

基于模拟退火的服务质量路由算法*

崔勇⁺, 吴建平, 徐恪

(清华大学 计算机科学与技术系, 北京 100084)

A QoS Routing Algorithm by Applying Simulated Annealing

CUI Yong⁺, WU Jian-Ping, XU Ke

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

+ Corresponding author: Phn: 86-10-62785822, Fax: 86-10-62788109, E-mail: cy@csnet1.cs.tsinghua.edu.cn

<http://netlab.cs.tsinghua.edu.cn/~cuiy>

Received 2002-06-20; Accepted 2002-09-17

Cui Y, Wu JP, Xu K. A QoS routing algorithm by applying simulated annealing. *Journal of Software*, 2003,14(5):877~884.

<http://www.jos.org.cn/10009-9825/14/877.htm>

Abstract: As a challenging problem of the upcoming next-generation networks, multi-constrained quality-of-service routing (QoSR) is to find a feasible path that satisfies the multiple constraints simultaneously. For the NP complete problem, the heuristic SA_MCP by applying the simulated annealing to Dijkstra's algorithm is proposed. SA_MCP first uses a non-linear energy function to convert multiple QoS weights to a single metric and then seeks to find a feasible path by simulated annealing. This paper overviews the simulated annealing method and analyzes the issues when it is applied to QoSR. Extensive simulations show the following conclusions: (1) SA_MCP has a high performance w.r.t. routing success ratio. (2) It has a good scalability in both network scale and weight number k . (3) It is insensitive to the distribution of QoS constraints. Furthermore, when most QoS requests are feasible, the running time of SA_MCP is about $O(k(m+n\log n))$, which is k times that of the traditional Dijkstra's algorithm.

Key words: simulated annealing; energy function; QoS routing; multiple constraints; scalability

摘要: 作为下一代互联网的核心问题之一,多约束的服务质量路由(QoSR)用来寻找一条同时满足多个约束条件的可行路径。然而,该问题具有NP完全的复杂度。将模拟退火引入多约束QoSR计算中,首先使用非线性能量函数将多个QoS度量转化成单一能量,然后基于模拟退火的方式求解最小能量路径。首先概述了模拟退火的方法,分析了在QoSR中应用模拟退火所面临的关键问题以及解决方案,然后给出了SA_MCP算法及其复杂性分析。实验结果表明,该算法具有很高的性能,同时对网络规模和约束个数都具有很好的扩展性,对QoS约束的分布状况也不敏感。此外,只要大部分QoS约束存在可行路径,算法的实际运行时间约为 $O(k(m+n\log n))$,即传统

* Supported by the National Natural Science Foundation of China under Grant Nos.90104002, 60203025 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant Nos.2002AA103067, 2001AA121013 (国家高技术研究发展计划(863))

第一作者简介: 崔勇(1976—),男,新疆乌鲁木齐人,博士生,主要研究领域为计算机网络体系结构,协议的仿真和测试,多目标优化的路由算法及性能评价。

Dijkstra 算法的 k 倍(k 为约束个数).

关键词: 模拟退火;能量函数;QoS 路由;多约束;可扩展性

中图法分类号: TP301 文献标识码: A

如何为应用提供不同的服务质量(QoS)保证是互连网络面临的一个重要难题^[1],而服务质量路由(QoS)则是其中的一个核心技术和热点问题^[2].QoS 的主要作用是为 QoS 业务寻找能够同时满足多种 QoS 约束的可行路径,虽然最初它是为集成服务模型而设计的,但是在区分服务(diffserv)模型中也需要给不同类型的业务提供不同的路径^[3-5].通常,QoS 限制可分为链路限制和路径限制.其中,链路限制可以转化为对整个路径上瓶颈链路的限制,如带宽;而路径限制则是对组成端到端路径上的所有链路的限制,如延迟.由于对链路限制来说,可以通过剪枝预先除去不符合要求的链路,从而保证在剩余子图中求得的路径符合链路限制.因此,本文主要考虑多重($k \geq 2$)路径限制(也称为多约束)的情况.

由于多约束的 QoS 是 NP 完全问题^[6],为此,研究人员设计了很多启发式算法.然而这些算法往往具有很大的局限性^[2]:(1) 计算复杂度过高,无法应用到实际环境中;(2) 算法性能较差,找不到实际存在的可行路径;(3) 算法只是针对某些特殊情况而设计,不具有普适性.本文则开创性地将模拟退火思想引入多约束 QoS 的计算,设计实现了 SA_MCP 算法.该算法首先使用非线性能量函数将多种 QoS 度量转化成单一能量值,然后再迭代使用 Dijkstra 算法对整个网络计算最短路径树 SPT.其中每次建立 SPT 时,以一定概率 $P(T)$ 选择非最佳路径,并根据所建立的 SPT 对所有节点进行标号,以供随着温度 T 的下降而重新计算 SPT 时使用.当温度逐步下降到 $T \rightarrow 0$ 并保证 $\lim_{T \rightarrow 0} P(T) = 0$ 时,只要网络中存在可行路径,该算法就能够从理论上保证找到一条可行路径.此外,实验也证明了该算法具有很好的性能(即以很大概率找到可行路径).

1 相关工作

由于寻找一条同时满足多种约束条件的路径具有 NP 完全的复杂度,因此人们设计了很多算法试图解决该问题.有关 QoS 算法更详细的讨论可参见文献[2,7].当网络中使用一类加权公平队列(WFQ)(如调度算法^[8])时,队列延迟、抖动、丢失率等都可以转化为带宽的函数而不再彼此独立,这样,原来多约束的 NP 完全问题就可以简化为多项式的复杂度^[9].Orda 对基于这种思想的 QoS 算法做了较为全面的研究^[10].此外,Orda 等人针对多约束的优化问题,通过量化花费函数设计了计算复杂度为 $O\left(\frac{1}{\varepsilon} H m \log C\right)$ 的预计算算法^[11],其中 H 为最长路径的跳数, C 为路径花费的上限.该算法的性能依赖于预先设定的最长路径跳数 H 和小量 ε 等.然而,这类算法一方面需要特定调度算法的支持,另一方面无法涉及到在高速网络中占据重要地位的传输延迟,因此这类算法的应用受到了很大的限制.

为了将多种度量转化为单一度量,基于能量函数的思想也得到了广泛关注. Neve 等人基于非线性函数,为 PNNI 协议设计了求解多约束的路由算法 TAMCRA^[12]及其用于 IP 网络的改进算法 SAMCRA^[13].这些算法首先构造非线性能量函数

$$g_{\lambda}(p) = \left(\frac{w_1(p)}{c_1}\right)^{\lambda} + \dots + \left(\frac{w_k(p)}{c_k}\right)^{\lambda},$$

并针对每个 QoS 请求使用启发式算法试图找到最小化 $g_{\lambda}(p)$ 所对应的路径.由于该花费函数的非线性特点将导致不能保证在多项式时间内找到最小化 $g_{\lambda}(p)$ 的路径 p ,因此算法使用计算 K 最短路径的 Dijkstra 算法的变体,其复杂度为 $O(Kn \log(Kn + K^3 km))$. Korkmaz 等人则设计了一种通过逆向标号过程求解多约束的最小花费路径的算法 H_MCOP^[14].该算法首先基于线性函数 $g_1(p)$,使用逆向 Dijkstra 算法对每个节点进行标号,然后在正向计算过程中基于非线性函数 $g_{\lambda_{\text{rel}}}(p)$,使用 Dijkstra 算法对每个节点及其标号进行评估.

2 背景知识

2.1 问题描述

用有向图 $G(V, E)$ 表示一个网络,其中 V 为节点集,元素 $v \in V$ 称为图 G 的一个顶点(节点); E 为弧集,元素 $e_{ij} \in E$, 记为 $e = v_i \rightarrow v_j$, 称为图 G 的一条边. 在 QoSR 中给每个链路 e 关联上一组相互无关的权值 $(w_1(e), w_2(e), \dots, w_k(e))$, 称为链路 e 的 QoS 度量, 简称为 $w(e)$. 其中 $w_l(e) \in R^+$ 为路径约束类型的度量, 对 $1 \leq l \leq k$. 也就是说, 对于路径 $p = v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_n$, 权值 $w_l(p) = \sum_{i=1}^n w_l(v_{i-1} \rightarrow v_i)$ 满足可加性^[14].

定义 1(多约束路径 MCP). 对于给定的有向图 $G(V, E)$, 包含源节点 s 、目标节点 t 和 $k \geq 2$ 重权值 $w_k(e) \in R^+$ 以及约束向量 $c = (c_1, c_2, \dots, c_k)$, 从 s 到 t 的路径 p 称为多约束路径, 如果 $w_l(p) \leq c_l$, 对 $1 \leq l \leq k$, 简称为 $w(p) \leq c$.

对于给定的 QoS 请求, QoS R 的主要任务就是在当前的网络状态下寻找满足要求的路径 MCP. Dijkstra 给出了单一度量下计算最短路径树 (SPT) 的算法, 具有较低的算法复杂度. 然而 MCP 问题涉及到同时考虑多种度量, 因此导致问题的复杂度为 NP 完全而无法直接使用原有算法. 一种可能的思路是将多种度量转化为单一度量, 以便使用原有算法.

定义 2. 称 $g(p) = \max_{l=1}^k (w_l(p)/c_l)$ 为路径 p 的能量函数, 表示该路径的耗费值, 其中 $c = (c_1, c_2, \dots, c_k)$ 为特定应用业务 QoS 请求的约束条件.

由于 $g(p)$ 为非线性函数, 因此并不能保证在多项式时间内找到一条具有最小能量值的路径. 例如, 在如图 1(a) 所示的网络中寻找从 s 到 t 的最小能量路径. 若使用原有 Dijkstra's 算法, 则建成的 SPT 如图 1(b) 所示, 其中从 s 到 t 的路径为 (sacdt), 该路径的能量为 16. 然而如图 1(c) 所示, $g(\text{sbcdt}) = 12$ 为最小能量值.

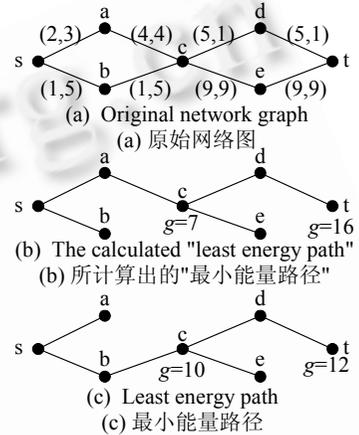


Fig. 1 Finding least energy path

图 1 最小能量路径的求解

2.2 模拟退火思想

由统计力学的研究表明, 在温度 T 下, 物体分子停留在状态 r 的概率满足波尔兹曼分布:

$$\Pr\{\bar{E} = E(r)\} = \frac{1}{Z(T)} \exp\left(-\frac{E(r)}{k_B T}\right), \quad (1)$$

其中, \bar{E} 为分子能量的随机变量, $E(r)$ 为分子在状态 r 所具有的能量, T 为温度, k_B 为波尔兹曼常数, $Z(T)$ 为标准化因子.

退火是一种物理过程: 金属物体加热到一定温度后, 随着温度逐步下降, 物体的分子以一定的概率 \Pr 处在不同的状态, 并满足波尔兹曼分布. 退火过程通常需要满足以下两个条件:

(1) 初始温度 $T=T_0$ 足够大, 使得分子停留在任意状态的概率相近. 若有

$$T_0 \gg E(r)/k_B, \quad (2)$$

则 $E(r)/k_B T_0 \approx 0$, 因此 $\Pr\{\bar{E} = E(r)\} \approx 1/Z(T_0)$, 即分子停留在每个状态的概率相近.

(2) 温度下降到 0 度时, 所有分子将以概率 1 停留在最低能量状态. 我们设分子在最低能量状态 r^* 所具有的能量为 0, 则当 $T \rightarrow 0$ 时有

$$\Pr\{\bar{E} = E(r)\} = \begin{cases} 1, & r = r^* \\ 0, & \text{others} \end{cases}. \quad (3)$$

模拟退火的思想最早由 Metropolis 提出, 并于 1983 年成功地应用在组合优化的问题上. 其思想是, 首先产生一个初始解作为当前解, 然后在当前解的邻域中, 以概率 $P(T)$ 选择一个非局部最优的解, 并令这个解为当前解再重复下去, 从而保证不会陷入局部最优. 在重复的过程中, 随着温度下降, 选择非局部最优解的概率 $P(T)$ 也降低, 直到 $T \rightarrow 0$ 时以 $P=1$ 选择局部的最优解.

模拟退火算法具有 4 个特点:(1) 每个状态都是可能到达的;(2) 在任意给定温度 T 下,分布是稳定的;(3)当温度渐近到 0 度时,模拟退火收敛到全局最优解;(4) 算法的收敛是不依赖于初始条件的.与禁忌搜索的算法相比,模拟退火最大的优点在于从理论上保证能够收敛到全局最优解.因此,为了避免陷入局部优化,从而提高 QoSR 算法的路由成功率(即算法性能),我们将模拟退火引入 QoSR 中.

3 基于模拟退火的启发式算法

3.1 算法思想

由于在路由计算中,只有端到端的一条完整路径才是一个解,因此将模拟退火引入到路由计算时,如何从一个解迭代到其邻域中的另一个解成为主要难题.为此,我们设计了多次迭代计算的方法,从一个解(当前路径)迭代到另一个解时使用 Dijkstra 算法,从而保证每个解都是一条端到端的完整路径.在使用 Dijkstra 算法计算 SPT 时,以概率 $P(T)$ 选择当前非最优的节点加入到部分建成的 SPT 中,从而保证该算法能够克服所有启发式算法都面临的局部优化问题.

设给定的 QoS 请求以 s 为源, t 为目的节点.该算法首先使用 Dijkstra 算法计算以 s 为根的最小跳数 SPT,并对网络中的所有节点标号,然后采用迭代的方法不断使用改进的 Dijkstra 算法对所有节点重新标号.每次重新标号时,都依据上一次的标号并以一定的概率 $P(T)$ 选择非最佳路径,其中 $\lim_{T \rightarrow 0} P(T) = 0$ 满足式(3).每进行一次标号后,温度 T 下降,从而保证随着标号次数的增加有 $T \rightarrow 0$.

3.2 伪码描述

基于上述模拟退火的思想 and Dijkstra 算法,我们针对多重路径约束的 QoSR 问题设计了启发式算法 SA_MCP,见算法 1.

算法 1. SA_MCP 算法伪码描述.

SA_Dijkstra(G, root, T)

1. FOR each node t in G
2. $g[t] = \text{INFINITY}$
3. SPT = {root}
4. $r[\text{root}] = 0$
5. NB = {root's neighbors}
6. WHILE NB is not empty
7. $u = \text{SA_Cheapest}(\text{NB}, T)$
8. addNode(SPT, u , NB)
9. FOR each node v in u 's neighbor
10. IF v is not in SPT THEN
11. SA_Relax(u, v)
12. FOR each node t in G
13. $d[t] = r[t]$

SA_Cheapest(NB, T)

1. $g^* = \min_{v \in \text{NB}} \max_{l=1}^k (r_l[v] + d_l[v]) / c_l$
2. FOR each v in NB
3. $E(v) = \max_{l=1}^k (r_l[v] + d_l[v]) / c_l - g^*$
4. $Z = \sum_{v \in \text{NB}} \exp(-E(v)/T)$

SA_MCP($G=(V,E), s, t, c, T_0, \text{grad}, I$)

1. $T = T_0$
2. Dijkstra(G, s); //get an initial solution
3. IF ($d[t] < c$) RETURN relevant path
4. FOR($i=0; i < I; i++$)
5. SA_Dijkstra(G, t, T); //from node t
6. IF ($d[s] < c$) RETURN relevant path
7. $T = T/\text{grad}$
8. SA_Dijkstra(G, s, T); //from node s
9. IF ($d[t] < c$) RETURN relevant path
10. $T = T/\text{grad}$
11. RETURN failure

5. $x \sim \text{uniform}(0, 1)$

6. sum = 0
7. FOR each u in NB
8. sum = sum + $\exp(-E(u)/T) / Z$
9. IF sum > x THEN RETURN u

- | | |
|---|--|
| addNode(SPT,u,NB)
1. NB=NB- $\{u\}$
2. SPT=SPT+ $\{u\}$
3. FOR each node v in u 's neighbor
4. IF v is not in SPT THEN
5. NB=NB+ $\{v\}$ | SA_Relax(u,v)
1. tmp= $\max_{l=1}^k (r_l[v]+w_l[u,v]+d_l[v])/c_l$
2. IF $g[v]>$ tmp THEN
3. $g[u]=$ tmp
4. $r[v]=r[u]+w(u,v)$
5. $\Pi r[v]=u$ |
|---|--|

算法输入图 G ,给定 QoS 请求的源节点 s ,目标节点 t 和约束向量 c ,其中 $c=(c_1,c_2,\dots,c_k)$ 为 k 重约束.如果从 s 到 t 的正向最小能量路径具有的 K 维度量 $d_l[t]$ 满足约束 c ,则返回该路径,否则拒绝该请求.算法中各个符号表示的意义见表 1.

Table 1 Meanings of the notations in Sa_Mcp

表 1 Sa_Mcp 算法中符号的意义

Symbol	Meanings	Symbol	Meanings
$T0$	Initial temperature	root	The root node for calculating the current SPT
$E(v)$	The energy of node v in formula (1)	$d[u]$	Backward weights: the k -dimensional weights of the path along the old SPT from its root to u
grad	Gradient for decreasing temperature	v	A child node of node u
l	Maximum number of iterations	$\Pi r[v]$	The precedent node of node v
Dijkstra(G,s)	Original Dijkstra's algorithm for SPT rooted by s	$g[u]$	The energy of node u
u	An intermediate node	NB	The set of the neighbors of the current SPT
SA_Dijkstra(G,s,T)	A heuristic for SPT rooted by s based on simulated annealing, where T is the temperature	$r[u]$	Forward weights: the k -dimensional weights of the path along the current SPT from its root to u
Z	The normal factor in formula (1)	g^*	A locally minimal energy
c	k -dimensional constraints of a QoS request	SPT	A partially created SPT

(1) SA_MCP 函数.该函数首先使用原有 Dijkstra 算法,以 s 为源计算最小跳数(Hop)SPT(第 2 行),其中沿着该 SPT 从 s 到 t 的路径为初始解.然后通过模拟退火不断地来回计算新的 SPT(第 4~第 10 行):先以 t 为根计算 SPT(第 5 行),再以 s 为根重新计算 SPT(第 8 行).每次使用 Dijkstra 和 SA_Dijkstra 函数计算出完整的 SPT 后,都对网络中所有的节点 u 更新 $d[u]$,使其保存沿着该 SPT 从 u 到根的能量(详见 SA_Dijkstra 函数中第 12,13 行).每次调用 SA_Dijkstra 函数时,则需要依据上一次更新的 d_l 来重新计算 SPT(详见 SA_Relax 函数第 1 行).此外,每次计算出一个新的 SPT 时,首先判断沿着该 SPT 从 s 到 t 的路径是否满足约束 c ,如果满足,则成功返回这条路径(第 3,6,9 行).每次使用模拟退火的 SA_Dijkstra 函数计算 SPT 时,则需要设定当前温度(第 7,10 行).

(2) SA_Dijkstra 函数.该函数首先是初始化(第 1~第 5 行).然后当 NB 非空时(第 6 行),以模拟退火的方式从 NB 中选择一个节点 u (第 7 行),并把这个节点加入到已经部分建好的 SPT 中(第 8 行),最后再松弛 u 的每个不在 SPT 中的邻居节点(第 9~第 11 行).在建成完整的 SPT 后,将沿着该 SPT 从 root 到达每个节点 t 的路径权值 $r_l[t]$ 保存在全局变量 $d_l[t]$ 中,以供下次重新计算 SPT 时使用(第 12,13 行).

(3) SA_Cheapest 函数.该函数集中体现了模拟退火的核心思想:在一定概率下选择非最优节点,并随着温度 T 的降低,这个概率趋于 0.在函数第 1 行中,

$$\max_{l=1}^k (r_l[v]+d_l[v])/c_l$$

表示如定义 1 所示的路径能量值.其中, $r_l[v]$ 代表了前向度量,即从当前 SPT 的 root 节点到达 v 的第 l 个权值; $d_l[v]$ 代表了后向度量,即从 v 到达此次计算的目标节点(上一次所计算 SPT 的 root 节点)的第 l 个权值,该权值由上一次计算出 SPT 时保存(详见 SA_Dijkstra 函数第 12,13 行).SA_Cheapest 函数首先在当前 SPT 的所有邻居节点中找到最小的能量值 g^* (第 1 行),然后计算模拟退火的能量值 $E(v)$ (第 2,3 行),从而保证最小的 $E(v)=0$.函数接下来计算式(1)中的标准化因子 Z (第 2 行),最后按照式(1)的分布概率选择并返回一个优化节点 u .

(4) addNode 函数.该函数与传统 Dijkstra 算法类似,主要是将 u 加入部分建成的 SPT 时,需要更改邻居节点集 NB,将 u 从 NB 中删除(第 1 行),将 u 的不在 SPT 中的邻居节点加入到 NB 中(第 3~第 5 行).

3.3 算法分析

下面来计算求解 k 约束问题时 SA_MCP 算法的时间复杂度.设在具有 k 种度量的网络 $G(V,E)$ 中, $m=|E|$ 为网

络的边数, $n=|V|$ 为节点数. 由于 Dijkstra 算法改进版本的时间复杂度为 $O(m+n\log n)$, SA_Dijkstra 函数的时间复杂度为 $O(km+kn\log n)$, 所以 SA_MCP 算法总的复杂度为 $O(Ik(m+n\log n))$, 其中 I 为模拟退火的最大迭代次数. 当网络中大部分 QoS 业务的约束条件能够满足时, 由于算法每次迭代前都检查是否已经具有可行路径, 因此能够避免多次迭代而很快找到一个可行路径. 所以上式为最坏情况下的时间复杂度, 而通常状况下的算法运行时间与最大迭代次数关系很小(后面的模拟实验将给出运行时间与迭代次数的关系).

由于模拟退火要求在初始状态下进行, 所以状态的选择具有较大的随机性, 也就是说, 初始温度 $T=T_0$ 足够大, 而在 SA_MCP 算法中, 能量 $E(v)$ 的最大值与 1 具有相近的数量级, 因此我们选择 $T_0=1$, 从而满足式(2). 此外, 模拟退火要求最后的温度 $T \rightarrow 0$, 以便所有的分子都位于能量最低的状态. 因此, 为了使温度能够迅速下降, 我们选择温度等比下降的比值 $\text{grad}=10$, 即迭代 $2I$, 此后 $T(2I)=10^{-2I}$, 从而 $T(2I) \ll E(v)$ 满足式(3). 后续的模拟实验表明, 使用这些参数能够取得很好的效果.

4 算法性能评价

性能评价主要包括两部分内容:(1) 在两重约束情况下评价 QoS 请求约束条件的分布与算法性能之间的关系;(2) 在均匀分布多重约束下的算法性能. 在每种情况下, 我们都将 SA_MCP 与目前所看到的性能最好的算法之一 H_MCOP^[14,15]进行了详细的比较.

在评价上述关系时, 我们基于 N 个节点的完全随机拓扑图^[16], 为每个链路产生了在 $[1,1000]$ 区间内均匀分布的 K 种度量 $w_k(e)$, 且相互无关. 我们分别模拟了网络节点数 N 为 50, 100, 200 和 500 的情况, 并针对每种情况产生了 10 个拓扑图, 在每个拓扑图上随机选取 100 次 QoS 请求的源-目的节点对(一个节点可能被选取多次), 每个源节点 s 使用 SA_MCP 算法计算最小能量路径. 算法评价中, 首先对这 100 次路由计算其成功率 SR(success ratio), 即算法能够找到可行路径的请求数目与所模拟的总请求数目的比值, 然后再通过对 10 个同类拓扑图所对应的 10 个成功率的统计, 得到其均值 SR.

4.1 两重约束下的算法性能

由于 QoSR 算法的性能受约束分布的影响很大, 因此我们在归一化链路度量的基础上, 对给定源-目的对 (s, t) 的 QoS 请求, 采用权重比例仿真法为每个 (s, t) 对产生 QoS 约束. 首先假设每个业务对不同度量 w_i 具有 a_i 的关注程度, 然后使用 Dijkstra 算法找到最小化线性能量函数:

$$g(s, t) = \sum_{i=1}^k a_i w_i(s, t)$$

的路径 $p(s, t)$. 在二维情况下, 我们可以简化为对给定的 $a_1 \in [0, 1]$, $a_2 = 1 - a_1$. 由于网络中不同应用业务对不同度量的关注程度不同, 我们设计了 3 种产生约束的方式:(1) 正态分布(NORMAL): $a_1 \sim \text{normal}(0.5, 0.16)$;(2) 均匀分布(UNIFORM): $a_1 \sim \text{uniform}(0, 1)$;(3) 反正态分布(AB_NORMAL): $a_1 \sim \text{normal}(0, 0.16)$ 且 $a_1 \in [0, 0.5]$. 其中, 为了保证在 99.7% 的概率下 a_1 与均值的偏差不超过 0.5, 我们在(反)正态分布中取标准差为 0.16.

图 2 给出了两重约束下, 模拟退火迭代次数与算法性能的关系以及与 H_MCOP 算法的比较, 其中横坐标为 QoS 约束的产生方式, 而纵坐标为路由成功率 SR, 表示路由算法的性能. 从图示数据可见, 迭代次数过小(如 $I=1$), 则不能取得很好的效果, 这主要是因为 $T_0=1$ 大于能量 $E(v)$, 使得最初的选路中随机性很强, 无法得到较优的路径. 而随着迭代次数的增加, SA_MCP 算法性能迅速提高, 并接近 100%. 这说明在寻找路径的过程中, 模拟退火的引入能够极大地提高算法性能.

H_MCOP 算法在不同的 QoS 约束下体现出不同的性能. 这主要是因为其第 1 次计算 SPT 时, 认为两个权重具有相同的重要性造成的. 当应用业务的约束条件 a_1 与 a_2 的值很接近时(如图示中的 NORMAL 分布), H_MCOP 算法性能较高; 反之, 在图示中的 NORMAL 分布下, H_MCOP 算法性能明显下降. SA_MCP 算法具有很强的适应能力, 在各种分布条件下, 都具有很高的性能. 此外, 随着网络规模的扩大, SA_MCP 算法依然能够保持很高的性能.

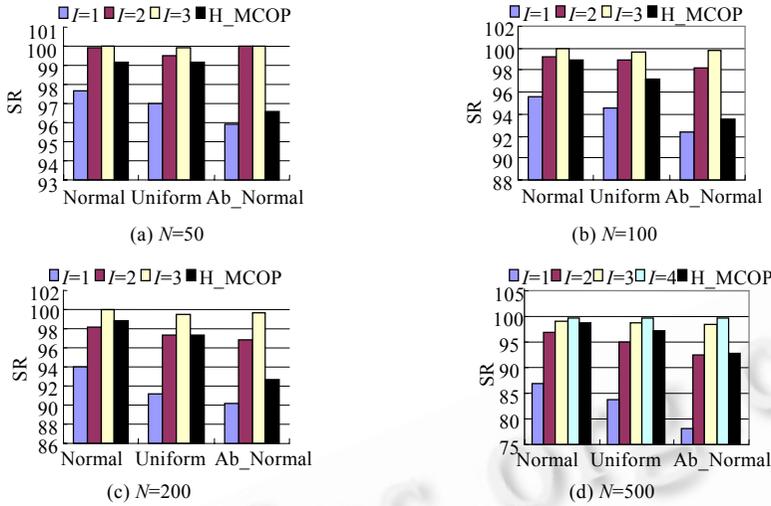


Fig.2 Performance evaluation with two constraints

图 2 两重约束下的算法性能

4.2 多重约束下的算法性能

为了在多约束($k>2$)的情况下研究搜索深度与算法性能之间的关系,我们针对每个源-目的对 (s,t) 产生 QoS 约束条件时,首先取随机数 $a_i \sim uniform(0,1)$,然后令 $a_i = b_i / \sum_{i=1}^k b_i$,从而再以 $g'_i(p) = \sum_{i=1}^k (a_i w_i(p))$ 为关键字构造最小能量路径 p ,然后以度量 $w(p)$ 为 QoS 请求的约束条件,即 $c(s,t) = w(p_i)$.图 3 给出了多约束条件对算法性能的影响,其中横坐标为 QoS 约束的个数 k ,而纵坐标 SR 为路由成功率.

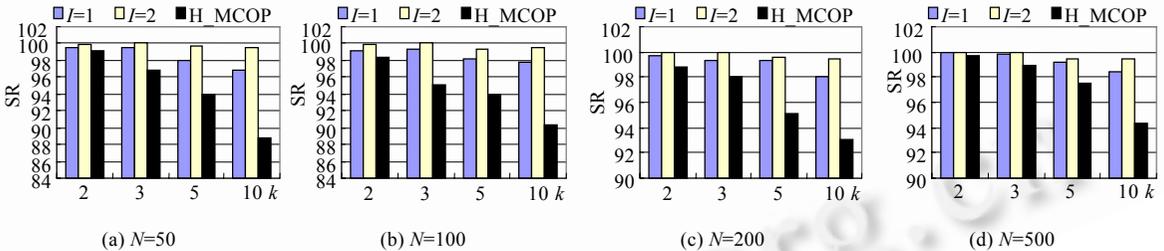


Fig.3 Performance evaluation with multiple constraints

图 3 多重约束下的算法性能

图 3 表明,当采用很小的迭代次数时,SA_MCP 算法就能够获得很高的性能.此外,随着约束个数 k 的增加,H_MCOP 算法性能下降很快,而 SA_MCP 算法依然能够保持很高的性能,并且对 QoS 约束个数的敏感性很小.同时,DF_MCP 算法对多约束下的网络规模也具有有良好的扩展性.

4.3 算法运行时间

为了说明 SA_MCP 算法运行时间与最大迭代次数的关系,我们使用 CPU 为 Pentium III 933 MHz,内存为 256M 字节的计算机统计了算法执行的 CPU 时间.以 $N=500$ 和 $k=10$ 为例,统计结果见表 2.由于算法在大部分情况下一次迭代就能够找到可行路径,只有在第 1 次迭代没有找到可行路径的情况下才进行第 2 次迭代.因此表 2 中的 CPU 时间表明最大迭代次数与算法的实际平均执行时间之间的关系很小,并不是理论上的线性比例关系.这与图 3 的实验结果相吻合,表明只需作一两次迭代就能够为大部分 QoS 请求找到可行路径.

Table 2 Running time of Sa_Mcp
表 2 Sa_Mcp 算法运行时间

Times of iterations	$I=1$	$I=2$	$I=3$	H_MCOP
Running time (ms)	19.54	20.36	21.07	21.90

5 结 论

QoSR 是一个 NP 完全问题,至今没有很好的算法.本文基于模拟退火和 Dijkstra 算法,针对多约束路由问题提出了一种启发式算法 SA_MCP.我们首先概述了模拟退火的基本思想,然后分析了将这种思想应用在路由领域所面临的主要问题以及相应的解决思路,进而给出了 SA_MCP 算法.该算法以最小跳数路径作为模拟退火的初始解,并使用该初始解对所有节点进行标号;然后不断迭代求解可行路径并对所有节点标号,而每次迭代过程中都使用上一次的标号来求解最小化非线性能量函数的 SPT.通过广泛而深入的实验,结果表明,SA_MCP 算法具有很高的性能,同时对网络规模(m,n)和约束个数(k)都具有很好的扩展性,对 QoS 约束的分布状况也不敏感.此外,虽然算法在最坏情况下的时间复杂度是 $O(k(m+n\log n))$,与最大迭代次数 I 成正比,但只要大部分 QoS 约束存在可行路径,则算法的实际运行时间约为 $O(k(m+n\log n))$,而与 I 无关.

References:

- [1] Xiao XP, Ni LM. Internet QoS: a big picture. *IEEE Network*, 1999,13(2):8~18.
- [2] Cui Y, Wu, JP, Xu K, Xu MW. Research on internetwork QoS routing algorithms: a survey. *Journal of Software*, 2002,13(11): 2065~2075 (in Chinese with English abstract).
- [3] de Oliveira JC, Scoglio C, Akyildiz IF, Uhl G. A new preemption policy for diffserv-aware traffic engineering to minimize rerouting. In: Lee D, Orda A, eds. *Proceedings of the IEEE INFOCOM 2002*. New York: IEEE Communication Society, 2002. 695~704.
- [4] Wang J, Nahrstedt K. Hop-by-Hop routing algorithms for premium-class traffic in diffserv networks. In: Lee D, Orda A, eds. *Proceedings of the IEEE INFOCOM 2002*. New York: IEEE Communication Society, 2002. 704~714.
- [5] IETF integrated services (diffserv) working group. <http://www.ietf.org/html.charters/diffserv-charter.html>.
- [6] Wang Z, Crowcroft J. Quality-of-Service routing for supporting multimedia applications. *IEEE Journal on Selected Areas in Communications*, 1996,14(7):148~154.
- [7] Chen SG, Nahrstedt K. An overview of quality-of-service routing for next-generation high-speed networks: problems and solutions. *IEEE Network*, 1998,12(6):64~79.
- [8] Bennett JCR, Zhang H. Hierarchical packet fair queuing algorithms. *ACM Computer Communication Review*, 1996,26(4): 143~156.
- [9] Pornavalai C, Chakraborty G, Shiratori N. QoS based routing algorithm in integrated services packet networks. *Journal of High Speed Networks*, 1998,7(2):99~112.
- [10] Orda A. Routing with end-to-end QoS guarantees in broadband networks. *IEEE/ACM Transactions on Networking*, 1999,7(3): 365~374.
- [11] Orda A, Sprintson A. QoS routing: the precomputation perspective. In: Sidi M, ed. *Proceedings of the IEEE INFOCOM 2000*. Israel: IEEE Communication Society, 2000. 128~136.
- [12] De Neve H, Van Mieghem PV. A multiple quality of service routing algorithm for PNNI. In: *Proceedings of the IEEE ATM Workshop*. IEEE Communication Society, 1998. 324~328.
- [13] Van Mieghem P, De Neve H, Kuipers F. Hop-by-Hop quality of service routing. *Computer Networks*, 2001,37:407~423.
- [14] Korkmaz T, Krunz M. Multi-Constrained optimal path selection. In: Sengupta B, ed. *Proceedings of the IEEE INFOCOM 2001*. Piscataway, NJ: IEEE Communication Society, 2001. 834~843.
- [15] Feng G, Douligeris C, Makki K, Pissinou N. Performance evaluation of delay-constrained least-cost QoS routing algorithms based on linear and nonlinear Lagrange relaxation. In: Veeraraghavan M, ed. *Proceedings of the IEEE ICC 2002*. New York: IEEE Communication Society, 2002. 2273~2279.
- [16] Zegura EW, Calvert KL, Donahoo MJ. A quantitative comparison of graph-based models for Internet topology. *IEEE/ACM Transactions on Networking*, 1997,5(6):770~783.

附中文参考文献:

- [2] 崔勇,吴建平,徐格,徐明伟.互网络服务质量路由算法研究综述. *软件学报*,2003,13(11):2065~2075.