文章编号:1001 - 9081(2008)05 - 1101 - 03

带多约束条件的最优路径选择算法研究

邹永贵,魏 来
(重庆邮电大学 移通学院,重庆 400065)
(5-529@163.com)

摘 要:传统的启发式算法把 NP 完全问题转化成一个能够在多项式时间内求解的 P 问题,却不能保证每次都得到最优路径。利用拉格朗日松弛法把该问题转换成一个 P 问题,利用次梯度算法来确定最优解,在降低算法时间复杂度的同时提高最优路径查找的成功率。通过实验和分析,该算法的有效性得到了验证,可以应用在地理信息系统和通信网络中。

关键词:拉格朗日松弛;多权值图;最优路径;多约束条件
中图分类号: TP393;TP301.6 **文献标志码**:A

Optimal path algorithm with multi-constrained condition

ZOU Yong-gui, WEI Lai
(*College of Mobile Telecommunications, Chongqing University of Posts and Telecommunications, Chongqing* 400065, *China*)

Abstract: Traditional heuristic algorithm converts the NP-complete problem to a simpler one that can be solved in polynomial time. However, it cannot guarantee a solution all the time. In this paper, the Lagrange relaxation was applied to traditional heuristic algorithm to improve the successful rate of finding an optimal path and to reduce the time complexity. Finally, the correctness and effectiveness of this algorithm are proved through experiment and analysis.

Key words: Lagrangean relaxation; multi-weighted graph; optimal paths; multi-constrained condition

0 引言

随着计算机的普及以及地理信息科学的发展,GIS 因其强大的功能得到日益广泛和深入的应用。网络分析作为 GIS 最主要的功能之一,在电子导航、交通旅游、城市规划以及电力、通信等各种管网、管线的布局设计中发挥了重要的作用,而网络分析中最基本最关键的问题是最短路径问题[1]。最短路径不仅仅指一般地理意义上的距离最短,还可以引申到其他的度量,如时间、费用、线路容量等。在现实世界中还存在这样一类模型:边对应的权值不是一个而是多个,而且源节点到某一节点的最短路径是有限制条件的,这就变成了多约束最优路径问题[4]。该类问题被广泛应用于汽车导航系统,以及 QoS 路由问题中[3-6]。

多约束最优路径问题(Multi-Constraint Optimal Path, MCOP)可描述为:一个有向的交通网络 $G(V, E)$ 中,V 表示节点的集合,E 表示边的集合;对于边 $(u, v) \in E$,该边对应 k 个权值用 $w_k(u, v)$,$k = 1,2,3\cdots,K$ 来表示,所有的权值均为非负。给定一个 k 约束 c_k,$k = 1,2,3,\cdots,K$,找出一个从源节点 s 到目的节点 t 的一个路径 p 满足两个条件:

$$w_k(p) = \sum_{(u,v)\in p} w_k(u,v) \leqslant c_k;\ k = 1,2,3,\cdots,K \quad (1)$$

$c_p = \sum_{(u,v)\in p} c(u,v)$ 所有满足条件(1)的可行路径中权值最小的路径 (2)

当 $K = 1$ 时 MCOP 问题就是单限制条件的最短路径问题,该问题也是 NP 完全问题[7]。与 MCOP 问题稍微不同的问题是多约束路径问题(Multi-Constrained Path, MCP),该类问题仅仅要求找到一条满足约束条件的可行路径。当 $K \geqslant 2$ 时 MCOP 问题也是一个 NP 完全问题[3,4]。约束最短路径问题(Restricted Shortest Path, RSP)和 MCP 问题都可以通过伪多项式时间算法来解决,该算法时间复杂度取决于边的实际权值(例如最大的边权值)和网络的规模[7,8]。但是这些算法在权值比较大时计算的代价比较昂贵。在这里仅仅考虑 MCOP 问题。

一般来讲,一个寻找双约束最短问题很容易扩展成多约束情况下的最优路径寻找问题。因此,本文致力于双约束问题的处理上,也就是 $K = 2$ 的情况,该类问题可以用 $MCP(G, s, t, w_1, w_2, c_1, c_2)$ 表示。在前面本文提到过 MCOP 问题是 NP 完全问题,除非把 NP 问题转化为 P 问题,不然没有算法能在多项式时间内找出可行解。

为了处理 $MCP(G, s, t, w_1, w_2, c_1, c_2)$ 问题,文献[6,9,10]提出了相关算法。文献[2]针对 $K = 2$ 时的 MCOP 问题提出了直观的、基于最小化边的线性组合权值的近似算法,该算法返回一个最好的路径。线性组合权值为 $l(e) = \alpha w_1(p) + \beta w_2(p)$,其中 $\alpha,\beta \in Z^+$,然后通过 Dijkstra 的最短路径算法进行求解。该算法最关键的地方在于怎么得到最合适的 α 和 β 使得该最优路径中的线性组合权值 $l(e)$ 满足各自的约束条件。在文献[2]中 α 和 β 两个的值根据最小化一个目标函数 $f(p) = \max\{fw_1(p);\ c_1\} + \max\{fw_2(p);\ c_2\}$ 来确定。文献[9,10]考虑到 $MCP(G, s, t, w_1, w_2, c_1, c_2)$ 问题,并提出了一个启发式算法把该类问题转化成简单和可解决的问题。该算法分为 2 步:

1)创建一个新的权值函数:

$$w'_2: E \to I,\ w'_2(u, v) = \left\lceil \frac{w_2(u, v) \cdot x}{c_2} \right\rceil \quad (3)$$

收稿日期:2007 - 11 - 29;**修回日期**:2008 - 01 - 10。 **基金项目**:国家 863 计划项目(2007AA12Z238)。

作者简介:邹永贵(1969 -),男,四川荣县人,副教授,主要研究方向:地理信息系统; 魏来(1982 -),男,河南信阳人,硕士研究生,主要研究方向:空间定位、网络优化。

其中 x 是一个给定的整数。该方法把原来的 $MCP(G, s, t, w_1, w_2, c_1, c_2)$ 问题转化成新的简单的 $MCP(G, s, t, w_1, w'_2, c_1, x)$ 问题。

2)用扩展 Dijkstra 算法或者 Bellman-Ford 算法在多项式时间内解决 $MCP(G, s, t, w_1, w'_2, c_1, x)$ 问题,这些近似和启发式算法时间复杂度都很高,而且经常找不出问题的解。

本文给出解决多约束最短路径问题的一个基于拉格朗日松弛的启发式算法。主要思想是利用拉格朗日松弛算法把原问题变成一个 P 问题[4],利用启发式算法找到算法可行解。

1 拉格朗日松弛算法

首先把 MCOP($K=2$) 问题转换成如下形式,假设 c_{ij} 表示弧 (i,j) 的代价,t_{ij} 表示对弧 (i,j) 的遍历时间:对于网络 $G=(N, A)$ 有:

$$z^* = \min \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{4}$$

$$\sum_j x_{ij} - \sum_j x_{ij} = \begin{cases} 1, & i = s \\ -1, & i = t \\ 0, & \text{其他} \end{cases} \tag{5}$$

$$\begin{cases} \sum_{(i,j)\in A} t_{ij}x_{ij} \leqslant T \\ \sum_{(i,j)\in A} c_{ij}x_{ij} \leqslant C \end{cases} \tag{6}$$

$x_{ij}=0$ 或 1 对于所有的弧 $(i,j)\in A$ 在此引入一个系数 $\mu \geqslant 0$,叫拉格郎日松弛乘子,把约束吸引到目标函数,得到一个新的函数:

$$z(\mu) = \min \sum_{(i,j)\in A} c_{ij}x_{ij} + \mu\Big(\sum_{(i,j)\in A} t_{ij}x_{ij} - T\Big) = \sum_{(i,j)\in A} (c_{ij} + \mu t_{ij})x_{ij} - \mu T \tag{7}$$

于是对于所有 $\mu \geqslant 0$,有 $z(\mu) \leqslant z^*$,因为 $\sum_{(i,j)\in A} t_{ij}x_{ij} \leqslant T$。假设 $L(\mu) = \min \sum_{(i,j)\in A} (c_{ij} + \mu t_{ij})x_{ij} - \mu T$,于是有 $L(\mu) \leqslant z(\mu) \leqslant z^*$,$L(\mu)$ 是 z^* 的一个下界。因为 T 是常数,当 μ 确定时,μT 也是常数,所以只需要求 $\sum_{(i,j)\in A} (c_{ij} + \mu t_{ij})x_{ij}$ 的最小值,于是可以用一个新的聚合权值 $w = c_{ij} + \mu t_{ij}$ 来代替原来的权值求解,这样该问题就变成了一个无约束最短路径问题,针对于不同的 μ 可以得到一个不同的下界。

为了获得一个最好的下界需要最大化函数 $L(\mu)$:

$$L^* := \max_{\mu\geqslant 0} L(\mu) \tag{8}$$

如果能够找到一个 μ,使 $L^* = L(\mu) = z(\mu) = z^*$,那么拉格郎日乘子 μ 对于原问题来说是最优的。当然算法并不能保证这个等式成立,所以只能想办法找出一个最合适的 μ,使得它最接近最优解。

问题的关键是如何找出最合适的 μ,在没有介绍算法之前先提出几个定理,这些定理是该算法的基础。

定理 1 如果 $p = Dijk(c+\alpha t)$, $q = Dijk(c+\beta t)$,$\alpha, \beta \in R^+$。

1) 如果 $\alpha \geqslant \beta$,那么有 $C_p \geqslant C_q$,$T_p \leqslant T_q$;

2) 如果 $C_p = C_q$,那么 $T_p = T_q$;

其中 p,q 表示路径,$p = Dijk(c+\alpha t)$ 含义是用 Dijkstra 算法以 $c+\alpha t$ 为边的权值求得的最短路径 p,C_i 表示路径 i 的代价权值,T_i 表示路径 i 的时延权值。

证明 因为 p 是当用 Dijkstra 算法对聚合权值 $c+\alpha t$ 求解时的最短路径,于是有 $C_p + \alpha T_p \leqslant C_q + \alpha T_q$;同理对路径 q 有 $C_q + \beta T_q \leqslant C_p + \beta T_p$;两个不等式联立有 $\alpha T_p \leqslant C_q + \alpha T_q - C_p \leqslant C_p + \beta T_p - \beta T_q + \alpha T_q - C_p = \beta T_p + (\alpha - \beta)T_q$;所以 $T_p \leqslant T_q$,同理 $C_p \geqslant C_q$。

定理 2 如果 $p = Dijk(c+\alpha t)$,$q = Dijk(c+\beta t)$,$r = Dijk(c+\gamma t)$,其中 $\beta < \gamma$,$T_r \neq T_q$,$\alpha = (C_r - C_q)/(T_q - T_r)$,那么 $C_r \geqslant C_p \geqslant C_q$,$T_r \leqslant T_p \leqslant T_q$。

证明 从定理 1 可得只需要证明 $\beta \leqslant \alpha \leqslant \gamma$ 即可。因为 $q = Dijk(c+\beta t)$,所以有 $C_q + \beta T_q \leqslant C_r + \beta T_r$,因为 $\beta < \gamma$ 所以有 $C_q \leqslant C_r$,$T_q \geqslant T_r$ 当 $T_r \neq T_q$ 时 $\beta \leqslant (C_r - C_q)/(T_q - T_r) = \alpha$。同理有 $\alpha \leqslant \gamma$。

总之,通过忽略约束条件(即松弛),把它们建立成目标函数,这样问题就变成了一个无约束条件的最优化路径问题,原问题的可行解当然同时也适合松弛后的解。结合定理 1、2 可知松弛乘子越大,路径的时延越大,代价也就越小。这说明只要满足时延约束条件 T,较大的松弛乘子可以获得较优的解。为了获得一个最优的解,必须想办法慢慢逼近这个松弛乘子,这里采用次梯度优化算法[11],但是实际问题中不可能迭代无穷多次,所以必须采用一种停止原则[12],文献[11]采用上下界相等的停止原则来获得最优的解,但是这种算法存在一个问题就是当对聚合权值求最短路径有两个或者多个相等结果的时候,选取不当的话会导致最终结果不是最优值。如下图边的权值对应是 (C,T),其中 C 表示代价,T 表示延迟,找出一个从节点 1 到节点 6 的对于权值 C 来说的最短路径并且满足 $T \leqslant 5$。

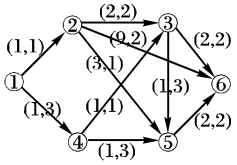

图 1 示例

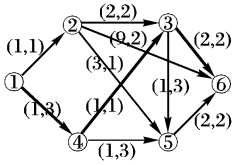

图 2 最小代价路径 P_c

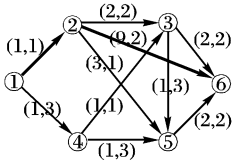

图 3 最小延迟路径 P_d

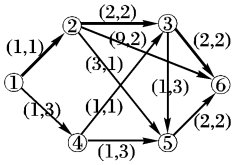

图 4 最优路径

图 1 为初始的代价和延迟图,对于图 2 中 $C(P_d) = 10$,$T(P_d) = 3$,图 3 中 $C(P_c) = 4$,$T(P_c) = 6$;根据以前的算法有 $\mu = (C(P_d) - C(P_c))/(T(P_c) - T(P_d)) = 2$。于是把图的权值转化成 $C + 2T$,然后再根据次梯度优化算法求出 μ,如图 5 所示。

图 5(a) 中可以看出结果 ①②⑤⑥ 作为最短路径 r,那么根据文献[11]的 LARAC 算法,因为 $T(r) = 4 \leqslant 5$,所以 P_d 被 r 替代,于是 $\mu = (C(P_d) - C(P_c))/(T(P_c) - T(P_d)) = 1$,把图的权值转化成 $C + T$,求最短路径后得到图 5(b) 黑线和虚线所示的三条最短路径 ①②⑤⑥、①②③⑥、①④③⑥,无

论 r 选取哪条路径都有 $c_\mu(r) = c_\mu(p_c) = 10$,所以结果有三种可能值,可以看出最优的结果明显是 ①②③⑥,如图 4 所示,其他两种并不是最优的结果。可见 LARAC 算法具有不稳定性,为了克服这个缺点,本文对 Dijksra 算法进行了适当的改进,定义了一个选取规则,从而使算法接近最优解。

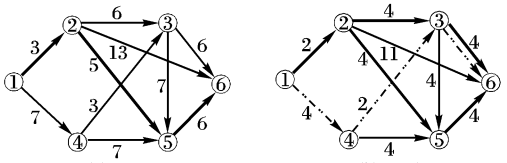

图 5　松弛乘子迭代下的最短路径图

定理 3　假设 P_1 和 P_2 两条路径有 $C(P_1) + \mu T(P_1) = C(P_2) + \mu T(P_2)$,其中 $\mu > 0$, $T(P_1) \leqslant T$,$T(P_2) \leqslant T$。

1) 如果 $C(P_1) < C(P_2)$,那么必有 $T \geqslant T(P_1) > T(P_2)$ 且 P_1 优于 P_2;

2) 如果 $C(P_1) = C(P_2)$,那么必有 $T(P_1) = T(P_2)$。

证明

1) 由 $C(P_1) + \mu T(P_1) = C(P_2) + \mu T(P_2)$ 得,$\mu = (C(P_1) - C(P_2))/(T(P_2) - T(P_1)) > 0$,而 $C(P_1) - C(P_2) > 0$,所以 $T(P_2) - T(P_1) > 0$,已知 $T(P_1) \leqslant T$, $T(P_2) \leqslant T$,于是得到 $T \geqslant T(P_1) > T(P_2)$。根据式(4)要找出一条路径 P,在满足时延约束的基础上,代价满足 $z^* = \min \sum_{(i,j)\in A} c_{ij}x_{ij} = \min(C(A))$,其中 A 表示所有可行的路径集合。P_1 和 P_2 同时满足时延约束 $T(P_1) \leqslant T$,$T(P_2) \leqslant T$,而代价 $C(P_1) < C(P_2)$,所以 P_1 要优于 P_2。

2) 已知 $C(P_1) + \mu T(P_1) = C(P_2) + \mu T(P_2)$, 当 $C(P_1) = C(P_2)$ 时,有 $\mu T(P_1) = \mu T(P_2)$,由于 $\mu > 0$,所以 $T(P_1) = T(P_2)$。

根据定理 1,2,3,改进算法如下:

```
E-LARAC ( G, s, t, w2, c, d, T )
  pd = EDijkstra( s, t, d )                          //获得最小时延
  if D ( pd ) > T return "there is no solution! "
                  //如果最小时延不满足时 延 T, 无解,直接退出
  pc = EDijkstra( s, t, c )                          //获得最小代价
  if D(pc) ≤ T return pc
          //如果最小代价满足时延约束 T,则为最优解,退出
  do(){
    μ = (C(pd) - C(pc))/(D(pc) - D(pd))              // 更新乘子
    r = EDijkstra(s, t, cμ)
    if (cμ(r) = cμ(pc) or cμ(r) = cμ(pd)) return r
                                          //满足停止原则退出
    else if D(r) ≤ T then return pd = r
                                      // 更新 pd 为更好的可行解
    else pc = r                       // 更新 pc 为更好的不可行解
  }while(1)
EDijkstra( s, t, c + μd )
  Initinilize()                                      //初始化
  u = s
  SPT = { s }                        //把源节点点加入最短路径集合
  do
    for each edge e = adj( u, v )
                          // 对所有与 u 相邻的节点点进行松弛
    if v isn' t in SPT
    Relax( u, v )
    while( v not equal t )
  return SPT                                  // 返回 s 到 t 的最短路径
Relax( u, v )
  temp = rμ ( u );
  cμ ( u, v ) = c ( u, v ) + μd ( u, v )
  if( rμ(u) + cμ(u, v) > rμ(v) ) continue
  else if ( rμ ( u ) + cμ ( u, v ) < rμ ( v ))
        rμ ( v ) = rμ ( u ) + cμ ( u, v )
        SPT = SPT ∪{ v }
  else if ( rc ( u ) + c ( u, v ) < rc ( v ))
        rμ ( v ) = rμ ( u ) + cμ ( u, v )
        SPT = SPT ∪{ v }
              //聚合权值相等时选代价较小的作为最短路径
  else continue              //聚合权值相等并且代价也相等时
                  //或者代价大的时候,不用更新最短路径
```

2　算法分析

2.1　正确性分析

由上述定理 1、2、3 可知算法的正确性,当算法停止的时候得出的解等于最好的可行解,也就是最优解。

2.2　时间复杂度

Dijkstra 算法的时间复杂度为 $O(n^2)$, E-LARAC 算法的执行次数为 k 次,k 由边的总数来决定,一般情况下可以求得 $k = m\log^3 m$[5],算法的时间复杂度为 $kn^2 = mn^2\log^3 m$,其中 n 为节点个数,m 为边的个数。

3　算法仿真

在图 6 所示的网络环境下进行算法的性能评估,该网络是通过修改 ANSNET[8] 而来,被广泛地用在评估路由算法[9]。这里考虑 2 约束路径问题,权值 D 统一分布在[0,50]内,权值 C 统一分布在[0,200]内。通过改变代价和时延约束来获得不同的实验数据,得到的数据如下,为了便于表示,用 $C1$ 表示 $D \in [100,115]$,$C \in [400,460]$;$C2$ 表示 $D \in [50,65]$, $C \in [200,260]$;$C3$ 表示 $D \in [75,90]$,$C \in [300,360]$;$C4$ 表示 $D \in [125,140]$,$C \in [500,560]$;$C5$ 表示 $D \in [150,165]$, $C \in [600,660]$。

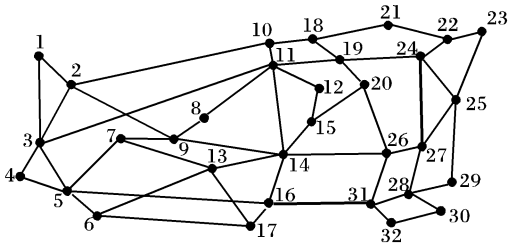

图 6　ANSNET 网络结构

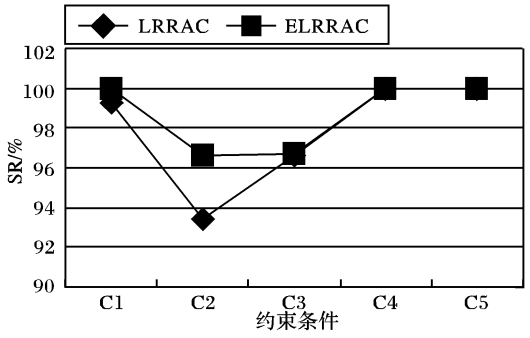

图 7　不同约束条件下的 SR 比较

SR 是请求的次数除以搜索的成功次数,SR 越高,算法性能越好。

(下转第 1110 页)

了其结构与特点,提出利用ToS字段来对关键业务流量进行控制的思想,并进行了实验验证。但由于ToS字段自身的限制,无法对流量进行更深层次的调度与控制。在DiffServ研究中,ToS字段已经扩充为DS字段,下一步工作可以考虑在DiffServ结构中通过DS字段来对更多类型的网络流量进行控制,并在队列中实现根据DS字段对数据包的区分对待。

参考文献:

[1] KELLY F P, MAULLOO A, TAN D, *et al.* Rate control for communication networks: Shadow prices, proportional fairness and stability [J]. Journal of the Operational Research Society, 1998, 49(3): 237 - 252.

[2] WANG WEI-HUA, PALANISWAMI M, LOW S H. Application-oriented flow control: Fundamentals, algorithms and fairness [J]. IEEE/ACM Transactions on Networking, 2006, 14(6): 1282 - 1291.

[3] LEE B K, KURIAN L, NPBENCH J. A benchmark suite for control plan and data plane application for network processors [C]// Proceedings of 21st International Conference on Computer Design (ICCD 2003). [S.l.]: IEEE Press, 2003: 226 - 233.

[4] Velocix. CacheLogic press and analyst presentation [EB/OL]. [2007 - 09 - 10]. http://www.cachelogic.com/research/.

[5] ALMSBERGER W. Linux traffic control-next generation [EB/OL]. [2007 - 09 - 15]. http://tcng.sourceforge.net/doc/.

[6] BROWN M A. Traffic control HOWTO, Version 1.0.2 [EB/OL]. [2007 - 09 - 18]. http://www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/pdf/Traffic-Control-HOWTO.pdf

[7] Agere systems proprietary. The challenge for next generation network processor [EB/OL]. [2007 - 09 - 16]. http://www.agere.com/docs/challenge_new.pdf

[8] LIU DUO, HUA BEI, HU XIANG-HUI, *et al.* High-performance packet classification algorithm for many-core and multithreaded network processor [C]// Proceedings of the 2006 International Conference on Compilers, Architecture and Synthesis for Embedded Systems. New York: ACM Press, 2006: 334 - 344.

[9] NIE XIAO-NING, UORDQVIST U, GAZSI L, *et al.* Network processors for access network (NP4AN): Trend and challenges [C]// Proceedings of the 2004 IEEE International SOC Conference. [S.l.]: IEEE Press, 2004: 265 - 269.

[10] ZHOU WEI-JIANG, LIN CHUANG, LI YIN, *et al.* Queue management for Qos provision build on network processor [C]// Proceedings of the 9th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS 2003). [S.l.]: IEEE Press, 2003: 219 - 224.

[11] Interl. Intel IXP2400 network processor data sheet [EB/OL]. (2004 - 02) [2007 - 08 - 26]. http://www.intel.com/design/network/datashts/301164.htm.

[12] GIORDANO S, PROCISSI G, ROSSI F, *et al.* Design of a multi-dimensional packet classifier for network processors [C]// Proceedings of the IEEE International Conference on Communications (ICC 2006). [S.l.]: IEEE Press, 2006: 503 - 508.

[13] Intel. Intel IXP2XXX product line of network processors [EB/OL]. [2007 - 09 - 10]. http://www.intel.com/cd/software/products/asmo-na/eng/index.htm.

(上接第1103页)

算法的执行时间如图8所示。

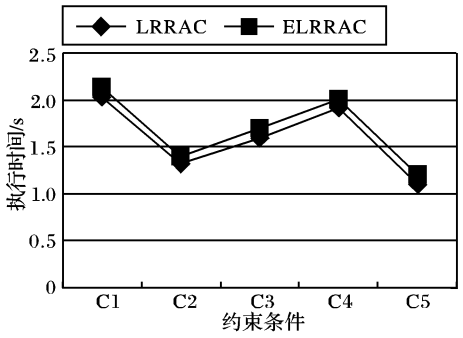

图8 不同约束下的执行时间比较

从图8可以看出,因为多了几次判断,算法的执行时间有所增加,但是增加的代价可以忽略不计。

4 结语

本文对拉格朗日松弛算法进行了研究,在深入分析拉格朗日松弛算法的基础之上,改进了次梯度算法中一个上下界相等的停止原则。由仿真可知该算法改进了原算法性能,并且没有提高算法的时间复杂度。

参考文献:

[1] CORMEN T H, LEISERSON C E, RIVEST R L, *et al.* Introduction to algorithms [M]. 2nd ed. New York: The MIT Press/McGraw-Hill Book Company, 2001.

[2] JAFFE J M. Algorithms for finding paths with multiple constraints [J]. Networks, 1984, 14(1): 95 - 116.

[3] AHUJA R K, MAGNANTI R L, ORLIN J B. Network flows: Theory, algorithms, and applications [M]. New Jersey: Prentice Hall, 1993.

[4] GAREY M R, JOHNSON D S. Computers and intractability, a guide to the theory of np-completeness [M]. New York: W H Freeman and Company, 1979.

[5] 徐凤生. 最短路径的求解算法[J]. 计算机应用, 2004, 24(5): 88 - 89.

[6] 周益民, 孙世新, 田玲. 一种实用的所有点对之间最短路径并行算法[J]. 计算机应用, 2005, 25(12): 2921 - 2922.

[7] HASSIN R. Approximation schemes for the restricted shortest path problem [J]. Mathematics of Operations Research, 1992, 17(1), 36 - 42.

[8] COMER D E. Internetworking with TCP / IP [M]. 3 rd ed. New York: Prentice Hall, 1995.

[9] CHEN SHI-GANG, NAHRSTEDT, K. On finding multi-constrained paths [C]// Proceedings of IEEE International Conference on Communications, 1998 (ICC'98). Washington, DC: IEEE Computer Society Press, 1998: 874 - 879.

[10] CHEN JING-CHAO. Efficient heuristic algorithms for finding multi-constrained paths [C]// Proceedings of IEEE International Conference on Communications, 2002 (ICC 2002). Washington, DC: IEEE Computer Society Press, 2002: 1074 - 1079.

[11] JÜTTNER A, SZVIATOVSZKI B, MECS I, *et al.* Lagrange relaxation based method for the QoS routing problem [C]// Proceedings of 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001). Washington, DC: IEEE Computer Society Press, 2001, 2: 859 - 868.

[12] 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.