International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol. 9, Issue.7, July 2021 E-ISSN: 2347-2693

Comparative Study between Greedy and Genetic Algorithms in Optimal Cellular Masts Hoisting Over Population Coverage

T.A. Enisuoh[1*], O.R. Okonkwo[2], N.N. Mbeledogu[3]

[1]Department of Computer Science and Informatics, Federal University Otuoke, P.M.B 126 Yenagoa, Bayelsa State, Nigeria
[2,3]Department of Computer Science, Nnamdi Azikiwe University, Awka, Anambra State, Nigeria

[*]Correspondence Author: tam4truth@yahoo.com, Tel,: +234 (0)8037767769

DOI: https://doi.org/10.26438/ijcse/v9i7.2634 | Available online at: www.ijcseonline.org

Received: 02/Jul/2021, Accepted: 19/Jul/2021, Published: 31/Jul/2021

Abstract- The need to meet the demands of subscribers of wireless services is imperative to Global System for Mobile (GSM) Communication companies. These demands which evolves round maintenance of good network coverage, reduction in service costs and improved Quality of Service (QoS) depends largely on the nature of cellular network masts hoisting. The existing optimization model however, do not handle hoisting of cellular network masts for small size populated areas effectively as a result of insufficient solution space and thus, suffers from poor network coverage. To overcome this challenge, Greedy Algorithm model was proposed. Object Oriented Analysis and Design Methodology (OOADM) was deployed and the proposed model was developed using Java programming language. A comparative analysis between the developed model and Genetic Algorithm was carried out with an aim of determining the best optimization technique for small size populated areas. The statistical results were significant in all the tests performed. The analysis of the results shows that Greedy Algorithm performed better than Genetic Algorithm in optimizing cellular network masts hoisting for small populated areas. Therefore, with this model, optimization of cellular network mast hoisting for small size populated area will be more accurate and hence, more reliable.

Keywords- Greedy Algorithm (GR) ; Genetic Algorithm (GA) ; Mast; Optimization; Coverage.

I. INTRODUCTION

The way cellular telecommunication is growing speedily from previous years is as a result of new emerging technologies been introduced into the market. Cellular network telecommunications normally have coverage over vast service area that is been partitioned into smaller regions known as sites/base station [1]. Due to the speedy growing rate of cellular network, alot of proliferation of cellular masts have been introduced over a particular area with the aim of attracting many subscribers in order to make more profit [2]. Having optimal coverage means getting the best from something so as to take the best decision. Optimization has really gotten the attention of many researchers in recent years as a result of the speedy evolution in the field of computer science with the aid of user-friendly software and high speed processors. The word optimization is very important in making any decision in life by different professionals in different fields. In addition, optimization can be applied to issues of cellular masts location with respect to population coverage and service quality delivery to GSM users by the telecommunication firms [3]. Hence, telecommunication network optimization becomes an important need for telecommunication firms to do their business effectively to attract more customers to make more profits. More so, [4] viewed mast as a location/place where antennas and electronic gazettes are designed to form a cell in a cellular network. According to [4], he noted that based on the technology masts operator use; a mobile operator can hoist many base stations, each having its own functions in that location. The masts are in different shapes, forms, and sizes and are seen as big towers [5].

II. RELATED WORK

Also, [6] presented a particular issue on Discrete Multiple Facility Location Problem (DMFLP) where multiple facilities of various types are located and assigned to users. Here, some facilities are located in order to service users around an area optimally in which locations and requirements are known. When sites are chosen from a given area, the corresponding/subsequent location problem turns to a Discrete Multiple Facility Location Problem. This issue considered depends on interactions if new facilities were present or not. Thereafter, new heuristic solution pattern, branch and bound algorithms were proposed. Then results obtained from data that are randomly generated are compared for optimal solutions, it showed that the proposed method was accurate and efficient.

However, [7] presented an all-purpose bi-level simulated annealing algorithm (BSA) for facility location problem to determine the demands of customers under the condition of minimum cost. This is depended on the idea and character of standard simulated annealing algorithm. From the researcher's findings, in solving the problem, the BSA was divided into two layers namely inner and outer layers respectively. As a result, the outer algorithm is optimized and the inner algorithm is optimized for allocation of customer's demand under the given decision of the outer algorithm.

More so, [8] presented a simple implementable algorithm which effectively determines the strategic positions of the cell towers. In addition, the algorithm helps to find out the optimal height of the tower at a chosen potential tower location. The approach applies a three stage algorithm that ensures that out of many potential tower locations, only the indispensible and optimal locations can be chosen.

III. METHODOLOGY

The researcher will develop the system using object-oriented analysis design methodology. Object-oriented Analysis Design Methodology (OOADM) is a software engineering approach that models a system as a group of interacting objects. Each object represents a component of the system being modelled and is typically characterized by its state (data elements), and its behaviour. The behaviour of the system results from the collaboration between these objects.

Greedy Algorithm

Greedy algorithm (GR) is known to be a problem solving algorithm that follows finite steps. With a given number of steps, it can be applied mathematically in solving optimization problems. To perform a given set of choice, the steps involve the repetition of the same operation. GR make choices that can be the best at that particular time with the hope that it will result to the global optimum solution unlike other algorithms that targets a single global optimal solution.

For example, if you have an objective function that needs to be optimized (either maximum or minimum) at a given point. A Greedy algorithm makes greedy choices at each step to ensure that the objective function is optimized. The GR algorithm has only one shot to compute the optimal solution so that it never goes back and reverse the decision.

In general, greedy algorithm has the following five components:

1. **Candidate Set:** This is where a solution is created.
2. **Selection Function:** This chooses the best candidate to be added to the solution.
3. **Feasibility Function:** This determines if a candidate can be used to contribute to a solution.
4. **Objective Function:** This assigns a value to a solution, or a partial solution.
5. **Solution Function:** This will indicate when we have discovered a complete solution.

Data Flow Diagram of Greedy Algorithm

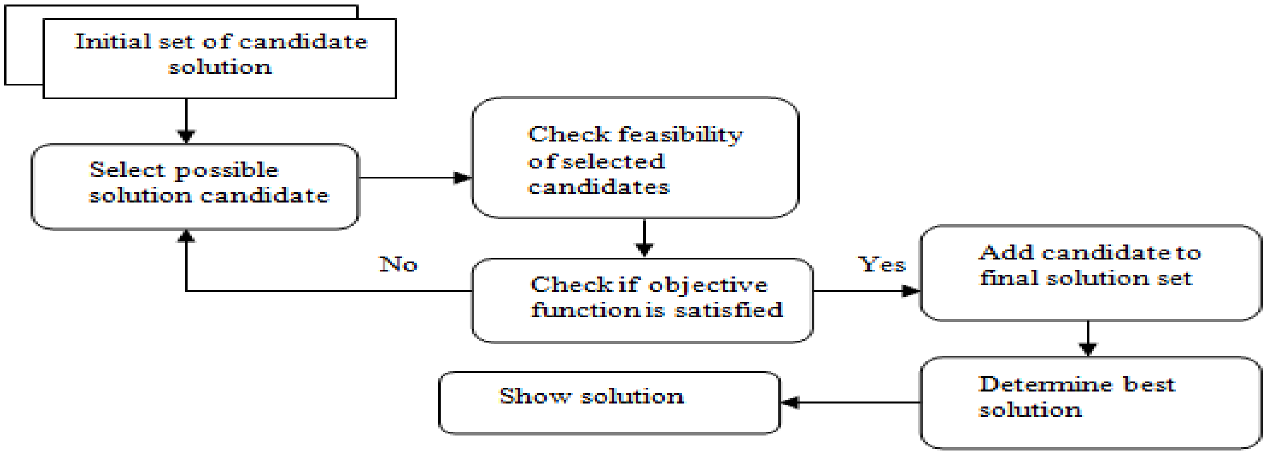

Figure 1 Data Flow Diagram of the Greedy Algorithm

Mathematical Model of Greedy Algorithm

The mathematical model for the greedy algorithm is as follows:

$S_w(F) = \{e \in E \setminus F \mid w(e) > 0, F \cup e \in F\}$ $(F \in F)$.

Where S is the initial set of elements and F is the final output containing the set of optimized elements.

The Pseudo-code of Greedy Algorithm

1. Initialize: F = null;
2. Iterate: While S_w (F) ≠ null:
   2.1 Choose $e \in S_w$ (F) of maximal possible weight w(e);
   2.2 Update F = (F ∪ e);
   3. Output F.

Genetic Algorithm

Genetic Algorithm (GA) is a pattern used to carry out random searching and optimization which has its application in many fields to solve optimization issues such as telecommunications industries, mostly, problems that have not been properly or well designed, it also relates with possible solutions of larger number. GA works with possible solutions that have been randomly generated in a given population. A chromosome is known to be an individual solution in the population. Also, a single string or gene can stand in for a chromosome which entails a segment or section of the given solution. The chromosomes should all have the same length in a given population.

Furthermore, a fitness function is made available to give the fitness value for all individuals that relies on the closeness of an individual towards an optimal solution, the larger the fitness value, the nearer that solution tends to optimality. Two parents chromosomes that have been randomly chosen can exchange information that are genetic in nature- this process is called crossover, in order to give birth to two new chromosomes called child or offspring. If the two parents earlier stated share a given pattern or style in their offspring in order to get better solution, mutation is used randomly on the selected chromosomes after the crossover process has been carried out.

After the crossover and mutation processes have taken place in the population, the next generation chromosomes are chosen. To make sure that this new generation of chromosome is as good as the last generation, the individuals that performed badly in the present generation can be exchanged by same number of individuals that performed very well from the last generation. This process continues till an ending condition of the algorithm is met as shown in the algorithm below.

Genetic Algorithm according [9]

```
 Int Main()
{
   Generate initial population
   Calculate the fitness of each individual
   while (not stopping criterion)
   {
   Select parents from population.
   Carry out crossover process to produce new offsprings.
   Carry out mutation process.
   Calculate fitness of each individual.
   Replace the parents by the corresponding offsprings in
   new generation.
   }
}
```

Data Flow Diagram of Genetic Algorithm

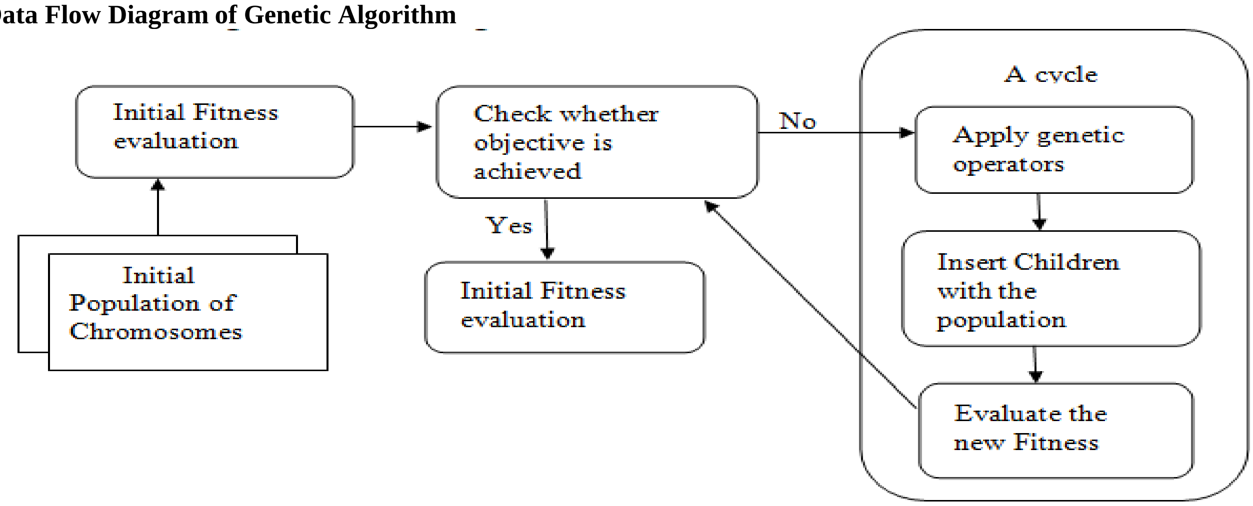

Figure 2 Data Flow Diagram of the Genetic Algorithm

- **Population Initialization:** The population here refers to chromosomes or individuals; the initial population is comprised of genes that define if an individual is right to be known as a candidate solution when optimization issues are looked at. The first population size is dependent on a given set of site requirement that encompass the number of masts needed to give enough coverage. The larger the first population the larger the time required to run GA because the GA gives most of its time on the population been evaluated.
- **Evaluation of Individuals:** This involves evaluating the fitness of an individual (chromosome), with the use of mathematical function that has been generated depending on the optimization objective. The fitness function is the backbone for having a suitable GA to solve problems that involve optimization.
- **Selection:** This process is done to show the survival of the fittest in the working principles of GA, each chromosome or individual is assigned a fitness value. Based on this, parents are chosen for crossover in which the system selects the best chromosomes that fits in the given population while keeping the diversity of the population intact.
- **Crossover:** It involves linking two chromosomes with the involvement of genes that give rise to new chromosomes having the characteristics of both parents.
- **Mutation:** This process is done to explain a genetic process conducted so as to allow adjustment in the given population which permits GA to look for all available search space and get a more optimal solution. For binary GA, it requires randomly changing 0 to 1 and vice versa. Mutation probability is advice to be low to the tune of 0.1% to avoid premature convergence and at most 1% as the case may be.

Mathematical Model of Genetic Algorithm

$$\overrightarrow{S(t)} = \frac{F\overrightarrow{P(t)}}{\sum_{j=0}^{2^l-1} F_{jj} P_j(t)}$$

Where $\overrightarrow{S(t)}$ is a column vector representing the probability of selecting a parent.
$\overrightarrow{P(t)}$ is the proportion of the population at generation t.
F is a two dimensional matrix with the property that
1. $F_{ij} = 0 \;\; \forall\, i, j$ and
2. $F_{i,i} = f(i)$.

The properties combined together means that every entry of F is zero except the diagonal entries $((i,i))$ which gives the fitness of the corresponding string i under proportional selection.
l is the length of a string.

Note:
1. This is an existing model for which you can easily find S(t) given F and P(t) or P(t) given F and S(t).
2. The population size is n.

Class Diagram of Greedy Cellular Network Mast Hoisting

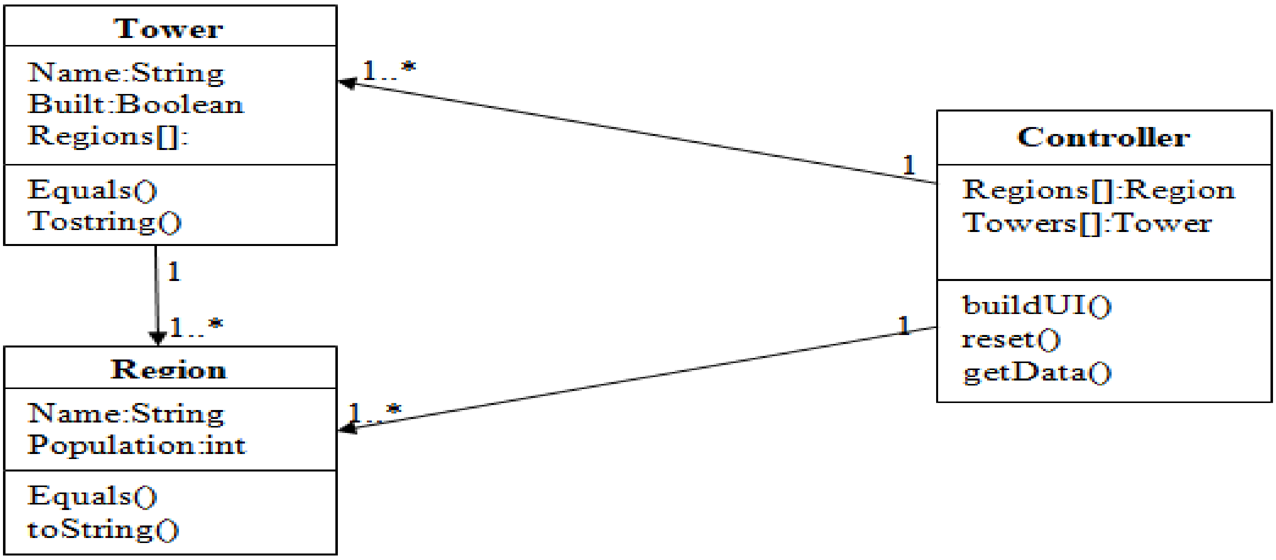

Figure 3 Class Diagram of Greedy Cellular Network Mast Hoisting

Tower class: The tower class represents a potential location for the placement of a telecom tower.
Region class: The region class represents a logical region such as towns in a local government area.
Controller class: The controller class encapsulates the various algorithms used to optimize the selection of location sites. It also handles the data input and output and the data flow between the various components of the system.

Model Diagram for Greedy Cellular Network Mast Hoisting (GCNMH)

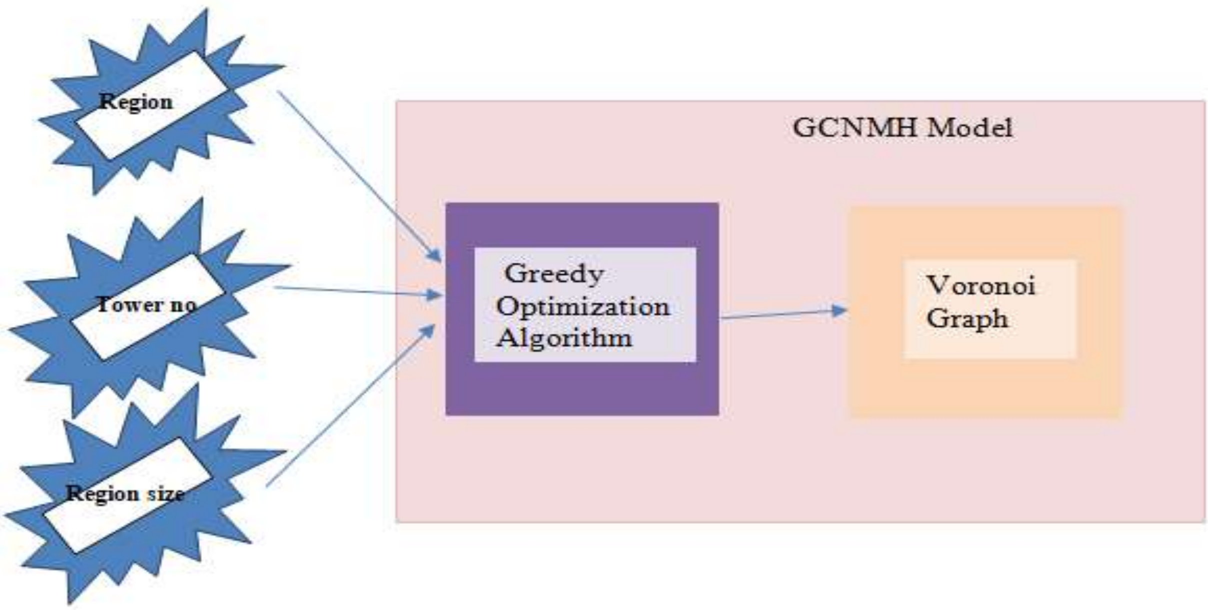

Figure 4 Greedy Cellular Network Mast Hoisting (GCNMH) model diagram.

The Program Algorithm of the Greedy Cellular Network

The following is the program algorithm in pseudo-code form.

Begin

Specify the number of regions and possible cell towers/sites to be mounted
Validate specified number of cell towers
Enter population data values
Enter tower sites
Assign region data to tower site
Apply the selected algorithm to the data
Generate aggregate population covered
Using the selected tower sites, generate voronoi diagram
Terminate the application

End

Activity Diagram of the Greedy Cellular Network Mast Hoisting

Figure 6 depicts the activity diagram of the new system which shows the flow of control with emphasis on the sequence and conditions of the flow.

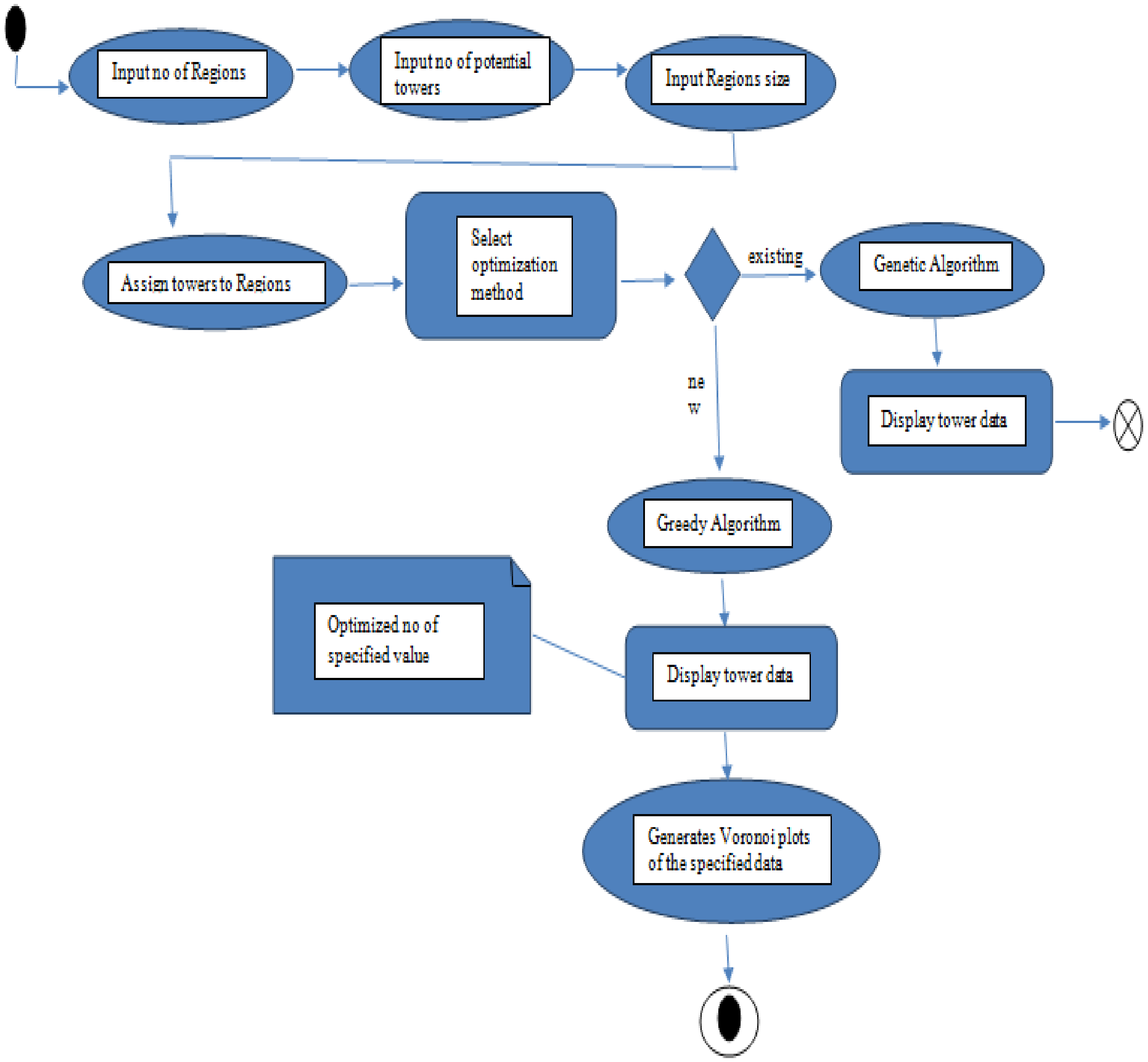

Figure 5 Activity Diagram of the Greedy Cellular Network Mast Hoisting

IV. RESULTS

Figure 6 Home screen

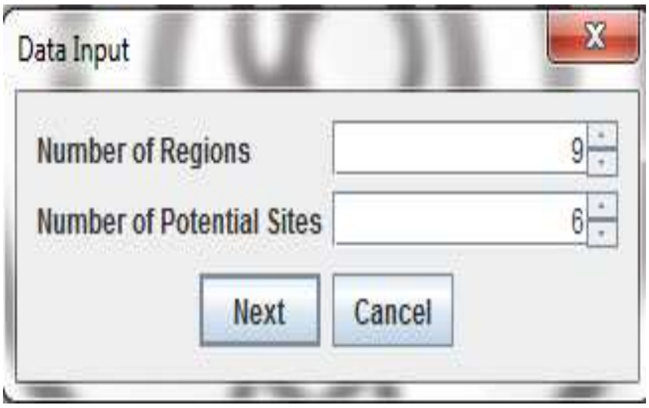

Figure 7 The data input screen

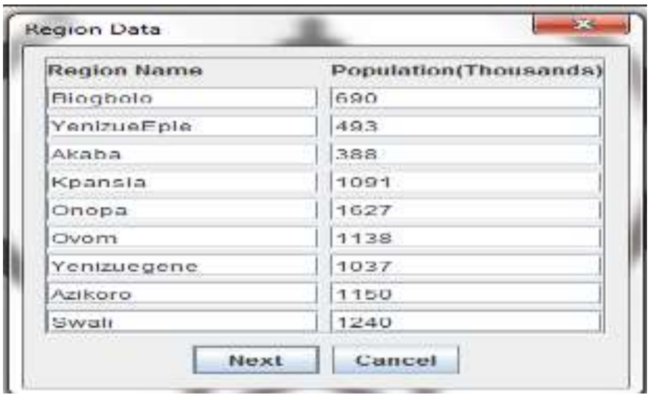

Figure 8 Tests result on region data

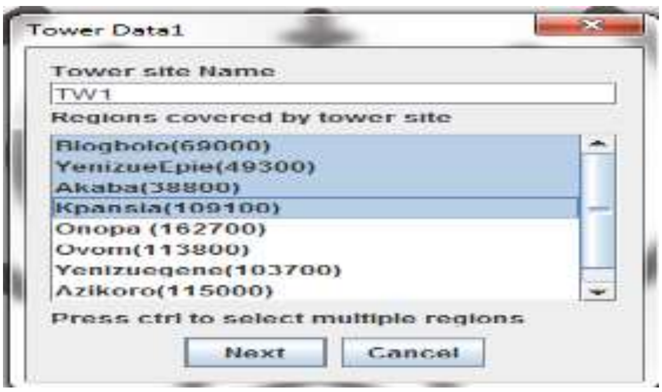

Figure 9 Tests on tower data inputted

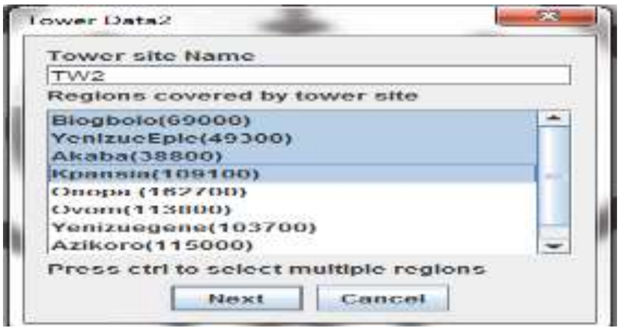

Figure 10 Tests on tower data inputted

Figure 11 Tests on tower data inputted

Figure 12 Tests on tower data inputted

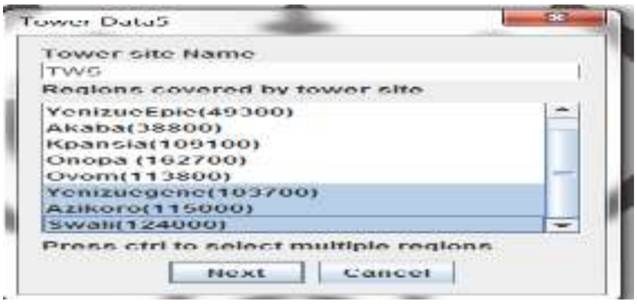

Figure 13 Tests on tower data inputted

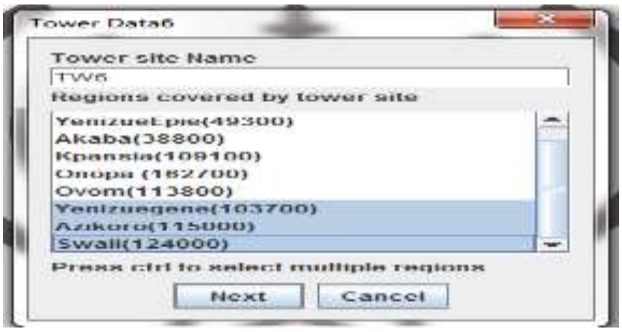

Figure 14 Tests on tower data inputted

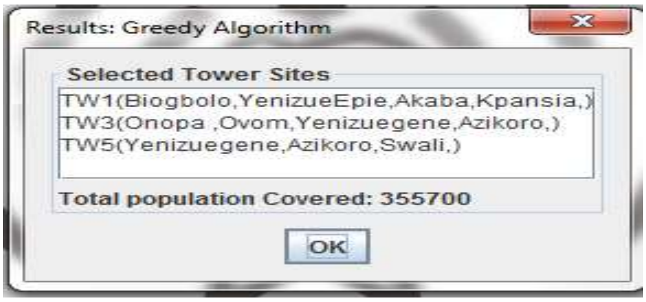

Figure 15 Results of Greedy Algorithm

Figure 16 Results of Genetic Algorithm

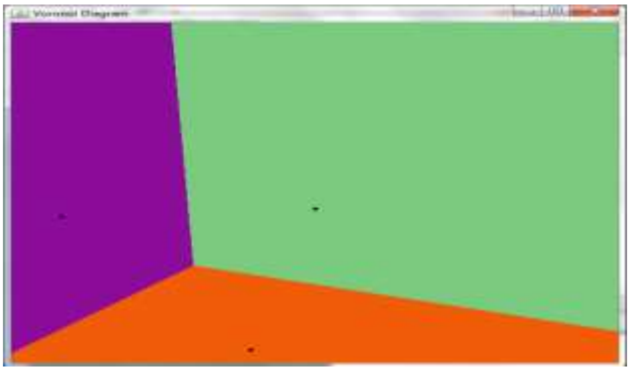
Figure 17 The voronoi output on coverage area of tower cells.

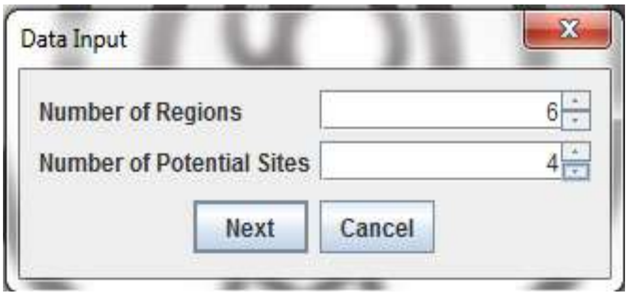

Figure 18 The data input screen

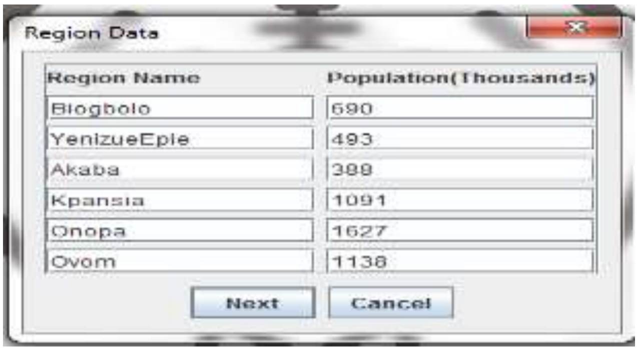

Figure 19 Tests result on region data

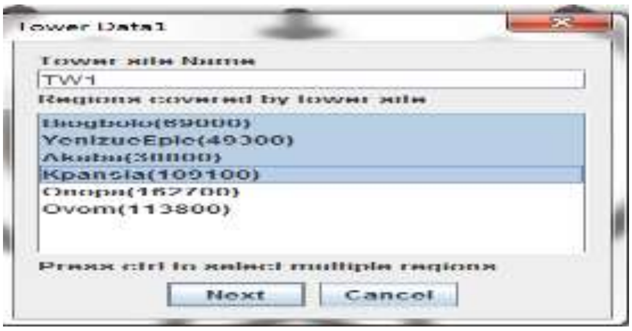

Figure 20 Tests on tower data inputted

Figure 21 Tests on tower data inputted

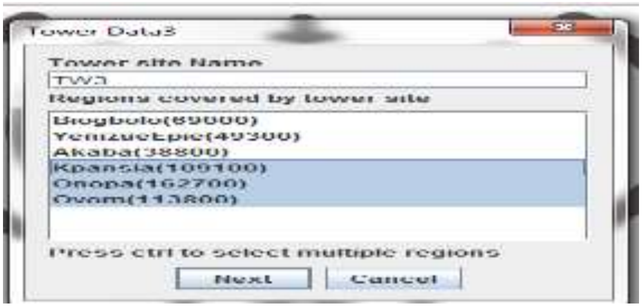

Figure 22 Tests on tower data inputted

Figure 23 Tests on tower data inputted

Figure 24 Results of Greedy Algorithm

Figure 25 Results of Genetic Algorithm

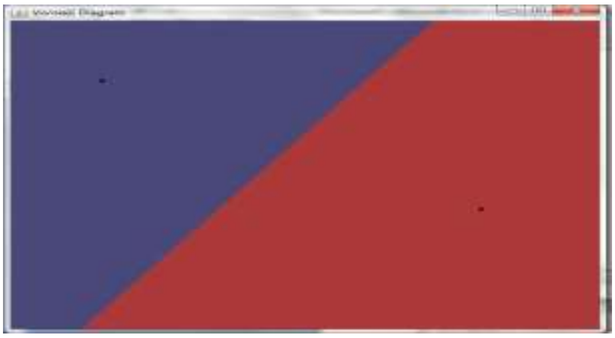

Figure 26: The voronoi output on coverage area of tower cells.

Result

When the Greedy and Genetic algorithms were compared to know which one is better in terms of optimization technique on coverage of cellular network masts hoisting for small size population, it was discovered that the Greedy Algorithm performed better than the Genetic algorithm with the statistical results obtained. Hence, with this model, optimization of cellular network mast hoisting for small size populated area will be more accurate and hence, more reliable.

Discussions

Figure 6: This Home screen shows immediately the program is run to enable other modules under file and optimize menus to be activated.

Figures 7 and 18: Are two different tests on region data generated with respect to the population size for each region.

Figures 8 and 19: Are two different tests on data input carried out to specify the numbers of regions the mast is expected to cover and the numbers of potential sites meant for those regions.

Figures 9 to 14 and 20 to 23: Are two different tests on tower data inputted for each tower earlier specified and multiple regions are selected to show optimal coverage of the total population in the generated regions.

Figures 15 to 16 and 24 to 25: Are two different results showing the comparison between Greedy and Genetic algorithms

Figures 17 and 26: Are two different results on voronoi output screen that shows the coverage area of tower sits.

Table 1: Comparisons of Greedy and Genetic Algorithms for Aggregate Population Covered.

S/N	Number of regions	Number of potential tower sites	Selected sites (Greedy Algorithm)		Selected sites (Genetic Algorithm)	
			Towers selected	Aggregate Population covered	Towers selected	Aggregate Population covered
1	9	6	TW1, TW3, TW5	355,700	TW1, TW4, TW5	335,4000
2	6	4	TW1, TW3	231,700	TW1, TW4	178,100

Sample tests were carried out using various region sizes (with their human population) and possible tower sites. The results of the tests are shown in the table above.

It can be noted from the tests that, in general, the greedy algorithm provides the solution which requires the least number of tower sites with larger population coverage.

It should be noted that in these tests;

1. Some tower sites cover more than a single region so tower coverage overlaps across regions.
2. The algorithms are designed to ensure that the selection of towers covers all specified regions.
3. The population aggregate takes into consideration the region overlaps.

Application Areas

This research work can be applied in the following areas:

1. It can be applied in ad hoc mobile networking to efficiently route packets with the fewest number of hops and the shortest delay time.
2. It can also be used in Activity Selection Problems.

V. CONCLUSION

In our world today new technologies are emerging day by day and telecommunication firms are really in the fore to solve daily life problems with first hand information. Mobile cell phone applications have made life comfortable and easy for customers of the various telecommunication firms. The essence of looking for optimal hoisting of cellular phone masts problems using computer-based algorithms is to provide a better quality of service and network coverage for users of cellular phone facilities who need good service in order to carry out their day-to-day business transactions, and reduction of cost needed to provide and maintain the cellular service. In this thesis work, greedy algorithm has been considered to handle the issue of optimal hoisting of cellular phone masts. The algorithm outputs voronoi diagram and provide the basis for a computer-based solution to the problem of determining an optimal location for mast hoisting.

Suggestion for Further Research

Analyze this Selection Greedy Algorithm with other types of Greedy Algorithms in respect to optimal hoisting of cellular phone mast to know which one have the best coverage.

REFERENCES

[1] T. D'Roza, G. Bilchev, "An overview of location-based services", BT Technology Journal, **Vol. 21, pp. 20-27, 2003.**

[2] V. F. O'Flaherty, "The Legal Landscape When a Tower Collapses", Journal of the Missouri Bar, **Vol. 57, 2001.**

[3] R. Haupt, S. Haupt, "Practical Genetic Algorithm" 2nd edition, John Wiley & Sons, publication, USA, ISBN 0-417-45565-2, **2004.**

[4] Zunia, "The Knowledge Exchange", **2011.**

[5] Mastsanity, "Mast Sanity Quick Guide To Mobile Phone Masts", Registered UK Charity no: **1109757, 2008.**

[6] S. Davood, M. Mahnam, A.S. Nookabadi, "An efficient approach to discrete Multiple Different Facility Location Problem," Service Operations and Logistics and Informatics, In the proceedings of 2008 IEEE International Conference, **Vol.2, pp2519, 2524, 2008.**

[7.] Ren et al., "Bilevel Simulated Annealing Algorithm for Facility Location Problem" In the proceedings of the 2008 Information Management, Innovation Management and Industrial Engineering. ICII International Conference, **Vol.3, pp 17,22, 2008.**

[8] K. Rakesh, M.S. Bhuvan, S. Chamarti, P. Bhat, M. Jothish and K. Annappa, "Algorithmic Approach for Strategic Cell Tower Placement", In the proceedings of the 2014 Fifth International Conference on Intelligent Systems, Modelling and Simulation, **2014.**

[9] C. Ming-Tsung, T. Shian-Shyong, "A genetic algorithm for multicast routing under delay constraint in WDM network with different light splitting", Journal of Information Science and Engineering, **Vol. 21, pp 85–108, 2005.**

[10] J. Hubaux, S. Znaty: In K. Terplan, P. Morreale., "The Telecommunications Handbook", CRC Press LLC, **2000.**

[11] International Journal Of Advanced Research in Computer and Communication Engineering **Vol.2, Issue 12, 2013.**

[12] M. Michael, "Voronoi Diagrams", The Mathematics Enthusiast: **Vol 1: No2, p.44-55, 2004.**

[13] L. Snyman," Practical Mathematical Optimization: An Introduction to Basic Optimization theory and Classical and New Gradient-Based Algorithms", Springer Science & Business Media, **2005.**

[14.] Szabó Gábor Vilmos,"Optimization Problems in Mobile Communication", **2005.**

[15.] A. Okabe, B. Boots, K. Sugihara & S.N. Chiu, "Spatial Tessellations – Concepts and Applications of Voronoi Diagrams", John Wiley publisher, 2nd edition., **2000.**

