A 0-1 MODEL FOR FIRE AND EMERGENCY SERVICE FACILITY LOCATION SELECTION:A CASE STUDY IN NIGERIA

AROGUNDADE O.T., AKINWALE A.T.; ADEKOYA A.F. AND AWE OLUDARE G.

Department Of Computer Science, University Of Agriculture, P.M.B 2240, Abeokuta, Ogun State, Nigeria.

E-mail: arogundade@acm.org, atakinwale@yahoo.com, lanlenge@yahoo.com and aweolu@yahoo.com

ABSTRACT:

Facility location selection problem is a variant of set covering problem. Set covering problem is a classical problem in computer science and complexity theory. In this paper two different techniques are applied to facility location problems. First, a mathematical model of facility location is introduced and solved by using optimization solver, TORA. Secondly, the balas additive algorithm of branch and bound techniques is used to solve the facility location problem. Tests were made using real life data from a city in Nigeria. We then observed that both algorithms indicate the same number of fire stations in different locations. Also the results obtained by applying and implementing balas additive were more explanatory by specifying the names of the locations where the facilities are to be located and the names of the locations to be served by each of the facilities.

Keywords: *Set covering problem, fire station, emergency service, branch and bound, integer linear programming.*

1 INTRODUCTION

Set covering problem is a classical problem in computer science and complexity theory, and is one of the most important discrete optimization problem because it serves as a model for real world problems. Real world problems that can be modeled as set covering problem include airline crew scheduling, nurse scheduling problems, resource allocation, assembly line balancing, vehicle routing, facility location problem which is the main focus of this work. Etc. Set covering problem is a problem of covering the row of an m-row/ n-column zero-one matrix with a subset of columns at minimal cost [1]. The set cover problem is a classic NP-hard problem studied extensively in literature, and the best approximation factor achievable for it in polynomial time is Ø (log n) [2, 3, 4].

A rich literature has been developed and several models have been formulated and applied to the facility location problems over the last few years. The complexity of these problems is due to the multitudes of quantitative and qualitative factors influencing location choices. However, investigators have focused on both algorithms and formulation in diverse setting in the private sector (e.g. industrial plants, retail facilities, telecommunication mast etc) and the public sectors (e.g. schools, health centers, ambulances, clinics etc). In this work, our interest is on one of the public sector facility location problem, the fire and emergency service location problem. In fact, fire and emergency service is crucial in saving lives and valuable properties and therefore must provide high level of quality services to ensure public safety. But providing these facilities effectively is a complex issue that especially depends on some factors and most especially on the best geographical location of the fire fighting and emergencies service facilities. The aim of this paper therefore is to use a Set Covering model to select the minimum fire stations that could serve all areas in a big city in such a way that each ward will have equal benefits in terms of services from the fire stations and also the facility will be strategically placed.

The process involves gathering data about all the wards in the city using the GPS (Global Point System) so as to get their distances from each other using GIS software (Geographical Information Service). We then developed a decision support sytem that determine the minimum number of fire stations needed to serve all the wards such that the distance between each ward and at least one station is less or equal 10 kilometers by solving the mathematical model of the set covering problem using the Balas Additive algorithm a special case of branch and bound that handles binary linear programming problem. The result obtained was compared to the result obtained from TORA solver.

2 LITERATURE REVIEW

The Classical Location Set Covering Problem involves finding the smallest number of facilities and their location so that demand is covered by at least one facility. It was first introduced by [12]. The problem represent several different application setting including the location of emergency service and the application setting including the location of emergency services and the selecting of conservative sites. The problem is called covering problem in that it requires that each demand be served or "covered" within some maximum time and distance standards. A demand is defined as covered if one or more facilities are located within the maximum distance or time standards of that demand. The second type of covering problem is called the Maximal Covering Location Problem [13]. Since the development of these two juxtaposed problems were formed, there have been numerous applications and extensions.

Set Covering Problem is one of the most prominent NP-complete problem.(An exhaustive algorithm must search through all 2^m subsets of S to find those which are covering subsets and then pick the minimal from among these [4] and can formally be defined as follow:

U is the universal set, S is a collection of subsets of U, and c: S -> N is a cost function. The goal is to find a collection S_1, S_2 ...,S_K of elements of S such that S_1 U S_2 U... U S_k = U with minimal total cost.[16].

Significant research has been directed towards the problem of locating and covering problems and several methods have been made to provide solutions specifically to the facilities location problem and these methods generally involve the use of queuing models [5], simulation and mathematical programming, also a combination of simulation model and heuristic search routines [6]. Also an extensive number of papers have been dedicated to the set covering problem (SCP) and many exact algorithms [7, 10] which can solve instances with up to few hundred rows and columns. A comparison of some exact algorithms can be found in [9]. Approximation algorithms plays an important role in solving SCP, given the limitation of exact methods and the large list of applications using large size SCP[12].

Virtually every heuristic approach for solving general integer problem has been applied to set covering problems. The set covering formulation naturally lends themselves to greedy start (i.e. an approach that at every iteration myopically chooses the next best solution without regards for its implication on future moves). Interchange approaches have also been applied; here a swap of one or more column is taken whenever such a swap improves the objective function value. Newer heuristic approaches such as genetic algorithm, probabilistic search [8], simulated annealing [11] and neural network have also been tried. Unfortunately, there has not been a comparative testing across such methods to determine under what circumstances a specific method might perform best. In addition, one can embed heuristic within an exact algorithm so that one can iteratively tighten the upper bound and at the same time one is attempting to get a tight approximation to the lower bound for this problem.

Problems arising in practice do not however have perfect or ideal matrices. Nevertheless, it has been observed in computational practice that as long as the problem to be solved are relatively of medium size, linear programming with branch and bound will provide integer solution quickly and optimally. However as the sub program size increases, the non integrality of the linear programming solution increases dramatically and does the length and size branching tree. It is for this large instance of problem that approximation techniques, reformulation and exact procedures have been developed that exploit the underlying structure of the problem.

Integer Linear Programming (ILPs) are linear programs in which some or all of the variables are restricted to integer (or discrete) values. ILP has important practical application. Unfortunately, despite decades of extensive research, computational experiences with ILP have been less than satisfactory. To date there does not exist an ILP computer code that can solve integer linear problems consistently [15].

2.1 *Problem Statement*

Consider a fire station location and allocation problem having the following features:

- A fire station located in a ward has to serve a set of wards.
- Each ward to be served must be located at fixed distance to the location of the fire station.
- The minimum number of fire stations that can serve all the wards must be determined.

The mathematical model of this problem is formulated as follow.

$$\text{Min } Z = \sum C_j x_j$$
$$j=\{1,2,\ldots n\}$$

Subjected to:

$$\sum a_{ij} x_j \geq 1$$
$$i = \{1,2,\ldots m\}$$

$$x_j = \{0,1\}$$

where C_j is the cost of installation, x_i represents a covering i. x_j which can take the value 0 or 1 depending on if ward i is in covering x_j.

3 Methodology

This paper aims to obtain an optimal solution to fire and emergency facilities location problem. We use the GPS (Global Point System) equipment to get the coordinates of all the wards in the city under consideration. From the screen of the equipment, we got the North-axis and the East-axis of every particular place we visited (37 wards). After the collection of the coordinates, we installed the GIS (Geographical Information System) software for analysis. We then supply the coordinates of each ward into the GIS which then locate the position of the wards on the map of Ogun state (see **figure 1**) and there after obtained the distance readings for each ward to the other.

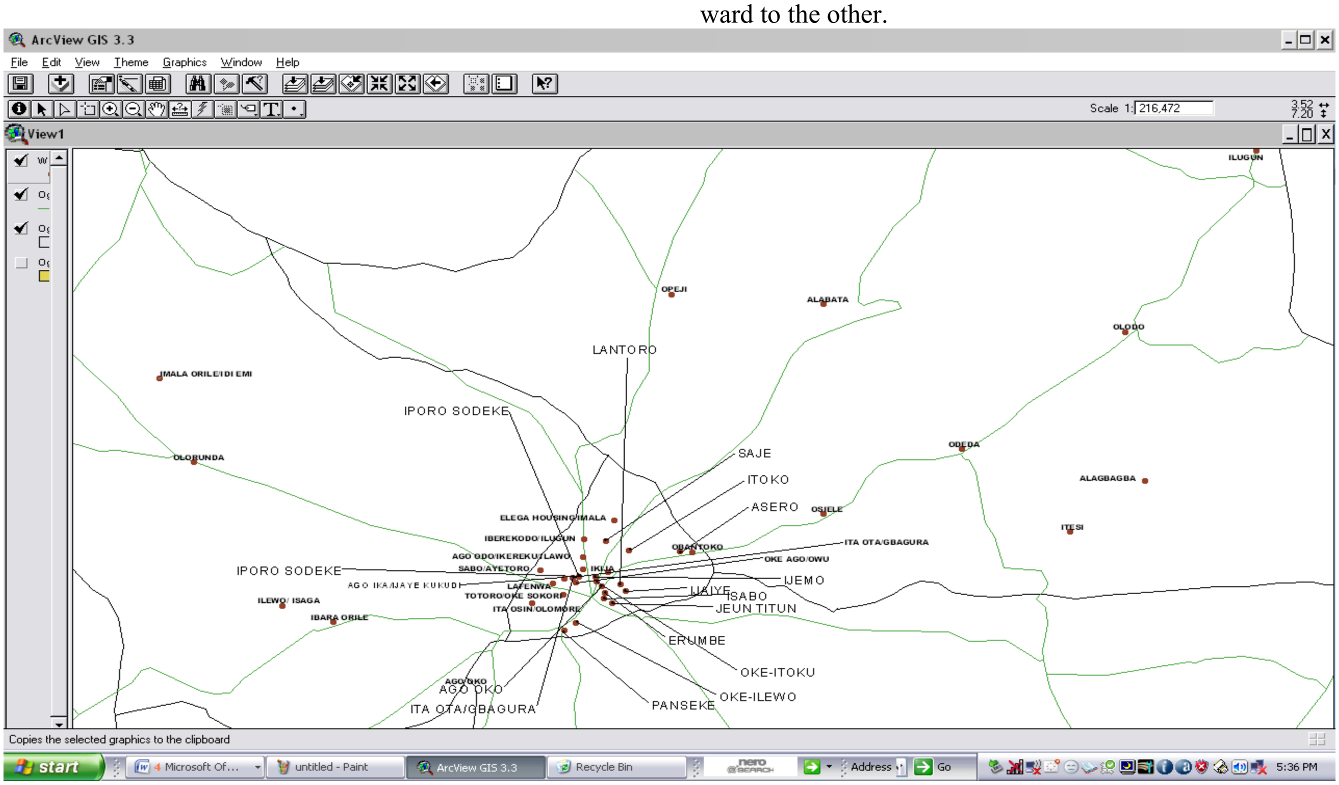

Figure 1. Wards location on the map

The result obtained from the distance reading is a 37 by 37 matrix which we then transformed into coverings according to a specified distance (precisely 10km from each wards). For example, the first cover which is {1,2,3,4,5,6,7,8, 9,10,15,21,25,26,27,28,29,30,31,32,33,34,35,36, 37} indicate those ward that can be covered within the range of 10km from ward 1. The part of the result of this process is shown in **figure 2.**

According to our first definition of set covering problem, the universal set U is {1, 2…37} and F = {C_1, C_2,…………….C_{37}}, now our aim is to find the minimum S a subset of F such that its union will give us U, and at this stage the wards are all covered with equal distances and the C_i picked are the wards where the fire station should be located. These data were then slotted into the balas additive and tora solver to solve the facility location problem. The results from the two algorithms were then compared to determine the optimal case.

1. **{1,2,3,4,5,6,7,8,9,10,15,21,25,26,27,28,29,30,31,32,33,34,35,36,37}** from Obantoko.
2. **{1,2,3,4,5,6,7,8,9,10,15,21,25,26,27,28,29,30,31,32,33,34,35,36,37}** from Ikija
3. **{1,2,3,4,5,6,7,8,9,10,15,21,25,26,27,28,29,30,31,32,33,34,35,36,37}** from Ago Oko

. . . .

. . . .

16 {16, 24} from Alagbagba

18 {18,21,23} from Osiele

. . . .

. . . .

37 {1,2,3,4,5,6,8,9,10,15,21,25,26,27,28,29,30,31,32,33,34,35,36,37} from Panseke

3.1 Models Used To Solve Fire And Emergency Facility Location Problem

3.1.1. Balas Additive Algorithm

The additive algorithm was one of the approaches known as branch and bound and is used to solve linear programs in **n** 0-1 variables by systematically enumerating a subset of 2^n possible binary n vectors, while using the logical implication of the 0-1 property to ensure that the whole set is implicitly examined. The technique employed in this algorithm is based on systematically assigning the value 0 and 1 to certain subset of variables and exploring the implications of these assignments by a sequence of logical tests. The simplicity of the procedure and its effectiveness when data are not too large makes it a better choice for this research work. Balas Additive algorithm required that the problem be put in standard form:

- The objective function is a form of minimization
- The m constraints are all inequalities of the form (≤)
- All the variables x_j are binary variables
- All objective function coefficients are non negative

Algorithm: Balas Additive Algorithm

1 Standardize the problem to the form:

$$\text{Min } Z = \sum_{j\in N} C_j x_j$$

s.t $\sum_{j\in N} a_{ij}x_j \le b_i$ for all $i \in M$.

where M = {1,2,…m} and N = {1,2,…n}

$x_j = \{0,1\}$, for all $j \in N$

2 Set an initial upper bound to Z = +∞, set i = 0, and J = { }.

3 Select the next partial solution J, solve the LPi of J and attempt to fathom using one of the three conditions listed below.

a. All completion violates one or more constraints. i.e compute
   i. $A= \{j{:}j \in N\text{-}J,\ a_{ij} \ge 0$ for all $i \in M$ such that $S_i \le 0\}$
   ii. N1 = N – J –A

   If NI = { } then fathom the partial solution J

b. All completion are inferior to the incumbent z' i.e compute
   i. $B = \{j : j \in N1,\ Z + C_j \ge Z'\}$
   ii. N2 = N1 – B

   If N2 = { } then fathom partial solution J

c. If constraint i is violated by the zero completion of the partial solution so that $S_i < 0$ i.e compute

   $C = \{i : S_i < \sum_{j\in N2} a_{ij}\ \text{-}\}$

   If C <> { } then fathom the partial solution J

   If all the fathom test fail, Goto step 6

4. If better solution is found, then update Z

5 If all elements of J is fathomed i.e underlined, then Z is optimal

Goto step 7

Else set J ← J,{-j} and repeat from step 3

6 Perform branching by:

i) Select free variable for forward step
ii) Set J ← J,{+j}

Set i = i + 1 and repeat step 3

7 Terminate

3.1.2. TORA

One of the powerful features of TORA is its graphical user interface (GUI) which enables users to express their problems in a natural way that is very similar to standard mathematical notation. This feature of GUI allows users to choose the next action being menu driven. This offers flexibility to users to increase or decrease the data size or to remove a particular variable completely.

TORA optimization solver has the following attributes:

a. Sets, which comprise of objects in programming model

b. Objective function of the problem

c. Constraints of Problem

d. input data

4 *IMPLEMENTATION AND RESULT*

4.1 *Format of input data*

In this paper, the input is 38x38 0-1 matrix where column 2-38 represents each covering and row 2-38 represents each ward. Therefore, for each column and row, the element is 1 if the ward is covered and 0 if not covered. E.g the name of the matrix is **a**, if **a[2][3]** = 1, it implies that ward 3 is covered by covering 2, otherwise it is not covered and the value will be 0. The whole input file format for this work is shown in **figure 4.** The format of the output is in form of a solution vector containing only zeros and ones i.e. 1 if a covering is selected and 0 if not selected. Each covering has specific name of wards covering other wards that are within the specified distance. (The name of each ward and the number attached to them is shown in **figure 3.**

1→Obantoko, 2→Ikija, 3→ Ago oko, 4→Elega Housing, 5→Iberekodo, 6→Ago ika, 7→Ayetoro, 8→Oke ago, 9→Totoro, 10→Ita osin, 11→Olorunda, 12→Imala Orile, 13→Ibara Orile, 14→Ilewo/isaga, 15→Ita ota, 16→Alagbagba, 17→Alabata, 18→Osiele, 19→Olodo, 20→Ilugun, 21→Ago odo, 22→Opeji, 23→Odeda, 24→Itesi, 25→Lafenwa, 26→Saje, 27→Itoko, 28→Ake, 29→Lantoro, 30→Ijemo, 31→Iporo sodeke, 32→Irunbe, 33→Ijaye, 34→Oke itoku, 35→Ijehun Titun, 36→Sabo, 37→Panseke.

Figure3 Names of wards and their number of identification

The input matrix shown in figure 4a and figure 4b was saved as text file and the balas additive algorithm was implemented using Java programming language.

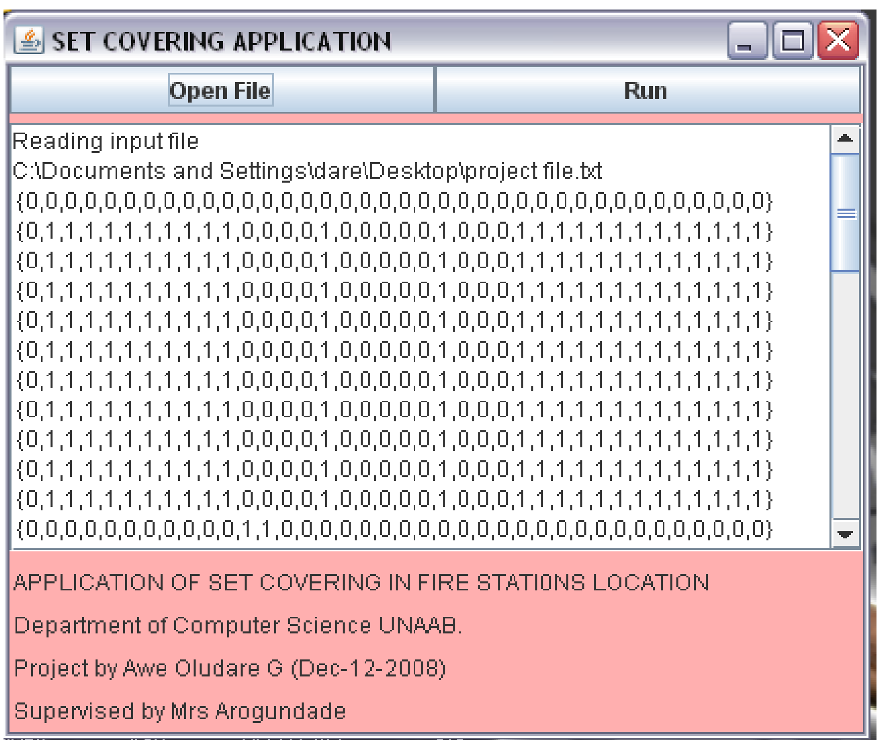

Figure 4a Input File Format

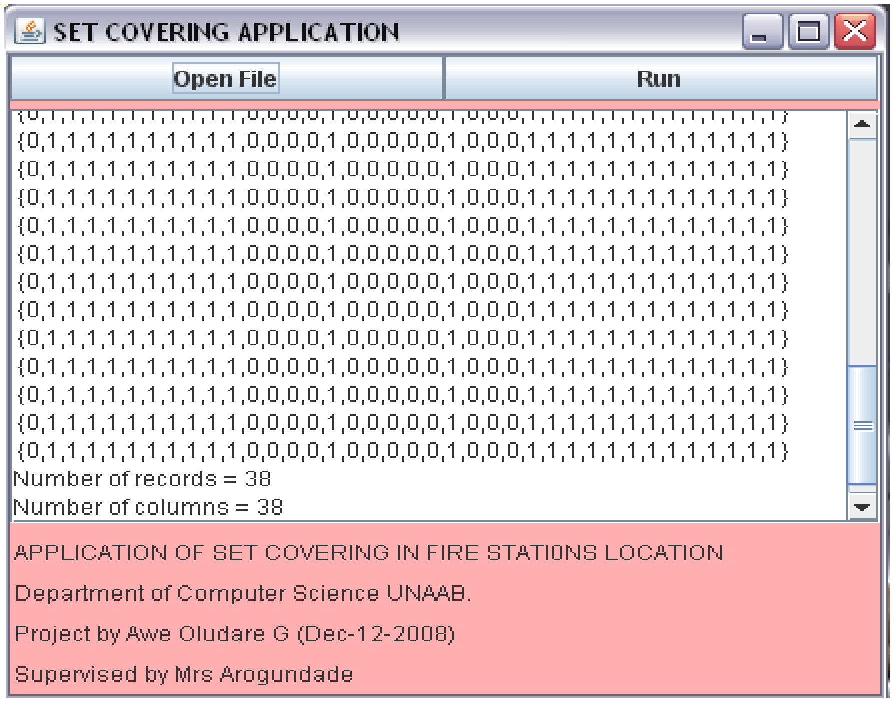

Figure 4b Input File Format

Once the input file has been selected, and then the program can be run to generate the output required. The result after the click of the "run" button is shown in figure 5 below.

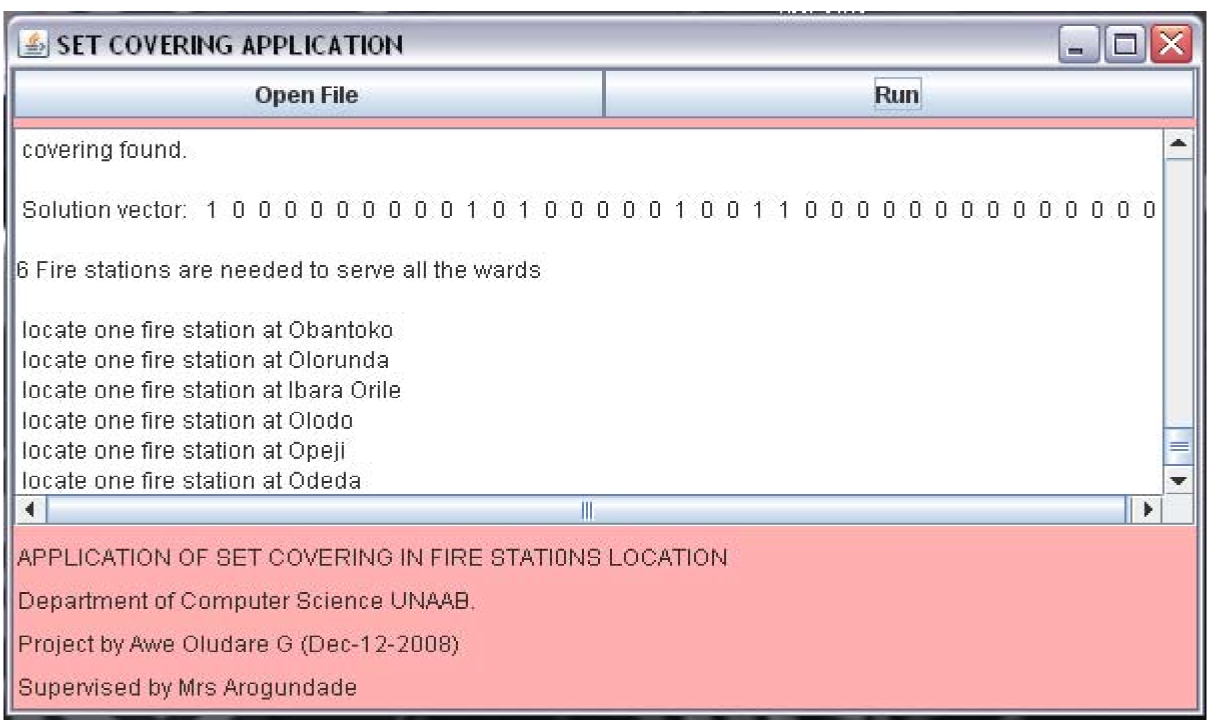

Figure 5 Set Covering output.

The balas additive result above showed that covering is found and also displayed the solution vector. It indicated that six fire stations are needed to serve all the wards and the locations of those stations are clearly stated.

The same result was obtained from TORA software in terms of optimality, but the locations are different and not clearly stated though it can be traced out.

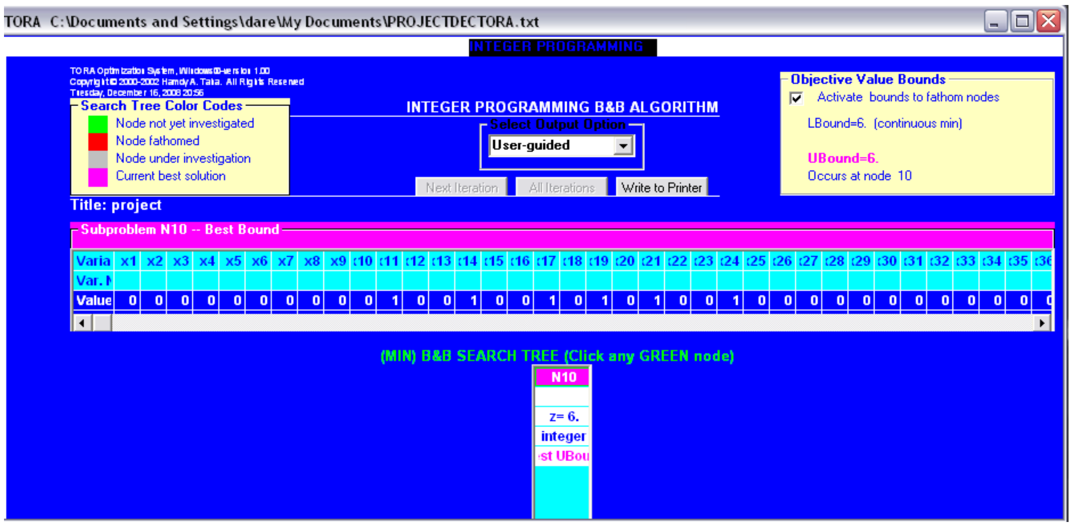

Figure 6 Result from TORA solver

Figure 6 shows the result of the TORA solver, the results is such that six fire stations are also needed to serve all the wards effectively but the locations of the fire service station are quite different from that of BALAS algorithm that was implemented. The locations indicated by the solver are locations 11, 14, 17, 19, 21, 24 which corresponds to the names of the following wards (Olorunda, Ilewo Isaga, alabata, Olodo, Ago-Odo and Itesi) as shown in figure 3. Tora solver does not list the names or numbers of the villages in its covering.

5 DISCUSSION

The necessity of the development of facilities location software for enhancing the decision making process and eventually productivity cannot be over-emphasized. The results obtained in this work showed that six fire stations are needed to serve every ward such that the maximum distance that a fire station service can go is 10 kilometers. It also showed that the location of the station should be Obantoko, Olorunda, Ibara orile, Olodo, Opeji and Odeda. The fire stations at Obantoko will render services to twenty five wards which are:Obantoko, Ikija, Ago-oko, Elega, iberekodo, Ago-Ika, Ayetoro, Oke-ago, totoro, Ita-osin, Ita-ota, Ago-odo, lafenwa, Saje, Itoko, Ake, Lantoro, Ijemo, Iporo-sodeke, irunbe, Ijaye, Oke-itoku, Ijeun-titun, Sabo and Panseke. Olorunda service station will serve two wards which are: Olorunda and Imala-Orile. Ibara-Orile service station will serve Ibara-orile and Ilewo-Isaga respectively. Olodo service station will serve three wards. They are: alagbagba, Olodo and Ilugun. Opeji will serve four wards which are: alabata, Osiele and Opeji. Finally odeda station will serve Odeda and Itesi. Though the result from TORA solver also indicated that six fire stations are needed, it did not specify the actual locations where the stations should be located. This shortcoming in TORA makes our output and implementation a better one. These results are presented in the table below for more clarity.

It shows the locations where the facilities are to be installed and also the villages to be covered by each of the facility (coverings) only for balas additive algorithm. The out put from balas additive algorithm does not show fair distribution. In the case of the facility in location Obantoko which is to serve 25 locations while others serve minimum of two locations and maximum of three. The facility in Obantoko will be overused.

Table 1: OUTPUT OF BALAS ADDITIVE ALGORITHM

Location identification number	Location name	Coverings
1	Obantoko	1,2,3,4,5,6,7,8,9,10,15,21,25,26,27,28,29,30,31,32,33,34,35,36,37
11	Olorunda	11,12
13	Ibara orile	13,14
19	Olodo	16,19,20
22	Opeji	17,18,22
23	Odeda	23,24

TABLE 2: OUTPUT OF TORA SOLVER

Location Identification Number	Location Name	Coverings
11	Olorunda	Not indicated
14	Ilewo-Isaga	Not indicated
17	Alabata	Not indicated
19	Olodo	Not indicated
21	Ago-odo	Not indicated
24	Itesi	Not indicated

6 CONCLUSIONS AND RECOMMENDATIONS

This result therefore should raise awareness and contribute to the aim of our government to adopt this tool which will definitely improve the functionality of fire stations in Nigeria by saving a lot of citizen's lives and properties. It should also be noted that the use of this system is not limited only to fire stations allocation alone, but also to other public facilities like schools, police station so as to increase response time and therefore reduce crime. It can also be used by private establishments.

REFERENCES

[1] J.E Beasley and P.C. Chu, "A genetic algorithm for the set covering problem", *European Journal of Operational Research*, vol. 94, 1996, pp.392-404.

[2] V.Chvatal. A greedy heuristic for the set covering problem. Mathematics of Operation Research, 4(3): 233-235, 1979.

[3] U.Fiege. A threshold for ln *n* for approximating set cover. Journal of the ACM, 45(4): 634-652, July 1998.

[4] D.S. Johnson. Approximation algorithms for combinatorial problems. J. Compute. System Science 9: 256-278, 1974.

[5] Larson Richard C., "A Hypercube Queuing Model for facility location and redistricting in urban emergency services", Computers and Operations Research 1 (1974) 67-95.

[6] Savas ES, "Simulation and cost-effectiveness analysis of New York's emergency ambulance service", Management Science 14 (1969) 608-627.

[7] M.L. Fisher and P.Kedia. Optimal Solution of set covering problems using dual heuristics. Management Science, 36: 674-688, 1990.

[8] Feo A. and G.C. Mauricio and A. Resende,(2002) "A Probabilistic Heuristic for a Computationally Difficult Set Covering Problem", Operations Research Letters, 8, 67-71.

[9] A. Caparara, M. Fischetti, and P. Toth. Algorithms for set coverin problems technical report OR-98-3, DEIS, University of Bologna, Italy 1998.

[10] N. Christofides and J.P Paix ao. Algorithms for large scale set covering problems. Annals of Operation Research, 43: 261-277, 1993.

[11] M.J Brusco, L.W. Jacobs, and G.M Thompson. A morphing procedure to supplement a simulated annealing heuristics for cost and coverage-correlated set covering problems. Annals of Operation Research, 86: 611-627, 1999.

[12] Toregas C., Swain R., Revelle C., and Bergman L., (1988) "The location of

emergency service facilities", Operation research 19, pg 1363-1373.

[13] Church Richard L., Gerard Ross a., (2003) "The multilevel location set covering model" Geographical Analysis Publication, pg 76-79.

[14] Garey M.R. and Johnson D.S., (2005) "Computer and interactability: a guide to the theory of NP-completeness", pg 34-38.

[15] Handy a. Taha., (2005) ' Operation Research: An introduction", Pearson education (Singapore) Pte. Ltd., pg 391-397.

[16] Daniel gulotta, (2006) "Application of linear Programming to set covering and related problems" Online journal of Operation Research, pg 1-2.