# Cost-effective infrastructure in a multi-antenna telescope layout

G. Grigorescu[1] P. Alexander[1,2] R. Bolton[1] R. McCool[3]

June 4, 2009

## Abstract

The future Square Kilometre Array (SKA) radio telescope is an interferometer array that will use a variety of collector types, including approximately 2500 dishes distributed with separations up to a few thousand kilometres, and about 250 aperture array (AA) stations located within 200 km of the core. The data rates associated with each individual collector are vast: around 10 GBytes/s for each dish and 2 TBytes/s for an AA station. As each of these must be connected directly to a central correlator, designing a cost-effective cabling and trenching infrastructure presents a great engineering challenge. In this paper we discuss approaches to performing this optimisation.

In graph theory, the concept of a minimum spanning tree (MST) is equivalent to finding the minimum total trench length joining a set of $n$ arbitrary points in the plane. We have developed a set of algorithms which optimise the infrastructure of any given telescope layout iteratively, taking into consideration not only trenching but also cabling and jointing costs as well. Solutions for an example configuration of telescope layout are presented. We have found that these solutions depend significantly on the collectors' output data rates. When compared to a "traditional" MST-based approach which minimises trenching costs only, our algorithms can further reduce total costs by up to 15-20%. This can influence greatly the SKA infrastructure related costs.

1. Astrophysics Group, Cavendish Laboratory, J.J. Thomson Avenue, Cambridge CB3 0HE, UK, Email: g.grigorescu@mrao.cam.ac.uk
2. Kavli Institute for Cosmology Cambridge, University of Cambridge, Madingley Road, Cambridge CB30HA, UK
3. Square Kilometre Array Program Development Office, Jodrell Bank Centre for Astrophysics, School of Physics and Astronomy, University of Manchester, Oxford Road, Manchester M13 9PL, UK

## 1 Introduction

The Square Kilometre Array (SKA) telescope project is a large international effort to build the next generation of radio telescope [1]. By providing wide field-of-view coverage and a sensitivity about two orders of magnitude better than the Very Large Array [2], the SKA will be an excellent survey instrument delivering transformational science across a wide range of astrophysical research areas [3].

The outline design of the SKA has been described in Schillizzi et al. [4]. The telescope will be an interferometer array, comprising approximately 2500 dishes, operating from about 700 MHz to 10 GHz. These will be distributed over continental distances (up to at least 3000 km from the centre) but will be concentrated in a core a few kilometres across. In addition to the dishes there will be around 250 aperture array stations, operating at lower frequencies, from about 70 MHz up to 1 GHz and distributed over a few hundred kilometres [5, 6]. Phased array feeds may be placed on some of the dishes to give an increased field-of-view.

The design of the SKA presents many technical challenges [7]. This paper is motivated by the challenges involved in transporting data signals around the telescope. Because the SKA will be an interferometer, the data from the individual collectors must be combined in a correlator, located near to the centre of the array. Although long-distance data transport is done commercially, because of its necessarily remote location, the SKA will require a dedicated fibre-optics network. With data rates of the order of 10 Gbytes/s from each dish and possibly 2 Tbytes/s for the AA stations, the total data rate to the correlator is set to be of the order of 520 Tbytes/s, dwarfing the current (end 2008) world internet traffic rates [8].

With so much data to transport over such large distances, it is likely that well over €100 million will need to be spent on equipment to generate, condition and receive the signals and on fibre and trenching to carry the data. Many millions of Euros can potentially be saved if the data-transport network can be optimised effectively by changing the fibre paths to trade off the amount of trenching that must be dug against the length of fibre and the quantity of networking that is required. Individual data links cannot be treated separately in the search for a global cost minimum and so, with close to 3,000 collectors, such a complex problem cannot be solved by hand. In this paper we present algorithms that we have developed which can be used to find a optimum networking layout for a given configuration of collectors. We illustrate these with some example networks.

## 2 Defining the problem

In a simplified approach, the main question to be answered is the following: given a collection of collectors located in a plane, where for each collector a position $(x, y)$ and an associated data rate are known, what would be the most cost effective way to route the signal of each collector down to one (or more) central base (CB) location(s)? An optimal solution will depend on parameters such as prices per unit length of trench and cable as well as on the position of the CB. The problem is often encountered in commercial telecommunication network design where constraints such as obstacles or topographically excluded regions can arise. Other applications dealing with similar optimisations include printed circuit board design [9], thin soap film solutions [10], urban planning of pipe networks, etc.

The problem is defined by several variables. These are:

- a telescope layout: a list of collectors, each collector $i$ being uniquely specified by an integer and having a position in space $(x_i, y_i)$ and an output data rate $D_i$[Gbytes/s].
- a list of one or more central bases, acting as signal sinks.
- a 2-dimensional site mask (terrain map) of local trenching costs per unit length acting as a trenching cost metric. In the simplified case, this value is constant over the entire telescope area. More realistically, this map can also simulate areas of no access by specifying extremely high trenching costs in those areas.
- a list of possible cable choices and the associated band-widths and costs per unit length. This is especially important in the context of individual collector output data rates possibly varying for each collector. We assume that more than one cable can be used in one trench.

Based on the above input, the expected solution is represented by an infrastructure layout specification of all trenching and cabling such that the total costs associated to this infrastructure are minimum.

## 3 Overview of relevant mathematics

From a mathematical point of view, the problem is similar to finding a minimum path length that interconnects $n$ vertices located in the plane. In graph theory, the problem is known as the *Euclidean minimum spanning tree* (EMST). An example is given in Fig. 1, where a set of 25 points in the plane are joined by a minimum path length, defined as a summation over all links $ij$ : $S_T = \sum s_{ij}$, with $s_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ as the Euclidean distance between points $i$ and $j$. Randomised algorithms of complexity $\mathcal{O}(n \log n)$ are known. For instance, Prim's algorithm or Krushal's algorithm can be employed [11, 12].

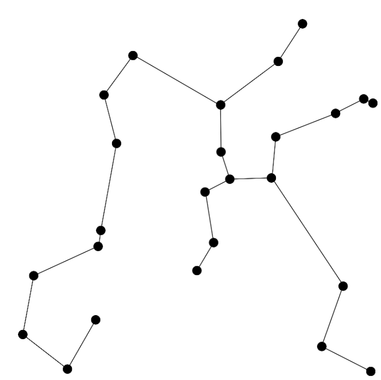

Figure 1: The Euclidean minimum spanning tree associated to a set of 25 points in the plane.

Finding a shorter total path length than the EMST can be achieved by introducing new vertices. Steiner showed that the total path length of a tree interconnecting a set of $n$ points lying in the plane can be reduced with respect to the EMST, by introducing a maximum of $n-2$ intermediate vertices which act as hub points, also known as Steiner points [13]. The Steiner points must have a degree of 3 - i.e. connect to three other vertices - and the three edges incident to such point must form three 120° angles. For a given set of points, there may be several Steiner trees. Finding the Steiner tree of a graph is, in general, an NP-complete problem, thought to be computationally difficult. In practice, heuristics and iteration are used: namely, a large collection of points is divided into sub-sets of three or four points and their Steiner trees are individually computed and then joined together. Steiner trees for sets of three and four points respectively are shown in Fig. 2. The length of a Steiner minimum spanning tree cannot be shorter than $\sqrt{3}/2$ (about 0.86) times the length of the EMST [14].

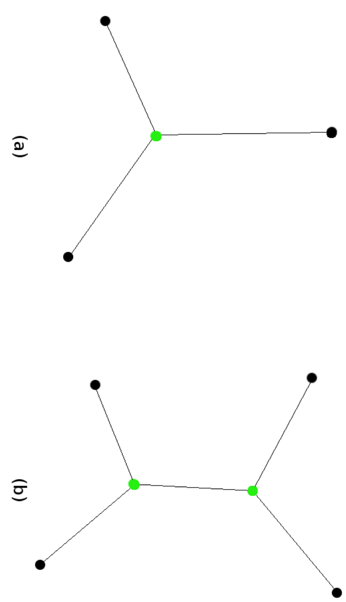


Figure 2: Examples of Steiner trees for sets of three points (a) and four points (b). Steiner points are shown in grey.

# 4 A cocktail of algorithms

Although there are obvious similarities between finding the Steiner tree of a graph and finding an optimal network infrastructure, a few important differences hint to a different optimum and therefore dictate a different approach. For instance, not all elements in a telescope network play an equal role: the location(s) of the CB(s) will dominate the optimisation. Also, certain configurations may decouple into several sets of collectors, each set being independently connected to the CB(s) from the others. Other crucial components to be taken into account, apart from the trenching, are the cables which, depending on band-width, come in different varieties with different costs. This is an essential consideration since different collectors may have significantly different data-rates. Variable trenching costs, depending on location, must also be allowed for, together with areas which are topographically inaccessible. All those differences can lead to an optimal infrastructure which may differ from the EMST. For example, in the extreme case of very expensive data links and low cost trenching, the solution for a sparse configuration consists of all the collectors being directly connected, through individual trenches, to the CB and thereby minimising the total cost, which is driven mainly by cost of the cable. This particular solution maximises the total trench length.

If both cable and trench costs are significant, the optimum solution for the network layout will lie between the extreme configurations of the EMST, which minimises trenching costs and the direct connecting of each collector to the CB, which minimises the cable costs.

The solution for the optimal network layout is found by iteration. We have developed a set of algorithms which can be applied successively. They each explore different approaches and aspects of the optimisation and several solutions may be generated by running different combinations of these algorithms. In general, these solutions converge to a similar final cost and configuration although computational times may differ substantially.

The total cost, $TC$, consists of three different contributions:

$$TC = K_t + K_c + K_e. \tag{1}$$

These contributions are as follows. $K_t = k_t \cdot \sum_{links} s(i,j)$ represents the trenching costs. $k_t$ is the trenching cost per unit length and may in general be position dependent. The summation is carried out over all links. $K_c = \sum_{cable\,types} \sum_{links} k_c \cdot s(i,j)$ is the total cost of cables, $k_c$ is the cable cost per unit length of a particular cable of type $c$. $K_e$ are costs associated with non-collector elements such as Steiner points.

We consider a telescope to be a collection of elements, each element being defined by the following attributes: a unique integer identifier; a cartesian position in the plane $(x, y)$; input, output and intrinsic signal data-rates, all of which can be zero; a pointer to the other element to which the current element is connected; a size and a cost; other variables related to book-keeping. In this way, no distinction is made between collectors, CB's and other passive elements of the telescope such as possible Steiner points which can be added during optimisation. The direction of the information flow is important as it is used to calculate the data rate transported through each trench. Elements are always connected such that no loops are created in the data flow.

The starting point we choose for the optimisation is to construct the minimal cable-cost network by connecting each collector directly to the nearest CB. Our experiments indicated that the final solution does not depend on this initial configuration. As mentioned above, the optimisation proceeds by employing a series of algorithms sequentially which we now discuss in detail.

*Connect Lines to Neighbours*: The purpose of this algorithm is to break longer more direct links from collectors further away from the CB into a network including more shorter links, making use of the elements closer to the core as hub points. It is implemented as follows: for each link in the telescope, let elements $i, j$ be the the start, stop points respectively of the link, where start and stop are defined by the data flow direction. A search for the closest element $k$ to the line $ij$ is performed. If the projection of $k$ on the line $ij$ falls inside the segment $ij$, the element $i$ is disconnected from $j$ and reconnected to $k$; a new $TC$ is computed; if $TC_{new} < TC_{old}$ then the newly made connection is retained.

*Add Steiner Points*: This routine deals with reshaping the connections for groups of three elements by introducing Steiner points, as shown in Fig. 2. First, the algorithm searches for the following: all elements $i$ to which at least two other elements connect: $j \to i$, $k \to i$; or all three elements $i$, $j$, $k$ are connected in a chain: $k \to j \to i$. If all the angles of the $ijk$ triangle are less than $120°$, the corresponding Steiner point $F$ is constructed. Then, both $k$ and $j$ elements are connected to $F$ and $F$ is further connected to $i$. $TC$ is recomputed and the new connections are accepted if $TC_{new} < TC_{old}$.

*Relax Steiner Points*: As the Steiner points added by the previous routine were constructed such that the total path length within the triangles was minimised, these points do not ensure cable costs are minimum. Nevertheless, the Steiner points are good first estimates. This routine allows the Steiner vertices created previously to relax by moving iteratively in four possible orthogonal directions. The step size is taken to be a fraction of the distance to the closest vertex of the triangle. The chosen direction of movement is the one which gives the largest negative gradient in $TC$. The variable step size ensures a good compromise between execution time and precision. There are two possible outcomes: either the Steiner point settles to a new position which minimises the costs for the triangle or the Steiner point moves such that it approaches one of existing vertices of the triangle until they coincide. In the latter case, the Steiner point is removed from the construction and new connections are made accordingly.

*Remove Steiner Points*: This routine removes all Steiner points which have lost their purpose as hub points. For instance, this situation can occur when, after creating and relaxing Steiner points, the *Connect Lines to Neighbours* algorithm is applied successfully once more, leaving some of the Steiner points with less than two affluents. In this case, those Steiner points are not useful any longer and removing them reduces the costs even further.

*Connect in Pairs*: This is a brute force algorithm which tries to minimise total cost by testing new

Approximating the surface of the Earth by a plane with an Euclidean metric remains valid for small distances. When the size of the entire SKA is considered, surface curvature plays an important role in accurately accounting for the distances along long baselines. Although the metric used in this paper is Euclidean, the arguments presented here hold for any other surface and metric.

links between all possible pairs of elements. It is implemented as follows: for all possible pairs $(i, j)$, let $d_i$ be the separation between element $i$ and its current connection, $d_j$ the separation between $j$ and its current connection and $d_{ij}$ the separation between $i$ and $j$. If $d_{ij} < d_i$ and $d_{ij} < d_j$, then $i$ is connected to $j$; $TC$ is recomputed. The newly made connection is kept only if $TC_{new} < TC_{old}$. This algorithm is extremely time consuming. Also, it is possible for the iterative scheme to get stuck in a local minimum. Nevertheless, the algorithm is helpful when improving on already optimised solutions and it can be further conditioned by restricting it to those pairs $(i, j)$ which pass certain criteria such as a minimum separation $d_{ij}$ or certain angle restrictions in the triangle formed by the elements $i$, $j$ and the closest CB to $j$.

*Reverse Lines*: Because links have directional flow and configurations are optimised iteratively, it can happen that solutions are trapped in local minima. This routine probes whether certain solutions are only local minima which differ substantially from a global minimum. Essentially, the routine tracks long linear (one-to-one) chain connections and tries to reverse either the whole chain or individual parts in order to reduce $TC$. Local minima happen more frequently in sparse, spiral-arm collector configurations where the flow winds around local CB's. A trivial example is given in Fig. 3 where configuration (a) is improved by tracking a linear chain of connections, disconnecting the downstream link, reversing the connection flow and reconnecting the chain at the old upstream end in the most cost-effective way, as in configuration (b).

*Connect Serially*: This algorithm cycles through the entire set of collectors in an orderly fashion, from the closest to the furthest of the collectors with respect to the CB, and attempts to join them serially such that the total cost decreases with every attempt. This algorithm performs well for configurations which spiral away and wind around the CB's. It is meant as an alternative to the *Connect Lines to Neighbours* algorithm, when the latter becomes stuck in a local minimum. Also, if run at first iteration, *Connect Serially* can improve the total cost significantly within one cycle.

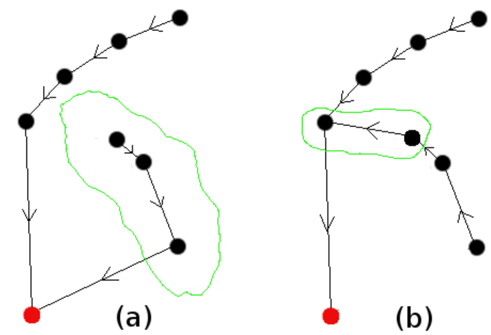


Figure 3: Example of *Reverse Lines* algorithm in action: the flow in a linear chain of connections is reversed (a), the most down stream link is disconnected and a more cost-effective connection is made at the up stream end (b).

# 5 Problem complexity and scaling

The scaling of algorithm complexity with the number of collectors is not trivial. First of all, the performance of all algorithms depends on the evolution of one parameter, the total cost TC. At each step within one cycle, changes in the infrastructure are performed only if TC decreases. Computing changes in TC is in itself a time-expensive task as it requires cycling through branches of collectors and summing individual costs corresponding to each encountered link in the network. Some of the optimisation algorithms scale as $\sim n$, such as *Connect Lines to Neighbours* or *Connect Serially*, as they cycle through the entire set of collectors. Some other algorithms, such as *Connect In Pairs* scale as $\sim n^2/2$ as they cycle through all pairs of collectors. Also, at each step within one algorithm cycle, other actions are performed, such as looking for the closest neighbour or closest link, which also may scale as $\sim n$. Indeed, the overall computation time increases in general at least as fast as $\sim n^2$. This scaling is inevitable and is justified by the fact that in our model the local change of a link impacts the global optimisation of the rest of the telescope, as illustrated in a trivial example in Fig. 4. For instance, attempting to disconnect the subset of collectors (1) from subset (2) and reconnect it to subset (3) will not only produce a shorter connecting trench between (1) and (2) but will affect many data rates flowing through some other trenches in all subsets: data rate output of subset (2) will decrease whereas that of subset (3) will increase. In consequence, the optimisation of cabling and possibly trenching within each subset will be affected, following the link change (1) → (2) to (1) → (3). Therefore, as local changes affect the global optimum, an optimum solution will require significant computation time. Fortunately, the total number of collectors for the SKA telescope will not exceed a few thousand. These configurations can therefore be optimised within reasonable times using the numerical computational framework discussed above.

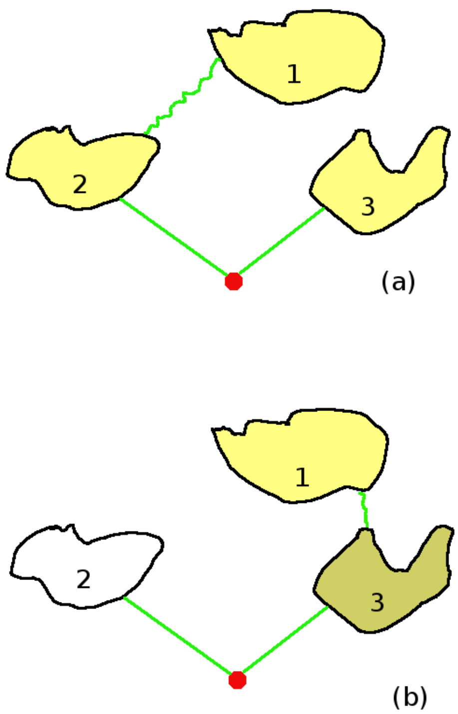


Figure 4: Example of how local change may affect the overall optimisation. Each 'blob' represents a subset of many collectors. All subsets have already been optimised within themselves. Attempting to disconnect the collector subset (1) from subset (2) and reconnect it to subset (3) will not only imply a shorter connecting trench but also a a change in the data rates flowing through many other trenches of subsets (2) and (3) and possibly (1).

# 6 Examples and discussion

Two examples are presented in Fig. 5 and 6. Some input parameters used for optimising the following configurations are listed in Table 1. The links were modelled using five options for cabling, namely optic fibre cables consisting of 1, 2, 6, 9 and 24 tubes. Each tube consists of 8 strands of fibre. Through each strand, we assume that 16 different colour channels can be broadcast, each channel having a maximum bandwidth of 10 Gbits/s. The input values for trench and cable costs were chosen to reflect realistic commercial values [15]. Cost wise, all Steiner points were treated as commercial optic fibre joints.

The two examples, Fig. 5 and 6, are two different optimisations of the same configuration of 165 collectors with a data rate of 2 TBytes/s each, consisting of a highly concentrated inner core of 1 km diameter and a sparse random distribution up to a 5 km diameter around the CB. This is a reasonable distribution of the central part of the AA station set. The infrastructure in Fig. 5 was optimised intentionally for total trench length only and it has an associated total cost of €898.55k, of which €560.4k are cable costs and €338.15k are trench costs. The infrastructure in Fig. 6 was optimised for total trench and cable costs combined and it has an associated total cost of €748.82k, of which €395.27k are cable costs and €353.56k are trench costs. As the individual data rate of each collector is high, about half of the total costs will be spent on cables. Therefore, this comparison shows that although the infrastructure in Fig. 5 has indeed a lower total trenching cost, the total cost is reduced in Fig. 6 by more than 16%. Namely, a trade-off of 10% in trench costs is necessary such that total costs are reduced by 16%. This is indeed a consequence of properly incorporating the cable costs into our optimisation procedure. It was also found that for the configuration shown in Fig. 5 and 6, the addition of Steiner points (not shown in the figures) would further reduce the total cost by about 1% only. This finding is justified by the fact that the inner core is too dense to accommodate cost-effective Steiner points. The quoted 1% reduction is due to only a few Steiner points, placed outside the inner core, in the sparse section of the configuration. There, the high-data rate and therefore the expensive cables conspire against the need of more

Steiner vertices, as that would require a higher total cable length. In general, Steiner points make a greater impact on the total cost for configurations with low data rate collectors and sparse collector distributions [16].

These examples show that a proper treatment of various infrastructure components and associated costs not only can reduce significantly the total cost but also can provide us with a valuable insight of how different components (Steiner vertices, cables, trenching) contribute differently to the total cost, as a function of collectors' layout and data rates.

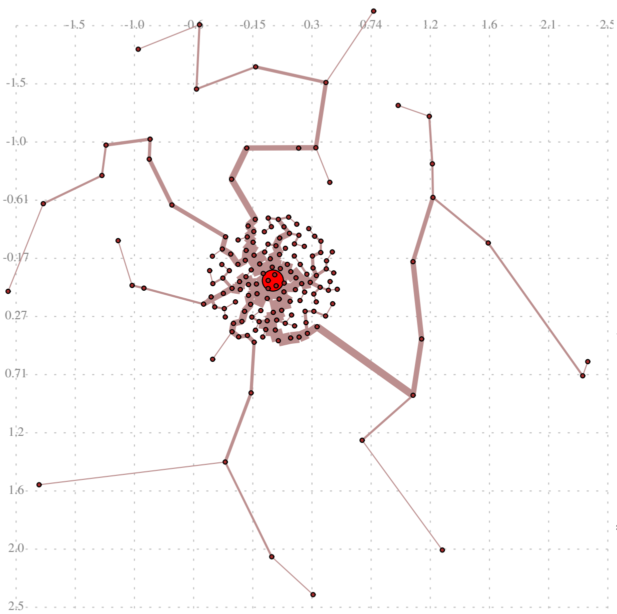

Figure 5: An example configuration of 165 collectors with a data rate of 2 TBytes/s each, consisting of a highly concentrated inner core of 1 km diameter and some sparse random distribution within 5 km diameter from the CB. The infrastructure was optimised for total trench length only. Total cost is €898.55k, of which €560.4k are cable costs and €338.15k are trench costs.

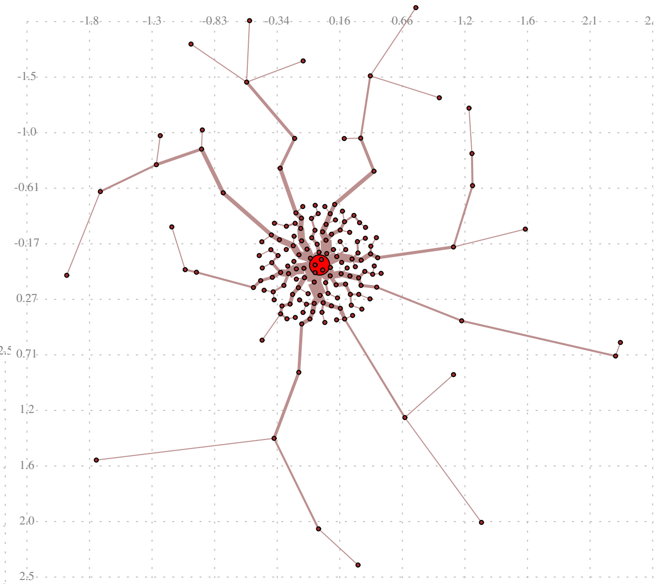

Figure 6: An example configuration of 165 collectors with a data rate of 2 TBytes/s each, consisting of a highly concentrated inner core of 1 km diameter and some sparse random distribution within 5 km diameter from the CB. The infrastructure was optimised for total trench and cable cost. Total cost is €748.82k, of which €395.27k are cable costs and €353.56k are trench costs.

| Name | Value |
|---|---|
| trench costs | €12k/km |
| cable costs (1 tube) | €0.85k/km |
| cable costs (2 tubes) | €1.2k/km |
| cable costs (6 tubes) | €2.05k/km |
| cable costs (9 tubes) | €2.7k/km |
| cable costs (24 tubes) | €5.1k/km |
| Steiner point costs | €0.4k |

Table 1: Input parameters used for optimising the infrastructure of the configuration presented in Fig. 5 and 6.

# References


[1] Square Kilometre Array. www.skatelescope.org/.

[2] Very Large Array. www.vla.nrao.edu/.

[3] eds: C. Carilli, S. Rawlings. Science with the Square Kilometre Array. *New Astronomy Reviews*, 48, 2004.

[4] R.T. Schilizzi et al. Preliminary Specifications for the Square Kilometre Array. www.skatelescope.org/PDF/memos/100_Memo_Schilizzi.pdf.

[5] R.C. Bolton et al. SKADS Benchmark Scenario Design and Costing 2 (The SKA Phase 2 AA Scenario). www.skads-eu.org/PDF/SKADS_D&C_second_document_05112008_final.pdf.

[6] P. Alexander et al. SKADS Benchmark Scenario Design and Costing. www.skads-eu.org/PDF/SKADS_Benchmark_Scenario_D+C_v1.0.pdf.

[7] P. J. Hall. The SKA: An Engineering Perspective. *Experimental Astronomy*, 17, 2004.

[8] Minnesota Internet Traffic Studies. www.dtc.umn.edu/mints/home.php.

[9] G. Bhaskar and T. T. Narendran. Grouping PCBs for set-up reduction: A maximum spanning tree approach. *International Journal of Production Research*, 34:621–632, 1996.

[10] Cyril Isenberg. *The Science of Soap Films and Soap Bubbles*. Dover Publications, 1992.

[11] R. C. Prim. Shortest Connection Networks and Some Generalizations. *Bell System Technical Journal*, 36:1389–1401, 1957.

[12] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 2001.

[13] F. K. Hwang, D.S. Richards, and P. Winter. The Steiner Tree Problem. *Annals of Discrete Mathematics*, 53, 1992.

[14] D. Z. Du and F. K. Hwang. A Proof of the Gilbert-Pollak Conjecture on the Steiner Ratio. *Algorithmica*, 7:121–135, 1992.

[15] Serenate project. www.serenate.org/publications/d14-serenate.pdf.

[16] B. Bell. http://cse.taylor.edu/~bbell/steiner/.