Cell Planning Using Genetic Algorithm and Tabu Search

Rodney S. Rambally, Avinash Maharajh
University of Trinidad and Tobago
rodney.rambally@utt.edu.tt

Abstract

This paper investigates the performance of the genetic algorithm and tabu search in solving the optimization problem of base station location. Optimization refers to maximizing radio coverage while minimizing equipment and maintenance costs. A comparative analysis of both the genetic algorithm and tabu search was undertaken. The effects of changing the population size as well as the type of selection were investigated for the genetic algorithm while the size of the candidate list and tabu tenure were examined for the tabu search. It was determined that the genetic algorithm performed best when tournament selection was employed with a population size of 10. A tabu tenure set to 1 and a candidate list size of 10 engendered the most optimal performance of the tabu list. Utilization of each algorithm's most optimal parameters allowed us to effectively compare the performances of the genetic algorithm with those of the tabu search.

1. Introduction

Acquiring a cell site and constructing a base station, as well as managing the other network-related costs such as operation and maintenance of the network, are very costly. It is thus necessary to examine the network topology in such a way as to minimize this cost and maximize the coverage of each site while maintaining a dependable network, limiting the amount of interference and the number of dropped calls. This paper investigates how the genetic algorithm and tabu search may be used to accomplish this objective; the paper also compares the performance of each of these techniques in meeting these objectives.

Finding the most optimal locations to place base stations poses one of the most challenging design problems in the planning and deployment of a communications network. Typically this procedure is conducted in an ad hoc fashion after manually inspecting maps depicting the propagation properties of the service area. A more efficient method is to utilize the data concerning the radio propagation characteristics of the service area, as well as a list of potential cell sites, in order to design the cells in such a way as to minimize the cost of equipment or base stations used while maximizing service or radio coverage provided in the area.

Because of the complex interactions within this optimization problem, the variety of algorithm parameters and neighborhood definitions, only experimental analysis can determine the performance and suitability of the genetic algorithm and the tabu search. Our objective was to compare, through experimental analysis, the performance of the genetic algorithm and tabu search for our optimization problem. MATLAB scripts and Java programs were used to manipulate the data and perform the searches.

2. Problem Formulation

Fundamentally we are searching for a process of optimizing base station location. To this end, we investigated the use of the genetic algorithm and tabu search in a manner such that radio coverage is maximized while equipment and maintenance cost are minimized. Since two different solutions were investigated, we found it meaningful to embark upon a comparison of the two methods which could then reflect their relative performances.

We had to find the parameters which best suit the genetic algorithm and the tabu search. The parameters which were investigated for the genetic algorithm were the type of selection used, the population size and the mutation operator. With the tabu search, the best values for the tabu tenure and candidate list size were identified. However, the challenging problem was in finding a level playing field over which these two very different algorithms could be compared. We found that by relying on the mutation operator to generate the next population of solutions, we were able to provide a common ground for comparison between the genetic algorithm and the tabu search. Since experimentation cannot be done easily through varying all three parameters simultaneously, we opted to base our comparisons for fixed values of the mutation operator. For a fixed mutation operator, we were able to determine the other parametric values of the

978-1-4244-4457-1/09/$25.00 ©2009 IEEE

genetic algorithm and tabu search that engendered the best solutions. We were then able to examine and compare these results.

Our optimization problem of dealing with base station locations involves several factors that must be included in an underlying cost function. At a given base station, a service coverage area may be determined with the use of a log-normal shadow fading model for radio propagation. The standard Hata model for power loss, in dB, at a distance *d* uses the following equation.

$$P_{loss}(d) = A + B \log(d) + N \qquad (1)$$

In this formula, *N* is a zero mean Gaussian random variable with a standard deviation of σ or variance σ^2. In our application, *A*, which is the path loss at a reference distance of 1 km, was chosen to be 50 dB. *B* is 10 multiplied by the path loss exponent which was set to 40. The variance, *N*, was set to 10. These values are somewhat typical of values commonly encountered in studies and application. They were picked in this case, not only to mimic real life situations, but also to provide a common ground for analysis of the two algorithms under inspection.

Using the formula given above, the radio path loss P_{loss} was calculated at each possible base location (for 10,000 points – see later). A cutoff value P^*, assigned a value of 100dB, was then chosen such that if $P^* \leq P_{loss}$ for any point then it is deemed that radio coverage was not adequate. The cost function must take into account the radio propagation characteristics of the service area which could be determined using either empirical path loss propagation models or ray-tracing software as well as the number of base stations. The cost function which we used is

$$f = k\frac{N_{BS}}{R^{\beta}} \qquad [1] \qquad (2)$$

where N_{BS} is the number of base stations. *R* is the radio coverage of the selected station or the percentage of locations in the service area which are covered by at least 1 base station. β symbolizes the weight attached to maximizing coverage in favor of minimizing the number of base stations and *k* is a scaling factor. We chose $\beta = 3$ and $k = 10^4$ for our problem.

We had to investigate which parameters engendered the most optimal performance for each search technique. To do this we formulated a square service area to be broken up into a grid of 100x100 points spaced 1 unit apart. Using MATLAB programs, 51 possible base station locations were randomly generated with a uniform distribution, as shown in Fig. 1.

Fig. 1 – Possible Base Station Locations

Three different variations of the mutation operator were used to define the neighborhood search space for both the genetic algorithm and the tabu search. These are referred to as Mutation Operator I, Mutation Operator II and Mutation Operator III. This was the major concept that allowed us to compare both algorithms without any bias.

Regarding Mutation Operator I, we followed along the lines of [1] and chose to randomly flip any one bit of the current solution. The only difference is that this randomly chosen bit is flipped with a probability of 1 (i.e. the bit is always flipped) instead of 0.9. We used Mutation Operator II to add a little more diversity in the sense that every bit of the current solution had a probability of being flipped. We varied this probability of mutation, Pm, for three different values, 0.001, 0.01 and 0.1. Mutation Operator III was the most loosely defined in the sense that it was used to simply and randomly pick a new point from the entire search space to be the next solution (rather than making decisions based on the current solution).

Regarding the genetic algorithm, there are several popular selection methods. For the purpose of our research, we focused on proportional selection, tournament selection and rank-based selection.

3. Related Work

We now summarize some of the major works that paved the way for cell planning and optimization. The problem of optimizing sites for base station transmitters in a microcell or PCS system has been addressed using the simulated annealing optimization technique [2]. The results show that plausible optimum transmitter sites can be successfully selected automatically regardless of how randomly the starting positions of the transmitters are selected.

In [3], the authors propose an interesting algorithm which uses a propriety branching scheme, an optimization gap tolerance of at least 1%, and two sets of global valid inequalities that tighten the upper bounds obtained from the linear programming relaxation.

In [4], the authors discuss integer programming models and discrete algorithms aimed at supporting the decisions in the process of planning locations for new base stations. The authors also present randomized greedy and stingy procedures as well as a randomized method combining the types of steps used in the integer programming models and discrete algorithms mentioned earlier.

The authors in [5] present a genetic algorithm to select AP devices, locations and antennas in WLAN planning. They also discuss AP configuration including transmission power and frequency channel in WLAN planning. Compared to manual network planning, the algorithm was able to create a network plan with 133 % capacity, 98 % coverage, and 93 % cost. Manually the corresponding network planning took hours, whereas the algorithm's computation time was 15 minutes.

In [6] the authors discuss the conflicting objectives involved in base station planning; they also characterise a multi-objective optimization problem. A genetic encoding of the third generation mobile network planning problem and parallel genetic algorithms to solve it is also presented.

The authors in [7] proposed the following efficient three step approach: a pre-processing phase based on a filtering principle, an optimization phase using tabu search, and a post optimization phase based on fine tuning. The first (pre-processing) is a parameterized phase allowing users to generate a variety of reduced sets of base stations of interest for devising an ultimate solution. The tabu search algorithm is based on a binary representation of the search space, integer techniques such as frequency-based tabu list management, and penalty-based diversification. For post optimization, various techniques were proposed, either to improve the objectives or to enhance constraint satisfaction. This approach was successfully applied to two large and realistic sets of test data, thereby proving its robustness, flexibility and effectiveness.

4. Effect of Population Size and Selection on Genetic Algorithm

With Mutation Operator I, the type of selection used is the major parameter contributing to the performance of a genetic algorithm. We utilized all three types of selection and then compared the results. The other major factor influencing the genetic algorithm is the size of the population. Fig. 2 shows the comparison of the average performance of the genetic algorithm for the three selection techniques with population sizes of 10, 30 and 50. All runs were conducted for 1000 cost function evaluations.

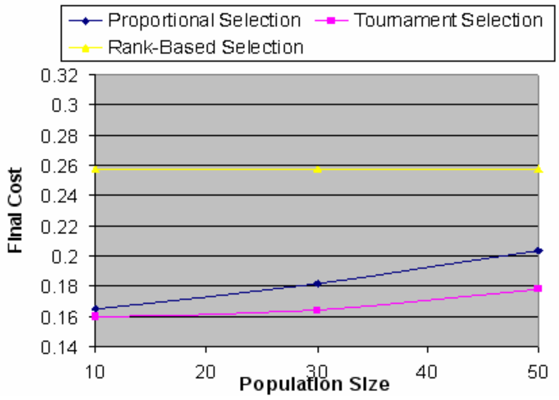

Fig. 2 – Performance of Genetic Algorithm

The graph indicates poor performance of rank-based selection which performs the same for all three population sizes tested with a final cost of about 0.258. The performance of tournament and proportional selection declines almost linearly with increasing population sizes. Both have their best performance at a population size of 10 where the tournament selection has a cost of 0.16 and the proportional selection has a cost of 0.165. The tournament selection outperformed the proportional selection at every level with the disparity in costs growing larger as the population size increased. This implies that tournament selection provides for a better method of assignment than the probability distribution used in the proportional selection method. Since the disparity becomes greater, this seems to indicate that tournament selection utilizes a more efficient method of choosing fitter parents for mutation. One explanation for the deteriorating performance of these two selection methods is that since we are comparing all the algorithms for the same number of evaluations, then the number of generations $G = (1000)/ (PS)$ is smaller where the size of the population is higher. This implies that the performance of the genetic algorithm improves linearly with the number of generations for these two selection schemes, at least for the first 1000 evaluations.

5. Effect of Tabu Tenure and Candidate List Size on Tabu Search

The tabu search was implemented with the criterion that any base station location that is selected within the last K iterations is considered tabu and is not to be selected at the current iteration. K is the tabu tenure. The candidate list v, which determines how many neighbors are considered at each step in the search, can also be varied.

Several algorithms were run and compared for different parameter values of K and v to determine their effects on the tabu search. This was done for the same starting point for 10 runs, each consisting of 1000 function evaluations. Thus the number of iterations $I = 1000/v$. The algorithms were compared for $K = \{1, 3, 4, 7, 9\}$ and $v = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20\}$. It is to be expected that a higher K would cause the performance to decline because, if the new base station located at some iteration in the search is a good one, it would not be considered for the next K generated solutions. With regards to the candidate list v, it is expected that performance increases as the size of the list increases since one can better sample the neighborhood of the current point in order to determine a good point to move to. The results are shown in Fig. 3.

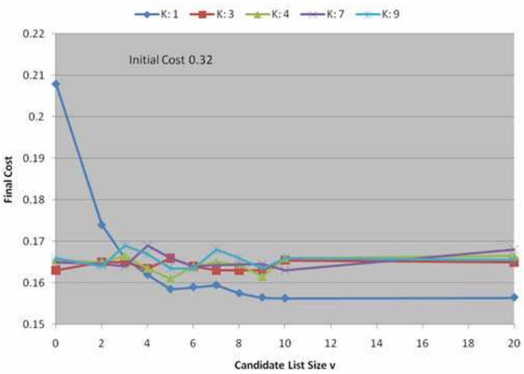

Fig. 3 – Performance of Tabu Search

The tabu search performs best when the tabu tenure is smallest, that is, when $K = 1$. Fig. 4 shows the minimum, average and maximum final costs obtained by the tabu search for $K = 1$ and the different values of v.

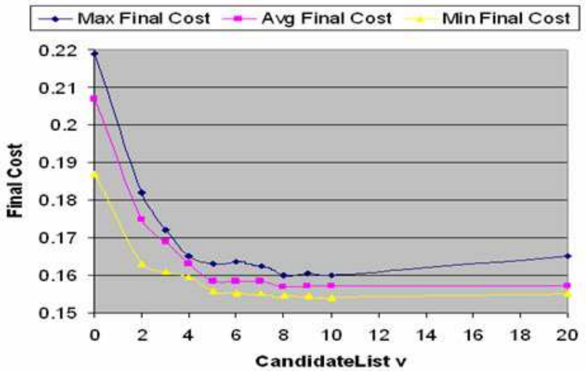

Fig. 4 – Performance of Tabu Search when K = 1

The algorithms performed progressively better as v is increased, reaching the best performance for a candidate list size of 10.

6. Comparative Analysis Using Mutation Operator I

The genetic algorithm and tabu search were first compared with each other using the best suited parameters discussed earlier. For the genetic algorithm we used tournament selection with a population size of 10, and for the tabu search tabu tenure K was set to 1 with the candidate list size v equal to 10. Both algorithms were then compared using Mutation Operator I; each was run 10 times for 10,000 function evaluations. Fig. 5 shows the maximum, average and minimum final costs obtained for each technique. While both algorithms performed similarly, the tabu search edged out the genetic algorithm both in terms of average final cost and variance. The genetic algorithm had maximum, average and minimum final costs of 0.159, 0.155 and 0.153 respectively while the tabu search had comparative values of 0.156, 0.154 and 0.153. This indicates that the tabu search was able to obtain the best solution in almost every run. The genetic algorithm possibly had a little more variance because it involved a population of solutions rather than 1 solution at a time.

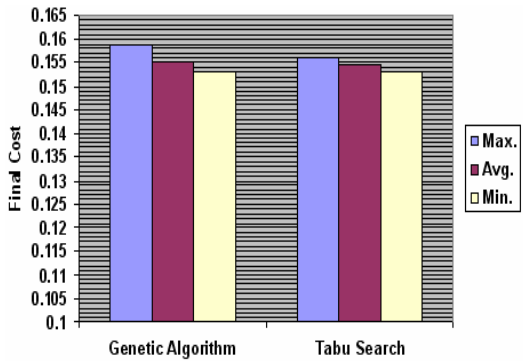

Fig. 5 – Comparison of Final Costs.

Fig. 6 shows the progress of both algorithms. While the tabu search is outperformed for a lower number of evaluations, it overtakes the genetic algorithm after about 90 iterations. However, both algorithms converge to a similar final cost of approximately 0.153. The tabu search had a final cost of 0.1528 which translated into 10 base stations with a radio coverage of 86.82%. The genetic algorithm's final cost was 0.1533, meaning that 11 base stations were selected that provided 89.52% coverage of the service area. This could be interpreted to mean that for a relatively smaller search space, the genetic algorithm may be the more efficient algorithm.

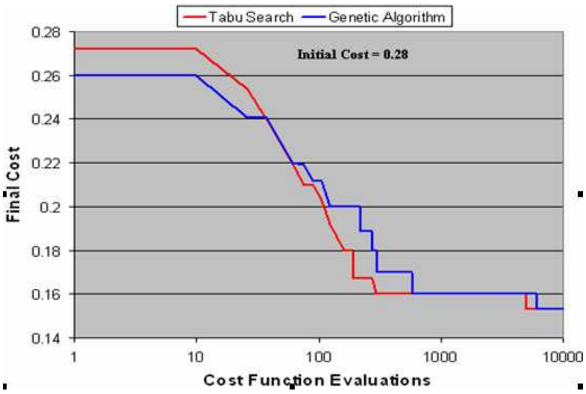

Fig. 6 – Comparison of Search Algorithms for 1 Sample Run.

7. Comparative Analysis Using Mutation Operator II

In order to investigate the performance of the genetic algorithm and tabu search in a more holistic manner, we needed to examine both algorithms for the other two mutation operators. Fig. 7 – 9 display the performances of both searches using Mutation Operator II and varying the value of *Pm* from 0.001 to 0.01 and finally 0.1 respectively. For each *Pm*, the algorithm was run 10 times for 10,000 function evaluations. The effect of increasing the probability of mutation leads to a more random neighborhood. For example, in the case where $Pm = 0.1$, there is a tenth probability of changing each bit in a single step. This will lead to neighborhood points differing on average by about 0.1 x 51 which is 5 bits.

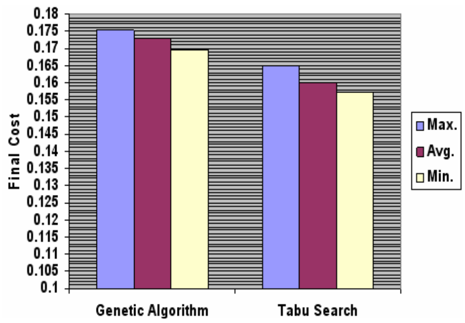

Fig. 7 – Comparison for Pm = 0.001

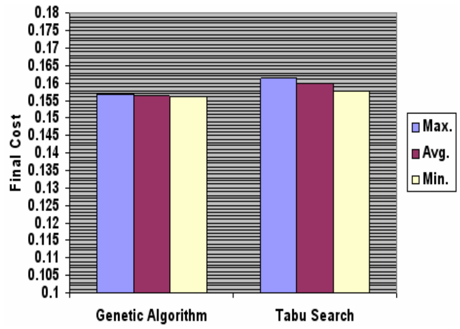

Fig. 8 – Comparison for Pm = 0.01

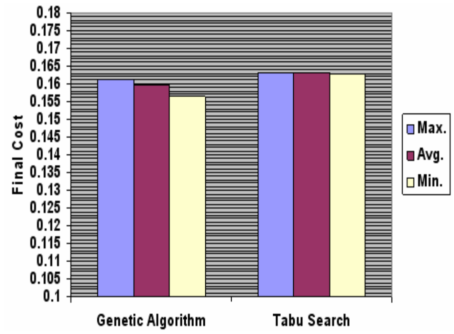

Fig. 9 – Comparison for Pm = 0.1

From Fig. 7 where $Pm = 0.001$, the tabu search, with an average final cost of approximately 0.160, outperformed the genetic algorithm, whose average final cost was 0.173. The tabu search performed well because the tabu list directed it out of the poor regions. The genetic algorithm performed poorly because this very low mutation probability did not lead to much diversity in the next generation; hence many more function evaluations were done with little variety. For $Pm = 0.01$ and $Pm = 0.1$ (Fig. 8 and Fig. 9), we obtained different results. In these two cases the genetic algorithm outperformed the tabu search. The next generation offered more diversity as the search space became more random. The tabu search provided a slightly worse final cost of 0.166 for $Pm = 0.1$. However, in the 10 runs observed, the final cost obtained showed little to no variance. We observed that changing the probability of mutation had a greater effect on the tabu search than on the genetic algorithm. This clearly demonstrated that changing the mutation operator does in fact alter the performance of both the genetic algorithm and tabu search.

8. Comparative Analysis Using Mutation Operator III

It is to be expected that for a totally random neighborhood both algorithms should perform equally.

This was verified by executing the searches using Mutation Operator III. We noticed that there was little or no variance. After the 10,000 function evaluations for 10 runs each, the final cost was approximately 0.26. This is a significantly worse value than those found using the other more structured mutation schemes. It can therefore be concluded that having a defined, logical neighborhood scheme can drastically improve the results rather than utilizing a totally random scheme.

9. Conclusion

Fig. 10 shows the best solution obtained after running all of the simulations. The best overall solution was encountered with the tabu search having a tabu tenure of $K = 1$ and candidate list size $v = 10$ while using Mutation Operator 1. The results of this tabu search yielded 86.82% coverage with only 10 base stations and having a final cost of 0.1528. This evaluation was accepted as the best overall solution because out of all the simulation evaluations which were performed, this gave the lowest cost thus indicating the most optimal solution. While in our case, the tabu search performed best, the genetic algorithm also performed well, actually outperforming the tabu search, in cases where the probability of mutation was set to a reasonable value.

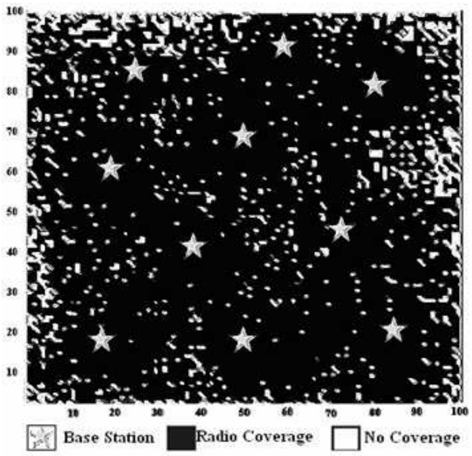

Fig. 10 – Best Final Solution for the Optimization of Base Station Locations.

Both algorithms performed poorly as the randomness of the mutation operator was increased. The tabu search suffered to a greater extent in its performance than the genetic algorithm. As the randomness increased the genetic algorithm started to outshine the tabu search even though its performance also declined. This result deviates from the findings of other research studies where the algorithms were compared. This indicates to us the importance of the mutation operator in the performance of these algorithms. When the mutation operator specified total randomness, we saw the results we had anticipated, i.e., both algorithms performed just as poorly with very little variance in their outputs.

We also saw that the use of these algorithms in the planning of cell site locations can lead to more favorable results. Instead of building 51 cell sites that will surely produce 100% coverage, we were able to reduce that number to 10 base stations that still provide 86.82% radio coverage. That translates into an 80% decrease in not only locations, but also a drastic reduction in network cost with only a minimal drop of 13% in radio coverage.

10. References

[1] P. Calegari, et. al. "Genetic approach to Radio Network Optimization for mobile systems", Vol. 2 IEEE 47th Vehicular Technology Conference, 1999, pp 755-759.

[2] H. R. Anderson, J. P. McGeehan, "Optimizing microcell base station locations using simulated annealing techniques", Proceedings of IEEE Vehi. Tech. Conf., vol. 2, June 1994, pp 858-862.

[3] J. Kalvenes, J. Kennington, E. Olinick, "Base station location and service assignment in WCDMA network", Technical Report 02-EMIS-03, Southern Methodist University, Oct. 2002.

[4] E. Amaldi, A. Capone, F. Malucelli, "Planning UMTS base station location: optimization models with power control and algorithms", IEEE Trans. Veh. Technol., vol. 2, issue 5, Sept. 2003, pp. 939-952.

[5] T. Vanhatupa, M. Hannikainen, T.D. Hamalainen, "Genetic Algorithm to Optimize Node Placement and Configuration for WLAN Planning", 4th International Symposium on Wireless Communication Systems, ISWCS 2007.

[6] C. Maple, J. Zhang, "Parallel Genetic Algorithms for Third Generation Mobile Network Planning", Proceedings of the International Conference on Parallel Computing in Electrical Engineering (PARELEC'04), 2004.

[7] M. Vasquez J.K Hao, "A heuristic approach for antenna positioning in cellular networks", Journal of Heuristics, vol. 7, no. 5, 2001, pp. 443-472.

[8] M. Mitchell, "An Introduction to Genetic Algorithms", MIT Press, Cambridge, MA, 1996.