

文章编号: 1006-2467(2005)04-0585-05

基于模拟退火的多约束路径优化选择算法

罗勇军, 石明洪, 白英彩

(上海交通大学, 计算机科学与工程系, 上海 200030)

摘 要: 多约束 QoS 路由是下一代互联网的核心问题之一, 为解决 QoS 路由算法容易陷入局部最优的问题, 将模拟退火方法与路由计算结合起来, 提出一种新的组合优化算法. 该算法可以及时从无效的迭代中跳出到其他空间, 提高搜索效率, 具有全局收敛性, 可在有限次数迭代中快速找到可行路径. 实验表明, 该算法性能高, 同时对网络规模和多约束个数具有很好的可扩展性.

关键词: 多约束; 服务质量; 路由; 全局优化

中图分类号: TP 393.02

文献标识码: A

A Simulated Annealing Based Algorithm for Multi-constrained Path Selection

LUO Yong-jun, SHI Ming-hong, BAI Ying-cai

(Dept. of Computer Science & Eng., Shanghai Jiaotong Univ., Shanghai 200030, China)

Abstract: Multi-constrained QoS routing is one of the challenging problems of the upcoming next-generation networks. Taking into account the local optimization of some QoS routing algorithms, this paper introduced a new combinatorial optimization algorithm, which combines the simulated annealing and k -shortest path algorithm. It is a good method to jump out to other useful space, and find the feasible paths. It can achieve good convergence. The simulation shows that the algorithm is efficient and has a good scalability in both network scale and weight number.

Key words: multiple constraint; quality of service (QoS); route; global optimization

多约束 QoS 路由是 NP 完全问题, 必须用启发式算法来解决. 由于多约束路径问题的复杂性, 对一个约束可行的路径, 往往对另一个不成立. 为简化这个问题, 在目前提出的算法中, 经常采用的方法有: ① 综合考虑各 QoS 参数, 用某个费用函数来统一处理多个约束之间的关系. 算法 H-MCOP^[1] 提出的费用函数是 g^* , 是多个参数的多项式组合. 算法 TAMCRA^[2] 提出的函数是各参数和约束之比的极值. ② 更有效地预测前向路径. 算法 H-MCOP 用反向 Dijkstra, 利用 $g^*, \lambda = 1$ 算出从目的节点开始的反

向的路径, 提供给正向寻径计算时, 从中间节点到目的节点的路径预测. TAMCRA 利用 k -shortest 算出来的前 k 个最优路径, 提供更多的路径选择. Chen^[3] 的预测则是扩展了前向路径的深度. ③ 为减少算法的复杂度, 把路径分成多个部分分别考虑. 例如, 可以把整个路径分为 2 部分, 即从源节点 s 到中间节点 u 的部分路径, 和从 u 到目的节点 t 的部分路径. 上述方法中, 组合优化问题的局部最优问题并未得到很好讨论. 在文献 [4] 中, 提出一种将模拟退火原理与 Dijkstra 算法结合的算法. 本文采用 k -

收稿日期: 2004-05-26

作者简介: 罗勇军 (1973-), 男, 湖北荆门人, 博士生, 主要研究方向为计算机网络、路由算法等.

白英彩 (联系人), 男, 教授, 博士生导师, 电话 (Tel.): 021-62933227, E-mail: bai-yc@cs.sjtu.edu.cn

shortest 算法,在寻径过程中,用模拟退火跳出局部最优,可以更好地解决局部最优和全局最优问题.

1 多约束路径选择问题和模拟退火原理

定义 1 多约束路径选择问题 (Multi-Constrained Path Problem, MCP): 一个网络,用有向图 $G=(V,E)$ 描述,其中: V 是网络节点集合, E 是网络链路集合. 每个链路 $\text{link}(i,j) \in E$ 的参数包括一个基本的费用参数 $c(i,j)$ 和 K 个加性的 QoS 参数 $w_k(i,j), k=1,2,\cdots,K$, 所有参数为非负数. 给定约束条件 α , 问题是找到一个从 s 到 t 的路径 p , 它满足条件:

$$w_k(p) \stackrel{\text{def}}{=} \sum_{(i,j) \in p} w_k(i,j) \leq \alpha$$

模拟退火算法^[5]是局部搜索算法的扩展,是一个全局最优算法. 退火是一种物理过程. 一种金属在加热到一定温度后,它的所有分子在状态空间 D 中自由运动. 随着温度下降,这些分子逐渐停留在不同的状态. 温度最低时,分子重新以一定结构排列. 统计力学研究表明,在温度 T , 分子停留在状态 r 满足波尔兹曼概率分布:

$$\Pr\{E = E(r)\} = \frac{1}{Z(T)} \exp\left[-\frac{E(r)}{k_B T}\right]$$

其中: E 为分子能量的一个随机变量; $E(r)$ 为状态 r 的能量; $k_B > 0$ 为波尔兹曼常数; $Z(T)$ 为概率分布的标准化因子,将概率归一化.

能量慢慢降低的过程,就是费用函数减小,到达最优解的过程. 根据定义 1, MCP 问题的目标是找出一个路径 p , 使 $w_k(p) \stackrel{\text{def}}{\leq} c_k$. 模拟退火算法的能量函数可参考上式定义.

定义 2 能量函数,

$$f(p) = \max\left\{\frac{w_1(p)}{c_1}, \frac{w_2(p)}{c_2}, \dots, \frac{w_K(p)}{c_K}\right\}$$

为 p 的能量函数. 其中, $w_k(p)$ 和 α 分别是 QoS 参数和对应的约束. 如果 p 是可行路径,必有 $f(p) \leq 1$.

2 多约束路径选择算法 DS-MCP

2.1 基本思想

Dijkstra 算法是单约束寻径问题的最好算法,它计算最短路径时,只在中间节点上存储一个最短路径,因此它不能处理多个参数约束或者非线性参数的路径问题. 文献 [6] 提出从 Dijkstra 算法推广出的 k -shortest 路径算法,它以最小化非线性费用函数为目标. 使用 k -shortest 路径算法,如果节点上存

储的路径数 M 足够大,它可以搜索到所有的源节点和目的节点间的路径,得到准确的解. 当然随着 M 的增大,需要的存储空间和计算时间都急剧增加. 在绝大多数情况下,在实际运算中取一个较小的 M 即可以满足要求. k -shortest 算法的关键在于,对 n 个节点的网络,普通 Dijkstra 算法的邻居队列有 n 个,而 k -shortest 算法的邻居队列大于 Mn 个,即每个节点可以多次进入邻居队列,每次进入时对上次的路径进行更新. 对每个节点来说,从源节点到它有 M 个路径. 总体上看,算法需要判断的路径数并不多,因为一个节点包含的路径数据,在前面节点的路径数据中已经包括了.

k -shortest 路径算法不能避免局部最优问题,而模拟退火的特点是可以从某些局部最优跳出来,扩大搜索范围. 在本算法中,把模拟退火原理和 k -shortest 算法结合,以得到更好的性能.

综上所述,本算法的基本过程是:

(1) 计算从源节点开始,在 k -shortest 搜索过程中,用模拟退火原理选择路径的下一条. 求解过程中,中间节点存储的路径也在不断调整变化.

(2) 温度的降低. 当上一步计算到目的节点后,降低温度,从源节点开始新一轮的计算. 由于很多节点已经存储了上一轮计算得到的路径,在新的计算过程中,将用新的路径替代原来的某些路径. 在这个过程中解得到了优化.

算法中参数 M 和温度下降次数 I 可调整. 较大的 M 和 I 能得到更好的解,同时也增加了开销.

2.2 算法描述

2.2.1 算法 DS-MCP 的伪代码

```
DS-MCP( $G=(V,E), s, t, t_0, I, c_k, k=1,2,\cdots,K, M$ )
 $T = t_0;$ 
for ( $i = 0; i < I; i++$ )
    DS- $k$ -shortest( $G, s, t, M$ );
    if ( $\text{Energy}(t[j]) < 1$ ) return path( $t[j]$ ); //return the path.
 $T = \frac{I-i}{I}t_0;$  //next  $T$ 
return failure;
DS- $k$ -shortest( $G, s, t, M, T$ )
for( $i = 1; i \leq N; i++$ ) //counter = 0, for all nodes
    counter[ $i$ ] = 0;
Energy( $s[1]$ ) = 0;
for( $i = 1; i \leq M; i++$ )
    Energy( $t[i]$ ) = Infinity;
Queue = {path( $s[1]$ )};
While Queue is not empty
    path( $u[j]$ ) = SA_Min(Queue,  $T$ );
```

```

for each node  $v$  in  $u$ 's neighbor
    if ( $v$  is not in  $\text{path}(u[j])$ )
        DS-Relax( $u, v$ );
DS-Relax( $u, v$ )
    Path =  $\text{path}(u[j]) + (u, v)$ ;
    for( $i = 1; i \leq M; i++$ ) //Check if Path is a better
        path;
        if ( $\text{Path} > \text{path}(v[i])$ )
            return;
    if ( $\text{Energy}(\text{Path}) \leq 1$ ) //Energy of  $\text{path}(u[j]) + (u, v)$ 
        If ( $\text{counter}(v) < M$ )
            counter( $v$ ) = counter( $v$ ) + 1;
            path( $v[\text{counter}(v)]$ ) = Path;
            Energy( $v[\text{counter}(v)]$ ) = Energy;
            Queue = Queue + path( $v[\text{counter}(v)]$ );
        else
            path( $v[j]$ ) = path in Node  $v$  with maximum Energy;
            if ( $\text{Energy}(\text{Path}) < \text{Energy}(\text{path}(v[j]))$ )
                path( $v[j]$ ) = Path;
                replace in Queue old path( $v[j]$ ) with new path
                    ( $v[j]$ );
SA-Min(Queue,  $T$ )
 $Z = \sum_{u[j] \in \text{Queue}} \exp(-\text{Energy}(u[j])/T)$ ;
 $x \sim \text{uniform}(0, 1)$ ;
sum = 0;
for each path ( $u[j]$ ) in Queue
    sum = sum +  $\exp(-\text{Energy}(u[j])/T) / Z$ ;
    if (sum >  $x$ )
        Queue = Queue - path( $u[j]$ );
    return path( $u[j]$ );

```

程序中: N 为网络节点数; $G_k[u]$ 为从源点 s 到节点 u 的一个路径上某参数权值之和, 共有 K 个; $\alpha(k=1, 2, \dots, K)$ 为 K 个 QoS 约束; t_0 为初始温度; I 为温度下降总次数; M 为 k -shortest 算法的每个节点存储的路径数, 共 M 个路径; $\text{Path}(u[j])$ 为节点 u 上第 j 个路径, $j=1, 2, \dots, M$, 路径从源节点 s 到 u ; $\text{Counter}[i]$ 为第 i 个节点存储的路径数, 最大值是 M ; $\text{Energy}(u[j])$ 为节点 u 上第 j 个路径的能量。

DS-MCP 是主程序。2~3 行调整退火温度, 多次计算新的路径。在每个温度的计算过程中, 各节点上存储了 M 个路径。这些路径在温度下降后, 被下一次计算更新, 以得到更好的结果。每次计算后, 都判断是否得到了解, 如果成功, 则返回这个路径。

函数 DS- k -shortest 是搜索过程的主体。队列 Queue 中的元素是从源节点 s 到各节点的路径。1~6 行是初始化, 每个节点上可以有 M 个路径, 用 counter[i] 计算节点 i 上已有的路径数目。算法开

始时, 队列 Queue 中是一个空路径 $\text{path}(s[1])$; 7~11 行为 k -shortest 算法搜索整个网络; 第 8 行为用模拟退火算法从队列中提取出某节点上的一个路径。9~11 为行松弛节点 u 的各个邻居。

函数 DS-Relax 是 DS- k -shortest 的松弛函数, 它检查节点 u 的邻居 v 上的新路径。新路径是否可用, 有 2 个条件:

(1) 新路径不能劣于 v 上所有原有路径。在多加性参数情况下, 一个路径劣于另一个路径的条件是, 它的所有参数长度都比另一个路径相应参数长度大。2~4 行检查条件 (1), 其中 $\text{Path} > \text{path}(v[i])$ 表示路径 Path 劣于路径 $\text{path}(v[i])$ 。

(2) 新路径的能量是否小于 1, 如果大于 1, 说明这个路径不可能满足 QoS 请求。第 5 行判断条件 (2)。6~15 行把满足条件的新路径加入到这个节点上, 并放到队列里。在 11~15 行中, 如果节点上已经存满了 M 个路径, 而新的路径比其中能量最大的那一个要小, 则用新路径替代它。

函数 SA-Min 的作用是根据模拟退火原理, 按照一定概率选择非最优路径。第 1 行计算归一化因子, 第 2 行得到一个均匀分布的随机数, 后面几行按波尔兹曼概率分布选择一个路径, 并把这个路径从队列 Queue 中拿出来。

本算法的特点是:

(1) 每个节点存储的路径数越多, 解的质量越好, 最后得到解的个数越多。

(2) 在某个温度下的搜索过程中, 算法可以跳出局部最优, 搜索更大的空间。

(3) 新温度下的搜索过程中, 可以被上一温度的较好搜索结果修正, 从而避免了重复的工作, 使新温度下的计算量不会比上次多。

(4) 在新的温度下, 算法可以跳出上一个温度搜索过的空间。而且由于温度的降低, 新的搜索空间比上一个温度搜索的空间更接近于解空间。

(5) 温度下降总次数越多, 算法得到解的机会越大。但是温度降到很低时, 继续降低温度对改善解的效果也减小, 因为算法会以很大的概率选择局部能量最小的路径, 使新温度下的搜索基本上沿着上次搜索过的顺序进行, 减小了改善解的趋势。

(6) QoS 参数越多, 模拟退火跳出局部最优的效果越明显。

2.2.2 算法 DS-MCP 的复杂度 在网络 $G(V, E)$ 中, 有 K 个 QoS 参数, 网络节点数目是 V , 链路数是 E 。每个节点存储的路径数是 M 。在 DS- k -shortest 中, 队列最大长度可能是 MV , SA-Min 从队列中拿

出一个路径的需要时间是 $\lg(MV)$, 那么队列操作共需要 $MV \lg(MV)$. DS- k -shortest 中处理每个节点的过程中, 每个链路上可能需要处理 M 次, 每次处理 K 个 QoS 参数. 判断一个路径是否可用需要 M 次, 将它加入队列可能需要处理 M 次. 综合起来, 处理所有的链路共需要 $M^3 k E$ 次. 在最坏情况下, 总时间复杂度是 $O(I(MV \lg(MV) + M^3 k E))$. 实际上, 在算法进行过程中, 只要得到解算法就退出, 因此实际运行时间大大减小.

3 算法性能评价

根据上述讨论, DS-MCP 算法的性能受到温度和节点存储路径数的影响. 在仿真实验中, 首先验证这些判断, 然后与目前性能最好的算法中的 H-MCOP^[1] 和 TAMCRA^[2] 比较.

3.1 实验参数和性能评价参数

(1) 拓扑图. 本实验使用的网络拓扑基于 Waxman 模型^[7,8], 分别产生节点数 N 为 50 100 200 400 的网络, 每种情况产生 10 个拓扑图.

(2) 链路参数. 在产生随机网络图后, 采用均匀分布产生每个链路的参数, 范围是 $[1, 100]$. 多个参数间是互不相关的.

(3) QoS 约束. 源-目的节点对 (s, t) 的 QoS 约束的选择很关键, 它决定了实际存在的可行路径的数量. 本实验用两种约束产生方法, 分别是 L_1 和 L_2

$$L_1: L_i = w_i(p), \quad i = 1, 2, \cdots, K$$

路径 p 是用第 1 个参数计算出来的最短路径, 若最短路径有多条, 则在这些路径中取第 2 个参数最小的路径, 以此类推, 以保证路径 p 的惟一性.

$$L_2: L_i = \max_{j=1, 2, \cdots, K} (w_i(p_j)), \quad i = 1, 2, \cdots, K$$

其中, p_j 是用参数 j 算出来的最短路径.

L_1 的条件十分严格, 可行路径是惟一的, 即解是惟一的. L_2 的条件较宽, 至少有 K 个可行路径.

(4) 性能评价参数. 在所有的仿真中, 采用的性能评价参数是成功率 R_s , 即算法能够找到的可行路径的请求数目与模拟的总请求数目的比值. 首先在节点数为 50, 100, 200, 400 的网络上分别随机选取 1 000, 1 500, 2 000, 3 000 个源-目的节点对, 然后对每个网络计算 QoS 路由成功率, 并对 10 个同类拓扑图的成功率取均值.

3.2 温度对 DS-MCP 算法性能的影响

图 1 给出 $K=2$ 约束下, DS-MCP 算法中温度下降次数 I 对性能的影响. 其中, 每个节点存储的路径数固定为 $M=2$. 图 2(a) 为用 L_1 产生 QoS 约束,

图 2(b) 为用 L_2 产生 QoS 约束. 从图 1 中可见, 用 L_1 产生的约束十分严格, 由于只有一个可行解, 网络越大, 找到路径的难度越大. 而用 L_2 产生的约束, 网络越大, 存在的可行路径越多, 找到一个可行解的可能性越大. 另外, 算法性能随着 I 增大而提高.

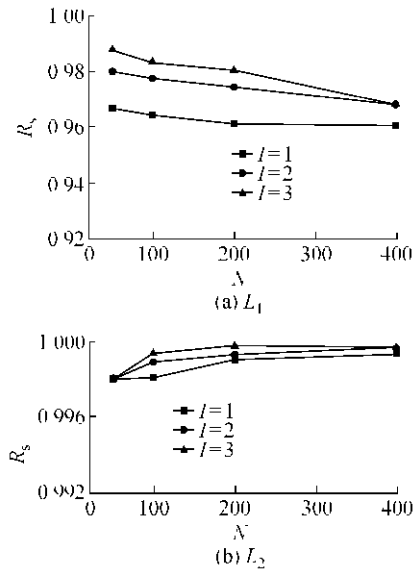


图 1 $K=2$ 约束下 I 对性能的影响
Fig. 1 The performance with different I at $K=2$

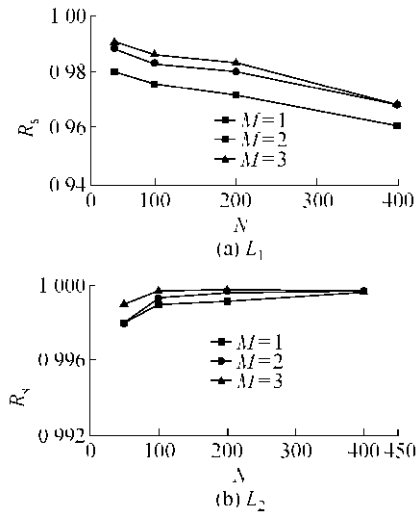


图 2 $K=2$ 约束下 M 对性能的影响
Fig. 2 The performance with different M at $K=2$

3.3 M 对 DS-MCP 算法性能的影响

图 2 给出 $K=2$ 约束下, 节点存储的路径数 M 对性能的影响. 其中 $I=3$. 在图 2(b) 中, $M=3$ 时, 算法性能已经非常高, 接近 100%, 说明 M 取一个较小的值就可以满足性能要求.

3.4 与其他算法的比较

图 3 中, 将本算法与 H-MCOP^[1] 和 TAMCRA^[2] 进行了比较. 实验中, TAMCRA 算法

中参数 $M=2$, DS-MCP 算法中参数为 $M=2, I=3$. 由图 3 可见, DS-MCP 算法在各种规模的网络中的性能是优异的.

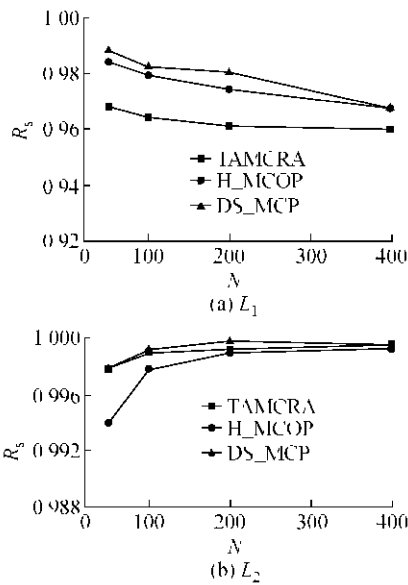


图 3 $K=2$ 约束下算法性能比较

Fig. 3 The performance under different algorithms at $K=2$

3.5 多约束情况下算法的性能

图 4 是在 $K>2$ 的情况下对多个算法性能的比较,网络节点数固定为 100.

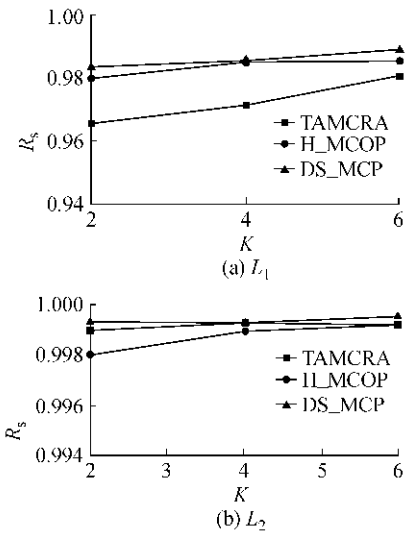


图 4 $K>2$ 约束下算法性能比较

Fig. 4 The performance under different algorithms at $K>2$

从图 4 中可以看出,算法性能并未随着约束个数 K 的增加而下降.一方面,和产生约束的实验方

法有关,约束个数越多,可行路径越多.另一方面,从图 4 可以看出,用本算法设计的非线性函数寻径,算法不受各参数之间关系和约束个数的影响.因此本算法对约束个数具有很好的可扩展性.

4 结 语

本文基于模拟退火和 k -shortest 算法,针对多约束路径选择问题提出一种有效的启发式算法.该算法在迭代过程中,很好地处理了局部最优和全局最优的关系,解决了 k -shortest 算法的局限.实验中证明,可以通过调整算法中的参数 I 和 M 实现不同的精度要求,并且在 I 和 M 很小的情况下即可以得到满意的结果.同时,该算法对网络规模和约束个数具有很好的可扩展性.

参考文献:

[1] Turgay K, Marwan K. Multi-constrained optimal path selection [A]. IEEE Computer Society, IEEE INFOCOMM [C]. Las Vegas, Nevada: Institute of Electrical Electronics Engineers Inc, 2001. 834-843.

[2] Neve H, Miegheem P. TAMCRA: A tunable accuracy multiple constraints routing algorithm [J]. Computer Communications, 2000, 23(11): 667-679.

[3] Chen S, Nahrstedt K. On finding multi-constrained paths [A]. IEEE ICC'98 [C]. Atlanta: Piscataway, 1998. 874-879.

[4] 崔勇,吴建平,徐恪.基于模拟退火的服务质量路由算法 [J].软件学报,2003,14(5): 877-884. CUI Yong, WU Jian-ping, XU Ke. A QoS routing algorithm by applying simulated annealing [J]. Journal of Software, 2003, 14(5): 877-884.

[5] 邢文训,谢金星.现代优化计算方法 [M].北京:清华大学出版社,1999. 90.

[6] Calvert K, Doar M, Zegura E. Modeling internet topology [J]. IEEE Communications Magazine, 1997, 35(6): 157-164.

[7] Chong E, Maddila S, Morley S. On finding single-source single-destination k shortest paths [A]. Seventh International Conference on Computing and Information [C]. Peterborough, Ontario: Canada: Trent University, 1995. 40-47.

[8] Waxman B. Routing of multipoint connections [J]. IEEE Journal on Selected Areas in Communications, 1988, 30(12): 1617-1622.