Global Search Techniques for Problems in Mobile Communications

Bhaskar Krishnamachari and Stephen B. Wicker
Wireless Multimedia Laboratory
School of Electrical Engineering
Cornell University

February 1999

1 Introduction

In the last two decades, wireless communication systems such as cordless phones, paging systems, wireless data networks, satellite-based and cellular mobile systems have been steadily increasing in both popular importance and technological sophistication. The first-generation wireless systems were developed in the late 1970's and 1980's and were based on analog technology (such as the Advance Mobile Phone Service (AMPS) by AT&T and Nordic Mobile Telephone (NMT) by Ericsson). As demand increased and digital technology matured in the 1980's and 1990's the second-generation digital wireless systems were designed (such as Global System for Mobile Communications (GSM) in Europe and Digital AMPS in North America). These systems, currently in use, offer higher system capacity and improved quality of service. Third-generation systems, referred to as personal communication systems (PCS), are currently under development and expected to be deployed at the beginning of the 21st century [1], [2].

The utilization of wireless communications has shown growth rates of 20-50% per year in various parts of the world. As the benefits of digital technology are realized, it is expected that there will be demand for the transmission of high-bandwidth, high-quality multimedia information content over these systems. The PCS goal is to provide to every user the ability to exchange such information securely with anyone, anytime, anywhere in the world, using a unique personal telecommunication number (PTN). To meet these challenging expectations, intensive research has been undertaken in recent years to develop a sophisticated PCS with increased network capacity and performance. One of the trends in this research has been the growing incorporation of Artificial Intelligence (AI) techniques into such systems [4].

Like all engineering endeavors, the subject of mobile communications also brings with it a whole host of complex design issues. A number of issues concerning resource allocation, design, planning, estimation and decision in mobile communications can be formulated as combinatorial optimization problems. Many of these problems are NP-hard [4], characterized by search spaces that increase exponentially with the size of the input. They are therefore intractable to solution using analytical approaches or simple deterministic algorithms that cannot terminate in polynomial time. Heuristic and stochastic optimization procedures offer more appropriate alternatives.

At about the same time as the advent of the first generation wireless systems, three robust and general global search techniques for NP-hard combinatorial optimization were invented - Genetic Algorithms (GA) (Holland, 1975 [5]), Simulated Annealing (SA) (Kirkpatrick et al., 1983 [6]), and Tabu Search (TS) (Glover, 1986 [15]). These techniques have proved to be very successful and their application to a large number of fields such as industrial production, management, financial services, game theory, telecommunications, graph theory, biological modeling, and VLSI. has been increasing steadily from the mid 80's through the 90's. Their application to the field of mobile telecommunications is still in its infancy, with most of the 20-odd papers that have been written on the subject having been published in the last five years. This chapter provides a fairly comprehensive survey of this still-nascent literature on the subject.

The twin goals of this chapter are a) to familiarize mobile communication engineers with global search techniques and their potential, and b) to provide those interested in these techniques with an idea of the nature and scope of their application to mobile communications. Accordingly, the rest of the chapter is organized as follows: the three global search techniques are described in section 2; section 3 provides a survey of the recent papers where these techniques have been applied to optimization for mobile communications; concluding comments are presented in section 4.

2 Global Search Techniques [1]

An optimization problem seeks to find the optimal solution $x^* = min\{f(x)|x \in S\}$, where f is the cost function (also known as the objective function) with domain S - the set of possible solutions. A combinatorial optimization problem is defined as an optimization problem where S has discrete members. When the problem complexity is low, either an exhaustive search of the space, or deterministic algorithms (such as Linear and Nonlinear Programming [49], Dynamic Programming [50], Branch-and-Bound algorithms and Polyhedral Cutting Plane approaches [51]) may be used to obtain the solution. For more difficult problems, heuristic and stochastic search techniques must be employed to find the optimal point in a large solution space.

A subset of S, designated $N(x)$, may be associated with each point $x \in S$. $N(x)$ is referred to as the neighborhood of x. Most search techniques of optimization operate by starting with some point x and exploring its neighborhood for solutions. Local search techniques such as steepest descent algorithms explore a neighborhood accepting each successive point as a solution only if it has a lower cost than the current solution. This causes entrapment in the point with the lowest cost function in the neighborhood - a local optimum. Global search techniques provide an escape from such traps by providing the ability to selectively accept successive points even if they have a higher cost than the current solution. We examine, in turn, Genetic Algorithms, Simulated Annealing and Tabu Search. These algorithms are very simple and easy to implement (perhaps the chief reason for their popularity), and are very general and robust in their simplest form - assuming nothing about the structure of the space being searched. To improve their efficiency when applied to specific problems, they may be modified suitably or even hybridized with heuristics-based local search techniques.

[1] There is some confusion regarding nomenclature in the literature. These techniques are sometimes referred to as "local" search techniques (e.g. [17]) because they proceed by searching neighborhoods in the search space. In this chapter, we distinguish between local search procedures (such as steepest descent algorithms) that are susceptible to being trapped in local optima, and global search procedures such as Genetic Algorithms, Simulated Annealing and Tabu Search, that are capable of escaping such minima and providing globally optimal solutions.

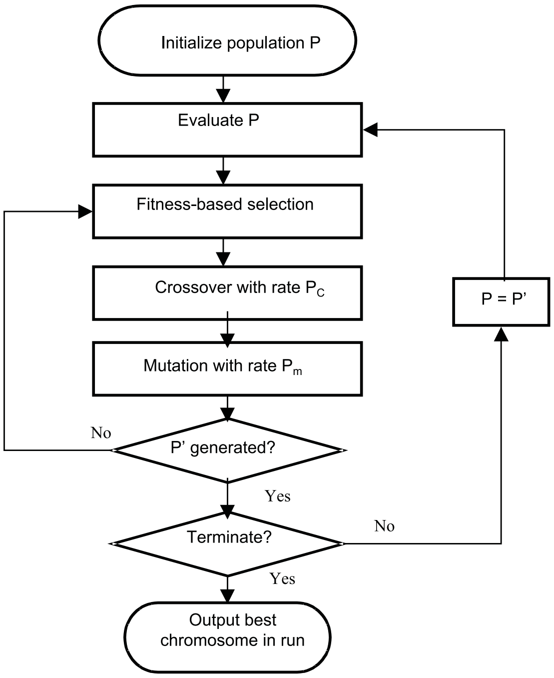

Figure 1: Genetic Algorithm

2.1 Genetic Algorithms

These algorithms derive their inspiration from the natural process of biological evolution. Solutions are encoded (often in binary) into strings or chromosomes. The algorithm operates on a population of these chromosomes, which evolve to the required solution through operations of fitness-based selection, reproduction with crossover and mutation that are fundamentally similar to their natural analogs. Figure 1 shows a diagram for the simple Genetic Algorithm. Using Markov chain modeling, it has been shown that GA's are guaranteed to converge asymptotically to the global optimum if an elitist strategy is used where the best chromosome at each generation is always maintained in the population [8]. More detailed descriptions of these algorithms, their implementation and applications can be found in [9], [10], [11].

Briefly, the process is as follows: an initial population P of N chromosomes is chosen. At each generation, a new population P' is generated by repeating the following procedure: first, two parents are picked from P in such a way that the probability of selection is proportional to their fitness. Then, new offspring are generated from these parents by crossover with some probability P_c. Finally, with a probability P_m (often chosen to be very low: 0.001 - 0.01), these recombined chromosomes are then randomly mutated (for binary GA this refers to complementing each bit with probability P_m). The population thus generated replaces the current population. The procedure is repeated until some termination criterion is satisfied. Termination criteria may be defined such that the algorithm stops when an acceptable solution is found in the population or after a predefined number of function evaluations or generations have been processed.

Very often, for specific applications, the crossover and mutation operators may be redefined or extended, and a local search may be incorporated at the end of each generation to speed up convergence. The encoding of the solution space into chromosomes, the size of the population, and the mutation and crossover probabilities all vary widely depending on the implementation.

Genetic Algorithms have been quite successful in a variety of applications and enjoy massive popularity, at least amongst members of the Evolutionary Computation community. Among the papers surveyed in this chapter, applications of Genetic Algorithms are by far the most numerous. The name *evolutionary telecommunications* has recently been suggested for this growing field.

2.2 Simulated Annealing

This algorithm derives its inspiration from the thermodynamic process by which solids are heated and cooled gradually (annealed) to a crystalline state with minimum energy. Simulated annealing operates on a single point (not a population of points as in GA), and at each step a point x' in $N(x)$ is generated from the current point x. If the point has a lower cost function than x it is accepted unconditionally, but even if it has a higher cost it is accepted probabilistically using the Metropolis criterion described below. This acceptance probability is proportional to the temperature T of the annealing process, which is lowered gradually as the algorithm proceeds. The procedure is illustrated in Figure 2.

The Metropolis criterion is as follows: for $x' \in N(x)$, the probability that x' is selected is

$$P_{x \to x'} = min\left(1, \exp\left[-\frac{f(x') - f(x)}{T}\right]\right) \tag{1}$$

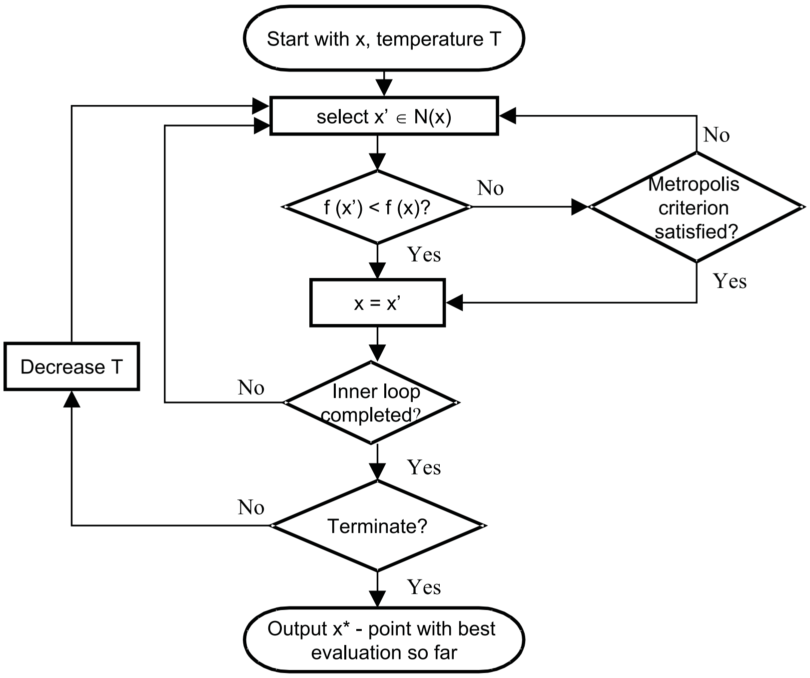

Figure 2: Simulated Annealing

When T is high initially, there is a greater probability of making uphill moves, which allows the search to fully explore the space. It has been shown, by modeling SA as Markov processes [12], that Simulated Annealing will converge asymptotically to the global optimum under two conditions:

- Homogeneous condition: if the temperature is lowered in any way to 0, while the length of the homogeneous Markov sequence (formed by the accepted points) at each temperature is increased to infinite length.
- Inhomogeneous condition: if irrespective of the length of these isothermal Markov chains, the cooling schedule is chosen such that T approaches 0 at a logarithmically slow rate.

Since in practice neither of these is possible in finite implementations, polynomial time approximations are used. The choice of cooling schedule and the length of the Markov chains at each temperature affects the quality of the results and the rate of convergence. The definition of the neighborhood function (which is usually based on some heuristic understanding of

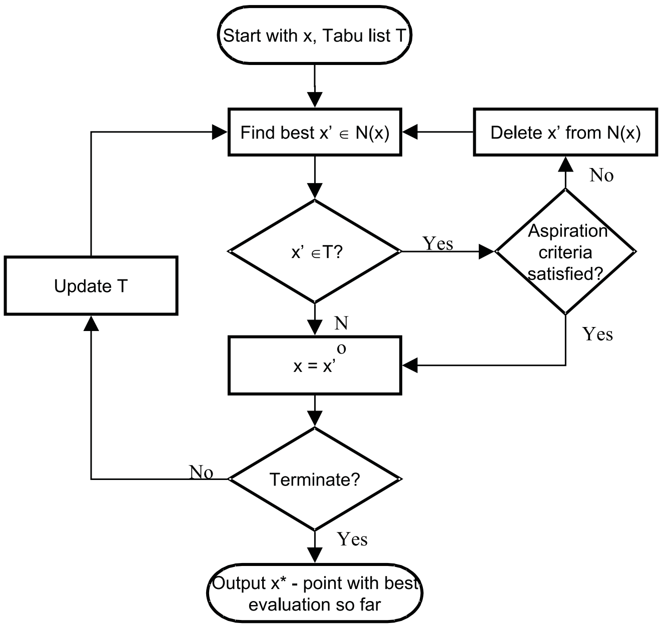

Figure 3: Tabu Search

the problem at hand) determines how new points are visited. The SA program is terminated if an acceptable solution is found or if a designated final temperature is reached.

Simulated Annealing also has quite a dedicated following and has been very successful in a broad range of NP-hard optimization problems. Quite a few papers surveyed in this chapter show the application of SA to mobile communications.

2.3 Tabu Search

Tabu Search is based on the premise that intelligent problem solving requires incorporation of adaptive memory [13], [14], [15]. Unlike GA's and SA, Tabu Search is not a stochastic search technique; it is a non-random meta-heuristic algorithm for combinatorial optimization , but like the other two techniques it also provides means for escaping local minima. Figure 3 illustrates how TS works.

In TS, a finite list of forbidden moves called the tabu list T is maintained.

At any given iteration, if the current solution is x, its neighborhood $N(x)$ is searched aggressively to yield the point x' which is the best neighbor such that it is not on the tabu list. Note that is not required that $f(x') \leq f(x)$, only that $f(x') = min(f(x^+)|x^+ \in N(x) \setminus T)$. As each new solution x' is generated, it is added to the tabu list and the oldest member of the tabu list is removed. Thus the tabu list prevents cycling by disallowing repetition of moves within a finite number of steps (determined by the size of the list). This, along with the acceptance of higher cost moves, prevents entrapment in local minima. It may also be desirable to include in the tabu list attributes of moves rather than the points themselves. Each entry in the list may thus stand for a whole set of points sharing the attribute. In this case, it is possible to allow certain solutions to be acceptable even if they are in the tabu list by using what are called *aspiration criteria*. For example, one such criterion is satisfied if the point has a cost that is lower than the current lowest cost evaluation. If a neighborhood is exhausted, or if the generated solutions are not acceptable, it is possible to incorporate into the search the ability to jump to a different part of the search space (this is referred to as *diversification*). One may also include the ability to focus the search on solutions which share certain desirable characteristic (*intensification*) by performing some sort of pattern recognition on the points that have shown low function evaluations in the past.

Tabu search is a meta-heuristic technique, and it must be adapted to the problem at hand for it to be efficient. The choice of moves that generate the neighborhood of a point is problem-specific. Different implementations can be generated by varying the definition and structure of the tabu list (for example by deciding how tabu attributes are determined), the aspiration criteria, intensification and diversification procedures etc. To speed up the search, faster ways to determine the best neighbor are required.

Like the other two global search techniques, TS has also been applied successfully to a large number of NP-hard optimization problems and has been shown to compare favorably with GA and SA [17]. A search of the literature, however, has not revealed extensive application of TS in mobile communications though there is certainly no *a priori* reason why it cannot be applied to the same problems as GA and SA.

3 Applications to Mobile Communications

In this section, We review papers describing the applications of these global search techniques to optimization problems in mobile communications. The number of problems in mobile communications that have been formulated and recognized as hard combinatorial optimization problems and processed with these algorithms is as yet quite small. Thus, while the papers discussed here by no means cover the entire range of possible applications of these techniques to mobile communications, they do represent nearly all such attempts to date, and should provide a good overview of the subject.

It will be noted that most of the papers surveyed here are concerned with the application of Genetic Algorithms. This is not the result of any inherent superiority in terms of efficiency or ease of programming in using GA's as opposed to SA and TS, but perhaps an indication of its general popularity among researchers interested in optimization for wireless telecommunications.

Several papers discussing global optimization for system-level resource allocation, design, and planning in cellular mobile systems are described. Papers covering the use of these techniques for lower-level problems such as optimal multi-user detection in CDMA technology, design of frames in TDMA schemes, blind channel identification and equalization are also described.

3.1 Design of Fixed Network Topology

In cellular communication networks, the mobile units are connected to the public networks, such as PSTN, ISDN and other data networks, through a network hierarchy determined by the system architecture. Figure 4 shows such an architecture for the European GSM standard [16]. The mobile stations (MS) in each cell communicate over the radio interface with the Base Transceiver Stations (BTS). BTSs connect to Base Station Controllers (BSC) via microwave links or dedicated leased lines on what is called the *Abis interface*. Each BSC may control several hundred BTSs, and some low-level functions such as mobile handoffs are made by the BSC. The BSCs in turn are connected to the Mobile Switching Centers (MSC) via the *A interface*. The MSC controls traffic among all the BSCs and there is a sub-network between MSCs at this level that includes units such as the Home Location Register (HLR), Visitor Location Register (VLR), the Authentication Center (AuC),

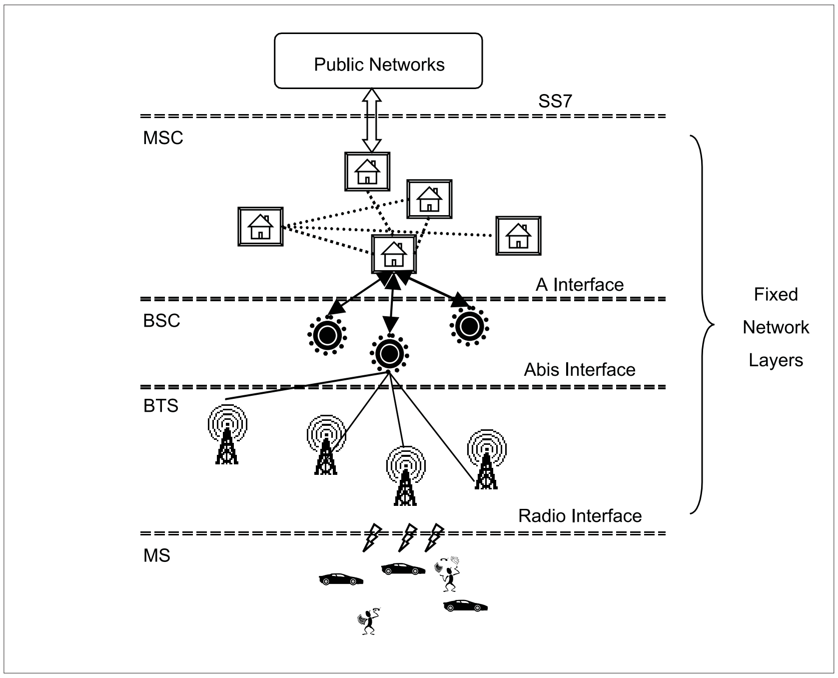

Figure 4: GSM Network Architecture

and the Point of Interconnect (POI) to the public networks. Thus there is a large part of the communication hierarchy in a mobile cellular system which is a fixed wired network.

The cost of the topology of a fixed network depends upon several factors, including the cost of the nodes, the cost of links, link flow and capacity, and constraints such as the maximum number of links for each node. The problem of designing minimum cost network topologies is closely related to the minimum spanning problem in graph theory and is often very complicated; an exhaustive enumeration of topologies to obtain the optimal arrangement becomes infeasible for even moderately sized networks. Network topology design is perhaps the most studied application of global search techniques to telecommunications (for example [18],[19], [20], [21]).

The design of the fixed portion of the GSM network and the closely related Digital Cellular System 1800 (DCS1800) is performed in [22] using a GA-based Genetic Optimizer for Topological Network Design (GOTND).

The cost function to be minimized is defined as follows:

$$f = \sum_{\forall n} C_n^{NODE} + \sum_{\forall p} C_p^{POI} + \sum_{\forall l \in L^{BTS\to BSC}, L^{BSC\to MSC}, L^{MSC\to MSC}} C_l^{LINK} \qquad (2)$$

subject to (i) $F_l \leq C_l, \forall l$ and (ii) $\xi \leq 0.001$, where C_n^{NODE} is the cost of all nodes of type n, C_p^{POI} is the cost of the pth POI, and C_l^{LINK} the cost of the l^{th} link which is one of types $L^{BTS\to BSC}$, $L^{BSC\to MSC}$, or $L^{MSC\to MSC}$; F_l, C_l represent the flow and capacity of each link; and ξ is the call blocking probability.

In this paper each candidate solution is represented by a set of seven chromosomes. The first four chromosomes represent the x and y coordinates for the BSCs and MSCs. The last three chromosomes describe the existence of links between BTSs and BSCs, between BSCs and MSCs and between MSCs themselves. The standard fitness based selection mechanism is used. Instead of the generic crossover and mutation operations, however, the GOTND employs 17 problem-specific operators. These include "Move one/some BSCs randomly", "move one/some MSCs randomly", "move MSC and BSC coupled randomly", "move MSC to BSC", "push BSC toward all connected BTS(s)", "push BSC(s) toward connected MSC" etc. among others. Different weighted combinations of these operators are tested to determine those that yield a fast convergence rate. Thus the operators that are more successful are used more often than less successful operators during the optimization.

In [22] the GOTND is applied to an example scenario with 100 BTSs, 4 BSCs and one MSC that is also the POI to the public telecommunication networks. The cost data for the links is assumed to be piecewise linear and was obtained from German Telecom. The results showed that a cost reduction of 19% over the best network in the first iteration could be achieved using the GOTND.

3.2 The Channel Allocation Problem

One of the most basic limitations of mobile radio communication systems is the restricted spectral bandwidth. Fundamental to the cellular concept is the idea of frequency reuse by which the same frequencies/channels can be reused in different geographical locations [2], [16]. Each such location is referred to as a cell (usually hexagonal in shape), and is allocated a set of channels according to the expected demand in that cell. The entire spectrum

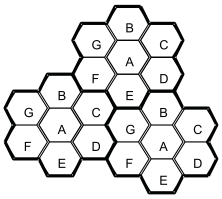

Figure 5: Frequency Reuse in 7 cell clusters

is allocated to a cluster of cells arranged in shapes that allow for uniform reuse patterns. The geometry of the hexagonal cells restricts the number of cells per sector N to the discrete values

$$N = i^2 + ij + j^2 \tag{3}$$

where i and j are integers. Thus clusters can only accommodate 1, 3, 4, 7 cells. Figure 5 shows an illustration of the frequency reuse in a 7-cell cluster system.

The allocation of the limited number of channels to each cell in a mobile communication system is a much-studied problem [24]. The channels must be allocated in such a way as to satisfy

- Co-channel interference constraints : These constraints are due to radio interference between cells that have the same channels,
- Co-cell (or adjacent channel) constraints : These constraints arise because of the radio interference between adjacent channels in the same cell that are not separated by some minimum spectral distance, and
- Cell demand requirements: These indicate how many channels are required in each cell due to their unique traffic patterns.

It has been shown that the channel allocation problem (CAP) is equivalent to the generalized coloring problem in graph theory, a problem that is known to be NP-complete [35]. There are essentially two kinds of allocation schemes - Fixed Channel Allocation (FCA) and Dynamic Channel Allocation (DCA). In FCA the channels are permanently allocated to each cell, while in DCA the channels are allocated dynamically upon request. DCA is desirable, but under heavy traffic load conditions, FCA outperforms most known DCA schemes [25]. Since heavy traffic conditions are expected in future generations of cellular networks, efficient FCA schemes become more important.

All of the papers surveyed that apply global search techniques to CAP are FCA schemes [26],[27], [28],[29], [30], [31].

Let us assume that there are C channels to be assigned to N cells. Typically, the interference constraints are modeled by an N-by-N compatibility matrix D. The diagonal elements of this matrix, $D_{x,x}$, represent the co-cell constraint - the number of frequency bands by which adjacent channels assigned to cell i must be separated . The non-diagonal elements $D_{x,y}$ represent the number of frequency bands by which channels assigned to cells x and y must differ. If the compatibility matrix is binary, then these constraints are expressed more simply - if the same channel cannot be reused by cells x and y then $D_{x,y} = 1$, and if it can be reused then $D_{x,y} = 0$.

The traffic requirements for the cells are modeled by a demand vector T of length N that represents the number of required channels in each of the cells. The assignment to be generated is denoted by an N x C binary matrix A which is chosen such that

$$A_{x,k} = \begin{cases} 1 & : \quad if\ cell\ x\ is\ assigned\ channel\ c_k \\ 0 & : \quad otherwise \end{cases} \tag{4}$$

In general, the cost due to the violation of interference constraints in this problem is given as:

$$f' = f_{co-cell} + f_{cochannel} = \alpha_1 \sum_{x}^{N} \sum_{i \neq j}^{C_x} \sum_{j}^{C_y} \Phi(i,j) + \alpha_2 \sum_{x \neq y}^{N} \sum_{y}^{N} \sum_{i}^{C_x} \sum_{j}^{C_y} \Psi(x_i, y_j) \tag{5}$$

where $\Phi(i,j) = \begin{cases} 0 & : \quad iff\ |c_i - c_j| < D_{x,x} \\ 1 & : \quad otherwise \end{cases}$ is a measure of the co-cell constraint satisfaction, and $\Psi(x_i, y_j) = \begin{cases} 0 & : \quad iff\ |x_i - x_j| < D_{x,y} \\ 1 & : \quad otherwise \end{cases}$ is a measure of the co-channel constraint satisfaction, x_i and y_j the assigned frequencies for the i^{th} and j^{th} channels of cells x and y respectively , C_x the number of channels in the x^{th} cell, and α_1, α_2 are constants to weigh the relative importance of the two constraints. The cost due to the violation of traffic demand requirements can be modeled explicitly as an error term:

$$f_{traffic} = \sum_{x}^{N} (T_x - \sum_{k} A_{x,k})^2 \tag{6}$$

The cost function to be minimized can be expressed as:

$$f = f' + f_{traffic} \tag{7}$$

If the traffic demand requirements are incorporated implicitly by only considering those assignments which satisfy them, then the cost function only consists of the interference-constraint violation term:

$$f = f' = f_{co-cell} + f_{cochannel} \tag{8}$$

subject to $\sum_k A_{x,k} = T_x, \forall x$. Some of the papers that describe the application of global search techniques to this problem are described below:

i) In [26], a Genetic Algorithm is used to determine an optimal channel assignment. In the encoding chosen in this paper for the GA, chromosomes whose total length is the sum of all channels required for each cell represent each possible allocation. The traffic demand is therefore incorporated into the representation. A typical chromosome consists of a linear arrangement of the channels for each cell listed in order. The standard mutation operator is chosen, while a slightly modified partially matched crossover (PMX) operator is designed that performs a crossover while resolving (correcting) any channel constraint-violations that may arise from the crossover to improve the performance of the algorithm. The algorithm was tested on a homogeneous cellular network of 49 cells where only three channels are available, and on data taken from an actual inhomogeneous cellular network consisting of 25 cells and 73 channels (known to be sufficient). In both cases the algorithm was able to generate conflict-free allocations. In the case of the inhomogeneous network example, the setting of a higher value of α_2(i.e. emphasizing co-channel constraint satisfaction over co-cell constraint satisfaction) was empirically observed to speed up convergence.

ii) Genetic Algorithms are also used in [27] for frequency planning. A binary compatibility matrix is used to model the co-channel constraints. The paper presents the performance of two versions of genetic algorithms - the simple GA with standard operators and binary representation, and a hybrid GA where an integer representation is used for the channels. For the simple GA, the demand requirements are explicitly incorporated into the cost function as in Equation (7). For the hybrid GA a representation which has the demand vector built into it is chosen. Thus a modified fitness function is chosen consisting only of the co-channel constraint satisfaction term. The hybrid GA uses uniform crossover instead of 1-point crossover (referring to the number of loci where recombination takes place). Furthermore, a steepest descent (local search) algorithm is incorporated at the end of each generation to perform a more thorough search of the solution space. The neighborhood for this local search is defined by selecting a different channel for each cell in conflict in a particular assignment and choosing the best such allocation.

The two algorithms were tested on cellular network scenarios with uniform and non-uniform traffic loads for two kinds of cellular systems - the standard planar system (with 12 channels and 21 cells) and a linear system (12 channels and 9 cells laid out side-by-side). It was observed empirically that the hybrid GA performs better than the simple GA, which, for a fixed upper limit on number of generations (around 200), was unable to resolve conflicts in all but the simplest cases.

iii) Yet another attempt at using GA for fixed channel assignment is described in [28]. The authors describe a modified genetic-fix algorithm that utilizes an encoding technique called the minimum-separation encoding scheme. The number of 1's in each row of the binary assignment matrix A corresponds to the number of channels allocated to the corresponding cell. To satisfy the demand requirement this would normally be constant. Each chromosome is a binary string that represents the matrix A through a concatenation of its rows. The genetic-fix algorithm defines mutation and crossover operators for a binary chromosome in such a way that the number of ones (the "weight" of the vector) is always preserved. The binary encoding is further compressed by taking into account the co-cell constraint that no two channels in the x^{th} cell can be closer than $d_{min} = D_{x,x}$. This would imply that each 1 in each row of the A matrix must be followed by $(d_{min} - 1)$ zeros. The minimum-separation encoding scheme takes advantage of this fact by eliminating $(d_{min} - 1)$ zeros following the 1. Thus the bit string <1000100100> is represented by <1011> when $d_{min} = 3$. This compression reduces the search space further. The implicit encoding of the co-cell constraints and demand requirements in this manner means that the cost function to be minimized only contains the co-channel constraint violation cost.

The genetic fix algorithm was tested on a suite of 5 problems with varying numbers of cells (4, 21 and 25), and numbers of channels (ranging from 11 to 309), along with different compatibility matrices and demand vectors. The results of this algorithm were compared with those from a paper [29] that applied a neural network to the first four of these problems using the frequency of convergence (the ratio of the number of successful results to the total number of runs) as a criterion. The genetic-fix algorithm was found to outperform the neural network in all cases. The fifth problem was for a 21-cell system where the algorithm was able to find a conflict free assignment with fewer channels than previously reported in the literature.

iv) Simulated Annealing is applied to the channel allocation problem in [30]. In the cost function used by the authors, only co-channel interference is considered, and the traffic demand term is explicitly specified. The initial temperature T_o for the annealing was chosen by starting with $T = 0$, and increasing it until the ratio χ of accepted to proposed transitions was between 0.7 and 0.9. The cooling schedule is chosen such that

$$T' = T \cdot \exp\left(-\frac{\lambda T}{\sigma}\right) \tag{9}$$

where is a parameter with value between 0 and 1, and is the standard deviation of the cost at level T. The transitions used to define the neighborhood of an acceptable solution include the basic "flip-flop" (add one channel to a cell, and remove one - preserving the number of channels assigned), and the dense-packing move (find all nearest co-channel cells, pick the channel that is most used in these cells and switch it on at the current cell; randomly select and turn off one of the other channels currently in use in this cell). To further enhance the ability of the annealing algorithm to evade local minima, the authors allow multiple flip-flops in one move (this permits greater movement in the solution space). The paper further discusses how additional constraints imposed on this basic allocation scheme, such as soft interference constraints, hard and soft pre-allocated channels, and minimized spectrum demand, may be incorporated into the cost function.

The Simulated Annealing algorithm is tested on a 7-cell cluster scheme for an inhomogeneous network with 239 cells, 38 channels, some pre-allocated channels and varying demand. Simple (only flip-flop moves) and sophisticated (flip-flop, dense-packing and multiple flip-flops) neighborhood schemes are compared in terms of convergence percentage, final cost, and computing time. As expected, the sophisticated schemes perform significantly better, demonstrating the importance of having a good neighborhood definition.

v) Simulated Annealing and Tabu Search are compared for the CAP in

[31]. The cost function to be minimized is given as

$$f'' = \sum_{j \in C} max_{x \in N}\{A_{x,j}\} + \frac{1}{\Psi} f \tag{10}$$

where f is as defined in Equation (8) with the traffic demand defined implicitly. The first term in the expression attempts to minimize the bandwidth allocated. Ψ is a small positive number that gives a larger weight to the second term that represents the interference constraints.

For the SA, the initial temperature is chosen as in [30] by using the acceptance ratio χ. The cooling schedule is taken from [12]:

$$T' = \frac{T}{1 + \frac{T \ln(1+\delta)}{3\sigma}} \tag{11}$$

where δ is a distance parameter chosen to be 0.5 in this paper. For the TS algorithm described in the paper, two tabu lists are maintained - one for the cell (of size 7), one for the channels (size 1). An aspiration criterion is used to allow even tabu moves if they are found to yield lower cost than the current best cost. The dense packing and basic flip-flop moves (described before) are used with equal weights to determine the neighborhood for both SA and TS algorithms.

Applying the algorithms to examples of various sizes (9 to 49 cells with traffic demand between 1 and 10), the authors observe that Simulated Annealing obtains fairly good solutions, while the Tabu Search is extremely ineffective. It took longer than 24 hours on a SUN SPARC 10 workstation for problem instances with 16 cells or more, and on the problem instance of 9 cells it did not match the performance of the SA algorithm in terms of the minimum number of channels used. It is, however, entirely possible that the dismal performance of the TS in this experiment may have been due to an inefficient implementation of the algorithm.

vi) The speculation regarding the poor results of TS in [31] is given some credence by the results in [32] where the use of Tabu Search for frequency assignment is investigated more thoroughly. The formulation of the constraints and the expression for the cost function used in this paper are identical to those given in Equation (5). To minimize the number of frequencies (NF) used in a channel assignment, the Tabu Search is carried out at an initially high value of NF. If a conflict-free assignment is found then the value of

NF is decreased by 1, and this is repeated until one can no longer find such a conflict-free assignment.

The co-cell constraints are incorporated into a candidate assignment x given as follows:

$$x = \{c_{1,1} \dots c_{1,T_1} \dots c_{x,1} \dots c_{x,T_x} \dots c_{N,1} \dots c_{N,T_N}\} \tag{12}$$

such that $\forall x \in 1 \dots N, \forall i, j \in 1 \dots T_x, |c_{x,i} - c_{x,j}| \geq D_{x,x}$, where $c_{x,j}$ is the j^{th} channel allocated to the x^{th} cell. This assignment satisfies the adjacent channel interference constraint by definition.

A neighbor x' of a solution point x can be obtained by changing the value of any channel that has been allocated in x such a way that the new value always satisfies the co-cell constraint. Further a candidate list of new neighbors V^* is specified as follows:

$$V^* = \{x' \; in \; N(x) | x' \; and \; x \; differ \; at \; the \; value \; of \; conflicting \; frequency\} \tag{13}$$

The size of this candidate list varies during the search, and helps the search to concentrate on influential moves while reducing the size of the neighborhood. Since only a part of the frequency assignment is changed during each move, the speed of evaluation is reduced by differentially updating the cost as each move is made. The size of the tabu list, k, is made to be proportional to the length of the candidate list at each iteration, $k = \alpha|V^*|$ and thus varies as the algorithm proceeds. Again, the simple aspiration criterion is chosen where the tabu status of a move is cancelled if it leads to a solution better than the best solution encountered so far.

The TS algorithm thus designed was applied on a suite of real-sized problems taken from data provided by the French National Research Center for Telecommunications and compared with a local search algorithm (steepest descent) and the best known results obtained with other methods including Simulated Annealing. For each TS run of 100,000 iterations, the steepest descent algorithm was run 20 times for 5,000 iterations. The local search algorithm repeatedly yielded sub-optimal results while the TS was able to find near-optimal solutions quite speedily for most of the instances selected. In comparing their Tabu Search algorithm with the Simulated Annealing results from [33], the authors showed that the TS algorithm was capable of matching, even outperforming SA in locating the minimal number of frequencies for channel allocation. The TS algorithm was also observed to be faster than the SA. Based upon further experimental evidence, the authors credit the performance of the algorithm to the usage of the conflict-based candidate list strategy and the incorporation of co-cell constraints into the candidate solution points (neither of which was done in [31], where TS did not fare well).

3.3 Optimal Base Station Location

One of the most challenging design problems in the planning and deployment of a mobile radio communication network is deciding on the optimal locations for base stations. Currently the location of base stations is often done in an *ad-hoc* manner via manual inspection of maps showing propagation properties of the area to be serviced. Given a list of potential sites in a service area where base stations may be located, the goal is to use knowledge of the radio propagation characteristics of the area to select sites in such a way as to minimize their number while maximizing coverage in the area. This would reduce network cost and complexity and be a great improvement upon the current methodology for setting up micro-cellular networks. The problem is very similar to the minimum dominating set (MDS) problem from graph theory, which is known to be NP-hard [4].

To perform such an optimization, the cost function necessarily involves the radio propagation characteristics of the area. This may be determined using sophisticated ray-tracing software (which could be a very time-consuming process), or by using empirical propagation models for path loss. Another issue in the design of the cost function is the trade-off between coverage and number of base stations. The more base stations there are, the greater coverage there is, but there is also correspondingly greater radio interference and network cost.

i) Genetic Algorithms are applied to this task in [34]. It is assumed that a list of N possible locations is known beforehand, such that if there were base stations at all those locations, 100% radio coverage is guaranteed. A binary encoding of the problem is chosen as follows: each chromosome consists of N bits, with a one at each bit if there is a base station at the location corresponding to that bit, and a zero otherwise. The simplest GA is first evaluated for the fitness function chosen as follows:

$$f = \frac{R^{\alpha}}{n} \tag{14}$$

where R is the radio coverage (the percentage of locations covered by the base stations selected), and n is the number of base stations that are present. α is a parameter that is tuned to favor coverage with respect to the number of transmitters and is assigned a value of 2 in this paper.

Experimental tests of this algorithm were performed on a 70 km x 70 km region in France for $N = 150$, with the radio coverage from each possible location being computed by a radio wave propagation simulation tool. A solution with 52 base stations covering 80.04% was found, but since the algorithm was found to be very slow it was parallelized using the parallel islands model of GA. In this version of the GA, the population is distributed into sub-populations (islands) where they evolve quasi-independently towards the solution, and some individuals in each generation are migrate from one island to another. A nearly linear speedup was observed using this technique and the resultant computational power enabled them to discover a good solution with only 41 base stations that covered 79.13% of the initial covered surface.

ii) The optimal base station location problem for micro-cells in an urban city environment is solved using Simulated Annealing in [36]. An empirically determined model for path loss in urban areas given in [37], which takes as parameters the heights above ground of the mobile and base station antennas, was used to calculate coverage. While the authors concede that the assumptions made regarding path loss are not necessarily valid in reality, they are sufficient to demonstrate the applicability of SA to this problem.

The service area is modeled as a Manhattan-like grid of streets dividing the area into square city blocks (10m x 10m). A power goal is set such that at each grid point i a certain minimum level of radio reception P_i^* is guaranteed. The cost function is then defined as follows:

$$f = \sum_{i \in A} (P_i^* - P_i)^2 \tag{15}$$

where A is the set of all grid-points where the actual received power P_i is less than the minimum requirement P_i^*.

Each possible solution is a location x. The neighborhood is defined by allowing the base-stations to make random moves while satisfying any constraints on the location such that the current temperature T at each stage defines the magnitude of these moves. The initial temperature T_o is chosen to be 80m, the cooling schedule is such that $T' = 0.8T$, and at each temperature 100 moves are allowed. The algorithm is stopped when the difference between the maximum and minimum costs at a given temperature is equal to the maximum change in an accepted move at the same temperature. Since the units of cost and temperature are different, the metropolis criterion is modified as follows:

$$P_{x \to x'} = min\left(1, \exp\left[-\frac{(\frac{f(x')-f(x)}{f(x)})}{(\frac{T}{T_o})^2}\right]\right) \tag{16}$$

This SA technique was tested on two problems - one involving the location of a single transmitter in a 4 x 4 grid. In this simple problem instance the optimum location could be determined by an exhaustive analysis of all possible grid locations. In all runs (with different initial starting positions) the optimization arrived at a final position within a few meters of the optimum location. Another problem with an 8 x 8 grid and base stations was performed and a figure showing the solution and resultant coverage is presented in the paper. The authors observe that different runs yield different base-station locations with the approximately the same overall costs indicating that near-optimum solutions are not unique.

3.4 Mobility Management

As mentioned before, one of the goals of PCS is to ensure that users are able to communicate with each other wherever they may be. Users should be able to roam at will, crossing cell boundaries system-wide while they are mobile without losing an ongoing call or being prevented from initiating or receiving calls. This necessitates mobility management, since the system needs to track the location of the user in order to provide service. For mobility management purposes, the system is divided into registration areas (RA) consisting of several cells, as seen in Figure **??**. The user's location is maintained in a location database, such as the Visitor Location Register in GSM mentioned before, and this database is updated each time users enter or leave the RA. This update is referred to as a location update or LU. When a mobile station is called, its current RA is determined from such a database, and all base stations in the RA broadcast paging messages over a reserved forward control channel (FOCC) to inform the mobile unit of this call request.

The LUs result in increased network traffic and also place a load on the distributed location databases, resulting in greater complexity of the databases' implementation. The paging messages also consume radio resources and their traffic is often limited by the bandwidth of the FOCC (a

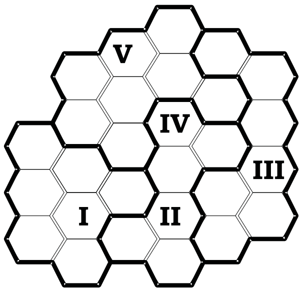

Figure 6: Registration Areas in a Cellular System

condition referred to as paging bound). It is thus desirable to design the system in such a way as to reduce both LU and paging costs. There is a tradeoff between the LU and paging costs which varies with the size of the RA: If the RA is large, there are fewer inter-RA crossings resulting in a lower LU cost, but the number of base stations that need to be paged increases correspondingly. Genetic Algorithms have been used to design optimal RA to reduce LU cost while maintaining paging bound constraints in [38]. Within each RA, only one mobile station responds to the multiple pages transmitted by base stations in the RA. To reduce the paging cost, it is possible to subdivide the RA into paging zones ordered based on a pre-determined probability of locating the mobile user at different locations within the RA. The optimal planning of such paging zones using GA is treated in [39].

i) The RA planning in [38] is based on the assumption that the cell planning has already been done and the LU and paging traffic have been estimated for each cell. It is assumed that the paging bound B for each RA is fixed. The RAs are planned so that they each contain a disjoint set of cells. The task is to design the RAs to minimize the following cost function:

$$f = \frac{\alpha_1}{2} \sum_{i=1}^{M} LU_i + \alpha_2 \sum_{i=1}^{M} max(0, (P_i - B)) \qquad (17)$$

where P_i and LU_i are the paging traffic and LU traffic respectively in the i^{th} registration area, and M is the total number of registration areas in the system. The second term in Equation (17) is non-zero if the paging bounding is exceeded in the i^{th} registration area due to an increased number of cells in it . and are constants used to weigh the relative importance of the LU cost and the paging cost.

A binary representation chosen for the chromosomes is based on the cell borders. The borders are numbered sequentially and the corresponding bit is one if that particular cell border is to be a border between two adjoining RAs, and zero otherwise. This representation facilitates the evaluation of the location update cost. The first term in Equation (17) can be rewritten for this border notation as follows:

$$f = \frac{\alpha_1}{2}\sum_{j=1}^{n} v_j w_j + \alpha_2 \sum_{i=1}^{M} max(0, (P_i - B)) \tag{18}$$

where n is the total number of borders, w_j the crossing intensity of the j^{th} border and v_j the value of the j^{th} bit of the chromosome being evaluated (i.e. a determination of whether the j^{th} border is an inter-RA border). It is also shown in the paper how additional terms may be incorporated into the cost function to deal with useless location updating (when no calls arrive between successive LUs) and presets that arbitrarily force certain borders in the chromosome to be 0 or 1. The standard GA operators of mutation and crossover are used. The mutation rate (initially 0.02) is halved after several generations to increase convergence. The GA is terminated after 1000 generations.

The GA is compared with hill-climbing for five systems with varying size (19 to 91 cells) where the paging cost for each cell and crossing intensity for each border were generated with a normal distribution. The result of RA planning over 100 runs shows that GA's consistently outperform hill-climbing providing improvements ranging from 10-30%. The authors conclude that GAs appear to be a valuable approach to the planning of registration areas in PCS networks.

ii) A tracking strategy utilizing the mobility patterns of individual mobile stations is used in [39] to plan the partition of RAs into paging zones to minimize paging cost. First a multilayered model is developed based on mobile-phone usage for different times of activity during a typical workday for each user - work, home, social. Each mobile user's activity is monitored for each such layer to develop statistical mobility patterns that describe the likelihood of locating each user in a particular cell during a particular time of the day. Then the cells in the RA are partitioned into paging zones for each user k, and layer j such that each zone consists of cells with similar likelihood of locating that user:

$$P_{k,j} = \{P_{k,j,l}\} \tag{19}$$

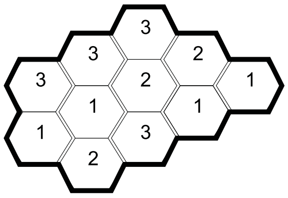

Figure 7: Cells belonging to three Paging Zones within a Registration Area (for a unique user)

These zones are arranged in order of descending likelihood of locating the user denoted by $prob(Pk, j, l)$ - the probability of locating the k^{th} user in activity layer j in the l^{th} zone. Now when a call is received for the user in the RA, the zones are paged in this order - which is more efficient than paging the entire area. The cost function derived in the paper for each paging zone assignment is based on the mobile location probability for each zone in a partition:

$$f = N(P_{k,j,1}) + \sum_{l}(1 - \sum_{i=1}^{l-1} prob(P_{k,j,i})) \cdot N(P_{k,j,l}) \cdot \alpha \cdot \beta \tag{20}$$

where $N(P_{k,j,l})$ represents the number of cells in the l^{th} zone, α the traffic in the FOCC per page, and β the traffic in the MSC per page. Since paging zones are unique for each user, this optimization must be carried out for each user separately.

The encoding chosen for the GA utilizes an integer representation. The cells in the RA are numbered sequentially and correspond to a locus on the chromosome. The value of the chromosome at that locus is the paging zone to which the cell belongs. Thus the string "11231" indicates that the first, second and fifth cells are in zone 1, the third cell in zone 2 and the fourth cell in zone 3. Zone 1 has the highest mobile location probability. The cells for a given paging zone need not be adjacent as seen in Figure 7. The standard genetic operators of crossover and mutation are used. The algorithm is tested for different numbers of paging zones (from 1 to 4) and it is observed that the greater the number of zones, the lower the paging cost. Even a 2-zone partition is observed to have a degree of magnitude lower cost than the traditional area-wide paging scheme.

3.5 Call Management

Another issue of mobile communication system planning is call management. One aspect is this is the determination of a call-admission policy - deciding under what conditions a new call to a mobile in a particular cell should be accepted. Allocating channels to every user whenever they are available (the admit-if-possible or AIP strategy) is not necessarily an optimal strategy because this may result in an inability to accept hand-off calls from a neighboring cell.

i) The problem of evolving an admission call policy for each cell based only on local state information (restricted to the cell and its neighbors) is considered in [40] using Genetic Algorithms. In a mobile communication system it is generally considered better to block a new call than drop an ongoing one. The cellular system performance (which depends upon the call admission policy) is the cost function to be minimized:

$$f = P_b + \omega P_h \tag{21}$$

where P_b is the new call blocking probability and P_h the handoff dropping probability. The factor ω is used to indicate the extent to which dropped calls are considered less desirable than blocked calls.

In each cell, the number of states refers to how many of its C channels are occupied. For a linear cellular system with N cells, the total number of global states is $(C+1)^N$ for each decision. The policy is constructed to consist of three binary decisions: 1) admit a new call 2) admit a handoff call from the left cell and 3) admit handoff from right cell. In a global admission policy there is one decision for each global state (thus $3(C+1)^N$ decisions in all). The states in a global admission policy satisfy the Markov property since each succeeding state depends only upon the previous state of the network. Markov decision process (MDP) based techniques such as Dynamic Programming [50] may be used to solve such a problem but the state space becomes too large for moderate sized networks. If a local policy is considered, and only state information from $2k$ nearest neighbors are used, the state space is $3(C+1)^{2k+1}$. In this case, however, the states are no longer Markov states and MDP techniques are no longer applicable. The state space is still very large: for an example with 9 channels, a policy with $k = 1$ (using information from the current cell and both immediate neighbors in a 1-D linear cellular system) requires 3000 bits (1 x 10903 possibilities).

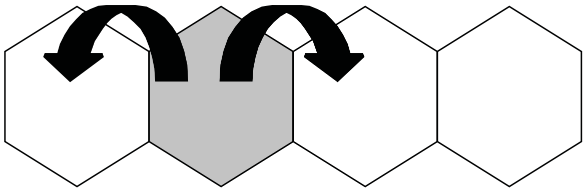

Figure 8: Linear Cellular System N=4, k=1 (2 neighbors considered)

A standard GA is used in [40], where each chromosome represents a local policy, with admit or reject decisions for the new call and handoff requests for each local state of the system. The local policies are evaluated via Monte Carlo simulation using the cost function described in Equation (21). The algorithm was tested on a small 4-cell system for both $k = 0$ and $k = 1$ (seen in Figure 8) to compare it with the optimal solution for a global policy obtained with MDP and the AIP strategy. As the factor w increases, performance is observed to become worse since more global information is needed to better determine handoff possibilities. But in all cases the solutions provided with GA give much better system performance than the AIP strategy. For a larger system with 16 cells and 9 channels the GA (for cases $k = 0$, and $k = 1$) was compared with AIP strategy and the best single-threshold handoff reservation policy (having a threshold t such that if cell occupancy $s > t$, new calls will not be accepted). Again the GA-based solutions (even for $k = 0$) were much better than either of these. Upon examination of the GA-based solutions for $k = 0$, the authors observed that they tended to suggest two-threshold policies with thresholds t_1 and t_2 such that new calls are accepted if the occupancy s is less than t_1 or greater than t_2, but not if it is between the two. The authors suggest an explanation for the result: when the cell is nearly empty ($s < t_1$) new calls may be admitted without fear of not having room for handoff calls. When it is a little full ($t_1 \leq s \leq t_2$) handoffs are expected and so some channels are reserved for handoff. When the cell is nearly completely occupied ($s > t_2$), it becomes less likely that the neighboring cells are occupied and hence there is a reduced handoff possibility and new calls are accepted again.

The GA was thus able to identify a novel policy structure. Based on a comparison of performances, the authors suggest that since the state space is considerably smaller for policies with $k = 0$ than $k = 1$ (30 bits vs. 3000 bits to represent the policy), and they offer nearly similar results, it is advisable to search for single-cell occupancy based policies.

3.6 Optimal Multi-user Detection in CDMA

The applications thus far presented are all geared towards high-level design of mobile communication networks. We now consider the difficult low-level optimization problems in mobile communications that have been tackled using global search techniques. One such problem is that of optimal multi-user detection in Code Division Multiple Access (CDMA) systems. Spread spectrum CDMA is a fairly new wide-band technique used to share the available bandwidth in a wireless communication system. In CDMA the message signal is multiplied by a large-bandwidth *spreading signal* - a pseudo-noise sequence with a chip rate much greater than the rate of the message. Each user is assigned a unique PN codeword that is nearly orthogonal to the other codewords. For the receiver to detect the message signal intended for the user, it must know the corresponding unique codeword used by the transmitter. In the single-user receiver the received waveform is input to a correlating matched filter and the signals of the other users are treated as noise (along with the noise due to the channel). But because the other codewords are still somewhat correlated with that of the receiving user's codeword, this receiver is not optimal in minimizing errors in detection. It has been shown that the optimal receiver would be one that detects the messages for all users simultaneously, i.e. one that performs simultaneous multiuser detection [41]. Formally, considering a synchronous CDMA channel with additive white Gaussian noise (AWGN) characteristics, given that the k^{th} user (of the K users sharing the channel) is assigned a codeword , the signal $r(t)$ at the receiver is:

$$r(t) = \sum_{k=1}^{K} b_k s_k(t) + \sigma n(t) \ , \ t \in [0, T] \tag{22}$$

where n(t) is a unit spectral density white Gaussian process and represents the transmitted bits in the message for the kth user. The output of a bank of matched filters that correlate this signal with the codewords for each user is given by:

$$y_k = \int_0^T r(t) \cdot s_k(t) dt \tag{23}$$

It has been shown that the output vector $\overrightarrow{y} = [y_1, y_2 \ldots y_K]$ thus obtained may be used to determine the transmitted message vector $\overrightarrow{b} = [b_1, b_2 \dots b_K]$ in an optimal manner [41]. If H is the nonnegative matrix of cross correlation between the assigned waveforms:

$$H_{i,j} = \int_0^T s_i(t) s_j(t) dt \tag{24}$$

then

$$\overrightarrow{y} = H \overrightarrow{b} + \overrightarrow{n} \tag{25}$$

(25) where $\overrightarrow{n}$ is a Gaussian K-vector with covariance matrix equal to $\sigma^2 H$. The optimal multiuser detection procedure determining the message that was most likely sent is to determine the vector $\overrightarrow{b}$ that maximizes the following expression:

$$f = 2 \overrightarrow{y}^T \overrightarrow{n} - \overrightarrow{b}^T H \overrightarrow{b} \tag{26}$$

This combinatorial optimization problem is known to be NP-hard, as the search space of the possible multi-user message increases exponentially with the number of users.

i) [42] describes the application of GA to optimal multiuser detection. It is assumed that the bits are being decoded sequentially (one bit for each user at any time) using the Viterbi algorithm. To speed up the search and reduce the computational complexity the authors suggest reducing the number of bits by utilizing the information given by the output of the matched filter bank. A confidence measure based on the probability of a bit error in single-user detection is used to determine if some of the bits may be considered as detected. The bits b_k are thus divided into two groups - the group of detected bits and the group of unknown bits. The search space is then restricted to finding values only for the unknown bits. In the Genetic Algorithm each chromosome consists of the number of bits that are still unknown. The standard mutation operation, uniform crossover and an elitist selection scheme were used. A total of 20 chromosomes are present in each generation and for each step of the detection process only 6 generations were performed (only 120 evaluations are performed in the optimization to detect the unknown bits).

To examine the proposed detector, Monte Carlo simulations were carried out comparing the GA-based detector with the linear decorrelating detector and the linear MMSE detector for $k = 20$ users. The authors show that the performance of the GA-based detection is very close to that of the MMSE detector and better than the decorrelating detector, and further that the proposed detector is actually computationally faster than these two linear detectors.

ii) A very brief paper [43] also describes the application of Genetic Algorithms to this problem. The authors reported using Monte Carlo simulations to study the performance of the genetic detector on a problem with 20 users and spreading sequences of length 31. The algorithms were compared for different types of initial guesses (random and those generated by sub-optimal detectors). They concluded that the algorithm was not robust when random initial guesses were used, but was able to improve the detection performance when combined with such sub-optimal detectors.

3.7 TDMA Frame Pattern Design

Besides CDMA, another multiple access scheme very commonly used in mobile communication systems is Time Division Multiple Access (TDMA). In TDMA the information for different users is carried in separate time slots in each frame of data. One of the goals of PCS is to be able to carry multimedia multi-rate information on the network. This is easily done in TDMA because users have the flexibility of determining different rates and error-control coding for traffic of different kinds of information- such as systems that carry both voice and data.

One way of integrating voice and data in TDMA is to divide the frame into fixed sub-frames for each. But this *frame pattern* has the significant disadvantage of not being able to adapt to traffic changes. A dynamic scheme that allows a data user to borrow voice slots when needed improves utilization. In schemes where a contention-based transmission technique is used for data, the positions of available data slots within a frame affect the performance of the system.

i) In [44] the problem of designing frame patterns to maximize throughput is considered and solved using a Simulated Annealing algorithm. It is assumed that each frame consists of N independent slots, N_t of which are assigned for data service (including nominal data slots and unassigned voice slots) while the rest are assigned for voice traffic. The frame pattern of the

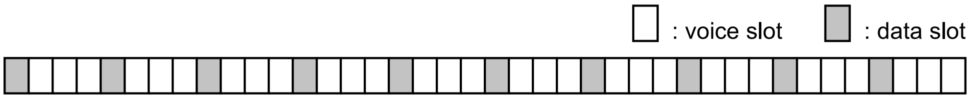

Figure 9: Optimal TDMA Frame Pattern (N = 40, Nt = 10, x = [4,4,4,4,4,4,4,4,4,4])

integrated TDMA system is defined as

$$\overrightarrow{x} = [x_1, x_2, ...x_{N_t}] \tag{27}$$

where $x_i \geq 1$ represent the number of slots between successive data slots. It is assumed that the slotted ALOHA random access protocol [7] is used for data transmission, that the queuing behavior of data for a given pattern is able to reach steady state and that data traffic is a Poisson process with mean arrival rate G packets/slot. For a given frame pattern, the mean data throughput is the function to be maximized. To perform minimization, the cost function may be taken as the negative of this throughput measure, and is therefore given by:

$$-\frac{G}{N_t}\sum_{i=1}^{N_t} x_i \cdot \exp(-G \cdot x_i) \tag{28}$$

The number of frame patterns for a given N_t and N can be very large indeed. For a TDMA frame with $N = 60$, and $Nt = 10$, there are as many as 7.5 x 1010 possibilities. Since a brute force technique is not possible in this huge search space, global search techniques are ideal for this NP-hard problem.

For the SA, the initial temperature T_o is chosen to be $10\sigma_\infty$ (the standard deviation of the cost of explored states when T is infinite). The neighborhood function is defined by a simple move (add 1 to one of the x_i, and subtract it from another randomly) and a complex move which is simply a succession of simple moves. A further adaptive scheme is used to determine the selection probabilities of the moves based on data collected during the execution of the algorithm. The standard metropolis criterion is used. The selected cooling schedule is similar to that in Equation (9), with an additional term ensuring a minimum rate of cooling. The algorithm is stopped when it is found that all the accessed states are of comparable costs at the given temperature.

The algorithm was tested for various loads G for a frame with $N = 40$ and N_t values of 5, 10 and 15. As anticipated, it was shown that there is

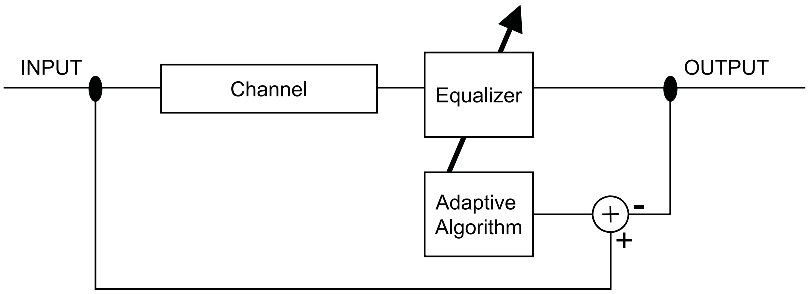

Figure 10: Adaptive Data Equalization

an optimal frame pattern for a given G and N_t but none for all G and N_t. Due to the restriction of integer solutions for the optimal frame pattern, it was found that the optimal frame pattern is insensitive to small changes in G and thus there are a finite number of total optimal frame patterns. The maximum throughput of 0.368 occurs at $G = Nt/N = 0.25$ when the frame pattern is uniformly distributed - each slot separated from the succeeding one by an inter-distance of 4 when $N_t = 10$ (shown in Figure 9). Comparing the results with random frame patterns, throughput gains ranging from 10% to 120% were observed by the authors.

3.8 Data Equalization

One major problem with digital communications channels is the distortion of data due to inter-symbol interference (ISI) [23]. The limited bandwidths and frequency selective fading (as observed in mobile radio environments) of these channels cause transmitted symbols to be smeared in time and interfere with adjoining bits. The effect of ISI can be reduced by *equalization*. An equalizing filter compensates for the characteristics of the channel and corrects the distortion, reducing errors in communication.

Most equalizers utilize filters where the output is non-recursively dependent upon present and previous values of the input and exhibit a finite impulse response (FIR). Recursive infinite impulse response (IIR) filters with feedback have outputs that depend not only on previous values of the input but also those of the output. Though susceptible to instability, IIR filters can perform better when the ISI extends over a long duration and typically require fewer weights (filter coefficients) to be adjusted to perform as well.

i) Figure 10 shows the block diagram of an adaptive equalization scheme. The problem of determining the filter coefficients of an adaptive IIR data equalizer is a difficult one. A GA-based algorithm for such equalization is presented in [45]. The chromosomes in the GA represent the filter coefficients (both feedback and feed-forward). The GA used is a simple binary encoded one. The cost function to be minimized is the mean square error (MSE) between the output of the equalizer and the input to the channel, which is known since a training data sequence is used to run the adaptive algorithm. The authors use a 6th order filter and test it on three different channel instances with varying characteristics. In these channels the bit-error rate due to ISI in the data transmitted without equalization ranged from 26% to 32%. After the adaptive equalization was performed, this error dropped to 0% in the first two cases, and 3% in the third case (which had channel noise added in besides the ISI distortion), demonstrating the effectiveness of the genetic adaptive algorithm.

3.9 Blind Channel Estimation

Adaptive channel equalizers utilize a training sequence at the start of transmission. But there are cases when the equalizers may have to start or restart without a training sequence - this is referred to as blind equalization. In the fading/multipath channels characteristic of mobile communications, such a restart may be required when there is temporary path loss due to a severe fading. Under such conditions blind channel identification schemes attempt to determine the coefficients of the channel from the statistical properties of the received signals, specifically higher order cumulants (HOC). This knowledge of the channel may then be used to design the equalizer. The application of GA and SA to the problem of blind channel identification using HOC fitting is presented in [46], [47], [48].

If the message transmitted is $s(k)$, the received signal $r(k)$ can be modeled as the result of ISI and noise as follows [46], [47] :

$$r(k) = \sum_{i=0}^{n_a} a_i s(k-i) + e(k) \tag{29}$$

The goal is then to determine the channel coefficients a_i (the channel is assumed to be of order n_a using only $r(k)$ and some knowledge of the statistical properties of $s(k)$. It has been shown in [48] that the problem can be expressed as the minimization of the following function:

$$f = J_4(a) = \sum_{\tau=-n_a}^{n_a} \left(C_{4,r}(\hat{\tau}, \tau, \tau) - \gamma_{4,s} \sum_{i=max(0,-\tau)}^{min(n_a, n_a-\tau)} a_i a_{i+\tau}^3 \right)^2 \tag{30}$$

where $C_{4,r}(\hat{\tau}, \tau, \tau)$ is the time estimate of the diagonal slice of the fourth-order cumulant for $r(k)$ and $\gamma_{4,s} = C_{4,s}(\tau_1, \tau_2, \tau_3)|_{\tau_1=\tau_2=\tau_3=0}$ is the *kurtosis* of $s(k)$.

i) In [46], the minimization is approached using a version of Genetic Algorithms called μGA - where the population size is much smaller than usual and the population is reinitialized randomly while retaining the best individual found up to that point after each run converges. The standard crossover operator is used, but no mutation operation is performed since the re-initialization procedure itself introduces diversity in the process. This process is repeated until no further improvement is seen. A 16-bit number between -1 and 1 represents each channel coefficient, and the encoding is such that these coefficients are arranged linearly in the chromosome.

The algorithm was tested on two channels with known coefficients. 50,000 noisy received data samples were used to compute the time estimate of the fourth-order cumulant. The final results were the average of 100 different runs where each run utilized a different sequence of received data samples and different initial populations. It was observed that the solutions always converged quickly to a value close to the global optimal channel estimate. The authors remark that the method is accurate and robust as demonstrated by the small standard deviations of the estimated channel coefficients over the different runs. They compare these results with results from a prior SA-based approach in terms of the number of function evaluations required for optimization. It is noted that the SA-based approach would have taken nearly six times as long for these problems. In a related paper ([48]) on GA-based blind channel estimation, third-order cumulant statistics are utilized instead and the simple Genetic Algorithm is used.

ii) The application of an adaptive Simulated Annealing (ASA) algorithm to blind channel identification with HOC fitting is described briefly in [47]. The function to be minimized is the same as given in Equation (30). The ASA algorithm, described in this paper, is a technique to speed up the convergence rate of Simulated Annealing by using estimates of the partial derivatives of the cost function (taken with respect to the solution coefficients) to modify the annealing rate every few moves. The algorithm was tested by applying it to a channel with known coefficients. As expected, ASA was observed to converge to the optimal solution much faster than the standard SA.

4 Conclusion

This chapter has surveyed the application of three global search techniques - Genetic Algorithms, Simulated Annealing, and Tabu Search – to difficult optimization problems in the subject of mobile communications. The papers described are all very recent, and they cover a broad spectrum of problems in mobile communications, from network planning and design issues to lower-level problems like detection of transmitted data and estimation of channel properties.

We have attempted to provide some detail on the implementations of the various techniques - this becomes a key issue with these algorithms as they are very general in their simplest form and there are a large number of parameters that may be selected to create different implementations. It must be observed, however, that these techniques appear to be quite robust and offer good results for a wide range of implementations as seen for example in the papers on channel assignment. There is, moreover, no empirical evidence of the inherent superiority of one of these techniques over the others for any given global optimization problem. In the few cases where the results of one technique are compared with another, the question of whether each technique was implemented to ensure the fastest convergence possible arises and remains unresolved because of the very nature of these algorithms. Researchers would do well not to ignore the possibilities inherent in each of these techniques and make their decisions on which technique to use based on at least a cursory comparison of their efficiency and ease of implementation.

The speed of convergence becomes important when these techniques are applied to time-critical problems where fast, online solutions are required. So far they have not been applied to such problems in wireless telecommunications because of the large number of function evaluations that are often required to find optimal or near-optimal solutions using these techniques. One of the trends of future research in this area would be the extension of the use of these algorithms to such optimization problems as well. In time, more problems in mobile communications may be identified where such techniques outperform existing schemes. It is already certain that the early success of these approaches will lead to their integration into the design process for future PCS networks.

References

[1] V.O.K. Li and X. Qiu, "Personal Communication Systems (PCS)," *Proceedings of the IEEE*, vol. 83, no. 9, pp. 1210-43, September, 1995.

[2] J.D. Gibson, Ed., *The Mobile Communications Handbook*, CRC Press, 1996.

[3] C. Muller, E.H. Magill, P. Prosser, and D.G. Smith, "Artificial Intelligence in Telecommunications," *IEEE Global Telecommunications Conference*, GLOBECOM '93, vol. 2, pp. 883-887, Nov. 1993.

[4] M.R. Garey and D.S. Johnson, *Computers and Intractability: a guide to the theory of NP-completeness*, W. H. Freeman & Company, 1979.

[5] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.

[6] S. Kirkpatrick, C.D. Gelatt jr., and M.P. Vecchi, "Optimization by Simulated Annealing," *Science*, no. 220, pp. 671-680, 1983.

[7] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 1992.

[8] G. Rudolph, "Convergence Analysis of Canonical Genetic Algorithms," *IEEE Transactions on Neural Networks*, vol. 5, no. 1, Jan. 1994.

[9] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, 1997.

[10] D. Goldberg, *Genetic Algorithms in search, optimization and learning*, Addison Wesley, 1989.

[11] T. Bck, D.B. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*, IOP Publishing and Oxford University Press, 1997.

[12] E.H.L. Aarts and J.H.M. Korst, *Simulated Annealing and Boltzmann Machines*, Wiley, 1989.

[13] F. Glover, "Tabu Search - Part I," *ORSA Journal on Computing*, vol. 1, 190-206, 1989.

[14] F. Glover, "Tabu Search - Part II," *ORSA Journal on Computing*, vol. 2, 4-32, 1989.

[15] F. Glover and M. Laguna, *Tabu Search*, Kluwer Academic Publishers, 1997.

[16] T. S. Rappaport, *Wireless Communications - Principles & Practice*, Prentice Hall, 1996.

[17] E. Aarts and J. K. Lenstra, *Local search in Combinatorial Optimization*, Wiley, 1997.

[18] K.T. Ko et al., "Using Genetic Algorithms to design mesh networks," *Computer*, vol.30, no.8, pp. 56-61, Aug. 1997.

[19] S. Pierre and A. Elgibaoui, "Improving communication network topologies using Tabu Search," *22nd Annual Conference on Local Computer Networks (LCN'97)*, pp. 44-53, 1997.

[20] G. Celli, E. Costamagna, and A. Fanni, "Genetic Algorithms for telecommunication network optimization," *IEEE International Conference on Systems, Man and Cybernetics, Intelligent Systems for the 21st Century*, pp. 1227-1232, 1995.

[21] E. Costamagna, A. Fanni, and G. Giacinto, "A Simulated Annealing algorithm for the optimization of communication networks," *URSI International Symposium on Signals, Systems, and Eletctronics – ISSSE '95*, pp. 40-8, 1995.

[22] M. Shahbaz, "Fixed network design of cellular mobile communication networks using Genetic Algorithms," *Fourth IEEE International Conference on Universal Personal Communications*, pp. 163-7, 1995.

[23] J. G. Proakis, *Digital Communications*, McGraw-Hill, 1995.

[24] I. Katzela and M. Nahshineh, "Channel assignment schemes for cellular mobile telecommunication systems: a comprehensive survey," *IEEE Personal Communications*, vol. 3, no. 3, pp. 10-31, June 1996.

[25] P. Raymond, "Performance analysis of cellular networks," *IEEE Transactions on Communications*, vol. 39, no. 3, pp. 542-548, 1994.

[26] W.K. Lai and G.C. Coghill, "Channel assignment through evolutionary optimization," *IEEE Transactions on Vehicular Technology*, vol. 45, no. 1, pp. 91 -96, Feb. 1996.

[27] F.J. Jaimes-Romero, D. Munoz-Rodriguez, and S. Tekinay, "Channel assignment in cellular systems using Genetic Algorithms," *IEEE 46th Vehicular Technology Conference*, vol. 2, 741-5, 1996.

[28] C.Y. Ngo, V.O.K. Li, "Fixed channel assignment in cellular radio networks using a modified Genetic Algorithm," *IEEE Transactions on Vehicular Technology*, vol. 47, no. 1, pp. 163-72, Feb. 1998.

[29] N. Funabiki and Y. Takefuji, "A neural network parallel algorithm for channel assignment problems in cellular radio networks," *IEEE Transactions on Vehicular Technology*, vol. 41, no. 4, pp. 430-437, 1992.

[30] M. Duque-Anton, D. Kunz, B. Ruber, "Channel Assignment for Cellular Radio Using Simulated Annealing," *IEEE Trans on Vehicular Technology*, vol. 42, no. 1, pp. 14-21, 1993

[31] K. Min-Jeong, "A comparison of two search algorithms for solving the channel assignment problem," *5th IEEE International Conference on Universal Personal Communications*, vol. 2, pp. 681-5, 1996.

[32] Hao Jin-Kao; R. Dorne, P. Galinier, "Tabu search for frequency assignment in mobile radio networks," *Journal of Heuristics*, vol. 4, no. 1, pp. 47-62, June 1998.

[33] A. Ortega et al., "Algorithmes de Coloration des Graph et d'Affectation des Frquences, Approche Gomtrique et Analytiques, Heuristiques, Recuit Simul, Programmation par Contraintes, Algorithme Gntiques," *Technical Report NT/PAB/SRM/RRm/4353*, CNET, Paris, 1995.

[34] P. Calegari et al., "Genetic approach to radio network optimization for mobile systems," *IEEE 47th Vehicular Technology Conference*, vol. 2, pp. 755-9, 1997.

[35] W.K. Hale, "Frequency Assignment: Theory and Applications," *Proceedings of the IEEE*, vol. 68, no. 12, pp. 1497-1514, Dec. 1980.

[36] H.R. Anderson, and J.P. McGeehan, "Optimizing Microcell Base Station Locations Using Simulated Annealing Techniques," *IEEE 44th Vehicular Technology Conference*, vol. 2, pp. 858-862, 1994.

[37] V. Erceg et al., "Urban/suburban out-of-sight propagation modeling," *IEEE Communications Magazine*, pp. 56-61, June 1992.

[38] T.P. Wang, S.Y. Hwang, and C.C. Tseng, "Registration Area Planning for PCS Networks Using Genetic Algorithms," *IEEE Transactions on Vehicular Technology*, vol. 47, no. 3, pp. 987-994, Aug. 1998.

[39] S. Junping, and H. C. Lee, "Optimal mobile location tracking by multi-layered model strategy," *Third IEEE International Conference on Engineering of Complex Computer Systems*, pp. 86-95, 1997.

[40] A. Yener, and C. Rose, "Genetic algorithms applied to cellular call admission: local policies," *IEEE Transactions on Vehicular Technology*, vol. 46, no. 1, pp. 72-9, Feb. 1997.

[41] S. Verdu, *Multiuser Detection*, Cambridge University Press, 1998.

[42] X.F. Wang, W. S. Lu, and A. Antoniou, "A Genetic Algorithm-based multiuser detector for multiple-access communications" *IEEE International Symposium on Circuits and System - ISCAS'98*, vol. 4, pp. 534-7, 1998.

[43] M.J. Juntti, T. Schlosser, and J. O. Lilleberg, "Genetic algorithms for multiuser detection in synchronous CDMA," *IEEE International Symposium on Information Theory*, pp. 492, 1997.

[44] C. J. Chang, C. H. Wu, "Optimal frame pattern design for a TDMA mobile communication system using a Simulated Annealing algorithm," *IEEE Transactions on Vehicular Technology*, vol. 42, no. 2, pp. 205-11, May 1993.

[45] M. S. White, S. J. Flockton, "A genetic adaptive algorithm for data equalization," *Proceedings of the First IEEE Conference on Evolutionary Computation*, vol. 2, pp. 665-669, 1994.

[46] S. Chen, Y. Wu, and S. McLaughlin, "Genetic algorithm optimization for blind channel identification with higher order cumulant fitting," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 4, Nov. 1997.

[47] S. Chen, B.L. Luk, Y. Liu, "Application of adaptive Simulated Annealing to blind channel identification with HOC fitting," *Electronics Letters*, vol. 34, no. 3, pp. 234-5, Feb. 1998.

[48] M.A. Alkanhal, S.A. Alshebeili, "Blind estimation of digital communication channels using cumulants and Genetic Algorithms", *International Conference on Consumer Electronics*, pp. 386-387, 1997.

[49] D.G. Luenberger, *Introduction to linear and nonlinear programming*, Addison-Wesley, 1973.

[50] W. Sacco, *Dynamic Programming : an elegant problem solver*, Janson Publications, 1987.

[51] G.L. Nemhauser, L.A.Wolsey, *Integer and Combinatorial Optimization*, Wiley, 1988.