International Journal of Computer Theory and Engineering, Vol. 1, No. 5, December, 2009
1793-8201

Comparative Analysis of Simulated Annealing and Tabu Search Channel Allocation Algorithms

Maninder Singh Kamboj, Jyotsna Sengupta, Department of Computer Science, Punjabi University, Patiala

***Abstract*—The major aim of the Frequency Allocation Problem in mobile radio networks is to allocate a limited number of frequencies to all radio cells in a network while minimizing electromagnetic interference due to the re-use of frequencies. This problem is identified as NP-hard, and of huge significance in practice since enhanced solutions will permit a telecommunications operator to administer bigger cellular networks. This paper presents and implements two algorithms tabu search and simulated annealing. The algorithms are tested on realistic and large problem instances and compared. Results of comparison show that the tabu search is less efficient than simulated annealing algorithm.**

***Index Terms*— Frequency Allocation Problem, Tabu Search, Simulated Annealing**

I. Introduction

The Frequency Allocation Problem (FAP) is one of the important applications in mobile radio networks engineering. Though diverse versions of the FAP can be defined, the key function is to allocate a restricted number of available frequencies to every cell in a mobile radio network whereas minimizing electromagnetic interference due to the reuse of frequencies. The complexity of this application comes from the fact that an adequate solution of the FAP should fulfill a set of multiple constraints, which impose contradictory objectives. The harshest constraint concerns a very restricted radio spectrum consisting of a little number of frequencies. Telecommunications operators must handle frequencies for their networks, whatsoever the traffic quantity to be covered, and this, in agreement with national international policy. This constraint imposes a very high degree of frequency reuse, which in turn increases the possibility of frequency interference. Apart from this constraint, there are other frequency interference constraints which state that frequencies allocated to various cells must satisfy a prearranged separation distance in the frequency domain.

The basic FAP is NP-hard in its easiest form because it is reduced to the graph coloring problem (Hale (1980)). More commonly, the problem is equal to the "set T-coloring problem" (Roberts (1991)). Hence, it is not likely to find any efficient algorithm for this problem. Many heuristic methods have been proposed so far, to deal with the FAP in cellular networks, including graph coloring algorithms (GCA) [5], constraint programming (CP)[2], artificial neural networks [6] [4], and evolutionary algorithms (EAs) [3]. In the setting of military application related techniques have been created and experimented on a closely related problem called "radio-link frequency assignment" (CALMAR [1]).

The paper is organized as follows. In Section 2, the FAP is modeled as an optimization problem. In Section 3, TS and SA are briefly reviewed. In Section 4, experimental results are presented and compared. Conclusions are given in the last section.

II. The Frequency Allocation Problem

A cellular network is expressed by a set $\{C_1, C_2 \ldots\ldots C_N\}$ of N cells, each cell C_i needs T_i frequencies. The amount Ti, called the traffic of Ci , is ascertained by an evaluation of the maximum possible number of communications which can concurrently occur within a cell.

The necessary FAP consists of allocating to each cell C_i of the network T_i frequencies taken from a set of vacant frequencies while abiding a set of frequency *interference constraints*. Since the number of vacant frequencies is very less and generally much lesser than the sum of the entire traffics of the network, frequencies should be reused by diverse cells in an assignment. This frequency reusing may result to frequency interference.

Interference occurs when two frequencies allocated to a same or two neighboring cells are not adequately separated. Therefore, the *frequency interference constraints* on a network are divided into *Co-channel, Adjacent channel, and Co-site constraints:*

- *Co-channel constraints:* A pair of transmitters situated at different sites must not be allocated the same frequency in case they are not sufficiently**,** geographically separated. If f_i and f_j are the frequencies assigned to transmitters i and j respectively, then this constraint is expressed: $f_i \neq f_j$.
- *Adjacent channel constraints:* Even if the above constraint is fulfilled (that is if $f_i \neq f_j$) interference may still occur.
- *Co-site frequency separation:* Any pair of frequencies at a site must be separated by a certain, fixed amount.

These interference constraints are suitably explained by a symmetric *compatibility matrix* $M[N, N]$, whereas N is the number of cells in the network and each element of M is a nonnegative integer. Let $f_{i;k}$ represent the value of the k^{th} frequency ($k \in \{1 \ldots. T_i\}$) of Ci and let $\{1 \ldots. N_F\}$ denote the

set of *NF* available frequency values, then the interference constraints are formulated as follows:

- $M[i; i]$ ($i \in \{1\ldots.N\}$) is the minimum frequency separation necessary to satisfy the co-cell constraints for the cell *Ci* .
  $\forall m; n \in \{1\ldots.Ti\}, m \neq n, |f_{i,m} - f_{i,n}| \geq M[i, i]$:
- $M[i; j]$ (i, $j \in \{1\ldots N\}$, $i \neq j$) represents the minimum frequency separation required to satisfy the adjacent-cell constraints between two cells C_i and C_j . $M[i, j] = 0$ means there is no constraint between the cells C_i and C_j .
  $\forall m \in \{1\ldots.. T_i\}, \forall n \in \{1\ldots.. T_j\}, |f_{i,m} - f_{i,n}| \geq M[i; j]$.

III. Channel Allocation algorithms

Various algorithms implemented in this paper are as follows:

A. Tabu Search

Tabu Search is a meta-heuristic created for tackling hard and large combinatorial optimization problems. Opposite to randomizing approaches such as Simulated Annealing where randomness is widely used, TS is based on the principle that intelligent search must embrace more efficient and systematic forms of direction such as memorizing and learning.

TS can be represented as a type of neighborhood search with a set of significant and complementary components. For a specified instance of an optimization problem characterized by a cost function *f and a* search space *S*, a function $N : S \rightarrow 2^S$ is first presented to find out a neighborhood. A typical TS algorithm starts with an initial configuration *s* in *S* and then continues iteratively to visit a sequence of locally best configurations following the neighborhood function. At every iteration, a *best* neighbor *s*' $\in N(s)$ is required to put back the present configuration even if *s*' does not progress the current configuration in terms of the cost function. To stay away from the problem of possible cycling and to permit the search to go past local optima, TS introduces the concept of *Tabu list*, one of the most vital components of the method.

A tabu list is a unique short-term memory with the purpose of maintaining a selective history *H*, composed of earlier encountered solutions or more normally pertinent attributes of such solutions. A simple TS approach based on this short-term memory *H* consists in forbiding solutions of *H* from being considered again for next *k* iterations (*k*, called tabu tenure). At each iteration, TS looks for a finest neighbor from this dynamically modified neighborhood $N(H, s)$, instead of $N(s)$ itself. Such strategy forbid tabu from being confined in short-term cycling and allows the search process to go beyond local optima.

When attributes of solutions are registered in tabu list, some unvisited, yet exciting solutions may be prohibited from being considered. *Aspiration criteria* might be used to conquer this problem. One commonly used aspiration criterion consists of removing a tabu classification from a move when the move leads to a solution superior than the best obtained so far.

TS utilize an aggressive search strategy to take advantage of its neighborhood. So, it is important to have special techniques and data structures which permit a quick updating of move evaluations, and lessen the effort of finding best moves. Furthermore, *candidate list strategies* can be used to restrict the neighbors to be considered at every iteration to a subset of the whole neighborhood. A typical strategy consists of determining subsets of prominent moves, such as those considered promising to direct to enhanced solutions. A superior candidate list strategy shared with an efficient technique for move evaluations is essential for high solution speed and good quality.

Here follows the pseudo code for *TS*. The variable BEST stores the best configuration that *TS* generated and TABU is the memory that the algorithm uses in order to mark the forbidden moves. The STOP variable controls the number of iterations of the algorithm. When this parameter is changed to TRUE the program is terminated. The neighborhood generation strategy is given in the following sections.

Step 1:
Generate a random or compute a valid initial configuration *A*
TABU $\leftarrow$ *A*
STOP $\leftarrow$ **false**
BEST $\leftarrow$ *A*

Step 2:
Chose the best feasible solution $B \in$
{ *NEIGHBOURHOOD(A)* – TABU }
if (*cost(B)* $\leq$ *cost(A)*) **then**
BEST $\leftarrow$ *B*
update TABU
update STOP
A $\leftarrow$ *B*

Step 3:
if (STOP = **true**) **then**
output BEST
else
goto *Step 3*

B. Simulated Annealing

Simulated annealing (SA) is a stochastic computational technique evolved from statistical mechanics for discovering near globally-minimum-cost solutions to big optimization problems. In several instances, determining the global minimum value of an objective function with various degrees of freedom subject to inconsistent constraints is an NP-complete problem, since the objective function will tend to have several local minima. A procedure for solving this type of optimization problems must sample the search space in such a way so that it has a high probability of finding the optimal or a near-optimal solution in a reasonable amount of time. In the last decade or so, simulated annealing has shown

itself to be a technique which fulfills these criteria for a wide variety of problems.

SA method itself has a direct similarity with thermodynamics, particularly with the manner that liquids freeze and crystallize, or metals cool and anneal. The molecules of a liquid move freely at high temperatures with respect to one another. If the liquid cools down slowly, thermal mobility is controlled. The atoms are able to line themselves up and create a pure crystal that is absolutely regular. For the system this crystal is the state of minimum energy, which would correspond to the optimal solution in a mathematical optimization problem. However, if a liquid metal cool down quickly, it fails reach a minimum energy state but rather higher energy state, in the mathematical sense, to a sub-optimal solution created by iterative improvement or hill-climbing.

Given a sequence of moves (i.e. neighbouring configurations), a simulated annealing thermodynamical system was assumed to change its arrangement from energy E_{old} to energy E_{new} with probability

$$prob = e^{(-(Enew-Eold)/Bt)}$$

where t and B are known as the Boltzmann constant. If $E_{new} < E_{old}$ then the new configuration has a lower energy state than the old one and the system always accepts this move. If $E_{new} > E_{old}$ then this new configuration can be accepted with probability prob ($0 < prob < 1$) and therefore, help the system to come out of a local minimum. This method, of constantly taking downward steps while at times taking upward steps is known as the Metropolis Algorithm.

Initialise t
Generate random configuration X_{old}
WHILE $t > t_{min}$ DO
FOR i = 1 to NUM_{loop} DO
generate new configuration, X_{new}, from X_{old}
calculate new energy, E_{new}
calculate $\Delta E = E_{new} - E_{old}$
IF $\Delta E < 0$ or random < prob = $e^{-\Delta E / t}$ THEN
Xold ← Xnew
Eold ← Enew
END IF
END FOR
reduce t (e.g. t = 0:9t)
END WHILE

IV. COMPARISON

The algorithms were implemented in C++ and tested on several instances. Small samples of these tests are presented here.

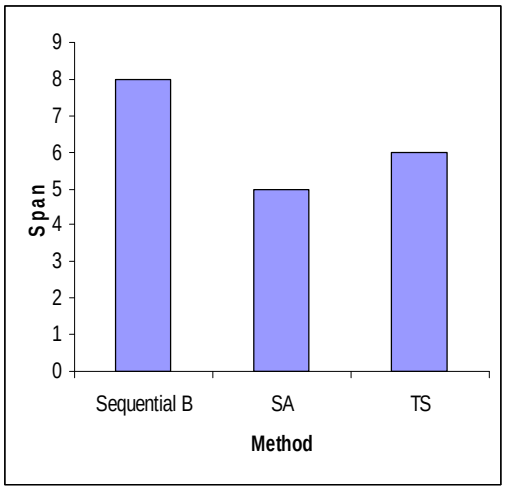

Fig. 1 Comparison of Algorithms for 15 vertices

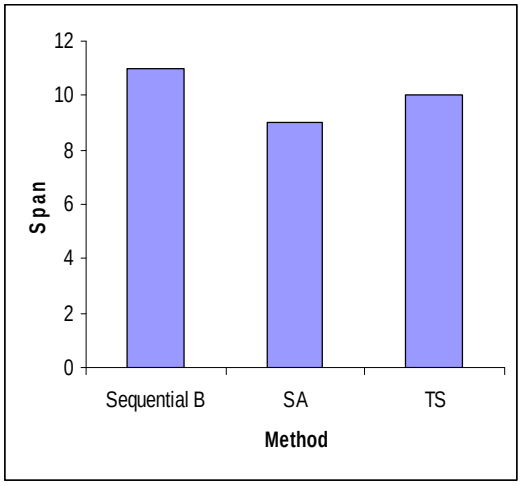

Fig. 2 Comparison of Algorithms for 20 vertices

The x-axis of the comparison chart represents the algorithms used and y-axis represents the number of channels used by each algorithm. The two algorithms (simulated annealing and tabu search) are tested on two instances having 15 and 20 nodes, and results are also compared with the Sequential B algorithm. The results of the comparison have shown that simulated annealing method performs better than tabu search.

V. CONCLUSION

We have presented two original parallel algorithms for solving the frequency assignment problem. *Tabu search is* less efficient than *Simulated Annealing*. The big advantage of *SA* is its capability to move to states of higher energy. On the other hand the version of *TS* we implemented does not support this feature. This is why *TS* cannot run away from likely topic minima and normally results inferior configurations. Tabu memory is not a mechanism to overcome topic minima but a method to evade cycling that is to create a larger amount of possible configurations in fewer iterations. From this conclusion discovers the need for a hybrid method that will provide *TS* with the capability to

allow moves to configurations of higher interference.

REFERENCES

[1] CALMAR, Symposium on Combinatorial Algorithms for Military Applications, Scheveningen, Holland, (1995).
[2] Caminada, A., "R´esolution du Probl`eme de l'Affectation des Fr´equences Par Programmation Par Contraintes," Technical Report FT.CNET/BEL/POH/CDI/7195/CA, CNET, Belfort, (1995).
[3] Dorne, R. and J.K. Hao, "Constraint Handling in Evolutionary Search: A Case Study on Frequency Assignment," *4th Int. Conf. on Parallel Problem Solving from Nature*, Lecture Notes in Computer Science 1141, Springer-Verlag, (1996), pp. 801–810.
[4] Funabiki, N. and Y. Takefuji, "A Neural Network Parallel Algorithm for Channel Assignment Problems in Cellular Radio Network," *IEEE Trans. on Vehicular Technology* 41, (1992), 430–437.
[5] Gamst, A., "A Resource Allocation Technique for FDMA Systems," *Alfa Frequenza* 57(2), (1988), 89–96.
[6] Kunz, D., "Channel Assignment for Cellular Radio Using Neural Networks," *IEEE Trans. on Vehicular Technology* 40, (1991), 188–193.