**LEARNING MACHINE > RRATE ALGORITHM**

**Parameters:**

|  |  |
| --- | --- |
| \_N | When \_Xvaluepath i ∈ i-e pair = \_N, stop learning stage |
| \_K | Quantity of candidates (k shortest paths) of each i-e pair |
| \_K1, K2 | Moderation parameters (recommend = 0.5) |

**Tóm tắt:**

RRATE gồm 2 pha:

* Offline: Thực hiện trước khi request đến, tìm \_K đường đi ngắn nhất cho mỗi cặp nguồn đích lưu vào \_AllKPaths và khởi tạo giá trị đua cho mỗi đường (thường khởi tạo = 0) \_XValues
* Online: gồm 2 giai đoạn (lưu ý là tại mỗi request chỉ tính toán cho cặp nguồn đích tương ứng với request)
  + Learning stage:
    - Tính toán cost cho mỗi đường tìm được trong \_AllKPathi-epair .
    - Chọn trong số các đường đã tính toán, đường có Cost nhỏ nhất và đáp ứng được nhu cầu request .
    - Tăng giá trị đua cho đường được chọn đó.
  + Post-Learning stage: thực hiện sau khi có 1 đường đạt mức đua = \_N
    - Chọn đường có giá trị đua lớn nhất thoả nhu cầu request làm kết quả.

Độ đáp ứng thuật toán phụ thuộc vào các tham số của bài toán đã đề cập trên.

Chi tiết từng bước đã được mô tả bên dưới.

**OFFLINE PHASE**

Find k shortest path s foreach i-e pair.

\_AllKPaths[i-epair] = YenAlgorithm (i-epair, k)

Set foreach path found in previous step x value = 0

\_Xvalue[path] = 0

First request came

**ONLINE PHASE**

**Xvalue path ∈ i-e pair = \_N?**

Post-Learning Stage

true

Learning Stage

**LEARNING MACHINE > RRATE ALGORITHM > LEARNING STAGE**

Find min cut set of i–e pair of request:

*minCutSet = \_FordFulkerson.FindMinCutSet(i-e pair)*

Foreach path in \_AllKPathi-epair

i-epair = (Request.Source, Request .Destination)

Find max Residual Bandwidth:

*maxResidualBandwidth = Max(ResidualBandwidth)link ∈ path*

Compute Cost of path

*\_PathCost[path] = \_K1 \* size of(commonLink) + \_K2 / maxResidualBandwidth*

Find set of min cut link of path:

*commonLink = {link: minCutSet ∩ path}*

Find resultPath

*resultPath = min(costpath∈\_AllKPaths[i-epair],path.Bandwidth>=demand)*

Increase value of candidate

\_Xvalue[resultPath]++

**LEARNING MACHINE > RRATE ALGORITHM > POST - LEARNING STAGE**

**Find resultPath**

*resultPath = Max(\_XValuepath∈\_AllKPaths[i-epair],path.Bandwidth>=demand)*