Weighted Clustering Coefficient Maximization For Air Transportation Networks

Julien Ponton¹, Peng Wei² and Dengfeng Sun²

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. $w_{ij}$ 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.

According to [9], and using formulation in (1), we can define $d_i$, 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 (ATNs) 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. 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 maximum 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.
As suggested in [10], and as we are considering only graphs of given order \(N\), we can use the reduced average weighted clustering coefficient, \(C^{\alpha}_{K}(G)\) defined by:

\[
C^{\alpha}_{K}(G) = \frac{\sum_{i} c_i^\alpha}{N \times C^{\alpha}(G)}
\]

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 \(\beta\).
- Moreover, to establish a new flight route, we need a minimum amount of traffic demand \(\alpha\).

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

\[
\max_{G} C^{\alpha}_{K}(G(V, E)) \quad \text{s.t.} \begin{cases} |V| = N \\ |E| = m \\ \forall (i,j), w_{ij} \in \{0, [\alpha, \beta]\} \end{cases} \quad (P)
\]

We denote \(C^{\alpha}_{\max}(N, m)\) 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 \(C^{\alpha}_{K}(G(V, E))\) 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 \(\beta\) is sufficiently close to \(\alpha\).

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:

\[
\forall i \in \{1, 2, \ldots, N\}, \frac{\alpha}{\beta} C_i(G) \leq C^\alpha_i(G) \leq \frac{\beta}{\alpha} C_i(G).
\]

Proof:

\[
c_i^\alpha(G) = \frac{1}{(d_i - 1)\sum_{h} w_{ih}} \sum_{j} \frac{w_{ij} + w_{jh}}{2} a_{ij} a_{ih} a_{jh} \\
\leq \frac{1}{(d_i - 1)\sum_{h} a_{ih}} \sum_{j} \beta a_{ij} a_{ih} a_{jh} \\
\leq \frac{\beta}{\alpha} \frac{1}{(d_i - 1)\sum_{h} a_{ih}} \sum_{j} a_{ij} a_{ih} a_{jh} \\
\leq \frac{\beta}{\alpha} C_i(G)
\]

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:

\[
\forall (N, m), \ C^{\alpha}_{\max}(N, m) \geq C_{\max}(N, m).
\]

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 \(\alpha\) and \(\beta\), we have \(\max w_{ij} C^\alpha_{K}(G) \geq C_{K}(G') = C_{K}(G)\).

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 \(\beta\) is close enough to \(\alpha\), 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:

\[
C^{\alpha}_{\max}(N, m) = \max w C^\alpha_{K}(G') \leq \frac{\beta}{\alpha} C_{K}(G') < C(G) = C_{\max}(N, m)
\]

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

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: \(\alpha\) or \(\beta\). More formally, the binary problem can be defined as following:

Problem 1:

\[
\max_{G} C^{\alpha}_{K}(G(V, E)) \\
\text{s.t.} \begin{cases} |V| = N \\ |E| = m \\ \forall (i,j), w_{ij} \in \{0, \alpha, \beta\} \end{cases}
\]

In this case, we can see that the function \(f_{N,m} : (\alpha, \beta) \mapsto C^{\alpha}_{\max}(N, m)\) depends only on \(\frac{\beta}{\alpha}\). So we can impose that \(\alpha = 1\) for instance, and only modify the value of \(\beta\). 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: \(\alpha\) or \(\beta\). 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:

\[
b_i = \frac{(A^2 W^*)_{ij}}{(d_i - 1)(\sum_{j=1}^{m} w_{ij} + (d_i - k)\alpha)}.
\]

where

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

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 \(m_\beta\). 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 \(\beta\) in the optimal graph is
lower than \(m_\beta\).

Proof:
Suppose that the optimal graph \(G_{opt}\) has more than \(m_\beta\)
edges weighted with \(\beta\). 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 \(a\) in \(G\) and \(\beta\) in \(G_{opt}\). For
instance, \(e = (i,k)\). Now if we assume that \(w_{ij}\) is a continuous
variable, we can compute the derivative of \(c_i^\beta(G)\) with respect
\(w_{ij}\):

\[
\frac{dc_i^\beta}{dw_{ij}} = \frac{dc_i^\beta(G)}{dw_{ik}} = \frac{\Sigma a_ia_ia_ia_k \Sigma w_{ij} - \Sigma \Sigma a_ia_ia_ia_k}{(d_i - 1)(\sum w_{ij})^2} = \frac{\Sigma \Sigma w_{ij}(a_ia_ia_ia_k - a_ia_ia_ia_k)}{(d_i - 1)(\sum w_{ij})^2} = \frac{\Sigma \Sigma w_{ij}y_{ij}}{(d_i - 1)(\sum w_{ij})^2},
\]

where \(y_{ij} = \Sigma a_ia_ia_ia_k - a_ia_ia_ia_k\). Hence the sign of this
derivative depends only on \(\Sigma w_{ij}y_{ij}\), which, if \(a\) is close to \(\beta\),
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 \(\beta\) to edge \(e_1\). Doing that for all the edges with \(a\) in \(G\)
and \(\beta\) 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 \(a\), and then set edges
to \(\beta\) one-by-one. When we find that by switching a new edge
from \(a\) to \(\beta\), the average weighted clustering coefficient is
decreasing, we stop. The algorithm is shown in Algorithm 1.

**Algorithm 1 Binary Case**

\[
G \leftarrow \text{Compute\_optimal\_non\_weighted\_graph}\ (N,m) \\
C \leftarrow \text{Compute\_clustering\_coefficient}\ (G) \\
\text{continue} \leftarrow \text{True} \\
m_\beta \leftarrow 1 \\
\text{while continue do} \text{True} \\
G \leftarrow \text{Compute\_optimal\_weighted\_graph}\ (G,m_\beta) \\
C_{new} \leftarrow \text{Compute\_the\_weighted\_clustering\_coefficient}\ (G) \\
\text{if } C_{new} \leq C \text{ then} \text{continue} \leftarrow \text{False} \\
\text{else} \\
m_\beta = m_\beta + 1 \\
C \leftarrow C_{new} \\
\text{end if} \\
\text{end while} \\
\text{return } [G, C]
\]

Note that we apply the branch & bound method in function
\text{Compute\_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
differentially close meaning.

![Fig. 2: Examples of optimal graphs for binary case (problem 1).](image)

a = 1, \(\beta = 2\). Top left corner: \(C_{R(4,4)} = 3.4, \ C_{R(4,4)} = 3.33\)  
- Top right corner: \(C_{R(6,9)} = 5.4375, \ C_{R(6,9)} = 5.4\)  
- Bottom left corner: \(C_{R(7,19)} = 6.4444, \ C_{R(7,19)} = 6.4667\)  
- Bottom right corner: \(C_{R(10,32)} = 9.4107, \ C_{R(10,32)} = 9.2817\)

C. General case

We now focus on the "relaxed" problem of the binary case,
that is \(w_{ij} \in \{0, [a, \beta]\}\). 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:

\[
C^w_R(G(N,m)) = \sum_{i=1}^{N} \frac{1}{d_i - 1} \sum_{j \neq i} w_{ij} a_{ij} \alpha_{ij} b_{ij},
\]

where \( b_{ij} = \sum_{l=1}^{N} a_{ij} a_{jl} \). 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:

\[
w^*_{ij} = \frac{w_{ij}}{\sum_{k=1}^{N} w_{ik}} \quad \text{and} \quad \overline{w} = \left( w_{11} w_{12} \ldots w_{1N} w_{21} \ldots w_{NN} \right)
\]

The constraint \( w^*_{ij} = \sum_{k=1}^{N} w_{ik} \) can be rewritten as:

\[
\overline{w} N(i-1) + \sum_{k=1}^{N} \overline{w}_{ik} = \overline{w}_{ij}
\]

\[
\Leftrightarrow \overline{w} A_{ij} = \overline{c}_{ij},\overline{w} = 0,
\]

where \( A_{ij} \) is a symmetric matrix and \( \overline{c}_{ij} \) a vector defined by:

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

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

\[
\overline{e} = \left( 0 \ldots 0 \ b_{12} b_{13} \ldots b_{1N} b_{21} \ldots b_{NN} \right),
\]

\[
\overline{\pi} = \alpha \left( a_{11} a_{12} \ldots a_{1N} a_{21} \ldots a_{NN} 0 \ldots 0 \right),
\]

\[
\overline{\beta} = \beta \left( a_{11} a_{12} \ldots a_{1N} a_{21} \ldots a_{NN} a_{11} a_{21} \ldots a_{NN} \right),
\]

\[
D_{\overline{\pi}} = \begin{pmatrix} 
\overline{w}_{12} - \overline{w}_{21} \\
\overline{w}_{13} - \overline{w}_{31} \\
\vdots \\
\overline{w}_{1N} - \overline{w}_{N1} \\
\overline{w}_{23} - \overline{w}_{32} \\
\vdots \\
\overline{w}_{N-1,N} - \overline{w}_{NN-1}
\end{pmatrix}
\]

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

**Problem 2:**

\[
\begin{align*}
\max \overline{e} \overline{\pi} \\
\text{s.t.} \quad D_{\overline{\pi}} = 0 \\
\overline{\pi} \leq \overline{\pi} \leq \overline{\beta}
\end{align*}
\]

**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:

\[
\frac{t^j}{A} \lambda^j A \lambda + h^j x + c \leq 0
\]

can be written as:

\[
\left( \frac{1+h^j x + c}{2} \right) \leq \left( 1 - \frac{h^j x - c}{2} \right)
\]

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. α = 1, β = 2.](image)

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 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.
IV. NON-HOMOGENEOUS WEIGHTS

In this section, we study the non-homogenous weights problem. When \( \beta \) is not close to \( \alpha \), 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 \( \Delta w^{\text{Rob}}(N - 1, m - d, D) \) 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, \( \Delta w^{\text{Rob}}(N - 1, m - d, D) = \sum_{i \in D} c_{ij}^{v}(G) - c_{ij}^{v}(G') \). Then, we have:

\[
\text{Theorem 3: Let } V \text{ be a set of } N \text{ nodes, and } v \text{ a node in } V, \text{ indexed by } j.
\]

\[
C^w_{\text{max}}(N,m) = \max_{D \subseteq V \setminus \{v\}} \left\{ C^w_{\text{max}}(N - 1, m - d) + \max_{w_{ij} \in [\alpha, \beta], i \in D} \{ c_{ij}^{v} + \Delta w^{\text{Rob}}(N - 1, m - d, D) \} \right\}
\]

\[
\text{Proof: Let } G(V,E) \text{ be a graph of order } N \text{ and size } m. \text{ Let } v \text{ be a node in } V, \text{ indexed by } j. \text{ We have:}
\]

\[
C^w_{\text{max}}(N,m) = \sum_{i=1}^{N} c_{ij}^{v} = \left( \sum_{i \neq j} c_{ij}^{v} \right) + c_{jj}^{v}
\]

and it follows that:

\[
C^w_{\text{max}}(N,m) = \max_{G(N,m)} \left\{ \sum_{i \neq j} c_{ij}^{v} + c_{jj}^{v} \right\}
\]

Thus, \( C^w_{\text{max}}(N,m) \) 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:

\[
C^w_{\text{max}}(N,m) = \max_{D \subseteq V \setminus \{v\}} \{ C^w_{\text{max}}(N - 1, m - d) + \Delta w^{\text{Rob}}(N - 1, m - d, D) \}
\]

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.
C. Variation with $\beta$

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 $\beta$. As in section III.B, we will use the function $f_{N,m}(\alpha, \beta) \Rightarrow C_{\max}(N, m)$.

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

Proof: If $\beta_1 \leq \beta_2$, then $\beta_1 \in [\alpha, \beta_2]$. Thus, when $\beta_2$ is the maximum bound, the optimal graph with $\beta_1$ 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 $\alpha$ and $\beta$. Figure 6 presents the results.

![Figure 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. $\alpha = 1$.](image)

From figure 6, we know that in most cases, the approximated graph is the same as the exact graph, even when $\beta$ becomes very different from $\alpha$. 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