

Résoudre le système :

$$\gamma_u^t = \sum_{(u,v) \in E} e^{-h_{u,v}^t} \gamma_v^t$$

avec : $\gamma_t^t = 1$

le schéma en bas de page représente le graph

voici le code java qui devrait correspondre au système d'équations:

```

//construction du système en bleue
//G.V() : nombre de noeuds
double[][][] gamma = new double[G.V()][G.V()][G.V()];
double[][] equivalent_number = new double[G.V()][G.V()];
//t: noeud destination
//u : noeud source
//v : noeud intermédiaire
for (int t = 0; t < G.V(); t++) {
    double matrix [][] = new double[G.V()][G.V()];
    final double[] constants = new double[(G.V())];
    for (int row=0; row < G.V(); row++) {
        for (int col=0; col < G.V(); col++) {
            matrix[row][col] = 0;
        }
    }
    for (int u = 0; u<G.V(); u++){
        if(t == u){
            constants[u] = 1;
        }
        else{
            constants[u] = 0;
        }
        if (u==t){
            matrix[u][t]=1;
            continue;
        }
        for (int v = 0; v < G.V(); v++) {
            if(u==v)
                matrix[u][v] = -1;
            else
                matrix[u][v]=Math.exp(-h[u][v][t]);

        }
    }
    final RealVector vector = new ArrayRealVector(constants);
    RealMatrix m = new Array2DRowRealMatrix(matrix);

    DecompositionSolver solver = new
org.apache.commons.math3.linear.LUDecomposition(m).getSolver();
    RealVector solution = solver.solve(vector);
    System.out.println("SOLVE SYSTEM For t: "+ t +"\n");
    for (int u = 0; u<G.V(); u++){
        Math.round((solution.getEntry(u)*100.00) /100.00)+"
\n");
    }
}
for (int u = 0; u<G.V(); u++){
    for (int t = 0; t<G.V(); t++){
        for (int v = 0; v<G.V(); v++){

            gamma[u][v][t] = equivalent_number[v]
[t]*Math.exp(-h[u][v][t]) ;
            // StdOut.print(" "+gamma[u][v][t]+ " ");

        }
    }
}
}

```

$$h_{u,v}^t = d_v^t + w_u^v - d_u^t$$

d_u^t The shortest distance from node u to node t. $d_t^t = 0$

$h_{u,v}^t$ Gap of shortest distance.

$d_v^t + w_u^v$ represents the distance from u to t when routed through neighboring node v.

calculer la fonction gamma pour un chemin P et une demande D:

$\Gamma_{PD}(h_{u,v}^t) = \gamma_v^t e^{-h_{u,v}^t}$ si $d_u^t \geq d_v^t$, sinon la valeur de la fonction est 0.

$e^{-h_{u,v}^t} \gamma_v^t$ (equivalent number)

exemple :

pour véhiculer un flot de 1 à 4 :

la fonction de distribution envoie une partie de 1->2->4 et une autre partie de 1->2->3->4

