2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

Weighted Clustering Coefficient Maximization For Air Transportation Networks

Julien Ponton[1], Peng Wei[2] and Dengfeng Sun[2]

Abstract— **In transportation networks the robustness of a network regarding nodes and links failures is a key factor for its design. At the same time, traveling passengers usually prefer the itinerary with fewer legs. The average clustering coefficient can be used to measure the robustness of a network. A high average clustering coefficient is often synonymous with a lower average travel distance and fewer number of legs. In this paper we present the average weighted clustering coefficient maximization problem, and give several solution methods based on branch and bound algorithm, dynamic programming and quadratically constrained programs.**

I. Introduction

An air transportation network (*ATN*) consists of distinct airports (cities) and direct flight routes between airport pairs [1]. We use a graph $G(V,E)$ to describe it, where the node set V represents all the N airports and the edge set E represents all the m direct flight routes between airports. We assume that this graph is weighted, and we use the *weighted adjacent matrix* $W = (w_{ij})_{i,j}$ to describe it. $w_{i,j}$ represents the total amount of traffic on the route (i,j), and $w_{ij} = 0$ if there is no route between airport i and airport j. In particular, $w_{ii} = 0$ and $w_{ij} = w_{ji}$ (in this paper the *ATN* is assumed to be symmetric). We also denote $A = (a_{ij})_{i,j}$ the *non-weighted adjacent matrix*. $a_{i,j}$ is simply defined as 1 if $w_{i,j} > 0$ and 0 otherwise. Finally, d_i represents the degree of the node i that is the number of its direct neighbors and t_i is the number of edges among its direct neighbors.

Air transportation networks (*ATN*s) have been widely studied [2], [3], [4]. In an *ATN*, reliability and well-connectivity of the flights routes are a major issue. Different metrics have been used to measure it: algebraic connectivity in [5], [6] and betweenness centrality in [7]. Here we propose to consider the average clustering coefficient (*ACC*) as a measurement of both robustness and well-connectivity of the network [8]. It has previously been shown in [9], [10], that the *ACC* is a proxy for increased robustness. Moreover, in an *ATN*, the robustness is above all local needs: when a flight route is deleted, companies want to re-route passengers with the fewest possible connections. And that is exactly what the weighted clustering coefficient defines: it shows the percentage of passengers that can be re-routed on other sides of every triangle [12]. Therefore the average weighted clustering coefficient can be seen as an interesting global measurement of local robustness for *ATNs*.

Contrary to other metrics that have been used to measure the robustness of a network, the *ACC* directly takes into account the average distance between two nodes. Indeed, a high average clustering coefficient in a given non-regular graph is often synonymous with a low average distance [10], and thus guarantees that the network is a "small-world", which has been proven to be important in the *ATN* [11]. Thus, in an *ATN*, where the average distance is very important and where the robustness is a local need, we won't need to add constraints to limit the maxium distance. This a major improvement and that's why we chose to study this new metric.

Figure 1 explains why a graph with a higher average clustering coefficient is more robust.

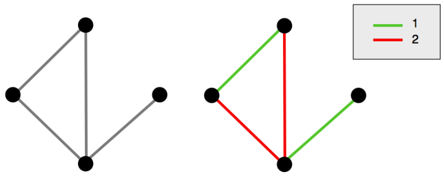

Fig. 1: $N = 4$, $m = 4$. *Left: non-weighted graph, Right: weighted graph. The right graph is more robust, because it will be possible to re-route more passengers in case of a link failure. The right graph is also the one with the highest average clustering coefficient (see figure 2).*

The (non-weighted) clustering coefficient of a node i, c_i, first defined by [8], is:

$$c_i = \begin{cases} 0 & \text{if } d_i = 0 \\ 1 & \text{if } d_i = 1 \\ t_i / \binom{d_i}{2} & \text{if } d_i \geq 2 \end{cases}$$

Using the non-weighted adjacent matrix A, we can rewrite c_i as:

$$c_i = \frac{1}{d_i(d_i-1)} \sum_{j,h} a_{ij} a_{ih} a_{jh} \qquad \text{when } d_i \geq 2 \tag{1}$$

According to [9], and using formulation in (1), we can define a weighted clustering coefficient of a node i, c_i^w as:

$$c_i^w = \frac{1}{(d_i-1)\sum_j w_{ij}} \sum_{j,h} \frac{w_{ij}+w_{ih}}{2} a_{ij} a_{ih} a_{jh}, \tag{2}$$

where $d_i \geq 2$($c_i^w = 0$ if $d_i = 0$ and 1 if $d_i = 1$).

To make computation easier, and using matrices definitions, we can rewrite c_i^w as:

$$c_i^w = \frac{(A^2 W)_{i,i}}{(d_i-1)\sum_j w_{ij}},$$

where $d_i \geq 2$.

Note that c_i^w is built to be like c_i in $[0,1]$, and if all the weights are the same, we have $c_i^w = c_i$. To avoid confusions, we can write $c_i^w(G)$. Then, the average weighted clustering coefficient $C^w(G)$ is:

$$C^w(G) = \frac{1}{N} \sum_i c_i^w$$

[1] J. Ponton is with the Department of Applied Mathematics, Ecole Polytechnique, 91120, Palaiseau, France julien.ponton at polytechnique.edu

[2] P. Wei and D. Sun are with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47907, USA weip,dsun at purdue.edu

978-3-952-41734-8/©2013 EUCA

As suggested in [10], and as we are considering only graphs of given order N, we can use the reduced average weighted clustering coefficient, $C_R^w(G)$ defined by:

$$C_R^w(G) = N \times C^w(G) = \sum_i c_i^w$$

The aim of this work is to determine the network which, for a given size m and order N, has the maximal average weighted clustering coefficient under constraints of limited air traffic on every flight route. We determine whether or not the structure of graphs with the highest average weighted clustering coefficient is similar to the one of non-weighted graphs, and present several algorithms to compute the optimal graph in both cases.

The rest of this paper is organized as follow. In section II, we model our problem and discuss its difficulty. In section III, we solve the average weighted clustering coefficient maximization problem in the homogeneous weights case. In section IV, we solve this problem when the weights are no longer homogeneous. Several applications to the *ATN* and several examples are presented in section III and IV. Section V concludes the paper.

II. Problem formulation and its NP-hardness

A. Problem formulation

The two following constraints will be applied to our maximization problem:

- For safety reasons and because the routes have a traffic throughput capacity, the edge weights have an upper bound β.
- Moreover, to establish a new flight route, we need a minimum amount of traffic demand α.

The Average Weighted Clustering Coefficient Maximization Problem for a given order N and a given size m can be written as follow:

$$\max_G C_R^w(G(V,E)) \quad s.t. \begin{cases} |V| = N \\ |E| = m \\ \forall (i,j),\, w_{ij} \in \{0, [\alpha, \beta]\} \end{cases} \tag{P}$$

We denote $C_{\max}^w(N,m)$ the value of the optimal solution in problem P, and $C_{\max}(N,m)$ the value of the optimal solution in the same problem when $\alpha = \beta$ (that is when the graph is unweighted).

B. NP-hardness

Suppose that we solve problem P in two-steps : first we find the optimal network structure design (i.e. the non-weighted graph), and then we weight the edges. The first step has been proven to be NP-hard in [10], thus, as a more difficult problem, problem P is NP-hard.

Note that even when the network structure is already fixed, the process of weighting the edges to maximize $C_R^w(G(V,E))$ is also a difficult problem as we will see in the next section.

III. Homogeneous weights

In this section, we study a relatively simple case of solving problem P, where β is sufficiently close to α.

A. Principle of solving problem

Here we give some useful properties, and the general principle of solving problem P.

Lemma 1: Let $G(N,m)$ be a weighted graph whose weights are in $[\alpha, \beta]$. Then:

$$\forall i \in \{1,2,...,N\}, \frac{\alpha}{\beta} c_i(G) \leq c_i^w(G) \leq \frac{\beta}{\alpha} c_i(G).$$

Proof:

$$\begin{aligned} c_i^w(G) &= \frac{1}{(d_i-1)\sum_j w_{ij}} \sum_{j,h} \frac{w_{ij}+w_{ih}}{2} a_{ij}a_{ih}a_{jh} \\ &\leq \frac{1}{(d_i-1)\sum_j \alpha a_{ij}} \sum_{j,h} \beta a_{ij}a_{ih}a_{jh} \\ &\leq \frac{\beta}{\alpha} \frac{1}{(d_i-1)d_i} \sum_{j,h} a_{ij}a_{ih}a_{jh} \\ &\leq \frac{\beta}{\alpha} c_i(G) \end{aligned}$$

And we can obtain the similar proof for the other inequality. ■

As a reminder, $C_{\max}(N,m)$ denotes the max in problem P when $\alpha = \beta$.

Lemma 2:

$$\forall (N,m),\ C_{\max}^w(N,m) \geq C_{\max}(N,m).$$

Proof: Let G be a weighted graph and G' the non-weighted graph with the same structure. As G' can be obtained from G by setting all its weights to a constant value between α and β, we have $\max_{w_{ij}} C_R^w(G) \geq C_R(G') = C_R(G)$ and $C_{\max}^w(N,m) \geq C_{\max}(N,m)$. ■

Theorem 1: Let G be the graph that maximizes the average non-weighted clustering coefficient for a given order N and a given size m. Let G' be the graph that maximizes the average weighted clustering coefficient for the same order and size. If β is close enough to α, then G and G' have the same structure.

Proof: Suppose that G and G' do not have the same structure. From lemma 1, we know that:

$$C_{\max}^w(N,m) = \max_w C_R^w(G') \leq \frac{\beta}{\alpha} C(G') < C(G) = C_{\max}(N,m)$$

The innequality $C_{\max}^w(N,m) < C_{\max}(N,m)$ is in contradiction with lemma 2. Thus, G and G' have the same structure. Note that this is true only if $\forall G' \neq G$, $\frac{\beta}{\alpha} C(G') < C(G)$. ■

The solution then follows from theorem 1. The weights and the structure of the network are uncorrelated. To find the optimal graph, we can use the method presented in [10] to find optimal network structure, and then optimize the weights using one of the methods presented below.

B. Binary case

Let's now simplify our problem, in which rather than considering that $w_{ij} \in \{0, [\alpha, \beta]\}$, we will consider $w_{ij} \in \{0, \alpha, \beta\}$. Thus, the weights of the edges can only take one of the 2 values: α or β. More formally, the binary problem can be defined as following:

Problem 1:

$$\max_G C_R^w(G(V,E))$$

$$s.t. \begin{cases} |V| = N \\ |E| = m \\ \forall (i,j),\, w_{ij} \in \{0, \alpha, \beta\} \end{cases}$$

In this case, we can see that the function $f_{N,m} : (\alpha, \beta) \mapsto C_{\max}^w(N,m)$ depends only on $\frac{\beta}{\alpha}$. So we can impose that $\alpha = 1$ for instance, and only modify the value of β. After computing the optimal non-weighted graph, which can be done in a pseudo-polynomial time (see [10]), we still have to weight the edges. To do so, a naive technique would be to try every possible combination, which complexity is $\theta(2^m)$. We will also present an heuristic to reduce this complexity. However, note that the binary case is a difficult problem, as it can be considered as an integer programming whose relaxation is NP-Hard, as we will see in the next section.

1) Branch and bound: To reduce this complexity, and as the maximization problem is not concave, we used a branch and bound algorithm: we weight the edges in an increasing order, and branch on the weight of the current edge: α or β. As for the (maximum) bound part of the algorithm, it is easy to see that we cannot reach a coefficient larger than the one obtained from the existing weighted graph, the rest of the edges being weighted to maximize the clustering coefficient. Thus, if we are currently branching on edge k, then the bound b_i for node i will be:

$$b_i = \frac{(A^2W^*)_{i,i}}{(d_i-1)(\sum_{j=1}^{k} w_{ij} + (d-k)\alpha)},$$

where

$$W^*_{p,q} = \begin{cases} \beta & \text{if } p = i \text{ and } q > k \\ \beta & \text{if } q = i \text{ and } p > k \\ W_{p,q} & \text{otherwise} \end{cases}$$

2) Increasing weights: Another useful property is given here, which reduces a lot of the complexity of the weighting process. Let $G_{\max}(N,m)$ be the optimal graph for the non-weighted case. Suppose, for instance, that we limit the number of edges of $G_{\max}$ that can be weighted with a maximum number m_β. We then try to maximize the weighted clustering coefficient on $G_{\max}$ with this new constraint. We denote $\Omega(G_{\max}, m_\beta)$ the sum of the weights of the edges of $G_{\max}$. Then we have of course $\Omega(G_{\max}, m_\beta) \leq m_\beta\beta + (m - m_\beta)\alpha$.

Lemma 3: If $\Omega(G_{\max}, m_\beta) < m_\beta\beta + (m - m_\beta)\alpha$, then the number of edges weighted with β in the optimal graph is lower than m_β.

Proof:

Suppose that the optimal graph G_{opt} has more than m_β edges weighted with β. Let G be the graph such that $\Omega(G, m_\beta) < m_\beta\beta + (m - m_\beta)\alpha$. We know that there is at least one edge, e, whose weight is α in G and β in G_{opt}. For instance, $e = (i,k)$. Now if we assume that $w_{i,j}$ is a continuous variable, we can compute the derivative of $c_i^w(G)$ with respect to w_e:

$$\begin{aligned} \frac{\partial c_i^w(G)}{\partial w_e} &= \frac{\partial c_i^w(G)}{\partial w_{ik}} \\ &= \frac{\sum_h a_{ik}a_{ih}a_{kh}\sum_l w_{il} - \sum_j\sum_h w_{ij}a_{ij}a_{ih}a_{jh}}{(d_i-1)(\sum_l w_{il})^2} \\ &= \frac{\sum_j\sum_h w_{ij}[a_{ik}a_{ih}a_{kh} - a_{ij}a_{ih}a_{jh}]}{(d_i-1)(\sum_l w_{il})^2} \\ &= \frac{\sum_j w_{ij}\gamma_{ij}}{(d_i-1)(\sum_l w_{il})^2}, \end{aligned}$$

where $\gamma_{ij} = \sum_h[a_{ik}a_{ih}a_{kh} - a_{ij}a_{ih}a_{jh}]$. Hence the sign of this derivative depends only on $\sum_j w_{ij}\gamma_{ij}$, which, if α is close to β, is always a constant sign. Then, as this term is supposed to be positive in G_{opt}, it is also positive in G, and thus we can assign β to edge e_1. Doing that for all the edges with α in G and β in G_{opt}, it will lead to a contradiction with the fact that $\Omega(G, m_\beta) < m_\beta\beta + (m - m_\beta)\alpha$.

■

3) Algorithm: We can now describe the algorithm: using lemma 3, we can set all the weights to α, and then set edges to β one-by-one. When we find that by switching a new edge from α to β, the average weighted clustering coefficient is decreasing, we stop. The algorithm is shown in Algorithm 1.

Algorithm 1 Binary Case

```
G ← Compute_the_optimal_non-weighted_graph (N,m)
C ← Compute_clustering_coefficient (G)
continue ← True
$m_\beta$ ← 1
while continue do
    G ← Compute_the_optimal_weighted_graph (G,$m_\beta$)
    $C_{new}$ ← Compute_the_weighted_clustering_coefficient (G)
    if $C_{new} \leq C$ then
        continue ← False
    else
        $m_\beta = m_\beta + 1$
        C ← $C_{new}$
    end if
end while
return [G, C]
```

Note that we apply the branch & bound method in function *Compute_the_optimal_weighted_graph*.

Figure 2 gives some examples of results that we found using Algorithm 1. Note that two nodes that are close in Figure 2 do not necessarily represent airports that are geographically close, and the length of the edges does not have geographically meaning.

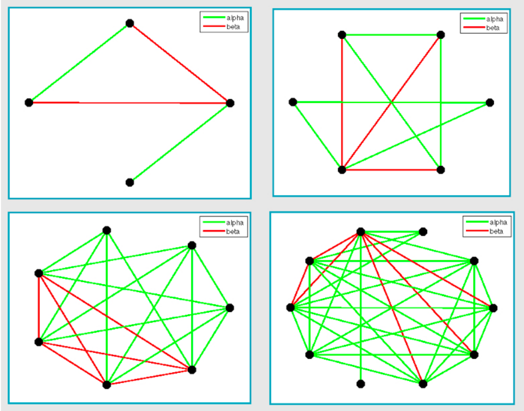

Fig. 2: *Examples of optimal graphs for binary case (problem 1). $\alpha = 1, \beta = 2$. Top left corner: $C_R^w(4,4) = 3.4$, $C_R(4,4) = 3.33$ - Top right corner: $C_R^w(6,9) = 5.4375$, $C_R(6,9) = 5.4$ - Bottom left corner: $C_R^w(7,19) = 6.6444$, $C_R(7,19) = 6.4667$ - Bottom right corner: $C_R^w(10,32) = 9,4107$, $C_R(10,32) = 9,2817$.*

C. General case

We now focus on the "relaxed" problem of the binary case, that is $w_{ij} \in \{0, [\alpha, \beta]\}$. This is a general problem, and we will begin with some properties on its difficulty and its solution method.

1) On the NP-Hardness of the problem: First, let's rewrite our problem:

$$\begin{aligned} C_R^w(G(N,m)) &= \sum_{i=1}^{N} \frac{1}{(d_i-1)\sum_j w_{ij}} \sum_{j,h} \frac{w_{ij}+w_{ih}}{2} a_{ij}a_{ih}a_{jh} \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{h=1}^{N} \frac{w_{ij}}{\sum_k w_{ik}} \frac{a_{ij}a_{ih}a_{jh}}{(d_i-1)} \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{w_{ij}}{\sum_k w_{ik}} b_{ij}, \end{aligned}$$

where $b_{ij} = \sum_{h=1}^{N} \frac{a_{ij}a_{ih}a_{jh}}{(d_i-1)}$. We know that the structure is already given, since it can be obtained from the non-weighted optimization with theorem 1. Thus, all b_{ij} are known. Now, we define w^*_{ij} and the vector $\overline{w}$ of size $2N^2$ as follow:

$$w^*_{ij} = \frac{w_{ij}}{\sum_k w_{ik}} \qquad \text{and} \qquad \overline{w} =$$
$$^t\left(\ w_{11}w_{12}\dots w_{1N}w_{21}\dots w_{nn}w^*_{11}w^*_{12}\dots w^*_{1N}w^*_{21}\dots w^*_{NN}\ \right)$$

The constraint $w^*_{ij} = \frac{w_{ij}}{\sum_k w_{ik}}$ can be rewritten as:

$$\begin{aligned} & \overline{w}_{N^2+N(i-1)+j} \sum_k w_{ik} = w_{ij} \\ \Leftrightarrow \quad & {}^t\overline{w} A^{ij} \overline{w} - {}^t c^{ij} \overline{w} = 0, \end{aligned}$$

where A^{ij} is a symetric matrix and c^{ij} a vector defined by:

$$A^{ij}_{kl} = \begin{cases} 1/2 & \text{if } k \in [\![N(i-1)+1, Ni]\!] \\ & \text{and } l = N^2+N(i-1)+j, \\ 1/2 & \text{if } l \in [\![N(i-1)+1, Ni]\!] \\ & \text{and } k = N^2+N(i-1)+j, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$c^{ij}_k = \begin{cases} 1 & \text{if } k = N(i-1)+j, \\ 0 & \text{otherwise.} \end{cases}$$

Finally, we define the vectors $\overline{b}, \overline{\alpha}, \overline{\beta}$ of size $2N^2$ and the matrix D of size $(N(N-1)/2 \times 2N^2)$ as:

$$\begin{aligned} \overline{b} &= {}^t\left(\ 0\dots 0 \quad b_{11}b_{12}\dots b_{1N}b_{21}\dots b_{NN}\ \right), \\ \overline{\alpha} &= \alpha \quad {}^t\left(\ a_{11}a_{12}\dots a_{1N}a_{21}\dots a_{NN} \quad 0\dots 0\ \right), \\ \overline{\beta} &= \beta \quad {}^t\left(\ a_{11}a_{12}\dots a_{1N}a_{21}\dots a_{NN}a_{11}a_{12}\dots a_{1N}a_{21}\dots a_{NN}\ \right), \end{aligned}$$

$$D\overline{w} = \begin{pmatrix} w_{12} - w_{21} \\ w_{13} - w_{31} \\ \vdots \\ w_{1N} - w_{N1} \\ w_{23} - w_{32} \\ \vdots \\ w_{N-1N} - w_{NN-1} \end{pmatrix}$$

Thus, our problem can be written as the following quadratically constrained linear program.

Problem 2:

$$\max_{\overline{w}} \overline{b}.\overline{w}$$
$$s.t. \begin{cases} \forall (i,j),\ {}^t\overline{w} A^{ij} \overline{w} - {}^t c^{ij} \overline{w} = 0 \\ D\overline{w} = \mathbf{0} \\ \overline{\alpha} \le \overline{w} \le \overline{\beta} \end{cases}$$

Theorem 2: Maximizing the average weighted clustering coefficient on a given graph structure is a NP-Hard problem.

Proof: It follows directly from problem 2 formulation and its quadratic constraints. ■

2) Elements of solution method: To solve problem 2, we rewrote the quadratic constraint using a second-order cone programming (SOCP) formulation. It is known that the general quadratic constraint:

$$^t x \, ^t A A x + {}^t b x + c \le 0$$

can be written as:

$$\left\| \begin{matrix} (1 + {}^t b x + c)/2 \\ Ax \end{matrix} \right\| \le (1 - {}^t b x - c)/2$$

Thus, the quadratic constraint can be written as a *SOCP*, and we can solve the whole problem using a *SOCP* solver. Here, we decided to use the matlab package *SDPT3* [13]. Another utility of this package is that it can solve the *lovasz theta number problem*, which is useful to compute the non-weighted optimal graph (see [10]). Figure 3 shows an interesting result of the relaxed problem.

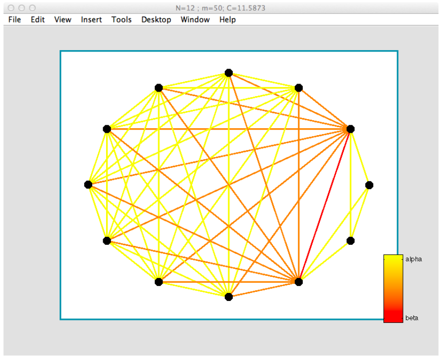

Fig. 3: *Example of an optimal graph for problem 2.* $\alpha = 1$, $\beta = 2$.

It shows in particular that the form of the *ATN* is preserved by maximizing the average weighted clustering coefficient: we find highly connected hubs, focus cities, and regional airports. The same structure is visible in most of the optimal graphs.

3) Application to regional airlines: For regional airlines in the USA, which are numerous and have to deal with a highly competitive market, robustness is a key point to provide an efficient service and avoid wasteful spendings. With a low number of destinations, a highly connected network, and routes with similar number of seats offered, these airlines fit perfectly in our model. Figure 4 presents some results for two relevant airlines: ExpressJet Airlines and AirTran Airways. This figure shows some interesting results: indeed, we see that the way of maximizing the average weighted clustering coefficient can be significantly different from a network to another. For AirTran Airways, the main idea is to redistribute the weights among the routes (which are more or less the same as in the real network); while for ExpressJet Airlines, the main idea is to redistribute the routes among all the airports in order to harmonize the whole network. In the real network of this airline, there are some airports connected with only a few others, but with a high traffic. And it is easy to understand that this is not really good for robustness, because it will be difficult to re-route passengers if one of these routes has a problem.

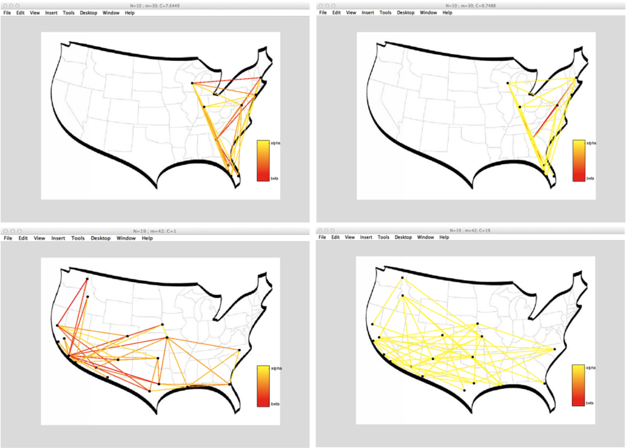

Fig. 4: *Example of regional airlines network optimization. Left: real network - Right: optimized network. Top: AirTran Airways - Bottom: ExpressJet Airlines. The weights are based on the number of flights over the year 2008 (data obtained from Research and Innovative Technology Administration).*

IV. Non-homogenous weights

In this section, we study the non-homogenous weights problem. When β is not close to α, we can no longer assume that the optimal graph has the same structure as the non-weighted case. Thus, it will be much more difficult to find the optimal graph, because we will have to optimize both the structure and the weights at the same time.

A. Dynamic programming formulation

In [10], the authors use a dynamic programming formulation to find the optimal graph in the non-weighted case. Using the same idea, we can dynamically find the optimal weighted graph. Let v be a node of the graph $G(V,E)$ connected to a subset D of V ($|D| = d$), and $G'(V',E')$ the same graph without the node v, nor its d connections. Obviously, we have $|V'| = N-1$ and $|E'| = m-d$. As in [10], we denote $\Delta C^w_R(N-1,m-d,D)$ the change in the weighted clustering coefficient of the d nodes of the set D of the graph G' when we connect the node v to them. More formally, $\Delta C^w_R(N-1,m-d,D) = \sum_{i\in D} c^w_i(G) - c^w_i(G')$. Then, we have:

Theorem 3: Let V be a set of N nodes, and v a node in V, indexed by j.

$$C^w_{\max}(N,m) = \max_{D\subset V\setminus\{v\}}\{C^w_{\max}(N-1,m-d) + \max_{w_{ij}\in[\alpha,\beta],\, i\in D}\{c^w_j + \Delta C^w_R(N-1,m-d,D)\}\}$$

Proof: Let $G(V,E)$ be a graph of order N and size m. Let v be a node in V, indexed by j. We have:

$$C^w_R(G) = \sum_{i=1}^{N} c^w_i = (\sum_{\substack{i=1\\ i\neq j}}^{N} c^w_i) + c^w_j$$

and it folllows that:

$$C^w_{\max}(N,m) = \max_{G(N,m)}\{\sum_{\substack{i=1\\ i\neq j}}^{N} c^w_i + c^w_j\}$$

Thus, $C^w_{\max}(N,m)$ is obtained by connecting optimally a new node to a given graph of order $N-1$ and size $m-d$, whose average weighted clustering coefficient is already maximum. Hence:

$$\begin{aligned} C^w_{\max}(N,m) &= \max_{\substack{D\subset V\setminus\{v\}\\ w_{ij}\in[0,\{\alpha,\beta\}]}} \{C^w_{\max}(N-1,m-d) \\ &\quad + c^w_j + \Delta C^w_R(N-1,m-d,D)\} \\ &= \max_{D\subset V\setminus\{v\}}\{C^w_{\max}(N-1,m-d) \\ &\quad + \max_{w_{ij}\in[\alpha,\beta],\, i\in D}\{c^w_j + \Delta C^w_R(N-1,m-d,D)\}\} \end{aligned}$$

■

Thanks to theorem 3, we can now compute the optimal graph with a dynamic programming formulation. Figure 5 shows an example of this technique an reveals that the optimal network preserves the difference of traffic among the routes.

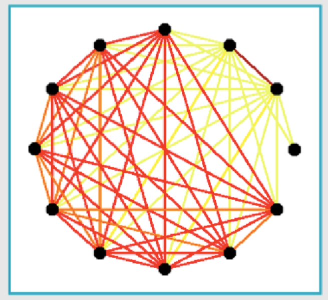

Fig. 5: $N = 12$, $m = 57$. *Exact solution for non-homogeneous weights.*

However, we still need to test every possible combination of D and w_{ij}, which is really costly. Again, for the binary case, we can use a branch & bound algorithm which will reduce the time needed. However, even with that technique, we cannot go farther than graphs with 14 or 15 nodes.

B. Algorithm

The complete algorithm for the binary case is given below. We used a branch and bound algorithm during the step-by-step weighting process.

Algorithm 2 Dynamic Programming Algorithm

```
$Graph \leftarrow Array(N,M)$
**for** $i = 1 \rightarrow N$ **do**
  **for** $j = 1 \rightarrow m$ **do**
    $C \leftarrow 0$
    $bound \leftarrow 0$
    **for all** $D$ subset **do**
      $G \leftarrow Graph(i-1, j-d)$
      **while** $Branch\_bound(G,D) \geq bound$ **do**
        $G \leftarrow Weighting\_process(G,D)$
      **end while**
      $C' \leftarrow Compute\_coefficient(G)$
      **if** $C' > C$ **then**
        $C \leftarrow C'$
        $Graph(i,j) \leftarrow G$
      **end if**
    **end for**
  **end for**
**end for**
**return** $Graph(N,m)$
```

C. Variation with β

As it is very difficult to get the optimal solution for bigger graphs, we present here a minimal bound which can be seen as an approximation of the average weighted clustering coefficient. To find this bound, we will study the variation of $C^w(G)$ with β. As in section *III.B*, we will use the function $f_{N,m} : (\alpha, \beta) \mapsto C^w_{\max}(N, m)$.

Lemma 4: $\forall\,(N, m), \quad f_{N,m}$ increases with β.

Proof: If $\beta_1 \leq \beta_2$, then $\beta_1 \in [\alpha, \beta_2]$. Thus, when β_2 is the maximum bound, the optimal graph with β_1 as the maximum bound is an acceptable graph in the maximization problem. ■

The next theorem is then obvious:

Theorem 4: The value of problem P for homogeneous weights is a minimal bound of the value of problem P for non-homogeneous weights.

Hence for big graphs, where it is impossible to compute the exact optimal, we can still use this approximation, which is easier to compute as seen in section 3. A natural question is whether or not this bound is a good approximation. To study this problem, we computed and compared the exact and the approximated optimal graph for different sizes, orders, and above all for different values of α and β. Figure 6 presents the results.

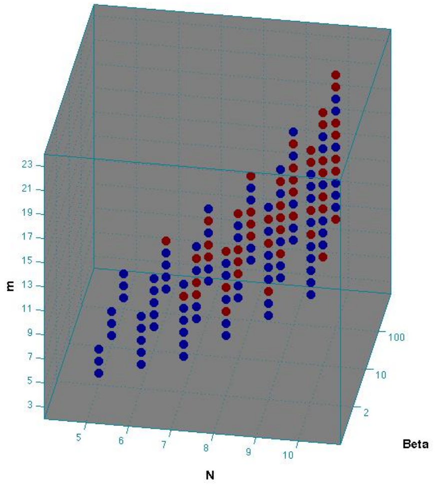

Fig. 6: *Comparison between exact optimal graphs and approximated optimal graphs. A blue circle means the two graphs are the same. A red circle means the two graphs are different.* $\alpha = 1$.

From figure 6, we know that in most cases, the approximated graph is the same as the exact graph, even when β becomes very different from α. This means that the bound presented in theorem 4 is often reached, and thus that it can be seen as a good approximation. In figure 6, when the two graphs are different, we can also compare the actual average clustering coefficient, in order to study how far the approximated graph is from the optimal solution. On average, we found that the approximated average clustering coefficient is 0.992 times the optimum.

V. Conclusion

In this work we have presented and studied a new problem concerning the optimization of the weighted clustering coefficient in the Air Transportation Network. This problem consists in optimizing both the structure and the weights of the network under several practical constraints.

We solved exactly the problem and showed that the structure of the optimal graph can be very different depending on whether the maximal and minimal traffic limits are close to each other.

Even if the complexity limits the size of the network we can optimize, regional airlines networks fit perfectly in our model. Moreover, good approximations can be found for bigger transportation networks. This novel work on robustness enhancement can therefore have practical applications and can lead to improvements from current ways of network design. In particular, we showed that the structure of an *ATN*, generally defined by hubs, focus cities and regional airports can be found on the optimal networks created with that metric.

Many improvements of this work can be considered, in particular for the non-homogeneous weights problem, in which the large complexity needs other methods to outperform our results.

References

[1] R. Guimera and L. Amaral, “Modeling the world-wide airport network,” *European Physical Journal B*, vol. 38, pp. 381–385, 2004.

[2] N. Alexandrov, “Transportation network topologies,” NASA Langley Research Center, Tech. Rep., 2004.

[3] P. A. Bonnefoy, “Scalability of the air transportation system and development of multi-airport systems: A worldwide perspective,” Ph.D. dissertation, Massachusetts Institute of Technology, 2008.

[4] S. R. Conway, “Scale-free networks and commercial air carrier transportation in the united states,” NASA Langley Research Center, Tech. Rep., 2004.

[5] H. Nagarajan, P. Wei, S. Rathinam, and D. Sun, “Air transportation network robustness optimization under limited legs itinerary constraint,” in *the 5th International Conference on Research in Air Transportation (ICRAT 2012)*, Berkeley, CA, USA, May 2012.

[6] G. Spiers, P. Wei, and D. Sun, “Algebraic connectivity optimization of the air transportation network,” in *American Control Conference*, Montreal, Canada, June 2012.

[7] R. Guimera, S. Mossa, A. Turtschi, and L. A. N. Amaral, “The worldwide air transportation network: Anomalous centrality, community structure, and cities’ global roles,” *Proceedings of the National Academy of Sciences, USA*, vol. 102, p. 7794, 2005.

[8] D. J. Watts and S. H. Strogatz, “Collective dynamics of ’small-world’ networks,” *Nature*, vol. 393, pp. 440–442, June 1998.

[9] A. Barrat, M. Barthelemy, R. Pastor-Satorras, and A. Vespignani, “The architecture of complex weighted networks,” *Proceedings of the National Academy of Sciences, USA*, vol. 101, pp. 3747–3752, 2004.

[10] D. Barmpoutis and R. M. Murray, “Networks with the smallest average distance and the largest average clustering,” 2010, arxiv preprint: pp. 1-28.

[11] L. A. N. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley, “Classes of small-world networks,” *Proceedings of the National Academy of Sciences, USA*, vol. 97, no. 21, pp. 11 149–11 152, October 2000.

[12] J. Huang and Y. Dang, “Research on the complexity of weighted air transportation network of express entreprise,” in *The 1st International Conference on Information Science and Engineering (ICISE2009)*, December 2009.

[13] K. C. Toh, R. H. Tutuncu, and M. J. Todd, “On the implementation and usage of sdpt3 - a matlab software package for semidefinite-quadratic-linear programming, version 4.0,” *Handbook on Semidefinite, Conic and Polynomial Optimization*, pp. 715–754, 2000.