Digital Business Ecosystem

Contract number° 507953

# Workpackage 11
# Dynamics of Networks

# Deliverable 11.4
# Report on hierarchical networks

Information Society Technologies

Project funded by the European Community under the “Information Society Technology” Programme

**Contract number:** 507953
**Project acronym:** DBE
**Title:** Digital Business Ecosystem

**Deliverable Nº:** D11.4
**Due date:** 05/2006
**Delivery date:** 07/2006

**Short description: The final report on network dynamics and restructuring. Note that this task has been brought forward by five months.**

**Author:** UBHAM
**Partners contributed:** STU, LSE, HWU
**Made available to:** DBE Consortium and European Commission

| Versioning | | |
|---|---|---|
| **Version** | **Date** | **Author, Organisation** |
| 1.0 | 28/06/2006 | Jonathan E. Rowe, Boris Mitavskiy (UBHAM) |
| 2.0 | 24/07/2006 | Jonathan E. Rowe, Boris Mitavskiy (UBHAM) |

**Quality check**
**1st internal reviewer: Maria Chli (HWU)**
**2nd internal reviewer: Miguel Vidal (SUN)**

# Contents

# Executive Summary

This report is the final report from UBHAM on WP11 (sub-task S2) brought forward from month 36. It is a summary of work done at UBHAM during the period January–April 2006.

The report is in four main sections. The first introduces the work and describes its relationship to other aspects of the DBE project. The second part then describes the network re-routing problem within the context of the DBE, and outlines our adaptive algorithm for addressing this problem. The third part presents the results of our further empirical studies on the network re-routing algorithm. This algorithm allows nodes in the DBE network that frequently communicate with each other to be brought closer together within the network, and potentially identifying clusters of users with common interests. It is shown that our algorithm can offer some improvements in communication costs relatively cheaply. Some improvements are discussed that should be investigated as further work. In the fourth part of the report we present some theoretical analysis, firstly of the algorithms behaviour with regard to improvements in the expected number of hops (path length) within the network, and secondly with regard to the algorithms ergodicity. This latter condition ensures that it is always possible for the network to adapt towards any target configuration, with non-zero probability.

The main "customers" of this work are SUN and HWU, who are concerned with the network aspects of the DBE infrastructure.

# Chapter 1

# Introduction and relationship to the project

This deliverable is the final report for sub-task S2 (Dynamics and clustering in hierarchical networks), which is scheduled to run from month 19 to month 36, and builds on the work of S1, already reported [7] and the preliminary report from S2 [6]. The objectives of this sub-task are:

1. To extend work on information propagation in simple networks to hierarchical ones.

2. To develop theory of structure and dynamics in such networks.

The sub-task interacts with S10 (Symbiosis and competition - HWU) which studies the effects on markets of information sharing in networks, and WP19 (SUN and HWU) looking at the specifics of the DBE peer-to-peer network behaviour (see, for example, [5, 4]).

In the previous report [6] we introduced the network re-routing problem and presented some preliminary experimental results of our proposed algorithm. We had originally planned to continue this effort in the following ways:

1. Further empirical studies of the algorithm on scale-free networks.

2. Theoretical analysis of the algorithm's properties.

3. Visit to SUN for large-scale empirical studies.

4. Extension to hierarchical clustering of nodes.

Unfortunately, due to personnel problems, UBHAM has had to withdraw from the project early. We have made good progress on the first two parts of the plan. However, we have not been able to arrange the visit to SUN, nor to implement the extension to hierarchical clustering. The man-months and funding for the remainder of the task have instead been reallocated to other partners.

In this report, then, we focus on our further work on the network re-routing problem, which is introduced in chapter 2, along with the basic algorithm. The results of our empirical studies are presented in chapter 3, and our theoretical results to date are presented in chapter 4.

# Chapter 2

# The network re-routing problem

## 2.1 The problem

In [6] we introduced the following problem regarding the dynamical restructuring of the DBE network. We imagine that users join the network in a rather ad hoc manner. It seems most likely that preferential attachment of users to already popular nodes will create a scale-free network topology. As the DBE is used, the user will transfer information from other nodes of the network. We would like to arrange things so that nodes that commonly transfer information between themselves are close together in the network. This is partly so that the communication time is more efficient, but also because this would lead, in the long term, to the identification of a potential cluster of users with similar interests and requirements. The problem is that the initial attachment of the user to the DBE network will not necessarily place it in the optimal position from this point of view. We would therefore like to be able to adapt the topology of the network so as to bring nodes which frequently communicate with each other closer together.

Obviously, the simplest solution would be to have a complete network, in which every node was a neighbour of every other node. However, this would then make the network too expensive, with many hubs and many connections (each node is effectively a hub). To try to maintain the sparsity of the network, then, we restrict the kinds of restructuring allowed to the replacement of one connection by another. In this way, the total number of connections in the network remains constant.

Formally, then, the problem is as follows. We are given a graph $G$ with nodes $V$ and edges $E$. From time to time, pairs of nodes communicate with each other. We assume that there is some (unknown) probability distribution $\mu$ over the set of pairs of nodes describing this communication. We follow some standard algorithm (e.g. Dijkstra's algorithm) to establish a path between the two nodes, and record the path length (number of hops required). We would like to minimise the average path length:

$$E(D) = \sum_{a,b \in V} \mu(a,b) d(a,b)$$

where $d(a,b)$ is the number of hops from $a$ to $b$. We seek to do this by incremen-

tally changing the network topology, replacing an existing edge with a new edge. The problem is to find an appropriate replacement strategy.

## 2.2 The algorithm

The algorithm introduced in [6] works as follows:

1. Select $a, b \in V$ according to $\mu$.
2. Find a shortest path $\gamma$ from $a$ to $b$.
3. For each internal node $v$ of the path, create a shortcut with probability $p$.
4. Go to 1.

To create a shortcut, consider the section of the path surrounding the chosen node $v$. That is, there are nodes $u$ and $w$ so that $(u, v, w)$ is in the path $\gamma$. We add a new edge $(u, w)$ to the graph, and delete either edge $(u, v)$ or $(v, w)$. The deleted edge is chosen randomly.

The idea of this algorithm is that the shortcut will reduce the path-length by one. Of course it may create problems for other paths, but the hope is that since it will be applied to the most frequently chosen paths, the overall effect will be beneficial.

It should be noted that the problem, as stated, has, as a special case, the Minimum Communication Cost Spanning Tree problem, which is already known to be NP-complete [1]. It is not feasible, then, to propose an efficient algorithm which will solve the problem exactly. The best we can hope for is that our heuristic algorithm will generate solutions of acceptable quality. Having said that, it is known that in the case where:

1. All pairs of nodes communicate with equal frequency, and
2. the number of allowed edges is one less than the number of nodes

that there is a polynomial-time algorithm, due to Gomory and Hu [2, 3]. However, in our situation, we do not have advance knowledge of the frequency with which pairs of nodes will communicate and it is very unlikely that this will be equal for all pairs of nodes, so their algorithm is not directly applicable. We do, though, have the advantage of having potentially more edges in the DBE network that the minimal set allowed in the Gomory-Hu algorithm.

# Chapter 3

# Empirical results with scale-free networks

## 3.1 Experiments

Following the set of experiments reported in [6], we continued our investigation of our algorithm on scale-free networks, with different probability distributions over the pairs of nodes selected for communication.

We assume that the DBE network will, at least initially, tend to grow following some sort of preferential attachment, as new users link to nodes that are already well-known. This will result in a scale-free network topology. We have therefore conducted experiments with such networks with between 50 and 200 nodes. We seek to simulate the situation that would occur when two nodes frequently share information, and yet are currently relatively far apart in the network. We hope our adaptive algorithm will reconfigure the network so as to bring them closer together. Consequently, for each network, we selected a subset of $10\%$ of the nodes with probability inversely proportional to the node's degree (i.e. the nodes which have smaller degree are more likely to get selected). Such a subset is likely to be fairly widely spread throughout the network and will tend not to include major hubs which are already well-connected.

The selected nodes where partitioned into two sets, $A$ and $B$ of equal sizes and the probability distribution $\mu$ was defined which samples every pair from the set $A \times B$ equally likely. That is, nodes from group A always want to communicate with nodes from group B. The adaptive algorithm should move these two subsets closer together in the network as it reconfigures the network topology.

The expected path length with respect to the probability distribution $\mu$ described above has been estimated by performing $50$ independent samplings of pairs of nodes with respect to $\mu$. Afterwards, 300 independent iterations of the re-routing algorithm were performed. A single iteration of the algorithm picks a pair of nodes $(x,\, y) \in V^2$ at random with respect to the distribution $\mu$ (notice that our distribution $\mu$ is concentrated on the pairs in $A \times B$ only so that with probability $1$ we choose a pair in $A \times B \subseteq V^2$). Now we use Dijkstra's algorithm to find the shortest path between $x$ and $y$ in the

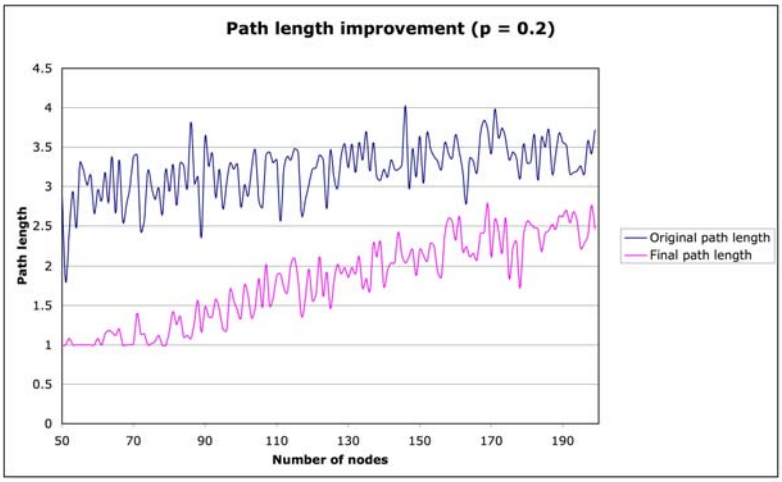
original network. Once a shortest path has been selected, we traverse the path from one end to another, replacing a consecutive pair of edges by a single edge joining non-common nodes and deleting one of the intermediate edges (see the previous section for more details on the analysis of this algorithm) with some probability $p$. Three independent experiments have been run with the values of $p = 0.2$, $p = 0.5$ and $p = 0.8$. Again, upon the completion of 300 such iterations, we estimate the average path length by performing $50$ independent samplings of pairs in $V^2$ with respect to $\mu$. The plots of the average shortest path length vs. the total number of nodes in the network before and after the re-routing algorithm have been produced. Also the plots of the average complexity of the Dijkstra's computation (the number of nodes encountered during a single iteration of the Dijkstra's algorithm) vs. the total number of nodes in the network have been constructed.

## 3.2 Results

From figures 3.1–3.3 we can see that the average path length after the algorithm has been run is significantly reduced. The same is true for the time complexity of running Dijkstra's algorithm for finding the shortest paths (measured by the number of nodes visited). Moreover, the reduction is stronger for larger values of $p$. A possible explanation for this is that a single step of the algorithm is unlikely to case to much harm. If a step leads to an increase in the mean path length, according to theorem 4.3.2, the only way this can happen is if the probability of joint occurrence of the consecutive edges which have been altered is smaller than the probability of their separate occurrence. But then, roughly speaking, the opposite will be true in the next step of the algorithm so that the algorithm is likely to correct itself in the sequel step. At the same time, the algorithm modifies the network more frequently after a fixed number of iterations when $p$ is larger. However, a deeper theoretical analysis is necessary to understand which parameters are more suitable for which networks. As the number of nodes gets larger, the improvement reduces regardless of the value of $p$. This is quite easy to understand: For larger networks, the percentage of the nodes selected increases and the number of possible pairs sampled increases quadratically. The number of iterations is usually insufficient to sample all the possible pairs, and, in fact, samples only a very limited number of these pairs. Completely different sets of pairs may be sampled during the mean path estimation after all the modifications are complete and before the iterations have been run. Unfortunately we do not possess sufficiently powerful equipment to run more more iterations for a large number of nodes, however, it seems the outcome should be roughly similar as it is for the small number of nodes.

## 3.3 Possible improvements

It would seem from the empirical data that it is a good strategy to make the edge replacement probability as high as possible. Taking this to the extreme, we could set $p = 1.0$ which effectively means that every time two nodes communicate, we place an edge directly between them and delete a random edge from the original path. However,



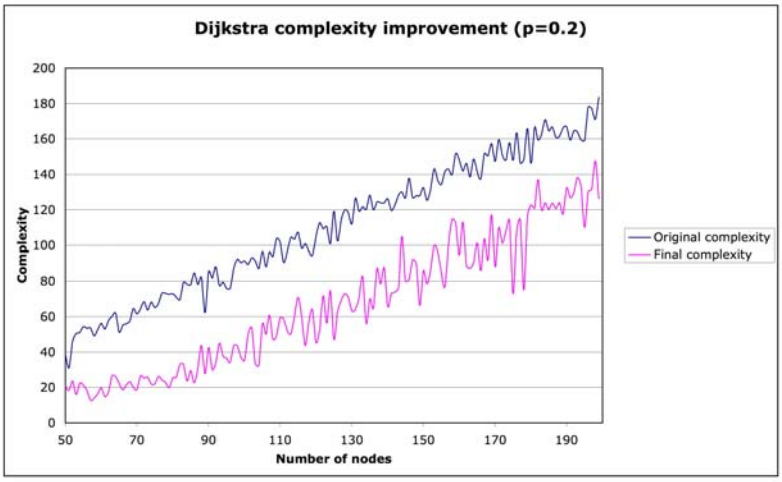
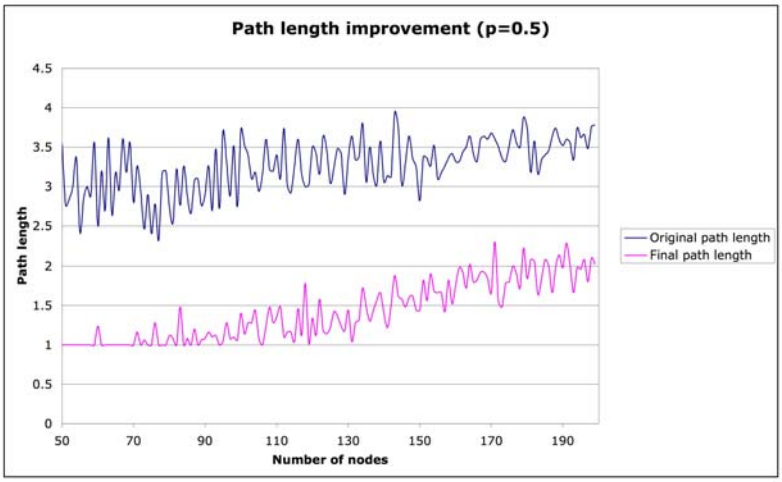




Figure 3.1: Improvement in expected path length, and time complexity of Dijkstra's algorithm, after running the adaptive re-routing algorithm for 300 iterations, with an edge replacement probability of 0.2



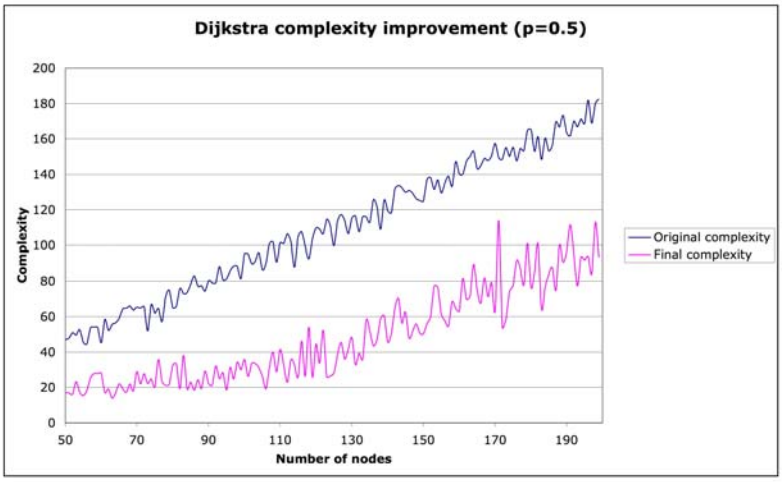
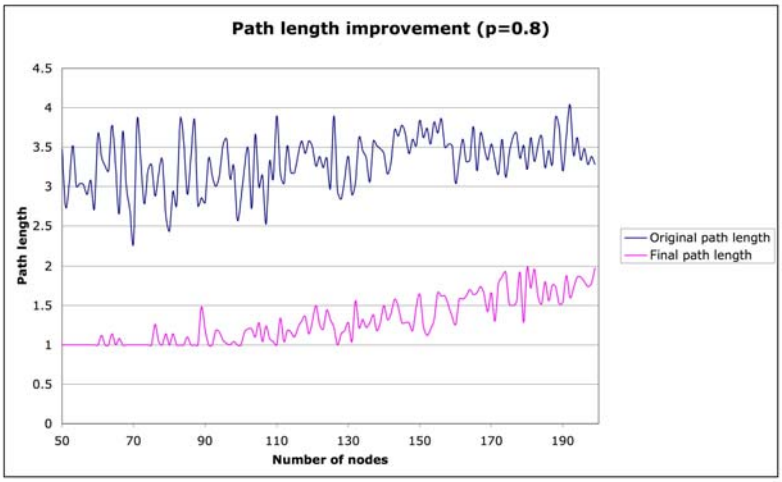




Figure 3.2: Improvement in expected path length, and time complexity of Dijkstra's algorithm, after running the adaptive re-routing algorithm for 300 iterations, with an edge replacement probability of 0.5



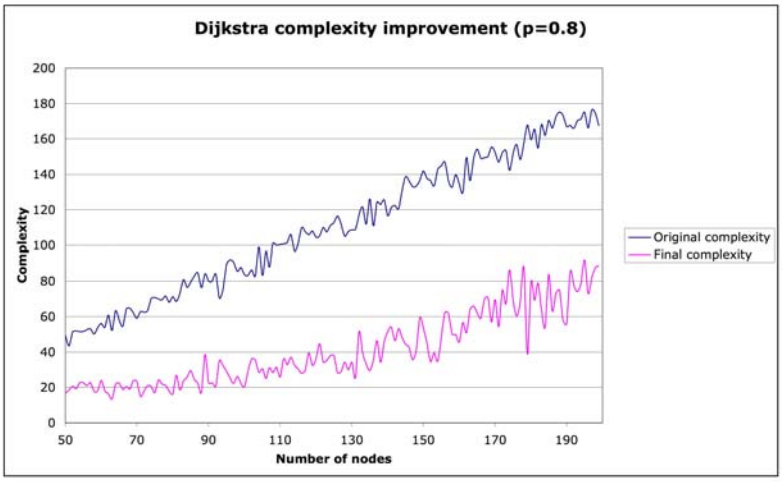




Figure 3.3: Improvement in expected path length, and time complexity of Dijkstra's algorithm, after running the adaptive re-routing algorithm for 300 iterations, with an edge replacement probability of 0.8

this trend in the data is probably caused by the rather special probability distribution which we used to model communications, in which all communications were restricted to being between two small subsets of the nodes. With a rather less strict set of communications, it is likely that the extreme choice $p = 1.0$ will be more likely to disrupt potentially useful pathways in the network.

However, this does lead to the question of the best choice of shortcut to make. A more sophisticated strategy would be to record, for each node in the network, the number of communications which are routed through it. One would then prefer to place shortcuts around nodes that are only used infrequently, since this is less likely to be disruptive. This protocol is one we would like to have validated on large-scale networks in collaboration with SUN.

A further improvement to the algorithm would have to take into account the fact that new users may join and leave the DBE network during its lifetime. There should not be any problem in accommodating new nodes in the network within the current algorithm (or its proposed extension). However, if a node leaves the network, then new routes would have to be found to accommodate this. Given the scale-free nature of the DBE network topology, most nodes would be able to leave (or go down temporarily) with only minor effects on the rest of the network. However, should a major hub leave the system, then our re-routing algorithm should be able to reconfigure the network efficiently. This would need to be tested empirically.

It is worth emphasising that, since the problem is an NP-complete one, we do not expect to be able to find optimal network configurations efficiently. Our algorithm will find local optima (with respect to the removal and addition of single edges), which may not necessarily be globally optimum. It is important to investigate exactly how good such local optima are. In fact it is possible to construct artificial networks (e.g. star-shaped networks) in which it is very hard for our algorithm to find good solutions. In such cases, the algorithm performs rather badly. However, it is very unlikely that such an artificial structure will be found in the DBE. One would therefore need to experiment with simulations of the DBE set-up to establish whether or not the algorithm performs well within the practical situation of the DBE. The experiments we reported are encouraging, but more needs to be done with large-scale simulations based on realistic data. We had planned to visit SUN to perform such experiments on their high-performance parallel machines, but unfortunately this visit had not taken place by the time we withdrew from the project. We urge SUN to continue with such experimentation if at all possible. Similarly, it is important that our proposed algorithm be compared to other approaches that may already be known. We have so far found the Gomory-Hu algorithm [2, 3], although this only applies to tree-structured networks. Again, this is work that should have been done before the completion of the project, which we have been unable to complete before our withdrawal.

# Chapter 4

# Theoretical analysis

## 4.1 Overview

In this section we present our preliminary theoretical analysis of our network re-routing algorithm. It is rather technical, so we first present a high-level summary.

First we have some mathematical observations in which we define the object of interest, namely the *expected shortest distance* in the network. That is the average number of hops that messages will make during transmissions between nodes in the network. We prove a simple result that relates this quantity to the frequency with which links (edges) in the network occur in transmissions. If an edge occurs with high frequency, then it can be considered to be rather important. The connection we prove is that the expected shortest distance in the network is equal to the sum of the edge frequencies.

Next we look at the local effects of a single move of our algorithm to try to establish conditions under which the replacement of one edge by another (a shortcut) will result in an improvement. Clearly, the shortcut will improve matters between the two nodes that have just communicated. The problem is the the edge that is being replaced might itself be important. Remembering that the importance of an edge is related to its frequency, we derive conditions on edge frequencies which ensure that the overall average shortest length is improved by the move.

Of course, this is only a local analysis, and may well lead the algorithm to a local optimum. In the next section, though, we at least prove that the algorithm has a chance to escape local optima since there is always a non-zero probability of moving between any two network configurations. technically, this means the algorithm is *ergodic*. In the worst case, of course, one may end up waiting a long time for such moves to be made. Further empirical studies are necessary to consider whether or not this is the case in the kinds of network structure generated by the DBE.

## 4.2 Basic mathematical observations

Suppose we are given a network (an undirected graph) $G = (V, E)$ and a probability distribution $\mu$ on $V^2$. For a given pair of nodes $x$ and $y$ denote by $d(x, y)$ the distance between $x$ and $y$ (i.e. the number of edges in the shortest path joining $x$ and $y$). We are interested in restructuring the network $G$ so as to preserve the number of of edges (not to make it bigger), but, at the same time, to minimize the expected shortest distance with respect to the distribution $\mu$, namely $E(D) = \sum_{x, y \in V} \mu(x, y) d(x, y)$. We make the following general observation first:

**Proposition 4.2.1** *Denote by $\Gamma$ a chosen collection of shortest paths: one for every pair of nodes. Then $E(D) = \sum_{e \in E} P_\Gamma(e)$ where $P_\Gamma(e)$ is the probability that the edge $e$ has been encountered in a shortest path from $\Gamma$ joining a pair selected with respect to the distribution $\mu$.*

*Proof:* Consider the characteristic function $\mathcal{X} : E \times V^2 \to \{0, 1\}$ defined as

$$\mathcal{X}(e, x, y) = \begin{cases} 1 & \text{if } e \text{ is an edge in the shortest path from } \Gamma \text{ joining } x \text{ and } y \\ 0 & \text{otherwise.} \end{cases}$$

First notice that for every $x, y \in V$ we can write $d(x, y) = \sum_{e \in E} \mathcal{X}(e, x, y)$ We then obtain

$$E(D) = \sum_{x, y \in V} \mu(x, y) d(x, y) =$$

$$= \sum_{x, y \in V} \mu(x, y) \sum_{e \in E} \mathcal{X}(e, x, y) = \sum_{e \in E} \sum_{x, y \in V} \mu(x, y) \mathcal{X}(e, x, y).$$

Notice that for a fixed edge $e \in E$, $P_\Gamma(e) = \sum_{x, y \in V} \mu(x, y) \mathcal{X}(e, x, y)$ is just the probability that the edge $e$ has been traversed by a path from $\Gamma$ joining some pair of nodes $(x, y)$ which is chosen randomly with respect to $\mu$. The desired conclusion that $E(D) = \sum_{e \in E} P_\Gamma(e)$ now follows. □

Notice that the choice of the collection of the shortest paths $\Gamma$ is not unique in general. As an immediate consequence of proposition 4.2.1 we deduce

**Corollary 4.2.2** *Given a graph $G = (V, E)$ and a probability distribution $\mu$ over $V^2$, let $\Gamma$ denote any collection of the shortest paths between the nodes of $G$ containing a unique path for every pair of nodes of $G$. Then it follows that $\sum_{e \in E} P_\Gamma(e)$ is independent of the choice of $\Gamma$.*

Similar results can be established by expressing the path length counting the nodes rather than the edges. The proof of proposition 4.2.1 goes through just fine when we use the characteristic function $\mathcal{Y}_\Gamma : V^3 \to \{0, 1\}$ defined as

$$\mathcal{Y}(v, x, y) = \begin{cases} 1 & \text{if } v \text{ is a node on the path from } \Gamma \text{ between } x \text{ and } y \\ 0 & \text{otherwise} \end{cases}$$

and then notice that $d(x,\, y) = (\sum_{v\in V} \mathcal{Y}(v,\, x,\, y)) - 1$ so that we obtain

$$E(D) = \sum_{x,\, y\in V} \mu(x,\, y)d(x,\, y) = \sum_{x,\, y\in V} \mu(x,\, y)((\sum_{v\in V} \mathcal{Y}(v,\, x,\, y)) - 1) =$$

$$= \sum_{x,\, y\in V} \mu(x,\, y) \sum_{v\in V} \mathcal{Y}(v,\, x,\, y) - \sum_{x,\, y\in V} \mu(x,\, y) =$$

$$= (\sum_{v\in V} \sum_{x,\, y\in V} \mu(x,\, y)\mathcal{Y}(v,\, x,\, y)) - 1.$$

Again, for every fixed node $v \in V$ we notice that $\sum_{x,\, y\in V} \mu(x,\, y)\mathcal{Y}(v,\, x,\, y)$ is the probability $P_\Gamma(v)$ that the node $v$ has been encountered in the unique path from $\Gamma$ joining the pair of nodes $(x,\, y)$ randomly chosen with respect to the distribution $\mu$. This produces a result analogous to proposition 4.2.1:

**Proposition 4.2.3** *Denote by $\Gamma$ a chosen collection of shortest paths: one for every pair of nodes. Then $E(D) = (\sum_{v\in V} P_\Gamma(v)) - 1$ where $P_\Gamma(v)$ is the probability that the node $v$ has been encountered in a shortest path from $\Gamma$ joining a pair selected with respect to the distribution $\mu$.*

## 4.3 Local algorithm analysis

We consider a simple online algorithm whose step is to remove an edge $e^-$ from the network and to insert another edge $e^+$ into the network. We shall now express the "improvement" or "worsening" that the algorithm creates after a single time step. We continue with the notation of the previous section: $G = (V,\, E)$ denotes our network and $G' = (V,\, (E \cup \{e^+\}) - \{e^-\})$ denotes the modified network. Our immediate goal is to express the difference between the expected average distances $D$ and $D'$ for the networks $G$ and $G'$ respectively. Recall that $\Gamma$ and $\Gamma'$ denote the collections of shortest paths between the pairs of nodes of $G$ and $G'$ respectively containing exactly one path for every pair of nodes $(x,\, y)$. Once the set of shortest paths $\Gamma$ has been chosen consider the subset $C(e^-,\, e^+) \subseteq \Gamma$ consisting of all the paths in $\Gamma$ which remain the shortest in $G'$ (i.e. these which do not pass through $e^-$ and also remain the shortest regardless of removing $e^-$ and adding $e^+$). Notice that $\Gamma'$ can be chosen so that $C(e^-,\, e^+) \subseteq \Gamma'$ (since these paths remain the shortest in $G'$) and for the rest of the pairs, we choose the new shortest path in $G'$. Denote this new set of shortest paths by $N'(e^-,\, e^+)$ and also let $N(e^-,\, e^+) = \Gamma - C(e^-,\, e^+)$. Notice also that

$$P_\Gamma(e) = P_{C(e^-,\, e^+)}(e) + P_{N(e^-,\, e^+)}(e)$$

where $P_{C(e^-,\, e^+)}(e)$ denotes the probability that the edge $e$ occurs in the shortest path from $\Gamma$ when sampling with respect to $\mu$ and $P_{N(e^-,\, e^+)}(e)$ denotes the probability that it occurs in the path from $N(e^-,\, e^+)$. Likewise,

$$P_{\Gamma'}(e) = P_{C(e^-,\, e^+)}(e) + P_{N'(e^-,\, e^+)}(e)$$

where $P_{N'(e^-,\,e^+)}(e)$ denotes the probability that the edge $e$ occurs in the path from $N'(e^-,\,e^+)$. Combining these observations with proposition 4.2.1 gives us

$$E(D) - E(D') = \sum_{e\in E} P_\Gamma(e) - \sum_{e\in E'} P_{\Gamma'}(e) =$$

$$= \sum_{e\in E-e^-} P_\Gamma(e) - \sum_{e\in E-e^-} P_{\Gamma'}(e) + P_\Gamma(e^-) - P_{\Gamma'}(e^+) =$$

$$= \sum_{e\in E-e^-} (P_{C(e^-,\,e^+)}(e)+P_{N(e^-,\,e^+)}(e)) - \sum_{e\in E-e^-} (P_{C(e^-,\,e^+)}(e)+P_{N'(e^-,\,e^+)}(e)) +$$

$$+P_\Gamma(e^-)-P_{\Gamma'}(e^+) = \sum_{e\in E-e^-} P_{N(e^-,\,e^+)}(e) - \sum_{e\in E-e^-} P_{N'(e^-,\,e^+)}(e)+P_\Gamma(e^-)-P_{\Gamma'}(e^+).$$

We summarize this in the following

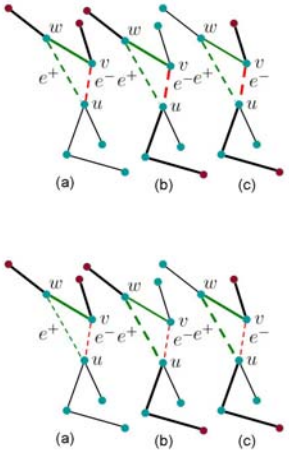
**Proposition 4.3.1** *Let $G = (V,\, E)$ denote a network and suppose we remove the edge $e^-$ from $E$ and add a new edge $e^+$ to $E$. This gives us a new network $G' = (V,\, (E \cup \{e^+\}) - \{e^-\})$. Let $\Gamma$ denote any particular choice of the shortest paths in the network $G$. Also, let $\Gamma'$ denote a choice of shortest paths having the property that $C(e^-,\, e^+) \subseteq \Gamma \cap \Gamma'$ (as mentioned before such a choice does exist) where $C(e^-,\, e^+) \subseteq \Gamma$ consisting of all the paths in $\Gamma$ which remain the shortest in $G'$ (i.e. these which do not pass through $e^-$ and also remain the shortest regardless of removing $e^-$ and adding $e^+$). Let $N(e^-,\, e^+) = \Gamma - C(e^-,\, e^+)$ and $N'(e^-,\, e^+) = \Gamma' - C(e^-,\, e^+)$. We then have*

$$E(D)-E(D') = \sum_{e\in E-e^-} P_{N(e^-,\,e^+)}(e) - \sum_{e\in E-e^-} P_{N'(e^-,\,e^+)}(e)+P_\Gamma(e^-)-P_{\Gamma'}(e^+).$$

Proposition 4.3.1 tells us, in particular, that the "add/remove an edge" type of an algorithm leads to an improvement if and only if $\sum_{e\in E-e^-} P_{N(e^-,\,e^+)}(e) - \sum_{e\in E-e^-} P_{N'(e^-,\,e^+)}(e) + P_\Gamma(e^-) - P_{\Gamma'}(e^+) > 0$. In fact, this quantity measures a single step improvement.

Next, we shall apply the above observations to deduce some basic theoretical properties of the algorithm described in the previous report. Lets recall how the algorithm works: We fix a small positive number $\epsilon \ll 1$. At every stage We sample a pair (x, y) of nodes randomly with respect to the probability distribution $\mu$. Next we find a shortest path joining $x$ and $y$ in the network $G$. We traverse the path starting with the node $x$ and whenever we encounter a consecutive pair of edges $(u,\, v)$ and $(v,\, w)$ along the path, with with probability $\epsilon$ we replace one of the edges (either $(u,\, v)$ or $(v,\, w)$) with the edge $(u,\, w)$. The decision which one of the edges is discarded is made with even probability $\frac{1}{2}$. Now suppose W.L.O.G. that $e^- = (u,\, v)$ and $e^+ = (u,\, w)$ (recall that $e^-$ denotes an edge that has just been removed while $e^+$ denotes the one that has been added). Once we select the collection of the shortest paths $\Gamma$, notice that for every path $\gamma \in N(e^-,\, e^+)$ joining a pair of nodes, say $x$ and $y$, the path $\gamma'$ in the new network obtained upon removing the edge $e^-$ and adding the edge $e^+$, constructed by replacing the edge $e^-$ with the consecutive pair of edges $\{v,\, w\}$ and $e^+$ (or $e^+$ and $\{w,\, v\}$ depending on the order) is longer by exactly one edge. Thus, even if $\gamma'$ is a shortest path between $x$ and $y$ in the new network, the shortest distance between $x$ and $y$ has

been increased by exactly one edge. In this case, we let the new set of shortest paths $\Gamma'$ contain the path $\gamma'$. Likewise, the paths which involve both edges, $e^-$ and $\{v, w\}$ are shortened by exactly one edge in the network $\Gamma'$. Again, we let $\Gamma'$ contain these paths as well (these where the edges $e^-$ and $\{v, w\}$ are replaced by the single edge $e^+$). The picture below illustrates all the possible situations:




The upper pictures show the shortest paths between the corresponding pairs of purple nodes. The case when the path remains unchanged is shown on the picture (a), the path which shall be shortened is displayed on the picture (b) and the path which shall become longer appears on the picture (c). The shortest paths in the modified graph appear on the bottom picture for all the corresponding cases.

To shorten the notation, let $e^* = \{v, w\}$. Now consider the possible re-routings that occur as a result of replacing $e^-$ with $e^+$. We will consider three cases:

**Case 1.** Paths that do not go through $e^-$ will either remain unchanged in $\Gamma'$ or may possibly be shortened by one if $e^*$ happens to create any fortuitous (i.e. unplanned) shortcuts.

**Case 2.** Paths that pass through $e^-$ but not through $e^*$. If there is an alternative shortest path, then this can be used and the path length remains unchanged. Otherwise, $e^-$ in the path has to be replaced by $e+, e^*$ which increases the path length by one.

**Case 3** . Paths that use both $e^-$ and $e^*$ are shortened by one as they now use the shortcut $e^+$.

We can get a lower bound on the size of the decrease in path length by ignoring fortuitous shortcuts in case 1, and alternative routes in case 2. Accordingly,

$$\sum_{e \in E - \{e^-,\, e^*\}} P_{N(e^-,\, e^+)}(e) \leq \sum_{e \in E - \{e^-,\, e^*\}} P_{N'(e^-,\, e^+)}(e)$$

According to proposition 4.3.1, we can now write

$$E(D)-E(D') = \sum_{e\in E-e^-} P_{N(e^-,\,e^+)}(e) - \sum_{e\in E-e^-} P_{N'(e^-,\,e^+)}(e) + P_\Gamma(e^-) - P_{\Gamma'}(e^+) =$$

$$= \sum_{e\in E-\{e^-,\,e^*\}} P_{N(e^-,\,e^+)}(e) - \sum_{e\in E-\{e^-,\,e^*\}} P_{N'(e^-,\,e^+)}(e) +$$

$$+ P_{N(e^-,\,e^+)}(e^*) - P_{N'(e^-,\,e^+)}(e^*) + P_\Gamma(e^-) - P_{\Gamma'}(e^+) \geq$$

$$\geq P_{N(e^-,\,e^+)}(e^*) - P_{N'(e^-,\,e^+)}(e^*) + P_\Gamma(e^-) - P_{\Gamma'}(e^+).$$

In summary, we have

$$E(D) - E(D') \geq P_{N(e^-,\,e^+)}(e^*) - P_{N'(e^-,\,e^+)}(e^*) + P_\Gamma(e^-) - P_{\Gamma'}(e^+).$$

We now proceed to analyze the differences $P_{N(e^-,\,e^+)}(e^*) - P_{N'(e^-,\,e^+)}(e^*)$ and $P_\Gamma(e^-) - P_{\Gamma'}(e^+)$ more explicitly. First note that we can write $N(e^-,\,e^+) = L(e^-,\,e^+) \cup S(e^-,\,e^+)$ where $L(e^-,\,e^+)$ is the set of these paths which have been made longer, and $S(e^-,\,e^+)$ is the set of these paths which have been shortened upon removal of $e^-$ and insertion of $e^+$ respectively. Notice that the edge $e^*$ does not occur in a path which has been lengthened by removal of $e^-$ (according to the previous discussion, all such paths must pass through $e^-$ and not through $e^*$) so that $P_{L(e^-,\,e^+)}(e^*) = 0$. Likewise, $P_{S(e^-,\,e^+)}(e^*)$ is the probability that $e^*$ is the edge of a path that has been shortened, i.e. a path which goes through both, $e^-$ and $e^*$. This is simply the probability that $e^-$ and $e^*$ occur jointly. We shall denote this probability by $P(e^- \wedge e^*)$. We now deduce that

$$P_{N(e^-,\,e^+)}(e^*) = P_{L(e^-,\,e^+)}(e^*) + P_{S(e^-,\,e^+)}(e^*) =$$

$$= 0 + P(e^- \wedge e^*) = P(e^- \wedge e^*).$$

In general we shall adopt the following notation: $P(e_1 \wedge e_2)$ is the probability that the edges $e_1$ and $e_2$ are encountered jointly, while $P(e_1 \wedge \overline{e_2})$ shall denote the probability that $e_1$ occurs and $e_2$ does not. Similarly, we deduce that

$$P_{N'(e^-,\,e^+)}(e^*) = P(e^* \wedge e^+) = P(e^- \wedge \overline{e^*}).$$

(Here the reader would do well to refer back to the picture above.) Finally, observe that whenever a path in $\gamma \in \Gamma$ involves $e^-$, the corresponding path $\gamma' \in \Gamma'$ involves $e^+$ (regardless of wether it is shortened or lengthened). It follows now that $P_\Gamma(e^-) = P_{\Gamma'}(e^+)$ so that $P_\Gamma(e^-) - P_{\Gamma'}(e^+) = 0$ and we finally conclude that

$$E(D) - E(D') \geq P_{N(e^-,\,e^+)}(e^*) - P_{N'(e^-,\,e^+)}(e^*) + P_\Gamma(e^-) - P_{\Gamma'}(e^+) =$$

$$= P(e^- \wedge e^*) - P(e^- \wedge \overline{e^*}).$$

We summarize this final observation below:

**Theorem 4.3.2** *Let $G = (V, E)$ denote a network. Let $u, v, w \in V$ and let $e^- = \{u, v\}$, $e^{interm} = \{v, w\}$. Suppose also that $e^+ = \{u, w\} \notin E$. Assume we remove the edge $e^-$ from $E$ and add a new edge $e^+$ to $E$. This gives us a new network $G' = (V, (E \cup \{e^+\}) - \{e^-\})$. Then, we have*

$$E(D) - E(D') \geq P(e^- \wedge e^*) - P(e^- \wedge \overline{e^*})$$

*where $P(e_1 \wedge e_2)$ is the probability that the edges $e_1$ and $e_2$ are encountered jointly, while $P(e_1 \wedge \overline{e_2})$ denotes the probability that $e_1$ occurs and $e_2$ does not.*

## 4.4 Ergodicity of the Algorithm:

This section is devoted to showing that given any two connected graphs $G$ and $G'$ on $n$ nodes and $k$ edges, there is a way to obtain $G'$ from $G$ by performing the steps of our algorithm. Before stating the formal theorem we want to establish, it is convenient to introduce the following group action on the set $\mathcal{G}(S, k)$ of connected graphs on the specified set $S$ of $n$ nodes and having a specified number $k$ of edges:

**Definition 4.4.1** *For every triple of nodes $\{i, j, k\} \subseteq S$ we introduce the following permutations on the set $\mathcal{G}(S, k)$ of connected graphs on the specified set $S$ of $n$ nodes and having a specified number $k$ of edges:*

$$\pi^{(a,b)}_{(a,b),(b,c)\to(a,c)}(G) = \begin{cases} G' = (S, E - \{(a, b)\} \cup \{(a, c)\}) & \text{if } (a, b) \text{ and } (b, c) \in E \\ & \text{and } (a, c) \notin E \\ G & \text{otherwise} \end{cases}$$

*where $\{a, b, c\} = \{i, j, k\}$ and $G = (S, E) \in \mathcal{G}(S, k)$.*

*We shall also denote by $\mathcal{A}$ the subgroup of permutations on $\mathcal{G}(S, k)$ generated by the set $\Delta$ of all possible permutations of the form described above.*

Applying a step of our algorithm to a graph $G$ can then be described as selecting a permutation from $\Delta \cup \{I\}$ with some probability (which does depend on $G$) and applying it to $G$. Notice that such a probability distribution over $\Delta \cup \{I\}$ can always be chosen so that every one of the elements in $\Delta \cup \{I\}$ is chosen with positive probability as long as the distribution $\mu$ on $V^2$ assigns a positive probability to every pair of nodes. it follows that a graph $G'$ can be obtained from a given graph $G \in \mathcal{G}(S, k)$ upon completion of finitely many steps of our algorithm if and only if there exists an element $a \in \mathcal{A}$ (the group generated by $\Delta$) such that $a \cdot G = G'$ where the action $\cdot$ is the usual function evaluation (which is the action induced by the generators). In other wards, $G'$ can be obtained from $G$ upon completion of finitely many steps of our algorithm if and only if $G'$ and $G$ are in the same orbit under the action of the group $\mathcal{A}$. In the language of group actions, saying that every $G' \in \mathcal{G}(S, k)$ can be obtained from $G$ upon completion of finitely many steps of our algorithm amounts to saying that the action of $\mathcal{A}$ on $\mathcal{G}(S, k)$ is transitive (there is only one orbit under this action).

**Definition 4.4.2** *We shall write $G \sim G'$ is and only if $G$ and $G'$ are in the same orbit under the action of $\mathcal{A}$.*

It is well known from group theory (and is very easy to show as well) that $\sim$ is an equivalence relation. Our goal in the current section is to establish the following result:

**Theorem 4.4.1** *The group action of $\mathcal{A}$ on $\mathcal{G}(S,\, k)$ is transitive, or, equivalently, $\forall\ G$ and $G' \in \mathcal{G}(S,\, k)$ we have $G \sim G'$.*

Theorem 4.4.1 is nontrivial and we shall break down the proof into several simple lemmas. First of all, the proof is based largely on the following concept:

**Definition 4.4.3** *Given a graph $G = (S,\, E)$, we shall say that a star set of $G$ is any collection of nodes $K \subseteq S$ such that for every $v \in S$ either $v \in K$ or $v$ is a neighbor of a node in $K$. We shall also refer to $k = \min\{|K| \mid K$ is a star set of $G\}$, the minimal cardinality of a star set of $G$, as the star number of a graph $G$. A graph having star number $1$ we shall name a generalized star. Any node of a generalized star comprising a singleton star set will be referred to as the star center.*

Obviously, a star number of a graph $G = (S,\, E)$ is always bounded above by the total number of nodes in $G$ since $S$ itself is a star set of $G$. In fact, it is bounded above by $\max\{1,\, |S| - 2\}$ since we can consider any spanning forest of $G$ and notice that every tree has at least two nodes of degree 1. by removing all the nodes in the spanning tree of degree 1 (except possibly one of them in case $G$ consists of only 2 nodes) we obtain a star set of $G$ of cardinality $\max\{1,\, |S| - 2\}$. Although the following property is not necessary for proving theorem 4.4.1, we believe it is worth mentioning:

**Proposition 4.4.2** *Let $K$ denote a star set of a graph $G = (S,\, E)$, and suppose $|K| > 1$. Fix $v \in K$ and let $p = v,\, x_1,\, \ldots, x_l,\, w$ denote a shortest path between $v$ and $K - \{v\}$ (of course, $w \in K - \{v\}$). Then $p$ consists of at most $4$ nodes. This means that we can write either $p = v,\, x_1,\, x_2,\, w$ or $p = v,\, x_1,\, w$ or $p = v,\, w$.*

*Proof:* Let $p = v, x_1,\, x_2, \ldots, w$. Note that $x_2$ is not a neighbor of $v$ (otherwise the path $p = v,\, x_2, \ldots, w$ would have been shorter). Since $K$ is a star set of $G$ $x_2$ must be a neighbor of at least some node in $K$. Since $v$ is not such a node, it must be a node $w \in K - \{v\}$ which implies that $p = v,\, x_1,\, x_2,\, w$. □

The reason we introduced the notion of a star number will become clear thanks to the following lemmas:

**Lemma 4.4.3** *Every graph $G \in \mathcal{G}(S,\, k)$ having star number $k > 1$ is equivalent (in the sense of definition 4.4.2) to a graph $G' \in \mathcal{G}(S,\, k)$ with star number $k - 1$.*

*Proof:* Fix a star set $K$ of $G$ of size $k$. pick $v \in K$ and note that $K - \{v\} \neq \emptyset$ (since $k > 1$). Since $G$ is a connected graph, there is a path in $G$ from $v$ to $K - \{v\}$. Let $p = v, x_1,\, x_2, \ldots, w$ with $w \in K - \{v\}$ denote a shortest such path. According to proposition 4.4.2, either $p = v,\, w$, $p = v,\, x_1,\, w$ or $p = v,\, x_1,\, x_2,\, w$. First, observe that $G \sim G''$ where $\{v,\, w\} \in E(G'')$. Indeed, if $p = v,\, w$ then $G'' = G$ will do. If $p = v,\, x_1,\, w$, let $G'' = \pi^{(v,\, x_1)}_{(v,\, x_1),\, (x_1,\, w) \to (v,\, w)}(G)$ and note that $\{v,\, w\} \in E(G'')$ (see definition 4.4.1). Finally, if $p = v,\, x_1,\, x_2,\, w$ then we first let $G''' = \pi^{(x_1,\, x_2)}_{(x_1,\, x_2),\, (x_2,\, w) \to (x_1,\, w)}(G)$ and then apply the argument of the previous sentence to

$G'''$. By transitivity of $\sim$ it suffices to show that $G'' \sim G'$ where $K - \{v\}$ is a star set of $G'$. Again, thanks to transitivity, it suffices to construct a sequence of graphs $G'' = G_1, G_2, \ldots, G_{l+1} = G'$ with $G_i \sim G_{i+1}$ whenever $1 \leq i \leq l-1$. To do so, enumerate the nodes $u_1, u_2, \ldots, u_l$ which are the neighbors of $v$ and not the neighbors of any node in $K - \{v\}$ (if there are no such nodes we would finish right away, since then $K - \{v\}$ is a star set of $G'' = G_1 = G'$). We now let $G_1 = G''$, and let $G_{i+1} = \pi^{(v,\, u_i)}_{(w,\, v),\, (v,\, u_i) \to (w,\, c)}(G_i)$. Notice that the edge $\{w, v\} \in E(G')$, and, by construction, remains an edge of every $G_i$. Therefore, every $G_i$ contains the edges of the form $\{w, u_j\}$ for whenever $1 \leq j < i$ (at the expense of loosing the edge $\{v, u_j\}$) so that $G' = G_{l+1}$ is our desired graph. The desired conclusion now follows. □

From lemma 4.4.3 together with transitivity of $\sim$ and the fact that a star number of a finite graph is finite, the following follows immediately:

**Lemma 4.4.4** *Every graph $G \in \mathcal{G}(S,\, k)$ is equivalent to a generalized star.*

Thanks to lemma 4.4.4 we only have to prove that any two generalized stars are equivalent. The reader can now appreciate the notion of a star number for our purposes. In fact, the following lemma narrows down the types of generalized stars we need to establish the equivalence of:

**Lemma 4.4.5** *Enumerate the nodes in $S$ as $0,\, 1,\, 2, \ldots, |S| - 1$ and let $G \in \mathcal{G}(S,\, k)$ denote a generalized star with star center $0$. Fix any $i$ satisfying $1 \leq i \leq |S| - 1$. Then $G \sim G'$ where $G'$ is the generalized star with the node $i$ being its star center (see definition 4.4.3).*

*Proof:* If $|S| = 1$ there is nothing to prove. Otherwise W.L.O.G. assume $i = 1$ (otherwise just renumber the nodes). Again, we exploit transitivity and construct a sequence

$$G = G_1,\, G_2, \ldots, G_{|S|-1} = G'$$

such that $G_l \sim G_{l+1}$ and every node $j$ with $j \leq l$ and $j \neq 1$ is a neighbor of $1$ in $G_l$. We let $G_1 = G$ and note that $0$ is a neighbor of $i$ since $0$ is a center of $G$. For $l$ satisfying $1 < l < |S|$, let $G_l = \pi^{(0,\, l)}_{(1,\, 0),\, (0,\, l) \to (1,\, l)}(G_{l-1})$. Since $0$ is a star center of $G$ and the edge $\{0, l\}$ was not deleted from any of $G_j$ for $j < l$ it follows that $G_l$ contains the edge $\{1,\, l\}$ at the expense of loosing the edge $\{0,\, l\}$. Notice that the intermediate graphs $G_i$ may have star number 2, but the final graph $G' = G_{|S|-1}$ is a generalized with star center $1$. □

Thanks to lemmas 4.4.4 and 4.4.5, all that remains to show to establish theorem 4.4.1 is that any two generalized stars having a common center (see definition 4.4.3) are equivalent under the relation $\sim$ of definition 4.4.2. We accomplish this task in three steps:

**Lemma 4.4.6** *Suppose we are a generalized star $G \in \mathcal{G}(S,\, k)$. Enumerate the nodes of $G$ as $0,\, 1, \ldots, |S|-1$ with $0$ denoting a star center. Suppose an edge $\{i,\, j\} \in E(G)$ and $\{i,\, l\} \notin E(G)$ for $i$, $j$ and $l > 0$. Then $G$ is equivalent via $\sim$ to the generalized star $G' \in \mathcal{G}(S,\, k)$ determined by the set of edges $E(G') = E(G) \cup \{\{i,\, l\}\} - \{\{i,\, j\}\}$.*

*In other words, $G$ is equivalent to the generalized star obtained from $G$ by removing the edge $\{i,\, j\}$ and inserting the edge $\{i,\, l\}$.*

*Proof:* First let $G'' = \pi^{(i,\,0)}_{(i,\,0),\,(0,\,l)\to(i,\,l)}(G)$ so that $E(G'') = E(G) \cup \{\{i,\, l\}\} - \{\{i,\, 0\}\}$ (notice that $G''$ may not be a generalized star). Now let $G' = \pi^{(i,\,j)}_{(i,\,j),\,(j,\,0)\to(i,\,0)}(G'')$ and notice that $E(G') = E(G'') \cup \{\{0,\, i\}\} - \{\{i,\, j\}\} = (E(G) \cup \{\{i,\, l\}\} - \{\{i,\, 0\}\}) \cup \{\{0,\, i\}\} - \{\{i,\, j\}\} = E(G) \cup \{\{i,\, l\}\} - \{\{i,\, j\}\}$ which is what we were after. □

**Lemma 4.4.7** *Suppose we are a generalized star $G \in \mathcal{G}(S,\, k)$. Enumerate the nodes of $G$ as $0,\, 1, \ldots, |S|-1$ with $0$ denoting a star center. Suppose an edge $\{i,\, j\} \in E(G)$ and $\{q,\, l\} \notin E(G)$ for $i$, $j$, $q$ and $l > 0$. Then $G$ is equivalent via $\sim$ to the generalized star $G' \in \mathcal{G}(S,\, k)$ determined by the set of edges $E(G') = E(G) \cup \{\{q,\, l\}\} - \{\{i,\, j\}\}$. In other words, $G$ is equivalent to the generalized star obtained from $G$ by removing the edge $\{i,\, j\}$ and inserting the edge $\{i,\, l\}$.*

*Proof:* If $q = i$ the situation is exactly that of lemma 4.4.6 and so we assume that $q \neq i$. In such a case, either one of the following mutually exclusive situations can occur: either $\{i,\, q\} \in E(G)$ or $\{i,\, q\} \notin E(G)$.

*Case 1:* $\{i,\, q\} \in E(G)$. In this case, according to lemma 4.4.6 $G \sim G''$ with $E(G'') = E(G) \cup \{\{q,\, l\}\} - \{\{i,\, q\}\}$ and, again thanks to lemma 4.4.6, $G'' \sim G'$ with $E(G') = E(G'') \cup \{\{i,\, q\}\} - \{\{i,\, j\}\} = (E(G) \cup \{\{q,\, l\}\} - \{\{i,\, q\}\}) \cup \{\{i,\, q\}\} - \{\{i,\, j\}\} = E(G) \cup \{\{q,\, l\}\} - \{\{i,\, j\}\}$

*Case 2:* $\{i,\, q\} \notin E(G)$. In this case, according to lemma 4.4.6 $G \sim G''$ with $E(G'') = E(G) \cup \{\{i,\, q\}\} - \{\{i,\, j\}\}$ and, again thanks to lemma 4.4.6, $G'' \sim G'$ with $E(G') = E(G'') \cup \{\{q,\, l\}\} - \{\{i,\, q\}\} = (E(G) \cup \{\{i,\, q\}\} - \{\{i,\, j\}\}) \cup \{\{q,\, l\}\} - \{\{i,\, q\}\} = E(G) \cup \{\{q,\, l\}\} - \{\{i,\, j\}\}$

Thus, in any of the possible cases we deduce that $G \sim G'$ with $E(G') = E(G) \cup \{\{q,\, l\}\} - \{\{i,\, j\}\}$ so that the desired conclusion follows at once. □

Lemma 4.4.7 brings us very close to reaching our final goal. Indeed, let $0,\, 1, \ldots, |S|-1$ enumerate the nodes of a generalized star graph $G \in \mathcal{G}(S,\, k)$ with $0$ denoting the star center. Notice that $G$ is uniquely determined by the binary sequence of length $\frac{(|S|-1)(|S|-2)}{2}$ indexed by all pairs $(i,\, j)$ satisfying $1 \leq i < j \leq |S| - 1$ and having a 1 in position $(i,\, j)$ if and only if $\{i,\, j\} \in E(G)$. Such a sequence contains exactly $k - |S| + 1$ 1s and the rest are zeros. Lemma 4.4.7 tells us that transposing a one and a zero in this sequence results in an equivalent generalized star. It is well known that every permutation is a product of transpositions and so we deduce

**Lemma 4.4.8** *Any two generalized stars in $\mathcal{G}(S,\, k)$ having a common star center are equivalent via $\sim$.*

In summary, combining lemmas 4.4.8 with lemma 4.4.5 tells us that any two generalized stars are equivalent. Now, given any two graphs $G$ and $G' \in \mathcal{G}(S,\, k)$, according to lemma 4.4.4, $G \sim G_1$ and $G' \sim G_2$ with $G_1$ and $G_2$ being generalized stars and, according to the previous sentence, $G_1 \sim G_2$ so that we finally have $G \sim G_1 \sim G_2 \sim G'$ so that theorem 4.4.1 now follows.

# Bibliography

[1] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness.* Freeman and Company, San Francisco., 1979.

[2] R. Gomory and T. Hu. Multiterminal network flows. *SIAM Journal on Applied Mathematics*, 9:551–570, 1969.

[3] T. Hu. Optimum communication spanning trees. *SIAM Journal on Computing*, 3(3):188–195, 1947.

[4] HWU. Derivation of network topology from combining graph-theoretic and system requirements. DBE Deliverable 19.2.

[5] SUN. Dbe nervous system. DBE Deliverable 19.1.

[6] UBHAM. Report on clustering in networks. DBE Deliverable 11.2.

[7] UBHAM. Report on the flow of software components in a static network. DBE Deliverable 11.1.