Intern. J. Computer Math., 2003, Vol. 80(4), pp. 441–455

EGONET: A GENETIC ALGORITHM MODEL FOR THE OPTIMISATION OF TELEPHONE NETWORKS

K. TOUT[a]*, E. VON GOLDAMMER[b] and D. J. EVANS[c]

[a]*Faculty of Sciences I, Lebanese University, Beirut, Lebanon;* [b]*Institut fuer Kybernetik & Systemtheorie, 44388 Dortmund, Germany;* [c]*Dept. of Computing, Nottingham Trent University, Nottingham, UK*

(Received 7 October 2002)

The aim of this paper is to study the use of genetic algorithms for the optimisation of telephone networks layout design. The genetic algorithms are used to design the network in a geographical sense with minimum costs depending on a set of real-world rules (constraints). A genetic algorithm tool (EGONET) is described. EGONET is written in ANSI-C and thus can be used on a wide range of platforms. EGONET was tested under Unix/Linux, OS/2 and Windows 98.

Keywords: Optimisation; Heuristic algorithms; Telephone networks; Genetic algorithms; Networks layout design

C.R. Categories: C.2.1, I.2.8

1 INTRODUCTION

The designers of a telephone network layout are mainly experts in the fields with a lot of experience. To develop new networks they use constraint rules or rules of thumb gained by experience. The rules used are dependent on the personal expertise of these people, which means that almost every expert uses his/her own design strategy. This implicates also that there is no well-defined, general accepted method to design a network layout. New techniques such as Genetic Algorithms can result in new strategies, which are probably better or at least can help to find guidelines to make the process of optimised-network layout design a more uniform process within a company.

2 SHORT DESCRIPTION OF THE PROBLEM

The basic structure of a telephone network is depicted in Figure 1. The transportation network transports messages from the switch board to the so called Cross Connection Points (CCP).

* Corresponding author. E-mail: ktout@ul.edu.lb

ISSN 0020-7160 print; ISSN 1029-0265 online
DOI: 10.1080/0020716021000059124

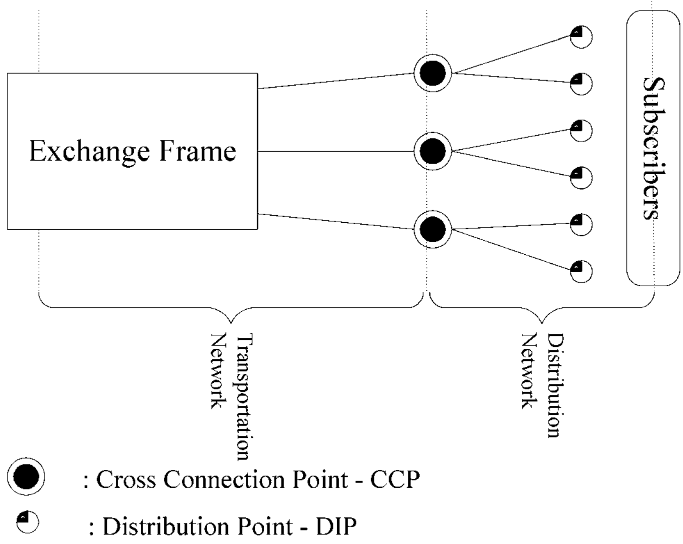

FIGURE 1 Principle structure of a telephone network.

The CCP in its turn connects several subscribers to the telephone network; this part is called the distribution network. The CCP is connected through cables to Distribution Points (DIPs). The DIPs are connected directly to a group of subscribers (telephone customers). If a cable is split or is bent within a certain angle, an Underground Room (UGR) is built.

The focus of this paper is on the optimisation of the distribution network. In other words: "How can all subscribers be connected to the network by travelling from the CCP through all DIPs with a minimum of costs for cables and underground rooms?"

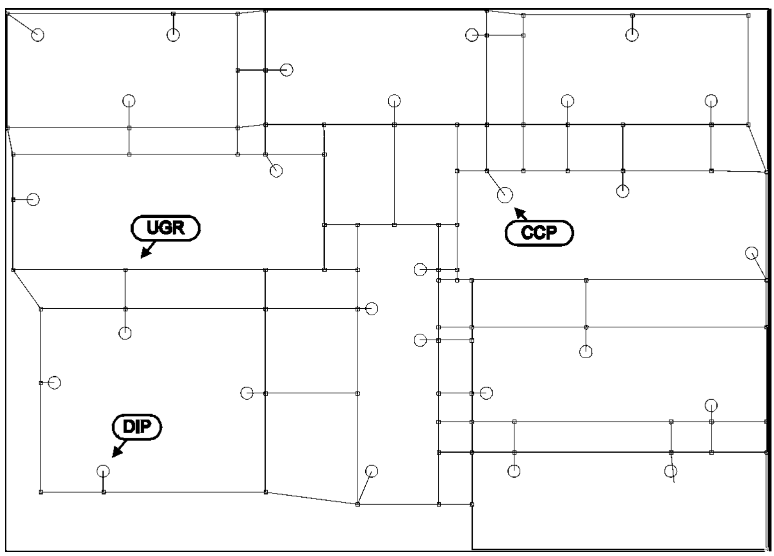

FIGURE 2 A potential network (P-NW) with all allowed connections between a given CCP and the DIPs.

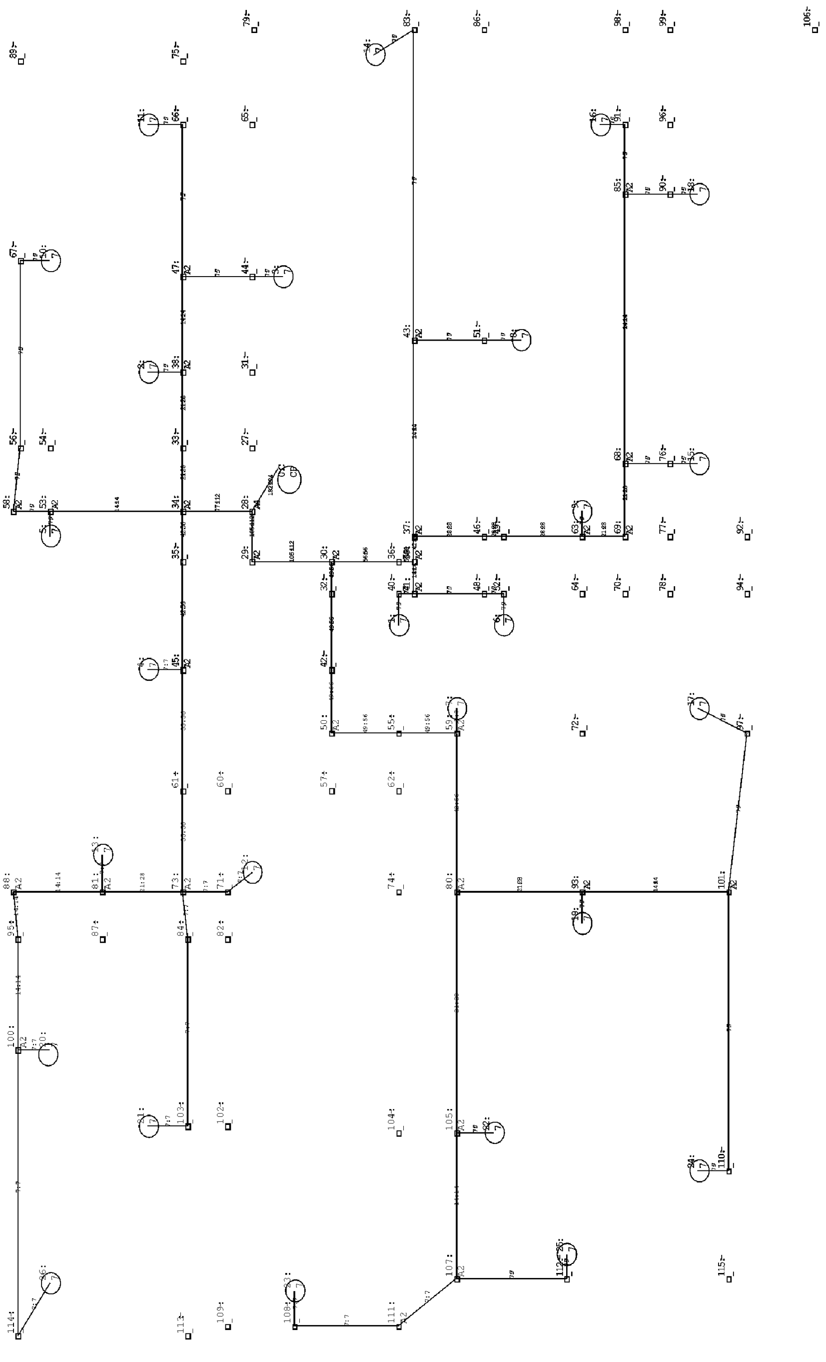

FIGURE 3(a) An optimised network (O-NW).

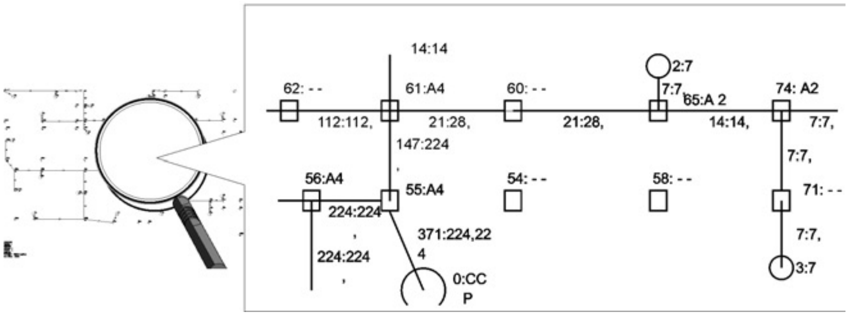

FIGURE 3(b) The optimised network depends on the cable capacities and the type of UGR.

In a geographical area a CCP is given together with several DIPs. There are certain connections allowed depending on some given rules (constraints) which results in a potential network that mathematically represents a graph.

A potential network (P-NW) is represented by a graph with three types of nodes (CCP, UGR, DIP). The potential network has to be optimised resulting in the optimised network (O-NW) which corresponds to a tree with the CCP as root and the DIPs as leaves. An optimised network O-NW for the P-NW given in Figure 2 is shown in Figures 3(a) and 3(b).

The capacity of a cable is equal to 7×2^n, where 7 is the number of pairs in the cable and n can have any value in the set $\{0, 1, 2, 3, 4, 5, 6, 7, 8\}$. In Tunisia, for example, the maximum capacity of a distribution cable is 448 pairs, which corresponds to 7×2^6. Cables connecting the CCP to the DIPs can be split into halves or quarters as described by the numbers in Figure 3. The costs for the cable layout depends strongly on the special configuration used, *e.g.* the costs for 3 cables with 7 pairs are different than for one cable with 21 pairs.

3 DESIGN AND CONSTRUCTION OF MAPS WITH TELEPHONE NETWORKS

In order to work with the optimisation tools developed within the project, an engineer has first to design a potential network (P-NW). The P-NW contains all possible connections, and all underground rooms (UGR) necessary for the realization of the P-NW. It also contains the CCP (central connection point). From the network P-NW, the optimised network (O-NW) can be calculated using the optimisation tools implemented into a CAD system.

In the past, the P-NW had to be drawn by hand on a paper copy of a map that contains buildings, streets, and footpaths. From these manually drawn maps the P-NW had to be extracted again by hand on a digitising board using an appropriate software tool.

This procedure for the design of an optimal telecommunication network (O-NW) is not acceptable. Therefore two possible alternative ways have been envisaged that could be used in the future by the engineers in Tunis and/or by other providers:

3.1 Bitmap-based Approach

With this approach, the manually drawn geographical map, the drawing, is scanned and the resulting bitmap gives the basis, *i.e.*, the graphical background, for the construction of the P-NW using adequate CAD software tools.

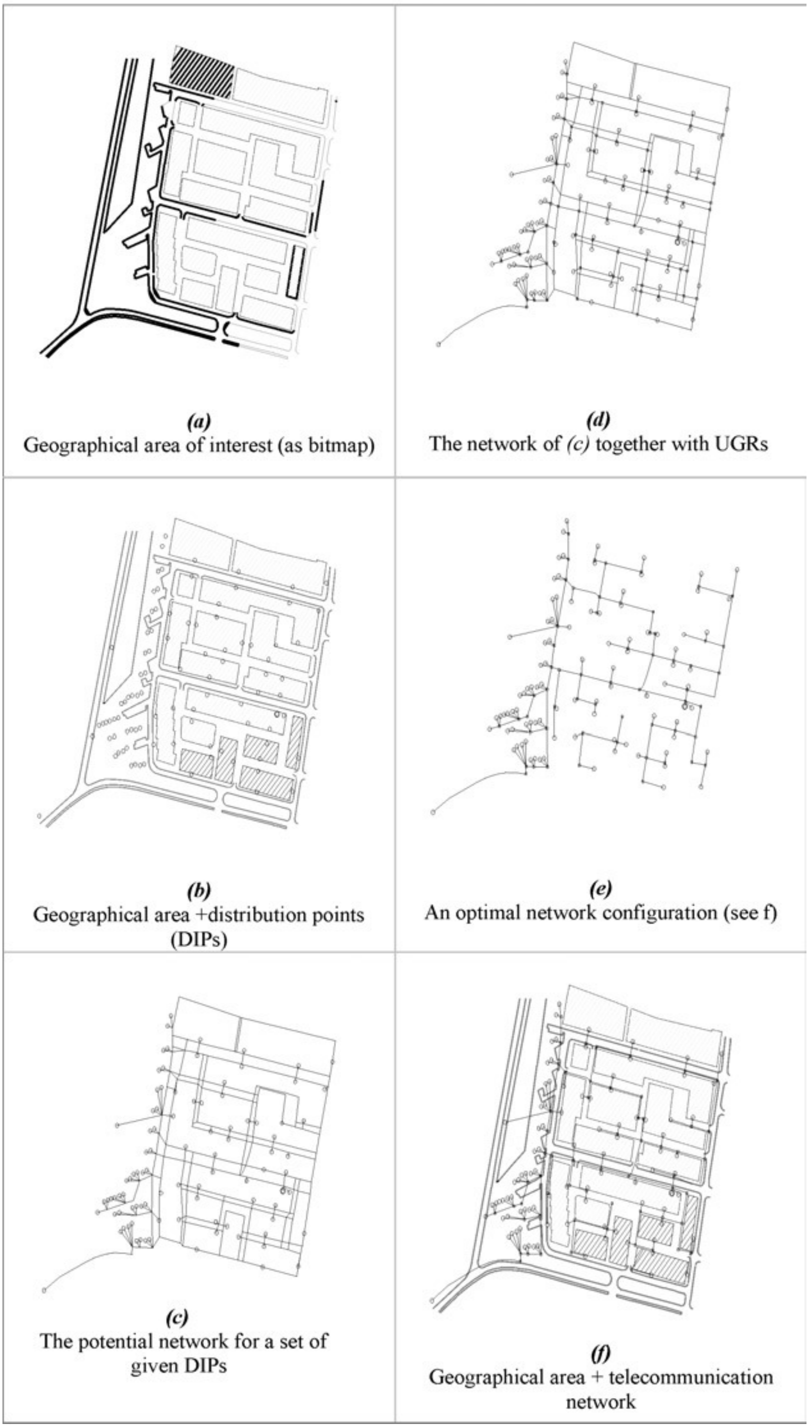

FIGURE 4 Steps towards an optimised telephone network (using the bitmap approach): (a) the map of the geographical area of interest (as bitmap background); (b) the distribution points and the CCP have been added to (a) using the CAD functions of the tool; (c) the resulting potential network; (d) the potential network with underground rooms (UGR); (e) the optimised network using the optimisation functions of the tool; (f) the result which can be plotted and electronically archived.

In this way the P-NW is obtained as a vector data file which can be transferred into a DXF file for the optimisation process. The resulting O-NW again can be transferred into a DXF file for the CAD system and can be plotted (in a correct scale) either separately or together with the underlying bitmap using standard CAD options. The different steps for the procedure are depicted in Figure 4.

3.2 Vectormap-based Approach

If the geographical map already exists in a vectorized and interpreted form, as it is the case for a Geographical Information System (GIS), the optimisation tool can be designed in such a way that the O-NW will be calculated without the foregoing manual construction of a P-NW.

4 THE OPTIMISATION TOOL

The developed telephone network optimisation tool consists of four different modules as depicted in Figure 5. These are:

EGONET This is a Genetic Algorithm tool written in ANSI-C. The tool can be implemented on all platforms using different operating systems. EGONET was tested under Unix/Linux, OS/2, and Windows 98.

Mutation Selection Threshold Accepting (MS-TA) This tool is also written in ANSI-C and therefore can be implemented on platforms using different operating systems.

CAD Functions Since the optimisation has to be performed on manually drawn (analogue) maps, the tool was implemented into a CAD software package. For this implementation the data exchange format is of great importance. Therefore a standard (vector) format was defined at the beginning of the project. The CAD functions of the tool are necessary for the construction of the potential telecommunication network (P-NW). This (computer aided) construction can be

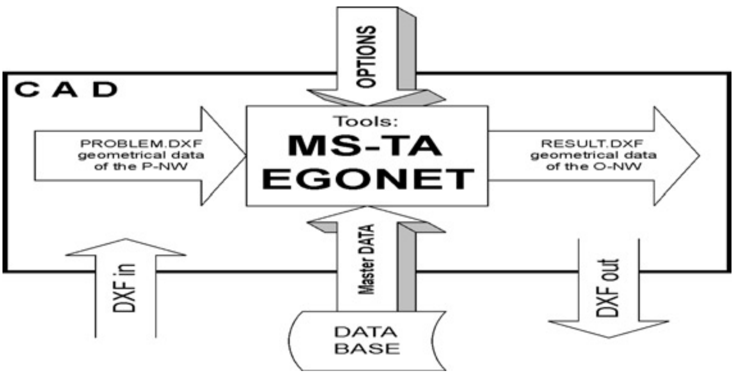

FIGURE 5 The optimisation tool.

done with the developed tool by using the corresponding bitmap as background. The bitmap can be obtained simply by scanning the manually drawn (paper) map.

Data Base Application Since the costs are variable, *i.e.*, they may differ from one country to another (*e.g.*, in Tunisia no manpower costs for digging is considered), a data base application was developed and implemented into the CAD software package. Again ANSI standards have been used in order to have a platform independent product.
The optimisation tool works as follows:

– Import data into the tool.
– Use EGONET to find a solution.
– Export data into a CAD system.
– Visualise both the original and the optimised network.

5 CONSTRAINTS

The constraints have been collected using a terminology as if the cable laying will be performed from the DIPs to the CCP. It is important to note that in reality the cable laying is performed from the CCP to the DIPs. However, it is simply a matter of semantics to speak of combining rather then dividing cables.

The following constraints only consider local networks in a smaller city area (with only one connection point, CCP). For larger networks, similar constraints are to be considered, however, the cable capacities, the number of CCPs, etc. will be different.

(1) *Size of the cable* (*number of wires within a cable rope*): $N = 7 \times 2^n$ with n as an integer in the range $0 \leq n \leq 6$ ($N = 7, 14, 28, 56, \ldots, 448$).
(2) *Diameter* (Q) *of the cable*: $Q \in \{0.4\,\text{mm}^2, 0.6\,\text{mm}^2, 0.8\,\text{mm}^2\}$. The decision about the diameter is done in advance depending on the total laying length and the total electrical resistance.
(3) *Combining of cable ropes*:
  i. Without unused capacity
  $N/2 + N/2 \rightarrow N$; $N/2 + N/4 + N/4 \rightarrow N$; $N/4 + N/4 + N/4 + N/4 \rightarrow N$
  ii. With unused capacities (max. $N/4$ unused)
  $N/2 + N/4 \rightarrow N$ ($N/4$ unused); $N/4 + N/4 + N/4 \rightarrow N$ ($N/4$ unused)
(4) *Parallel laying of cable ropes within one connection*: Parallel laying is only allowed if combining of the ropes is not possible.
(5) *Cable branching*: If there is a cable with a capacity of N and an unused capacity of $N/4$ then it is possible to connect a cable with $N/4$ to it – the resulting capacity is then N.
(6) *Maximum length of cables*: The total length of the cable between CCP and the subscriber has to satisfy the following constraint: $d \Leftarrow \min\{13/a, 1040/r\}$ where d is the length of the cable, a is the transmission loss and r the loop resistance.
(7) *Necessity for the installation of an UGR*: UGR are built for the combining of cables, first, if only a DIP has to be connected, an UGR is not necessary or, second, if angular variation is more than 10° for cable laying then an UGR is built.
(8) *Types of UGRs*: The type depends on the highest value N of the cable's capacity and the cable's diameter Q.
(9) *DIP capacity*: It is given as $N = 7 \times 2^n$.

(10) *Types of tubes*: Tubes are used for the laying of cables. The type (size) of the tubes depends on the capacity and the diameter of the cables.
(11) *Cables laying*: The laying of cables is done in tubes and depends on the combination of different types of tubes used.

6 COSTS

Since the costs are strongly case dependent, *i.e.*, they differ, for example, from one country to another and from one soil class to another, a data base application was written in order to make the optimisation tools case- and conditions-independent. In the following some of the costs will be outlined:

(1) *Cables*:
  – The costs depend on the capacity N and the diameter Q of the cables.
  – The cost of a single pair of wires is lower for cables with higher capacities N;
(2) *Laying cables*:
  – From the size of the cables (capacity) the tubes for the laying are determined. The result is a combination of different types of tubes (for example: 3-times type_1 + 0-times type_2 + 1-times type_3). The combination determines the laying costs.
  – The laying costs increase in proportion to the number of tube types used.
  – The laying costs increase in proportion to the number of tubes used.
  – The installation of two tubes from the same type in one connection is less expensive than the installation of one tube's type in two connections. (for example: 1-times_type_1 → 6.50$/m and 2-times_type_1 → 8.00$/m).
(3) *Installation of UGR*:
  – If the installation of an UGR is necessary then the type of UGR (and its costs) is determined by the maximum value of the capacity N and the diameter Q of the corresponding cables.
  – The size of an UGR increases with the cable capacity.
(4) *Combining cables*:
  – The combining costs depend on the cable capacities.
  – If there are several sequential steps of combining, then the costs have simply to be added.

7 THE GENETIC ALGORITHMS TOOL (EGONET)

The terminology concerning *genetic algorithms* differs slightly in the literature. The following conditions are generally accepted.

A population of individuals has to exist. All individuals are strings based on an alphabet. The individuals are different. There exist some genetic operators which are applied on the individuals to cause some changes to their structures. A function exists which defines a fitness for each individual. After some changes, a re-orientation (reproduction) within the population of individuals occurs (Holland, 1975; Goldberg, 1988; Tout *et al.*, 1995). The genetic algorithm can be described as follows:

(1) Choose an initial population P with N individuals and define P' as an empty set;
(2) Calculate the fitness for all individuals of the population P;

(3) Carry out the following operations:
   – Recombination (crossing over) with probability $p(C)$,
   – Mutation with probability $p(M)$,
   – Reproduction with probability $p(R)$;
(4) Add the new and selected individuals to the new population P';
(5) If the number of individuals is smaller than N continue at (3), else at (6);
(6) The new individuals represent a new population P'; check the conditions for a termination, if these conditions are not fulfilled then $P := P'$ and continue at (2) and consider P' as an empty set;
(7) Select the individual with the highest value of fitness as a solution to the problem.

The size of the population is normally chosen between 50 and few hundreds of individuals. The recombination (or crossover) rate should be higher than 0.5 (mostly 0.6). The mutation rate should be normally small, *e.g.*, $p(M) \leq 1/N$.

Genetic algorithms are suitable for solving a variety of optimisation problems provided that the optimisation problem can be parameterised and a fitness function can be defined. Other heuristic methods applicable to the optimisation problem can be found in Dueck and Scheuer (1990), Dueck (1993), Kirkpatrick *et al.* (1983), Schwefel (1987) and Kershenbaum (1993).

7.1 Implementation

EGONET is a Genetic Algorithm that can be used as an add-on to all CAD systems since it is independent from the operating system used. Special tools for a proper CAD design of the distribution points (DIPs), the underground rooms (UGRs) and the cable connections (including their capacities) have been implemented into the CAD-system. With the help of these tools it is possible to design a P-NW on the background of a scanned bitmap that contains all geographical information such as the position of houses, streets, sidewalks, etc. The calculation of the O-NW can be called directly from the CAD system. The O-NW is visualized on the bitmap and can be plotted either separately or together with the geographical background (the bitmap representing houses, streets, etc.) using standard CAD functions. For EGONET a graphical interface for Windows (EGOWIN) was developed.

EGONET gets the information about a given network in the form of a DXF file (created with AutoCAD). This includes all the information about the DIPs, UGRs, the CCP, and all possible connections in the network (see Tab. I).

Using this format the names of the layers must be correct, otherwise, the program will not be able to extract the data from the input file.

The end point of each connection must be a central point of CCP, DIP or UGR. In order to find the correct interpretation, a difference of maximal 0.5 m between a node and the end point of a connection is allowed.

TABLE I Layers Structure of the DXF Files.

Name of layer	*Nodes*	*Representation of node*
CCP	CCP	Circles with its central point
DIP	DIPs	Circles with its central point
UGR	UGRs	Rectangle with its central point (point of intersection of diagonals)
LINES	Connections	Line with two nodes

7.2 Representation of the Data

Adjacency Matrix

Since the original telecommunication network is represented as an undirected graph it can be represented as the upper half of an *adjacency matrix*. Several operations on the adjacency representation are known in the literature (Michalewicz, 1994; Davis, 1991) and are being implemented in the developed tools.

The matrix elements are 0 (no connection) and 1 (connection) resulting in 1 bit for each matrix element. A further reduction in the number of matrix elements is possible if we assume that the DIPs are not connected to each other, an assumption which corresponds to the reality. This results in a reduction of $(d^2 - d)/2$ matrix elements, where d stands for the number of DIPs. For 150 DIPs this yields a reduction of 11,175 Bits = 1397 Bytes.

Adjacency List

Each list begins with the starting node followed by all connected nodes:

$$
\begin{array}{llllll}
\mathrm{DIP}_1, & \mathrm{UGR}_a, & \mathrm{UGR}_b, \ldots & \mathrm{UGR}_c, & \mathrm{CCP} \\
\mathrm{DIP}_2, & \mathrm{UGR}_i, & \mathrm{UGR}_j, \ldots & \mathrm{UGR}_k, & \mathrm{CCP} \\
\vdots & \vdots & \vdots \quad \vdots & \vdots & \vdots \\
\mathrm{DIP}_n, & \mathrm{UGR}_x, & \mathrm{UGR}_y, \ldots & \mathrm{UGR}_z, & \mathrm{CCP}
\end{array}
$$

Each line within such a list corresponds to a path from a DIP_i (Distribution Point i) to the CCP (Cross Connection Point) of the telephone network. The size of the list depends on the number of nodes N and the number of connections per node.

All DIPs and UGRs are represented by numbers specifying their distance to the CCP, which results in a simple data file:

$$\mathrm{CCP}_0,\ \mathrm{DIP}_1,\ \mathrm{DIP}_2,\ \ldots,\ \mathrm{DIP}_n,\ \ldots,\ \mathrm{UGR}_n,\ \mathrm{UGR}_{n+1},\ \ldots,\ \mathrm{UGR}_{n+m}$$

The adjacency list may be represented either by a list of lists of integers or as nodes connected by pointers. The representation we have used is a list where the number of nodes are given as short value and where the list is terminated by −1, *i.e.*, the data are structured as follows:

```
typedef short *tPath       for a single list
typedef tPath *AdjList     for the adjacency list
```

This data representation scheme is being used to store the telephone network data in EGONET.

7.3 Path Generation

There are different ways to generate a path:

Directed

The method starts with a DIP. The following node is the one with the lowest number (this corresponds to the shortest distance to the CCP). Each node is used only once. Dead-end paths are cancelled. A path is created when the CCP is reached.

THE ALGORITHM

```
(1) Start with DIPi;
(2) Find all possible connections excluding nodes already exist-
    ing in the path (eliminate cycles);
(3) IF the CCP corresponds to one of the possibilities THEN GOTO
    (7);
(4) IF the set of possibilities is empty THEN mark this point as
    ''visited'' AND GO one node back within the path AND GOTO (2);
(5) Select the node with the lowest number as next node;
(6) GOTO (2);
(7) Endpoint is the CCP.
```

Statistical

This is the method being used in the EGONET. The method starts with a DIP. The following node is selected randomly from the neighbouring nodes. Dead-end paths are cancelled. A path is created if the CCP is reached.

THE ALGORITHM

```
(1) Start with DIPi;
(2) Find all possible connections excluding nodes already exist-
    ing in the path (eliminate cycles);
(3) IF the CCP corresponds to one of the possibilities THEN GOTO
    (7);
(4) IF the set of possibilities is empty THEN mark this point as
    ''visited'' AND GO one node back within the path AND GOTO (2);
(5) Select randomly one of the possibilities as next node;
(6) GOTO (2);
(7) Endpoint is the CCP.
```

Statistical Weighted

The method starts with a DIP. All neighbouring nodes are identified and weighted according to their distance to the CCP. The nodes weights are then summed up and a Roulette wheel-like method is used to select randomly a node. This guarantees that nodes with lower distances to the CCP have higher chances to be selected. A path is created if the CCP is reached.

THE ALGORITHM

```
(1) Start with DIPi;
(2) Find all possible connections excluding nodes already exist-
    ing in the path (eliminate cycles);
(3) IF the CCP corresponds to one of the possibilities THEN GOTO (7);
(4) IF the set of possibilities is empty THEN mark this point as
    ''visited'' AND GO one node back within the path AND GOTO (2);
(5) Select using roulette wheel-like method one of the possibi-
    lities as next node;
(6) GOTO (2);
(7) Endpoint is the CCP.
```

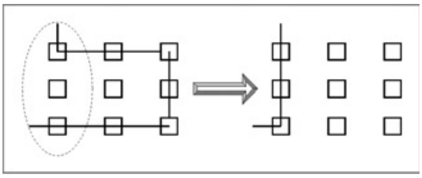

FIGURE 6 Local mutation type 1.

8 GENETIC OPERATORS

8.1 Population

This is a set of individuals. An individual is a complete network with all its paths. Each path corresponds to one DIP.

8.2 Mutation

Simple mutations such as accidental changes of numbers within a path are not reasonable since in most cases the results are not useful. For example, if in the path ... $\mathtt{UGR}_i$-$\mathtt{UGR}_j$-$\mathtt{UGR}_k$- ... the number of $\mathtt{UGR}_j$ changes, this only can be done if the new UGR is connected with both *i* and *k*. Therefore it is necessary to introduce so-called *intelligent* mutations. The followings are different mutation techniques used within EGONET:

Local Mutation

Parts of the path under consideration will be examined for possible changes in the UGRs sequences and if so will be modified accordingly.

Consider the following path: $\mathrm{DIP}_1, \ldots, \mathrm{UGR}_i, \ldots, \mathrm{UGR}_k, \ldots, \mathrm{UGR}_j, \ldots, \mathrm{CCP}$

- If a possible connection exists between UGR_i and UGR_j then connect the two nodes directly (see Fig. 6).
- If a UGR_m exists that can be connected to both UGR_i and UGR_j then replace the sub-path ..., UGR_k, ... with UGR_m (see Fig. 7).

Pattern Matching Mutations

Because it is possible, even quite likely, that certain UGRs are present in more than one path, it is possible to cross these paths at these nodes. In other word, this mutation operator is able to swap sub-paths between different paths. Consider the following paths:

$$\text{P1: } \mathrm{DIP}_1, \ldots, \mathrm{UGR}_4, \ldots, \mathrm{UGR}_7, \ldots, \mathrm{CCP}$$

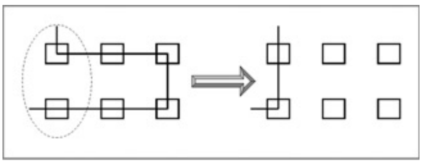

FIGURE 7 Local mutation type 2.

and

$$\text{P2: } \text{DIP}_2, \ldots, \text{UGR}_7, \ldots, \text{UGR}_4, \ldots, \text{CCP}$$

It is possible to cross these paths at the locations UGR_4 and UGR_7. Suppose we want to cross the part after the UGR_7 from P1 into P2. To do this straightforwardly is not advised as the resulting path would look like this:

$$\text{DIP}_1, \ldots, \text{UGR}_4, \ldots, \text{UGR}_7, \ldots, \text{UGR}_4, \ldots, \text{CCP}$$

This path would be *illegal* since it has the cycle on node UGR_4. To obtain a *legal* path the string will be checked for such double occurrences and if it finds one, it can safely delete the sub-path in between including the second occurrence of the node ($\ldots$, $UGR_7, \ldots$, UGR_4).

New Paths Generation Mutation

An individual is randomly selected and all its paths are regenerated. The probability for the mutation (*Mutationprob*) is considered by default to be equal to 0.025.

(1) Select a path i from the old network;
(2) Select a random number r;
(3) IF $r \Leftarrow$ *Mutationprob* THEN generate a new path i with all its properties and incorporate it within the new network;
(4) IF not all paths in the old network are done GO TO (1);
(5) Ready.

This mutation should be used with some care since good structures may be destroyed; however, in many cases new good individual paths may be introduced. This mutation has proved to be very useful and has improved the behaviour of the GA considerably.

8.3 Crossover (Recombination)

Two parent individuals (father and mother) with high fitness values are first selected. One path is selected randomly from the father network with the probability OVCPOP and a second path is likewise selected from the mother but with a probability of (1.0 − OVCPOP).

In a statistically selected (child) individual, which is characterised by a low fitness value, the corresponding paths are substituted by the paths from the parent networks. The population is then sorted and selected by the fitness of its individuals.

SELECTION
(1) Choose a father network with high fitness from the population;
(2) Choose a father network with high fitness from the population;
(3) Choose a child network with low fitness from the population;
(4) Crossover the paths from the father and mother networks with the corresponding paths from the child network (*see Crossover algorithm*);
(5) Calculate the fitness for the child network;
(6) Sort the population according to the fitness values;
(7) IF not the end GO TO (1)
(8) Ready.

CROSSOVER (RECOMBINATION)
(1) Select a path from the father network;
(2) Select a path from the mother network;

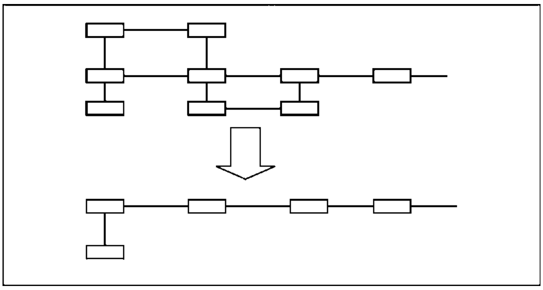

FIGURE 8 Repairing the network.

(3) Select a random number r;
(4) IF $r \Leftarrow$ OVCPOP then substitute the path i in the child network with the one from the father network else substitute it with the path from the mother network;
(5) Repair the child network;
(6) IF not all paths in parents networks are done GO TO (1);
(7) Ready.

As mentioned above a repair procedure for the resulting network is used which primarily eliminates the cycles within the paths of the network (see Fig. 8):

8.4 Fitness

The fitness function works in two phases. First the function determines the lengths and the capacities of all edges needed for the network. An edge is a connection between two nodes, *e.g.* if an edge is member of 3 paths in a network, then this edge has a capacity of $3 \times 7 = 21$ wires (7, because until now, each DIP has this capacity). At the same time the number of UGRs in the network and the lengths of all edges are calculated.

In the second phase the costs of all edges necessary for the description of the network are added to the total cost. Now each edge is to be divided into several cables, starting with the cable with the highest fitting capacity (see Tab. II). If, for example, an edge is member of 5 paths in a network, this edge has a capacity of $5 \times 7 = 35$ wires. This capacity is divided into

TABLE II Necessary Divisions Between Two Cables.

	Min						
Max	*7*	*14*	*28*	*56*	*112*	*224*	*448*
7	–						
14	KT14	–					
28	KT28	KT28	–				
56	KT56, KT14	KT56	KT56	–			
112	KT112, KT28	KT112, KT28	KT112	KT112	–		
224	KT224, KT56, KT14	KT224, KT56	KT224, KT56	KT224	KT224	–	
448	KT448, KT112, KT28	KT448, KT112, KT28	KT448, KT112	KT448, KT112	KT448	KT448	–

2 cables. One cable with a capacity of 28 wires and another cable with a capacity of 7 wires since $28 + 7 = 35$ wires. Knowing the length of the edge from the first phase, the cost of the edge is calculated as the sum of the costs of both cables. Furthermore the costs of laying the cables and installing the UGRs are then added.

The costs for cables divisions in UGRs are negligibly low so that we have decided against considering them in our fitness calculation. This decision does not affect the goodness of the resulting networks but spares us some extra (unneeded) calculations which reduces the total computation time.

9 CONCLUSION

For our standard problem depicted above in Figure 4, EGONET has found during several tests, networks with a fitness of approximately US $30,000. This value is only an upper bound of the real costs, which are approximately US $25,000. The difference results from the fact that the UGRs were calculated using an average cost value of US $200 and not the real cost. With real UGRs costs being used, the model has succeeded in finding networks with respectively US $29,000 and even US $30,000 fitness values.

EGONET has a reliable convergence rate. It always succeeds in finding a good network with a relatively small number of iterations and recombinations.

With EGONET we are confident that we have developed an application and a general approach to solving the problem of network layout design. In contrast with classical methods of optimisation, our method can be extended to the problem of finding a network layout in *full* detail. The starting point of our work was the optimisation of telecommunication networks, but along the road this was extended to optimising the layout of networks different in detail and possibly more complex. These domains currently include Gas, Electricity and Cable TV. The constraint editor built for defining and analysing network layout specifications in a generic – knowledge based – approach to solving similar problems with different constraints in the shortest amount of time possible. At the end of our work, the network design application is heavily parameterised by the knowledge of the network design engineer. Examinations of the optimisation speed have revealed that as few as five runs with 10,000 networks generated per run could find a near-optimal network.

References

Davis, L. (1991). *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York.

Dueck, G. (1993). New optimization heuristics – the great deluge algorithm and the record-by-record travel. *Journal of Computational Physics*, **104**, 86–92.

Dueck, G. and Scheuer, T. (1990). Threshold accepting: A general purpose optimization algorithm superior to simulated annealing. *Journal of Computational Physics*, **90**, 161–175.

Goldberg, D. E. (1988). *Genetic Algorithms in Search, Optimisation & Machine Learning*. Addison-Wesley.

Holland, J. H. (1975). *Adaptation in Natural and Artificial System*, MIT Press.

Kershenbaum, A. (1993). *Telecommunication Network Design Algorithms.* McGraw Hill Publishers, N.Y.

Kirkpatrick, S., Gelatt, C. D. and Veechi, M. P. (1983). Optimization by simulated annealing. *Science*, **220**, 671–680.

Michalewicz, Z. (1994). *Genetic Algorithms + Data Structures = Evolution Programs*, 2nd extended ed., Springer Verlag.

Schwefel, H. P. (1987). Collective phenomena in evolutionary systems. Reprints of the *Annual Meeting of Int. Soc. for General Systems Research*, Vol. 2. Budapest, pp. 1025–1033.

Tout, K., Ribeiro-Filho, J. and Mignot, B. (1995). A cross-platform parallel genetic algorithms programming environment. *Parallel Algorithms and Applications*, Vol. 7, pp. 255–266.