QoS 路由近似算法的研究*)

张信明 陈国良 黄刘生 徐 云
(中国科技大学计算机科学技术系 国家高性能计算中心 合肥230027)

Research on the Approximate Algorithms for QoS Routing

ZHANG Xin-Ming CHEN Guo-Liang Huang Liu-Sheng XU Yun
(Department of Computer Science and Technology, University of Science and Technology of China, National High Performance Computing Center, Hefei 230027)

Abstract In general, finding a feasible route subject to multiple additive QoS constraints is an NP-complete problem. In this paper, we first analyze the constraints and their properties that can reflect the basic characteristics of a network and support the fundamental QoS requirements. Then, we show that the unicast QoS routing problem can be generalized as a Multiple Constrained Path(MCP) or Restricted Shortest Path(RSP) problem. Finally, We investigate the efficient approximate algorithms for MCP and RSP, and how to convert a RSP problem to an easier MCP problem.

Keywords QoS routing, Approximate algorithm, MCP, RSP, NPC

QoS 需求可分为基于路径的与基于链路的两种。称一个具有足够资源来满足应用 QoS 需求的路径为可行路径。另外，优化准则可以进一步缩小在可行路径间的选择范围。QoS 路由的目标是寻找最优的可行路径。

基于 QoS 的路由是当前的一个研究热点[1~15]，其目标是在确定满足多个约束条件(QoS 需求)的路径的同时实现网络资源的高利用率。基于 QoS 的路由对资源预置的服务(如 IntServ, ATM)与非资源预置的服务(如 DiffServ)都是需要的。如，ATM PNNI(Private Network Node Interface)协议基于 QoS 路由是通过源节点为连接需求决定适宜的路径来实现的；DiffServ 的基于 QoS 路由可由网络管理者以流量工程为目的进行；基于 QoS 的路由也可用来保证聚集流的 SLA (service level agreement)。选路的两个基本任务是：①分发网络状态信息；②根据网络状态信息，寻求满足给定约束条件的可行路径。

1 QoS 路由的约束条件

一般的 QoS 路由问题包含多个约束条件，如时延、时延抖动、分组丢失率、带宽与成本等[1]。

1.1 约束条件选择准则

选路约束条件代表了网络的特征；因此，它们是决定 QoS 路由算法的复杂性、可支持 QoS 保证的范围的主要因素。需要考虑的因素主要有：①对任何选择的约束条件而言，必须存在计算路径的有效算法以保证路由协议能够扩展至 Internet 等大型网络；②约束条件必须反映网络的基本特性；约束条件应当包含支持基本 QoS 需求的信息；③约束条件之间必须是相互正交的以保证约束条件不包含冗余信息。

1.2 约束条件的性质

QoS 约束条件分为可加的、可乘的与瓶颈的三种，即 $c_p=\Sigma_{e\in p}c_e$、$c_p=\prod_{e\in p}c_e$、$c_p=\min_{e\in p}c_e$(e 表示链路，p 表示路径，c 表示约束条件)。时延、时延抖动、成本为可加约束条件，分组丢失率为可乘约束条件，而带宽为瓶颈约束条件。通过在式 $c_p=\prod_{e\in p}c_e$ 两边取对数可将可乘的约束条件化为可加的约束条件。

1.3 约束条件的选取

多个约束条件肯定可以更准确地模型化一个网络；然而问题是寻找受多个条件约束的路径是非常困难的，解决此问题的多项式算法可能不存在。仅有两个约束条件的被称为"最短权重约束路径"的简单问题为 NPC 问题。时延和带宽是实时多媒体传输必须保证的重要约束条件；时延抖动约束可以通过在接收端采用缓冲技术来解决；现在多数网络多媒体应用都附带错误恢复功能，能够容忍一定范围的分组丢失率；成本是评价网络使用效率的重要因素。故在设计 QoS 路由算法时主要考虑时延、带宽和成本三个约束条件。通过修剪不满足 QoS 需求的所有链路，瓶颈约束条件可作为预处理步骤来处理，这样就可以仅考虑可加约束条件时延与成本。

2 MCP 与 RSP 问题

MCP(Multiple Constrained Path)[15]问题：网络以有向图 $G=(V,E)$表示，V、E 分别是节点、链路集合。每一个链路 $e=(u,v)\in E$ 与两个非负可加 QoS 参数 $c_1(e)$、$c_2(e)$相联系。给定约束条件 δ_1、δ_2，MCP 问题是从源节点 s 到目标节点 d 寻找满足 $c_1(p)\leqslant\delta_1$、$c_2(p)\leqslant\delta_2$的路径 p。其中 $c_1(p)=\Sigma_{e\in p}c_1(e)$，$c_2(p)=\Sigma_{e\in p}c_2(e)$。

RSP(Restricted Shortest Path)[14]问题：网络以有向图 $G=(V,E)$表示，V、E 分别是节点、链路集合。每一个链路 $e=(u,v)\in E$ 与两个非负可加 QoS 参数、$c_1(e)$、$c_2(e)$相联系。给定约束条件 δ_2，RSP 问题是从源节点 s 到目标节点 t 寻找满足$\min_p c_1(p)$、$c_2(p)\leqslant\delta_2$的路径 p。

MCP、RSP 问题系 NPC 问题，即不存在有效的多项式算

*)本课题的研究得到国家重点基础发展规划973(G1998030403)和中国科学院高水平大学建设(KYZ2706)项目的资助。**张信明** 博士，副教授，主要研究领域为计算机网络与操作系统。**陈国良** 教授，博士生导师，主要研究领域为并行算法及其应用，并行计算机体系结构。**黄刘生** 教授，博士生导师，主要研究领域为高性能计算。**徐 云** 博士，副教授，主要研究领域为并行与随机算法。

法。任何解决 RSP 问题的方法均可应用于 MCP 问题。MCP 与 RSP 问题均可通过伪多项式算法解决，算法的复杂性依赖于链路权重(如最大链路权重)及网络规模。当链路权重、网络规模较大时算法的计算代价较大。

3 MCP 问题近似算法[11]

3.1 层次最短路径算法

下面给出在组合权重 $l(e)=\alpha w_1(e)+\beta w_2(e)$ 下使用的层次 Dijkstra 最短路径算法。除了寻找在 $l(e)$ 条件下的最短路径，层次最短路径算法还确定在所有最短路径中 $w_1(\cdot)$、$w_2(\cdot)$ 的最小值。为实现上述任务，就需要对标准 Dijkstra 算法的松驰过程进行修改；层次算法比标准算法增加了表示不同权重下的最短路径的成本 $w_1[u]$、$w_2[u]$，所有最短路径中 w_1、w_2 的最小值 min_$w_1[u]$、min_$w_2[u]$。

```
Relax (u,v)
if (d[v]>d[u]+l(u,v)) then
    d[v]:=d[u]+l(u,v)
    π[v]:=u
    w1[v]:=w1[u]+w1(u,v)
    w2[v]:=w2[u]+w2(u,v)
    min_w1[v]:=w1[v]
    min_w2[v]:=w2[v]
else if (d[v]=d[u]+l(u,v))then
    if(min_w1[v]>min_w1[u]+w1(u,v))then
        min_w1[v]:=min_w1[u]+w1(u,v)
     if(min_w2[v]>min_w2[u]+w2(u,v))then
        min_w2[v]:=min_w2[u]+w2(u,v)
```

3.2 MCP 问题的基本近似算法

```
MCP-APPROX(G(V,E),s,t,c1,c2)
/* Find a path p from s to t in the network G=(V,E) */
/* such that w1(p)=w1[t]≤c2 and w2(p)=w2[t]≤c2 */
  l(e)=w1(e)+w2(e)(∀ e∈E)
Execute hierarchical Dijkstra's algorithm with link weights {l(e):
e∈E}
if (w1[t]≤c1 and w2[t]≤c2)then
    return SUCCESS
if (w1[t]>c1 and w2[t]>c2)then
    return FAILURE
if (min_w1[t]≤c1 and min_w2[t]≤c2)then
    return FAILURE
if (min_w2[t]≤c2)then
    Execute Binary_Search(i=1,j=2)
 else if (min_w1[t]≤c1)then
    Execute Binary_Search(i=2,j=1)
Binary_Search(i,j)
  k_min=1
  k_max=n·max{wj(e)|e∈E}
while (k_min<=k_max)do
    k=⌈√(k_min * k_max)⌉
    l(e)=k wi(e)+wj(e)(∀ e∈E)
    Execute hierarchical Dijkstra's algorithm with link weights {l
(e):e∈E}
    if (w1[t]≤c1 and w2[t]≤c2) then
       return SUCCESS
    if (min_wj[t]≤cj)then
       k_min=k+1
   else
       k_max=k-1
```

4 RSP 问题近似算法

4.1 PATH 与 PARTITION 问题

对于节点数为 n、链路数为 m 的网络，令 $c_e(d)$ 为链路 e 在时延 d 下所需成本；PATH 代表端到端(源节点 s 至目的节点 t)时延约束条件为 D 时，如何选择一个路径 p、确定路径上各链路的时延限制并使各链路的成本之和最小化；PARTITION 代表将 PATH 的基于路径问题映射到基于链路问题，即当路径 p($e(p)$个链路)选定后，如何确定路径 p 上各链路的时延限制并使各链路的成本之和最小化。

对 PATH 与 PARTITION 的近似算法是基于对 RSP 问题的近似。RSP[12,14]是一个受限的 PATH 问题(每个链路具有给定的成本及时延)。

定理1 RSP 问题具有复杂性为 $O(\frac{1}{\varepsilon}mn\log\log U)$ 的 ε-近似算法(U 为最优解 OPT 的上界)。

定理2 PATH 问题具有复杂性为 $O(X\frac{1}{\varepsilon}mn\log\log U)$ 的 ε-近似算法($X=\min[D,\frac{\log C}{\varepsilon}+\log D,\frac{n}{\varepsilon}+\log D]$)。

定理3 PARTITION 问题具有复杂性为 $O(Y\frac{1}{\varepsilon}e^2(p)\log\log U)$ 的 ε-近似算法($Y=\min[D,\frac{\log C}{\varepsilon}+\log D,\frac{e(p)}{\varepsilon}+\log D]$)。

4.2 算法

算法思路：假定存在一个可用来决定 OPT 是否大于给定值 V 的精确测试过程，则就从 OPT 的上界 U 开始并使用上述过程进行二分搜索以获得 OPT 的精确值。

4.2.1 精确测试过程

```
EXACT
1 initialization
2    for all c≥0,set gs(c)=0
3    for all v≠s,set gv(0)=∞
4        for c=1,2,3,Λ
5    for all e,compute de(c)
6    for all v≠s,set gv(c)=min_{b:b<c} min_{u:uv∈E}[gu(b)+duv(c-b)]
7    if (gt(c)≤D)then
8        output OPT=c and its corresponding path, exit.
```

算法采用动态规划技术，$g_v(c)$是合计成本为 c 的 s-v 的最小约束条件路径。OPT 是时延最多为 D 的 s-t 路径的最小时延，即 $g_t(OPT)\leqslant D$ 且对任何 $c<OPT$，$g_t(c)>D$；所以使 $g_t\leqslant D$ 的最小 c 为 OPT。行5的运行时间为 $O(m\log D)$，行6的运行时间为 $O(mc)$，行4的循环次数为 OPT，所以 EXACT 的运行时间为 $O(mOPT\log D+mOPT^2)$。

4.2.2 近似测试过程

```
TEST(V)
1 initialization
2      for all c≥0, set gs(c)=0
3      for all v≠s, set gv(0)=∞
4 for c=1,2,…,⌊n/ε⌋
5      for all e, compute d̂e(c)
6      for all v≠s, set gv(c)=min_{b:b<c} min_{u:uv∈E}[gu(b)+d̂uv(c-
          b)]
7      if (gt(c)≤D) then
8         output OPT≤(1+ε)V,exit
9 output OPT≥V
```

TEST 是采用 RAS(rounding-and-scaling)技术的一个多项式时间的 ε-测试过程。TEST 与 EXACT 的区别有两点：成本 $c_e(d)$依因子 $V\varepsilon/n$ 按比例缩小；行4的 for 循环中 $c=1,2,\cdots,\lceil n/\varepsilon\rceil$。$\hat{c}_e(d)$、$\hat{d}_e(c)$分别表示按比例缩小的成本与其反函数。$\hat{c}_e(d)=\lfloor\frac{c_e(d)}{V\varepsilon/n}\rfloor$，$\hat{d}_e(c)$表示在成本函数 $\hat{c}$ 下在链路 e 产生的成本最多为 c 的最小时延。$\hat{c}_e(\cdot)$与 $\hat{d}_e(\cdot)$的计算时间分别为 $O(1)$、$O(\log D)$。

引理1 TEST 是一个 ε-近似测试，其运行时间为 $O(\frac{n}{\varepsilon}m\log D+\frac{n^2}{\varepsilon^2}m)$。

若 TEST 输出 $OPT\leqslant(1+\varepsilon)V$，则 TEST 发现了成本函数 $\hat{c}_e(d)$下的时延最多为 D、成本最多为 n/ε 的路径。令此路径为 P，下面求 P 在原成本函数 $c_e(\cdot)$下的成本。根据 $\hat{c}_e(\cdot)+1=\lfloor\frac{c_e(\cdot)}{V\varepsilon/n}\rfloor+1>\frac{c_e(\cdot)}{V\varepsilon/n}$ 得 P 在成本函数 $c_e(\cdot)$下的成本 $\Sigma_{e\in P}c_e(\cdot)<\Sigma_{e\in P}(\hat{c}_e(\cdot)+1)V\varepsilon/n=V\varepsilon/n(\Sigma_{e\in P}\hat{c}_e(\cdot)+\Sigma_{e\in P}1)=V\varepsilon/n$($\hat{c}_e(\cdot)$下 P 的费用$+P$ 的链路数)$\leqslant V(\varepsilon/n)(n/\varepsilon+n)=(1+\varepsilon)V$。由于 OPT 受 P 成本的限制，所以 OPT 最多为$(1+\varepsilon)V$。

若 TEST 输出 $OPT \geqslant V$，则对所有成本 $\hat{c}_e(\cdot) \leqslant \lfloor n/\varepsilon \rfloor$ 的路径条件 $g_e(c) \leqslant D$ 均不成立。在成本函数 $\hat{c}_e(\cdot)$ 下时延最多为 D 的每个路径成本至少为 n/ε。所以在成本函数 $c_e(\cdot)$ 下所有这些路径的成本至少为 $(n/\varepsilon)(V\varepsilon/n)=V$，即 OPT 至少为 V。运行时间的获得方法与 EXACT 类似。

4.2.3 近似算法

```
APPROX
1  initialization
2      set L=1
3      set U=nC
4  while (U>2L) do
5      set V=√(U·L)
6      if (TEST(V)) outputs OPT≥V, set L=V
7      if (TEST(V))outputs OPT≤(1+ε)V, set U=(1+ε)V
8  set ĉ_e(d)=⌊c_e(d)/(Lε/n)⌋
9  call EXACT with cost ĉ_e(d)
```

引理2 APPROX 是一个运行时间为 $O((\frac{n}{\varepsilon}+\log D)\frac{1}{\varepsilon}mn\log\log U)$ 的 ε-近似算法。

令 P 为在成本函数 $c_e(\cdot)$ 下的成本为 OPT 的最优路径，$\hat{P}$ 为按比例缩小成本 $\hat{c}_e(\cdot)$ 下由 EXACT 得到的路径。由 $\hat{P}$ 为成本 $\hat{c}_e(\cdot)$ 下最优得 $\Sigma_{e\in \hat{P}}\hat{c}_e(\cdot) \leqslant \Sigma_{e\in P}\hat{c}_e(\cdot)$ (1)；而 $\Sigma_{e\in P}\hat{c}_e(\cdot) = \Sigma_{e\in P}\lfloor \frac{c_e(\cdot)}{L\varepsilon/n} \rfloor \leqslant \Sigma_{e\in P}\frac{c_e(\cdot)}{L\varepsilon/n} = \frac{OPT}{L\varepsilon/n}$ (2)，$\Sigma_{e\in \hat{P}}\hat{c}_e(\cdot) = \Sigma_{e\in \hat{P}}\lfloor \frac{c_e(\cdot)}{L\varepsilon/n} \rfloor \geqslant \Sigma_{e\in \hat{P}}(\frac{c_e(\cdot)}{L\varepsilon/n}-1) \geqslant \Sigma_{e\in \hat{P}}\frac{c_e(\cdot)}{L\varepsilon/n}-n$ (3)，由(1)、(2)得 $\Sigma_{e\in \hat{P}}\hat{c}_e(\cdot) \leqslant \frac{OPT}{L\varepsilon/n}$ (4)，由(3)、(4)得 $\Sigma_{e\in \hat{P}}\frac{c_e(\cdot)}{L\varepsilon/n}-n \leqslant \frac{OPT}{L\varepsilon/n}$，所以 $c_e(\cdot)$ 下路径 $\hat{P}$ 的成本为 $\Sigma_{e\in \hat{P}}c_e(\cdot) \leqslant OPT+L\varepsilon$。由于 L 是 OPT 的下界，故 $c_e(\cdot)$ 下路径 $\hat{P}$ 的成本最多为 $OPT+OPT\varepsilon=(1+\varepsilon)OPT$。

APPROX 的运行包括行9的 EXACT 运行时间以及 while 循环中的 TEST 运行时间。而 EXACT 的 for 循环的执行次数为 $\Sigma_{e\in \hat{P}}\hat{c}_e(\cdot)$，由(4)得 $\Sigma_{e\in \hat{P}}\hat{c}_e(\cdot) \leqslant \frac{OPT}{L\varepsilon/n}$，又 EXACT 是在 $OPT \leqslant U \leqslant 2L$ 情况下被执行的，即 $\Sigma_{e\in \hat{P}}\hat{c}_e(\cdot) \leqslant \frac{OPT}{L\varepsilon/n} \leqslant 2n/\varepsilon$，所以 EXACT 的运行时间为 $O(\frac{n}{\varepsilon}m\log D+\frac{n^2}{\varepsilon^3}m)$。TEST 被执行 $\log\log U$ 次，每次执行的时间为 $O(\frac{n}{\varepsilon}m\log D+\frac{n^2}{\varepsilon^2}m)$。故 APPROX 的运行时间为 $O((\frac{n}{\varepsilon}+\log D)\frac{1}{\varepsilon}mn\log\log U)$。

5 搜索空间的降低 SSR

由于 MCP 问题比 RSP 问题容易一些，所以一般说来解决 MCP 问题的启发式算法在运行时间上更有效，那么若将 RSP 问题转换为 MCP 问题亦是很有意义的。DCCR(for Delay-Cost-Constrained Routing)[6]首先根据网络状态引入一个成本界限，然后在一个新的非线性路径时延、成本权重函数下应用 k 最短路径算法来有效地寻找受时延、成本约束的路径。由于不能同时满足时延、成本约束的路径被修剪掉，所以搜索空间得到了降低。与 TAMCRA(解决 MCP 问题的算法)相似，权重函数更倾向于较低成本的路径。由于使用紧一些的成本界限可以提高算法的准确性与速度，所以作为一个改进，使用了由 Bolkh/Gutin 的算法优化了搜索空间。通过分析，算法的复杂性的渐近性能在数量级上与单约束条件最短路径算法(如 Dijkstra 算法)相同。下面介绍 SSR+DCCR 算法。

通过为 MCP 问题定义一个适宜的成本界限将 RSP 问题转化为 MCP 问题并保证 RSP 的解依然是 MCP 问题的一个可行解。此目标可通过使用足够松的成本界限来实现。算法的起点是以最大时延路径作为 MCP 问题的一个可行解，最大时延路径的成本作为成本界限。若不存在更小成本的可行路径，则最大时延路径必然是最优路径。因此，就可将 RSP 问题转化为在一个新 MCP 问题的解空间中搜索接近最优的解。由于同时满足时延、成本约束的路径相对较少，所以 MCP 问题的解空间显然要比 RSP 问题的解空间要小。为了搜索最小成本的解空间，需要检查 MCP 问题的可行路径(即同时满足时延、成本约束的路径)。对于上述目标我们可以使用任何已知的最短路径算法(如 Dijkstra，Bellman-Ford 等)；但是由于最短路径算法只有一个约束条件，就需要定义一个可以融合所有链路约束条件的权重函数，这样最终可以发现一个能同时最小化所有链路约束条件的解。一种混合约束条件的简单方法是使用线性函数(如 $w(e)=\alpha c(e)+\beta d(e)$)作为边的新权重。该方法的优点在于将对路径的多个约束条件(时延、成本)转化为单一的路径权重($\delta=\delta_c+\beta\delta_d$)约束，因而易于实现。然而，线性权重函数不能完全反映路径的实际特性；这就会出现根据新的权重函数获得的最优路径可能违反原有的两个约束条件，而次最优解反倒可以满足原有的两个约束条件。使用非线性函数可以克服上述问题，Neve/Miegeham 在 TAMCRA[13]算法中提出凹函数 $\max\{c(p)/\delta_c, d(p)/\delta_d\}$。使用非线性权重函数的一个问题是路径的权重与路径中所有链路权重的和不相等，即 $W(p) \neq \Sigma_{e\in p}w(e)$。由于记录路径的累积时延与累积成本是较容易的，所以可以通过路径时延与成本的函数(即 $W(p)=f(c(p),d(p))$，而不是链路权重的和)计算路径权重来解决上述问题。使用非线性权重函数的一个更严重的问题是非线性函数不具备最优子结构性质，即最优路径的一部分不一定是最短路径。因此，类似 Dijkstra 的最短路径算法有时找不到最小权重路径。解决上述问题的一种方法是采用 k 最短路径算法。在寻找离所有界限很远的路径方面 TAMCRA 的非线性(最大)权重函数是可行的，但 TAMCRA 并不优化任何约束条件。最大权重函数在处理时延、成本约束条件时并无偏好，这对于寻找具有最小成本路径的问题来说是不合适的。

小结 确定满足一定约束条件的可行路径是提供端到端 QoS 保证的基础，而寻找受限于多个可加约束条件的路径问题为 NPC 问题。本文首先研究了能够反映网络基本特性并包含支持基本 QoS 需求信息的选路约束条件及其性质，然后将单播 QoS 路由问题抽象为 MCP 与 RSP 问题并给出解决 MCP、RSP 问题的高效近似算法，最后研究了如何将 RSP 问题转化为相对简单的 MCP 问题。组播 QoS 路由[2]所涉及的树优化问题(使组播树的总计成本最小化，即 Steiner 树问题)、带宽受限的 Steiner 树问题(约束条件属瓶颈性质)、延迟受限的 Steiner 树问题(约束条件属可加或可乘性质)、带宽与延迟均受限的 Steiner 问题等均是 NPC 问题。

参考文献

1 Wang Z, Crowcroft J. Quality of service routing for supporting multimedia applications. IEEE Journal on Selected Areas in Communications, 1996, 14(7):1228～1234

2 Wang Bin, Hou J C. Multicast routing and its QoS extension: problems, algorithms, and protocols. IEEE Network, 2000 (January/February):22～36

3 Widyono R. The design and evaluation of routing algorithms for real-time channels:[Technical Report TR-94-024]. University of California at Berkeley, June 1994

(下转第96页)

体沿 X、Y、Z 轴三个方向的边均5等分，从而得到了6×6×6=216个椭球图标。可以看出，采用的图标离散地表示了实体中的应力分布情况，但图标不能太多，太多的图标造成视觉上的混乱，可视化也就失去了意义。

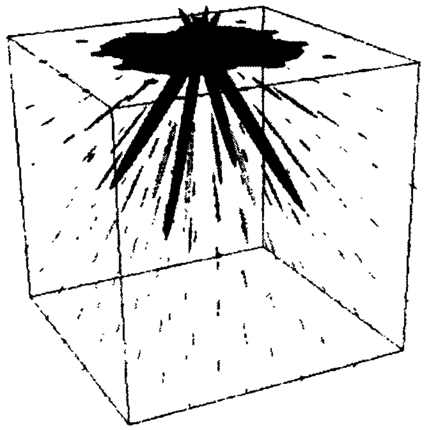

图2 椭圆图标可视化应力场

超流线可视化应力场的算法如下：

1)对离散点集进行 Delaunay 三角剖分，构造它们之间的拓扑关系；

2)指定超流线的起始点；

3)判断点所在的四面体；

4)插值计算出所要显示图标的点的九个分量值，采用体积加权的方法，对每个分量单独进行插值；

5)将插值得到的九个张量分量写成一个3×3的矩阵，得到一个实对称矩阵，采用 Jacobi 方法计算该矩阵的三个特征值和特征向量；

6)根据计算出的特征值和特征向量将除纵向特征向量外的另两个特征向量映射为椭圆或十字叉；

7)根据选取的纵向特征向量，采用 Runge-Kutta 方法计算超流线轨迹线上的下一个点；

8)若该点在实体内或轨迹线的长度小于给定长度，转3)；否则，继续；

9)根据超流线轨迹线上点处的椭圆或十字叉构造管或螺旋线，即为超流线。

图3是用超流线可视化应力场的效果图。

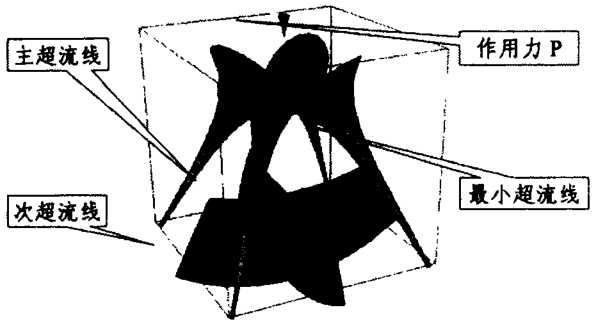

图3 应力场中的超流线

结论 我们以弹性力学中的 Boussinesq 问题为例，采用椭球图标和超流线对该问题的结果数据进行了可视化显示。可以看出，可视化结果比较合理地反映了实体中的三个主应力及剪切应力分布的情况。但由于张量的复杂性，对于复杂拓扑结构的实体的非对称张量场以及高阶张量场的可视化研究目前仍然十分有限。随着国内外对这方面研究的不断深入，相信在不久的将来，一定会取得更多的突破性进展。

参考文献

1 石教英，蔡文立. 科学计算可视化算法与系统. 科学出版社，1996

2 Sadda A S. Elasticity Theory and Applications. Pergamon Press Inc. , New York, NY, 1974

3 蒋尔雄，高坤敏，吴景琨. 线性代数. 人民教育出版社，1978

4 Delmarcells T, Hesselink L. Visualizing Second Order Tensor Fields with Hyperstreamlines. IEEE CG&A, 1993, 13(4)

(上接第107页)

4 Salama H F, Reeves D S, Viniotis Y. A distributed algorithm for delay-constrained unicast routing. In: Proc. of IEEE INFOCOM'97, 1997. 84～91

5 Chen S, Nahrstedt K. On finding multi-constrained paths. In: Proc. of IEEE ICC'98, 1998. 874～879

6 Guo L, Matta I. Search space reduction in QoS routing. In: Proc. of the 19th IEEE Intl. Conf. on Distributed Computing Systems, 1999. 142～149

7 Ma Q, Steenkiste P. Routing traffic with quality-of-service guarantees in integrated services networks. In: Proc. of NOSSDAV'98, 1998

8 Guerin R, Orda A. QoS routing in networks with inaccurate information: theory and algorithms. In: Proc. of IEEE INFOCOM'97, 1997. 75～83

9 Lorenz D H, Orda A. Optimal partition of QoS requirements on unicast paths and multicast trees. In: Proc. of IEEE INFOCOM'99, 1999

10 Orda A. Routing with end-to-end QoS guarantees in broadband networks. IEEE/ACM Transactions on Networking, 1999, 7(3): 365～374

11 Korkmaz T, Krunz M, Tragoudas S. An efficient algorithm for finding a path subject to two additive constraints. Computer Communications, 2002, 25(3): 225～238

12 Ergun F, Sinha R, Zhang L. QoS routing with performance-dependent costs. In: Proc. of IEEE INFOCOM'2000

13 Neve D, Mieghem V. TAMCRA: a tunable accuracy multiple constraints routing algorithm. Computer Communications, 2000, 23(2000): 667～679

14 Hassin P. Approximation schemes for the restricted shortest path problem. Mathematics of Operations Research, 1992, 17(1): 36～42

15 Jaffe J M. Algorithms for finding paths with multiple constraints. Networks, 1984, 14: 95～116