼在游动过程中会自然聚集成群, 这是一种保证群体的生存和躲避危害而形成的一种生活习性。

鱼的追尾和聚群行为形成鱼群, Reynolds^[17]认为, 鱼类群集的形成只需要每条鱼遵循一些局部的相互作用规则即可, 而群集现象作为整体模式会自发地从个体的局部相互作用中突现出来。Reynolds 提出了 3 条规则: 1) 分隔规则, 尽量避免与邻近伙伴过于拥挤; 2) 对准规则, 尽量与邻近伙伴的平均方向一致; 3) 内聚规则, 尽量向邻近伙伴的中心移动。人工鱼群算法在分析鱼群行为特点的基础上定义了人工鱼的各种行为, 形成了基于行为的寻优机制。随机游动和觅食行为体现人工鱼的个体行为, 追尾和聚群行为体现人工鱼的社会行为。

3.2 双域模型

待求解问题的最优解在搜索空间中一定存在于可行域一侧且靠近可行域和非可行域的边界附近。然而, 对于复杂优化问题, 在解空间中要确定实际的可行域和非可行域边界是比较困难的。本文在人工鱼的搜索功能上将搜索空间分为可行域和非可行域, 满足约束条件的人工鱼在可行域以搜索最小代价多播树为目标, 而不满足约束条件的人工鱼在非可行域以游向可行域为目标, 分别实施觅食、追尾、聚群和随机游动等行为。人工鱼在游动中可能会从可行域进入非可行域, 因此, 对人工鱼位置的评价采用下述规则: 1) 可行域的人工鱼优于非可行域的人工鱼; 2) 可行域的人工鱼代价越小越优; 3) 非可行域的人工鱼超出约束的程度越小越优。

3.3 求解最优多播树

3.3.1 人工鱼编码

如何将一棵多播树映射成一条人工鱼并有利于人工鱼行为的实现, 是需要解决的首要问题。观察一棵树, 从叶节点向根节点追溯, 除根节点外, 每个节点只有一个前驱节点。因此, 本文采用前驱

节点指向的编码方法，可以方便地用数组结构对人工鱼进行编码。网络拓扑和多播树如图1所示。

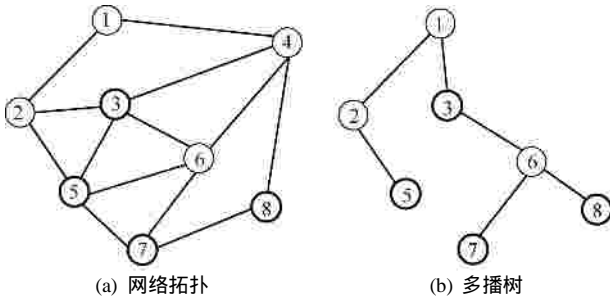


图1 网络拓扑和多播树

例如，对于图1(a)所示的网络拓扑，设源节点为1，目的节点为{3,5,7,8}，一棵多播树可用数组结构形式表示如图2所示。

元素	1	1	1	#	2	3	6	6
下标	1	2	3	4	5	6	7	8

图2 多播树的数组结构形式

根节点1的前驱节点是它本身，节点2和节点3的前驱节点是节点1，‘#’表示节点4没有前驱节点，不是多播树成员，节点5和节点6的前驱节点分别是节点2和节点3，节点7和节点8的前驱节点是节点6。由上述前驱节点指向关系表示的多播树如图1(b)所示。由于树的前驱节点的唯一性，这种追溯前驱节点的编码方法不会产生环路。

3.3.2 人工鱼行为算子

将一条人工鱼从位置 X 向目标位置 X_{Aim} 游动到达位置 X' ，记为 $X'=Toward(X, X_{Aim})$ 。对于用多播树表示的人工鱼，这一过程是通过树的结构演化实现的。具体方法是，将 X_{Aim} 中不同于 X 的各个分量取出，得到一些零散结构；将这些零散结构添加到 X 的树结构上，得到一个有环图；然后，随机删除有环的边，得到一棵新树 X' ， X' 具有 X 和 X_{Aim} 的部分结构。人工鱼位置的优劣直接用多播树代价评价，即 $f(\bullet)=C(T(s, M))$ ，其值越小则越优。采用2条人工鱼 X_i 和 X_j 不相等位的个数表示它们在解空间的距离，将人工鱼 X 在视觉半径 $Visual$ 内的空间称为人工鱼 X 的视野。双域模型的人工鱼行为算法伪代码描述如下。

1) 随机游动行为

```
CFish  $X_{Aim}$ ;
Do{
```

```

 $X_{Aim}.Construct\_Tree(Graph G)$ ;
}While( $Distance(X, X_{Aim}) > Visual$ );
 $X' = Toward(X, X_{Aim})$ ;
2) 觅食行为
Switch
{
Case1:  $X$  在可行域
Int Attempt =  $ATTEMPT$ ;
Bool Done = false;
Do{
Generate  $X_{Aim}$ ;
If ( $f(X_{Aim}) < f(X)$ )
{  $X = X_{Aim}$ ;
Done=true; }
Else
Attempt - - ;
} While (Attempt > 0 && (! Done));
If (! Done)
执行随机游动;
Break;
Case2:  $X$  在非可行域
Int Attempt =  $ATTEMPT$ ;
Bool Done = false;
Do{
Generate  $X_{Aim}$ ;
If( $X_{Aim}$  在可行域)
{  $X = X_{Aim}$ ;
Done=true; }
If( $X_{Aim}$  在非可行域)
{
If( $X_{Aim\_Delay} < X\_Delay$ )
{  $X = X_{Aim}$ ;
Done=true; }
If( $X_{Aim\_Delay} = X\_Delay$  &&
 $f(X_{Aim}) < f(X)$ )
{  $X = X_{Aim}$ ;
Done=true; }
}
}
If(!Done)
Attempt - - ;
} While (Attempt > 0 && (! Done));
If (! Done)
执行随机游动;
```

```

    Break;
}
3) 追尾行为
CFish XMin;
Int nf=0; // X 邻域的人工鱼数
For(int i=0; i<POPULATION- 1; i++)
    If(Distance(X, X[i]) < Visual)
    {
        If(f(X[i])<f(XMin))
            XMin =X[i];
        nf ++;
    }
If(nf ==0)
    X.Prey();
Else
{
    Switch
    {
        Case1: X 和 XMin 均在可行域
            If (f(XMin)·nf < f(X)·d) //d 为拥挤度因子
                X' =Toward(X, XMin);
            Else
                X.Prey();
            Break;
        Case2: X 在可行域, XMin 在非可行域
            X.Prey();
            Break;
        Case3: X 在非可行域, XMin 在可行域
            X' =Toward(X, XMin);
            Break;
        Case4: X 和 XMin 均在非可行域
            If (XMin_Delay<X_Delay)
                X' =Toward(X, XMin);
            Else
                X.Prey();
            Break;
    }
}

```

4) 聚群行为

计算人工鱼视野内全部人工鱼的中心 X_c , 每条人工鱼向 X_c 游动。聚群行为实现的关键是如何确定以多播树为对象的各种树结构的中心。本文采用的方法是合并所有人工鱼所表示的树结构, 形成一个图 G' , 然后, 在图 G' 中随机生成一棵多播树作为 X_c 。这样,

X_c 概率性地具有群中人工鱼的某些结构, 可近似地代表当前群的结构。为避免拥挤, 用拥挤度因子 d 进行控制。X 向 X_c 的聚群行为与上述追尾行为执行的 4 种情况相同, 只是目标位置为 X_c 。伪代码略。

3.4 迭代更新

在一个迭代周期中, 一条人工鱼根据环境执行某一个或几个行为, 然后, 比较每个行为产生的结果, 用最好的结果更新该人工鱼, 并将其记录为标本。如果下一次迭代产生的人工鱼优于该标本, 则用其更新标本。将标本作为当前代种群的最优人工鱼, 当算法结束时标本作为搜索结果。

3.5 收敛性分析

文献[18]从理论上证明了量子粒子群优化(QPSO)算法的全局收敛性。QPSO 是一种基于迭代的群智能优化(SIO)算法, 该算法能够收敛到全局最优解必须要具有 3 个性能^[18]: 1)有一个能够在待求解空间产生解的操作, 并保证产生的新个体优于当前个体; 2)个体向群体当前最优平均中心靠拢; 3)群体最优解具有单调性。

人工鱼群算法是一种新型群智能进化算法, 具有与 QPSO 类似的群体进化特性。人工鱼在可行域的觅食行为中, 若 $f(X_{Aim}) < f(X)$, 则用 X_{Aim} 替换 X , 因而可以保证产生的新个体优于当前个体, 满足 SIO 的第一个性能要求; 人工鱼的聚群行为通过追尾算子使人工鱼视野内的全部人工鱼向当前最优平均中心 X_c 聚集, 满足 SIO 的第 2 个性能要求; 算法中人工鱼的迭代更新方式和标本可保证群体最优解具有单调性, 满足 SIO 的第 3 个性能要求。而且, 非可行域中人工鱼向可行域的游动可增强算法搜索到最优解的能力。因此, 本文算法具有收敛到全局最优解的性能。

文献[19]给出了进化算法的收敛性定理: 若一个进化算法(EA)满足 2 个条件: 1)种群序列 $p(1), p(2), \dots, p(t), \dots$ 是单调的, 即对 $\forall t, p(t+1)$ 中任意解不劣于 $p(t)$ 中任意解; 2)对可行域中任意 2 个点 $\overset{r}{X}$ 和 $\overset{r}{X'}$, $\overset{r}{X'}$ 是从 $\overset{r}{X}$ 通过变异和重组可达的, 则 EAs 以概率 1 收敛到问题的全局最优解。下面利用该定理给出本文算法收敛性证明。

设种群中最优人工鱼为 $X_m(t)$, 人工鱼的随机游动算子, 觅食算子和追尾算子产生的新位置分别为 $X_r(t)$ 、 $X_p(t)$ 和 $X_f(t)$, 取最好的位置为

$$X(t) = \arg \min \{ f[X_r(t)], f[X_p(t)], f[X_f(t)] \} \quad (5)$$

此时种群中的最优人工鱼更新为

$$X_m(t+1) = \begin{cases} X(t), & (f[X(t)] < f[X_m(t)]) \\ X_m(t), & (f[X(t)] \geq f[X_m(t)]) \end{cases} \quad (6)$$

式(5)和式(6)保证了种群序列是单调的。

人工鱼在可行域的觅食算子中存在的操作，即如果 $f(X_{Aim}) < f(X)$ ，则执行 $X = X_{Aim}$ ，随机游动算子和追尾游动算子中的操作 $X' = Toward(X, X_{Aim})$ 均是 X 向 X_{Aim} 可达的，由文献[19]的定理可得，当 $t \geq 8$ 时算法收敛到全局最优解。

4 数值实验

算法采用 Visual C++ 6.0 编程 在 Intel (R) Core II processor T5 600 (1.83 GHz)上运行。

4.1 与启发式算法的比较

DCMPH 算法^[1]是性能比较好的求解时延约束多播树的启发式算法，将本文算法与文献[1]的算法进行比较。图 3 是随机生成的具有 26 个节点 65 条

边的网络拓扑，每条边定义 3 个属性(代价、带宽、时延)。

任选 5 个多播需求，带宽需求和时延约束分别为 $B=5, D=15$ 。算法参数设置为：人工鱼数量 20， $ATTEMPT=15, Visual=8, d=12$ ，迭代次数 100。本文算法和 DCMPH 算法搜索到的最优多播树如表 1 所示，表中的多播树用连接节点的形式表示。从表 1 可以看出，本文算法搜索到的 5 棵多播树均明显优于用 DCMPH 算法求得的多播树。由于本文算法是全局搜索，而 DCMPH 是基于贪心算法设计的，容易陷入局部最优解，因而算法的求解质量好于 DCMPH。

4.2 与 GA 和 PSO 的比较

文献[2]和文献[11]分别提出了求解时延多播树的遗传(GA)算法和粒子群优化(PSO)算法，为了检验本文算法(AFSA)的性能，与这 2 种进化算法进行

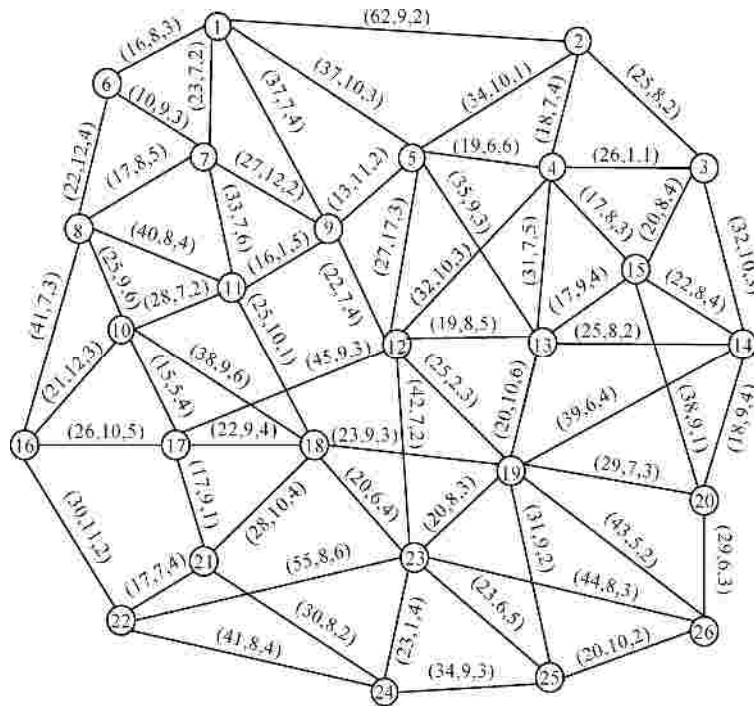


图 3 网络拓扑

表 1 最优多播树

源节点? 目的节点	树的结构(本文算法)	代价(本文算法)	树的结构(DCMPH ^[1])	代价(DCMPH ^[1])
1? 3,12,18,22,26	1 5 2 3, 5 12 23 18, 23 22, 23 25 26	283	1 2 3, 2 5 12 23 18, 23 26, 23 22	309
3? 8,18,20,22,24	3 2 5 9 7 8, 5 12 17 21 22, 17 18, 21 24, 3 14 20	324	3 14 20, 14 19 18, 19 25 24, 3 2 5 12 23 22, 5 9 7 8	417
5? 10,17,20,24,25	5 13 12 17 10, 13 19 20, 19 25 24	228	5 12 17 10, 17 21 24 25, 5 2 3 15 20	285
12? 3,6,14,16,24	12 13 14 3, 12 9 7 6, 12 17 21 24, 17 16	253	12 13 14 3, 12 9 7 6, 7 8 16, 12 17 21 24	285
20? 2,8,9,17, 25	20 19 13 5 2, 19 25, 19 18 11 8, 18 17, 5 9	272	20 26 25, 20 19 18 17, 18 11 8, 20 15 3 2 5 9	318

比较。数据对象采用 4.1 节中的一个多播需求 {5, 10, 17, 20, 24, 25}。分别执行 3 种算法各 20 次，平均进化曲线如图 4 所示。可以看出，AFSA 进化曲线的收敛速度和收敛值均明显优于 GA 和 PSO 的进化曲线，表明算法具有较好的收敛性。

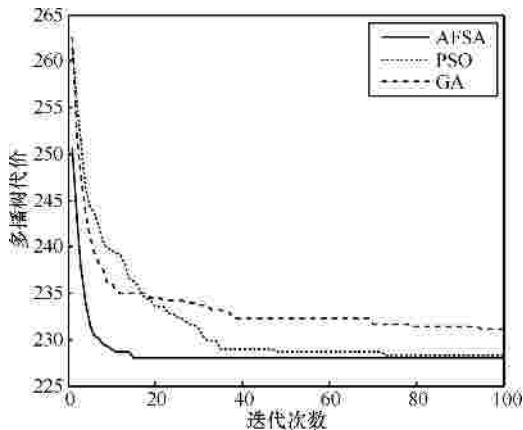


图 4 平均进化曲线

在图 3 的网络中随机选择不同数目的目的节点进行测试，3 种算法的求解结果如图 5 所示。从比较中可以看出，AFSA 的求解质量最好。

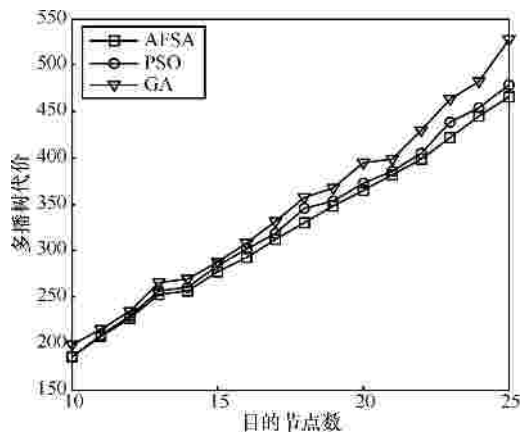


图 5 不同目的节点数的解

本文还针对不同网络拓扑进行了测试，随机生成网络节点数分别为 40、50、60、70、80、90、100 等 7 种网络拓扑，目的节点数分别取网络节点数的 20%，执行 3 种算法各 20 次，得到最优解的平均值和平均算法耗时如图 6 和图 7 所示，图 6 中的数值是对 GA 获得值的相对值。可以看出，AFSA 的求解质量优于其他 2 种算法，当网络规模增大时，优势明显增大。图 7 中 AFSA 的耗时稍大于 PSO，这是由于双域模型中增加了对非可行个体处理的算子。然而，正是这些算子提高了 AFSA 的求解质量。

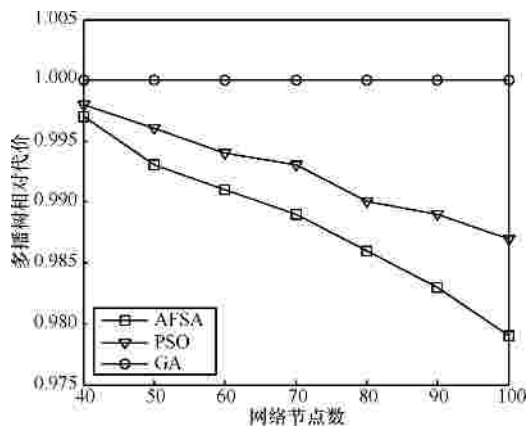


图 6 不同网络规模的最优解比较

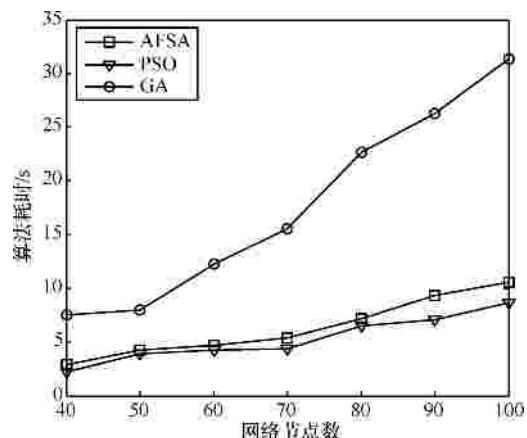


图 7 算法耗时

4.3 双域模型与罚函数方法比较

分别执行双域模型的人工鱼群算法(T-AFSA)和采用罚函数对非可行解处理的人工鱼群算法(P-AFSA)各 20 次，种群中非可行解人工鱼数量与种群数量比率的变化如图 8 所示。可以看出，双域模型的非可行解人工鱼数量明显少于罚函数的方法，这可以提高在可行域搜索到最优解的可能性。实验结果中，T-AFSA 比 P-AFSA 搜索到最优解的速度更快。

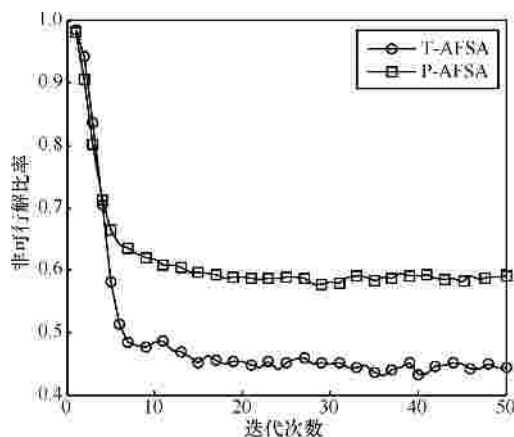


图 8 非可行解比率

5 结束语

本文提出了求解时延约束多播树的双域模型人工鱼群算法。基于树的前驱节点指向的编码方法可以直观地表示一棵多播树,使树的生成比较简单且避免了环路的生产。人工鱼分别在可行域和不可行域按不同的寻优目标自适应地实施游动行为,有效地利用了非可行个体。数值实验结果表明,提出的算法具有较好的求解时延约束最小代价多播树的性能。双域模型人工鱼算法还可以应用于其他优化问题。

参考文献：

- [1] 周灵, 孙亚民. 基于 MPH 的时延约束 Steiner 树算法[J]. 计算机研究与发展, 2008, 45(5):810-816.
ZHOU L, SUN Y M. A delay-constrained Steiner tree algorithm using MPH[J]. Journal of Computer Research and Development, 2008, 45(5): 810-816.
- [2] 刘伟彦, 张顺颐. 下一代网络中基于遗传算法的 QoS 多播路由算法[J]. 电子与信息学报, 2006, 28(11): 2157-2161.
LIU W Y, ZHANG S Y. A QoS multicast routing algorithms based on genetic algorithm in next-generation networks[J]. Journal of Electronics & Information Technology, 2006, 28(11): 2157-2161.
- [3] VIJAYALAKSHMI K, RADHAKRISHNAN S. Artificial immune based hybrid GA for QoS based multicast routing in large scale networks[J]. Computer Communications, 2008, 31(17): 3984-3994.
- [4] 范一鸣, 余建军, 方志敏. 融合小生境机制的 QoS 多播路由遗传模拟退火算法[J]. 通信学报, 2008, 29(5): 65-71.
FAN Y M, YU J J, FANG Z M. Hybrid genetic simulated annealing algorithm based on niching for QoS multicast routing[J]. Journal on Communications, 2008, 29(5): 65-71.
- [5] ZHANG L, CAI L B, LI M. A method for least-cost QoS multicast routing based on genetic simulated annealing algorithm[J]. Computer Communications, 2009, 32(1): 105-110.
- [6] 章兢, 王炼红. 基于改进克隆选择算法的时延受限多播路由[J]. 通信学报, 2009, 30(7): 54-58.
ZHANG J, WANG L H. Multicast routing with delay constrain based on improved CSA[J]. Journal on Communications, 2009, 30(7): 54-58.
- [7] WANG H, SHI Z, LI S. Multicast routing for delay variation bound using a modified ant colony algorithm[J]. Journal of Network and Computer Applications, 2009, 32(1): 258-272.
- [8] WANG H, XU H, YI S W. A tree-growth based ant colony algorithm for QoS multicast routing problem[J]. Expert Systems with Applications, 2011, 38(9): 11787-11795.
- [9] HUANG C J, CHUANG Y T, HU K W. Using particle swarm optimization for QoS in ad-hoc multicast[J]. Engineering Applications of Artificial Intelligence, 2009, 22(8): 1188-1193.
- [10] WANG H, MENG X X, LI S, *et al.* A tree-based particle swarm optimization for multicast routing[J]. Computer Networks, 54(15): 2775-2786.
- [11] SUN J, FANG W, WU X J, *et al.* QoS multicast routing using a quantum-behaved particle swarm optimization algorithm[J]. Engineering Applications of Artificial Intelligence, 2011, 24(1):123-131.
- [12] CHOOTINAN P, CHEN A. Constraint handling in genetic algorithms using a gradient-based repair method[J]. Computers & Operations Research, 2006, 33(8): 2263-2281.
- [13] Salcedo S. A survey of repair methods used as constraint handling techniques in evolutionary algorithms[J]. Computer Science Review, 2009, 3(3): 175-192.
- [14] WU W H, LIN C Y. The second generation of self-organizing adaptive penalty strategy for constrained genetic search[J]. Advances in Engineering Software, 2004, 35(12): 815-825.
- [15] 李晓磊, 邵之江, 钱积新. 一种基于动物自治体的寻优模式: 鱼群算法[J]. 系统工程理论与实践, 2002, 22(11): 32-38.
LI X L, SHAO Z J, QIAN J X. An optimizing method based on autonomous animals: fish-swarm algorithm[J]. Systems Engineering-Theory & Practice, 2002, 22(11): 32-38.
- [16] 黄光球, 朱华平, 周静. 用鱼群算法求解石油运输系统多级站定位优化问题[J]. 系统工程理论与实践, 2008, 28(3): 94-102.
HUANG G Q, ZHU H P, ZHOU J. An optimization method of multistage stations locating in oil transportation based on fish-swarm algorithm[J]. Systems Engineering-Theory & Practice, 2008, 28(3): 94-102.
- [17] REYNOLDS C W. Flocks, herds, schools: a distributed behavioral model[J]. Computer Graphics, 1987, 21(4): 25-34.
- [18] 方伟, 孙俊, 谢振平等. 量子粒子群优化算法的收敛性分析及控制参数研究[J]. 物理学报, 2010, 59(6): 3686-3694.
FANG W, SUN J, XIE Z P, *et al.* Convergence analysis of quantum-behaved particle swarm optimization algorithm and study on its control parameter[J]. Acta Physica Sinica, 2010, 59(6): 3686-3694.
- [19] BÄCK T. Evolutionary Algorithms in Theory and Practice[M]. New York: Oxford University Press, 1996.

作者简介：



马炫 (1962-), 男, 陕西西安人, 博士, 西安理工大学教授, 主要研究方向为智能计算和模式识别。



刘庆 (1983-), 男, 陕西西安人, 西安理工大学硕士生, 主要研究方向为进化计算。