

# 求解时间窗口网络中前 $k$ 条最短路径的方法

郭荣梅, 胡小兵

(中国民航大学电子信息与自动化学院, 天津 300300)

**摘 要:** 在一个时间窗口网络中寻找前  $k$  条最短路径是一项具有挑战性的任务. 在时间窗口网络中, 一个节点可能只有在某些特定的时间窗口内才能通行. 现有的研究大都假设运动体可以立即通过可通行节点, 或者在暂不可通行节点处等待直到未来时间窗口的开始时刻才通过. 本文针对一个更一般的时间窗口情况, 其中运动体一旦到达节点, 可以选择在节点的时间窗口中的任何离散时刻通过该节点. 本文将这样的时间窗口网络称为拓展时间窗口网络, 其解空间大小和复杂程度都显著增加. 通过模拟水面上的自然涟漪扩散现象, 本文提出了一种有效的涟漪扩散算法, 用于求解拓展时间窗口网络中的前  $k$  条最短路径. 除了一对一问题之外, 涟漪扩散算法 (ripple spreading algorithm, RSA) 还扩展到一对多问题. 在一对多问题中, 需要找到从给定起点到网络中的每个其他节点的所有前  $k$  条最短路径. 新方法具有最优性的理论保证, 其计算复杂度仅为  $O(k \times N_{\text{ATU}} \times N_L)$ , 其中  $N_L$  是网络中链接的数量,  $N_{\text{ATU}}$  是涟漪通过链接平均所需的仿真时间单位数. 实验结果证明了 RSA 的有效性.

**关键词:** 前  $k$  条最短路径问题; 拓展时间窗口网络; 涟漪扩散算法

**中图分类号:** TP301.6      **文献标识码:** A      **文章编号:** 0372-2112 (2020)07-1387-09

**电子学报 URL:** <http://www.ejournal.org.cn>

**DOI:** 10.3969/j.issn.0372-2112.2020.07.019

## An Effective Method to Find the $k$ Shortest Paths in a Generalized Time-Window Network

GUO Rong-mei, HU Xiao-bing

(College of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China)

**Abstract:** It is a challenging task to find the  $k$  shortest paths in a time-window network. A node may only be accessible within some specific time windows. In the existing researches, an assume is made that a traveller can pass through an accessible node immediately or wait until the next accessible time window. This paper targets a more general case where a traveller, once arrived at a node, may choose to pass through the node at any discrete times in the time windows of the node. In such a generalized time-window network, the complexity increases significantly, as the size of solution space soars up exponentially. By imitating the natural ripple-spreading phenomenon on a liquid surface, an effective ripple-spreading algorithm (RSA) is proposed for the  $k$  shortest paths problem in a generalized time-window network ( $k$ -SPPGTW). Besides one-to-one  $k$ -SPPGTW, the RSA is also extended to one-to-all  $k$ -SPPGTW, where all the  $k$  shortest paths from a given source to every other node in the network need to be found. The new method has a theoretically guaranteed optimality. The computational complexity of the RSA is  $O(k \times N_{\text{ATU}} \times N_L)$ , where  $N_L$  is the number of links in the network, and  $N_{\text{ATU}}$  is the average simulated time units for a ripple to travel through a link. The effectiveness and efficiency of the RSA for the  $k$ -SPPGTW are demonstrated by some preliminary experimental results.

**Key words:**  $k$  shortest paths problem; generalized time-window network; ripple-spreading algorithm

### 1 引言

最短路径问题 (Shortest Path Problem, SPP) 是图论和智能计算研究中的一个经典和重要问题<sup>[1]</sup>. 路径优

化具有很强的研究与应用价值, 在许多控制, 决策和管理问题当中也发挥着重要的作用, 如车辆路径<sup>[2]</sup>, 物流配送<sup>[3]</sup>, 蚁群算法<sup>[4,5]</sup>等. 前  $k$  条最短路径问题 ( $k$  Shortest Paths Problem,  $k$ -SPP) 是对第一最短路径问题的拓

展,它的目的是寻找路网中起点和终点之间的多个备选优化路径,形成最短路径组,以最大程度满足用户对不同路径的选择需求<sup>[6]</sup>.  $k$ -SPP 对模型评估,灵敏度分析和替代方案生成都有很大帮助,它具有广泛的应用背景,如生物学工程<sup>[7]</sup>,多目标优化<sup>[8]</sup>,机器人运动规划<sup>[9]</sup>,公路和电力线工程<sup>[10]</sup>,网络连接路由<sup>[11,12]</sup>等,长期以来一直受到研究人员的关注<sup>[13-19]</sup>.

文献[20]报告了一个关于时间窗口网络中的前  $k$  条最短路径问题的研究 ( $k$  Shortest Paths Problem with Time Window,  $k$ -SPPTW), 即节点只在给定的时间窗口内可以通行. 运动体可以选择立即通过一个节点(如果运动体在到达节点时该节点是可以通行的), 或者在节点处等待(如果运动体在到达节点时该节点是不可通行的), 直到未来时间窗口的开始时刻才能通过该节点. 与文献[20]不同, 本文将考虑运动体一旦到达节点, 可以选择在节点可通行的任何离散时刻通过该节点. 事实上,  $k$ -SPPTW 只是本文所提出的新问题的一个特例, 所以将新问题称为在拓展时间窗口网络中求解前  $k$  条最短路径问题 ( $k$  Shortest Paths Problem with Generalized Time Window,  $k$ -SPPGTW).

自 1950 年以来, 已经有许多用于求解经典  $k$ -SPP 的算法, 例如文献[13~19], 这些算法大都需要在计算第  $k$  条最短路径之前确定前  $(k-1)$  条最短路径, 因为经典  $k$ -SPP 的最优性原理指出: 第  $k$  条最短路径可由根据前  $j$  条最短路径所重构的路网而推导得出,  $j < k$ <sup>[4]</sup>. 本文要使用一种新的算法, 它不需要前  $(k-1)$  条最短路径的信息来计算第  $k$  条最短路径, 也不需要重构路网<sup>[13,14]</sup>. 新算法通过模拟水面上的自然涟漪扩散现象, 当有  $k$  个涟漪到达终点时将能找出前  $k$  条最短路径, 新算法被称为  $k$ -SPP 的涟漪扩散算法 (Ripple Spreading Algorithm, RSA)<sup>[13,14]</sup>. 本文旨在将 RSA 扩展到  $k$ -SPPGTW, 以获得比文献[20]更有效的方法. 文献[20]仅考虑了一对一  $k$ -SPPTW 问题, 即找到从给定起点到给定终点的  $k$  条最短路径. 除了一对一的  $k$ -SPPGTW, 本文还将尝试一对多  $k$ -SPPGTW, 即仅需运行一次 RSA 算法, 就能求解出从给定起点到路网中每个其他节点的所有前  $k$  条最短路径.

## 2 拓展时间窗口网络的数学模型

假设路网  $G(V, E)$  由节点集  $V$  和链接集  $E$  组成,  $V$  中包括起点和终点在内的  $N_N$  个不同节点, 并且链接集  $E$  在节点之间具有  $N_L$  条链接. 该路网可以记录为  $N_N \times N_N$  的邻近矩阵  $A$ , 矩阵  $A(i, j) = 1, i = 1, \dots, N_N, j = 1, \dots, N_N$ , 定义了一条链接, 即从节点  $i$  到节点  $j$  的有向链接, 否则,  $A(i, j) = 0$  表示从节点  $i$  到节点  $j$  没有有向链接. 本文假设  $A(i, i) = 0$ , 即不允许自连接链接. 每个链

接  $A(i, j)$  有一个成本  $C(i, j)$ , 本文中为通过该链接所需要的时间.

在时间窗口网络中, 每个节点都有一些时间窗口, 节点只有在时间窗口内是可以通行的. 设  $W_i(w, 1)$  和  $W_i(w, 2)$  分别记录节点  $i$  的第  $w$  个时间窗口的开始时刻和结束时刻,  $w = 1, \dots, N_{TW}(i)$ ,  $N_{TW}(i)$  是节点  $i$  的时间窗口总数. 例如, 对于图 1 中的节点 A,  $W_A = [3, 5; 7, 11]$  表示节点 A 有 2 个时间窗口, 它们确定节点 A 只能在时间段  $[3, 5]$  和  $[7, 11]$  期间通行. 在本文中, 节点的可通行性是基于离散时间单位定义的. 所以图 1 中的节点 A 实际上可以在离散时刻  $t = 3, 4, 5, 7, 8, 9, 10, 11$  时通行. 假设运动体在时间  $t = 5$  到达节点 A, 那么运动体可以选择在时刻  $t = 5, 7, 8, 9, 10$  或 11 时通过节点 A (需要强调的是, 在文献[20]的  $k$ -SPPTW 框架中, 运动体只可以在时刻  $t = 5$  或 7 时通过节点 A; 所以说, 本文的  $k$ -SPPGTW 远比  $k$ -SPPTW 复杂).

假设候选路径记录为向量  $R$ , 元素  $R(i, 1) = j$  表示节点  $j$  是路径  $R$  中的第  $i$  个节点,  $i = 1, \dots, N_p$  和  $j = 1, \dots, N_N$ , 其中  $N_p$  为路径  $R$  上的节点数目 (包括起点和终点). 显然,  $R(1, 1)$  是起点,  $R(N_p, 1)$  是终点. 元素  $R(i, 2)$  是选择通过路径  $R$  中的第  $i$  个节点的离散时刻. 本文不允许循环路径, 这意味着没有节点可以多次出现在同一条路径中, 即

$$R(i, 1) \neq R(j, 1), \text{ 如果 } i \neq j, i = 1, \dots, N_p, j = 1, \dots, N_p \quad (1)$$

对于给定路径  $R$  中的两个连续节点, 比如节点  $R(i, 1)$  和  $R(i+1, 1)$ ,  $i = 1, \dots, N_p - 1$ , 最早通过节点  $R(i+1, 1)$  的时间是

$$t_{ET, R(i+1, 1)} = \max(R(i, 2) + C(R(i, 1), R(i+1, 1)), \min(W_{R(i+1, 1)}(w, 1))) \quad (2)$$

其中  $w$  是节点  $R(i+1, 1)$  的第  $w$  个时间窗口, 满足

$$R(i, 2) + C(R(i, 1), R(i+1, 1)) \leq W_{R(i+1, 1)}(w, 2) \quad (3)$$

需要注意的是: 式(2)中的  $t_{ET, R(i+1, 1)}$  是通过节点  $R(i+1, 1)$  的可能的最早时间. 运动体可以选择在时刻  $t_{ET, R(i+1, 1)}$  通过节点  $R(i+1, 1)$ , 也可以选择等待直到后面的其它可通过时刻. 事实上, 通过节点  $R(i+1, 1)$  的时间  $R(i+1, 2)$  可以是满足以下条件的任何离散时刻

$$R(i+1, 2) = t_{PT, R(i+1, 1)} \geq t_{ET, R(i+1, 1)} \quad (4)$$

$$t_{PT, R(i+1, 1)} \in \bigcup_{w \text{ 满足式(3)}} [W_{R(i+1, 1)}(w, 1), W_{R(i+1, 1)}(w, 2)] \quad (5)$$

即, 可以选择在时刻  $t_{ET, R(i+1, 1)}$  通过节点  $R(i+1, 1)$ , 也可以选择节点  $R(i+1, 1)$  的所有时间窗口中的所有晚于时刻  $t_{ET, R(i+1, 1)}$  的任意一个离散时刻通过节点  $R(i+1, 1)$ .

根据式(2)~(5)的定义可知:运动体一旦到达一个节点,可以选择在该节点可通行的任何离散时刻通过该节点.

将路网中的第一最短路径定义为具有最小总成本(即最小旅行时间)的任何路径,如下:

$$f_{\min,1} = \min_{R \in \Omega} R(N_p, 2) \quad (6)$$

满足式(1)~(5)以及

$$A(R(i, 1), R(i+1, 1)) = 1, i = 1, \dots, N_p - 1 \quad (7)$$

其中  $\Omega$  是连接起点和终点的所有可行路径的集合.

对于任何  $k > 1$ , 令  $R_{SP,k}$  表示路网中具有第  $k$  最短旅行时间的路径.  $R_{SP,k}$  必须满足以下条件

$$R_{SP,k-1}(N_{SP,k-1}, 2) < R_{SP,k}(N_{SP,k}, 2) < R_{SP,k+1}(N_{SP,k+1}, 2) \quad (8)$$

根据上述数学模型的定义,  $k$ -SPPGTW 旨在从  $\Omega$  中找出  $R_{SP,1}, R_{SP,2}, \dots, R_{SP,k}$ . 为了实现这一点, 现有算法通常需要依次计算  $R_{SP,1}, R_{SP,2}, \dots, R_{SP,k}$ , 并且对于任何  $1 < j \leq k$ ,  $R_{SP,1}$  到  $R_{SP,j-1}$  的信息是计算  $R_{SP,j}$  必不可少的. 例如, 在文献[20]中, 就是通过比较那些删除前  $(j-1)$  条最短路径中所包含的链路的所有可能组合而重新构建的路网中的所有第一条最短路径来求解第  $j$  条最短路径. 本文针对  $k$ -SPPGTW 所提出的涟漪扩散算法(RSA)将以一种完全不同于文献[20]的方式工作, 不需要前  $(j-1)$  条最短路径的信息或路网重构.

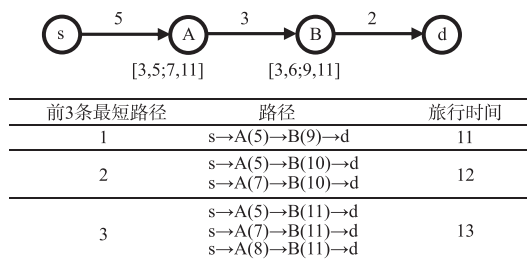


图1  $k$ -SPPGTW示意图

图1给出了  $k$ -SPPGTW 的一个图例. 在图1中, 根据经典  $k$ -SPP 的观点(即没有考虑节点处的时间窗口), 从起点  $s$  到终点  $d$  只有1条路径. 根据  $k$ -SPPTW, 则有2条不同的路径, 即  $s \rightarrow A(5) \rightarrow B(9) \rightarrow d$  (该路径意味着在时刻  $t=5$  离开节点  $A$ , 并且在时刻  $t=9$  离开节点  $B$ ), 和  $s \rightarrow A(7) \rightarrow B(10) \rightarrow d$  (该路径意味着在时刻  $t=7$  离开节点  $A$ , 并且在时刻  $t=10$  离开节点  $B$ ). 然而在本文提出的  $k$ -SPPGTW 框架中, 实际上有6条不同的路径, 其中有1条是第一最短的, 有2条是第二最短的, 其余3条是第三最短的. 显然本研究中的  $k$ -SPPGTW 比  $k$ -SPPTW 具有更大的解空间. 实际上, 前者涵盖了后者, 而后者只是前者的一个特例, 因此, 经典  $k$ -SPP 方法<sup>[13-19]</sup>和  $k$ -SPPTW 方法都不能解决本文所提出的  $k$ -SPPGTW 问题.

### 3 $k$ -SPPGTW 的涟漪扩散算法(RSA)

#### 3.1 RSA 的基本思想

文献[14]阐述了在自然涟漪扩散现象中所反映的优化原理, 即: 涟漪在水面以相同的速度向所有方向扩散, 因此根据空间点到涟漪中心点的距离, 涟漪将依次到达各个空间点, 并且总是首先到达最近的空间点. 基于这种涟漪扩散优化原理, 文献[14]提出了涟漪扩散算法(RSA)来求解第一最短路径问题. 文献[13,14]则将 RSA 进一步扩展用于求解经典  $k$ -SPP 问题.

简单地说, 用于求解第一最短路径的 RSA 就像是在路网上进行一场涟漪接力赛: 第一个涟漪从起点开始; 当涟漪到达直接连接但未访问过的节点时, 该节点将被激活以产生自己的涟漪; 当有涟漪第一次到达终点时, 涟漪接力赛停止, 第一最短路径就可通过回溯该涟漪的路径求得<sup>[14]</sup>; 在整个涟漪接力赛过程中, 所有涟漪始终以相同的预设恒定速度扩散.

涟漪扩散优化原理可以很容易地应用到节点上没有时间窗口的  $k$ -SPP. 在涟漪接力赛中寻找第一最短路径时, 每个节点可以产生最多1个涟漪; 为了找到前  $k$  条最短路径, 只需允许每个节点可产生多个涟漪. 让涟漪从起点开始, 并且仅沿着路网中节点之间的链接扩散, 一旦涟漪到达新节点, 它将在节点处触发新的涟漪(无论该节点之前是否已经产生过涟漪), 并且新的涟漪也将仅沿着链接扩散并在更远的节点处触发涟漪. 由于所有的涟漪以相同的速度扩散, 显然首先到达终点的涟漪就是沿着第一最短路径传播的, 而到达终点的第  $k$  个涟漪则是沿着第  $k$  条最短路径传播的. 在模拟涟漪接力赛时, 不需要计算从起点到任何中间节点的任何路径长度, 需要做的就是简单地模拟涟漪的扩散和激活, 即涟漪是否到达链接的末端节点, 以及哪个涟漪由哪个涟漪触发, 当涟漪一个接一个地到达终点时, 可以通过简单地回溯第  $k$  个到达终点的涟漪的传播路径, 就能得出第  $k$  条最短路径. 对于  $k$ -SPP 来说, RSA 和大多数现有方法之间的根本区别在于: 现有方法必须基于前  $(j-1)$  条最短路径信息重构路网并计算所有重构网络中的第一最短路径, 以便找到原始路网中的第  $j$  条最短路径, 直到  $j=k$ . 而 RSA 不需要这些路径信息, 也不需要重构路网, 只需在原始路网中模拟一次涟漪接力赛, 就可找出所有前  $k$  条最短路径. 关于求解经典  $k$ -SPP 问题的 RSA 的更多细节, 可参见文献[13,14].

#### 3.2 将 RSA 扩展到 $k$ -SPPGTW

为了将 RSA 扩展应用到本文的  $k$ -SPPGTW, 需要修改文献[13,14]中的 RSA 以允许涟漪的等待和共同触发行为. 也就是说, 需要给涟漪引入等待行为和共同触

发行为. 简单而言, 如果一个涟漪到达节点, 它可能在节点处触发多达  $k$  个新的涟漪, 并且这些新涟漪的开始扩散时间可以是节点可通行时的任意离散时刻, 换句话说, 新的涟漪将在节点处等待, 即在其分配的开始扩散时间到来之前不会开始扩散, 这就定义了涟漪的等待行为. 如果一个涟漪到达了一个已经由其他涟漪触发了一些等待状态的涟漪的节点, 那么就说这些等待状态的涟漪是由多个涟漪共同触发的, 需要记录哪些涟漪共同触发了节点上的等待状态涟漪, 这就定义了涟漪的共同触发行为. 通过引入等待和共同触发行为, RSA 可以扩展适用于  $k$ -SPPGTW. 需要注意的是, 一个激励涟漪可能在节点处触发多达  $k$  个新涟漪, 因为为了找到前  $k$  条最短路径, 理论上, 每个节点只需要产生不多于  $k$  个涟漪即可.

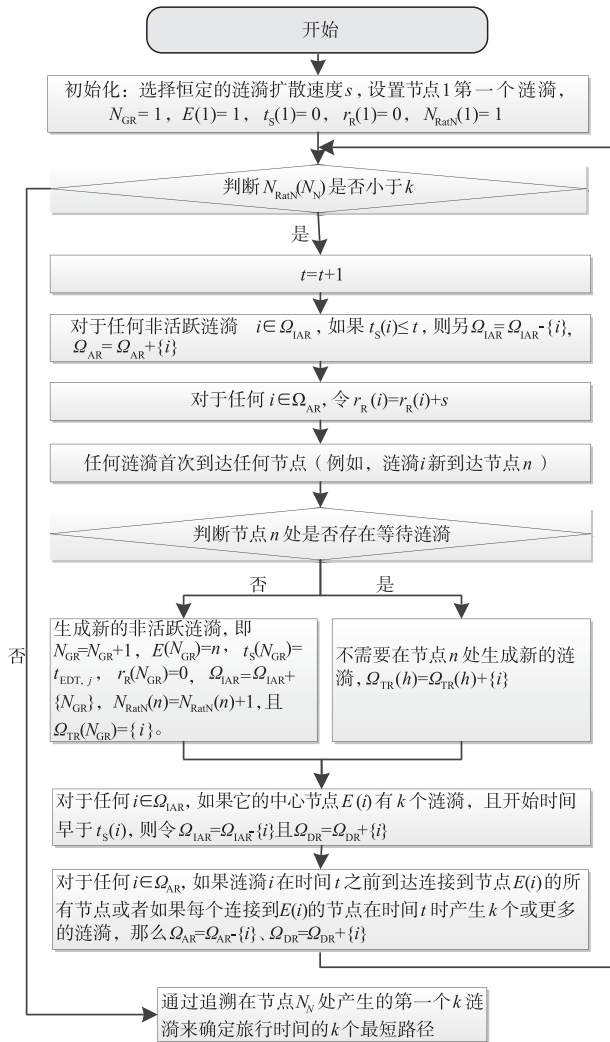


图2 RSA求解 $k$ -SPPGTW的流程图

求解一对一  $k$ -SPPGTW 的 RSA 的流程图见图 2. 图中假设起点和终点分别是节点 1 和节点  $N_N$ ,  $N_{GR}$  为目前所产生的涟漪数;  $E(i)$  为涟漪  $i$  的中心点节点,  $i = 1,$

$\dots, N_{GR}$ ; 设  $\Omega_{TR}(i)$  是共同触发涟漪  $i$  的涟漪集合;  $t_s(i)$  和  $r_r(i)$  表示涟漪  $i$  的开始时间和当前半径;  $\Omega_{IAR}, \Omega_{AR}$  和  $\Omega_{DR}$  分别表示非活跃涟漪集合, 活跃涟漪集合和消亡涟漪集合;  $N_{Rain}(n)$  为在节点  $n$  处生成的涟漪数,  $n = 1, \dots, N_N$ .

根据图 2 中的步骤, 如文献[14]中所证明的, 为了保证最优性, RSA 的涟漪扩散速度必须满足

$$0 < s \leq a_{\min} \quad (9)$$

其中  $a_{\min}$  是路网中的最小链接长度; 新的涟漪 (节点 1 处的第一个涟漪除外) 总是初始化为非活跃涟漪, 并且在开始时间到来之前不会开始扩散, 这就是涟漪的等待行为; 此外, 如果多个激励涟漪需要在某个节点的同一个可通行离散时刻触发该节点处的新涟漪, 就在该节点的该离散时刻仅共同触发一个新涟漪, 这引就是涟漪的共同触发行为. 因此, 上述 RSA 流程图引入了涟漪等待和共触发行为, 这对解决  $k$ -SPPGTW 至关重要.

基于图 1 的路网, 图 3 示例了 RSA 的涟漪接力赛如何求解出拓展时间窗口网络中的前  $k$  条最短路径. 这里为了简单起见, 设置  $k = 3$ , 即我们的目标是找到图 1 路网中的前 3 条最短路径. 在图 3 中, 假设终点 (即节点 d) 总是可通行的, 涟漪接力赛从起点 (即节点 s) 的涟漪 R1 开始, 涟漪 R1 在时刻  $t = 5$  到达节点 A, 并在节点 A 触发 3 个涟漪, 即 R2, R3 和 R4, 开始时刻分别为  $t_s(2) = 5, t_s(3) = 7$  和  $t_s(4) = 8$ . 因此, R2 从  $t = 5$  开始扩散, 而 R3 和 R4 则在节点 A 处等待. R1 在  $t = 5$  之后消失, 因为 R1 已到达所有连接到 R1 中心点节点的节点. R3 和 R4 分别在时刻  $t = 7$  和  $t = 8$  开始扩散, R2 在  $t = 8$  时到达节点 B, 在节点 B 触发 3 个涟漪, 即 R5, R6 和 R7, 开始时刻分别为  $t_s(5) = 9, t_s(6) = 10, t_s(7) = 11$ . 然后, 在  $t = 8$  之后 R2 消失. R5 在  $t = 9$  开始扩散, 而 R6 和 R7 仍然在节点 B 处等待. R3 在  $t = 10$  到达节点 B, 因此, R3 共同触发 R6 和 R7. 然后 R6 在  $t = 10$  时开始扩散. 在  $t = 10$  之后 R3 消失. R5 在  $t = 11$  时到达节点 d, 从而找到了第一最短路径:  $s \rightarrow A(5) \rightarrow B(9) \rightarrow d$ , 其旅行时间为 11. R7 仍在节点处等待. R4 在  $t = 11$  时到达节点 B, 因此, R4 共同触发 R7. R7 在  $t = 11$  开始扩散. 在  $t = 11$  之后, R4 和 R5 都消失. R6 在  $t = 12$  到达节点 d, 从而找到了第二最短路径:  $s \rightarrow A(5) \rightarrow B(10) \rightarrow d$  和  $s \rightarrow A(7) \rightarrow B(10) \rightarrow d$ , 两者的旅行时间都为 12. 在  $t = 12$  之后, R6 消失. R7 在  $t = 13$  到达节点 d, 从而找到了第三最短路径:  $s \rightarrow A(5) \rightarrow B(11) \rightarrow d, s \rightarrow A(7) \rightarrow B(11) \rightarrow d$  和  $s \rightarrow A(8) \rightarrow B(11) \rightarrow d$ , 它们的旅行时间都为 13.

从图 3 中可以清楚地看到 RSA 根本不需要重建任何路网或者在重建路网中重新求解第一最短路径. 利

用涟漪的等待和共同触发行为, RSA 只需要在最初给定的时间窗口网络中模拟一次涟漪接力赛, 就可以找

出前  $k$  条最短路径.

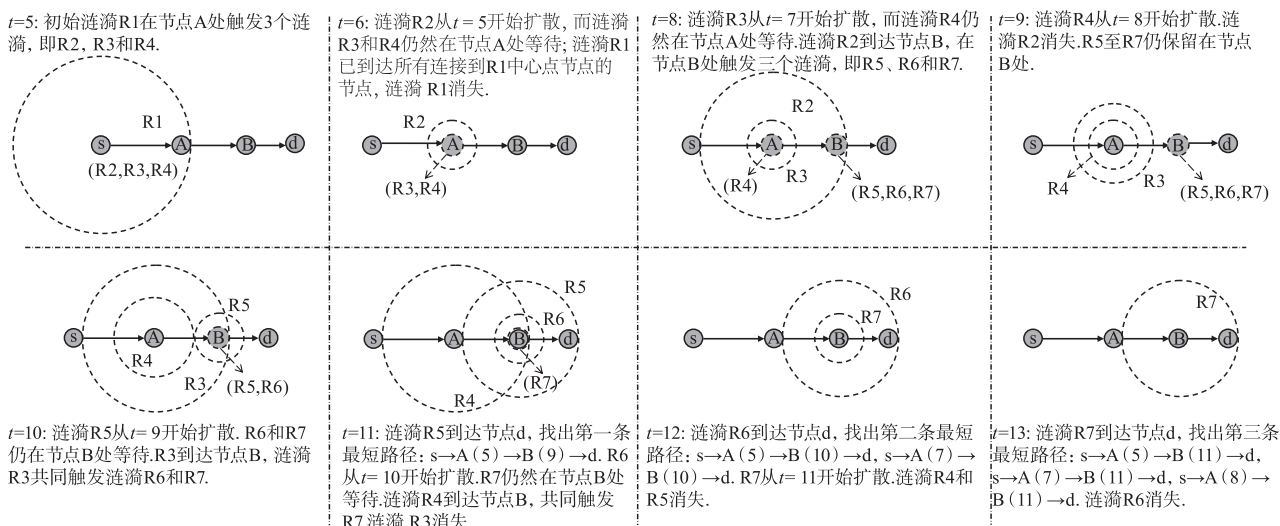


图3 RSA求解 $k$ -SPPGTW的涟漪接力赛

## 4 对 RSA 的理论分析

### 4.1 最优性

关于针对  $k$ -SPPGTW 的 RSA 的最优性有以下定理:

**定理** 在求解  $k$ -SPPGTW 问题的 RSA 的涟漪接力赛过程中, 第  $j$  个到达终点的涟漪决定了拓展时间窗口网络中的第  $j$  条最短路径,  $1 \leq j \leq k$ .

**证明** 求解  $k$ -SPPGTW 的 RSA 的最优性是由自然涟漪扩散现象中所反映的优化原理所保证的, 即: 涟漪在所有方向以相同的速度扩散, 因此它总是首先到达最近的点. 节点上的时间窗口不会影响这种涟漪扩散优化原则, 因为当一个节点变为可通行/不可通行状态时, 它对所有涟漪一视同仁都是可通行/不可通行. 所以, 参照文献[14]中所给出的第一最短路径的数学分析方法, 可以很容易地推导出在涟漪接力赛中第  $j$  个到达终点的涟漪决定了拓展时间窗口网络中的第  $j$  条最短路径,  $1 \leq j \leq k$ .

### 4.2 计算复杂性

求解  $k$ -SPPGTW 的 RSA 中的基本计算步骤是在一个仿真时间单位中沿着链接扩散涟漪. 基本计算步骤简单地包括加法运算和比较运算: 按涟漪扩散速度增加涟漪半径, 然后将新半径与链接的长度进行比较. 假设路网具有  $N_N$  个节点,  $N_L$  条链接 (因此, 每个节点平均具有  $N_L/N_N$  条链接); 并且假设涟漪平均需要  $N_{ATU}$  个仿真时间单位以扩散通过一条链接.

关于求解  $k$ -SPPGTW 的 RSA 的计算复杂性, 需要在最坏的情况下检查每个节点需要进行多少次基本计

算步骤. 由于起点仅产生一个涟漪, 因此平均需要  $N_{ATU} \times N_L/N_N$  个基本计算步骤; 终点不需要计算步骤; 对于其余  $(N_N - 2)$  个节点, 每个节点可以生成多达  $k$  个涟漪, 然后平均需要高达  $k \times (N_L/N_N - 1) \times N_{ATU}$  步长. 因此, RSA 大约需要执行  $N_{BCS}$  个基本计算步骤就能找到前  $k$  条最短路径

$$N_{BCS} = N_{ATU} \times N_L/N_N + (N_N - 2) \times k \times (N_L/N_N - 1) \times N_{ATU} < N_{ATU} \times N_L/N_N + k \times N_{ATU} \times N_L \quad (10)$$

对于大规模网络, 一般有  $N_L \gg N_L/N_N$ , 这意味着求解  $k$ -SPPGTW 的 RSA 的计算复杂度可以被评估为  $O(k \times N_{ATU} \times N_L)$ . 文献[20]中方法的计算复杂度为  $O(k \times r \times N_N^3)$ . 因为  $N_{ATU} \times N_L$  与  $r \times N_N$  相当, 所以很明显, 本文的 RSA 的计算复杂度远低于文献[20].

### 4.3 扩展应用到一对多的 $k$ -SPPGTW 问题

本文第 3 节中所提出的 RSA 仅针对一对一的  $k$ -SPPGTW 问题, 即在时间窗口网络中找到连接给定起点和终点的前  $k$  条最短路径. 这里将考虑一对多的  $k$ -SPPGTW 问题, 即需要在拓展时间窗口网络中找到从给定起点到每个其他节点的所有前  $k$  条最短路径.

传统的做法是: 一对多的  $k$ -SPPGTW 问题可以分解为  $(N_N - 1)$  个单独的一对一  $k$ -SPPGTW 问题, 因此可以应用针对一对一  $k$ -SPPGTW 问题的方法进行  $(N_N - 1)$  次运算来求解.

事实上, RSA 在求解一对多的  $k$ -SPPGTW 问题时, 并不需要运行  $(N_N - 1)$  次, 而是仍旧只需要在原始路网中进行一次涟漪接力赛, 然后就可以找到从给定起点到时间窗口网络中每个其他节点的所有前  $k$  条最短路径. 为实现这一目标, 只需要修改 3.2 小节中 RSA 的个

别步骤,即改变涟漪接力的终止条件.在求解一对一  $k$ -SPPGTW 问题时,当有  $k$  个或更多涟漪到达给定终点时,涟漪接力赛就会终止.而在求解一对多  $k$ -SPPGTW 问题时,除非每个节点(除了给定的起点)都已经产生了  $k$  个涟漪,或者所有涟漪都已经消失,否则涟漪接力赛将一直进行下去.

虽然一对多  $k$ -SPPGTW 可被视为  $(N_N - 1)$  个单独的一对一  $k$ -SPPGTW,但很容易推导出,求解一对多  $k$ -SPPGTW 的 RSA 具有与求解一对一  $k$ -SPPGTW 的 RSA 完全相同的计算复杂度,即  $O(k \times N_{ATU} \times N_L)$ .这是因为在第 4 小节中分析针对一对一  $k$ -SPPGTW 的 RSA 的计算复杂度时,我们使用了最坏的情况来评估计算量的上界;在这种最坏的情况下,每个节点(除了起点之外)都产生了  $k$  个涟漪,这与求解一对多  $k$ -SPPGTW 的 RSA 的涟漪接力赛正好相同.

## 5 实验结果

现有文献中缺少  $k$ -SPPGTW 的测试问题.本节参考文献[20]中  $k$ -SPPTW 的测试路网,然而  $k$ -SPPTW 只是本文所提出的  $k$ -SPPGTW 的一个特例,所以,必须对文献[20]的测试路网进行一些必要的修改,以使其适合本研究的实验需求.通过参考和修改  $k$ -SPPTW 的测试路网,得到了如图 4 和图 5 两个测试案例问题.其中图 4 较为简单,图 5 较为复杂,图中的每条链接都有自己指定的方向.

实验中,通过运行仅仅一次针对一对多  $k$ -SPPGTW 的 RSA,旨在找到从起点  $s$  到网络中的每个其他节点的所有前 3 条最短路径.需要注意的是,对于图 4 中的节点 A 和节点 B,以及图 5 中的节点 A,各有总共 1 条可行路径.运行 RSA 的结果见表 1 和表 2.

通过对图 4 和图 5 中路网进行理论分析,可以看出所对应的表的结果在理论上均为正确的,这证明了本文提出的 RSA 算法对求解  $k$ -SPPGTW 的有效性.

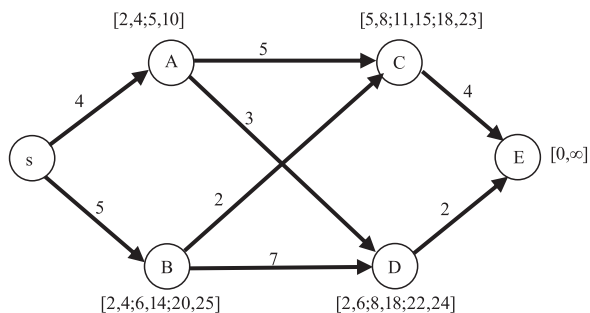


图4 实验所用的时间窗口网络1

由于本文提出的  $k$ -SPPGTW 是一个全新的、更为复杂的问题,现有的其它方法(例如文献[20]的方法)求解不了,所以不能直接用于对比实验.事实上,

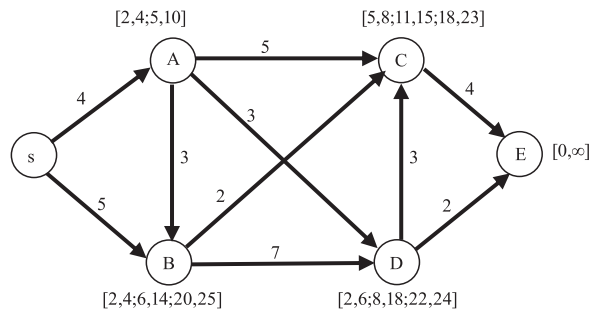


图5 实验所用的时间窗口网络2

由于  $k$ -SPPTW 的特殊性,文献[20]也只给出了一个类似图 4 的简单测试实验,而没有与其它方法的对比实验.为了更好地展示本文新方法与文献[20]方法的差异和特点,这里我们基于图 4 和图 5 的网络给出一些与文献[20]方法的近似对比实验.所谓近似对比实验,是指在求解图 4 和图 5 的问题时,文献[20]方法按其自身的  $k$ -SPPTW 定义只能选择时间窗口的开始时间通过节点(例如,假如文献[20]方法在时刻  $t=6$  到达图 4 中的节点 C,则其可以选择在时刻  $t=6, 11$  或  $18$  通过节点 C),而本文方法则可以选择时间窗口内的任意离散时刻通过节点(例如,假如本文方法在时刻  $t=6$  到达图 4 中的节点 C,则其可以选择在时刻  $t=6 \sim 8, 11 \sim 15$  或  $18 \sim 23$  之间的任意离散时刻通过节点 C).所以,虽然同是图 4 和图 5 的网络,但是本文方法要求解的问题远比文献[20]方法要求解的问题复杂得多.文献[20]方法按  $k$ -SPPTW 定义求解图 4 和图 5 的结果见表 3 和表 4.对比 RSA 的表 1 和表 2 的结果,可见文献[20]方法所求解  $k$ -SPPTW 的答案总体要简单.因为图 4 和图 5 的网络比较简单,文献[20]方法平均用了 0.63 秒求解一对多  $k$ -SPPTW,而 RSA 只用了约 0.01 秒求解一对多  $k$ -SPPGTW.下面进一步对比两种方法在图 4 和图 5 网络中的计算复杂度.根据文献[20]方法在求解一对一  $k$ -SPPTW 的复杂度公式  $O(k \times r \times N_N^3)$ ,其在图 4 和图 5 网络中求解一对多  $k$ -SPPTW 的计算复杂度分别为  $O(8640)$  和  $O(10800)$ .而本文的 RSA 在图 4 和图 5 网络中求解一对多  $k$ -SPPGTW 的计算复杂度分别为  $O(48)$  和  $O(57)$ .所以在图 4 和图 5 的算例中,两种方法的计算复杂度理论上相差  $10^2$  的级别.而两种方法实际运行时间相差不到  $10^2$  的级别,主要是因为实验是在 Matlab 环境中进行的,RSA 的 0.01 秒中可能受计算机后台程序运行的影响很大.表 1 到表 4 的结果已经表明 RSA 要求解的  $k$ -SPPGTW 比文献[20]方法所求解的  $k$ -SPPTW 的难度大,而 RSA 的计算复杂度却远小于文献[20]方法的计算复杂度.所以本文提出的方法具有明显的优势.

表 1 RSA 求解图 4 中从  $s$  到每个其它节点的前 3 条最短路径

终点	第 $i$ 条最短的路径	路径	旅行时间
A	1	$s \rightarrow A(4)$	4
B	1	$s \rightarrow B(6)$	6
C	1	$s \rightarrow B(6) \rightarrow C(8)$	8
	2	$s \rightarrow B(7) \rightarrow C(11)$ $s \rightarrow B(8) \rightarrow C(11)$ $s \rightarrow B(9) \rightarrow C(11)$ $s \rightarrow A(4) \rightarrow C(11)$ $s \rightarrow A(5) \rightarrow C(11)$ $s \rightarrow A(6) \rightarrow C(11)$	11
	3	$s \rightarrow A(7) \rightarrow C(12)$ $s \rightarrow B(10) \rightarrow C(12)$	12
D	1	$s \rightarrow A(4) \rightarrow D(8)$ $s \rightarrow A(5) \rightarrow D(8)$	8
	2	$s \rightarrow A(6) \rightarrow D(9)$	9
	3	$s \rightarrow A(7) \rightarrow D(10)$	10
E	1	$s \rightarrow A(4) \rightarrow D(8) \rightarrow E(10)$ $s \rightarrow A(5) \rightarrow D(8) \rightarrow E(10)$	10
	2	$s \rightarrow A(6) \rightarrow D(9) \rightarrow E(11)$	11
	3	$s \rightarrow A(7) \rightarrow D(10) \rightarrow E(12)$	12

表 2 RSA 求解图 5 中从  $s$  到每个其它节点的前 3 条最短路径

终点	第 $i$ 条最短的路径	路径	旅行时间
A	1	$s \rightarrow A(4)$	4
B	1	$s \rightarrow B(6)$	6
	2	$s \rightarrow A(4) \rightarrow B(7)$	7
	3	$s \rightarrow A(5) \rightarrow B(8)$	8
C	1	$s \rightarrow B(6) \rightarrow C(8)$	8
	2	$s \rightarrow B(7) \rightarrow C(11)$ $s \rightarrow B(8) \rightarrow C(11)$ $s \rightarrow B(9) \rightarrow C(11)$ $s \rightarrow A(4) \rightarrow C(11)$ $s \rightarrow A(5) \rightarrow C(11)$ $s \rightarrow A(6) \rightarrow C(11)$ $s \rightarrow A(4) \rightarrow B(7) \rightarrow C(11)$ $s \rightarrow A(5) \rightarrow B(8) \rightarrow C(11)$ $s \rightarrow A(6) \rightarrow B(9) \rightarrow C(11)$ $s \rightarrow A(4) \rightarrow D(8) \rightarrow C(11)$ $s \rightarrow A(5) \rightarrow D(8) \rightarrow C(11)$	11
	3	$s \rightarrow A(7) \rightarrow C(12)$ $s \rightarrow B(10) \rightarrow C(12)$ $s \rightarrow A(4) \rightarrow B(10) \rightarrow C(12)$ $s \rightarrow A(5) \rightarrow B(10) \rightarrow C(12)$	12

续表 2

终点	第 $i$ 条最短的路径	路径	旅行时间	
C	3	$s \rightarrow A(6) \rightarrow B(10) \rightarrow C(12)$ $s \rightarrow A(7) \rightarrow B(10) \rightarrow C(12)$ $s \rightarrow A(4) \rightarrow D(9) \rightarrow C(12)$ $s \rightarrow A(5) \rightarrow D(9) \rightarrow C(12)$ $s \rightarrow A(6) \rightarrow D(9) \rightarrow C(12)$	12	
	D	1	$s \rightarrow A(4) \rightarrow D(8)$ $s \rightarrow A(5) \rightarrow D(8)$	8
		2	$s \rightarrow A(6) \rightarrow D(9)$	9
		3	$s \rightarrow A(7) \rightarrow D(10)$	10
E	1	$s \rightarrow A(4) \rightarrow D(8) \rightarrow E(10)$ $s \rightarrow A(5) \rightarrow D(8) \rightarrow E(10)$	10	
	2	$s \rightarrow A(6) \rightarrow D(9) \rightarrow E(11)$	11	
	3	$s \rightarrow A(7) \rightarrow D(10) \rightarrow E(12)$	12	

表 3 文献[20]方法求解图 4 中从  $s$  到其它节点的前 3 条最短路径

终点	第 $i$ 条最短的路径	路径	旅行时间
A	1	$s \rightarrow A(4)$	4
B	1	$s \rightarrow B(6)$	6
C	1	$s \rightarrow B(6) \rightarrow C(8)$	8
	2	$s \rightarrow A(4) \rightarrow C(11)$ $s \rightarrow A(5) \rightarrow C(11)$	11
	3	$s \rightarrow B(20) \rightarrow C(22)$	22
D	1	$s \rightarrow A(4) \rightarrow D(8)$ $s \rightarrow A(5) \rightarrow D(8)$	8
	2	$s \rightarrow B(6) \rightarrow D(13)$	13
E	1	$s \rightarrow A(4) \rightarrow D(8) \rightarrow E(10)$ $s \rightarrow A(5) \rightarrow D(8) \rightarrow E(10)$	10
	2	$s \rightarrow B(6) \rightarrow C(8) \rightarrow E(12)$	12
	3	$s \rightarrow A(4) \rightarrow C(11) \rightarrow E(15)$ $s \rightarrow A(5) \rightarrow C(11) \rightarrow E(15)$ $s \rightarrow B(6) \rightarrow D(13) \rightarrow E(15)$	15

表 4 文献[20]方法求解图 5 中从  $s$  到其它节点的前 3 条最短路径

终点	第 $i$ 条最短的路径	路径	旅行时间
A	1	$s \rightarrow A(4)$	4
B	1	$s \rightarrow B(6)$	6
	2	$s \rightarrow A(4) \rightarrow B(7)$	7
	3	$s \rightarrow A(5) \rightarrow B(8)$	8

续表 4

终点	第 $i$ 条最短的路径	路径	旅行时间
C	1	s→B(6)→C(8)	8
	2	s→A(4)→C(11) s→A(5)→C(11) s→A(4)→B(7)→C(11) s→A(5)→B(8)→C(11) s→A(4)→D(8)→C(11) s→A(5)→D(8)→C(11)	11
	3	s→B(20)→C(22) s→A(4)→B(20)→C(22) s→A(5)→B(20)→C(22)	22
D	1	s→A(4)→D(8) s→A(5)→D(8)	8
	2	s→B(6)→D(13)	13
	3	s→A(4)→B(7)→D(14)	14
E	1	s→A(4)→D(8)→E(10) s→A(5)→D(8)→E(10)	10
	2	s→B(6)→C(8)→E(12)	12
	3	s→B(6)→C(11)→E(15) s→A(4)→C(11)→E(15) s→A(5)→C(11)→E(15) s→A(4)→B(7)→C(11)→E(15) s→A(5)→B(8)→C(11)→E(15) s→A(4)→D(8)→C(11)→E(15) s→A(5)→D(8)→C(11)→E(15) s→B(6)→D(13)→E(15)	15

## 6 结论

现有研究中关于时间窗口网络中的前  $k$  条最短路径问题 ( $k$ -SPPTW), 是假设运动体可以立即通过可通行节点, 或者在暂不可通行节点处等待直到未来时间窗口的开始时刻就立刻通过. 本文针对时间窗口网络的更一般情况 (即: 拓展时间窗口网络), 其中运动体一旦到达节点, 可以选择在节点时间窗口中的任何离散时刻通过该节点, 因此问题解空间呈指数级增长, 问题也具有更高的复杂程度. 为了解决拓展时间窗口网络中的前  $k$  条最短路径问题 ( $k$ -SPPGTW), 本文引入涟漪等待和共同触发行行为来改进文献 [13, 14] 中的涟漪扩散算法 (RSA), 并将 RSA 进一步扩展应用到一对多  $k$ -SPPGTW 问题, 求解从给定起点到网络中的每个其他节点的所有前  $k$  条最短路径. 本文对 RSA 的最优性和计算复杂性进行了理论分析, 然后给出一些实验结果以证明 RSA 对求解  $k$ -SPPGTW 的有效性. 未来工作将会更加关注新方法的应用研究, 例如, 通过结合文献 [8]

中利用计算各单一优化目标下前  $k$  个最优单目标解而求解完整 (非部分或近似) Pareto 前沿的多目标优化方法, 本文所提出的 RSA 将有助于求解更复杂的多目标优化问题的完整 Pareto 前沿.

## 参考文献

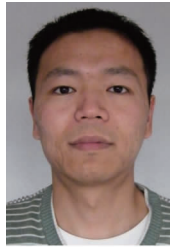
- [1] 邦迪 J A, 默蒂 U S R. 图论及其应用 (吴望名, 等译) [M]. 北京: 科学出版社, 1984.
- [2] 杜鹏楨, 唐振民, 等. 不确定环境下基于改进萤火虫算法的地面自主车辆全局路径规划方法 [J]. 电子学报, 2014, 42(3): 616–624.  
DU Peng-zhen, TANG Zhen-min, et al. Global path planning method for ALV based on improved glowworm swarm optimization under uncertain environment [J]. Acta Electronica Sinica, 2014, 42(3): 616–624. (in Chinese)
- [3] 郎茂祥, 胡思继. 用混合遗传算法求解物流配送路径优化问题的研究 [J]. 中国管理科学, 2002, 10(5): 51–56.  
LANG Maoxiang, HU Siji. Research on the optimization of logistics distribution routing problem with hybrid genetic algorithm [J]. Chinese Journal of Management Science, 2002, 10(5): 51–56. (in Chinese)
- [4] 赵娟平, 高宪文, 等. 移动机器人路径规划的改进蚁群优化算法 [J]. 控制理论与应用, 2011, 28(4): 457–461.  
ZHAO Juanping, GAO Xianwen, et al. Improved ant colony optimization algorithm for mobile robot path planning [J]. Control Theory & Applications, 2011, 28(4): 457–461. (in Chinese)
- [5] 李擎, 张超, 等. 一种基于粒子群参数优化的改进蚁群算法 [J]. 控制与决策, 2013, 28(6): 873–883.  
LI Qing, ZHANG Chao, et al. An improved ant colony algorithm based on particle swarm optimization [J]. Control and Decision, 2013, 28(6): 873–883. (in Chinese)
- [6] 高松, 陆锋.  $K$  则最短路径算法效率与精度评估 [J]. 中国图象图形学报, 2009, 14(8): 1677–1683.  
GAO Song, LU Feng. The  $K$ th shortest path algorithms: Accuracy and efficiency evaluation [J]. Journal of Image and Graphics, 2009, 14(8): 1677–1683. (in Chinese)
- [7] Dunn DA, WD Grover, MH MacGregor. Comparison of  $k$ -shortest paths and maximum flow routing for network facility restoration [J]. IEEE Journal on Selected Areas in Communications, 1994, 12(1): 88–99.
- [8] Hu XB, M Wang, E Di Paolo. Calculating complete and exact Pareto front for multi-objective optimization: a new deterministic approach for discrete problems [J]. IEEE Transactions on Systems, Man and Cybernetics, Part B, 2013, 43(3): 1088–1101.
- [9] 刘长安, 鄢小虎, 等. 基于改进蚁群算法的移动机器人动态路径规划方法 [J]. 电子学报, 2011, 39(5): 1120–1124.

- LIU Chang-an, YAN Xiao-hu, et al. Dynamic path planning for mobile robot based on improved ant colony optimization algorithm [J]. *Acta Electronica Sinica*, 2011, 39 (5): 1120 – 1124. (in Chinese)
- [10] 胡军红, 郭秀成, 等. 基于  $K$  最短路径的现代有轨电车线网优化 [J]. *东南大学学报*, 2017, 47 (6): 1275 – 1278.
- HU Junhong, GUO Xiucheng, et al. Optimization of modern tram network based on  $K$  shortest path [J]. *Journal of Southeast University*, 2017, 47 (6): 1275 – 1278. (in Chinese)
- [11] 齐小刚, 刘三阳. 一种基于  $K$  最短路径的 QoS 路由选择算法 [J]. *吉林大学学报*, 2005, 35 (5): 527 – 530.
- QI Xiaogang, LIU Sanyang. A QoS routing algorithm based on  $K$  shortest path [J]. *Journal of Jilin University*, 2005, 35 (5): 527 – 530. (in Chinese)
- [12] 刘佳, 夏少芳, 等. 复杂网络中最短  $K$  条路径问题的求解算法研究 [J]. *计算机应用*, 2008, 28 (4): 952 – 956.
- LIU Jia, XIA Shaofang, et al. Research on algorithm for solving the shortest  $K$ -path problem in complex networks [J]. *Journal of Computer Applications*, 2008, 28 (4): 952 – 956. (in Chinese)
- [13] Hu XB, M Wang, D Hu, MS Leeson, EL Hines, E Di Paolo. A ripple-spreading algorithm for the  $k$  shortest paths problem [A]. 3rd Global Congress on Intelligent Systems (GCIS 2012) [C]. Wuhan, 2012. 202 – 208.
- [14] Hu XB, M Wang, MS Leeson, EL Hines, E Di Paolo. A deterministic agent-based path optimization method by mimicking the spreading of ripples [J]. *Evolutionary Computation*, 2016, 24 (2): 319 – 346.
- [15] Hoffman W, R Pavley. A method for the solution of the  $N$ th best path problem [J]. *Journal of the Association for Computing Machinery*, 1959, 6: 506 – 514.
- [16] Yen JY. Finding the  $k$  shortest paths in a network [J]. *Management Science*, 1971, 17 (11): 712 – 716.
- [17] Eppstein D. Finding the  $k$  shortest paths [J]. *SIMA J Computation*, 1998, 28 (2): 652 – 673.
- [18] Aljazzar H, S Leue.  $K^*$ : A heuristic search algorithm for finding the  $k$  shortest paths [J]. *Artificial Intelligence*, 2011, 175 (18): 2129 – 2154.
- [19] 赵礼峰, 于汶雨. 一种求解  $K$  最短路径问题的新算法 [J]. *计算机技术与发展*, 2015, 25 (11): 67 – 70.
- ZHAO Lifeng, YU Wenyu. A new algorithm for solving  $K$  shortest path problem [J]. *Computer Technology and Development*, 2015, 25 (11): 67 – 70. (in Chinese)
- [20] Chen YL, Yang HH. Finding the first  $K$  shortest paths in a time-window network [J]. *Computers & Operations Research*, 2004, 31: 499 – 513.

### 作者简介



**郭荣梅** 女, 1996 年出生, 河南新乡人, 中国民航大学电子信息与自动化学院电子与通信工程专业硕士研究生, 主要研究方向为机场安全应急。  
E-mail: 2447927534@qq.com



**胡小兵 (通信作者)** 男, 1975 年出生, 四川攀枝花人, 中国民航大学电子信息与自动化学院教授、博士生导师, 主要研究方向为计算智能、复杂系统工程和综合风险管理。  
E-mail: huxb1975@163.com