

# 基于 Pareto 最优的 QoS 路由算法\*

郑彦兴<sup>1+</sup>, 田菁<sup>2</sup>, 窦文华<sup>1</sup>

<sup>1</sup>(国防科学技术大学 计算机学院,湖南 长沙 410073)

<sup>2</sup>(国防科学技术大学 机电工程与自动化学院,湖南 长沙 410073)

## A QoS Routing Algorithm Based on Pareto Optimal

ZHENG Yan-Xing<sup>1+</sup>, TIAN Jing<sup>2</sup>, DOU Wen-Hua<sup>1</sup>

<sup>1</sup>(School of Computer Science, National University of Defense Technology, Changsha 410073, China)

<sup>2</sup>(School of Mechanical Engineering & Automation Control, National University of Defense Technology, Changsha 410073, China)

+ Corresponding author: E-mail: yxzheng@nudt.edu.cn, http://www.nudt.edu.cn

Received 2003-12-12; Accepted 2005-01-07

Zheng YX, Tian J, Dou WH. A QoS routing algorithm based on Pareto optimal. *Journal of Software*, 2005, 16(8):1484-1489. DOI: 10.1360/jos161484

**Abstract:** A novel unicast QoS routing algorithm is proposed to address the two additive constraints routing problem. The algorithm is based on the concept of Pareto optimal and dynamic weight coefficient mechanism. Normally to attain a high performance, the algorithm only needs to run 2~3 times of the Dijkstra algorithm. Extensive simulations have shown that the algorithm is very efficient and converges quickly.

**Key words:** QoS routing; Pareto optimal; dynamic weight coefficient; dominated path

**摘要:** QoS 路由是 QoS 框架中的重要组成部分,旨在寻找多约束条件下的可行路径.在解决多约束(MCP)问题时,引入了 Pareto 最优概念.基于此概念,提出了基于 Pareto 最优的 QoS 权重空间划分模型.在该模型中,根据路由请求与 MCP 问题解的关系,很容易判定路由请求是否能够被满足.在模型基础上,提出了用于解决具有两可加约束的动态权重系数路由算法 PODWCA,它平均只需要运行 2~3 次,Dijkstra 算法就能达到很高的性能.仿真结果验证了 PODWCA 算法的有效性.

**关键词:** QoS 路由;Pareto 最优;动态权重系数;支配路径

中图法分类号: TP393 文献标识码: A

目前,越来越多的网络应用需要服务质量保证,尤其是多媒体应用大多有时延、带宽以及费用的要求.一般说来,QoS 度量尺度(metrics)有 3 种类型,可加尺度,如时延、代价等;可乘尺度,如丢失率等;极大极小尺度,如带宽等.对于极大极小约束,往往可以在路由算法之前采用一定的缩减步骤进行处理;可乘尺度可以转化为可加尺度,因此,QoS 路由面临的主要问题来自可加尺度.

**定义 1.** 多约束问题(MCP).网络  $G(N,E)$ ,  $N$  是节点集合,  $E$  是边集.每条边对应一个多维权重向量

\* 作者简介: 郑彦兴(1974 - ),男,黑龙江海伦人,博士,主要研究领域为网络路由算法;田菁(1980 - ),女,博士生,主要研究领域为任务规划;窦文华(1946 - ),男,教授,博士生导师,主要研究领域为高级计算机网络.

$w(w_1, w_2, \dots, w_m)$ .  $w_i$  是可加权重(weight),  $i=1, \dots, m$ . 给定  $m$  个约束,  $L_{ij}, j=1, \dots, m$ , 寻找从源节点  $s$  到目的节点  $d$  的路径  $P$ , 使得

$$w_i(P) = \sum_{e \in P} w_i(e) \leq L_i, \quad i=1, \dots, m.$$

路径  $P$  又可以写成  $P(w_1, w_2, \dots, w_m)$  的形式. 满足所有  $m$  个约束的路径称为一个可行路径, 可能存在多条从  $s$  到  $d$  的路径同时满足约束. 当  $m=2$  时, MCP 问题就是具有两个可加约束的路由问题, 该问题是 NP 完全问题. 目前已有很多用来解决 NP-完全问题的启发式算法<sup>[1-7]</sup>. Yong Cui<sup>[6]</sup>提出了解决多约束的预计算算法 MEFPA. 当  $k=2$  时, MEFPA 为二约束 QoS 路由算法, 本文称其为“2EFPA”.

## 1 Pareto 最优

### 1.1 符号约定

$s$  表示源节点,  $d$  表示目的节点.  $P_{sd}$  表示  $s$  和  $d$  之间的所有路径集合. Dijkstra( $\alpha, \beta$ ) 基于单一权重  $w' = \alpha w_1 + \beta w_2, \alpha + \beta = 1$  的 Dijkstra 算法,  $P(w_1^{\alpha, \beta}, w_2^{\alpha, \beta})$  表示运行 Dijkstra( $\alpha, \beta$ ) 算法得到的路径.

### 1.2 路径代价函数

我们采用了线性路径代价定义形式, 即把两个链路权重线性相加:

$$w(P) = \sum_{i=1}^n \alpha w_1(e_i) + \beta w_2(e_i) = \alpha w_1(P) + \beta w_2(P), \quad \alpha + \beta = 1.$$

这样, Dijkstra 算法就可以基于单一权重而被使用.

#### 1.2.1 QoS 权重空间

对于路径  $P = n_1 \rightarrow n_2 \rightarrow \dots \rightarrow n_k$ , 每条边具有两个可加权重, 则路径  $P$  每个权重的和可以表示成如下形式:

$$w_j(P) = \sum_{i=2}^k w_j(n_{i-1} \rightarrow n_i), \quad j \in \{1, 2\}.$$

因此, 路径  $P$  又可写成  $P(w_1(P), w_2(P))$ .

定义 2. QoS 权重空间.  $W^2 = W_1 \times W_2$  被称为 QoS 权重空间, 如果对任何  $P \in G(N, E)$ ,  $w_j(P) \in W_j, j \in \{1, 2\}$ .

定义 3. 映射  $f: W^2 = W_1 \times W_2$  被称为 QoS 权重空间, 如果对任何  $P \in G(N, E)$ ,  $w_j(P) \in W_j, j \in \{1, 2\}$ .

### 1.3 Pareto 最优

定义 4. 支配路径. 路径  $P(u_1, \dots, u_m)$  支配路径  $P(v_1, \dots, v_m)$ , 当且仅当  $\forall i \in \{1, \dots, m\}, u_i \leq v_i \wedge \exists i \in \{1, \dots, m\} : u_i < v_i$ , 在本文的后面, 我们将采用  $P(u_1, \dots, u_m) < P(v_1, \dots, v_m)$  来表示路径  $P(u_1, \dots, u_m)$  支配路径  $P(v_1, \dots, v_m)$ .

定义 5. Pareto 最优. 路径  $P(w_1, w_2) \in P_{sd}$  称为 Pareto 最优的, 如果不存在路径  $P'(w'_1, w'_2) \in P_{sd}$ , 使得  $f(P'(w'_1, w'_2)) = (w'_1, w'_2)$  支配  $f(P(w_1, w_2)) = (w_1, w_2)$ , 即  $(w'_1, w'_2) < (w_1, w_2)$ .

定义 6. Pareto 集  $P_{\text{true}}$  及 Pareto front( $PF_{\text{true}}$ ).

对于 MCOP 问题,  $P_{\text{true}} = \{P(w_1, w_2) \in P_{sd} \mid P(w_1, w_2) \text{ 为 MCOP 问题的 Pareto 最优解}\}$ .

Pareto front  $PF_{\text{true}} = \{f(P(w_1, w_2)) \mid P(w_1, w_2) \in P_{\text{true}}\}$ . 由于路径  $P \in P_{sd}$  的非连续性,  $PF_{\text{true}}$  是  $W^2$  中的点集.

定理 1. 对任何  $(\alpha, \beta), \alpha + \beta = 1$ , 路径  $P(w_1^{\alpha, \beta}, w_2^{\alpha, \beta})$  都是 Pareto 最优的, 即  $(w_1^{\alpha, \beta}, w_2^{\alpha, \beta}) \in PF_{\text{true}}$ .

证明: 采用反证法. 如果路径  $P(w_1^{\alpha, \beta}, w_2^{\alpha, \beta})$  不是 Pareto 最优的, 则存在路径  $P'(w_1^{\alpha', \beta'}, w_2^{\alpha', \beta'}) \in PF_{\text{true}}$ , 使得  $(w_1^{\alpha', \beta'}, w_2^{\alpha', \beta'}) < (w_1^{\alpha, \beta}, w_2^{\alpha, \beta})$ , 即  $w_1^{\alpha', \beta'} \leq w_1^{\alpha, \beta}, w_2^{\alpha', \beta'} \leq w_2^{\alpha, \beta}$ , 并且  $\exists i \in \{1, 2\} : w_i^{\alpha', \beta'} < w_i^{\alpha, \beta}$ . 因此,  $\alpha w_1^{\alpha', \beta'} + \beta w_2^{\alpha', \beta'} < \alpha w_1^{\alpha, \beta} + \beta w_2^{\alpha, \beta}$ . 这意味着当运行 Dijkstra( $\alpha, \beta$ ) 时, 返回路径是  $P'(w_1^{\alpha', \beta'}, w_2^{\alpha', \beta'})$ , 而不是  $P(w_1^{\alpha, \beta}, w_2^{\alpha, \beta})$ , 与事实矛盾.

## 2 具有 2 可加约束 QoS 路由问题的 Pareto Front

定理 2. 对于权重系数  $(\alpha_i, \beta_i), \alpha_i + \beta_i = 1, i \in \{1, 2\}$ , 如果  $\alpha_1 > \alpha_2$ , 则 (1)  $w_1^{\alpha_1, \beta_1} \leq w_1^{\alpha_2, \beta_2}$ , 并且 (2)  $w_2^{\alpha_1, \beta_1} \geq w_2^{\alpha_2, \beta_2}$ .

证明: (1) 由已知条件可知:

$$\alpha_1 w_1^{\alpha_1, \beta_1} + \beta_1 w_2^{\alpha_1, \beta_1} < \alpha_1 w_1^{\alpha_2, \beta_2} + \beta_1 w_2^{\alpha_2, \beta_2} \Rightarrow \alpha_1 (w_1^{\alpha_1, \beta_1} - w_1^{\alpha_2, \beta_2}) < \beta_1 (w_2^{\alpha_2, \beta_2} - w_2^{\alpha_1, \beta_1}) = (1 - \alpha_1) (w_2^{\alpha_2, \beta_2} - w_2^{\alpha_1, \beta_1}) \quad (i)$$

$$\alpha_2 w_1^{\alpha_2, \beta_2} + \beta_2 w_2^{\alpha_2, \beta_2} < \alpha_2 w_1^{\alpha_1, \beta_1} + \beta_2 w_2^{\alpha_1, \beta_1} \Rightarrow \alpha_2 (w_1^{\alpha_2, \beta_2} - w_1^{\alpha_1, \beta_1}) < \beta_2 (w_2^{\alpha_1, \beta_1} - w_2^{\alpha_2, \beta_2}) = (1 - \alpha_2)(w_2^{\alpha_1, \beta_1} - w_2^{\alpha_2, \beta_2}) \quad (ii)$$

$$(i)+(ii) \Rightarrow (w_1^{\alpha_1, \beta_1} - w_1^{\alpha_2, \beta_2}) < (w_2^{\alpha_1, \beta_1} - w_2^{\alpha_2, \beta_2}) \quad (iii)$$

如果  $w_1^{\alpha_1, \beta_1} > w_1^{\alpha_2, \beta_2}$ , 则(iii)  $\Rightarrow (w_1^{\alpha_2, \beta_2}, w_2^{\alpha_2, \beta_2}) < (w_1^{\alpha_1, \beta_1}, w_2^{\alpha_1, \beta_1})$ . 根据定理 1,  $(w_1^{\alpha_1, \beta_1}, w_2^{\alpha_1, \beta_1})$  及  $(w_1^{\alpha_2, \beta_2}, w_2^{\alpha_2, \beta_2})$  都在 Pareto front 中,  $(w_1^{\alpha_2, \beta_2}, w_2^{\alpha_2, \beta_2})$  不能支配  $(w_1^{\alpha_1, \beta_1}, w_2^{\alpha_1, \beta_1})$ , 因此,  $w_1^{\alpha_1, \beta_1} \leq w_1^{\alpha_2, \beta_2}$ .

(2) 根据定理 1,  $(w_1^{\alpha_2, \beta_2}, w_2^{\alpha_2, \beta_2})$  和  $(w_1^{\alpha_1, \beta_1}, w_2^{\alpha_1, \beta_1})$  不能相互支配, 再由式 (1) 的结论  $w_1^{\alpha_1, \beta_1} \leq w_1^{\alpha_2, \beta_2}$  可知,  $w_2^{\alpha_1, \beta_1} \geq w_2^{\alpha_2, \beta_2}$ .

定理 3. 对于 2 可加约束 MCOP 问题, 对所有的  $(w_1, w_2) \in PF_{true}$ , (1)  $w_1^{1,0} \leq w_1$ ; (2)  $w_2^{1,0} \geq w_2$ ; (3)  $w_1^{0,1} \geq w_1$ ; (4)  $w_2^{0,1} \leq w_2$ .

证明: 采用反证法. (1) 如果存在路径  $P'(w'_1, w'_2)$  满足  $w'_1 < w_1^{1,0}$ , 则 Dijkstra(1,0) 返回的路径将是  $P'(w'_1, w'_2)$ , 而不是  $P(w_1^{1,0}, w_2^{1,0})$ . 这与事实矛盾.

(2) 如果存在路径  $P''(w''_1, w''_2)$ , 使得  $w''_2 < w_2^{1,0}$ , 则根据 (1) 的结论可知,  $w_1^{1,0} \leq w''_1$ , 因此  $(w''_1, w''_2)$  支配  $(w_1^{1,0}, w_2^{1,0})$ , 这与定理 1 矛盾. (3)、(4) 的证明与 (1)、(2) 的证明类似, 略.

### 3 采用 Pareto Front 对 $W^2$ 进行划分

定义 7. 支配区域. 向量  $w = (w_1, w_2)$  的支配区域定义为  $D(w) \triangleq \{ (w'_1, w'_2) \mid (w_1, w_2) \leq (w'_1, w'_2) \}$ .

定义 8. 可行区域  $A_{feasible}$ .  $A_{feasible} = \bigcup_{v \in PF_{true}} D(v) \cup PF_{true}$ .

定理 4. 如果路由请求约束  $c = (c_1, c_2) \in A_{feasible}$ , 则该请求可被满足, 即存在从  $s$  到  $d$  能满足约束的路径.

证明: 根据  $A_{feasible}$  的定义,  $\exists v \in PF_{true}$  使得  $v = (w_1, w_2)$  支配  $c = (c_1, c_2)$ , 即  $w_1 \leq c_1$  并且  $w_2 \leq c_2$ , 因此路径  $f^{-1}(v)$  能够满足请求. 前面已经提到,  $f^{-1}(v)$  代表的可能不止一条路径, 其中的任何一条路径都能够满足路由请求.

定义 9. 不可行区域  $A_{unfeasible}$ .  $A_{unfeasible} = A_{bedom} \cup A_{10} \cup A_{01}$ .

这里,  $A_{bedom} = \{ v = (w_1, w_2) \mid \exists v' \in PF_{true}, \text{使得 } (w_1, w_2) < v' \}$ ;

$$A_{10} = \{ (w_1, w_2) \mid w_1 < w_1^{1,0}, (w_1, w_2) \in W^2 \};$$

$$A_{01} = \{ (w_1, w_2) \mid w_2 < w_2^{0,1}, (w_1, w_2) \in W^2 \}.$$

定理 5. 如果路由请求  $c = (c_1, c_2) \in A_{unfeasible}$ , 则对于任意路径  $P \in P_{sd}$  都不能满足该请求.

证明: 根据定义  $A_{unfeasible} = A_{bedom} \cup A_{10} \cup A_{01}$  可知,  $c \in A_{bedom}$ , 或者  $c \in A_{10}$ , 或者  $c \in A_{01}$ . (1) 如果  $c \in A_{bedom}$ , 则存在  $v \in PF_{true}$ , 使得  $c < v$ . 假设存在路径  $P(w_1, w_2)$  能够满足请求, 即  $w_1 \leq c_1$  并且  $w_2 \leq c_2$ , 则意味着  $(w_1, w_2) < v$ , 因此  $v \notin PF_{true}$ , 这与事实矛盾. (2) 如果  $c \in A_{10}$ , 则  $c_1 < w_1^{1,0}$ . 根据定理 3, 对任何  $v(w_1, w_2) \in PF_{true}$ , 都有  $w_1 \geq w_1^{1,0}$ , 因此  $c_1 < w_1$ . 这意味着 Pareto front 中的任何元素都不能满足路由请求. 因此, 当  $c \in A_{10}$  时, 没有路径能够满足路由约束请求. (3) 证明与 (2) 的证明类似, 略.

定义 10. 未知区域  $A_{unknown}$ .  $A_{unknown} = W^2 / (A_{feasible} \cup A_{unfeasible})$ .

如果能够确定找到了  $PF_{true}$  中的所有元素, 未知区域  $A_{unknown}$  实质上成为不可行区域的一部分. 然而, 到目前为止, 还没有办法得到  $PF_{true}$  的确切大小. Miegheem<sup>[7]</sup> 指出了  $PF_{true}$  大小的可能最大值. 然而,  $PF_{true}$  的实际大小一般是远远低于该值的. 本文的仿真结果表明, 2 可加约束路由问题的 Pareto front 是很小的. 只要恰当地选择权重系数, 则可以找到绝大多数  $PF_{true}$  中的元素. 因此, 问题变成了如何选择权重系数, 使得在实际问题允许的计算代价下, 找到绝大多数  $PF_{true}$  中的元素.

2EFPA 把每个权重分为  $b$  个等级,  $b$  是一个很小的整数. 然后基于  $B = C_{b+k-2}^{k-1}$  个不同权重系数组合运行  $B$  次 Dijkstra 算法, 对于 2 可加约束问题,  $B=b$ . 我们做了大量的仿真实验, 证明 2EFPA 执行了较多的冗余计算. 每个实验随机生成一个 Waxman 网络图, 网络每条边上有两个权重, 都服从区间  $[1, 300]$  内的均匀分布. 随机选择一个源节点及 20 个不同于源节点的目的节点, 统计源节点到目的节点的平均冗余路径. 网络的规模分别为 50、100、200, 在每个规模上进行 20 次上述实验, 得到如图 1 所示的仿真结果. 横坐标是实验标识(EID), 纵坐标是平均冗余

路径(ARP).

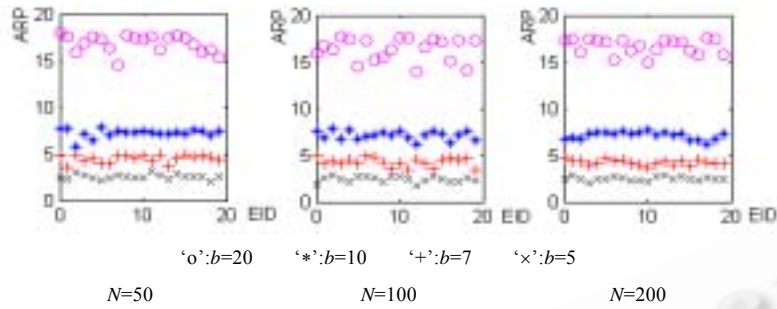


Fig.1 2EFPA conducts more redundant computation

图 1 2EFPA 执行了大量冗余计算

### 4 PODWCA 算法

通过以上分析,我们提出了基于 Pareto 最优及动态权重系数的 QoS 路由算法,在一定程度上避免了冗余计算,并且算法收敛速度非常快.

**定理 6.**  $(\alpha_1, \beta_1)$  和  $(\alpha_2, \beta_2)$  是不同的两个权重系数组合,且  $1 \geq \alpha_1 > \alpha_2 \geq 0$ , 如果  $p(w_1^{\alpha_1, \beta_1}, w_2^{\alpha_1, \beta_1})$  与  $p(w_1^{\alpha_2, \beta_2}, w_2^{\alpha_2, \beta_2})$  是同一条路径,即  $w_1^{\alpha_1, \beta_1} = w_1^{\alpha_2, \beta_2}, w_2^{\alpha_1, \beta_1} = w_2^{\alpha_2, \beta_2}$ , 则对于任意  $(\alpha, \beta), \alpha_1 > \alpha > \alpha_2, p(w_1^{\alpha, \beta}, w_2^{\alpha, \beta})$  与  $p(w_1^{\alpha_1, \beta_1}, w_2^{\alpha_1, \beta_1})$  也是同一条路径,即  $w_1^{\alpha, \beta} = w_1^{\alpha_1, \beta_1}, w_2^{\alpha, \beta} = w_2^{\alpha_1, \beta_1}$ .

**证明:**根据定理 2,  $\alpha_1 > \alpha \Rightarrow w_1^{\alpha, \beta} \geq w_1^{\alpha_1, \beta_1}; \alpha > \alpha_2 \Rightarrow w_1^{\alpha_2, \beta_2} = w_1^{\alpha, \beta}$ . 根据给定条件有:  $w_1^{\alpha_1, \beta_1} = w_1^{\alpha_2, \beta_2}$ , 因此  $w_1^{\alpha_2, \beta_2} = w_1^{\alpha, \beta}$ ; 基于同样原因  $w_2^{\alpha_2, \beta_2} = w_2^{\alpha, \beta}$ .

定理 6 的意义是非常明显的,即如果  $Dijkstra(\alpha_1, \beta_1)$  及  $Dijkstra(\alpha_2, \beta_2)$  返回相同的路径, 那么没有必要再考虑任何满足  $\alpha_1 > \alpha > \alpha_2$  的权重系数  $(\alpha, \beta)$ . 这样能在一定程度上避免冗余计算.

PODWCA 算法的主要思想是维护一个区间列表(interval list,简称 ITL).ITL 中包含即将处理的区间.每次 PODWCA 处理 ITL 中的第 1 个区间,处理完毕后,该区间从 ITL 中删除.Dijkstra 算法所依赖的权重系数之一  $\alpha$  会根据当前区间调整自身的值.PODWCA 首先通过运行  $Dijkstra(1,0)$  计算最短路径  $P_1(w_1, w_2)$ . 如果  $(w_1, w_2) < (c_1, c_2)$ ,PODWCA 将返回路径  $P_1$  并终止.否则,如果请求  $(c_1, c_2)$  落在不可行区域  $A_{unfeasible}$ .PODWCA 就知道在  $s$  和  $d$  之间没有能够满足约束的路径,算法也将终止.如果上述条件都不满足,PODWCA 将运行  $Dijkstra(0,1)$  计算最短路径  $P_2$ .得到  $P_2$  后,PODWCA 将首先检查  $P_1$  和  $P_2$  是否为同一条路径,如果是,则算法终止(根据定理 6,不论  $(\alpha, \beta)$  取何值,都不可能再有与  $P_1$  不同的路径).应该注意的是,在算法运行过程中,未知区域  $A_{unfeasible}$  一直处于变化之中,因此,算法将再一次检查路由请求是否落在  $A_{unfeasible}$  内.最后,PODWCA 把区间  $[0,1]$  添加到 ITL 列表中,开始循环处理 ITL 的操作.

每次 PODWCA 处理一个区间  $[x,y](y>x)$  时,它已经知道由  $Dijkstra(x,1-x)$  和  $Dijkstra(y,1-y)$  返回的路径  $P_x$  及  $P_y$ .PODWCA 将把区间中点作为权重系数第 1 个分量  $\alpha$  的值,并运行  $Dijkstra((x+y)/2, 1-(x+y)/2)$  获得路径  $P_{middle}$ . 然后算法判断  $P_{middle}$  是否与  $P_x$  和  $P_y$  的其中之一相同.(1) 如果与  $P_x$  和  $P_y$  的其中之一相同,假设  $P_{middle}$  和  $P_x$  是同一条路径,则算法将添加区间  $[(x+y)/2, y]$  到 ITL 的末尾,而忽略区间  $[x, (x+y)/2]$ . 因为根据定理 6,对任何  $z \in (x, (x+y)/2)$ , 算法  $Dijkstra(z, 1-z)$  将得到与  $P_x$  相同的路径.(2) 如果  $P_{middle}$  与  $P_x$  和  $P_y$  都不相同,则记录  $P_{middle}$ , 并把区间  $[x, (x+y)/2]$  和  $[(x+y)/2, y]$  添加到 ITL 的末尾.注意,  $P_x$  及  $P_y$  不会是相同的路径,否则区间  $[x,y]$  不会被添加到 ITL. 每次 PODWCA 处理 ITL 中的第 1 个元素,并把新区间添加到列表末尾,保证了每次处理的区间都是 ITL 中最大的. 算法处理完区间  $[x,y]$  后,将其从 ITL 中删除,然后开始处理 ITL 中的下一个区间.

### 5 性能评估

由上面的分析可知,PODWCA 的终止条件是非常灵活的,因此很难估计它的计算复杂性.我们从以下两方

面来评价 PODWCA 算法的性能和计算复杂性:绝对性能和相对性能.

### 5.1 绝对性能

我们采用未知区域的大小来评价算法的绝对性能,同时给出了 2EFPA 的绝对性能作为比较.仿真采用 Waxman 网络模型,节点数目分别为 50,100,200.网络每条边上的两个权重服从区间[1,300]内的均匀分布.对每次实验,我们随机生成一个 Waxman 拓扑图,随机生成一个源节点和目的节点,并保证源节点和目的节点之间至少两跳,然后统计未知区域的大小,重复实验 1 000 次,统计未知区域大小的平均值,作为算法运行  $b$  次 Dijkstra 算法的绝对性能.如图 2 所示,图中的横坐标是算法运行 Dijkstra 的次数  $b$ ,纵坐标是未知区域的平均大小.从图中可以看出,算法 PODWCA 的未知区域比 2EFPA 的未知区域随着  $b$  的增大下降要快得多,并最终在运行大约 15 次 Dijkstra 算法后收敛到一个常值.PODWCA 算法收敛快的一个重要原因是采用了动态权重系数机制.

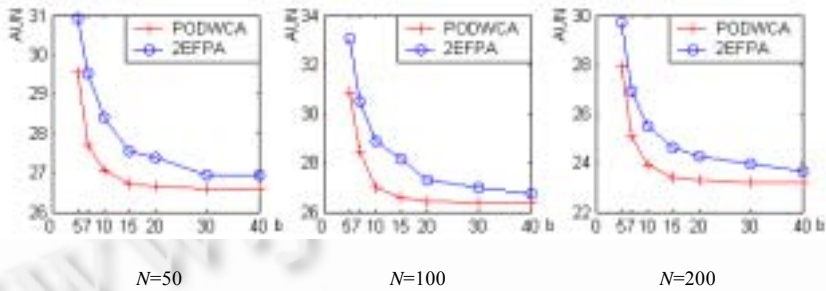


Fig.2 The comparison of absolute performance between PODWCA and 2EFPA

图2 算法 PODWCA 与 2EFPA 的绝对性能比较

### 5.2 相对性能和相对计算复杂性

Cui 通过仿真验证了  $b=7$  时,2EFPA 的性能要超过 H\_MCOP 的性能.因此,我们考虑在何种计算代价下,PODWCA 能够达到当  $b=7$  时,2EFPA 的性能.实际上,当  $b=7$  时,2EFPA 达到的权重系数粒度为  $1/6$ .因此,只要在算法 PODWCA 的终止条件中加入一条:当 ITL 中的最大区间长度小于  $1/6$  时,算法也终止.此时算法 PODWCA 的性能与当  $b=7$  时 2EFPA 的性能大致相当.

我们为节点对  $(s,d)$  生成路由请求  $(c1,c2)$ ,满足  $c1$  服从区间  $[x1,2 \times x2 - x1]$  内的均匀分布, $c2$  服从区间  $[y2, 2 \times y1 - y2]$  内的均匀分布.这样生成的约束能够覆盖整个未知区域,使得算法在最苛刻的条件下进行评估.网络模型及链路权重的生成与评估绝对性能时相同.图 3 显示了算法 PODWCA 的相对性能.纵坐标是成功率(被满足的约束个数与生成的总约束个数的比值).横坐标是算法 2EFPA 运行 Dijkstra 算法的次数.

图 4 记录了当算法 PODWCA 达到上述性能时的平均计算复杂性.横坐标是算法 2EFPA 运行 Dijkstra 算法的次数.纵坐标是 PODWCA 达到相应 2EFPA 性能时运行 Dijkstra 算法的平均次数.从图 4 中不难看出,要达到算法 2EFPA 当  $b=7,10,20$  时的性能,PODWCA 平均只需运行 Dijkstra 算法 2~3 次.算法 PODWCA 的高效主要得益于(1) 动态权重机制,使得算法能够迅速达到很小的权重系数粒度;(2) 基于 Pareto optimal 的算法终止机制,使得 PODWCA 算法能够快速判断路由请求是否可行.

## 6 结 论

本文把 Pareto 最优概念引入具有 2 可加约束 QoS 路由问题,提出了完整的基于 Pareto 最优的 MCOP 问题权重空间划分理论.本文给出的一些定理和结论,除了二维空间特有的性质以外,都可以推广到多维空间.如 Pareto 最优概念、3 个区域的划分等.本文还提出了基于 Pareto 最优和动态权重系数机制的路由算法 PODWCA.仿真实验表明,PODWCA 算法在很低的计算复杂性下,就可以达到很高的性能.

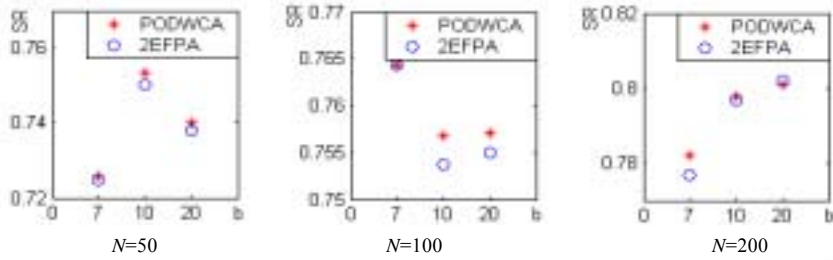


Fig.3 Comparison of relative performance between PODWCA and 2EFPA

图 3 PODWCA 和 2EFPA 的相对性能

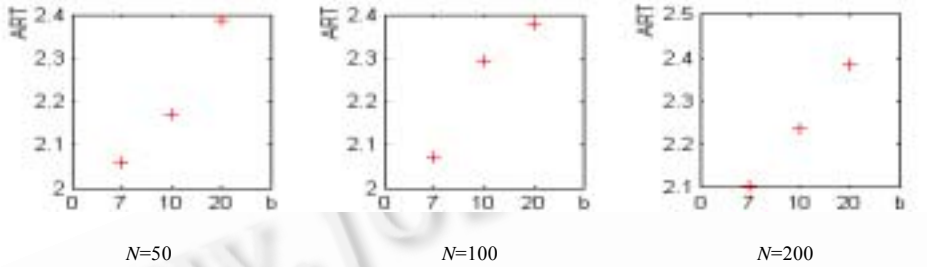


Fig.4 The relative complexity of PODWCA

图 4 算法 PODWCA 的相对复杂性

**References:**

[1] Orda A. Routing with end-to-end QoS guarantees in broadband networks. *IEEE/ACM Trans. on Networking*, 1999,7(3):365–374.

[2] Lorenz DH, Orda A, Raz D, Shavitt Y. Efficient QoS partition and routing of unicast and multicast. In: *Proc. of the IEEE 8th Int'l Workshop on Quality of Service (IWQoS)*. 2000. 75–83.

[3] Juttner A, Szviatovszki B, Mecs I, Rajko Z. Lagrange relaxation based method for the QoS routing problem. In: *Proc. of the INFOCOM 2001, Vol 2*. 2001. 859–868.

[4] van Mieghem P, De Neve H, Kuipers FA. Hop-by-Hop quality of service routing. *Computer Network*, 2001,37(3-4):407–423.

[5] Korkmaz T, Krunz M. Multi-Constrained optimal path selection. *IEEE INFOCOM*, 2001. 834–843.

[6] Cui Y, Xu K, Wu JP. Precomputation for multi-constrained QoS routing in high-speed networks. *IEEE INFOCOM*, 2003.

[7] Van Mieghem P, Kuipers FA. Concepts of exact QoS routing algorithms. *IEEE/ACM Trans. on Networking*, 2004,12:851–864.